

JAERI-M

6 9 5 4

ANISN-JR, A ONE-DIMENSIONAL DISCRETE
ORDINATES CODE FOR NEUTRON AND
GAMMA-RAY TRANSPORT CALCULATIONS

February 1977

K. KOYAMA, Y. TAJI, K. MINAMI, T. TSUTSUI,
T. IDETA and S. MIYASAKA

日本原子力研究所
Japan Atomic Energy Research Institute

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問合せは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

ANISN-JR, A One-Dimensional Discrete Ordinates Code
for Neutron and Gamma-Ray Transport Calculations

Kinji KOYAMA, Yukichi TAJI, Kazuyoshi MINAMI*,
Tuneo TSUTSUI, Takashi IDETA and Shun-ichi MIYASAKA

Division of Reactor Engineering, Tokai, JAERI

(Received January 29, 1977)

The ANISN code available from RSIC is designed to solve the one-dimensional Boltzmann equation for deep penetration problems taking into consideration the anisotropic scattering by Legendre expansion of the scattering cross sections.

To extend its applicability for shielding analyses, the code has been modified by adding options of calculating the reaction rates distributions from detector response, generating the volume-flux weighted cross sections in arbitrary regions or zones and plotting the neutron or gamma-ray spectra and the reaction rates distributions.

The formats of input data necessary in the options are described in detail.

*) Fujitsu Ltd.

中性子・ガンマ線輸送1次元Snコード, ANISN-JR

日本原子力研究所・東海研究所・原子炉工学部

小山謹二・田次邑吉・南 多善*

筒井恒夫・出田隆士・宮坂駿一

(1977年1月29日受理)

1次元輸送計算コードANISNは、散乱の異方性を高次のルジャンドル展開頃まで扱うことができるが、さらに遮蔽計算への適応性を高め、使い易くするために、ANISNに1) 検出器のレスポンスから反応率の空間分布を計算する、2) 中性子およびガンマ線のエネルギー・スペクトルの空間依存性を考慮し、任意の領域の平均断面積を作成する、そして3) エネルギー・スペクトル、反応率の空間分布の作図、などに代表される機能を追加した。この報告はANISN-JRのユーザー・マニュアルとして種々のオプションを使用するために必要な入力データと、その定義をまとめたものである。

*）富士通株式会社

CONTENTS

1. Introduction	1
2. Additional Data Description	2
3. Limitation for the Additional Data	6
4. Data Description for ANISN-PLOT	8
Appendix A Data Description for ANISN	12
Appendix B Detailed Data Notes	17
Appendix C Free-Form FIDO Format	25
Appendix D Sample Input List for ANISN-JR	31
Appendix E Sample Input List for ANISN-PLOT	32
Appendix F Program List of ANISN-PLOT	33

1. Introduction

The one-dimensional transport calculation code ANISN¹⁾ currently available from RSIC (Reactor Shielding Information Center, Oak Ridge National Laboratory) is widely used as a standard shielding code. However, the analyses of shield experiments, require, various quantities such as energy spectra, reaction rates, heat generation, radiation damage, and biological dose. These quantities are obtained by multiplying the fluxes in the shield by response function of detectors. In addition, for heterogeneous zone consisting of media of thicknesses of more than one mean free path, cross sections are sometimes required to be averaged with the flux distribution over the zone for effective shield calculations.

In order to make these treatments possible and to increase utility of ANISN, some optional functions are added to the code as follows:

- (1) print the total fluxes at boundary points of all mesh intervals.
(The original ANISN prints the total fluxes at midpoints only.)
- (2) calculate, print and plot the spectrum of ϕ_g , $\phi_g/\Delta E_g$ or $\phi_g/\Delta U_g$ (lethargy width).
- (3) print the angular fluxes at only required mesh boundaries or midpoints (maximum 10 points). The original ANISN prints at midpoints of all meshes, and therefore the number of print pages becomes vast according to the number of the spatial and angular meshes.
- (4) use the asymmetric quadrature set.
- (5) calculate and plot the reaction rates for neutron and gamma-ray detectors, and collapse the response functions of detectors.
- (6) generate volume-flux weighted cross sections for arbitrary zone or region. (In the original ANISN, the cross sections can be collapsed only for a homogeneous zone or region.)
- (7) collapse into few group cross sections in ANISN, DOT, or TWOTRAN format. (In TWOTRAN format, the l -th Legendre coefficient of the scattering cross section is divided by $(2l + 1)$ and the cross section of $(n,2n)$ reaction is added for use of the coarse-mesh rebalance technique.)
- (8) multiply the average cross section by the density factor, when an option of density factors are used (IDFM = 1).

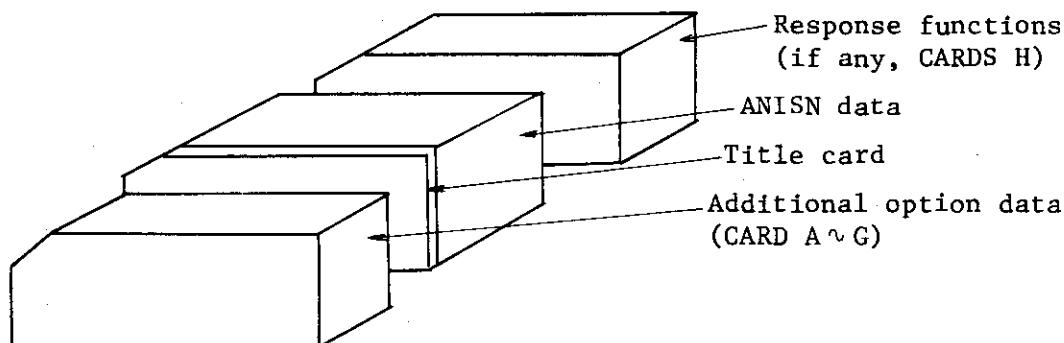
- (Note-1) In original ANISN and the present version, the adjoint calculations can not be performed with the group independent cross section tape ($ID2 = 1$) but with the tape generated from the step-three of the RADHEAT-V3.³⁾
- (Note-2) To plot the energy spectrum and reaction rates distributions, a utility code of ANISN Plot is used.

The ANISN code with the above mentioned new options is named as ANISN-JR.

In the sections 2 and 3 of the present report, additional input instructions for the ANISN-JR are described in detail with its limitation. The data description for the ANISN-PLOT is described in the section 4. For convenience for users, the data description and detailed data notes of the original ANISN are quoted from the reference 1) in appendixes A and B. In addition, the free FIDO format used in the program is cited from the reference 2) in appendix C.

2. Additional Data Description

The data for the additional options are given before the ANISN original input data. If the reaction rates are required, the response functions of detectors follow after the ANISN data as shown below.



(1) CARD A Format (2I5)

1. IANISN the format of group independent cross section tape.
 - 0 — same as the original
 - 1 — for the tape produced by RADHEAT-V3
2. ITMAX maximum execution time in minutes. If ITMAX = 0, this option is ignored.

ANISN checks this time at the end of each outer iteration and, if it has been exceeded, the problem is terminated with full output.

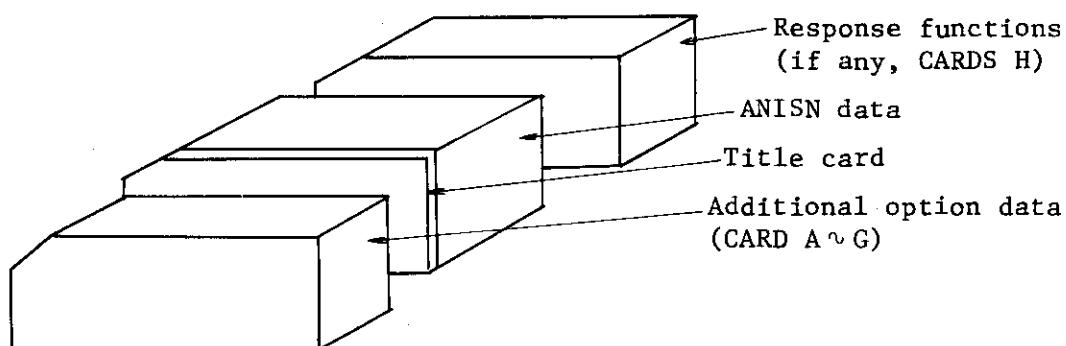
- (Note-1) In original ANISN and the present version, the adjoint calculations can not be performed with the group independent cross section tape ($ID2 = 1$) but with the tape generated from the step-three of the RADHEAT-V3.³⁾
- (Note-2) To plot the energy spectrum and reaction rates distributions, a utility code of ANISN Plot is used.

The ANISN code with the above mentioned new options is named as ANISN-JR.

In the sections 2 and 3 of the present report, additional input instructions for the ANISN-JR are described in detail with its limitation. The data description for the ANISN-PLOT is described in the section 4. For convenience for users, the data description and detailed data notes of the original ANISN are quoted from the reference 1) in appendixes A and B. In addition, the free FIDO format used in the program is cited from the reference 2) in appendix C.

2. Additional Data Description

The data for the additional options are given before the ANISN original input data. If the reaction rates are required, the response functions of detectors follow after the ANISN data as shown below.



(1) CARD A Format (2I5)

1. IANISN the format of group independent cross section tape.
 - 0 — same as the original
 - 1 — for the tape produced by RADHEAT-V3
2. ITMAX maximum execution time in minutes. If ITMAX = 0, this option is ignored.

ANISN checks this time at the end of each outer iteration and, if it has been exceeded, the problem is terminated with full output.

(2) CARD B Format (10I5)

1. IIBOUD 0 — print total fluxes at midpoint of each mesh interval
 (same as original)
 1 — print total fluxes at boundary of each mesh interval
2. IISPTM 0 — no effect
 1 — calculate and print neutron spectra $\phi_g/\Delta E_g$ at each mesh
 2 — calculate and print neutron spectra $\phi_g/\Delta U_g$ at each mesh
 Energy group boundaries are specified by CARDS C and D.
3. IIANLL 0 — print angular fluxes at all meshes (same as original)
 1 — print angular fluxes at only required meshes
 The numbers of required meshes are given by CARDS E and F.
 If angular fluxes are required, ID1 = 1 must be entered in 15\$ array.
4. IGMNEW number of total energy groups (equal to IGM in 15\$ array)
5. IGMNEU number of neutron groups. If neutron only, IGMNEU=IGMNEW.
6. NACTPR 0 — no effect
 1 — print the activities for each group and zone
7. NREACT 0 — no effect
 1 — calculate and print reaction rates for neutrons and gamma rays
 2 — calculate reaction rates and also collapse response functions of detectors
 3 — collapse and punch response functions of detectors
 The response functions for neutrons and gamma rays are given by CARDS H. For collapsing the response functions, IFG = 1 in 15\$ array, and 27\$ and 28\$ arrays must be entered.
8. NASYMM 0 — no effect
 1 — use asymmetric quadrature set.
 The angular quadrature weights and cosines are given by 6* and 7* arrays. In the case NASYMM = 1, the reflective condition for left boundary can not be used.
9. NRESAT 0 — no effect
 1 — write/read final fluxes on a tape for use as an initial flux guess for the next run
 Final fluxes are written on the logical unit No.15 and read from No.14. For the first run, NRESAT=1, IFN=1 or 2 (in 15\$ array) must be specified, and for the following, NRESAT=1 and IFN = 3.

10. NXOUT 0 — no effect
 1 — obtain the few group cross sections for DOT, TWOTRAN or MORSE
 When NXOUT = 1, CARD G is necessary.
11. NPLOT 0 — no effect
 1 — obtain the file for plotting
- (3) CARD C Format (I5) (input if IISPTM=1)
 1. NOYGRE number of energies to be specified in CARDS D
 NOYGRE = IGM + 1, for neutron or gamma-ray problem
 NOYGRE = IGM + 2, for coupled neutron and gamma-ray problem
- (4) CARD D Format (8E10.5) (input if IISPTM=1)
 (YGREENE(I), I=1, NOYGRE)
 Energy group boundaries are given in descending order of energy in eV.
- (5) CARD E Format (I5) (input if IIANLL=1)
 1. NOANNO number of mesh boundaries to print angular fluxes (≤ 10)
- (6) CARD F Format (10I5) (input if IIANLL=1)
 (NOANLL(I), I=1, NOANNO)
 Mesh boundaries numbers to print angular fluxes.
- (7) CARD G Format (6I5) (input if NXOUT=1)
 1. NACT number of detectors for which the response functions are to be stored.
 2. IDOT 0 — calculate collapsed cross sections in TWOTRAN format
 1 — calculate collapsed cross sections in DOT format
 3. NPU output of cross section tables
 0 — only print
 1 — punch
 2 — write on a tape (logical unit No.40)
 3 — add cross sections to the tape produced in previous cases.
 4. IGNT number of groups of $(n, 2n)$ down-scattering. Normally, total number of neutron groups is entered. When IDOT = 1 (DOT format), IGNT is put to zero.

5. IGMN number of energy groups to be obtained (from highest energy).
 6. ITL2 length of the output cross section table.
 Specify ITL2 = NACT + IGMN + 3 for DOT and ITL2 = NACT + IGMN + 5 for TWOTRAN.

(8) CARDS H (input if NREACT#0)

CARDS H give the names of detectors and their response functions after the data for the first case of ANISN inputs. When the neutron and gamma-ray responses are calculated, CARDS H must be given respectively, that is, at first the neutron's data and then the gamma-ray's data.

CARD H-1 Format (I5)

1. NELM number of detectors for which the response functions for neutrons or gamma-rays are stored.

CARD H-2 Format (I5, 4A4) (input if NELM#0)

1. NAME identification number for the detector
 2. DXCM title of the detector

CARDS H-3 Format (8E10.3) (input if NELM#0)

(OSIG(I), I=1, NG)

Response function of the detector in descending order of energy.

NG is the number of neutron groups or gamma-ray groups.

CARD H-2 and CARDS H-3 are repeated by NELM times.

(9) Data for the cell weighting for arbitrary zones

This option is specified in ICON and IPUN of 27\$ array of ANISN data. Cell numbers to be weighted are given in 29\$ array added newly to the ANISN input.

27\$ Few group parameters [5] (IFG=1)

1. ICON 0 --- no effect
 1 --- not changed
 2 --- not changed

3 — macroscopic cross section desired (cell-weighted for
arbitrary zones)
2. IHTF not changed
3. IHSF not changed
4. IHMF not changed
5. IPUN 0 — no effect
1 — not changed
2 — obtain weighted cross sections on tape F11.

28\$ Not changed

29\$ Cell numbers for each zone. This number becomes material number
for cell weighted cross sections. [IZM]
(IFG=1, ICON=3)

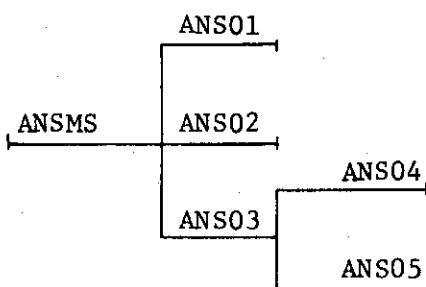
3. Limitations for the Additional Data

- (1) For options of IIBOUD=1 (print total flux at the boundary point) and IISPTM=1 (print spectrum), IDAT1#2 (fluxes in core) must be entered in 15\$ array.
Dimensional limitation : $IGM*(IM+1) \leq 30000$.
- (2) For option of IIANLL=1 (print angular fluxes), maximum number of intervals to be printed is 10.
- (3) Detailed print of activities.
Dimensional limitation: $IGM*ID3*(IM+IZM+1) \leq 30000$, where ID3 is number of activities in 15\$ array.
- (4) For calculations of reaction rates, maximum number of detectors, energy groups and mesh intervals are respectively 25, 100 and 400.
- (5) Output of the weighted cross sections in DOT or TWOTRAN format.
Dimensional limitation: $(IHMF+ITL2)*IGMF \leq 1040$.

The input/output files assigned by the ANISN-JR are tabulated in Table 1 except the F05 and F06 logical units for the cards input and output list respectively. A diagram of the ANISN-JR overlay structure and assignment of subroutines for each segment is shown in Fig. 1. A sample input is shown in appendix D.

Table 1. Input/output file assignment

Logical Unit No.	Comment
F01	Scratch file
F02	Scratch file (use if IDAT1=1 or 2)
F03	Scratch file (use if IDAT1=2)
F04	Library tape
F08	Scratch file (use if IDAT1=2)
F09	Scratch file (use if NXOUT=1)
F11	Output of the weighted cross section (use if IPUN=2)
F14	Input of the restart file (use if NRESAT=1)
F15	Output of the restart file (use if NRESAT=1)
F20	Scratch file (use if IIBOUD=1)
F30	Output of the plotting data (use if NPLOT=1)
F40	Output of the DOT or TWOTRAN cross section (use if NPLOT=1)



Segment		Subroutine				
ANSMS	FTMAIN ITIME	ANISN CLOCK	CÖNTRL ADDR	ERRÖ CLEAR	WOT	SPIE
ANS01	PLSNT S805	FIDÖ S814	FFREAD WOT8	TP S966	ADJNT	S804
ANS02	GUTS ST	S807 CELL	S810 S851	S821	S824	S833
ANS03	FINPR NWSUB2	FINPRL ERRMSG	PUNSH ACTPRT	DTFPUN WOTYT	FLTFX	NWSUBL
ANS04	BT	SUMARY	FACTR	SUBNW3	SUBNW4	
ANS05	FEWG	WATE	CÖNVT	CRATE		

Fig. 1 Simplified structure of the ANISN-JR

$\underline{3}$ — macroscopic cross section desired (cell-weighted for arbitrary zones)
 2. IHTF not changed
 3. IHSF not changed
 4. IHMF not changed
 5. IPUN 0 — no effect
 1 — not changed
 $\underline{2}$ — obtain weighted cross sections on tape F11.

28\$ Not changed

29\$ Cell numbers for each zone. This number becomes material number for cell weighted cross sections. [IZM]
 (IFG=1, ICON=3)

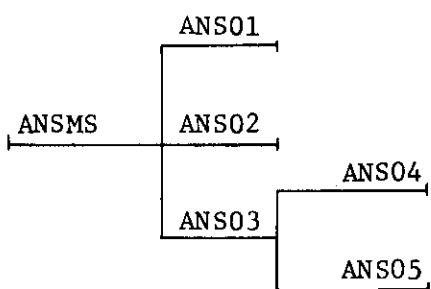
3. Limitations for the Additional Data

- (1) For options of IIBOUD=1 (print total flux at the boundary point) and IISPTM=1 (print spectrum), IDAT1#2 (fluxes in core) must be entered in 15\$ array.
 Dimensional limitation : $IGM*(IM+1) \leq 30000$.
- (2) For option of IIANLL=1 (print angular fluxes), maximum number of intervals to be printed is 10.
- (3) Detailed print of activities.
 Dimensional limitation: $IGM*ID3*(IM+IZM+1) \leq 30000$, where ID3 is number of activities in 15\$ array.
- (4) For calculations of reaction rates, maximum number of detectors, energy groups and mesh intervals are respectively 25, 100 and 400.
- (5) Output of the weighted cross sections in DOT or TWOTRAN format.
 Dimensional limitation: $(IHMF+ITL2)*IGMF \leq 1040$.

The input/output files assigned by the ANISN-JR are tabulated in Table 1 except the F05 and F06 logical units for the cards input and output list respectively. A diagram of the ANISN-JR overlay structure and assignment of subroutines for each segment is shown in Fig. 1. A sample input is shown in appendix D.

Table 1. Input/output file assignment

Logical Unit No.	Comment
F01	Scratch file
F02	Scratch file (use if IDAT1=1 or 2)
F03	Scratch file (use if IDAT1=2)
F04	Library tape
F08	Scratch file (use if IDAT1=2)
F09	Scratch file (use if NXOUT=1)
F11	Output of the weighted cross section (use if IPUN=2)
F14	Input of the restart file (use if NRESAT=1)
F15	Output of the restart file (use if NRESAT=1)
F20	Scratch file (use if IIBOUD=1)
F30	Output of the plotting data (use if NPLOT=1)
F40	Output of the DOT or TWOTRAN cross section (use if NPLOT=1)



Segment		Subroutine				
ANSMS	FTMAIN ITIME	ANISN CLOCK	CONTRL ADDR	ERR0 CLEAR	WOT	SPIE
ANS01	PLSNT S805	F1D0 S814	FFREAD WOT8	TP S966	ADJNT	S804
ANS02	GUTS ST	S807 CELL	S810 S851	S821	S824	S833
ANS03	FINPR NWSUB2	FINPRI ERRMSG	PUNSH ACTPRT	DTFPUN WOTYT	FLTFX	NWSUB1
ANS04	BT	SUMARY	FACTR	SUBNW3	SUBNW4	
ANS05	FEWG	WATE	CONVT	CRATE		

Fig. 1 Simplified structure of the ANISN-JR

4. Data Description for ANISN-PLOT

As a supporting package for ANISN-JR, a graph plotting program ANISN-PLOT has been developed to plot the results; 1) neutron and gamma-ray energy spectrum, 2) flux distribution of neutron and gamma-ray for an arbitrary energy group, and 3) distribution of reaction rates calculated from the detector response.

These data are transferred from the logical unit F30 of ANISN-JR shown in previous section. A sample input and the program list of ANISN-PLOT is given in appendixes E and F, respectively.

A. Title and parameters

CARD A-1 Format (18A4)

1	(TITLE(I), I=1, 18)
2	(TITLE(I), I=19, 36)

title of this problem.

CARD A-2 Format (E10.6)

1 XNF	normalization factor, each value for Y axis is devided by this.
-------	---

CARD A-3 Format (6I5)

1 IF1	=0, no effect, =1, plot the neutron energy spectrum,
2 IF2	=0, no effect, =1, plot the gamma-ray energy spectrum,
3 IF3	=0, no effect, =1, plot the neutron flux distribution calculated at midpoints of mesh intervals (IIBOUD=0, and need CARDs D), =2, plot the neutron flux distribution calculated at mesh boundary points (IIBOUD=1, and need CARDs D).
4 IF4	=0, no effect, =1, plot the gamma-ray flux distribution calculated at midpoints of mesh intervals (IIBOUD=0, and need CARDs E),

=2, plot the gamma-ray flux distribution calculated at mesh boundary points (IIBOUD=1, and need CARDS E).

5 IF5 =0, no effect,
 =1, plot the neutron reaction rate distribution
 calculated at mispoints of mesh intervals
 (IIBOUD=0, NREACT#0, and need CARDs F),
 =2, plot the neutron reaction rate distribution
 calculated at mesh boundary points (IIBOUD=1,
 NREACT#0, and need CARDs F).

6 IF6 =0, no effect,
 =1, plot the gamma-ray reaction rate distribution
 calculated at midpoints of mesh intervals
 (IIBOUD=0, NREACT#0, and need CARDs G),
 =2, plot the gamma-ray reaction rate distribution
 calculated at mesh boundary points (IIBOUD=1,
 NREACT#0, and need CARDs G).

B. Plotting the neutron energy spectrum (input if IF1=1)

CARD B-1 Format (215)

1 I1 ≥ 1 , neutron energy group at upper energy boundary
 on X axis,
 2 I2 ≤ 100 , neutron energy group at lower energy
 boundary on X axis (must be I1 I2).

CARD B-2 Format (1615)

1 N \leq 20, number of spectra to be plotted,
2 (JK(I), I=1, N),
 spatial mesh numbers to be plotted.

CARD B-3 Format (15, 17A4)

1 IXN \leq 17, number of words to be written as the caption
 of X-axis (4 characters/word),
2 (IX(I), I=1, IXN)
 caption of X-axis.

CARD B-4 Format (T5, 17A4)

1 ZYN \leq 17, number of words to be written as the caption
of Y-axis.

2 (IY(I), I=1, IYN)
caption of Y-axis.

C. Plotting gamma-ray energy spectrum (input if IF2=1)
same as CARDs B series (the highest energy group of gamma-ray
must be counted as one).

D. Plotting the neutron flux distribution (input if IF3 \geq 1)

CARD D-1 Format (2I5)

1 I1 \geq 1, mesh number corresponding to the left boundary
 of X-axis,
2 I2 \leq 400, mesh number corresponding to the right boundary
 of X-axis.

CARD D-2 Format (16I5)

1 N \leq 20, number of distributions to be plotted
2 (JK(I), I=1, N)
neutron energy group numbers to be plotted.

CARD D-3 Format (I5, 17A4)

1 IXN \leq 17, number of words to be written as the caption
 of X-axis,
2 (IX(I), I=1, 17)
caption of X-axis.

CARD D-4 Format (I5, 17A4)

1 IYN \leq 17, number of words to be written as the caption
 of Y-axis,
2 (IY(I), I=1, 17)
caption of Y-axis.

E. Plotting the gamma-ray flux distribution (input if IF4 \geq 1)

same as CARDs D series.

F. Plotting the neutron reaction rate distribution (input if IF5 \geq 1)

CARD F-1 Format (2I5)

1 I1 \geq 1, mesh number corresponding to the left boundary
 of X-axis,

2 I2 ≤ 400 , mesh number corresponding to the right boundary of X-axis.

CARD F-2 Format (16I5)

1 N ≤ 20 number of reaction rate distributions to be plotted,

2 (JK(I), I=1, N)
identification numbers of the detector responses
(NAME in CARD H-2 of ANISN-JR)

CARD F-3 Format (I5, 17A4)

1 IXN ≤ 17 , number of words to be written as the caption of X-axis,

2 (IX(I), I=1, IXN)
caption of X-axis.

CARD F-4 Format (I5, 17A4)

1 IYN ≤ 17 , number of words to be written as the caption of Y-axis,

2 (IY(I), I=1, IYN)
caption of Y-axis.

G. Plotting the gamma-ray reaction rate distribution (input if IF6 ≥ 1)

same as CARDs F series.

References

- 1) Ward W. Engle, Jr.: "A USERS MANUAL FOR ANISN: A One Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering", K-1693, (1967).
- 2) Green N.M. et al.: "XLACS: A Program to Produce Weighted Multigroup Neutron Cross Section from ENDF/B", ORNL-TM-3646(AMPX-2), (1972).
- 3) Koyama K. et al.: "RADHEAT-V3: A Code System for Analyzing the Radiation Transport in a Nuclear Reactor and Shield", to be publish.

Appendix A Data Description for ANISN

This section is intended to be used as a guide in preparing problems for ANISN. The appendix B presents a more detailed description of the data. The quantity in brackets is the array dimension and the expression in braces is the condition requiring that array or set of arrays. Arrays or sets of arrays which are not required should not be entered. If no condition is specified the array is required. Note that a T must follow each of the five sets of arrays if that set is entered.

- A. LIMI card - format (6X, I6) This card contains the number of locations available for ANISN data. (not used in ANISN-JR)
- B. Title card - format (12A6)
- C. Parameters

15\$ Integer parameters [36]

1. ID problem ID number
2. ITH 0 - forward solution
1 - adjoint solution
3. ISCT maximum order of scatter found in any zone
4. ISN order of angular quadrature
5. IGE 1 - slab; 2 - cylinder; 3 - sphere
6. IBL left boundary condition
0 - vacuum (no reflection)
1 - reflection
2 - periodic
3 - white/albedo
7. IBR right boundary condition, same options as IBL
8. IZM number of zones or regions
9. IM number of mesh intervals
10. IEVT eigenvalue type
0 - fixed source
1 - k calculation
2 - α calculation
3 - concentration search
4 - zone width search
5 - outer radius search
6 - buckling search
11. IGM number of energy groups

12. IHT position of σ_{total} in cross section table
13. IHS position of σ_{gg} (self-scatter) in cross section table
14. IHM length of cross section table
15. MS cross section mixing table length (10\$, 11\$, 12*)
16. MCR number of cross section sets to be read from cards (14*)
17. MTP number of cross section sets to be read from tape (13\$)
18. MT total number of cross section sets (elements + mixtures)
19. IDFM 0 - density factors (21*) not used
1 - density factors used
20. IPVT 0 - no effect
1 - enter k_0 as PV (16*)
2 - enter α_0 as PV
21. IQM 0 - no effect
1 - enter distributed source (17*)
22. IPM 0 - no effect
1 - enter shell source by group and angle (18*)
IM - enter shell source by interval, group, and angle
23. IPP interval number which contains shell source if IPM = 1;
0 otherwise
24. IIM inner iteration maximum
25. ID1 0 - no effect
1 - print angular flux
2 - punch scalar flux
3 - both 1 and 2
26. ID2 0 - no effect
1 - use specially prepared group independent cross
section tape (contains MTP materials)
2 - use cross sections and fixed source from previous
problem
27. ID3 0 - no effect
N - compute N activities by zone where N is any
positive integer
28. ID4 0 - no effect
1 - compute N activities by interval where N refers
to ID3
29. ICM outer iteration maximum

30. IDAT1 0 - all data in core
 1 - cross sections and fixed sources stored on tape
 2 - fluxes and currents on tape also
31. IDAT2 0 - no effect
 1 - execute diffusion solution for specified groups
 (24\$)
32. IFG 0 - no effect
 1 - flux weight cross sections (27\$ and 28\$)
33. IFLU 0 - step model used when linear extrapolation yields
 negative flux (mixed mode)
 1 - use linear model only
 2 - use step model only
34. IFN 0 - enter fission guess (2*)
 1 - enter flux guess (3*)
 2 - use fluxes from previous case
35. IPRT 0 - print cross sections
 1 - do not print cross sections
36. IXTR 0 - calculate P_L scattering constants (Legendre
 coefficients)
 1 - read P_L constants from cards (34*)

16* Floating point parameters [14]

1. EV first guess for eigenvalue
 2. EVM eigenvalue modifier
 3. EPS epsilon - accuracy desired
 4. BF buckling factor, normally 1.420892
 5. DY cylinder or plane height for buckling correction
 6. DZ plane depth for buckling correction
 7. DFM1 transverse dimension for void streaming correction
 8. XNF normalization factor
 9. PV 0.0, k_0 , or α_0 according to IPVT = 0, 1, or 2
 10. RYF λ_2 relaxation factor, normally 0.5
 11. XLAL point flux convergence criterion if entered greater
 than zero
 12. XLAH upper limit for $|1.0 - \lambda_1|$ used in linear search
 13. EQL eigenvalue change epsilon
 14. XNPM new parameter modifier
- NOTE: The above data is followed by a T.

D. Cross Sections {ID2 = 0}

- 13\$ Library ID numbers [MTP] {MTP > 0}
 14* Cross sections [MCR × IGM × IHM] {MCR > 0}

NOTE: If entered, the above data is followed by a T.

E. Fixed Source {IEVT = 0 and ID2 < 2}

- 17* Distributed source [IGM × IM] {IQM = 1}
 18* Shell source [IGM IPM × MM] {IPM > 0}

NOTE: If entered, the above data is followed by a T.

F. Flux or Fission Guess {IFN < 2}

- 2* Fission density [IM] {IFN = 0}
 3* Flux guess [IGM × IM] {IFN = 1}

NOTE: If entered, the above data is followed by a T.

G. Remainder of Data

- 1* Fission spectrum [IGM]
 4* Radii by interval boundary [IM + 1]
 5* Velocities [IGM]
 6* Angular quadrature weights [MM]^{a)}
 7* Angular quadrature cosines [MM]
 8\$ Zone numbers by interval [IM]
 9\$ Material numbers by zone [IZM]
 10\$ Mixture numbers in mixing table [MS] {MS > 0}
 11\$ Component numbers in mixing table [MS] {MS > 0}
 12* Number densities in mixing table [MS] {MS > 0}
 19\$ Order of scatter by zone [IZM] {ISCT > 0}
 20* Radius modifiers by zone [IZM] {IEVT = 4}
 21* Density factors by interval [IM] {IDFM = 1}
 22\$ Material numbers for activities [ID3] {ID3 > 0}
 23\$ Cross section table position for activities [ID3] {ID3 > 0}
 24\$ Diffusion calculation markers [IGM] {IDAT2 = 1}
 25* Albedo by group - right boundary [IGM] {IBR = 3}
 26* Albedo by group - left boundary [IGM] {IBL = 3}

a) MM = ISN + 1 for plane or shpere
 MM = (ISN × (ISN + 4))/4 for cylinder

27\$ Few group parameters [5] {IFG = 1}

1. ICON 0 - no effect
 1 - micro cross sections desired
 2 - macro cross sections desired (minus implies cell weighting)
 3 - macro cross section desired (cell-weighting for arbitrary zones)
2. IHTF position of σ_{total} in weighted cross section
3. IHSF position of $\sigma_{g \rightarrow g}$ in weighted cross sections (minus implies upscatter removal)
4. IHMF table length of weighted cross sections
5. IPUN 0 - no effect
 1 - punch weighted cross sections
 2 - obtain weighted cross section on tape F11

28\$ Few group number for each multigroup [IGM] {IFG = 1}

29\$ Cell numbers for each zone, which become the numbers for cell weighted cross sections [IZM] (IFG = 1, ICON = 3)

34* PL scatter constants [JT × MM]^{b)} {IXTR = 1}

NOTE: The above data is followed by a T.

b) JT = ISCT for plane or sphere
 JT = (ISCT × (ISCT + 4))/4 for cylinder
 NOTE: JT is truncated to the next lower integer for cylinders when ISCT is odd.

Appendix B Detailed Data Notes

This section presents a more detailed definition of selected parameters and arrays. Some comments derived from experience are also included.

A. Boundary conditions - IBL, IBR, 25*, 26*

The white boundary condition causes the entering flux at the specified boundary to be isotropic. This is accomplished by summing the flux leaving the system and returning an average flux in all directions. The albedo for each group specifies the fraction of the flux leaving to be returned. If the albedo is not specified it is assumed to be 1.0. The white boundary is recommended for the outer boundary of spherical and cylindrical cells.

B. Cross section mixing table - MS, 10\$, 11\$, 12*

The cross section mixing table is used to combine elements into macroscopic mixtures and to specify the method of the concentration search. Experience will reveal that only the imagination limits its flexibility. The following table illustrates the three types of operations performed by the mixing table.

	<u>10\$</u>	<u>11\$</u>	<u>12*</u>
1.	M	O	X
2.	M	N	X
3.	M	M	0.0

1. Multiply all cross sections in material M by X.
2. Multiply all cross sections in material N by X and add to corresponding cross sections in material M.
3. Multiply all cross sections in material M by EV, the eigenvalue (concentration search)

C. Cross sections - 13\$, 14*

ANISN expects a table of cross sections for each group, g, of each material in the following format:

<u>Position</u>	<u>Cross section type</u>
1	activity
.	"
.	"
IHT-2	absorption
IHT-1	nu x fission
IHT	total
IHT+1	$\sigma_{g+NUS \rightarrow g}$
.	.
.	.
IHS-1	$\sigma_{g+l \rightarrow g}$
IHS	$\sigma_{g \rightarrow g}$
IHS+1	$\sigma_{g-1 \rightarrow g}$
.	.
.	.
IHM	$\sigma_{g-NDS \rightarrow g}$

upscatter^{c)}downscatter^{c)}

Thus the parameters IHT, IHS, and IHM completely describe the format of the cross sections. If there are no activity cross sections, IHT = 3. If there is no upscatter IHS = IHT + 1. If there is no downscatter IHM = IHS (i.e. a one group problem). If there is upscatter ANISN will compute a total upscatter cross section for each group of each material and place that cross section in position IHS + 1. The activity cross sections are used only for activities (22\$, 23\$).

The P_L cross section tables must correspond in format to the P_0 tables even though the transfer coefficients are the only numbers used. Note that the P_L cross sections must contain a $(2L + 1)$ term. Previous S_n codes supplied this term internally. (e.g. DTF-II multiplied the P_1 cross sections by 3.0) This factor may be included externally or internally via the mixing table.

c) NUS is the number of groups of upscatter.
NDS is the number of groups of downscatter.

D. Material numbers - 10\$, 11\$, 9\$, 22\$

All cross section sets, whether elements or mixtures, are referred to by a continuous set of material numbers. In particular, the materials supplied in card form (14*) become materials 1 through MCR the materials read from a library tape become MCR + 1 through MCR + MTP, and any number greater than MCR + MTP but less than or equal to MT refers to a mixture.

When the order of scatter for any zone (19\$) is greater than zero, ANISN expects the P_1 cross sections to be material $M + 1$, the P_2 cross sections to be $M + 2$, etc. where M is the P_0 material number specified in the 9\$ array.

E. Density factors - IDFM, 21*

All cross sections appropriate to an interval are multiplied by the density factor for that interval. Thus one may easily and efficiently describe a void or a density variation by interval.

F. IPVT and PV - used in search

If IPVT = 1, ANISN will search for the parameter which results in a multiplication factor of PV. If IPVT = 2, ANISN will search for the parameter which results in a multiplication factor of 1.0 when $\alpha = PV$. If IPVT = 0, ANISN will search for a multiplication factor of 1.0 with $\alpha = 0.0$.

G. Distributed source - IQM, 17*

The distributed source is entered by group and interval as follows: group 1, interval 1 through IM; group 2, etc.

H. Shell source - IPM, IPP, 18*

If IPM = 1, the shell source is entered by group and angle for interval IPP as follows: group 1, angle 1 through angle MM; group 2 etc.

If IPM = IM, the shell source is entered by group, interval and angle as follows: group 1, interval 1, angle 1 through angle MM; interval 2, etc.

I. Special cross section tape - ID2 = 1

A special purpose program is available which will prepare a group

independent cross section tape for ANISN. This tape is required if the complete input cross section matrix [(MCR + MTP) \times IGM \times IHM] is larger than the number of data locations available.

J. Activities - ID3, ID4, 22\$, 23\$

Activities may be computed by zone and interval as specified in ID3 and ID4. The zone activity is a total reaction rate and the interval activity is per unit volume. The following table illustrates the use of activity specifications.

	<u>22\$</u>	<u>23\$</u>
1.	1	3
2.	-5	1
3.	7	-1
4.	-3	-1

1. Compute activity for material 1, cross section position 3 in the intervals and/or zones in which material 1 appears.
2. Compute activity for material 5, cross section position 1 in all intervals and/or zones.
3. Computer activity for material 7, position 1 in appropriate intervals and/or zones and multiply interval activities by 1.0, $2\pi r$, or $4\pi r^2$ for slab, cylinder, or sphere respectively.
4. Compute activity for material 3, position 1 in all intervals and/or zones and multiply interval activities by geometry factor.

K. Auxiliary tape storage - IDAT1

If IDAT1 is specified as zero, ANISN will use the most efficient tape storage possible and modify IDAT1 accordingly. IDAT1 may be specified as 1 or 2 if cross section and/or flux tapes are available from a previous problem or if ID2 = 1.

L. Diffusion theory solution - IDAT2, 24\$

If IDAT2 = 1 the 24\$ array must be entered. A zero implies a transport solution and a one implies a diffusion solution for the corresponding group. If IFN = 0, ANISN will use diffusion theory on the first outer iteration.

M. Weighted cross sections = IFG, 27\$, 28\$

When microscopic weighted cross sections are requested (ICON = 1), a set of cross sections is produced for each component of each material in each zone. When macroscopic cross sections are requested (ICON = 2), a set of cross sections is produced for each material in each zone. The cross sections are weighted by the flux or current in the zone in which the material appears. Since the mixing table is used to determine the components of a material, MS should not be zero when ICON = 1.

If the cross section structure specified for the weighted cross sections will not accommodate the complete multigroup scattering matrix, the "extra" transfer coefficients are placed such that they transfer as far down (or up) as possible.

If complete removal of the upscatter is desired, IHSF should be minus. |IHSF| should be the position of the self scatter cross section before the upscatter is removed. IHMF should be the final table length. After the upscatter is removed, IHSF will be IHTF + 1. The upscatter is removed by subtracting the reaction rate due to $\sigma_{j \rightarrow i}$ from the reaction rate due to $\sigma_{i \rightarrow j}$ where $j > i$. Thus the net transfer rate between groups j and i is preserved.

N. Starting guess - IFN, 2*, 3*

If IFN is specified as zero, ANISN will execute a diffusion solution for the first outer iteration. Since this is undesirable for fixed source calculations where one normally desires a zero flux guess one may set IFN = 1 and enter no guess. Simply enter a card with a T in column three for that section of data.

P. EV and EVM guesses

<u>Ievt</u>	<u>EV</u>	<u>EVM</u>
0	0.0	0.0
1	0.0	0.0
2	best guess for $\alpha^d)$	0.0
3	1.0	-0.1
4	0.0	-0.1
5	outer radius	-(10% of outer radius)
6	1.0	-0.1

d) Zero is the best guess unless one is reasonably sure that his guess is close to the answer.

When IEVT = 0 there is no eigenvalue (EV).

When IEVT = 1 the multiplication factor (k) is the eigenvalue.

When IEVT = 2 α is the eigenvalue.

When IEVT = 3 the eigenvalue is defined by its use in the mixing table.

When IEVT = 4 the eigenvalue is used as follows:

$$\Delta R_I = \Delta R_I^0 (1.0 + EV \times RM_Z)$$

where ΔR_I^0 is the initial ΔR

RM_Z is the radius modifier (20*)

When IEVT = 5 the outer radius is the eigenvalue.

When IEVT = 6, $EV = DY/DY^0 = DZ/DZ^0$

where DY^0 and DZ^0 are input.

R. Convergence - EPS, XLAL, RYF

The inner or flux iterations are considered converged when both the integral self-scatter error and the integral removal error are less than EPG or when the maximum flux deviation is less than EPS. EPG is related to EPS by a normalization factor, the total source divided by IGM. Since the integral tests are sometimes easily satisfied, a point flux convergence may be specified. If XLAL is greater than zero, the inner iterations are not considered converged until the maximum flux deviation is less than XLAL.

The outer or power iteration is considered converged when the total source ratio between successive iterations differs from 1.0 by less than EPS, the total scatter ratio differs from 1.0 by less than EPS/RYF and the upscatter ratio differs from 1.0 by less than EPS/RYF.

S. Buckling correction - BF, DY, DZ

ANISN computes a correction factor of the DB^2 term for finite transverse dimensions. The correction is applicable only with "transport corrected" P_0 cross sections where position IHT is occupied by the transport cross section.

T. Void streaming correction - DFML

Since the DB^2 term is not applicable to a void region, ANISN computes a simple correction^{e)} which effectively removes the transverse

e) Olsen, T.: "Void Streaming in SN calculations," Nucl. Sci. Eng., 21, 271 (1965).

component of each angular flux in the void region. This correction term is not included in the calculation of the absorption reaction rate as are the DB² losses. This omission causes the neutron balance to differ from 1.0. If DFML is zero, no correction is computed for the void regions.

U. Normalization

When IEVT is greater than zero the total fission source is normalized to XNF. When IEVT is equal to zero the total fixed source is normalized to XNF and the fission source, if any, is unnormalized. If XNF = 0.0 there is no normalization.

V. Searches - XLAH, EQL, XNPM

When the absolute value of the difference between two successive lambdas (λ_1) is less than EQL, the eigenvalue, EV, is changed. The first EV change is the result of adding or subtracting the eigenvalue modifier, EVM. The second EV change is the result of a linear extrapolation. To prevent large changes early in the calculation, the absolute value of the difference between 1.0 and λ_1 is not allowed to exceed XLAH. To prevent oscillations when using the linear search, the extrapolation is limited by SNPM. The third EV change is the result of the quadratic search. The quadratic search is used until the absolute value of $1.0 - \lambda_1$ is less than EQL. At this point, the linear search is used to complete the problem. XLAH is normally 0.05 and XNPM is normally 0.75. EQL should be the larger of 0.001 and three times EPS. In cases where EPS is quite small, EQL may be less.

W. Multiple cases

The ANISN data arrays are stored in core in the order in which they are numbered. For example the fission density (2*) follows the fission spectrum (1*). No data is destroyed between cases. If problem dimensions change, the repositioned arrays are simply read into core over the previous data. The result is that all arrays following and including the first array to be repositioned must be respecified. It should be noted that the 15\$ and 16* parameter arrays are exceptions to the above discussion and are never destroyed. Multiple cases in which IDAT1 changes will not retain data properly. If IDAT1 =

1, multiple cases will retain data properly only if ID2 = 2 in all cases following case 1. If IDAT1 = 2, multiple cases will retain data properly only if ID2 = 2 and IFN = 2 in all cases following case 1. If there is upscatter, multiple cases will retain data properly only if ID2 = 2 or if the complete cross section matrix is read in all cases.

In cases where the data in a particular section remains the same in multiple cases, one may enter a card containing only a T in any of the six appropriate columns for that section of data (e.g. the 15\$, 16* section).

If any case is preceded by an adjoint solution, the following arrays must be respecified if they are required for the next case: 14*, 17*, 18*, 3*, 1*, 5*, 24\$, 25*, 26*.

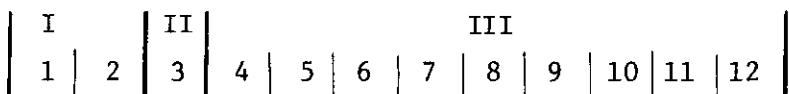
If multiple cases are completely independent of each other (i.e. all data is specified in each case) the data of each case may be terminated with a T in the third column of two successive fields on the same card. This "double T" is used in lieu of the normal single T. If ANISN terminates a case for any reason, the code will search for the "double T" and attempt to execute the following problem. This option may also be used for independent sets of multiple cases.

Appendix C Free-Form FIDO Format

The input to ANISN-JR is given in the ANISN format, and hence it is able to use the free-form FIDO format improved by Ward Engle of ORNL. In this section, the definitions of the format are described by quoting reference 2).

C.1 Type 1 Format

Each card is divided up into six 12-digit data fields which are in turn divided up into 3 subfields, illustrated in the following figure. Only one data field is shown.



The first subfield is a two-digit integer; the second subfield contains either a \$, *, R, I, T, S, F, A, C, E, Q, L, N, M, O, U, V, Z, +, -, or a blank. The third subfield contains either a fixed or floating point number. The contents of the first two subfields will define the operation to be performed on the third field.

Blank fields are ignored. One can use any or all fields on a card. For example, a box of blank cards sandwiched anywhere in a data array would be completely ignored.

Each data array is identified by a two-digit integer in a first subfield. There are both fixed and floating point arrays; a fixed point array is designated by a \$ in the second subfield, a floating point array by an *.

The second subfield contains an operator which specifies the type of operation to be performed on the data. The possible operators are listed below.

Array Operators

\$ indicates the beginning of an integer array. The first subfield

contains a one- or two-digit number identifying the array.

* indicates the beginning of a floating point array. The first subfield identifies the array.

R indicates that the entry in the third subfield is to be repeated the number of times specified in the first subfield.

I indicates linear interpolation between the entry in the third subfield and the entry in the third subfield of the next data field. The number in the first subfield gives the number of points to be placed equally spaced in the specified range.

T indicates termination of data reading for a block. XLACS can require several data blocks and each block must be ended with a T. A block can contain any number of arrays. Data on a card after a T will be ignored.

S indicates skip. The first subfield defines the number of entries to be skipped. The third field can contain the first entry following the skips. A blank third subfield would be ignored.

F is used to fill the remainder of an array with the item given in the third subfield.

A is used to address a particular location in an array. This location is specified in the third subfield; the first subfield is blank.

C is used to obtain a count of the number of items read into an array up to the point where the C is placed. An integer ZZ in front of the C will be used as identification in producing a message as follows:

XX ENTRIES READ IN THE YY ARRAY at ZZC.

E may be used to end specifying data for an array. This option is particularly useful when it is desired to replace only some items in a particular array. The items in question are replaced, and the use of an E prevents having to count and skip to the end of the array.

Q is used to repeat sequences of numbers. The length of the sequence is defined in the third subfield. The number of times to repeat the sequence is given in the first subfield.

L is similar to I except that a logarithmic interpolation is performed between the entry points. This option is particularly useful for defining

energy structures equally spaced in lethargy.

N is used to repeat a sequence of numbers in reverse order. The length of the sequence is defined in the third subfield. The number of times to repeat a sequence is given in the first subfield.

M is used to negate and repeat an inverted sequence. The length of the sequence is given in the third subfield. The number of times to repeat a sequence is given in the first subfield.

O is used to turn on (or off) the card image edit of ANISN input data. As with the C option, an integer in front of the O identifies the particular entry. The default (starting) condition is not to edit the data.

U is used to replace the ANISN input format for an array. The array number is given in the first subfield. The format, written in normal FORTRAN, is specified on the card immediately following the card containing a U. The parentheses normally capsulating a format should be included.

V specifies that the array identified in the first subfield will be read according to the last variable format read in.

Z is used to specify a string of zeroes; e.g., 49Z would place forty-nine zeros into an array.

+ or - indicates exponentiation. The data in the third field is multiplied by $10^{\pm N}$, where N is an integer in the first subfield. This option allows one to specify a number in up to nine significant digits.

Integer data in the third subfield must be right adjusted. Floating point data may be written with or without an exponent. If the decimal is omitted, it is assumed to be immediately to the left of the exponent field. If there is no exponent, the decimal point is assumed to be to the extreme right of the nine-column subfield.

Input Restrictions

The following restrictions must be observed when using the ANISN input format:

- (1) Blank data fields are ignored.
- (2) If the interpolation option (I) is used, the next data field may not be either blank or an A entry.

(3) The third subfield of a data field containing a \$ or an * may contain an integer, N. The next data entry is assumed to be the $(N+1)^{th}$ member of the array. Normally the third subfield is blank and is ignored.

(4) All data arrays must be filled with the correct number of entries. A data array is ended by either starting a new data array or by ending a data block.

The sample data sheet which follows illustrates possible uses of many of the ANISN input options. This is shown for illustrative purposes only and should not be construed to represent meaningful data.

C.2 Type 2 Format (Free Form)

The transferral of input data to input forms or punched cards for a code requiring significant amounts of input is always a time consuming, distasteful and error-prone process. The original ANISN formats were designed to help reduce these difficulties. The options are convenience features. The usefulness of the "F" option which fills an array is obvious, but it is somewhat harder to see the practical uses for some of the more obscure ones like N, M, and Q; however, frequent use will turn up situations where these options are invaluable. For example, the S_n cosines are negated and reflected about 90° , a fact which suggests the use of the M option.

There are justifiable complaints with the input formats; for example, where convenience options are not applicable, data can be hard to write because of the manner in which the data fields are spread on the card. This is especially true of integer arrays, where the data are right adjusted in 12-column fields. The ANISN input forms help to some extent, but the actual keypunching is still troublesome.

The input format has been greatly improved by Ward Engle of ORNL who has designed and implemented an all-FORTRAN free-form, ANISN input scheme which has data items separated by blanks (as others do), but still allows all of the important convenience features of the earlier formats. The restrictions on the use of this input are essentially that the user write the data in a form that he can interpret within the context of the ANISN options. Data is easily written and keypunched, since there is no worry about which type character falls in which column or how many blanks are left between entries.

The free-form input can be interspersed with the fixed form input. To select free-form, an array is identified as either a \$\$ or a ** array, for integer and floating point arrays, respectively.

The restrictions are:

(1) Any third subfield (data entry) must be followed by one or more blanks. This is an obvious restriction, otherwise data interpretation would be impossible.

(2) Only columns 1-72 are used.

(3) Numbers with exponents must not have imbedded blanks; e.g., use 1.0E+4, not 1.0 E+4 or 1.0E+ 4.

(4) The old + or - options (2nd subfield) are not operational.

Significance requirements which led to the development of this option can be had directly.

(5) No more than 9 digits in a number can be entered. The exponent is not counted; e.g., 9234+09, 923400000+1 will work, 9234000000 will not work. Nine-digit accuracy is clearly beyond the significance available for single precision IBM 360 floating point operations.

(6) A blank must not appear between items which fall in the first and second subfields with the old format, e.g., 24R, not 24 R. Note that the 99 restriction on the number of repeats, interpolations, etc., has been eliminated.

(7) The Z-entry must be entered as 738Z, not Z738. The old format allowed either.

(8) The Q, M, N entries must be specified as Q4, not 4Q. The old format allows either. An entry like 3Q4 accomplishes the same as Q4 Q4 Q4. This is now true for either format.

The character ')' in column 1 of a card will cause the contents of the card to be listed as comments, while the data is read in. Column 2 should contain the proper carriage control character; e.g., blank,o,1,2, etc. This card is ignored as a data card. This option is also available with the old formats.

Some examples of the new format are given below:

1\$\$25R1_0_4_3Q3_2\$\$_3R42_E_T

The first 25 entries of the 1\$ array are 1's followed by 0 and 4 and then the sequence 1 0 4 is repeated three times. The 2\$ array has three 42's and then data input to the array ends. The T terminates a data block.

```
42**_0.0_0.1666667_0.3333333_N2  
43**_-1.0_-0.8819171_0.3333333 M2
```

This example puts 0.0, 0.1666667, 0.3333333, 0.333333, 0.1666667 in the 42* array and -1.0, -0.8819171, -0.3333333, 0.3333333, 0.8819171 in the 43* array.

Appendix D Sample Input List of ANISN-JR

.....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

```

*HFORT
CCCCCC ANISN-JR MAIN PROGRAM CCCCCCCCCCCCCC
COMMON/BULKBU/ D(1),LIM1,DUMY(50000)
C
C      SET BLANK COMMON SIZE
LIM1=50000
CALL ANISN
STOP
END
*HLIED RFNAME=J9103,ANISNJR,SIMPL=OVLY,EDIT=YES,RBSPC=100,DIRCT=50
SGMT ANSMS
SELECT (FTMAIN,CLEAR,CONTRL,ERRO,WOT,SPIE,ITIME,CLOCK,ADDR,ANISN)
SGMT ANS01,CHN=ANSMS
SELECT (PLSNT,FIDO,TP,ADJNT,S804,S805,S814,WOT8,S966,FFREAD)
SGMT ANS02,CHN=ANSMS
SELECT (GUTS,S807,S810,S821,S824,S833,DT,CELL,S851)
SGMT ANS03,CHN=ANSMS
SELECT (FINPR,FINPR1,PUNSH,DTFPUN,FLTFX,NWSUB1,NWSUB2,ERRMSG,ACTPRT,
WOTYT)
SGMT ANS04,CHN=ANS03
SELECT (BT,SUMARY,FACTOR,NWSUB3,NWSUB4)
SGMT ANS05,CHN=ANS03
SELECT (FEWG,WATE,CONVT,CRATE)
FIN
*HRUN      SIZE=20
*DISK      F01,TRK=300
*DISK      F02,TRK=300
*DISKTO    F04,J9103,REG10
*DISK      F11
*DATA
   1   0
   0   1   1   34   26   0   0   0   0   0   0
  36
  1.5 E+07  1.05E+07  6.5 E+06  4.0 E+06  2.5 E+06  1.4 E+06  0.8 E+06  0.4 E+06
  2.0 E+05  1.0 E+05  4.65E+04  2.15E+04  1.0 E+04  4.65E+03  2.15E+03  1.0 E+03
  4.65E+02  2.15E+02  1.0 E+02  4.65E+01  2.15E+01  1.0 E+01  4.65   2.15
  1.0   0.465   1.0 E-03   1.0 E+07   8.0 E+06   6.0 E+06   4.0 E+06   2.5 E+06
  1.2 E+06  5.0 E+05  1.0 E+05  1.0 E+04
   4
   7   22   37   52
.....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

```

SAMPLE INPUT NO.1 FOR ANISN-JR																				
15**	100	0	3	4	3	1	0	5	60	0	34	5	6	39	0	0	4	4	1	0
16**	1.05	0.01	1.0-4	1.421	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.5	5.0-4			
13**	4R10	T																		
17**	5R9,5824-4	55R0,0	5R1,6925-2	55R0,0	5R8,4287-2	55R0,0	5R1,7297-1	55R0,0	5R2,6260-1	55R0,0	5R2,0558-1	55R0,0	5R1,4845-1	55R0,0	5R6,6319-2	55R0,0				
	5R2,6377-2	55R0,0	5R1,0467-2	55R0,0	5R3,4358-3	55R0,0	5R1,0915-3	55R0,0	5R3,4860-4	55R0,0	5R1,1132-4	55R0,0	5R3,4905-5	55R0,0	5R1,1081-5	55R0,0				
	5R3,5287-6	55R0,0	5R1,1050-6	55R0,0	5R3,5064-7	55R0,0	5R1,1161-7	55R0,0	F0,0	T										
3**	F0,0	T																		
1**	F0,0																			
4**	5R1	0,0	60,0																	
5**	F1,0																			
6**	0,0	0,1666667	0,3333333	0,3333333	0,1666667															
7**	-0,9367418	-0,8688903	-0,3500212	0,3500212	0,8688903															
8**	5R1	10R2	15R3	15R4	15R5															
9**	5R1																			
19**	5R3																			
21**	5R0,7	55R1,0																		
27**	3	5	6	39	2															
28**	2R1	3R2	4R3	4R4	5R5	2R6	5R7	8	4R9	2R10	2R11									
29**	2R20	30	2R40																	
	T																			

Appendix E Sample Input List of ANISN-PLOT

```

*DFORT SFNAME=J1026,SANISNPL
*DLMEDRUN GRFO=ON,COMLIB=CALL
*GCOM35
*DISKTO F01,J9196,FCACASE2
*DATA
FBR EXPERIMENTS CASE NO.2
***** B1 *****
1.0
   1   1
   1 100
11   1   17   21   25   26   34   39   42   48   54   67
 9 NEUTRON ENERGY (EV)
 9 NEUTRON SPECTRUM
   1 20
11   1   17   21   25   26   34   39   42   48   54   67
 9 GAMMA-RAY ENERGY (EV)
 9 GAMMA SPECTRUM

```

Appendix F Program List of ANISN-PLOT

```

C      READ(1) ((XNEU(1,J),J=1,IM1),I=1,IGNEU)
C      C      16 FORMAT(8H *XNEU*/(1P10E12,3))
C      C      300 IF(IGGAM,EQ,0) GO TO 400
C      C      READ(1) ((ANGAM(1,J),J=1,IM1),I=1,IGGAM)
C      C      400 READ(1) NNELM
C      C      WRITE(6,18) NNELM
C      C      18 FORMAT(8H *NNELM*/10)
C      C      IMP=IM
C      C      IF(NNELM,EQ,1) IMP=IMP+1
C      C      IF(NNELM,EQ,0) GO TO 600
C      C      DO 500 I=1,NNELM
C      C      500 IF(IGEMLM,LT,0) GO TO 100
C      C      READ(1) ((CM(N,I),N=1,18)
C      C      100 IF(IGEMLM,LT,0) GO TO 100
C      C      WRITE(6,22) CM(N,I),N=1,18)
C      C      READ(1) ((AFLXN(1,N),N=1,IMP)
C      C      100 WRITE(6,19) AFLXN(1,N),N=1,IMP
C      C      19 FORMAT(8H *AFLXN*/(1P10E12,3))
C      C      600 READ(1) NGELM
C      C      WRITE(6,20) NGELM
C      C      20 FORMAT(8H *NGELM*/10)
C      C      IF(NGELM,EG,0) GO TO 800
C      C      DO 700 I=1,NGELM
C      C      700 IF(NGELM,LT,0) GO TO 700
C      C      READ(1) ((CMG(N,I),N=1,18)
C      C      800 WRITE(6,22) CMG(N,I),N=1,18)
C      C      22 FORMAT(1X,18A4)
C      C      READ(1) ((AFLXG(1,N),N=1,IMP)
C      C      800 READ(1) ((CMG(N,I),N=1,18)
C      C      900 WRITE(6,22) CMG(N,I),N=1,18)
C      C      900 CONTINUE
C      C      DO 2000 J=1,IM1
C      C      2000 DO 2000 I=1,IGNEU
C      C      2000 IF(XNEU(I,J),LT,0) XNEU(I,J)=ABSCXNEU(I,J)
C      C      2000 CONTINUE
C      C      DO 3000 I=1,IGGAM
C      C      3000 IF(ANGAM(I,J),LT,0) ANGAM(I,J)=ABSCXGAM(I,J)
C      C      3000 CONTINUE
C      C      3000 FORMAT(8H *R*/(1P10E12,3))
C      C      14 FORMAT(8H *(MA(I),I=1,IM1)
C      C      15 FORMAT(8H *MA*/(10I10))
C      C      IF(IGNEU,EQ,0) GO TO 300
C      C

```

```

00 CONTINUE
DO 40000 J=1,IMP
DO 50000 I=1,NELM
IF (AFLXN(I,J) .LT. 0) AFLXN(I,J)=ABS(AFLXN(I,J))
CONTINUE
DO 60000 I=1,NELM
IF (AFLXG(I,J) .LT. 0) AFLXG(I,J)=ABS(AFLXG(I,J))
CONTINUE
CALL GRAPH (ANISN,VENEU,VEGAN,R,MA,CM,CMG,XNNEU,XNGAM,AFLXN,
AFIXG)
CALL PLOT(0,0,999)
REWIND 1
CALL PLOT(WX+120,-14,0,-3)
STOP
END

00 CONTINUE
CALL GRAPH (ANISN,VENEU,VEGAN,R,MA,CM,CMG,XNNEU,XNGAM,AFLXN,
AFIXG)
CALL PLOT(WX+120,-14,0,-3)
REWIND 1
CALL PLOT(0,0,3)
RETURN
END

SUBROUTINE CLOSE(WX)
CALL PLOT(WX+120,-14,0,-3)
RETURN
END

SUBROUTINE FIXNUM(X,XMIN,XMAX,WX,XDEC)
DIMEN(X(20))
XMIN=INT(XMIN)/10**10
XMAX=(INT(XMAX)/10)*10+10
WRITE(6,100) XMIN,XMAX,WX
100 FORMAT(1X,5E20.5)
NDEC=10
XDEC=XMAX-XMIN)/WX
2 NLINE=INT(XMAX-XMIN)/NDEC+1
IF (NDEC/XDEC,GT,10.0) GO TO 3
NDEC=NDEC+10
GO TO 2
3 XNUM=MIN
DO 1 I=1,LINE
XX=FLOAT(NDEC*(I-1))/XDEC
XXX=XX**2
CALL PLOT(X,0,3)
CALL PLOT(X,5,0,2)
CALL NUMBER(XXX,*7.0,2,8,XNUM,0,0,-1)
CONTINUE
CALL PLOT(WX,0,3)
RETURN
END

SUBROUTINE GRAPH(ANISN,VENEU,VEGAN,R,MA,CM,CMG,XNNEU,XNGAM,AFLXN,
AFLXG)
1
C
DIMENSION ANISN(36),VENEU(101),VEGAN(101),XNNEU(50),XNGAM(50,400),
1 XNNEU(100,400),XNGAM(50,400)
2 CM18_20,AFLXN(20,000,CMG18_20),AFLXG(20,400)
DIMENSION X(*,01),Y100_150,YT(400,5),JK(20)
DIMENSION IX(18),LY(18)
COMMON /TITLE/IX,N,IY,N,IV,XMIN,XMAX,YM(N,YMAX
COMMON /PPPP/WX,WY
COMMON /JJJJ/JJ,JKMAX
COMMON /OPT/IF1,IF2,IF3,IF5,IF6,IBOUND
REAL/NORM
C
2000 READ(5,9,END=1000) ANISN
FORMAT(1.8A4)
READ(5,10) NORM
FORMAT(8E10.0)
10 FORMAT(6E0) NORM
WRITE(6,E0) NORM
20 FORMAT(1X,7H *NORM*,1PE12.3)
READ(5,11) IF1,IF2,IF3,IF4,IF5,IF6
FORMAT(10.5)
11 FORMAT(1F11.11) [F1,1*IF2*IF3*IF4*IF5*IF6
WRITE(6,21) [F1*IF2*IF3*IF4*IF5*IF6]
21 FORMAT(1X,17H [F1,1*IF2,..,1*IF6*,6]10)
C
1 IF (IF1,E0,0) GO TO 100
FORMAT(11)
C
CALL TITLE(1,1,(2,JK,JKMAX)
MAX=12-1
IMAX=1-IMAX+1
CALL TRANS(X,VENEU,1,1,12)
WRITE(6,22) CX(1),1,1,IMAX)
XH*X /*(IP10E12,3)
22 FORMAT(1X,6H * X /*(IP10E12,3))
CALL TRANS(Y,XNNEU,JK,JKMAX,11,12,100)
DO 1 JJ=1,JKMAX,2
J1=JJ+1
DO 2 I=1,IMAX
TEMP=1.0/NORM
Y(1,1)=Y(1,1)*TEMP
1 IF (JJ=0,JKMAX) GO TO 2
CONTINUE
2
C
WRITE(6,23) CY(1,1),1,1,IMAX
IF (JJ=0,JKMAX) GO TO 3
WRITE(6,24) CY(1,2),1,1,IMAX
FORMAT(1X,9H*YT(1,1)*(1P,0E12,.3))
3
CONTINUE
CALL PLFXOOC(X,Y(1,1),IMAX1,0,0)
IF (JJ,NE,JKMAX) GO TO 44

```

```

      CALL SWRITE(WX,WY,JK(JJ),0,0,0,DUM)
      CALL PLOT(0,0,0,4444)
      CALL CLOSE(WX)
      GO TO 4
 44  CONTINUE
      CALL SWRITE(WX,WY,JK(JJ),JK(JJ),0,0,0,DUM)
      CALL PLOT(0,0,0,4444)
      CALL PLFX0(X,YT(1,2),IMAX1,0,1)
 45  CONTINUE
      CALL PLOT(0,0,8888)
      CALL PLOT(0,0,1666)
 1   CONTINUE
C     100 CONTINUE
      IF(IF2,EQ,0) GO TO 200
C
      CALL TITLE((1,12,JK,JKMAX)
      IMAX=12-1
      IMAX1=IMAX+1
      CALL TRANSXY(VEGAM,(1,12)
      WRITE(6,122) (X(I),I=1,IMAX1)
 122  FORMAT(1X,6H*(1)12,3)
      CALL TRANSY(XNGAM,JK,JKMAX,(1,12,50)
      DO 101 I=1,JKMAX,12
      J1=J1+1
      DO 102 I=1,MAX
      TEMP=1.0*NORM
      YT(1,1)=Y(1,JJ)*TEMP
      IF(JJ,EQ,JKMAX) GO TO 102
      YT(1,2)=Y(1,JJ)*TEMP
      CALL MINMAX(XMIN,XMAX,YT,YMIN,YMAX,IMAX1,2)
      WRITE(6,123) (YT(I,1),I=1,IMAX)
 123  FORMAT(1X,9H*T(1,1)/1P10I12,3)
      IF(JJ,EQ,JKMAX) GO TO 103
      WRITE(6,124) (YT(I,2),I=1,IMAX)
 124  FORMAT(1X,9H*VT(1,2)/1P10E12,3)
 103  CONTINUE
      CALL PLFX0(X,YT(1,2),IMAX1,0,0)
      IF(JJ,NE,JKMAX) GO TO 144
      CALL SWRITE(WX,WY,JK(JJ),0,0,0,DUM)
      CALL PLOT(0,0,0,4444)
      CALL CLOSE(WX)
      GO TO 104
 144  CONTINUE
      CALL SWRITE(WX,WY,JK(JJ),JK(JJ),0,0,0,DUM)
      CALL PLOT(0,0,0,4444)
      CALL PLFX0(X,YT(1,2),IMAX1,0,1)
 104  CONTINUE
      CALL PLOT(0,0,8888)
      CALL PLOT(0,0,1666)
 101  CONTINUE
C
      200 CONTINUE
END

```



```

PLF01070
PLF01080
PLF01090
PLF01100
PLF01110
PLF01120
PLF01130
PLF01140
PLF01150
PLF01160
PLF01170
PLF01180
PLF01190
PLF01200
PLF01210
PLF01220
PLF01230
PLF01240
PLF01250
PLF01260
PLF01270
PLF01280
PLF01290
PLF01300
PLF01310
PLF01320
PLF01330
PLF01340
PLF01350
PLF01360
PLF01370
PLF01380
PLF01390
END

51   IN=1
      IF (N-1,NE,1) CALL PLOT(XIN,YIN,2)
      IF (1,E0,1) CALL PLOT(XIN,0,1,2)
      55  IF ((N-1),GE,2) GO TO 51
          CALL PLOT(XIN,0,1,0,444)
      === SHIFT ORIGIN
          CALL TSMB(CANISN,12,IX,IXN,1,YIN)
          CALL PLOT(0,0,0,444)
          IF (PLOT, E0,0) RETURN
      DO CONTINUE
      === LINE
          JA=1ABS(C.)
          I=0
      === DO LOOP
          51  I=I+1
              X=(ALOGIO(X(I))-FLOAT(KX))*DECX
              Y=(ALOGIO(Y(I))-FLOAT(KY))*DECY
              IF (1,E0,1) CALL PLOT(X,I,0,3)
              IF (C,J,E0,1) CALL DASHP(X,Y,I+1,0)
              IF (C,J,E0,0) GO TO 561
              JD=C-I-((I-1)/JA)*JA
              IF ((JD,F0,0) CALL SYMBOL (XI,YI,2,L,O,-1)
              IF (C,J,L,T,0) GO TO 565
          IN=I+2
              XIN=(ALOGIO(X(IN))-FLOAT(KX))*DECX
              IF (N-1,NE,1) YIN=(ALOGIO(Y(IN))-FLOAT(KY))*DECY
              CALL DASHP(XIN,YI,1,0)
              IF (N-1,NE,1) CALL DASHP(XIN,YIN,1,0)
              55  IF ((N-1),GE,2) GO TO 551
                  CALL PLOT(0,0,0,1444)
                  CALL PLOT(WX+120,-14,-3)
                  CALL PLOT (0,10,1,3)
          RETURN
      END

      SUBROUTINE PLOT5(XX,YY,MM,NORM,IV,JV,CMM,IFG)
      DIMENSION XC(401),YC(IV,JV),XM(400),Y(401),Y(20,400),M(400)
      DIMENSION JK(20),Y(400,5),JK(15),MM(400)
      COMMON /PPPP/WX,Y,ANISN,CM,CMG
      COMMON /OPT/F1,F2,IF3,IF4,IF5,IF6,1BOUND
      COMMON /TITLE/IX,IXN,IX,YIN,1,YMIN,YMAX
      DIMENSION IX(20),IX(18),Y(18),Y(15,20)
      COMMON /TITLE/IXN,IX,YIN,1,YMIN,YMAX,XMAX,YMAX
      REAL NORM
      1FF=IFG-1
      IF (1FF,NE,1) BOUND STOP *ERROR STOP MID OR BOUNDARY POINT*
      X=W*280,
      Y=W*200,
      CALL TITLE(11,12,JK,JMAX)
      IMAX=12-1
      IMAX1=MAX
      IF (IFG,E0,2) IMAX1=IMAX1+1
      CALL TRANSX(X,XX,11,12)
      CALL TRANY(Y,YY,JK,JMAX,11,12,1,12)
      C
      CALL TRANSM(M,MM,[1,12]
      WRITE(6,20) CM([1],[1,1,MAX])
      20 FORMAT(IX,20,15)
      CALL WRITE(X,IMAX1,4H* R ,4H*
      L,1
      00 212 [1,2,1,MAX
      IF (M(1),E0,M(1-1)) GO TO 212
      XM(L)=X(K[1])
      L=L+1
      212 CONTINUE
      IMAX=-1
      IF (L,E0,1) GO TO 21
      CALL WRITE(XM,LMAX,4H*XMM*,4H
      ) )
      21 CONTINUE
      DO 30 J=1,JKMAX
          30 WRITE(6,31) '(Y(J,1),J=1,[MAX1]
      31 FORMAT(IX,',*XNNEN,*'(1P10E12,3))
      IF (IFG,E0,2) GO TO 2360
      DO 211 I=1,1,MAX
          211 X(I)=X(I)*(X(I+1)-X(I))/2,
              WHITE(6,214) '(X(I),I=1,[MAX1)
              214 FORMAT(IX,7H * X * /(1P10E12,3))
      2360 CONTINUE
      IF (JKMAX,GT,5) GO TO 360
      DO 1 I=1,1,MAX1
          DO 1 J=1,JKMAX
              YT(I,J)=Y(J,I)/NORM
      1 CONTINUE
      CALL MNMA (X,XMIN,XMAX,YT,YMIN,YMAX,[MAX1,JKMAX]
      DO 3 J=1,JKMAX
          WRITE(6,10) J,(YT(I,J),I=1,[MAX1)
      10 FORMAT(IX,7H * YT *,I2/(1P10E12,3))
      3 CONTINUE

```

```

4 CALL SEMLOG(X,YT(1,1),IMAX1,10,0,1)
  IF(JKMAX.EQ.1) GO TO 400
  DO 5 J=2,JKMAX
    CALL SEMLOG(X,YT(1,J),IMAX1,10,1,J)
  5 CONTINUE
  400 CONTINUE
    CALL SWRITE(WX,WY,0,0,5,JK,JKMAX,CMM)
    CALL PLOT(0,0,444)
    IF(L,E@,1) GO TO 9
    DECX=(XMAX-XMIN)/WX
    DO 8 IM=1,IMAX
      XM(1M)=(XM(1M)-XM(IN))/DECX
      CAL FSEGMAXC(1M),0,XM(1M),WY)
    8 CONTINUE
    CALL PLOT(0,0,444)
  9 CONTINUE
    CALL CLOSE(WX)
    CALL MINMA(CX,XMIN,XMAX,YT,YMIN,YMAX,IMAX1,5)
    DO 13 J=1,5
      300 DO 11 I=1,IMAX1
        DO 11 JT=1,5
          11 YT(I,JT)=YT(I,J)/NORM
          CALL SEMLOG(X,YT(1,I),IMAX1,10,1,J)
          DO 13 J=1,5
            13 WRITE(6,10) J,YT(I,J),I-1,IMAX1
            CALL SEMLOG(X,YT(1,12),IMAX1,10,0,1)
            DO 15 JT=2,5
              15 YT(I,JT)=YT(I,J)/NORM
              CALL MINMA(CX,XMIN,XMAX,YT,YMIN,YMAX,IMAX1,5)
              DO 13 J=1,5
                13 WRITE(6,10) J,YT(I,J),I-1,IMAX1
                CALL SEMLOG(X,YT(1,12),IMAX1,10,0,1)
                DO 15 JT=2,5
                  15 YT(I,JT)=YT(I,J)/NORM
                  CALL SEMLOG(X,YT(1,I),IMAX1,10,1,J)
                  15 CONTINUE
                    CALL SWRITE(WX,WY,0,0,5,JK,5,CMM)
                    CALL PLOT(0,0,444)
                    IF(L,E@,1) GO TO 25
                    DECX=(XMAX-XMIN)/WX
                    DO 7 IM=1,IMAX
                      XM(1M)=(XM(1M)-XM(IN))/DECX
                      CAL FSEGMAXC(1M),0,XM(1M),WY)
                    7 CONTINUE
                    CALL PLOT(0,0,444)
  25 CONTINUE
        CALL CLOSE(WX)
        CALL PLOT(0,0,888)
        CALL PLOT(0,0,666)
        IF(JKMAX.GT.10) GO TO 40
        DO 16 I=1,IMAX1
          DO 16 JT=6,JKMAX
            JA=JT-5
            JK(JA)=JK(JT)
            41 YT(I,JA)=YT(I,T)/NORM
            JKMAX=JKMAX-5
            DO 17 J=1,JKMAX
              17 WRITE(6,10) J,YT(I,J),I-1,IMAX1
              CALL MINMA(CX,XMIN,XMAX,YT,YMIN,YMAX,IMAX1,JKMAX)
              CALL SEMLOG(X,YT(1,I),IMAX1,10,0,1)
  400 CONTINUE
    CALL SEMLOG(WX,WY,0,0,5,JK,JKMAX,CMM)
    DO 18 J=2,JKMAX5
      CALL SEMLOG(X,YT(1,J),IMAX1,1,1,J)
  18 CONTINUE
  500 CONTINUE
    CALL SWRITE(WX,WY,0,0,5,JK,JKMAX5,CMM)
    CALL PLOT(0,0,444)
    IF(L,E@,1) GO TO 509
    DO 508 IM=1,LMAX
      CALL FSEGMAXC(1M),0,XM(1M),WY)
    508 CONTINUE
    CALL PLOT(0,0,444)
  509 CONTINUE
    CALL CLOSE(WX)
    CALL PLOT(0,0,888)
    CALL PLOT(0,0,666)
    40 CONTINUE
    DO 51 I=1,IMAX1
      DO 51 JT=6,10
        51 YT(I,JT)=YT(I,J)/NORM
        CALL MINMA(CX,XMIN,XMAX,YT,YMIN,YMAX,IMAX1,5)
        DO 53 J=1,5
          53 WRITE(6,10) JJ,YT(I,J),I-1,IMAX1
          CALL SEMLOG(X,YT(1,I),IMAX1,1,0,1)
          CALL PLOT(0,0,444)
        53 CONTINUE
        CALL SWRITE(WX,WY,0,0,5,JK,5)
        IF(L,E@,1) GO TO 58
        DO 55 JT=2,5
          55 YT(I,JT)=YT(I,J)/NORM
          CALL SEMLOG(X,YT(1,I),IMAX1,1,1,J)
          CALL PLOT(0,0,444)
        55 CONTINUE
        CALL PLOT(0,0,444)
        57 CONTINUE
        58 CONTINUE
        CALL CLOSE(WX)
        CALL PLOT(0,0,777)
        CALL PLOT(0,0,666)
        JKMAX1=JKMAX-10
        DO 41 I=1,IMAX1
          DO 41 JT=1,JKMAX
            JA=JT-10
            JK(JA)=JK(JT)
            41 YT(I,JA)=YT(I,T)/NORM
            DO 42 J=1,JKMAX1
              42 WRITE(6,10) JJ,YT(I,J),I-1,IMAX1
              CALL MINMA(CX,XMIN,XMAX,YT,YMIN,YMAX,IMAX1,JKMAX1)
              CALL SEMLOG(X,YT(1,I),IMAX1,1,0,1)
  PL000530
  PL000540
  PL000550
  PL000560
  PL000570
  PL000580
  PL000590
  PL000600
  PL000610
  PL000620
  PL000630
  PL000640
  PL000650
  PL000660
  PL000670
  PL000680
  PL000690
  PL000700
  PL000710
  PL000720
  PL000730
  PL000740
  PL000750
  PL000760
  PL000770
  PL000780
  PL000790
  PL000800
  PL000810
  PL000820
  PL000830
  PL000840
  PL000850
  PL000860
  PL000870
  PL000880
  PL000890
  PL000900
  PL000910
  PL000920
  PL000930
  PL000940
  PL000950
  PL000960
  PL000970
  PL000980
  PL000990
  PL001000
  PL001010
  PL001020
  PL001030
  PL001040
  PL001050
  PL001060

```

```

CALL PLOT(0,0,444)
IF (JKMAX1.EQ.1) GO TO 600
DO 43 J=2,JKMAX1
CALL SEMLOGIX,Y(I,J),IMAX1,1,1,J)
43 CONTINUE
600 CONTINUE
CALL SWRITE(WX,WY,0,0,5,JJK,JKMAX1)
IF (L,EQ.1) GO TO 609
DO 608 IM=1,MAX
CALL FSEG(MC(M,IM),0,XM(M),WY)
608 CONTINUE
609 CONTINUE
CALL CLOSE(WX)
CALL PLOT(0,0,777)
CALL PLOT(0,0,666)
RETURN
END

```

```

PL001610          SUBROUTINE SEMLOG(X,Y,N,J,IPLT,L)
PL001620          CCCCCC  FAILY FIRED LOG-LOG PLOT
PL001630          CAUTION   X,Y MUST BE POSITIVE
PL001640          C          PLOT FROM (X(I,J),Y(I,J)) TO (X(N),Y(N))
PL001650          C          J IS THE ALTERNATE NUMBER OF DATA POINT TO PLOT A SYMBOL
PL001660          C          SEE GRAPHIC PLOTTER MANUAL
PL001670          C          THIS GRAPH RANGES OVER 0,0 TO 280,0 (MM) ON X-AXIS
PL001680          C          AND SHIFT ORIGIN 400,0 TO RIGHT
PL001690          C          CHARACTERISTIC OF THIS ROUTINE
PL00170          C          1. NEED NOT TWO REST STORAGE
PL001710          C          2. DATA (X AND Y) IS NOT DELETED
PL001720          C          CCCCCC  FIXED PARAMETER
PL001730          C          GX,GY = ORIGIN    (0,0,0)
PL001740          C          WX = WIDTH OF X-AXIS (MM) --- 180.0
PL001750          C          WY = WIDTH OF Y-AXIS (MM) --- 180.0
PL001760          C          L = SYMBOL NO. --- 1
PL001770          C          DIMENSION ID(20),IX(18),IY(18)
PL001780          C          DIMENSION XC18,YC18
COMMON /PPPP/WX,WY,ANISN,CH,CMG
COMMON /TITLE/IN,IX,LYN,Y,XMIN,XMAX,YMIN,YMAX
COMMON ANISN(36)*CM(18,20),CMG(18,20)
COMMON LOGLN
IF(IPLT,NE,0) GO TO 100
C***** FIX
CAL PLOT(20,14,0,-3)
GX=0,
GY=0,
CAL FIXNUM(X,XMIN,XMAX,WX,DECK)
C***** X-AXIS
CAL PLOT(0,0,0,444)
YML=ALOG10(YMIN)
YMA=ALOG10(YMAX)
KY=INT(YML)
IF (YML.LT.0,) KY=KY-1
MY=INT(CYML)+1
IF (CYML.LT.0,) MY=MY-1
NY=MY-KY
IF (NY.LE.2) GO TO 5
2 DECY=MY/FLOAT(NY)
LOGLN=0
Y=0
DO 41 I=1,NY
HY=DECY*FLOAT(I-1)
CALL SYMBOL(-10,HY,2,8,2H10,0,0,2)
EY=FLOAT(CY+(I-1))
CALL NUMBER(-7,0,HY-5,0,2,0,EY,0,0,1)
41 CONTINUE
CALL SYMBOL(-10,0,WY,2,8,2H10,0,0,2)
CALL NUMBER(-7,0,WY+5,0,2,1,FLOAT(MY),0,0,1)
GO TO 43
C

```


J A E R I - M 6954

```

SUBROUTINE TRANSM(M,MM,[1,12])
  DIMENSION M(400),MM(400)
  IMAX=12-11
  DO 1 I=1,IMAX
    1 I=I+1-1
    M(I)=MM(I,1)
    RETURN
  END

SUBROUTINE SWRITE(X,Y,[1,12,*1OPT,JK,IMAX,CMM)
  DIMENSION JK(20)
  COMMON /PPPP/ WX,WY,AN(SN,CM,
  DIMENSION AN(SN,36),CR(18,20),CMG(18,20),CMM(18,20)
  XX=X+0,0
  IF (IOPT,EQ,2) GO TO 100
  Y=Y+20.
  CALL SYMBOL(WX,WY,5,0,5H - R=0,05)
  CALL NUMBER(WX+3.,WY,5,0,0,FLOAT([1],0,0,-1)
  IF (I2.EQ.0) RETURN
  WY=WY-10.
  CALL SYMBOL(WX,WY,5,0,5H R=0,05)
  CALL NUMBER(WX+23.,WY,5,0,0,FLOAT([12],0,0,-1)
  RETURN
  100 CONTINUE
  DO1=1,IMAX
    I=JK(I)
    YY=Y-10.*FLOAT([1]-1)+10,0)
    CALL SYMBOL(CXX,YY,5,0,10,0,-2)
    CALL SYMBOL(CXX+10,0,YY,5,0,0,CMM(I,1)),0,0,5)
    1 CONTINUE
  END

SUBROUTINE TRANSX(CX,XN,[1,12])
  DIMENSION X(401),XN(401)
  IMAX=12-11
  DO 1 I=1,IMAX
    1 I=I+1-1
    X(I)=XN(I,1)
    RETURN
  END

SUBROUTINE TRANSY(CY,YN,[1,12])
  DIMENSION Y(100,20),YN((KK,400),JK(20))
  IMAX=12-11
  DO 1 J=1,JKMAX
    JJ=JK(J)
    DO 1 I=1,IMAX
      II=I+1-1
      Y(I,J)=YN(I,1,J,J)
    1 CONTINUE
  END

SUBROUTINE TRANJCY,YN,JK,JKMAX,[1,12,JK(20)]
  DIMENSION Y(20,400),YN((IV,IV),JK(20))
  IMAX=12-11
  IF (IFG,EQ,2) JKMAX=JKMAX+1
  DO 1 I=1,JKMAX
    I=JK(I)
    DO 1 J=1,JKMAX
      JJ=J+1-1
      Y(I,J)=YN(I,1,J,J)
    1 CONTINUE
  END

SUBROUTINE TRANJY,YN,JK,JKMAX,[1,12,IV,JV,IFG)
  DIMENSION Y(20,400),YN((IV,IV),JK(20))
  IMAX=12-11
  IF (IFG,EQ,2) JKMAX=JKMAX+1
  DO 1 I=1,JKMAX
    I=JK(I)
    DO 1 J=1,JKMAX
      JJ=J+1-1
      Y(I,J)=YN(I,1,J,J)
    1 CONTINUE
  END

SUBROUTINE TITLE([1,12],JK,JKMAX)
  DIMENSION JK(20),IX(18),IY(18)
  COMMON /TITLE/ IX,N,IX,LYN,IY,XMIN,XMAX,YMIN,YMAX
  READ(5,13) I1,I2
  READ(5,14) JKMAX,(JK(I),I=1,JKMAX)
  21 FORMAT(16I5)
  READ(5,10) IXN,(IX(I),I=1,IXN)
  READ(5,10) IYN,(IY(I),I=1,IYN)
  10 FORMAT(15,17A4)
  C
  WRITE(6,12) I1,I2
  WRITE(6,13) JKMAX,(JK(I),I=1,JKMAX)
  WRITE(6,14) IXN,(IX(I),I=1,IXN)
  WRITE(6,14) IYN,(IY(I),I=1,IYN)
  C
  12 = I2 + 1
  12 FORMAT(1X,'11.I2',*2I16)
  13 FORMAT(1X,'JKMAX',/(10I10))
  14 FORMAT(1X,18A4)
  C
  RETURN
  END

TRA00010
TRA00020
TRA00030
TRA00040
TRA00050
TRA00060
TRA00070
TRA00080
TRA00090
TRA00100
TRA00110
SWR00100
SWR00110
SWR00120
SWR00130
SWR00140
SWR00150
SWR00160
SWR00170
SWR00180
SWR00190
SWR00200
SWR00210
SWR00220
SWR00230
TIT00010
TIT00020
TIT00030
TIT00040
TIT00050
TIT00060
TIT00070
TIT00080
TIT00090
TIT00100
TIT00110
TIT00120
TIT00130
TIT00140
TIT00150
TIT00160
TIT00170
TIT00180
TIT00190
TIT00200
TIT00210
TIT00220
TIT00230
TIT00240
TIT00250

```

```

SUBROUTINE TSYMBC(ID, IDN, IX, IXN, IY, IYN)
DIMENSION ID(16), IX(18), IY(18)
CALL SYMBOL(-50, -20, 5, IX, 0, 4*IXN)
CALL SYMBOL(8, 230, 15, ID, 0, 4*IDN)
CALL SYMBOL(8, 220, 15, ID(19), 0, 4*IDN)
CALL SYMBOL(-20, 40, 5, IY, 90, 4*IYN)
RETURN
END
TSY00010
TSY00020
TSY00030
TSY00040
TSY00050
TSY00060
TSY00070
TSY00080

```

```

SUBROUTINE WRITE(X, IMAX1, HOL1, HOL2)
DIMENSION X(400)
WRITE(6,1) HOL1, HOL2, (X(I), I=1, IMAX1)
1 FORMAT(1X, 2A4/(IP10E12, 3))
RETURN
END
WR100010
WR100020
WR100030
WR100040
WR100050
WR100060

```