NL95FM087

TU Delft

Interfacultair Reactor Instituut

IRI-131-92-002

SCALPLO - A Universal Program for Plotting Flux Output from SCALE Modules and Related Programs

A. Hersman, P.F.A. de Leege and J.E. Hoogenboom
Delft University of Technology, Interfaculty Reactor Institute
Mekelweg 15, 2629 JB Delft, The Netherlands

USER's MANUAL

Delft, April 1992

1. **NAME OR DESTINATION OF PROGRAM** - SCALPLO.
A Universal Program for Plotting Flux Output from SCALE Modules and Related Programs, Version 1.0.

2. **COMPUTER FOR WHICH THE PROGRAM IS DESIGNED AND OTHER MACHINE VERSION PACKAGES AVAILABLE**

Program-name	Package-ID	Orig. Computer	Test Computer
SCALPLO-1.0		VAX 4300	

3. **DESCRIPTION OF PROGRAM OR FUNCTION** - SCALPLO is a plot program, designed to plot flux, power and spectrum information. Data exchange between SCALE modules and SCALPLO is via CCCC-interface files. As not all modules can produce these files, there are special routines supplied with SCALPLO that can produce CCCC-like files. These routines can be included in the code and for XSDRNPM, CITATION, ANISN and DOT, the place to include these routines is supplied.

4. **METHOD OF SOLUTION** - SCALPLO consists of two sections. Firstly the preprocessor, which selects and reads the required data. Secondly the plot section which produces the plot on the selected output device.

5. **RESTRICTIONS ON THE COMPLEXITY OF THE PROBLEM** - SCALPLO requires DISSPLA version 11.0 or higher. The choice of output device depends on the devices installed.

6. **TYPICAL RUNNING TIME** - Depends on the amount of data to be read and the output device selected. Running time will be less than one minute.

7. **UNUSUAL FEATURES OF THE PROGRAM** - Three subroutines are supplied to make CCCC-like interface files readable to SCALPLO and it is possible to add them to other codes. Description of adapting XSDRNPM, CITATION, ANISN and DOT is added to the manual. The BOLD-VENTURE module already uses CCCC interface files. It is also possible to enter ASCII X-Y data for fast evaluation of simple data.

8. **RELATED AND AUXILIARY PROGRAMS** -

9. **STATUS** -

10. **REFERENCES** -

11. **MACHINE REQUIREMENTS** -
3 Mbytes of virtual memory. No scratch disk is used.

12. **PROGRAMMING LANGUAGE(S) USED** -
FORTRAN-77 (full standard, except the input statement NAMELIST option, accepted by many compilers)

13. **OPERATING SYSTEM UNDER WHICH PROGRAM IS EXECUTED** -
VMS 5.4 (DEC VAX 4300 cluster)

14. **OTHER PROGRAMMING OR OPERATING INFORMATION OR RESTRICTIONS** - SCALPLO uses DISSPLA which has to be available on location. SCALPLO routine INISYS is the appropriate place to enter the DISSPLA initialisations.

15. **NAME AND ESTABLISHMENT OF AUTHORS** -

A. Hersman, P.F.A. de Leege and J.E. Hoogenboom
Delft University of Technology
Interfaculty Reactor Institute
Mekelweg 15
2629 JB Delft
The Netherlands
E-mail : HERSMAN@IRI.TUDELFT.NL
Tel : ++31 15 78 39 31
Telefax : ++31 15 78 64 22

16. **MATERIAL AVAILABLE** -

SCALPLO FORTRAN source	: 2328 lines
CCCC-routines FORTRAN source	: 219 lines
ASCII Test input file	: 37 lines
ASCII Test data file	: 24 lines
ASCII Test list file	: 103 lines

17. **CATEGORY** - M, P

KEYWORDS -
COMPUTER GRAPHICS, DATA ANALYSIS

1. Introduction to SCALPLO

Reactor core and shielding codes generally produce a vast amount of flux values, depending on the number of energy groups, the number of dimensions in the geometry considered and the number of mesh intervals for each dimension. Such an amount of data can only be judged by graphical display of selected data. As most computer codes in this field use their own data format for the flux output, no generally applicable flux plotting program is available. Standardization of flux and geometry output seems very difficult to arrange, even within a code system like SCALE, as it contains modules developed by many different authors.

One can develop a flux plotting program that has separate interfaces to the output produced by all kinds of flux calculation codes, however, severe difficulties are involved. The program will permanently stay in development as interfaces to new codes are to be included. More severe is the problem that not all codes in current use can produce the necessary output. Therefore, another way was followed at the Interfaculty Reactor Institute (IRI) in Delft, which requires only minor modifications of the flux calculation programs, if they are not already preprogrammed to produce the necessary output in a standardized way. Although making changes to existing codes will be done in many cases, one will prefer to limit this to the minimum. Therefore, additional routines are developed that facilitates the writing of the requested data from the flux calculation program, so that the inclusion of a few CALL statements to those subroutines at the appropriate place in the program suffices.

The plot program itself should have sufficient options for choosing that part of the available data to be plotted. For its plot output it must rely on a more basic plotting package, which should be widely available and can produce plot output for all current hardware devices.

1.1 SCALPLO plot options

The FORTRAN-77 program SCALPLO is being developed ~~at IRI to meet the~~ above mentioned requirements. It is designed to make an easy and quick graphic survey of flux and/or power data calculated with SCALE ~~[1]~~ modules or other core calculation or shielding codes. The basic plot functions it can perform are one- and two-dimensional plots of flux or power distributions and flux energy spectra. More specifically it can produce plots of the flux distribution in a one-dimensional geometry for one or more energy groups in one figure. It can also plot the flux distribution along a cut through a two- or three- dimensional geometry along one of the coordinate axes and it can plot a two-dimensional view of the flux distribution of a two-dimensional geometry or of a plane cut through a three-dimensional geometry. The same can be done for the power distribution in a system. Furthermore SCALPLO can plot the particle flux spectrum as a function of energy, either as group fluxes or as group fluxes per unit energy or per unit lethargy.

The control input to SCALPLO offers options to select only part of the coordinate axes to plot the flux or power distribution in order to zoom in on the geometry of the system. Likewise, the energy range for flux spectra can also be chosen by input. For the two-dimensional plot the viewing point can also be chosen by input. The overall scale of a plot can, of course, be controlled by input as well as text to be plotted in a figure and along the axes, the character style and height. When more than one line is drawn in a picture, a legend can also be produced in the plot. The control data is read from an input file which structure consists of input blocks controlling the preprocessor (to read the plot data) and input blocks controlling the plot section of the SCALPLO program (see figure 1).

Although SCALPLO was developed for easy plotting of flux profiles and energy spectra, it is also suitable as a general plotting program for all kind of data which can be represented on a X-Y scale.

The plot module is based on the DISSPLA FORTRAN graphics library [2]. This package is widely used in reactor physics codes like the SCALE system and NJOY code. It provides a variety of subroutines for creating complete pictures for data representation, including annotated axes, text and legends. It is virtually device independent as it can produce plot output for a graphics terminal as well as for many hardware plotting devices using a variety of plot output standards, among which GKS. DISSPLA is almost computer independent. Only its initialization is installation dependent.

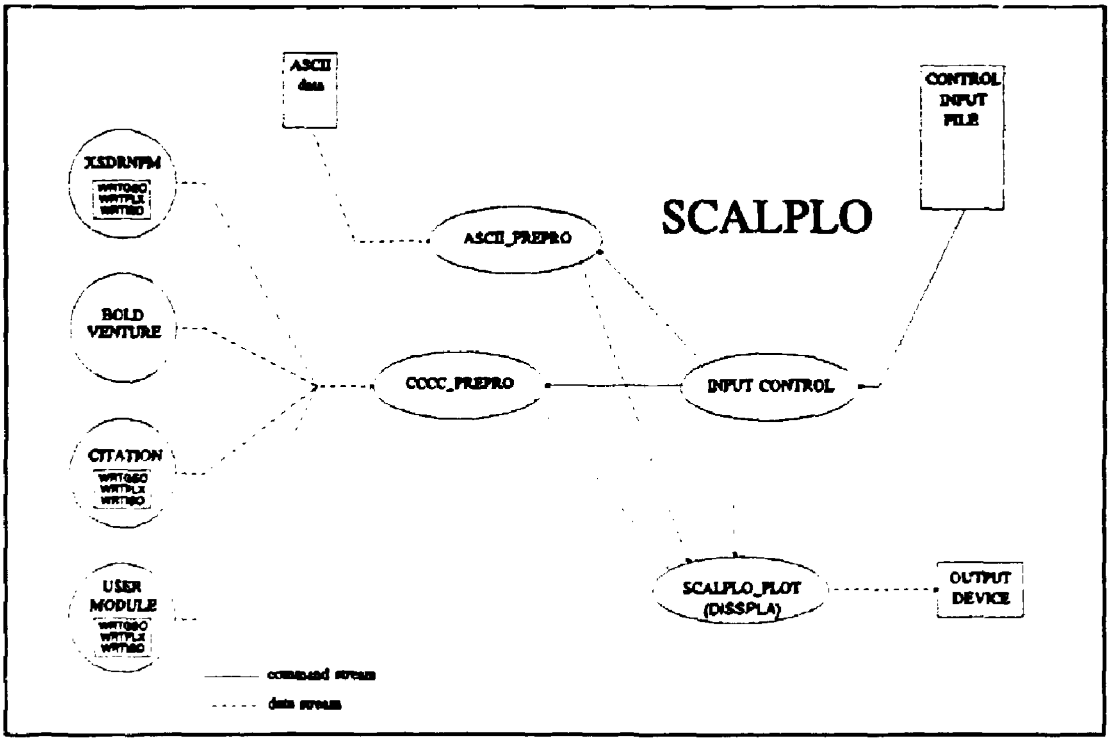


Fig. 1. Schematic diagram of the SCALPLO program

2. Environment and installation

SCALPLO is a universal FORTRAN-77 program based on DISSPLA version 11.0, the graphical package used in several modules of the SCALE system and NJOY code. It is therefore necessary to have this package installed on your system. Two routines in SCALPLO are not universal F-77; subroutine INISYS and INPCON. INISYS is a routine where DISSPLA and the output plot device are initialized. At IRI this is a subroutine developed by the system manager. It is device and system dependent, so no universal routine can be supported. The INPCON routine is the input routine that reads the input control file and starts the different SCALPLO modules. It uses namelist-directed read statements for user friendly input of control parameters. The namelist option is not a standard FORTRAN feature, but is available in almost all F-77 compilers on different computers.

3. Description of input data files

SCALPLO can read ASCII and CCCC interface file format. The ASCII input file type is used to enter X/Y data to plot, whereas the CCCC interface file format is used to read flux, spectrum and geometry data produced by another program.

3.1 The ASCII input file

The ASCII input format is a character file with a comment line as first line. The X/Y data follows at every new line for every new point. This X/Y data can be separated by either blanks or a comma. The ASCII preprocessor will automatically determine the number of data points in the ASCII file.

3.2 Interface files with CCCC format

For the format of the data files containing the flux and geometry data needed for a plot, the CCCC format was chosen. CCCC stands for Committee on Computer Code Coordination. This committee established standards and procedures to facilitate the development of readily exchangeable reactor physics codes [3]. Part of this effort was the definition of standard data interfaces between codes to facilitate linking the output of one code to the input of another code. The CCCC format was chosen because it is a well-considered and well-documented standard that covers all kind of data that may be used in reactor physics codes. In fact, part of the CCCC interface files are already programmed for output in some SCALE modules and other codes and several subroutines defined by the standard to facilitate reading and writing interface files are widely used in SCALE modules and other codes. The current version of the standard (Version IV) was established in 1976.

The CCCC standard defines files for different types of nuclear data, reactor specification and particle and power distribution data. For the plotting program SCALPLO the flux distribution file RTFLUX and the geometry description file GEODST are the most important ones. For neutron spectrum plots the ISOTXS file for multi group cross section data is used to retrieve the energy group boundaries.

The general setup for CCCC interface files is as follows. The first record is the file identification. It contains the CCCC file type, the user identification and the version number. The second record is the file control data. It contains specific constants for the CCCC file. The third and following records are data records.

The control data of the flux distribution file contains the number of geometric dimensions of the problem (1-3), the number of energy groups and the number of mesh intervals per dimension. The geometry description file GEODST allows for a large number of different geometries in one, two and three dimensions with rectangular, cylindrical, spherical, hexagonal and other coordinates.

4. Making a plot with SCALPLO

4.1 Introduction

The SCALPLO program opens an ASCII control input file on FORTRAN unit LUNIN (5) to read the control variables to create a plot. To make a plot with SCALPLO it is necessary to enter the sections global control, preprocessor control and plot control. The first control section initializes some global variables, like page size, character style, etc. to define the working space on the output device; this is described in 4.2. The second control section describes the kind of preprocessor to get the right plot data from the data input file(s); this is explained in 4.3. The third and last section is the actual plot control control section. Here the variables are entered to draw the plot on the output device (chapter 4.4). Most variables from the input sections have default values, shown in Tables 1, 2 and 3, respectively, others have to be entered.

As far as possible all the variables used in SCALPLO have the same name as those in the DISSPLA manual, but some are slightly altered. The connection between SCALPLO and DISSPLA functions and variables is shown in Table 7, Appendix C. The two basic commands of SCALPLO are START and END. The first command starts a control section like: GLOBAL (chapter 4.2), PREPRO (chapter 4.3) and PLOT (chapter 4.4). Whereas END is used to end a control section, a plot section, or the SCALPLO program. Remember that, whereas the SCALPLO commands START and END have to start at the first position of a line, the namelist input block has to start at the second position of a line. The namelist starts with a $ or & (with IBM compilers use only $).

4.2 The global control section

The first line of a control file must be START SCALPLO. This command starts the namelist input of the global variables shown in Table 1. These variables can be entered in random order. The namelist input starts with $GLOBAL at the second position of the next line. Now enter the appropriate variables according to the namelist input rules (see your FORTRAN compiler documentation) and end it with $END (except for IBM you can also use & in case of $). When no control variables have to change it is permitted to enter an empty namelist: $GLOBAL $END.

The complete list of namelist variables of this section is listed in the output file at unit number LUNLIS. Every variable in the global section has a default value (listed in Table 1) and is given in brackets. Values are kept until they are changed. In the next chapter the control variables are explained in order of their function in SCALPLO.

4.2.1 SCALPLO unit numbers

LUNMSH [8] unit number of the geometry interface file GEODST for the CCCC preprocessors.

LUNDAT [9] unit number of the data interface file. The ASCII data file for the ASCII preprocessor or the RTFLUX data file for the CCCC preprocessor.

LUNISO [10] unit number of the ISOTXS interface file for the CCCC spectrum preprocessor.

LUNLIS [6] unit number of the output file used for listing the namelist variables and possible messages from the SCALPLO preprocessors.

LUNERR [6] unit number for SCALPLO error messages.

4.2.2 Setting up a plot space on the output device

The plot is drawn on a plot page. The dimensions of this page are defined by PAGEX and PAGEY. Mostly the plot page match the physical limits of the device, but this is not necessary. The measure of these and other values can be user defined with the SCALE variable, the default is inch. If the exact dimensions of the plot page are not known, the variable LSCALE can be set to 'SCREEN' to ensure that the plot always fits within the device plotting limits.

SCALE [1.0] defines the scale unit, from units to inches. Enter SCALE = 0.3937 to get measures in centimetres. When SCALE is altered, make sure that other values are also in the correct measure.

PAGEX [8.0], PAGEY [11.5] define the physical limits of the plot page. They are device dependent and have to be entered in scale units.

LSCAL ['SCREEN'] defines the method in which the plot area (PAGEX and PAGEY) is transformed to the device hardware plotting surface. The options are:
= 'DOWN', scales down any plot exceeding device plotting limits,
= 'CLIP', clips the plot at device plotting limits,
= 'SCREEN', scales the plot up or down until it fills the plotting surface in at least one direction,
= 'NONE', performs no page scaling or clipping,
= 'ABORT', omits the entire plot if it exceeds either the X or Y dimension of the plotting device.

LROT ['AUTO'] defines the orientation of the plot on paper, the options are:
= 'COMIC', the orientation is horizontal,
= 'MOVIE', the orientation is vertical,
= 'AUTO' , the orientation automatically switches between 'COMIC' and 'MOVIE', depending on the selected output device.

4.2.3 Defining global control data

KDIM [1] defines the plot dimension. KDIM=1 gives a X/Y plot with lines. If KDIM=2 the plot is a X/Y/Z plot with a map.

SHBRD [.TRUE.] logical to show border around plot page, defined by PAGEX and PAGEY.

LSTYLE ['SIMPLX'] defines the character style for the plot. The styles that can be selected are listed in Table 4.

HITE [0.2] height of all the characters in scale units. All the other heights (like header characters) are relative to this height.

XPAGE [6.0], YPAGE [6.0] the length of the X and Y axis if KDIM=1. If KDIM=2 these are the dimensions of the area on which the 3-D axes will be projected (see DISSPLA manual). Enter values in scale units.

SHFRM [.TRUE.] logical to show a frame around the plot area, defined by XPAGE and YPAGE.

SHLEG [.FALSE.] logical to select the legend box
= .TRUE.: draw legend box and enter the parameters LEGNME (section 4.2.3) and LEGTXT (section 4.3.2).

HITLEG [1.0]
height of the characters in the legend box, relative to HITE.

LEGNME [' ']
the name of the legend box (displayed at the top of the legend box), use a maximum of 20 characters.

FRMLEG [0]
draws a frame around the legend box
(see DISSPLA) :
= 1 , draws a frame,
= 0 , does not draw a frame.

If KDIM=2:

Some parameters only affect the plot when KDIM=2. These parameters describe the way in which the 2-dimensional data is represented. For more detail see the DISSPLA manual.

XAXIS [5.0], YAXIS [5.0], ZAXIS [5.0]
the virtual length of the axis in scale units.
This virtual axes system is projected on the plot page (defined by XPAGE and YPAGE) when the plot is drawn.

IXDIM [0], IYDIM [0]
define the size of the 2-dimensional array used by DISSPLA to draw the map (DISSPLA call SURMAT). If 0, SCALPLO uses the same size as the data array in the geometry file.

IXPTS [1], IYPTS [1]
define the grating of the map to be drawn. There is a line every IXPTS-th and IYPTS-th point of the 2-dimensional array defined with IXDIM and IYDIM.

As mentioned before, the namelist input section must be terminated by $END.

4.3 Preprocessor control

The data has to be read before plotting. To read data from different file types, preprocessors are written; routine PREASC for ASCII input and PRECCC for the CCCC interface files. To select a preprocessor enter START PREPRO at the beginning of the first column and $PREPRO on the next line starting at the second position. This will start the namelist input of the variables needed to select the right plot data from an input data file. The data has to be entered according to the namelist input. Close the namelist input with $END. The namelist data is listed on LUNLIS and the selected preprocessor starts to run. End the preprocessor section with the command END PREPRO. Comments from the preprocessor about geometry, etc., will also be listed on LUNLIS. It is possible to enter nothing in the namelist. In that case all the defaults for this section are used.

To plot more than one line in the same plot, repeat the above mentioned commands from START PREPRO as many times as there are lines in the plot (with a maximum of 10 lines). Then give the command to plot (chapter 4.4). The variables which can be used are listed in Table 2. They are explained in the next sections.

4.3.1 Select preprocessor type and optional file names

LPREPR describes the preprocessor to be used (Table 5):
['RTFLU'] = 'ASCII' the ASCII preprocessor, one ASCII input file on unit LUNDAT is read,
= 'RTFLU' the CCCC flux preprocessor, two files are read according to the CCCC format;
GEODST (unit LUNMSH): the geometry description,
RTFLUX (unit LUNDAT): the flux data,
= 'SPECT' the CCCC spectrum preprocessor, three files are read to produce a neutron spectrum;
GEODST (unit LUNMSH): the geometry,
RTFLUX (unit LUNDAT): the flux per group,
ISOTXS (unit LUNISO): the energy group boundaries.

MSHFLE optional file name for the mesh data (see LUNMSH, section 4.2.1).
[EMPTY*]

DATFLE optional file name for the data file (see LUNDAT, section 4.2.1).
[EMPTY*]

ISOFLE optional file name for the ISOTXS CCCC interface file (see LUNISO,
[EMPTY*] section 4.2.1).

*) Name is filled with blanks.

4.3.2 Select plot data

In most cases there is more data in the file than has to be plotted. This section describes the control variables needed to select the data from the interface files.

NGROUP the group number of the data in the flux file (e.g. number of flux group) to
[1] use in the flux plot. Not used with the ASCII preprocessor.

LEGTXT the text displayed in the legend box for the line type belonging to the data
[EMPTY*] specified by the preprocessor run.

XMIN [0.0], XMAX [0.0]

defines the minimum and maximum values of X data to be read from the data file to be plotted. If both XMIN and XMAX are equal, the minimum and maximum values of the geometry data are used.

If KDIM=2 :

YMIN [0.0], YMAX [0.0]

the same as XMIN an XMAX, but for the Y data range.

If 2- or 3-D data is available, the next variables can be used to select a plane or a line from this data.

LPLANE ['XY']

used if there is a data set with 3-dimensional data and a plane of data is to be extracted from that data set to plot.
LPLANE='XY' : extract XY plane at the Z position RPLANE.
LPLANE='XZ' : extract XZ plane at the Y position RPLANE.
LPLANE='YZ' : extract YZ plane at the X position RPLANE.

RPLANE [0.0]

the position of the plane to extract from the 3-D data

LLINE ['X']

used to plot a cross section in the X or Y direction in XY, XZ or YZ planes (X, Y coordinates are used in XY, XZ or YZ planes).
LLINE='X' : extract a line perpendicular to the X axis, at Y = RLINE
LLINE='Y' : extract a line perpendicular to the Y axis, at X = RLINE

RLINE *[0.0]*

the position of the line to extract from a plane,
in case of LLINE='X' ; this is the Y position,
in case of LLINE='Y' ; this is the X position.

N.B.: In case of 1-D input data, this has no effect.
In case of 3-D input data, first select a plane with LPLANE and RPLANE, then treat this plane as input data for LLINE and RLINE)

If the spectrum preprocessor is selected, the next variables are used to control the data to be read.

SPCTYP [1]

selects the spectrum type.
= 1 , group flux,
= 2 , flux per unit energy,
= 3 , flux per unit lethargy.

NGMIN the minimum group number to be read from the flux file in case of a spectrum plot. If zero, the minimum group number of the input data is used.

NGMAX the maximum group number to be read in case of a spectrum plot.
[0] If zero, the maximum group number of the input data is used.

4.4 Plot control

When the data is known to SCALPLO (a preprocessor has been selected), enter the plot control section to specify the plot with START PLOT and $PLOT at the second position of the next line. Now enter in namelist style the control data needed to produce the plot. When this is done, enter $END. The namelist data is written on unit LUNLIS. If an empty namelist is entered, all the defaults for this section are used. The control data (and their defaults) are listed in Table 3 and are explained in the next sections.

If different plots with the same preprocessor data are needed, just re-enter the plot section from START PLOT with the changed control variables. To start a new plot in the same run with new preprocessor data, re-enter the preprocessor and the plot section. This can be repeated as often as desired.

The command END PLOT is used to end the plot on a page. To get more than one plot on a page, first read the data for the first plot, plot this data at the right place on the plot page (use XPHYS an YPHYS) and do <u>NOT</u> enter END PLOT, but enter the preprocessor section to read the data of the second plot. Then plot the data again, but at a different position. If the plot is ready enter the command END PLOT. Remember that, when all the plotting on one page is done, the END PLOT command has to be given to let DISSPLA close the page in an orderly way.

4.4.1 Drawing of the axes

SHAXES selects the plotting of the X, Y and Z axis in the plot,
[.TRUE.] = .TRUE. , draw the axes.

XPHYS the X position of the lower left corner of the plot, relative to the lower left
[-1.0] corner of the plot page (defined by PAGEX and PAGEY). If XPHYS is negative the plot is centred in the middle of the plot page.

YPHYS the Y position of the lower left corner of the plot.

LINXAX defines the X axis type,
[.TRUE.] = .TRUE. , linear X axis,
= .FALSE. , logarithmic X axis.

LINYAX defines the Y axis type (see LINXAX).
[.TRUE.]

LINZAX defines the Z axis type (see LINXAX).
[.TRUE.]

LXNAME text at the X, Y and Z axis, maximum length of 60 characters.
['X-AXIS'],
LYNAME
['Y-AXIS'],
LZNAME
['Z-AXIS']

LOWNAX logical, used to select user defined scaling of the axis,
[.FALSE.] = .FALSE. , auto scaling,
= .TRUE. , user defined scaling. Enter the limits and step size of the markers on the X axis (XORIG, XSTEP and XXMAX).
NB: With logarithmic axes enter only XORIG and XXMAX.

LOWNAY logical, used to select user defined scaling of the Y axis,
[.FALSE.] = .FALSE. , auto scaling,
= .TRUE. , user defined scaling. Enter the limits and step size of the markers on the Y axis (YORIG, YSTEP and YYMAX).
NB: With logarithmic axes enter only YORIG and YYMAX.

If LOWNAX=.TRUE. :

XORIG defines the X origin of the axis. Values in axis units, depending on the geometry of the data.

XSTEP defines X step size (between tick marks) in axis units, with linear axis only.

XXMAX defines the maximum X value at the axis in axis units.

If LOWNAY=.TRUE. :

YORIG defines the Y origin of the axis. Values in axis units, depending on the geometry of the data.

YSTEP defines Y step size (between tick marks) in axis units, with linear axis only.

YYMAX defines the maximum Y value at the axis in axis units.

If KDIM=2 :

XABS [-50.0], YABS [-50.0] ZABS [20.0]
define the viewpoint in absolute 3-D coordinates, relative to the position of the zero crossing of the axis. These values are in scale units. For more information see the DISSPLA manual.

4.4.2 Plotting the data

IMARK [0]
specifies the curve marker type and the connection of points,
= 0 , points connected with no symbols drawn,
= +I , points connected and a symbol every I-th point,
= -I , points not connected but a symbol every I-th point.

LINTYP ['LINEAR']
describes the representation of the line; see table 6.

If LINTYP='BARS' :

BARWTH [-0.5]
the width parameter for the bars,
> 0 , width of bars in scale units,
< 0 , absolute value of the ratio of the bar width to the gap width.

If LINTYP='RASPLN' :

TENSN [0.0]
a positive 'tension' number that determines the extent of smoothing in case of rational spline interpolation of a 1-D curve.

If KDIM = 2 :

IXMDF [1], IYMDF [1]
used for smoothing of the map to be plotted, together with WEIGHT (see DISSPLA manual).

WEIGHT [2.0]
weight factor for smoothing.

4.4.3 The header and legend lines

Header lines are always centred above the plot, whereas the legend can be placed at a user defined position. If the header is selected (HEADER=.TRUE.) there will always be four lines plotted.

HEADER [.FALSE.]
logical to print the header,
= .TRUE. ; print header.

HEAD 4*[' ']
array holding the header text; HEAD(1..4). Use a maximum of 60 characters per line. The header is centred above the plot.

HTMULT [1.0] [0.75] [0.75] [0.5]
array with the height of the header lines; HTMULT(1..4). This height is relative to the character height HITE (section 4.2.3).

XLEG [-1.0], YLEG [0.0]
the position of the legend box on the plot page (origin is in lower left corner of the plot area, defined with XPAGE and YPAGE). Values in scale units. Default position of the legend box (selected when XLEG is negative) is centred at the top of the plot area, defined by XPAGE and YPAGE.

5. Examples

5.1 Single curve, ASCII preprocessor

This first example shows the use of the ASCII input processor. The data is put in an ASCII file according to chapter 3.1. This input file is shown in appendix A.
The control input file and the plot are on the next pages.

In the global section the dimension is set to 1, and the character style is set to COMPLX. Furthermore the dimensions of the plot page and area are changed to suite the plot. In the preprocessor section the preprocessor is set to ASCII and the data input file is set to 'SCALPLO$REFCOPY:TEST_1.DAT'. In the plot section the header and axes text are set and the line type is 'SPLINE'. The axes are user defined and the appropriate values are set.

```
START SCALPLO
 $GLOBAL
    KDIM   = 1
    LSTYLE = 'COMPLX'  LSCAL  = 'NONE'
    PAGEX  = 8.0       PAGEY  = 5.5
    XPAGE  = 6.0       YPAGE  = 2.5
 $END
START PREPRO
 $PREPRO
    LPREPR = 'ASCII'
    DATFLE = 'SCALPLO$REFCOPY:TEST_1.DAT'
 $END
END PREPRO
START PLOT
 $PLOT
     LINTYP    = 'SPLINE'
     HEADER    = .TRUE.
     HEAD(1)   = 'pressure profile'
     HEAD(2)   = 'model rocket motortest'
     HEAD(3)   = 'fuel: Kalinitrox with additives'
     HEAD(4)   = 'NAVRO - Holland'
     LXNAME    = 'time [seconds]'
     LYNAME    = 'pressure [bar]'
     LOWMAX    = .TRUE.  ,  LOWNAY    = .TRUE.
     XORIG     = 0.0     ,  XSTEP     = 5.0    ,   XXMAX     = 25.0
     YORIG     = 0.0     ,  YSTEP     = 2.5    ,   YYMAX     = 7.5
 $END
END PLOT
END SCALPLO
```

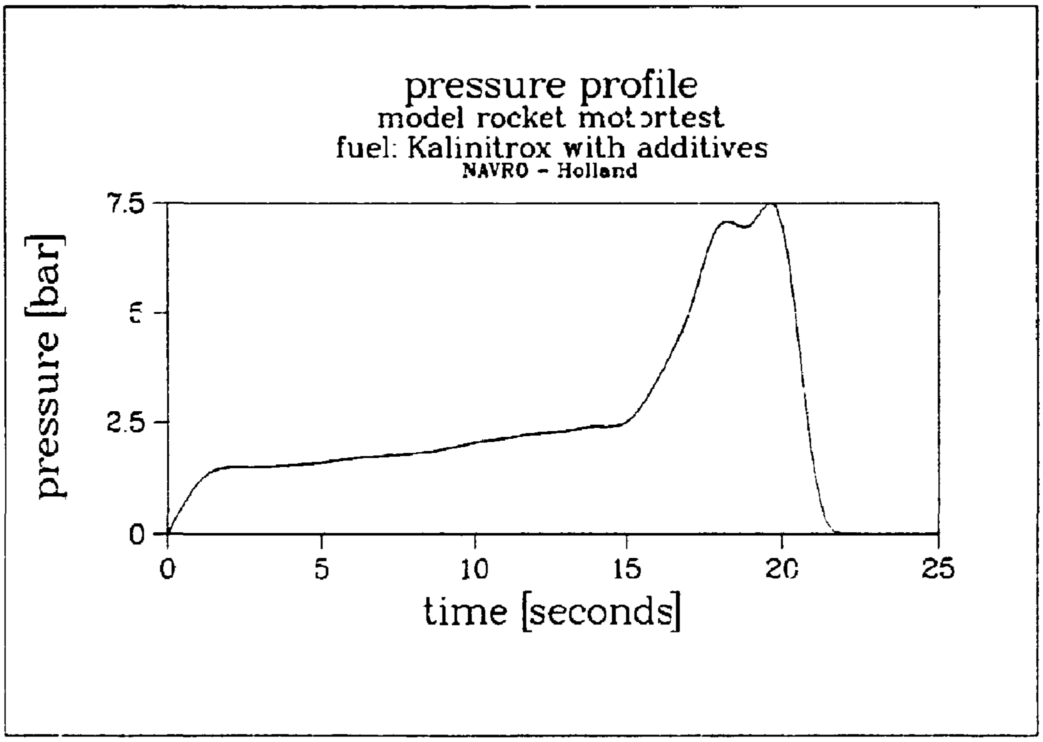


fig. 2. Example 1: Single curve with the ASCII preprocessor

5.2 Multi curve, CCCC preprocessor

The next plot data is obtained from a 172 group XSDRNPM calculation. To produce the CCCC interface files, the supplied subroutines WRTFLX and WRTGEO have been added to the XSDRNPM source (Chapter 6.4). The interface files are binary files and therefore not listed in this manual. The control input file is listed below.

```
START SCALPLO
 $GLOBAL
  KDIM = 1
  LSTYLE = 'COMPLX'  LSCAL = 'NONE'
  PAGEX = 8.8        PAGEY = 6.0
  XPAGE = 6.0        YPAGE = 2.5
  SHLEG = .TRUE.     LEGNME = 'GROUP'
  HITLEG = 0.6
 $END
START PREPRO
 $PREPRO
  LPREPR = 'RTFLU'
  MSHFLE = 'SCALPLO$REFCOPY:TEST_10.GEO'
  DATFLE = 'SCALPLO$REFCOPY:TEST_10.FLX'
  NGROUP = 30
  LEGTXT = 'nr.  30 (fast)'
 $END
END PREPRO
START PREPRO
 $PREPRO
  NGROUP = 160
  LEGTXT = 'nr. 160 (thermal)'
 $END
END PREPRO
START PLOT
 $PLOT
  LXNAME = 'X dimension'
  LYNAME = 'flux'
  HEADER = .TRUE.
  HEAD(1) = 'EPRI/PWR S8P3  172 groups'
  HEAD(2) = 'IRI'
  HEAD(3) = 'TUDelft'
  LOWNAX = .TRUE.   XORIG = 0.1   XSTEP = 0.1   XXMAX = 0.9
  LOWNAY = .TRUE.   YORIG = 0.1   YSTEP = 0.05  YYMAX = 0.2
 $END
END PLOT
END SCALPLO
```

This input file produces figure 3.

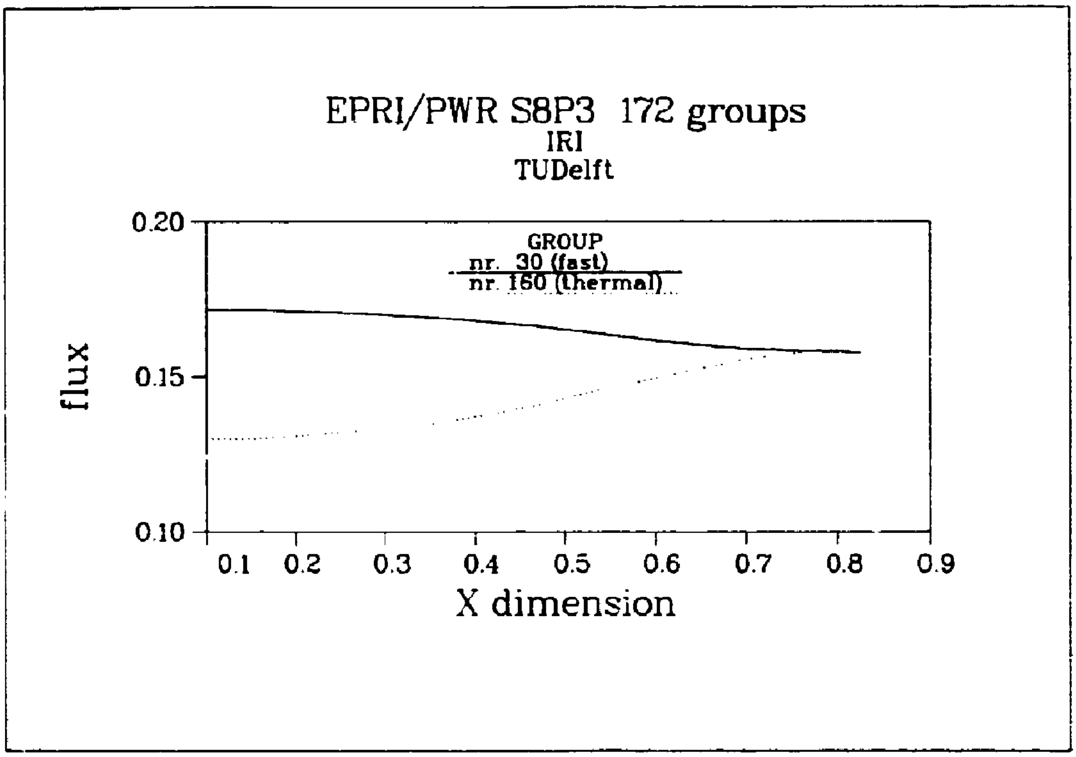


Fig. 3. Example 2: Radial flux distribution in a pin cell

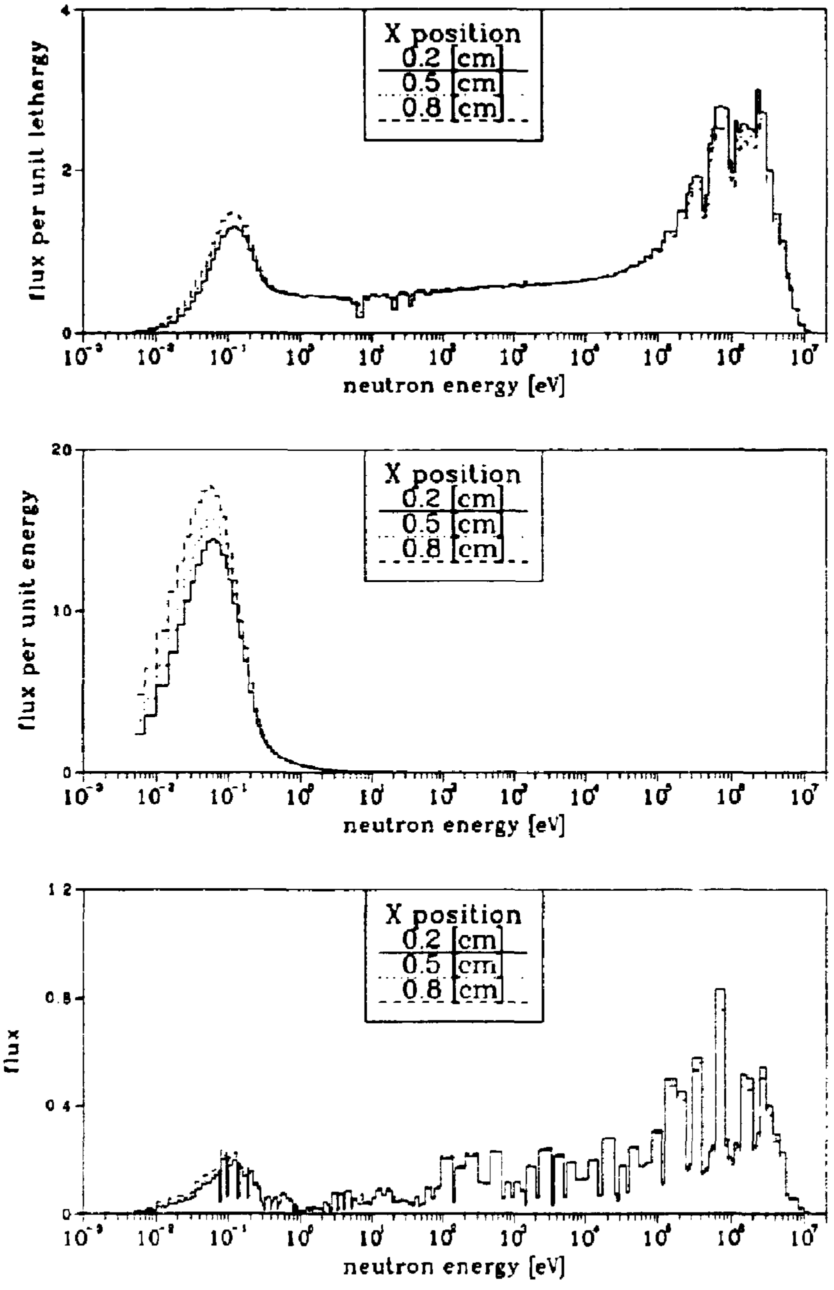
5.3 Multi curve, multi spectrum plot

The third example shows the possibilities of the spectrum preprocessor. There are three different spectra to plot, the flux, the flux per unit energy and the flux per unit lethargy. Every plot shows the spectrum at three different positions. The input data files are the RTFLUX, the GEODST and the ISOTXS files from the XSDRNPM calculation of the previous example. The input control file is listed below.

```
START SCALPLO
 $GLOBAL
  LSTYLE='COMPLX'
  SCALE=0.3937, PAGEX=17.5, XPAGE=13.0, PAGEY=26.0, YPAGE=5.5, HITE=0.6
  LSCAL='NONE', SHLEG=.TRUE., FRMLEG=1, HITLEG=0.5, LEGNME='X position'
 $END
START PREPRO
 $PREPRO
  LPREPR='SPECT' ,DATFLE='SCALPLO$REFCOPY:TEST_10.FLX'
  MSHFLE='SCALPLO$REFCOPY:TEST_10.GEO'
  ISOFLE='SCALPLO$REFCOPY:TEST_10.ISO'
  SPCTYP=1, XMIN = 0.2, LEGTXT='0.2 [cm]'
 $END
END PREPRO
START PREPRO
 $PREPRO  XMIN = 0.5  LEGTXT='0.5 [cm]'  $END
```

```
END PREPRO
START PREPRO
 $PREPRO   XMIN = 0.8   LEGTXT='0.8 [cm]'   $END
END PREPRO
START PLOT
 $PLOT
  XPHYS=2.0, YPHYS=1.5, LINXAX=.FALSE., LXNAME='neutron energy [eV]'
  LYNAME='flux', HITAX=0.4
 $END
START PREPRO
 $PREPRO   SPCTYP=2     XMIN = 0.2   LEGTXT='0.2 [cm]'   $END
END PREPRO
START PREPRO
 $PREPRO   XMIN = 0.5   LEGTXT='0.5 [cm]'   $END
END PREPRO
START PREPRO
 $PREPRO   XMIN = 0.8   LEGTXT='0.8 [cm]'   $END
END PREPRO
START PLOT
 $PLOT
  YPHYS=9.0, LINXAX=.FALSE., LYNAME='flux per unit energy'
 $END
START PREPRO
 $PREPRO   SPCTYP=3     XMIN = 0.2    LEGTXT='0.2 [cm]'   $END
END PREPRO
START PREPRO
 $PREPRO   XMIN = 0.5   LEGTXT='0.5 [cm]'   $END
END PREPRO
START PREPRO
 $PREPRO   XMIN = C.8   LEGTXT='0.8 [cm]'   $END
END PREPRO
START PLOT
 $PLOT
  HEADER=.TRUE., HEAD(1)='EPRI/BWR S8P3'
  HEAD(2)='multi plot, multi curve', HEAD(3)='spectrum plot'
  YPHYS=16.5, LINXAX=.FALSE., LYNAME='flux per unit lethargy'
 $END
END PLOT
END SCALPLO
```

EPRI/BWR S8P3
multi plot, multi curve
spectrum plot



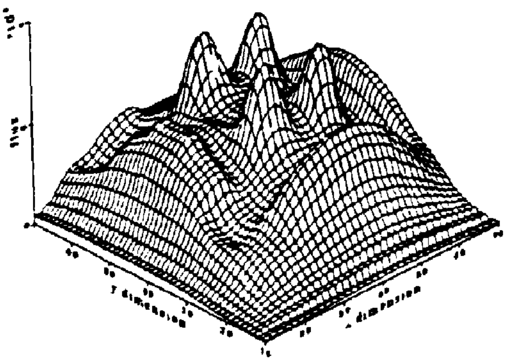
5.4 Two dimensional, multi plot

At IRI the routines WRTGEO and WRTFLX are also added to the CITATION code. This CITATION code is used for whole core calculations of the research reactor HOR. This reactor core consists of 32 MTR-type fuel assemblies, four of which may contain a control rod. The core is surrounded at one side by beryllium reflector elements. At all sides a thick water reflector is present. The fuel assemblies have been homogenized in this calculation, so that there is no fine structure of the flux through the plates of a fuel assembly. As the control rods were assumed to be fully withdrawn from the core with their space replaced by water, a considerable flux peaking can be seen at the control elements in the upper plot, whereas in the lower plot some dips can be seen in the fast flux profile. The input control file is printed below.

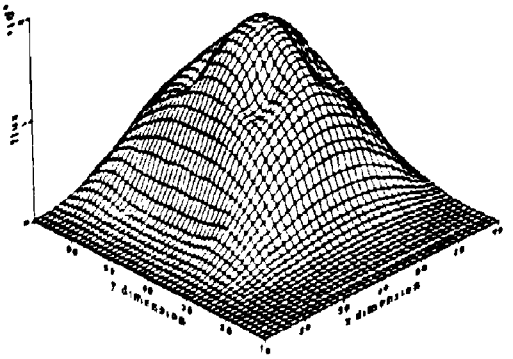
```
START SCALPLO
 $GLOBAL
 KDIM=2, IXPTS=2, IYPTS=2, HITE=0.5, LSTYLE='COMPLX'
 SCALE=0.3937, PAGEX=17.0, PAGEY=26.0, XPAGE=15.0, YPAGE=9.0
 XAXIS=6.0, YAXIS=6.0, ZAXIS=4.0
 $END
START PREPRO
 $PREPRO
 MSHFLE='SCALPLO$REFCOPY:TEST_12.GEODST'
 DATFLE='SCALPLO$REFCOPY:TEST_12.RTFLUX'
 NGROUP=1
 $END
END PREPRO
START PLOT
 $PLOT
 XPHYS=1.0, YPHYS=1.0, HITAX=0.3
 LOWNAX=.TRUE., XORIG=10.0, XSTEP=10.0, XXMAX=70.0
 LOWNAY=.TRUE., YORIG=10.0, YSTEP=10.0, YYMAX=70.0
 IXMDF=2, IYMDF=2, WEIGHT=2.0, ZABS=40.0
 HEADER=.TRUE., HEAD(1)='CITATION calculation, fast flux'
 LXNAME='x dimension', LYNAME='y dimension', LZNAME='flux'
 $END
START PREPRO
 $PREPRO
 MSHFLE='SCALPLO$REFCOPY:TEST_12.GEODST'
 DATFLE='SCALPLO$REFCOPY:TEST_12.RTFLUX'
 NGROUP=5
 $END
END PREPRO
START PLOT
 $PLOT
 YPHYS=13.5, HEAD(1)='CITATION calculation, thermal flux'
 HEAD(2)='Interfaculty Reactor Institute'
 HEAD(3)='Delft University of Technology'
 HEAD(4)='The Netherlands'
 $END
END PLOT
END SCALPLO
```

CITATION calculation, thermal flux

Interfaculty Reactor Institute
Delft University of Technology
The Netherlands



CITATION calculation, fast flux



6. Special subroutines supplied with SCALPLO

Some SCALE modules already provide output capabilities in CCCC format. These are mostly limited to ISOTXS cross section data files (e.g. XSDRNPM). The 1-, 2- and 3-D diffusion code BOLD VENTURE [5] already includes the RTFLUX flux file and GEODST geometry description file. If the required CCCC interface files are not available in a flux calculation module, a few special subroutines are included in the SCALPLO package which will create the appropriate CCCC files by a call to that routine.

Although the CCCC standard defines the ATFLUX interface file for adjoint functions (which differs from the RTFLUX file in the order the energy groups are listed), the ATFLUX file is not used with the SCALPLO program. For plotting, the adjoint function can be treated exactly as the regular flux and can also be output with the same call to subroutine WRTFLX.

6.1 The geometry file GEODST

To create a geometry description file the subroutine WRTGEO is available with the following arguments (the variable names are taken from the CCCC description)

```
SUBROUTINE WRTGEO(LUN, HUSE, IGOM, NCINTI, NCINTJ, NCINTK,
                  XMESH, YMESH, ZMESH)
```

The arguments have the following meaning:

LUN   Unit number; user defined.
HUSE   User identification; user defined. Array with two character elements: CHARACTER*8 HUSE(2), but according to the description of the CCCC interface structure, only the first 6 characters of each one are in use.
IGOM   Geometry type according to CCCC GEODST description.
 1 = slab
 2 = cylinder
 3 = sphere
 6 = X-Y
 7 = R-Z
 8 = theta-R
 9 = uniform triangular
10 = hexagonal
11 = R-theta
12 = R-theta-Z
13 = R-theta-alpha
14 = X-Y-Z
NCINTI   Number of first dimension mesh intervals.
NCINTJ   Number of second dimension mesh intervals,
 = 1 if 1-D mesh.
NCINTK   Number of third dimension mesh intervals,

= 1 if 1- or 2-D mesh.

XMESH Mesh boundaries in the X direction.
YMESH Mesh boundaries in the Y direction.
ZMESH Mesh boundaries in the Z direction.

For including a call to subroutine WRTGEO in an existing SCALE module or other flux calculation code, the above defined arguments must be replaced by actual variable names used in that code. For the output logical unit number LUN a logical unit number not yet in use in that program can be chosen. The contents of the character string HUSE is up to the user as it is used for identification only. Some knowledge of the code under consideration is required to get the correct variable names and the place to enter the described calls. Examples for some codes will follow in chapter 6.4.

6.2 The flux file RTFLUX

To create a RTFLUX data file the subroutine WRTFLX is available with the following arguments

```
SUBROUTINE WRTFLX(LUN, HUSE, IGOM, NGROUP, NINTI, NINTJ,
                  NINTK, FREG, MULT, NCTRL)
```

The arguments have the following meaning:

LUN Unit number; user defined.
HUSE User identification (same as above).
IGOM Geometry type according to CCCC GEODST description.
NGROUP Number of energy groups.
NINTI Number of first dimension mesh intervals.
NINTJ Number of second dimension mesh intervals.
NINTK Number of third dimension mesh intervals.
FREG Multi dimensional (regular or adjoint) flux by interval and group.
MULT Description of precision of FREG,
= 1 : FREG is single precision data array
= 2 : FREG is double precision data array, but will be written as single precision.
NCTRL Control variable to select the 'output mode'
= 0 : close unit LUN
= 1 : write RTFLUX in one call
(open, write, close)
= 2 : write identification records (open,write)
use only with a 2 or 3 dimensional flux array
NGROUP must be the maximum number of groups to follow before closing the file
= 3 : write flux to the file for NGROUP groups (first open with NCTRL=2, then write flux with one or more calls with NCTRL=3 and close unit with NCTRL=0).

6.3 The cross section file ISOTXS

To make a spectrum plot, the energy boundaries of the flux groups are needed. This data is available in the CCCC file ISOTXS. If a program cannot produce this file, the routine WRTISO is available to make a ISOTXS-like file. In XSDRN these boundaries are available, so the call to WRTISO can be added to the source. In ANISN and DOT these boundaries are not available in the code. The subroutine WRTISO has the following arguments

```
SUBROUTINE WRTISO(LUN,HUSE,NGROUP,EMAX,EMIN)
```

LUN	Unit number; user defined.
HUSE	User identification (same as with WRTGEO)
NGROUP	Number of energy groups.
EMAX	Array with energy boundaries, NGROUP elements.
EMIN	Minimum energy boundary.

6.4 Implementing WRTGEO, WRTFLX and WRTISO in different codes

To create a GEODST and a RTFLUX file with the SCALE module XSDRNPM the following calls can be placed in subroutine OUTPUT

```
      CALL WRTGEO(66,'XSDRNP  M-S      ',IGE,IM,1,1,D(LR),DUM,DUM)
      CALL WRTFLX(67,'XSDRNP  M-S      ',IGE,IGM,IM,1,1,D(LXN),2,1)
```

resulting in a GEODST interface file written on logical unit 66 and a RTFLUX file on unit 67. These units are chosen because they are unused units in the SCALE-4 system. The variable DUM is a dummy argument passed to the routine. As can be seen from the second last argument of WRTFLX, the data in the flux-array is double precision and has to be converted into single precision before it is written (MULT=2). The last argument of WRTFLX (NCTRL) equals 1, so the whole RTFLUX file is written in one call to WRTFLX.

To create the ISOTXS file in XSDRNPM the call to routine WRTISO is placed in the subroutine EDIT, where the energy boundaries are known.

```
      CALL WRTISO(68,'XSDRNP  M-S      ',NGN-1,E,E(NGN))
```

resulting in an ISOTXS file is written on FORTRAN unit 68.

In the ANISN code the implementation is almost the same, except for the call to WRTISO (ANISN does not have the energy boundaries). Here the routines are placed in the FINPR1 routine.

```
      CALL WRTGEO(66,'ANISN            ',IGE,IM,1,1,D(LRA),DUM,DUM)
      CALL WRTFLX(67,'ANISN            ',IGE,IGM,IM,1,1,D(LXN),2,1)
```

This also results in writing a GEODST file on unit 66 and a RTFLUX file on unit 67.

More changes are needed for the codes DOT and CITATION to produce the correct output. This is caused by the way in which the flux data is available, on a scratch unit or in an array. The 2-dimensional DOT code is changed in the routine S8850, where the next lines are added after the call to WOT10

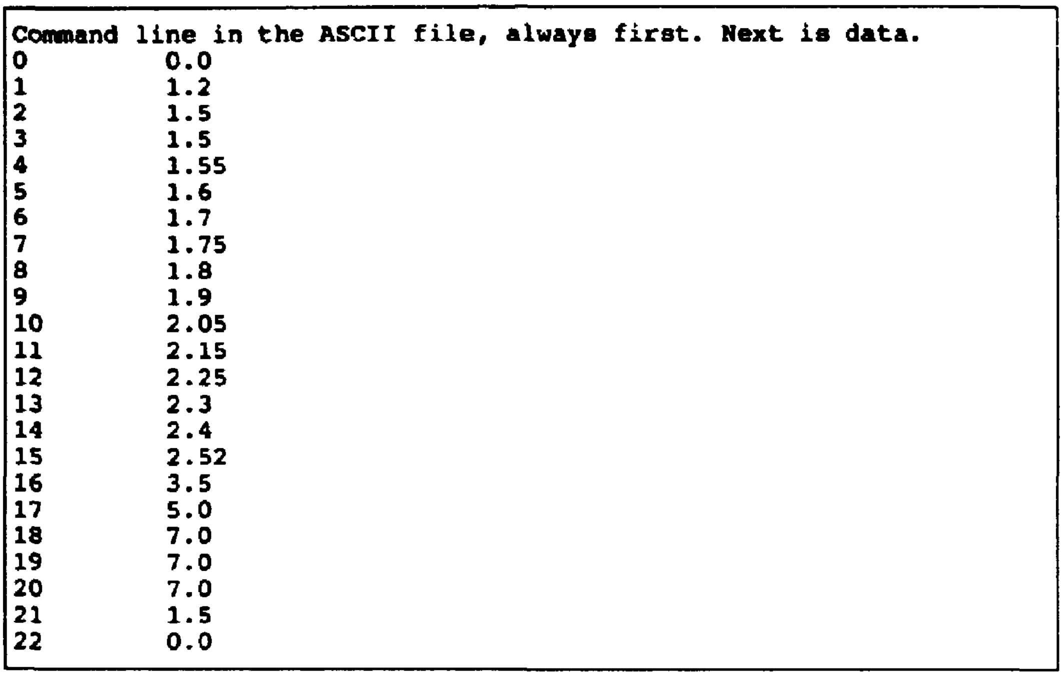
```
CSCALPLO
      IDUM1=IGE+6
      IF (IGE.EQ.3) IDUM1=11
      CALL WRTGEO(66,'DOT 3.5         ',IDUM1,IM,JM,1,R1,Z1,RDUM)
      CALL WRTFLX(67,'DOT 3.5         ',IDUM1,IGM,IM,JM,1,N2,1,2)
      DO 1051 IIG=1,IGM
      IF(IFOT.EQ.0) GO TO 1451
      IG1 = 1
      CALL WANDR1(NFLUX1,N2(1,1,IG1),IMJM,2)
 1451 CALL WRTFLX(67,'DOT 3.5         ',IDUM1,1,IM,JM,1,N2(1,1,IIG),1,3)
 1051 CONTINUE
      CALL WRTFLX(67,'DOT 3.5         ',IDUM1,1,IM,JM,1,N2,1,0)
      REWIND NFLUX1
CSCALPLO-END
```

Firstly the geometry definition is recalculated into the CCCC description, then the GEODST file is written. Next the first two records of RTFLUX are written (NCTRL=2) and then the loop over IGM number of groups is entered. Now data is written with NCTRL=3 and at the end the RTFLUX file is closed with NCTRL=0 and the flux scratch file is rewound. Note that the flux is written by group, so NGROUP must be 1.

The following changes are needed for CITATION. The common block AMESH has to be included in the chosen subroutine EIGN to write the geometry. The next source lines are to be written at the end of the EIGN routine.

```
CSCALPLO           WRITE GEODST TO UNIT 90
      IGOMSC = NUAC(5)
      IF (NUAC(5).EQ.9)  IGOMSC = 10
      IF (NUAC(5).EQ.10) IGOMSC = 9
      IF (NUAC(5).EQ.11) IGOMSC = 14
      CALL WRTGEO(90,'CITATI  ON      ',IGOMSC,JMAX,IMAX,KBMAX,
     +           X,Y,Z)
      IF (IGOMSC.LT.11) THEN
CIRIVAX - VAX ONLY
C  CONVERT G_FLOAT TO D_FLOAT, IF G_FLOAT IS USED
         ICOUNT = JIDX*KBDX*KDX
         CALL MTH$CVT_GA_DA(F2,P2,ICOUNT)
C  WRITE FLUX OUTPUT TO UNIT 91
         CALL WRTFLX(91,'CITATI  ON      ',IGOMSC,KMAX,JMAX,IMAX,
     +   1,P2,2,1)
C  CONVERT BACK TO G_FLOAT
         CALL MTH$CVT_DA_GA(P2,P2,ICOUNT)
      ELSE
         CALL WRTFLX(91,'CITATI  ON      ',IGOMSC,KMAX,JMAX,IMAX,
     +   KBMAX,P2E,1,1)
      ENDIF
CSCALPLO-END
```

Appendix A. ASCII input file, example 1



```
Command line in the ASCII file, always first. Next is data.
0         0.0
1         1.2
2         1.5
3         1.5
4         1.55
5         1.6
6         1.7
7         1.75
8         1.8
9         1.9
10        2.05
11        2.15
12        2.25
13        2.3
14        2.4
15        2.52
16        3.5
17        5.0
18        7.0
19        7.0
20        7.0
21        1.5
22        0.0
```

Appendix B. List of SCALPLO error messages

```
==> ERROR OPENING MESH FILE (LUNMSH) <==
==> ERROR OPENING DATA FILE (LUNDAT) <==
==> ERROR OPENING ISOTXS FILE (LUNISO) <==
```

Error occurred in opening one of the units; check file names.

```
==> MESH FILE (MSHFLE) IS NOT A -GEODST- FILE <==
==> DATA FILE (DATFLE) IS NOT A -RTFLUX- FILE <==
==> X-SECTION FILE (ISOFLE) IS NOT A -ISOTXS- FILE <==
```

The logical unit is not connected to the right CCCC interface file.

```
==> MESH AND DATA FILE DO NOT COINCIDE <==
```

The CCCC mesh and data file (GEODST & RTFLUX) are not from the same run, so they can have different geometry and/or different origin.

```
==> ENERGY GROUPS DIFFER IN INPUT FILES <==
```

The input files (GEODST & RTFLUX) have different number of energy groups.

```
==> ERROR OCCURRED OPENING ASCII DATA FILE <==
```

```
==> ERROR OCCURRED READING ASCII DATA FILE <==
```

```
==> WARNING, ARRAY TO BIG TO MAKE SPECTRUM <==
```

There is too much data in the data array to convert to a spectrum, not all the data is converted. There is a maximum of 5000 points to convert to a spectrum. Change input boundaries in the preprocessor control section.

Appendix C. Tables

Table 1

global control data

```
START SCALPLO
 $GLOBAL
   variable = data
  . . .
 $END
```

variable	default value	description
FRMLEG	0	draws a frame around the legend box
HITE	0.2	height of characters in plot (in scale units)
HITLEG	1.0	height of the characters in the legend box, relative to HITE
IXDIM	0	number of 2-D X array elements if 0, the number of the mesh data array is used
IXPTS	1	defines the grating of the map to be drawn
IYDIM	0	number of 2-D Y array elements if 0, the number of the mesh data array is used
IYPTS	1	defines the grating of the map to be drawn
KDIM	1	plot dimension (1 or 2)
LEGNME	' '	name of legend box (max 20)
LROT	'AUTO'	orientation of the plot page on the device plotting area, selections: 'COMIC', 'MOVIE', 'AUTO'
LSCAL	'SCREEN'	defines the method by which the plot page is scaled to the device plotting surface, selections: 'DOWN', 'CLIP', 'SCREEN', 'NONE', 'ABORT'
LSTYLE	'SIMPLEX'	defines character style of text in plot (see table 4)
LUNDAT	9	unit of data input
LUNERR	6	unit of SCALPLO error messages
LUNISO	10	unit of CCCC interface file ISOTXS
LUNLIS	6	unit of list output
LUNMSH	8	unit of mesh input
PAGEX	8.0	X dimension of the page in scale units
PAGEY	11.5	Y dimension of the page in scale units
SCALE	1.0	scaling from units to inches (inches/unit)
SHBRD	.TRUE.	logical to show border around plot page
SHFRM	.TRUE.	logical to show a frame around the plot area
SHLEG	.FALSE.	logical to select legend box
XAXIS	5.0	virtual length of the 3-D X axis

XPAGE	6.0	length of X axis (if 1-D) or X dimension of area to project 3-D drawing
YAXIS	5.0	virtual length of the 3-D Y axis
YPAGE	6.0	length of Y axis (if 1-D) or Y dimension of area to project 3-D drawing
ZAXIS	5.0	virtual length of the 3-D Z axis

Table 2

preprocessor control data

```
START PREPRO
 $PREPRO
   variable = data
   . . .
 $END
```

variable	default value	description
DATFLE	' '	optional file name for data
ISOFLE	' '	optional file name for CCCC ISOTXS file
LEGTXT	' '	text in legend box for the line read by the preprocessor
LLINE	'X'	direction of line to extract from data, X or Y
LPLANE	'XY'	plane to extract from data
LPREPR	'RTFLU'	preprocessor selection (see table 5)
MSHFLE	' '	optional file name for mesh data
NGROUP	1	select the group number in data file to plot
NGMIN	0	first number of group to plot, in case of spectrum
NGMAX	0	last number of group to plot
RLINE	0.0	location of the line to extract, in mesh units
RPLANE	0.0	position of plane to extract, in mesh units
SPCTYP	1	select spectrum type
XMAX	0.0	maximum X data value to read from data
XMIN	0.0	minimum X data value to read from data
YMAX	0.0	maximum Y data value to read from data
YMIN	0.0	minimum Y data value to read from data

Table 3

plot control data

```
START PLOT
 $PLOT
   variable = data
   . . .
 $END
```

variable	default value	description
BARWTH	-0.5	width parameter used when LINTYP = 'BARS'
HEAD	(1) = ' ' (2) = ' ' (3) = ' ' (4) = ' '	array with header text, maximum of 60 characters per line
HEADER	.FALSE.	logical to print header
HTMULT	(1) = 1.0 (2) = 0.75 (3) = 0.75 (4) = 0.5	height of header line, relative to HITE
IMARK	0	selects the marker type at data points
IXMDF	1	X distance for smoothing of a plane
IYMDF	1	Y distance for smoothing of a plane
LINTYP	'LINEAR'	description of 1-D line type (see table 6)
LINXAX	.TRUE.	logical defining linear or logarithmic X axis
LINYAX	.TRUE.	logical defining linear or logarithmic Y axis
LINZAX	.TRUE.	logical defining linear or logarithmic Z axis
LOWNAX	.FALSE.	logical to select user defined scaling of the X axis
LOWNAY	.FALSE.	logical to select user defined scaling of the Y axis
LXNAME	'X-AXIS'	text next to X axis, max. 60 characters
LYNAME	'Y-AXIS'	text next to Y axis, max. 60 characters
LZNAME	'Z-AXIS'	text next to Z axis, max. 60 characters
SHAXES	.TRUE.	logical to show the X, Y and Z axes
TENSN	0.0	a 'tension' number used when LINTYP = 'RASPLN'
WEIGHT	2.0	weight factor for 2-D smoothing
XABS	-50.0	X distance of view point
XLEG	-1.0	X position of legend box
XORIG		user defined X axis origin
XPHYS	-1.0	X position of the lower left corner of the plot area on the plot page
XSTEP		user defined step size between tick marks on the X axis
XXMAX		user defined maximum value on the X axis
YABS	-50.0	Y distance of view point

YLEG	0.0	Y position of legend box
YORIG		user defined Y axis origin
YPHYS		Y position of the lower left corner of the plot area in the plot page
YSTEP		user defined step size between tick marks on the Y axis
YYMAX		user defined maximum value on the Y axis
ZABS	20.0	Z distance of view point

Table 4

possible character styles to be used with LSTYLE	
type	DISSPLA equivalence
'SIMPLX'	call SIMPLX
'COMPLX'	call COMPLX
'FUTURA'	call FUTURA
'CARTOG'	call CARTOG
'FASHON'	call FASHON
'SCMPLX'	call SCMPLX
'TRIPLX'	call TRIPLX
'GOTHIC'	call GOTHIC
'SWISSL'	call SWISSL
'SWISSM'	call SWISSM
'SWISSB'	call SWISSB
'DUPLX'	call DUPLX
'SERIF'	call SERIF
'LOGO1'	call LOGO1

Table 5

preprocessors	
type	description (see also appendix B)
'ASCII'	ASCII input data file
'RTFLU'	CCCC interface input data files for flux plots, GEODST & RTFLUX
'SPECT'	CCCC interface input data files for spectrum, GEODST, RTFLUX & ISOTXS

Table 6

smoothing technique to be used with LINTYP	
type	DISSPLA equivalence
'LINEAR'	call LINEAR
'STEP'	call STEP
'BARS'	call BARS(BARWTH)
'POLY3'	call POLY3
'SPLINE'	call SPLINE
'RASPLN'	call RASPLN(TENSN)

Table 7

relation between SCALPLO and DISSPLA variable names		
SCALPLO name	DISSPLA name	used in DISSPLA call
FRMLEG	THKNSS	THKFRM
HITE	HITE	HEIGHT
IMARK	IMARK	CURVE
HEAD IHEAD HTMULT	LHEAD IHEAD HTMULT	HEADIN
LXNAME IXNAME	LXNAME IXNAME	XNAME, X3NAME
LYNAME IYNAME	LYNAME IYNAME	YNAME, Y3NAME
LZNAME IZNAME	LZNAME IZNAME	Z3NAME
IXDIM IYDIM IXPTS IYPTS	IXDIM IYDIM IXPTS IYPTS	SURMAT
IXMDF IYMDF WEIGHT	IX IY WEI	MDFMAT
LSCAL	LSCAL	HWSCAL
LROT	LROT	HWROT
PAGEX PAGEY	PAGEX PAGEY	PAGE
SCALE	SCALE	UNITS
XPAGE YPAGE	XAXIS YAXIS	AREA2D
XLEG YLEG	XPOS YPOS	LEGEND
XORIG YORIG ZORIG XSTEP YSTEP ZSTEP XXMAX YYMAX ZZMAX	XORIG,X3ORIG YORIG,Y3ORIG Z3ORIG XSTP,X3STP YSTP,Y3STP Z3STP XMAX,X3MAX YMAX,Y3MAX Z3MAX	GRAF,GRAF3D,XLOG, YLOG,RNDLIN,RNDLOG
XAXIS YAXIS ZAXIS	X3AXIS Y3AXIS Z3AXIS	VOLM3D
XMIN YMIN XMAX YMAX	XMIN YMIN XMAX YMAX	SURSZE

XABS	XABS	VUABS
YABS	YABS	
ZABS	ZABS	

HP 13.11.

References

[1] SCALE-4 - A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation, NUREG/CR-0200 Rev. 4 (ORNL/NUREG/CSD-2/R4) Vols. I, II and III, Oak Ridge National Laboratory, 1990.

[2] CA DISSPLA User Manual, Computer Associates International, Inc., Version 11, 1989.

[3] O'Deli, R.D., Standard Interface Files and Procedures for Reactor Physics Codes, Version IV, Report LA-6941, Los Alamos Scientific Laboratory, 1977.

[4] Flower, T.B., Vondy, D.R., Cunningham, G.W., CITATION; Nuclear Reactor Core Analysis Code, Report ORNL-TM-2496, Rev. 2, Oak Ridge National Laboratory, 1971.

[5] The Bold Venture Computation System for Nuclear Reactor Core Analysis, Version III, Report ORNL-5711 , Oak Ridge National Laboratory, 1984.