IJS Delovno Poročilo
IJS−DP−6948
Izdaja/Revision 5, februar 2009

CORD-2 Package Users' Guide

Andrej Trkov

	Ime in priimek	Datum	Podpis
Izdelal			
Pregledal			
Odobril			

Institut "Jožef Stefan", Ljubljana, Slovenija

Izvajalec:	Institut "Jožef Stefan" 1001 Ljubljana Jamova 39 Odsek za reaktorsko fiziko
Naslov poročila:	CORD-2 Package Users' Guide
Avtorji poročila:	Andrej Trkov
Št. delov. poročila:	IJS−DP−6948
Kopije:	IJS knjižnica (1x) QA arhiv odseka (1x + original) avtorji
Komisija za pregled:	dr. Igor Lengar univ. dipl. fiz. Luka Snoj

Ljubljana, februar 2009

NASLOV POROČILA:

IJS−DP−6948
februar 2009

CORD-2 Package Users' Guide

Andrej Trkov

KLJUČNE BESEDE:

Nuklearni projekt sredice, upravljanje z gorivom, tlačnovodni reaktor

Poročilo je podano v angleščini.

REPORT TITLE:

IJS REPORT 6948
February 2009

CORD-2 Package Users' Guide

Andrej Trkov

KEYWORDS:

Nuclear core design, Core Management, PWR

Contents

1 Introduction

The CORD-2 package is designed to provide a modern, independent calculational tool for reactor core calculations, based on strictly non-commercial software. It provides options that are essential for modelling the advanced features of fuel assemblies. The package provides not only the calculational modules, but also the data management support facilities. It has been implemented on VAX/VMS and on PC/DOS, but extension to other systems is quite straightforward.

2 Program description

The CORD-2 package consists of two basic reactor physics codes, namely WIMS [1] and `GNOMER` [2]. WIMS is a well known and widely used lattice code. Version `WIMS-D/4` is available from the NEA Data Bank in Paris. `GNOMER` solves the neutron diffusion equation in three-dimensional cartesian geometry by the Green's function NOdal MEthod [3]. It also includes advanced features for cross section homogenization and a simple thermohydraulics module from the `CTEMP` code [4], so that thermal feedbacks can be taken into account. The remaining parts of the CORD-2 software are utility codes which perform various functions, such as: library maintenance, data managements and input preparation for the reactor physics codes.

A set of data libraries is defined to contain various classes of data, such as reactor core geometry, material composition etc. All codes of the package share the data from the same libraries. This reduces the user-input to the essential parameters which need to be varied in a calculational sequence and simplifies the quality assurance and quality control procedures. The list of libraries is the following:

DIMlib - dimensions and geometry description,

FASlib - fuel assembly ident, mass, enrichment and burnup history,

CORlib - core distributions data (e.g. power, temperature, burnup etc.),

ISOlib - burnup-history dependent fuel isotopic inventory,

WIMlib - summary file for WIMS material assignment,

RCFlib - reactivity coefficient library.

The libraries depend strongly on the reactor type, for which the calculations are performed. They can not be included in the package, except as a test example for a typical case. They have to be generated by the user. A number of utility codes are available in `CORD-2` to help the user in this task.

3 Prerequisites

It is assumed that the `WIMS-D/4` and `GNOMER` codes are installed and operational. Although they form an essential part of `CORD-2`, they are treated as separate packages and should be obtained from a code distribution centre separately.

The `CORD-2` coding is standard FORTRAN-77 as much as possible, so it is practically machine-independent. Alternate coding is provided where necessary in the FORTRAN source, so that relevant code can be activated or deactivated automatically with the `CNVRTF` code, which is also supplied with the package. The `CNVRTF` code identifies special coding in the source which is to be processed, when it is enclosed in statements of the form

```
C*****nnnn
```

where ”`nnn`” is a 4-character keyword (any subsequent characters are ignored and may be used for comments). At present the following keywords are defined:

`" VAX"` default source code distribution suitable for execution on VAX.

`" VXG"` the G-floating option for VAX. It is related solely to the floating point arithmetic constants in `EILIBE.FUL` and must appear simultaneously with the `" VAX"` option.

`" VXD"` is an alternative to the `"VXG"` option for the D-floating arithmetic compilation. It must appear simultaneously with the `" VAX"` option.

`" STANDARD"` relevant to `OPEN` statements in the `DCKSPL` program only.

`" EILIBE"` in `CORD2.FUL` source to activate calls to the mathematical library routines from LINPACK, EISPACK and BLAS libraries, which are public domain software available through NETLIB and are included on the `EILIBE.FUL` file.

`" IMSL"` in `CORD2.FUL` source if an IMSL mathematical program package library is available on site (it is not part of the package and is fully replaced where necessary by equivalent NETLIB routines, activated with the `EILIBE` keyword.

`" PCN"` applicable to the `EILIBE.FUL` source only, in the case of Lahey, NDP and Professional Fortran compilers.

`" PCM"` applicable to the `EILIBE.FUL` source only, if the user requires the MicroSoft compiler.

A realistic calculation requires a sequence of codes to be executed. To control such calculational sequences, procedures in job-control-language are required. These are strongly machine dependent. The use of such procedures has been minimized as much as possible, but could not be entirely avoided.

The program has been tested on various VAX machines, compiled with standard options, and on a PC with the MicroWay NDP Fortran-486 Ver.4.4.0. Procedures for automatic generation of the executable code on the above configurations is supplied with the package.

The source code on a PC has been generated with 540 Kbytes of memory available. Problems may be encountered if less memory is available.

4 CORD-2 Module description and input instructions

The most recent documentation on individual modules can be found in the source code in comments following the module declaration. Such comments begin with `"C-M"` in the first three columns. Further information is also available in a similar way, as evident from the list below:

`C-T` Title (i.e. name) of the program, function or subroutine,

`C-P` Purpose of the routine, stated in a few words,

`C-D` Description, usually stated for routines of more general interest and importance,

`C-M` Manual of user instructions for the programs and main modules,

`C-A` Author(s) names and affiliation,

`C-V` Version stated as ”yy/mm” with a brief description of the changes to keep track of the revisions.

On VAX, such information can be extracted very conveniently by the system search utility as shown in the example below:

```
"$ SEARCH ’name’.FUL/OUT=’name’.TXT "C-T",C-M","C-V"
```

where `’name’` is the name of the source file. On pages that follow, the users’ guide to individual modules has been extracted directly from the source code and reflects the status of the code at the time when this document was prepared. There may be changes in the source code introduced subsequently. Any changes observed in the source code supersede those in this document.

The codes are listed in alphabetical order, except for the first two which are the Fortran source maintenance codes.

4.1 Manual for Program CNVRTF

The program is intended for FORTRAN source code maintenance. It is assumed that the source code contains special labels which enclose machine-dependent coding which can be activated or de-activated by the CNVRTF code. The label is a record of the form:

  "C*****nnnn"

where the string of 5 characters "C****" identifies the keyword bearing record, the 6-th character is ignored (for backward compatibility with some older codes) and "nnnn" is a 4-character keyword, such as " VAX", " CDC", etc. The remaining characters on the record are not checked. Of the source code sections enclosed by keywords, only those are activated, for which the respective keywords are found on the input list. All others are deactivated.

Processing of the PROGRAM card for CDC requires special processing since it contains information on the files to be used. If the PROGRAM card is the first record on the input file, it is deactivated, except if " CDC" appears on the list of requested keywords.

An additional feature of the code is to label the source code in columns 72-80 by decks, similarly to the conventions of the CDC system UPDATE. If the record begins with "*", the deckname is read from columns 7-10. On output in columns 72-80 the deckname (4-characters) and the record number in that deck are written (the deckname record is the first record in the deck). If no "deck" records are found, the source deck is not labelled.

```
WARNINGS: - Each section of the special code on the source file
            must begin and end with the same keyword,
          - sections with different keywords may not be nested.
```

Limitations:
The code is an extension of the earlier CVRT code. It has been designed to be machine-independent as much as possible. It is a very simple code without extensive checking capabilities, so it is the responsibility of the user to make sure that the statements in the input source file are valid and to adapt the CNVRTF source for his machine if necessary.

Instructions:
Unit-6 is the default output devices (i.e. the terminal screen) on which the input requests are displayed. Unit-5 is the default input device (i.e. the keyboard) from which the input (source code) filename, the modified output code filename and the keywords are read. The filenames may be up to 40 characters long. The keywords are four-character strings and must match those in the source exactly. Up to 20 keywords may be entered.

NOTE: To avoid using non-standard Fortran, the output file is of standard Fortran type. Care must be take when viewing, printing or transfering such files to other systems, not to loose the first character.

4.2 Manual for Program DCKSPL

The FORTRAN source file is assumed to contain records befinning with "*DECK dkname" where 'dkname' is the appropriate name. Each deck is written onto a new file named 'dkname.ext' where 'ext' is the file extension requested from input. If the extension is not "dck" of "DCK", the record "$DECK dkname" is not transferred to output.

An output file DCKLST.TMP is also prepared, on which all the decknames on the source file are listed in the format "CALL FORL dkname". This is to allow the design of batch procedures for automatic compilation of all decks.

Instructions:
The following input parameters are requested from input:

- Source filename
- Extension forsplit deck files.

4.3 Manual for Program CORBRN

The design CORlib file is scanned. The burnup or time dependent parameters such as critical boron concentration [ppm], maximum enthalpy rise Fdh [fraction] or axiall offset [%] are extracted and tabulated on the output file as a function of the cycle burnup [MWd/tU] or time [minutes]. The radial power offset is the average power of the central fuel assemblies compared to the nominal average assembly power. Central assemblies are those for which the distance from the core centre to the assembly centre is less than 1/4 of the diameter of the core

Instructions:
The input and the output filenames are requested from input. The data header comment, which is to appear at the beginning of a data set on the output file, is also requested from input. The burnup or time on the CORlib header cards is assumed monotonic increasing, otherwise a new data set is assumed and a new data header comment is requested from input. The data output format is consistent with the PLOTTAB graphics code.

4.4 Manual for Program CORAXP

The program is designed to process CORlib files which contain core-averaged axial distributions (power, temperature etc.), such as can be produced by CORDSP. Data sets that can not be processed are skipped. The distributions are interpolated to a specified grid and printed in two-column format, which is compatible with the PLOTTAB code.

Assembly-wise axial distributions can also be processed. The assembly consecutive number af tho form "Ass.   n" appears on the header label. Assemblies are numbered left-to-right, top-to-bottom, as ordered in the CORlib file. The numbering depends on the symmetry option in the CORlib file. For example, in quadrant symmetry numbering starts from the central assembly and continues for the bottom-right quadrant.

Note that the whole input CORlib file is scanned and processed. It is the user's responsibility to place the selected data on the CORlib file. It is recommended to use the CORDSP code for the purpose.

Procedures:

The CORlib convention is observed that axial distributions are given in bottom-to-top order using CORlib format "1Tnn", where 'nn' is the number of axial regions. If the x-y format is used, the data are assumed in top-to-bottom order using format "nnT01". The default 10-region relative thicknesses are (1,1,2,4,4,4,4,2,1) while the 24-region mesh is assumed equidistant. For any other axial mesh the region thicknesses must be given explicitly in the CORlib file under keyword "$* REGAX" using format "1Tnn". Different axial mesh representations can be used on the same file, provided the appropriate mesh is defined (default 10 and 24 regions or any other mesh defined with "$* REGAX").

Given coarse mesh interval thicknesses and the average function value in that interval, quadratic splines are constructed such that the function and the first derivative are continuous. In each interval the Legendre polynomial expansion coefficients are determined. By definition, the average function value is the P0 coefficient. A tridiagonal system of equations is solved to find the P2 coefficients. The P1 coefficients can then be determined quite easily.

Having the polynomial expansion coefficients, the smooth function is generated on the fine mesh, defined from input. The derivative is also calculated on the same mesh.

Instructions:

The filenames of the source CORlib file and the output PLOTTAB file may be entered on input interactively to override the defaults. Similarly, the vertical axis coordinates that imply core top and bottom can also be entered. The number of intervals for the output grid can be redefined. The default output interval thickness is such that the last point corresponds to the core bottom, but it may be redefined (be careful of the sign).

NOTE: The quadratic spline interpolation routine works with region thicknesses and not the absolute coordinate values. For generating the function values at a given fine mesh it is assumed that the first point on the list is the origin of the local axes, from which the original coarse mesh axial region increments are defined. This implies that the first point ("top" of the core) is fixed. If the region thickness for the output mesh is changed, the last point need not correspond to the true "core bottom".

4.5 Manual for Program CORCRW

Process the GNOMER output CORlib file. Search for the K-eff printout at various control rod (CR) positions. Convert CR positions to steps and calculate the integral rod worth.
If requested, interpolate by cubic splines to a fixed CR position grid on which the integral and differential worths are given. Otherwise, present data on coarse mesh. Differential worths on the boundary are calculated as the average of the gradients in the adjacent regions.

Instructions:

The source CORlib filename is requested from input. Default output filenames generation is attempted. The filenames may be changed if desired. Since the control rod position in the CORlib file includes the reflector, the assumed reflector thickness, the step position at rods withdrawn and the step size must be specified. If position indication is in "steps withdrawn", then the step size is -ve. Defaults are suggested, but they may be redefined if required.

If spline interpolation is requested, the differential and integral worth values are printed at NPJ equidistant points (default 46). Optionaly, parabolic fit can be performed on differential values. This procedure mitigates to some extent fine mesh oscilations.

Control rod position on the CORlib file is given in cm, rounded to integer. For most practical purposes this is sufficient, but when differential worths are calculated, this rounding may cause unphysical oscillations. Assuming that the K-eff values on the CORlib file are given on an equidistant mesh, the mesh read by CORCRW can be corrected accordingly by responding with "Y" to the prompt. If a mesh correction exceeds 2% of the total core height, a warning is issued.

```
File units:  1 - GNOMER output CORlib file (requested from input,
             2 - Integral worth output file (in PLOTTAB format),
             4 - Differential worth output file (PLOTTAB format),
             5 - keyboard input,
             6 - terminal screen,
             8 - tabulated text output file.
```

4.6 Manual for Program CORDFP

This is a trivial CORlib comparison program which only treats a single power distribution data set in two separate files and calculates the differences. Depending on the selected input option the per-cent relative differences or absolute differences scaled by one hundred (i.e.: per-cent difference with respect to the core average power) are printed.

Instructions:
The reference, the compared and the output filenames are requested from input. The flag IAB is requested to print the per-cent relative differences (IAB=0) or the scaled absolute differences (IAB=1). The first data type on the reference file under the keyword '$* PWRXY' is sought and compared to the first comparable data type on the compared file. The differences are written to the output file in CORlib format under the keyword '$* PERXY'.

4.7 Manual for Program CORDSP

The program displays and compares data contained in CORlib files. Full core or quarter core options are available. The display screen width is variable. A selection of data for display is made. The data are presented on screen and printed on the output file if desired. In the data comparison mode, the selected data types from the first CORlib file are compared to equivalent data types on the second file.

Instructions:
The program is designed to run interactively. Input requests are issued on the terminal screen and parameters are read from the keyboard. The filename of the first CORlib file is entered. The file is scanned and a list of all data types is prepared. The data are copied onto a scratch file SCR1.COR, performing data condensation or expansion, depending on the requested display format option (see comments below). Averaging of the axial distribution data is also performed, if requested.
NOTE: At present, a simplified model for radial avrageing is used, which does not check for cases where channel-normalized

axial distribution data representation is given (such as usually used for burnup data). In such cases the calculated averages are incorrect and NO warning message is printed.

The display window width is defined from input. It applies to the screen display as well as to the output file width. It is rounded to multiples of 8 columns. The display window width may be redefined by selecting an appropriate option on input.

```
WARNING: on VAX the display window width is limited to 132 columns.
         Requesting a larger window width causes program failure.
```

From the menu, actions can be selected which include the following:

```
    List CORlib file contents ...........  1
    Mark CORlib file data for display ....  2
    Display selected CORlib file data ....  3
    Comapre selected CORlib file data ....  4
    Write selected data to new CORlib file  5
    Print a page to Log-file ............  6
    Re-display last page ................  7
    Return to file selection options .....  8
    Return to display selection options...  9
```

The 'list' option (1) identifies different data types on the file which may depend on burnup, power and control rod position. For data retrieval only the dependence on burnup is considered.

Data display is designed primarily for assembly-averaged quantities. The core-average axial distributions can also be displayed, but the format is less convenient for displaying the full core axial distributions. Special options for this purpose will be implemented at a later stage. Note that a selection of data for display (2) must be made before the data display option (3) is actually selected. Although any number of entries can be displayed on the same figure, the output is clearer if not more than four data types are selected.

In the data comparison option (4), additional CORlib filenames are requested. The files are scanned and copied onto the scratch files SCRi.COR, performing data condensation or expansion as necessary. Selected data sets are compared one by one. The reference data from the first file and the absolute differences from the compared files are shown on the same diagram. After the display, the program returns to the options menu so that the data can be printed, for example. By selecting the data comparison option again, the next data type from the selected list is compared, until all data are processed.
is requested. The file is scanned and copied onto the second scratch file SCR2.COR, performing data condensation or expansion as necessary. Selected data sets are compared one by one. The reference data from the first file, the compared data from the second file and the absolute differences are shown on the same diagram. After display, the program returns to the options menu so that the data can be printed, for example. By selecting the data comparison option again, the next data type from the selected list is compared, until all data are processed.

The selected data can be written onto a new CORlib file. (option 5). The filename is entered on the first write request.
NOTE: if the selected data are to be written to the Log-file as well as to a CORlib file, make sure that the Log-file is written first, otherwise the output page may get corrupted, since the work array is shared my different modules of the code.

The displayed data can be written to an output file (option 6). The output filename is entered on the first printout request.

The reference file from which the data are displayed may be changed (option 8). This option is useful also if the display configuration is to be changed (for example, from full core to quarter core display). The program tests if the entered filename string is the same, in which case no action is taken. Use different case letters or place a blank before the filename when the same file is to be processed for a different display configuration.

Displayed format (option 9) is either full-core or quarter-core (including the octant-symmetry option), and the display width can be entered from input. Averaging to quarter core can be done assuming either standard or cyclic quadrant symmetry. Valid core symmetry options ICS are listed below:

```
  ICS = 1 - full core display, expansion assuming standard
            quadrant symmetry is performed if necessary.
       -1 - same as above but cyclic quadrant symmetry is assumed
            for expansion where necessary.
        4 - bottom-right core quadrant is displayed. Condensation
            is performed assuming standard quadrant symmetry.
       -4 - same as above but cyclic quadrant symmetry is assumed
            where necessary.
        8 - bottom-right core quadrant is displayed. Condensation
            is performed assuming octant symmetry.
NOTE: Expansion to full core representation can not be performed
      unless the first data type on the file is given in full core
      geometry. If the data on the files are all in quarter core
      configuration and no expansion to full core is required,
      enter ICS=4 to avoid program failure.
```

Files, logical units and default filenames:

Unit	Dflt.filenm.	Description
-		
1	K.COR	Reference CORlib file for display
2	K.COR	Second CORlib file for comparison
21	SCR1.COR	Scratch-1 (fixed filename)
20+i	SCRi.COR	Scratch-i (fixed filename)
4	SCR2.COR	Scratch-2 (fixed filename)
5		Keyboard input
6		Terminal screen
8	CORDSP.LOG	Output file
9	CORDSP.COR	New CORlib file

4.8 Manual for Program COREDT

The program is designed for general editing and merging of the CORlib files which are produced in the process of design calculations, namely the design file containing the core radial and axial distributions, pin burnups, cross sections, etc. The main CORXSA cross section generation input file is also updated.

The required input parameters are read from the main CORXSA input, in which case the updated Design CORlib filename is the same as the source filename. If the CORXSA input filename is blank, the edit input instructions are expected from keyboard. In this case, a different filename for the edited Design CORlib may be specified. The main difference between the manual input and the automatic sequence is that in the manual input case only the specified design file is updated. When the input instructions are obtained from the CORXSA input file, all processable files are updated, namely the main design CORlib file, the pin burnup file, the cross sections archival file, the current cross sections set and the main CORXSA input file itself.

Data types in the Design CORlib files are updated by the data contained in the Edit CORlib file. The relative power level, xenon level, control rod position and the boron concentration on the data header card are defined from the data set giving the average assembly power distribution. If the cycle burnup matches the EOC burnup on the data set giving the assembly average burnup distribution, all data sets headers are corrected to this EOC burnup.

When burnup steps are specified, the assembly (or pin) burnup distributions are updated. The cross sections files are modified accordingly (if present on the files CORE.XSR, CORE.XSP and if the cross sections archival CORlib file is defined with the '$* XSRLIB ' input command).

The CORXSA input file is also updated. If the boron concentration is found to be non-zero, the '$* BORON ' command is modified. If burnup is greater than zero, the '$* XENON ' and '$* SAMARIU' commands are added (if not already present) to request equilibrium Xenon and Samarium concentrations, respectively. The iteration step on the '$* POWERIT' command is incremented. If the iteration step equals the maximum number of iterations, the burnup entries in the CORlib files are incremented to the next value on the list of the '$* BURNUP' input command. Setting MXIT to zero is recommended only when the design CORlib file already exists and is not to be modified (for example, to generate cross sections for different core symmetry directly from the CORXSA sequence without iterations).

If the Edit CORlib filename is specified, it is assumed to contain the data to be added or replaced. A blank may be entered if a burnup increment is requested by interpolating the data already in the source Design CORlib file.

The requested cycle burnup step is the burnup to which the data in the Design CORlib file are to be modified. It serves as the flag to activate the assembly burnup increment calculation. At least the power distribution data must be available at suitable burnup steps so that the average power distribution at half the last burnup step can be retrieved by interpolation. Note that only the last burnup step can normally be corrected by specifying a reference burnup on the source Design CORlib file and the requested burnup, equal to the last burnup step on the file. Otherwise a warning is printed and the user is requested to confirm the continuation of the edit, in which case the new requested burnup step is added at the end.

The reference burnup on the source Design CORlib file can be redefined explicitly (it is needed if a burnup increment calculation is performed), otherwise the burnup from the last burnup step in the library is assumed.

The burnup on the Edit Corlib file can also be defined from input, in case it is not defined on the Edit CORlib file itself.

The symmetry flag is defined according to the usual conventions:

```
             4=reflective quadrant,
            -4=cyclic quadrant,
             8=octant symmetry.
```

It is used only if expansion of the data to full core are necessary.

To avoid handling excessively large files, the pinwise distributions are not expected to appear on the same file as the core distribution data. This also resolves the ambiguity in the format where the assembly-core distributions and pin-wise core distributions appear under the same CORlib header keyword.

When pin-wise distributions are updated, the source design CORlib file must contain at least the pin burnup distribution for the previous burnup step. For example, if there are NFD assemblies across the core axis and NFP fuel pins along the assembly sidelength, the pin-wise distributions are square arrays of order NFD*NFP. Note that the arrays are rather large, therefore it is recommended that no axial distributions are present on the file, since they would occupy an excessive amound of memory (and disk space). This is consistent with the assumption that the power distribution within an assembly is separable and distributions are given as axial averages. This is not strictly true, especially when axial blankets are present, but it is not expected to cause a significant error. Consistent with the above assumptions it is not necessary to specify the pin mass distributions. The pin burnup increment is calculated from the core average burnup increment and the relative pin power. This may lead to a slight inconsistency if one attempts to calculate assembly average burnup from the pin burnups. As a rule, pin burnup distributions are given in order to define relative pin burnup distributions within an assembly. Average assembly burnups and axial distributions should be obtained from the Design CORlib file.

To handle the axially averaged pin power distributions (Fdh by definition), the '$* PWRXY' and '$* FDHXY' keywords are accepted.

The cross sections archival library is updated automatically if the library filename is specified on the '$* XSRLIB ' input command on the main CORXSA input file. The data on the Edit CORlib file in CORlib format are entered. The default filename is XSBLIB.COR, but may be redefined with the '$* XSBLIB ' command on the CORXSA input file. The appropriate file can be generated from the usual cross section files with the XSTOCO utility code before running COREDT.

Input instructions:
All the required parameters can be extracted grom the general CORXSA input file. The following keywords are processed from by the CXSINP routine:

```
'$* CORDSN ' - define the source design CORlib file which is
               to be updated.
'$* CORPWR ' - define the Edit CORlib file.
'$* PINBRN ' - define the source pinwise distributions
               CORlib file.
'$* PINPWR ' - define the pinwise distributions Edit CORlib file
               (default: CORFDH.COR).
'$* XSRLIB ' - define the source cross sections archival
               CORlib file.
'$* XSBLIB ' - define the cross sections Edit CORlib file
               (default: XSBLIB.COR).
'$* FMFLIB ' - define the source pin power form factors CORlib
               file to be updated with the data on FASFMF.COR
'$* DIMLIB ' - define the core dimensions DIMlib file.
'$* POWERIT' - define the cross sections iteration paramteres:
                Col. Name  Description.
               11-20 MXIT  Max.No. of iterations.
               21-30 ITER  Current iteration index.
               31-40 ITXS  Number of iterations with explicit
                           cross section preparation with the
                           CORXSA procedure (default=1).
               41-50 CBTOL Tolerance on the boron concentration
                           change from the previous CORXSA
                           iteration to force termination and
                           proceed to the next calculational step.
'$* BURNUP ' - define the requested burnup steps [MWd/tU].
               Burnup steps may be define in incremental form. The
               format for such entries is " Xnnbbbbbb"
               nn is the number of times an increment is added the
                  previous burnup step.
               X  is the increment qualifier which can either be
                  "A" or "P".
               bbbbbb defines the increment. If X="A" than the
                  increment has a constant value "bbbbbb". In the
                  case of the "P" qualifier, the increment increase
                  on every step (i.e. the first increment is "bbbbb
                 the second 2x"bbbbbb", the third 3x"bbbbbb", etc.
If the CORXSA input file is not specified or if "-" is entered,
the required input parameters must be specified explicitly as
follows:
 - Filename of the source CORlib file to be updated,
 - Filename of the updated (output) CORlib file,
 - Filename of the Edit CORlib file,
 - Filename of the DIMlib library file,
 - Symmetry flag for the Edit CORlib file in case expansion to
   full core is necessary. Valid entries are:
     1 - no symmetry,
     4 - reflective quadrant symmetry,
    -4 - cyclic quadrant symmetry,
     8 - octant symmetry.
 - Cycle burnup step to which the burnup increment from the
   previous step is to be calculated. This entry is the flag
   for a burnup increment calculation.
  Negative value for the requested cycle burnup is a flag
  that the absolute value is the last burnup step and forces
  cycle burnup to be overwritten with the entered value.
 - burnup on the EDIT CORlib file [MWd/tU].
 - Reference burnup on the source CORlib file [MWd/tU].
 - Axial mesh flag Mz.
     blank  - default 10 is assumed.
     Mz<0   - the default mesh is subdivided -Mz subintervals.
     Mz>0   - relative mesh thicknesses are read (free format)
              in bottom-to-top order (arbitrary units).
```

4.9 Manual for Program CORFMT

```
Reformat a set of data into an 8-column format, 10 words per line.
Data are assumed to be delimited with blanks. Up to 20 words can
be specified on an input line. If the number of elements per row
is specified, data entries in a row are centered. Entering blank
or zero suppresses blank-filling and shifting of the entries.
The resulting file conforms to CORlib library requirements.
  Exceptions to the above rules are comments and keyword lines.
A record beginning with 8 or more blanks is treated as a comment
and is copied directly to output. If a record begins with "$", the
blank in column-3 is not treated as a delimiter. Also, regardless
of the words encountered on input, ten 8-column words are copied
to output.

Files:  unit  2 - input file (Filename requested from input)
              4 - re-formatted file (Filename requested from input)
              5 - keyboard input (interactive)
              6 - terminal screen output (interactive)
```

4.10 Manual for Program CORSMR

```
The design CORlib file is scanned. Summary information that is
useful for evaluating a core loading pattern is extracted and
printed to an output file. Parameters are sorted by region
where applicable, in addition to the cumulative total. Region
equivalences can be defined to reduce or group together a
number of regions.
```

Instructions:
The source design CORlib and the output filenames are requested from input.

Region labels can be equivalenced or redefined. The new region label is requested from input. Then follow the requests for region names which are to be equivalenced to the new name, until a blank is encountered. The region equivalencing procedure can be repeated if desired. Summary results are printed on the output log file.

4.11 Manual for Program CORPPK

From the fine mesh power distribution over homogenized nodes and inter-assembly form factors, given for each node in sequence, the heterogeneous global power distribution is reconstructed.

Maximum peaking factors and integral peak powers are picked and printed. The power form factors, which are superimposed onto the global fine mesh solution, are read from a separate file. They correspond to the heterogeneous solution over each node with fully reflective boundary conditions. A better approximation is also considered in which more realistic boundary conditions are used (such as the critical albedo condition). In this case an equivalen solution over homogenized nodes, with identical boundary conditions is read from a third file. The power form factors are defined as the ratio of the heterogeneous to the homogeneous node solution.

All power distributions are assumed to be given pin-by-pin for each node (i.e. fuel assembly) separately in CORlib format and in the usual node numbering sequence. Octant, quadrant or full core in 2D or 3D representations are allowed. Node location in the core is extracted from the global fine mesh solution usually generated by GNOMER (default filename CORHET.COR). The node location is read from the 6-th word (columns 41-48) of the CORlib header record. In 2D cases, columns 41-44 contain the string " Loc", followed by the core y- and x-coordinate indices (two characters each). In core quadrant and octant symmetries node numbering starts from the core centre. In full core geometry node numbering includes the reflector node (i.e. the lowest node index in the x- and y-direction is 2) In 3D cases columns 42-43 contain the z-coordinate. The coordinate indices usually correspond to the fuel assemblies. Note that the solution over homogenized nodes may be given on a uniformly denser mesh that the power form factors (i.e. an integer mesh densification factor in the active core) in which case fine mesh axial slices are averaged appropriately. The axial mesh of the reconstructed heterogeneous power corresponds to the mesh at which the power form factors are specified.

Burnup has the tendency to smooth out the peaking factors due to heterogeneities. If the heterogeneous assembly calculation is done assuming some average pin burnup BRN, an approximate correction can be applied by modifying the form factor FF according to the following empirical relation [1]:

```
  FF = 1 + (FF-1) * exp(-BRN/56686 )
```

Note that this correction should not be applied if burnup-dependent pin cross sections are used. If the solution over the homogenized nodes is given on a denser mesh, the burnup data are also expected on a denser mesh and averaged accordingly.

Instructions:
The filenames for the global fine mesh power distribution is
Input parameters can be read from the master CORXSA input

file. The filename is defined interactively on the first record of the default input file (usually the keyboard) in response to the prompt on the screen. Alternatively, input parameters may be defined interactively in response to the prompts

The filename for the global fine mesh power distribution is requested from input, followed by the axial mesh densification factor, if the homogenized solution is given on a denser mesh than the power form factors. Then the heterogeneous node power distribution and the homogeneous node power distribution filenames are requested. Defaults are suggested, but may be overridden by entering a different filename. If "-" is entered for the heterogeneous assembly solution, only the global solution over homogenized nodes is reconstructed. If "-" is entered for the homogenized assembly solution, the heterogeneous assembly solution is used directly as the form factors (identified by the keywords "$* FMFXY" of "$* PWRXY"), otherwise the form factors are calculated as the ratio of the heterogeneous to homogeneous assembly solution and printed to the "CORFMF.COR" file.

The output filename is defined next, to contain the axially averaged fine mesh pin power distributions in CORlib format (i.e. the enthalpy rise Fdh). A default name is suggested. If "-" is entered, the reconstruction of the global pin-wise power distributions is skipped.

To activate the burnup correction, the filename must be defined which contains the burnup for each axial region of each assembly on a separate record in columns 1-10. The core conditions file CORE.XSP for the region macroscopic cross sections for GNOMER can be specified. The number of the axial slices for an assembly must match the axial mesh of the global homogenized assembly solution.

To be able to reserve sufficient space for the storage of all relevant data, the maximum dimension of the core must be known. This can be either entered numerically as an integer. Otherwise the entry is interpreted as a filename and the number of assemblies across the core diameter is extracted from the record with the "$* CONFD" keyword in the DIMlib file or from a CORlib file containing the loading pattern data identified with the '$* FASXY' keyword.

Core symmetries can be considered. The following represent valid input entries for the symmetry flag:

```
   0 No symmetry
   1 Quarter core symmetry, no expansion to full core requested.
   4 Expand to reflective quadrant symmetry (TO BE CHECKED!).
  -4 Expand to cyclic quadrant symmetry.
   8 Expand to octant symmetry.
```

The log file CORPPK.LOG is opened automatically and contains the interpreted information on the processed output. Note that the axial region coordinate for Fqt (ie. the 3D peaking factor) excludes the reflector regions, although the location index in the assembly-wise files are usually specified including the reflector

Two auxilliary files CORHET.COR and FASFMF.COR are also opened to contain the reconstructed heterogeneous global solution in CORlib format and the form factors, given node-by-node.

By default, standard axial mesh of 10 axial slices is assumed. The mesh may be redefined when reading input from the master CORXSA input file.

References:

[1] M.Kromar: Empirical relation for the burnup correction of the form factors, Institute Jozef Stefan, Ljubljana, Slovenia, Private communication, (1995).

4.12 Manual for Program COTOGR

The 2-D core-wise pin power distribution file from CORD-2 is read. An (ASCII).GRD file such as recognized by the SURFER for WINDOWS is prepared. It contains the following records:

- header string "DSAA"
- number of nodes in the x and y direction (NMX, NMY)
- initial and final values on the x-coordinate (X0,XM)
- initial and final values on the y-coordinate (Y0,YM)
- minimum and maximum function value (P0,PMX)
- NMY logical records containing NMX function values, specified on equidistant intervals in the x-direction. Each logical record contains up to ten eight-character entries per line, terminated by a blank line.

Instructions:
The source CORlib filename is defined from input. A default filename is suggested, but it may be overridden by entering a new name. The mesh densification factor is defined next. It represents the number of points per cell in each direction. All points in a cell are assigned the same function value, to approximate the histogram representation on a plot. Two points per cell per direction were found to give good results for quarter core configurations. Finally, the maximum function is defined. This is necessary if several diagrams are prepared, so that they are all drawn to the same scale. If zero is entered, the maximum function value encountered in the file is adopted.

NOTE: The x- and y-axes are oriented contrary to the usual layout in CORD-2 files.

4.13 Manual for Program CORLOA

Given the core loading scheme(in CORlib format), extract data from the fuel assembly FASlib library and construct a CORlib core design input data file.

Procedures:
Given the loading pattern, all available information from the FASlib and DIMlib files is assembled and included in the Design CORlib file (for example: burnup distribution, control rod location, pin burnup, etc.).

Average operating conditions during burnup are determined approximately. From the actual assembly burnup and from the respective average cycle burnups, the average power during burnup can be determined. Using this average power distribution, the average moderator density and fuel temperature distributions are calculated.
NOTE: Cycle dependent data from previous burnup cycles are not considered when calculating average operating conditions during burnup.

Instructions:
The program is designed to run interactively. Data are entered from keyboard in response to the prompts on the screen:

- Enter the generic CORXSA input or the Load CORlib filename containing fuel assembly names (full core representation) and and burnable poison rod. If the generic input option is used the required input filenames and other data are extracted from the '$* CORLOA ', '$* CORDSN ', '$* PINPWR ' '$* RCFLIB ' and '$* CORSECT' keywords, respectively.

```
   The '$* AXMESH' keyword is checked for the presence
   of non-standard axial mesh discretisation.
   No other input is required. The generic input can also
   be piped to the default input unit. In this case the
   first record must begin with '$* ', but it MUST NOT be
   one of the above keywords (it may be '$* COMMENT', for
   example).
 - Enter the output Design CORlib filename.
 - Enter the output Pin Burnup CORlib filename. Entering blank
   causes pin burnup data to be ignored.
 - Enter the DIMlib filename.
 - Enter the core symmetry flag if expansion is necessary (-1 for
   cyclic quadrant symmetry)

File units:
  1 - DIMlib library (Default DIMLIB)
  2 - input Load CORlib file (default CORLOA)
  3 - output Pin Burnup file (default PINDSN)
  4 - output Design file (default CORDSN)
  5 - keyboard input
  6 - console for run-time messages
  8 - full log file displaying the results of the data checking tha
      can be done at this level (file CORLOA.LOG opened explicitly)
```

4.14 Manual for Program CORRCF

```
The CORlib file containing K-eff values at various core conditions
is assumed available. The excess reactivities are calculated with
reference to the critical state or another core condition, selected
from input. All the data in the CORlib file are processed.

Input instructions:
  The CORlib filename and the output filename are requested from
input. Next, the option selection menu appears. The boron
integral worth, and control rod worth with boron and isothermal
temperature coefficient at critical boron concentration are
currently available. The output filename may also be redefined.
The list of available parameters for reference is presented (it
may be the reference temperature or the reference control rod
configuration).
  The multiplication factor values are read from the CORlib file.
It may be the effective (K-EFF) or the infinite medium (K-INF)
value. Appropriate selection is requested.
  When boron integral worth is requested, the reference control
rod configuration can be selected from input and the total core
reactivity at the specified core configuration is printed.
Polynomial least squares fit may be performed, if desired. In
this case the boron concentrations list may be redefined in terms
of the starting value, final value and increment (note: the
increment in the last interval may be reduced to match the
specified final boron concentration). In addition, when the
starting boron concentration is zero, the printed reactivity
is shifted to zero at zero boron concentration. This option
is convenient, because it is consistent with the conventions
of the RCFlib reactivity coefficient library.
  When isothermal temperature coefficient requested, the reference
control rod configuration can be selected from input. If
blank is entered, tha tables for all control rods are generated.
The polynomial order can be specified for fitting the integral
worth as a function of temperature, so that the derivative can
be calculated (i.e. the reactivity coefficient). The isothermal
reactivity coefficient is tabulated at the temperatures and
```

boron concentrations found on the source CORlib file.

When control rod worth and other reactivity coefficients are requested for printout, the reference temperature may be selected from input. Reactivities are interpolated by fitting a second order polynomial for the boron worth and third order polynomial for the temperature worth (if sufficient points are available). For each control rod configuration at critical boron concentration the boron and the isothermal temperature coefficient are printed. Also, the control rod worth in [ppm] and [pcm] is printed with respect to the first CR configuration on the list (usually ARO) when control rods are inserted individually, or with respect to the previous configuration when control rods are inserted in sequence.

After printing the results the control returns to the options selection menu.

4.15 Manual for Program CORSHF

The program is designed to as a fuel management utility, to help the user to reshuffle the fuel assemblies consistently, while preparing a new core loading scheme. The old loading scheme from the source file in CORlib format is read, the assemblies are shuffled (or new assemblies loaded) and a new CORlib file with the modified loading scheme is written. This utility also allows to re-sort a labelled cross section file, corresponding to the new loading scheme.

Instructions:
The name is requested from input for the source data file containing an old loading scheme which is to be reshuffled. For the new loading scheme the same geometry is assumed.

The core symmetry for the original and the new loading scheme is requested. The following symmetry option flags are valid:

 1 - full core (no symmetry),
 4 - quadrant reflective symmetry,
-4 - quadrant cyclic symmetry,
 8 - octant symmetry.

Note that for assembly repositioning, higher symmetry can be defined for the new core compared to the original core. Also, conversion from reflective to cyclic quadrant symmetry (and vice versa) is permissible. For core shuffling the symmetry flag for the original core is ignored.

The new cycle length [MWd/tU] is also requested from input. The cycle number is incremented if assembly repositioning option is used (see description below).

To generate a new core loading pattern from an existing one, the following options are available:

1 - Reposition the assemblies from the original scheme to the new scheme. In this case the new scheme is blank-filled. A message is issued if any locations in the new scheme are left blank (compared to the original scheme) before termination. Note that it is customary to move the assemblies across the core centre to the new position. This is taken into account when assembly position coordinates are defined on the bottom-right quadrant (see note on coordinates).

2 - Swap the assemblies within a core quadrant. This option is is similar to the previous one except that the assemblies at the specified locations are interchanged.

3 - Symmetric locations in the core are loaded by assemblies specified from input.

4 - Write the auxilliary file "CORSHF.INP" on which the assemblies are listed which belong to symmetric locations

in the core. The core locations for the sequence are given left-to-right, top-to-bottom for the whole core, bottom-right quadrant or bottom-bottom-right octant, depending on the selected symmetry flag.

5 - Sort a labelled master cross section file, core conditions file and the form-factors file (if any) to correspond to the sequence on the input file "CORSHF.INP", such as prepared under option 4 above. The cross sections file, the core conditions file and the form factors file (if any) are processed.

6 - Sort a labelled master cross section file corresponding to the new loading scheme. Considering the specified symmetry flag, the source cross section file is scanned and a new file is written with cross sections sorted in order as specified in the new scheme. The filenames are requested from input.

7 - Display the original core loading scheme. The bottom-right (or bottom-bottom-right) core section is displayed, depending on the symmetry flag of the core.

8 - Display the new core loading scheme. The bottom-right (or bottom-bottom-right) core section is displayed, depending on the symmetry flag of the core.

9 - Display the assembly repositioning instructions. The bottom-right (or bottom-bottom-right) core section is displayed, depending on the symmetry flag of the core.

4.16 Manual for Program LIBXSA

All the necessary instructions and a selection of appropriate calculation options are contained in a single input file, which is common with the CORXSA procedure. LIBXSA provides a quick but less accurate means of cross sections preparation for HFP calculations. The Design CORlib file is read and this information is sufficient to generate the cross sections and the core conditions files. The core conditions at which the cross sections are defined are nominal. It is expected that the cross sections will be modified with the reactivity coefficient method (or otherwise) to correspond to the true operating conditions.

Instructions:
The CORXSA general input filename is entered from keyboard in response to the prompt. All input instructions in the CORXSA file are identified by keywords. Data on the input records are assumed in 10-column format, unless specified otherwise. A more exhaustive list of acceptable keywords is given in the CORXSA instructions.
The keywords recognized by LIBXSA are the following:
Currently accepted keywords are the following:

```
 '$* XSRFAS ' Define the Cross Sections Library filename, which
              is expected as the additional parameter (Col.:11-50).
 '$* COARSGR' This command is interpreted to determine the number
              of coarse groups from columns 11-80. By default NGC=2
 '$* CYCLE  ' Define the current cycle number for which the calcu-
              lations are performed. It is given in columns 11-20.
 '$* CORSECT' Requested core symmetry section may be defined. One
              additional parameter FCR in columns 11-20 implies:
                1 - full core without symmetry,
                4 - reflective core quadrant symmetry (default),
               -4 - cyclic   core quadrant symmetry,
                8 - core octant symmetry.
 '$* AXMESH ' Mesh definition in the axial region. One additional
              integer parameter LAX in columns 11-20 defines the
```

```
                number of axial regions. It can take the following
                values:
                 LAX = 20 - double axial mesh (see below LAX=10)
                            Note: Cross sections are calculated on
                            double mesh; compare to LA=10, LXF=2
                            where only the diffusion solution and
                            thermohydraulic calculations are done
                            on the double mesh).
                       10 - fine axial mesh (10 regions in the ratio
                            (6,6,12,24,24,24,24,12,6,6)
                        6 - not operational
                        3 - not tested
                        1 - a single (average) axial region.
                      n<0 - Axial slice thicknesses are defined
                            from input; |n| values are read from
                            one or more recortds that follow
                           (arbitrary units, normalised internally).
                            Note:
                            The actual axial slice thicknesses are
                            interpreted and written to the design
                            CORlib file by the CORLOA code. They
                            must not be redefined during the run.
                All other values are illegal (default LAX=1).
 '$* BURNUP ' Cycle burnup at which the relevant data are retrieved
                from the CORlib file is given in Col.11-20.
                If this command is not specified, cycle burnup
                as found on the '$* FASXY' data set in the CORlib
                file is assumed.
 '$* CORDSN ' Complete core design specifications filename (prepa-
                red by the CORLOA program or otherwise) is expected a
                the additional parameter (Col.:11-50).
 '$* END    ' CORXSA Input completed.

Note: All records which do not begin with any of the above keywords
      are ignored. Similarly, all records after the keyword
      '$* END    ' are also ignored.

WARNING:
  The generated cross sections do not have any P1 components
  due to burnup gradients. The request for such calculations
  with the '$* PINBRN ' command should appear on the input
  instructions set.

Files used:
   1 - CORXSA input file (name requested from keyboard input,
       default CORXSA.INP)
   2 - Core design specifications file (name in CORXSA input,
       default CORDSN)
   3 - cross sections file to be written (name in CORXSA input,
       default CORE.XSR)
   4 - cross section core conditions specifications file to be
       written (name in CORXSA input, default CORE.XSP)
   5 - Keyboard input
   6 - Terminal screen
   8 - LIBXSA.LOG diagnostics output file (filename fixed)
```

4.17 Manual for Program CORXSA

All the necessary instructions and a selection of appropriate
calculation options are contained in a single CORXSA input file.
The Design CORlib file is read and this information is sufficient

```
to generate the WICORD input file so that WIMS inputs for all
types of cells in the core can be calculated. Core conditions at
which the assembly cross sections are calculated is also
produced on a separate file.

Instructions:
The CORXSA input filename is entered from keyboard in response
to the prompt.
  All input instructions in the CORXSA input file are identified
by keywords (interpreted in the CORDIN routine). Data on the input
records are assumed in 10-column format, unless specified otherwise
Currently accepted keywords are the following:
 '$* POWERIT' Define the external iteration sequence of CORXSA
              procedure.
              NOTE: This command is not processed by CORXSA. It is
                    required by COREDT.
 '$* FEWGROU' Define transport group structure. With an additional
              parameter, one of the following may be selected:
               -1 - 18-groups,
               -2 - 32-groups, (default in WICORD and REFXSA),
               -3 - 69-groups,
               >0 - number of transport groups entered explicitly
                    with grouping given on the following record(s)
                    in free-format (i.e. for each transport group
                    the last WIMS group is specified, belonging to
                    that group).
              NOTE: This command is transferred by CORXSA to WICORD
                    input directly without checking.
 '$* MACROGR' Define explicitly the macro-group structure for fuel
              assembly homogenization. One additional parameter
              specifies the number of macrogroups NGM. On the
              next record, grouping is defined (i.e. for each
              macrogroup, the last transport group is specified,
              belonging to that group). This input option is
              mandatory if '$* FEWGROU' other than default is
              specified. By default NGM=10 and group boundaries
              are at transport groups (3 5 6 10 14 21 25 32).
                Normally the Effective Diffusion Homogenization(EDH
              method is applied on cell homogenized cross section.
              To suppress the EDH correction, enter negative NGM.
              NOTE: This command is transferred by CORXSA to WICORD
                    input directly without checking. It is used
                    exclusively to prepare input for XSWOUT.
 '$* COARSGR' Define explicitly the coarse group structure for
              global reactor core calculations. One additional
              parameter specifies the number of coarse groups NGC.
              By default NGC=2 and group boundaries are (6 10). On
              the next record, grouping is defined (i.e. for each
              coarse group, the last macrogroup is specified,
              belonging to that group). This input option is
              mandatory if '$* MACROGR' other than default is
              specified.
              NOTE: This command is transferred by CORXSA to WICORD
                    input directly without checking.
 '$* XSAOPT ' Assembly cross section homogenization options for
              for GNOMER input preparation. Parameters ICR,ICN,IHM
              are read from columns 18,19 and 20, respectively:
                ICR =0 - no criticality search,
                     1 - critical buckling search,
                     4 - critical albedo search (default),
                ICN =1 - diffusion constant condensation (default),
                     4 - transport cross sect. condensation (1/3D),
                IHM =1 - flux and volume weighted homogenization,
                     3 - effective diffusion homog. (default).
              NOTE: This command is transferred by CORXSA to WICORD
                    input directly without checking.
```

```
             NOTE: The above defaults are valid only when this
                   option card is absent altogether, otherwise
                   each parameter must be specified explicitly.
             See GNOMER input instructions on printout options
             '   CRITICL' and '   XSAVER' for more details.
'$* REFXSA ' Specifications for generating the reflector constants
             NOTE: This command is not processed by CORXSA. It
                   is required by the REFXSA code, where the input
                   instructions are defined in more detail.
'$* CYCLE  ' Current cycle number for which the calculations are
             performed is given in columns 11-20. The value is
             transferred to the WICORD input file without checking
'$* CORSECT' Requested core section may be defined. One additional
             parameter FCR in columns 11-20 implies
               1 - full core without symmetry,
               4 - standard core quadrant symmetry (default),
              -4 - cyclic   core quadrant symmetry,
               8 - core octant symmetry.
'$* AXMESH ' Mesh definition in the axial region. Two additional
             integer parameter LAX and LXF are read from columns
             11-20 and 21-30, respectively, with the following
             interpretation:
              LAX = 20 - double axial mesh (see below LAX=10)
                         Note: Cross sections are calculated on
                         double mesh; compare to LA=10, LXF=2
                         where only the diffusion solution and
                         thermohydraulic calculations are done
                         on the double mesh).
                    10 - fine axial mesh (10 regions in the ratio
                         (6,6,12,24,24,24,24,12,6,6)
                     6 - not operational
                     3 - not tested
                     1 - a single (average) axial region.
                   n<0 - Axial slice thicknesses are defined
                         from input; |n| values are read from
                         one or more recortds that follow
                        (arbitrary units, normalised internally).
                         Note:
                         The actual axial slice thicknesses are
                         interpreted and written to the design
                         CORlib file by the CORLOA code. They
                         must not be redefined during the run.
              LXF >  1 - In the diffusion calculation the axial
                         mesh density is increased by subdividing
                         each axial region into LXF subregions.
             NOTES:- All LAX values other than those listed above
                     are illegal.
                   - LXF entry is valid only when LAX>1.
                   - Parameter LXF isnot used in CORXSA.
'$* FASGAP ' Request to treat the inter-assembly gap explicitly.
             The gap half-thickness is calculated from the
             difference in the assembly and the pin pitch. The
             input command is transferred to the WICORD input and
             affects cell dimensions (taken in this case from the
             pin pitch rather than the assembly pitch) and
             subsequently in FASXSA when preparing the GNOMER
             input for assembly calculations.
'$* FASBUGR' Fuel assembly burnup gradients are considered
             explicitly (provided that relevant data exist in the
             fuel assembly library),
             NOTE: currently inactive. Use '$* PINBRN ' instead.
'$* PINBRN ' Pin burnups within an assembly are treated in detail
             explicitly (separability is assumed in the axial
             direction). Two additional parameters are expected:
              Col. Name Description
             11-50 FNBC filename containing the relative pin
```

```
                     burnups
           51-60 NBG  Number of burnup points to describe the
                      pin burnup (default 3, one point is
                      always the average, additional points
                      are uniformly spaced from min. to max.
                      pin burnup).
                      If NBG<0 the pin burnups are processed
                      but the burnup gradients in the cross
                      sections are not calculated.
'$* BURNUP ' Cycle burnup at which the relevant data are retrieved
             from the CORlib file is given in Col.11-20.
             If this command is not specified, cycle burnup
             as found on the '$* FASXY' data set in the CORlib
             file is assumed. Columns from 21-80 are used
             to define additional burnup steps. They are processed
             by the COREDT program, where additional instructions
             can be found. The burnup value is transferred to the
             WICORD input file without checking.
'$* CYCBRN ' Cycle burnup length [MWd/tU] is given in Col.11-20
             processed by the COREDT program.
'$* XENON  ' Xenon concentration scaling factor (default=1).
             Omitting the card, No-Xenon is assumed),
'$* SAMARIU' Samarium concentration scaling factor (default=1).
             Omitting the card, peak Sm scaling factor 1.3 is
             assumed,
'$* ZEROPWR' Redefine average moderator temperature and density on
             two additional parameters:
             Col.:11-20 - TMO average moderator temperature [K],
                  21-30 - GMO average moderator density [g/cm3],
                          if GMO>2, ignore,
             NOTE: Columns 21-80 can contain the temperature list,
                   interpreted by the WICEDT code, where further
                   details on input instructions can be found.
'$* POWERP ' Define the relative power:
             Col.:11-20 - PWP relative core power level [%]
             WARNING:
             It is generally assumed that the main design CORlib
             file is specified at "full power" only. The retrieval
             routines interpolate the distributions by burnup and
             do not check the power and xenon levels. It is
             permissible to append special calculation results at
             intermediate power levels (and arbitrarily chosen
             burnup) at the end of the design CORlib file, but
             not in the middle. When distributions are read from
             the design CORlib file to initiate the calculations
             at intermediate power levels, the input cycle burnup
             step must match the one in the library to the nearest
             10 MWd/tU (i.e. no interpolation by burnup is allowed
             in this case).
'$* BORON  ' Specify the boron concentration in Col.11-20 [ppm].
             The value is transferred to the WICORD input file
             without checking.
             NOTE: Columns 21-80 can contain the boron conc. list,
                   interpreted by the WICEDT code, where further
                   details on input instructions can be found.
'$* CRODS  ' Request to generate control rod cross section data;
             If only one specific set of control rods is required,
             its designation may be specified in columns 11-20
             (2-char. designation, '**' = '  ' ==> all rods).
'$* PINPWR ' Request to calculate the pin-power distributions. The
             additional parameter in columns 11-50 is the filename
             containing pin-wise core power distribution.
'$* WIMLIB ' WIMS library specifications. Two additional parameter
             are required:
             Col.:11-50 - WIMS library summary filename
                          (default: WIMLIB)
```

```
              Col.:51-60 - WIMS library identification name
                           (default: IJS0)
 '$* DIMLIB ' Core Dimensions Library filename is expected as the
              additional parameter (Col.:11-50)
 '$* WICINP ' WICORD input filename is expected as the additional
              parameter (Col.:11-50).
              NOTE: this command must precede any other commands
                    which are transferred directly to output.
 '$* CORDSN ' Complete core design specifications filename (prepare
              by the CORLOA program or otherwise) is expected as th
              additional parameter (Col.:11-50)
 '$* ISOLIB ' Isotopic Inventory Library filename is expected as th
              additional parameter (Col.:11-50)
 '$* CORXSR ' Cross sections filename for the UNRODDED core is
              the additional parameter (Col.:11-50). Note that
              "CORE.XSR" is always the standard cross section
              file generated in the CORXSA or LIBXSA sequence.
              The filename redefinition takes effect only in
              the cross sections editing sequence with XSREDT.
 '$* CORXSP ' Cross section core condition specifications
              filename for the UNRODDED core is the additional
              parameter (Col.:11-50). Note that "CORE.XSP" is
              always the the standard cross section file
              generated in the CORXSA or LIBXSA sequence.
              The filename redefinition takes effect only in
              the cross sections editing sequence with XSREDT.
 '$* CROXSR ' Cross sections filename for the RODDED core is
              the additional parameter (Col.:11-50). Note that
              "CROD.XSR" is always the standard cross section
              file generated in the CORXSA sequence.
              The filename redefinition takes effect only in
              the cross sections editing sequence with XSREDT.
 '$* CROXSP ' Cross section core condition specifications
              filename for the RODDED core is the additional
              parameter (Col.:11-50). Note that "CROD.XSP" is
              always the the standard cross section file
              generated in the CORXSA sequence.
              The filename redefinition takes effect only in
              the cross sections editing sequence with XSREDT.
 '$* XSRLIB ' Burnup-dependent cross sections library for a
              particular cycle in CORlib format.
              (Not used in CORXSA)
 '$* XSBLIB ' Cross sections file for a particular burnup step
              in CORlib format.
              (Not used in CORXSA)
 '$* XSRFAS ' Parametrised fuel assembly cross sections library
              in CORlib format. If present on input, CORXSA
              execution is terminated and the "LIBXSA" keyword
              is written to the CORXSA.TMP file, indicating
              that assembly cross sections are to be generated
              from the library.
 '$* WIMSD5 ' Define LATTICE instead of DANCOFF in WIMS-D input.
              The option is applicable to in WIMSD5 code only.
              (Not used in CORXSA, transferred to WICORD input).
 '$* END    ' CORXSA Input completed.

Note: All records which do not begin with any of the above keywords
      are ignored. Similarly, all records after the keyword
      '$* END    ' are also ignored.

Files used:
   1 - CORXSA input file (name requested from keyboard input,
       default CORXSA.INP)
   2 - Core design specifications file (name in CORXSA input,
       default CORDSN)
   3 - cross section core conditions specifications file to be
```

written (name in CORXSA input, default CORE.XSP)
4 - WICORD input file to be written (name in CORXSA input, default WICORD.INP)
5 - Keyboard input
6 - Terminal screen
8 - CORXSA.LOG diagnostics output file (filename fixed)

4.18 Manual for Program FASBUP

Given the core description data on a CORlib file, perform automatic update of the FASlib fuel assembly library files. If a FASlib file exists, check the data, otherwise create a new file. Note that the program can not create entries with varying enrichment and mass distribution in the axial direction (Vantage-5 fuel). For IFBA fuel, B-10 axial distribution is assumed to be 0.59 [mg/cm] in regions 3 to 8 and zero otherwise.

The data type "$* FASXY" must be present on the CORlib file since the FASlib filenames are extracted from it. In addition, the following data types are checked for consistency: "$* REGXY", "$* TYPXY", "$* MASXY", "$* ENRXY" and "$* IFBXY". To perform a burnup update, the following data types are requested: "$* PWRXY", "$* BPRXY", "$* BRNXY", "$* BRNAX", "$* TFBXY", "$* TFBAX", "$* GMBXY", "$* GMBAX". If non-standard axial mesh is defined, the axial slice thicknesses are read from the "$* REGAX" data type.

If the pin burnup file is specified (giving whole-core pin-by-pin burnup distribution), the data at the appropriate burnup are extracted and a scratch file is created giving pin burnup distributions for all assemblies in sequence.

Input instructions:
- Enter the initials of the operator (two characters),
- Enter the comment describing the Update,
- Enter the core description CORlib filename,
- Enter the pin burnup CORlib filename ["-" to ignore],
- Enter the core symmetry (-1 for cyclic quadrant symmetry)
- Enter the requested cycle burnup step Bu (MWd/tU)
  BU > 100 000 indicates EOC BU
  BU < 0 indicates no BU edit requested :
- If execution is performed interactively and a message indicating an unacceptable error is noted, processing can be stopped before the update is performed by responding with an "N" to the prompt.

4.19 Manual for Program FASEDT

The Fuel assembly files are an important link in the chain of the Core Design calculations. Any errors on these files which escape error trapping may cause errors in the calculations which could be very difficult to identify. It is therefore essential that editing of these files eliminates possibilities of trivial errors as much as possible, providing means of QA/QC at the same time.

The program is designed to run interactively.

Input Instructions:

Fuel assembly identification is requested from input. If the corresponding file exists, it is displayed. If not, a new file is open

The Keyword of the data type to be entered (or corrected) is requested from input. At present the following options are

```
available:
'$* IDENT': fuel assembly region assignment and fuel type are
            requested (Fuel type examples: WH, KW, V5),
'$* REGAX': This option is required if non-standard axial mesh
            is defined for the fuel assembly. The number of
            axial slices is requested from input, followed by
            the respective slice thicknesses (bottom-to-top)
            in arbitrary units. Normalisation to the active
            core height is done within CORD-2.
'$* MASSU': mass of heavy metal [kg] is requested (assumed homogene
            ously distributed. If zero is specified or if Vantage-5
            fuel (type V5) is defined, then the mass of the central
            (reg. 2 -- 9) and peripheral regions (1 and 10) are
            requested,
'$* ENRIC': enrichment [%] is requested (assumed homogeneously
            distributed. If zero is specified or if Vantage-4 fuel
            (type V5) is defined, then the enrichment of the
            central (reg. 2 -- 9) and peripheral regions (1 and 10)
            are requested,
'$* IFBA ': number of IFBA pins in the assembly is requested,
            followed by the boron-10 loading
            [default 0.59 [mg/cm] distributed homogeneously
            in regions 3 - 8 of the default slices only].
            IFBA loading pattern in a fuel assembly is uniquely
            determined by the number of B-10 loaded pins and the
            fuel type.
            Note: even if fuel design specifications remain
                  the same and only the IFBA loading pattern
                  within the assembly changes, a new fuel type
                  must be assigned).
  Before closing the file, the edit description and initials of the
person doing the edit are requested in order to record the editing
history of the file.

Files:
  1  - Scratch file SCR1. opened explicitly, discarded on exit,
  2  - Source .FAS file (if exists), filename defined internally,
  3  - Scratch file SCR2. opened explicitly, discarded on exit,
  4  - Edited .FAS file, filename defined internally,
  5  - Keyboard input,
  6  - Terminal screen output.
```

4.20 Manual for Program FASGET

```
Given the core loading pattern, copy the required files from the
FASlib fuel assembly library. The target is the default directory.

Instructions
------------
A record is read from the default input and is interpreted as
follows:
- If the record begins with "$* ", the default input is assumed
  to be the generic CORXSA input file on which the FASlib
  library and the loading pattern CORlib library names are
  defined.
- Otherwise the record is interpreted as a filename. The records
  on the file are scanned for keywords:
   * The file is recognised as the generic CORXSA input file if
     it contains the '$* CORLOA ' and '$* FASLIB ' keywords
     defining the CORlib loading pattern file and the FASlib
     library files, respectively.
```

* The file is recognised as the merged FASlib library if it contains the '$* IDENT' keyword. In this case the CORlib loading pattern filename is expected from the second entry on the default input.
* The file is recognised as the CORlib loading pattern file if it contains the '$* FASXY' keyword. In this case the merged FASlib library filename is expected from the second entry on the default input.

File units:
1 - Generic CORXSA input.
2 - CORlib file containing fuel assembly core loading pattern (full core) and region assignment.
3 - Merged FASlib library file.
4 - Output single assembly faslib file.
5 - Default input (keyboard) or generic CORXSA input.
6 - Console output for run-time messages.
8 - full log file displaying the results of the data checking that can be done at this level. (file FASGET.LOG is opened explicitly).

4.21 Manual for Program FASSMR

A FASLIB merged library file is processed to produce a summary over assemblies of a selected grouping property. The grouping property can be the assembly type, the region or the assembly name. The output list can contain the data for all different grouping properties or for a specific one, if this is specified. The user may select to list characteristics of all assemblies of the specified grouping property or only the average values.

Instructions

The filename of the FASlib library in merged form is requested from input. The user is then prompted to specify the grouping property by entering the appropriate number to select grouping by assembly type, region or assembly name. A specific name can be defined (for example, grouping by assembly type "V5" will process only assemblies of this type). If blank is entered, all assemblies are processed. In addition, selection by cycle number or by burnup can be made if the appropriate key is entered. In the selection by cycle number, the value "n" is requested. If "n" is positive, only the assemblies loaded in cycle "n" or later are printed. If "n" is negative, only assemblies loaded in cycle "-n" or earlier are printed. Similarly, in the selection by burnup, the burnup threshold "b" is requested. If "b" is positive, assemblies with burnup not exceeding "b" are printed. If "b" is negative, assemblies burnt more or equal to the threshold are printed. Finally, a selection is made to either list the characteristic data of the assemblies with the selected grouping property, or to average the data. In the "list" option, the assembly name, type, region, number of IFBA pins, mass, enrichment, last cycle position (cycle number and location) and burnup are specified. An auxilliary file with a fixed filename "CORLOA.COR" is also opened, which contains the listed assemblies in CORlib format. In the "average" option, the number of assemblies with the specified grouping property, the total mass, the average enrichment and the average burnup are given.

4.22 Manual for Program FASSPL

A file where all F.A. library files are merged together is split so that individual F.A. library files are created. File names consist of the F.A. identifier with extension .FAS

Instructions:
The merged FASlib filename is requested from input. The FAS files are written to the default directory.

4.23 Manual for Program FASXSA

Two input files for GNOMER diffusion code are prepared so that cross sections for all assemblies (rodded and unrodded on separate input files) can be generated automatically. Basic input data are read from the WICORD input file (usually prepared by the CORXSA program). Some additional data are extracted from the Core Dimensions library. Buckling and cell dimensions are read from the relevant WIMS input files, therefore WICORD must be executed before FASXSA.

WICORD input filename and the generated GNOMER input filenames are requested from keyboard interactively. All other files are assigned internally.

Specific input records which are recognised and processed by the FASXSA code are characterised by the following

```
 '$* MACROGR' number of macrogroups in the assembly solution
              (default 10, actual macrogroup structure is not
              checked by FASXSA).
 '$* COARSGR' number of coarse groups in the global core solution
              and the coarse grouping set, i.e. for each coarse
              group, the last macrogroup belonging to that coarse
              group (default 10, actual coarse group structure is
              not checked by FASXSA).
 '$* XSAOPT ' Cross section avrageing options ICR,ICN,IHM for GNOME
              are transferred in Col.18,19,20, respectively.
              Definitions are given in GNOMER input description:
               ICR     - criticality search '   CRITICL' command,
               ICN,IHM - homog.options of '   XSAVER ' command.
 '$* FASGAP ' flag to request explicit treatment of the inter-
              assembly gap (pin pitch and assembly pitch are
              retrieved from the DIMlib library). Note that the
              gap half-thickness smaller than 0.01 cm is ignored
              and a warning is printed.
 '$* PINPWR ' flag to request the calculation of the pin-power
              distributions. The heterogeneous assembly pin-power
              distribution is written on the FASHET.COR file. The
              corresponding homogenized assembly pin-power distri-
              bution is written on the FASHOM.COR file. A scratch
              file FASXSA.TMP is opened to contain temporarily
              the homogenized cross sections.
 '$* FASXSA ' basic keyword which indicates the beginning of the
              WICORD input sets for a particular fuel assembly.
              Any general input options described above must
              precede this command. Additional parameters have the
              following meaning:
              Col.11-20 - Number of cell types (minimum of 2 for
                          fuel and water cells, or higher if
                          special cells such as BPR's and IFBA
                          pins are present)
              Col.21--> - Keywords by which special cells are
                          recognised (for example '$* BPRnn' for
                          BPR's and '$* IFBnn' or '$* IFnnn' for
```

IFBA pins). The keywords are 8-characters long, placed right-justified into 10-character input words (i.e.: keywords if any begin in columns 23, 33, 43 etc.).

Files used:
1 - Core dimensions library (assigned from WICORD input file)
2 - FASXSA (=WICORD) input file, requested from keyboard
3 - WIMS input files (assigned from WICORD input)
4 - GNOMER input file which is to be written
5 - Keyboard input
6 - terminal screen
7 - coarse group cross sect.files (unit declared but not used)

4.24 Manual for Program FOTOPT

The FOLLOW output file named "FOLLOW.OUT" (fixed filename) is processed to extract the boron concentration as a function of burnup. The first point is assumed before full power conditions are established and is treated separately. The succeeding points within a burnup interval of BSTP [MWd/tU] are read. The points where measured boron concentration is zero or which are labelled as invalid are skipped. The points are averaged and their standard deviation is calculated. The average is recalculated, neglecting the points which deviate by more than one standard deviation. The output is written on the file "FOLLOW.PNT" (fixed filename). It begins with a header record followed by data in 11-column format, such as accepted by PLOTTAB.
Three additional files: F1.CUR, F2.CUR AND F3.CUR are opened to contain the relative power, temperature and control rod position at each burnup point, respectively. The burnup averaged values are written to "FOLLOW.LST".

4.25 Manual for Program INTOCO

The program converts INCORE long output to CORlib format of the CORD-2 system. In addition, the axial power distributions for each assembly are verified by comparison to the core average axial power distribution. If differences in excess of a specified tolerance limit are encountered, the axial power distribution for those assemblies are printed in PLOTTAB format.

The program is compatible with the output format of the following versions of the INCORE code:
INCORE 3D 3.8.4.C.
INCORE 3D 7.4.5
INCORE 2D 3.6

Procedures:

The program tries to determine automatically the number of points per trace, the number of assemblies and the power. The loading scheme is determined from the output page where the measured axial peaking factors are given. The assembly axial relative power distributions are read, assuming they are given on a uniform axial mesh. They are integrated to determine the assembly average power and the core average power (note that the assembly average power in INCORE output is not processed). Assembly axial power distributions are integrated to 10 CORlib axial regions (defined by the DATA statement NPC and WTC for the number of regions and relative thicknesses).

Input instructions:

The source INCORE output filename is requested from input.

The CORlib output filename is requested. If a blank is entered, CORlib file preparation is skipped. Additional parameters which could not be determined from the INCORE output are requested interactively. These include:

```
  - cycle number
  - cycle burnup step    [MWd/tU]
  - core power level     [%]
  - control rod position [cm]
  - boron concentration  [ppm]
```

If power level is greater than 15%, equil.xenon is assumed. Control rod position must be specified in the format "Xnnn" where "X" is the control rod bank identifier and nnn is the control rod position in [steps]. Conversion to [cm], measured from the top of the reflector is performed internally, assuming Abs(ST0) steps in the core (-ve sign implies -ST0 steps at ARO configuration), a rodded length of STL cm, total core length of COL cm and reflector thickness of RFL cm. The parameters are defined in a DATA statement in the code (the convention for defining control rod position in CORlib files has been adopted because the diffusion code, which normally generates the control rod position, could not easily convert to "steps" in a general way).

The number of axial mesh slices NPC is requested. If blank, the default 10 is assumed. If negative, the default mesh is subdivided into -NPC subintervals. If positive, the relative mesh thicknesses are read (free format) in botto-to-top order. Normalisation of the thicknesses is not necessary as it is done internally.

The threshold trace difference is requested (M. Kromar ???)

The PLOTTAB curves filename is requested. If a blank is entered, preparation of the PLOTTAB file is skipped.

The threshold power difference is requested from input. The axial power distribution in an assembly is renormalized so that its average power matches the core average power. The distribution is compared to the core average axial power distribution. If the absolute difference is greater than the threshold, the assembly power distribution is added to the printout list.

The number of curves per plot NPR is requested. This means that every NPR-th curve is the core average axial power distribution. This option is introduced for convenience so that in PLOTTAB the reference core average axial power distribution can be plotted on every picture. Repetitive printout of the core average axial power distribution can be suppressed by selecting NPR=1.

4.26 Manual for Program ISOPLT

ISOPLT The program ISOPLT writes the files isoplt.cur to plot the isotopic composition from different libraries. The output files are ready to be used as input files for PLOTTAB code system.

```
Input data
==========
```

The following data are requested from keyboard:

1.) Isotopic composition library file names (up to 10)

```
2.) Assembly type (WH, V5, ...)
3.) Fuel enrichment index.
4.) Moderator temperature index.
5.) Fuel temperature index
6.) Boron concentration [ppm]
7.) Number of inserts (NBPR)
8.) IBFA index


Mode of operation
=================
After reading input data, the vector of input conditions
is set up and a searching is performed for each library
considered in the running. If the record is found the
corresponding isotopic composition is read in to a core
array. Afterwards, taking the first input library as
reference, a search is now carried out by isotopes to
write the isoplt.cur file. If the same isotope is found
on the rest of the libraries a new curve is added to
isoplt.cur. Finally, plotting control information is
written on the isoplt.p92 file. The code is designed in
such a way that all isotopes of the reference (first)
library are always plotted. Correct global plotting
information is written accordingly.

Input units
===========
20+I, where I is the input library index
   5, keyboard
   6, output console

Output units
============
   2, isoplt.cur
   3, isoplt.p92
```

4.27 Manual for Program ISOFMT

```
Read an Isotopic Inventory Library file in standard format with
Isotopic masses given of a fine burnup mesh and rewrite the library
with burnup steps on coarse mesh (<10 steps, internally defined).
Files:  1 - Source library file (Filename requested from input)
        2 - Target library file (Filename requested from input)
        5 - Keyboard input
        6 - Diagnostics output (screen)
        7 - ISOFMT.LOG warnings and error message output
```

4.28 Manual for Program ISOLIB

```
A WICORD Input file is generated in order to prepare several
WIMS input files so that Fuel pin isotopic inventory can be
calculated.

Instructions:
The filenames of the CORD-2 libraries are defined from input
(DIMlib, WIMS library identifier, WIMlib). Then follow the
definitions of the fuel assembly type (normally two characters),
cell type (2=standard fuel cell, 12=fuel cell with IFBA,
```

42=fuel cell loaded with Gd2O3 burnable poison), mass of initial heavy metal per assembly [kg], average boron concentration during burnup [ppm] and the number of assembly inserts (burnable poison rods are usually implied). If a Gd2O3 loaded cell is specified, the weight fraction of Gd2O3 is also requested. Optionally, the mass of U-metal (normalized as if all pins were made of the same material) are specified, when different from surrounding fuel pins. Similarly, an enrichment multiplying factor can be defined by which the enrichment of the central Gd2O3 loaded pin is multiplied, if necessary.

The enrichments, fuel temperatures and moderator densities are predefined internally. The available set of values is displayed. The selection is made by entering the initial and the final index of the displayed values. Note that the assembly linear power is correlated with the fuel and cladding temperatures. Similarly the moderator temperature is correlated with the moderator density. Otherwise the parameters are assumed independent.

The WICORD input file is written to WICORD.INP file so that WIMS input can be generated for a burnup over the assembly life. Note that assemblies (or their sections), which are operating at low power, will achieve a lower discharge burnup.

4.29 Manual for Program ISOVEC

Appropriate parameters (from the list of which the isotopic inventory library ISOlib is tabulated) is requested from input. The library is scanned and the isotopic vector printed to output.

4.30 Manual for Program ISOWIC

Input for WICORD is prepared. Relevant information is requested from input interactively. The fuel isotopic inventory vector is read from the ISOLIB library by specifying the appropriate burnup.

anual : A WICORD input file is prepared to generate the corresponding WIMSD inputs. The WIMSD runs consist in a k-infinity calculation vs burnup for one of the lattice models considered in the CORD-2 system. The material composition as a funcion of burnup is taken from the specified ISOlib library at the parameters entered as input. If desired, some isotopes are removed from the composition vector. The isotopes that should be removed are entered in an exclusion file. It could be useful to study the impact on reactivity of a given set of isotopes. The code requests input data from the standard input device and writes the corresponding WICORD input file.

Instructions:
The filenames of the CORD-2 libraries are defined from input (DIMlib, ISOlib, WIMlib). Then follow the definitions of the fuel assembly type (normally two characters), cell type (2=standard fuel cell, 12=fuel cell with IFBA), mass of initial heavy metal per assembly [kg], average boron concentration [ppm] and the number of assembly

inserts.
The enrichments, fuel temperatures and moderator densities are predefined internally. The available set of values is displayed. The selection is made by entering the initial and the final index of the displayed values. Note that the assembly linear power is correlated with the fuel and cladding temperatures. Similarly the moderator temperature is correlated with the moderator density. Otherwise the parameters are assumed independent.

```
      Input units
      ===========
         1, isotopic composition library
         3, Isotope Exclusion file (default: NONE = include all)
         5, keyboard
         6, output console

      Output units
      ============
         2, WICORD output file

 Format of the exclusion file
  ===


    LINE          DATA    FORMAT    Example
    ====          =====   ======   =========
 First line:    $* ISOMR   (A8)     $* ISOMR
                  .         .         8O   0
                  .         .        61Pm148
                  .         .       561Pm147
  i-th line:    IZZSYAAA   (A8)      92U 235
                  .         .        94Pu239
                  .         .        94Pu240
                  .         .        94Pu241
  Last line:    $* ENDMR   (A8)     $* ENDMR
```

where IZZSYAAA is the library identifier for a given isotope. IZZ is the Z number (with extensions), SY is the chemical symbol and AAA is the mass number. These identifiers can be extracted from the corresponding WIMlib file.
The data are read using the A8 FORTRAN format descriptor.

4.31 Manual for Program ISTOPT

Read an Isotopic Inventory Library file in standard format with Isotopic masses given of a fine burnup mesh and rewrite the file in PLOTTAB curves format.

```
Files:  1 - Isotopic inventory file (Filename requested from input)
        2 - PLOTTAB curve file (Filename requested from input)
        5 - Keyboard input
        6 - Diagnostics output (screen)
```

4.32 Manual for Program KITOCO

A file containing the K-inf values for each fuel assembly is read. The format corresponds to that written by the XSTOKI code. The

K-inf values on the file are assigned to assemblies in sequence as found on the CORlib file, defined from input. The rearranged K-inf values are printed in CORlib format.

Input instructions:
The names for the CORlib file containing the assembly loading pattern, the K-inf data and the output CORlib file are requested from input. Default filenames are suggested, but may be overridden by defining new filenames.

4.33 Manual for Program KINDIF

Filenames are requested for the reference and the compared file containing K-inf values (such as produced by program XSTOKI). The output report filename is then requested. The comparison may be split into separate parts, taking NST values at a time (i.e. the grouping set length). Critical buckling or the K-inf values may be presented. On the report file the differences are presented numerically and graphically. For example, the grouping option is useful to compare differences along a fuel assembly length (otherwise the displayed differences are swamped by large differences between different fuel assemblies

4.34 Manual for Program LEASQP

====

A set of points is fitted to a polynomial of specified order. Define the source data and output filenames interactively from terminal screen. Define also the fitted polynomial order.

Frequently, the data to be fitted lie over a relatively narrow range either in the argument, the function or both. This causes a numerically ill-posed problem. The pivot coordinates (XP,YP) may be specified, to which the origin of a new coordinate system is shifted by linear transformation. Polynomial coefficients are calculated in the new coordinate system.

Optionally, an output file is generated, tabulating the fitted polynomial (or its n-th derivative) over a specified number of equidistant mesh points. The output points are defined in the original (input) coordinate system.

The source data file is read, assuming two values per record (argument and function). If the first record can not be read successfully, it is interpreted as comment and transferred directly to the output file. A blank card is interpreted as an end of a data set. The next data set may follow (starting with a comment or with numeric values)

The input and output format is consistent with the one required for PLOTTAB files.

4.35 Manual for Program PTTOIS

This is an ad-hoc program for converting curve data (.CRV) prepared for drawing by PLOTTAB into Isotopic Inventory Standard data format (.ISO). Isotope identification is extracted from the PLOTTAB comment card assuming "zz-SS-aaa" representation where zz is the atomic number, aa the mass number and SS the chemical symbol

Data are read in sequence (several isotopes are assumed present) until an End-of-file is encountered.

4.36 Manual for Program QDRTLT

The purpose of this program is to provide a correction for tilts in the measured data in a reactor core. Symmetry considerations across core quadrants and octants are considered.

Primary quadrants (NW, NE, SW, SE) are defined by dividing the core along the Cartesian coordinate axes with origin at the core centre. Secondary quadrants (N, S, W, E) are defined by dividing the core across the diagonals (which also define the core octants).

Define power tilt (T) as the ratio between the quadrant average power (Pq) and the core average power (Pa). Reactor core power distribution is measured over the whole core while calculations are normally done for a core quadrant or octant. In the case of octant-symmetric core loading, the power tilts over the core octants should also be 1 (neglecting the small differences due to engineering tolerances and burnup). This is equivalent to requesting that tilts across primary and secondary quadrants are all equal to 1. When core loading is quadrant-symmetric, the tilts across the primary quadrants should still be 1, but the restriction for tilts over the secondary quadrants is relaxed. They may differ from 1, but they should be equal in the N/S quadrants and in the W/E quadrants, respectively.

In real power distribution measurements, power tilts may arise due to other (thermohydraulic) core properties which cannot be controlled. For a consistent comparison between measurement and calculation, these tilts must be accounted for. Tilts across primary quadrants are eliminated by simple averaging over the quadrants. Tilts across the secondary quadrants are only averaged to some extent. Namely, the quadrant-averaged power distribution is sensitive to variations in the average tilts along the axes (i.e. the average tilt in the N/S quadrants compared to the average tilt in the W/E quadrants). These tilts must be treated explicitly.

Octant assembly power symmetry is incorporated provisionally. The printed errors correspond to Octant average power symmetry.

Procedures:
Assume some form for the tilt distribution function which can be:
Trigonometric (in polar coordinates):

```
 T = sin(r*Pi/(2*R))
   * ( C1          + C2*sin(t)   + C3*cos(t)   + C4*sin(2*t)
     + C5*cos(2*t) + C4*sin(3*t) + C5*cos(3*t)  )
```

Polynomial (in Cartesian coordinates):

```
 T =  C1     + C2*x   + C3*y     + C4*x*y
   +  C5*x*x + C6*y*y + C7*x*x*y + C8*y*y*x
```

The origin is at the core centre, dimensions are in units of assembly side.

The core power distribution is read. Given the quadrant-averaged power distributions from input, the smooth tilt distribution is calculated and input power distribution corrected accordingly to compensate for the assumed power tilts.

When octant symmetry is requested, the tilt distribution function is constructed such that tilts across all quadrants will be compensated completely. In the case of quadrant symmetry some residual tilts in the secondary quadrants will remain, equal to the average of the tilts in the N/S quadrants and the W/E quadrants, respectively.

Input instructions:

- Filenames containing the raw power distribution data and the tilt corrected distributions are requested interactively.
- Tilt distribution form function flag is requested. Numeric value of 1 corresponds to the trigonometric form, otherwise the polynomial form is used.
- When consistent power distribution data are given on input (i.e. with non-zero entries on all symmetric core locations) the quadrant tilts are calculated and printed on screen. They are used to calculate the tilt distribution function. If desired, each of them may be redefined from input by entering a new value in response to the prompt (e.g. entering 1's for tilts in secondary quadrants N, S, W, E will eliminate the tilt corrections in these quadrants).
- If full octant symmetry is desired, respond with 1 to the prompt.
- If condensation to a core quadrant is desired, respond with 1 to the prompt.

Note that the raw power distributions are given in the standard CORlib format, identified by the keyword '$* DRRXY' (for detector reaction rates) or '$* PWRXY' for power distribution data. They must be given in the full core radial distributions format with the 'P' format identifier in columns 51-60. The additional assumed convention is that non-blank entries in the data file imply the presence of a fuel assembly, in order to define the geometry. For non-instrumented channels, zero may be entered (zero entries will not be tilt-corrected).

The data in the output file will always be identified with the keyword "$* PWRXY". They will be reduced to quarter core geometry by simple averaging, if requested. In this case a data set identified by keyword "$* PERXY" will also be produced to represent the "standard deviation" in the measured power in symmetric core locations (i), where standard deviation is defined by

```
          Per = (1/4) * Sum(i=1,4) (Pi - Pav)**2

          where:  Pav = (1/4) * Sum(i=1,4)  Pi
```

Note that this program may also be used to calculate the standard deviations of measured data on symmetric core locations without performing tilt corrections by setting all tilts to 1.

4.37 Manual for Program REFXSA

From a given WIMS input file for a fuel supercell, from the DIMlib library and from information on the input file, a WIMS input for the reflector is generated.

The reference fuel cell serves as a source of data to define the material composition of the layers next to the reflector. The first five annuli and materials are considered, assuming the following sequence:

1 - Fuel pellet.
2 - Gap.
3 - Cladding.
4 - Spacer grids.
5 - Coolant.

```
Instructions:
The input filename is requested from input (usually the main
CORXSA input file). Alternatively, the explicit input instructions
can be entered manually, if the line input begins with "$* ".
Only three keywords are recognized: '$* WIMLIB ', '$* DIMLIB '
and '$* REFXSA '.
'$* WIMLIB ' WIMS library specifications. Two additional parameter
             are required:
             11-50 - WIMS library summary filename
             51-60 - WIMS library identification name
'$* DIMLIB ' Core Dimensions Library filename is expected as the
             additional parameter in columns 11-50
'$* REFXSA ' Specifications for generating the reflector constants.
             The following additional parameter are expected in
             columns:
             11-50 - reference fuel supercell filename for
                     generating the reflector constants,
             51-60 - baffle thickness (default 3.794) [cm],
                     (by volume conservation around effective
                     core radius),
             61-70 - baffle+water thickness (default 15.427) [cm]
             71-75 - volume fraction of stainless steel in water
                     behind the baffle (default 0.05208 = 5.208%)
                     NOTE: the same composition for water is
                           assumed as for the coolant in the fuel
                           supercell. To this composition, SS
                           is admixed homogeneously, if specified.
                           To suppress SS addition and override the
                           default value, enter a negative number.
             76-80 - if > 0, total reflector thickness [cm],
                     if < 0, -multiple of cell sidelength,
                     (default -16, implying reflector thickness
                     16-times the cell sidelength).
All other input instructions encountered on input are ignored.
```

4.38 Manual for Program SHTOLO

```
Reformat the file SHEMA.nCY which contains the loading scheme
for cycle "n" into standard CORlib format. Data are assumed to
be delimited with blanks. Full core configuration is assumed.
 Input instructions:
Instructions:
Only the cycle number and cycle length [MWd/tU] are requested
interactively form keyboard input.

Files:  unit  2 - input file (filename defined internally)
              4 - CORlib output file (filename defined internally)
              5 - keyboard input
              6 - terminal screen
```

4.39 Manual for Program WHTOCO

```
The program is intended as an aid for formatting digitized core
design information in CORlib format. The radial distribution data
are particularly tedious to enter by hand.
  The source file is a an ASCII text file, produced by digitizing
a page from a core design document. The information is usually
numeric, but alphanumeric entries (such as fuel assembly loading
```

pattern or region assignment data) can also be processed. The data block can be preceeded by a block (single entries per row) which carry information such as the current cycle burnup step, critical boron concentration, etc. This information is processed and can optionally be entered at a specified place on the CORlib header record of each data entry. No other information should be present on the file. Blank lines are ignored. Lines containing the form-feed character are also ignored, except if they appear in the middle of a data block. This is considered an error and processing is terminated.

Instructions:

The source text file and the CORlib output filenames are requested from input.

Then the number of entries per row is specified. This value appears as the array dimension in the CORlib header format descriptor. Symmetric arrays are assumed at present). The number of data entries per row encountered on the file must not exceed the row length.

In the design document it is customary to specify more than one data type per assembly, which appear in blocks containing more than one row. The number of rows (i.e. data types) is requested from input.

Finally, the keyword to identify each data type is requested from input. From there onwards the data is processed automatically. It is recommended to check the CORlib file which is produced by a visual inspection.

4.40 Manual for Program WGTDET

The presence of a control rod creates a perturbation in the neutron flux distribution which is reflected in the detector response even when the core average power remains the same. To eliminate this virtual power change indication, the flux redistribution correction factors are introduced. They are defined as the ratio of the weighted power distribution Fx for the x-rodded and Fo for the unrodded core. The weights are the probabilities for the neutron born at a particular location in the core to reach the detector.

Instructions

------------

The names are requested from input for files containing the weight, the core power distributions corresponding to specific control rod configurations of interest and the output file onto which the correction factors are printed.

The weights are expected in core quadrant configuration.

The rodded core power distributions can be given either in core quadrant of full core representation.

4.41 Manual for Program WICEDT

The CORXSA filename is requested from input. This file is scanned to define the WICORD filename, the list of requested boron concentrations and the list of isothermal temperatures (fuel, cladding, moderator), or fuel temperatures explicitly. The CORXSA input file is exited such that the first entry after the keyword on the input line carries the current parameter value from the list which follows (up to 6 entries, 10 char. long). The list may be extended on additional records, if necessary. If the current value of a parameter equals the last on the list, the command "EXIT" is issued on the CORXSA.TMP file, except when boron concentration and isothermal temperature are both varied. In this case boron concentration variation is repeated for each entry of the isothermal temperature. The "EXIT" command is then issued when all boron concentrations and temperatures have been processed. Under normal circumstances, the "WICORD" command appears in the CORXSA.TMP file.

The WICORD input file is opened and processed. If the boron concentration or temperature specification input command is encountered ('   BORON ',' TMPFUEL',' TMPCLAD',' TMPMODE'), the parameter value is corrected accordingly. The number of records considered for correction are counted and printed before program termination.

Changes in the WICORD input imply also changes in the core parameters at which the cross sections are calculated. Since the CORXSA code which normally prepares the CORE.XSP file needs not to be executed when the WICORD input file is edited, the WICEDT code opens the CORE.XSP file (if it exists) and edits the appropriate entries as necessary.

Although WICEDT copies all records while editing the CORXSA input file, it does not check the contents. The only commands which are processed are identified by the following keywords:

```
'$* WICINP ' to define the WICORD filename (default WICORD.INP),
'$* BORON  ' to define the boron concentrations [ppm],
'$* ZEROPWR' to define the absolute isothermal temperatures  [K],
'$* DELTISO' to define the isothermal temperature increments [K],
'$* DELTMOD' to define the moderator  temperature increments [K],
'$* DELTFUE' to define the fuel       temperature increments [K],
```

The input records are formatted in the usual 10-column format. The first entry after the keyword defines the current value of the parameter which is processed by other codes. The remaining entries (up to six) represent the list of values which are to be considered. To extend the list to more than six parameters, the last entry on the list should be preceeded by the "+" sign to flag that the next record is part of the list. Additional entries are read in the same manner, up to six entries per record, each ten characters long, starting from column 31. The reserved arrays allow up to 20 entries.

4.42 Manual for Program WICORD

The program WICORD was designed to help the user to prepare WIMS input in accordance with the models defined by the CORD-2 system for PWR Core Design calculations. Some assumptions about geometry and specific input options related to the CORD-2 supercell model are defined internally, but the plant-dependent dimensions and other parameters are read from the CORlib Core Dimensions Library file.

The program was derived from the WIMINP program by S.Slavic. Input instructions differ in format from WIMINP although as a special feature for the convenience of WIMINP users, the program recognizes the old input, re-formats it and makes it available on the WICORD.SCR file.

Instructions:
For simplicity of operation, default values of parameters are

built into the program so that the required input instructions are reduced to a minimum. Each input data record begins with a keyword which uniquely determines the input data option that follows. The order in which the input data options are specified is not important, except in the few cases which are explicitly noted in the instructions.

At present a free formatting capability is not available. The data are expected in ten-column format.

Several WIMS input files can be prepared in a single WICORD run by specifying appropriate WICORD input data sets one after the other in the input file.

The following input options are available:

```
'$* FEWGROU' Define the group system for the transport calculation.
             One additional parameter in Col.11-20 defines the
             selected group structure and may take one of the
             following values:
              -3  to use the full 69-group system,
              -2  to use the 32-group transport system (default),
              -1  to use the 18-group transport system,
              ng  to define a new group system of 'ng' groups.
                  In this case, 'ng' numbers are read in free forma
                  so that the i-th number defines the last group of
                  the original WIMS library which belongs to
                  transport group 'i'.
'$* FASGAP ' Request explicit treatment of the inter-assembly gap
             (the cell dimensions are processed from the pin pitch
             rather than the assembly pitch). Once defined, the
             option applies to all subsequent WICORD input.
'$* WIMSD5 ' Specify the LATTICE and REGULAR keywords instead of
             the internally calculated DANCOFF input option for WIM
             iput (applicable to WIMSD5 code only).
'$* END    ' End of all WICORD input.
'$* WICORD ' The main input card that marks the beginning of a
             WICORD input data set. Additional parameters:
              lbl - character string (up to 20 characters) to
                    identify the cell (for example: the fuel
                    assembly location, or the fuel and moderator
                    temperatures in the case of the isotopic
                    inventory library preparation). By default
                    the label 'lbl' is blank.
             A WICORD input data set includes subcommands which
             are listed below.
'   WIFLNM ' Filename for the WIMS input file. Default name
             is WICINP.INP.
'   COMMENT' Character string (up to 70 characters) which appears
             as a comment on the WIMS input file. Default is blank.
'   CELLTYP' Selection of the cell type according to the CORD-2
             model. Additional parameters are:
              ity - Cell type (Col.11-20) where valid values are:
                     1 - pin cell calculation,
                     2 - fuel super-cell calculation,
                    12 - fuel with integral fuel burnable absorber
                         super-cell calculation,
                    22 - axial blanket of the fuel with integral
                         fuel burnable absorber super-cell
                         calculation,
                    32 - fuel super-cell calculation without the
                         extra water layer on the outside,
                    42 - fuel super-cell with Gd2O3 burnable
                         absorber loaded in the central cell,
                     3 - water super-cell calculation,
                     4 - burnable-poison super-cell calculation,
                     5 - control rod super-cell calculation
                    Default cell type is pin cell.
              fty - Fuel assembly type (Col.21-30) where fty is
                    a character string, consistent with the
                    assembly type designation in the DIMlib file.
              icy - Cycle number (used in case any cycle-dependent
                    data are retrieved from the DIMlib file).
'   FASMUEN' To model fresh fuel assemblies it is convenient to
             specify only the mass of uranium an the enrichment,
             rather than the isotope mass fractions. Furthermore,
             a calculation of an "average" fresh fuel assembly is
             often desired. This input option is designed for
             such purpose. Two additional parameter are:
              nel - number of fuel assemblies to be averaged.
                    (Col.11-20). Default is one.
              mfu - material vector number to which the data
                    apply (Col.21-30). Default value is 9
                    (regular fuel). If the central fuel pin
                    is different from the surrounding fuel,
                    the second '   FASMUEN' card is specified
                    with mfu value equal to 10.
             Two additional records are expected, each containing
             'nel' values in ten-column format. The first set
             contains the masses of uranium metal per fuel assembly
             and the second contains the enrichment. These values
             are used to calculate the average mass and average
             enrichment of the fuel assemblies, which are used in
             turn to define the fuel isotopic inventory vector at
             zero burnup.
'   BUSTEPS' The presence of this card implies that at least one
             burnup step is to be performed. One additional
             parameter is:
              nbu - number of burnup steps, exclusive of the
                    calculation at zero burnup. (Col.11-20).
                    Default is zero.
'   BUMWDT ' When nbu>0 then burnup steps must be specified
             explicitly. Additional record(s) containing nbu
             burnup values in units [MWd/tU] is expected
             (ten columns per value).
             NOTE: The number of burnup steps (opt.: '   BUSTEPS)
             must appear before this card. The option '   BUDAYS'
             and this option exclude each other.
'   BUDAYS ' When nbu>0 then burnup steps must be specified
             explicitly. An additional record containing nbu
             burnup increments in units [days] is expected.
             NOTE: the same note as for '   BUMWDT ' applies.
'   BORON  ' Specify boron concentration at each burnup step.
             One additional parameter may be defined:
              cbo - a non-blank entry in Col.11-20 implies that
                    boron concentration is constant (units [ppm])
                    for all burnup steps.
             If 'cbo' is blank, additional record(s) containing
             'nbu' values of boron concentrations is expected. A
             different boron concentration can be assigned to each
             burnup step.
             NOTE: the same note as for '   BUMWDT ' applies.
'   SAMARIU' Request equilibrium samarium in WIMS.
'   POWERTU' Specific power [MW/tU] is specified in Col.11-20.
'   TMPFUEL' Fuel temperature [K] is specified in Col.11-20.
'   TMPCLAD' Cladding temperature [K] is specified in Col.11-20.
'   TMPMODE' Moderator temperature [K] is specified in Col.11-20.
'   RHOMODE' Moderator density [g/cm3] is specified in Col.11-20.
'   IFBA   ' The presence of this card implies the presence if
             integral burnable absorbers in the fuel lining.
'   BPR    ' The presence of this card indicates that burnable
             poison rods are present in the fuel assembly.
'   GD2O3  ' This card indicates that Gd2O3 loaded fuel pin is to
```

```
               be calculated. One additional parameter is expected,
               in columns 11-20, specifying the weight percent of
               Gd2O3 in the fuel.
'   SUPPRES'   Enables WIMS output suppression. The presence of
               this input option triggers suppression of all output
               except that of WIMS chains 1, 2 and 12. These default
               settings can be modified by additional integer
               parameters 'Si' in columns 11-80 (ten characters
               each). If 'Si'>0) then the output of WIMS} chain 'i'
               is suppressed. Alternately, if 'Si'<0 then the output
               of WIMS} chain 'i' is enabled.
               More than one '   SUPPRES' cards can be specified.
'   VANTAG5'   This option triggers VANTAGE-5 fuel specifications.
               One additional parameter defines the number of IFBA
               pins in the assembly. A negative value implies the
               axial blanket.
'   MATVECT'   Isotopic composition vector is defined from input.
               Additional parameters are:
                'nl'  - length of the vector (Col.11-20),
                'nmt' - Number index of the vector to be overwritten
                        (Col.21-30), which may be the following:
                          1 - helium-filled fuel-cladding gap,
                          2 - zircalloy cladding (borated for
                              IFBA pins),
                          3 - inconel,
                          4 - coolant (borated water),
                          5 - control rods (Ag/In/Cd)
                          6 - air-gap,
                          7 - pyrex glass,
                          8 - stainless steel (with a trace of
                              fissile material),
                          9 - fuel.
                'lza' - material specification flag. (Col.31-40).
                        If 'lza' is non-zero, the material numbers
                        are assumed to be given in ZA form and
                        converted to MAT numbers corresponding to
                        the WIMS library defined by the '   WIMLIB'
                        command.
                'wgt' - total weight of the material. If specified,
                        it is used to renormalize the fractional
                        material composition.
               On the record(s) that follow, a total of 'nl'/2 pairs
               of values are read, namely the material designations
               and their weight fractions in the mixture.
'   WIMLIB '   WIMS library definition. The WIMS library summary file
               is read to assign correct WIMS material numbers from
               their ZA designation. Additional parameters:
                'FLNM' - WIMS library summary filename (Col.11-50,
                         default: WIMLIB.SMR).
                'WLIB' - WIMS library identification name (Col.51-60,
                         default: IJS0).
'   DIMLIB '   Define the DIMlib filename containing core dimensions.
'   ENDWICO'   End of a WICORD data set for one WIMS input. Additiona
               input data sets may follow, beginning with the
               '$* WICORD ' command.

Files used:
  1 - Core dimensions library file (defined on input,
      default: DIMLIB)
  2 - WIMS library summary file (defined on input,
      default: WIMLIB)
  3 - Input instructions (Filename defined from keyboard input)
  5 - Keyboard input (to define Input instructions Filename),
  6 - Terminal output for diagnostics,
  7 - Scratch file (contains non-standard input records)
 10 - WIMS input (Filename defined from input)
```

4.43 Manual for Program WIMSMR

```
Read the WIMS library file and write the cross section library
summary information which provides information such as:
 - library type and name (eg.: WIMS IJS0 ),
 - number of groups (total, fast, resonance),
 - number of materials (total, fissile, fission products),
 - isotope material number (MAT) assignment table,
 - atomic mass and atomic number,
 - NF-flag, No.of temperatures, No.of resonance tables.
The contents of the library are checked for consistency (material
number assignment, material declarations in the burnup chains,
cross section consistency). All messages are written to the
file WIMSMR.LOG.

Instructions:
The Wims library Ident name and its File name are requested from
Unit 5 (suitable for keyboard input, 2 records, format A8 and A20,
respectively). Summary file is written on Unit 2, error messages
and warnings on Unit 7.
Files used:   1  WIMS library (filename assigned from input)
              2  Output Library summary file (assigned from input)
              5  Input (keyboard)
              6  Diagnostics output (screen)
              7  Log-file WIMSMR.LOG (opened explicitly).
```

4.44 Manual for Program WITOIS

```
A WIMS output from the calculation for an effective fuel cell of
PWR fuel assembly to EOL burnup is processed. The results are enter
into the Isotopic inventory library.
  The number densities are read from Chain2 output and the
burnup is estimated from the last burnup step preceeding
the Chain2 output.
  The number densities are read for the MFU-th material
where MFU index is specified from input.
  The burnup is calculated by summing burnup increments
calculated from the overall burnup and power level of the
supercell. The total burnup increment from the previous
burnup step and the average power level during the burnup
step are calculated. The power sharing of the MRE-th
burnable region is taken into account where MRE is
specified from input. In the case of pin cell calculations
with a single fuel region the average burnup from the
WIMS-D output is used.

Files used:
  1  WIMLIB Summary of the WIMS library (fixed filename)
  2  Output Isotopic inventory library,
  3  WIMS output file to be processed (requested from input),
  5  Keyboard input,
  6  Diagnostics output.

Manual
=====
A WIMS output is processed and k-infinity is extracted
```

from chain 14.
The burnup is taken from the Comment card generated by WICORD for non burnup calculations or it is calculated by summing burnup increments calculated from the overall burnup and power level of the supercell. The total burnup increment from the previous burnup step and the average power level during the burnup step are calculated.
In this case, the power sharing of the MTB-th burnable region is taken into account where MTB is specified from the standard input. If the burnup should be read from the Comment Card written by WICORD, enter a negative value for MTB.

The following input data is interactively requested from the standard input:

1. WIMSD output filename
2. PLOTTAB formatted curve filename
3. PlOTTAB formated curve file header (label)
4. MTB-th burnable material region to calculate local burnup. If negative number is entered, it is assumed that the value of BURNUP (BU) will be read from the Comment card prepared by WICORD (This feature is used by the code ISOWKI)

Unit numbers and files used:

1 WIMS output file to be processed (requested from input),
2 Output curve file (requested from input),
5 Keyboard (Standard) input,
6 Diagnostics output.

4.45 Manual for Program XSREDH

EDH method is applied to obtain effective diffusion few-group cross sections. Cylindrical 1D geometry is assumed for the EDH correction. Group condensation is also performed, if requested.

Instructions:
First, the source cross section filename is entered from input. The output cross section filename is entered next. Then the number of fine groups NGF on the input cross section file is defined. The number of coarse macro-groups NGC can also be defined, in which case the macro-group structure must also be given. It is entered as a set of NGC numbers, separated by a blank (free-format), giving the last fine group belonging to the macro-group. The last fine group belonging to the last macro-group must equal NGF by definition. If blank is entered for NGC, the request for the macro-group structure is skipped and no group condensation is performed.

In addition to the usual cross sections, the input cross sections file contains the group average flux, the boundary flux, the outer boundary current, the group buckling-squared, the K-eff and the effective cell radius. The group average flux values must always be given. They are needed for condensation as well as for the EDH correction. If the boundary flux values are not specified, the EDH correction is skipped. If a reflector cell is processed, the current on the outer boundary must be specified and slab geometry is assumed. The current on the inner boundary is calculated as the difference between the total group leakage and the current on the outer boundary. The data are read for each group (NGF+5+5 values). The buckling can be group-dependent, but if zero is entered, the value from the previous group is adopted (a constant buckling may be specified by defining its value for the first group only). For the K-eff and the effective cell radius it is sufficient to specify non-zero entries in only one of the groups, since they do not depend on energy.

NOTE: Buckling-squared (not buckling) is entered.

Cross sections file format:
The cross sections file format is similar to that defined in the GNOMER and BINODE codes, except for the additional parameters. The entries are expected in 10-column format, up to eigh entries per record. The data for each group start on a new record and contain NGF+10 entries. The cross sections are sorted in the following order for each group:

- Diffusion constant [cm],
- Absorption cross section [1/cm],
- Fission yield [1/cm],
- Neutron spectrum group fraction,
- Fission cross section [1/cm] (or fission energy per unit neutron flux [pJ/cm], to yield directly the power),
- Scattering matrix elements [1/cm],
- ON INPUT : Average cell flux (rel.units),
  ON OUTPUT: EDH heterogeneity factor (only when the number of groups is neither 3 nor 11),
- ON INPUT ONLY: Cell boundary flux [rel.units],
- ON INPUT ONLY: Cell outer boundary current when reflector element is processed [rel.units],
- ON INPUT ONLY: Buckling-squared.
- ON INPUT ONLY: K-eff of the supercell,
- ON INPUT ONLY: Effective cell radius (normal cells) or reflector HALF-thickness for slabs with an outer boundary [cm].

Error conditions:
The EDH iterations may terminate with an error flag, which can be interpreted as follows:

```
         0 - normal termination.
 -100*MXIT - no convergence on the diffusion constant in EDH
             iterations (MXIT=40 at present).
-(5000+NIT)- solution is singular at EDH iteration NIT.
```

References:

[1] A.Trkov, M.Ravnik: Effective Diffusion Homogenization of Cross Sections for Pressurized Water Reactor Core Calculations, Nucl.Sci.Eng. Vol.116, No.2, pp.86-95, Feb.1994.

4.46 Manual for Program XSREDT

The program is designed to perform various editing operations on the file containing the cross sections. The program is designed to run interactively.

Input Instructions:
To allow automatic processing and for consistency with other CORD-2 modules the first filename (in the manual input mode the source cross sections file) may be the main CORXSA input filename. The file is scanned for the presence of the "$* AXMESH", "$* COARSGR" $* CRODS " and "$* PINBURN" keywords. If at least the first keyword is present, automatic cross section expansion option is activated and no additional input is necessary. Otherwise, the required

additional input parameters must be entered as follows. Note that the source cross section files are always "CORE" and "CROD" for the unrodded and the rodded core, respectively, with extension "XSR" and "XSP" for the cross sections and the core conditions. The output filenames by default take the form "COnn" and "CRnn", where "nn" is the number of regions. The output filename convention can be overruled with the '$* CORXSR ', '$* CORXSP ', '$* CROXSR ' and 'CROXSP ' input instructions, giving the appropriate filenames in columns 11-50.

WARNING: "CORE.XS*" and "CROD.XS*" are reserved names and should not be used as parameters to redefine the filenames.

First the source and the target cross section filenames are requested, followed by the assumed number of groups. This input is essential and common to all data processing options. The edited cross sections are written to the output file and to a scratch file (which is deleted on program termination). A whole sequence of editing operations may be performed one after the other without redefining the source and the output file, because on the next editing sequence the input is routed automatically to the scratch file and the output file is rewound.

Next, the key-number to request a specific editing option is entered. The following options are available:

1 Edit individual cross section types in selected groups by scaling (this may be required occasionally for test purposes, usually on cell cross sections).

2 Split cross sections into multi-region sets. Core cross sections are usually calculated for a fixed number of axial regions. Sometimes it is required to perform calculations on a denser axial mesh, for example, in calculations with thermohydraulic feedbacks where every axial region must have its own cross section set assigned so that feedback corrections can be applied on the fine mesh. Another example are calculations involving control rods. This edit options defines a cross section set, which corresponds to the coarse axial mesh. By specifying the multiplicity factor for each coarse mesh region, the cross section set is expanded to correspond to the fine axial mesh. The same cross sections are assigned to all fine mesh regions belonging to one coarse mesh region. The core conditions specifications file is also processed in parallel (if specified). Requested additional parameters are the following:

```
  FLNC - Core conditions specifications file (if any),
  NRE  - Number of regions per set,
  NMTi - Multiplicity of the i-th region (i.e. the cross
         sections for the i-th region are copied NMT times).
         Multiplicity is specified for each of the NRE
         regions which make one cross section set.
```

The procedure is repeated automatically for all cross section sets on the file.

3 Make region-wise adjustment to K-inf. A repetitive pattern of regions is assumed (for example, corresponding to one axial channel of the core, similarly as in the previous option). By entering a requested change in K-inf (in units pcm), the thermal absorption cross section is adjusted to correspond to the requested reactivity change. This option, in combination with the one above can be applied to approximately modify cross sections for the assembly spacer grids. Requested additional parameters are the following:

```
  NRE  - Number of regions per set. By default, cross
         section sets are processed in sequence until an
         end-of-file is encountered. By entering -ve NRE,
         the number of sets to be processed NST can be
```

```
          entered explicitly. This option may be used to
          modify the cross sections for a particular
          channel. For example, if the K-inf of the cross
          sections belonging to the j-th channel are to be
          reduced by 200 pcm (each channel containing "m"
          regions), the sequence of operations would be
          the following:
            NRE -1, NST=(j-1)*m, DRO1=   0 ; (j-1)*m x-sect.
                                             sets unaltered
            NRE -1, NST= m,      DRO1=-200 ; m sets modified,
            NRE  1,              DRO1=   0 : rest unaltered.
          Note that a set defined above contains a single
          region. See also instructions for the next entry.
   DROi - Requested change in reactivity (units: pcm) for the
          i-th region. The requested change in reactivity is
          specified for each of the NRE regions which make
          one cross section set.
 NOTE: At present this option is limited to 2-group cases.

4 Interpolate cross sections
  NOTE: This option is not yet implemented.

9 Redefine the essential parameters (filenames and the number
  of groups). Note that the previously processed files are
  closed so a file processed in a previous step may now be
  defined as input, if desired.
```

4.47 Manual for Program XSRLBL

The region-wise cross section data files are labelled with either the assembly name or the core coordinates on the RHS margin in columns 81-90. The assembly-name option is used in the full core case, while the core coordinate option is used when core symmetry is specified. The core geometry and assembly names are read from the CORlib load file. If control rod locations are present on the CORlib file, the rodded core cross sections are also labelled.

Instructions:
Filenames for the CORlib load file and cross section source and labelled output files are requested from input. If control rod locations are found on the CORlib load file, the filenames for the rodded core cross section source and labelled output files are also requested from input. If blank is entered for the source filename, rodded core cross section file processing is skipped.

Cross section files may be generated for multiple axial regions and different core symmetries. The number of axial regions and the core symmetry flag ICS are requested from input. Valid core symmetry flag values are:

```
  0 or 1  - full core symmetry,
   +/- 4  - quadrant core symmetry,
       8  - octant symmetry.
```

4.48 Manual for Program XSRGET

The program reads the cross section library in CORlib format, retri
the cross sections and core parameters at the specified cycle burnu

```
performs expansion or contraction to account for core symmetry,
adds specified axial buckling difference to correct for axial offse
implements axial region multiplicity if explicit calculations on a
axial mesh are required.

Instructions
------------
The following input parameters are expected from the default input
(usually keyboard) in response to the prompt on the default output
(terminal screen):

FLIB  - Source cross section library of core cross sections in
        CORlib format.
FLOU  - Output cross section file in CORlib format for the selected
        burnup step (default XSRGET.TMP).
FLLD  - Core loading pattern in CORlib format (only required if cor
        expansion or condensation is performed).
ICY   - Cycle number.
PCY   - Requested cycle burnup step [MWd/tU].
MLTR  - Axial slice multiplicity (each cross section set in the
        output cross section file is written MLTR-times; the cross
        sections in the CORlib output file are not affected).
DBSQ  - Axial buckling-squared difference (correction term for
        predicted axial offset). The value W*D*DBSQ is added to the
        absorption cross section in each group where D is the
        diffusion constant and W is the weight proportional to the
        axial slice number with value 1 at the top of the core and
        -1 at the core bottom. The correction is only applied in th
        cross section output file and not in the CORlib output file
```

4.49 Manual for Program XSTOCO

```
The program converts XSR and corresponding XSP files in COR format.
Name of the main CORXSA.INP file is requested from the input
(default is CORXSA.INP). From there necessary informations to form
the header of the output COR file (default XSBLIB.COR; can be
changed through '$* XSBLIB ' card in the main CORXSA.INP)
are retrieved. The header keyword depends on data type:
'$* XSCOR'  core region
'$* XSREF'  reflector region
'$* XSCRO'  rodded assemblies
The rest is in standard format with few exceptions:
Columns 45-46 (NXSR) number of processed values from XSR file
                     in each set (5+NG+NP)*NG
Columns 47-48 (NG)   number of XS energy groups
Columns 49-56        temperature instead of power level if P=0

Default XS names : CORE.XSR & XSP
                   REFL.XSR & XSP
                   CROD.XSR & XSP (processed only if '$* CRODS  '
                                   card is present)
can be changed through the manual selection ("-") on input.

NOTE: Processed data are appended to the data already existing
      on XSBLIB file.
```

4.50 Manual for Program XSTOKI

A cross section file such as produced by XSWOUT (strictly limited to the 2-group case) is read. The cross section and the output filenames are requested from terminal input. An input buckling may be specified. From the cross sections the K-inf and the total critical buckling (i.e.: calculated + input) is calculated and written on output

4.51 Manual for Program XSWOUT

Wims output is read and cell-averaged cross sections are calculated. Homogenization can be done using the Effective Diffusion Homogenization method (EDH), or else the simple flux and volume weighting can be applied.

The leakage edit of WIMS is applicable only for fuel-bearing regions and is incorrect if one tries to obtain homogenized cross sections for non-fuelled cells, such as water channels or burnable poison cells. For this reason the Cell-edit in WIMS is processed, in which the cross sections are homogenized by flux and volume weighting.

Definitions:

TRANSPORT CROSS SECTION is reconstructed from the absorption cross section and the scattering matrix. The diffusion constant is defined (D = 1 / 3.SigmaTransport).

ABSORPTION CROSS SECTION is defined as the sum of the capture and fission cross sections.

FISSION YIELD is the product of the fission cross section and the number of neutrons per fission.

NEUTRON SPECTRUM GROUP FRACTION is the fraction of the neutrons born in a particular group in fission.

FISSION ENERGY PER NEUTRON FLUX is the fission cross section scaled with the energy released per fission, such that the product with the neutron flux gives directly the power.

SCATTERING MATRIX is given in terms of the cross sections for scattering from a particular group into all other groups.

CONDENSATION is performed using the cell-average flux from the Cell-edit of WIMS.

EDH method is applied by default to obtain effective diffusion few-group cross sections. Cylindrical 1D geometry is assumed for the EDH correction. The cell boundary flux value is approximated by the average flux in the outermost region of the cell. The exception is the case when a FREE command is encountered on WIMS input. It is assumed that reflector constants are to be generated. Boundary flux is approximated by the innermost region of the cell and a zero-flux boundary condition is taken on the outer boundary of the equivalent homogenized problem.

Instructions:
The following input parameters are read from input:

```
  FLNX  - output cross sections filename
  NG,NP - number of macrogroups NG and the number of additional
          cross section parameters NP. The parameters are read
          from 10-column fields, but blanks are ignored, so the
          entries need not be right-justified.
            By default NP=1 to imply the EDH correction to the
          cross sections and the printout of the EDH flux
          scaling factor. NP>1 may be entered to reserve space
          for the P1 cross section components (for consistency
          with cross sections from other sources), although
          these are all zero in XSWOUT.
            To force suppression of the EDH correction,
          enter negative NG.
```

```
            To apply the EDH method on the cross sections
          but to suppress the printout of the EDH factor
          set NP<0.
  NR(g) - the list containing last groups of the transport
          group structure, belonging to the few-group set
          (i.e.same type of specification as for the FEWGROUP
          command in WIMS). If blank is entered for the previous
          input request (NG, NP), a 10-group set is assumed and
          no request for the group boundaries is issued. By
          default, groups (3,5,6,10,14,21,25,32) of the WIMS
          transport groups are used.
  FLNO -  WIMS output filename to be processed. Requests for
          FLNO are repeated so that several WIMS files can be
          processed in sequence and homogenized cross sections
          written on the same output cross sections file.
          Processing is terminated when a blank filename is
          encountered.

Output cross section format:
  The cross sections are printed in 10-column format which is
compatible with the GNOMER and BINODE codes. The cross sections
are sorted in following order for each group:
 - Diffusion constant [cm],
 - Absorption cross section [1/cm],
 - Fission yield [1/cm],
 - Neutron spectrum group fraction,
 - Fission cross section [1/cm] (or fission energy per unit
   neutron flux [pJ/cm], when burnup printout is available),
 - Scattering matrix elements [1/cm],
 - EDH heterogeneity factor (only when EDH correction to
   the cross sections is specified and the number of groups
   is not 3 or 11).
The output record is limited to 80-columns. If more than three
groups are requested, the cross sections for each group occupy
more than one record. Similarly, more than two records are
needed if the number of groups is greater than 11.

Error conditions:
The EDH iterations may terminate with an error flag, which can be
interpreted as follows:
         0 - normal termination.
 -100*MXIT - no convergence on the diffusion constant in EDH
             iterations (MXIT=40 at present).
-(5000+NIT)- solution is singular at EDH iteration NIT.
```

References:

[1] A.Trkov, M.Ravnik: Effective Diffusion Homogenization of Cross Sections for Pressurized Water Reactor Core Calculations, Nucl.Sci.Eng. Vol.116, No.2, pp.86-95, Feb.1994.

5 CORD-2 Library Formats and Description

5.1 DIMlib Format Specifications

The core dimensions data library is formatted in 8-column format. An entry begins with a header record and may be followed by additional records if necessary. Header records begin with specific Keywords to identify individual data types. Provision is made for entries of a certain data type to be distinguished on the basis of the fuel assembly type and cycle number.

NOTE: When formatting the data, make sure that any cycle-dependent or assembly-type-dependent entries precede the generic entries (with blank cycle and assembly type fields)
The structure of each entry header is the following:

Col.	Description
1 - 8	Keyword
9 - 16	data entry
17 - 24	cycle number (only for cycle-dependent entries)
25 - 30	fuel assembly type (only for ass.type dependent entries)

At present the following keywords have been defined:

```
$* CONFA Number of fuel assemblies in the core             -
$* CONFD Number of fuel assemblies across core diameter    -
$* CORAD Effective core radius                            [cm]
$* COHGT Reactor core height                              [cm]
$* COCZP Core Cold-Zero-Power temperature                 [K]
$* COHZP Core Hot-Zero-Power temperature                  [K]
$* COTIN Core coolant inlet temperature AT HFP conditions [K]
$* COMFL Core coolant mass flow rate                      [kg/s]
$* COPSP Primary system pressure                          [bar]
$* COPWN Core nominal thermal power                       [MW]

$* COCRO Number (NC1) of control rods in the core;
         the following record(s) contains NC1 strings
         (8 characers long). The leading 4 characters contain
         control rod type identification (eg.: A, B, C, D, SA, SB),
         the next four character define the coordinates in the
         core (numeric: W-E, alphabetic: N-S, example: B07, G13...).
$* CRO** Number (MC1) and loc. of control rod pins for control
         rod type '**' (where '**' is a 2-character control rod
         designation, such as defined in '$* COCRO'.
         The following record contains MC1 numbers for pin
         locations in the array: LOC(IX,IY)=IY*100+IX )

$* CRRAD CR absorber material radius                      [cm]
$* CRDGA CR absorber to lining gap thickness              [cm]
$* CRDSS CR Stainless steel lining thickness              [cm]

$* BPR12 Number and location of  12-BPR assemblies
         (same format as FAWHL)
$* BPR16 Number and location of 16-BPR assemblies
         (same format as FAWHL)
$* BPR20 Number and location of 20-BPR assemblies
         (same format as FAWHL)
$* BPRIN BPR pyrex glass inner radius                     [cm]
$* BPROU BPR pyrex glass outer radius                     [cm]
$* BPVSI Volume of BPR inner stainless steel lining       [cm3]
$* BPVSO Volume of BPR outer stainless steel lining       [cm3]
$* FANFP Number of fuel pins in a fuel assembly          -
$* FANPA Assembly pin array size (NPA=17 for 17x17 fuel)   -
$* FAWHL Number (NWH) and loc. of water holes in an ass.   -
         (second record contains NWH numbers for water hole
         locations in the array: LOC(IX,IY)=IY*100+IX )
$* FAPCH Fuel assembly pitch (centre-to-centre)           [cm]
$* FARFP Radius of a fuel pellet                          [cm]
$* FADGA Fuel pellet/cladding gap thickness               [cm]
$* FADCL Fuel cladding thickness                          [cm]
$* FARTH Radius of the guide thimble                      [cm]
$* FADTH Thickness of the guide thimble                   [cm]
$* FACMX Inter-assembly coolant mixing factor              -
$* FAVIN Volume of inconel per assembly                   [cm3]

$* PNPCH Fuel pin pitch (centre-to-centre)                [cm]
         NOTE: This dimension is used to define cell
         dimensions ONLY when the inter-assembly gap is
         treated explicitly. Otherwise the fuel assembly
         pitch "$* FAPCH" is used

$* MTSST Stainless steel composition. The parameter defines the
         composition vector length Abs(NL) which is read from the
         records that follow in the format (10F8.0), giving the
         density and pairs of values (Mat, Wgt). Parameter Wgt
         is always the weight fraction. If NL>0 the Mat
         numbers correspond to the WIMS material numbers, which
         are library dependent. It is preferable to specify NL<0
         in which case Mat are the ZA material designations,
         normally consisting of ZA=1000*Z+A where Z is the atomic
         number and A is the mass number of the isotope. For
         natural elements A=0. Special ZA numbers are assigned
         for some materials:
            1056 - stainless steel
            4107 - Silver-107 as a structural material (non-burnable)
            4109 - Silver-109 as a structural material (non-burnable)
            4115 - Indium-115 as a structural material (non-burnable)

$* MTPYR Pyrex glass composition for BPR. The first material is assumed
         B-10 and the second oxygen. The format is the same as
         for the "$* MTSST" keyword.

$* MTCRO Control rod composition. The format is the same as
         for the "$* MTSST" keyword.

$* MTINC Inconel composition. The format is the same as
         for the "$* MTSST" keyword.

$* MTZRC Zircally composition. The last pair of entries on the list refer
         to B-10 when IFBA fuel is used. Default concentration of B-10 is
         expressed in mg/cm and is processed internally into correct
         number density. The format is the same as for the
         "$* MTSST" keyword.

$* MTVEC Composition of various extra materials that the user may wish
         to define. Additional parameter in columns 9-16 gives the
         number of materials NM. The next record defines:
           IDX  Material index, (limit: IDX<40) (Col. 1-8).
                For new user-defined materials use IDX>10.
           NL   Vector length, as defined for the "$* MTSST"
                keyword (Col. 9-16)
         Records that follow give the density, material idents and weight
         fractions as explained for the "$* MTSST$ keyword.
         The above data are read for all NM materials.

         NOTE: The "$* MTVAR" keyword could be used to define the
```

```
               composition of zircalloy, stainless steel, etc. with
               the corresponding values IDX<10, but the use of special
               keywords for these materials (described above) is preferred.

$* MTGTH Guide thimble material, selected by index as follows:
           2  Zircalloy (default)
           3  Inconel
           8  Stainless steel

$* MTBP1 BPR central region material, selected by index (Col. 9-16)
         of materials (MATVEC) as follows:
           1  Helium
           2  Zircalloy (same as cladding)
           3  Inconel
           4  Coolant (borated water)
           6  Air gap (default)
           7  Pyrex glass
           8  Stainless steel
         Alternatively, a user-defined material index IDX can be specified
         as defined with the '$* MTVEC' keyword.

$* MTCR1 Alternative CR material index (absorber follower) (Col. 9-16)
         as for '$* MTBP1'

$* BUCSQ Global buckling to correct for systematic discrepancies due to
         CORD-2 models, WIMS-D models or the multigroup library.
```

An example of the DIMlib file for the Almaraz-II nuclear power plant is given in Appendix A.1

5.2 FASlib Format Specifications

The design objectives in defining the format of the fuel assembly library were the following:

- The data for each fuel assembly are contained on a separate file. The filename consists of the assembly identification name (4 characters) and the extension ".FAS".
- As an alternative library format, all individual files can be merged into a single data file, since assembly identification is included explicitly in each fuel assembly data set.
- A data type in an assembly data set may consist of one or more records, 80 characters long. Data are formatted in 8 characters columns. Each data type begins with a unique keyword. The advantages of such file organization are:
  - great flexibility in file organization,
  - new data types can be defined and added to the library if necessary,
  - the initial data base can contain a minimum amount of information and can be extended if necessary and when data become available,
  - comments can be included in the library freely, since only records beginning with specific keywords are processed by the retrieval routines.
- A data set for a particular assembly begins with the `$* IDENT` keyword which is explained below.

For convenience, some additional constraints were imposed to simplify the interpretation of data:

- axial distributions are given on 10 axial regions with relative thicknesses

  $$(1, 1, 2, 4, 4, 4, 4, 2, 1, 1).$$

  Such choice of the axial mesh was made because it offers adequate resolution near the edges, where large flux gradients are found and because the region boundaries agree with some physical dimensions in the fuel assembly mechanical design.

- assembly quadrant distributions are given for quadrants (A,B,C,D) in sequence, where quadrant locations are identified in the diagram below:

A	B
C	D

Keywords which are currently defined include the following:

`$* IDENT` for fuel assembly identification. Parameters are:

Column	Description
1- 8	keyword `$* IDENT`,
9-16	fuel assembly name (4-character string),
17-24	fuel assembly region assignment,
25-32	fuel assembly type (for example: WH for Westinghouse standard fuel, V5 for Vantage-5 fuel, KW for KWU fuel etc.),
41-48	edit ident of the last modification (8-character string of the format `XXyymmdd` where `XX` are the initials of the operator performing the edit, `yy` is the year, `mm` the month and `dd` the day of the last edit.

`$* MASSU` to define the mass of heavy metal in the assembly. Parameters are:

Column	Description
1- 8	keyword `$* MASSU`,
9-16	total mass of heavy metal in the assembly [kg]. If the total mass is zero then the mass distribution is assumed to be non-homogenous. Another record is expected (10 values, 8-columns each) to define the mass of heavy metal in each of the axial regions [kg], ordered bottom to top.

`$* ENRIC` to define the enrichment of uranium (or percent weight of fissile material in the heavy metal). Parameters are:

Column	Description
1- 8	keyword `$* ENRIC`,

9-16	average enrichment of uranium in the assembly [%]. If the average enrichment is zero then the enrichment distribution is assumed to be non-homogenous. Another record is expected (10 values, 8-columns each) to define the enrichment of uranium in each of the axial regions [%], ordered bottom to top.

`$* IFBA` to define the number of IFBA pins in the assembly and the ${}^{10}B$ axial distribution. Parameters are:

Column	Description
1- 8	keyword `$* IFBA` ,
9-16	not used
17-24	number of IFBA pins in the assembly, Another record must always be present to define the ${}^{10}B$ concentration per unit length of a fuel pin in each of the axial regions $[mg/cm]$, ordered bottom to top.

`$* FABRN` to define the burnup distribution. Parameters are:

Column	Description
1- 8	keyword `$* FABRN`,
9-16	cycle number (integer, right-justified),
17-24	current cycle burnup $[MWd/tU]$,
25-32	End-of-cy only be defined precisely after the cycle is completed),
33-40	Assembly location (3-character string, right-justified with uppercase alphabetical index for the y-coordinate and numerical index for the x-coordinate. Numbering starts in the top-left corner of a rectangular grid),
41-48	number of BPR's in the assembly in the current cycle.

Additional data may be defined by auxilliary keywords (beginning with blanks instead of the `"$*"` string) which are the following:

`FABAV` to define average burnup. Parameters are:

Column	Description
1- 8	keyword `FABAV`,
9-16	accumulated assembly burnup $[MWd/tU]$,
17-24	assembly power $[MW]$.

`FABAX` to define axial burnup distribution. The relative burnup factors for each axial zone are given on the next record, such that the product with the average accumulated burnup gives the burnup of each axial zone. Data are given bottom-to-top.

`FABQD` assembly quadrant burnup distribution. The relative burnup factors for each assembly quadrant (A,B,C,D) are given on the next record such that the product with the average accumulated burnup (and the corresponding axial burnup factor) gives the burnup of each assembly quadrant (for all axial zones).

`FABRN` to terminate a burnup data set.

An example of the *Faslib* library file for an assembly which has been in the core for three cycles is shown in Appendix A.2.

5.3 CORlib Format Description

The convention for all *Corlib* files is the extension `".COR"`. The rest of the filename is arbitrary but the filename structure `Annxxx.COR"` is recommended where `"A"` is one-character plant identification, `nn` the cycle number (two digit integer) and `"xxx"` is a character string to distinguish different *Corlib* files (for example `K01LOA.COR` for Krško cycle-1 Loading Pattern file, `K02BOC.COR` for the cycle-2 design file etc.).

The *Corlib* library files contain core operational data obtained either from calculations or from the actual measurements. All data are formatted in columns 8-characters long. There may be up to 10 such columns per record.

Each data set is preceeded by a header record which begins with a keyword, unique for each data type. The structure of the header record is the following:

Col. Description

1- 8 keyword (list of currently defined keywords is given below),

9-16 cycle number (integer),

17-24 current cycle burnup [MWd/tU],

25-32 cycle burnup at End-of-cycle [MWd/tU],

33-40 data format descriptor `" CnnnAmm"` where `"C"` identifies character or numeric data, `"A"` denotes format type (currently defined types are `"P", "Q", "T", "X"` and `"Z"`), and `"nnn"` and `"mm"` denote a three-digit and two-digit integers. The precise meaning of the symbols is explained in Table 5.1,

41-48 label (arbitrary, may be used for plant identification or for other purposes),

49-56 fractional power level with respect to the nominal power,

57-64 fractional xenon concentration relative to equilibrium concentration,

65-72 control rod configuration

73-80 boron concentration [ppm].

The header record is followed by the data set. A data set consists of a matrix of values or character strings (`n` $\times$ `m`, say). First, `m` data are written 8-characters per datum and up to 10 values per record. If (`m` $>$ 10), the data occupy more than one record. The process is repeated for all `n` rows of data.
Currently defined keywords are the following:

Table 5.1: Data format descriptor specifications

Symbol	Description
`"C"`	Blank for numeric data, `C` for character data only,
`"A"`	The symbol `"A"` takes one of the following character values: `P` denotes full core description where `nnn` is the maximum number of assemblies in a row (i.e. over the core diameter). The data set consists of a matrix of `nnn`×`nnn` values. Locations in the matrix, which are outside the core are left blank. The value `mm` is the multiplicity number which means, that the data sections of the form above are written `mm` times. Normally `mm` is zero but it may equal to the number of groups (in the case where the energy group fluxes are tabulated, for example), or to the number of axial regions when this format is used to tabulate 3D distributions (radial distributions are given in sequence for each axial region). `Q` denotes quarter core description where `nnn` is the maximum number of fuel assemblies in a row of a core quadrant. The bottom-right quadrant is always assumed. Same conventions regarding `nnn` and `mm` apply as for the `P`-type format. `T` tabular data format where a matrix of `nnn`×`mm` values is given. `X` denotes special format for the full core axial distributions data where `nnn` is the number of assemblies in the core (if not specified) and `mm` is the number of axial regions (assumed 10, if not specified). Conventions are the same as for the `T`-type format (for example, in the case of the Krško NPP with 121 assemblies in the core and for 10 axial regions, the data set is a matrix of 121 × 10 values). `Z` denotes special format for the quarter core axial distributions data where `nnn` is the number of fuel assemblies in the core quadrant. Other conventions are the same as for the `X`-type format.

`$* FASXY` Fuel assembly core loading scheme (entries are four-character fuel assembly identifiers, for example A-01, C-03, HH11).

`$* BPRXY` Number of burnable poison rods at specific core locations.

`$* REGXY` Fuel assembly region assignment in the core.

`$* TYPXY` Fuel assembly types.

`$* MASXY` Mass of heavy metal [Kg].

`$* ENRXY` Fissile material enrichment [%].

`$* OLDXY` Position in the previous cycle (character data consisting of cycle number and standard location designation).

`$* CROXY` Location of control rods in the core. The control rod designation labels (for example: A, B, C ,D, SA, SB) appear on locations where control rods enter the core.

`$* PABXY` Average assembly power during burnup (as a fraction of the nominal power).

`$* BRNXY` Assembly average burnup [MWd/tU].

`$* BRNAX` Assembly axial burnup distribution (normalised to average burnup)

`$* PWRXY` Assembly average relative power distribution.

`$* PWRAX` Assembly axial relative power distribution.

`$* TMOXY` Assembly average moderator temperature [K].

`$* TMOAX` Assembly axial distribution of moderator temperature [K].

`$* TFUXY` Assembly average fuel temperature [K].

`$* TFUAX` Assembly axial distribution of fuel temperature [K].

`$* TCLXY` Assembly average cladding temperature [K].

`$* TCLAX` Assembly axial distribution of cladding temperature [K].

`$* GMOXY` Assembly average moderator density [g/cm^3].

`$* GMOAX` Assembly axial distribution of moderator density [g/cm^3].

`$* TFBXY` Assembly average fuel temperature during burnup (normalized to average).

`$* TFBAX` Axial distribution of average fuel temperature during burnup (normalised to average).

`$* GMBXY` Average moderator density during burnup [g/cm^3].

`$* GMBAX` Axial distribution of average moderator density during burnup (normalized to average).

An example of the Load CORlib file for Almaraz-II Cycle-1 is shown in Appendix A.3

5.4 ISOlib Format Specifications

The convention for the extension of ISOlib filenames is `".ISO"`. The rest of the filename is arbitrary. The ISOlib library files contain isotopic composition vectors as a function of burnup. All data are formatted in columns 8-characters long. There may be up to 10 such columns per record. A data set is preceeded by a header record which begins with a keyword. A single keyword ”`$* ISOFU`” is defined. The structure of the header record is the following:

Col.	**Description**
1- 8	keyword: ”`$* ISOFU`” for standard fuel pins (cell type 2), ”`$* ISOFB`” for fuel pins with integral burnable absorbers IFBA (cell type 12),
9-16	not used,
17-24	not used,
25-32	fuel assembly type designation
33-40	data format descriptor `"  nnnTmm"` where `nnn` is the number of “rows” and `mm` is the number of data columns (each 8-characters long). Note that a “row” can span over more than one record if `nn` is greater than ten.
41-48	average moderator density during burnup [g/cm^3],
49-56	average fuel temperature during burnup [K],
57-64	average moderator temperature during burnup [K] (not used),
65-72	average boron concentration during burnup [ppm],
73-80	number of inserts (BPR’s) during burnup.

The header record is followed by the data set which consists of a block `mmm.nn` of entries. The first entry in row-1 is the enrichment, followed by `nn-1` burnup values [MWd/tU], at which the isotopic composition vectors are tabulated. Column-1 of the remaining rows identifies the isotope in the form `zzhhaaa` where `zz` is the atomic number, `hh` the chemical symbol (left justified) and `aaa` the mass number. The following `nn-1` columns give the masses of individual isotopes at the burnup specified in row-1 in units [g] per *ton* of heavy metal at zero burnup. The exception is the last row, which is reserved for the reactivity decrease or the reactivity coefficient as a function of burnup (it is not used at present). An example of an ISOlib file for 2.1% enriched fuel without inserts, burning with a fuel temperature of $600\,K$ and surrounded by moderator at density $0.7\,g/cm^3$ is shown in Appendix A.4. A complete set of such files, copied to a single file make the master ISOlib isotopic composition vector library.

5.5 WIMlib Format Specifications

The WIMlib library is a summary of the WIMS multigroup library, which provides a table relating the material designation (of the form `zzhhaaa` where `zz` is the atomic number, `hh` the chemical symbol and `aaa` the mass number) and the WIMS material number. It also provides the atomic weights for a consistent calculation of material composition in other programs.

The data in the library are stored in records, each containing ten 8-character words. The header record is identified by the keyword `$* ISOMT`, followed by the data. The description of the library contents is given below.

```
Header:  1  Keyword "$* ISOMT".
         2  Library type ("WIMS" in this case).
         3  Library name or designation. This can be used to distinguish
            calculations with different recommendations for selecting
            materials from the same library. For example, label "IJS0"
            is defined to activate the original actinides. Similarly,
            label "IJS1" can be defined in which the "recommended"
            data sets would be those added to the library (see notes on
            the material version number below).
         4  Number of groups (total).
         5  Number of fast groups.
         6  Number of resonance groups.
         7  Number of materials.
         8  Number of fissile materials.
         9  Number of fission products.
        10  Not used.

Data:    1  Isotope/material designation 'vzzhhaaa', where v is the
            material version number, in case there are more than one
            data sets for the same material in the library.
         2  Material number (MAT).
         3  Atomic weight (atomic mass units).
         4  Atomic number.
         5  Number of temperature tabulations of thermal data.
         6  Number of sets of resonance tables.
         7  Trigger NF (see WIMS library format description).
         8  Fission product (FP) and yield data length.
         9  Parameter NF in FP and yield data (-99 implies: not defined).
        10  not used.

Other conventions:
 - Multiple data sets for a particular material are preceeded by a version
   number (v) which corresponds to the fourth digit in the MAT number,
 - the "recommended" data set appears without the above mentioned number,
 - material designation must be such that unique identification is possible
   from the ZA value (1000*Z + A)
 - Z-value of composite or other special materials is zero
 - A is zero for a natural mixture of isotopesof an element.
```

An example of the WIMlib library is given in Appendix A.5

5.6 RCFlib Format Specifications

The RCFlib reactivity coefficient library is an auxilliary library which is strictly related to the `GNOMER` code to allow thermohydraulic feedback corrections to be applied to the calculated power and temperature distributions. The format is similar to that of the CORlib library. Detailed description and an example is given in the documentation for the `GNOMER` code.

6 Test case description

The selected test case is the Almaraz-II NPP, Cycle-1 which is representative of a 1000 MW(e) pressurized water reactor nuclear power plant. The test case is an international benchmark [6], used in an International Atomic Energy Agency coordinated research project on In-core Fuel Management Code Package Validation. Best estimate core geometry data and some measured results are available.

The test cases describe realistically all important stages of a design calculation. Design accuracy is requested and global calculations are done in three-dimensional cartesian geometry, therefore some of the tests are computationally very intensive.

6.1 Test-1: Expand the FASlib files

In this test the merged FASlib library file from the distribution is split with the `FASSPL` utility code into FASlib files for individual fuel assemblies of the Almaraz-II NPP, Cycle-1. The purpose of the test is to make sure that the CORD-2 package is installed and that the setting are correct, so that CORD-2 utility programs can be executed. The generated FASlib files are used in subsequent tests.

6.2 Test-2: Make the Design CORlib file

From the distribution CORD-2 libraries (DIMlib, FASlib and loading scheme CORlib files), the design CORlib file `A01DSN.COR` is constructed for the Almaraz-II NPP, Cycle-1 at beginning of cycle (BOC). It is needed in nearly all subsequent tests.

The generated `A01DSN.COR` file can be compared with the corresponding `A01BOC.COR` file on the package distribution to check that the data from the FASlib files are assembled correctly into the CORlib file.

6.3 Test-3: Calculate power distribution at HZP

In this test the information in the Design CORlib file is used to generate the cross sections for the Amaraz-II NPP, Cycle-1 core at BOC and Hot-Zero-Power (HZP) conditions in the `CORXSA` procedure. The power distribution is calculated with `GNOMER` and added to the Design CORlib file with the `COREDT` utility code.

The calculated power distribution in the updated `A01DSN.COR` file can be compared to the reference in the distribution file `A01DSN.REF` with the `CORDSP` utility code.

6.4 Test-3a: Calculate control rod worth

The cross sections calculated in Test-3 are used to make a series of calculations with GNOMER for core configurations at different control rod positions. From the calculated values of the multiplication factor k_{ff}, the differential and the integral control rod worths can be obtained. The resulting CORlib file `CRWK.COR` can be processed with the `CORCRW` utility code to sort the differential and the integral rod worths in PLOTTAB format.

6.5 Test-4: Calculate power distribution at HFP

The cross sections calculated in Test-3 are used to calculate approximately the power distribution at Hot-Full-Power (HFP) conditions without xenon (NoXe). The `GNOMER` code is used. The feedback effects are treated entirely through corrections to cross sections by the reactivity coefficient method. A generic reactivity coefficient library RCFlib is used.

The calculated power distribution in the updated `A01DSN.COR` file can be compared to the reference in the distribution file `A01DSN.REF` with the `CORDSP` utility code.

6.6 Test-5: Calculate reactivity coefficients.

A sequence of calculations is performed in which the cross sections are calculated at different boron concentrations and core (isothermal) temperatures for the rodded and the unrodded core of the Almaraz-II NPP, Cycle-1 at BOC. For each case, the values of the multiplication factor are collected in a CORlib file for subsequent analysis. For example, they can be used to improve the RCFlib library. The cross sections are calculated in 2D geometry since the core is axially homogeneous. They are expanded to 24 axial regions for the K-eff and power distribution calculations. In general, the cross sections are usually generated for ten axial regions.

The output CORlib file RCFK.COR can be processed with the CORRCF utility code to calculate some commonly needed reactivity coefficients, critical boron concentrations and control rod worths, which can be compared to the measured values. The RCFK.COR file can be compared to the reference distribution file RCFREF.COR.

6.7 Test-6: Generate the ISOlib library

This is a rather lengthy test case in which the entire isotopic composition vector library `ISOlib` is generated. It is not included in the distribution and is needed if burnup calculations are performed.

Note: in a realistic situation, the ISOlib library is generted once and for all, for a particular fuel type.

6.8 Test-7: Calculate burnup for Cycle-1

In this test case, the full three-dimensional calculations of cross sections are demonstrated, since the core is no longer axially homogeneous due to burnup. The power and temperature distributions are calculated and a critical boron concentration search is performed. Several burnup steps to the end-of-cycle (EOC) are made. The computation is rather lengthy. The calculated power, and burnup distribution in the updated `A01DSN.COR` file can be compared to the reference in the distribution file `A01REF.COR` with the `CORDSP` utility code.

6.9 Test-8: Update the FASlib library

Using the design CORlib file produced in previous test examples, the FASlib files are updated.

7 References

[1] J.R.Askew, F.J.Fayers, P.B.Kemshell, *A General Description of the Code WIMS*, J.Br.Nucl.Energy Soc., **5**, p.564, (1966).

[2] A.Trkov, *GNOMER - Multigroup 3-Dimensional Neutron Diffusion Nodal Code with Thermohydraulic Feedbacks*, Institute Jožef Stefan, Ljubljana, Slovenia, IJS-DP-6688.

[3] A.Trkov, M.Najžer, L.Škerget, *Variant of Green's Function Nodal Method for Neutron Diffusion*, J.Nucl.Sci.Technol., **27**, 8, pp.766-777, (1990).

[4] M.Kromar, S.Slavič, A.Trkov: *CTEMP: A Code for Thermohydraulic Calculations*, Institute Jožef Stefan, Ljubljana, Slovenia, IJS-DP-6143, Feb.1991.

[5] A.Trkov, M.Ravnik, *Effective Diffusion Homogenization of Cross Sections for Pressurized Water Reactor Core Calculations*, Nucl.Sci.Eng., **116**, 2, pp.86-95, (1994).

[6] M.J.Crijns (Scientific secretary): *In-Core Fuel Management Code Package Validation for PWR's*, Coordinated research project, International Atomic Energy Agency, Vienna, IAEA-TECDOC-815, Sep.1995.