

ORNL/CSD/TM-9

User's Guide for ICE

S. K. Fraley

BLANK PAGE

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy \$4.00, Microfiche \$2.25

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Energy Research and Development Administration, United States Nuclear Regulatory 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.

Contract No. W-7405 eng 26

COMPUTER SCIENCES DIVISION

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, 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.

USER'S GUIDE FOR ICE

S. K. Fraley

JULY 1976

Work performed for the U.S. Nuclear Regulatory Commission under Interagency Agreement 40-494-75.

NOTICE

This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

UNION CARBIDE CORPORATION, NUCLEAR DIVISION
operating the
Oak Ridge Gaseous Diffusion Plant • Oak Ridge National Laboratory
Oak Ridge Y-12 Plant • Paducah Gaseous Diffusion Plant
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

DISTRIBUTION

TABLE OF CONTENTS

ABSTRACT	v
INTRODUCTION	1
ICE INPUT DATA FORMAT	2
Table IFLAG Options	3
Input/Output Device Specifications	6
REFERENCES	7
APPENDIX A: FIDO (ANISN) Input System	8
Fixed-Field Input	9
Free-Field Input	14
User-Field Input	15

ABSTRACT

ICE is a cross-section mixing code which will accept cross sections from an AMPX working library and produce mixed cross sections in the AMPX working library format, ANISN format, and the group independent ANISN format. User input is in the free-form or fixed-form FIDO structure. The code is operable as a module in the AMPX system.

INTRODUCTION

ICE* is a cross-section mixing code which will accept cross sections from an AMPX¹ working library and produce mixed cross sections in three formats: (1) AMPX working library format, (2) ANISN² format, and (3) group independent ANISN format. User input is in the free-form or fixed form FIDO structure. Section 13 of Ref. 1, which explains the use of the FIDO system, is included as an Appendix to this manual.

The code was developed to allow efficient mixing with minimum user effort and with reduced core storage requirements. The code is operable as a module in the AMPX system.

*Intermixed Cross Sections Effortlessly.

BLANK PAGE

INPUT DATA FORMAT

Card A (20A4)

Title card

FIDO Data Block 1

05 [4]

- | | |
|-----------|--|
| 1. INTAPE | Input tape unit; default 4 |
| 2. IOT1 | AMPX working library output unit; default 3 |
| 3. IOT2 | ANISN output unit*; default 20 |
| 4. IOT3 | ANISN group independent output unit*; default 22 |

*If punched output is designated, the unit will be set to 7.

15 [6]

- | | |
|-------------|--|
| 1. MIX | Number of cross section mixtures to be made |
| 2. NMIX | Number of mixing operations (elements times density operations) to be performed |
| 3. IFLAG(1) | Set greater than zero if AMPX working library output desired. See options in table; default 0 |
| 4. IFLAG(2) | Set greater than zero if ANISN format output desired. See options in table; default 0 |
| 5. IFLAG(3) | Set greater than zero if group independent ANISN output desired. See options in table; default 0 |
| 6. KOPT | Option to be used to calculate χ (chi) of the mixture; default 2 |

KOPT = 0, χ of element with the largest sum of $v(I) \sum_f(I) \phi(I)$ over all groups is used

KOPT = 1, $\bar{\lambda}$ calculated based on an input flux distribution

KOPT = 2, $\bar{\lambda}$ calculated based on a default thermal reactor flux distribution

KOPT = 3, $\bar{\lambda}$ calculated based on a default fast reactor flux distribution

KOPT = 4, $\bar{\lambda}$ calculated based on the fluxes provided in the input tape for each element in the working library

T - Terminate Block I

Table IFLAG Options

Print Options

0	No cross section printing
1	Print 1D cross sections (available only in AMPX working library format)
2	Print 1D cross sections and P_0 cross sections (not available in ANISN group independent format)
3	Complete cross section printing

Output Options

00	No tape or punch output
10	Binary tape output
20	Free-form FIDO punch output (not available in AMPX working library format)
30	Fixed form FIDO punch output (not available in AMPX working library format)

IFLAG should be set to the sum of the print option and output option desired.

FIDO Data Block 2

2\$ [NMIX]

1. (KM(I), I=1, NMIX) Mixture numbers - values range from
1 to MIX

3\$ [NMIX]

1. (KE(I), I=1, NMIX) Element identifiers

4* [NMIX]

1. (RHO(I), I=1, NMIX) Densities

5\$ [MIX]

1. (NCOEF(I), I=1, MIX) Number of Legendre coefficients, in-
cluding P_0 , to be mixed for each
mixture

6* [NG+4]

1. (FLUX(I), I=1, NG) Flux to be used in calculation of χ
if KOPT = 1; default (FLUX(I) = 1.0,
I=1, NG) where NG = number of neutron
groups

NOTE: The four following parameters are only required if KOPT = 2, i.e.,
a standard thermal reactor flux is calculated.

2. TEMP Temperature in °K to be used for Max-
wellian distribution; default 300
3. THETA Temperature in eV to be used for fis-
sion spectrum; default 1.3×10^6
4. AKT Number of KT at which 1/E distribution
and Maxwellian distribution j in;
default 5

5. FCUT Energy in eV at which fission spectrum
and 1/E distribution join; default
 67.5×10^3

8\$ [4]

1. IHT Position of σ_T in ANISN format output;
default 3

2. IFT Position of within group scattering in
ANISN format output; default IHT
plus number of thermal groups

3. IHM Table length in ANISN format output;
default IHS plus full downscatter

4. LOPT Set greater than zero if truncations
of cross sections defined by the
above parameters are to be applied
to the AMPX working library also;
default 0

T - Terminate Data Block 2

FIDO Data Block 3

8\$ [NMIX] Required only if IFLAG(1) > 0

1. (MID(I), I=1, MIX) Mixture ID numbers for AMPX working
library; default (MID(I)=I, I=1, MIX)

9\$ [N] Required only if IFLAG(2) or IFLAG(3) > 0

1. (MAD(I), I=1, N) ANISN mixture ID numbers. NCOEF(I)
ID's are required for each mixture

for a total of $\sum_{I=1}^{\text{mix}} \text{NCOEF}(I)$

Default as follows:

P_0 for 1st mixture	ID=100
P_1 for 1st mixture	ID=101
P_2 for 1st mixture	ID=102
.	.
.	.
P_0 for 2nd mixture	ID=200
P_1 for 2nd mixture	ID=201
.	.
.	.

10\$ [IHT] Required only if IFLAG(2) or IFLAG(3) > 0

1. IMT(I), I=IHT, 1, -1 ENDF-MT numbers of the ID cross sections to be used in the ANISN cross-section tables in inverted order; default first five positions: 1(σ_T), 1452($v\sigma_f$), 27(σ_a), 18(σ_f), and 1018 (χ). All other positions: 0

T - Terminate Data Block 3

Input/Output Device Specifications

The following devices are normally needed in running ICE:

Logical Number

3	AMPX working library output unit
4	AMPX working library input unit
7	Punched card output unit
8	Direct access scratch device*
9	Direct access scratch device*
20	ANISN output unit
21	ANISN group independent output unit

*Direct access space is automatically provided on Units 8 and 9 when ICE is used as a module of AMPX.

REFERENCES

1. N. M. Greene, J. L. Lucius, W. E. Ford, III, R. Q. Wright, and L. M. Petrie, "AMPX: A Modular Code System for Generating Coupled Multi-group Neutron-Gamma Libraries from ENDF/B," ORNL/TM-3706 (March, 1976).
2. W. W. Engle, Jr., "A User's Manual for ANISN," K-1693 (March, 1967).

APPENDIX A
FIDO (ANISN) Input System

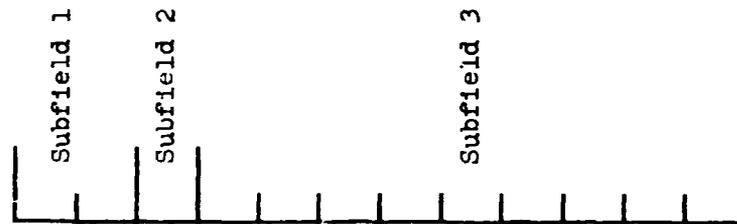
The FIDO input method is especially devised to allow the entering or modifying of large data arrays with minimum effort. Special advantage is taken of patterns of repetition or symmetry wherever possible. The FIDO system was patterned after the input method used with the FLOCO coding system at Los Alamos,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

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

Three major types of input are available: fixed-field input, free-field input, and user-field input.

Fixed-Field Input

Each card is divided into six 12-column data fields, each of which is divided into three subfields. The following sketch illustrates a typical data field. The three subfields always comprise 2, 1, and 9 columns, respectively.



To begin the first array of a block, an array originator field is placed in any field on a card:

Subfield 1: An integer array identifier < 100 specifying the data array to be read in.

Subfield 2: An array-type indicator:

"S" if the array is integer data

"*" if the array is real data

Subfield 3: Blank

Data are then placed in successive fields until the required number of entries has been accounted for. A sample data sheet shown below illustrates this input.

In entering data, it is convenient to think of an "index" or "pointer" which is under control of the user, and which specifies the position in the array into which the next data entry is to go. The pointer is always positioned at array location #1 by entering the array

originator field. The pointer subsequently moves according to the data operator chosen. Blank fields are a special case, in that they do not cause any data modification and do not move the pointer.

A data field has the following form:

Subfield 1: The data numerator, an integer < 100 . We refer to this entry as N_1 in the following discussion.

Subfield 2: One of the special data operators listed below.

Subfield 3: A nine-character data entry, to be read in F9.0 format.

It will be converted to an integer if the array is a "\$" array or if a special array operator such as Q is being used. Note that an exponent is permissible but not required. Likewise, a decimal is permissible but not required. If no decimal is supplied it is assumed to be immediately to the left of the exponent, if any; and otherwise to the right of the last column. This entry is referred to as N_3 in the following discussion.

A list of data operators and their effect on the array being input follows:

"Blank" indicates a single entry of data. The data entry in the third subfield is entered in the location indicated by the pointer, and the pointer is advanced by one. However, an entirely blank field is ignored.

"+" or "-" indicates exponentiation. The data entry in the third field is entered and multiplied by $10^{\pm N_1}$, where N_1 is the

data numerator in the first subfield, given the sign indicated by the data operator itself. The pointer advances by one. In cases where an exponent is needed, this option allows the entering of more significant figures than the blank option.

"&" has the same effect as "+".

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

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

"L" indicates logarithmic interpolation. The effect is the same as that of "I" except that the resulting data are evenly separated in log-space. This is especially convenient for specifying an energy mesh.

"Q" is used to repeat sequences of numbers. The length of the sequence is given by the third subfield, N_3 . The sequence of N_3 entries is to be repeated N_1 times. The pointer advances by $N_1 * N_3$. If either N_1 or N_3 is 0, then a sequence of $N_1 + N_3$ is

repeated one time only, and the pointer advances by $N_1 + N_3$.

This feature is especially valuable for geometry specification.

The "N" option has the same effect as "Q", except that the order of the sequence is reversed each time it is entered. This is valuable for the type of symmetry possessed by S_n quadrature coefficients.

"M" has the same effect as "N" except that the sign of each entry in the sequence is reversed each time the sequence is entered.

For example, the entries:

1 2 3 2M2

would be equivalent to

1 2 3 -3 -2 2 3.

This option is also useful in entering quadrature coefficients.

"Z" causes $N_1 + N_3$ locations to be set to 0. The pointer is advanced by $N_1 + N_3$.

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

"Ø" causes the print trigger to be changed. The trigger is originally off. Successive "Ø" fields turn it on and off alternately. When the trigger is on, each card image is listed as it is read.

"S" indicates that the pointer is to skip N_1 positions leaving those array positions unchanged. If the third subfield is blank, the pointer is advanced by N_1 . If the third subfield is non-blank

that data entry is entered following the skip, and the pointer is advanced by $N_1 + 1$.

"A" moves the pointer to the position, N_3 , specified in the third subfield.

"F" fills the remainder of the array with the datum entered in the third subfield.

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

The reading of data to an array is terminated when a new array origin field is supplied, or when the block is terminated. If an incorrect number of positions has been filled, an error edit is given, and a flag is set which will later abort execution of the problem. FIDO then continues with the next array if an array origin was read. Otherwise, it returns control to the calling program.

A block termination consists of a field having "T" in the second subfield. All entries following "T" on a card are ignored, and control is returned from FIDO to the calling program.

Comment cards can be entered within a block by placing an apostrophe (') in column 1. Then columns 2-80 will be listed, with column 2 being used for printer carriage control. Such cards have no effect on the data array or pointer.

Free-Field Input

With free-field input, data are written without fixed restrictions as to field and subfield size and positioning on the card. The options used with fixed-field input are available, although some are slightly restricted in form. In general, fewer data cards are required for a problem, the interpreting print is easier to read, a card listing is more intelligible, the cards are easier to keypunch, and certain common keypunch errors are tolerated without affecting the problem. Data arrays using fixed- and free-field input can be intermingled at will within a given block.

The concept of three subfields per field is still applicable to free field input, but if no entry for a field is required, no space for it need be left. Only columns 1-72 may be used, as with fixed-field input.

The array originator field can begin in any position. The array identifiers and type indicators are used as in fixed-field input. The type indicator is entered twice, to designate free-field input (i.e., "\$\$" or "**"). The blank third subfield required in fixed-field input is not required. For example:

31**

indicates that array 31, a real-data array, will follow in free-field format.

Data fields may follow the array origin field immediately. The data field entries are identical to the fixed-field entries with the following restrictions:

- (1) Any number of blanks may separate fields, but at least one blank must follow a third subfield entry if one is used.
- (2) If both first- and second-subfield entries are used, no blanks may separate them, i.e., 24S, but not 24 S.
- (3) Numbers written with exponents must not have imbedded blanks, i.e., 1.0E+4, 1.0E4, 1.0+4, or even 1+4, but not 1.0 E4.
- (4) In third-subfield data entries, only 9 digits, including the decimal but not including the exponent field, can be used, i.e., 123456.89E07, but not 123456.789E07.
- (5) The Z entry must be of the form: 738Z, not Z738 or 738 Z.
- (6) The + or - data operators are not needed and are not available.
- (7) The Q, N, and M entries are restricted: 3Q4, 1N4, or M4, but not 4Q, 4N, or 4M.

User-Field Input

If the user follows the array identifier in the array originator field with the character "U" or "V", the input format is to be specified by the user. If "U" is specified, the FORTRAN FORMAT to be used must be supplied in columns 1-72 of the next card. The format must be enclosed by the usual parentheses. Then the data for the entire array must follow on successive cards. The rules of ordinary FORTRAN input as to exponents, blanks, etc., apply. If the array data do not fill the last card, the remainder must be left blank.

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