TECHNICAL REPORT
IGE–174 Rev. 5

A USER GUIDE FOR DRAGON

Version DRAGON_000331 Release 3.04

G. MARLEAU, A. HÉBERT AND R. ROY

Institut de génie nucléaire
Département de génie mécanique
École Polytechnique de Montréal
April 2000

COPYRIGHT NOTICE FOR DRAGON

ACKNOWLEDGMENTS

The computer code DRAGON results from a concerted effort made at École Polytechnique de Montréal. The main authors of this report would therefore like to express their thanks to École Polytechnique de Montréal for its support along the years as well as to the graduate students and research associates which have contributed to the development of DRAGON along the years. We would also like to thank Kevin E. Kohler at the Nova Southeastern University Oceanographic Center for letting us use and distribute a PostScript utility module derived from his PSPLOT package. Finally, the DRAGON team would never have survived without the financial support of the Natural Science and Engineering Research Council of Canada (NSERC), Hydro–Québec, Atomic Energy of Canada limited (AECL) and the CANDU Owners Group (COG).

SUMMARY

The computer code DRAGON contains a collection of models which can simulate the neutron behavior of a unit cell or a fuel assembly in a nuclear reactor. It includes all of the functions that characterize a lattice cell code, namely: the interpolation of microscopic cross sections which are supplied by means of standard libraries; resonance self-shielding calculations in multidimensional geometries; multigroup and multidimensional neutron flux calculations which can take into account neutron leakage; transport-transport or transport-diffusion equivalence calculations as well as editing of condensed and homogenized nuclear properties for reactor calculations; and finally isotopic depletion calculations.

The code DRAGON contains a multigroup iterator conceived to control a number of different algorithms for the solution of the neutron transport equation. Each of these algorithms is presented in the form of a one group solution procedure where the contributions from other energy groups are included in a source term. The current version of DRAGON contains three such algorithms. The JPM option which solves the integral transport equation using the $J_{\pm}$ method, (interface current method applied to homogeneous blocks); the SYBIL option which solves the integral transport equation using the collision probability method for simple one dimensional (1–D) or two dimensional (2–D) geometries and the interface current method for 2–D cartesian or hexagonal assemblies; and the EXCELL option which solves the integral transport equation using the collision probability method for more general 2–D geometries and for three dimensional (3–D) assemblies.

The execution of DRAGON is controlled by the generalized GAN driver. It is modular and can be interfaced easily with other production codes.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The computer code DRAGON[1–4] results from an effort made at École Polytechnique de Montréal to rationalize and unify into a single code the different models and algorithms used to solve the neutron transport equation. One of the main concerns of the DRAGON development team was to ensure that the structure of the code was such that the development and implementation of new calculation techniques would be facilitated. DRAGON is therefore a lattice cell code which is divided into many calculation modules linked together using the GAN generalized driver[5,6]. These modules exchange information only via well defined data structures.

The two main components of the code DRAGON are its multigroup flux solver and its one group collision probability (CP) tracking modules. The CP modules all perform the same task but using different levels of approximation.

The JPM tracking option uses the interface current technique at the level of each homogeneous zones associated with a geometry ($J_{\pm}$ method).[7] Such calculations can be performed through the use of the `JPMT:` module.[8–12] In this case one can either built the complete collision probabilities matrix or generate a response matrix both of which can be processed by the general multigroup solver. This last method permit a non iterative calculation of the one group neutron flux carried out using sparse matrix algebra.

The SYBIL tracking option emulates the main flux calculation option available in the APOLLO–1 code,[13,14] and includes a new version of the EURYDICE–2 code which performs reactor assembly calculations in both rectangular and hexagonal geometries using the interface current method. SYBIL is slightly more accurate than JPM due to the fact that it performs a complete calculation of the collision probabilities on the whole or a large part of the domain therefore avoiding for a large number of interfaces the angular flux approximation. The option is activated when the `SYBILT:` module is called.

The EXCELL tracking option is used to generate the collision probability matrices for the cases having cluster, two dimensional or three dimensional mixed rectangular and cylindrical geometries.[15,16] A cyclic tracking option is also available for treating specular boundary conditions in two dimensional rectangular geometry.[17–20] EXCELL calculations are performed using the `EXCELT:` module.

After the collision probability or response matrices associated with a given cell have been generated, the multigroup solution module can be activated. This module uses the power iteration method and requires a number of iteration types.[21] The thermal iterations are carried out by DRAGON so as to rebalance the flux distribution only in cases where neutron undergo upscattering. The power iterations are performed by DRAGON to solve the fixed source or eigenvalue problem in the cases where a multiplicative medium is analyzed. The effective multiplication factor (k_{eff}) is obtained during the power iterations. A search for the critical buckling may be superimposed upon the power iterations so as to force the multiplication factor to take on a fixed value.[22]

DRAGON can access directly microscopic cross-section libraries defined according to the following standard formats: MATXS,[23–25] WIMS–D4[26–28], WIMS–AECL[29] and APOLLO.[13] It has the capability of exchanging macroscopic cross-section libraries with a code such as TRANSX–CTR or TRANSX–2 by the use of GOXS and ISOTXS format files.[23,30] The macroscopic cross section can also be read in DRAGON via the input data stream.

2 GENERAL STRUCTURE OF DRAGON INPUT

The input to DRAGON is set up in the form of a structure containing commands which call successively each of the calculation modules required in a given transport calculation.

2.1 Data organization

The structure of the input data is independent of the physical or computational characteristics of the host system. The physical characteristics of the input data is a collection of sequential records. These characters are by necessity ASCII characters. The logical organization of an input deck is in the form of a sequential structure of input variables presented in free format. This structure must be located in the first 72 columns of each record in the input stream. Characters located in column 73 and above can be used to identify the records and are treated as comments. An input variable can be defined in one of two ways.

- As a set of consecutive characters containing no blanks; it will be considered by DRAGON automatically as being either an integer, a real or a character variable depending on the format of the input variable.
- As a set of characters enclosed between quotation marks (’). In this case, the input variable is always considered to be a character variable.

The only separator allowed between two input variables is a single or a set of blanks (not enclosed between quotation marks). A single input variable cannot span two records. Comments can be included in the input deck in one of the following ways:

- characters in column 73 or above on each record are considered to be comments;
- all the information following the ‘`;`’ keyword on a record are not considered by the generalized driver;
- each record starting with the characters ‘`*`’ is considered to be commented out;
- all the characters on a given record following a ‘`!`’ are considered to be commented out.

This users guide was written using the following conventions:

- An input structure represents a set of input variables. It is identified by a name in boldface surrounded by parenthesis. For example, the complete DRAGON input deck is represented by the structure **(DRAGON)**;
- A standard DRAGON data structure represents a set if records and directory stored in a hierarchical format on a direct access XSM file or in memory via a linked list.[4] It is identified by a name in small capital letter. For example, the data structure ASMPIJ contains the multigroup collision probability matrices generated by the `ASM:` module of DRAGON;
- The variables presented using the typewriter font are character variables used as keywords. For example `GEO:` is the keyword required to activate the geometry reading module of DRAGON.
- The variables in italics are user defined variables. When indexed and surrounded by parenthesis they denote arrays. If they are in lower case they represent either integer type (starting with *i* to *n*) or real type (starting with *a* to *h* or *o* to *z*) variables. If they are in upper case they represent character type variables. For example, *iprint* must be replaced in the input deck by an integer variable, (*energy*(g), g=1,*ngroup*+1) states that a vector containing *ngroup*+1 real elements is to be read while *FILE* must be replaced by a character variable, its maximum size being specified. No character variable can exceed 72 character in length.
- The variables or structures surrounded by single square brackets ‘[ ]’ are optional.
- The variables or structures surrounded by double square brackets ‘[[ ]]’ are also optional. However, they can be repeated as many times as required.

- The variables or structures surrounded by braces and separated by vertical bars '{ | | }' represents various calculation options available in DRAGON. Only one of these options is permitted.
- The variables or structures surrounded by '>>' and '<<' represents CLE-2000 output parameters.[5,6]

When a fixed default value is specified for an optional parameter in a structure, it can be modified only locally and is reset to the original default value each time the module is called. When a floating default value is specified for a variable, it is saved and can be used in later calls to this module. In DRAGON, almost every default value is a floating value, with the exception of the parameter *iprint*, which is set to 1 and is used to control the amount of information printed in the module. Departure from this general rule will be indicated in the following sections.

2.2 DRAGON Data Structure and Module Declarations

DRAGON is built around the GAN generalized driver.[5,6] Accordingly, all the modules that will be used during the current execution must be first identified. One must also define the format of each data structure that will be processed by these modules. Then, the modules required for the specific DRAGON calculation are called successively, information being transferred from one module to the next via the data structures. Finally, the execution of DRAGON is terminated when it encounters the `END:` module even if it is followed by additional data records in the input data stream. The general input data structure therefore follows the calling specifications given below:

Table 1: Structure **(DRAGON)**

[`MODULE` (*MODNAME*(i), $i = 1, N_M$) `;`]
[`LINKED_LIST` (*STRNAME*(i), $i = 1, N_L$) `;`]
[`XSM_FILE` (*STRNAME*(i), $i = 1, N_X$) `;`]
[`SEQ_BINARY` (*STRNAME*(i), $i = 1, N_B$) `;`]
[`SEQ_ASCII` (*STRNAME*(i), $i = 1, N_A$) `;`]
[[**(module)** `;`]]
`END: ;`

where

`MODULE` keyword used to specify the list of modules to be used in this DRAGON execution.

MODNAME list of N_M `character*12` name of DRAGON or utility module. The list of DRAGON module is provided in Section 2.3. The number of module declared N_M depends on the particular application of DRAGON.

`LINKED_LIST` keyword used to specify which data structures will be stored in linked lists.

`XSM_FILE` keyword used to specify which data structures will be stored on XSM format files.

`SEQ_BINARY` keyword used to specify which data structures will be stored on sequential binary files.

`SEQ_ASCII` keyword used to specify which data structures will be stored on sequential ASCII files.

STRNAME list of N_L, N_X, N_B or N_A `character*12` name of DRAGON data structure. The list of DRAGON data structure is presented in Section 2.4. The number of module of each type declared depends on the particular application of DRAGON.

(module) input specifications for a DRAGON or utility module. For the DRAGON specific modules these input structures will be defined in Section 3.

`END:` keyword to call the normal end-of-execution utility module.

`;` end of record keyword. This keyword is used by DRAGON to delimit the part of the input data stream associated with each module.

Note that the user generally has the choice to declare most of the data structures in the format of a linked list to reduce CPU times or as a XSM file to reduce memory resources. Some exceptions to this general rule are the tracking files as we will see in Section 3.4. In general, the data structure are stored on the sequential ASCII files only for backup purposes. The input data normally ends with a call to the `END:` module.

Each **(module)** specification contains a description of the execution modules to be called and its associated input structure. All these modules, except the `END:` module may be called more than once.

2.3 The DRAGON Modules

The code DRAGON has been divided into 19 main calculations sequences to which is generally associated a single calculation module. The only exception to this rule is the tracking sequence to which is associated four different modules, one for each of the standard CP calculation options and an additional module for diffusion calculations. However, this later module can only be used indirectly in the edition module of DRAGON. These modules perform the following tasks:

`MAC:` module used to generate or modify a DRAGON MACROLIB (see Section 2.4) which contains the group ordered macroscopic cross sections for a series of mixture (see Section 3.1). This MACROLIB can be either an independent data structure or it can be included as a substructure in a MICROLIB. The spatial location of these mixtures will be defined using the `GEO:` module (see Section 3.3).

`LIB:` module used to generate or modify a DRAGON MICROLIB (see Section 2.4) that can read a number of different types of microscopic cross-section libraries (see Section 3.2). Each such access requires a double interpolation (temperature, dilution) carried out by a subroutine specifically tailored to each type of library. Currently the formats DRAGLIB[36], MATXS[23], WIMS–D4[26,27], and WIMS–AECL[29] are supported. After having reconstructed the microscopic cross sections for each isotope, they are then multiplied by the isotopic concentrations (particles per cm^3) and combined in such a way as to produce an embedded MACROLIB (see Section 2.4). The spatial location of these mixtures will be defined using the `GEO:` module (see Section 3.3).

`GEO:` module used to generate or modify a geometry (see Section 3.3).

`JPMT:` the standard tracking module based on the $J_{\pm}$ technique (see Section 3.4).

`SYBILT:` the standard tracking module based on the Interface Current technique (see Section 3.4).

`EXCELT:` the standard tracking module based on the Collision Probability technique (see Section 3.4).

`BIVACT:` the non standard diffusion tracking module (see Section 3.4).

`SHI:` module used to perform self-shielding calculations (see Section 3.5).

`ASM:` module which uses the tracking information to generate a multigroup response or collision probability matrix (see Section 3.6).

`EXCELL:` module which combines the tracking module `EXCELT:` and the assembly module `ASM:` thereby avoiding the generation of an intermediate binary tracling file (see Section 3.6).

`FLU:` module which uses the multigroup response or collision probability matrix to solve the transport equation for the flux (see Section 3.7).

`MOCC:` module to solve the transport equation for the flux using the cyclic method of characteristics (see Section 3.8).[31, 32] (Release 3.04)

`EDI:` editing module (see Section 3.9).

`EVO:` burnup module (see Section 3.10).

`CPO:` reactor database construction module (see Section 3.11). (Release 3.03)

`INFO:` utility module to compute the density and isotopic contents of heavy or light water and the isotopic contents of UO_2 or $ThUO_2$ fuels (see Section 3.12).

`CFC:` module used to create a reactor cross section database with Feedback coefficients (see Section 3.13).[33–35]

`MRG:` module used to pre-homogenize a geometry tracked using the module `EXCELT:` (see Section 3.14). (Release 3.03)

`PSP:` module to generate PostScript images for 2–D geometries which can be tracked using the module `EXCELT:` (see Section 3.15). (Release 3.04)

2.4 The DRAGON Data Structures

The transfer of information between the DRAGON execution modules is ensured by well defined data structure. They are generally created or modified directly by one of the modules of DRAGON or by one of the utility modules. Here we will give a brief description of these data structures but their complete contents can be found in an independent report.[4]

MACROLIB a standard data structure used by DRAGON to transfer group ordered macroscopic cross sections between its modules. It can be a stand-alone structure or it can be included into a larger structure, such as a MICROLIB or an EDITION structure. When used by a DRAGON module it must be stored either in a linked list or an XSM file. It can be created by the `MAC:`, `LIB:` and `EDI:` modules. It can also be modified by the `SHI:` and `EVO:` modules. Such a structure (either stand-alone or as part of a MICROLIB) is also required for a successful execution of the `ASM:` and `FLU:` modules.

MICROLIB a standard data structure used by DRAGON to transfer microscopic and macroscopic cross sections between its modules. It always include a MACROLIB substructure. It can be a stand-alone structure or included into a larger structure, such as an EDITION structure. When used by a DRAGON module it must be stored either in a linked list or an XSM file. It can be created by the `LIB:` and `EDI:` modules. It can also be modified by the `MAC:`, `SHI:` and `EVO:` modules.

GEOMETRY a standard data structure used by DRAGON to transfer the geometry between its modules. It can be a stand-alone structure or included into a larger structure, such as another GEOMETRY structure. When used by a DRAGON module it must be stored either in a linked list or an XSM file. It can be created by the `GEO:` module. Such a structure is also required directly for a successful execution of the tracking modules (`JPMT:`, `SYBILT:` and `EXCELT:`).

TRACKING a standard data structure used by DRAGON to transfer the general tracking information between its modules. It is a stand-alone structure. When used by a DRAGON module it must be stored on a linked list. It can be created by the `JPMT:`, `SYBILT:` and `EXCELT:` modules. Such a structure is also required directly for a successful execution of the `ASM:` module.

ASMPIJ a standard data structure used by DRAGON to transfer the multigroup response and collision probability matrices between its modules. It is a stand-alone structure. When used by a DRAGON module it must be stored on a linked list or an XSM file. It is created by the `ASM:` module. Such a structure is also required directly for a successful execution of the `FLU:` module.

FLUXUNK a standard data structure used by DRAGON to transfer the flux between its modules. It is a stand-alone structure. When used by a DRAGON module it must be stored on a linked list or an XSM file. It is created by the `FLU:` and `MOCC:` modules. Such a structure is also required for a successful execution of the `EDI:` and `EVO:` modules.

EDITION a standard data structure used by DRAGON to store condensed and merged microscopic and macroscopic cross sections. It is a stand-alone structure but can contain MACROLIB and MICROLIB substructure. When used by a DRAGON module it must be stored on a linked list or an XSM file. It is created by the `EDI:` module. Such a structure is also required for a successful execution of the `CPO:` module.

BURNUP a standard data structure used by DRAGON to store burnup information. It is a stand-alone structure that must be stored on a linked list or an XSM file. It is created by the `EVO:` module. Such a structure is also required for a successful execution of the `CPO:` module.

CPO a standard data structure used by DRAGON to store reactor related diffusion coefficients and microscopic and macroscopic cross sections. It is a stand-alone structure that must be stored on a linked list or an XSM file. It is created by the `CPO:` module.

FBMXSDB a standard data structure used by DRAGON to store a full reactor cross section database with Feedback coefficients. It is a stand-alone structure that must be stored on a linked list or an XSM file. It is created by the `CFC:` module. (Release 3.03)

2.5 Main Updates in DRAGON

The current DRAGON package (DRAGON_000331 Release 3.04) has seen a large number of changes since the first official release of the code (DRAGON_960627). Here we will present the main changes that were introduced along the years to simplify the input and to increase the code capability.

2.5.1 Release 3.02 of DRAGON

(Release 3.02)

Here most of the modifications are related to the treatment of the microscopic cross section libraries while other were implemented to automatize or simplify the input file. The main changes in the input data are for the modules `LIB:`, `SHI:`, `ASM:`, `EDI:` and `INFO:`.

Modifications to the `LIB:` module

The following modifications can be found in Section 3.2.

`MXIS` keyword used to redefine the maximum number of isotopes per mixture.

nmisot the maximum number of isotopes per mixture. By default up to 200 different isotopes per mixture are permitted.

`CTRA` keyword to specify the type of transport correction that should be generated and stored on the MICROLIB. The options following this keyword have been changed.

`NONE` new keyword to specify that no transport correction should be used in this calculation.

`OLDW` keyword to specify that a WIMS type transport correction based on the linearly anisotropic scattering cross sections is to be used for the total and isotropic scattering cross sections. This correction assumes that the micro-reversibility principle is valid only for groups energies less than 4.0 eV. For the remaining groups a $1/E$ flux spectrum is considered in the evaluation of the

transport correction. In addition for WIMS–D4 and WIMS–AECL libraries, linearly anisotropic diagonal scattering cross sections are generated in the cases where the transport correction differs from 0.0 while no anisotropic scattering cross sections is provided on the library. This option was inserted for compatibility with the WIMS transport correction in older versions of DRAGON.

`WIMS` keyword to specify that the transport correction is to be used for the total and isotropic scattering cross sections. In the case where a library in the APOLLO, MATXS format are considered, this correction is identical to the `OLDW` option described above. For WIMS–D4 and WIMS–AECL format libraries, one uses directly the transport correction provided on the library. This is the new recommended option. It should replace the WIMS transport correction used in older version of DRAGON.

`WIMSAECL` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the WIMS–AECL format. This option is identical to the option `WIMS` .

`WIMSD4` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the WIMS–D4 format.

ndepl number of isotopes in the depleting chain.

(descdepl) input structure describing the depletion chain (see Section 3.2.2). It has been completely changed.

Modifications to the `ASM:` module

The following modifications can be found in Section 3.6.

`ARM` keyword to specify that an assembly calculation is carried out without building the full collision probability matrices. This option can only be used for a geometry tracked using the `JPMT:` module. By default, the `PIJ` option is used.

`PIJ` keyword to specify that the standard collision probabilities must be computed. This is now the default option.

Modifications to the `EDI:` module

The following modifications can be found in Section 3.9.

`COND` keyword to specify that a group condensation of the flux is to be performed. The options following this keyword have been modified.

energy array of decreasing energy limits (in eV) that will be associated with each of the *ngcond* condensed groups. We must have $ngcond \leq ngroup+1$. Note that if an energy limit is located between two energy groups, the condensation group will include this associated energy group. In the case where two energy limits fall within the same energy group the lowest energy will be dropped. Finally the maximum and minimum energy limits can be skipped since they will be taken automatically from the information available in the library.

`PERT` keyword to specify that first order perturbations for the microscopic cross sections are to be saved on *EDINAM*.

`ISOTXS` keyword to specify that the set of microscopic cross section generated by the `FLIB`, `MICR` and `ACTI` command will also be saved on a microscopic group neutron cross section library in the ISOTXS–IV format.

New modules

The following module is now available in DRAGON:

`EXCELL:` New module which combines the tracking module `EXCELT:` and the assembly module `ASM:` thereby avoiding the generation of an intermediate binary tracling file (see Section 3.6).

2.5.2 Release 3.03 of DRAGON

(Release 3.03)

Here most of the modifications are related to the treatment of geometry. The main changes in the input data are for the modules `MAC:`, `GEO:` and `EXCELT:`. Two new modules are also available in DRAGON, namely the `CFC:` and `MRG:` modules.

Modifications to the `MAC:` module

The main modifications here are related to the fact that one can now combine a set of MACROLIB and MICROLIB data structure into a single final MACROLIB data structure (see Section 3.1).

Modifications to the `GEO:` module

The modifications that can be found in Section 3.3 are related to the fact that one can now subdivide arbitrarily the Cartesian mesh associated with a `CARCEL`, `CARCELX`, `CARCELY` and `CARCELZ` type geometry while in the previous version of the code the subdivision of the Cartesian mesh associated with such geometry was very restrictive.

Modifications to the `EXCELT:` module

A new option for track normalization option is now available. This option is only supported when the `EXCELT:` module is called.

Modifications to the `ASM:` and `EXCELL:` modules

A new collision probability normalization method is now available. It is also the new default option.

New modules

The following modules are now available in DRAGON:

`CFC:` New module used to create a reactor cross section database with Feedback coefficients (see Section 3.13).

`MRG:` New module used to pre-homogenize a geometry tracked using the module `EXCELT:` (see Section 3.14).

2.5.3 Release 3.04 of DRAGON

(Release 3.04)

Here the modifications are related to the homogenization technique used in DRAGON. The main changes in the input data are for the module `EDI:`. Two new modules are also available in DRAGON, namely the `MOCC:` and `PSP:` modules.

Modifications to the `EDI:` module

The main modifications are related to the fact that one can now select various homogenization options for the anisotropic scattering cross section option `P1SCAT` (see Section 3.9).

Modifications to the `EVO:` module

The main modifications were designed to simplify the `EVO:` input data (see Section 3.10.1).

Modifications to the `INFO:` module

The modifications are required by changes in the GAN generalized driver.[5, 6]

New modules

The following modules are now available in DRAGON:

`MOCC:` New module used to solve the transport equation using the method of cyclic characteristics (see Section 3.8). This module can only be used with EXCELL type 2–D geometries.[31, 32]

`PSP:` New module used to generate a PostScript image for a DRAGON geometry (see Section 3.15). This module can only generate images for a EXCELL type 2–D geometries.

3 THE DRAGON MODULES

The input to DRAGON is set up in the form of a structure containing commands which call successively each of the calculation modules required in a given transport calculation.

3.1 The MAC: module

In DRAGON, the macroscopic cross sections associated with each mixture are stored in a MACROLIB (as an independent data structure or as part of a MICROLIB) which may be generated using one of three different ways. First, one can use directly the input stream already used for the remaining DRAGON data. The second method is via a GOXS format binary sequential file.[23] It should be noted that a number of GOXS files may be read successively by DRAGON and that it is possible to combine data from GOXS files with data taken from the input stream. The third input method is through a file which already contains a MACROLIB. One can also transfer the macroscopic cross sections to a GOXS format binary file if required.

The general format of the data for the MAC: module is the following:

Table 2: Structure **(MAC:)**

```
{ MACLIB := MAC: [ MACLIB ] :: (descmac)
  | MICLIB := MAC: MICLIB :: (descmac)
  | MACLIB := MAC: [ MACLIB ] [ OLDLIB ] :: (descmac)
  }
```

where the first form is for the case where a single macroscopic library is involved, the second form correspond to the case where a single microscopic library is involved, and the third form is valid when two macroscopic and microscopic libraries are to be combined (Release 3.03). The meaning of each of the terms above is:

MACLIB character*12 name of a MACROLIB that will contain the macroscopic cross sections. If *MACLIB* appears on both LHS and RHS, it is updated; otherwise, it is created. If *MACLIB* is created, all macroscopic cross sections are first initialized to zero.

MICLIB character*12 name of a MICROLIB. Only the MACROLIB data substructure of this MICROLIB is then updated. This is used mainly to associate fixed sources densities with various mixtures. If any other cross section is modified for a specific mixture, the microscopic and macroscopic cross sections are no longer compatible. One can return to a compatible library using the library update module (see Section 3.2).

OLDLIB character*12 name of a MACROLIB or a MICROLIB which will be used to update or create the *MACLIB* MACROLIB.

(descmac) macroscopic input data structure for this module (see Section 3.1.1).

3.1.1 Input structure for module MAC:

In the case where there are no *OLDLIB* specified, the **(descmac)** input structure takes the form:

Table 3: Structure **(descmac)**

[EDIT *iprint*]
[NGRO *ngroup*]
[NMIX *nmixt*]
[NIFI *nifiss*]
[ANIS *naniso*]
[CTRA { OFF | APOL | WIMS *igroup* }]
[NALP *nalbp*]
[ALBP (*albedp*(i),i=1,*nalbp*)]
[WRIT *GOXSWN*]
[ENER (*energy*(g), g=1,*ngroup* +1)]
[ADD]
[[{ READ [(*imat*(i), i=1,*nmixt*)] *GOXSRN* [DELE] | READ INPUT [[**(descxs)**]] }]]
[NORM]

In the case where *OLDLIB* is specified, the **(descmac)** input structure takes the form:

Table 4: Structure **(descmac)**

[EDIT *iprint*]
[CTRA OFF]
[[MIX *numnew* [*numold* { UPDL | OLDL }]]]

with

EDIT keyword used to modify the print level *iprint*.

iprint index used to control the printing in this module. It must be set to 0 if no printing on the output file is required. The macroscopic cross sections can written to the output file if the variable *iprint* is greater than or equal to 2. The transfer cross sections will be printed if this parameter is greater than or equal to 3. The normalization of the transfer cross sections will be checked if *iprint* is greater than or equal to 5.

NGRO keyword to specify the number of energy groups for which the macroscopic cross sections will be provided. This information is required only if *MACLIB* is created and the cross sections are taken directly from the input data stream.

ngroup the number of energy groups used for the calculations in DRAGON. By default *ngroup*=1.

NMIX keyword used to define the number of material mixtures. This information is required only if *MACLIB* is created and the cross sections are taken directly from the input data stream or from a GOXS file.

nmixt the maximum number of mixtures (a mixture is characterized by a distinct set of macroscopic cross sections) the MACROLIB may contain. By default *nmixt*=1.

`NIFI` keyword used to specify the maximum number of fissile spectrum associated with each mixture. Each fission spectrum generally represents a fissile isotope. This information is required only if *MACLIB* is created and the cross sections are taken directly from the input data stream.

nifiss the maximum number of fissile isotopes per mixture. By default *nifiss*=1.

`ANIS` keyword used to specify the maximum level of anisotropy permitted in the scattering cross sections. This information is required only if *MACLIB* is created and the cross sections are taken directly from the input data stream.

naniso number of Legendre orders for the representation of the scattering cross sections. By default *naniso*=1 corresponding to the use of isotropic scattering cross sections.

`CTRA` keyword to specify the type of transport correction that should be generated and stored on the MACROLIB. All the modules that will read this MACROLIB will then have access to transport corrected cross sections. By default there is no transport correction when the MACROLIB is created from the input or GOXS files.

`OFF` deactivate the transport correction.

`APOL` keyword to specify that an APOLLO type transport correction based on the linearly anisotropic scattering cross sections is to be used for the total and isotropic scattering cross sections. This correction assumes that the micro-reversibility principle is valid for all energy groups. This option is valid only if the P_1 scattering cross sections exist on the MACROLIB.

`WIMS` keyword to specify that a WIMS–AECL type transport correction based on the linearly anisotropic scattering cross sections is to be used for the total and isotropic scattering cross sections. This correction assumes that the micro-reversibility principle is valid only for groups with an index greater than that specified by the reference group *igroup*. For the remaining groups a $1/E$ flux spectrum is considered in the evaluation of the transport correction.

igroup group number with lowest energy limits which will use a $1/E$ flux spectrum. For the WIMS–AECL 69 groups Winfrith library, *igroup*=27 and the micro-reversibility principle is a used only for group 28 to 69.

`NALP` keyword to specify the maximum number of physical albedo which will be read. These can be used by the `GEO:` module (see Section 3.3).

nalbp the maximum number of physical albedo. By default *nalbp*=0.

`ALBP` keyword used for the input of the physical albedo array.

albedp physical albedo array. A maximum of *nalbp* entries can be specified.

`WRIT` keyword used to write cross section data to a GOXS file.

GOXSWN `character*7` name of the GOXS file to be created or updated.

`ENER` keyword to specify the energy group limits.

energy energy (eV) array which define the limits of the groups (*ngroup*+1 elements). Generally *energy*(1) is the highest energy.

`ADD` keyword for adding increments to existing macroscopic cross sections. In this case, the information provided in **(descxs)** represents incremental rather than standard cross sections.

`READ` keyword to specify the input file format. One can use either the input stream (keyword `INPUT`) or a GOXS format file.

imat array of mixture identifiers to be read from a GOXS file. The maximum number of identifiers permitted is *nmixt* and the maximum value that *imat* may take is *nmixt*. When *imat* is 0, the corresponding mixture on the GOXS file is not included in the MACROLIB. In the cases where *imat* is absent all the mixtures on the GOXS file are available in a DRAGON execution. They are numbered consecutively starting at 1 or from the last number reached during a previous execution of the `MAC:` module.

GOXSRN `character*7` name of the GOXS file to be read.

`DELE` keyword to specify that the GOXS file is deleted after being read (Release 3.03).

`INPUT` keyword to specify that mixture cross sections will be read on the input stream.

(descxs) structure describing the format used for reading the mixture cross sections from the input stream (see Section 3.1.2).

`NORM` keyword to specify that the macroscopic scattering cross sections and the fission spectrum have to be normalized. This option is available even if the mixture cross sections were not read by the `MAC:` module.

`MIX` keyword to specify that the macroscopic cross sections associated with a mixture is to be created or updated.

numnew mixture number to be updated or created on the output MACROLIB.

numold mixture number on an old MACROLIB or MICROLIB which will be used to update or create *numnew* on the output MACROLIB.

`OLDL` the macroscopic cross sections associated with mixture *numold* are taken from *OLDLIB*. This is the default option.

`UPDL` the macroscopic cross sections associated with mixture *numold* are taken from *MACLIB*.

3.1.2 Macroscopic cross section definition

Table 5: Structure **(descxs)**

```
MIX [ matnum ]
      [ EFISS (efiss(i), i=1,nifiss ) ]
      [ TOTAL (xssigt(g), g=1,ngroup) ]
      [ NUSIGF ((xssigf(i,g), g=1,ngroup), i=1,nifiss) ]
      [ NFTOT ((xsfiss(i,g), g=1,ngroup), i=1,nifiss) ]
      [ CHI ((xschi(i,g), g=1,ngroup), i=1,nifiss) ]
      [ FIXE (xsfixe(g), g=1,ngroup) ]
      [ SCAT (( nbscat(l,h), ilastg(l,h),(xsscat(l,h,g),
            g=1,nbscat(l,h) ), h=1,ngroup), l=1,naniso) ]
```

`MIX` keyword to specify that the macroscopic cross sections associated with a new mixture are to be read.

matnum identifier for the next mixture to be read. The maximum value permitted for this identifier is *nmixt*. When *matnum* is absent, the mixtures are numbered consecutively starting with 1 or with the last mixture number read either on the GOXS or the input stream.

`EFISS` keyword to specify the energy released per fission for each fissile isotopes. (Release 3.04)

efiss energy (MeV) released per fission for each fissile isotopes. (Release 3.04)

`TOTAL` keyword to specify that the total macroscopic cross sections for this mixture follows.

xssigt array representing the multigroup total macroscopic cross section (Σ^g in cm^{-1}) associated with this mixture.

`NUSIGF` keyword to specify that the macroscopic fission cross section multiplied by the average number of neutron per fission for this mixture follows.

xssigf array representing the multigroup macroscopic fission cross section multiplied by the average number of neutron per fission ($\nu\Sigma^g_{f,i}$ in cm^{-1}) for all the fissile i isotopes associated with this mixture.

`NFTOT` keyword to specify that the macroscopic fission cross section multiplied by the average number of neutron per fission for this mixture follows. (Release 3.04)

xsfiss array representing the multigroup macroscopic fission cross section ($\Sigma^g_{f,i}$ in cm^{-1}) for all the fissile i isotopes associated with this mixture. (Release 3.04)

`CHI` keyword to specify that the fission spectrum for this mixture follows.

xschi array representing the multigroup fission spectrum (χ^g_i) for all the fissile isotopes i associated with this mixture.

`FIXE` keyword to specify that the fixed neutron source density for this mixture follows.

xsfixe array representing the multigroup fixed neutron source density for this mixture (S^g in $s^{-1}\mathrm{cm}^{-3}$).

`SCAT` keyword to specify that the macroscopic scattering cross section matrix for this mixture follows.

nbscat array representing the number of secondary groups g with non vanishing macroscopic scattering cross section towards the primary group h considered for each anisotropy level associated with this mixture.

ilastg array representing the group index of the most thermal group with non vanishing macroscopic scattering cross section towards the primary group h considered for each anisotropy level associated with this mixture.

xsscat array representing the multigroup macroscopic scattering cross section ($\Sigma^{g\rightarrow h}_{sl}$ in cm^{-1}) from the secondary group g towards the primary group h considered for each anisotropy level associated with this mixture. The elements are ordered using decreasing secondary group number g, from *ilastg* to (*ilastg*$-$*nbscat*$+1$), and an increasing primary group number h. Examples of input structures for macroscopic scattering cross sections can be found in Section 4.1.

3.2 The `LIB:` module

The general format of the input data for the `LIB:` module is the following:

Table 6: Structure **(LIB:)**

MICLIB `:=` `LIB:` [*MICLIB* [*OLDLIB*]] `::` **(desclib)**

where

MICLIB `character*12` name of the MICROLIB that will contain the internal library. If *MICLIB* appears on both LHS and RHS, it is updated; otherwise, it is created.

OLDLIB `character*12` name of a read-only MACROLIB, MICROLIB or BURNUP data structure. In the case where a MACROLIB is considered, it is included directly in the *MICLIB* before updating it. If it is a second MICROLIB or a BURNUP data structure, the number densities for the isotopes in file *MICLIB* will be replaced selectively by those found in *OLDLIB*.

(desclib) input structure for this module (see Section 3.2.1).

3.2.1 Data input for module `LIB:`

In the case where *OLDLIB* is absent or represents a MACROLIB, **(desclib)** takes the form:

Table 7: Structure **(desclib)** for old MACROLIB

[`EDIT` *iprint*]
[`MXIS` *nmisot*]
[`NMIX` *nmixt*]
[`CTRA` { `NONE` | `APOL` | `WIMS` | `OLDW` }] [`ANIS` *naniso*]
[`PROM`]
[`ADED` *nedit* (*HEDIT*(*i*), *i*=1,*nedit*)]
[[{ `CDEPCHN` | `RDEPCHN` }] `DEPL` { `LIB:` { `DRAGON` | `WIMS` | `WIMSAECL` | `WIMSD4` }
`FIL:` *NAMEFIL* | *ndepl* **(descdepl)** }]
[[`MIXS` `LIB:` { `DRAGON` | `MATXS` | `MATXS2` | `WIMSD4` | `WIMS` | `WIMSAECL` | `APLIB1` }
`FIL:` *NAMEFIL* [[**(descmix1)**]]]]

Alternatively if *OLDLIB* is absent or represents a second MICROLIB, **(desclib)** takes the form:

Table 8: Structure **(desclib)** for old MICROLIB

[`EDIT` *iprint*]
`MAXS` [[**(descmix2)**]]

Finally, if *OLDLIB* represents BURNUP structure, **(desclib)** takes the form:

Table 9: Structure **(desclib)** for old BURNUP

[EDIT *iprint*]
BURN { *iburn* | *tburn* } [[**(descmix2)**]]

with

EDIT — keyword used to modify the print level *iprint*.

iprint — index used to control the printing in this module. It must be set to 0 if no printing on the output file is required while values >0 will increase in steps the amount of information transferred to the output file.

MXIS — keyword used to redefine the maximum number of isotopes per mixture.

nmisot — the maximum number of isotopes per mixture. By default up to 200 different isotopes per mixture are permitted.

NMIX — keyword used to define the number of material mixtures. This data is required if *MICLIB* is created.

nmixt — the maximum number of mixtures (a mixture is characterized by a distinct set of macroscopic cross sections).

CTRA — keyword to specify the type of transport correction that should be generated and stored on the MICROLIB. All the modules that will read this MICROLIB will then have access to transport corrected cross sections. The default is no transport correction.

NONE — keyword to specify that an no transport correction should be used in this calculation.

APOL — keyword to specify that an APOLLO type transport correction based on the linearly anisotropic scattering cross sections is to be used for the total and isotropic scattering cross sections. This correction assumes that the micro-reversibility principle is valid for all energy groups. This type of correction uses the P_1 scattering cross sections present on the original library.

OLDW — keyword to specify that a WIMS type transport correction based on the linearly anisotropic scattering cross sections is to be used for the total and isotropic scattering cross sections. This correction assumes that the micro-reversibility principle is valid only for groups energies less than 4.0 eV. For the remaining groups a $1/E$ flux spectrum is considered in the evaluation of the transport correction. In addition for WIMS–D4 and WIMS–AECL libraries, linearly anisotropic diagonal scattering cross sections are generated in the cases where the transport correction differs from 0.0 while no anisotropic scattering cross sections is provided on the library. This option was inserted for compatibility with the WIMS transport correction of older versions of DRAGON (Release 3.02).

WIMS — keyword to specify that the transport correction is to be used for the total and isotropic scattering cross sections. This type of correction uses directly the transport correction cross section provided on the original library. Such information is available only in WIMSD4, WIMSAECL and DRAGON type format libraries. In the case where a library of another type is considered, this correction is identical to the OLDW option described above (Release 3.02).

`ANIS` keyword to specify the maximum level of anisotropy for the scattering cross sections.

naniso number of Legendre orders for the representation of the scattering cross sections. Isotropic scattering is represented by *naniso*=1 while *naniso*=2 represents linearly anisotropic scattering. Generally the linearly anisotropic scattering contributions are taken into account via the transport correction (see `CTRA` keyword) in the transport calculation. For B_1 or P_1 leakage calculations, the linearly anisotropic scattering cross sections are taken into account explicitly. The default value is *naniso*=2.

`PROM` keyword to specify that prompt neutron are to be considered for the calculation of the fission spectrum. By default, the contribution due to delayed neutron is considered. This option is only compatible with a `MATXS` or `MATXS2` format library.

`ADED` keyword to specify the input of additional cross sections to be treated by DRAGON. These cross sections are not needed to solve the transport equation but are recognized by the `EDI:` module.

nedit number of types of additional cross sections.

HEDIT `character*6` name of an additional cross-section type. This name also corresponds to vectorial reactions in a `MATXS` and `MATXS2` format library not automatically recognized by DRAGON. For example:

`NWT0`/`NWT1`=P_0/P_1 library weight functions.
`NTOT0`/`NTOT1`=P_0/P_1 neutron total cross sections.
`NELAS`=Neutron elastic scattering cross sections (MT=2).
`NINEL`=Neutron inelastic scattering cross sections (MT=4).
`NG`=Neutron radiative capture cross sections (MT=102).
`NUDEL`=Number of delayed secondary neutron (Nu-D / MT=455).
`NFSLO`=$\nu*$slow fission cross section.
`NHEAT`=Heat production cross section.
`CHIS`/`CHID`=Slow/delayed fission spectrum.
`NF`/`NNF`/`N2NF`/`N3NF`=$\nu*$partial fission cross sections (MT=19, 20, 21 and 38).
`N2N`/`N3N`/`N4N`=(n,2n), (n,3n), (n,4n) cross sections (MT=16, 17 and 37).
`NP`/`NA`=(n,p) and (n,α) transmutation cross sections (MT=103 and 107).

By default, DRAGON will always attempt to recover the additional cross sections `NG`, and `NHEAT` which may be required for the depletion calculations.

`CDEPCHN` keyword to specify that a complete depletion chain is to be considered. As a result the isotopes in a depletion chain (specified by keyword `DEPL`) not present in a mixture containing burnup material will be added automatically with 0.0 concentrations. This is the default option when the keyword `DEPL` is activated. (Release 3.04)

`RDEPCHN` keyword to specify that a reduced depletion chain is to be considered. As a result the isotopes in a depletion chain (specified by keyword `DEPL`) not present in a mixture containing burnup material will not be added automatically. (Release 3.04)

`DEPL` keyword to specify that the isotopic depletion (burnup) chain is to be read. For a given `LIB:` execution only one isotopic depletion chain can be read.

`MIXS` keyword to specify that the mixture description is to be read. For a given `LIB:` execution more than one cross-section library can be read.

`LIB:` keyword to specify the type of library from which the isotopic depletion chain or microscopic cross section is to be read. It is optional when preceded by the keyword `DEPL` in which case the isotopic depletion chain is read from the standard input file.

`DRAGON` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the DRAGLIB format.[36]

`MATXS` keyword to specify that the microscopic cross sections are in the MATXS format of NJOY–II and NJOY–89 (no depletion data available for libraries using this format).

`MATXS2` keyword to specify that the microscopic cross sections are in the MATXS format of NJOY–91 (no depletion data available for libraries using this format).

`WIMSD4` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the WIMS–D4 format.

`WIMS` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the WIMS–AECL format.

`WIMSAECL` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the WIMS–AECL format.

`APLIB1` keyword to specify that the microscopic cross sections are in the APOLLO–1 format (no depletion chains available for libraries using this format).

`FIL:` keyword to specify the name of the file where is stored the isotopic depletion data.

NAMEFIL `character*8` of the library where the isotopic depletion chain or the microscopic cross sections are stored.

ndepl number of isotopes in the depleting chain.

(descdepl) input structure describing the depletion chain (see Section 3.2.2).

`MAXS` keyword to specify that the mixture density on *MICLIB* are to be modified. If *OLDLIB* is present and **(descmix2)** is absent, a direct one to one correspondence between the isotope on both libraries is assumed. If *OLDLIB* and **(descmix2)** are present, only the mixture on the library file specified by **(descmix2)** are updated using information from the *OLDLIB*. If *OLDLIB* is absent and **(descmix2)** is present, only the mixture on *MICLIB* specified by **(descmix2)** are updated.

`BURN` keyword to specify that the mixture density on *MICLIB* are to be updated using information taken from *OLDLIB*. If **(descmix2)** is absent, a direct one to one correspondence between the isotope on *OLDLIB* and *MICLIB* is assumed. If **(descmix2)** is present, only the mixture specified by **(descmix2)** are updated using information from *OLDLIB*.

iburn burnup step from the burnup file to use. This step must be already present on the burnup file.

tburn burnup time in days from the burnup file to use. This time step must be already present on the burnup file.

(descmix1) input structure describing the isotopic and physical properties of a given mixture (see Section 3.2.3).

(descmix2) input structure describing perturbations to the isotopic and physical properties of a given mixture (see Section 3.2.3).

3.2.2 Depletion data structure

The structure **(descdepl)** describes the heredity of the radioactive decay and the neutron activation chain to be used in the isotopic depletion calculation.

Table 10: Structure **(descdepl)**

```
CHAIN
[[ NAMDPL izae
      [ { STABLE | [[ { DECAY dcr |
            NG | N2N | N3N | N4N | FISSION qf } ]] } ]
      [ FROM [[ { DECAY | NG | N2N | N3N | N4N | FISSION }
            [[ yield NAMPAR ]] ]] ] ]]
ENDCHAIN
```

with:

`CHAIN` keyword to specify the beginning of the depletion chain.

NAMDPL `character*8` name of an isotope (or isomer) of the depletion chain that appears in the cross-section library.

izae six digit integer representing the isotope. The first two digits represent the atomic number of the isotope; the next three indicate its mass number and the last digit indicates the excitation level of the nucleus (0 for a nucleus in its ground state, 1 for an isomer in its first exited state, etc.). For example, ^{238}U in its ground state will be represented by *izae*=922380.

`STABLE` non depleting isotope.

`DECAY` indicates that a decay reaction takes place either for production of this isotope or its depletion.

`NG` indicates that a capture reaction takes place either for production of this isotope or its depletion.

`N2N` indicates that the reaction

$$n + ^{A} X \rightarrow 2n + ^{A-1} X$$

takes place either for production of this isotope or its depletion.

`N3N` indicates that the reaction

$$n + ^{A} X \rightarrow 3n + ^{A-2} X$$

takes place either for production of this isotope or its depletion.

`N4N` indicates that the reaction

$$n + ^{A} X \rightarrow 4n + ^{A-3} X$$

takes place either for production of this isotope or its depletion.

`FISSION` indicates that a decay reaction takes place either for production of this isotope or its depletion.

dcr radioactive decay constant (in 10^{-8} s^{-1}) of the isotope. By default, *dcr*=0.0.

qf energy (in MeV) recoverable per fission of this fissile isotope. It must include the energy associated with the gamma rays generated by radiative capture. By default, *qf*=0.0 MeV.

yield production yield for fission it is expressed in % while for other reaction is is given in absolute value.

NAMPAR `character*8` name of the a parent isotope (or isomer) that appears in the cross-section library.

`ENDCHAIN` keyword to specify the end of the depletion chain.

3.2.3 Mixture description structure

The structure **(descmix1)** is used to describe the isotopic composition and the physical properties, such as the temperature and density, of a mixture.

Table 11: Structure **(descmix1)**

```
MIX [ matnum ] {
      [temp [ denmix ] ]
            [[ [ NAMALI = ] NAMISO dens [ { dil | INF } ]
            [ inrs ] [ DBYE tempd ] [ SHIB NAMS ] [ THER HINC [ TCOH HCOH ] ntfg ] ]] |
      COMB [[ mati relvol ]] }
```

where:

`MIX` keyword to specify the number identifying the next mixture to be read.

matnum mixture identifier. The maximum value that *matnum* may have is *nmixt*. When *matnum* is absent, the mixtures are numbered successively starting from 1 if no mixture has yet been specified or from the last mixture number specified + 1.

temp absolute temperature (in Kelvin) of the isotopic mixture. It is optional only when this mixture is to be updated, in which case the old temperature associated with the mixture is used.

denmix mixture density in $g\ cm^{-3}$.

NAMALI `character*8` alias name for an isotope to be used locally. When the alias name is absent, the isotope name used locally is identical to the isotope name on the library.

`=` keyword to specify to which isotope in a library is associated the previous alias name.

NAMISO `character*12` name of an isotope present in the library which is included in this mixture.

dens isotopic concentration of the isotope *NAMISO* in the current mixture in $10^{24}\mathrm{cm}^{-3}$. When the mixture density *denmix* is specified, the relative weight percentage of each of the isotopes in this mixture is to be provided.

dil group independent microscopic dilution cross section (in barns) of the isotope *NAMISO* in this mixture. It is possible to recalculate a group dependent dilution for an isotope by the use of the `SHI:` module (see Section 3.5). In this case, the dilution is only used as a starting point for the self-shielding iterations and has no effect on the final result. If the dilution is not given or is larger than an 10^{10} barns, an infinite dilution is assumed.

`INF` keyword to specify that a dilution of 10^{10} barns is to be associated with this isotope. This value represents an infinite dilution (the isotope is present in trace amounts only). It is possible to recalculate a group dependent dilution for an isotope by the use of the `SHI:` module (see Section 3.5). In this case, the dilution is only used as a starting point for the self-shielding iterations and has no effect on the final result. If the dilution is not given an infinite dilution is assumed.

inrs number of the resonant region associated with this isotope. By default *inrs*=0 and the isotope is not a candidate for self-shielding. When *inrs*$\neq$0, the isotope can be self-shielded using the

`SHI:` module (see Section 3.5) where it is assumed that a given isotope distributed with different concentrations in a number of mixtures and having the same value of *inrs* will share the same fine flux. This approximation is similar to the PIC approximation of Livolant and Jeanpierre. Should we wish to self-shield both the fuel sheaths and the fuel it is important to assign a different *inrs* number to each. If a single type of fuel is located in different mixture in *onion-peel fashion*, it is necessary to attribute a single *inrs* value to this fuel.

`DBYE` keyword to specify that the absolute temperature of the isotope is different from that of the isotopic mixture. This option is useful to define Debye-corrected temperature.

tempd absolute temperature (in Kelvin) of the isotope. By default *tempd=temp*.

`SHIB` keyword to specify that the name of the isotope containing the information related to the self-shielding is different from the initial name of the isotope.

NAMS `character*8` name of a record in the library containing the self-shielding data. This name is required if the dilution is not infinite or a non zero resonant region is associated with this isotope and *NAMS* is different from *NAMISO*.

`THER` keyword to specify that the thermalization effects are to be included with the cross sections when using a `MATXS` or `MATXS2` format library.

HINC `character*6` name of the incoherent thermalization effects which will be taken into account. The incoherent effects are those that may be described by the $S(\alpha,\beta)$ scattering law. The value `FREE` is used to simulate the effects of a gas.

`TCOH` keyword to specify that coherent thermalization effects will be taken into account.

HCOH `character*6` name of the coherent thermalization effects which will be taken into account. The coherent effects are the *vectorial reactions* in the `MATXS` or `MATXS2` format library where the name is terminated by the '$' suffix. They are generally available for graphite, beryllium, beryllium oxide, polyethylene and zirconium hydroxide.

ntfg number of energy groups that will be affected by the thermalization effects.

`COMB` keyword to specify that this mixture represents a combination of previously defined mixtures.

mati number associated with a previously defined mixture. In order to insert some void in a mixture use *mati*=0. If the mixture is not already defined one assumes that it represents a voided mixture.

relvol relative volume V_m occupied by mixture *mati*=m in *matnum*. Two cases can be considered, namely that where the density ρ_m of each mixture *mati* is provided along with the weight percent for each isotopes J (W_m^J) and the case where the explicit concentration N_m^J of each isotope in a *mati* was provided (it is forbidden to combined two mixtures with different isotopic content description). In the case where the initial mixtures are defined using densities ρ_m, the density (ρ_k) and volume (V_k) of the final mixture will become:

$$V_k = \sum_m V_m$$

$$\rho_k = \frac{1}{V_k} \sum_m \rho_m V_m$$

and the weight percent will be changed in a consistent way, namely

$$W_{k,J} = \frac{\rho_m V_m W_{m,J}}{\rho_k V_k}$$

When the explicit concentration are given we will use:

$$N_{k,J} = \frac{V_m N_{m,J}}{V_k}$$

Note that in the structure **(descmix1)** one only needs to describe the isotopes initially present in each mixture. DRAGON will then automatically associate with each depleting mixture the additional isotopes required by the available burnup chain. Moreover, the microscopic cross-section library associated with these new isotopes will be the same as that of their parent isotope. For example, suppose that mixture 1 contains isotope `U235` which is to be read on the DRAGON format library associated with file `DRAGLIB`. Assume also that the depletion chain, which is written on the WIMS–AECL format library associated with file `WIMSLIB`, states that isotope `U236` (initially absent in the mixture) can be generated form `U235` by neutron capture. Then, one can either specify explicitly from which library file the microscopic cross sections associated with isotope `U236` (zero concentration) are to be read, or omit `U236` from the mixture description in which case DRAGON will assume that the microscopic cross sections associated with isotope `U236` are to be read from the same library as the cross section for isotope `U235`. Note that the isotopes added automatically will remain at infinite dilution.

The structure **(descmix2)** is used to describe the modifications in the isotopic composition of a mixture.

Table 12: Structure **(descmix2)**

`MIX` *matnum* [*matold*] [*relden*] [*NAMALI dens*]

where:

`MIX`	keyword to specify the number identifying the next mixture to be updated.
matnum	mixture identifier on *MICLIB*.
matold	mixture identifier on *OLDLIB*.
relden	relative density of updated mixture. The concentration of each isotope in the mixture is to be multiplied by this factor whether it comes from *MICLIB*, from *OLDLIB* or is specified explicitly using *dens*.
NAMALI	`character*8` alias name for an isotope on *MICLIB* to be modified.
dens	isotopic concentration of the isotope *NAMISO* in the current mixture in $10^{24}\mathrm{cm}^{-3}$. When *denmix* is specified, the isotopic concentration becomes *dens*$\times$*relden*.

3.3 The `GEO:` module

The `GEO:` module is used to create or modify a geometry. The geometry definition module in DRAGON permits all the characteristics (coordinates, region mixture and boundary conditions) of a simple or complex geometry to be specified. The method used to specify the geometry is independent of the discretization module to be used subsequently. Each geometry is stored in the form of a GEOMETRY data structure under its given name. It is always possible to modify an existing geometry or copy it under a new name. The calling specifications are:

Table 13: Structure **(GEO:)**

{ *GEONAM* `:=` `GEO:` { *GEONAM* | *OLDGEO* } `::`
| *GEONAM* `:=` `GEO: ::` **(descgtyp)** } **(descgcnt)**

where

GEONAM `character*12` name of the GEOMETRY created or modified.

OLDGEO `character*12` name of a read-only GEOMETRY. The type and all the characteristics of *OLDGEO* will be copied onto *GEONAM* before this later geometry is modified.

(descgtyp) structure describing the geometry type of *GEONAM* (see Section 3.3.1).

(descgcnt) structure describing the characteristics of a geometry (see Section 3.3.1).

3.3.1 Data input for module `GEO:`

Structures **(descgtyp)** and **(descgcnt)** are used to define respectively the type of geometry that will be define and the contents of this geometry (dimensions, materials, boundary conditions). The module `GEO:` can be recursively called from **(descgcnt)** as an embedded module, in order to define sub-geometries:

Table 14: Structure **(descgtyp)**

{ `VIRTUAL` | `HOMOGE` | `SPHERE` *lr* |
`CAR1D` *lx* | `CAR2D` *lx ly* | `CAR3D` *lx ly lz* |
`HEX` *lh* | `HEXZ` *lh lz* |
`TUBE` *lr* | `TUBEX` *lr lx* | `TUBEY` *lr ly* | `TUBEZ` *lr lz* |
`CARCEL` *lr* [*lx ly*] | `CARCELX` *lr* { *lx* | *lx ly lz* } |
`CARCELY` *lr* { *ly* | *lx ly lz* } | `CARCELZ` *lr* { *lz* | *lx ly lz* } |
`HEXCEL` *lr* | `HEXCELZ` *lr lz* |
`GROUP` *lp* }

Table 15: Structure **(descgcnt)**

[`EDIT` *iprint*]
(descBC)
(descSP)
(descPP)
(descNSG)

continued on next page

Structure **(descgcnt)** continued from last page

[[`:::` *SUBGEO* `:=` `GEO:` { **(descgtyp)** | *SUBGEO* | *OLDGEO* } **(descgcnt)** `;`]]

where

`VIRTUAL`	keyword to specify that a virtual geometry description follows. This type of geometry is used to complete an assembly that has irregular boundaries.
`HOMOGE`	keyword to specify that a infinite homogeneous geometry description follows.
`SPHERE`	keyword to specify that a spherical geometry (concentric spheres) description follows.
`CAR1D`	keyword to specify that a one dimensional plane geometry (infinite slab) description follows.
`CAR2D`	keyword to specify that a two dimensional cartesian geometry description follows.
`CAR3D`	keyword to specify that a three dimensional cartesian geometry description follows.
`HEX`	keyword to specify that a two dimensional hexagonal geometry description follows.
`HEXZ`	keyword to specify that a three dimensional hexagonal geometry description follows.
`TUBE`	keyword to specify that a cylindrical geometry (infinite tubes or cylinders) description follows.
`TUBEX`	keyword to specify that a polar geometry ($R-X$) description follows (Release 3.04).
`TUBEY`	keyword to specify that a polar geometry ($R-Y$) description follows (Release 3.04).
`TUBEZ`	keyword to specify that a polar geometry ($R-Z$) description follows.
`CARCEL`	keyword to specify that a two dimensional mixed cartesian cell (concentric tubes surrounded by a rectangle) description follows.
`CARCELX`	keyword to specify that a three dimensional mixed cartesian cell with tubes oriented along the $X-$axis description follows.
`CARCELY`	keyword to specify that a three dimensional mixed cartesian cell with tubes oriented along the $Y-$axis description follows.
`CARCELZ`	keyword to specify that a three dimensional mixed cartesian cell with tubes oriented along the $Z-$axis description follows.
`HEXCEL`	keyword to specify that a two dimensional mixed hexagonal cell (concentric tubes surrounded by a hexagon) description follows.
`HEXCELZ`	keyword to specify that a three dimensional mixed hexagonal cell with tubes oriented along the $Z-$axis description follows.
`GROUP`	keyword to specify that a *do-it-yourself* type geometry description follows.
lx	number of subdivisions along the $X-$axis (before mesh splitting).
ly	number of subdivisions along the $Y-$axis (before mesh splitting).
lz	number of subdivisions along the $Z-$axis (before mesh splitting).
lr	number of cylinders or spherical shells (before mesh splitting).

lh number of hexagon in an axial plane (including the virtual hexagon).

lp number of types of cells (number of cells inside which a distinct flux will be calculated).

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing in this module. It must be set to 0 if no printing on the output file is required, to 1 for minimum printing (fixed default value) and to 2 for printing the geometry state vector.

(descBC) structure defining the boundary conditions associated with a geometry (see Section 3.3.2).

(descSP) structure defining the spatial coordinates associated with a geometry (see Section 3.3.3).

(descPP) structure defining the physical properties associated with a geometry (see Section 3.3.4).

(descNSG) structure used to specify the properties of non standard geometries (see Section 3.3.5).

SUBGEO `character*12` name of the directory that will contain the sub-geometry.

OLDGEO `character*12` name of a parallel directory containing an existing sub-geometry. The type and all the characteristics of *OLDGEO* will be copied onto *SUBGEO*.

Note that all the geometries described above are called “pure geometries” when they do not contain sub-geometries. When they do contain sub-geometries they will be called “composite geometries”.

3.3.2 Boundary conditions

The inputs corresponding to the **(descBC)** structure are the following:

Table 16: Structure **(descBC)**

[X- { VOID | REFL | TRAN | SYME | ALBE { *albedo* | *icode* } | DIAG }]
[X+ { VOID | REFL | TRAN | SYME | ALBE { *albedo* | *icode* } | DIAG }]
[Y- { VOID | REFL | TRAN | SYME | ALBE { *albedo* | *icode* } | DIAG }]
[Y+ { VOID | REFL | TRAN | SYME | ALBE { *albedo* | *icode* } | DIAG }]
[Z- { VOID | REFL | TRAN | SYME | ALBE { *albedo* | *icode* } }]
[Z+ { VOID | REFL | TRAN | SYME | ALBE { *albedo* | *icode* } }]
[R+ { VOID | REFL | ALBE { *albedo* | *icode* } }]
[HBC { S30 | SA60 | SB60 | S90 | R120 | R180 | SA180 | SB180 | COMPLETE }
{ VOID | REFL | SYME | ALBE { *albedo* | *icode* } }]

where:

`X-`/`X+` keyword to specify the boundary conditions associated with the negative or positive X surface of a cartesian geometry.

`Y-`/`Y+` keyword to specify the boundary conditions associated with the negative or positive Y surface of a cartesian geometry.

`Z-`/`Z+` keyword to specify the boundary conditions associated with the negative or positive Z surface of a cartesian geometry.

`R+` keyword to specify the boundary conditions associated with the outer surface of a cylindrical or spherical geometry.

`VOID` keyword to specify that the surface under consideration has zero reentrant angular flux.

`REFL` keyword to specify that the surface under consideration has a reflective boundary condition.

`TRAN` keyword to specify that the surface under consideration is connected to the opposite surface of a cartesian domain. This option provides the facility to treat an infinite geometry with translation symmetry. The only combinations of translational symmetry permitted are:

- Translation along the $X-$axis

  `X- TRAN X+ TRAN`

- Translation along the $Y-$axis

  `Y- TRAN Y+ TRAN`

- Translation along the $Z-$axis

  `Z- TRAN Z+ TRAN`

`SYME` keyword to specify that the cartesian surface under consideration is virtual and that a reflection symmetry is associated with the adequately directed axis running through the center of the cells closest to this surface.

`DIAG` keyword to specify that the cartesian surface under consideration has the same properties as that associated with a diagonal through the geometry. Note that two and only two `DIAG` surfaces must be specified. The diagonal symmetry is only permitted for square geometry and in the following combinations:

```
X+ DIAG Y- DIAG
```

or

```
X- DIAG Y+ DIAG
```

`ALBE` keyword to specify that the surface under consideration has an arbitrary albedo.

albedo geometric albedo corresponding to the boundary condition `ALBE` (*albedo*$>$0.0).

icode index of a physical albedo corresponding to the boundary condition `ALBE`. The numerical values of the physical albedo are supplied by the module `MAC:` (see Section 3.1).

`HBC` keyword to specify the boundary conditions associated with the outer surface of an hexagonal geometry.

`S30` keyword to specify an hexagonal symmetry of one twelfth of an assembly (see Figure 1).

`SA60` keyword to specify an hexagonal symmetry of one sixth of an assembly of type A (see Figure 1).

`SB60` keyword to specify an hexagonal symmetry of one sixth of an assembly of type B (see Figure 2).

`S90` keyword to specify an hexagonal symmetry of one quarter of an assembly (see Figure 2).

`R120` keyword to specify a rotation symmetry of one third of an assembly (see Figure 3).

`R180` keyword to specify a rotation symmetry of a half assembly (see Figure 3).

`SA180` keyword to specify an hexagonal symmetry of half a type A assembly (see Figure 4).

`SB180` keyword to specify an hexagonal symmetry of half a type B assembly (see Figure 5).

`COMPLETE` keyword to specify a complete hexagonal assembly (see figure Figure 6).

3.3.3 Spatial properties of geometry

The **(descSP)** structure has the following contents:

Table 17: Structure **(descSP)**

[`MESHX` (*xxx*(i), i=1,*lx*+1)]
[`SPLITX` (*ispltx*(i), i=1,*lx*)]
[`MESHY` (*yyy*(i), i=1,*ly*+1)]
[`SPLITY` (*isplty*(i), i=1,*ly*)]
[`MESHZ` (*zzz*(i), i=1,*lz*+1)]
[`SPLITZ` (*ispltz*(i), i=1,*lz*)]
[`RADIUS` (*rrr*(i), i=1,*lr*+1)]
[`OFFCENTER` (*disxyz*(i), i=1,3)]
[`SPLITR` (*ispltr*(i), i=1,*lr*)]
[`SIDE` *sidhex*]
[`NPIN` *npins*] [`RPIN` *rpins*] [`APIN` *apins*] [`DPIN` *dpins*]

`MESHX` keyword to specify the spatial mesh defining the regions along the $X-$axis.

xxx array giving the X limits (cm) of the regions making up the geometry. These values must be given in order, from `X-` to `X+`. If the geometry presents a diagonal symmetry the same data is also used along the $Y-$axis.

`SPLITX` keyword to specify that a mesh splitting of the geometry along the $X-$axis is to be performed.

ispltx array giving the number of zones that will be considered for each region along the $X-$axis. If the geometry presents a diagonal symmetry this information is also used for the splitting along the $Y-$axis. By default, *ispltx*=1.

`MESHY` keyword to specify the spatial mesh defining the regions along the $Y-$axis.

yyy array giving the Y limits (cm) of the regions making up the geometry. These values must be given in order, from `Y-` to `Y+`.

`SPLITY` keyword to specify that a mesh splitting of the geometry along the $Y-$axis is to be performed.

isplty array giving the number of zones that will be considered for each region along the $Y-$axis. By default, *isplty*=1 unless a diagonal symmetry is used in which case *isplty*=*ispltx*.

`MESHZ` keyword to specify the spatial mesh defining the regions along the $Z-$axis.

zzz array giving the Z limits (cm) of the regions making up the geometry. These values must be given in order, from `Z-` to `Z+`.

`SPLITZ` keyword to specify that a mesh splitting of the geometry along the $Z-$axis is to be performed.

ispltz array giving the number of zones that will be considered for each region along the $Z-$axis. By default, *ispltz*=1.

`RADIUS` keyword to specify the spatial mesh along the radial direction.

rrr array giving the radial limits (cm) of the annular regions (cylindrical or spherical) making up the geometry. It is used for the following geometries: `TUBE`, `TUBEZ`, `SPHERE`), `CARCEL`, `CARCELX`, `CARCELY`, `CARCELZ`, `HEXCEL` and `HEXCELZ`. It is important to note that we must have *rrr*(1)=0.0.

`OFFCENTER` keyword to specify that the cylindric regions in a `CARCEL`, `CARCELX`, `CARCELY` and `CARCELZ` geometry can now be displaced from the center of the cell. This option will only be treated when the `EXCELT:` and `EXCELL:` modules are used (Release 3.03).

disxyz array giving the x (*disxyz*(1)), y (*disxyz*(2)) and z (*disxyz*(3)) displacement (cm) of the cylindric regions from the center of a Cartesian cell (Release 3.03).

`SPLITR` keyword to specify that a mesh splitting of the geometry along the radial direction is to be performed.

ispltr array giving the number of zones that will be considered for each region along the radial axis. A negative value results in a splitting of the regions into zones of equal volumes; a positive value results in a uniform splitting along the radial direction. By default, *ispltr*=1.

`SIDE` keyword to specify the length of a side of a hexagon.

sidhex length of a side of a hexagon (cm).

`NPIN` keyword to specify the number of pins located in a cluster geometry. It can only be used for `TUBE` sub-geometry.

npins the number of pins associated with this sub-geometry in the primary geometry.

`RPIN` keyword to specify the radius of an imaginary cylinder where the centers of the pins are to be placed in a cluster geometry. It can only be used for `TUBE` sub-geometry.

rpins the radius (cm) of an imaginary cylinder where the centers of the pins are to be placed.

`APIN` keyword to specify the angle of the first pin centered on an imaginary cylinder in a cluster geometry. It can only be used for `TUBE` sub-geometry.

apins the angle (radian) of the first pin in the ring.

`DPIN` keyword to specify the angular separation between the pins in the ring centered on an imaginary cylinder in a cluster geometry. It can only be used for `TUBE` sub-geometry.

dpins the angular separation (radian) between two pins in the ring.

The user should be warned that the maximum number of zones resulting from the above description of a geometry L_t should not exceed the limits imposed by *maxreg* and defined in the tracking module `JPMT:`, `SYBILT:` or `EXCELT:` (see Section 3.4). For pure geometry with splitting we can define the variables L_x, L_y, L_z, L_r and L_h as:

$$L_x = \sum_{i=1}^{lx} ispltx(i)$$

$$
\begin{aligned}
L_y &= \sum_{i=1}^{ly} ispltY(i) \\
L_z &= \sum_{i=1}^{lz} ispltZ(i) \\
L_r &= \sum_{i=1}^{lr} |ispltR(i)| \\
L_h &= lh
\end{aligned}
$$

and L_t will be given by:

- CAR1D geometry.

$$L_t = L_x$$

- HEX geometry.

$$L_t = L_h$$

- TUBE and SPHERE geometries.

$$L_t = L_r$$

- CAR2D geometry without diagonal symmetry.

$$L_t = L_x L_y$$

- CAR2D geometry with diagonal symmetry.

$$L_t = \frac{L_x(L_y+1)}{2} = \frac{(L_x+1)L_y}{2}$$

- TUBEZ geometry.

$$L_t = L_z L_r$$

- CAR3D geometry without diagonal symmetry.

$$L_t = L_x L_y L_z$$

- CAR3D geometry with diagonal symmetry.

$$L_t = \frac{L_x(L_y+1)L_z}{2} = \frac{(L_x+1)L_y L_z}{2}$$

- HEXZ geometry.

$$L_t = L_z L_h$$

- HEXCEL geometries.

$$L_t = (L_r + 1)$$

- HEXCELZ geometries.

$$L_t = L_z(L_r + 1)$$

- CARCEL geometries.

$$L_t = L_x L_y(L_r + 1)$$

- CARCELX geometry.

$$L_t = L_x L_y L_z(L_r + 1)$$

- CARCELY geometry.

$$L_t = L_x L_y L_z(L_r + 1)$$

- CARCELZ geometries.

$$L_t = L_x L_y L_z(L_r + 1)$$

For mixed geometries, it is important to ensure that L_t which represents the sum over all the sub-geometries of the total number of regions L_t^i associated with each pure sub-geometry i computed using the technique described above. For cluster geometries, only one region is associated with each zone in a pin even if this pin is repeated *npins* times.

3.3.4 Physical properties of geometry

In addition to specifying the mixture associated with each region in the geometry, the **(descPP)** structure is also used to provide information on the sub-geometry required in this geometry. For example, an optional procedure in DRAGON groups together regions so as to reduce the number of unknowns *maxreg* in the flux calculation. In this way, only the merged regions contribute to the cost of the calculation. However, the following points must be considered:

1. All the cells belonging to the same merged region must have the same nuclear properties and dimensions.
2. The grouping procedure is based on the approximation that all the regions belonging to the same merged region share the same flux.
3. The merging can also take into account region orientation (by a rotation and/or transposition) before they are merged. This procedure facilitates the merging of regions when a DIAG or SYME boundary condition is used.

The **(descPP)** structure has the following contents:

Table 18: Structure **(descPP)**

[MIX (*imix*(i), $i = 1, N_t$)]
[CELL (*HCELL*(i), $i = 1, N_t$)]
[MERGE (*imerge*(i), $i = 1, N_t$)]
[TURN (*HTURN*(i), $i = 1, N_t$)]
[CLUSTER (*NAMPIN*(i) ,$i = 1, N_p$)]

where N_p is the number of pin types in the cluster and N_t, which is valid only for pure geometry, is computed in just the same way as L_t except that one uses:

$$\begin{aligned} L_x &= lx \\ L_y &= ly \\ L_z &= lz \\ L_r &= lr \\ L_h &= lh \end{aligned}$$

The inputs associated with this structure have the following meaning:

MIX keyword to specify the isotopic mixture number associated with each region inside the geometry. When diagonal symmetries are considered, only the mixture associated with regions inside the symmetrized geometry need to be specified. When a sub-geometry is located inside symmetrized geometry but outside the calculation region it must be declared *virtual* (for example, the corners of a nuclear reactor).

imix array of mixture numbers associated with a region *imix*$\leq$*maxmix* (see Sections 3.1 and 3.2). If *imix*=0, the corresponding volume is replaced by a void region. These values must be specified in the following order for most geometries (including CARCELZ geometry):

1. radially from the inside out.
2. from surface X- to surface X+
3. from surface Y- to surface Y+
4. from surface Z- to surface Z+

In the cases where a CARCELX geometry is defined then we will use

1. radially from the inside out.
2. from surface Y- to surface Y+
3. from surface Z- to surface Z+
4. from surface X- to surface X+

For a CARCELY geometry the following order is considered:

1. radially from the inside out.
2. from surface Z- to surface Z+

3. from surface `X-` to surface `X+`
4. from surface `Y-` to surface `Y+`

`CELL` keyword to specify the location of the sub-geometry called *generating cells* in a cartesian geometry.

HCELL array of sub-geometry `character*12` names which will be superimposed upon the current cartesian geometry. The same sub-geometry may appear in different positions within the global geometry if the material properties and dimensions are identical. The concept of sub-geometry is useful for the `JPMT:` and `SYBILT:` calculation options since the collision probability matrix associated with each sub-geometry is computed independently of its location in the geometry. In general, the neutron flux in identical sub-geometry located at different locations will be different even if they are associated with the same collision probability matrix. These sub-geometry names must be specified in the following order:

1. from surface `X-` to surface `X+`
2. from surface `Y-` to surface `Y+`
3. from surface `Z-` to surface `Z+`

`MERGE` keyword to specify that some sub-geometries or regions must be merged.

imerge array of numbers that associate a global sub-geometry or region number with each sub-geometry or region. All the sub-geometries or regions with the same global number will be attributed the same flux.

`TURN` keyword to specify that some sub-geometries must be rotated in space before being located at a specific position.

HTURN array of `character*1` keywords to rotate conveniently each sub-geometry. The letters `A` to `L` are used as keywords to specify these rotation. For cartesian geometries, the eight possible orientations are shown in figure Figure 7 while for hexagonal geometries the permitted orientations are shown in figure Figure 8. For 3–D cells, the same letters can be used to describe the rotation in the $X-Y$ plane. However, an additional $-$ sign can be glued to the 2–D rotation identifier to indicate reflection of the cell along the $Z-$axis (`-A` to `-L`).

`CLUSTER` keyword to specify that pin (cylindrical) sub-geometry will be inserted in the geometry (see Figure 9).

NAMPIN array of cylindrical sub-geometry `character*12` name representing a pin. This sub-geometry must be of type `TUBE`.

3.3.5 Non standard geometries

Finally the structure **(descNSG)** provides the possibility to define non standard geometries such as double-heterogeneity and *do-it-yourself* assemblies:

Table 19: Structure **(descNSG)**

[`BIHET` { `TUBE` | `SPHE` } *nmistr nmilg*

continued on next page

Structure **(descNSG)** continued from last page

(*ns*(i), i=1,*nmistr*)
((*rs*(i,j), j=1,*ns*(i)+1), i=1,*nmistr*)
(*milie*(i), i=1,*nmilg*)
(*mixdil*(i), i=1,*nmilg*)
((*fract*(i,j), j=1,*nmistr*) ([(*mixgr*(i,j,k), k=1,*ns*(j))], j=1,*nmistr*), i=1,*nmilg*)]
[POURCE (*pcinl*(i), i=1,*lp*)]
[PROCEL ((*pijcel*(i,j), j=1,*lp*), i=1,*lp*)]

where

BIHET keyword to specify that a sub-geometry made up of spherical or cylindrical micro structures is to be inserted into the current geometry. Each micro structure can be composed of many micro volumes.[37,38]

TUBE keyword to specify that the micro structures are of a cylindrical geometry;

SPHE keyword to specify that the micro structures are of a spherical geometry.

nmistr number of micro structure types in the region. Each type of micro structure is characterized by its dimension and may have distinct volumetric concentrations in each of the macro geometry volumes. All the micro structures of a given type have the same nuclear properties in a given macro volume. The micro structures of a given type may have different nuclear properties within different macro volumes.

nmilg number of micro structures regions.

ns array giving the number of sub-regions (tubes or spherical shells) in the micro structures. Each type of micro structures may contain a different number of micro volumes.

rs array giving the radius of the tubes or spherical shells making up the micro structures. For each type of micro structure i, we will have an initial radius of $rs(1,i) = 0.0$.

milie array giving the composite mixture number associated with each region in the micro structures. These composite mixture numbers must be greater than the maximum number of real mixtures *maxmix*.

mixdil array giving the mixture number associated with each region of the geometry where the micro structure is to be inserted. It is required that *mixdil*$\leq$*maxmix*.

fract array of volumetric concentration (V_G/V_R) of each micro structures (volume V_G) in a given region (volume V_R) of the geometry.

mixgr array giving the mixture number associated with each region of the micro structures. Note that *mixgr* should be specified only for the regions of the micro structure which have a concentration *fract*$>$0.It is required that *mixgr*$\leq$*maxmix*.

POURCE keyword to specify that a *do-it-yourself* type geometry is to be defined, that is to say a geometry resembling the multicell geometry seen in APOLLO–1.[13] This option permits the interactions between different arbitrarily and statistically (default option) arranged cells in an infinite lattice to be treated. The cells are identified by the information following the keyword CELL. The user must ensure that the total number of regions appearing in all the cells must be less than *maxreg*.

pcinl array giving the proportion of each cells type in the lattice such that:

$$|\sum_{i=1}^{lp} pcinl(i) - 1.| < 10^{-5}$$

PROCEL keyword to specify that in a *do-it-yourself* type geometry rather than using a statistical arrangement of cells, a pre-calculated cell distribution is to be considered.

pijcel array giving the pre-calculated probability for a neutron leaving a cell of type i to enter a cell of type j without crossing any other cell. We require:

$$|S(i)pcinl(i)pijcel(i,j) - S(j)pcinl(j)pijcel(j,i)| < 10^{-4}$$

where $S(i)$ and $S(j)$ are the exterior surfaces area of the cells of type i and j respectively.

Examples of geometry definitions for DRAGON can be found in Section 4.2.

3.4 The tracking modules

A tracking module is required to analyze a spatial domain (geometry) assuming a given algorithm will be used for the collision probability calculations. It performs zone numbering operations, volume and surface area calculations and generates the required integration lines for a geometry that was previously defined in the GEO: module. These operations are carried out differently depending on the solution algorithm used.

Four different modules are available for tracking in DRAGON. The JPMT: module is used to perform an interface current tracking inside homogeneous region.[8–12]. The SYBILT: module is used for interface current tracking inside heterogeneous blocks. The EXCELT: module is used to perform full cell collision probability tracking with isotropic[15, 16] or specular[17–20] surface current. These are the transport tracking modules which can be used everywhere in the code where tracking information needs to be generated. The final module BIVACT: is used to perform a diffusion type 2–D tracking which may be required for homogenization purposes.[39] In this case, a finite element discretization will be considered. Note that this module can only be used in the EDI: module since it is not compatible with the flux solution FLU: and self-shielding SHI: modules.

None of these modules can analyzed all of the geometry available in the code DRAGON. In general, the restrictions that apply to a given tracking module result directly from the approximation associated with this method. Moreover, in other instances, some geometries which would have had the same tracking file generated by two different method, such as tube geometry for the SYBILT: and EXCELT: module, have been made available only to one of these tracking module (module SYBILT: in this case).

Accordingly, only the following geometries can be analyzed by the module EXCELT::

1. The two dimensional geometries CAR2D and HEX including respectively CARCEL and HEXCEL sub-geometries.
2. The two dimensional cluster geometries corresponding to a TUBE sub-geometry superimposed on a global TUBE, CARCEL or HEXCEL geometry. Here, the only restriction is that the pins are fully located inside the annular part of the cell and they do not overlap even if they can overlap with internal annular regions.
3. The three dimensional geometry CAR3D including CARCELX, CARCELY and CARCELZ sub-geometries and the three dimensional geometry HEXZ including HEXCELZ sub-geometries.

The geometries that can be treated by the module SYBILT: are

1. The homogeneous geometry HOMOGE.
2. The one dimensional geometries SPHERE, TUBE and CAR1D.
3. The two dimensional geometries CAR2D and HEX including respectively CARCEL and HEXCEL sub-geometries as well as VIRTUAL sub-geometries.

4. Two dimensional non standard geometries containing micro structures.
5. The double heterogeneity option.

The module JPMT: can be used to analyze:

1. The one dimensional geometries SPHERE, TUBE and CAR1D.
2. The two dimensional geometries CAR2D and HEX including respectively CARCEL and HEXCEL sub-geometries as well as VIRTUAL sub-geometries.
3. The two dimensional cluster geometries corresponding to a TUBE sub-geometry superimposed on a global TUBE, CARCEL or HEXCEL geometry. Here, each cluster must be located between two independent annular regions and the clusters must not overlap.
4. The three dimensional TUBEZ geometry.
5. The double heterogeneity option.

Finally, the BIVACT: module can only process directly:

- The two dimensional geometries CAR2D and HEX.

The general information resulting from these tracking is stored in a TRACKING data structure. For the JPMT: and EXCELT: modules, an additional sequential binary tracking file may be generated.

The global numbering of the zones in a geometry proceeds following an order of priorities given by:

- the different rings of a cylindrical or spherical region starting with the inner most after mesh splitting;
- for a cluster regions located in a ring, one first numbers the region inside the pin in the same way as for cylindrical regions and finishes by associating the next region number to the shell of the global geometry which contains this pin. If two cluster types are located in a given ring, they are classified according to increasing *rpins* and *apins* and then numbered in this order. Cluster overlapping annular region are numbered before considering the annular regions.
- the zones in ascending order corresponding to the first axial component (normally X) after mesh splitting;
- the zones in ascending order corresponding to the second axial component (normally Y) after mesh splitting;
- the hexagonal zones corresponding to the order described in Figure 1 to 6.
- the sub-geometry of type CARCELX, CARCELY and CARCELZ are numbered assuming that the third component corresponds to X, Y and Z respectively.

We should also note that symmetry conditions implicitly force the grouping of certain calculation zones. Some other restriction are also imposed for the EXCELT: calculation modules:

- supports only one symmetry condition per axis on the external boundaries of a cartesian assembly; symmetry boundary conditions are not permitted for a single cell.
- allows collision probability calculations to be carried out in exact geometries, where the cells are grouped in such a way as to form a unique geometry. No approximations are made at the cell interfaces.
- only isotropic reflection boundary conditions on the external surfaces of the domain are available for 3–D geometries.

The calling specifications for each of these modules is:

Table 20: Structure **(EXCELT:)**

TRKNAM [*TRKFIL*] `:=` `EXCELT:` [*TRKNAM*] [*TRKFIL*] *GEONAM* `::` **(desctrack) (descexcel)**

Table 21: Structure **(JPMT:)**

TRKNAM [*TRKFIL*] `:=` `JPMT:` [*TRKNAM*] [*TRKFIL*] *GEONAM* `::` **(desctrack) (descjpm)**

Table 22: Structure **(SYBILT:)**

TRKNAM `:=` `SYBILT:` [*TRKNAM*] *GEONAM* `::` **(desctrack) (descsybil)**

Table 23: Structure **(BIVACT:)**

TRKNAM `:=` `BIVACT:` [*TRKNAM*] *GEONAM* `::` **(desctrack) (descbivac)**

where

TRKNAM `character*12` name of the TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information. If *TRKNAM* also appears on the RHS, the previous tracking parameters will be applied by default on the current geometry.

TRKFIL `character*12` name of the sequential binary tracking file used to store the tracks lengths. This file is always required for the `EXCELT:` module, either on the LHS, on the RHS or on both sides. It is also required if the `JPMT:` module is applied to a cluster type geometry. In the case where *TRKFIL* appears on both LHS and RHS, the existing tracking file is modified by the module while if *TRKFIL* appears only on the RHS, the existing tracking file is read but not modified.

GEONAM `character*12` name of the GEOMETRY data structure.

(desctrack) structure describing the general tracking data (see Section 3.4.1)

(descexcel) structure describing the transport tracking data specific to `EXCELT:` (see Section 3.4.2).

(descsybil) structure describing the transport tracking data specific to `SYBILT:` (see Section 3.4.3).

(descjpm) structure describing the transport tracking data specific to `JPMT:` (see Section 3.4.4).

(descbivac) structure describing the diffusion tracking data specific to `BIVACT:` (see Section 3.4.5).

3.4.1 The general tracking data

Table 24: Structure **(desctrack)**

[`EDIT` *iprint*] [`TITL` *TITLE*] [`MAXR` *maxreg*] [{ `RENO` | `NORE` | `RENM` }]

with

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing of this module. The amount of output produced by this tracking modules will vary substantially depending on the print level specified. For example,

- when *iprint*=0 no output is produced;
- when *iprint*=1 a minimum amount of output is produced; the main geometry properties are printed (fixed default option);
- when *iprint*$\geq$2 In addition to the information printed when using *iprint*=1 the zone numbering (zones associated with a flux) is printed;

`TITL` keyword which allows the run title to be set.

TITLE the title associated with a DRAGON run. This title may contain up to 72 characters. The default when `TITL` is not specified is no title.

`MAXR` keyword which permits the maximum number of regions to be considered during a DRAGON run to be specified.

maxreg maximum dimensions of the problem to be considered. The default value is set to the number of regions previously computed by the `GEO:` module. However this value is generally insufficient if symmetries or mesh splitting are specified.

`RENO` keyword to specify the use of the automatic procedure for integration lines normalization to the fine mesh volumes. This normalization procedure should always be used to ensure neutron balances for each fine mesh zone. It is the default option for transport tracking and is forbidden for the `BIVACT:` diffusion tracking module.

`NORE` keyword to specify that the automatic normalization of the integration lines should be deactivated. It is forbidden for the `BIVACT:` diffusion tracking module.

`RENM` keyword to specify the use of the automatic procedure for integration lines normalization to the merged volumes. This normalization procedure should always be used to ensure neutron balances for each merged zone. This option is only valid when the `EXCELT:` module is called (Release 3.03).

3.4.2 The `EXCELT:` specific tracking data

Table 25: Structure **(descexcel)**

[`ANIS` *nanis*]
[{ `PISO` | `PSPC` [`CUT` *pcut*] }]
[`TRAK` { `TISO` [{ `EQW` | `GAUS` }] *nangl dens* [`CORN` *pcorn*] [`SYMM` *isymm*]
| `TSPC` [`MEDI`] *nangl dens* }]

where

`ANIS` keyword to specify the order of anisotropy in collision probability.

nanis order of anisotropy in collision probability calculation. A default value of 1 represents isotropic calculations while a value of 2 correspond to linearly anisotropic collision probability. For the `PIJK` option, a value of 2 is required (see Section 3.6).

`PISO` keyword to specify that a collision probability calculation with isotropic reflection boundary conditions is required. It is the default option if a `TISO` type integration is chosen. To obtain accurate transmission probabilities for the isotropic case it is recommended that the normalization options in the `ASM:` module be used.

`PSPC` keyword to specify that a collision probability calculation with specular reflection boundary conditions required; this is the default option if a `TSPC` type integration is chosen. This calculation is only possible if the file was initially constructed using the `TSPC` option.

`CUT` keyword to specify the input of cutting parameters for the specular integration.

pcut real value representing the maximum error allowed on the exponential function used for specular collision probability calculations. Tracks will be cut at a length such that the error in the probabilities resulting from this reduced track will be of the order of *pcut*. By default, there is no cutting of the tracks and *pcut*=0.0. If this option is used in an entirely reflected case, it is preferable to use the `NORM` command in the `ASM:` module.

`TRAK` keyword to specify the tracking parameters to be used.

`TISO` keyword to specify that isotropic tracking parameters will be supplied. This is the default tracking option for cluster geometries.

`EQW` keyword to specify the use of equal weight quadrature. This option is valid only if an hexagonal geometry is considered.

`GAUS` keyword to specify the use of the Gauss–Legendre or the Gauss–Jacobi quadrature. This option is valid only if an hexagonal geometry is considered.

`TSPC` keyword to specify that specular tracking parameters will be supplied. This option is invalid if an hexagonal geometry is considered.

`MEDI` keyword to specify that instead of selecting the angles located at the end of each angular interval, the angles located in the middle of these intervals are selected. This is particularly useful if one wants to avoid tracking angles that are parallel to the $X-$ or $Y-$axis as its is the case when the external region of a `CARCEL` geometry is voided.

nangl angular quadrature parameter. For applications involving 3–D cells, the choices are *nangl*=2, 4, 8, 10, 12, 14 or 16; these angular quadratures EQ_n present a rotational symmetry about the three cartesian axes. For 2–D isotropic applications, any value of *nangl* may be used, equidistant angles will be selected. For 2–D specular applications the input value must be of the form $p+1$ where p is a prime number (for example p=7,11, etc.); the choice of *nangl* = 8, 12, 14, 18, 20, 24, or 30 are allowed. For cluster type geometries the default value is *nangl*= 10 for isotropic cases and *nangl*=12 for specular cases.

dens real value representing the density of the integration lines (in cm^{-1} for 2–D cases and cm^{-2} for 3–D cases). This choice of density along the plan perpendicular to each angle depends on the geometry of the cell to be analyzed. If there are zones of very small volume, a high line density is essential. This value will be readjusted by `EXCELT:`. In the case of the analysis of a cluster type geometry the default value of this parameter is $5/r_m$ where r_m is the minimum radius of the pins or the minimum thickness of an annular ring in the geometry.

`CORN` keyword to specify that the input of the parameters used to treat the corners for the isotropic integration.

pcorn maximum distance (cm) between a line and the intersection of $n \geq 2$ external surfaces where track redistributing will take place. Track redistributing will take place if a line comes close to the intersection of $n \geq 2$ external surfaces. In this case the line will be replicated n times, each of these lines being associated with a different external surface, while its weight is reduced by a factor of $1/n$. This allows for a better distribution of tracks which are relatively close to n external surfaces. By default, there is no treatment of the corners and *pcorn*=0.0.

`SYMM` keyword to specify that the geometry has a rotation symmetry.

isymm integer value describing the rotation symmetry of the geometry. The fixed default of this parameter is 1.

3.4.3 The `SYBILT:` specific tracking data

Table 26: Structure **(descsybil)**

[`MAXJ` *maxcur*] [`MAXZ` *maxint*]
[`HALT`]
[`QUA1` *iqua1*] [`QUA2` *iqua2 nsegment*] [{ `EQW` | `GAUS` }]
[{ `ROTH` | `ROT+` | `DP00` | `DP01` }]
[`ASKE`] [`LIGN`] [`RECT`]
[`QUAB` *iquab*]

where

`MAXJ` keyword to specify the maximum number of interface currents surrounding the blocks in the calculations.

maxcur the maximum number of interface currents surrounding the blocks. The default value is *maxcur*=max(18,4×*maxreg*) for the `SYBILT:` module.

`MAXZ` keyword to specify the maximum amount of memory required to store the integration lines.

maxint the maximum amount of memory required to store the integration lines. The default value is *maxint*=10000.

`HALT` keyword to specify that the program is to be stopped at the end of the geometry calculations. This option permits the geometry inputs to be checked, the number of blocks and interface currents to be calculated, and a conservative estimate of the memory required for storing the tracks to be made for mixed geometries.

`QUA1` keyword to specify the one dimensional integration parameters.

iqua1 number of basis points for the angular integration of the blocks in a one dimensional geometry. This parameter is not used for `CAR1D` geometries. If a Gauss–Legendre or Gauss–Jacobi quadrature is used, the values of *iqua1* allowed are: 1 to 20, 24, 28, 32 or 64. The default value is *iqua1*=5.

`QUA2` keyword to specify the two dimensional integration parameters.

iqua2 number of basis points for the angular integration of the blocks in a two dimensional geometry appearing during assembly calculations. If a Gauss–Legendre or Gauss–Jacobi formula is used the values allowed for *iqua2* are: 1 to 20, 24, 28, 32 or 64. The default value is *iqua2*=6 and represents the number of angles in $(0, \pi/4)$ for cartesian geometries and $(0, \pi/6)$ for hexagonal geometries.

nsegment number of basis points for the spatial integration of the blocks in a two dimensional geometry appearing during assembly calculations. The default value is *nsegment*=3.

`EQW` keyword to specify the use of equal weight quadrature.

`GAUS` keyword to specify the use of the Gauss–Legendre or the Gauss–Jacobi quadrature. This is the default option.

`ROTH` keyword to specify that the isotropic (DP_0) components of the current at cell interface is used with the incoming current being averaged over all the faces surrounding a cell. The global collision matrix is calculated in a annular model. Only used when 2–D assembly of cells are considered.

`ROT+` keyword to specify that the isotropic (DP_0) components of the current at cell interface is used. The global collision matrix is calculated in a annular model. Only used when 2–D assembly of cells are considered.

`DP00` keyword to specify that the isotropic (DP_0) components of the current at cell interface is used. The global collision matrix are computed explicitly. Only used when 2–D assembly of cells are considered.

`DP01` keyword to specify that the linearly anisotropic (DP_1) components of the current at cell interface are used. This hypothesis implies 12 currents per cell in a cartesian geometry and 18 currents per cell for a hexagonal geometry. Linearly anisotropic reflection is used. Only used when 2–D assembly of cells are considered.

ASKE keyword to specify the use of an *Askew* cylinderization which preserves both the external surface of the cells and the material balance of the external crown (by a modification of its concentration). By default a *Wigner* cylinderization is used which preserves the volume of the external crown. Note, that an assembly of a number of rectangular cells having unequal volumes requires an *Askew* cylinderization. This applies only in cases where the external surface is annular using the ROTH or ROT+ options. Only used when 2–D assembly of cells are considered.

LIGN keyword to specify that all the integration lines are to be printed. This option should only be used when absolutely necessary because it generates a rather large amount of output. Only used when 2–D assembly of cells are considered.

RECT keyword to specify that square cells are to be treated as if they were rectangular cells, with the inherent loss in performance that this entails. This option is of purely academic interest.

QUAB keyword to specify the initial number of basis point for the numerical integration of each micro structure in cases involving double heterogeneity.

iquab the number of basis point for the numerical integration of the collision probabilities in the micro volumes using the Gauss–Jacobi formula. The values permitted are: 1 to 20, 24, 28, 32 or 64. The default value is *iquab*=5.

3.4.4 The JPMT: specific tracking data

Table 27: Structure **(descjpm)**

[MAXJ *maxcur*] [MAXZ *maxint*]
[HALT] [OLD]
[{ IP00 | SP01 | IP01 }]
[QUA1 *iqua1*] [QUA2 *iqua2 nsegment*] [{ EQW | GAUS }]
[{ ROTH | ROT+ | DP00 | DP01 }]
[ASKE] [LIGN] [RECT]
[{ RECD | RECR }]
[{ BP00 | BP01 }] [QUAB *iquab*]

where

MAXJ keyword to specify the maximum number of interface currents surrounding the blocks in the calculations.

maxcur the maximum number of interface currents surrounding the blocks. The default value is *maxcur*=max(50,6×*maxreg*) for the JPMT: module.

MAXZ keyword to specify the maximum amount of memory required to store the integration lines. Not used for cluster geometries.

maxint the maximum amount of memory required to store the integration lines. The default value is *maxint*=10000.

`OLD` keyword to specify that a set of previously calculated collision probabilities previously saved on *TRKNAM* is to be recovered. This option is of interest in cases where the coolant occupies a region of a complex geometry (such as a fuel assembly or bundle) and calculations of isotopic evolution (burnup) or resonance self-shielding are required. By default, all the probabilities are recalculated even if only one isotopic mixture is modified.

`HALT` keyword to specify that the program is to be stopped at the end of the geometry calculations. This option permits the geometry inputs to be checked, the number of blocks and interface currents to be calculated, and a conservative estimate of the memory required for storing the tracks to be made for mixed geometries. This keyword is not used for cluster type geometries.

`IP00` keyword to specify that an isotropic angular flux between each block is used. (default option for the `TUBEZ` geometries).

`SP01` keyword to specify that a linearly anisotropic angular flux between each block is used in combination with linearly anisotropic boundary conditions (default option for all geometries except `TUBE`, `SPHERE` and `TUBEZ`).

`IP01` keyword to specify that a linearly anisotropic angular flux between each block is used in combination with isotropic boundary conditions (default option for the `TUBE` and `SPHERE` geometries).

`QUA1` keyword to specify the one dimensional integration parameters.

iqua1 number of basis points for the angular integration of the blocks in a one dimensional geometry. This parameter is not used for `CAR1D` geometries. If a Gauss–Legendre or Gauss–Jacobi quadrature is used, the values of *iqua1* allowed are: 1 to 20, 24, 28, 32 or 64. The default value is *iqua1*=5.

`QUA2` keyword to specify the two dimensional integration parameters.

iqua2 number of basis points for the angular integration of the blocks in a two dimensional geometry appearing during assembly and cluster calculations. If a Gauss–Legendre or Gauss–Jacobi formula is used the values allowed for *iqua2* are: 1 to 20, 24, 28, 32 or 64. The default value is *iqua2*=6 and represents the number of angles in $(0, \pi/4)$ for cartesian geometries and $(0, \pi/6)$ for hexagonal geometries.

nsegment number of basis points for the spatial integration of the blocks in a two dimensional geometry appearing during assembly and cluster calculations. The default value is *nsegment*=3.

`EQW` keyword to specify the use of equal weight quadrature.

`GAUS` keyword to specify the use of the Gauss–Legendre or the Gauss–Jacobi quadrature. This is the default option.

`ROTH` keyword to specify that the isotropic (DP_0) components of the current at cell interface is used with the incoming current being averaged over all the faces surrounding a cell. The global collision matrix is calculated in a annular model. Only used when 2–D assembly of cells are considered.

`ROT+` keyword to specify that the isotropic (DP_0) components of the current at cell interface is used. The global collision matrix is calculated in a annular model. Only used when 2–D assembly of cells are considered.

`DP00` keyword to specify that the isotropic (DP_0) components of the current at cell interface is used. The global collision matrix are computed explicitly. Only used when 2–D assembly of cells are considered.

`DP01` keyword to specify that the linearly anisotropic (DP_1) components of the current at cell interface are used. This hypothesis implies 12 currents per cell in a cartesian geometry and 18 currents per cell for a hexagonal geometry. Linearly anisotropic reflection is used. Only used when 2–D assembly of cells are considered.

`ASKE` keyword to specify the use of an *Askew* cylinderization which preserves both the external surface of the cells and the material balance of the external crown (by a modification of its concentration). By default a *Wigner* cylinderization is used which preserves the volume of the external crown. Note, that an assembly of a number of rectangular cells having unequal volumes requires an *Askew* cylinderization. This applies only in cases where the external surface is annular using the `ROTH` or `ROT+` options. Only used when 2–D assembly of cells are considered.

`LIGN` keyword to specify that all the integration lines are to be printed. This option should only be used when absolutely necessary because it generates a rather large amount of output. Only used when 2–D assembly of cells are considered.

`RECT` keyword to specify that square cells are to be treated as if they were rectangular cells, with the inherent loss in performance that this entails. This option is of purely academic interest.

`RECD` keyword to specify the use of the direct reconstruction method for the collision probabilities (method with refraction effects). Only used when cluster geometries are considered.

`RECR` keyword to specify the use of the surface fractioning reconstruction method for the collision probabilities (no refraction effect but twice the number of interfaces). This is the default option. Only used when cluster geometries are considered.

`BP00` keyword to specify the use of an isotropic angular flux (DP_0) approximation between the micro volumes making up the micro structures in a case involving the treatment of double heterogeneity.

`BP01` keyword to specify the use of a linearly anisotropic angular flux (DP_1) approximation between the micro volumes making up the micro structures in a case involving the treatment of double heterogeneity. In all cases, an approximation of isotropic angular flux is used on the interface between the micro structures and the macro volumes. This is the default option.

`QUAB` keyword to specify the initial number of basis point for the numerical integration of each micro structure in cases involving double heterogeneity.

iquab the number of basis point for the numerical integration of the collision probabilities in the micro volumes using the Gauss–Jacobi formula. The values permitted are: 1 to 20, 24, 28, 32 or 64. The default value is *iquab*=5.

3.4.5 The `BIVACT:` specific tracking data

Note that this tracking option can be used only indirectly through the `SPH` homogenization option (see Section 3.9.1):

Table 28: Structure **(descbivac)**

[{ `PRIM` | `DUAL` } [*ielem icol* [*isplh*]]]

where

`PRIM` keyword to set a primal finite element (classical) discretization.

`DUAL` keyword to set a mixed-dual finite element discretization.

ielem order of the finite element representation. The values permitted are 1 (linear polynomials), 2 (parabolic polynomials), 3 (cubic polynomials) or 4 (quartic polynomials). By default *ielem*=1. Discretization of a hexagonal geometry is only available with *ielem*=1.

icol type of quadrature used to integrate the mass matrices. The values permitted are 1 (analytical integration), 2 (Gauss–Lobatto quadrature) or 3 (Gauss–Legendre quadrature). By default *icol*=2. The analytical integration corresponds to classical finite elements; the Gauss–Lobatto quadrature corresponds to a variational or nodal type collocation and the Gauss–Legendre quadrature corresponds to superconvergent finite elements.

isplh type of hexagonal mesh splitting. This data is given only if the geometry is 2–D hexagonal. The values permitted are 1 (full hexagon), 2 for splitting each hexagon into 6 triangles, 3 for splitting each hexagon into 24 triangles, 5 for splitting each hexagon into 96 triangles, 9 for splitting each hexagon into 384 triangles and 17 for splitting each hexagon into 1536 triangles.

Various finite element approximations can be obtained by combining different values of *ielem* and *icol*:

- `PRIM 1 1` : Linear finite elements;
- `PRIM 1 2` : Mesh corner finite differences;
- `PRIM 1 3` : Linear superconvergent finite elements;
- `PRIM 2 1` : Quadratic finite elements;
- `PRIM 2 2` : Quadratic variational collocation method;
- `PRIM 2 3` : Quadratic superconvergent finite elements;
- `PRIM 3 1` : Cubic finite elements;
- `PRIM 3 2` : Cubic variational collocation method;
- `PRIM 3 3` : Cubic superconvergent finite elements;
- `PRIM 4 2` : Quartic variational collocation method;
- `DUAL 1 1` : Mixed-dual linear finite elements;
- `DUAL 1 2` : Mesh centered finite differences;
- `DUAL 1 3` : Mixed-dual linear superconvergent finite elements (numerically equivalent to `PRIM 1 3`);
- `DUAL 2 1` : Mixed-dual quadratic finite elements;
- `DUAL 2 2` : Quadratic nodal collocation method;
- `DUAL 2 3` : Mixed-dual quadratic superconvergent finite elements (numerically equivalent to `PRIM 2 3`);
- `DUAL 3 1` : Mixed-dual cubic finite elements;
- `DUAL 3 2` : Cubic nodal collocation method;

- `DUAL 3 3 :` Mixed-dual cubic superconvergent finite elements
  (numerically equivalent to `PRIM 3 3`);
- `DUAL 4 2 :` Quartic nodal collocation method;

3.5 The `SHI:` module

The self-shielding module in DRAGON, called SHIBA[40], allows the energy dependent dilution parameter (microscopic dilution cross section) associated with each resonant isotope, identified as such by the *inrs* parameter defined in Section 3.2, to be recalculated. The general format of the data for this module is:

Table 29: Structure **(SHI:)**

MICLIB `:=` `SHI:` { *MICLIB* | *OLDLIB* } *TRKNAM* [*TRKFIL*] `::` **(descshi)**

where

MICLIB `character*12` name of the MICROLIB that will contain the microscopic and macroscopic cross sections updated by the self-shielding module. If *MICLIB* appears on both LHS and RHS, it is updated; otherwise, the internal library *OLDLIB* is copied into *MICLIB* and *MICLIB* is updated.

OLDLIB `character*12` name of a read-only MICROLIB that is copied into *MICLIB*.

TRKNAM `character*12` name of the required TRACKING data structure.

TRKFIL `character*12` name of the sequential binary tracking file used to store the tracks lengths. This file is given if and only if it was required in the previous tracking module call (see Section 3.4).

(descshi) structure describing the self-shielding options.

Each time the `SHI:` module is called, a sub-directory is updated in the MICROLIB data structure to hold the last values defined in the **(descshi)** structure. The next time this module is called, these values will be used as floating defaults.

3.5.1 Data input for module `SHI:`

Table 30: Structure **(descshi)**

[`EDIT` *iprint*]
[`GRMIN` *lgrmin*] [`GRMAX` *lgrmax*]
[`MXIT` *imxit*] [`EPS` *valeps*]
[{ `LJ` | `NOLJ` }] [{ `GC` | `NOGC` }] [`NOTR`] [`PIJ`]

where

EDIT	keyword used to modify the print level *iprint*.
iprint	index used to control the printing of this module. The amount of output produced by this tracking module will vary substantially depending on the print level specified.
GRMIN	keyword to specify the minimum group number considered during the self-shielding process.
lgrmin	first group number considered during the self-shielding process. By default, *lgrmin* is set to 1 for all the libraries except for those in the WIMSAECL and WIMSD4 format where this information is take explicitly from these libraries.
GRMAX	keyword to specify the maximum group number considered during the self-shielding process.
lgrmax	last group number considered during the self-shielding process. By default, *lgrmax* is set to the group closer to 4.0 eV in the library.
MXIT	keyword to specify the maximum number of iterations during the self-shielding process.
imxit	the maximum number of iterations. The default is *imxit*=20.
EPS	keyword to specify the convergence criterion for the self-shielding iteration.
valeps	the convergence criterion for the self-shielding iteration. By default, *valeps* is set to 1.0×10^{-4}.
LJ	keyword to activate the Livolant and Jeanpierre normalization scheme which modifies the self-shielded averaged neutron flux in heterogeneous geometries. By default the Livolant and Jeanpierre normalization scheme is not activated.
NOLJ	keyword to deactivate the Livolant and Jeanpierre normalization scheme which modifies the self-shielded averaged neutron flux in heterogeneous geometries. This is the default option.
GC	keyword to activate the Goldstein–Cohen approximation in cases where Goldstein–Cohen parameters are stored on the internal library. These parameters are not available if the resonant isotopes are interpolated from an APLIB1 or MATXS type library. This is the default option.
NOGC	keyword to deactivate the Goldstein–Cohen approximation in cases where Goldstein–Cohen parameters are stored on the internal library. These parameters are not available if the resonant isotopes are interpolated from an APLIB1 or MATXS type library. By default the Goldstein–Cohen approximation is used when possible.
NOTR	keyword to deactivate the transport correction option for self-shielding calculations (see CTRA in Sections 3.1 and 3.2).
PIJ	keyword to specify the use of complete collision probabilities for the JPMT: calculation option. By default, a fast reconstruction algorithm based on sparse matrix algebra is used.

3.6 The assembly modules

We will now describe the assembly modules which can be used to prepare the group dependent complete collision probability or the assembly matrices required by the flux solution module of DRAGON. There are two assembly modules: ASM: and EXCELL:. The assembly module ASM: is generally called after a tracking module; it recovers tracking lengths and material numbers from the sequential tracking file and then computes the collision probability matrices under various normalization. The EXCELL: module can also be used to perform the work of both the EXCELT: and the ASM: modules for computing collision probabilities in 3–D geometries. This last module has been programmed to enhance the capability and performance of collision probability calculations for 3–D applications. The EXCELL: module does not keep the tracking file, the file is rebuilt at the same time as

collision probabilities are computed. This feature enables users to do fine mesh calculations in 3–D domains without any limitation on the size of the tracking file. The calling specifications are:

Table 31: Structure **(ASM:)**

PIJNAM `:=` `ASM:` [*PIJNAM*] *LIBNAM TRKNAM* [*TRKFIL*] `::` **(descasm)**

and

Table 32: Structure **(EXCELL:)**

PIJNAM TRKNAM `:=` `EXCELL:` *GEONAM LIBNAM* `::` **(desctrack) (descXL)**

where

PIJNAM `character*12` name of ASMPIJ data structure containing the system matrices. If *PIJNAM* appears on the RHS, the **(descasm)** information previously stored in *PIJNAM* is kept.

LIBNAM `character*12` name of the MACROLIB or MICROLIB data structure that contains the macroscopic cross sections (see Sections 3.1 and 3.2).

TRKNAM `character*12` name of the TRACKING data structure containing the tracking (see Section 3.4).

TRKFIL `character*12` name of the sequential binary tracking file used to store the tracks lengths. This file is given if and only if it was required in the previous tracking module call (see Section 3.4).

GEONAM `character*12` name of the GEOMETRY data structure for the `EXCELL:` module (see Section 3.3).

(descasm) structure containing the input data to this module (see Section 3.6.1).

(desctrack) structure containing the general tracking data to the `EXCELL:` module (see Section 3.4.1).

(descXL) structure containing the input data to the `EXCELL:` module (see Section 3.6.2).

3.6.1 Data input for module `ASM:`

Table 33: Structure **(descasm)**

[`EDIT` *iprint*]

continued on next page

Structure **(descasm)** continued from last page

```
[ { ARM [ NOR2 ] |
{ PIJ | PIJK } [ SKIP ] [ [ NORM ] ALBS ] [ NAME NMPIJ ] ]
[ PNOR { NONE | DIAG | GELB | HELI | NONL } ]
```

where

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing of this module. The amount of output produced by this tracking module will vary substantially depending on the print level specified.

`ARM` keyword to specify that an assembly calculation is carried out without building the full collision probability matrices. This option can only be used for a geometry tracked using the `JPMT:` module. By default, the `PIJ` option is used.

`NOR2` keyword to specify that the matrix required for residual calculation is not required. This is active only when the `JPMT:` tracking module is called. In order to use this option the variational acceleration technique in module `FLU:` must be deactivated (see Section 3.7).

`PIJ` keyword to specify that the standard collision probabilities must be computed. This is the default option.

`PIJK` keyword to specify that both the directional and standard collision probabilities must be computed. Moreover, the additional directional collision probability matrix can only be used if *nanis* is set to 2 in Section 3.4 and `HETE` is activated in Section 3.7. Finally, the PIJK option is only available for 2–D geometries analyzed with the module `EXCELT:`. By default, the `PIJ` option is used.

`SKIP` keyword to specify that only the complete collision probability matrix p_{ij}^g is to be computed. In general, the scattering modified collision probability matrix $p_{s,ij}^g$ is also computed using:

$$p_{s,ij}^g = \left[I - p_{ij}^g \Sigma_{s0}^{g\to g}\right]^{-1} p_{ij}^g$$

where $\Sigma_{s0}^{g\to g}$ is the within group isotropic scattering cross section. When available, $p_{s,ij}^g$ is used in the flux solution module in such a way that for the groups where there is no upscattering, the thermal iteration is automatically deactivated. In the case where the `SKIP` option is activated, the p_{ij}^g matrix is used and thermal iterations are required in every energy group.

`NORM` keyword to specify that the collision probability matrix is to be normalized in such a way as to eliminate all neutron loss (even if the region under consideration has external albedo boundary conditions which should result in neutron loss). When used with a void boundary condition (zero reentrant current), this option is equivalent to imposing *a posteriori* a uniform reentrant current.

`ALBS` keyword to specify that a consistent Selengut normalization of the collision probability matrix is to be used both for the flux solution module (see Section 3.7) and in the equivalence calculation (see Section 3.9). This keyword results in storing the escape probabilities P_{iS} in *PIJNAM*. For all the cases where this option is used, it is necessary to define a geometry with `VOID` external boundary conditions (see Section 3.3).

`NAME` keyword to specify that the complete collision probability matrices are to be computed even if they are not required in the flux solution module (keyword `PIJ` or `SKIP` absent) and stored under a specific name on *PIJNAM*.

NMPIJ name under which the complete collision probability matrices are saved.

PNOR keyword to specify that the collision, leakage and escape probability matrices are to be normalized in such a way as to satisfy explicitly the neutron conservation laws. This option compensates for the errors which will arise in the numerical evaluation of these probabilities and may result in non conservative collision probability matrices. The default option is now HELI while it was formerly GELB (Release 3.03).

NONE keyword to specify that the probability matrices are not to be renormalized.

DIAG keyword to specify that only the diagonal element of the probability matrices will be modified in order to insure the validity of the conservation laws.

GELB keyword to specify that the Gelbard algorithm will be used to normalize the collision probability matrices.[41]

HELI keyword to specify that the Helios algorithm will be used to normalize the collision probability matrices.[42]

NONL keyword to specify that a non linear multiplicative algorithm will be used to normalize the collision probability matrices.[41]

3.6.2 Data input for module EXCELL:

Table 34: Structure **(descXL)**

```
TRAK [ SUBG nsubg ]
[ PNOR { NONE | DIAG | GELB | HELI | NONL } ]
[ [ NORM ] ALBS ] [ SKIP ]
TISO nangl dens [ CORN pcorn ] [ SYMM isymm ]
```

where

SUBG keyword to specify the number of groups in each subgroup for collision probability calculations.

nsubg number of groups in each subgroup in collision probability calculations. The default value is the total number of groups contained in the *LIBNAM* object. However, in applications needing a large amount of memory to store group=dependent collision probability, this number can be smaller (the minimal value is indeed 1). In all cases, the tracking file is rebuilt for every subgroup, and the collision probability matrices are computed by block of *nsubg* groups until all groups are processed.

All other keywords and values have already been defined in the previous section (see Section 3.6.1) or in the EXCELT: section (see Section 3.4.2). Note that the EXCELL: module is limited to 3–D geometries.

3.7 The FLU: module

The FLU: module is used to solve the linear system of multigroup collision probability or response matrix equations in DRAGON. The calling specifications are:

Table 35: Structure **(FLU:)**

FLUNAM `:=` `FLU:` [*FLUNAM*] *PIJNAM LIBNAM TRKNAM* `::` **(descflu)**

where

FLUNAM `character*12` name of the FLUXUNK data structure containing the solution. If *FLUNAM* appears on the RHS, the solution previously stored in *FLUNAM* is used to initialize the new iterative process; otherwise, a uniform unknown vector is used.

PIJNAM `character*12` name of the ASMPIJ data structure containing the group dependent system matrices (see Section 3.6).

LIBNAM `character*12` name of the MACROLIB or MICROLIB data structure that contains the macroscopic cross sections (see Sections 3.1 and 3.2).

TRKNAM `character*12` name of the TRACKING data structure containing the tracking (see Section 3.4).

(descflu) structure containing the input data to this module (see Section 3.7.1).

The TRACKING and the MACROLIB substructure used to build the structure *PIJNAM* are automatically recovered in read-only mode by the generalized driver.

3.7.1 Data input for module `FLU:`

Table 36: Structure **(descflu)**

[`EDIT` *iprint*]
[`INIT` { `OFF` | `ON` ((*fluxes*(i, g), i=1,*nregio*), g=1,*ngroup*) }]
[{ `FLX` | `PAF` | `AF` }]
`TYPE` { `N` | `S` | `K` [**(descleak)**] | { `B` | `L` } **(descleak)** }]
[`THER` [*maxthr*] [*epsthr*]]
[`EXTE` [*maxout*] [*epsout*]]
[`UNKT` [*epsunk*]]
[`REBA` [`OFF`]]
[`ACCE` *nlibre naccel*]

where

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing of this module. The amount of output produced by this tracking module will vary substantially depending on the print level specified.

`INIT` keyword to specify the neutron flux initialization option used.

`OFF` keyword to specify that the initial neutron flux distribution is not to be initialized.

ON	keyword to specify that the initial neutron flux distribution follows.
fluxes	array of average flux per region and per group.
FLX	keyword to specify that a flux solution is to be considered. This is the default option. (Release 3.04)
PAF	keyword to specify that a pseudo-adjoint flux solution is to be considered.[43] (Release 3.04)
AF	keyword to specify that a pseudo-adjoint flux solution is to be considered and that both the pseudo-adjoint and adjoint flux are to be saved on the FLUXUNK data structure. (Release 3.04)
TYPE	keyword to specify the type of flux flux calculation to be performed.
N	keyword to specify that no flux calculation is to be performed. This option is usually activated when one simply wishes to initialize the neutron flux distribution and to store this information in *FLUNAM* (see ON parameter above).
S	keyword to specify that a fixed source problem is to be treated. Such problem can also include fission source contributions.
K	keyword to specify that a fission source eigenvalue problem is to be treated. The eigenvalue is then the effective multiplication factor with a fixed buckling. In this case, the fixed source, if any is present on the MACROLIB or MICROLIB data structure, is not used.
B	keyword to specify that a fission source eigenvalue problem is to be treated. The eigenvalue in this case is the critical buckling with a fixed effective multiplication factor. The buckling eigenvalue has meaning only in the case of a cell without leakages (see the structure **(descBC)** in Section 3.3.2). It is also possible to use an open geometry with VOID boundary conditions provided it is closed by the ASM: module (see Section 3.6.1) using the keywords NORM or ALSB.
L	keyword to specify that a non multiplicative medium eigenvalue problem is to be treated. The eigenvalue in this case is the critical buckling with vanishing fission cross sections. The buckling eigenvalue has meaning only in the case of a cell without leakages (see the structure **(descBC)** in Section 3.3.2). It is also possible to use an open geometry with VOID boundary conditions provided it is closed by the ASM: module (see Section 3.6.1) using the keywords NORM or ALSB.
(descleak)	structure describing the general leakage parameters options (see Section 3.7.2).
THER	keyword to specify that the control parameters for the thermal iterations are to be modified.
maxthr	maximum number of thermal iterations. The fixed default value is 2×*ngroup*-1 (using scattering modified CP) or 4×*ngroup*-1 (using standard CP).
epsthr	convergence criterion for the thermal iterations. The fixed default value is 5.0×10^{-5}.
EXTE	keyword to specify that the control parameters for the external iteration are to be modified.
maxout	maximum number of external iterations. The fixed default value for a case with no leakage model is $2 \times n_f - 1$ where n_f is the number of regions containing fuel. The fixed default value for a case with a leakage model is $10 \times n_f - 1$.
epsout	convergence criterion for the external iterations. The fixed default value is 5.0×10^{-5}.
UNKT	keyword to specify that the flux/current error tolerance in the outer iteration.
epsunk	convergence criterion for flux/current in the outer iteration. The fixed default value is *epsunk*=*epsthr*.

`REBA` keyword used to specify that the flux rebalancing option is to be turned on or off in the thermal iteration. By default (floating default) the flux rebalancing option is initially activated. This keyword is required to toggle between the on and off position of the flux rebalancing option.

`OFF` keyword used to deactivate the flux rebalancing option. When this keyword is absent the flux rebalancing option is reactivated.

`ACCE` keyword used to modify the variational acceleration parameters. This option is active by default (floating default) with *nlibre*=3 free iterations followed by *naccel*=3 accelerated iterations.

nlibre number of free iterations per cycle of *nlibre*+*naccel* iterations.

naccel number of accelerated iterations per cycle of *nlibre*+*naccel* iterations. Variational acceleration may be deactivated by using *naccel*=0. This is required when the `NOR2` is used in the `ASM:` module (see Section 3.6.1).

3.7.2 Leakage model specification structure

The **(descleak)** structure allows the following information to be specified:

Table 37: Structure **(descleak)**

```
{ LKRD | P0 | P1 | B0 | B1 | B0TR }
{ SIGS | PNL | ALBS | HETE [ { G | R | Z | X | Y } ] }
[ { BUCK { valb2 | [ G valb2 ] [ R valbr2 ] [ Z valbz2 ] [ X valbx2 ] [ Y valby2 ] } |
   KEFF valk | IDEM [ { B2 | DB2 } ] } ]
```

`LKRD` keyword used to specify that the leakage coefficients are recovered from data structure named *FLUNAM*.

`P0` keyword used to specify that the leakage coefficients are calculated using a P_0 model.

`P1` keyword used to specify that the leakage coefficients are calculated using a P_1 model.

`B0` keyword used to specify that the leakage coefficients are calculated using a B_0 model. This is the default value when a buckling calculation is required (`B`).

`B1` keyword used to specify that the leakage coefficients are calculated using a B_1 model.

`B0TR` keyword used to specify that the leakage coefficients are calculated using a B_0 model with transport correction.

`SIGS` keyword used to specify that an homogeneous buckling correction is to be applied on the diffusion cross section ($\Sigma_s - dB^2$).

`PNL` keyword used to specify that the elements of the collision probability (`SKIP`) or the scattering modified collision probability matrix are multiplied by the adequate non leakage homogeneous buckling dependent factor.[44] .This is the default option when a buckling calculation is required (`B`) or a fission source eigenvalue problem (`K`) with imposed buckling is considered.

ALBS keyword used to specify that an homogeneous buckling contribution is introduced by a group dependent correction of the albedo.[45] It is then necessary to define the geometry with an external boundary condition of type VOID (see Section 3.3.2) and to close the region in module ASM: using the ALBS option (see Section 3.6.1).

HETE keyword used to specify that the leakage and anisotropic effects will be taken into account using a consistent P_n[46] or a B_n[47,48] model. The heterogeneous buckling contribution is introduced in the B_n model using the PIJK method. It is activated only if ANIS 2 is specified in module EXCELT: (see Section 3.4.2) and the option PIJK is used in module ASM: (see Section 3.6.1). Otherwise, a consistent P_n model is used.

G keyword used to specify that the buckling search will assume all directional buckling to be identical (floating default option).

R keyword used to specify that a radial buckling search will be considered assuming an imposed Z direction buckling.

Z keyword used to specify that a Z direction buckling search will be considered assuming an imposed X direction and Y direction buckling.

X keyword used to specify that a X direction buckling search will be considered assuming an imposed Y direction and Z direction buckling.

Y keyword used to specify that a Y direction buckling search will be considered assuming an imposed X direction and Z direction buckling.

BUCK keyword used to specify the initial (for a buckling eigenvalue problem) or fixed (for a effective multiplication constant eigenvalue problem) buckling.

G keyword used to specify that the buckling in the X direction, Y direction and Z direction are to be initialized to *valb2*/3 (floating default).

R keyword used to specify that the buckling in the X direction, and Y direction are to be initialized to *valbr2*/2.

Z keyword used to specify that the buckling in the Z direction, is to be initialized to *valbz2*.

X keyword used to specify that the buckling in the X direction, is to be initialized to *valbx2*.

Y keyword used to specify that the buckling in the Y direction, is to be initialized to *valby2*.

valb2 value of the fixed or initial total buckling in cm^{-2}. The floating default value is

$$valb2 = valbx2 + valby2 + valbz2.$$

valbr2 value of the fixed or initial radial buckling in cm^{-2}. The floating default value is

$$valb2 = valbx2 + valby2.$$

valbz2 value of the fixed or initial Z direction buckling in cm^{-2}. By default *valbz2*=0.0 cm^{-2}. If *valb2* is specified then *valbz2*=*valb2*/3.

valbx2 value of the fixed or initial Z direction buckling in cm^{-2}. By default *valbx2*=0.0 cm^{-2}. If *valb2* is specified then *valbx2*=*valb2*/3. If *valbr2* is specified then *valbx2*=*valbr2*/2.

valby2 value of the fixed or initial Z direction buckling in cm^{-2}. By default *valby2*=0.0 cm^{-2}. If *valb2* is specified then *valby2*=*valb2*/3. If *valbr2* is specified then *valby2*=*valbr2*/2.

`KEFF` keyword used to specify the fixed (for a buckling eigenvalue problem) effective multiplication constant.

valk value of the fixed effective multiplication constant. The fixed default value is *valk*=1.0.

`IDEM` keyword used to specify that the initial (for a buckling eigenvalue problem) or fixed (for a effective multiplication constant eigenvalue problem) leakage is to be read from the data structure *FLUNAM*.

`B2` keyword used to specify that only the buckling is to be read from the data structure *FLUNAM*. This is the default value. (Release 3.04)

`DB2` keyword used to specify that the initial buckling and diffusion coefficients are to be read from the data structure *FLUNAM*. (Release 3.04)

3.8 The `MOCC:` module

(Release 3.04)

The `MOCC:` module can be used to solve the transport equation using the method of cyclic characteristics in DRAGON.[31,32] The calling specifications are:

Table 38: Structure **(MOCC:)**

FLUNAM `:=` `MOCC:` [*FLUNAM*] *LIBNAM TRKNAM TRKFIL* `::` **(descmoc)**

where

FLUNAM `character*12` name of the FLUXUNK data structure containing the solution. If *FLUNAM* appears on the RHS, the solution previously stored in *FLUNAM* is used to initialize the new iterative process; otherwise, a uniform unknown vector is used.

LIBNAM `character*12` name of the MACROLIB or MICROLIB data structure that contains the macroscopic cross sections (see Sections 3.1 and 3.2).

TRKNAM `character*12` name of the TRACKING data structure containing the tracking (see Section 3.4).

TRKFIL `character*12` name of the sequential binary tracking file used to store the tracks lengths. This file is given if and only if it was required in the previous tracking module call (see Section 3.4).

(descmoc) structure containing the input data to this module (see Section 3.8.1).

3.8.1 Data input for module `MOCC:`

Table 39: Structure **(descmoc)**

```
[ EDIT iprint ]
[ EXAC ]
[ NBPN nl ]
[ { LCMD | GAUS | CACA | CACB } ]
TYPE { N | S | K [ (descleakmoc) ] | B [ (descleakmoc) ] }
[ THER [ maxthr ] [ epsthr ] ]
[ EXTE [ maxout ] [ epsout ] ]
[ UNKT [ epsunk ] ]
[ NOBA ]
[ ACCE nlibre naccel ]
```

where

EDIT	keyword used to modify the print level *iprint*.
iprint	index used to control the printing of this module. The amount of output produced by this tracking module will vary substantially depending on the print level specified.
EXAC	keyword used to specify that exact exponential functions are to be used for neutron path attenuation. By default, approximate values for the exponential function derived from second order local polynomials are considered.
NBPN	keyword used to specify the expansion order in Legendre polynomial for the flux used in the calculation.
nl	the expansion order in Legendre polynomial for the flux used in the calculation. By default *nl*=0.
LCMD	keyword to specify that optimized polar integration angles are to be selected.[49] This is the default option.
GAUS	keyword to specify that Gauss polar integration angles are to be selected.
CACA	keyword to specify that CACTUS type equal weight polar integration angles are to be selected.[50]
CACB	keyword to specify that CACTUS type uniformly distributed integration polar angles are to be selected.[50]
TYPE	keyword to specify the type of flux flux calculation to be performed.
N	keyword to specify that no flux calculation is to be performed. This option is usually activated when one simply wishes to initialize the neutron flux distribution and to store this information in *FLUNAM* (see ON parameter above).
S	keyword to specify that a fixed source problem is to be treated. Such problem can also include fission source contributions.
K	keyword to specify that a fission source eigenvalue problem is to be treated. The eigenvalue is then the effective multiplication factor with a fixed buckling. In this case, the fixed source, if any is present on the MACROLIB or MICROLIB data structure, is not used.

B	keyword to specify that a fission source eigenvalue problem is to be treated. The eigenvalue in this case is the critical buckling with a fixed effective multiplication factor. The buckling eigenvalue has meaning only in the case of a cell without leakages (see the structure **(descBC)** in Section 3.3.2). It is also possible to use an open geometry with VOID boundary conditions provided it is closed by the ASM: module (see Section 3.6.1) using the keywords NORM or ALSB.
(descleakmoc)	structure describing the general leakage parameters options (see Section 3.8.2).
THER	keyword to specify that the control parameters for the thermal iterations are to be modified.
maxthr	maximum number of thermal iterations. The fixed default value is 2×*ngroup*-1 (using scattering modified CP) or 4×*ngroup*-1 (using standard CP).
epsthr	convergence criterion for the thermal iterations. The fixed default value is 5.0×10^{-5}.
EXTE	keyword to specify that the control parameters for the external iteration are to be modified.
maxout	maximum number of external iterations. The fixed default value for a case with no leakage model is $2 \times n_f - 1$ where n_f is the number of regions containing fuel. The fixed default value for a case with a leakage model is $10 \times n_f - 1$.
epsout	convergence criterion for the external iterations. The fixed default value is 5.0×10^{-5}.
UNKT	keyword to specify that the flux/current error tolerance in the outer iteration.
epsunk	convergence criterion for flux/current in the outer iteration. The fixed default value is *epsunk*=*epsthr*.
NOBA	keyword used to specify that the flux rebalancing option is to be turned on or off in the thermal iteration. By default (floating default) the flux rebalancing option is initially activated.
ACCE	keyword used to modify the variational acceleration parameters. This option is active by default (floating default) with *nlibre*=3 free iterations followed by *naccel*=3 accelerated iterations.
nlibre	number of free iterations per cycle of *nlibre*+*naccel* iterations.
naccel	number of accelerated iterations per cycle of *nlibre*+*naccel* iterations. Variational acceleration may be deactivated by using *naccel*=0. This is required when the NOR2 is used in the ASM: module (see Section 3.6.1).

3.8.2 Leakage model specification structure for the method of characteristics

The **(descleakmoc)** structure allows the following information to be specified:

Table 40: Structure **(descleakmoc)**

{P0 | P1 | B0 | B1 | B0TR }
{ SIGS | PNL }
[{ BUCK *valb2* | KEFF *valk* }] }]

`P0` keyword used to specify that the leakage coefficients are calculated using a P_0 model.

`P1` keyword used to specify that the leakage coefficients are calculated using a P_1 model.

`B0` keyword used to specify that the leakage coefficients are calculated using a B_0 model. This is the default value when a buckling calculation is required (`B`).

`B1` keyword used to specify that the leakage coefficients are calculated using a B_1 model.

`B0TR` keyword used to specify that the leakage coefficients are calculated using a B_0 model with transport correction.

`SIGS` keyword used to specify that an homogeneous buckling correction is to be applied on the diffusion cross section ($\Sigma_s - dB^2$).

`PNL` keyword used to specify that the elements of the collision probability (`SKIP`) or the scattering modified collision probability matrix are multiplied by the adequate non leakage homogeneous buckling dependent factor.[44] .This is the default option when a buckling calculation is required (`B`) or a fission source eigenvalue problem (`K`) with imposed buckling is considered.

`BUCK` keyword used to specify the initial (for a buckling eigenvalue problem) or fixed (for a effective multiplication constant eigenvalue problem) buckling.

valb2 value of the fixed or initial total buckling in cm^{-2}. The floating default value is

$$valb2 = valbx2 + valby2 + valbz2.$$

`KEFF` keyword used to specify the fixed (for a buckling eigenvalue problem) effective multiplication constant.

valk value of the fixed effective multiplication constant. The fixed default value is *valk*=1.0.

3.9 The `EDI:` module

The `EDI:` module supplies the main editing options to DRAGON. It can be use to compute the reaction rates, average and condensed cross sections and to store this information on a file for further use. The calling specifications are:

Table 41: Structure **(EDI:)**

EDINAM `:=` `EDI:` [*EDINAM*] *FLUNAM LIBNAM TRKNAM* [*REFGEO REFPIJ* [*SPHGEO*]] `::` **(descedi)**

where

EDINAM `character*12` name of the EDITION data structure where the edition results will be stored.

FLUNAM `character*12` name of the FLUXUNK data structure containing a transport solution (see Section 3.7).

LIBNAM `character*12` name of the MACROLIB or MICROLIB data structure that contains the macroscopic cross sections (see Sections 3.1 and 3.2).

TRKNAM `character*12` name of the TRACKING data structure containing the tracking (see Section 3.4).

SPHGEO `character*12` name of the GEOMETRY data structure that can be used in the SPH equivalence procedure. In some cases the module

REFGEO `character*12` optional name of the GEOMETRY data structure that was used for the original flux calculation (see Section 3.3). Compulsory for calculation involving SPH homogenization.

REFPIJ `character*12` optional name of the ASMPIJ data structure that was used for the original flux calculation (see Section 3.6). Compulsory for calculation involving SPH homogenization.

SPHGEO `character*12` optional name of the GEOMETRY data structure that can be used in the SPH equivalence procedure. In some cases the module `EDI:` can automatically build a macro geometry, however it is always preferable to specify explicitly the macro geometry to be used.

(descedi) structure containing the input data to this module (see Section 3.9.1).

The GEOMETRY, TRACKING, MACROLIB and MICROLIB data structure used to build the structure *FLUNAM* are automatically recovered in read-only mode by the generalized driver. If a Selengut normalization is requested, the ASMPIJ data structure is also recovered.

3.9.1 Data input for module `EDI:`

Table 42: Structure **(descedi)**

[`EDIT` *iprint*]
[`P1SCAT` { `FLUX` | `CURRENT` | `COHERENT` | `DIRECTION` }]
[`UPS`]
[`MERG` { `COMP` | `NONE` |
`REGI` (*iregm*(i), $i = 1, N_r$) |
`MIX` (*imixm*(i), $i = 1, N_m$) }]
[`TAKE` {
`REGI` (*iregt*(i), $i = 1, N_r$) |
`MIX` (*imixt*(i), $i = 1, N_m$) }]
[{ `P0W` | `P1W` }]
[`COND` { `NONE` | ({ *icond*(g) | *energy*(g) }, $g = 1, N_g$)] }
[`MICR` [`ISOTXS`] { `ALL` | `NONE` | *nis* (*HISO*(i), i=1,*nis*)]
[`FLIB` [`ISOTXS`] { `ALL` | `NONE` }]
[`ACTI` [`ISOTXS`] { `NONE` | (*imixa*(i), $i = 1, N_m$)] }
[`SAVE` [`ON` { *DIRN* | *idirn* }]]
[`PERT`]
[`STAT` { `ALL` | `RATE` | `FLUX` | `DELS` } [`REFE` { *DIRO* | *idiro* }]]
[`NBAL`]
[`SPH` **(descsph)**]

where

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing of this module. The *iprint* parameter is important for adjusting the amount of data that is printed by this calculation step:

- *iprint*=0 results in no output;
- *iprint*=1 results in the average and integrated flux being printed (floating default);
- *iprint*=2 results in the reaction rates being printed;
- *iprint*=3 is identical to the previous option, but the condensed and/or homogenized vectorial cross sections are also printed;
- *iprint*=4 is identical to the previous option, but the condensed and/or homogenized transfer cross sections are also printed.

`P1SCAT` keyword to specify the type of homogenization to be considered for the anisotropic component of the scattering cross section (Release 3.04).

`FLUX` keyword to specify the flux/volume homogenization for the anisotropic component of the scattering cross section. This is the default option technique used when no leakage model is used (Release 3.04).

`CURRENT` keyword to specify the current/volume homogenization for the anisotropic component of the scattering cross section. This is the default option technique used when an homogeneous leakage model is used (Release 3.04).

`COHERENT` keyword to specify a coherent directional averaged current/volume homogenization for the anisotropic component of the scattering cross section (Release 3.04)..[51]

`DIRECTION` keyword to specify a coherent directional current/volume homogenization for the anisotropic component of the scattering cross section.This is the default technique used when an heterogeneous leakage model is used (Release 3.04).[52]

`UPS` keyword to specify that the reaction rates and the condensed and/or homogenized cross sections are corrected so as to eliminate upscattering. This option is useful for reactor analysis codes which cannot take into account such cross sections.

`NONE` keyword to deactivate the effect of previous use of a the preceding keyword.

`MERG` keyword to specify that the neutron flux is to be homogenized over specified regions or mixtures.

`REGI` keyword to specify that the homogenization of the neutron flux will take place over the following regions. Here $N_r \leq$*maxreg* with *maxreg* the maximum number of regions for which solutions were obtained.

iregm array of homogenized region numbers to which are associated the old regions. In the editing routines a value of *iregm*=0 allows the corresponding region to be neglected.

`MIX` keyword to specify that the homogenization of the neutron flux will take place over the following mixtures. Here we must have $N_m \leq$*maxmix* where *maxmix* is the maximum number of mixtures in the macroscopic cross section library.

imixm array of homogenized region numbers to which are associated the isotopic mixtures. In the editing routines a value of *imixm*=0 allows the corresponding isotopic mixtures to be neglected. For a mixture in this library which is not used in the geometry one should insert a value of 0 for the new region number associated with this mixture.

`COMP` keyword to specify that the a complete homogenization is to take place.

`TAKE` keyword to specify that the neutron flux is to be edited over specified regions or mixtures.

`REGI` keyword to specify that the editing of the neutron flux will take place over the following regions. Here $N_r \leq$*maxreg* with *maxreg* the maximum number of regions for which solutions were obtained.

iregt regions where the editing will take place. The new region numbers associated with these editing regions are numbered sequentially.

`MIX` keyword to specify that the homogenization of the neutron flux will take place over the following mixtures. Here we must have $N_m \leq$*maxmix* where *maxmix* is the maximum number of mixtures in the macroscopic cross section library.

imixt mixtures where the editing will take place. Each mixture used here must be used in the geometry on is considering.

`P0W` keyword to specify that the P_1 information is to be homogenized and condensed using the scalar flux. This is the default option.

`P1W` keyword to specify that the P_1 information is to be homogenized and condensed using a current recovered from a consistent P_1 or from a consistent heterogeneous B_1 model.

`COND` keyword to specify that a group condensation of the flux is to be performed.

icond array of increasing energy group limits that will be associated with each of the N_g condensed groups. The final value of *icond* will automatically be set to *ngroup* while *icond*$>$*ngroup* will be dropped from the condensation. We must have $N_g \leq$*ngroup*.

energy array of decreasing energy limits (in eV) that will be associated with each of the N_g condensed groups. We must have $N_g \leq$*ngroup+1*. Note that if an energy limit is located between two energy groups, the condensation group will include this associated energy group. In the case where two energy limits fall within the same energy group the lowest energy will be dropped. Finally the maximum and minimum energy limits can be skipped since they will be taken automatically from the information available in the library.

`MICR` keyword to specify that the condensation and homogenization procedure will be used to associate microscopic cross sections to the isotopes present in the homogenized regions. The macroscopic cross sections and the diffusion coefficients are weighted by the multigroup flux appearing in the regions where the isotopes are present. The resulting nuclear properties are saved on *EDINAM* when the `SAVE` keyword is present.

`FLIB` keyword similar `MICR` except that the burnup chain are also saved on *EDINAM* when the `SAVE` keyword is present. In addition one fission spectrum per fissile isotope is generated rather than the unique fission spectrum for all fissile isotopes generated when `MICR` is used. The use of this keyword is required if burnup using the condensed and homogenized library is to be considered since in this case the file *EDINAM* contains a MICROLIB.

`ALL` keyword to specify that all the isotopes present in the homogenized region are to be processed.

nis number of isotopes isotopes present in the homogenized region to be processed.

HISO array of `character*8` isotopes names to be processed.

`ACTI` keyword to specify that microscopic activation data will be edited for the isotopes associated with the specified mixture. This information correspond to the microscopic cross section associated with each isotope in a given macro group and macro region assuming a concentration for this isotope of 1.0 cm^{-3} in each region. This keyword is followed by *nacti* material mixture indices, where *nacti*$\leq$*maxmix*.

imixa array of material mixture indices which contains the isotopes for which activation data is to be generated. $nmix \leq maxmix$. Even mixture not used in the geometry can be considered here.

ISOTXS keyword to specify that the set of microscopic cross section generated by the FLIB, MICR and ACTI command will also be saved on a microscopic group neutron cross section library in the ISOTXS–IV format. This will generate a file for each final region specified by the TAKE or MERG keyword, numbered consecutively (IFILE). The name of the file (NISOTXS) is built using the command

```
WRTIE(NISOTXS,'(A6,I6.6)') 'ISOTXS',IFILE
```

SAVE keyword to specify that the flux, the macroscopic and microscopic cross sections and the volumes corresponding to homogenized regions are to be saved on *EDINAM*. In the case where the FLIB option is activated, the saved information is in the form of a MICROLIB, otherwise a MACROLIB is store on a subdirectory of EDITION.

ON keyword to specify on which directory of *EDINAM* this information is to be stored.

DIRN name of the directory on which the above information is to be stored.

idirn number associated with a directory of *EDINAM* on which the above information is to be stored. To each number *idirn* is associated a directory name CDIRN='REF-CASE'//CN where CN is a character*4 variable defined by WRITE(CN,'(I4)') idirn.

PERT keyword to specify that first order perturbations for the microscopic cross sections are to be saved on *EDINAM*.

STAT keyword to specify that a comparison between the current and a reference set of reaction rates and/or integrated flux is to be performed.

ALL keyword to specify that the relative differences in the reaction rates and the integrated flux are to be printed.

RATE keyword to specify that the relative differences in the reaction rates are to be printed.

FLUX keyword to specify that the relative differences in the integrated flux are to be printed.

DELS keyword to specify that the absolute differences in the macroscopic cross section are to be printed.

REFE keyword to specify the directory of *EDINAM* where the reference data requires for the comparison is stored. When this keyword is absent, the last reaction rates and integrated flux saved on *EDINAM* are used.

DIRO name of the directory from which the reference information is taken.

idiro number associated with an directory of *EDINAM* on which the reference information is stored. To each number *idirn* is associated a the directory CDIRN='REF-CASE'//CN where CN is a character*4 variable defined by WRITE(CN,'(I4)') idirn.

NBAL keyword to specify the editing of the four factors computed from a group balance. In this case, the user must specify explicitly a three group condensation.

SPH keyword to specify that an equivalence calculation, between the current micro geometry and a macro geometry to be specified, is to be performed using the SPH technique. The resulting SPH factors are automatically used for the flux and the microscopic and macroscopic cross sections homogenization and condensation.

(descsph) structure used to specify the information required for the SPH calculations (see Section 3.9.2).

3.9.2 Description of the equivalence information

This structure is used to specify the type of equivalence calculation where the flux and the condensed and/or homogenized cross sections are corrected by SPH factors, in such a way as to respect a specified transport-transport or transport-diffusion equivalence criteria.[44, 45, 53] This structure is defined as:

Table 43: Structure **(descsph)**

```
[ SELE ] [ MGEO MACGEO ]
{ OFF | SPRD SPHNAM | HOMO | ALBS |
        ::: EXCELT: (desctrack) (descexcel) |
        ::: SYBILT: (desctrack) (descsybil) |
        ::: JPMT: (desctrack) (descjpm) |
        ::: BIVACT: (desctrack) (descbivac) }
```

where

SELE keyword to specify the use of Selengut normalization. In all cases where this option is used it is necessary to define the geometry with VOID external boundary conditions (see Section 3.3.2) and to close the region for the collision probability calculations using the ALBS option (see Section 3.6.1).

MGEO keyword to specify the macro geometry to be used. In some special cases where 2–D Cartesian assemblies are considered, a macro geometry named SPH$GEO can be automatically constructed by homogenizing the sub-geometries in a geometry. However, for most problems this is not the case and the macro geometry should be specified explicitly.

MACGEO character*12 name of the macro geometry to use. This name should be identical to *SPH-GEO*.

OFF keyword to specify the SPH factors are all set to 1.0, meaning no correction. This is the floating default. This keyword is useful to get rid of a SPH correction which have been set by a previous EDI: call.

SPRD keyword to specify the SPH factors are read on *EDINAM*.

SPHNAM name of the directory from which the SPH factors are to be read (Release 3.04).

HOMO keyword to specify that the SPH factors are calculated assuming the macro geometry is equivalent to a complete homogenization of the current micro geometry. The options MERG COMP must then be specified. In this case the neutron flux (transport or diffusion) will be uniform, which allows the SPH factors to be obtained (one per macro group) using a non iterative strategy. For a given macro group the SPH factor will be equal to the ratio between the average flux of the region and the surface flux if the SELE option is used otherwise the SPH factor are all set equal to 1.0 (no correction). The SELE option allows an SPH factor equal to the inverse of the discontinuity factor to be calculated.

`ALBS` keyword to specify that the albedo of the geometry are to be taken into account in the complete homogenization process. Thus the `MERG` and `COMP` options must be specified. The SPH factors are obtained using a transport-transport equivalence based on a calculation using the collision probabilities. This option requires a geometry with `VOID` (see Section 3.3.2) external boundary conditions to be closed using `ALBS` in modules `ASM:` and `FLU:` (see Sections 3.6.1 and 3.7.1).[45]

`EXCELT:` keyword to specify that the SPH factors are obtained using a transport-transport equivalence calculation where the macro geometry is processed using the `EXCELT:` tracking module.

`SYBILT:` keyword to specify that the SPH factors are obtained using a transport-transport equivalence calculation where the macro geometry is processed using the `SYBILT:` tracking module.

`JPMT:` keyword to specify that the SPH factors are obtained using a transport-transport equivalence calculation where the macro geometry is processed using the `JPMT:` tracking module.

`BIVACT:` keyword to specify that the SPH factors are obtained using a transport-diffusion equivalence calculation where the macro geometry is processed using the `BIVACT:` diffusion tracking module. This option requires to use one of the keywords `LKRD`, `P0`, `P1`, `B0`, `B1` or `B0TR` in the flux calculation (see Section 3.7.1) so as to supply diffusion coefficients.

(desctrack) structure of the general tracking options (see Section 3.4.1).

(descexcel) structure of the `EXCELT:` tracking options (see Section 3.4.2).

(descsybil) structure of the `SYBILT:` tracking options (see Section 3.4.3).

(descjpm) structure of the `JPMT:` tracking options (see Section 3.4.4).

(descbivac) structure of the `BIVACT:` tracking options (see Section 3.4.5).

3.10 The `EVO:` module

The `EVO:` module in DRAGON allows the isotopic densities and the macroscopic cross sections to be updated following a in-core (i.e., burnup) or out-of-core depletion. The general format of the data which is used to control the execution of the `EVO:` module is the following:

Table 44: Structure **(EVO:)**

BRNNAM MICNAM `:=` `EVO:` [*BRNNAM*] { *MICNAM* | *OLDMIC* } *TRKNAM* [*FLUNAM*] `::`
(descevo)

where

BRNNAM `character*12` name of the BURNUP data structure that will contain the depletion history as modified by the depletion module. If *BRNNAM* appears on both LHS and RHS, it is updated; otherwise, it is created.

MICNAM `character*12` name of the MICROLIB that will contain the update macroscopic cross sections. If *MICNAM* appears on both LHS and RHS, it is updated; otherwise, the internal library *OLDMIC* is copied in *MICNAM* and *MICNAM* is updated.

OLDMIC `character*12` name of a read-only MICROLIB that is copied in *MICNAM*.

TRKNAM `character*12` name of a read-only TRACKING constructed for the depleting geometry. This information is required both for in-core and out-of-core depletion cases.

FLUNAM `character*12` name of a read-only FLUXUNK. This information is used only for in-core depletion cases.

(descevo) structure containing the input data to this module (see Section 3.10.1).

For the in-core depletion cases, the tracking TRACKING data structure on which *FLUNAM* is based, is automatically recovered in read-only mode from the generalized driver dependencies.

3.10.1 Data input for module EVO:

Table 45: Structure **(evo)**

[EDIT *iprint*]
[{ SAVE *xts* { S | DAY | YEAR } [{ FLUX *flux* | POWR *power* }] | NOSA }]
[EPS1 *valeps1*] [EPS2 *valeps2*] [EXPM *valexp*] [H1 *valh1*]
[{ RUNG | KAPS }]
[DEPL { *xti xtf* | *dxt* } { S | DAY | YEAR } [{ COOL | FLUX *flux* | POWR *power* }]]
[SET *xtr* { S | DAY | YEAR }]

where

EDIT keyword used to modify the print level *iprint*.

iprint index used to control the printing of the module. The amount of output produced by this tracking module will vary substantially depending on the print level specified.

SAVE keyword to specify that the results of the last transport calculation and the current isotopic concentration must be stored on *BRNNAM* on a sub-directory corresponding to a specific time. By default this data is stored at a time corresponding to *xti*.

NOSA keyword to specify that the results of the last transport calculation and the current isotopic concentration will not be stored on *BRNNAM*. By default this data is stored at a time corresponding to *xti*.

SET keyword used to recover the isotopic concentration already stored on *BRNNAM* on a sub-directory corresponding to a specific time. By default this data is recovered from a time corresponding to *xtf*.

DEPL keyword to specify that a burnup calculation between an initial and a final time must be performed. In the case where the SAVE keyword is absent, the initial isotopic concentration will be stored on *BRNNAM* on a sub-directory corresponding to the initial time. If the SET keyword is absent, the isotopic concentration corresponding to the final burnup time will be recovered from the FLUXUNK structure.

xti initial time associated with the burnup calculation. By default *xti* is the final time reached at the last depletion step. If this is the first depletion step, *xti*=0. The name of the sub-directory where

this information is stored will be given by 'DEPL-DAT'//CNN where CNN is a `character*4` variable defined by `WRITE(CNN,'(I4)') INN` where INN is an index associated with the time *xti*. The initial values are recovered from this sub-directory in *BRNNAM*.

dxt time interval for burnup calculation. The initial time *xti* in this case is taken as the final time reached at the last burnup step. If this is the first depletion step, *xti*=0 (Release 3.04).

xtf end of time for the burnup calculation. The results of the isotopic depletion calculations are stored in the tables associated with a sub-directory whose name is constructed in the same manner as the *xti* input. In the case where the time interval *dxt* is provided then *xtf*=*xti*+*dxt*.

xts time associated with the last transport calculation. The name of the sub-directory where this information is to be stored is constructed in the same manner as the for *xti* input. By default (fixed default) *xts*=*xti*.

xtr time associated with the next flux calculation. The name of the sub-directory where this information is to be stored is constructed in the same manner as for the *xti* input. By default (fixed default) *xtr*=*xtf*.

S keyword to specify that the time is given in seconds.

DAY keyword to specify that the time is given in days.

YEAR keyword to specify that the time is given in years.

COOL keyword to specify that a zero flux burnup calculation is to be performed.

FLUX keyword to specify that isotopic a constant flux burnup calculation is to be performed.

flux flux expressed in $\mathrm{cm}^{-2}\mathrm{s}^{-1}$. In the case where *flux*$\leq$0.0, the calculations are performed as if the COOL option was used.

POWR keyword to specify that isotopic a constant power burnup calculation is to be performed.

power power expressed in KW$\times$Kg^{-1} = MW$\times$tonne^{-1}. In the case where *power*$\leq$0.0, the calculations are performed as if the COOL option was used.

EPS1 keyword to specify the tolerance used in the algorithm for the solution of the depletion equations.

valeps1 the tolerance used in the algorithm for the solution of the depletion equations. The default value is 1.0×10^5.

EPS2 keyword to specify the tolerance used in the search algorithm for a final fixed power (used if the POWR option is activated).

valeps2 the tolerance used in the search algorithm for a final fixed power. The default value is 1.0×10^{-4}.

EXPM keyword to specify the selection criterion for non-fissile isotopes that are at saturation.

valexp the isotopes for which $\lambda \times (xtf - xti) \geq valexp$ will be treated by a saturation approximation. Here, λ is the sum of the radioactive decay constant and microscopic neutron absorption rate. The default value is 80.0. In order to remove the saturation approximation for all isotopes set *valexp* to a very large number such as 1.0×10^5.

H1 keyword to specify an estimate of the relative width of the time step used in the solution of burnup equations.

valh1 relative width of the time step used in the solution of burnup equations. An initial time step of $\Delta_t = valh1 \times (xtf - xti)$ is used. This value is optimized dynamically by the program. The default value is 1.0×10^{-4}.

`RUNG` keyword to specify a solution of the depletion equations using the 5^{th} order Runge–Kutta algorithm.

`KAPS` keyword to specify a solution of the depletion equations using the 4^{th} order Kaps–Rentrop algorithm.[54] This is the default value.

3.11 The `CPO:` module

The `CPO:` module is used to generate the reactor cross-section database required for a full core calculation. The calling specifications are:

Table 46: Structure **(CPO:)**

CPONAM `:=` `CPO:` [*CPONAM*] *EDINAM* [*BRNNAM*] `::` **(desccpo)**

where

CPONAM `character*12` name of the CPO data structure containing the reactor database. Additional contributions can be included in the reactor cross-section database if *CPONAM* appears on the RHS.

EDINAM `character*12` name of the read-only EDITION data structure containing the results from the `EDI:` module.

BRNNAM `character*12` name of the read-only BURNUP data structure containing the depletion history. This information is given only if the reactor database is to contain burnup dependent data.

(desccpo) structure containing the input data to this module (see Section 3.11.1).

The MICROLIB or MACROLIB data structure on which *EDINAM* and *BRNNAM* are based, is automatically recovered in read-only mode from the generalized driver dependencies.

3.11.1 Data input for module `CPO:`

Table 47: Structure **(desccpo)**

[`EDIT` *iprint*]
[`B2`] [`NOTR`]
{ `STEP` *NOMDIR* | `BURNUP` *PREFIX* }
[[`EXTRACT` { `ALL` | *NEWNAME* (*OLDNAME*(i), i=1,niext) }]]
[`NAME` *NDIR*]

where

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing of this module. The amount of output produced by this tracking module will vary substantially depending on the print level specified.

B2 keyword to specify that the buckling correction (dB^2) is to be applied to the cross section to be stored on the reactor database. By default (fixed default), such a correction is not taken into account.

NOTR keyword to specify that the cross section to be stored on the reactor database are not to be transport corrected. By default (fixed default), transport corrected cross section are considered when the CTRA option is activated in MAC: or LIB: (see Sections 3.1 and 3.2).

STEP keyword to specify that a specific cross section directory stored in *EDINAM* via the SAVE option in the EDI: module is to be transferred to *CPONAM*.

NOMDIR character*12 name of the specific cross section directory to be treated.

BURNUP keyword to specify that a chain of cross section directory stored in *EDINAM* via the SAVE option in the EDI: module will be transferred to *CPONAM*.

PREFIX character*8 prefix name of the cross section directory to be treated. DRAGON will transfer into the reactor database all the directory with full name *NOMDIR* created using

WRITE(*NOMDIR*,'(A8,I4)') *PREFIX*, NB

where NB is an integer greater than 0 indicating the depletion step index.

EXTRACT keyword to specify that the contribution of some isotopes to the macroscopic cross sections associated with each homogenized mixture should be extracted before being stored on the reactor database. The microscopic cross sections and concentrations associated with these isotopes should also be generated and stored on the reactor database.

ALL keyword to specify that all the isotopes processed using the MICR option of the EDI: module should be extracted from the macroscopic cross sections associated with each homogenized mixture.

NEWNAME character*12 name under which a given set of extracted isotope will be stored on the reactor database.

OLDNAME array of character*8 name of isotopes to be extracted from the macroscopic cross section associated with each homogenized mixture.

NAME keyword to specify the prefix for the name of the sub-directory where the information corresponding to a single homogenized region will be stored. The fixed default is *NDIR*='COMPO '.

NDIR character*8 prefix for the name of the sub-directory. The complete name is constructed by the concatenation of *NDIR* with a four digit integer value.

3.12 The INFO: module

The INFO: module is mainly used to compute the number densities for selected isotopes at specific local conditions. The module can also be used to compute the water density $\rho(T,P)$ according to the assumed temperature T and purity P. In that case, the compound water density for a mix of light and heavy water is

$$\rho(T,P) = \frac{100\ \rho_{H_2O}(T)\ \rho_{D_2O}(T)}{P\ \rho_{H_2O}(T) + (1-P)\ \rho_{D_2O}(T)}\ .$$

Temperature tabulations for $\rho_{H_2O}(T)$ and $\rho_{D_2O}(T)$ are the same as those of the WIMS–AECL code.[29] The calling specifications are:

Table 48: Structure **(INFO:)**

`INFO: ::` **(descinfo)**

where

(descinfo) structure containing the input data to this module (see Section 3.12.1).

3.12.1 Data input for module `INFO:`

Table 49: Structure **(descinfo)**

[`EDIT` *iprint*] [`LIB:` { `DRAGON` | `MATXS` | `MATXS2` | `WIMSD4` | `WIMS` | `WIMSAECL` | `APLIB1` } `FIL:` *NAMEFIL*] [`TMP:` *temp* { `K` | `C` }] [`PUR:` *purity* { `WGT%` | `ATM%` }] [`CALC DENS WATER` >>*dens*<<] [`ENR:` *enrichment* { `WGT%` | `ATM%` }] [[`ISO:` *nbiso* (*ISONAM*(i), i=1,nbiso) { `GET MASS` (>>*mass*(i)<<, i=1,nbiso) | `CALC WGT%` { `D2O` >>*nh1*<< >>*hd2*<< >>*no16*<< | `UO2` >>*nu5*<< >>*hu8*<< >>*no16*<< | `THO2` >>*nth2*<< >>*nu3*<< >>*no16*<< } }]]

where

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing of the module. The amount of output produced by this tracking module will vary substantially depending on the print level specified.

`LIB:` keyword to specify the type of library from which the isotopic mass ratio is to be read.

`DRAGON` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the DRAGLIB format.

`MATXS` keyword to specify that the microscopic cross sections are in the MATXS format of NJOY–II and NJOY–89.

`MATXS2` keyword to specify that the microscopic cross sections are in the MATXS format of NJOY–91.

`WIMSD4` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the WIMS–D4 format.

`WIMS` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the WIMS–AECL format.

`WIMSAECL` keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the WIMS–AECL format.

`APLIB1` keyword to specify that the microscopic cross sections are in the APOLLO–1 format (no depletion chains available for libraries using this format).

`FIL:` keyword to specify the name of the file where is stored the mass ratio data.

NAMEFIL `character*8` name of the library where the mass ratio are stored.

`TMP:` keyword to specify the isotopic temperature.

temp temperature given in Kelvin (`K`) or Celsius (`C`).

`PUR:` keyword to specify the water purity, that is fraction of heavy water in a mix of heavy and light water.

purity water purity in weight percent (`WGT%`) or atomic percent (`ATM%`).

`ENR:` keyword to specify the fuel enrichment.

enrichment fuel enrichment in weight percent (`WGT%`) or atomic percent (`ATM%`).

`ISO:` keyword to specify an isotope list. This list will be used either for getting mass values of isotopes or for computing number densities.

nbiso number of isotopic names used for a calculation (limited to $nbiso \leq 3$).

ISONAM `character*12` name of an isotope.

`GET MASS` keyword to recover the mass values as written in the library. It returns the mass value of each isotope in the output parameter *mass*.

`CALC` keyword to ask the module to compute some parametric values. It returns one value in the output parameter *dens*.

`DENS WATER` set of keywords to recover the water density as a function of its temperature and purity. This option requires the setting of temperature and purity, and it does not affect any given list of isotope names.

`WGT% D2O` keywords to recover 3 number densities for a compound mixture of heavy and light water. The isotope list is assumed to contain ^{1}H, ^{2}D and ^{2}O. Temperature and purity are supposed to be available. It returns concentration of these isotopes in the output parameters *nh1*, *nd2* and *no16*.

`WGT% UO2` keywords to recover 3 number densities for a compound mixture of Uranium oxide. The isotope list is assumed to contain ^{235}U, ^{238}U and ^{16}O. The ^{235}U enrichment is supposed to be available. Note that the number densities will sum to 100. It returns concentration of these isotopes in the output parameters *nu5*, *nu8* and *no16*.

`WGT% THO2` keywords to recover 3 number densities for a compound mixture of Thorium/Uranium oxide. The isotope list is assumed to contain ^{232}Th, ^{233}U and ^{16}O. The ^{233}U enrichment is supposed to be available. Note that the number densities will sum to 100. It returns concentration of these isotopes in the output parameters *nth2*, *nu3* and *no16*.

The `INFO:` module works the following way. For any isotope list given, mass extraction or a calculation process is expected. Once this calculation is done, it is possible to list other isotopes and ask for further calculations. Finally note that the number of output (denoted by $>>param<<$) parameters must be equal to the number of isotopes names given, plus the water density when a command `CALC DENS WATER` is issued.

3.13 The CFC: module

(Release 3.03)

The CFC: module is used to generate a Feedback Model database required for a full core calculation.[33–35] The general specifications of this module are:

Table 50: Structure **(CFC:)**

CFCNAM := CFC: [*CFCNAM*]
(*CPONAM*(i), i=1,28) :: **(desccfc)**

where

CFCNAM character*12 name of the FBMXSDB data structure containing the Feedback Model reactor database. The reactor database can be updated if *CFCNAM* appears on the RHS.

CPONAM character*12 name of read only CPO data structures. There are 28 different CPO data structures required here each containing respectively

1. the reactor reference cross section.
2. cell cross section for the first fuel temperature.
3. cell cross section for the second fuel temperature.
4. cell cross section for the first coolant temperature.
5. cell cross section for the second coolant temperature.
6. cell cross section for the first moderator temperature.
7. cell cross section for the second moderator temperature.
8. cell cross section for the first coolant density.
9. cell cross section for the second coolant density.
10. cell cross section for the first moderator density.
11. cell cross section for the second moderator density.
12. cell cross section for a different concentration of boron.
13. cell cross section for a different moderator purity.
14. cell cross section for a different concentration of xenon.
15. cell cross section for a different concentration of samarium.
16. cell cross section for a different concentration of neptunium.
17. cell cross section for the spectral mixed effect fuel/coolant density.
18. cell cross section for the spectral mixed effect coolant density/temperature.
19. cell cross section for low power history.
20. cell cross section for intermediate power history.
21. cell cross section for high power history.
22. reactor reference moderator cross section.
23. moderator cross section for the first moderator temperature.

24. moderator cross section for the second moderator temperature.
25. moderator cross section for the first moderator density.
26. moderator cross section for the second moderator density.
27. moderator cross section for a different concentration of boron.
28. moderator cross section for a different moderator purity.

(desccfc) structure containing the input data to this module (see Section 3.13.1).

3.13.1 Data input for module `CFC:`

Table 51: Structure **(desccfc)**

[`INFOR` *TITLE*]
[`DNAME` *RNANE*]

where

`INFOR` key word which allows to set the title.

TITLE `character*72` title associated to the reactor database generated.

`DNAME` key word which permits to set a specific database name in the data structure.

RNAME `character*12` name of the feedback database.

3.14 The `MRG:` module

(Release 3.03)

The `MRG:` module is used to pre-homogenize a geometry after it has been tracked with the `EXCELT:` tracking module (see Sections 3.4 and 3.4.2).[55,56] The general specifications of this module are:

Table 52: Structure **(MRG:)**

TRKNEW TFILNEW `:=` `MRG:` *TRKOLD TFILOLD* `::`
(descmrg)

where

TRKNEW `character*12` name of the new TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information after the pre-homogenization process.

TFILNEW `character*12` name of the new sequential binary tracking file used to store the tracks lengths after the pre-homogenization process has take place.

TRKOLD `character*12` name of the old TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information before the pre-homogenization process.

TFILOLD `character*12` name of the old sequential binary tracking file used to store the tracks lengths before the pre-homogenization process has take place.

(descmrg) structure containing the input data to this module (see Section 3.14.1).

3.14.1 Data input for module `MRG:`

Table 53: Structure **(descmrg)**

[`EDIT` *iprint*]
[`REGI` (*irmrg*(i), i=1,*nreg*)]
[`SURF` (*ismrg*(i), i=1,*nsur*)]

where

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing in this module. It must be set to 0 if no printing on the output file is required.

`REGI` keyword to specify that regions will be pre-homogenized.

irmrg new region numbers associated with old region numbers. Two or more old regions can be combined together only if they contain the same mixture. The number *nreg* of old region is that printed after the execution of the `EXCELT:` module.

`SURF` keyword to specify that surfaces will be pre-homogenized.

ismrg new surface numbers associated with old surface numbers. Two or more old surface can be combined together only if they are associated with the same albedo. The number *nsur* of old surfaces is that printed after the execution of the `EXCELT:` module.

3.15 The `PSP:` module

(Release 3.04)

The `PSP:` module is used to generate a graphical file in a PostScript ASCII format for a DRAGON 2–D geometry which can be analyzed using the `EXCELT:` tracking module (see Sections 3.4 and 3.4.2). The module `PSP:` is based on the PSPLOT FORTRAN library from Nova Southeastern University.[57] Since only a few routines PSPLOT routines were required and because additional PostScript routine not present in the original package were needed, the routines have been completely readapted to DRAGON. These routines are no longer machine dependent and are now Y2K safe. The PostScript files generated by DRAGON can be viewed by any PostScript

viewer, such as Ghostview[58] or sent to a printer compatible with this language. In DRAGON the PSP: module is activated using the following list of commands:

Table 54: Structure **(PSP:)**

PSGEO := PSP: *PSGEO* { *GEONAM* | *TRKNAM* } [*FLUNAM*] :: **(descpsp)**

where

PSGEO character*12 name of the file that will contain the graphical description in a POSTSCRIPT format. This file must have a sequential ASCII format.

GEONAM character*12 name of a read-only GEOMETRY (see Section 3.3).

TRKNAM character*12 name of an EXCELL type read-only TRACKING (see Section 3.4).

FLUNAM character*12 name of an optional read-only FLUXUNK (see Section 3.7). It is required only if a flux mapping plot is requested.

(descpsp) structure containing the input data to this module (see Section 3.15.1).

3.15.1 Data input for module PSP:

Table 55: Structure **(descpsp)**

[EDIT *iprint*]
[FILL { NONE | GRAY | RGB | CMYK | HSB } [NOCONTOUR]]
[TYPE { REGION | MIXTURE | FLUX | MGFLUX }]

where

EDIT keyword used to modify the print level *iprint*.

iprint index used to control the printing in this module. It must be set to 0 if no printing on the output file is required.

FILL keyword to specify the drawing options.

NONE keyword to specify that only region contour are to be drawn.

GRAY keyword to specify that the regions will be filled with various levels of gray.

RGB keyword to specify that the regions will be filled with various colors taken using the RGB color scheme.

CMYK	keyword to specify that the regions will be filled with various colors taken using the CMYK color scheme.
HSB	keyword to specify that the regions will be filled with various colors taken using the HSB color scheme. This is the default option.
NOCONTOUR	keyword to specify that the contour lines delimiting each region will not be drawn.
TYPE	keyword to specify the type of graphics generated.
REGION	keyword to specify that different colors or gray levels will be associated with each region. This is the default option.
MIXTURE	keyword to specify that different colors or gray levels will be associated with each mixture.
FLUX	keyword to specify that the group integrated flux is to be drawn.
MGFLUX	keyword to specify that the group flux is to be drawn.

4 EXAMPLES

We will now present a few examples of DRAGON input structures in such as to clarify and illustrate some of the options presented in Section 3.

4.1 Scattering cross sections

In DRAGON, the angular dependence of the scattering cross section is expressed in a Legendre series expansion of the form:

$$\Sigma_s(\Omega \cdot \Omega') = \Sigma_s(\mu) = \sum_{l=0}^{L} \left(\frac{(2l+1)}{4\pi} \right) \Sigma_{s,l} P_l(\mu).$$

Since the Legendre polynomials satisfy the following orthogonality conditions:

$$\int_{-1}^{1} d\mu P_l(\mu) P_m(\mu) = \left(\frac{2\delta_{l,m}}{(2l+1)} \right),$$

we will have

$$\Sigma_{s,l} = \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\varphi \Sigma_s(\mu) P_l(\mu) = 2\pi \int_{-1}^{1} d\mu \Sigma_s(\mu) P_l(\mu).$$

Let us now consider the following three group (*ngroup*=3) isotropic and linearly anisotropic scattering cross sections (*L=naniso*=2) given by:

l	g	$\Sigma_{s,l}^{g\to1}$ (cm^{-1})	$\Sigma_{s,l}^{g\to2}$ (cm^{-1})	$\Sigma_{s,l}^{g\to3}$ (cm^{-1})
	1	0.90	0.80	0.00
0	2	0.00	0.70	0.60
	3	0.00	0.30	0.40
	1	0.09	0.05	0.08
1	2	0.00	0.07	0.06
	3	0.03	0.00	0.04

In DRAGON this scattering cross section must be entered as

```
SCAT  (* L=0 *) 1 1 (* 3->1 *)      (* 2->1 *)      (* 1->1 *) 0.90
                3 3 (* 3->2 *) 0.30 (* 2->2 *) 0.70 (* 1->2 *) 0.80
                2 3 (* 3->3 *) 0.40 (* 2->3 *) 0.60 (* 1->3 *)
SCAT  (* L=1 *) 3 3 (* 3->1 *) 0.03 (* 2->1 *) 0.00 (* 1->1 *) 0.09
                2 2 (* 3->2 *)      (* 2->2 *) 0.07 (* 1->2 *) 0.05
                3 3 (* 3->3 *) 0.04 (* 2->3 *) 0.06 (* 1->3 *) 0.08
```

4.2 Geometries

In order to illustrate the use of the various geometries presented in Section 3.3, lets us consider a few examples that can be treated by DRAGON.

- 1–D Slab geometry (see Figure 10):

  This geometry can be analyzed using the `JPMT:` and `SYBILT:` tracking modules:

```
PLATE := GEO: :: CAR1D 6
    X- VOID X+ ALBE 1.2
    MESHX  0.0 0.1 0.3 0.5 0.6 0.8 1.0
    SPLITX     2   2   2   1   2   1
    MIX        1   2   3   4   5   6 ;
```

- 2–D Cartesian geometry containing micro structures (see figure Figure 11):

  This geometry can be analyzed only using SYBILT: tracking modules:

```
CARNSG := GEO: :: CAR2D 3 3
  X- DIAG X+ REFL Y- SYME Y+ DIAG
  CELL  C1  C1  C2
            C3  C2
                C3
  BIHET SPHE  2  2
              3  3
              0.0 0.1 0.2 0.3 0.0 0.2 0.4 0.5
              4 5 1 1 0.4 0.0 3 1 3 0.2 0.1
              1 2 1 2 3 1
  ::: C1 := GEO: CAR2D 1 1
    MESHX 0.0 1.45 MESHY 0.0 1.45 MIX 4  ;
  ::: C2 := GEO: C1
    MIX 1  ;
  ::: C3 := GEO: CARCEL 2
    MESHX 0.0 1.45 MESHY 0.0 1.45
    RADIUS 0.0 0.6 0.7
    MIX        5   2   1 ; ;
```

- Cylindrical and Cartesian cluster geometry (see Figure 12):

  The first two geometry, namely ANNPIN and CARPIN can be analyzed using both the JPMT: and EXCELT: tracking modules since the pins in the clusters are all located between annular region. For the last two geometries, ANNSPIN and CARSPIN, which are based on ANNPIN and CARPIN respectively, they can only be treated by the EXCELT: tracking modules since the pins in the clusters intersect the annular regions defined by the SPLITR option. This later option which was selected to ensure a uniform thickness of 0.25 cm for each the annular region in the final geometries.

```
ANNPIN := GEO: :: TUBE 3
  R+ REFL  RADIUS 0.0 0.75 2.75 4.75
  MIX 2 1 3
  CLUSTER C1 C2
  ::: C1 := GEO: TUBE 2
    MIX 2 4 RADIUS 0.0 0.3 0.6
    NPIN 4  RPIN 1.75  APIN 0.523599 ;
  ::: C2 := GEO: C1
    NPIN 2  RPIN 3.75  APIN 1.570796 ; ;
CARPIN := GEO: :: CARCEL 3
  X- REFL X+ REFL Y- REFL  Y+ REFL
  MESHX 0.0 10.0 MESHY -5.0 5.0
  RADIUS 0.0 0.75 2.75 4.75
  MIX 2 1 3 3
  CLUSTER C1 C2
  ::: C1 := GEO: TUBE 2
    MIX 2 4 RADIUS 0.0 0.3 0.6
    NPIN 4  RPIN 1.75  APIN 0.523599 ;
  ::: C2 := GEO: C1
    NPIN 2  RPIN 3.75  APIN 1.570796 ; ;
ANNSPIN := GEO: ANNPIN ::
```

```
  SPLITR   3 8 8 ;
CARSPIN := GEO: CARPIN ::
  SPLITR   3 8 8 ;
```

Note that even if `MESHX` and `MESHY` differ in `CARPIN`, the annular regions and pins will still be localized with respect to the center of the cell located at $(x, y) = (5.0, 0.0)$ cm.

- 2–D hexagonal geometry (see Figure 13):

  This geometry can be analyzed using the `JPMT:`, `SYBILT:` and `EXCELT:` tracking modules:

```
HEXAGON := GEO: :: HEX 12
  HBC S30 ALBE 1.6
  SIDE 1.3
  MIX 1 1 1 2 2 2 3 3 3 4 5 6
;
```

- 3–D Cartesian supercell (see Figure 14):

  This geometry can only be analyzed using the `EXCELT:` tracking modules:

```
SUPERCELL := GEO: :: CAR3D 4 4 3
  X- REFL   X+ REFL
  Y- REFL   Y+ REFL
  Z- REFL   Z+ REFL
  CELL A1 C1 D1 A3   A2 C2 D2 D2   A2 C2 C2 C2   A2 C2 C2 C2
       C3 C3 D3 A4   C4 C4 D4 D4   C4 C4 C4 C4   C4 C4 C4 C4
       C3 C3 D3 A4   C4 C4 D4 D4   C4 C4 C4 C4   C4 C4 C4 C4
  ::: C1 := GEO: CAR3D 1 1 1
    MESHX 0.0 1.0 MESHY 0.0 1.5 MESHZ 0.0 2.0
    MIX   1 ;
  ::: C2 := GEO: C1 MESHY 0.0 1.0 ;
  ::: C3 := GEO: C1 MESHZ 0.0 1.0 ;
  ::: C4 := GEO: C2 MESHZ 0.0 1.0 ;
  ::: D1 := GEO: C1 MIX 2  ;
  ::: D2 := GEO: C2 MIX 2  ;
  ::: D3 := GEO: C3 MIX 2  ;
  ::: D4 := GEO: C4 MIX 2  ;
  ::: A1 := GEO: CARCELY 2 1
    MESHX 0.0 1.0 MESHY 0.0 1.5 MESHZ 0.0 2.0
    RADIUS 0.0 0.4 0.45
    MIX        3   4   1 ;
  ::: A2 := GEO: A1 MESHY 0.0 1.0 ;
  ::: A3 := GEO: CARCELZ 2 1
    MESHX 0.0 1.0 MESHY 0.0 1.5 MESHZ 0.0 2.0
    RADIUS 0.0 0.3 0.35
    MIX        5   6   1 ;
  ::: A4 := GEO: A3 MESHZ 0.0 1.0 ; ;
```

- Multicell geometry in a 2–D hexagonal lattice (see Figure 15).

  Here we are considering an infinite lattice having two types of cells such that

$$\begin{pmatrix} \text{pource}(1) \\ \text{pource}(2) \end{pmatrix} = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \text{procel}(1,1) & \text{procel}(1,2) \\ \text{procel}(2,1) & \text{procel}(2,2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1/2 & 1/2 \end{pmatrix}$$

This lattice, can be represented either in a *do-it-yourself* type geometry (HEXDIY) or directly (HEXDIR):

```
HEXDIY := GEO: :: GROUP 2
  POURCE 0.3333333 0.66666667
  PROCEL 0.0       1.0
         0.5       0.5
  CELL   C1 C2
  ::: C1 := GEO: TUBE 1
    RADIUS 0.0 1.1822093 MIX 1 ;
  ::: C2 := GEO: C1 MIX 2 ; ;
HEXDIR := GEO: :: HEX 2
  HBC S30 SYME SIDE 1.3 MIX 1 2 ;
```

The first lattice can only be analyzed using SYBILT: and JPMT: while the second lattice can be analyzed using all the tracking modules of DRAGON.

4.3 Macroscopic cross sections examples

The sample test cases we will consider here use the MAC: module to enter macroscopic cross sections directly into DRAGON. They are numbered successively from **TCM01** to **TCM05**.

4.3.1 ***TCM01*** *– Annular region*

This sample input is used to analyze the annular cell presented in Figure 16. It uses two groups macroscopic cross sections provided directly by the user. Two types of solutions are provided here, one with a complete collision probability calculation (SYBILT:) and one using the $J_{\pm}$ method (JPMT:). Note that for the second flux calculation the initial flux distribution is taken from the existing FLUXUNK structure which already contains the flux distribution from the SYBILT: calculation.

Input data for test case: **TCM01**

```
*----
* TEST CASE TCM01
* MACROSCOPIC CROSS SECTIONS
* FISSION SOURCE PROBLEM
* 1-D ANNULAR CELL
*
* REF: none
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  MACRO ANGEO TRACK SYS FLUX EDITION ;
SEQ_ASCII
  RES ;
MODULE
  GEO: SYBILT: JPMT: MAC: ASM: FLU: EDI: DELETE: END: ;
*----
*  Macroscopic XS
*----
```

```
MACRO := MAC: ::
  NGRO 2 NMIX 2 NIFI 1
  READ INPUT
  MIX 1   TOTAL 0.222222 0.833333
    SCAT   1 1  0.19222          2 2 0.75333 0.02
    NUSIGF  0.0  0.135 CHI 1.0 0.0
  MIX 2   TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.04
  ;
*----
*  Geometry :  ANGEO - Annular 2 regions
*----
ANGEO := GEO: :: TUBE 2
  R+ REFL
  RADIUS 0.0 0.19653 1.0
  MIX        1       2
  SPLITR     1       4    ;
*----
*  Tracking : SYBILT
*  Solution : PIJ
*    1- KEFF WITHOUT BUCKLING
*    2- BUCKLING WITH KEFF=1
*    3- LEAKAGE WITH KEFF=1
*----
TRACK := SYBILT: ANGEO ::
  TITLE 'TCM01: ANNULAR GEOMETRY WITH MACROSCOPIC XS (SYBIL)'
  EDIT 1 MAXR 5 QUA1 5 ;
SYS := ASM: MACRO TRACK ;
FLUX := FLU:      SYS MACRO TRACK ::
  TYPE K ;
EDITION := EDI:         FLUX MACRO TRACK ::
  EDIT 3 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B0 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B0 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE ;
FLUX SYS TRACK := DELETE: FLUX SYS TRACK ;
*----
*  Tracking : JPMT
*  Solution : ASM
*    1- KEFF WITHOUT BUCKLING
*    2- BUCKLING WITH KEFF=1
*    3- LEAKAGE WITH KEFF=1
*----
TRACK := JPMT: ANGEO ::
  TITLE 'TCM01: ANNULAR GEOMETRY WITH MACROSCOPIC XS (JPM)'
  EDIT 1 MAXR 5 IP01 QUA1 5 ;
SYS := ASM: MACRO TRACK ::
```

```
  ARM ;
FLUX := FLU:      SYS MACRO TRACK ::
  TYPE K ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE STAT ALL REFE 1 ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B0 PNL EXTE 5.0E-5 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE STAT ALL REFE 2 ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
   TYPE L B0 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE STAT ALL REFE 3 ;
FLUX SYS := DELETE: FLUX SYS ;
*----
*  Tracking : JPMT
*  Solution : PIJ
*    1- KEFF WITHOUT BUCKLING
*    2- BUCKLING WITH KEFF=1
*    3- LEAKAGE WITH KEFF=1
*----
SYS := ASM: MACRO TRACK ;
FLUX := FLU:      SYS MACRO TRACK ::
  TYPE K ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE STAT ALL REFE 4
  ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B0 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE STAT ALL REFE 5 ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B0 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE STAT ALL REFE 6 ;
RES := EDITION ;
END: ;
QUIT "LIST" .
```

4.3.2 ***TCM02*** *– The Stankovski test case.*

This test case represents a one group calculation of a 7×7 PWR assembly. The reaction rates obtained from DRAGON can be compared with those obtained using the MARSYAS code.[18,19,59] The corresponding geometry is shown in Figure 17 where the cell numbers generated by DRAGON are shown.

Input data for test case: **TCM02**

```
*----
*  TEST CASE TCM02
*  MACROSCOPIC CROSS SECTIONS
*  FIXED SOURCE PROBLEM
```

```
*  FOR 1/8 7X7 PWR ASSEMBLY
*
*  REF: Z. Stankovski, Nucl. Sci. Eng. 92, 255 (1986)
*       R. Roy et al. Advances in Mathematics, Computation
*       and Reactor Physics, April 28 - May 2 1991, Pittsburgh
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  PWR TRACK MACRO SYS FLUX EDITION ;
SEQ_BINARY
  PWRTRK ;
MODULE
  GEO: EXCELT: MAC: ASM: FLU: EDI: END: DELETE: ;
*----
* Macroscopic XS
*----
MACRO := MAC: ::
  NGRO 1 NMIX 3
  READ INPUT
  MIX 1 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX 2 TOTAL  0.625  SCAT 1 1  0.355  FIXE 0.000
  MIX 3 TOTAL 14.000  SCAT 1 1  0.000  FIXE 0.000
  ;
*----
*  Geometry : PWR - Cartesian 4X4
*  Tracking : EXCELT
*----
PWR := GEO: :: CAR2D 4 4
  X- DIAG  X+ REFL Y- SYME  Y+ DIAG
  CELL   P F F F
           F F F
             F F
               F
  ::: F := GEO: CARCEL 1
    RADIUS 0.000 0.450 MIX 2  1
    MESHX -0.625  0.625 SPLITX 2
    MESHY -0.625  0.625 SPLITY 2
    ;
  ::: P := GEO: F
    MIX  3  1  SPLITR 3
    ;
  ;
TRACK PWRTRK := EXCELT: PWR ::
  TITLE 'TCM02: STANKOVSKI PWR ASSEMBLY'
  MAXR 58 CUT 1.E-4 TRAK TSPC 12 8.0
  ;
*----
*  Solution : FIXED SOURCE PROBLEM
*  Editing  : Absorption rates taken from STANKOVSKI
*    cell 1 = 0.83799 0.00689 cell 2 = 0.73979 0.03571
```

```
*    cell 3 = 0.82218 0.03991 cell 4 = 0.85166 0.04104
*    cell 5 = 0.78722 0.03824 cell 6 = 1.67049 0.08092
*    cell 7 = 1.71199 0.08252 cell 8 = 0.85350 0.04120
*    cell 9 = 1.72122 0.08328 cell 10= 0.86023 0.04174
*    NOTE: There is a factor 4.0 with the EDI: results of DRAGON
*----
SYS := ASM: MACRO TRACK PWRTRK :: ;
FLUX := FLU:       SYS MACRO TRACK ::
  TYPE S ;
EDITION := EDI: FLUX MACRO TRACK ::
  EDIT 2 SAVE
  MERGE REGION  1  1  1  2  3  4  3  4  5  6  5  6  7  8  7  8
  9 10  9 10  9 10 11 12 11 12 11 12 11 12 13 14 13 14 13 14 13 14
  15 16 15 16 15 16 17 18 17 18 17 18 17 18
  19 20 19 20 19 20
  ;
PWRTRK := DELETE: PWRTRK ;
END: ;
QUIT "LIST" .
```

4.3.3 ***TCM03*** *– Watanabe and Maynard problem with a void region.*

This test case is a one group problem with a central void region (see Figure 18). This benchmark was first proposed by Watanabe and Maynard. Akroyd and Riyait used it to analyze the performance of various codes.[18,19,60]

Input data for test case: **TCM03**

```
*----
*  TEST CASE TCM03
*  MACROSCOPIC CROSS SECTIONS
*  FIXED SOURCE PROBLEM
*  CARTESIAN 3 X 3 ASSEMBLY
*
*  REF: Akroyd and Riyait, Ann. Nucl. Energy 16, 1 (1989)
*       R. Roy et al. Advances in Mathematics, Computation
*       and Reactor Physics, April 28 - May 2 1991, Pittsburgh
*       R. Roy, Ann. Nucl. Energy 18, 511 (1991)
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
   WATA WAT08 WAT16 WAT24 TRACK MACRO SYS FLUX EDITION ;
SEQ_BINARY
   WATATRK ;
MODULE
  GEO: EXCELT: MAC: ASM: FLU: EDI: DELETE: END: ;
*----
* Macroscopic XS
*----
MACRO := MAC: ::
```

```
  NGRO 1 NMIX 3
  READ INPUT
  MIX 1 TOTAL 0.2 SCAT 1 1 0.19 FIXE 6.4
  MIX 2 TOTAL 0.2 SCAT 1 1 0.19
  MIX 3 TOTAL 0.0 SCAT 1 1 0.00
  ;
*----
*  Geometry : WATA   -  3 X  3 REGIONS
*             WAT08  -  8 X  8 REGIONS
*             WAT16  - 16 X 16 REGIONS
*             WAT24  - 24 X 24 REGIONS
*  Tracking : EXCELT
*----
WATA := GEO: :: CAR2D 3 3
  X- DIAG X+ VOID Y- REFL Y+ DIAG
  MESHX 0.00 1.25 5.00 10.00  MESHY 0.00 1.25 5.00 10.00
  MIX  1  3  2
          3  2
             2
  ;
WAT08 := GEO: WATA ::
  SPLITX  1  3  4  SPLITY  1  3  4
  ;
WAT16 := GEO: WATA ::
  SPLITX  2  6  8  SPLITY  2  6  8
  ;
WAT24 := GEO: WATA ::
  SPLITX  3  9 12  SPLITY  3  9 12
  ;
*----
*  Tracking : EXCELT - WAT08
*  Solution : FIXED SOURCE PROBLEM
*  Editing  : 1- UPPER QUADRANT FLUX
*             2- FLUX AT X=5.625CM
*----
TRACK WATATRK := EXCELT: WAT08 ::
  TITLE 'TCM03: WATANABE-MAYNARD 8X8 '
  MAXR 300 CUT 1.E-4 TRAK TSPC 12 4.0
  ;
SYS := ASM: MACRO TRACK WATATRK ::
  SKIP
  ;
FLUX := FLU: SYS MACRO TRACK ::
  TYPE S THER 1.E-6 100 EXTE 1.E-6 100
  ;
EDITION := EDI: FLUX MACRO TRACK ::
  EDIT 2 SAVE
  MERGE REGION
  0  0  0  0   0  0  0  0
     0  0  0   0  0  0  0
        0  0   0  0  0  0
           0   0  0  0  0
```

```
                1  2  3  4
                   5  6  7
                      8  9
                        10
  ;
EDITION := EDI: EDITION  FLUX MACRO TRACK ::
  EDIT 2 SAVE
  MERGE REGION
  0  0  0  0   1  0  0  0
     0  0  0   2  0  0  0
        0  0   3  0  0  0
           0   4  0  0  0
               5  6  7  8
                  0  0  0
                     0  0
                        0
  ;
TRACK WATATRK SYS FLUX := DELETE: TRACK WATATRK SYS FLUX ;
*----
*  Tracking : EXCELT - WAT16
*  Solution : FIXED SOURCE PROBLEM
*  Editing  : 1- UPPER QUADRANT FLUX
*             2- FLUX AT X=5.625CM
*----
TRACK WATATRK := EXCELT: WAT16 ::
  TITLE 'TCM03: WATANABE-MAYNARD 16X16 '
  MAXR 300 CUT 1.E-4 TRAK TSPC 12 8.0
  ;
SYS := ASM: MACRO TRACK WATATRK ::
  SKIP
  ;
FLUX := FLU: SYS  MACRO TRACK ::
  TYPE S THER 1.E-6 100 EXTE 1.E-6 100
  ;
EDITION := EDI: EDITION  FLUX  MACRO TRACK ::
  EDIT 2 SAVE
  MERGE REGION
  0 0 0 0 0 0 0 0  0  0   0  0  0  0  0  0
    0 0 0 0 0 0 0  0  0   0  0  0  0  0  0
      0 0 0 0 0 0  0  0   0  0  0  0  0  0
        0 0 0 0 0  0  0   0  0  0  0  0  0
          0 0 0 0  0  0   0  0  0  0  0  0
            0 0 0  0  0   0  0  0  0  0  0
              0 0  0  0   0  0  0  0  0  0
                0  0  0   0  0  0  0  0  0
                   1  1   2  2  3  3  4  4
                      1   2  2  3  3  4  4
                          5  5  6  6  7  7
                             5  6  6  7  7
                                8  8  9  9
                                   8  9  9
                                     10 10
```

```
                                                   10
    ;
  EDITION := EDI: EDITION FLUX  MACRO TRACK ::
    EDIT 2 SAVE
    MERGE REGION
    0 0 0 0 0 0 0 0  1  1  0  0  0  0  0  0
      0 0 0 0 0 0 0  2  2  0  0  0  0  0  0
        0 0 0 0 0 0  3  3  0  0  0  0  0  0
          0 0 0 0 0  4  4  0  0  0  0  0  0
            0 0 0 0  5  5  0  0  0  0  0  0
              0 0 0  6  6  0  0  0  0  0  0
                0 0  7  7  0  0  0  0  0  0
                  0  8  8  0  0  0  0  0  0

                     9 10 12 13 14 15 16 17
                       11 12 13 14 15 16 17
                           0  0  0  0  0  0
                              0  0  0  0  0
                                 0  0  0  0
                                    0  0  0
                                       0  0
                                          0
    ;
  TRACK WATATRK SYS FLUX := DELETE: TRACK WATATRK SYS FLUX ;
  *----
  *  Tracking : EXCELT - WAT24
  *  Solution : FIXED SOURCE PROBLEM
  *  Editing  : 1- UPPER QUADRANT FLUX
  *             2- FLUX AT X=5.625CM
  *----
  TRACK WATATRK := EXCELT: WAT24 ::
    TITLE 'TCM03: WATANABE-MAYNARD 24X24 '
       MAXR 300 CUT 1.E-4 TRAK TSPC 12 12.0
    ;
  SYS := ASM: MACRO TRACK WATATRK ::
    SKIP
    ;
  FLUX := FLU: SYS  MACRO TRACK ::
    TYPE S THER 1.E-6 100 EXTE 1.E-6 100
    ;
  EDITION := EDI: EDITION  FLUX  MACRO TRACK ::
    EDIT 2 SAVE
    MERGE REGION
    0 0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
      0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
        0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
          0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
            0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
              0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
                0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
                  0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
                    0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
                      0 0 0  0  0  0  0  0  0  0  0  0  0  0  0
                        0 0  0  0  0  0  0  0  0  0  0  0  0  0
                          0  0  0  0  0  0  0  0  0  0  0  0  0
                             0  0  0  0  0  0  0  0  0  0  0  0
                                1  0  0  2  0  0  3  0  0  4  0
                                   0  0  0  0  0  0  0  0  0  0
                                      0  0  0  0  0  0  0  0  0
                                         5  0  0  6  0  0  7  0
                                            0  0  0  0  0  0  0
                                               0  0  0  0  0  0
                                                  8  0  0  9  0
                                                     0  0  0  0
                                                        0  0  0
                                                          10  0
                                                              0
  ;
EDITION := EDI: EDITION  FLUX  MACRO TRACK ::
  EDIT 2 SAVE
  MERGE REGION
  0 0 0 0 0 0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  0  0  0  0
    0 0 0 0 0 0 0 0 0 0 0 0  0  2  0  0  0  0  0  0  0  0  0  0
      0 0 0 0 0 0 0 0 0 0 0  0  3  0  0  0  0  0  0  0  0  0  0
        0 0 0 0 0 0 0 0 0 0  0  4  0  0  0  0  0  0  0  0  0  0
          0 0 0 0 0 0 0 0 0  0  5  0  0  0  0  0  0  0  0  0  0
            0 0 0 0 0 0 0 0  0  6  0  0  0  0  0  0  0  0  0  0
              0 0 0 0 0 0 0  0  7  0  0  0  0  0  0  0  0  0  0
                0 0 0 0 0 0  0  8  0  0  0  0  0  0  0  0  0  0
                  0 0 0 0 0  0  9  0  0  0  0  0  0  0  0  0  0
                    0 0 0  0 10  0  0  0  0  0  0  0  0  0  0
                      0 0  0 11  0  0  0  0  0  0  0  0  0  0
                        0  0 12  0  0  0  0  0  0  0  0  0  0
                           0 13  0  0  0  0  0  0  0  0  0  0
                             14 15 16 17 18 19 20 21 22 23 24
                                 0  0  0  0  0  0  0  0  0  0
                                    0  0  0  0  0  0  0  0  0
                                       0  0  0  0  0  0  0  0
                                          0  0  0  0  0  0  0
                                             0  0  0  0  0  0
                                                0  0  0  0  0
                                                   0  0  0  0
                                                      0  0  0
                                                         0  0
                                                            0
  ;
WATATRK  := DELETE: WATATRK  ;
END: ;
QUIT "LIST" .
```

4.3.4 ***TCM04*** *– Adjuster rod in a CANDU type supercell.*

This test case represents a two group calculation of incremental cross sections resulting from the insertion of stainless steel adjuster rods in a CANDU–6 supercell (see Figure 19).

Input data for test case: **TCM04**

```
*----
*  TEST CASE TCM04
*  MACROSCOPIC CROSS SECTIONS
*  FISSION SOURCE PROBLEM
*  CANDU 3-D ADJUSTER ROD 1/8-ASSEMBLY
*
*  REF: none
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  BC TRACK  MACRO SYS  FLUX  EDITION
     TRACK2       SYS2 FLUX2 EDITION2    ;
SEQ_BINARY
  BCTRK ;
MODULE
  GEO: EXCELT: EXCELL: MAC: ASM: FLU: EDI:  DELETE: END: ;
*----
* Macroscopic XS
*----
MACRO := MAC: ::
 NGRO 2 NMIX 4 NIFI 1
 READ INPUT
 MIX 1  TOTAL     3.22798014E-1 3.81341100E-1
        NUSIGF    5.46564534E-3 7.17375278E-2
        CHI       1.0           0.0
        SCAT 2 2  3.13575147E-4 3.11233580E-1
             2 2  3.24143648E-1 2.19577667E-3
 MIX 2  TOTAL     1.49818063E-1 1.59792125E-1
        SCAT 2 2  7.40572286E-5 1.47693634E-1
             2 2  1.57371104E-1 1.30506000E-3
 MIX 3  TOTAL     2.60458171E-1 3.77224326E-1
        SCAT 2 2  5.98954648E-5 2.49342978E-1
             2 2  3.77127469E-1 1.11155845E-2
 MIX 4  TOTAL     2.60458171E-1 3.77224326E-1
        SCAT 2 2  5.98954648E-5 2.49342978E-1
             2 2  3.77127469E-1 1.11155845E-2
 ;
*----
*  Geometry : BC - 3D Cartesian assembly with annular regions
*  Tracking : 1) EXCELT
*             2) EXCELL (includes ASM and does not require track file)
*----
BC := GEO: :: CAR3D 3 2 2
```

```
    X- REFL X+ SYME Y- REFL Y+ SYME Z- REFL Z+ SYME
    CELL  M   MX  MX MX  FXY MXY   M   MX  BX MX  FXY BXY
    TURN  A   A   A  F   A   A     A   A   A  F   A   A
    ::: M := GEO: CAR3D 1 1 1
      MESHX  0.0 7.14375 MESHY 0.0 7.14375 MESHZ -8.25500 +8.25500
      SPLITZ 2 MIX 3
      ;
    ::: MX := GEO: M
      MESHX -7.14375 +7.14375 SPLITX 2
      ;
    ::: MXY := GEO: MX
      MESHY -7.14375 +7.14375 SPLITY 2
      ;
    ::: BX := GEO: CARCELY 2 1
      MESHX -7.14375 +7.14375 SPLITX  2
      MESHY  0.0     +7.14375
      MESHZ -8.25500 +8.25500 SPLITZ  2
      RADIUS 0.0 3.5100 3.8100
      MIX       3      4      3
      ;
    ::: BXY := GEO: BX
      MESHY -7.14375 +7.14375 SPLITY 2
      ;
    ::: FXY := GEO: CARCELZ 2 1
      MESHX -7.14375 +7.14375 SPLITX  2
      MESHY -7.14375 +7.14375 SPLITY  2
      MESHZ -8.25500 +8.25500 SPLITZ  2
      RADIUS 0.0 5.16890 6.58750
      MIX       1       2       3
      ;
    ;
  TRACK BCTRK := EXCELT: BC ::
    TITLE 'TCM04: TWO GROUPS CANDU 3-D ADJUSTER ROD ASSEMBLY '
    MAXR 40 TRAK TISO 4 2.5
    ;
  SYS := ASM: MACRO TRACK BCTRK :: ;
  TRACK2 SYS2 := EXCELL: BC MACRO ::
    TITLE 'TCM04: TWO GROUPS CANDU 3-D ADJUSTER ROD ASSEMBLY '
    MAXR 40 TRAK NORM TISO 4 2.5
    ;
  *----
  *  Solution : K-EFFECTIVE
  *  Editing  : Compute reference reaction rates
  *----
  FLUX := FLU: SYS MACRO TRACK ::
    TYPE K
    ;
  EDITION := EDI: FLUX MACRO TRACK ::
    EDIT 3 UPS  MERG COMP SAVE ON 'NOROD'
    ;
  FLUX := FLU: FLUX SYS2 MACRO TRACK2 ::
    TYPE K
```

```
  ;
EDITION := EDI: EDITION FLUX MACRO TRACK2 ::
  EDIT 3 UPS  MERG COMP STAT ALL REFE 'NOROD'
  ;
EDITION2 := EDI: FLUX MACRO TRACK2 ::
  EDIT 3 UPS  MERG COMP SAVE ON 'NOROD'
  ;
SYS SYS2 TRACK2 := DELETE: SYS SYS2 TRACK2 ;
*----
*  Modify Macrolib for adjuster rod material
*  Solution : K-EFFECTIVE
*  Editing  : Compute Delta-Sigma
*----
MACRO := MAC: MACRO ::
  READ INPUT
  MIX 4  TOTAL     6.96358740E-1 1.12379551E+0
         SCAT 2 2  2.55611958E-4 6.77430272E-1
              2 2  9.55488145E-1 3.16311372E-3
  ;
SYS := ASM: MACRO TRACK BCTRK ::
  ;
TRACK2 SYS2 := EXCELL: BC MACRO ::
  TITLE 'TCM04: TWO GROUPS CANDU 3-D ADJUSTER ROD ASSEMBLY '
  MAXR 40 TRAK NORM TISO 4 2.5
  ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE K
  ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 UPS  MERG COMP STAT DELS REFE 'NOROD'
  ;
FLUX := FLU: FLUX SYS2 MACRO TRACK2 ::
  TYPE K
  ;
EDITION2 := EDI: EDITION2 FLUX MACRO TRACK2 ::
  EDIT 3 UPS  MERG COMP STAT DELS REFE 'NOROD'
  ;
BCTRK := DELETE: BCTRK ;
END: ;
QUIT "LIST" .
```

*4.3.5 **TCM05** – Comparison of leakage models*

This test presents various homogeneous and heterogeneous leakage models on a simple cell (see Figure 20).[61]

Input data for test case: **TCM05**

```
*----
*  TEST CASE TCM05
*  MACROSCOPIC CROSS SECTIONS
*  FISSION SOURCE PROBLEM
```

```
*  2-D CARTESIAN/ANNULAR CELL
*  Validating leakage options
*    TYPE K B L FOR MOSTELC (NO VOID)
*    TYPE K B L FOR MOSTELCV (MOSTELC WITH VOID)
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  MOSTELC MOSTELCV TRACK MACRO SYS FLUX EDITION ;
SEQ_BINARY
  TRKSPC ;
MODULE
  GEO: EXCELT: MAC: ASM: FLU: EDI: DELETE: END: ;
*----
* Macroscopic XS
*----
MACRO := MAC: ::
 NGRO 2 ANIS 2 NMIX 2 NIFI 1
  READ INPUT
  MIX 1   TOTAL       0.222222  0.833333
    SCAT   1 1  0.19222 2 2  0.75333   0.02
          1 1  0.1     2 2  0.5       0.01
    NUSIGF      0.0       0.135  CHI         1.0       0.0
  MIX 2   TOTAL       0.166667  1.111111
    SCAT   2 2  0.00015   0.126667 2 2  1.10111   0.04
           2 2  0.0001    0.1      2 2  0.5       0.01
  ;
*----
*  Geometry : MOSTELC  - Cartesian 2D cell without void region
*             MOSTELCV - Cartesian 2D cell with void region
*----
MOSTELC := GEO: :: CARCEL 2
  X- REFL X+ REFL    Y- REFL Y+ REFL
  MESHX 0.0 1.26209  MESHY 0.0 1.26209
  RADIUS 0.0 0.39306 0.45802 SPLITR  2 1
  MIX 1 2 2  ;
MOSTELCV := GEO: MOSTELC ::
  MIX 1 0 2  ;
*----
*  Tracking : EXCELT - MOSTELC
*             ANIS 2 for adequate dimensions in PIJK
*  Solution : TYPE K, B or L
*  Leakage  : B1 PNL, B1 HETE
*----
TRACK TRKSPC := EXCELT: MOSTELC ::
  TITLE 'TCM05: ANNULAR GEOMETRY WITH MACROSCOPIC XS'
  MAXR 5 ANIS 2 TRAK TISO 12 20.0  ;
SYS := ASM: MACRO TRACK TRKSPC ::
  PIJK  ;
FLUX := FLU:       SYS MACRO TRACK ::
  TYPE K ;
```

```
EDITION := EDI:         FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE K B1 PNL BUCK 1.51429E-03 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 PNL KEFF 1.199538 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 PNL ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B1 PNL ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := DELETE: FLUX ;
FLUX := FLU:       SYS MACRO TRACK ::
  TYPE K B1 HETE BUCK 1.50298E-03 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 HETE KEFF 1.199538 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 HETE R BUCK Z 5.00993E-04 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 HETE Z BUCK R 1.001986E-03 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 HETE ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B1 HETE R BUCK Z 5.00993E-04 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B1 HETE Z BUCK R 1.001986E-03 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B1 HETE ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX SYS := DELETE: FLUX SYS ;
```

```
*----
*  Tracking : EXCELT - MOSTELCV only update TRACK TRKSPC files
*             since only change is in one material
*  Solution : TYPE K, B or L
*  Leakage  : B1 PNL, B1 HETE
*----
TRACK TRKSPC := EXCELT: TRACK TRKSPC MOSTELCV ::
  TITLE 'TCM05: ANNULAR GEOMETRY WITH MACROSCOPIC XS (VOID)'  ;
SYS := ASM: MACRO TRACK TRKSPC ::
  PIJK  ;
FLUX := FLU:       SYS MACRO TRACK ::
  TYPE K   ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE K B1 PNL BUCK 1.40181E-03 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 PNL KEFF 1.228007 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 PNL ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B1 PNL ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := DELETE: FLUX ;
FLUX := FLU:       SYS MACRO TRACK ::
  TYPE K B1 HETE BUCK 1.40181E-03 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 HETE KEFF 1.228007 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 HETE R BUCK Z 5.00993E-04 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 HETE Z BUCK R 1.001986E-03 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE B B1 HETE ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
```

```
  TYPE L B1 HETE R BUCK Z 5.00993E-04 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B1 HETE Z BUCK R 1.001986E-03 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B1 HETE ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 2 SAVE ;
TRKSPC := DELETE: TRKSPC ;
END: ;
QUIT "LIST" .
```

*4.3.6 **TCM06** – Buckling search without fission source*

This test is for an homogeneous water cell. A buckling eigenvalue problem is solved in the absence of fission source for the neutron flux distribution inside this cell.

Input data for test case: **TCM06**

```
*----
*  TEST CASE TCM06
*  MACROSCOPIC CROSS SECTIONS
*  BUCKLING SEARCH PROBLEM WITHOUT FISSION SOURCE
*  HOMOGENEOUS GEOMETRY
*
*  REF: none
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  WATER TRACK MACRO SYS FLUX EDITION ;
MODULE
 GEO: SYBILT: MAC: ASM: FLU: EDI: END: ;
*----
*  Macroscopic XS
*----
MACRO := MAC: ::
  EDIT 2 NGRO 1 ANIS 2 NMIX 1 NIFI 0
  READ INPUT
  MIX 1 TOTAL 3.59 SCAT 1 1 3.57 1 1 2.38
  ;
*----
*  Geometry :  WATER - Homogeneous geometry
*  Tracking : SYBILT
*----
WATER := GEO: :: HOMOGE
  MIX 1
```

```
  ;
TRACK := SYBILT: WATER ::
  TITLE 'TCM06: ENE6101 EXAM'
  MAXR 1
  ;
*----
*  Solution : TYPE  L
*  Leakage  : B0 PNL, P0 PNL, B1 PNL, P1 PNL
*----
SYS := ASM: MACRO TRACK :: ;
FLUX := FLU: SYS MACRO TRACK ::
  TYPE L B0 SIGS EXTE 5 1.0E-5 BUCK -0.07 ;
EDITION := EDI: FLUX MACRO TRACK ::
  EDIT 3 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L P0 SIGS EXTE 5 1.0E-5 BUCK -0.07 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L B1 SIGS EXTE 5 1.0E-5 BUCK -0.07 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE ;
FLUX := FLU: FLUX SYS MACRO TRACK ::
  TYPE L P1 SIGS EXTE 5 1.0E-5 BUCK -0.07 ;
EDITION := EDI: EDITION FLUX MACRO TRACK ::
  EDIT 3 SAVE ;
END: ;
QUIT "LIST" .
```

*4.3.7 **TCM07** – Test of boundary conditions*

This test is for a 2–D Cartesian cell with reflective and void boundary conditions.

Input data for test case: **TCM07**

```
*----
*  TEST CASE TCM07
*  MACROSCOPIC CROSS SECTIONS
*  FISSION SOURCE PROBLEM
*  2-D CARTESIAN CELL
*  REFLECTIVE AND VOID BOUNDARY CONDITIONS
*
*  REF: none
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  MACRO LATGEOR LATREGR SYSR FLUXR EDITR
        LATGEOV LATREGV SYSV FLUXV EDITV ;
SEQ_BINARY
  TRKR TRKV ;
```

```
MODULE
  MAC: GEO: EXCELT:  ASM: FLU: EDI: DELETE: END: ;
*----
* Macroscopic XS
*----
MACRO := MAC: ::
  NGRO 1 NMIX 2 NIFI 1
  READ INPUT
  MIX  1 TOTAL 0.75 SCAT 1 1 0.50 NUSIGF 1.00  CHI 1.0
  MIX  2 TOTAL 0.75 SCAT 1 1 0.50
  ;
*----
*  Geometry : LATGEOR - Cartesian 2D with reflection BC
*             LATGEOR - Cartesian 2D with void BC
*  Tracking : EXCELT
*----
LATGEOR  := GEO: :: CAR2D 2 2
  X- REFL X+ REFL MESHX 0.00 1.0 2.00 SPLITX  4 4
  Y- REFL Y+ REFL MESHY 0.00 1.0 2.00 SPLITY  4 4
  MIX  1  2  2  2  ;
LATGEOV  := GEO: LATGEOR  ::
  X- VOID X+ VOID Y- VOID Y+ VOID  ;
LATREGR TRKR := EXCELT: LATGEOR ::
  TITLE 'LATHROP *** P1 ANISOTROPE '
  MAXR 64 TRAK TISO 49 20.0  ;
LATREGV TRKV := EXCELT: LATGEOV ::
  TITLE 'LATHROP *** P1 ANISOTROPE '
  MAXR 64 TRAK TISO 49 20.0  ;
*----
*  Solution : TYPE K
*----
SYSR := ASM: MACRO LATREGR TRKR :: ;
FLUXR := FLU: SYSR MACRO LATREGR ::
  TYPE K ACCE 3 0 ;
EDITR := EDI: FLUXR MACRO LATREGR ::
  EDIT 1 ;
SYSV := ASM: MACRO LATREGV TRKV :: ;
FLUXV := FLU: SYSV MACRO LATREGV ::
  TYPE K ACCE 3 0 ;
EDITV := EDI: FLUXV MACRO LATREGV ::
  EDIT 1 ;
LATGEOR LATREGR SYSR FLUXR EDITR TRKR
   LATGEOV LATREGV SYSV FLUXV EDITV TRKV := DELETE:
LATGEOR LATREGR SYSR FLUXR EDITR TRKR
   LATGEOV LATREGV SYSV FLUXV EDITV TRKV ;
MACRO := DELETE: MACRO ;
END: ;
QUIT "LIST" .
```

4.3.8 ***TCM08*** *– Fixed source problem with fission*

This test is for a 2–D Cartesian cell which contains both a fission and a fixed source.

Input data for test case: **TCM08**

```
*----
*  TEST CASE TCM08
*  MACROSCOPIC CROSS SECTIONS
*  FIXED SOURCE PROBLEM WITH FISSILE MATERIAL
*  FOR 1/8 7X7 PWR ASSEMBLY
*
*  REF: TCM02
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  PWRF TRACF SYSF FLUXF EDITF PWRS TRACS SYSS FLUXS EDITS MACRO ;
SEQ_BINARY
  PWRTRKF PWRTRKS ;
MODULE
  GEO: EXCELT: MAC: ASM: FLU: EDI: DELETE: END: ;
*----
* Macroscopic XS
*----
MACRO   :=      MAC:    ::
  NGRO 1 NMIX 4 NIFI 1
  READ INPUT
  MIX 1 TOTAL  1.250  SCAT 1 1  1.242
        NUSIGF 0.15  CHI 1.0
  MIX 2 TOTAL  0.625  SCAT 1 1  0.355
        FIXE 0.000
  MIX 3 TOTAL 14.000  SCAT 1 1  0.000
        FIXE 1.000
  MIX 4 TOTAL  1.250  SCAT 1 1  1.242
        FIXE 0.000
  ;
*----
*  Geometry : PWRF  - Cartesian 2D assembly with fission
*             PWRS  - Cartesian 2D assembly without fission
*  Tracking : EXCELT
*----
PWRF    :=      GEO:    :: CAR2D 4 4
  X- DIAG  X+ REFL Y- SYME  Y+ DIAG
  CELL   P F F F
           F F F
             F F
               F
  ::: F := GEO: CARCEL 1
    RADIUS 0.000 0.450
    MIX 2  1
    MESHX -0.625  0.625 SPLITX 2
```

```
      MESHY -0.625  0.625 SPLITY 2  ;
    ::: P := GEO: F
      MIX  3  1
      SPLITR 3     ;
    ;
  PWRS    :=      GEO:    :: CAR2D 4 4
    X- DIAG  X+ REFL Y- SYME  Y+ DIAG
    CELL   P F F F
             F F F
               F F
                 F
    ::: F := GEO: CARCEL 1
      RADIUS 0.000 0.450
      MIX 2  4
      MESHX -0.625  0.625 SPLITX 2
      MESHY -0.625  0.625 SPLITY 2 ;
    ::: P := GEO: F
      MIX  3  4
      SPLITR 3     ;
    ;
  TRACF PWRTRKF := EXCELT: PWRF    ::
    TITLE 'TCM08: STANKOVSKI PWR ASSEMBLY'
    MAXR 58 TRAK TISO 12 8.0  ;
  SYSF := ASM: MACRO TRACF PWRTRKF :: ;
  TRACS   PWRTRKS :=      EXCELT: PWRS    ::
    TITLE 'TCM08: STANKOVSKI PWR ASSEMBLY'
    MAXR 58 TRAK TISO 12 8.0  ;
  SYSS := ASM: MACRO TRACS PWRTRKS :: ;
  *----
  *  Solution : TYPE K to test if k < 1.0
  *             TYPE S to include fixed source
  *----
  FLUXF := FLU: SYSF MACRO TRACF ::
    TYPE K ;
  EDITF := EDI: FLUXF MACRO TRACF ::
    EDIT 2 SAVE
    MERGE REGION
     1  1  1  2  3  4  3  4  5  6  5  6  7  8  7  8
     9 10  9 10  9 10 11 12 11 12 11 12 11 12 13 14 13 14 13 14 13 14
    15 16 15 16 15 16 17 18 17 18 17 18 17 18
    19 20 19 20 19 20  ;
  EDITF := DELETE: EDITF ;
  *----
  *  SINCE KEFF < 1  DO FIXED SOURCE PROBLEM
  *  (FIXED AND FISSION SOURCES TAKEN INTO ACCOUNT)
  *----
  FLUXF := FLU: FLUXF SYSF MACRO TRACF ::
    TYPE S ;
  EDITF := EDI: FLUXF MACRO TRACF ::
    EDIT 2 SAVE
    MERGE REGION
     1  1  1  2  3  4  3  4  5  6  5  6  7  8  7  8
```

```
    9 10  9 10  9 10 11 12 11 12 11 12 11 12 13 14 13 14 13 14 13 14
   15 16 15 16 15 16 17 18 17 18 17 18 17 18
   19 20 19 20 19 20  ;
*----
*  Solution : TYPE S only since no fission
*----
*
*  IF KEFF < 1  DO FIXED SOURCE PROBLEM PROBLEM
*  (FIXED AND FISSION SOURCES TAKEN INTO ACCOUNT)
*
FLUXS := FLU: SYSS MACRO TRACS ::
  TYPE S ;
EDITS := EDI: FLUXS MACRO TRACS ::
  EDIT 2 SAVE
  MERGE REGION
   1  1  1  2  3  4  3  4  5  6  5  6  7  8  7  8
   9 10  9 10  9 10 11 12 11 12 11 12 11 12 13 14 13 14 13 14 13 14
  15 16 15 16 15 16 17 18 17 18 17 18 17 18
  19 20 19 20 19 20  ;
PWRTRKS PWRTRKF := DELETE: PWRTRKS PWRTRKF ;
END: ;
QUIT "LIST" .
```

4.4 WIMS–D4 microscopic cross section examples

The test cases we will consider here use the `LIB:` module to enter microscopic cross sections taken from a WIMS–D4 69 groups library. We will assume that this library is located in file `WNEALIB`. The test cases are numbered successively from **TCWD01** to **TCWD12**.

4.4.1 ***TCWD01*** *– The Mosteller benchmark*

This benchmark uses both a cartesian 2–D cell with a central annular pin and an equivalent annular cell (see Figure 20).[61] No depletion information is required in this case since the module `EVO:` will not be executed. A comparison between various calculation options is provided here. We first consider the annular geometry with a `JPMT:` self-shielding and a `SYBILT:` transport calculation. This is then repeated for the cartesian 2–D cell. Finally, we used an isotropic (`TISO`) and a specular (`TSPC`) `EXCELT:` tracking successively for the self-shielding and transport calculations.

Input data for test case: **TCWD01**

```
*----
*  TEST CASE TCWD01
*  MOSTELLER BENCHMARK: 1-D ANNULAR CELL AND 2-D CARTESIAN CELL
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*
*  REF: R. Mosteller et al. Nucl. Sci. Eng. 107, 265 (1991)
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
```

```
  MOSTELA MOSTELC DISCR1 DISCR2 LIBRARY CP CALC OUT ;
SEQ_BINARY
  TRKSPC ;
MODULE
  LIB: GEO: JPMT: SYBILT: EXCELT: SHI: ASM: FLU: EDI:
  DELETE: END: ;
*----
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 3 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX 1 600.0                          O16      = '16'     4.61309E-2
    U235     = '235'    1.66078E-4 1 SHIB '235.4'
    U238     = '238'    2.28994E-2 1 SHIB '238.4'
  MIX 2 600.0
    Zr91     = '91'     3.83243E-2
  MIX 3 600.0
    H1H2O    = '2001'   4.42326E-2   O16H2O   = '16'     2.21163E-2
    BNat     = '11'     1.02133E-5
  ;
*----
*  Geometry MOSTELA : annular 3 region geometry
*           MOSTELC : Cartesian 3 region geometry
*----
MOSTELA := GEO: :: TUBE 3
  R+ REFL RADIUS 0.0 0.39306 0.45802 0.71206 SPLITR 2 1 1
  MIX 1 2 3 ;
MOSTELC := GEO: :: CARCEL 2
  X- REFL X+ REFL MESHX 0.0 1.26209
  Y- REFL Y+ REFL MESHY 0.0 1.26209
  RADIUS 0.0 0.39306 0.45802 SPLITR 2 1
  MIX  1 2 3 ;
*----
*  Case 1 -- annular
*  Self-Shielding calculation JPM
*  Transport calculation      SYBIL
*  Flux calculation for K no leakage
*----
DISCR1 := JPMT: MOSTELA ::
  TITLE 'TCWD01: MOSTELLER BENCHMARK (JPM / SYBIL)'
  MAXR 4 IP01 QUA1 5 ;
LIBRARY := SHI: LIBRARY DISCR1 :: ;
DISCR2 := SYBILT: MOSTELA ::
  TITLE 'TCWD01: MOSTELLER BENCHMARK (JPM / SYBIL)'
  MAXR 4 QUA1 5 ;
CP := ASM: LIBRARY DISCR2 :: ;
CALC := FLU: CP LIBRARY DISCR2 ::
  TYPE K ;
OUT := EDI: CALC LIBRARY DISCR2 ::
  EDIT 4 MERG MIX 1 2 3 COND 4.0 SAVE ;
DISCR1 DISCR2 CP := DELETE: DISCR1 DISCR2 CP ;
```

```
*----
*  Case 2 -- Cartesian
*  Self-Shielding calculation JPM
*  Transport calculation      SYBIL
*  Flux calculation for K no leakage
*----
DISCR1 := JPMT: MOSTELC ::
  TITLE 'TCWD01: MOSTELLER BENCHMARK (JPM / SYBIL)'
  MAXR 4 IP01 QUA1 5 QUA2 12 5 ;
LIBRARY := SHI: LIBRARY DISCR1 :: ;
DISCR2 := SYBILT: MOSTELC ::
  TITLE 'TCWD01: MOSTELLER BENCHMARK (JPM / SYBIL)'
  MAXR 4 QUA1 5 QUA2 12 5 ;
CP := ASM: LIBRARY DISCR2 :: ;
CALC := FLU: CALC CP LIBRARY DISCR2 ::
  TYPE K ;
OUT := EDI: OUT CALC LIBRARY DISCR2 ::
  EDIT 1 MERG MIX 1 2 3 COND 4.0 STAT ALL REFE 1 ;
DISCR1 DISCR2 CP := DELETE: DISCR1 DISCR2 CP ;
*----
*  Case 3 -- annular
*  Self-Shielding calculation EXCEL-ISO
*  Transport calculation      EXCEL-ISO
*  Flux calculation for K no leakage
*----
DISCR1 TRKSPC := EXCELT: MOSTELC ::
  TITLE 'TCWD01: MOSTELLER BENCHMARK (EXCELL)'
  MAXR 4 TRAK TISO 12 20.0 ;
LIBRARY := SHI: LIBRARY DISCR1 TRKSPC :: ;
CP := ASM: LIBRARY DISCR1 TRKSPC :: ;
CALC := FLU: CALC CP LIBRARY DISCR1 ::
  TYPE K ;
OUT := EDI: OUT CALC LIBRARY DISCR1 ::
  EDIT 1 MERG MIX 1 2 3 COND 4.0 STAT ALL REFE 1 ;
DISCR1 TRKSPC CP := DELETE: DISCR1 TRKSPC CP ;
*----
*  Case 4 -- Cartesian
*  Self-Shielding calculation EXCEL-SPC
*  Transport calculation      EXCEL-SPC
*  Flux calculation for K no leakage
*----
DISCR1 TRKSPC := EXCELT: MOSTELC ::
  TITLE 'TCWD01: MOSTELLER BENCHMARK (EXCELL)'
  MAXR 4 TRAK TSPC 12 20.0 ;
LIBRARY := SHI: LIBRARY DISCR1 TRKSPC :: ;
CP := ASM: LIBRARY DISCR1 TRKSPC :: ;
CALC := FLU: CALC CP LIBRARY DISCR1 ::
  TYPE K ;
OUT := EDI: OUT CALC LIBRARY DISCR1 ::
  EDIT 1 MERG MIX 1 2 3 COND 4.0 STAT ALL REFE 1 ;
DISCR1 TRKSPC CP := DELETE: DISCR1 TRKSPC CP ;
END: ;
```

```
QUIT "LIST" .
```

4.4.2 ***TCWD02*** *– A* 17 × 17 *PWR type assembly*

This test case represents a production calculation of a normal PWR assembly with cell grouping (`MERGE` and `TURN` options). Its configuration is shown in Figure 21.

Input data for test case: **TCWD02**

```
*----
*  TEST CASE TCWD02
*  17 X 17 PWR ASSEMBLY WITHOUT POISON
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*
*  REF: none
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  ASSMB DISCR1 DISCR2 LIBRARY CP CALC OUT COMPO ;
SEQ_ASCII
  RES ;
MODULE
  LIB: GEO: JPMT: SYBILT: SHI: ASM: FLU: EDI: CPO:
  DELETE: END: ;
*----
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 8 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX 1 579.9
    H1H2O    = '2001'   4.76690E-2   O16H2O   = '16'     2.38345E-2
    BNat     = '11'     2.38103E-5
  MIX 2 579.9                         O16      = '16'     3.06711E-4
    Cr52     = '52'     7.54987E-5   Fe56     = '56'     1.47624E-4
    Zr91     = '91'     4.18621E-2
  MIX 3 579.9
    H1H2O    = '2001'   4.65292E-2   O16H2O   = '16'     2.32646E-2
                                      Cr52     = '52'     4.79927E-5
    Fe56     = '56'     4.45845E-5   Ni58     = '58'     1.13521E-4
                                      Mo95     = '95'     4.03755E-6
                                      Al27     = '27'     2.35231E-6
    Mn55     = '55'     4.15901E-7
    BNat     = '11'     2.32761E-5
    Zr91     = '91'     8.92427E-4
  MIX 4 579.9                         O16      = '16'     2.87335E-4
    Cr52     = '52'     7.07291E-5   Fe56     = '56'     1.38298E-4
    Zr91     = '91'     3.92175E-2
  MIX 5 579.9
```

```
      H1H2O    = '2001'   4.71346E-2   O16H2O   = '16'     2.35673E-2
                                        Cr52     = '52'     2.24991E-5
      Fe56     = '56'     2.09013E-5   Ni58     = '58'     5.32188E-5
                                        Mo95     = '95'     1.89281E-6
                                        Al27     = '27'     1.10277E-6
      Mn55     = '55'     1.94976E-7
      BNat     = '11'     2.35598E-5
      Zr91     = '91'     4.18372E-4
    MIX 6 579.9
      H1H2O    = '2001'   4.71676E-2   O16H2O   = '16'     2.35838E-2
                                        Cr52     = '52'     2.11122E-5
      Fe56     = '56'     1.96130E-5   Ni58     = '58'     4.99383E-5
                                        Mo95     = '95'     1.77614E-6
                                        Al27     = '27'     1.03479E-6
      Mn55     = '55'     1.82957E-7
      BNat     = '11'     2.35753E-5
      Zr91     = '91'     3.92583E-4
    MIX 7 579.9
      H1H2O    = '2001'   4.72020E-2   O16H2O   = '16'     2.36010E-2
                                        Cr52     = '52'     1.96591E-5
      Fe56     = '56'     1.82630E-5   Ni58     = '58'     4.65011E-5
                                        Mo95     = '95'     1.65389E-6
                                        Al27     = '27'     9.63569E-7
      Mn55     = '55'     1.70365E-7
      BNat     = '11'     2.35914E-5
      Zr91     = '91'     3.65562E-4
    MIX 8 933.6                         O16      = '16'     4.49355E-2
      U235     = '235'    7.39237E-4 1 SHIB '235.4'
      U238     = '238'    2.17285E-2 1 SHIB '238.4'
    ;
  *----
  *  Geometry ASSMB : a 17 X 17 normal PWR assembly
  *  contains  C1 : cell without fuel
  *            C2 : normal fuel cell
  *            C3 : peripheral cell
  *            C4 : corner cell
  *----
  ASSMB := GEO: :: CAR2D 9 9
    X- DIAG X+ REFL Y- SYME Y+ DIAG
    CELL C1 C2 C2 C1 C2 C2 C1 C2 C3
            C2 C2 C2 C2 C2 C2 C2 C3
               C2 C2 C2 C2 C2 C2 C3
                  C1 C2 C2 C1 C2 C3
                     C2 C2 C2 C2 C3
                        C1 C2 C2 C3
                           C2 C2 C3
                              C2 C3
                                 C4
    MERGE 1  3 12 11 12 12 11 12 15
             4  6  5  6  6  5  6  8
               13  5  6  6  5  6  8
                   2  5  5 10  5  8
```

```
                    13  5  5  6  8
                        2  5  7  8
                           13  7  8
                              14  8
                                  9
  TURN  H  H  B  H  H  B  H  H  A
           H  G  G  H  G  G  H  A
              A  E  E  F  E  E  A
                 H  H  F  H  H  A
                    H  E  G  H  A
                       H  H  A  A
                          H  A  A
                             A  A
                                A
  ::: C1 := GEO: CARCEL 2
    MESHX 0.0 1.26472 MESHY 0.0 1.26472
    RADIUS 0.0 0.572435 0.613142  MIX 1 2 3 ;
  ::: C2 := GEO: C1  RADIUS 0.0 0.412660 0.474364  MIX 8 4 5 ;
  ::: C3 := GEO: C2  MESHX 0.0 1.31472 MIX 8 4 6 ;
  ::: C4 := GEO: C3  MESHY 0.0 1.31472  MIX 8 4 7 ;
  ;
*----
*  Self-Shielding calculation JPM
*  Transport calculation      SYBIL
*  Flux calculation for B1 homogeneous leakage
*  Editing using SPH model for transport-diffusion
*----
DISCR1 := JPMT: ASSMB ::
  TITLE 'TCWD02: 17 X 17 MULTICELL PWR BENCHMARK WITHOUT POISON'
  MAXR 400 OLD ;
LIBRARY := SHI: LIBRARY DISCR1 :: ;
DISCR2 := SYBILT: ASSMB ::
  TITLE 'TCWD02: 17 X 17 MULTICELL PWR BENCHMARK WITHOUT POISON'
  MAXR 400 ;
CP := ASM: LIBRARY DISCR2 :: ;
CALC := FLU: CP LIBRARY DISCR2 ::
  TYPE B B1 ;
OUT := EDI: CALC LIBRARY DISCR2 ASSMB ::
  EDIT 3 UPS SAVE COND 4.0 SPH
  ::: BIVACT: PRIM  1 2 EDIT 0 ;
  ;
COMPO := CPO: OUT ::
  EDIT 1 STEP 'REF-CASE   1' EXTRACT ALL NAME COMPO ;
RES := COMPO ;
END: ;
QUIT "LIST" .
```

4.4.3 ***TCWD03*** *– An hexagonal assembly*

This test case represents a production calculation of a typical hexagonal control assembly. Its configuration is presented in Figure 22.

Input data for test case: **TCWD03**

```
*----
*  TEST CASE TCWD03
*  MULTICELL HEXAGONAL ASSEMBLY WITH POISON
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*
*  REF: none
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  ASSMBH DISCR1 DISCR2 LIBRARY CP CALC OUT COMPO ;
SEQ_ASCII
  RES ;
MODULE
  GEO: JPMT: SYBILT: LIB: SHI: ASM: FLU: EDI: CPO:
  DELETE: END: ;
*----
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 11 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX 1 579.9
    H1H2O    = '2001'   4.76690E-2   O16H2O   = '16'     2.38345E-2
    BNat     = '11'     2.38103E-5
  MIX 2 579.9                         O16      = '16'     3.06711E-4
    Cr52     = '52'     7.54987E-5   Fe56     = '56'     1.47624E-4
    Zr91     = '91'     4.18621E-2
  MIX 3 579.9
    H1H2O    = '2001'   4.65292E-2   O16H2O   = '16'     2.32646E-2
                                      Cr52     = '52'     4.79927E-5
    Fe56     = '56'     4.45845E-5   Ni58     = '58'     1.13521E-4
                                      Mo95     = '95'     4.03755E-6
                                      Al27     = '27'     2.35231E-6
    Mn55     = '55'     4.15901E-7
    BNat     = '11'     2.32761E-5
    Zr91     = '91'     8.92427E-4
  MIX 4 933.6                         O16      = '16'     4.49355E-2
    U235     = '235'    7.39237E-4 1 SHIB '235.4'
    U238     = '238'    2.17285E-2 1 SHIB '238.4'
  MIX 5 579.9                         In115    = '115'    7.57464E-3
    Cd113    = '113'    2.62493E-3
    Ag109    = '109'    4.49188E-2
  MIX 6 579.9                         Cr52     = '52'     1.52702E-2
```

```
      Fe56     = '56'     5.57670E-2   Ni58     = '58'     7.51418E-3
      Mn55     = '55'     8.02943E-4
    MIX 7 579.9
      H1H2O    = '2001'   3.06466E-2   O16H2O   = '16'     1.53233E-2
      Fe56     = '56'     5.27485E-5   Cr52     = '52'     2.69769E-5
      BNat     = '11'     1.53077E-5
      Zr91     = '91'     1.49580E-2
    MIX 8 579.9
      H1H2O    = '2001'   4.65292E-2   O16H2O   = '16'     2.32646E-2
                                       Cr52     = '52'     4.79927E-5
      Fe56     = '56'     4.45845E-5   Ni58     = '58'     1.13521E-4
                                       Mo95     = '95'     4.03755E-6
                                       Al27     = '27'     2.35231E-6
      Mn55     = '55'     4.15901E-7
      BNat     = '11'     2.32761E-5
      Zr91     = '91'     8.92427E-4
    MIX 9 579.9                        O16      = '16'     2.87335E-4
      Cr52     = '52'     7.07291E-5   Fe56     = '56'     1.38298E-4
      Zr91     = '91'     3.92175E-2
    MIX 10 579.9
      H1H2O    = '2001'   4.71346E-2   O16H2O   = '16'     2.35673E-2
                                       Cr52     = '52'     2.24991E-5
      Fe56     = '56'     2.09013E-5   Ni58     = '58'     5.32188E-5
                                       Mo95     = '95'     1.89281E-6
                                       Al27     = '27'     1.10277E-6
      Mn55     = '55'     1.94976E-7   BNat     = '11'     2.35598E-5
      Zr91     = '91'     4.18372E-4
    MIX 11 579.9
      H1H2O    = '2001'   4.71676E-2   O16H2O   = '16'     2.35838E-2
                                       Cr52     = '52'     2.11122E-5
      Fe56     = '56'     1.96130E-5   Ni58     = '58'     4.99383E-5
                                       Mo95     = '95'     1.77614E-6
                                       Al27     = '27'     1.03479E-6
      Mn55     = '55'     1.82957E-7
      BNat     = '11'     2.35753E-5
      Zr91     = '91'     3.92583E-4
    ;
  *----
  *  Geometry ASSMBH :  hexagonal assembly with poison
  *  contains  C1 : cell without fuel
  *            C2 : poison cell
  *            C3 : normal fuel cell
  *            C4 : peripheral cell
  *----
  ASSMBH := GEO: :: HEX 36
    HBC S30 REFL
    CELL  C1 C3 C3 C3 C3 C3 C2 C3 C3 C3 C2 C3 C3 C3 C3 C3 C3 C2
          C3 C3 C2 C3 C3 C3 C3 C3 C3 C3 C3 C3 C4 C4 C4 C4 C4 C4
    TURN   A  A  A  A  A  A  A  A  B  D  A  I  A  C  F  J  B  A
           F  A  A  E  E  A  A  E  A  A  A  A  A  A  A  A  A  A
    MERGE  1  2  3  4  5  4  6  7  8  7  9  8 10  7  7  4  7 11
          12 13 14 15 12 16 17 12 16 18 18 19 20 21 21 22 22 23
```

```
    ::: C1 := GEO: HEXCEL 2
      SIDE 0.707297 RADIUS 0.0 0.412282 0.475917
      MIX 1 2 3 ;
    ::: C2 := GEO: HEXCEL 5
      SIDE 0.707297 RADIUS 0.0 0.25057 0.354359 0.436 0.486 0.6125
      MIX 5 5 5 6 7 8 ;
    ::: C3 := GEO: C1  MIX 4 9 10 ;
    ::: C4 := GEO: C3  MIX 4 9 11 ;
    ;
  *----
  *  Self-Shielding calculation JPM
  *  Transport calculation      SYBIL
  *  Flux calculation for B1 homogeneous leakage
  *  Editing using SPH model for transport-diffusion
  *----
  DISCR1 := JPMT: ASSMBH ::
    TITLE 'TCWD03: MULTICELL HEXAGONAL ASSEMBLY WITH POISON'
    MAXR 400 MAXZ 15000 OLD ;
  LIBRARY := SHI: LIBRARY DISCR1 :: ;
  DISCR2 := SYBILT: ASSMBH ::
    TITLE 'TCWD03: MULTICELL HEXAGONAL ASSEMBLY WITH POISON'
    MAXR 400 MAXZ 15000 ;
  CP := ASM: LIBRARY DISCR2 :: ;
  CALC := FLU: CP LIBRARY DISCR2 ::
    TYPE B B1 ;
  OUT := EDI: CALC LIBRARY DISCR2 ASSMBH ::
    EDIT 3 UPS SAVE COND 4.0 SPH
    ::: BIVACT: PRIM 1 2 1 ;
    ;
  COMPO := CPO: OUT ::
    STEP 'REF-CASE   1' EXTRACT ALL NAME COMPO ;
  RES := COMPO ;
  END: ;
  QUIT "LIST" .
```

*4.4.4 **TCWD04** – A Cylindrical cell with burnup*

This test case represents a burnup calculation for the mosteller annular geometry (see Figure 20).

Input data for test case: **TCWD04**

```
*----
*  TEST CASE TCWD04
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*  ANNULAR MOSTELLER BENCHMARK WITH BURNUP
*
*  REF: R. Mosteller et al. Nucl. Sci. Eng. 107, 265 (1991)
*
*----
*  Define variables
*----
```

```
INTEGER
  istep := 1 ;
REAL
  evobeg evoend ;
REAL
  step2 step3   step4   step5 :=
  1.0   27.1739 67.9348 135.8696 ;
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY MOSTELAS MOSTELA TRACKS TRACK SYS FLUX BURNUP
  EDITION COMPO ;
SEQ_ASCII
  RES ;
MODULE
  GEO: SYBILT: LIB: SHI: ASM: FLU: EVO: EDI: CPO:
  DELETE: END: ;
*----
*  Depletion data from file WNEALIB format WIMSD4
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 3 CTRA WIMS
  DEPL LIB: WIMSD4 FIL: WNEALIB
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX 1 600.0                          O16      = '16'     4.61309E-2
    U235     = '235'    1.66078E-4 1 SHIB '235.4'
    U238     = '238'    2.28994E-2 1 SHIB '238.4'
    U236     = '236'    0.0        1
    Pu239    = '3239'   0.0        1 SHIB '3239.1'
  MIX 2 600.0
    Zr91     = '91'     3.83243E-2
  MIX 3 600.0
    H1H2O    = '2001'   4.42326E-2   O16H2O   = '16'     2.21163E-2
    BNat     = '11'     1.02133E-5
  ;
*----
*  Geometry MOSTELAS : 3 regions annular cell for self-shielding
*           MOSTELA  : 4 regions annular cell for transport
*----
MOSTELAS := GEO: ::  TUBE 3
  R+ REFL RADIUS 0.0 0.39306 0.45802 0.71206
  MIX 1  2  3 ;
MOSTELA  := GEO:  MOSTELAS ::
  SPLITR 2  1  1 ;
*----
*  Self-Shielding calculation SYBIL
*  Transport calculation      SYBIL
*  Flux calculation for keff with imposed buckling
*  using B1 homogeneous leakage model
*----
```

```
TRACKS := SYBILT: MOSTELAS ::
  TITLE 'TCWD04: MOSTELLER BENCHMARK WITH BURNUP'
  EDIT 1 MAXR 3 ;
LIBRARY := SHI: LIBRARY TRACKS :: ;
TRACK := SYBILT: MOSTELA ::
  TITLE 'TCWD04: MOSTELLER BENCHMARK WITH BURNUP'
  EDIT 1 MAXR 4 ;
SYS := ASM: LIBRARY TRACK :: ;
FLUX := FLU: SYS LIBRARY TRACK ::
  TYPE K B1 PNL BUCK 0.2948E-2 ;
EDITION := EDI: FLUX LIBRARY TRACK ::
  EDIT 3 MERG COMP COND 4.0 SAVE  ;
*----
*  Burnup loop: for first step BURNUP is created
*  while for other steps it is modified
*  two burnup per step:
*  1) get a first approximation of final composition followed
*     by a transport calculation
*  2) use approximation for final flux distribution to get a
*     better approximation for final composition
*----
EVALUATE evoend := 0.0 ;
WHILE evoend step2 < DO
  EVALUATE evobeg := evoend ;
  EVALUATE evoend := step2 ;
  IF istep 1 = THEN
    BURNUP LIBRARY := EVO: LIBRARY FLUX TRACK ::
      SAVE <<evobeg>> DAY
      DEPL <<evobeg>> <<evoend>> DAY POWR 36.8
      SET <<evoend>> DAY ;
  ELSE
    BURNUP LIBRARY := EVO: BURNUP LIBRARY FLUX TRACK ::
      SAVE <<evobeg>> DAY
      DEPL <<evobeg>> <<evoend>> DAY POWR 36.8
      SET <<evoend>> DAY ;
  ENDIF ;
  LIBRARY := SHI: LIBRARY TRACKS :: ;
  SYS := DELETE: SYS ;
  SYS := ASM: LIBRARY TRACK :: ;
  FLUX := FLU: FLUX SYS LIBRARY TRACK ::
    TYPE K B1 PNL BUCK 0.2948E-2 ;
  BURNUP LIBRARY := EVO: BURNUP LIBRARY FLUX TRACK ::
    SAVE <<evoend>> DAY
    DEPL <<evobeg>> <<evoend>> DAY POWR 36.8
    SET <<evoend>> DAY ;
  LIBRARY := SHI: LIBRARY TRACKS :: ;
  SYS := DELETE: SYS ;
  SYS := ASM: LIBRARY TRACK :: ;
  FLUX := FLU: FLUX SYS LIBRARY TRACK ::
    TYPE K B1 PNL BUCK 0.2948E-2 ;
  EDITION := EDI: EDITION FLUX LIBRARY TRACK ::
    EDIT 3 SAVE  ;
```

```
    EVALUATE step2 step3 step4 step5 :=
             step3 step4 step5 step2 ;
    EVALUATE istep := istep 1 + ;
  ENDWHILE ;
  COMPO := CPO: EDITION BURNUP ::
    BURNUP REF-CASE NAME COMPO  ;
  RES := COMPO ;
  END: ;
  QUIT "LIST" .
```

*4.4.5 **TCWD05** – A CANDU–6 type annular cell with burnup*

This test case represents the typical CANDU type cell with an annular moderator region defined in Figure 23. Both its cross section and depletion data are taken from the same WIMS–D4 file. Depletion calculations are performed for 50 days at a fixed power.[20]

Input data for test case: **TCWD05**

```
*----
*  TEST CASE TCWD05
*  CANDU-6 ANNULAR CELL
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*  POWER (KW)          =  615.00000
*  BURN POWER (KW/KG) =   31.97130
*  URANIUM MASS        =   19.23600
*  UO2 REAL DENSITY    =   10.59300
*  UO2 EFF DENSITY     =   10.43750
*  UO2 TEMPERATURE     =  941.28998
*  ENRICHMENT          =    0.71140
*  COOLANT D2 AT %     =   99.222
*  MODERATOR D2 AT %   =   99.911
*  NUMBER OF DAYS      =   50
*
*  REF: R. Roy et al.  Ann. Nucl. Energy 21, 115 (1994)
*
*----
*  Define variables and initialize
*     Burnup paremeters
*     a) Power
*        =  31.9713 kw/kg for    0.0 to 300.0 days
*     b) Burnup time interval Delt
*        =     1 day  for    0 to    1 day
*        =     4 days for    1 to    5 days
*        =     5 days for    5 to   10 days
*        =    10 days for   10 to   50 days
*        =    20 days for   50 to  150 days
*        =    50 days for  150 to  300 days
*     c) Days with burnup interval changes
*        = 1.0, 5.0, 10.0, 50.0, 150.0 and 300.0 days
*     d) Burnup control time variables Timei, Timef
*        Timei = initial time
```

```
*        Timef = final time
*----
REAL
  Power    Delt Timec Timei Timef :=
  31.9713  1.0  1.0   0.0   0.0 ;
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY CANDU6S CANDU6F VOLMATS VOLMATF PIJ FLUX BURNUP EDITION
  COMPO1 COMPO2 ;
SEQ_BINARY
  INTLINS INTLINF ;
SEQ_ASCII
  RESFUEL RESMODE ;
MODULE
  GEO: EXCELT: LIB: SHI: ASM: FLU: EVO: EDI: CPO:
  DELETE: END: ;
*----
*  Depletion data from file WNEALIB format WIMSD4
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 10 CTRA WIMS
  DEPL LIB: WIMSD4 FIL: WNEALIB
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX  1  560.66   0.81212           O16      = '16'     7.99449E-1
    D2D2O    = '8002'   1.99768E-1   H1H2O    = '2001'   7.83774E-4
  MIX  2  560.66   6.57
    BNat     = '11'     2.10000E-4
    Zr91     = '91'     9.75000E+1
  MIX  3  345.66   0.0014            He4      = '4'      1.00000E+2
  MIX  4  345.66   6.44              Fe56     = '56'     1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '11'     3.10000E-4
    Zr91     = '91'     9.97100E+1
  MIX  5  345.66   1.082885          O16      = '16'     7.98895E-1
    D2D2O    = '8002'   2.01016E-1   H1H2O    = '2001'   8.96000E-5
  MIX  6  941.29  10.4375010         O16      = '16'     1.18473E+1
    Xe135    = '135'    0.0
    U235     = '235'    6.27118E-1 1 SHIB '235.4'
    U238     = '238'    8.75256E+1 1 SHIB '238.4'
    U236     = '236'    0.0        1
    Pu239    = '3239'   0.0        1 SHIB '3239.1'
  MIX  7  COMB 6 1.0
  MIX  8  COMB 6 1.0
  MIX  9  COMB 6 1.0
  MIX 10   560.66   6.44             Fe56     = '56'     1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '11'     3.10000E-4
    Zr91     = '91'     9.97100E+1
  ;
```

```
*----
*  Geometry CANDU6S : 13 regions annular cluster for self-shielding
*           CANDU6F : 31 regions annular cluster for transport
*----
CANDU6S := GEO: ::  TUBE 5
  R+ REFL RADIUS  0.00000 5.16890 5.60320  6.44780 6.58750 16.12171
  MIX  1 2 3 4 5
  CLUSTER ROD1 ROD2 ROD3 ROD4
  ::: ROD1 := GEO: TUBE 2 MIX 6 10 NPIN  1 RPIN 0.0000 APIN 0.0000
    RADIUS 0.00000 0.6122 0.6540 ;
  ::: ROD2 := GEO: ROD1   MIX 7 10 NPIN  6 RPIN 1.4885 APIN 0.0000 ;
  ::: ROD3 := GEO: ROD1   MIX 8 10 NPIN 12 RPIN 2.8755 APIN 0.261799 ;
  ::: ROD4 := GEO: ROD1   MIX 9 10 NPIN 18 RPIN 4.3305 APIN 0.0 ;
  ;
CANDU6F := GEO: CANDU6S :: SPLITR  6 1 1 1 10
  ::: ROD1 := GEO: ROD1 SPLITR 2 1 ;
  ::: ROD2 := GEO: ROD2 SPLITR 2 1 ;
  ::: ROD3 := GEO: ROD3 SPLITR 2 1 ;
  ::: ROD4 := GEO: ROD4 SPLITR 2 1 ;
  ;
*----
*  Self-Shielding calculation EXCEL
*  Transport calculation      EXCEL
*  Flux calculation for keff
*----
VOLMATS INTLINS := EXCELT: CANDU6S ::
  TITLE 'TCWD05: CANDU-6 ANNULAR POWER= 31.971 FUEL TEMP= 941.29'
  EDIT 0 MAXR 13 TRAK TISO 5 10.0 SYMM 12 ;
LIBRARY := SHI: LIBRARY VOLMATS INTLINS ::
  EDIT 0 ;
VOLMATF INTLINF := EXCELT: CANDU6F ::
  TITLE 'TCWD05: CANDU-6 ANNULAR POWER= 31.971 FUEL TEMP= 941.29'
  EDIT 0 MAXR 31 TRAK TISO 5 10.0 SYMM 12 ;
PIJ := ASM: LIBRARY VOLMATF INTLINF :: ;
FLUX := FLU: PIJ LIBRARY VOLMATF ::
  TYPE K ;
EDITION := EDI: FLUX LIBRARY VOLMATF ::
  COND 4.0 MERGE MIX 0 0 0 0 1 0 0 0 0 0 SAVE ON 'EDITMOD' ;
EDITION := EDI: EDITION FLUX LIBRARY VOLMATF ::
  COND 4.0 MERGE COMP MICR 1 Xe135 SAVE ;
*----
*  Burnup loop: for first step BURNUP is created
*  while for other steps it is modified
*----
WHILE Timei Timec < DO
  EVALUATE Timef := Timei Delt + ;
  IF Timei 0.0 = THEN
    BURNUP LIBRARY := EVO: LIBRARY FLUX VOLMATF ::
      DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
  ELSE
    BURNUP LIBRARY := EVO: BURNUP LIBRARY FLUX VOLMATF ::
      DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
```

```
    ENDIF ;
    LIBRARY := SHI: LIBRARY VOLMATS INTLINS ::
      EDIT 0 ;
    PIJ := DELETE: PIJ ;
    PIJ := ASM: LIBRARY VOLMATF INTLINF :: ;
    FLUX := FLU: FLUX PIJ LIBRARY VOLMATF ::
      TYPE K ;
    EDITION := EDI: EDITION FLUX LIBRARY VOLMATF ::
      SAVE  ;
  *----
  *  change delta t for burnup and final time if required
  *----
    IF Timef Timec = THEN
      IF Timec 150.0 = THEN
        EVALUATE Delt Timec := 50.0 300.0 ;
      ENDIF ;
      IF Timec 50.0 = THEN
        EVALUATE Delt Timec := 20.0 150.0 ;
      ENDIF ;
      IF Timec 10.0 = THEN
        EVALUATE Delt Timec := 10.0 50.0 ;
      ENDIF ;
      IF Timec 5.0 = THEN
        EVALUATE Delt Timec := 5.0 10.0 ;
      ENDIF ;
      IF Timec 1.0 = THEN
        EVALUATE Delt Timec := 4.0 5.0 ;
      ENDIF ;
    ENDIF ;
    EVALUATE Timei := Timef ;
  ENDWHILE ;
  *----
  *  Save calculation results in CPO format file
  *----
  COMPO1 := CPO: BURNUP EDITION ::
    BURNUP REF-CASE EXTRACT Xe135 Xe135  NAME MIXTRXE ;
  RESFUEL := COMPO1 ;
  COMPO2 := CPO: EDITION ::
    STEP 'EDITMOD' NAME MIXTMOD ;
  RESMODE := COMPO2 ;
  INTLINF INTLINS := DELETE: INTLINF INTLINS ;
  END: ;
  QUIT "LIST" .
```

*4.4.6 **TCWD06** – A CANDU–6 type supercell with control rods*

This test case treats both the CANDU cell with a cartesian moderator region (similar to the cell described in defined Figure 23) and the supercell containing a stainless steel rod which can be either in the inserted or extracted position (see Figure 19). Two groups incremental cross sections corresponding to the rod in the inserted and extracted position with respect to the original supercell containing only 3–D fuel elements are computed.[20]

Input data for test case: **TCWD06**

```
*----
*  TEST CASE TCWD06
*  CANDU-6 CARTESIAN CELL
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*  STAINLESS STELL RODS IN 3D SUPERCELL
*
*  REF: R. Roy et al.  Ann. Nucl. Energy 21, 115 (1994)
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY LIBRARY2 CANDU6F CANDU6S TRACK SYS FLUX EDITION BCO BCI ;
SEQ_BINARY
  INTLIN ;
MODULE
  GEO: JPMT: EXCELT: LIB: SHI: ASM: FLU: EDI:
  DELETE: UTL: END: ;
*----
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  EDIT 0 NMIX 10 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX  1  560.66   0.81212           O16      = '16'     7.99449E-1
    D2D2O    = '8002'   1.99768E-1   H1H2O    = '2001'   7.83774E-4
  MIX  2  560.66   6.57
    BNat     = '11'     2.10000E-4
    Zr91     = '91'     9.75000E+1
  MIX  3  345.66   0.0014            He4      = '4'      1.00000E+2
  MIX  4  345.66   6.44              Fe56     = '56'     1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '11'     3.10000E-4
    Zr91     = '91'     9.97100E+1
  MIX  5  345.66   1.082885          O16      = '16'     7.98895E-1
    D2D2O    = '8002'   2.01016E-1   H1H2O    = '2001'   8.96000E-5
  MIX  6  941.29  10.4375010         O16      = '16'     1.18473E+1
    U235     = '235'    6.27118E-1 1 SHIB '235.4'
    U238     = '238'    8.75256E+1 1 SHIB '238.4'
  MIX  7  COMB 6 1.0
  MIX  8  COMB 6 1.0
  MIX  9  COMB 6 1.0
  MIX 10   560.66   6.44              Fe56     = '56'     1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '11'     3.10000E-4
    Zr91     = '91'     9.97100E+1
 ;
*----
*  CELL CALCULATION
*  Geometry CANDU6S : 14 regions Cartesian cluster for self-shielding
```

```
*           CANDU6F : 32 regions Cartesian cluster for transport
*           BCO     : 48 regions 3D Cartesian geometry
*           BCI     : 48 regions 3D Cartesian geometry
*----
CANDU6S := GEO: ::  CARCEL  5
  X+ REFL X- REFL MESHX -14.2875 14.2875
  Y+ REFL Y- REFL MESHY -14.2875 14.2875
  RADIUS  0.00000 5.16890 5.60320  6.44780 6.58750 14.00
  MIX  1 2 3 4 5 5
  CLUSTER ROD1 ROD2 ROD3 ROD4
  ::: ROD1 := GEO: TUBE 2 MIX 6 10 NPIN  1 RPIN 0.0000 APIN 0.0000
    RADIUS 0.00000 0.6122 0.6540 ;
  ::: ROD2 := GEO: ROD1   MIX 7 10 NPIN  6 RPIN 1.4885 APIN 0.0000 ;
  ::: ROD3 := GEO: ROD1   MIX 8 10 NPIN 12 RPIN 2.8755 APIN 0.261799 ;
  ::: ROD4 := GEO: ROD1   MIX 9 10 NPIN 18 RPIN 4.3305 APIN 0.0 ;
  ;
CANDU6F := GEO: CANDU6S :: SPLITR  6 1 1 1 10
  ::: ROD1 := GEO: ROD1 SPLITR 2 1 ;
  ::: ROD2 := GEO: ROD2 SPLITR 2 1 ;
  ::: ROD3 := GEO: ROD3 SPLITR 2 1 ;
  ::: ROD4 := GEO: ROD4 SPLITR 2 1 ;
  ;
*----
*  Self-Shielding calculation EXCEL
*  Transport calculation      EXCEL
*  Flux calculation for keff
*----
TRACK INTLIN := EXCELT: CANDU6S ::
  TITLE 'TCWD06: CANDU-6 CARTESIAN FUEL TEMP= 941.29'
  EDIT 0 MAXR 14 TRAK TISO 29 20.0 SYMM 4 ;
LIBRARY := SHI: LIBRARY TRACK INTLIN ::
  EDIT 0 ;
TRACK INTLIN := DELETE: TRACK INTLIN ;
TRACK INTLIN := EXCELT: CANDU6F ::
  TITLE 'TCWD06: CANDU-6 CARTESIAN FUEL TEMP= 941.29'
  EDIT 0 MAXR 32 TRAK TISO 29 20.0 SYMM 4 ;
SYS := ASM: LIBRARY TRACK INTLIN ::
  EDIT 0 ;
FLUX := FLU: SYS LIBRARY TRACK ::
  TYPE K ;
*----
*  Microscopic cross sections from WIMSLIB for stainless steel
*  MIX 1,2,3 from EDI: : fuel, structure material and moderator
*  MIX 4 stainless steel rods
*----
EDITION := EDI: FLUX LIBRARY TRACK ::
  EDIT 0 MERGE MIX 1 2 2 2 3 1 1 1 1 1 SAVE ON SSRODS ;
EDITION := UTL: EDITION :: STEP UP SSRODS ;
LIBRARY2 := LIB: EDITION ::
  EDIT 0 NMIX 4 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX 4    345.66                   Fe56     = '56'     6.19027E-2
```

```
     Cr52     = '52'     1.56659E-3   Ni58     = '58'     6.83337E-3
     Si29     = '29'     7.79072E-4   C12      = '12'     1.46552E-4
     Mn55     = '55'     1.25431E-3
   ;
EDITION TRACK INTLIN SYS FLUX := DELETE:
  EDITION TRACK INTLIN SYS FLUX ;
*----
*  SUPERCELL CALCULATION
*  Geometry BCO     : 27 regions 3D Cartesian geometry with rods out
*           BCI     : 27 regions 3D Cartesian geometry with rods in
*----
BCO := GEO: :: CAR3D 3 2 2
  X- REFL X+ SYME Y- REFL Y+ SYME Z- REFL Z+ SYME
  CELL  M   MX  MX MX  FXY MXY  M   MX  BX MX  FXY BXY
  TURN  A   A   A  F   A   A    A   A   A  F   A   A
  ::: M := GEO: CAR3D 1 1 1 MIX 3
    MESHX  0.0 7.14375
    MESHY  0.0 7.14375
    MESHZ -8.25500 8.25500 SPLITZ 2  ;
  ::: MX := GEO: M  MESHX -7.14375 +7.14375 SPLITX 2 ;
  ::: MXY := GEO: MX MESHY -7.14375 +7.14375 SPLITY 2 ;
  ::: BX := GEO: CARCELY 2 1 MIX 3 3 3
    MESHX -7.14375 7.14375 SPLITX  2
    MESHY 0.0 7.14375
    MESHZ -8.25500 8.25500 SPLITZ 2
    RADIUS 0.0 3.5100 3.8100 ;
  ::: BXY := GEO: BX MESHY -7.14375 +7.14375 SPLITY 2 ;
  ::: FXY := GEO: CARCELZ 2 1 MIX 1  2  3
    MESHX -7.14375 7.14375 SPLITX 2
    MESHY -7.14375 7.14375 SPLITY 2
    MESHZ -8.25500 8.25500 SPLITZ 2
    RADIUS 0.0 5.16890 6.58750 ;
  ;
BCI := GEO: BCO ::
  ::: BX := GEO: BX  MIX 3 4 3 ;
  ::: BXY := GEO: BXY MIX 3 4 3 ;
  ;
*----
*  Transport calculation      EXCEL
*  Flux calculation for keff
*  Homogenized properties for rod out
*----
TRACK INTLIN := EXCELT: BCO ::
  EDIT 0 MAXR 40 TRAK TISO 2 1.0 ;
SYS := ASM: LIBRARY2 TRACK INTLIN ::
  EDIT 0 ;
FLUX := FLU: SYS LIBRARY2 TRACK ::
  TYPE K ;
EDITION := EDI: FLUX LIBRARY2 TRACK ::
  EDIT 2 MERG COMP COND 4.0 SAVE ON  'NOBC' ;
SYS TRACK INTLIN := DELETE: SYS TRACK INTLIN ;
*----
```

```
*  Transport calculation      EXCEL
*  Flux calculation for keff
*  Homogenized properties for rod in
*----
TRACK INTLIN := EXCELT: BCI ::
  EDIT 0 MAXR 40 TRAK TISO 2 1.0 ;
SYS := ASM: LIBRARY2 TRACK INTLIN ::
  EDIT 0 ;
FLUX := FLU: FLUX SYS LIBRARY2 TRACK ::
  TYPE K ;
EDITION := EDI: EDITION FLUX LIBRARY2 TRACK ::
  EDIT 2 MERG COMP COND 4.0 STAT DELS REFE 'NOBC' ;
TRACK INTLIN SYS := DELETE: TRACK INTLIN SYS ;
END: ;
QUIT "LIST" .
```

*4.4.7 **TCWD07** – A CANDU–6 type calculation using various leakage options*

This test case treats the CANDU cell with a cartesian moderator region (similar to the cell described in defined Figure 23) using various leakage options.

Input data for test case: **TCWD07**

```
*----
*  TEST CASE TCWD07
*  CANDU-6 CARTESIAN CELL
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*  TEST VARIOUS LEAKAGE OPTIONS
*
*  REF: R. Roy et al.  Ann. Nucl. Energy 21, 115 (1994)
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY CANDU6S CANDU6T CANDU6SV CANDU6TV TRACK
  SYS FLUX EDITION ;
MODULE
  GEO: EXCELT: LIB: SHI: ASM: FLU: EDI:
  DELETE: END: ;
SEQ_BINARY
  INTLIN ;
*----
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  EDIT 0 NMIX 10 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX  1  560.66   0.81212           O16      = '16'     7.99449E-1
    D2D2O    = '8002'   1.99768E-1   H1H2O    = '2001'   7.83774E-4
  MIX  2  560.66   6.57
```

```
    BNat     = '11'     2.10000E-4
    Zr91     = '91'     9.75000E+1
  MIX  3  345.66   0.0014           He4      = '4'      1.00000E+2
  MIX  4  345.66   6.44             Fe56     = '56'     1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '11'     3.10000E-4
    Zr91     = '91'     9.97100E+1
  MIX  5  345.66   1.082885         O16      = '16'     7.98895E-1
    D2D2O    = '8002'   2.01016E-1   H1H2O    = '2001'   8.96000E-5
  MIX  6  941.29  10.4375010        O16      = '16'     1.18473E+1
    U235     = '235'    6.27118E-1 1 SHIB '235.4'
    U238     = '238'    8.75256E+1 1 SHIB '238.4'
  MIX  7  COMB 6 1.0
  MIX  8  COMB 6 1.0
  MIX  9  COMB 6 1.0
  MIX 10   560.66   6.44             Fe56     = '56'     1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '11'     3.10000E-4
    Zr91     = '91'     9.97100E+1
  ;
*----
*  Geometry CANDU6S : GEOMETRY FOR SELF-SHIELDING  (NO VOID)
*           CANDU6F : GEOMETRY FOR TRANSPORT       (NO VOID)
*           CANDU6FV: GEOMETRY FOR TRANSPORT       (COOLANT VOID)
*           CANDU6FV: GEOMETRY FOR TRANSPORT       (COOLANT VOID)
*----
CANDU6S := GEO: :: CARCEL  5
  X+ REFL X- REFL MESHX -14.2875 14.2875
  Y+ REFL Y- REFL MESHY -14.2875 14.2875
  RADIUS  0.00000 5.16890 5.60320 6.44780 6.58750 14.00
  MIX     1       2       3       4       5       5
  CLUSTER ROD1 ROD2 ROD3 ROD4
  ::: ROD1 := GEO: TUBE 2 MIX  6 10 NPIN  1 RPIN 0.0000 APIN 0.0000
    RADIUS 0.00000 0.6122 0.6540 ;
  ::: ROD2 := GEO: ROD1   MIX  7 10 NPIN  6 RPIN 1.4885 APIN 0.0000   ;
  ::: ROD3 := GEO: ROD1   MIX  8 10 NPIN 12 RPIN 2.8755 APIN 0.261799 ;
  ::: ROD4 := GEO: ROD1   MIX  9 10 NPIN 18 RPIN 4.3305 APIN 0.0      ;
  ;
CANDU6T := GEO: CANDU6S :: SPLITR  6 1 1  1 10
  ::: ROD1 := GEO: ROD1 SPLITR 2 1 ;
  ::: ROD2 := GEO: ROD2 SPLITR 2 1 ;
  ::: ROD3 := GEO: ROD3 SPLITR 2 1 ;
  ::: ROD4 := GEO: ROD4 SPLITR 2 1 ;
  ;
CANDU6SV := GEO: CANDU6S  :: MIX 0 2 3 4 5 5 ;
CANDU6TV := GEO: CANDU6SV :: SPLITR  6 1 1  1 10
  ::: ROD1 := GEO: ROD1 SPLITR 2 1 ;
  ::: ROD2 := GEO: ROD2 SPLITR 2 1 ;
  ::: ROD3 := GEO: ROD3 SPLITR 2 1 ;
  ::: ROD4 := GEO: ROD4 SPLITR 2 1 ;
  ;
*----
```

```
*  CASE WITH NO VOID
*  Self-Shielding calculation EXCEL
*  Transport calculation      EXCEL
*  Flux TYPE K AND B WITH VARIOUS LEAKAGE OPTIONS
*----
TRACK INTLIN := EXCELT: CANDU6S ::
  TITLE 'TCWD07: CANDU-6 CARTESIAN FUEL TEMP= 941.29'
  EDIT 0 MAXR 14 TRAK TISO 7 20.0 SYMM 4 ;
LIBRARY := SHI: LIBRARY TRACK INTLIN ::
  EDIT 0 ;
TRACK INTLIN := DELETE: TRACK INTLIN ;
TRACK INTLIN := EXCELT: CANDU6T ::
  TITLE 'TCWD07: CANDU-6 CARTESIAN FUEL TEMP= 941.29'
  EDIT 0 MAXR 32 ANIS 2 TRAK TISO 7 20.0 SYMM 4 ;
SYS := ASM: LIBRARY TRACK INTLIN ::
  EDIT 0 PIJK ;
FLUX := FLU: SYS LIBRARY TRACK ::
  TYPE K ;
EDITION := EDI: FLUX LIBRARY TRACK ::
  EDIT 3 SAVE COND 4.0 TAKE REGI 1 4 7 10 16 24 ;
FLUX := FLU: FLUX SYS LIBRARY TRACK ::
  TYPE B B1 PNL ;
EDITION := EDI: EDITION FLUX LIBRARY TRACK ::
  EDIT 3 SAVE COND 4.0  TAKE REGI 1 4 7 10 16 24 ;
FLUX := FLU: FLUX SYS LIBRARY TRACK ::
  TYPE B B1 HETE ;
EDITION := EDI: EDITION FLUX LIBRARY TRACK ::
  EDIT 3 SAVE COND 4.0 TAKE REGI 1 4 7 10 16 24 ;
FLUX SYS TRACK INTLIN := DELETE: FLUX SYS TRACK INTLIN ;
*----
*  CASE WITH COOLANT VOIDED
*  Self-Shielding calculation EXCEL
*  Transport calculation      EXCEL
*  Flux TYPE K AND B WITH VARIOUS LEAKAGE OPTIONS
*----
TRACK INTLIN := EXCELT: CANDU6SV ::
  TITLE 'TCWD07: CANDU-6 CARTESIAN FUEL TEMP= 941.29'
  EDIT 0 MAXR 14 TRAK TISO 7 20.0 SYMM 4 ;
LIBRARY := SHI: LIBRARY TRACK INTLIN ::
  EDIT 0 ;
TRACK INTLIN := DELETE: TRACK INTLIN ;
TRACK INTLIN := EXCELT: CANDU6TV ::
  TITLE 'TCWD07: CANDU-6 CARTESIAN FUEL TEMP= 941.29'
  EDIT 0 MAXR 32 ANIS 2 TRAK TISO 7 20.0 SYMM 4 ;
SYS := ASM: LIBRARY TRACK INTLIN ::
  EDIT 0 PIJK ;
FLUX := FLU: SYS LIBRARY TRACK ::
  TYPE K ;
EDITION := EDI: EDITION FLUX LIBRARY TRACK ::
  EDIT 3 SAVE COND 4.0 TAKE REGI 1 4 7 10 16 24 ;
FLUX := FLU: FLUX SYS LIBRARY TRACK ::
  TYPE B B1 PNL ;
```

```
EDITION := EDI: EDITION FLUX LIBRARY TRACK ::
  EDIT 3 SAVE COND 4.0  TAKE REGI 1 4 7 10 16 24 ;
FLUX := FLU: FLUX SYS LIBRARY TRACK ::
  TYPE B B1 HETE ;
EDITION := EDI: EDITION FLUX LIBRARY TRACK ::
  EDIT 3 SAVE COND 4.0 TAKE REGI 1 4 7 10 16 24 ;
FLUX SYS TRACK INTLIN := DELETE: FLUX SYS TRACK INTLIN ;
END: ;
QUIT "LIST" .
```

*4.4.8 **TCWD08** – Burnup of an homogeneous cell*

This case illustrate the burnup of an homogeneous cell that spends the first 1000 days in a reactor before being removed. The depletion of the isotopes in this cell for an additional 1000 days outside of the core is also investigated.

Input data for test case: **TCWD08**

```
*----
*  TEST CASE TCWD08
*  HOMOGENEOUS DEPLETION CASE
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*
*  REF: None
*
*----
*  Define variables
*     Burnup paremeters
*     a) Power
*        =  600.0 kw/kg for    0.0 to 1000.0 days
*        =    0.0 kw/kg for 1000.0 to 2000.0 days
*     b) Burnup time interval Delt
*        =    10 days for    0 to   50 days
*        =    50 days for   50 to  500 days
*        =   100 days for  500 to 1000 days
*        =  1000 days for 1000 to 2000 days
*     c) Editing time Timec
*        = 0.0, 50.0, 500.0, 1000.0 and 2000.0 days
*     d) Burnup control time variables Timei, Timef
*        Timei = initial time
*        Timef = final time
*     d) Print variable Iprint
*        = 1 reduced print
*        = 3 full print
*----
REAL
  Power  Delt  Timec Timei Timef :=
  600.0  10.0  50.0  0.0   0.0 ;
INTEGER
  Iprint :=  1 ;
*----
```

```
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY HOM TRACK PIJ FLUX BURNUP EDITION ;
MODULE
  GEO: SYBILT: LIB: SHI: ASM: FLU: EVO: EDI:
  DELETE: END: ;
*----
*  Depletion data from file WNEALIB format WIMSD4
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 1 CTRA WIMS
  DEPL LIB: WIMSD4 FIL: WNEALIB
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX 1 300.0                          H1       = '2001'   2.00000E+1
    U235     = '235'    1.0        1 SHIB '235.4'
    U236     = '236'    0.0        1
  ;
*----
*  Geometry HOM : Homogeneous geometry
*----
HOM := GEO: :: HOMOGE
  MIX 1 ;
*----
*  Self-Shielding calculation SYBIL
*  Transport calculation      SYBIL
*  Flux calculation for keff
*----
TRACK := SYBILT: HOM  ::
  TITLE 'TCWD08: HOMOGENEOUS BENCHMARK WITH BURNUP' ;
LIBRARY := SHI: LIBRARY TRACK :: ;
PIJ := ASM: LIBRARY TRACK :: ;
FLUX := FLU: PIJ LIBRARY TRACK ::
  TYPE K ;
EDITION := EDI: FLUX LIBRARY TRACK ::
  COND 4.0 MERGE COMP SAVE ;
*----
*  Burnup loop: for first step BURNUP is created
*  while for other steps it is modified
*----
WHILE Timei Timec < DO
  EVALUATE Timef := Timei Delt + ;
  IF Timef Timec = THEN
    EVALUATE Iprint := 3 ;
  ELSE
    EVALUATE Iprint := 1 ;
  ENDIF ;
  IF Timei 0.0 = THEN
    BURNUP LIBRARY := EVO: LIBRARY FLUX TRACK ::
      EDIT <<Iprint>> DEPL <<Timei>> <<Timef>> DAY
      POWR <<Power>> ;
```

```
    ELSE
      BURNUP LIBRARY := EVO: BURNUP LIBRARY FLUX TRACK ::
        EDIT <<Iprint>> DEPL <<Timei>> <<Timef>> DAY
        POWR <<Power>> ;
    ENDIF ;
    LIBRARY := SHI: LIBRARY TRACK :: EDIT 0 ;
    PIJ := DELETE: PIJ ;
    PIJ := ASM: LIBRARY TRACK :: ;
    FLUX := FLU: FLUX PIJ LIBRARY TRACK ::
      TYPE K ;
    IF Iprint  3 = THEN
      EDITION := EDI: EDITION FLUX  LIBRARY TRACK ::
        SAVE ;
    ENDIF ;
  *----
  *  change delta t for burnup, final time and power if required
  *----
    IF Timef Timec = THEN
      IF Timec 1000.0 = THEN
        EVALUATE Power Delt Timec := 0.0 1000.0 2000.0 ;
      ENDIF ;
      IF Timec 500.0 = THEN
        EVALUATE Delt Timec := 100.0 1000.0 ;
      ENDIF ;
      IF Timec 50.0 = THEN
        EVALUATE Delt Timec := 50.0 500.0 ;
      ENDIF ;
    ENDIF ;
    EVALUATE Timei := Timef ;
  ENDWHILE ;
  END: ;
  QUIT "LIST" .
```

*4.4.9 **TCWD09** – Testing boundary conditions*

This case test different boundary conditions for the Mosteller cell (see Figure 20).

Input data for test case: **TCWD09**

```
*----
*  TEST CASE TCWD09
*  MOSTELLER BENCHMARK FOR 1-D ANNULAR CELL
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*  REFLECTIVE AND VOID BC
*
*  REF: None
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
```

```
  MOSTELA MOSTELV VOLMAT LIBRARY PIJ FLUX OUT ;
MODULE
  LIB: GEO: JPMT: SYBILT: SHI: ASM: FLU: EDI:
  DELETE: END: ;
*----
*  Microscopic cross section from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 3 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX 1 600.0                          O16      = '16'     4.61309E-2
    U235     = '235'    1.66078E-4 1 SHIB '235.4'
    U238     = '238'    2.28994E-2 1 SHIB '238.4'
  MIX 2 600.0
    Zr91     = '91'     3.83243E-2
  MIX 3 600.0
    H1H2O    = '2001'   4.42326E-2   O16H2O   = '16'     2.21163E-2
    BNat     = '11'     1.02133E-5
  ;
*----
*  Geometry MOSTELA : Annular cell with reflective BC
*           MOSTELV : Annular cell with void BC
*----
MOSTELA := GEO: :: TUBE 3
  RADIUS 0.0 0.39306 0.45802 0.71206
  SPLITR     2       1       1
  MIX        1       2       3
  R+ REFL ;
MOSTELV := GEO: MOSTELA ::
  R+ VOID ;
*----
*  Self-Shielding calculation JPM
*  Transport calculation      SYBIL
*  Flux calculation for keff
*----
VOLMAT := JPMT: MOSTELA ::
  TITLE 'TCWD09: JPM TRACK MOSTELLER BENCHMARK REFLECTIVE BC '
  MAXR 4 IP01 QUA1 5 ;
LIBRARY := SHI: LIBRARY VOLMAT :: ;
VOLMAT := DELETE: VOLMAT ;
VOLMAT := SYBILT: MOSTELA ::
  TITLE 'TCWD09: SYBIL TRACK MOSTELLER BENCHMARK REFLECTIVE BC '
  MAXR 4 QUA1 5 ;
PIJ := ASM: LIBRARY VOLMAT ::  ;
FLUX := FLU: PIJ LIBRARY VOLMAT ::
  TYPE K ;
OUT := EDI: FLUX LIBRARY VOLMAT ::
  EDIT 4 MERG MIX 1 2 3 COND 4.0 SAVE ;
PIJ VOLMAT := DELETE: PIJ VOLMAT ;
VOLMAT := SYBILT: MOSTELV ::
  TITLE 'TCWD09: SYBIL TRACK MOSTELLER BENCHMARK VOID BC '
  MAXR 4 QUA1 5 ;
```

```
PIJ := ASM: LIBRARY VOLMAT :: ;
FLUX := FLU: FLUX PIJ LIBRARY VOLMAT ::
  TYPE K ;
OUT := EDI: OUT FLUX LIBRARY VOLMAT ::
  EDIT 4 MERG MIX 1 2 3 COND 4.0 SAVE ;
OUT FLUX PIJ LIBRARY VOLMAT := DELETE:
  OUT FLUX PIJ LIBRARY VOLMAT  ;
END: ;
QUIT "LIST" .
```

4.4.10 ***TCWD10*** *– Fixed source problem in multiplicative media*

This case verify the use of a fixed source inside a cell where fission also takes place.

Input data for test case: **TCWD10**

```
*----
*  TEST CASE TCWD10
*  MOSTELLER BENCHMARK FOR 1-D ANNULAR CELL
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*  FIXED SOURCE PROBLEM IN MULTIPLICATIVE MEDIA
*
*  REF: None
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  MOSTELA VOLMAT LIBRARY PIJ FLUX OUT ;
MODULE
  LIB: MAC: GEO: JPMT: SYBILT: SHI: ASM: FLU: EDI:
  DELETE: END: ;
*----
*  Microscopic cross section from file WNEALIB format WIMSD4
*  Fixed source of 1.0E5 in group 6
*----
LIBRARY := LIB: ::
  EDIT 0 NMIX 3  CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX 1 600.0                          O16      = '16'     4.61309E-2
    U235     = '235'    1.66078E-4 1 SHIB '235.4'
    U238     = '238'    2.28994E-2 1 SHIB '238.4'
  MIX 2 600.0
    Zr91     = '91'     3.83243E-2
  MIX 3 600.0
    H1H2O    = '2001'   4.42326E-2   O16H2O   = '16'     2.21163E-2
    BNat     = '11'     1.02133E-5
  ;
LIBRARY  := MAC: LIBRARY  ::
  EDIT 0
  READ INPUT
```

```
    MIX 3 FIXE
    0.0 0.0 0.0 0.0 0.0 1.0E+5 0.0 0.0 0.0 0.0
    0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    ;
*----
*  Geometry MOSTELA : Annular cell with reflective BC
*----
MOSTELA := GEO: :: TUBE 3
  RADIUS 0.0 0.39306 0.45802 0.71206
  SPLITR     2       1       1
  MIX        1       2       3
  R+ REFL ;
*----
*  Self-Shielding calculation JPM
*  Transport calculation      SYBIL
*  Flux calculation for keff
*----
VOLMAT := JPMT: MOSTELA ::
  TITLE 'TCWD10: JPM TRACK MOSTELLER BENCHMARK'
  MAXR 4 IP01 QUA1 5 ;
LIBRARY := SHI: LIBRARY VOLMAT :: ;
VOLMAT := DELETE: VOLMAT ;
VOLMAT := SYBILT: MOSTELA ::
  TITLE 'TCWD10: SYBILT TRACK MOSTELLER BENCHMARK'
  MAXR 4 QUA1 5 ;
PIJ := ASM: LIBRARY VOLMAT :: ;
FLUX := FLU: PIJ LIBRARY VOLMAT  ::
  TYPE K ;
OUT := EDI: FLUX LIBRARY VOLMAT  ::
  EDIT 4 MERG MIX 1 2 3 COND 4.0 ;
FLUX := DELETE: FLUX ;
FLUX := FLU: PIJ LIBRARY VOLMAT  ::
  TYPE S EXTE 30 UNKT 1.0E-3 ;
OUT := EDI: OUT FLUX LIBRARY VOLMAT  ::
  EDIT 4 MERG MIX 1 2 3 COND 4.0 ;
OUT FLUX PIJ LIBRARY VOLMAT := DELETE:
  OUT FLUX PIJ LIBRARY VOLMAT  ;
END: ;
QUIT "LIST" .
```

4.4.11 ***TCWD11*** *– Two group burnup of a CANDU–6 type cell*

This case is similar to **TCWD05** except that the burnup module uses DRAGON generated two groups time dependent microscopic cross sections.

Input data for test case: **TCWD11**

```
*----
*  TEST CASE TCWD11
*  CANDU-6 ANNULAR CELL
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*  TWO GROUP BURNUP
*  POWER (KW)          =  615.00000
*  BURN POWER (KW/KG) =   31.97130
*  URANIUM MASS        =   19.23600
*  UO2 REAL DENSITY    =   10.59300
*  UO2 EFF DENSITY     =   10.43750
*  UO2 TEMPERATURE     =  941.28998
*  ENRICHMENT          =    0.71140
*  COOLANT D2 AT %     =   99.222
*  MODERATOR D2 AT %   =   99.911
*  NUMBER OF DAYS      =   50
*
*  REF: R. Roy et al.  Ann. Nucl. Energy 21, 115 (1994)
*
*----
*  Define variables
*     Burnup paremeters
*     a) Power
*        =  31.9713 kw/kg for    0.0 to 300.0 days
*     b) 69 Groups Burnup time interval Delt
*        =   300 day  for    0 to  300 day
*     c) 2 Groups Burnup time interval Delt
*        =     1 day  for    0 to    1 day
*        =     4 days for    1 to    5 days
*        =     5 days for    5 to   10 days
*        =    10 days for   10 to   50 days
*        =    20 days for   50 to  150 days
*        =    50 days for  150 to  300 days
*     c) Days with burnup interval changes
*        = 1.0, 5.0, 10.0, 50.0, 150.0 and 300.0 days
*     d) Burnup control time variables Timei, Timef
*        Timei = initial time
*        Timef = final time
*----
REAL
  Power    Delt   Timec   Timei Timef :=
  31.9713  1.0    300.0   0.0   0.0 ;
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY CANDU6S CANDU6F VOLMATS VOLMATF PIJ FLUX BURNUP EDITION ;
SEQ_BINARY
  INTLINS INTLINF ;
SEQ_ASCII
  RES ;
```

```
MODULE
  GEO: EXCELT: LIB: SHI: ASM: FLU: EVO: EDI:
  DELETE: UTL: END: ;
*----
*  Depletion data from file WNEALIB format WIMSD4
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 10 CTRA WIMS
  DEPL LIB: WIMSD4 FIL: WNEALIB
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX  1  560.66   0.81212           O16      = '16'     7.99449E-1
    D2D2O    = '8002'   1.99768E-1   H1H2O    = '2001'   7.83774E-4
  MIX  2  560.66   6.57
    BNat     = '11'     2.10000E-4
    Zr91     = '91'     9.75000E+1
  MIX  3  345.66   0.0014            He4      = '4'      1.00000E+2
  MIX  4  345.66   6.44              Fe56     = '56'     1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '11'     3.10000E-4
    Zr91     = '91'     9.97100E+1
  MIX  5  345.66   1.082885          O16      = '16'     7.98895E-1
    D2D2O    = '8002'   2.01016E-1   H1H2O    = '2001'   8.96000E-5
  MIX  6  941.29  10.4375010         O16      = '16'     1.18473E+1
    Xe135    = '135'    0.0
    U235     = '235'    6.27118E-1 1 SHIB '235.4'
    U238     = '238'    8.75256E+1 1 SHIB '238.4'
    U236     = '236'    0.0        1
    Pu239    = '3239'   0.0        1 SHIB '3239.1'
  MIX  7  COMB 6 1.0
  MIX  8  COMB 6 1.0
  MIX  9  COMB 6 1.0
  MIX 10   560.66   6.44             Fe56     = '56'     1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '11'     3.10000E-4
    Zr91     = '91'     9.97100E+1
  ;
*----
*  Geometry CANDU6S : 13 regions annular cluster for self-shielding
*           CANDU6F : 31 regions annular cluster for transport
*----
CANDU6S := GEO: ::  TUBE 5
  R+ REFL RADIUS  0.00000 5.16890 5.60320  6.44780 6.58750 16.12171
  MIX  1 2 3 4 5
  CLUSTER ROD1 ROD2 ROD3 ROD4
  ::: ROD1 := GEO: TUBE 2 MIX 6 10 NPIN  1 RPIN 0.0000 APIN 0.0000
    RADIUS 0.00000 0.6122 0.6540 ;
  ::: ROD2 := GEO: ROD1   MIX 7 10 NPIN  6 RPIN 1.4885 APIN 0.0000 ;
  ::: ROD3 := GEO: ROD1   MIX 8 10 NPIN 12 RPIN 2.8755 APIN 0.261799 ;
  ::: ROD4 := GEO: ROD1   MIX 9 10 NPIN 18 RPIN 4.3305 APIN 0.0 ;
  ;
CANDU6F := GEO: CANDU6S :: SPLITR  6 1 1 1 10
```

```
    ::: ROD1 := GEO: ROD1 SPLITR 2 1 ;
    ::: ROD2 := GEO: ROD2 SPLITR 2 1 ;
    ::: ROD3 := GEO: ROD3 SPLITR 2 1 ;
    ::: ROD4 := GEO: ROD4 SPLITR 2 1 ;
    ;
  *----
  *  Self-Shielding calculation EXCEL
  *  Transport calculation      EXCEL
  *  Flux calculation for keff
  *----
  VOLMATS INTLINS := EXCELT: CANDU6S ::
    TITLE 'TCWD11: FEW GROUP BURNUP / SELF-SHIELDING TRACKING'
    EDIT 0 MAXR 13 TRAK TISO 5 10.0 SYMM 12 ;
  LIBRARY := SHI: LIBRARY VOLMATS INTLINS ::
    EDIT 0 ;
  VOLMATF INTLINF := EXCELT: CANDU6F ::
    TITLE 'TCWD11: FEW GROUP BURNUP / TRANSPORT TRACKING'
    EDIT 0 MAXR 31 TRAK TISO 5 10.0 SYMM 12 ;
  PIJ := ASM: LIBRARY VOLMATF INTLINF :: ;
  FLUX := FLU: PIJ LIBRARY VOLMATF ::
    TYPE K ;
  EDITION := EDI: FLUX LIBRARY VOLMATF CANDU6F ::
    MERG REGI  6  6 10  7  7 10  1  1  8  8 10  1
               1  9  9 10  1  1  2  3  4  5  5  5
               5  5  5  5  5  5  5
    COND 4.0 FLIB  ALL SAVE
    SPH MGEO CANDU6F
      ::: EXCELT: EDIT 0 MAXR 31 TRAK TISO 5 10.0 SYMM 12 ;
    ;
  *----
  *  69 group Burnup
  *----
  BURNUP LIBRARY := EVO: LIBRARY FLUX VOLMATF ::
    EDIT 3 EXPM 200.0 DEPL <<Timei>> <<Timec>> DAY POWR <<Power>> ;
  LIBRARY := SHI: LIBRARY VOLMATS INTLINS ::
    EDIT 0 ;
  PIJ := DELETE: PIJ ;
  PIJ := ASM: LIBRARY VOLMATF INTLINF :: ;
  FLUX := FLU: FLUX PIJ LIBRARY VOLMATF ::
    TYPE K ;
  EDITION := EDI: EDITION FLUX LIBRARY VOLMATF CANDU6F ::
    PERT
    SPH MGEO CANDU6F
    ::: EXCELT: EDIT 0 MAXR 31 TRAK TISO 5 10.0 SYMM 12 ;
    ;
  BURNUP FLUX PIJ LIBRARY INTLINS VOLMATS CANDU6S := DELETE:
    BURNUP FLUX PIJ LIBRARY INTLINS VOLMATS CANDU6S ;
  EDITION := UTL: EDITION :: STEP UP 'REF-CASE   1' ;
  LIBRARY :=  EDITION ;
  EDITION := UTL: EDITION :: STEP DOWN ;
  EDITION := DELETE: EDITION ;
  PIJ := ASM: LIBRARY VOLMATF INTLINF :: ;
```

```
FLUX := FLU: PIJ LIBRARY VOLMATF ::
  TYPE K ;
EDITION := EDI: FLUX LIBRARY VOLMATF ::
  EDIT 1 MERGE COMP FLIB  ALL SAVE ;
EVALUATE Timec := 1.0 ;
WHILE Timei Timec < DO
  EVALUATE Timef := Timei Delt + ;
  IF Timei 0.0 = THEN
    BURNUP LIBRARY := EVO: LIBRARY FLUX VOLMATF ::
      EDIT 3 DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
  ELSE
    BURNUP LIBRARY := EVO: BURNUP LIBRARY FLUX VOLMATF ::
      EDIT 3 DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
  ENDIF ;
  PIJ := DELETE: PIJ ;
  PIJ := ASM: LIBRARY VOLMATF INTLINF :: ;
  FLUX := FLU: FLUX PIJ LIBRARY VOLMATF ::
    TYPE K ;
  EDITION := EDI: EDITION FLUX LIBRARY VOLMATF ::
    SAVE  ;
*----
*  change delta t for burnup and final time if required
*----
  IF Timef Timec = THEN
    IF Timec 150.0 = THEN
      EVALUATE Delt Timec := 50.0 300.0 ;
    ENDIF ;
    IF Timec 50.0 = THEN
      EVALUATE Delt Timec := 20.0 150.0 ;
    ENDIF ;
    IF Timec 10.0 = THEN
      EVALUATE Delt Timec := 10.0 50.0 ;
    ENDIF ;
    IF Timec 5.0 = THEN
      EVALUATE Delt Timec := 5.0 10.0 ;
    ENDIF ;
    IF Timec 1.0 = THEN
      EVALUATE Delt Timec := 4.0 5.0 ;
    ENDIF ;
  ENDIF ;
  EVALUATE Timei := Timef ;
ENDWHILE ;
RES := EDITION ;
EDITION BURNUP FLUX PIJ LIBRARY INTLINF VOLMATF CANDU6F := DELETE:
  EDITION BURNUP FLUX PIJ LIBRARY INTLINF VOLMATF CANDU6F ;
END: ;
QUIT "LIST" .
```

4.4.12 ***TCWD12*** *– Mixture composition*

This case illustrates the use of the `INFO:` module of DRAGON (see Section 3.12) as well as the new `COMB` option in the module `LIB:` (see Section 3.2).

Input data for test case: **TCWD12**

```
*----
*  TEST CASE TCWD12
*  WIMSD4 69 GROUPS LIBRARY FILE WNEALIB
*  GENERATE A LIBRARY USING INFO AND OTHER OPTIONS
*
*  REF: None
*
*----
*  Define variables and initialize
*    Coolant properties
*    a) Input
*       TempCool    = Coolant temperature (K)
*       Purity      = D2/(D2+H1) Weight % ratio in Coolant
*    b) Output
*       DensCool    = Coolant Density (g/cm**3)
*       WH1C        = H1 Weight % in Coolant
*       WD2C        = D2 Weight % in Coolant
*       WO16C       = O16 Weight % in Coolant
*    Fuel properties
*    a) Input
*       TempFuel    = Fuel temperature  (K)
*       Enrichment  = U235/(U235+U238) Weight % ratio in Fuel
*       DensFuel    = Fuel Density (g/cm**3)
*    b) Output
*       WU235F      = U235 Weight % in Fuel
*       WU238F      = U238 Weight % in Fuel
*       WO16F       = O16 Weight % in Fuel
*----
REAL
  TempCool Purity TempFuel Enrichment DensFuel   :=
  560.66   99.95  941.29   0.72       10.437501 ;
REAL
 WH1C   WD2C   WO16C  DensCool
 WU235F WU238F WO16F ;
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY ;
MODULE
  LIB: INFO:
  END: ;
*----
*  Get Coolant properties
*----
```

```
ECHO
  "Input - Coolant temperature (K)             " TempCool ;
ECHO
  "Input - D2/(D2+H1) Weight % ratio in Coolant" Purity ;
INFO: ::
  TMP: <<TempCool>> K
  PUR: <<Purity>>   WGT%
  CALC DENS WATER >>DensCool<<
  LIB: WIMSD4 FIL: WNEALIB
  ISO: 3 '2001'   '8002'   '16'
  CALC WGT% D2O >>WH1C<< >>WD2C<< >>WO16C<<
  ;
ECHO
  "Output - Coolant Density (g/cm**3)" DensCool ;
ECHO
  "Output - H1 Weight % in Coolant   " WH1C ;
ECHO
  "Output - D2 Weight % in Coolant   " WD2C ;
ECHO
  "Output - O16 Weight % in Coolant  " WO16C ;
*----
*  Get Fuel properties
*----
ECHO
  "Input - Fuel temperature (K)                  " TempFuel ;
ECHO
  "Input - U235/(U235+U238) Weight % ratio in Fuel" Enrichment ;
ECHO
  "Input - Fuel Density (g/cm**3)                " DensFuel ;
INFO: ::
  ENR: <<Enrichment>> WGT%
  LIB: WIMSD4 FIL: WNEALIB
  ISO: 3 '235'    '238'    '16'
  CALC WGT% UO2 >>WU235F<< >>WU238F<< >>WO16F<<
  ;
ECHO
  "Output - U235 Weight % in Fuel" WU235F ;
ECHO
  "Output - U238 Weight % in Fuel" WU238F ;
ECHO
  "Output - O16 Weight % in Fuel " WO16F ;
*----
*  Microscopic cross sections from file WNEALIB format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 8 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: WNEALIB
  MIX  1  <<TempCool>> <<DensCool>>  O16      = '16'     <<WO16C>>
    D2D2O    = '8002'   <<WD2C>>     H1H2O    = '2001'   <<WH1C>>
  MIX  2  <<TempFuel>> <<DensFuel>>  O16      = '16'     <<WO16F>>
    U235     = '235'    <<WU235F>> 1 SHIB '235.4'
    U238     = '238'    <<WU238F>> 1 SHIB '238.4'
```

```
    MIX  3 COMB 1 0.5 0 0.5
    MIX  4 COMB 1 0.1 2 0.9
    ;
  END: ;
  QUIT "LIST" .
```

REFERENCES

[1] G. Marleau, R. Roy and A. Hébert, *DRAGON: A Collision Probability Transport Code for Cell and Supercell Calculations*, Report IGE–157, Institut de génie nucléaire, École Polytechnique de Montréal, Montréal, Québec (1994).

[2] G. Marleau, A. Hébert and R. Roy, "New Computational Methods Used in the Lattice Code DRAGON", *Topical Meeting on Advances in Reactor Physics*, pp 1.177–1.188, Charleston, South Carolina, March 8–11 1992;

[3] A. Hébert, G. Marleau and R. Roy, "Application of the Lattice Code DRAGON to CANDU Analysis", *Trans. Am. Nucl. Soc.*, **72**, 335 (1995);

[4] A. Hébert, G. Marleau and R. Roy, *A Description of the DRAGON Data Structures. Version DRAGON_000331 Release 3.04*, Report IGE–232 Rev. 3, Institut de génie nucléaire, École Polytechnique de Montréal, Montréal, Québec (2000).

[5] R. Roy and A. Hébert, *The GAN Generalized Driver*, Report IGE–158, Institut de génie nucléaire, École Polytechnique de Montréal, Montréal, Québec (2000).

[6] R. Roy, *The CLE-2000 Tool-Box*, Report IGE–163, Institut de génie nucléaire, École Polytechnique de Montréal, Montréal, Québec (1999).

[7] F.E. Driggers, *A Method for Calculating Neutron Absorption and Flux Spectra at Epithermal Energies*, Report AECL–1996, Atomic Energy of Canada Limited, Chalk River, Ontario (1964).

[8] G. Marleau, M.L. Vergain, A. Hébert and R. Roy, "Computation of the DP1 Collision Probabilities for Spherical and Cylindrical Geometries", *Ann. nucl. Energy*, **17**, 119–134 (1990).

[9] G. Marleau and A. Hébert, "An Integral Transport Method for Treating CANDU and GCR Clusters", *Progress in Nuclear Energy*, **18**, 197–205 (1986).

[10] G. Marleau, R. Roy and A. Hébert, "Analytic Reductions for Transmission and Leakage Probabilities in Finite Tubes and Hexahedra", *Nucl. Sci. Eng.*, **104**, 209–216 (1990).

[11] M. Ouisloumen, G. Marleau, A. Hébert and R. Roy, "Computation of Collision Probabilities for Mixed Hexagonal-Cylindrical Geometries Using the DP1 Approximation to the $J_{\pm}$ Technique", *International Topical Meeting on Advances in Mathematica, Computation and Reactor Physics*, pp 2.2.1.1–2.2.1.12, Pittsburgh, Pennsylvania, April 28 – May 2, 1991.

[12] G. Marleau and A. Hébert, "Analysis of Cluster Geometries Using the DP1 Approximation of the $J_{\pm}$ Technique", *Nucl. Sci. Eng.*, **111**, 257–270 (1992).

[13] A. Hoffman et al., *APOLLO: Code Multigroupe de résolution de l'équation du transport pour les neutrons thermiques et rapides*, Note CEA–N–1610, Commisariat à l'énergie Atomique, France (1973).

[14] A. Hébert, *Développement de la méthode SPH: Homogénéisation de cellules dans un réseau non uniforme et calcul des paramètres de réflecteur*, Note CEA–N–2209, Commisariat à l'énergie Atomique, France (1981).

[15] R. Roy, A. Hébert and G. Marleau, "A Transport Method for Treating Three-Dimensional Lattices of Heterogeneous Cells", *Nucl. Sci. Eng.*, **101**, 217–225 (1989).

[16] R. Roy, G. Marleau, J. Tajmouati and D. Rozon, "Modeling of CANDU Reactivity Control Devices with the Lattice Code DRAGON", *Ann. nucl. Energy*, **21**, 115–132 (1994).

[17] R. Roy, "Anisotropic Scattering for Integral Transport Codes. Part 1. Slab Assemblies", *Ann. nucl. Energy*, **17**, 379–388 (1990).

[18] R. Roy, "Anisotropic Scattering for Integral Transport Codes. Part 2. Cyclic Tracking and its Application to XY Lattices", *Ann. nucl. Energy*, **18**, 511–524 (1991).

[19] R. Roy, A. Hébert and G. Marleau, "A Cyclic Tracking Procedure for Collision Probability Calculations in 2–D Lattices", *International Topical Meeting on Advances in Mathematica, Computation and Reactor Physics*, pp 2.2.4.1–2.2.4.14, Pittsburgh, Pennsylvania, April 28 – May 2, 1991.

[20] G. Marleau and R. Roy, "Use of Specular Boundary Conditions for CANDU Cell Analysis", *Fourth International Conference on Simulation Methods in Nuclear Engineering*, pp 5B.3.1–5B.3.13, Montréal, Québec, June 2–4, 1993.

[21] G. Marleau and A. Hébert, "Solving the Multigroup Transport Equation Using the Power Iteration Method", *1985 Simulation Symposium on Reactor Dynamics and Plant Control*, Kingston, Ontario, April 22–23, 1985.

[22] G. Marleau and A. Hébert, "Introduction of an Improved Critical Buckling Search in WIMS", *1986 Simulation Symposium on Reactor Dynamics and Plant Control*, Hamilton, Ontario, April 21–22, 1986.

[23] R.E. Macfarlane, *TRANSX–CTR: A code for Interfacing MATXS Cross-Section Libraries to Nuclear Transport Codes for Fusion Systems Analysis*, Report LA–9863–MS, Los Alamos Scientific Laboratory, Los Alamos, New Mexico (1984).

[24] *MATXS7A – 69 Neutron Group Cross Section Library in MATXS*, DLC–117, RSIC Data Library Collection, Oak Ridge National Laboratory (1985).

[25] P. Vontobel and S. Pelloni, "New JEF/EFF Based MATXS–Formatted Nuclear Data Libraries", *Nucl. Sci. Eng.*, **101**, 298 (1989).

[26] J.R.Askew, F.J. Fayers and P.B. Kemshell, "A General Description of the Lattice Code WIMS", *J. Brit. Nucl. Energy Soc.*, **5**, 564, (1966).

[27] C.J. Taubman, *The WIMS 69–Group Library Tape 166259*, Report AEEW–M1324, U.K. Atomic Energy Authority, Winfrith (1975).

[28] J.J. Kim, J.T. Lee and H.R. Kim, "Generation and Benchmarking of a 69 Group Cross Section Library for Thermal Reactor Applications", *J. Korean Nucl. Soc.*, **21**, 245 (1989).

[29] J.V. Donnelly, *WIMS–CRNL, A User's Manual for the Chalk River Version of WIMS*, Report AECL–8955, Atomic Energy of Canada Limited, Chalk River, Ontario (1986).

[30] R.E. Macfarlane, *TRANSX–2: A Code for Interfacing MATXS Cross-Section Libraries to Nuclear Transport Codes*, Report LA–12312–MS, Los Alamos Scientific Laboratory, Los Alamos, New Mexico (1992).

[31] R. Roy, "The Cyclic Characteristics Method", *International Conference on the Physics of Nuclear Science and Technology*, Long Island, New York, October 5–8, 1998,

[32] R. Roy, "The Cyclic Characteristics Method with Anisotropic Scattering", *M& C'99 Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, Madrid, Spain, September 27–30, 1999,

[33] M.T. Sissaoui and G. Marleau, "Application of the Feedback Model for the History Base Using DRAGON", *19th CNS Simulation Symposium*, Hamilton, Ontario, October 16–17, 1995.

[34] M.T. Sissaoui, G. Marleau, D. Rozon , "Accounting For Actinide Burnup History in CANDU Reactor simulation", *Advances In Nuclear Fuel Management II*, Myrtle beach, South Carolina, March 23–26, 1997.

[35] M.T. Sissaoui, G. Marleau, D. Rozon , "CANDU Reactor Simulations Using the Feedback Model with Actinide Burnup History", *Nuclear Technology*, **125**, 197 (1999).

[36] A. Hébert, *DRAGON: Description of the DRAGLIB Format*, Report IGE–117, Institut de génie nucléaire, École Polytechnique de Montréal, Montréal, Québec (1991).

[37] A. Hébert, "A Collision Probability Analysis of the Double-Heterogeneity Problem", *Nucl. Sci. Eng.*, **115**, 177 (1993).

[38] A. Kavenoky, *Calcul et utilisation des probabilités de première collision pour les milieux hétérogènes à une dimension*, Note CEA–N–1077, Commisariat à l'énergie Atomique, France (1969).

[39] A. Hébert, "Application of a Dual Variational Formulation to Finite Element Reactor Calculations", *Ann. nucl. Energy*, **20**, 823 (1993).

[40] A. Hébert and G. Marleau, "Generalization of the Stamm'ler Method for the Self-Shielding of Resonant Isotopes in Arbitrary Geometries", *Nucl. Sci. Eng.*, **108**, 230 (1991).

[41] R. Roy and G. Marleau, "Normalization Techniques for Collision Probability Matrices", *PHYSOR–90*, pp IX.40–IX.49, Marseille, France, April 23–27, 1990.

[42] E.A. Villarino, R.J.J. Stamm'ler, A.A. Ferri and J.J. Casal, "HELIOS: Angularly Dependent Collision Probabilities", *Nucl. Sci. Eng.*, **112**, 16–31 (1992).

[43] T. Courau and G. Marleau , "Calculating Adjoint Fluxes in the Code DRAGON Using the Collision Probability Method", *PHYSOR 2000 International Topical Meeting, Advances in Reactor Physics and Mathematics and Computation into the Next Millennium*, Pittsburgh, PA, May 7–11, 2000.

[44] A. Hébert, "A Consistent Technique for the Pin-by-Pin Homogenization of a Pressurized Water Reactor Assembly", *Nucl. Sci. Eng.*, **113**, 227 (1993).

[45] A. Hébert and G. Mathonnière, "Development of a Third-Generation Superhomogénéisation Method for the Homogenization of a Pressurized Water Reactor Assembly", *Nucl. Sci. Eng.*, **115**, 129 (1993).

[46] G. Rimpault, "Algorithmic Features of the ECCO Cell Code for Treating Heterogeneous Fast Reactor Subassemblies", *International Conference on Mathematics and Computations, Reactor Physics, and Environmental Analyses*, Portland, Oregon, April 30 – May 4, 1995.

[47] P. Benoist, J. Mondot and I. Petrovic, "Calculational and Experimental Investigations of Void Effect – A Simple Theoretical Model for Space-Dependent Leakage Treatment of Heterogeneous Assemblies", *Nucl. Sci. Eng.*, **118**, 197 (1994).

[48] I. Petrovic, P. Benoist and G. Marleau, "A Quasi-Isotropic Reflecting Boundary Condition for the Heterogeneous Leakage Model Tibere", *Nucl. Sci. Eng.*, **122**, 151 (1996)

[49] T. Jevremovic, T. Postma, J. Vujic and K. Tsuda, "A Comparative Study of the NEACRP Nine-Pin Supercell Benchmark by the Two-Dimensional General Geometry Transport Codes," *Trans. Am. Nucl. Soc.*, **79**, pp. 169 (1998).

[50] M.J. Halsall, *CACTUS, A Characteristics Solution to the Neutron Transport Equation in Complicated Geometries*, Report AEEW-R 1291, Atomic Energy Establishment, Winfrith (1980).

[51] G. Marleau and E. Debos, "Homogenization of Linearly Anisotropic Scattering Cross Sections in a Consistent B_1 Heterogeneous Leakage Model", *Nineteenth Annual Conference of the Canadian Nuclear Society*, Toronto, Ontario, October 18–21, 1998.

[52] G. Marleau, "Coherent Anisotropic Scattering Homogenization in the B_1 Leakage Model", *M&C'99 Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, Madrid, Spain, September 27-30, 1999.

[53] A. Hébert, "Development of a Second Generation SPH Technique for the Pin-by-Pin Homogenization of a Pressurized Water Reactor Assembly in Hexagonal Geometry", *Trans. Am. Nucl. Soc.*, **71**, 253 (1994).

[54] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes, Second Edition (FORTRAN Version)*, Cambridge University Press, Cambridge (1994).

[55] G. Marleau, "Fine Mesh 3–D Collision Probability Calculations Using the Lattice Code DRAGON", *International Conference on the Physics of Nuclear Science and Technology*, Long Island, New York, October 5–8, 1998,

[56] G. Marleau, "New Geometric Capabilities of DRAGON", *Nineteenth Annual Conference of the Canadian Nuclear Society*, Toronto, Ontario, October 18–21, 1998,

[57] K.E. Kohler, *PostScript for Technical Drawings PSPLOT: A FORTRAN-Callable PostScript Plotting Library User's Manual*, Technical Report Nova Southeastern University, Oceanographic Center, 8000 North Ocean Drive, Dania, Florida; One can get a feel for the flavor of PSPLOT at http://www.nova.edu/ocean/ while access to the full psplot library is via anonymous ftp: whitetip.ocean.nova.edu (137.52.16.19) in the directory psplot.

[58] T.O. Theisen, *Ghostview: An X11 user interface for Ghostscript*, This program is free software under the term of the GNU general public licence as published by the Free Software Fundation.

[59] Ž. Stankovski, "Refinement of the Substructure Method for Integral Transport Calculations", *Nucl. Sci. Eng.*, **92**, 255 (1986).

[60] R.T. Akroyd and N.S. Riyait, "Iteration and Extrapolation Methods for the Approximate Solution of the Even-Parity Transport Equation for systems with voids", *Ann. nucl. Energy*, **16**, 1 (1989).

[61] R.D. Mosteller, L.D. Eisenhart, R.C. Little, W.J. Eich and J. Chao, "Benchmark Calculations for the Doppler Coefficient of Reactivity", *Nucl. Sci. Eng.*, **107**, 265 (1991).

FIGURES

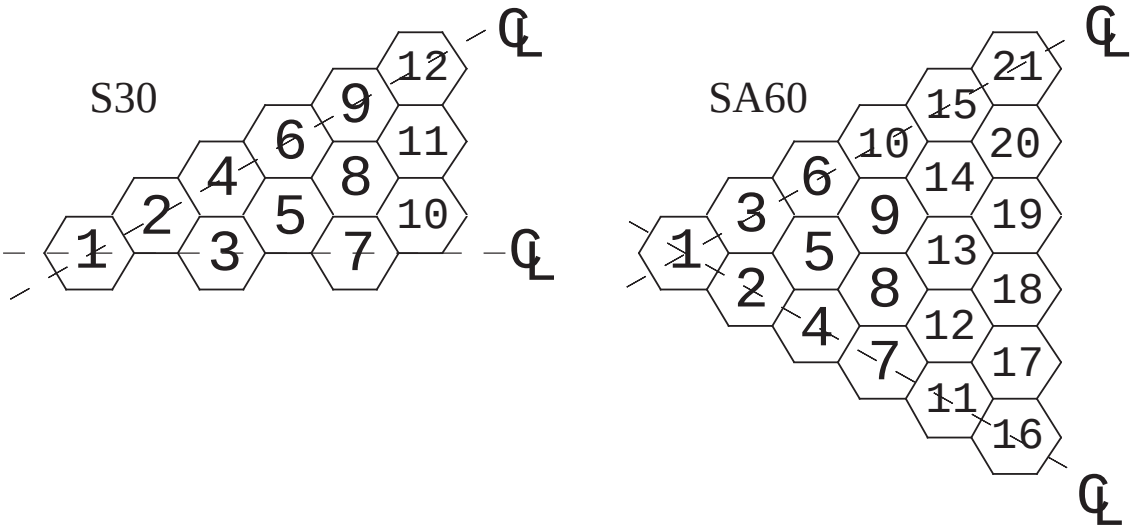

Figure 1: Hexagonal geometries of type S30 and SA60

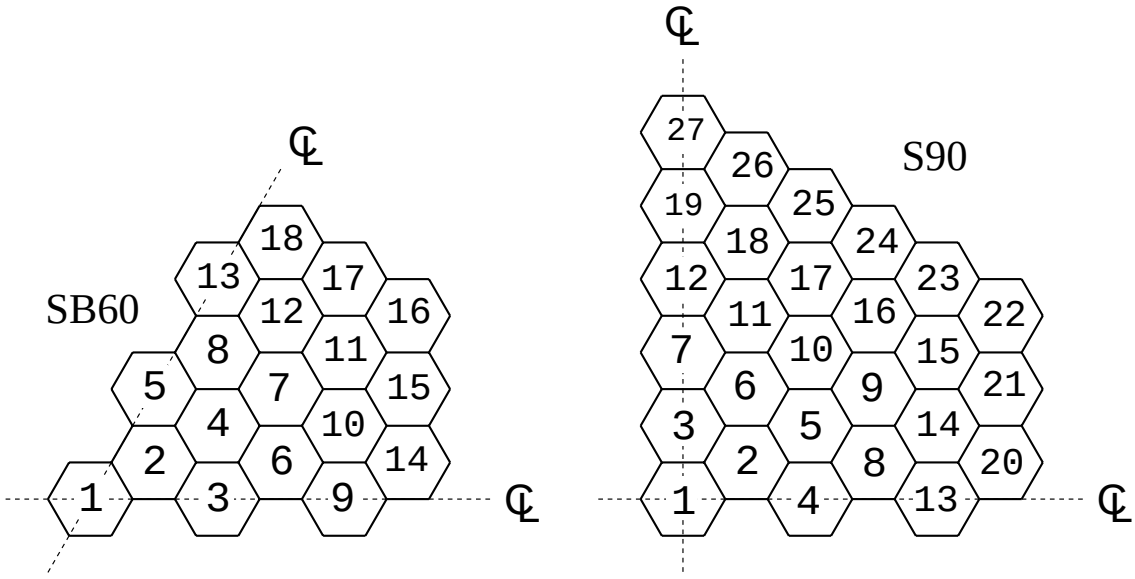

Figure 2: Hexagonal geometries of type SB60 and S90

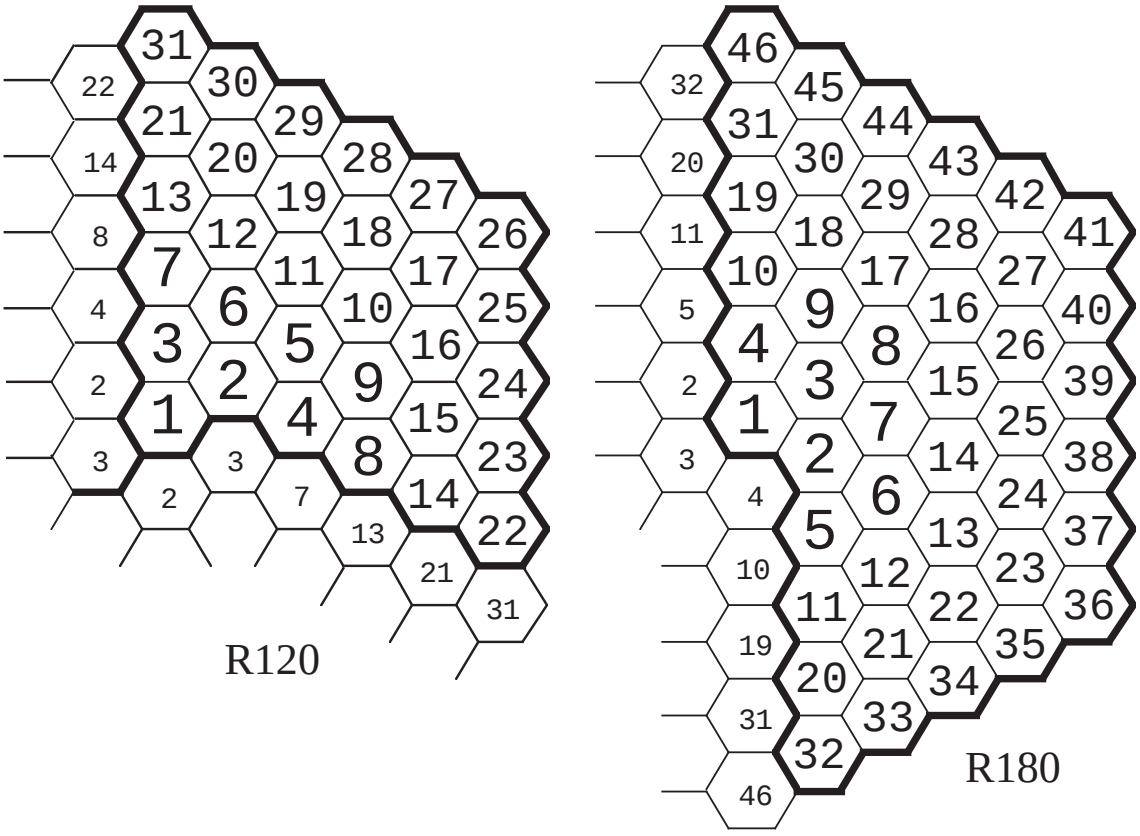

Figure 3: Hexagonal geometries of type R120 and R180

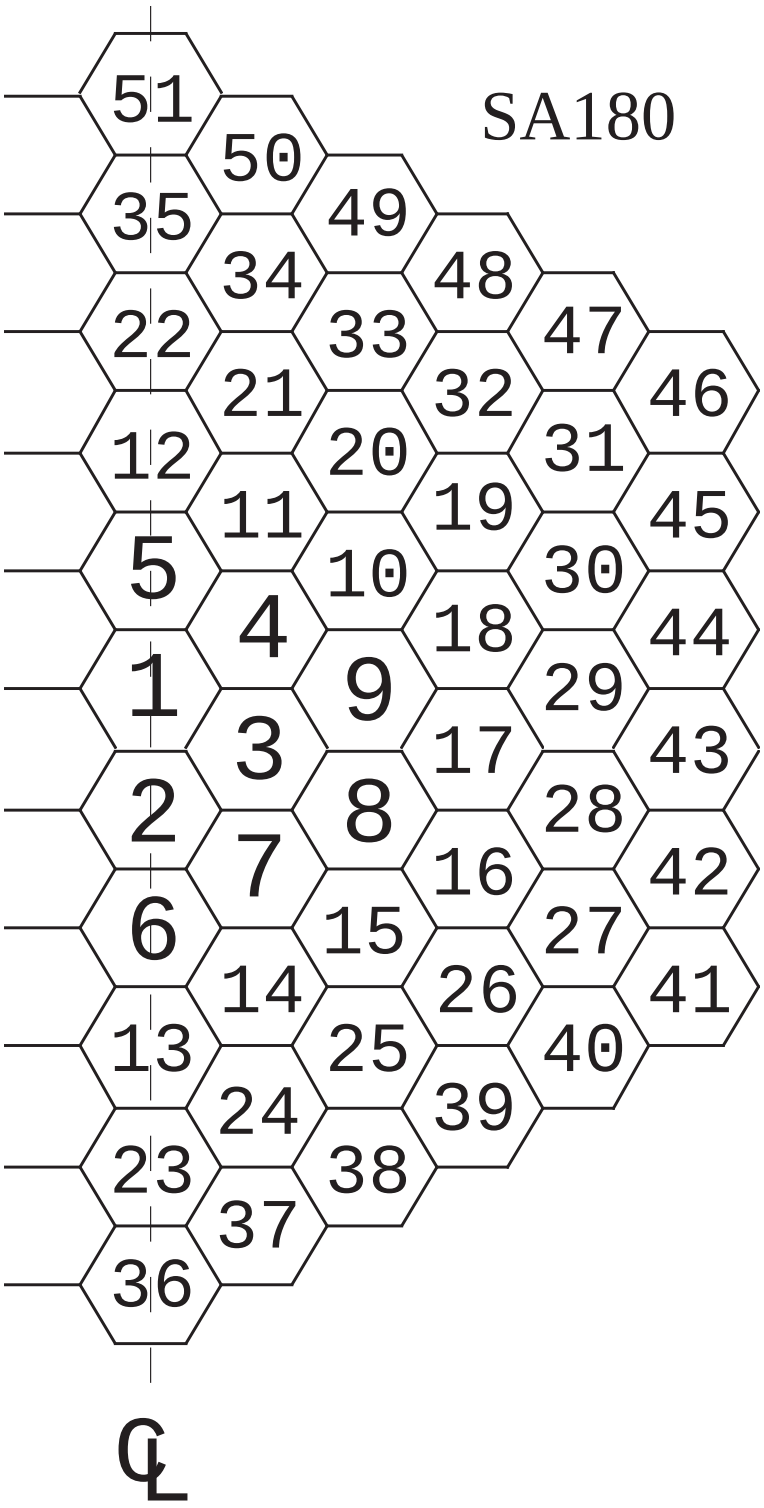

Figure 4: Hexagonal geometry of type SA180

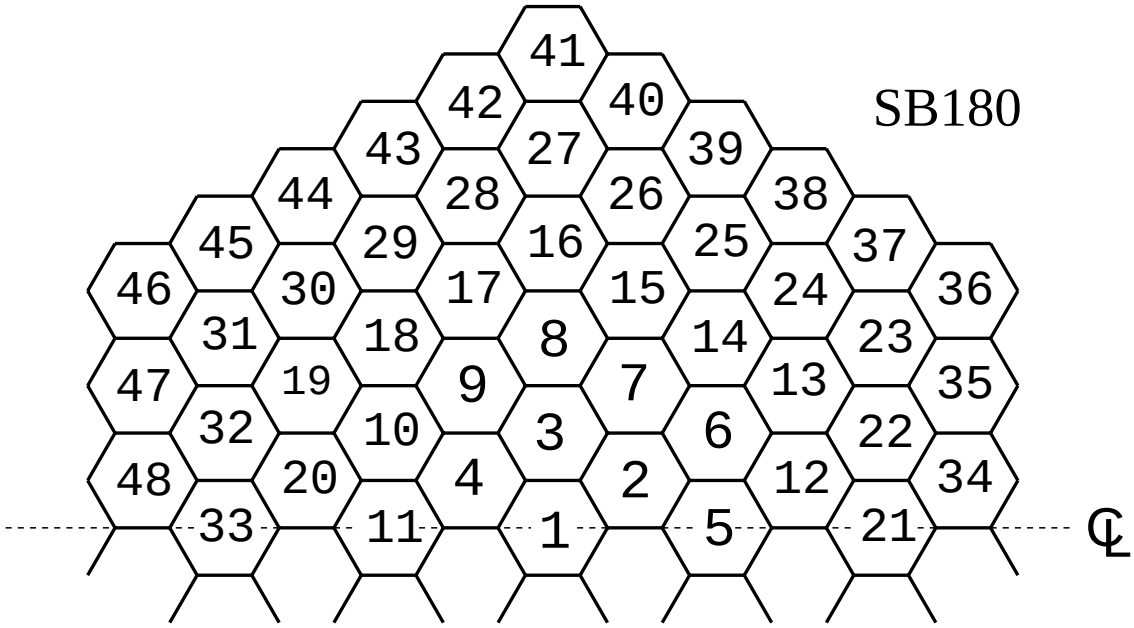

Figure 5: Hexagonal geometry of type SB180

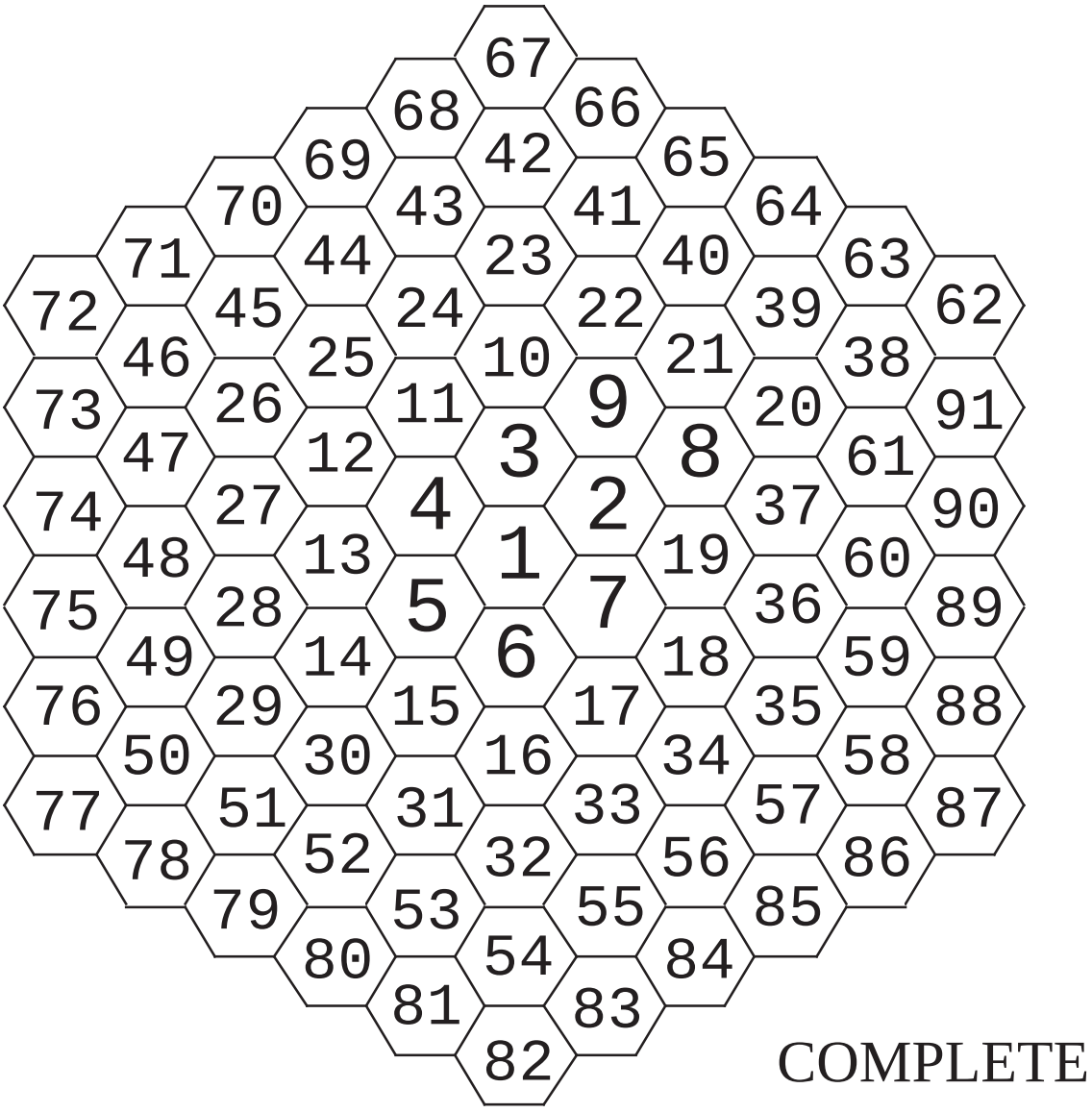

Figure 6: Hexagonal geometry of type COMPLETE

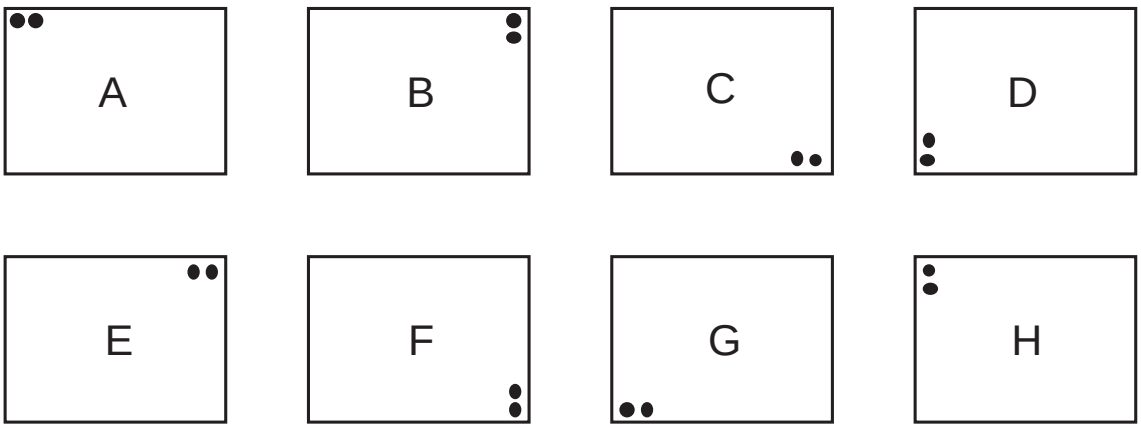

Figure 7: Description of the various rotations allowed for Cartesian geometries

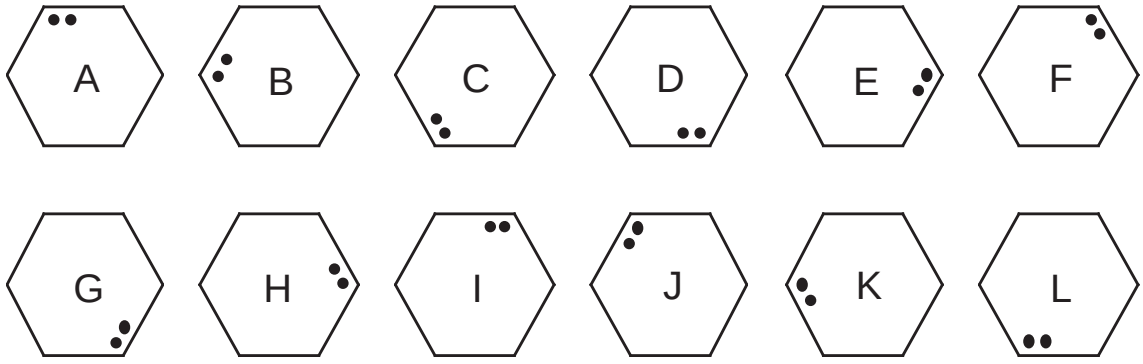

Figure 8: Description of the various rotation allowed for hexagonal geometries

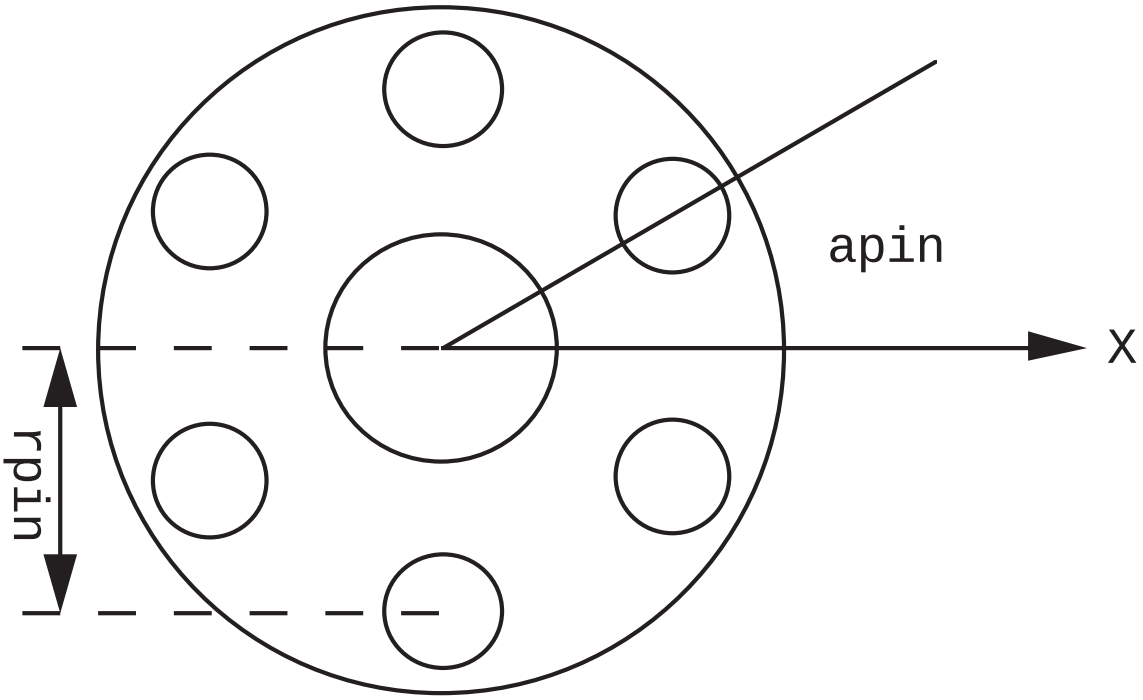

Figure 9: Typical cluster geometry

Figure 10: Slab geometry with mesh splitting

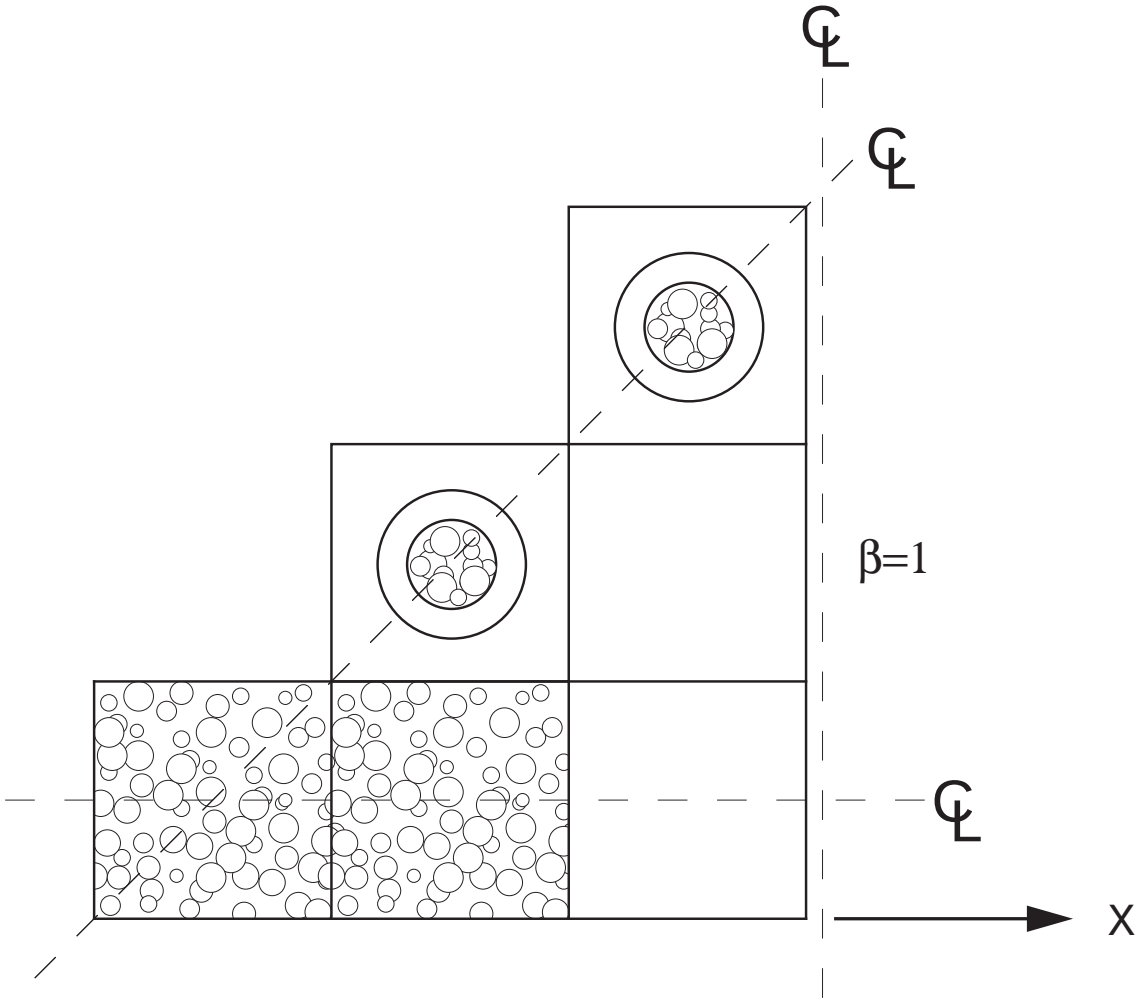

Figure 11: Two dimensional Cartesian assembly containing micro structures

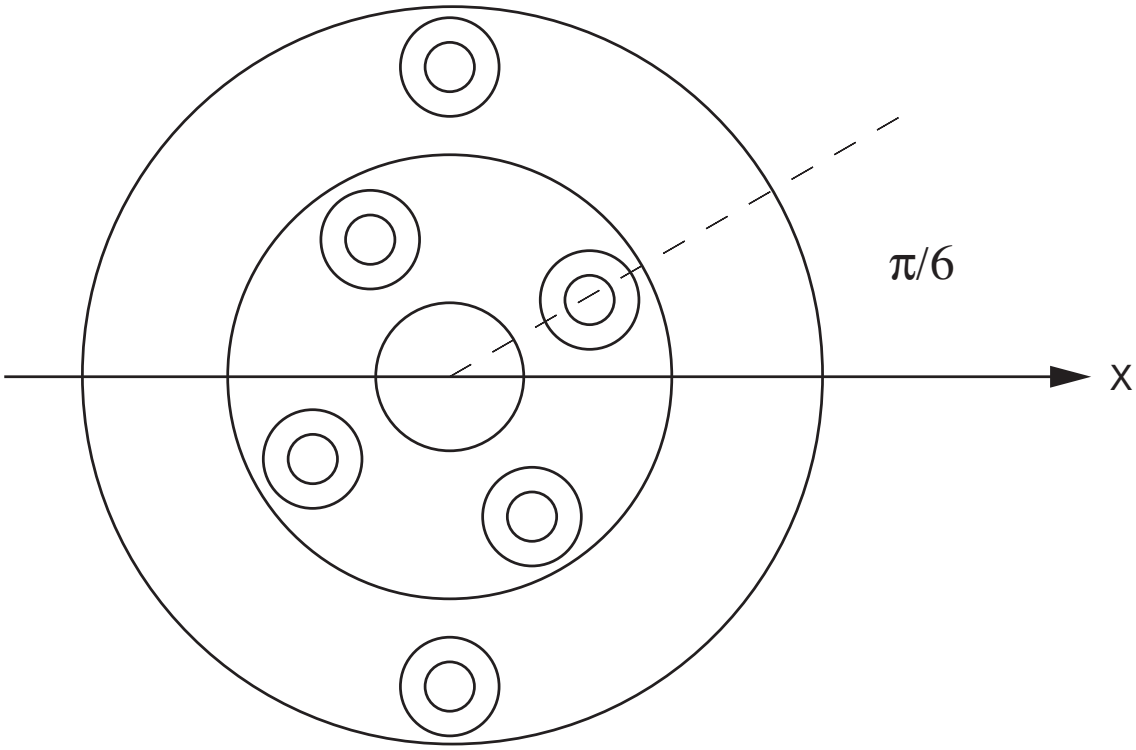

Figure 12: Cylindrical cluster geometry

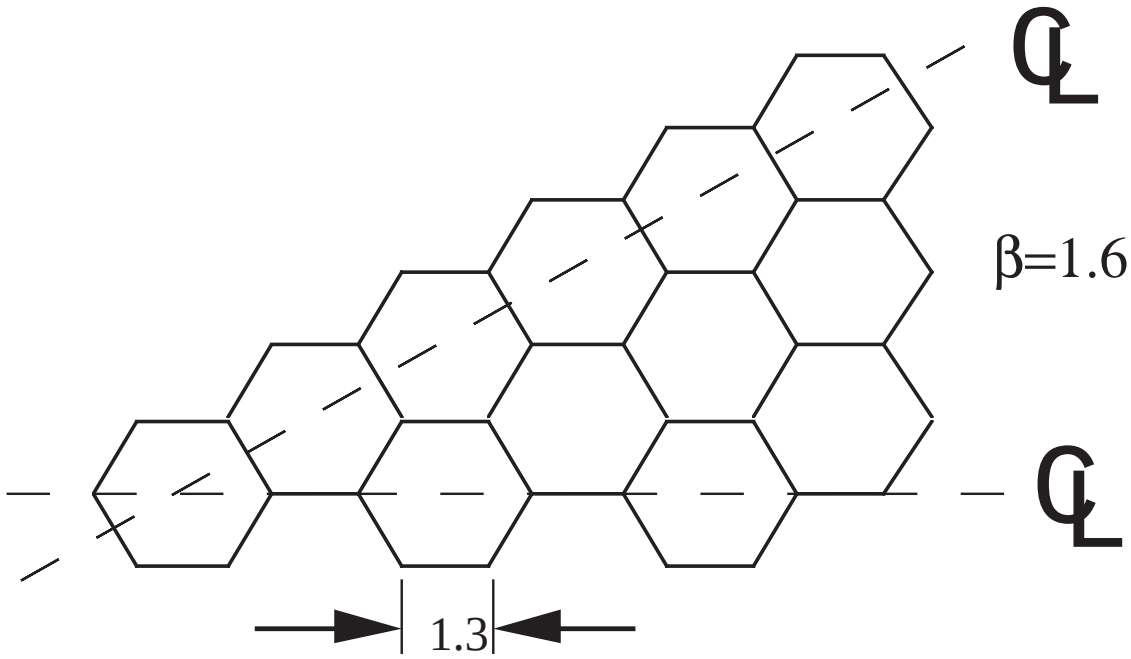

Figure 13: Two dimensional hexagonal geometry

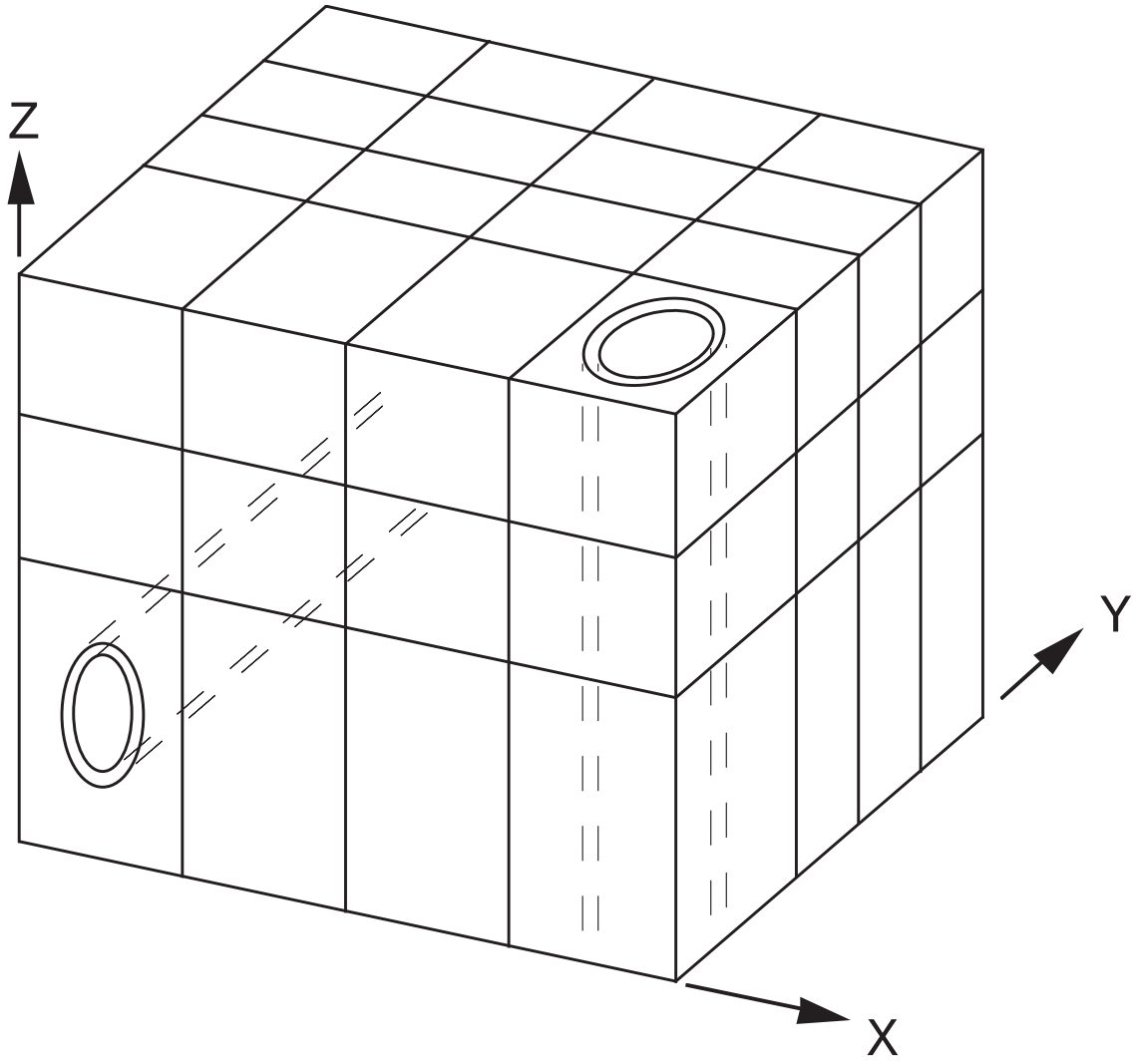

Figure 14: Three dimensional Cartesian supercell

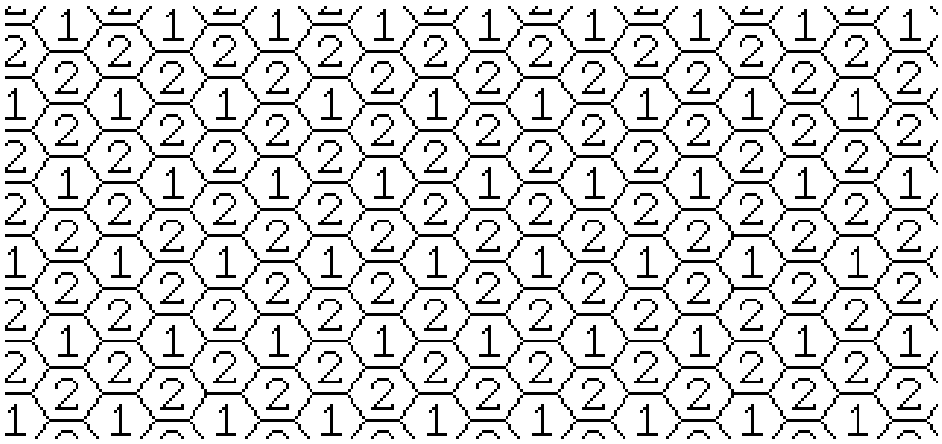

Figure 15: Hexagonal multicell lattice geometry

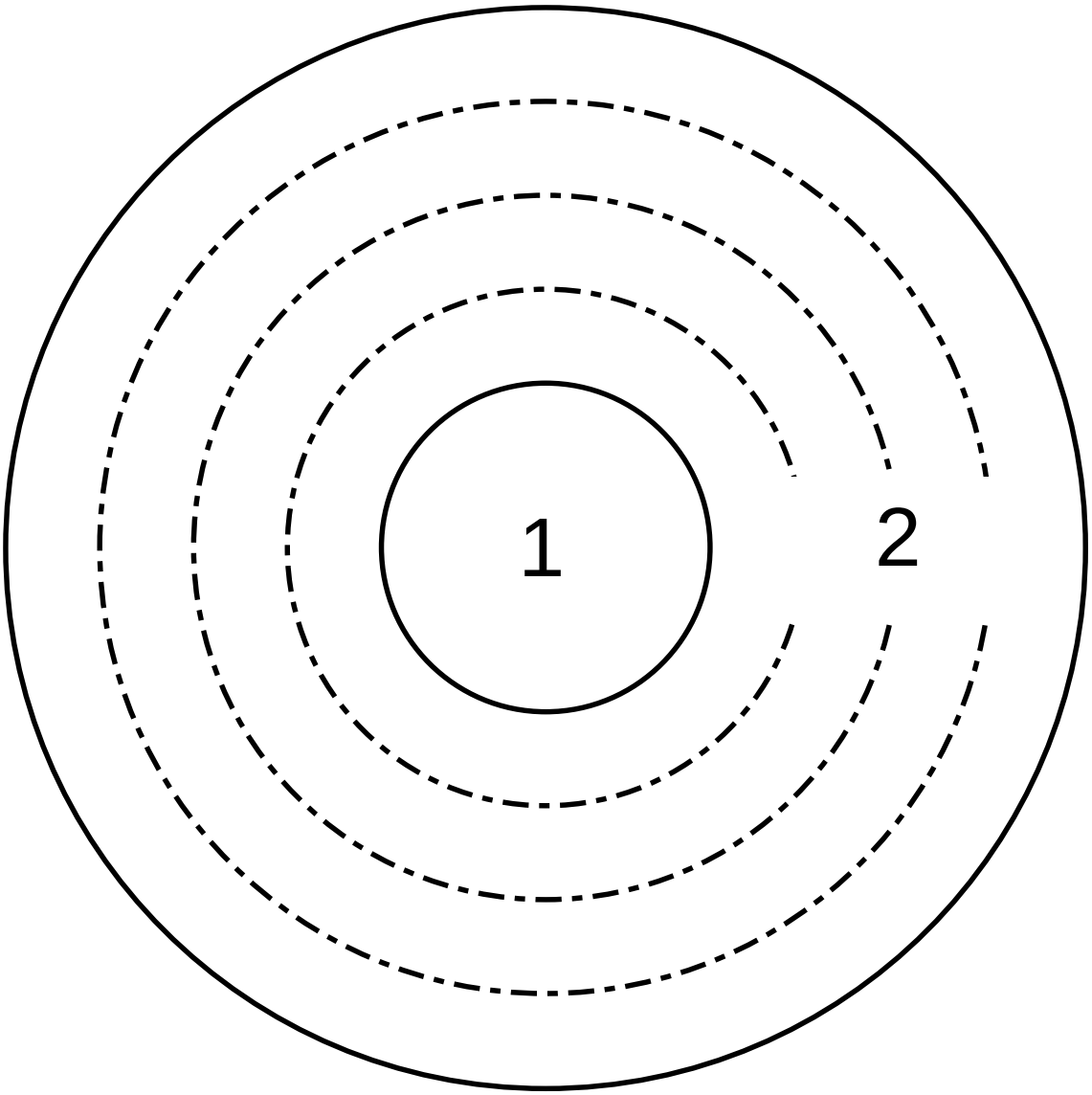

Figure 16: Geometry for test case **TCM01** for an annular cell with macroscopic cross sections.

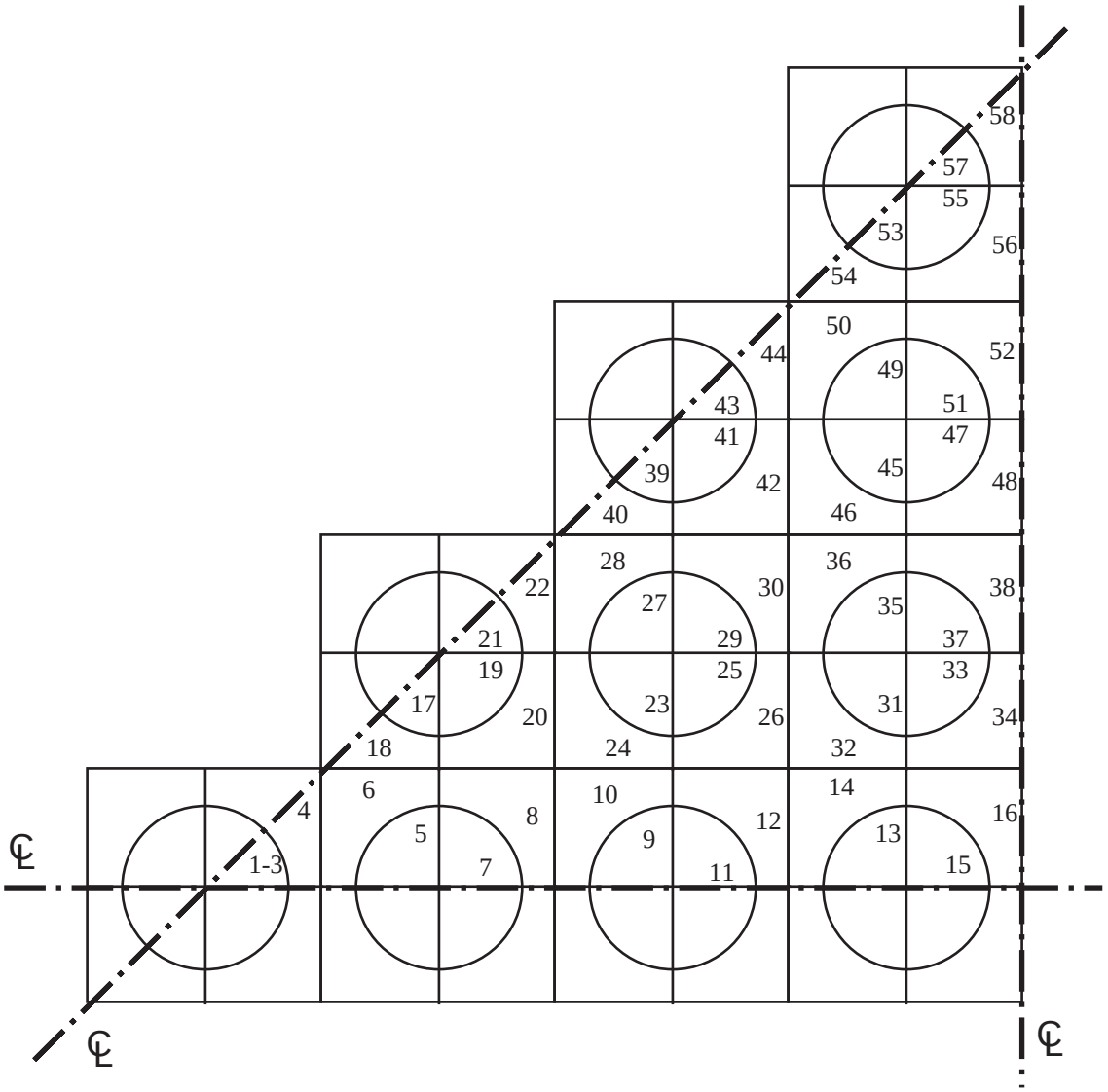

Figure 17: Geometry for test case **TCM02**.

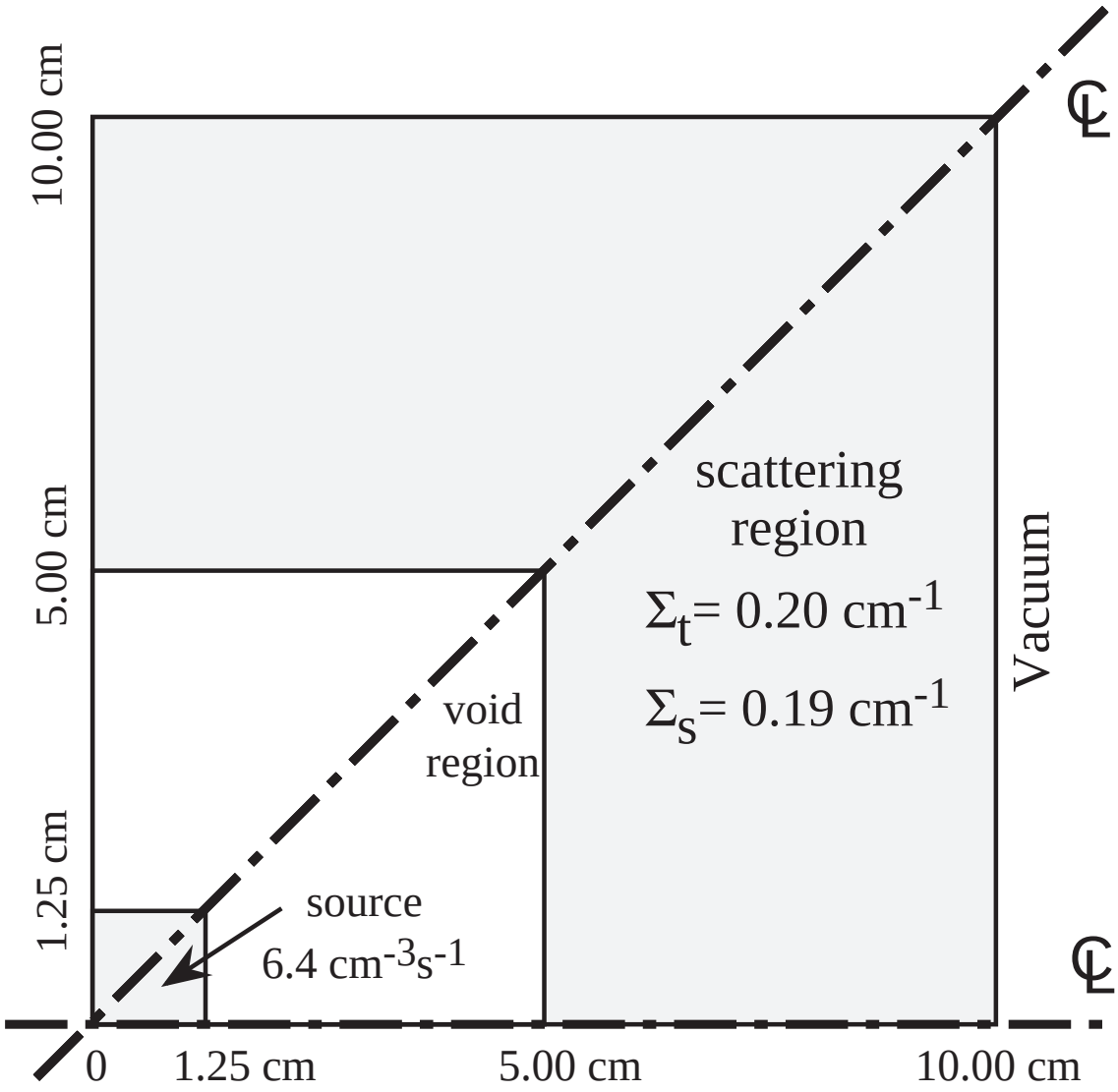

Figure 18: Geometry for test case **TCM03**.

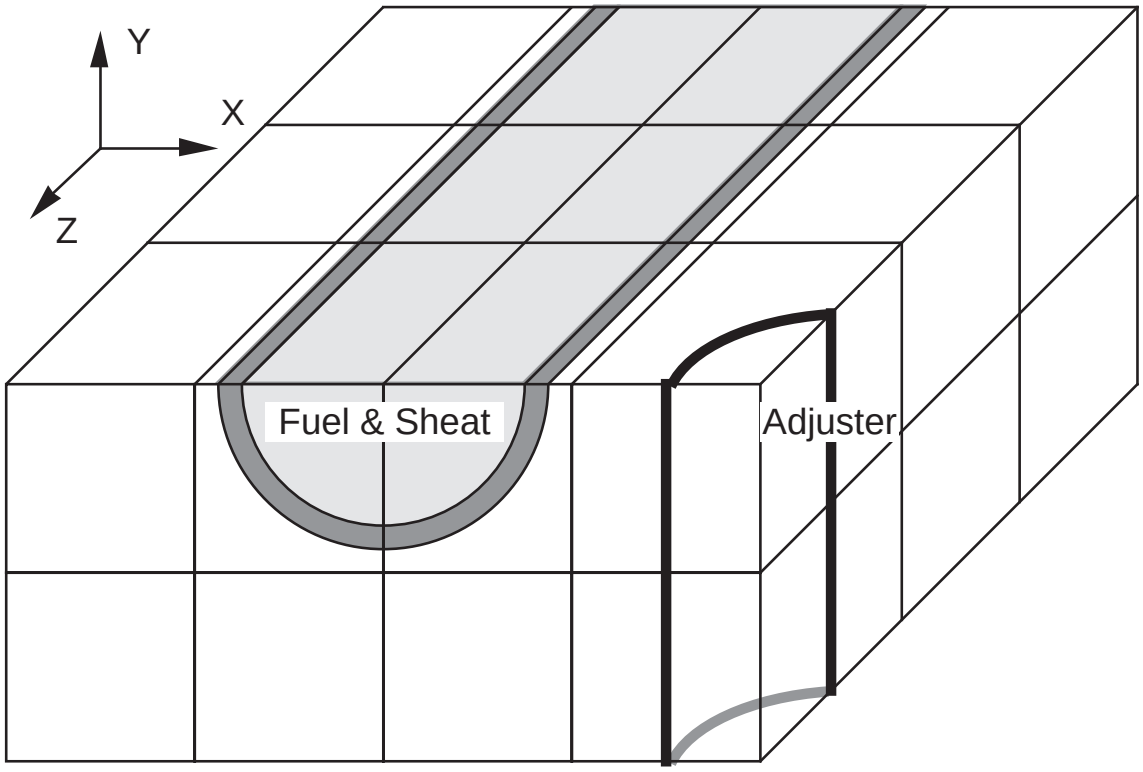

Figure 19: Geometry of the CANDU–6 supercell with stainless steel rods.

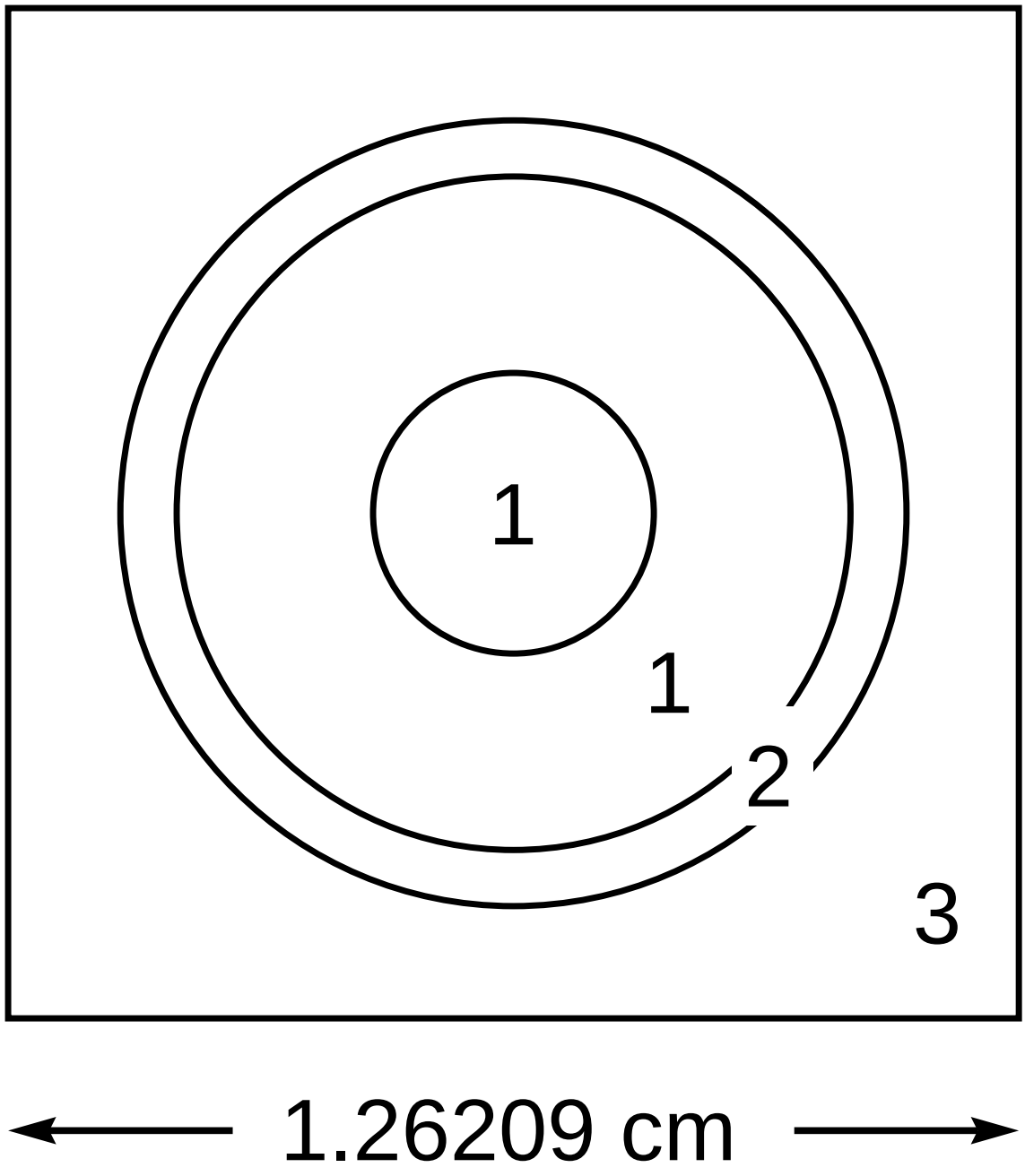

Figure 20: Geometry for the Mosteller benchmark problem.

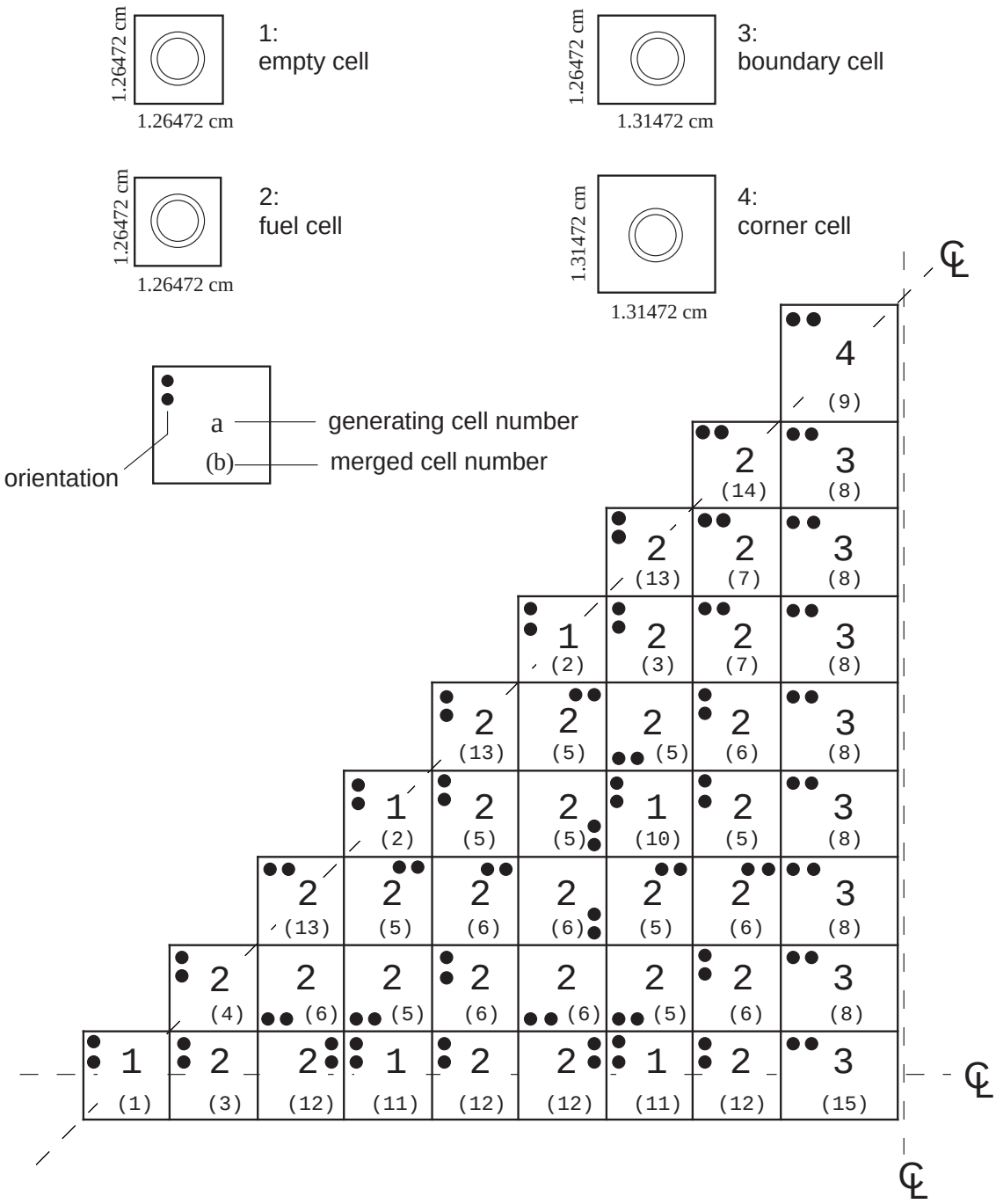

Figure 21: Geometry for test case **TCWD02**.

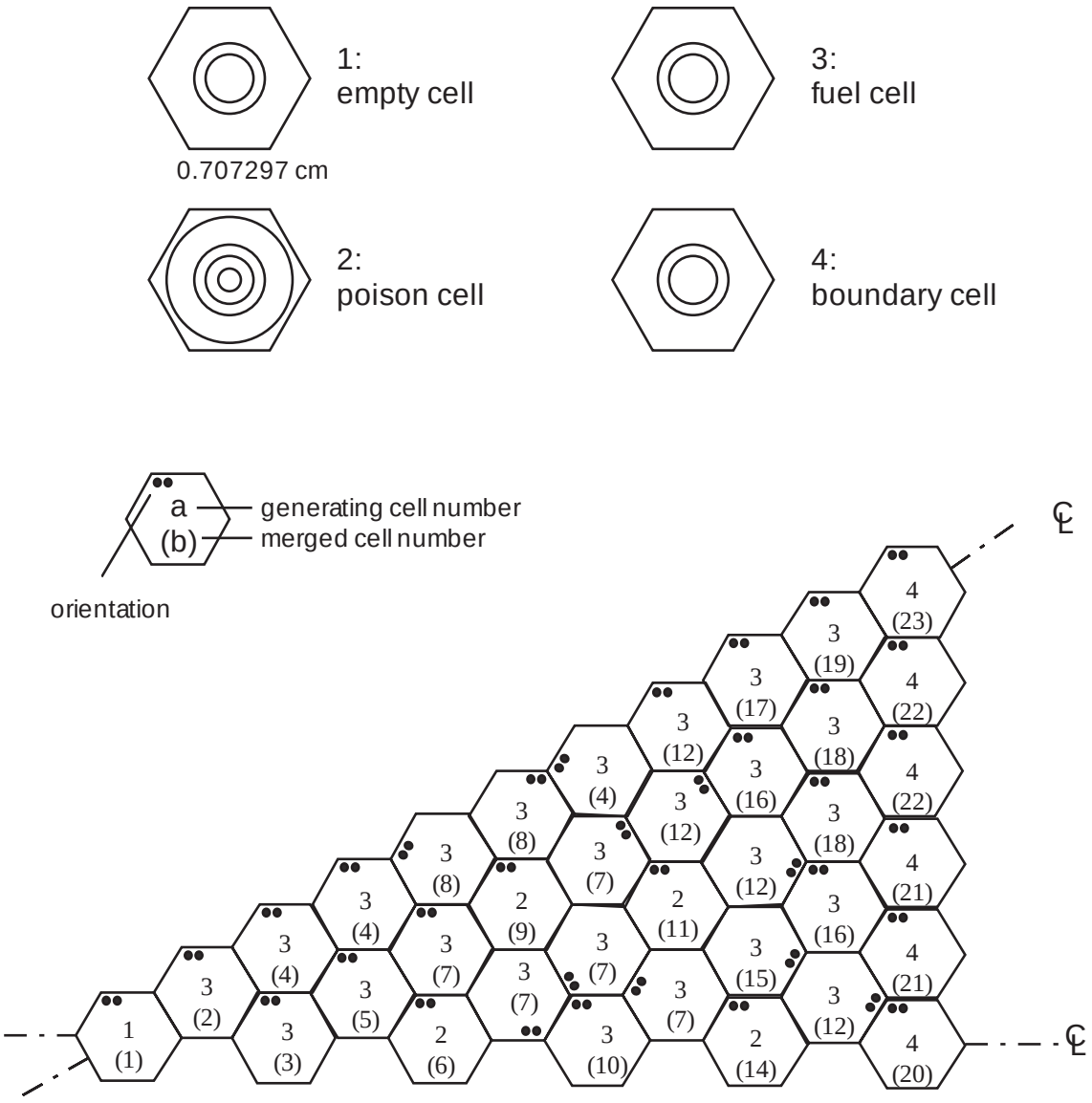

Figure 22: Geometry for test case **TCWD03**.

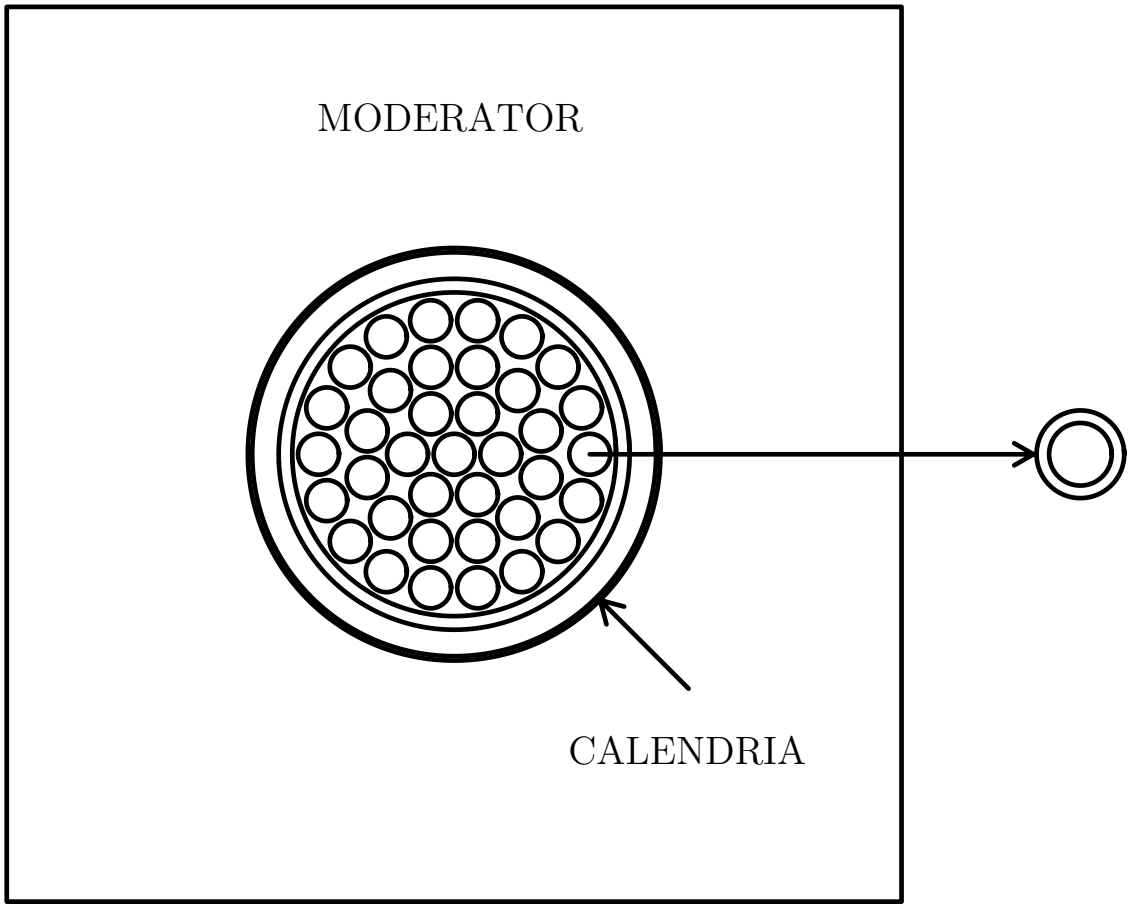

Figure 23: Geometry of the CANDU–6 cell.

INDEX