95,403

NOVEMBER 1973

MATT-1008

SWAN: A CODE FOR THE ANALYSIS
AND OPTIMIZATION OF FUSION
REACTOR NUCLEONIC CHARACTERISTICS

BY

E. GREENSPAN, W. G. PRICE, JR.,
AND H. FISHMAN

PLASMA PHYSICS LABORATORY

PRINCETON UNIVERSITY
PRINCETON, NEW JERSEY

This work was supported by U. S. Atomic Energy Commission Contract AT(11-1)-3073. Reproduction, translation, publication, use, and disposal, in whole or in part, by or for the United States Government is permitted.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SWAN: A Code for the Analysis and Optimization of Fusion
Reactor Nucleonic Characteristics

E. Greenspan*, W. G. Price, Jr., and H. Fishman
Plasma Physics Laboratory, Princeton University
Princeton, New Jersey 08540

ABSTRACT

This report is intended as a User's Manual for "SWAN" — a code written for perturbation theory analysis and optimization of the nucleonic characteristics of fusion reactor blankets. SWAN is also applicable to any problem described by the inhomogeneous linear transport equation, e.g., radiation shield problems.

The optimization method is based on variational techniques. The variables of the optimization are material densities, with no restriction on their number. One joint restraint may be imposed in addition to upper and lower limits on each density. The parameter to be extremized and the restraint may be either a weight-type (linear) or nucleonic (bilinear) functional.

The transport calculations for SWAN are performed with the one-dimensional discrete-ordinate code ANISN.

* Present address: Nuclear Research Center, P. O. Box 9001, Beer-Sheva, Israel.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

TABLE OF CONTENTS

1. INTRODUCTION
 2. PROBLEM STATEMENT AND TERMINOLOGY
 3. SPECIAL FEATURES OF SWAN
 4. GENERAL DESCRIPTION OF SWAN
 - The Modules
 - The Data Files
 5. INPUT DATA DESCRIPTION
 - Driver Data,
 - ANISN Data
 - SWIF Data
 6. DETAILED DATA NOTES
 - Multiple Input Files
 - SHUFFL Control Cards
 - DRIVER Data
 - ANISN Data
 - SWIF Data
 - Cross Section Libraries
 7. OUTPUT DATA DESCRIPTION
 8. SAMPLE PROBLEM
 9. PROGRAMER'S GUIDE
 10. CALCULATIONAL PROCEDURES
 - Effectiveness Functions
 - Perturbation Parameters
 - Determination of New Density Distribution
- REFERENCES
- APPENDIX: PPL-FIDO

1. INTRODUCTION

The code SWAN was developed primarily for the analysis and optimization of the nucleonic characteristics of a Controlled Thermonuclear Reactor (CTR). It is also applicable, however, for the analysis and optimization of radiation shields or, in general, of any nuclear system that is described by the inhomogeneous linear transport equation. The needs for, and special features of the nucleonic analysis and optimization of CTR's are reviewed in a different report.¹

SWAN is composed of two major modules: SWIF and ANISN. ANISN is a one-dimensional discrete ordinates transport code² and SWIF is a code developed for perturbation calculations and optimization studies. The development of SWAN has reached such a phase that the code can be applied to a variety of problems useful in the analysis and evaluation of conceptual designs of CTR's. This report is aimed at providing the information required for using SWAN, for modifying and extending the code, as well as for adapting it to other computers. (SWAN was written in FORTRAN-IV using an IBM 360/91 computer; the coding is almost completely computer-type independent.)

The report is divided into four parts. The first part (Secs. 2-4) gives a general description of the problems solved by SWAN, special features of the program, and its structure. The second part (Secs. 5-8) is a user's guide. The third part (Sec. 9) contains information about the structure of SWAN that may be helpful for modifying the program and for transferring it to other

computers. The last part (Sec. 10) gives details about the computational procedures used in SWIF. The theoretical basis for the optimization method is summarized in a separate report.¹

A thorough familiarity with ANISN is required for the use of SWAN. For the sake of completeness and convenience the input specifications for ANISN have been included in this report. Note that these specifications refer to the version of ANISN currently operational at the Plasma Physics Laboratory, rather than the version distributed by the Radiation Shielding Information Center at Oak Ridge. These versions are functionally equivalent, but a few extra input and output options which do not appear in the RSIC documentation³ have been added at PPL.

2. PROBLEM STATEMENT AND TERMINOLOGY

The type of optimization problems that can be handled by SWAN can be characterized as follows :

Given the external source distribution

$$S(x) \equiv S(\underline{r}, E, \underline{\Omega}) \quad (1)$$

and the atomic density distribution $N_i(\underline{r})$ for all I materials the density of which are variables of the optimization, find the material density distribution that will extremize the functional $F_e(N_1, N_2, \dots, N_I)$ subject to:

1. the constraints imposed by the density limits

$$N_i^{\min}(\underline{r}) \leq N_i(\underline{r}) \leq N_i^{\max}(\underline{r}) \leq N_i^0(\underline{r}) \quad (2)$$

where these densities are assumed constant in each zone

2. the constraint on the total volume fraction available:

$$\sum_{i=1}^I \frac{N_i(\underline{r})}{N_i^0(\underline{r})} = \text{const} \leq 1.0 \quad (3)$$

and

3. preservation of the value of the restraints [denoted by the functionals $F_c(N_1, N_2, \dots, N_I)$] imposed on the problem.

The characteristic to be extremized (F_e) as well as the restraints (F_c) can be of either one of two general categories: a weight-type characteristic or a nucleonic characteristic. A weight-type characteristic is any characteristic expressible in the form

$$F_w(N_1, N_2, \dots, N_I) = \sum_{i=1}^I \int d\underline{r} C_{w,i}(\underline{r}) N_i(\underline{r}) , \quad (4)$$

to be referred to as a weight functional. A nucleonic characteristic is one expressible in the form of a bilinear functional:

$$F_b(N_1, N_2, \dots, N_I) = \int d\underline{r} [\langle \phi, S_b^+ \rangle + \langle \phi_b^+, S \rangle - \langle \phi_b^+, H\phi \rangle] \quad (5)$$

(using the notation $\langle \ , \ \rangle$ for $\iint dE d\Omega$) where the flux $\phi \equiv \phi(\underline{r}, E, \Omega)$ and the adjoint $\phi_b^+ \equiv \phi_b^+(\underline{r}, E, \Omega)$ are the solutions of, respectively, the linear Boltzmann equation and its adjoint

$$H\phi(\underline{r}, E, \Omega) = S(\underline{r}, E, \Omega) \quad \text{and} \quad H^+ \phi_b^+(\underline{r}, E, \Omega) = S_b^+(\underline{r}, E, \Omega). \quad (6)$$

Consequently the functional has the value

$$F_b = \int d\underline{r} \langle \phi, S_b^+ \rangle = \int d\underline{r} \langle \phi_b^+, S \rangle . \quad (7)$$

The adjoint source term, S_b^+ , is to be selected so as to give the adjoint function, and consequently the functional, an adequate physical meaning.

An Effectiveness Function may be defined for a bilinear functional as:

$$e_{b,i}(\underline{r}) \equiv \left\langle \phi, \frac{\delta S_b^+}{\delta N_i} \right\rangle - \left\langle \phi_b^+, \frac{\delta H}{\delta N_i} \phi \right\rangle \quad (8)$$

and for a weight functional as:

$$e_{w,i}(\underline{r}) \equiv C_{w,i}(\underline{r}) . \quad (9)$$

The Substitution Effectiveness Function (SEF) is defined as follows:

$$Q_i(\underline{r}) \equiv e_i(\underline{r}) - e_I(\underline{r}) \frac{N_I^0(\underline{r})}{N_i^0(\underline{r})} \quad (10)$$

so that

$$\delta F = \sum_{i=1}^I \int d\underline{r} e_i(\underline{r}) \delta N_i = \sum_{i=1}^{I-1} \int d\underline{r} Q_i(\underline{r}) \delta N_i \quad (11)$$

Equations (10) and (11) hold for both bilinear and weight functionals.

The material densities for the n^{th} iteration are obtained from the parameters in the $(n-1)$ iteration from the relation:

$$N_i^n(\underline{r}) = N_i^{n-1}(\underline{r}) + A_i^n Q_{e,i}^{n-1}(\underline{r}) + B_i^n Q_{c,i}^{n-1}(\underline{r}) \quad (12)$$

The procedure for calculating the A_i^n and B_i^n are described in detail in Sec. 10.

3. SPECIAL FEATURES OF SWAN

The functional to be extremized and the restraints can be any combination of bilinear or weight functionals, as long as at least one bilinear functional (expressing a nucleonic characteristic) is included. Note that this initial version of SWAN will accept only one restraint.

SWAN can accommodate any number of materials, the density of which may be either fixed or variable (subject to optimization). The effectiveness functions, SEF's, and perturbation integrals may be calculated for all the materials, including those not included in the transport calculation model.

The optimization method of SWIF is compatible with any transport code that can provide flux and adjoint distributions (one adjoint solution is required for each bilinear functional). As incorporated in SWAN, the optimization and perturbation calculations can be carried out for any external source (non-fissioning) problem that can be solved by ANISN, e.g., radiation shielding problems.

The number of independent variables for the SWIF optimization is $(I-1)*K$, where I is the number of materials of variable density and K is the number of material zones. In comparison, the number of variables for codes which optimize by moving zone boundaries is proportional to $K*D$, where D is the number of spatial dimensions.

The ability of SWAN to compute bilinear terms can provide the basis for general perturbation calculations and sensitivity studies. The structure of SWAN allows the easy addition of new

modules to provide extra analysis functions. Furthermore, transport codes other than ANISN (e.g., for two-dimensional analysis) can be incorporated into SWAN without great difficulty.

Most of the input to SWAN is specified in the PPL-FIDO format (as described in the Appendix). Either ANISN or SWIF may be executed alone, or they may be run in series (starting with either one). The memory size required for SWAN may be kept within the size required for ANISN alone by overlaying the major modules.

4. GENERAL DESCRIPTION OF SWAN

This section gives only an outline of the operation of SWIF, sufficient to understand the input data requirements. Detailed information about various parts of SWAN are given in later sections.

SWAN is a modular system that consists of two major modules with a driver, and seven data files. A block diagram showing these modules and data files is given in Fig. 1. The figure also shows the possible channels for the flow of information between the different units.

The Modules

DRIVER is the master program that controls the sequence of the calculations according to the options specified in the input. The user may, for example, request execution of SWIF without ANISN, ANISN without SWIF, or iteration of ANISN and SWIF starting with either module. A full description of the various options is incorporated into the DRIVER input data description.

SHUFFL reads the card input stream and prepares the primary data files for ANISN and for SWIF: FSWIF, VSWIF, FANISN, and VANISN.

ANISN performs the transport calculations for both the flux and adjoint distributions.

SWIF is the program that calculates the effectiveness functions, substitution effectiveness functions, and the new density distributions. It may also modify those input data arrays of ANISN and SWIF that depend on the density distribution.

734271

Fig. 1. "Problem modules and data files of SWAN."

The Data Files

Of the seven primary data files required by SWAN, two are cross section data libraries (XSECLIB and ACTVLIB), two are input data files for ANISN (FANISN and VANISN), two are input data files for SWIF (FSWIF and VSWIF), and one (XTROUT) allows communication from ANISN to SWIF.

XSECLIB contains the cross section data (transport and activity) required for ANISN calculations as well as for the calculation of the effectiveness functions by SWIF.

ACTVLIB contains the cross section data required by SWIF for the calculation of the contribution of the adjoint source term to the effectiveness functions, as well as for the preparation of the new adjoint source terms after a density variation.

XSECLIB and ACTVLIB are not optional; although ANISN can read cross sections directly from cards SWIF cannot. A description of the data format for these libraries is included in the Detailed Data Notes section.

VANISN and FANISN contain the data required by ANISN. All data arrays that depend on the density distributions are included in VANISN, so that they may be modified by SWIF if SWAN is iterating. All ANISN input data arrays that do not change during the iterations are included in FANISN. (The PPL-FIDO input processor allows multiple input files.)

VSWIF and FSWIF are similarly the variable and the fixed data files, respectively, required for SWIF input. SWIF modifies its own variable data file when SWAN is iterating.

XTROUT obtains the results from each ANISN calculation. Part of this data is required as input for SWIF: SWIF cannot be run without a suitable XTROUT file. XTROUT must be rewritten after each ANISN run in the iteration procedure of an optimization calculation. Although the user of SWAN needs to know only how to request the XTROUT option from ANISN, a description of this file is included with the Programmer's Guide.

An additional file, SWFOUT, may be used. This file receives most of the results of each SWIF run in the same way that XTROUT receives the results of each ANISN run. It is not required for the operation of SWAN, but may be useful in performing subsidiary calculations. A description of SWFOUT is included with the Programmer's Guide.

5. INPUT DATA DESCRIPTION

A condensed summary of the input specification required for SWAN is given below. It is intended to serve as a guide for data preparation. Additional data notes are given in the next section. As in the ANISN manual, a quantity in brackets is an array dimension while an expression in braces describes the condition under which the specific option or array is required. An array that is not required should not be specified in the input. The input for all data is, except where otherwise stated, in the PPL-FIDO format.

The input for SWAN consists of three data types: Driver data, ANISN data, and SWIF data. The data within the data sets for ANISN and for SWIF is arranged in blocks. Each block is terminated by a T. The input data arrays for ANISN and SWIF can be arranged in any order within a block. Following the data for DRIVER, one can specify either ANISN or SWIF data. Appropriate SHUFFL control cards have to be interleaved within ANISN and SWIF input arrays in a manner that will provide for correct loading of the data into the data files (see description in Sec. 6).

DRIVER DATA

Title card - format (12A4)

Option specification card - format (5I3)

1. IFIL - Preparation of input data files (FSWIF, VSWIF, FANISN, VANISN)
 - 0 - use existing data files
 - 1 - prepare new data files
2. IOPT - Sequence of calculations
 - 0 - ANISN run only
 - 1 - SWIF run only
 - 2 - start with ANISN followed by SWIF
 - 3 - start with SWIF followed by ANISN
3. IEF - Purpose of SWIF calculations
 - 0 - optimization calculations
 - 1 - effectiveness functions only
4. IMFL - Modifications to input files VANISN and VSWIF
 - 1 - SWIF will initialize contents of files {IEF=1}
 - 0 - SWIF will not modify contents of files
 - 1 - SWIF will modify contents of files {IEF=0}
5. IMAX - Maximum number of iterations on material density distributions {IOPT $\geq$ 2; IEF = 0}.

ANISN DATA

(This is a slight revision of section III of the ANISN manual, with changes made to describe the current PPL version of ANISN.)

Title card - format (12A4, 12X, 112)

1. Title - 48 characters maximum
2. 0 or maximum execution time in hundredths of seconds
- 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*) {IPM = 0}

22. IPM 0 - no effect
1 - enter shell source (18*)
by group and angle {IQM = 0}
IM - enter shell source by interval, group, and
angle {IQM = 0}
23. IPP interval number which contains shell source if
IPM = 1; 0 otherwise
24. IIM inner iteration maximum
25. ID1 0 - no effect
add 1 to print angular flux
add 2 to punch scalar flux
add 4 to write the XTROUT information on unit 3
add 8 to suppress printing of the source and
scalar flux
26. ID2 0 - no effect
1 - use "TAPEMAKER"-prepared group independent
cross section tape (containing MTP materials)
2 - use cross sections and fixed source from
previous problem
3 - use internal TAPEMAKER option
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) and print them
-1 print and also punch activities by interval

29. ICM outer iteration maximum
use ICM = 1 if IEVT = 0
ICM may be 0 if IFN < 0 and IPRT = 4 or 5
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
>0 - execute diffusion solution for specified groups
(24\$) as long as outer iteration # $\leq$ IDAT2
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
3 - weighted difference model (preferred option)
4 - linear model, weighted model if negative
34. IFN 0 - enter fission guess (2*)
1 - enter flux guess (3*)
2 - use fluxes from previous case
<0 - restart by reading XTROUT data from
unit # |IFN|
35. IPRT 0 or 2 or 4 - print cross sections
1 or 3 or 5 - do not print cross sections
0 or 1 - print all balance tables
2 or 3 - print only summary balance table
4 or 5 - do not print any balance table

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. DFML transverse dimension for void streaming correction
8. XNP 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.

Cross sections {ID2 = 0}

13\$ Library ID numbers [MTP] {MTP > 0}

14* Card cross sections [IHM × IGM × MCR] {MCR > 0}

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

Cross Sections {ID2 = 3}

13\$ Library ID numbers [MTP] {MTP > 0}

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

14* Card cross sections [IHM × IGM], MCR sets {MCR > 0}

NOTE: Each card cross section set is entered as an individual 14* array, followed by a T.

Fixed Source {IEVT = 0 and ID2 ≠ 2}

17* Distributed source [IM × IGM] {IQM = 1} (source per unit volume)

18* Shell source [MM × IPM × IGM] {IPM > 0} (flux on outside of interval)

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

Flux or Fission Guess {IFN = 0 or 1}

2* Fission density [IM] {IFN = 0}

3* Flux guess [IM × IGM] {IFN = 1}

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

Remainder of Data

1* Fission spectrum [IGM]

4* Radii by interval boundary [IM + 1]

5* Velocities [IGM] (must truly be velocity if IEVT = 2)

6* Angular quadrature weights [MM] *

*
MM = ISN + 1 for plane or sphere
MM = (ISN × (ISN + 4))/4 for cylinder

- 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\$ Position numbers for activities [ID3] {ID3 > 0}
- 24\$ Diffusion calculation or
infinite medium markers [IGM] {IDAT2 > 0}
- 25* Albedo by group - right boundary [IGM] {IBR = 3}
- 26* Albedo by group - left boundary [IGM] {IBL = 3}
- 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)
- 2. IHTF position of σ_{total} in weighted cross sections
- 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 - print without punching
 - 1 - punch and print weighted cross sections
 - 1 - punch without printing

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

29\$ summation limits for activities [2 × ID3]{ID3 > 0}

34* P_L scatter constants [JT × MM]* {IXTR = 1}

38* densities for use in
activity calculation [IM × ID3] {ID3 > 0}

NOTE: The above data is followed by a T.

* 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.

SWIF DATA

Title card - format (10L1, 10I1, 15A4)

1. Indicators to control printout: Use T for print, F for do not print.

Column 1: Addressing parameters for MUCHO common block

2: Fluxes and currents

3: Adjoints as given by ANISN

4: Inverted adjoints

5: XSECLIB and ACTVLIB cross sections used by SWIF

6: Terms of effectiveness functions

7: Density dependent input arrays

8-10: Not used

2. Indicators to control output of revised density dependent data arrays.

Column 11: Not used

12: for SWIF 11* Array

13: for ANISN 12* Array

14: for ANISN 17* Array

15: for ANISN 38* Array

16-20: Not used

The following key is used for specifying the type of output for these arrays:

0 - no output

1 - print only

2 - punch only

3 - print and punch

4 - write on VANISN/VSWIF

5 - print and VANISN/VSWIF

6 - punch and VANISN/VSWIF

7 - print, punch, VANISN/VSWIF

3. Title - 60 characters maximum

Initial Data

1\$ Integer Parameters [12]

1. INF number of functionals (≤ 2)
2. ISA number of bilinear functions ($\leq$ INF)
3. IMM type of extremum
+1 maximum
-1 minimum
4. ISK number of materials of variable concentration
5. ISD total number of materials for SWIF ($\geq$ ISK)
6. IDN density adjustment mode
0 fill-up {I01(2) $\neq$ 0}
1 rescale
7. ITA selection of amplitude, A, for density variation
-2 perturbation calculation of fractional change of functional {INF < 2 }
-1 perturbation calculation of absolute change of functional
0 prescribed density change {IBO = 0}
N externally specified. Calculation is repeated N times, with an order of magnitude increase in A after each time
8. IBO relation between the B and A coefficients
0 - independent B for each material
1 - fixed ratio of B/A for all materials

9. ICT convergence test

-1 maximum possible change in functional {I01(2)≠0}

0 bypass

1 relative maximum change of functional

2 relative maximum change of
ratio of functionals {INF = 2}

10. SAVE preparation of SWFOUT file

0 do not prepare

1 write the SWFOUT information on unit 30

11-12. not used

2* Floating Point Parameters [6]

1. EPDEN density deviation check

2. ADSCAL scale factor for adjoints

3-6. not used

Note: The above data is followed by a T.

Input Control Indicators

3\$ I01:specify 0/1 for bilinear/weight functional [INF]

4\$ Number of materials ($ISAS \leq ISD$) that go into each adjoint
source term [ISA]

Note: The above data is followed by a T.

Material Specification

5\$ Sequence numbers of materials that go into the adjoint
source terms [$\Sigma ISAS$]

6\$ ACTVLIB ID for corresponding materials [$\Sigma ISAS$]

7\$ XSECLIB ID for all materials of variable concentration
[ISK × (ISCT + 1)].

Material Densities

- 8* Maximum atomic density of each material of variable concentration if it were to occupy all the volume available for the materials of variable concentration [IZM $\times$ ISK]
- 9* Maximum permissible atomic density of each material of variable concentration [IZM $\times$ ISK]
- 10* Minimum allowable atomic density of each material of variable concentration [ISM $\times$ ISK]
- 11* Current atomic densities of all the materials [IZM $\times$ ISD]

Other Parameters

- 12* Coefficients (A) for density modification [ISK - 1] {IEF = 0}
- 13* Weight coefficients (C_w) for materials used in weight functionals [IM $\times$ ISD] {INF > ISA}
- 51* Convergence parameters for substitution effectiveness functions [(ISK - 1) * INF] {IEF = 0}

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

- 48\$ Sequence numbers of materials as they appear in ANISN input array number 38* [ID3] {IMFL = 1}.

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

6. DETAILED DATA NOTES

Multiple Input Files

All the card input data for a SWAN run may be supplied as one deck on unit 5. The case title and SWAN control parameters are read by DRIVER, and then all other cards are processed by SHUFFL to produce the ANISN and SWIF data files.

ANISN and SWIF read cards with a PPL version of the FIDO free-format card scanner, which is described in the Appendix. FIDO may be directed to change its logical input unit number; this allows the convenient segregation of SWIF from ANISN and fixed from variable data. The SHUFFL control cards are used to specify the transfer of initial data from the main card input stream on unit 5 to the proper SWIF and ANISN input files.

All of the SWIF data except the 11* array should be directed to unit 52 (FSWIF): this one array which depends on the material densities should be sent to unit 22 (VSWIF) with a SHUFFL control card 22X ahead of it and 52X behind. These will cause SHUFFL to switch its output unit and generate appropriate FIDO redirection cards. The 11* array is put into its own file so that revised data, generated by SWIF, can be written in its place without having to worry about the other arrays.

For the same reason the 12* arrays, the first 38* array, and all but the first 17* arrays for ANISN must be directed to unit 21 (VANISN) with 21X and 51X SHUFFL control cards, while all the rest of the ANISN cards are copied to unit 51 (FANISN). (The first 17*

array, the source for the flux density case, is assumed not to vary.) Note that SHUFFL must be alerted with "51" and "52" control cards (not 51X and 52X) at the beginning of the ANISN and SWIF data blocks, respectively, otherwise the initial portion of the fixed data will be copied to the SHUFFL default punch unit.

SHUFFL Control Cards

SHUFFL is a routine to rearrange card-image files. It reads a card stream, looking for control cards with the two characters "% in columns 1 and 2, and copies non-control cards to output units as directed. SHUFFL reads cards from unit NIN until an end-of-file is sensed. Cards are written onto unit NPUN, and echoed with comments onto unit NOUT. These units are set initially to: NIN = 5, NOUT = 6, NPUN = 51 (FANISN). The unit numbers can be changed, and the punching and editing suppressed, by use of the optional control cards. (But control cards are always edited.)

Following is the control card format, by columns:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	80	,
			+			X				X				+				
"	%	Δ	-	A	A	Y	Δ	B	B	Y	Δ	C	C	-	Δ		comments	
			Δ			Δ				Δ				Δ				

the "%" in columns 1 and 2 is required

Δ stands for a blank space

if column 15 is -, editing of punched cards is suppressed

if column 15 is +, editing is enabled

if $0 < CC \leq 99$, NOUT = CC before editing this card

if $0 < BB \leq 99$, NIN = BB and if column 11 is Y, unit BB is rewound

if column 4 is -, punching of input cards is suppressed

if column 4 is +, punching is enabled

if $0 < AA \leq 99$, and column 7 is X or Y, a punched card is generated

first and then NPUN = AA. The punched card has the form

X
Δ AA Y Δ Δ "Comments

i.e., it is a FIDO redirection card for unit AA. If column 7 is blank, this punched card is not generated.

Any field may be left blank and no actions will be performed;
fields which appear to be in error will be treated as though
blank and flagged with ERROR!

Note that all control cards would be interpreted by PPL-FIDO
as comment cards.

DRIVER Data

Input data file preparation - IFIL

Before any ANISN run, the input data files VANISN and FANISN have to be filled. Similarly before any SWIF run, the input data files VSWIF and FSWIF have to be filled. If all necessary input data files have been filled previously, use IFIL = 0 and supply only the DRIVER input. If any changes have to be made in one of the input data files, use IFIL = 1 and read all the input data for that file. When the first ANISN run follows a SWIF run, there is no need to define the file VANISN from cards; this file can be defined by SWIF (IMFL = ± 1 options).

Sequence of calculations - IOPT

A SWAN calculation that includes ANISN needs the XSECLIB cross section library and the data files VANISN and FANISN. A SWAN calculation that includes SWIF will require the two cross section libraries XSECLIB and ACTVLIB, the two input data files VSWIF and FSWIF, and the special data file XTROUT.

Purpose of calculations - IEF

SWAN can be used to calculate effectiveness functions and perturbation parameters only, by IEF setting to 1. In this option the program skips all the calculations associated with density modifications. With IEF = 1, SWAN does not return to ANISN after the SWIF run (i.e., the IMAX option is bypassed).

Modifications to input files - IMFL

The user can make SWIF prepare the variable arrays of ANISN (17*, 12* and 38*) by specifying the option IMFL = -1. In order to use this option it is necessary to provide the XTROUT file for at least one ANISN run as well as all the input for SWIF. The XTROUT file may be prepared by hand or with a dummy ANISN run setting ICM = 0, since the fluxes computed by ANISN are not required for this option. When IMFL = -1, IMAX should be set to 1.

Maximum number of iterations - IMAX

For the purpose of defining the iteration number, an iteration terminates whenever a SWIF run is terminated (regardless of the initial step in the calculations).

ANISN Data

All the ANISN runs for a given problem (on a given iteration) have to be performed in sequence. This sequence should start with a flux solution followed by adjoint solutions; the first adjoint should refer to the functional to be extremized (if it is a nucleonic characteristic). All adjoint runs should have the same geometry description, group structure and $S_n - P_\ell$ specifications as for the flux run. The preparation of the XTROUT file requires this stacking of all ANISN calculations (per iteration) in one run.

After completing each case, ANISN determines whether or not to proceed with another (e.g., an adjoint) by trying to read another card. If it detects an end-of-file, or if there is a card with exactly "END " in columns 1-4, then it stops; otherwise it looks for another full case with its own title card, etc. If SWAN will be executing ANISN more than once (IOPT = 2 or 3, IEF = 0) then termination with the "END " card is necessary, to allow the ANISN data files to be rewound.

A comment on the adjoint source may be useful. SWIF utilizes nucleonic functionals which estimate $\int d\underline{r} \langle \phi, S^+ \rangle$, so the obvious choice for S^+ is some kind of macroscopic cross section. In this case the appropriate choice of the input XNP is probably 0.0, implying no renormalization, so that the adjoint assumes its real physical meaning. This is in contrast to the direct flux, which is commonly normalized with $XNP \neq 0.0$ relative to a source distribution function.

Several new options are available with the PPL version of ANISN; these are discussed below. Refer to the ANISN User's Guide for descriptions of the regular ANISN options.

ID1 - The appropriate value for ID1 is obtained by adding together the values representing each of several possible options:

(a) add 1 to print the angular flux; (b) add 2 to punch the scalar flux; (c) add 4 to write the output from this case on the XTROUT file (described in detail in the Programmer's Guide); and (d) add 8 to suppress the printing of the source and the scalar flux. Option (c) was added to allow auxiliary programs to obtain the ANISN results, option (d) was added to save paper.

IPRT - The expanded choice of values for IPRT is provided to reduce the expense of balance table computation in many-zone systems and to allow the $ICM = 0$ option.

IFN - When $IFN = -N$, a flux guess is read from an XTROUT-format file with logical unit number N. This allows one, for example, to restart a previous case or to use an old flux result as a starting guess.

ICM - It is now meaningful to set $ICM = 0$. No flux calculation will be performed, so the activities and group collapsing will be performed using the flux guess, e.g., the flux in the XTROUT file if $IFN < 0$. If $IFN \geq 0$ then IPRT must be 4 or 5 since the angular flux guess required for balance table computation will not exist.

ID2 - The TAPEMAKER program, which prepares a special space-saving cross section input "tape" for ANISN, has been dismembered and reconstituted as an integral part of the PPL version of ANISN.

When $IDA = 3$ ANISN will try, after reading the 15\$ and 16* arrays and their "T," to read a 13\$ array (with MTP entries) terminated with a "T," and then/or a set (MCR decks) of 14* arrays, each one terminated with a "T".

Cross section mixing is done in the regular ANISN manner as specified by the 10\$, 11\$, and 12* arrays entered later. The special "tape" is not made available for later use, but the size of memory required is greatly reduced when this option is used.

ID3 - When activities are to be computed ($ID3 > 0$) two new arrays are required to control the energy and spatial summations.

The 29\$ array - $J(2, ID3)$ - should be entered as a series of pairs of numbers, one pair for each activity. The first number of the Kth pair, $J(1, K)$, is set to 1 if it is < 1 or $> IGM$, while the second number, $J(2, K)$, is set to IGM if it is < 1 or $> IGM$ (thus 0,0 is equivalent to 1, IGM). The summation over energy for each activity will be controlled by the corresponding entries in the 29\$ array: only the groups from $J(1, K)$ through $J(2, K)$ will be added. As an example of its use, the 29\$ array can be set to extract neutron-only integrals in coupled problems.

The 38* array - $D(IM, ID3)$ - is used to supply optional material density profiles for each activity. If $M(K)$, the material number provided in the 22\$ array for the Kth activity, is positive, then the real density profile of material $M(K)$ in each interval is used as the density for the activity calculation; however, if $M(K)$ is negative, then the entries $D(I, K)$ are used

(except that if $D(I,K) = 0$ for $I = 1$ to IM then $D(I,K)$ is reset to 1.0). A new column in the activity table print-out contains "Real," "1.0," or "38*" to indicate which option was used.

SWIF Data

Density Dependent Arrays Manipulation

The indicators to manipulate the density dependent arrays, columns 11-20 on the title card, must be correlated with the SHUFFL control cards. The indicator for an array which SWIF can revise should be set to 4, 5, 6 or 7 if the modified array is to be transferred to the variable file VANISN or VSWIF. Such an array must be initialized, presumably by either a previous SWIF run or by a SHUFFL control card.

The punch options - 2, 3, 6 and 7 - will cause card images of the specified array to be written on unit 7 in PPL-FIDO "Fixed" format.

Number of Functionals - INF, ISA

The present version of SWAN can handle one or two functionals per problem. These are the functional to be extremized and up to one additional restraint. The total number of functionals includes both bilinear and weight functionals. The number of bilinear functions is $ISA \leq INF$.

Number of Materials - ISK, ISD

ISD is the total number of materials that are of variable density and/or contribute to the functionals (either to the functional to be extremized or to the restraint). Out of these ISD materials, ISK ($\leq ISD$) are of variable density (active participants in the optimization). A given material is counted only once.

Density Adjustments - IDN

After every density change a check is made (in subroutine NEWDEN) to find whether the new density distribution exceeds the permissible limits, $N_i^{\max}(\underline{r})$ or $N_i^{\min}(\underline{r})$. In case the density limits were exceeded, NEWDEN adjusts the new density distribution using one of the following options:

(a) IDN = 0. In the zones in which the density of a given material exceeded its upper or lower limit, the density is brought back to the constraint. For problems with no restraints, this is the only density modification performed. For problems with a restraint of a bilinear functional type, this option should not be used (as it cannot assure the preservation of the restraint). The IDN = 0 option can preserve, however, certain types of restraints in the category of weight functionals. These are all those weight functionals in which the density of each material is weighted with a constant (space independent) factor. It is up to the user to decide whether it is appropriate to use this option or not. For a problem with a restraint, the total quantity of each material is preserved by adjusting the densities in the zones in which the densities did not reach their limits. When the adjustment requires addition of materials, this adaption is accomplished, for a maximization problem, by first filling up the zone with the largest value of the substitution effectiveness function and then proceeding to the zone of second largest SEF, etc. For a minimization problem, the addition starts at the zone of the lowest SEF. When the adjustment involves the subtraction of some quantity of the material, this subtraction starts at the zone of the maximum

SEF for a minimization problem, and at the zone of the minimum SEF for a maximization problem.

(b) IDN = 1. The value of the A coefficients are adjusted so that the zone density which most exceeded its limit is just restored to that limit. The densities in the other zones are automatically determined in accordance with the regular density variation procedures. For problems with restraints, this procedure will always preserve the restraint to first order.

Determination of the A_i and B_i Coefficients - ITA, IBO

The density modifications following a given iteration are determined by the A_i and B_i coefficients [see Eq. (12)]. The B_i coefficients are automatically determined by the program to preserve the restraint (see Sec. 10). For problems with no restraint $B_i = 0$. The amplitude of the density variation can be determined by the user via the specification of the A_i coefficients. There are four options for the determination of the A_i coefficients:

ITA = -2. Do not use this option for problems with a restraint. The A_i coefficients are calculated by the program (subroutine NEWDEN) to bring about a specified fractional change in the functional to be extremized. This change is a fraction of the maximum possible functional variation towards its extremum [the term P_8 of Eq. (36)], as predicted by first order perturbation theory. The fraction is specified by the user in the 12* array of SWIF input. If used for a problem with a bilinear functional restraint, SWIF interpreted this option as ITA = -1.

ITA = -1. This option is similar to the previous one except that the 12* array specifies the absolute change in the functional. Be careful not to ask for too large a change in the functional.

ITA = 0. The A_i coefficients are calculated by the program to bring about a specified change in the material density: the maximum absolute density change is specified in the 12* array (the value for the ISKth material is disregarded by the program). This option should not be used with the IBO = 1 option.

ITA = N, N = 1,2,3,... . The value of the A_i coefficients are specified by the user (12*). When $N > 1$ the program repeats the density change calculations N times. Before each new series of calculations (excluding the first one) the program multiplies the A_i coefficients used for the previous calculations by a factor of 10.

The method for the calculation of the A_i and B_i coefficients is determined by the IBO option (see Sec. 10). When IBO = 0, the coefficients of the different materials are independent. When IBO = 1, it is assumed that $B_i/B_1 = A_i/A_1$. The coefficient A_1 is to be specified so as to determine the desired amplitude of density variation. The IBO = 1 option should not be used with the ITA = 0 option. When IBO = 0, the ITA = -1, -2 options take the A_1 coefficient as a constant coefficient for all the materials (disregarding the specified value of A_i , $i > 1$).

Convergence Test - ICT

There are four options for testing the convergence of the iterative optimization calculations:

ICT = -1. The convergence test is performed with the maximum possible variation in the functional towards its extremum [the term P_8 of Eq. 36] as predicted by perturbation theory calculations. When the value of P_8 is not larger than ϵ_1 , convergence has been reached. Do not use this for problems with a restraint.

ICT = 0. No convergence test is made.

ICT = 1,2. The convergence test is based on the maximum relative deviation of the SEF. In the ICT = 1 option, convergence is reached when:

$$|Q_{e,i}^{\max} - Q_{e,i}^{\min}| / |Q_{e,i}^{\max}| \leq \epsilon_i, \quad i = 1, 2, \dots, \text{ISK} - 1$$

where $Q_{e,i}^{\max}$ and $Q_{e,i}^{\min}$ are, respectively, the maximum and minimum values of the SEF for the functional to be extremized and the i th isotope. If $Q_{e,i} = \text{constant}$ for all zones (a weight functional, for example), this convergence test is ignored.

In the ICT = 2 option, the convergence test is:

$$\left| \frac{(Q_{e,i}/Q_{c,i})_{\text{zone of } \max Q_{e,i}} - (Q_{e,i}/Q_{c,i})_{\text{zone of } \min Q_{e,i}}}{(Q_{e,i}/Q_{c,i})_{\text{zone of } \max Q_{e,i}}} \right| \leq \epsilon_i$$

$$i = 1, 2, \dots, \text{ISK} - 1.$$

This option can be used only for problems involving a restraint.

In both the ICT = 1 or 2 options, the convergence test for a given material is skipped in any zone in which the material has reached one of its density limits.

The convergence criteria ϵ_i are defined in the 51* array of SWIF input. These criteria are to be specified for the ISK - 1 materials for INF functionals. When ICT = -1 only one value, the first one in the 51* array, is used.

Density Deviation Check - EPDEN

When the absolute difference between the density and the density limits (either maximum or minimum) is smaller than EPDEN, the density is assumed to be at its limit. If in a given zone this limit is reached for all materials of variable concentration, this zone is excluded from the search for the optimum density distribution in the iterations that follow.

In the course of a series of optimization calculations it may be desirable to reinclude, in the search for the optimum density distribution, one or all the zones in which the density came within EPDEN of its limit. This can be accomplished by changing the value of EPDEN to be smaller than the difference between the actual density and the density limits.

Scale Factor for Adjoints - ADSCAL

All the moments of the adjoint function calculated by ANISN are multiplied by this factor prior to use by SWIF. This option enables one to adjust the normalization factor for the adjoint (if, for some reasons, it was not set correctly for the ANISN calculations). Usually this factor should be set to 1.0. The rescaled adjoint flux and angular moments can be printed by SWIF.

Functional Type Specification - I01 (3\$)

The first functional has to be the functional to be extremized. At least one functional has to be a bilinear functional. All other SWIF data has to be specified consistently with the order of the functionals.

Material Specifications - ISAS (4\$), IIS (5\$), LAS (6\$), LID (7\$)

The user should decide upon the sequence of materials for SWIF. In determining this sequence one should follow these rules:

(1) The first ISK materials should be the ones with variable concentration.

(2) The ISK_{th} material is the one the previous materials are substituted for, in calculating the substitution effectiveness functions.

(3) The sequence of the first (ISK-1) materials, and of the (ISK + 1) to ISD materials, can be arbitrary.

Once the sequence of materials is determined, all input specifications (cross sections, densities, A_i coefficients, etc.) should be given in the same sequence.

The ISK materials can include materials not included in the system, and hence, not included in the ANISN input. The inclusion of such materials can be useful for effectiveness calculations. If included, their densities (11*) should be negligible.

Arrays 5\$ and 6\$ can include any of the ISD materials. Array 7\$ specifies the ISK materials. (In this array, the XSECLIB library ID for all ($\ell + 1$) components of a given material have to be specified, P_0 through P_ℓ .)

Material Densities - DOMZ(8*), DMX(9*), DMN(10*), DP(11*)

The atomic densities are specified by zone in units of $(\text{cm} \times \text{barn})^{-1}$. The atomic densities of the first material are specified for every zone, followed by the atomic densities of the second material, and so on. The DOMZ array gives the density the material would have were it to occupy all the volume fraction

available in the given zone for the materials of variable concentration. In zones for which $DOMZ = 0.0$ the corresponding material is excluded from the search for optimal distribution.

The DMX and DMN arrays give the maximum and minimum density, respectively, that the material can have in a given zone. They are to be determined in accordance with the problem. When $IDN = 1$ and $ISK > 2$ set $DMX(ISK) = DOMZ(ISK)$. This will assure preservation of the constraint of total volume fraction.

The DP array gives the actual density distribution corresponding to a given system. The data for the first ISK materials in this array will be changed during the optimization procedure by SWIF. The DP array should be specified for all ISD materials. The initial densities of any of the ISK-1 materials should differ from the corresponding density limits by more than EPDEN (unless EPDEN equals zero).

Weight Coefficients - WF(13*)

Weight coefficients are to be specified (only when $INF > ISA$) by interval for all ISD materials of variable concentration. For materials that do not contribute to the weight functional weight coefficients of zero should be specified. In most problems the weight coefficients are interval independent.

Material Sequence for Activities - XMF(48\$)

Array 48\$ is required only when SWIF is to modify the 38* array of ANISN. In location N of the 48\$ array, enter the SWIF sequence number for the material an activity of which is called for in the same location N of the ANISN 38* array. Only sequence

numbers corresponding to the ISK materials of variable concentration are to be specified in the 48\$ array. Enter zero for activities of any material other than these.

Cross Section Libraries

The data libraries XSECLIB and ACTVLIB are assumed to be in the format of standard ANISN binary libraries. This consists of one or more pairs of FORTRAN "unformatted" records (description and data) plus a terminator record. Each pair is written into the library on unit #L with the FORTRAN statements:

```
WRITE(L) NG, NP, NC, MARKER, (NAME(K), K = 1, 12)
```

```
WRITE(L) [(X(I,J), I = 1, NP), J = 1, NG]
```

Here X (I,J) is a matrix, e.g., of ANISN cross sections, dimensioned NP by NG, with "MARKER" as its ID # in the library and a 48-character "NAME." The control number NC should ordinarily be zero; the terminator record (logical end-of-data) is a dummy description with NC = 7. ANISN and SWIF look in XSECLIB for transport cross section matrices dimensioned IHM by IGM; SWIF looks in ACTVLIB for cross section vectors with NP = 1.

The cross sections used for the calculation of the activities in ANISN should be the same cross sections used for calculating the adjoint source term for the corresponding reactions. This will assure that the nucleonic characteristics calculated by ANISN in the density distribution (flux) and probability (adjoint) spaces will be identical.

7. OUTPUT DATA DESCRIPTION

The output starts with the print out of the options specified for DRIVER. Then the cards processed by SHUFFL are printed unless a SHUFFL control card has requested otherwise.

The card images read by ANISN and SWIF from their input files can be printed by judicious use of the "O" option code of the PPL-FIDO card scanner. The switch which requests this card echoing is initially off, but may be turned on with a "+10" option code; a "-10" option code will turn it back off.

For the output of ANISN the reader is referred to the ANISN user's manual. The SWIF output is described briefly below. The SWAN sample case (Sec. 8) illustrates the output formats.

(1) ANISN title card and integer parameters (array 15\$) are printed.

(2) SWIF integer parameters (1\$) and floating point parameters (2*) are printed.

(3) If column 1 of the SWIF title card is T, addressing parameters for the common block MUCHO are printed.

(4) If column 2 of the SWIF title card is T, the scalar flux and higher flux moments (up to the order l corresponding to the P_l approximation) are printed.

(5) If column 3 of the SWIF title card is T, the angular moments of the adjoint solutions are printed. These moments are the ones calculated by ANISN as multiplied throughout by EDSCAL.

(6) If column 4 of the SWIF title card is T, the adjoint angular moments are reprinted, but this time inverted with respect to the energy group structure and with respect to the direction of

angular variable. This transforms the ANISN adjoints, which are calculated in inverted order, back to the physically meaningful order, in which form they are used in SWIF.

(7) If column 5 of the SWIF title card is T, the cross sections from the ACTVLIB library (for ISAS materials) and the cross sections from the XSECLIB library (for ISK materials) are printed out.

(8) If column 6 of the SWIF title card is T, tables of the terms used for calculating the effectiveness functions are printed out. The data of these tables is arranged (horizontally) in the following order (from left to right):

I	interval number
NB	number of functional
NA	number of bilinear functional
NI	number of material of variable concentration
IG	energy group number
AC	isotropic component of the AC term [Eq. (22)]
BC	isotropic component of the BC term [Eq. (24)]
CC	isotropic component of the CC term [Eq. (29)]
ISCT	maximum order (ℓ) of the P_ℓ approximation used
AC	sum of all ℓ, n components of AC [Eq. (22)]
BC	sum of all ℓ, n components of BC [Eq. (24)]

The above data is useful for debugging purposes. It can also be useful as the basis for perturbation and sensitivity calculations.

Output items (7) and (8) are printed in pairs, material by material. The sequence of output items (5) - (8) are repeated for every bilinear functional.

(9) The isotropic components and the anisotropic components of the effectiveness functions are printed by interval for every material and for every bilinear functional.

(10) These are followed by the print out of the effectiveness functions and of the substitution effectiveness functions. The last row of the first table gives the integrated (over all the volume) effectiveness function.

(11) Following is a print out of perturbation parameters. For details about these parameters, consult Sec. 10. The interpretation of the parameters printed as "maximum functional change" and "absolute maximum functional change" should be made with care. These changes are computed for each functional, postulating density changes subject to the density constraint but not to the restraint functional. Changes in both the functional to be extremized and the restraint functional (if any) are computed. If the problem is a maximization one ($IMM = 1$), the greatest increase in each functional is computed, whereas the greatest decrease is calculated in a minimization problem.

This is the end of SWIF output when the option for effectiveness calculation ($IEF = 1$) is used. The rest of the output data is connected with the density variation.

(12) Results from convergence tests are printed.

(13) Values of density variation coefficients A_i and B_i and listed followed by arrays of old densities, new densities and relative density variation. If the new density distribution exceeds one of the prescribed limits, the density distribution is adjusted according to the option specified in the input, and the density tables are reprinted.

When $ITA = N > 0$, the above data is repeated N times. After each iteration the value of the A_i coefficients is increased by a factor of 10. This option is provided for survey calculations that are aimed at providing a guide for the determination of the value of the A_i coefficients to be used for the optimization. Usually this option is used with the $IEF = 0$, $IMFL = 0$, and $IMAX = 1$ option.

(14) When the $IEF = 0$, $IMFL = 0$ options are used, the density dependent input data arrays can be printed out as specified in columns 11 - 20 of the SWIF title card.

8. SAMPLE PROBLEM

A listing of the input data cards for a sample problem is given below. The data is arranged in three sections: DRIVER input (cards 1-3), ANISN input (cards 4-78), and SWIF input (cards 79-99). The sample problem is characterized as follows:

- (1) A slab geometry, 80 cm wide, with 16 intervals and 16 zones.
- (2) The $S_4 - P_1$ approximation is used.
- (3) Eight energy groups.
- (4) The medium is composed of three materials, one of which has a fixed density.
- (5) The problem is considered to be an optimization problem. One iteration is called for.
- (6) There is no restraint.
- (7) /BULKBU/ requires 5333 words,
/MUCHO/ requires 1946 words.

```

1 > SAMPLE PROBLEM - DRIVER INPUT
2 > 1 2 0 1 1
3 >"% +51
4 > SAMPLE PROBLEM - ANISN FLUX DATA
5 >+10
6 >15$$ 1 0 1 4 1 1 0 16 16 0 8 3 4 11 120 0 7 37 1 0 1 0 0 50
7 > 4 0 4 1 1 0 0 0 4 1 3 0
8 >16** 2R0.0 0.00001 4R0.0 1.0 0.0 0.5 0.00001 3R0.0 T
9 >13$$ 21 22 33 34 93 94 95 T
10 >17** 1.0 F0.0 T
11 >3** 16R1.0 F0.0
12 > T
13 >1** 8R0.0
14 >4** 15I0.0 80.0
15 >5** 8R1.0
16 > 6* 0.0 0.1666667 0.3333333 0.3333333 0.1666667
17 > 7* -.9367418 -.9688903 -.3500212 0.3500212 0.8688903
18 >3$$ 14I1 16
19 >9$$ 2R8 10
20 > 12 14 16 18 20 22 24 26 28 30
21 > 32 34 36
22 >10$$ 4R8 4R9 4R10 4R11 4R12 4R13 4R14 4R15 4R16 4R17 4R18
23 > 4R19 4R20 4R21 4R22 4R23 4R24 4R25 4R26 4R27 4R28 4R29
24 > 4R30 4R31 4R32 4R33 4R34 4R35 4R36 4R37
25 >11$$ 0 6 1 3 0 7 2 4 14Q8
26 >"% +21X 12**
27 >12** 0.0 0.0345 0.00001 0.0139 29Q4
28 >"% +51X
29 >19$$ 16R1
30 >21** 1.0E-15 F1.0
31 >12** 0.0 0.0345 0.00001 0.0139 29Q4
32 >"% +51X
33 >19$$ 16R1

```

```

34 >21** 1.0E-15 F1.0
35 >22$$ -1 -3 -5 -6
36 >23$$ 1 1 3 1
37 >"% +21X
38 >38**
39 >0.0 15R0.00001 0.0 15R0.0139 0.0 15R0.0345 1Q16
40 >"% +51X
41 >29$$ 1 7 3Q2
42 >T
43 >CASE END
44 > SAMPLE PROBLEM - ANISN ADJOINT DATA
45 >15$$ 1 1 1 4 1 1 0 15 16 0 8 3 4 11 120 0 7 37 1 0 1 0 0 50
46 > 4 0 0 0 1 0 0 0 4 1 3 0
47 >16** 2R0.0 0.00001 4R0.0 0.0 0.0 0.5 0.00001 3R0.0 T
48 >13$$ 21 22 33 34 93 94 95 T
49 >"% +21X
50 >17**
51 >0.0 15R0.010257
52 >0.0 15R0.012159
53 >0.0 15R0.0009903
54 >0.0 15R0.0007638
55 >0.0 15R0.002413
56 >0.0 15R0.007080
57 >0.0 15R1.28966
58 >16R0.0
59 >"% +51X
60 >T
61 >3** 16R1.0 F0.0
62 > T
63 >1** 8R0.0
64 >4** 15I0.0 80.0
65 >5** 8R1.0
66 > 6* 0.0 0.1666667 0.3333333 0.3333333 0.1666667

```

```

67 > 7*          -.9367418  -.8688903  -.3500212  0.3500212  0.8688903
68 >8$$ 14I1 16
69 >9$$ 2R8 10
70 >      12 14 16 18 20 22 24 26 28 30
71 >      32 34 36
72 >10$$ 4R8 4R9 4R10 4R11 4R12 4R13 4R14 4R15 4R16 4R17 4R18
73 > 4R19 4R20 4R21 4R22 4R23 4R24 4R25 4R26 4R27 4R28 4R29
74 > 4R30 4R31 4R32 4R33 4R34 4R35 4R36 4R37
75 >11$$ 0 6 1 3 0 7 2 4 14Q8
76 >"% +21X      12**
77 >T
78 >END OF ANISN CASES
79 >"% +52
80 >FFFFFFFFF0444400000 SAMPLE PROBLEM - SWIF DATA
81 >+10
82 >1$$ 1 1 1 2 2 0 -2 1 -1 F0
83 >2** .00001 1.0 F0.0 T
84 >3$$ 0
85 >4$$ 1 T
86 >5$$ 2
87 >6$$ 751
88 >7$$ 21 22 94 95
89 >8** 0.0 15R0.0927 0.0 15R0.0345
90 >9** 0.0 15R0.0927 0.0 15R0.0345
91 >10** F0.0
92 >"% +22X
93 >11** 0.0 15R1.1E-5 0.0 15R0.0345
94 >"% +52X
95 >12** 0.1
96 >51** 0.05
97 > T
98 >48$$ 1 0 2 2
99 > T

```


9. PROGRAMMER'S GUIDE

All the routines in SWAN have been written in IBM-360 FORTRAN for the H-level compiler. The actual compilations have been made with the H-Extended-Plus Compiler for the 360/91, but this was done only to produce more efficient object code.

Memory Size Requirements

The variable array dimensioning scheme used in ANISN to minimize memory requirements has been carried over to SWIF. By overlaying the ANISN common/BULKBU/with the SWIF common/MUCHO/it has been found that SWAN requires no larger memory than ANISN alone.

The complete overlay structure for SWAN is shown in Fig. 2 below. It consists of a root segment containing the DRIVER program, a first level overlay containing the ANISN or SWIF control programs and common, and second level overlays containing the routines for the various options within ANISN or within SWIF. All routines not explicitly located in an overlay are placed in the root segment.

Fig. 2. The overlay structure of SWAN.

The memory size required to execute SWAN with this overlay structure on the IBM/360 has been approximately (in bytes) the larger of $(102,000 + 4 \times \text{words in /BULKBU/})$ or $(99,000 + 4 \times \text{words in /MUCHO/})$, plus I/O Buffers. The sizes of /BULKBU/ and /MUCHO/ are set by the declarations in the subroutines ANISN and SWIF, respectively. It is up to the user to run only those cases which fit his compiled version of SWAN.

The number of words required in /BULKBU/ by a given set of data will be approximately the larger of:

$$\begin{aligned} & \text{(a) } (MCR + MTP) * IHM * \begin{cases} IGM & \text{if } ID2 = 0 \\ \text{else } 1 \end{cases} \\ \text{or (b) } & [IM * (JT + 1) + MM] * \begin{cases} IGM & \text{if } IDAT1 \neq 2 \\ \text{else } 1 \end{cases} \\ & + [IHM * MT + IPQM] * \begin{cases} IGM & \text{if } IDAT1 = 0 \\ \text{else } 1 \end{cases} \\ & + [ISCT + 3 * JT + ID3 + 6 * MM + 14] * IM \end{aligned}$$

where $IPQM = \begin{cases} IPM * MM & \text{if } IPM \neq 0 \\ \text{or } IM & \text{if } IQM \neq 0 \end{cases}$ and the other symbols are defined in the ANISN input section.

Similarly, the number of words needed for /MUCHO/ will be approximately:

$$\begin{aligned} & 500 + (8 * ISK + 4) * IZM \\ & + [(15 + ID3) * ISK + 4] * IM \\ & + [(ISA + 1) * JT * IM + (ISCT + 1) * IHM] * IGM \end{aligned}$$

The exact number of words required for a particular case will be indicated in the ANISN and SWIF output. If the approximate formulas given above are not satisfactory, these exact sizes may be found ahead of time by running the auxiliary program "SIZERS" and supplying as input on unit 5 the following data in NAMELIST format:

&PARAMS	INF = n,	ISK = n,	ISD = n,
ISA = n,	ISAS(1) = n,	...	ISAS(ISA) = n,
ISCT = n,	ISN = n,	IGE = n,	
IBL = n,	IBR = n,	IZM = n,	IM = n,
IGM = n,	IHT = n,	IHS = n,	IHM = n,
MS = n,	MCR = n,	MTP = n,	MT = n,
IDFM = n,	IQM = n,	IPM = n,	
ID2 = n,	ID3 = n,	ID4 = n,	
IDAT1 = n,	IDAT2 = n,	IFG = n,	
IPRT = n,	&END		

The parameters have all been described in the ANISN and SWIF input sections. More than one such namelist may be input in one run of SIZERS.

INPUT/OUTPUT UNITS

At least 14 I/O unit numbers are used by SWAN, as indicated in Fig. 1. Additional units may be specified to satisfy input redirection requests (X or Y options) by the PPL-FIDO card scanner. Format descriptions and IBM/360 JCL examples of the basic units will be given below.

Unit 5: This is the input unit from which the basic data cards are read. It is assumed to supply 80-column card images.

```
//FT05F001 DD *
```

Unit 6: This is the printed output unit, and is assumed to be a line printer accepting a carriage control character and 132 printing columns.

```
//FT06F001 DD SYSOUT = A
```

Unit 7: This is the punched output unit, and is assumed to punch 80-column cards.

```
//FT07F001 DD SYSOUT = B
```

Units 21, 22, 51, & 52: These are the ANISN and SWIF data input units. They (and others, if FIDO redirection is used) are written and read as 80-column card images. They must be rewindable, and have enough capacity to hold the ANISN and SWIF data cards.

```
//FT21F001 DD Unit = SYSDA, SPACE = (TRK,(5,5)),  
DCB = (RECFM = FB, LRECL = 80,  
BLKSIZE = 7280)
```

Units 4 and 12: These are the cross section libraries XSECLIB and ACTVLIB. They must be produced by an auxiliary program, and should have structures appropriate to FORTRAN "unformatted" reads and writes.

```
//FT04F001    DD    DISP = (SHR, KEEP), DSN = XSECLIB,  
                UNIT = 2314, VOL = SER = PPL789,  
                DCB = (RECFM = VBS, LRECL = 7290,  
                BLKSIZE = 7294)
```

Units 3 and 30: These are the XTROUT and SWFOUT files which receive the new output of ANISN and SWIF as a series of FORTRAN "unformatted" writes. Their contents and required sizes are described later in this section.
(Unit 30 is not required if SAVE = 0.)

```
//FT03F001    DD    DISP = (NEW, PASS), DSN = XTROUT,  
                UNIT = 2314, VOL = SER = PPL789,  
                DCB = (RECFM = VBS, LRECL = 7290,  
                BLKSIZE = 7294)
```

Units 1, 2, 8, & 9: These are scratch units required by ANISN (but not SWIF). They are written and read "unformatted."

```
//FT01F001    DD    UNIT = SYSDA, SPACE = (TRK, (100,100)),  
                DCB = (RECFM = VBS, LRECL = 7290,  
                BLKSIZE = 7294).
```

XTROUT

The ANISN XTROUT option, triggered by adding in 4 when computing ID1, dumps out most of the ANISN input and the raw computed fluxes. The successive dumps from stacked cases follow each other in the XTROUT file. The following parameters not defined in the ANISN input are needed to describe the output format:

IP	= number of points = IM + 1	
IHP	= number of positions in internal cross section tables = IHM + 1 if there is upscattering; else = IHM	
MM	= number of discrete angles = ISN + 1 for slab or sphere; = ISN* (ISN + 4)/4 for cylinder	
JT	= 1 if ISCT = 0; else = number of Legendre moments: ISCT for slab or sphere; ISCT* (ISCT + 4)/4 for cylinder	
MP	= 1 if IPM = 0; = MM if IPM ≠ 0	
IMP	= IM if IQM > 0; = IPM if IPM > 0	
DAT	= date in A8 format: MM/DD/YY	
TOD	= time of day in A8 format: HH.MM.SS.	
KS	= MS, or = 1 if MS = 0	Be very careful if
KID	= ID3, or = 1 if ID3 = 0	MS or MTP or ID3 = 0:
KTP	= MTP, or = 1 if MTP = 0	dummy arrays will be written

The following FORTRAN-like statements describe the output:

1.	WRITE (3)	T,DAT,TOD		T(12) is the problem title
2.	"	KFIX(36)	15\$	integer constants
3.	"	FLT(14)	16*	real constants
4.	"	XKI(IGM)	1*	fission spectrum
5.	"	FD(IM)	2*	fission density guess
6.	"	R(IP)	4*	radii
7.	"	VE(IGM)	5*	velocities
8.	"	W(MM)	6*	quadrature weights
9.	"	DSN(MM)	7*	quadrature cosines
10.	"	MA(IM)	8\$	zone numbers
11.	"	MZ(IZM)	9\$	material numbers
12.	"	MB(KS)	10\$	mixing numbers
13.	"	MC(KS)	11\$	component numbers
14.	"	XMD(KS)	12\$	component densities
15.	"	MTT(KTP)	13\$	ID'S of materials on tape
16.	"	J5(IZM)	19\$	order of scatter
17.	"	RM(IZM)	20*	radius modifiers
18.	"	DF(IM)	21*	density factors
19.	"	J3(KID)	22\$	materials for activities
20.	"	J4(KID)	23\$	positions for activities
21.	"	IGT(IGM)	24\$	diffusion markers
22.	"	ART(IGM)	25*	right albedo
23.	"	ALFT(IGM)	26*	left albedo
24.	"	IFGP(5)	27\$	few group parameters

25.	WRITE(3)	IFGG(IGM)	28\$	few group numbers
26.	"	J6(2,KID)	29\$	activity group limits
27.	"	V(IM)	30*	interval volumes
28.	"	AA(IP)	31*	boundary areas
29.	"	WD(MM)	32*	weight*cosine
30.	"	MR(MM)	33\$	index of reflection angle
31.	"	PNC(MM,JT)	34*	Legendre coefficients
32.	"	CF(IM,KID)	38*	activity densities

DO 1 J = 1, IGM

WRITE(3) (XN(I,J), I = 1, IM) scalar flux

WRITE(3) (B(M,J), M = 1, MM) boundary flux

WRITE(3) ((XND(I,M,J), I = 1, IP),
M = 1, MM) angular flux

IF (ISCT.EQ.0) GO TO 1

WRITE(3) ((XJ(I,L,J), I = 1, IM),
L = 1, JT) Legendre components

1 CONTINUE

DO 2 J = 1, IGM

WRITE(1 or 2)

& ((CRX(I,J,K), I = 1, IHP), K = 1, MT) cross sections

IF((IQM*IM + IPM*MM).EQ.0) GO TO 2

WRITE (1 or 2)

& ((Q(M,I,J), M = 1, MP), I = 1, IMP) source

2 CONTINUE

RETURN

SWFOUT

The SWIF SWFOUT option, triggered by setting SAVE = 1, dumps out most of the SWIF input and output. The following parameters not defined in the ANISN or SWIF input are needed to describe the output format:

IP = number of points = IM + 1

ISJ = ISK - 1

ISCT1 = order of scattering X-Sect expansion + 1 = ISCT + 1

ISASUM = sum of entries in array ISAS (4\$)

KID = ID3, or 1 if ID3 = 0

DAY = date in A8 format: MM/DD/YY

TIME = time of day in A8 format: HH.MM.SS

JT = 1 if ISCT = 0; else = number of Legendre moments:

ISCT for slab or sphere;

ISCT * (ISCT + 4) / 4 for cylinder

The following FORTRAN-like statements describe the output:

1. WRITE(30) SWTITL(15),DAY,TIME SWTITL is title card for SWIF
2. " D(1),LIM1,INTAN(36) LIM1 is length of MUCHO
INTAN integer constants (ANISN 15\$)
3. " INT(12) 1\$ integer constants
4. " FLT(6) 2* floating point constants
5. " SCOMON(18) common block SWIFCM
6. " KID = ID3, or 1 if ID3 = 0
7. " ICOMON(9) common block IDSWIF
8. " DCOMON(3) common block DRIVE
9. " I01(INF) 3\$ functional type
10. " ISAS(ISA) 4\$ number of materials for each S^+
11. " IIS(ISASUM) 5\$ sequence number for above materials
12. " LAS(ISASUM) 6\$ ACTVLIB ID for above materials
13. " LID(ISK,ISCT1) 7\$ XSECLIB ID for ISK materials
14. " DOMZ(ISM,ISK) 8* max possible densities, by zone
15. " DMX(IZM,ISK) 9* max permissible densities
16. " DMN(IZM,ISK) 10* min permissible densities
17. " DP(IZM,ISD) 11* actual densities
18. " AA(ISJ) 12* density modification coefficients
19. " WF(IM,ISD) 13* weight coefficients
20. " EPS(ISJ,INF) 51* convergence parameters
21. " ACTOR(KID) 48\$ sequence number for 38* array
22. " V(IM) volume by intervals
23. " XN(IM,IGM) scalar flux
24. " XJ(IM,IGM,JT) flux Legendre components

```
25. WRITE(30)  XNA(IM,IGM,ISA)      adjoint fluxes
26.   "        XJA(IM,IGM,JT,ISA)    adjoint Legendre components
27.   "        AC(IM,ISK,ISA)        AC component of eff. func.
28.   "        BC(IM,ISK,ISA)        BC component of eff. func.
29.   "        CC(IM,ISK,ISA)        CC component of eff. func.
30.   "        PER(8,ISK,INF)        perturbation integrals
31.   "        H(IP,ISD,INF)         perturbation parameters
32.   "        HR(IM,ISJ,INF)        SEF by intervals
33.   "        BB(ISJ)               Bi coefficients
34.   "        ACI(IM,ISK,ISA)       isotropic component of AC
35.   "        BCI(IM,ISK,ISA)       isotropic component of BC
36.   "        VZ(IZM)               volume by zones
37.   "        HRZ(IZM, ISJ, INF)    SEF by zone
38.   "        DOM(IM, ISK)          max possible density by intervals
39.   "        RX(IZM,ISK)           density limits indicators

      IF(IEF.NE.0) GO TO 1
40. WRITE(30)  DN(IZM,ISD)          new density distribution
      1  CONTINUE
```

SWIF Subroutines

The principal functions of the various SWIF subroutines are summarized below:

SWIF: Allocates the maximum size for MUCHO and initializes MUCHO arrays to zero. Rewinds I/O units and reads output option tables and title information (included in the SWIF title card). Calls SETUP.

SETUP: Reads ANISN 15\$ array (integer parameters) from XTROUT and SWIF input arrays 1\$, 2*, 3\$, and 4\$. Computes the relative addresses of MUCHO arrays and the total size required for this common block. Reads in the rest of SWIF input data. Calls UNIT3 (to read that part of SWIF input taken from XTROUT) and reads in cross sections from ACTVLIB (required for the adjoint source terms). Transfers control to EFFECT. For optimization problems (IEF = 0), SETUP calls NEWDEN. If contents of variable input files are to be changed (IMFL = ± 1), NEWRAY is called. SETUP also controls the preparation of the SWFOUT file (SAVE = 1).

UNIT 3: Reads that part of XTROUT (including ANISN input and output data) required as input for SWIF.

EFFECT: This is the heart of SWIF. It calculates the effectiveness functions and the perturbation parameters. It also reads the cross section tables from XSECLIB (material by material).

NEWDEN: Is the central subroutine responsible for defining the new density distributions in the iterative optimization procedure. It performs convergence tests and calculates the new density distributions using one of the several options specified

in the input. If the new densities exceed the permissible limits, NEWDEN adjusts the density distribution.

NEWRAY: Controls the preparation of the density-dependent input arrays for ANISN and SWIF, using the new density distribution defined by NEWDEN. NEWRAY can also be used to prepare the variable input arrays using the fixed arrays input (the IMFL = -1 option). The input arrays to be prepared by NEWRAY, as well as their output forms, are specified on the SWIF title card.

DENTAB: Sets the variable array 38* of ANISN and 11* of SWIF.

SOURCE: Sets the source terms (17*) for the ANISN adjoint runs.

MIX: Sets the densities in the mixing table of ANISN input (array 12*).

PUFIDO: Produces punched FIDO type images on the disk.

SWFOUT, INPOUT, EFFOUT, and NEWOUT: Are used to prepare I/O unit SWFOUT (if SAVE = 1). SWFOUT dumps SWIF input parameters. INPOUT dumps input arrays read into SWIF as FIDO input. EFFOUT dumps the arrays used in and set by EFFECT. NEWOUT dumps the new densities set by NEWDEN.

10. CALCULATIONAL PROCEDURES

This section summarizes the basic expressions used by SWAN for computing selected quantities. The energy variable is treated in the group formulation.

Effectiveness Functions

The effectiveness function is defined in Eq. (8). In formal notation this function can be written as follows:

$$e(\underline{r}) = -AC(\underline{r}) + BC(\underline{r}) + CC(\underline{r}) \quad (13)$$

where

$$AC(\underline{r}) \equiv \int_0^\infty dE \int_{4\pi} d\Omega \phi^+(\underline{r}, E, \Omega) \sigma_t(E) \phi(\underline{r}, E, \Omega) \quad (14)$$

$$BC(\underline{r}) \equiv \int_0^\infty dE \int_{4\pi} d\Omega \int_0^\infty dE' \int_{4\pi} d\Omega' \phi^+(\underline{r}, E', \Omega') \sigma_s(E \rightarrow E', \Omega \rightarrow \Omega') \phi(\underline{r}, E, \Omega) \quad (15)$$

$$CC(\underline{r}) \equiv \int_0^\infty dE \sigma_+(E) \phi(\underline{r}, E) \quad (16)$$

and $\sigma_+(E)$ is used for defining the adjoint source term [Eq. (6)]:

$$S^+(\underline{r}, E) = \sum_i N_i(\underline{r}) \sigma_{+,i}(E) \quad (17)$$

In the above the subscripts for the type of functional (b) and for the material (i) are omitted [excluding Eq. (17)].

The angular flux and adjoint distributions are now expanded in the spherical harmonics $Y(\underline{\Omega})$:

$$\phi(\underline{r}, E, \underline{\Omega}) = \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \phi_{\ell m}(\underline{r}, E) Y_{\ell m}(\underline{\Omega}) \quad (18)$$

$$\phi^+(\underline{r}, E, \underline{\Omega}) = \sum_{k=0}^K \sum_{n=-k}^k \phi_{kn}^+(\underline{r}, E) Y_{kn}(\underline{\Omega}) \quad (19)$$

and using the orthonormality property:

$$\int_{4\pi} d\underline{\Omega} Y_{\ell m}(\underline{\Omega}) Y_{\ell' m'}^*(\underline{\Omega}) = \delta_{\ell \ell'} \delta_{m m'} \quad (20)$$

and the relation:

$$Y_{\ell m}(\underline{\Omega}) = (-1)^m Y_{\ell, -m}^*(\underline{\Omega}) \quad (21)$$

the multigroup expression for AC can be written as follows:

$$AC(\underline{r}) = \sum_{g=1}^G \sigma_t^g \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} (-1)^m \phi_{\ell m}^g(\underline{r}) \phi_{\ell-m}^{+g}(\underline{r}) \quad (22)$$

where the flux moments are defined by:

$$\phi_{\ell m}^g(\underline{r}) \equiv \int_{4\pi} d\underline{\Omega} \phi^g(\underline{r}, \underline{\Omega}) Y_{\ell m}^*(\underline{\Omega}) \quad (23)$$

The multigroup expression used for the calculation of the BC term is:

$$BC(\underline{r}) = \sum_{\ell=0}^L \frac{4\pi}{(2\ell+1)} \sum_{m=-\ell}^{\ell} (-1)^m \sum_{g=1}^G \sum_{h=g}^G \phi_{\ell m}^g(\underline{r}) \sigma_{sl}^{g \rightarrow h} \phi_{\ell-m}^{+h}(\underline{r}) \quad (24)$$

In deriving this equation the differential scattering cross section has been expanded into a Legendre polynomial series:

$$\sigma_s^{g \rightarrow h}(\mu) = \sum_{\ell=0}^L \sigma_{s\ell}^{g \rightarrow h} P_{\ell}(\mu) \quad (25)$$

where

$$\sigma_{s\ell}^{g \rightarrow h} = \frac{2\ell+1}{2} \int_{-1}^1 \sigma_s^{g \rightarrow h}(\mu) P_{\ell}(\mu) d\mu \quad (26)$$

and

$$\int_{-1}^1 P_{\ell}(\mu) P_{\ell'}(\mu) d\mu = \frac{2}{2\ell+1} \delta_{\ell\ell'} \quad (27)$$

Use is also made of the spherical harmonics addition theorem:

$$P_{\ell}(\mu) = \frac{4\pi}{2\ell+1} \sum_{m=-\ell}^{\ell} Y_{\ell m}^*(\underline{\Omega}') Y_{\ell m}(\underline{\Omega}) \quad (28)$$

where μ is $\underline{\Omega} \cdot \underline{\Omega}'$, the cosine of the polar angle between the two directions $\underline{\Omega}$ and $\underline{\Omega}'$.

The multigroup expression for the CC term [Eq. (16)] is:

$$CC(\underline{r}) = \sum_{g=1}^G \sigma_+^g \phi^g(\underline{r}) \quad (29)$$

where $\phi^g(\underline{r})$ is the scalar flux.

The moments of the flux and of the adjoint distributions $[\phi_{\ell m}^g(\underline{r})$ and $\phi_{\ell m}^{+g}(\underline{r})]$ are calculated routinely by ANISN and are transferred from ANISN to SWIF via the XTROUT file. The ANISN adjoint moments are inverted in SWIF before being used for the calculation of the effectiveness functions. The inversion,

both in the energy and in the angular variables, is done as follows:

$$\phi_{lm}^{+g}(r) = (-1)^{l+m} [\phi_{lm}^{+(G-g+1)}]_{\text{ANISN}} \quad (30)$$

This inversion produces the physically meaningful adjoint.

The cross sections σ_t^g and σ_{sl}^{g+h} are the standard ANISN cross sections. The cross sections σ_+^g for the calculation of the adjoint source term are to be selected in accordance with the physical meaning of the adjoint solution. They are to be obtained from ACTVLIB.

The substitution effectiveness function is obtained from the effectiveness functions [Eq. (13)] according to its definition in Eq. (10).

Perturbation Parameters

SWIF calculates (subroutine EFFECT) the following perturbation parameters (listed in the order in which they are printed out):

(a) The variation in the value of the functional due to 1% increase in the atomic density in the interval:

$$\int_{\text{interval}} d\underline{r} e_i(\underline{r}) N_i(\underline{r}) / 100, \quad i = 1, 2, \dots, I, I + 1 \dots J \quad (31)$$

where J is the total number of materials that contribute to the functionals (ISD) and I is the number of materials with variable concentration (ISK). The last row (interval "IM + 1") in the tables gives the total contribution of the material to the

reaction rate of interest:

$$\int d\underline{r} N_i(\underline{r}) \sum_{g=1}^G \sigma_{+,i}^g \phi^g(\underline{r}) \quad (32)$$

A summation of Eq. (32) for all the materials gives the value of the nucleonic characteristic as calculated using the reaction-rate formulation.

Zones for which $N_i^0 = 0$ (DOMZ = 0) do not contribute to the perturbation parameters. For the purpose of perturbation calculations it is allowable to define fictitious densities.

(b) The integrated (total) effectiveness function:

$$P_1 = \int d\underline{r} e_i(\underline{r}) \quad (33)$$

A similar quantity is printed also as the last row (interval IM + 1) of the Effectiveness Function (H) print out. The latter value, however, includes also contribution from intervals for which $N_i^0 = 0$.

$$(c) \quad P_3 = \int d\underline{r} e_i(\underline{r}) N_i(\underline{r}) / 100 \quad , \quad i = 1, 2, \dots, I \quad (34)$$

(d) The same as (c) but including contributions from zones for which $e_i(\underline{r}) > 0$.

$$(e) \quad P_2 = \int d\underline{r} Q_i(\underline{r}) \quad i = 1, 2, \dots, I-1 \quad (35)$$

$$(f) \quad P_4 = \int d\underline{r} Q_i(\underline{r}) N_i(\underline{r}) / 100 \quad i = 1, 2, \dots, I-1 \quad (36)$$

(g) The same as (f) but including contributions from $Q_i(\underline{r}) > 0$ only.

(h) Maximum possible functional variation due to the substitution of a given material for the I^{th} material. For a maximum problem:

$$\begin{aligned}
 P_4 = & \int d\underline{r} Q_i(\underline{r}) [N_i^{\max}(\underline{r}) - N_i(\underline{r})] && \text{for } Q_i(\underline{r}) > 0 \\
 & + \int d\underline{r} |Q_i(\underline{r})| [N_i(\underline{r}) - N_i^{\min}(\underline{r})] && \text{for } Q_i(\underline{r}) < 0 \\
 & && i = 1, 2, \dots, I-1
 \end{aligned} \quad (37)$$

(i) Total possible variation in the functional

$$P_8 = \int d\underline{r} Q^*(\underline{r}) \Delta N^*(\underline{r}) \quad (38)$$

where $Q^*(\underline{r}) \Delta N^*(\underline{r})$ is selected for each zone from among all the $I-1$ isotopes to give the maximum contribution to the functional variation at $\underline{r}$.

Determination of New Density Distribution

There are four major steps in the determination of the new density distribution: Determination of the A_i coefficients, determination of the B_i coefficients, the calculation of the new density distributions [see Eq. (12)], and readjustment of the new density distributions if they have overflowed the specified density limits.

There are two basic options for the definition of the coefficients A_i and B_i :

(1) The B_i coefficients are independent of each other. This is the approach of Abagyan et al. (For details see Ref. [1]).

(2) There are different A_i and B_i coefficients for each material but the relation $B_i/B_j = A_i/A_j$ is satisfied for all the materials.

Option 1 (IBO = 0)

The density variation at the n^{th} iteration is calculated from the expression:

$$\delta N_i^n(\underline{r}) \equiv N_i^n(\underline{r}) - N_i^{n-1}(\underline{r}) = A_i^n Q_{e,i}^{n-1}(\underline{r}) + B_i^n Q_{c,i}^{n-1}(\underline{r}) \quad (39)$$

With $\delta N_i^n(\underline{r})$ of Eq. (39) the variation of the restraint becomes:

$$\delta F_c^n = \sum_{i=1}^{I-1} \int d\underline{r} [A_i^n Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r}) + B_i^n Q_{c,i}^{n-1}(\underline{r}) Q_{c,i}^{n-1}(\underline{r})] \quad (40)$$

The coefficients B_i^n are determined to give $\delta F_{c,i}^n = 0$, that is:

$$B_i^n = -A_i^n [\int d\underline{r} Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r})] / [\int d\underline{r} Q_{c,i}^{n-1}(\underline{r})^2] \quad (41)$$

$$i = 1, 2, \dots, I-1$$

For problems with no restraints the B_i^n coefficients are set to zero.

The three options for the calculation of the A_i^n coefficients (in the fourth option, ITA = N, these coefficients are used as specified in the input), are:

(a) ITA = 0

Inserting B_i^n [Eq. (41)] back into Eq. (39) one gets:

$$\delta N_i^n(\underline{r}) = A_i^n \left[Q_{e,i}^{n-1}(\underline{r}) - \frac{\int d\underline{r} Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r})}{\int d\underline{r} [Q_{c,i}^{n-1}(\underline{r})]^2} Q_{c,i}^{n-1}(\underline{r}) \right] \quad (42)$$

The A_i^n coefficients are calculated by SWAN using Eq. (42) as follows:

$$A_i^n = a_i^n / \left[Q_{e,i}^{n-1}(r_m) - \frac{\int dr Q_{c,i}^{n-1}(r) Q_{e,i}^{n-1}(r)}{\int dr [Q_{c,i}^{n-1}(r)]^2} Q_{c,i}^{n-1}(r_m) \right] \quad (43)$$

where r_m designates the zone in which the density change will be the maximum (in absolute value) and a_i^n are the input data specified in the 12* array. The value of a_j^n should be the maximum density change desired for material i at the n^{th} iteration (in the units used for the material densities).

(b) ITA = -1

With Eq. (42) for the density variation, the total variation in the functional to be extremized becomes:

$$\delta F_e^n = \sum_{i=1}^{I-1} A_i^n \left(\int dr [Q_{e,i}^{n-1}(r)]^2 - \frac{[\int dr Q_{c,i}^{n-1}(r) Q_{e,i}^{n-1}(r)]^2}{\{\int dr [Q_{c,i}^{n-1}(r)]^2\}} \right). \quad (44)$$

The program assumes for ITA = -1 and ITA = -2 that $A_i^n = A^n =$ constant. This coefficient is calculated as follows:

$$A^n = a_1^n / \sum_{i=1}^{I-1} \left(\int dr [Q_{e,i}^{n-1}(r)]^2 - \frac{[\int dr Q_{c,i}^{n-1}(r) Q_{e,i}^{n-1}(r)]^2}{\{\int dr [Q_{c,i}^{n-1}(r)]^2\}} \right). \quad (45)$$

The value of a_1^n should be the absolute change desired for the functional (to first order) to be extremized, at the n^{th} iteration. It should be given in absolute value and in the units used for the functional. The rest of the a_i^n ($i \neq 1$) entries of the 12* array are disregarded.

(c) ITA = -2

In this option the A^n of Eq. (45) is multiplied by P_8 [Eq. (38)]: the maximum possible variation of the functional to be extremized (in the direction of its optional value) as predicted by first order perturbation theory. The value of a_1^n for this option should be the fraction of the maximum possible functional variation.

Option 2 (IBO = 1)

This option assumes that

$$A_i^n/A_1^n = B_i^n/B_1^n = \eta_i^n \quad (46)$$

With these relations the equations corresponding to Eqs. (40) - (45) are revised as follows:

$$\delta F_c^n = A^n \sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r}) + B^n \sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} [Q_{c,i}^{n-1}(\underline{r})]^2 \quad (47)$$

where $A^n \equiv A_1^n$ and $B^n \equiv B_1^n$.

$$B_i^n = -A_i^n \sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r}) / \sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} [Q_{c,i}^{n-1}(\underline{r})]^2 \quad (48)$$

$$i = 1, 2, \dots, I-1$$

$$\delta N_i^n(\underline{r}) = A^n \left(\eta_i^n Q_{e,i}^{n-1}(\underline{r}) - \frac{\sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r})}{\sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} [Q_{c,i}^{n-1}(\underline{r})]^2} \eta_i^n Q_{c,i}^{n-1}(\underline{r}) \right) \quad (49)$$

$$i = 1, 2, \dots, I-1$$

(a) ITA = 0

$$A_i^n = a_1^n / \eta_i^n \left(Q_{e,i}^{n-1}(\underline{r}_m) - \frac{\sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r})}{\left\{ \sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} [Q_{c,i}^{n-1}(\underline{r})]^2 \right\}} Q_{c,i}^{n-1}(\underline{r}_m) \right). \quad (50)$$

The coefficients a_i^n are to be specified by the user so that $\eta_i^n = a_i^n / a_1^n$.

(b) ITA = -1

$$\delta F_e^n = A^n \left(\sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} [Q_{e,i}^{n-1}(\underline{r})]^2 - \frac{\sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r})^2}{\left\{ \sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} [Q_{c,i}^{n-1}(\underline{r})]^2 \right\}} \right). \quad (51)$$

$$A^n = a_1^n / \left(\sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} [Q_{e,i}^{n-1}(\underline{r})]^2 - \frac{\sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} Q_{c,i}^{n-1}(\underline{r}) Q_{e,i}^{n-1}(\underline{r})^2}{\left\{ \sum_{i=1}^{I-1} \eta_i^n \int d\underline{r} [Q_{c,i}^{n-1}(\underline{r})]^2 \right\}} \right). \quad (52)$$

(c) ITA = -2

The same procedure employed for the IBO = 0 option is used also here.

ACKNOWLEDGMENT

This work was supported by U. S. Atomic Energy Commission Contract AT(11-1)-3073; use was made of computer facilities supported in part by National Science Foundation Grant NSF-GP579; support was also received from the Edison Electric Institute Project RP 113.

REFERENCES

- [1] E. Greenspan, "A Method for the Optimization of Fusion Reactor Neutronic Characteristics," Proceedings of the Topical Meeting on Mathematical Models and Computation Techniques for Analysis of Nuclear Systems (American Nuclear Society, Hinsdale, Illinois, 1973), to be published. Also issued as MATT-981 (Princeton Plasma Physics Laboratory) 1973.
- [2] W. W. Engle, Jr., "A User's Manual for ANISN, A One-Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering," Oak Ridge Gaseous Diffusion Plant Report K-1693 (1967). This report is included in Ref. [3].
- [3] Radiation Shielding Information Center, "ANISN, a One-Dimensional Discrete Ordinates Transport Code," Computer Code Collection CCC-82, Oak Ridge National Laboratory (1972).

APPENDIX

PPL-FIDO is an input processing routine which allows large amounts of data to be specified in a very compact notation. It is a revision of the FIDO subroutine written for ANISN by W. W. Engle, with improved internal processing and an extended set of data codes. It is used in SWAN to read the ANISN input (initially from unit 51) and the SWIF input (initially from unit 52). Cards are read and scanned when PPL-FIDO is called; it prints various messages on unit 6 to indicate problems and progress; and it returns when a code "T" is encountered in the data stream. A full listing of the cards read may be obtained by using the "+10" option.

Each card image is scanned and interpreted as a number of data "fields," each with three "subfields": I, an integer; C, a code character; and V, a real or integer value. Each field may specify one or more data entries.

The data are read into arrays identified by code number. The user, through particular code characters, must specify the array numbers, the data type (real or integer), and the scan mode (fixed-format or free-format). Each array must be filled (with data or skips for every element) before a new array may be started. A code "T" or an end-of-file will terminate the scan.

The columns past a quotation mark (") are ignored and may be used for comments. If the " is in column one, the whole card will be printed along with the FIDO comments.

In Fixed-format scan mode the cards are read, six fields per card, with a standard (6(I2,A1,E9.0)) format. Fields which are all blank are ignored.

In Free-format scan mode the cards are read character-by-character, so the subfields need not be aligned to any particular columns. The following rules must be observed.

- 1) Columns 73-80 are ignored.
- 2) The I and C fields must not be separated by blanks.
- 3) The V field must be followed by a blank (or column 73).
- 4) Numbers must not have embedded blanks (e.g., in their exponents).
- 5) Commas are treated like blanks.
- 6) The +, -, and & codes are not recognizable.
- 7) No more than 9 digits can be entered in sequence (in either the fraction or the exponent of a number).
- 8) The "V" field must be given for the codes A, F, I, J, K, L, M, N, P, Q, R, and "blank" (i.e., where a V is ordinarily used).
- 9) The "V" field must not be given for the codes B, C, D, E, G, H, O, S, T, W, X, Y, Z, and \$, *, @, and % (i.e., where a V is ignored if present in fixed format).

If the code character is "+" or "&", then I, C, and V are immediately changed to $V = V * (10. ** (+I))$, C = blank, I = 0. If the code is "-", the subfields are changed to $V = V * (10. ** (-I))$, C = blank, I = 0. The effects of the other codes are described below. In these descriptions, it is assumed that the "array" being filled is the vector A of dimension M, and that A(N) is the next unfilled element. I, C, and V are the three data subfields, and W is used for the next V subfield.

<u>Code</u>	<u>Effect</u>
-------------	---------------

"blank"	if the format code is blank, $A(N) = V$, $N = N + 1$, and I is ignored.
A	address: $N = V$, I is ignored. Any value V is allowed, but a warning is given if $N \leq 0$ or $N > M$
B	variable format: the following card is read immediately with a (20 A4) format and saved for use as the variable format with the W option or for "%" arrays.
C	count: print I and N-1, the number of elements currently filled
D	unformatted unit: if $-100 < I < 100$ and $I \neq 0$, set unit number for unformatted "@" arrays equal to $ I $, and if $I < 0$, this unit is rewind
E	end array: $N = M + 1$, I and V are ignored
F	if $N > M$, a warning is printed if $N \leq M$, elements A(N) through A(M) are set = V, and $N = M + 1$; I is ignored

<u>Code</u>	<u>Effect</u>
G	Input unit after EOF: if $0 < I < 100$, the input unit number is set to I after an end-of-file; otherwise (the initial condition) the scan will terminate after an E-O-F.
H	Hollerith input: the Hollerith mode switch is flipped between off and on. When off, the apostrophe (') is illegal except in column one, where it signifies a comment card to be printed. When on (the initial condition) the apostrophe is recognized as the delimiter for Hollerith input
I	linear interpolation ($I \geq 0$ is valid): I points are inserted between V and W: $A(N) = V$; then as K goes from 1 to I, $A(N + K) = V + (W - V) * K / (I + 1)$; finally $N = N + I + 1$
J	sequence repeat with increment: if either I or V is 0, set $V = V + I$ and $I = 1$, then repeat the sequence $\{A(N - V), \dots A(N - 1)\}$ I times starting at A(N), but increment the values in the sequence by one before each repeat; finally set $N = N + I * V$
K	fixed increment ($I \geq 0$ is valid): as K goes from 1 to I, $A(N + K - 1) = A(N + K - 2) + V$; finally $N = N + I$
L	logarithmic interpolation ($I \geq 0$ is valid): I points are inserted between V and W: $A(N) = V$; then as K goes from 1 to I, $A(N + K) = V * (W/V)^{K/(I + 1)}$; finally $N = N + I + 1$

<u>Code</u>	<u>Effect</u>
M	reversed-sign inverted sequence repeat; if either I or V is 0, set $V = I + V$ and $I = 1$, then repeat the V elements of the sequence $\{A(N - 1), \dots, A(N - V)\}$ I times starting at $A(N)$, and set $N = N + I * V$
N	inverted sequence repeat: if either I or V is 0, set $V = I + V$ and $I = 1$ then repeat the V elements of the sequence $\{A(N - 1), \dots, A(N - V)\}$ I times starting at $A(N)$, and set $N = N + I * V$
O	set the print trigger: if $I > 0$ it is set on; if $I < 0$ it is set off; if $I = 0$ it is flipped. If the trigger is on, input cards are listed; it is initialized off
P	scale factor: for $K = 1$ to $N - 1$, $A(N) = V * A(N)$. I is ignored
Q	sequence repeat: if either I or V is 0, set $V = V + 1$ and $I = 1$, then repeat the V elements of the sequence $\{A(N - V), \dots, A(N - 1)\}$ I times starting at $A(N)$; finally set $N = N + I * V$
R	repeat ($I > 0$ is valid): V is repeated I times starting at $A(N)$, then $N = N + I$
S	skip: $N = N + I$; V is ignored
T	terminate: write out $N - 1$, the number of elements filled in the current array. If $N \neq 1$ and $N \neq M + 1$, an error flag is turned on; if the write flag or the error flag is on, the elements are printed out. I is printed and the data reading terminates

<u>Code</u>	<u>Effect</u>
W	write flag: I is printed out; V is ignored. The write flag, which is turned off whenever a new array is started, is turned on. When the current array processing finishes, its elements will be written out in a standard format if $I \geq 0$, or with the variable format if $I < 0$
X	change input unit: the input logical unit number is changed to I. V is ignored. (Only the last X or Y on a card will be effective.)
Y	change input unit and rewind: the input logical unit number is changed to I, and a REWIND I is executed. V is ignored. (Only the last X or Y on a card will be effective.)
Z	zero: set $A(N)$ through $A(N + I - 1) = 0$ and set $N = N + I$. V is ignored.

Code Effect

\$ new array: first all the processing for the "T" code
\$\$
* except the scan termination is performed. Then the
**
% dimension M for the new array A specified by I is
%%
@ determined, and N is set to 1. The card scan is
@@ resumed in Fixed-format if the code is a single
character or in Free-format if the code is a double
character.

If the code is \$, integer values are read into A.

If the code is *, floating point values are
read into A.

If the code is %, A(1) through A(N) are read
with the variable format specified with the B
option.

If the code is @, A(1) through A(N) are read
"unformatted" from the unit specified with the
D option.

When the Hollerith switch is off, the ' may only appear
in column one to indicate a comment card. When the
switch is on, a pair of apostrophes delimit a string
of characters to be read into A(N), A(N + 1), ...as if
with an (A1) format. (An interior double-prime is
used to actually read an apostrophe.)

? Entering a question mark will elicit the date
of the latest revision of this data processing routine.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission to the extent that such employee or contractor prepares, handles or distributes, or provides access to, any information pursuant to his employment or contract with the Commission.