**FORM: Perspective Plotting Code for
One- and Two-Dimensional Fluxes**

D. T. Ingersoll

MASTER

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION · FOR THE DEPARTMENT OF ENERGY

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.

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 \$4.50; Microfiche \$3.00

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, contractors, subcontractors, or their employees, makes any warranty, express or implied, nor assumes any legal liability or responsibility for any third party's use or the results of such use of any information, apparatus, product or process disclosed in this report, nor represents that its use by such third party would not infringe privately owned rights.

Contract No. W-7405-eng-26

Neutron Physics Division

FORM: PERSPECTIVE PLOTTING CODE FOR
ONE- AND TWO-DIMENSIONAL FLUXES

D. T. Ingersoll

Date Published - May, 1978

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, 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.

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

Table of Contents

	<u>Page</u>
Abstract	v
A. Code Description.	1
B. Input	2
C. Storage and Time Requirement	4
D. Data Notes.	4
E. ANISN Flux Plot	8
F. JCL for FORM.	9
G. FIDO Input System	10
H. CONV Program Description.	18
Sample Problem 1.	20
Sample Problem 2.	20
Sample Problem 3.	23
Sample Problem 4.	26
References.	29

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

Abstract

FORM (Fluxes Over Region Map) is a perspective plotting code for displaying two-dimensional fluxes projected above the corresponding region boundary map. The code uses the extensive and versatile DISPLA graphics package available at ORNL to provide a number of useful features including the plotting of R- θ fluxes. FORM uses FIDO input format and is designed to read flux tapes generated by the DOT 3.5 code. While providing a few unique features, FORM is intended as a limited alternative to the more extensive GOFRR plotting code.

A. Code Description

FORM is a two-dimensional surface plotting code designed specifically to generate perspective plots of DOT 3.5¹ scalar fluxes. It can also be used to plot DOT IV² fluxes if the flux tape modification code RTFLUM³ is used. In addition, ANISN³ fluxes can be plotted as a function of distance and energy. For this option, the auxiliary program CONV (discussed in Sec. H) is available which divides the energy-dependent ANISN fluxes by group energy-width or lethargy-width. With proper choice of input, FORM can be used to plot any two-dimensional surface. Primary features of FORM are:

- (1) Fluxes are plotted with all three axes labeled and scaled, or plotted as though projected directly above a zone boundary map. In the latter case, a separate nonprojected zone map is also plotted. (For extra class, the projected zone map can be plotted in a second pen color!)
- (2) Multiple views of flux surface from arbitrary viewing points can be specified.
- (3) Fluxes can be plotted by energy group or summed over any or all groups. The same option allows activities to be plotted.
- (4) Fluxes can be plotted in either linear or logarithmic fashion.
- (5) Minimum and maximum flux values can be input explicitly or generated internally.
- (6) FORM uses FIDO input format with many of the arrays identical to DOT input arrays.
- (7) R- θ fluxes are plotted in true polar coordinates.
- (8) Multiple cases may be stacked with only new information required.

B. Input

Problem Title Card

20A4 FORMAT

1\$ [10] Control Parameters

- 1) IGM Number of energy groups.
- 2) IM Number of X-R-R intervals.
- 3) JM Number of Y-Z- θ intervals.
- 4) NPLOT Number of distinct flux plots using different sets
 of energy group multipliers (6* array). (See Data
 Note 1) [Default = 1]
- 5) NVU Number of different viewpoints for each flux plot.
 [Default = 1]
- 6) ILL 0/1 Plot linear/log flux in Z-direction. [Default = 0]
- 7) MNMX 0/1 Calculate/Read minimum and maximum flux limits.
 MNMX = 0 plots entire flux range. [Default = 0]
- 8) IREG = 0 Plot flux surface only.
 = 1 Plot 2-D region (zone) map projected below flux
 surface. Also plot one nonprojected region map.
 = IZM Plot 2-D material map. IZM must equal number of
 regions as indicated in 8\$ and 9\$ arrays.
 (See Data Note 2) [Default = 0]
- 9) IGEOM 0/1/2 X-Y/R-Z/R- θ Geometry [Default=1] (See Data Note
 3).
- 10) NFLX Logical unit number of input scalar flux. [Default = 4]

5*	[8]	Plotting parameters
1)	HLGTH	Horizontal length of plot area in inches (≤ 8.0). [Default = 8.0].
2)	VLGTH	Vertical length of plot area in inches (≤ 10.5). [Default = 9.0].
3)	XREL	Relative X-axis length. [Default = 1.0].
4)	YREL	Relative Y-axis length. [Default = 1.0]. (See Data Note 4)
5)	ZREL	Relative Z-axis length. [Default = 1.0].
6)	XDIV	X-R-R units per axis label. (See Data Note 5)
7)	YDIV	Y-Z- θ units per axis label. (See Data Note 5)
8)	RREL	Location of projected region map given in <u>relative</u> Z-axis units. The map will appear RREL units below the minimum flux value. (Arbitrary if IREG = 0) [Default = 1.0]
T	Data Block 1	Terminator
2*	[JM + 1]	Y-Z- θ interval boundaries
4*	[IM + 1]	X-R-R interval boundaries
6*	[NPLLOT * IGM]	Energy group multipliers
8\$	[IM * JM]	Region number by interval
9\$	[IZM]	Material number by region (IREG = IZM)
10*	[3*NVCU]	Viewing coordinates in <u>relative</u> X, Y, and Z units. (See Data Note 6) [Default = 3R3.0]
12*	[2*NPLLOT]	Minimum and maximum flux values. (Not required if MNMX = 0).
T	Data Block 2	Terminator

Plot Labels [2* NPLOT cards]

Plot Title	Columns 1 - 40 of first card
X-R-R Axis Title	Columns 1 - 20 of second card
X-Z- θ Axis Title	Columns 21 - 40 of second card
Z (flux) Axis Title	Columns 41-72 of second card

Note: Characters enclosed in ()'s will be plotted as lower case,
and "(" will plot a single "(".

C. Storage and Time Requirements

The amount of common storage needed to execute FORM can be determined from:

$$\text{NSTORE} = 20 + 3(\text{NVU} + \text{IM} * \text{JM}) + \text{NPLOT} * (2 + \text{IGM}) + \text{SCRATCH}$$

$$\text{where SCRATCH} = \begin{cases} 7200 & \text{IGEOM} = 2 \\ 2(\text{IM} + \text{JM}) & \text{IGEOM} < 2 \end{cases}$$

On the IBM 360/91, the REGION requirement for FORM is approximately:

$$\text{REGION} \simeq 256\text{K} + 4 * \text{NSTORE}$$

The CPU time requirement is closely related to the number and type of plots to be drawn. Experience has shown that on the IBM 360/91, the total CPU time can be roughly estimated by:

$$\text{T(sec)} \simeq 4 + 2 * (\# \text{ of region maps}) + 3 * (\# \text{ of surface plots})$$

There will always be NPLOT * NVU) surface plots, plus an additional (NPLOT * NVU + 1) region maps if IREG $\neq$ 0.

D. Data Notes

Note # 1 NPLOT (1\$ array)

The energy-dependence of the flux can be investigated using

0's and 1's in the 6* array allows the user to determine

which groups are to be summed and plotted. NPLOT >1 permits multiple summations within one job. Also, cross sections or response functions can be entered into the 6* array in order to plot activities. These activities will appear in all zones, however, since the 6* array is entered by group only. The user must specify NPLOT*IGM entries in the 6* array.

Note # 2

IREG (1\$ array)

With IREG = 0, the flux surface is plotted with all three axes labeled as supplied by the user. IREG = 1 will plot the flux surface "floating" directly above a region (or zone) map specified by the 8\$ array. The Z axis does not appear in this case unless RREL = 0.0. Also a stand-alone nonprojected region map is plotted -- only once per job. IREG = IZM will plot a material map instead of a region map, where IZM is the number of regions indicated in the 8\$ array and is the the number of entries in the 9\$ array. If IREG < 0, the projected region (or material) map is plotted in a second color specified as PEN # 2. This option is only available when using the CALCOMP 925/1036 plotters at K-25 and X-10

Note # 3

IGEOM (1\$ array)

The IGEOM parameter has a hidden feature which is useful to the serious R-θ plotter. With IGEOM<2, the X and Y grid of the flux surface has a one-to-one correspondence to the mesh

defined by the 4* and 2* arrays. For IGEOM = 2, the X and Y grid is independent of R and θ with normally 60 equally spaced lines in each of the X and Y directions. It is generally more presentable, however, to scale the number of X and Y lines according to their respective axis lengths which is determined by $|\cos\theta_{\max}|$. For example, a 45° (or 135°) problem would appear best if 60 by 43 lines were used. This is accomplished by setting IGEOM = 6043, i.e., $\text{IGEOM} = (100 * \# \text{ of X-lines}) + (\# \text{ of Y-lines})$.

NOTE: This option markedly affects the required number of storage locations. If the IGEOM > 2 option is used, then the 7200 value of the SCRATCH term in the storage relation (sec. C) is replaced by $2(\# \text{ of X-lines}) * (\# \text{ of Y-lines})$.

Note # 4

XREL, YREL, ZREL (5* array)

The 3-D workbox within which the flux surface is plotted is determined by XREL, YREL, and ZREL which specify the relative lengths of the X, Y, and Z edges of the workbox.

The absolute values of XREL, YREL, and ZREL are unimportant; i.e., 1, 1, 1 yields the same result as 100, 100, 100 -- both produce a cubical plot. Values of 10, 10, 1 would produce a plot that was very flat in the Z (flux) direction. The absolute values of XREL, YREL, and ZREL do affect the choice of RREL and the viewing coordinates (10* array) which are given in terms of the same units.

Note # 5

XDIV, YDIV (5* array)

The X-R-R and Y-Z- θ axes are labeled in terms of "user

units" which are generally centimeters (2* and 4* arrays). The axes will be divided into intervals with numbers written at the division marks according to XDIV and YDIV which specify the number of user units per axis interval on the radial and axial axes, respectively. Specifying XDIV equal to the difference between the last and first entries in the 4* array will cause numbers to be written only at the axis extremities, and similar for YDIV and the 2* array.

Note # 6

Viewpoint (10* arrays)

The viewpoint for the perspective plots are specified by viewing coordinates which are defined in relative X, Y, and Z units. (See Data Note # 3) The viewpoint is only restricted not to lie within the workbox defined by the coordinate points (0, 0, 0) and (XREL, YREL, ZREL). For most problems, an adequate viewpoint is (3*XREL, 3*YREL, 3*ZREL). Moving closer to the workbox increases the perspective effect while moving away tends to "square-off" the workbox so that in the limit of a very distant viewpoint, the 3-D plot will be isometric; i.e., parallel lines are plotted parallel.

Note # 7

Multiple cases

Multiple cases can be stacked by immediately following the final plot title card of the first case with the problem title card of the next case. Only new information need be entered in the second case except that: (a) both data block T's must be entered, (b) all title cards must be entered, and (c) if any dimensioning changes are made, all

arrays must be re-entered to assure proper alignment in storage.

E. ANISN Flux Plot

In order to plot ANISN³ fluxes as a function of position and energy, the following input changes should be made:

- (1) In the 1\$ array, set IGM = -1

.IM = no. of energy groups

NPLOT = 1 only

IREG = 0 only

- (2) In the 2* array, enter the energy group boundaries. For most problems this will cause a bunching of lines at low energy. It may be better to enter group numbers instead (0 to IGM) or to use log(energy)
- (3) Enter 1.0 in the 6* array.
- (4) Do not enter the 8\$ array.
- (5) In the 5* array, the value of YDIV will depend on the user's choice of the 2* array. However, if the 2* array has descending values, YDIV must be negative.
- (6) If ANISN fluxes are used directly, the surface may show apparent structure relating to the energy group structure. The auxiliary program CONV (described in a later section) can be used to remove this effect by dividing the fluxes by the group energy-widths or lethargy-widths. CONV can also be used to plot ANISN activities as demonstrated in Sample Problem 3.

F. JCL for FORM

Below is given the required JCL to run FORM. It does not include the needed JCL for plotting, which varies with the methods used (i.e., diskplot, plottape, etc).

```
(X-10)  //FORM EXEC FORTHCLG, REGION = 350K
        //FORT.SYSIN DD *
            COMMON DM(10000)
            NSTORE = 10000
            CALL CONTRL(NSTORE,DM)
            STOP
            END
        /*
        //LKED.SYSLIB DD
        //          DD
        //          DD
        //          DD DSN=DISSPLA.LOAD, DISP=SHR
        //          DD DSN=X.JVP15335.WARLIB, DISP=SHR
*       //LKED.LMOD DD DSN=X.DTI14647.PLIB,UNIT=3330,VOL=SER=ZX1111,DISP=SHR
        //LKED.SYSIN DD *
            INCLUDE LMOD(FORM)
        /*

            {flux and plot DD cards}

        //GØ.FT05F001 DD *

            {data}

        //
```

(K-25) To run FORM at K-25, replace only the starred statement with the following:

```
//LKED.LMOD DD DSN=X.DTI14647.PLIB, VOL=SER=XXX117,DISP=SHR,
// UNIT=2314
```

G. FIDO 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 forming 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.

* FIDO description taken from W. A. Rhoades.

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:

"\$" if the array is integer data

"*" if the array is read 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 the 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 no 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.

H. CONV Program Description

CONV is a straightforward program to divide ANISN³ fluxes by group energy-width or lethargy-width. In addition, a group multiplier array is available in order to convert fluxes to activities.

Input Description:

Title Card 20A4 FORMAT

1\$ [5] Control parameters.

- 1) IGM Number of energy groups.
- 2) IM Number of intervals.
- 3) LORE 0/1 Divide by lethargy/energy.
- 4) NIN Input flux logical unit number.
- 5) NOUT Output flux (activity) logical unit number.

T Data Block 1 Terminator

2* [IGM+1] Energy group boundaries (upper boundaries + lower boundary
 of last group).

6* [IGM] Energy group multipliers.

T Data Block 2 Terminator

Notes:

- (1) If $IGM < 0$, the logarithm of the energy group boundaries will be punched in free-form FIDO format. If this option is used, a DD card for FT07F001 must be supplied.
- (2) CONV is available as a load module at both X-10 and K-25 sites. The JCL is similar to the JCL needed for FORM. Sample Problem 3 shows the complete input listing of a CONV example.

Sample Problem 1

The first sample problem demonstrates a FORM plot of a single-group flux with a projected region boundary map. The example is a result of a DOT IV R-Z calculation and shows the fast neutron flux in the experimental assembly of the GCFR Grid Plate Shield Confirmation Experiment. A complete input listing of the job submitted to the IBM 360/195 is given on the next page followed by the two resulting plots: a nonprojected region boundary map and a linear surface plot projected above the same region map. The first job step uses RTFLUM (or RTFLUX) to convert the DOT IV flux tape to a DOT 3.5 formatted data set. The new data set is then passed to the second job step which generates the sample plots. The RTFLUM step also serves the useful task of transferring the flux tape to a direct access device. This is especially important if NPLOT is greater than 2 or 3 since each NPLOT causes a rewind of the data set which would tend to wear a magnetic tape excessively.

Sample Problem 2

The second example is a part of the same computer job that created the first sample problem, demonstrating the use of multiple cases. The resulting plot shows the total log (flux) drawn without a region map. This example also shows that regardless of the viewing point, the z axis (flux axis) is always located on the left side of the figure.

Complete JCL and input listing of sample problems 1 and 2:

```

/*RELAY          REMOTES
95
/*SECCARD
//DTIFORM JOB (14647), 'MPD 6025 INGERSOLL'
/*CLASS C0195=50S, REGION=350K, IO=20
//DMS EXEC XEQ, PROG=DRIVER, TIME=1440, REGION=350K
//GO, FT05F001 DD UNIT=SYS0A, SPACE=(80,(50,25)),
// DCR=(RECFM=FB, LRECL=80, BLKSIZE=3200, BUFGNO=1)
//GO, FT01F001 DD UNIT=TAPE9, DISP=(OLD,KEEP),
// DSN=X, DTIG2647, GR25, VOL=SER=A01624, LABEL=(1,SL)
//GO, FT02F001 DD UNIT=SYS0A, DISP=(NEW,PASS), DSN=66TMDFLX,
// DCR=(RECFM=VS, BLKSIZE=13030, BUFGNO=1), SPACE=(13030,(71,51))
//GO, FT08F001 DD UNIT=SYS0A, SPACE=(80,(50,25)),
// DCR=(RECFM=FB, LRECL=80, BLKSIZE=3200, BUFGNO=1)
//GO, FT09F001 DD DDNAME=SYSDN
=RTFLUX
RTFLUX:  CHANGE DDIV FLUXES TO DD3.5 FLUXES FOR FORM
10
15*  1  2  0  30  3  60  0  1  0  0  7  42  59  3  30  7  7  1  0  0
T
=END
//FORM EXEC FORTHCLG, REGION=350K, PLUT=PLT
//FORT, SYSDN DD *
C
C  ALLOCATE COMMON STORAGE FOR  F O R M
C
COMMON DM(10000)
NSTORE=10000
CALL CONTRL(NSTORE,DM)
STOP
END
/*
//LKFD, SYSLIB DD
//
//
//
// DSN=DISPLA, LOAD, DISP=SHR
// DSN=X, JVD15335, KAWLIB, DISP=SHR
//LKED, LMOD DD UNIT=2314, DSN=X, DT114647, PLIR, VOL=SER=XXX117, DISP=SHR
//LKED, SYSDN DD *
INCLUDE LMOD(FORM)
ENTRY MAIN
/*
//GO, FT54F001 DD UNIT=TAPE9, DCR=(RECFM=VS, LRECL=364, BLKSIZE=368)
//GO, FT04F001 DD DSN=66TMDFLX, DISP=(OLD,DELETE), DCR=BUFGNO=1
//GO, FT05F001 DD *
FORM:  SAMPLE PROBLEM 1 - GCFR STREAMING EXPERIMENT
0
15*  7  42  59  1  1  0  0  1  1  4
5**  7.5  8.5  1.0  2.5  2.5  20.0  50.0  2.5
T
2**  1310.0  12145.0  1194.848  11195.8  141.203  71141.52  172.0
71172.318  202.48
4**  410.0  6.65  517.707  14.546  5115.729  22.755  1123.938  1126.911
2127.864  34.10  3138.418  1153.34  7154.293  83.82
6**  1.0  62
R55
4R1  10R2  4R1  4R2  2R3  18R4  2042
5R1  8R2  6R1  3R2  2R3  18R4  3042
6R1  6R2  8R1  2R2  2R3  18R4  3042
7R1  4R2  10R1  1R2  2R3  18R4  2042
5  6  1002  2R3  18R4  12042
5  6  1002  2R3  10R10  8R4  1042
5  6  1002  2R3  8R7  2R10  8R4  11042
5  6  1002  2R3  4R9  4R7  2R10  8R4
5  6  1002  2R3  3R8  9  4R7  2R10  8R4  7042
24R9  27R8  9  4R7  2R10  8R4  7042
10**  100.  -100.  85.
T
RADIUS ((CM))  SAMPLE 1: FAST FLUX
HEIGHT ((CM))
FORM:  SAMPLE PROBLEM 2 - GCFR STREAMING EXPERIMENT
15*  7  42  59  1  1  1  0  0  1  4
5**  7.5  8.5  1.0  2.5  2.0  20.0  50.0  0.0
T
6**  7R1.0
10**  -100.  -100.  85.
T
RADIUS ((CM))  SAMPLE 2: TOTAL FLUX
HEIGHT ((CM))  LOG FLUX
//

```

ORNL-DWG-78-1223

SAMPLE 1: FAST FLUX

ORNL-DWG-78-1224

SAMPLE 2: TOTAL FLUX

ORNL-DWG-1111

Sample Problem 3

The third sample problem is primarily an example of using FORM to plot ANISN fluxes, in this case from a CTR standard blanket calculation. The program again used two job steps with the first step preparing the data set for the second step. The first step used CONV to convert the ANISN scalar fluxes to ${}^6\text{Li}(n,\alpha)$ reaction rates and also to divide the activity by group lethargy widths. The activity plotted by FORM in the second job step shows non-zero values in all intervals despite the fact that ${}^6\text{Li}$ material appears in only specific regions. This demonstrates one limitation of CONV.

The energy axis used log(energy) values which had been punched by CONV in an earlier job, and shows the use of a negative YDIV when the energy values are descending. This approach is generally better than using group numbers since it shows the sometimes irregular energy structure.

JCL and input listing for Sample Problem 3:

```
/*RELAY          REMOTE5
95
/*SFQCARD
//DTIFORM JOB (14647),'INPD 6025 INGERSOLL'
/*CLASS CPU95=50S,REGION=350K,IO=20
//CONV EXEC FORTHCLG,REGION=350K
//FORT.SYSIN DD *
COMMON IA(10000)
CALL FIXFLX(IA)
STOP
END

/*
//LKED.SYSLIB DD
//          DD
//          DD
//          DD DSN=EX.JVP15335.WARLIB,DISP=SHR
//LKED.LMOD DD UNIT=2314,DSN=EX.DTI14647,PLIB,VOL=SER=XXX117,DISP=SHR
//LKED.SYSIN DD *
INCLUDE LMOD(CONV)
ENTRY MAIN
/*
//GO.FT03F001 DD UNIT=TAPF9,DISP=(OLD,KEEP),
//          DSN=EX.DTIME975,STHLK,CCCC1,VOL=SER=A07524,LABEL=(5,SL)
//GO.FT04F001 DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=66TMPFLX,
//          DCH=(RECFM=VS,BLKSIZE=13030,BUFNO=1),SPACE=(13030,(71,51))
//GO.FT07F001 DD SYSOUT=B
//GO.FT05F001 DD *
CONV: DIVIDE ANISN FLUXES BY GROUP LETHARGY WIDTH
0
155 101 104 0 3 4
T
```


JCL and input listing for Sample Problem 3:

2**

1.4918367	1.3498667	1.2214067	1.1051757	1.0000067	1.7333367		
8.1873166	7.4081866	5.7032066	6.0653166	5.4881266	9.0483766	100	GP 1
4.4932966	4.0657066	3.6788066	3.3287166	3.0119466	4.9658566	100	GP 2
2.4659766	2.2313066	2.0189766	1.8268466	1.6529966	2.7253266	100	GP 3
1.3533566	1.2245666	1.1080366	1.0025966	9.0718065	1.4956966	100	GP 4
7.4273665	6.7205565	5.0810165	5.5023265	4.9787165	8.2085065	100	GP 5
4.0762265	3.6883265	3.3373365	3.0197465	2.7323765	4.5049265	100	GP 6
2.2370865	2.0241965	1.8315665	1.6572765	1.4995665	2.4723565	100	GP 7
1.2277365	1.1109065	8.6517064	6.7379564	5.2475264	1.3568665	100	GP 8
3.1827864	2.4787564	1.9304664	1.5034464	1.1708864	4.0867764	100	GP 9
7.1017563	5.5308563	4.3074363	3.3546363	2.6125963	9.1188263	100	GP10
1.5846163	1.2341063	9.6111762	7.4851962	5.8294762	2.0346863	100	GP11
3.5357562	2.7536562	2.1445462	1.6701762	1.3007362	4.5400062	100	GP12
7.8893361	6.1442261	4.7851261	3.7266561	2.9023261	1.0130162	100	GP13
1.7603561	1.3709661	1.0677061	8.3152960	6.4759660	2.2603361	100	GP14
3.9278760	3.0590260	2.3823760	1.8553960	1.4449860	5.0434860	100	GP15
8.76425-1	6.82561-1	5.31579-1	4.13990-1	0.00100-1	1.1253560	100	GP16
LI-6(N,ALPHA) ACTIVITY CROSS SECTION							GP17

```

6U
(6E12,5)
2.37866E-02 2.57771E-02 2.73254E-02 3.23276E-02 3.59338E-02 4.03635E-02
4.55345E-02 5.07895E-02 5.67982E-02 6.36567E-02 7.11032E-02 7.99702E-02
8.88012E-02 9.93139E-02 1.11740E-01 1.24760E-01 1.37094E-01 1.54150E-01
1.77763E-01 2.03743E-01 2.28294E-01 2.48562E-01 2.61383E-01 2.58116E-01
2.50229E-01 2.46362E-01 2.45959E-01 2.48430E-01 2.53564E-01 2.61496E-01
2.72664E-01 2.87484E-01 3.07405E-01 3.34965E-01 3.73567E-01 4.27350E-01
5.09970E-01 6.40150E-01 8.62220E-01 1.25930E 00 1.97786E 00 3.00470E 00
3.42341E 00 7.70542E 00 1.87530E 00 1.35488E 00 1.06246E 00 8.94694E-01
7.96181E-01 7.37248E-01 6.91240E-01 6.83167E-01 7.15148E-01 7.70125E-01
8.45179E-01 9.37665E-01 1.04799E 00 1.17820E 00 1.32160E 00 1.49020E 00
1.68310E 00 1.90270E 00 2.15300E 00 2.43658E 00 2.75981E 00 3.11988E 00
3.53880E 00 4.01130E 00 4.53340E 00 5.14620E 00 5.83300E 00 6.60370E 00
7.48890E 00 8.48750E 00 9.60670E 00 1.08950E 01 1.24000E 01 1.39870E 01
1.59070E 01 1.80220E 01 2.03910E 01 2.31140E 01 2.61530E 01 2.95860E 01
3.36180E 01 3.81460E 01 4.31670E 01 4.89800E 01 5.55150E 01 6.29100E 01
7.11710E 01 8.08220E 01 9.13749E 01 1.03660E 02 1.17460E 02 1.33170E 02
1.50720E 02 1.71020E 02 1.94080E 02 2.19100E 02 2.48799E 02

```

```

T
//FORM EXFC FORTHCLG,REGION=350K,PL0T=PLT
//FORT.SYSIN DD *
C
C ALLOCATE COMMON STORAGE FOR F O R M
C
COMMON DM(25000)
NSTORE=25000
CALL CONTRL(NSTORE,DM)
STOP
END
/*
//LKED.SYSLIR DD
// DD
// DD
// DD DSN=DISSPLA,LOAD,DISP=SHR
// DD DSN=X.JVP15335,WARLIR,DISP=SHR
//LKED.LMOD DD UNIT=2314,DSN=X,DTI14647,PLIR,VOL=SER=XXX117,DISP=SHR
//LKED.SYSLIR DD *
INCLUDE LMOD(FORM)
ENTRY MAIN
/*
//GO.FT54F001 DD UNIT=TAPE9,DCB=(RECFM=VS,LRECL=364,BLKSIZE=368)
//GO.FT04F001 DD DSN=66TMDPLX,DISP=(OLD,DELETE),DCB=BUFNO=1
//GO.FT05F001 DD *
FORM: SAMPLE PROBLEM 3 * ANISN FLUX FROM CTR STANDARD BLANKET
O
155 -1 62 101 1 1 1 0 0 1 4
5** 7.0 8.0 1.0 1.0 1.0 50.0 -2.0 0.0
T
2**
723885-5 717372-5 713029-5 708685-5 704343-5 7 695656-5 691314-5 101 1
686971-5 682628-5 678285-5 673942-5 669599-5 665257-5 660914-5 656571-5 101 2
652228-5 647885-5 643542-5 639199-5 634856-5 630513-5 62617-4 621827-5 101 3
617484-5 613141-5 608798-5 604455-5 600112-5 595769-5 591426-5 587084-5 101 4
58274-4 578398-5 574055-5 569712-5 565369-5 561026-5 556683-5 55234-4 101 5
547997-5 543654-5 539311-5 534968-5 530625-5 526282-5 521939-5 517596-5 101 6
513253-5 50891-4 504568-5 49371-4 482853-5 471995-5 461138-5 450281-5 101 7
439423-5 428566-5 417709-5 406851-5 395994-5 385137-5 374279-5 363422-5 101 8
352564-5 341707-5 33085-4 319992-5 309135-5 298278-5 28742-4 276563-5 101 9
265705-5 254848-5 243991-5 233133-5 222276-5 211419-5 200561-5 189704-5 101 10
178847-5 167989-5 157132-5 146274-5 135417-5 12456-4 113702-5 102845-5 101 11
919877-6 811304-6 70273-5 594157-6 485582-6 377009-6 268435-6 159862-6 101 12
512873-7 -572855-7 -165859-6 -274432-6 -38301-5 -4 101 13
4** 0. 150. 21200. 51200.5 21203.5 291204. 141264. 21294. 300.
6** 1.0
10** 100.0 -100.0 120.0
T

```

```

RADIUS ((CM))
LOG((ENERGY))
LOG FLUX
//

```


SAMPLE 3: ANISN ACTIVITY

Sample Problem 4

The fourth sample problem demonstrates the plotting of DOT IV R- θ fluxes. This particular example shows the total neutron flux in a conceptual nondestructive assay facility which is designed to determine the plutonium content of a fast reactor fuel subassembly. The DOT IV fluxes had been previously prepared by RTFLUM and stored on a semi-permanent direct access device. The program was run on the IBM 360/91 and used the disk plot option available at the X-10 site. It should be noted that FORM selected the X and Y axis so as to just circumscribe the R- θ region. Also, the IGEOM>2 option was used to yield equal line spacing in the X and Y directions.

JCL and input listing of Sample Problem 4:

```
//DTIFORM JOB (14647),'MDD 6025 INGERSOLL'
//**CLASS CPU91=305,REGION=350K,10=10
// EXEC FORTHCLG,REGION=350K
//FORT.SYSIN DD *
C      ALLOCATE COMMON STORAGE FOR F O R M
C
      COMMON DM(12000)
      NSTORE=12000
      CALL CONTRL(NSTORE,DM)
      STOP
      END
/*
//LKFD.SYSLIB DD
//          DD
//          DD
//          DD DSN=DISPLA,LOAD,DISP=SHR
//          DD DSN=X,JVP15335,WALTH,DISP=SHR
//LKED.PLOTSUBS DD DSN=JGSPLOTH,DISP=SHR
//LKED.LMOD DD DSN=X,DTI14647,PLTH,UNIT=3330,VOL=SER=ZX1111,DISP=SHR
//LKFD.SYSIN DD *
      INCLUDE PLOTSUBS
      INCLUDE LMOD(FORM)
      ENTRY MAIN
/*
//GO.FT49F001 DD UNIT=IN2002,DISP=(NEW,KEEP),SPACE=(3208,(90,10),RLSE),
// DCH=(RECFM=VS,LRECL=3204,RLKSIZE=3208),DSN=PL0T00.DTI1
//GO.FT04F001 DD DSN=DTI14647,NDARTF,DISP=(SHR,KEEP)
//GO.FT05F001 DD *
      FORM: SAMPLE PROBLEM #4 * N D A R=THETA FLUX
      O
15* 51 32 26 1 1 1 0 1 6043 4
5** 7.5 8.5 2.0 1.5 3.0 10.0 10.0 2.0
T
2** 710.0 1310.02047 110.09538 110.10855 0.1250
4** 0.0 2.8174 3.9845 4.8800 5.6349 116.3 218.0 318.4 12.21
    13.45 15.0 16.55 5117.79 3130.5 3139.5 45.0
6** F1.0
8**
      5R1 2R2 3R3 14R4 7 2R8 7 4R4 3032
      5R1 2R2 3R3 14R4 4R7 4R4 3032
      5R1 2R2 3R3 22R4 4R4 13032
      5R1 2R2 3R3 4R4 4R5 14R4 1032
      5R1 2R2 3R3 4R4 5 2R6 5 14R4 1032
10** 100.0 130.0 120.0
T
      SAMPLE 4: R-THETA FLUX
      RADIUS ((CM)) RADIUS ((CM))
//
```

ORNL-DWG-78-1228
SAMPLE 4. R-Theta Flux

REFERENCES

- (1) W. A. Rhoades and F. R. Mynatt, "The DOT III Two-Dimensional Discrete Ordinates Transport Code," ORNL-TM-4280, September 1973.
- (2) W. A. Rhoades, D. B. Simpson, R. L. Childs, and W. W. Engle, Jr., "The DOT IV Two-Dimensional, Discrete-Ordinates Transport Code with Space-Dependent Mesh and Quadrature," ORNL-5283 (1977).
- (3) W. W. Engle, Jr., "A Users Manual for ANISN - A One-Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering," K-1693, March 30, 1967.
- (4) F. B. Sadler, D. L. Selby, "GOFRR: A Computer Code to Generate Graphical Output for Fluxes and Reaction Rates," ORNL/TM-5063 (1977).

INTERNAL DISTRIBUTION

1. L. S. Abbott	44. C. O. Slater
2. R. G. Alsmiller, Jr.	45. E. T. Tomlinson
3. D. E. Bartine	46. J. S. Tang
4. T. J. Burns	47. C. R. Weisbin
5. S. N. Cramer	48. J. R. White
6. M. B. Emmett	49. G. E. Whitesides
7. W. W. Engle, Jr.	50. L. R. Williams
8. G. F. Flanagan	51. M. L. Williams
9. H. Goldstein (Consultant)	52. B. A. Worley
10-27. D. T. Ingersoll	53. A. Zucker
28. H. E. Knee	54. P. Greebler (Consultant)
29. F. C. Maienschein	55. W. B. Loewenstein (Consultant)
30. E. M. Oblow	56. R. E. Uhrig (Consultant)
31. J. V. Pace	57. R. Wilson (Consultant)
32. R. W. Peelle	58-59. Central Research Library
33-40. Radiation Shielding
Information Center	60. ORNL Y-12 Technical Library
Document Reference Section
41. W. A. Rhoades	61-62. Laboratory Records Department
42. D. L. Selby	63. Laboratory Records ORNL RC
43. D. B. Simpson	64. ORNL Patent Office

EXTERNAL DISTRIBUTION

- 65. DOE Oak Ridge Operations, Research and Technical Support
Division: Director
- 66. DOE Oak Ridge Operations, Reactor Division: Director
- 67-68. DOE Division of Reactor Research and Development, Washington,
D.C. 20545: Director
- 69-95. Technical Information Center (TIC)