TECHNICAL REPORT
IGE–174 Rev. 13 (Release 3.06O)

A User Guide for DRAGON

Release 3.06O (2016/02/01)

G. MARLEAU, A. HÉBERT AND R. ROY

Institut de génie nucléaire
Département de génie physique
École Polytechnique de Montréal
February 2016

Copyright Notice for DRAGON

The development of DRAGON has been financially supported along the years, directly or indirectly, by various organizations including École Polytechnique de Montréal, Hydro-Québec and the Hydro-Québec chair in nuclear engineering, the Natural Science and Engineering Research Council of Canada (NSERC), Atomic Energy of Canada limited (AECL) and the CANDU Owners Group (COG).

Acknowledgments

The computer code DRAGON results from a concerted effort of professors, research scientists and graduate students 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 as well as to the graduate students and research staff that 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 that can simulate the neutron behavior of a unit cell or a fuel assembly in a nuclear reactor. It includes all the functions that characterize a lattice cell code, namely:

- interpolation of microscopic cross sections supplied by standard libraries;
- resonance self-shielding calculations in multidimensional geometries;
- multigroup and multidimensional neutron flux calculations that 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;
- isotopic depletion or fuel burnup calculations.

Two neutron flux solution procedures are currently programmed in DRAGON:

- the collision probability method (CPM);
- the method of characteristics (MOC).

Both procedures rely on the same basic approximation, namely the sources (fission, scattering or external) inside each region over which the integrated flux is evaluated is assumed flat. In addition, they are obtained by integrating the transport equation numerically over the neutron directions and space. As a result, DRAGON has been written in such a way that the various numerical quadrature options found in the code (the so-called *tracking procedures*) are all coherent and the information they generate is compatible with both CPM and MOC.

The execution of DRAGON is managed via the GAN generalized driver. The code is modular and can be interfaced easily with other production codes, including the finite reactor code DONJON.

Contents

List of Figures

List of Tables

1 INTRODUCTION

The computer code DRAGON[1–6] results from an effort made at École Polytechnique de Montréal to rationalize and unify into a single code different models and algorithms used to perform lattice cell calculations based on solutions to the neutron transport equation. One of the main concerns of the DRAGON development team has always been to ensure that the structure of the code remains such that the development and implementation of new calculation techniques is facilitated. DRAGON is therefore a lattice cell code divided into many calculation modules that are linked using the GAN generalized driver[7,8]. These modules exchange information only via well defined data structures.

The main components of the code DRAGON are:

- the library access module that reads various microscopic cross section library formats;
- the tracking modules that process a geometry and generate integration lines to be used for collision probability (CP) evaluation or to solve the transport equation using the method of characteristics (MOC);
- the multigroup flux solver used to obtain CP based solutions to the transport equation;
- dedicated modules to solve the multigroup transport equation using the method of characteristic;
- the burnup module that solves the Bateman equation to determine the isotopic contents of various mixtures (in a lattice or out of core) with time;
- the editing module that is used for homogenization and condensation of microscopic and macroscopic cross sections using a flux volume weighting as well as equivalence procedures to preserve reaction rates.

One important feature of DRAGON is its ability to process and track a selection of geometries in one, two and three dimensions. In addition the tracking modules come in various flavor depending on the level of approximation used in the CP or MOC solution to the transport equation. For example:

- the JPM tracking module uses the interface current technique for each homogeneous zones associated with a geometry ($J_{\pm}$ method);[9]
- the SYBIL tracking module emulates the main flux calculation option available in the APOLLO-1 code,[10,11] and includes a new version of the EURYDICE-2 code that performs reactor assembly calculations in both rectangular and hexagonal geometries using the interface current method;
- the EXCELL tracking module generates the CP and MOC tracking lines for isolated 2-D CANDU fuel clusters and for two or three dimensional assemblies containing mixed rectangular/cylindrical cells;[12,13]
- the NXT tracking is a generalization of the EXCELL tracking module to assemblies 2-D and 3-D clusters cells.[14–16]

In principle, SYBIL is 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 the use of an angular approximation for the flux on a large number of interfaces. Similarly, the standard tracking option of the EXCELL and NXT modules, while being more expensive from the point of view of computing time, yield results that are more precise than those based on a SYBILT tracking because no approximation is required at region interface. Finally, the cyclic tracking option programmed in the EXCELL and NXT module provides even better results since it can treat explicitly specular boundary conditions (limited to two dimensional rectangular geometry).[17–19]

After the collision probability or response matrices associated with a given lattice have been generated, the multigroup solution module can be activated. This module uses the power iteration method and requires a number of iteration types.[20] The thermal iterations are carried out by DRAGON so as to rebalance the flux distribution only in cases where neutrons undergo upscattering. The power iterations are performed to solve the fixed source or eigenvalue problem in the cases where a multiplicative medium is analyzed. For problems without eternal sources, the effective multiplication factor (k_{eff}) is obtained at the end of 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 an imposed value.[21] Similar procedures are also used to solve the MOC transport equation.[22–26]

DRAGON can access directly microscopic cross-section libraries having the following standard formats:

- DRAGLIB[27];
- MATXS[28–30];
- WIMS–D4[31–34];
- WIMS–AECL[35].

It can also exchange macroscopic cross-section libraries with codes such as TRANSX–CTR or TRANSX–2 by the use of GOXS and ISOTXS format files.[28, 36] The macroscopic cross section can also be read in DRAGON via the input data stream (useful for few groups benchmarking exercises).

2 GENERAL STRUCTURE OF DRAGON INPUT

The input to DRAGON is in the form of an input data structure containing commands that call successively the calculation modules of DRAGON or the CLE-2000 procedures[7,8] required in a given lattice cell calculation. The DRAGON CLE-2000 procedures are themselves DRAGON input data structures and have the same syntax as the main DRAGON data structure.

In general, the main input data structure for a DRAGON execution is an ASCII file that is identified by 12 characters (case dependent) including an optional `.x2m` extension. A CLE-2000 procedure is also a file identified by 12 characters (again case dependent). However, two file format options are available for these procedures. First, the procedure can be provided in an ASCII format (identified by the `.c2m` extension). In this case, it is compiled by the GAN generalized driver (preprocessed) and the resulting information is stored in a direct access binary file (extension `.o2m`). This is the file that is processed during a DRAGON execution. One can also provide the procedure directly via a direct access binary file (extension `.o2m`). In that case, the GAN generalized driver transfers directly the information to DRAGON for processing. This is mainly used to store reference procedures because they are impossible to modify (the explicit contents of these compiled procedures is also hidden from the user).

2.1 Data organization

The instructions to control the execution of DRAGON are stored in a file (also known as the the input deck or the INPUT data structure) as a collection of sequential `ASCII` record. The logical organization of the input deck is in the form of a list of free format input variables and keywords. The instructions must be located in the first 72 columns of each record in the input stream. Characters located in column 73 and higher are treated as comments (they can be used to identify a record). An input variable can be defined in one of two ways.

- As a set of consecutive characters containing no blanks; it will be automatically interpreted by DRAGON as being either an `INTEGER`, a `REAL`, a `DOUBLE PRECISION` or a `CHARACTER` variable depending on the format of the input variable. The identification of `INTEGER`, `REAL` and `DOUBLE PRECISION` variables follows the FORTRAN prescriptions, everything else is automatically assumed to represent a character variable.
- As a set of characters enclosed between quotation marks (`'␣'`). In this case, the information is always assumed to represent a character variable.

The only separator allowed between two input variables is one or more blank character (not enclosed between quotation marks). A single input variable cannot span two records. Comments can also be included in the input deck as folows:

- characters in column 73 or higher on a record;
- each record starting with the character `*`;
- characters on a record following a `!`;
- characters on a record following the `;` keyword.

These comments are not transferred to DRAGON during the execution but are useful to document the input data structure.

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 of 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 letters. 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 characters 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 curly braces and separated by vertical bars `{␣|␣|␣}` represent various calculation options available in DRAGON. Only one of these options is permitted.
- The variables or structures surrounded by `>>␣<<` represents CLE-2000 output parameters.[7,8]

When a fixed default value is specified for an optional parameter in a structure, then that parameter is reinitialized to its default value every time the module is called. When a floating default value is specified, it is saved on the output data structure and can be used in future calls to this module provided it is then provided as input to the module (read only or update mode). In DRAGON, most default value are floating, the exception being the parameter *iprint* (default value of 1) that is used to control the amount of information printed by each module. Departure from this general rule will be indicated clearly in the following sections.

2.2 DRAGON Data Structure and Module Declarations

DRAGON is built around the GAN generalized driver.[7,8] Accordingly, all the modules and procedures that are used in an input deck must be declared. One must also define the format of each data structure that will be processed by these modules. The modules and procedures required for a specific calculation are called successively, the information generated by the execution of one or several modules being transferred (as requested by the user) to other modules via the data structures. Finally, the execution is terminated by calling the `END:` module. This is true even if additional data records may be present 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), $i = 1, N_A$) `;`]
[`PROCEDURE` (*PROCNAME*(i), $i = 1, N_M$) `;`]
[[**(module)** `;`]]
`END: ;`

where

`MODULE` keyword used to specify the list of modules to be used in this 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 while the utility modules are provided in the `GANLIB` user guide.[7] The number of module declared N_M depends on the particular application of DRAGON.

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

`XSM_FILE` keyword used to specify the data structures that will be stored in XSM format files.

`SEQ_BINARY` keyword used to specify the data structures that will be stored in sequential binary files.

`SEQ_ASCII` keyword used to specify the data structures that will be stored in sequential ASCII files.

STRNAME list of N_L, N_X, N_B or N_A `character*12` name of data structures. The type and format (`LINKED_LIST`, `XSM_FILE`, `SEQ_BINARY` and `SEQ_ASCII`) of various DRAGON data structures is presented in Section 2.4.

`PROCEDURE` keyword used to specify the user defined procedures to be used in this DRAGON execution.

PROCNAME list of N_M `character*12` name of DRAGON procedure. These procedures are stored in a file with name *PROCNAME*`.c2m` and contain standard DRAGON instructions.[7,8]

(module) input specifications for a DRAGON or utility module. For the DRAGON specific modules, these input structures are defined in Section 3. For utility modules, the equivalent information is provided in report IGE-158[7] and IGE-163[8].

`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. Several exceptions to this general rule exist including the tracking files (`SEQ_BINARY`) and the PostScript graphical file (`SEQ_ASCII`). The data structures stored on `LINKED_LIST` and `XSM_FILE` can be archived on sequential ASCII files for backup purpose. The input data normally ends with a call to the `END:` module.

Finally **(module)** contains a description of the execution modules to be called as well as its associated input structure. All the modules can be called more than once.

2.3 The DRAGON Modules

The code DRAGON is divided into 27 main calculation modules. Some of these modules perform identical tasks but using different calculation techniques (i.e. the five tracking modules) while others combine several modules (i.e. `EXCELL:` that combines `ASM:` and `EXCELT:`). One of the modules, namely `BIVACT:`, can only be called indirectly as a sub-module of `EDI:`. These modules perform the following tasks:

`MAC:` generates or modifies a DRAGON MACROLIB (see Section 2.4) that contains the group ordered macroscopic cross sections for a series of mixtures (see Section 3.1). The MACROLIB generated can be an independent data structure or included as a substructure in a MICROLIB. The spatial positioning of these mixtures in a geometry is provided by the `GEO:` module (see Section 3.3).

`LIB:` generates or modifies a DRAGON MICROLIB (see Section 2.4). It can read different formats of microscopic cross-section libraries (see Section 3.2). Currently the DRAGLIB[27], MATXS[28,29], WIMS-D4[31–34], and WIMS-AECL[35] formats are supported. After having interpolated the microscopic cross-sections for each isotope in temperature and dilution, 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 positioning of these mixtures in a geometry is provided by the `GEO:` module (see Section 3.3).

`GEO:` generates or modifies a geometry (see Section 3.3).

`EXCELT:` tracks the geometry using the standard EXCELL procedure for 2–D and 3–D assemblies of mixed cartesian/annular or hexagonal/annular cells as well as isolated 2-D cells containing CANDU-type clusters (see Section 3.4.2).

`NXT:` tracks the geometry using the NXT procedure for 2–D and 3–D assemblies of cells containing CANDU-type clusters (see Section 3.4.3).

`SYBILT:` tracks the geometry using the interface current technique (see Section 3.4.4).

`JPMT:` tracks the geometry using the $J_{\pm}$ technique (see Section 3.4.5).

`BIVACT:` tracks the geometry using a diffusion type algorithm (see Section 3.4.6). This module can only be called indirectly as a sub-module of `EDI:`.

`SHI:` performs resonance self-shielding calculations (see Section 3.5).

`ASM:` generates multigroup response or collision probability matrices (see Section 3.6.1) based on tracking information.

`EXCELL:` combines the `EXCELT:` tracking module and the assembly module `ASM:` thereby avoiding the generation of a binary tracking file (see Section 3.6.2).

`FLU:` solves the transport equation for the flux using the multigroup response or collision probability matrices (see Section 3.7).

`MOCC:` solves the transport equation for the flux using the cyclic method of characteristics with mirror like boundary conditions for 2-D geometry (see Section 3.8.1).[22, 23]

`MCU:` solves the transport equation for the flux using the method of characteristics with white boundary conditions for 3-D geometry (see Section 3.8.2).[24–26]

`EDI:` performs editing for the flux, cross sections and reaction rates (see Section 3.9) according to regional homogenization and energy condensation requirements.

`EVO:` solves the Bateman equations for the time (burnup) dependent isotopic contents of the mixtures in a MICROLIB (see Section 3.10).

`INFO:` computes 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).

`CPO:` creates a simplified reactor cross section database (see Section 3.11).

`CFC:` creates a reactor cross section database with Feedback coefficients (see Section 3.13).[37–39]

`MRG:` pre-homogenizes a geometry tracked using the module `EXCELT:` or `NXT:` (see Section 3.14). This module can also segment `NXT:` based tracking files for use with multistep CP integration in the `ASM:` module.

`PSP:` generates PostScript images for 2–D geometries that can be tracked by `EXCELT:` or `NXT:` (see Section 3.15).

`SAD:` computes the generalized adjoint fluxes associated with homogenized and condensed cross-sections (see Section 3.16).[40–44]

`PER:` performs perturbation theory calculations (see Section 3.17).[40–44]

`HST:` manages a full reactor execution in DONJON[45] using explicit DRAGON calculations for each cell (see Section 3.18).[46–49]

`TLM:` creates Matlab[50] `m-file` to generate a graphics representation of the `NXT:` tracking lines (see Section 3.19).[51]

`FMT:` performs dedicated editing for the flux, cross sections and reaction rates (see Section 3.9) according to regional homogenization and energy condensation requirements for other codes (see Section 3.20).[52]

`ITR:` performs dedicated editing to generate TRIPOLI importance files (see Section 3.21).[53]

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 by one of the DRAGON or utility modules. Here we will give a brief description of these data structures, their complete contents being described in report IGE-232.[4]

The execution of a sequence of module by DRAGON can be replaced by a sequence of DRAGON executions calling a single module provided the data structure generated by the different modules are save on a physical support (created on an `XSM` file or exported to a `XSM` or a sequential `ASCII` file). These files can be re-imported for a new DRAGON execution and then used for subsequent calculations. This is also the preferred method for restarting an execution that has been stopped prematurely.

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 includes 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 store the geometry description. 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 is created by the `GEO:` module. It is required for a successful execution of the modules `JPMT:`, `SYBILT:`, `EXCELT:`, `EXCELL:`, `NXT:`, `BIVACT:`. It can also be used by the `PSP:` module (`EXCELT:` compatible 2–D geometries).

TRACKING a standard data structure used by DRAGON to store the general tracking information. It is a stand-alone structure. When used by a DRAGON module it must be stored on a linked list (or an XSM file when the `NXT:` module is considered). It can be created by the `JPMT:`, `SYBILT:`, `EXCELT:`, `EXCELL:` and `NXT:` modules. It is required for a successful execution of the `ASM:`, `FLU:`, `MOCC`, `MCU:`, `EDI:`, `EVO:`, `SAD:` and `PER:` modules. It can also be used by the `MRG:` and `PSP:` modules.

ASMPIJ a standard data structure used by DRAGON to store the multigroup response and collision probability matrices. 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:` and `EXCELL:` module. It is required for a successful execution of the `FLU:` module.

FLUXUNK a standard data structure used by DRAGON to store the fluxes, the adjoints, the generalized adjoints, the multiplication constant (multiplicative problem without leakage and no external

sources), the leakage coefficients and the buckling (multiplicative problem with leakage and no external sources). 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:`, `MOCC:`, `MCU:` and `SAD:` modules. It is required for a successful execution of the `EDI:` and `EVO:` modules. It can also be used by the `PSP:` module.

EDITION — a standard data structure used by DRAGON to store condensed and homogenized microscopic and macroscopic cross sections. It is a stand-alone structure that can contain embedded MACROLIB and MICROLIB sub-structures. 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. It is 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. It can also be used by the `LIB:` and `HST:` module.

CPO — a standard data structure used by DRAGON to store a simplified reactor dabase. 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. It is required for a successful execution of the `CFC:` module. It can be used by the `CRE:` module of DONJON.[45]

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. It can be used by the `AFM:` module of DONJON.[39, 45]

HISTORY — a standard data structure that contains the information required to ensure a smooth coupling of DRAGON with DONJON when an history based full reactor calculation is to be performed. It is a stand-alone structure that must be stored on a linked list or an XSM file. It is only used by the `HST:` module of DRAGON. It can be used and modified by several modules of DONJON.[45–48]

3 THE DRAGON MODULES

The input to DRAGON has the form of a series of input data structures that are call successively during the execution. These input data structures can be interspaced with calls to utility modules, procedures and GANLIB structure and variables definitions. Each input data structure contains an execution command (a module) and data (instructions for the module). In the following sub-section, we will describe the input data structures associated with each modules of DRAGON.

3.1 The MAC: module

The MAC: module is used to store the macroscopic cross sections associated with a mixture in a MACROLIB. The MAC: module can process the information in one of three different ways. First, it can read the cross sections directly from the input stream. It can also read this information from a GOXS format binary sequential file.[28] 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. Finally, the MACROLIB can also be created using information on a MICROLIB or EDITION data structure. This module can also be used to transfer the macroscopic cross sections stored in a MACROLIB to a GOXS format binary file.

The general format of the MAC: input data structure is the following:

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

{ *MACLIB* := MAC: [*MACLIB*] :: **(descmac) (descmaci)** |
MICLIB := MAC: *MICLIB* :: **(descmac) (descmaci)** |
MACLIB := MAC: [*MACLIB*] *OLDLIB* :: **(descmac) (descmacm)** |
MICLIB := MAC: [*MICLIB*] *OLDLIB* :: **(descmac) (descmacm)** }

The first form is for the case where a single MACROLIB is involved (creation or update), the second form corresponds to the case where a single MICROLIB is to be updated, the third form is valid when two MACROLIB are to be combined and finally the fourth form is used to combine two MICROLIB. The meaning of each of the terms above is:

MACLIB character*12 name of a MACROLIB that will contain the macroscopic cross sections. When *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 that will be used to update or create a MACROLIB or a MICROLIB.

(descmac) general MAC: processing instructions (see Section 3.1.1).

(descmaci) instructions to read the macroscopic cross sections from the input data stream (see Section 3.1.2).

(descmacm) instructions to transfer the macroscopic cross sections from *OLDLIB* to *MACLIB* or *MICLIB* (see Section 3.1.3).

*3.1.1 The **(descmac)** input structure for* MAC:

The **(descmac)** input structure takes the form:

Table 3: Structure **(descmac)**

```
[ EDIT iprint ]
[ NMIX nmixt ]
[ NIFI nifiss ]
[ ANIS naniso ]
```

Here

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 be written to the output file if the variable *iprint*$\geq$2. The transfer cross sections will be printed if *iprint*$\geq$5. The normalization of the transfer cross sections will be checked if *iprint*$\geq$10.
NMIX	keyword used to define the number of material mixtures. This information is required when the number of mixtures on the *MACLIB* is to be increased. When *MACLIB* is in creation mode, one assumes that at least one mixture will be added and *nmixt*=1. When *MACLIB* is in update mode *nmixt* is selected as the maximum mixture number currently stored on *MACLIB*.
nmixt	the maximum mixture number (a mixture is characterized by a unique set of macroscopic cross sections) that will be defined in this execution of the MAC: module. The value effectively used by DRAGON will be the maximum between *nmixt* and the maximum mixture number defined on *MACLIB*.
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 from the input data stream.
nifiss	the maximum number of fissile spectrum 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 P_0 (isotropic) scattering cross sections. A value of *naniso*=2 indicates that P_1 (linearly anisotropic) scattering cross sections will be provided as input data.

*3.1.2 The **(descmaci)** input structure for* MAC:

The **(descmaci)** input structure takes the form:

Table 4: Structure **(descmaci)**

```
[ NGRO ngroup ]
[ CTRA { OFF | APOL | WIMS igroup } ]
[ NALBP 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) ]] [[ (descmpa) ]] } ]]
[ NORM ]
```

with

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 for which macroscopic cross sections will be provided. By default *ngroup*=1.

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 have access to this transport correction to produce transport corrected cross sections. By default, there is no transport correction when the MACROLIB is created from the input or GOXS files.

OFF — do not use the transport correction ($\Sigma^g_{tr,m}$) stored on the MACROLIB.

ON — use the transport correction ($\Sigma^g_{tr,m}$) stored on the MACROLIB.

APOL — keyword to specify that an APOLLO type transport correction ($\Sigma^g_{tr,m}$) based on the linearly anisotropic scattering cross sections will be computed and 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 P_1 scattering cross sections are available on the MACROLIB.

WIMS — keyword to specify that a WIMS-AECL type transport correction ($\Sigma^g_{tr,m}$) based on the linearly anisotropic scattering cross sections will be computed and 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 and WIMS-D4 69 groups libraries, *igroup*=27 and the micro-reversibility principle is used only for group 28 to 69.

NALBP — keyword to specify the maximum number of physical albedos that will be read. These can be used by the GEO: module (see Section 3.3).

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

ALBP — keyword used for the input of the group independent physical albedo arrays.

albedp group independent physical albedo arrays. 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 the first element in the array *energy* is considered to be the highest energy that can be reached by the neutron.

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.

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 3.1.4).

(descmpa) structure describing the format used for reading multigroup physical albedos from the input stream (see Section 3.1.5).

NORM keyword to specify that the fission spectrum will be normalized. This implies that the fission energy spectrum χ_D^g that will be stored in the output MACROLIB will satisfy:

$$\sum_{g=1}^{G} \chi_D^g = 1$$

This option is available even if the mixture cross sections were not read by the MAC: module.

*3.1.3 The **(descmacm)** input structure for MAC:*

The **(descmacm)** input structure takes the form:

Table 5: Structure **(descmacm)**

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

Here

CTRA	keyword to specify the transport correction option. All the modules that will read this MACROLIB will have access to this transport correction to produce transport corrected cross sections. By default there is no transport correction.
OFF	do not use the transport correction ($\Sigma^g_{tr,m}$) stored on the MACROLIB.
ON	use the transport correction ($\Sigma^g_{tr,m}$) stored on the MACROLIB.
MIX	keyword to specify that the macroscopic cross sections associated with a mixture are 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 that 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*.

3.1.4 Macroscopic cross section definition

Table 6: Structure **(descxs)**

MIX [*mixnum*]
[EFISS (*efiss*(i), i=1,*nifiss*)]
[TOTAL (*xssigt*(g), g=1,*ngroup*)]
[TRAN (*xssigtr*(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*)]
[FIXA (*xsfixa*(g), g=1,*ngroup*)]
[FIXG (*xsfixg*(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.

mixnum identifier for the next mixture to be read. The maximum value permitted for this identifier is *nmixt*. When *mixnum* is absent, the mixtures are numbered consecutively starting at 1 or after 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 isotope.

efiss energy (MeV) released per fission for each fissile spectrum.

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

xssigt multigroup total macroscopic cross sections (Σ_m^g in cm^{-1}) associated with mixture m.

`TRAN` keyword to specify that the macroscopic cross sections associated with the transport correction for this mixture follows.

xssigtr multigroup transport correction macroscopic cross sections ($\Sigma_{tr,m}^g$ in cm^{-1}) associated with mixture m.

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

xssigf multigroup macroscopic fission cross sections multiplied by the average number of neutrons per fission ($\nu\Sigma_{f,i,m}^g$ in cm^{-1}) for fissile spectrum i and mixture m.

`NFTOT` keyword to specify that the macroscopic fission cross sections for this mixture follows.

xsfiss multigroup macroscopic fission cross sections $\Sigma_{f,i,m}^g$ in cm^{-1}) for fissile spectrum i and mixture m.

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

xschi multigroup fission spectrum ($\chi_{I,m}^g$) for fissile spectrum i and mixture m.

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

xsfixe multigroup fixed neutron source densities (S_m^g in $s^{-1}\mathrm{cm}^{-3}$) for mixture m.

`FIXA` keyword to specify that adjoint fixed neutron source densities for this mixture follows.

xsfixa multigroup adjoint fixed neutron source densities ($S_{A,m}^g$ in cm^{-1}) for mixture m.

`FIXG` keyword to specify that fixed generalized adjoint neutron source densities for this mixture follows.

xsfixg multigroup generalized adjoint fixed neutron source densities ($S_{G,m}^g$ in cm^{-1}) for mixture m.

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

nbscat array that provides the number of groups (g) for which macroscopic scattering cross section ($\Sigma_{sl,m}^{g\to h}$) towards the group (h) will be provided for each anisotropy level associated with this mixture. The scattering cross sections for the remaining groups vanish.

ilastg array that provides the group index associated with the most thermal group for which macroscopic scattering cross section towards group h will be provided for each anisotropy level associated with this mixture.

xsscat multigroup macroscopic scattering cross sections ($\Sigma_{sl,m}^{g\rightarrow h}$ in cm^{-1}) from the scattering from group g towards group h. The elements are ordered in a decreasing order from group number g=*ilastg* to g =(*ilastg*-*nbscat*+1), and an increasing order from $h = 1$ to $h = G$. An example of an input structure for macroscopic scattering cross sections can be found in Section 4.1.

3.1.5 Multigroup physical albedo definition

Table 7: Structure **(descmpa)**

ALBPG [*ialbp*] (*albp*(g), g=1,*ngroup*)]

ALBPG keyword to specify that the multigroup physical albedos are to be read.

ialbp identifier for physical albedo to be read. The maximum value permitted for this identifier is *nalbp*.

albp multigroup data associated with physical albedo *ialbp*.

3.2 The LIB: module

The general format of the LIB: input data structure is the following:

Table 8: Structure **(LIB:)**

{ *MICLIB* := LIB: [*MICLIB*] :: **(desclib)** |
MICLIB := LIB: [*MICLIB* [*MICOLD*]] :: **(desclibupd)** |
MICLIB := LIB: [*MICLIB* [*BRNOLD*]] :: **(desclibbrn)** }

The first form is for the case where a single MICROLIB is involved (created or updated), the second form corresponds to the case where a MICROLIB is updated or created using the information available on a second MICROLIB, Finally, the third form is for the case where a MICROLIB is updated using the information available on a BURNUP data structure. The meaning of each of the terms above is:

MICLIB character*12 name of the data structure that will contain the MICROLIB.

MICOLD character*12 name of a read-only MICROLIB data structure. In the case where a second MICROLIB data structure is provided, the number densities for the isotopes in file *MICLIB* will be replaced selectively by those found in *MICOLD*.

BRNOLD character*12 name of a read-only BURNUP data structure. In the case where a BURNUP data structure is provided, the number densities for the isotopes in file *MICLIB* will be replaced selectively by those found in *BRNOLD*.

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

(desclibupd) input structure for updating the MICROLIB based on information available on a second MICROLIB data structure (see Section 3.2.2).

(desclibbrn) input structure for updating the MICROLIB based on information available on a BURNUP data structure (see Section 3.2.3).

3.2.1 General `LIB:` input structure

The general format of **(desclib)** is of the form:

Table 9: Structure **(desclib)**

[`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)**]]]]

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 mixtures. This data is required if *MICLIB* is created.

nmixt the maximum number of mixtures.

`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 no transport correction should be used in this calculation.

`APOL` keyword to specify that an APOLLO type transport correction ($\Sigma^g_{tr,m}$) based on the linearly anisotropic scattering cross sections will be computed and 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 P_1 scattering cross sections are available on the original library.

`OLDW` keyword to specify that a WIMS-AECL type transport correction ($\Sigma^g_{tr,m}$) based on the linearly anisotropic scattering cross sections will be computed and used for the total and isotropic scattering cross sections. This correction assumes that the micro-reversibility principle is valid only for groups with 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 WIMSD4 and WIMS-AECL libraries, linearly anisotropic scattering cross sections (diagonal term only) are generated in the cases where the transport correction differs from 0.0 and no anisotropic scattering cross sections are provided on the original library. This option was inserted for compatibility with the WIMS transport correction implemented in older versions of DRAGON.

`WIMS` keyword to specify that the transport correction ($\Sigma^g_{tr,m}$) is to be used for the total and isotropic scattering cross sections. This type of correction reads directly the transport correction cross sections provided on the original library. This information is available only in `WIMSD4` and `WIMS-AECL` format libraries. In the case where a library of another type is considered, this correction is identical to the `OLDW` option.

`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. The linearly anisotropic scattering contributions are generally 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 only prompt neutron are to be considered for the calculation of the fission spectrum. By default, the contributions due to delayed neutron are also 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 (see Section 3.9).

nedit number of types of additional cross sections.

HEDIT `character*6` name of an additional cross-section type. This name also corresponds to vector 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 weighted neutron total cross sections.
`NELAS`=Neutron elastic scattering cross sections (MT=2).
`NINEL`=Neutron inelastic scattering cross sections (MT=4).
`NG`=Neutron 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 NG, and NHEAT cross sections since they may be required for 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. These isotopes will be flagged as non-resonant. To ensure that the resonant isotopes are processed by the self-shielding module (see Section 3.5), they must be inserted manually in the mixture with 0.0 concentration and flagged as such (associate a resonant region to the isotopes).

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.

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. The energy group structure of the two libraries must be compatible. For burnup calculations, the depletion chain of the two libraries must also be compatible.

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.[27]

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. This keyword has the same meaning as the keyword WIMS.

FIL: keyword to specify the name of the file from which the isotopic depletion chain or microscopic cross section is to be read.

NAMEFIL character*64 name 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 3.2.4).

(descmix1) input structure describing the isotopic and physical properties of a given mixture (see Section 3.2.5).

3.2.2 Instruction for updating a MICROLIB *using a second* MICROLIB *in* `LIB:`

The general format of **(desclibupd)** is of the form:

Table 10: Structure **(desclibupd)**

[`EDIT` *iprint*]
`MAXS` [[**(descmix3)**]]

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.

`MAXS` keyword to specify that the mixture density on *MICLIB* are to be modified. If *MICOLD* is present and **(descmix3)** is absent, a direct one to one correspondence between the isotope on both libraries is assumed. If *MICOLD* and **(descmix3)** are present, only the mixture on the library file specified by **(descmix3)** are updated using information from the *MICOLD*. If *MICOLD* is absent and **(descmix3)** is present, only the mixture on *MICLIB* specified by **(descmix3)** are updated.

(descmix3) input structure describing perturbations to the isotopic and physical properties of a given mixture (see Section 3.2.7).

3.2.3 Instruction for updating a MICROLIB *using a* BURNUP *in* `LIB:`

The general format of **(desclibbrn)** is of the form:

Table 11: Structure **(desclibbrn)**

[`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.

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

iburn burnup step selected on *BRNOLD*. This step must be present on the burnup file.

tburn burnup time in days on *BRNOLD*. This time step must be present on the burnup file.

(descmix2) input structure describing perturbations to the isotopic and physical properties of a given mixture (see Section 3.2.6).

3.2.4 Depletion data structure

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

Table 12: Structure **(descdepl)**

```
CHAIN
[[ NAMDPL [ izae ]
      [[ { DECAY dcr |
            reaction [ energy ] } ]]
      [ { STABLE |
            FROM [[ { DECAY | reaction } [[ yield NAMPAR ]] ]] } ] ]]
ENDCHAIN
```

with:

CHAIN keyword to specify the beginning of the depletion chain.

NAMDPL `character*12` 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 represent 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.

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

reaction `character*6` identification of a neutron-induced reaction that takes place either for production of this isotope, its depletion, or for producing energy. Examples of reactions follow:

`NG` a radiative capture reaction takes place either for production of this isotope, its depletion or for producing energy.

`N2N` represents the reaction:

$$ {}^{1}_{0}n + {}^{A}_{Z}\mathrm{X} \rightarrow 2\,{}^{1}_{0}n + {}^{A-1}_{Z}\mathrm{X} $$

`N3N` represents the reaction:

$$ {}^{1}_{0}n + {}^{A}_{Z}\mathrm{X} \rightarrow 3\,{}^{1}_{0}n + {}^{A-2}_{Z}\mathrm{X} $$

`N4N` represents the reaction:

$$ {}^{1}_{0}n + {}^{A}_{Z}\mathrm{X} \rightarrow 4\,{}^{1}_{0}n + {}^{A-3}_{Z}\mathrm{X} $$

`NP` represents the reaction:

$$ {}^{1}_{0}n + {}^{A}_{Z}\mathrm{X} \rightarrow {}^{1}_{1}p + {}^{A}_{Z-1}\mathrm{Y} $$

`NA` represents the reaction:

$$ {}^{1}_{0}n + {}^{A}_{Z}\mathrm{X} \rightarrow {}^{4}_{2}\mathrm{He} + {}^{A-3}X_{Z-2}\mathrm{Y} $$

`NFTOT` a fission reaction takes place.

energy energy (in MeV) recoverable per neutron-induced reaction. If the energy production associated with radiative captures is not provided explicitly, it should be included in the energy released by fission. By default, *energy*=0.0 MeV.

`STABLE` non depleting isotope. Such an isotope may produces energy by neutron-induced reactions (such as radiative capture).

`FROM` indicates that this isotope is produced from decay or neutron-induced reactions.

yield production yield for fission (expressed in %) or for other reaction (given in absolute value).

NAMPAR `character*12` name of the parent isotope (or isomer) that appears in the cross-section library.

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

*3.2.5 Format for **(descmix1)***

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

Table 13: Structure **(descmix1)**

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

where:

`MIX` keyword to specify the number identifying the 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 $\mathrm{g}\times\mathrm{cm}^{-3}$.

NAMALI `character*8` alias name to be used locally for an isotope. 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 When the mixture density *denmix* is specified, this parameter is the relative weight percentage of the isotope in this mixture. Otherwise, the parameter is the isotopic concentration of the isotope *NAMISO* in the mixture in $10^{24}\times\mathrm{cm}^{-3}$.

dil group independent microscopic dilution cross section (in barns) of the isotope *NAMISO* in this mixture. The group dependent dilution for an isotope can be computed using 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 10^{10} barns, an infinite dilution is assumed.

`INF` keyword to specify that an infinite dilution (10^{10} barns) is to be associated with this isotope. This value implie that the isotope is present in trace amounts only.

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-Jeanpierre. Should one 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*12` 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 that will be taken into account. The coherent effects are the *vector 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.

`NOEV` keyword to force a mixture or a nuclide to be non-depleting (even in cases where it is potentially depleting). Note that the mixtures or nuclides keep their capability to produce energy.

`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, is is replaced by void.

relvol relative volume V_m occupied by mixture m=*mati* in *matnum*. Two cases can be considered.

1. The density ρ_m of each mixture *mati* is provided along with the weight percent for each isotopes J (W_m^J). In this case the density (ρ_k) and volume (V_k) of the final mixture become

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

   The weight percent of isotope J for the combined mixture is then given by

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

2. The explicit concentration N_m^J of each isotope J in each material m is provided and

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

   It is forbidden to combine two mixtures with different isotopic content description.

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`. Also assume 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`. Remember that the isotopes added automatically will remain at infinite dilution (not affected by resonance self-shielding calculations).

*3.2.6 Format for **(descmix2)***

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

Table 14: Structure **(descmix2)**

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

where:

`MIX` keyword to specify the number identifying the next mixture to be updated. If no mixture is specified then all the mixtures are updated.

matnum mixture identifier on *MICLIB*.

matold mixture identifier on *BRNOLD*. When *matold* is not specified this mixture is not updated.

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 $10^{24}\times\text{cm}^{-3}$. When *dens*$\geq 0$, the isotopic concentration for this isotope becomes *dens*, while all the other isotopes take the value specified on *BRNOLD*. When *dens*$= -1.0$, the isotopic concentration of this isotope is not updated.

*3.2.7 Format for **(descmix3)***

The structure **(descmix3)** is used to describe the modifications in the isotopic composition of a mixture taken from an old MICROLIB data structure.

Table 15: Structure **(descmix3)**

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

where:

`MIX` keyword to specify the number identifying the next mixture to be updated. If no mixture is specified then all the mixtures are updated.

matnum mixture identifier on *MICLIB*.

matold mixture identifier on *MICOLD*. When *matold* is not specified this mixture is not updated.

relden relative density of updated mixture. The concentration of each isotope in the mixture will be

multiplied by this factor independent of the fact that the original concentrations were defined in *MICLIB*, *MICOLD* 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 $10^{24}\times\text{cm}^{-3}$. When *dens*$\geq 0$, the isotopic concentration for this isotope becomes *dens*$\times$*relden*, while all the other isotopes are multiplied by *relden* only. When *dens*$= -1.0$, the isotopic concentration of this isotope is not updated while all the other isotope concentrations are multiplied by *relden*.

3.3 The GEO: module

The GEO: module is used to create or modify a geometry. All the characteristics (dimensions, region contents and boundary conditions) of simple or complex geometries are specified using this module. The specifications of the geometry are independent of the tracking module to be used subsequently. Each geometry is stored in a GEOMETRY data structure under its given name. Once a geometry has been specified, it can be updated through a new call to the GEO: module. The calling specifications to create a geometry with the GEO: module are provided in Table 16 while Table 17 and Table 18 describe respectively the format required to modify or import a geometry.

Table 16: Structure **(GEO:)** to create a geometry

GEONAM `:=` `GEO: ::` **(descgtyp)**
(descgcnt)

Table 17: Structure **(GEO:)** to modify an existing geometry

GEONAM `:=` `GEO:` *GEONAM* `::`
(descgcnt)

Table 18: Structure **(GEO:)** to import and modify an existing geometry

GEONAM `:=` `GEO:` *OLDGEO* `::`
(descgcnt)

The definition of the parameters used in Tables 16 to 18 follows.

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* are first copied to *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 3.3.2).

3.3.1 Geometry types

The structure **(descgtyp)** presented in Table 19 is used to define the type of geometry that will be considered.

Table 19: Structure **(descgtyp)**

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

where

VIRTUAL keyword to specify a virtual geometry. This type of geometry is used to complete an assembly that has irregular boundaries.

HOMOGE keyword to specify a infinite homogeneous geometry.

SPHERE keyword to specify a spherical geometry (concentric spheres). This option is only supported by the SYBILT: tracking module (see Section 3.4).

TUBE keyword to specify a 2-D cylindrical (infinite tubes or cylinders) geometry. This geometry can contain an imbedded $X - Y$ Cartesian mesh.

TUBEX keyword to specify a 3-D cylindrical (along the X axis) geometry. This geometry can contain an imbedded $X-Y-Z$ Cartesian mesh. This option is only supported for cluster sub-geometries in the NXT: tracking module (see Section 3.4).

TUBEY keyword to specify a 3-D cylindrical (along the Y axis) geometry. This geometry can contain an imbedded $X-Y-Z$ Cartesian mesh. This option is only supported for cluster sub-geometries in the NXT: tracking module (see Section 3.4).

TUBEZ keyword to specify a 3-D cylindrical (along the Z axis) geometry. This geometry can contain an imbedded $X-Y-Z$ Cartesian mesh.

CAR1D keyword to specify a single Cartesian cell geometry in 1-D with an imbedded Z mesh or an assembly of 1-D Cartesian cells. This option is only supported by the SYBILT: tracking module (see Section 3.4).

CAR2D keyword to specify a single Cartesian cell geometry in 2-D with an imbedded $X-Y$ mesh or an assembly of 2-D Cartesian cells.

CARCEL keyword to specify a 2-D Cartesian cell geometry with embedded annular regions (concentric tubes surrounded by a rectangle).

CAR3D keyword to specify a single Cartesian cell geometry in 3-D with an imbedded $X-Y-Z$ mesh or an assembly of 3-D Cartesian cells.

CARCELX keyword to specify a 3-D Cartesian cell geometry with embedded cylinders oriented along the $X-$axis.

CARCELY keyword to specify a 3-D Cartesian cell geometry with embedded cylinders oriented along the $Y-$axis.

CARCELZ keyword to specify a 3-D Cartesian cell geometry with embedded cylinders oriented along the $Z-$axis.

HEX keyword to specify a single 2-D hexagonal cell geometry or a 2-D assembly of hexagonal cells.

HEXT keyword to specify a single 2-D hexagonal cell geometry having a triangular mesh. This option is only supported by the NXT: tracking module (see Section 3.4).

HEXCEL keyword to specify a single 2-D hexagonal cell geometry containing concentric annular regions. This option is not supported by the NXT: tracking module (see Section 3.4).

HEXTCEL keyword to specify a single 2-D hexagonal cell geometry having a triangular mesh and containing concentric annular regions.

HEXZ keyword to specify a single Z directed 3-D hexagonal cell geometry or a 3-D assembly of Z directed hexagonal cells.

HEXTZ keyword to specify a single Z directed 3-D hexagonal cell geometry having a triangular mesh (plane $X-Y$). This option is only supported by the NXT: tracking module (see Section 3.4).

HEXCELZ keyword to specify a single Z directed 3-D hexagonal cell geometry containing concentric Z directed cylinders. This option is not supported by the NXT: tracking module (see Section 3.4).

HEXTCELZ keyword to specify a single Z directed 3-D hexagonal cell geometry a triangular mesh and containing concentric Z directed cylinders.

GROUP keyword to specify a *do-it-yourself* type geometry.

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).

nhr number of concentric hexagons in a `HEXT` or `HEXTZ` cell (see Figure 1). This will lead to an hexagon subdivided into $6N^2$ identical trangles.

lp number of types of cells (number of cells inside which a distinct flux will be calculated) for a *do-it-yourself* type geometry.

3.3.2 Geometry contents

The structure **(descgcnt)** illustrated in Table 20 is used to define the contents of a geometry (dimensions, materials, boundary conditions). As one can see, sub-geometries included in a geometry can also be defined by calling recursively the module `GEO:` from **(descgcnt)** (embedded module). Currently, geometries definitions are limited to 4 recursion levels and most tracking modules can only process the first two levels.

Table 20: Structure **(descgcnt)**

[`EDIT` *iprint*]
(descBC)
(descSP)
(descPP)
(descNSG)
[[`:::` *SUBGEO* `:=` `GEO:` { **(descgtyp)** | *SUBGEO* | *OLDGEO* } **(descgcnt)** `;`]]

The following notation is used in Table 20:

`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 3.3.3).

(descSP) structure defining the spatial coordinates associated with a geometry (see Section 3.3.4).

(descPP) structure defining the physical properties associated with a geometry (see Section 3.3.5).

(descNSG) structure used to specify the properties of non standard geometries (see Section 3.3.6).

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 to *SUBGEO* before the updates specified by **(descgcnt)** are applied.

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

(descgcnt) structure describing the characteristics of *SUBGEO* (see Section 3.3.2).

3.3.3 Boundary conditions

The data corresponding to the **(descBC)** structure is presented in Table 21.

Table 21: Structure **(descBC)**

```
[ X- { VOID | REFL | SSYM | TRAN | SYME | ALBE { albedo | icode } | DIAG } ]
[ X+ { VOID | REFL | SSYM | TRAN | SYME | ALBE { albedo | icode } | DIAG } ]
[ Y- { VOID | REFL | SSYM | TRAN | SYME | ALBE { albedo | icode } | DIAG } ]
[ Y+ { VOID | REFL | SSYM | TRAN | SYME | ALBE { albedo | icode } | DIAG } ]
[ Z- { VOID | REFL | SSYM | TRAN | SYME | ALBE { albedo | icode } } ]
[ Z+ { VOID | REFL | SSYM | 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 } } ]
```

Here:

`X-` keyword to specify the boundary conditions associated with the negative X surface in a Cartesian geometry.

`X+` keyword to specify the boundary conditions associated with the positive X surface in a Cartesian geometry.

`Y-` keyword to specify the boundary conditions associated with the negative Y surface in a Cartesian geometry.

`Y+` keyword to specify the boundary conditions associated with the positive Y surface in a Cartesian geometry.

`Z-` keyword to specify the boundary conditions associated with the negative Z surface in a Cartesian geometry.

`Z+` keyword to specify the boundary conditions associated with the positive Z surface in 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. For most tracking modules, this implies white boundary conditions. The main exceptions to this rule are

- 1-D Cartesian geometries analyzed using `SYBILT:`.

- 2-D Cartesian geometries processed by EXCELT: or NXT: using the cyclic tracking technique.

In both cases, mirror like rather than white boundary conditions are considered. In DRAGON assemblies are never unfolded to take into account a REFL boundary condition.

SSYM keyword to specify that the surface under consideration has a reflective boundary condition. The main difference between REFL and SSYM is that the later option results in the assembly being unfolded along the reflective boundary. Accordingly, SSYM implies the use of a mirror like reflection.

TRAN keyword to specify that periodic boundary conditions are considered. The surface under consideration is therefore connected to the opposite surface in the Cartesian domain. The only combinations of periodic boundary conditions permitted are:

- Periodicity along the $X-$axis

```
X- TRAN X+ TRAN
```

- Periodicity along the $Y-$axis

```
Y- TRAN Y+ TRAN
```

- Periodicity 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 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. For most calculations, this implies white boundary conditions. The main exception to this rule is when cyclic tracking in 2-D is considered. Note that the cell is never unfolded to take into account a ALBE boundary condition.

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 2). This option is not yet supported by the NXT: tracking module (see Section 3.4).

SA60 keyword to specify an hexagonal symmetry of one sixth of an assembly of type A (see Figure 2). This option is not yet supported by the NXT: tracking module (see Section 3.4).

`SB60` keyword to specify an hexagonal symmetry of one sixth of an assembly of type B (see Figure 3). This option is not yet supported by the `NXT:` tracking module (see Section 3.4).

`S90` keyword to specify an hexagonal symmetry of one quarter of an assembly (see Figure 3). This option is not yet supported by the `NXT:` tracking module (see Section 3.4).

`R120` keyword to specify a rotation symmetry of one third of an assembly (see Figure 4). This option is not yet supported by the `NXT:` tracking module (see Section 3.4).

`R180` keyword to specify a rotation symmetry of a half assembly (see Figure 4). This option is not yet supported by the `NXT:` tracking module (see Section 3.4).

`SA180` keyword to specify an hexagonal symmetry of half a type A assembly (see Figure 5). This option is not yet supported by the `NXT:` tracking module (see Section 3.4).

`SB180` keyword to specify an hexagonal symmetry of half a type B assembly (see Figure 6). This option is not yet supported by the `NXT:` tracking module (see Section 3.4).

`COMPLETE` keyword to specify a complete hexagonal assembly (see figure Figure 7).

3.3.4 Spatial description of geometry

The **(descSP)** structure is described in Table 22.

Table 22: 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)]
[`SPLITR` (*ispltr*(i), i=1,*lr*)]
[`OFFCENTER` (*disxyz*(i), i=1,3)]
[`SIDE` *sideh* [*hexmsh*]]
[`SPLITH` *isplth*]
[{ `NPIN` *npins*
{ [`RPIN` { *rpins* | (*rpins*(i), i=1, *npins*) }]
[`APIN` { *apins* | (*apins*(i), i=1, *npins*) }] |
[`CPINX` (*xpins*(i), i=1, *npins*)]
[`CPINY` (*ypins*(i), i=1, *npins*)]
[`CPINZ` (*zpins*(i), i=1, *npins*)] }
| `DPIN` *dpins* }]

Here

`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 important to note that we must have *rrr*(1)=0.0.

`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.

`OFFCENTER` keyword to specify that the concentric annular regions in `CARCEL`, `CARCELX`, `CARCELY` and `CARCELZ` geometries are displaced with respect to the center of the Cartesian mesh. This displacement also affects the `TUBE`, `TUBEX`, `TUBEY` and `TUBEZ` pin clusters locations. This option is only processed by the `EXCELT:`, `NXT:` and `EXCELL:` tracking modules.

disxyz array giving the x (*disxyz*(1)), y (*disxyz*(2)) and z (*disxyz*(3)) displacements (cm) of the concentric annular regions with respect to the center of the Cartesian mesh.

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

sideh length of one side of a hexagon (cm).

hexmsh triangular mesh for `HEXT` and `HEXTZ` hexagonal geometries. By default, *hexmsh*=*sideh*/*nhr*. When *hexmsh* is provided, it is used instead of the default value with the following constraints

$$sideh \leq nhr \times hexmsh < sideh + hexmsh$$

The triangles in the last hexagonal ring are truncated at *sideh* (see Figure 8).

`SPLITH` keyword to specify that a triangular mesh splitting level for `HEXT` and `HEXTZ` type geometries is to be considered. This is valid only if *nhr*=1.

isplth value of the triangular mesh splitting. Its use is similar to *nhr* except that each sector of the hexagonal cell will be filled by a unique mixture.

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

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

`DPIN` keyword to specify the pin density in a geometry that contains clusters. A number $N_{p,r}$ of pins that will be placed randomly in the geometry with

$$N_{p,r} = \text{NINT}\left(\frac{d_{p,r}V_c}{V_p}\right)$$

where $d_{p,r}$ is the pin density, V_g the volume of the cell containing these pins and V_p the volume of this pin type. The function NINT() provides the nearest integer associated with its real argument. It can only be used for `SPHERE`, `TUBE`, `TUBEX`, `TUBEY` and `TUBEZ` sub-geometry.

dpins the pin density $d_{p,r}$.

`RPIN` keyword to specify the radius of an imaginary cylinder where the centers of the pins are to be placed in a cluster geometry.

rpins the radius (cm) of an imaginary cylinder where the centers of the pins are to be placed. In the case where a single value is provided for *rpins*, all the pins are located at the same distance from the center of the cell (taking account the offset provided by the keyword `OFFCENTER`).

`APIN` keyword to specify the angle of the first pin or each pin centered on an imaginary cylinder in a cluster geometry.

apins the angle (radian) of the first pin in the ring if only one value is provided for *apins* (angular spacing of the pins being $2\pi/$*npins*) or the angle of each pins in the ring.

`CPINX` keyword to specify the x position where the centers of the pins are to be placed in a cluster geometry.

xpins the x position (cm) where the centers of the pins are to be placed.

`CPINY` keyword to specify the y position where the centers of the pins are to be placed in a cluster geometry.

ypins the y position (cm) where the centers of the pins are to be placed.

`CPINZ` keyword to specify the z position where the centers of the pins are to be placed in a cluster geometry.

zpins the z position (cm) where the centers of the pins are to be placed.

The user should be warned that the maximum number of zones resulting from the above description of a geometry L_{zones} 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, L_h and L_t as:

$$L_x = \sum_{i=1}^{lx} \mathit{ispltx}(i)$$
$$L_y = \sum_{i=1}^{ly} \mathit{isplty}(i)$$
$$L_z = \sum_{i=1}^{lz} \mathit{ispltz}(i)$$
$$L_r = \sum_{i=1}^{lr} |\mathit{ispltr}(i)|$$
$$L_h = \mathit{lh}$$
$$L_t = \begin{cases} 6 \times \mathit{nhr}^2 & \text{if } \mathit{nhr} > 1 \\ 6 \times \mathit{isplith}^2 & \text{otherwise} \end{cases}$$

and L_{zones} will be given by:

- `SPHERE` geometry.

$$L_{\text{zones}} = L_r$$

- `TUBE` geometry.

$$L_{\text{zones}} = L_x L_y L_r$$

- `TUBEX` geometry.

$$L_{\text{zones}} = L_x L_y L_z L_r$$

- `TUBEY` geometry.

$$L_{\text{zones}} = L_x L_y L_z L_r$$

- `TUBEZ` geometry.

$$L_{\text{zones}} = L_x L_y L_z L_r$$

- `CAR1D` geometry.

$$L_{\text{zones}} = L_x$$

- CAR2D geometry
  - without diagonal symmetry.

$$L_{\text{zones}} = L_x L_y$$

  - with diagonal symmetry.

$$L_{\text{zones}} = \frac{L_x(L_y+1)}{2} = \frac{(L_x+1)L_y}{2}$$

- CARCEL geometries.

$$L_{\text{zones}} = L_x L_y (L_r+1)$$

- CAR3D geometry
  - without diagonal symmetry.

$$L_{\text{zones}} = L_x L_y L_z$$

  - with diagonal symmetry.

$$L_{\text{zones}} = \frac{L_x(L_y+1)L_z}{2} = \frac{(L_x+1)L_y L_z}{2}$$

- CARCELX geometry.

$$L_{\text{zones}} = L_x L_y L_z (L_r+1)$$

- CARCELY geometry.

$$L_{\text{zones}} = L_x L_y L_z (L_r+1)$$

- CARCELZ geometries.

$$L_{\text{zones}} = L_x L_y L_z (L_r+1)$$

- HEX geometry.

$$L_{\text{zones}} = L_h$$

- HEXT geometry.

$$L_{\text{zones}} = L_t$$

- `HEXCEL` geometries.

$$L_{\text{zones}} = (L_r + 1)$$

- `HEXTCEL` geometries.

$$L_{\text{zones}} = L_t(L_r + 1)$$

- `HEXZ` geometry.

$$L_{\text{zones}} = L_z L_h$$

- `HEXTZ` geometry.

$$L_{\text{zones}} = L_z L_t$$

- `HEXCELZ` geometries.

$$L_{\text{zones}} = L_z(L_r + 1)$$

- `HEXTCELZ` geometries.

$$L_{\text{zones}} = L_z L_t(L_r + 1)$$

For cluster geometries, only one region is associated with each zone in a pin even if this pin is repeated *npins* times.

3.3.5 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-geometries required in this geometry. An optional procedure can also be used to groups together regions so as to reduce the number of unknowns 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 dimensions and contain the same mixtures.
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 contents of the **(descPP)** structure is presented in Table 23.

Table 23: Structure **(descPP)**

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

Here N_p is the number of pin types in the cluster. In addition to the real (physical) mixture *imix* present in a given region of space and specified by the keyword MIX, a virtual mixture *ihmix* can also be provided using the keyword HMIX. This mixture can be used to identify the regions that will be combined in the EDI: module to create homogenized region *ihmix* (see Section 3.9). Here N_t is computed in a way similar to L_{zones} namely

- SPHERE geometry.

$$N_t = lr$$

  The real and virtual mixtures are given in the following order

  1. radially outward ($l = 1, lr$).

- TUBE geometry.

$$N_t = lr \times lx \times ly$$

  The real and virtual mixtures are given in the following order

  1. radially outward ($l = 1, lr$) and such that *imix* (*ihmix*) is arbitrary (not used) if radial region l does not intersect Cartesian region (i, j);
  2. from surface X- to surface X+ ($i = 1, lx$ for each j);
  3. from surface Y- to surface Y+ ($j = 1, ly$).

- TUBEX geometry.

$$N_t = lr \times ly \times lz \times lx$$

  The real and virtual mixtures are given in the following order

  1. radially outward ($l = 1, lr$) and such that *imix* (*ihmix*) is arbitrary (not used) if radial region l does not intersect Cartesian region (j, k, i);
  2. from surface Y- to surface Y+ ($j = 1, ly$ for each k and i);
  3. from surface Z- to surface Z+ ($k = 1, lz$ for each i);
  4. from surface X- to surface X+ ($i = 1, lx$).

- TUBEY geometry.

$$N_t = lr \times lz \times lx \times ly$$

The real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr$) and such that *imix* (*ihmix*) is arbitrary (not used) if radial region l does not intersect Cartesian region (k, i, j);
2. from surface Z- to surface Z+ ($k = 1, lz$ for each i and j);
3. from surface X- to surface X+ ($i = 1, lx$ for each j);
4. from surface Y- to surface Y+ ($j = 1, ly$).

- TUBEZ geometry.

$$N_t = lr \times lx \times ly \times lz$$

The real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr$) and such that *imix* (*ihmix*) is arbitrary (not used) if radial region l does not intersect Cartesian region (i, j, k);
2. from surface X- to surface X+ ($i = 1, lx$ for each j and k);
3. from surface Y- to surface Y+ ($j = 1, ly$ for each k);
4. from surface Z- to surface Z+ ($k = 1, lz$).

- CAR1D geometry.

$$N_t = lx$$

The real and virtual mixtures are then given in the following order

1. from surface X- to surface X+ ($i = 1, lx$).

- CAR2D geometry
  - without diagonal symmetry.

$$N_t = lx \times ly$$

The real and virtual mixtures or cells are then given in the following order

1. from surface X- to surface X+ ($i = 1, lx$ for each j);
2. from surface Y- to surface Y+ ($j = 1, ly$).

  - with diagonal symmetry (X- and Y+).

$$N_t = \frac{lx \times (lx + 1)}{2}$$

The real and virtual mixtures or cells are then given in the following order

1. from surface X- to surface X+ ($i = j, lx$ for each j);
2. from surface Y- to surface Y+ ($j = 1, ly$).

– with diagonal symmetry (X+ and Y-).

$$N_t = \frac{lx \times (lx + 1)}{2}$$

The real and virtual mixtures or cells are then given in the following order

1. from surface X- to surface X+ ($i = 1, j$ for each j);
2. from surface Y- to surface Y+ ($j = 1, ly$).

- CARCEL geometries.

$$N_t = (lr + 1) \times lx \times ly$$

The real and virtual mixtures are then given in the following order

1. radially outward ($l = 1, lr$) and such that *imix* (*ihmix*) is arbitrary (not used) if radial region l does not intersect Cartesian region (i, j);
2. $l = lr+1$ for the mixture outside the annular regions but inside Cartesian region (i, j);
3. from surface X- to surface X+ ($i = 1, lx$ for each j);
4. from surface Y- to surface Y+ ($j = 1, ly$).

- CAR3D geometry

– without diagonal symmetry.

$$N_t = lx \times ly \times lz$$

The real and virtual mixtures or the cells are then given in the following order

1. from surface X- to surface X+ ($i = 1, lx$ for each j and k);
2. from surface Y- to surface Y+ ($j = 1, ly$ for k);
3. from surface Z- to surface Z+ ($k = 1, lz$).

– with diagonal symmetry (X- and Y+).

$$N_t = \frac{lx \times (lx + 1)}{2} \times lz$$

The real and virtual mixtures or the cells are then given in the following order

1. from surface X- to surface X+ ($i = j, lx$ for each j and k);
2. from surface Y- to surface Y+ ($j = 1, ly$) for each k);
3. from surface Z- to surface Z+ ($k = 1, lz$).

– with diagonal symmetry (X+ and Y-).

$$N_t = \frac{lx \times (lx + 1)}{2} \times lz$$

The real and virtual mixtures or the cells are then given in the following order

1. from surface X- to surface X+ ($i = 1, j$ for each j and k);
2. from surface Y- to surface Y+ ($j = 1, ly$ for each k);
3. from surface Z- to surface Z+ ($k = 1, lz$).

- CARCELX geometry.

$$N_t = (lr + 1) \times ly \times lz \times lx$$

The real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr$) and such that *imix* (*ihmix*) is arbitrary (not used) if radial region l does not intersect Cartesian region (j, k, i);
2. $l = lr+1$ for the mixture outside the annular regions but inside Cartesian region (j, k, i);
3. from surface Y- to surface Y+ ($j = 1, ly$ for each k and i);
4. from surface Z- to surface Z+ ($k = 1, lz$ for each i);
5. from surface X- to surface X+ ($i = 1, lx$).

- CARCELY geometry.

$$N_t = (lr + 1) \times lz \times lx \times ly$$

The real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr$) and such that *imix* (*ihmix*) is arbitrary (not used) if radial region l does not intersect Cartesian region (k, i, j);
2. $l = lr+1$ for the mixture outside the annular regions but inside Cartesian region (k, i, j);
3. from surface Z- to surface Z+ ($k = 1, lz$ for each i and j);
4. from surface X- to surface X+ ($i = 1, lx$ for each j);
5. from surface Y- to surface Y+ ($j = 1, ly$).

- CARCELZ geometries.

$$N_t = (lr + 1) \times lx \times ly \times lz$$

The real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr$) and such that *imix* (*ihmix*) is arbitrary (not used) if radial region l does not intersect Cartesian region (i, j, k);
2. $l = lr+1$ for the mixture outside the annular regions but inside Cartesian region (i, j, k);
3. from surface X- to surface X+ ($i = 1, lx$ for each j and k);
4. from surface Y- to surface Y+ ($j = 1, ly$ for each k).
5. from surface Z- to surface Z+ ($k = 1, lz$).

- HEX geometry.

$$N_t = lh$$

The real and virtual mixtures or the cells are given in the order provided in Figures 2 to 7.

- HEXT geometry.

  Three options are possible here:

  – All the triangles in an hexagonal crown have the same mixture. In this case

  $$N_t = nhr$$

  and the real and virtual mixtures are given from each crown starting at the center of the cell.

  – All the triangles in an hexagonal crown in a given sector have the same mixture. In this case

  $$N_t = 6 \times nhr$$

  and the real and virtual mixtures are given in the following order

  1. from each crown in sector j starting from the center of the cell;
  2. for each sector $j = 1, 6$.

  – All the triangles contain a different mixture. In this case

  $$N_t = 6 \times nhr^2$$

  and the real and virtual mixtures are given in the following order

  1. from each triangle l ($l = 1, 2 \times nhc - 1$) in hexagonal crown i of sector j. Figure 1 illustrates region and surface ordering in the case where the default value of *hexmsh* is used and Figure 8 the same information when a different value of *hexmsh* is provided.
  2. from each crown in sector j starting from the center of the cell;
  3. for each sector $j = 1, 6$.

- HEXCEL geometries.

  $$N_t = (lr + 1)$$

  The real and virtual mixtures are given in the following order

  1. radially outward ($l = 1, lr$);
  2. $l = lr+1$ for the mixture outside the annular regions but inside the hexagon.

- HEXTCEL geometries.

  Three options are possible here:

  – All the triangles in an hexagonal crown have the same mixture. In this case

  $$N_t = (lr + 1) \times nhr$$

  and the real and virtual mixtures are given in the following order

  1. radially outward ($l = 1, lr + 1$) for each crown ($l = lr + 1$ is for the part of crown outside the annular regions);
  2. from each crown starting from the center of the cell.

  – All the triangles in an hexagonal crown in a given sector have the same mixture. In this case

  $$N_t = 6 \times (lr + 1) \times nhr$$

  and the real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr + 1$) for each crown of each sector ($l = lr + 1$ is for the part of crown outside the annular regions);
2. from each crown in sector j starting from the center of the cell;
3. for each sector $j = 1, 6$.

– All the triangles contain a different mixture. In this case

$$N_t = 6 \times (lr + 1) \times nhr^2$$

and the real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr + 1$) for each triangle ($l = lr + 1$ is for the part of triangle outside the annular regions);
2. from each triangle l ($l = 1, 2 \times nhc - 1$) in hexagonal crown i of sector j. Figure 1 illustrates region and surface ordering in the case where the default value of *hexmsh* is used and Figure 8 the same information when a different value of *hexmsh* is provided.
3. from each crown in sector j starting from the center of the cell;
4. for each sector $j = 1, 6$.

- HEXZ geometry.

$$N_t = lh \times lz$$

The real and virtual mixtures or the cells are given in the following order

1. according to Figures 2 to 7 for plane k;
2. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

- HEXTZ geometry.

Three options are again possible here:

– All the triangles in an hexagonal crown in a plane have the same mixture. In this case

$$N_t = nhr \times lz$$

and the real and virtual mixtures are given in the following order

1. from each crown starting from the center of the cell;
2. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

– All the triangles in an hexagonal crown in a given sector in a plane have the same mixture. In this case

$$N_t = 6 \times nhr \times lz$$

and the real and virtual mixtures are given in the following order

1. from each crown in sector j starting from the center of the cell;
2. for each sector $j = 1, 6$;
3. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

– All the triangles contain a different mixture. In this case

$$N_t = 6 \times nhr^2 \times lz$$

and the real and virtual mixtures are given in the following order

1. from each triangle l ($l = 1, 2 \times nhc - 1$) in hexagonal crown i of sector j. Figure 1 illustrates region and surface ordering in the case where the default value of *hexmsh* is used and Figure 8 the same information when a different value of *hexmsh* is provided.
2. from each crown in sector j starting from the center of the cell;
3. for each sector $j = 1, 6$;
4. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

- HEXCELZ geometries.

$$N_t = (lr + 1) \times lz$$

The real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr$) for plane k;
2. $l = lr+1$ for the mixture outside the annular regions but inside the hexagonal region on plane k;
3. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

- HEXTCELZ geometries.

$$N_t = 6 \times nhr^2 \times (lr + 1) \times lz$$

The real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr$) for plane k;
2. $l = lr+1$ for the mixture outside the annular regions but inside a triangle.
3. from each triangle l ($l = 1, 2 \times nhc - 1$) in hexagonal crown i of sector j. Figure 1 illustrates the triangular regions and external surfaces ordering in each z plane in the case where the default value of *hexmsh* is used and Figure 8 the same information when a different value of *hexmsh* is provided.
4. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

- HEXTCELZ geometries.

Three options are possible here:

– All the triangles in an hexagonal crown have the same mixture. In this case

$$N_t = (lr + 1) \times nhr \times lz$$

and the real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr + 1$) for each crown ($l = lr + 1$ is for the part of crown outside the annular regions);
2. from each crown starting from the center of the cell;
3. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

– All the triangles in an hexagonal crown in a given sector have the same mixture. In this case

$$N_t = 6 \times (lr + 1) \times nhr \times lz$$

and the real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr + 1$) for each crown of each sector ($l = lr + 1$ is for the part of crown outside the annular regions);

2. from each crown in sector j starting from the center of the cell;
3. for each sector $j = 1, 6$;
4. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

– All the triangles contain a different mixture. In this case

$$N_t = 6 \times (lr + 1) \times nhr^2 \times lz$$

and the real and virtual mixtures are given in the following order

1. radially outward ($l = 1, lr + 1$) for each triangle ($l = lr + 1$ is for the part of triangle outside the annular regions);
2. from each triangle l ($l = 1, 2 \times nhc - 1$) in hexagonal crown i of sector j. Figure 1 illustrates region and surface ordering in the case where the default value of *hexmsh* is used and Figure 8 the same information when a different value of *hexmsh* is provided.
3. from each crown in sector j starting from the center of the cell;
4. for each sector $j = 1, 6$.
5. from lowest (Z-) to highest (Z+) plane ($k = 1, lz$).

The meaning of the parameters presented in Table 23 follows:

MIX — keyword to specify the real (physical) isotopic mixture number associated with each region inside the geometry. When diagonal symmetries are considered, only the mixture associated with regions inside the original geometry need to be specified. Here $n_t \leq N_t$.

imix — array of real (physical) mixture numbers associated with a region. If *imix*=0, the corresponding volume is filled with void.

HMIX — keyword to specify the virtual mixture number associated with each region inside the geometry. When diagonal symmetries are considered, only the mixture associated with regions inside the original geometry need to be specified. Here $n_t \leq N_t$.

ihmix — array of virtual mixture numbers associated with each region. This information is processed by the NXT: tracking module (see Section 3.4) for use by the EDI: module (see Section 3.9). In the case where *ihmix*=0, the corresponding volume is not considered in the homogenization process.

REPEAT — keyword to specify that the previous list of mixtures (real or virtual) will be repeated. This is valid only when N_t/n_t is an integer. If this keyword is absent and $n_t < N_t$, then the missing mixtures will be replaced with void (*imix*=*ihmix*=0).

CELL — keyword to specify the sub-geometries (*generating cells*) that fill a Cartesian or hexagonal assembly of cells. When a sub-geometry is located inside a geometry but outside the calculation region it must be declared *virtual* (for example, the corners of a nuclear reactor).

HCELL — array of sub-geometry character*12 names to fill the Cartesian or hexagonal assembly of cells. 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 same order as for real and virtual mixtures.

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 9 while for hexagonal geometries the orientations available are illustrated in figure Figure 10. 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 sub-geometry will be inserted in the geometry (see Figure 11).

NAMPIN array of sub-geometry `character*12` names representing pins. These sub-geometries must be of type TUBE, TUBEX, TUBEY or TUBEZ.

3.3.6 Non standard geometries

The structure **(descNSG)** provides the possibility to define non standard geometries such as double-heterogeneity and *do-it-yourself* assemblies (see Table 24).

Table 24: Structure **(descNSG)**

[BIHET { TUBE | SPHE } *nmistr nmilg*
(*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*)]

The parameters use in Table 24 have the following meaning:

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.[54, 55]

TUBE keyword to specify that the micro structures are cylinders;

SPHE keyword to specify that the micro structures are spheres.

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, the initial radius must be $rs(1,i) = 0.0$.

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.

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.

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.[10] 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. The constraint

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

where $S(i)$ and $S(j)$ are the exterior surface areas of the cells of type i and j respectively, must be satisfied.

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

3.4 The tracking modules

The tracking modules perform an analysis of the geometry, including region volume and surface area calculations, and generate the integration lines for a geometry that was previously defined in the GEO: module. These operations are carried out differently depending on the tracking algorithm considered.

Five different tracking modules are available in DRAGON:

1. The JPMT: module which is used to perform an interface current tracking inside homogeneous region.[56–60];
2. The SYBILT: module which is used for interface current tracking inside heterogeneous blocks;
3. The EXCELT: module performs the tracking over the complete geometry with isotropic[12, 13] or specular[2, 17–19] surface current;
4. The NXT: module is a generalization of EXCELT: to more complex geometry, including assemblies of clusters in 2-D and 3-D. It also performs the tracking over the complete geometry;
5. The BIVACT: module is used to perform a 2-D diffusion like tracking that may be required for homogenization purposes.[61]

The first four tracking modules can be used to generate the information required for a solution to the transport equation. The last module (BIVACT:) can only be used in the EDI: module since it is not compatible with the flux solution FLU: and self-shielding SHI: modules. The general information resulting from these tracking modules is stored in a TRACKING data structure. For the JPMT:, EXCELT: and NXT: modules, an additional sequential binary tracking file may be generated.

None of these modules can analyze all the geometries that can be defined with the GEO: module. In general, some restrictions apply to each tracking option as a function of the approximations associated with the specific transport solution method. For instance

- Geometries that can be analyzed by the module EXCELT:
  a) 2-D geometries (CAR2D and HEX) that contain sub-geometries (CARCEL and HEXCEL respectively).
  b) 2-D cluster geometries corresponding to a TUBE sub-geometry superimposed on a global TUBE, CARCEL or HEXCEL geometry (here the main 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 internal annular regions).
  c) 3-D assemblies (CAR3D or HEXZ) that respectively contain Cartesian/annular (CARCELX, CARCELY and CARCELZ) and hexagonal/annular sub-geometries.
- Geometries that can be analyzed by the module NXT:
  a) Cartesian 2-D and 3-D geometries that can be processed by the EXCELT: module described above.
  b) A generalization of the Cartesian 2-D and 3-D geometries that can be processed by EXCELT: that includes pin clusters.
  b) A generalization of the hexagonal 2-D and 3-D assemblies that can be processed by EXCELT: and contain hexagons with a triangular mesh (HEXT or HEXTZ) and can include pin clusters.
- Geometries that can be analyzed by the module SYBILT:
  a) Homogeneous geometry HOMOGE.
  b) 1-D geometries SPHERE, TUBE and CAR1D.
  c) 2-D geometries CAR2D and HEX including CARCEL and HEXCEL sub-geometries as well as VIRTUAL sub-geometries;.
  d) 2-D non standard geometries containing micro structures.
  e) Double heterogeneity option.
- Geometries that can be analyzed by the module JPMT:
  a) 1-D geometries SPHERE, TUBE and CAR1D.
  b) 2-D geometries CAR2D and HEX including CARCEL and HEXCEL sub-geometries as well as VIRTUAL sub-geometries.

c) 2-D cluster geometries corresponding to a TUBE sub-geometry superimposed on a global TUBE, CARCEL or HEXCEL geometry (each cluster must be located between two independent annular regions and the clusters must not overlap).

d) 3-D TUBEZ geometries.

e) Double heterogeneity option.

- Geometries that can be analyzed by the module BIVACT:

a) 2-D Cartesian (CAR2D) geometries.

b) 2-D hexagonal (HEX) geometries.

The global numbering of the zones for a specific geometry proceeds following an order that is generally dependent on the specific tracking module selected. For more information on the region ordering selected by the EXCELT: module one can consult reference [5] while for the NXT: module the information is provided in reference [14].

The calling specifications for each of these modules are provided in Tables 25 to 29.

Table 25: Structure **(EXCELT:)**

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

Table 26: Structure **(NXT:)**

{ *TRKNAM* := NXT: *GEONAM* : : **(desctrack) (descnxt)** |
TRKFIL TRKNAM := NXT: *TRKNAM* : : **(desctrack) (descnxt)** |
TRKFIL TRKNAM := NXT: *GEONAM* : : **(desctrack) (descnxt)** }

The first form for the structure NXT: is used to analyze the geometry and track but will not generate a tracking file (see Table 26). The second form can be used to generate a tracking file from a compatible TRACKING data structure. Finally the last form generates both the tracking file and the TRACKING data structure from the GEOMETRY data structure. Thus, even if *TRKFIL* is not provided the tracking of the geometry may still take place and will be validated. The track normalization factors required to ensure volume preservation will also be computed and stored on *TRKNAM*. This information is required by the ASM: module for collision probability integration with online track generation (see Section 3.6).

Table 27: Structure **(JPMT:)**

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

Table 28: Structure **(SYBILT:)**

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

Table 29: Structure **(BIVACT:)**

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

In Tables 25 to 29 we have used:

TRKNAM `character*12` name of the TRACKING data structure that will contain region volume and surface area information in addition to region identification pointers and other tracking information.

TRKFIL `character*12` name of the sequential binary tracking file used to store the tracks lengths.[6] This file is always required for the `EXCELT:` module. It is also required if the `JPMT:` module is applied to a cluster type geometry. It is optional for the `NXT:` module.

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

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

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

(descnxt) structure describing the tracking data specific to `NXT:` (see Section 3.4.3).

(descsybil) structure describing the tracking data specific to `SYBILT:` (see Section 3.4.4).

(descjpm) structure describing the tracking data specific to `JPMT:` (see Section 3.4.5).

(descbivac) structure describing the tracking data specific to `BIVACT:` (see Section 3.4.6).

3.4.1 The general tracking data

This data structure is described in Table 30.

Table 30: Structure **(desctrack)**

[`EDIT` *iprint*] [`TITL` *TITLE*] [`ANIS` *nanis*] [{ `RENO` | `NORE` | `RENM` | `REND` }]

Here

`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 each tracking module vary substantially depending on the print level specified, the geometry analyzed and the tracking options selected. For example, with the `NXT:` module, one generally has

- if *iprint*=0, no output is produced.
- if *iprint*$\leq$1, a minimum amount of output is produced (main geometry properties, maximum and average errors on regional volume and surface area resulting from the tracking).
- if *iprint*$\leq$2, local errors on regional volume and surface area resulting from the tracking are also produced.
- if *iprint*$\geq$1000, explicit follow up of the tracking process as it progresses through the code.

`TITL` keyword to specify the title for this tracking file.

TITLE the `character*72` title associated with this tracking file. By default, *TITLE* is a series of 72 blank characters.

`ANIS` keyword to specify the order of anisotropy in collision probability or for the method of characteristics.

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

`RENO` keyword to specify the use of the automatic procedure for integration lines normalization to the fine mesh volumes. This normalization procedure ensures neutron balance for each fine mesh zone. It is the default option for transport based tracking modules (not valid for the `BIVACT:` tracking module).

`NORE` keyword to specify that automatic normalization of the integration lines should be deactivated.

`RENM` keyword to specify that the automatic normalization procedure for integration lines will be applied to the merged volumes. This normalization procedure ensures neutron balance for each merged zone. This option is only valid when the `EXCELT:` module is called.

`REND` keyword to specify that the automatic normalization procedure for integration lines will be applied to the merged volumes for each tracking direction. This normalization procedure ensures neutron balance on the fine mesh for each direction. This option is only valid when the `NXT:` module is called.

3.4.2 The `EXCELT:` specific tracking data

Table 31: Structure **(descexcel)**

[`MAXR` *maxreg*]

continued on next page

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

```
[ { PISO | PSPC } ]
[ CUT pcut ]
[ { GAUS | LCMD | CACA | CACB } npol ]
[ TRAK [ CORN pcorn ]
       { TISO [ { EQW | GAUS } ] nangl dens [ densz ] | TSPC [ MEDI ] nangl dens }
       [ SYMM isymm ] ]
```

where

MAXR	keyword to define the maximum number of regions that will be generated for this geometry.
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.
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 (this is the default option in ASM:).
PSPC	keyword to specify that a collision probability calculation with mirror like reflection or periodic boundary conditions is 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 exponential cutoff parameter for cyclic collision probability or characteristic integration (specular tracking only).
pcut	real value cutoff 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, the tracks are extended to infinity and *pcut*=0.0. If this option is used in an entirely reflected case, it is recommended to use the NORM command in the ASM: module.
GAUS	keyword to specify that a gaussian quadrature for the polar integration is to be used. Used by the method of characteristics with cyclic tracking.
LCMD	keyword to specify that optimized polar integration angles are to be selected.[62] This is the default option. Used by the method of characteristics with cyclic tracking.
CACA	keyword to specify that CACTUS type equal weight polar integration angles are to be selected.[63] Used by the method of characteristics with cyclic tracking.
CACB	keyword to specify that CACTUS type uniformly distributed polar integration angles are to be selected.[63] Used by the method of characteristics with cyclic tracking.
npol	the polar quadrature order. Used by the method of characteristics with cyclic tracking.
TRAK	keyword to specify the tracking parameters to be used.
CORN	keyword to specify the corner cutoff.

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

`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.[64]

`GAUS` keyword to specify the use of the Gauss-Legendre 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 for 2-D hexagonal or annular geometries and for general 3-D geometries.

`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 is the case when the external region of a `CARCEL` geometry is voided.

nangl angular quadrature parameter where

- For 3-D geometry, only the `EQW` option is permitted with *nangl*=2, 4, 8, 10, 12, 14 or 16.[6]
- For 2-D isotropic tracking, any value of *nangl* may be used and equidistant angles will be selected.
- For 2-D specular tracking 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.

dens real value representing the approximate density of the integration lines (in cm^{-1} for 2-D Cartesian and 3-D hexagonal geometries and cm^{-2} for 3-D Cartesian geometries). The choice of density along the plan perpendicular to each track direction 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 cluster type geometries 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.

densz real value representing the density of the integration lines (in cm^{-1} for 3-D hexagonal geometries).

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

isymm integer value describing the rotational symmetry of the cluster geometry (invariant under 2π/*isymm* rotation). The fixed default of this parameter is 1.

3.4.3 The `NXT:` specific tracking data

Table 32: Structure **(descnxt)**

```
[ { PISO | PSPC } ]
[ CUT pcut ]
[ CORN pcorn ]
[ { SYMM isymm | NOSY ]
[ {TISO | TSPC }
    [ { EQW | MEDI | PNTN | SMS } ] nangl
    [ { GAUS | CACA | CACB | LCMD | TRAD | TRAA } npol ]
    dens [ densl2 ] ]
[ NOTR ]
[ NBSLIN nbslin ]
[ LONG ]
```

where

PISO — see Section 3.4.2.

PSPC — see Section 3.4.2.

CUT — see Section 3.4.2.

pcut — see Section 3.4.2.

CORN — see Section 3.4.2.

pcorn — see Section 3.4.2.

SYMM — keyword to specify the level to which the tracking will respect the symmetry of the geometry.

isymm — level to which the tracking will respect the symmetry of the geometry. For 2-D and 3-D Cartesian geometries it must takes the form *isymm*=$2S_x + 4S_y + 16S_z$ where

- $S_x = 1$ if the X symmetry is to be considered and $S_x = 0$ otherwise.
- $S_y = 1$ if the Y symmetry is to be considered and $S_y = 0$ otherwise.
- $S_z = 1$ if the Z symmetry is to be considered and $S_z = 0$ otherwise.

NOSY — keyword to specify the full tracking will take place irrespective of the symmetry of the geometry. This is equivalent to specifying *isymm*=0.

TISO — see Section 3.4.2.

TSPC — see Section 3.4.2.

EQW — keyword to specify the use of equal weight quadrature (azimuthal integration in 2-D using a trapezoidal quadrature or EQ_N directional quadrature in 3-D[64]).

MEDI — see Section 3.4.2.

PNTN — keyword to specify that Legendre-Techbychev quadrature will be selected (only valid for 3-D geometries).[65]

SMS — keyword to specify that Legendre-trapezoidal quadrature will be selected (only valid for 3-D geometries).[66]

nangl azimuthal or directional (3D) quadrature order. Depending on the geometry and the tracking options *nangl* will take different meaning:

- For Cartesian and hexagonal 2-D geometries with isotropic tracking (only option permitted is `EQW` which is the default value), the azimuthal angle φ is discretized using a double *nangl* trapezoidal quadrature ($0 \leq \varphi \leq \pi/2$ and $\pi/2 \leq \varphi \leq \pi$).
- For Cartesian 2-D geometries with cyclic (specular) tracking (only `MEDI` option permitted, the default being to use end of range azimuthal position), the azimuthal angle φ is discretized using a specialized numerical quadrature. In this case *nangl* must be of the form $p + 1$ where p is a prime number (for example p=7,11, etc.); only the *nangl* = 8, 12, 14, 18, 20, 24, or 30 are allowed. By default the polar angle quadrature is a Gauss-Legendre quadrature of order *nangl*/2+1 (it can also be specified independently).
- For Cartesian and hexagonal 3-D geometries with the `EQW` option, Carlson equal weight directional quadrature are selected.[64] The only values of *nangl* permitted are then 2, 4, 8, 10, 12, 14 or 16 corresponding to 1, 3, 10, 21, 28 and 36 directions per quadrant in the upper half sphere.
- For Cartesian and hexagonal 3-D geometries with the `PNTN` or `SMS` options, *nangl* must be an even number smaller than 46.[14]

`GAUS` see Section 3.4.2.

`CACA` see Section 3.4.2.

`CACB` see Section 3.4.2.

`LCMD` see Section 3.4.2.

`TRAD` keyword to specify that the polar integration will be carried out using an order *npol* trapezoidal quadrature over $\mu = \cos\vartheta$ ($0 \leq \mu \leq 1$).

`TRAA` keyword to specify that the polar integration will be carried out using an order *npol* trapezoidal quadrature over ϑ ($0 \leq \vartheta \leq \pi/2$).

npol the polar quadrature order.

dens real value representing the density of the integration lines (in cm^{-1} for 2-D cases and cm^{-2} for 3-D cases 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 `NXT:`.

`NOTR` keyword to specify that the geometry will not be tracked. This is useful to verify if the geometry is adequate (can be processed by the module) before the tracking process as such is undertaken. For 2-D geometries, the tracking data structure generated can be used directly by the `PSP:` module (see Section 3.15) to illustrate the geometry.

`NBSLIN` keyword to change the default value for the maximum number of segments per lines.

nbslin maximum number of segments per lines. By default *nbslin*=100000.

`LONG` keyword to specify that a "long" tracking file will be generated. This option is required if the tracking file is to be used by the `TLM:` module (see Section 3.19).

3.4.4 The `SYBILT:` specific tracking data

Table 33: Structure **(descsybil)**

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

where

MAXR	keyword to specify the maximum number of flux regions for this geometry.
maxreg	maximum number of flux regions for this geometry. 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.
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$\times$*maxreg*).
MAXZ	keyword to specify the maximum amount of memory required to store the tracking lines.
maxint	the maximum amount of memory required to store the tracking lines. The default value is *maxint*=10000.
HALT	keyword to stop the execution at the end of the geometry analysis. 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 1-D integration parameters.
iqua1	number of basis points for the angular integration of the blocks in a 1-D 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 2-D integration parameters.
iqua2	number of basis points for the angular integration of a 2-D cell inserted in an assembly. 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 a 2-D cell inserted in an assembly. The default value is *nsegment*=3.
EQW	keyword to specify the use of an equal weight quadrature method.
GAUS	keyword to specify the use of the Gauss-Legendre or the Gauss-Jacobi quadrature method. This is the default option.

`ROTH` keyword to specify that the isotropic (DP_0) components of the current at cell interfaces 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 for 2-D assemblies of cells.

`ROT+` keyword to specify that the isotropic (DP_0) components of the current at cell interfaces is used. The global collision matrix is calculated in a annular model. Only used for 2-D assemblies of cells.

`DP00` keyword to specify that the isotropic (DP_0) components of the current at cell interfaces is used. The global collision matrix is computed explicitly. Only used for 2-D assemblies of cells.

`DP01` keyword to specify that the linearly anisotropic (DP_1) components of the current at cell interfaces 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 conditions are used. Only used for 2-D assemblies of cells.

`ASKE` keyword to specify the use of an *Askew* cylinderization of the cells that preserves both the external surface and the material balance of the external crown (by a modification of its concentration). By default a *Wigner* cylinderization of the cell is used that preserves the volume of the external crown. Note, that an assembly containing a number of rectangular cells having unequal volumes requires Askew cylinderization. This option can only be used if the `ROTH` or `ROT+` options are also activated. Only used for 2-D assemblies of cells.

`LIGN` keyword to specify that all the integration lines are to be printed. This option should be used with care because it can generate a rather large amount of output. Only used for 2-D assemblies of cells.

`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.

3.4.5 The `JPMT:` specific tracking data

Table 34: Structure **(descjpm)**

[`MAXR` *maxreg*]
[`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

MAXR	see Section 3.4.4.
maxreg	see Section 3.4.4.
MAXJ	see Section 3.4.4.
maxcur	see Section 3.4.4.
MAXZ	see Section 3.4.4. This option is not used for cluster geometries.
maxint	see Section 3.4.4.
OLD	keyword to specify that a set of previously calculated collision probabilities 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	see Section 3.4.4. This option is not used for cluster 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	see Section 3.4.4.
iqua1	see Section 3.4.4.
QUA2	see Section 3.4.4.
iqua2	see Section 3.4.4.
nsegment	see Section 3.4.4.
EQW	see Section 3.4.4.
GAUS	see Section 3.4.4.
ROTH	see Section 3.4.4.
ROT+	see Section 3.4.4.
DP00	see Section 3.4.4.
DP01	see Section 3.4.4.
ASKE	see Section 3.4.4.
LIGN	see Section 3.4.4.
RECT	see Section 3.4.4.

`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` see Section 3.4.4.

iquab see Section 3.4.4.

3.4.6 The `BIVACT:` specific tracking data

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

Table 35: Structure **(descbivac)**

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

where

`MAXR` ksee Section 3.4.4.

maxreg see Section 3.4.4.

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

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

ielem order of the finite element representation. The values allowed for Cartesian geometries are

ielem=1 for linear polynomials,
ielem=2 for parabolic polynomials,
ielem=3 for cubic polynomials, and
ielem=4 for quartic polynomials.

By default *ielem*=1. For hexagonal geometries, only *ielem*=1 is allowed.

icol type of quadrature used to integrate the mass matrices. The values allowed are

icol=1 for analytical integration,

icol=2 for Gauss-Lobatto quadrature, and

icol=3 for 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 valid only for 2-D hexagonal geometries. The values allowed are

isplh=1 for full hexagon,

isplh=2 for for splitting each hexagon into 6 triangles,

isplh=3 for splitting each hexagon into 24 triangles,

isplh=5 for splitting each hexagon into 96 triangles,

isplh=9 for splitting each hexagon into 384 triangles, and

isplh=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 (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 (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 (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[67], is used to compute 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. Using this information, it then recomputes the self-shielded cross section for these resonant isotopes (saved on the MICROLIB) as well as updates the embedded MACROLIB. The general input format for this module is presented in Table 36.

Table 36: Structure **(SHI:)**

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

Here

MICLIB `character*12` name of the MICROLIB that will contain the microscopic and macroscopic cross sections updated by the self-shielding module.

OLDLIB `character*12` name of a read-only MICROLIB that is copied into *MICLIB*. The library *OLDLIB* is first copied to *MICLIB* before this library is updated.

TRKNAM `character*12` name of the TRACKING data structure associated with the GEOMETRY considered for self-shielding calculations.

TRKFIL `character*12` name of the sequential binary tracking associated with the *TRKNAM* TRACKING data structure. This file is required and only if it was produced by the tracking module that generated *TRKNAM* (see Section 3.4).

(descshi) structure describing the self-shielding options.

Each time the `SHI:` module is called, the MICROLIB data structure in such a way that the information provided in the **(descshi)** input structure is saved. The next time this module is called, these values will be extracted from the MICROLIB and used as floating defaults.

3.5.1 Data input for module SHI:

Table 37: Structure **(descshi)**

[`EDIT` *iprint*]
[`GRMIN` *lgrmin*] [`GRMAX` *lgrmax*]
[`MXIT` *imxit*] [`EPS` *valeps*]
[{ `LJ` | `NOLJ` }] [{ `GC` | `NOGC` }] [{ `NOTR` }]
[`LEVE` *level*] [`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 provided explicitly in the library.

`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 closest to 4.0 eV for all the libraries except for those in the `WIMSAECL` and `WIMSD4` format where this information is provided explicitly.

`MXIT` keyword to specify the maximum number of iterations used in 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*=1.0×10^{-4}.

`LJ` keyword to activate the Livolant-Jeanpierre normalization scheme that modifies the self-shielded averaged neutron flux in heterogeneous geometries. By default the Livolant-Jeanpierre normalization scheme is not activated.

`NOLJ` keyword to deactivate the Livolant-Jeanpierre normalization scheme. This is the default option.

`GC` keyword to activate the Goldstein-Cohen approximation in cases where Goldstein-Cohen parameters are stored on the microscopic cross section library. These parameters are not available if the resonant isotopes are interpolated from a `MATXS` type library. This is the default option.

`NOGC` keyword to deactivate the Goldstein-Cohen approximation even if Goldstein-Cohen parameters are stored on the microscopic cross section library.

`NOTR` keyword to deactivate the transport correction option for self-shielding calculations (see `CTRA` in Sections 3.1 and 3.2).

`LEVE` keyword to specified the self-shielding correction option. When this option is not specified the option *level*=0 is selected.

level the self-shielding correction option. The following values are allowed

level=1 the original Stamm'ler model model is used;

level=1 the original Stamm'ler model with Nordheim approximation is used;

level=2 the Stamm'ler model with Nordheim approximation and Riemann integration is used.[68]

`PIJ` keyword to specify the use of complete collision probabilities for the `JPMT:` tracking option. By default, a fast reconstruction algorithm based on sparse matrix algebra is used for this tracking option.

3.6 The assembly modules

DRAGON contains two assembly modules, ASM: and EXCELL:, that are used to prepare the group dependent complete collision probability or the assembly matrices required by the flux solution module. The ASM: module first recovers tracking lengths and material numbers from the sequential tracking file and then computes the collision probability matrices associated with the problem. The EXCELL: module, which is programmed to enhance the capability and performance of collision probability calculations, can also be used to perform the work of both the EXCELT: and the ASM: modules for computing collision probabilities in 3–D geometries. As a result, EXCELL: module does not save the tracking lines to a file, but generates them when required by the CP integration process. A similar feature has been programmed in the ASM: module for NXT: based tracking. Thus, calling ASM: with a NXT: TRACKING data structure but without the associated tracking file, will replace all the access to the tracking file with call to the NXT: tracking subroutines that will generate online the integration lines required.

The input specifications for these modules are presented in Tables 38 and 39.

Table 38: Structure **(ASM:)**

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

Table 39: Structure **(EXCELL:)**

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

Here

PIJNAM character*12 name of ASMPIJ data structure containing the system matrices.

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 (see Section 3.4).

TRKFIL character*12 name of the sequential binary tracking file associated with the *TRKNAM* TRACKING data structure. This file is required if it is produced by the tracking module (see Section 3.4), the only exception being when a NXT: based TRACKING data structure is available.

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 3.4.2).

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

3.6.1 Data input for module `ASM:`

Table 40: Structure **(descasm)**

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

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

iprint — index used to control the printing of this module. The amount of output produced 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. Only the variational acceleration technique in module FLU: uses this information (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. 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:.

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 up-scattering, 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 sated on *PIJNAM*.

NMPIJ `character*12` 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 that arise due to the numerical evaluation of the probabilities that may result in non-conservative collision probability matrices. The default option normalization is `HELI`..

`NONE` keyword to specify that the probability matrices are not to be normalized for neutron conservation.

`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.[69]

`HELI` keyword to specify that the HELIOS algorithm will be used to normalize the collision probability matrices.[70]

`NONL` keyword to specify that a non linear multiplicative algorithm will be used to normalize the collision probability matrices.[69]

`ALLG` keyword to specify that the contribution of a tracking line to the multigroup collision probabilities will be processed before the next tracking line is analyzed. This means that for a multigroup problem the tracking file is read once. The default option is to generate the collision probability matrices group by group implying multiple readings of the tracking file. The major drawback of using the `ALLG` keyword is that the space requirement for the problem is $N \times N \times G$ for a N region G groups problem while only a $N \times N$ array is required when this option is not activated.

3.6.2 Data input for module `EXCELL`:

Table 41: Structure **(descXL)**

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

`EDIT` see Section 3.6.1.

iprint see Section 3.6.1.

`NORM` see Section 3.6.1.

`ALBS` see Section 3.6.1.

`SKIP` see Section 3.6.1.

`PNOR` see Section 3.6.1.

`NONE` see Section 3.6.1.

`DIAG` see Section 3.6.1.

`GELB` see Section 3.6.1.

`HELI` see Section 3.6.1.

`NONL` see Section 3.6.1.

`TRAK` see Section 3.4.2.

`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 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.

`TISO` see Section 3.4.2.

nangl see Section 3.4.2.

dens see Section 3.4.2.

`CORN` see Section 3.4.2.

pcorn see Section 3.4.2.

`SYMM` see Section 3.4.2.

isymm see Section 3.4.2.

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 input specifications for this module are presented in Table 42.

Table 42: Structure **(FLU:)**

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

Here

FLUNAM `character*12` name of the FLUXUNK data structure containing the solution. If *FLUNAM* appears on the RHS, the solution previously stored in *FLUNAM* can be used to initialize the iterative solution process.

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 (see Section 3.4).

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

3.7.1 Data input for module `FLU:`

Table 43: 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*]
[EGPA *epsgpa*] [CGPA *congpa*] [DECO { ON | OFF }]

Here

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

iprint index used to control the printing of this module. The amount of output produced 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.

PAF keyword to specify that a pseudo-adjoint flux solution is to be considered.[40]

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.

TYPE keyword to specify the type of flux or adjoint calculations 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 3.3.3). 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 an eigenvalue problem for a non multiplicative medium is to be treated. The eigenvalue in this case is the critical buckling. The buckling eigenvalue has meaning only in the case of a cell without leakages (see the structure **(descBC)** in Section 3.3.3). 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).

EGPA — keyword to specify that the control parameters for the generalized adjoint flux iteration are to be modified.

epsgps — relative convergence criterion for the adjoint flux. The explicit convergence criteria for the generalized adjoint will be given by *epsgps*×*epsunk*. By default, *epsgps*=10.

CGPA	keyword to specify the contamination factor for the generalized adjoint.
congpa	contamination factor for the adjoint flux. By default, *congpa*=100.
DECO	keyword to activate or deactivate the decontamination option.
OFF	keyword used to deactivate the decontamination option.
ON	keyword used to activate the decontamination option.

3.7.2 Leakage model specification structure

The **(descleak)** structure is described in Table 44.

Table 44: 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 the data structure *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 performed (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 matrices are multiplied by the adequate non leakage homogeneous buckling dependent factors.[71]. This is the default option when a leakage (B) or a fission source eigenvalue problem (K) calculation is performed with imposed buckling.
ALBS	keyword used to specify that an homogeneous buckling contribution is introduced by a group dependent correction of the albedo.[72] It is then necessary to define the geometry with an external boundary condition of type VOID (see Section 3.3.3) 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[73] or B_n[74,75] 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 assumes that the directional bucklings are all identical (floating default option).

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

Z keyword used to specify that a Z direction buckling search will be considered with imposed buckling in the X and Y directions.

X keyword used to specify that a X direction buckling search will be considered with imposed buckling in the Y and Z directions. with imposed Y and Z direction bucklings.

Y keyword used to specify that a Y direction buckling search will be considered with imposed buckling in the X and Z directions.

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 bucklings in the X, Y and Z directions are to be initialized to *valb2*/3 (floating default).

R keyword used to specify that the buckling in the X and Y directions 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

$$valbr2 = valbx2 + valby2$$

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 X 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 Y 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.

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

3.8 The `MOCC:` and `MCU:` modules

The `MOCC:` and `MCU:` modules can be used respectively to solve the transport equation using the method of cyclic characteristics.

In the case of the `MOCC:` module, specular boundary conditions in 2-D geometries are considered (cyclic tracking required).[22,23] The `MCU:` module can be used for 3-D geometries with isotropic boundary conditions.[24–26] The calling specifications are presented in Tables 45 and 46.

Table 45: Structure **(MOCC:)**

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

Table 46: Structure **(MCU:)**

{ *FLUNAM TRKNAM* `:=` `MCU:` [*FLUNAM*] *GEONAM LIBNAM TRKFIL* `::` **(descmcu) (desctrak)**
| *FLUNAM* `:=` `MCU:` [*FLUNAM*] *TRKNAM TRKFIL LIBNAM* `::` **(descmcu)** }

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 iterative process.

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 associated with the *TRKNAM* TRACKING data structure. This file is required if it is produced by the tracking module (see Section 3.4), the only exception being when a `NXT:` based TRACKING data structure is available.

GEONAM `character*12` name of GEOMETRY data structure that contains a physical description of the problem to be solved.

(descmoc) structure containing the input data for the `MOCC:` module (see Section 3.8.1).

(descmcu) structure containing the input data for the `MCU:` module (see Section 3.8.2).

(descexcel) structure containing the input data for tracking the geometry (see Section 3.4.2).

3.8.1 Data input for module MOCC:

Table 47: Structure **(descmoc)**

[EDIT *iprint*]
[EXAC]
[NBPN *nl*]
TYPE { N | S | K [**(descleak)**] | B [**(descleak)**] }
[DFLX { ON | OFF }]
[AJCB { ON | OFF }]
[ADJ { SAJ0 | SAJ1 }]
[ANGL]
[THER [*maxthr*] [*epsthr*]]
[EXTE [*maxout*] [*epsout*]]
[UNKT [*epsunk*]]
[NOBA]
[ACCE *nlibre naccel*]

EDIT	keyword used to modify the print level *iprint*.
iprint	index used to control the printing of this module. The amount of output produced 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.
TYPE	see Section 3.7.1.
N	see Section 3.7.1.
S	see Section 3.7.1.
K	see Section 3.7.1.
B	see Section 3.7.1.
(descleak)	see Section 3.7.2.
DFLX	to activate or deactivate the flux calculator. By default DFLX is ON.
AJCB	to activate or deactivate the adjoint calculator. By default AJCB is OFF.
ON	to turn the DFLX or AJCB option ON.
OFF	to turn the DFLX or AJCB option OFF.

ADJ	to select the adjoint calculation option.
SAJ0	similar to TYPE K (eigenvalue calculation without external fixed source).
SAJ1	similar to TYPE S (external fixed source are taken into account).
ANGL	to save the angular fluxes and adjoint on the FLUXUNK data structure.
THER	see Section 3.7.1.
maxthr	see Section 3.7.1.
epsthr	see Section 3.7.1.
EXTE	see Section 3.7.1.
maxout	see Section 3.7.1.
epsout	see Section 3.7.1.
UNKT	see Section 3.7.1.
epsunk	see Section 3.7.1.
NOBA	keyword used to specify that the flux rebalancing option is to be turned off in the thermal iteration. By default the flux rebalancing option is activated.
ACCE	see Section 3.7.1.
nlibre	see Section 3.7.1.
naccel	see Section 3.7.1.

3.8.2 Data input for module MCU:

Table 48: Structure **(descmcu)**

[EDIT *iprint*]
TYPE { N | S | K [**(descleak)**] | B [**(descleak)**] }
[THER [*maxthr*] [*epsthr*]]
[EXTE [*maxout*] [*epsout*]]
[NOBA]
[ACCE *nlibre naccel*] [SCR *maxscr*] [ETAB *ON OFF*] [ITLM]

EDIT	keyword used to modify the print level *iprint*.
iprint	index used to control the printing of this module. The amount of output produced will vary substantially depending on the print level specified.
TYPE	see Section 3.7.1.

N	see Section 3.7.1.
S	see Section 3.7.1.
K	see Section 3.7.1.
B	see Section 3.7.1.
(descleak)	see Section 3.7.2.
THER	see Section 3.7.1.
maxthr	see Section 3.7.1.
epsthr	see Section 3.7.1.
EXTE	see Section 3.7.1.
maxout	see Section 3.7.1.
epsout	see Section 3.7.1.
NOBA	see Section 3.8.1.
ACCE	see Section 3.7.1.
nlibre	see Section 3.7.1.
naccel	see Section 3.7.1.
SCR	keyword used to modify the number of iterations in the self-collision rebalancing procedure.
maxscr	the number of iterations in the self-collision rebalancing procedure. The default value is *maxscr*=5.
ETAB	keyword to specify the option for using exponential tables.
ON	to specify that the exponential tables will be used.
OFF	to specify that the exponential tables will not be used.
ITLM	keyword to specify that the effective number of thermal iterations m at outer iteration n is $m = \min(n, \textit{maxthr})$ where *maxthr* is the maximum number of thermal iterations.

3.9 The EDI: module

The EDI: module performs the main editing calculations in DRAGON. It can compute reaction rates, average and condensed cross sections and fluxes. It can also store both macroscopic and microscopic cross sections respectively in the form of a MACROLIB or a MACROLIB on an EDITION data structure for further use. The input specifications for this module are presented in Table 49.

Table 49: Structure **(EDI:)**

EDINAM := EDI: [*EDINAM*] *FLUNAM LIBNAM TRKNAM* [*REFGEO REFPIJ* [{ *SPHGEO* | *SPHTRK SPHLINE* }]] : : **(descedi)**

EDINAM `character*12` name of the EDITION data structure where the editing 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 or microscopic 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).

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

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

SPHGEO `character*12` name of the equivalence GEOMETRY data structure to be used for SPH factors calculation. By default, the original transport GEOMETRY is used. In some cases, the module `EDI:` can also automatically build a equivalence GEOMETRY based on the original geometry. However, it is always preferable to specify explicitly the equivalence GEOMETRY or TRACKING data structure to be considered in such calculations.

SPHTRK `character*12` name of the equivalence TRACKING data structure to be used for SPH factors calculation. By default, the original transport TRACKING is used. It is always preferable to specify explicitly the equivalence GEOMETRY or TRACKING data structure to be considered in such calculations.

SPHLINE `character*12` name of the tracking file associated with the *SPHTRK* data structure.

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

3.9.1 Data input for module `EDI:`

Table 50: Structure **(descedi)**

[`EDIT` *iprint*]
[`P1SCAT` { `FLUX` | `CURRENT` | `COHERENT` | `DIRECTION` }]
[`UPS`]
[`MERG` { `COMP` | `GEO` | `HMIX` | `CELL` |
`REGI` (*ireg*(i), $i = 1, N_r$) |
`MIX` (*imixt*(i), $i = 1, N_m$) |
`NONE` }]
[`TAKE` { `REGI` (*ireg*(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` [`ISOT`] { `ALL` | `NONE` | *nis* (*HISO*(i), i=1,*nis*)]
[`FLIB` [`ISOT`] { `ALL` | `NONE` }]
[`ACTI` [`ISOT`] { `NONE` | (*imixa*(i), $i = 1, N_m$)] }

continued on next page

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

[SAVE [ON { *DIRN* | *idirn* }]]
[PERT]
[STAT { ALL | RATE | FLUX | DELS } [REFE { *DIRO* | *idiro* }]]
[NBAL]
[SPH **(descsph)**]

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 means no output;

iprint≥ 1 the average and integrated flux are printed (floating default);

iprint≥ 2 the reaction rates are printed;

iprint≥ 3 the condensed and homogenized vector cross sections are printed;

iprint≥ 4 the condensed and homogenized scattering cross sections are printed.

P1SCAT — keyword to specify the type of homogenization to be considered for the anisotropic component of the scattering cross section.

FLUX — keyword to specify the flux/volume homogenization for the anisotropic component of the scattering cross section. This is the default option used when no leakage model is considered.

CURRENT — keyword to specify the current/volume homogenization for the anisotropic component of the scattering cross section. This is the default option used when an homogeneous leakage model is considered.

COHERENT — keyword to specify a coherent directional averaged current/volume homogenization for the anisotropic component of the scattering cross section.[76]

DIRECTION — keyword to specify a coherent directional current/volume homogenization for the anisotropic component of the scattering cross section. This is the default option used when an heterogeneous leakage model is considered.[77]

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

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

MERG — keyword to specify that the neutron flux to be edited will be homogenized over a various number of regions.

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

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

GEO — keyword to specify that the homogenization region will be selected by comparing a calculation geometry with an homogenization geometry. Only a reduced number of EXCELT: geometries can now be processed using this option.

`HMIX` keyword to specify that the homogenization region will be selected using the information provided by the `HMIX` option in the `GEO:` module (see Section 3.3.5). In this case, all the regions associated with a virtual homogenization mixture will be homogenized. If the virtual homogenization mixtures were not defined in the geometry, the real mixtures are used instead (see `MIX` keyword in Section 3.3.5). This option is valid only for `NXT:` based TRACKING data structure (this option uses the information stored on the reference TRKNAM data structure).

`CELL` keyword to specify that one homogenization region will be associated with each cell in an assembly geometry. Only geometries analyzed using the `NXT:` tracking module can be processed using this option (this option uses the information stored on the reference TRKNAM data structure). In the case where the geometry is not built with the `CELL` option (see Section 3.3.5), this option is identical to the `COMP` option as if the assembly was composed of a single cell. Homogenization region identification is printed on the output file on a 3-D Cartesian grid that takes into account the internal symmetries of the geometries.

`REGI` keyword to specify that the homogenization of the neutron flux will take place over specific regions. Here, $N_r \leq$*maxreg* with *maxreg* the maximum number of regions for which a transport solution was obtained.

`MIX` keyword to specify that the homogenization of the neutron flux will take place over specific physical mixtures. Here, $N_m \leq$*maxmix* with *maxmix* the maximum number of mixtures in the macroscopic cross section library.

ireg array of homogenized mixtures numbers to which are associated the regions used for the flux calculation (`MERG` option) or array of regions where the editing will take place (`TAKE` option). A value of *iregm*=0 means that the corresponding region is not considered in the homogenization process.

imixt array of homogenized mixture numbers to which are associated the physical mixtures (`MERG` option) or array of mixture numbers where the homogenization will take place (`TAGE` option). A value of *imixm*=0 means that the corresponding physical mixture is not considered in the homogenization process. For library mixtures not used in the geometry, *imixm*=0 should be used.

`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.

`FLIB` keyword similar to `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 original MICROLIB are to be processed.

nis number of isotopes present in the original MICROLIB to be processed.

HISO array of `character*8` isotope 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 mixtures not used in the geometry can be considered here.

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

```
WRITE(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` or `MACR` options are activated, the information is saved in the form of a MICROLIB. Otherwise, a MACROLIB is created on a subdirectory of *EDINAM*.

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

DIRN `character*12` 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` defined as

```
WRITE(CDIRN,'(A8,I4)') 'REF-CASE',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 `character*12` 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` defined using:

```
WRITE(CDIRN,'(A8,I4)') 'REF-CASE',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 transport geometry and an homogenization geometry, 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.[71, 72, 78] The input specifications for this structure are presented in Table 51.

Table 51: Structure **(descsph)**

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

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 3.3.3) 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.

`MTRK` keyword to specify that the macro TRACKING *SPHTRK* and tracking file *SPHLINE* provided will be used for homogenization.

`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 SPH calculation.

`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.

`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 direct strategy. For a given macro group, the SPH factor will be equal to the ratio of the average flux in the region to 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 prodices SPH factors that are equal to the inverse of the discontinuity factors.

`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 3.3.3) external boundary conditions to be closed using `ALBS` in modules `ASM:` and `FLU:` (see Sections 3.6.1 and 3.7.1).[72]

`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.

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

(descnxt) structure of the `NXT:` tracking options (see Section 3.4.3).

(descsybil) structure of the `SYBILT:` tracking options (see Section 3.4.4).

(descjpm) structure of the `JPMT:` tracking options (see Section 3.4.5).

(descbivac) structure of the `BIVACT:` tracking options (see Section 3.4.6).

3.10 The EVO: module

The EVO: module is used for in-core (i.e., burnup) or out-of-core isotopic depletion calculations. Both the isotopic densities and the macroscopic cross sections present in the MICROLIB are updated at the end of each calculations. The general format of the data which is used to control the execution of this module is presented in Table 52.

Table 52: 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.

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 microcopic cross section library *OLDMIC* is copied in *MICNAM* and then updated.

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

TRKNAM character*12 name of a read-only TRACKING 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).

3.10.1 Data input for module EVO:

Table 53: Structure **(evo)**

[EDIT *iprint*]
[{ SAVE *xts* { S | DAY | YEAR } [{ FLUX *flux* | POWR *power* | W/CC *wcc* }] | NOSA }]
[EPS1 *valeps1*] [EPS2 *valeps2*] [EXPM *valexp*] [H1 *valh1*]
[{ SATOFF | NSAT | SAT }] [{ NODI | DIRA }]
[{ TIXS | TDXS | NOEX }]
[{ GLOB | NOGL }]
[EXTR]
[{ RUNG | KAPS }]
[MIXP (*imixp*(i), $i = 1, N_m$)]
[MIXB (*imixb*(i), $i = 1, N_m$)]
[DEPL { *xti xtf* | *dxt* } { S | DAY | YEAR } [{ COOL | FLUX *flux* | POWR *power* | W/CC *wcc* }]]
[SET *xtr* { S | DAY | YEAR }]

`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*.

`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*. The isotopic concentrations corresponding to this specific time will be used to update the *MICNAM* data structure.

`MIXB` keyword to specify the mixtures that will be burned. By default, all the mixtures that can burn will.

imixb list of mixtures to burn. All the mixtures not specified in this are assumed to have isotopic contents constant in time. The maximum number of mixtures that can be provided is that specified in the MICROLIB data structure.

`MIXP` keyword to specify which mixtures will be used for the power normalization. By default, all the mixtures in the cell can contribute to the power production. When the the keyword `MIXB` is specified the list *imixp* is assumed to be identical to *imixb*. The irradiation stored in the BURNUP data structure will be that associated with the mixture specified by this keyword.

imixp list of mixtures mixtures considered for power normalisation. The maximum number of mixtures that can be provided is that specified in the MICROLIB data structure.

`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 in *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 `EVONAM` where this information is stored will be given by

```
WRITE(EVONAM,'(A8,I4)') 'DEPL-DAT',inn
```

where `inn` is an index associated with the time *xti*.

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.

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 burnup calculation at constant flux 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 a burnup calculation at constant power (KW/Kg) is to be performed.

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

W/CC keyword to specify that a burnup calculation at constant power (W/cm^3) is to be performed.

wcc power expressed in W/cm^3. In the case where *wcc*$\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 10^{-5}.

EPS2 keyword to specify the tolerance used in the search algorithm for a final fixed power (used if the POWR and W/CC options are activated).

valeps2 the tolerance used in the search algorithm for a final fixed power. The default value is 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 10^5 or use the keyword SATOFF.

SATOFF keyword to specify that the saturation model for the non-fissile isotopes will not be used.

NSAT keyword to specify that the saturation model for the non-fissile isotopes will not be used. This keyword has the same effect as keyword SATOFF.

SAT keyword to specify that the saturation model for the non-fissile isotopes will be used. This is the default option.

NODI keyword to specify that no Dirac delta function will be used in the saturation model. For isotope k at saturation one therefore uses

$$N_k(t) = \frac{S_k(t)}{\Lambda_k(t)}$$

`DIRA` keyword to specify that a Dirac delta function will be used in the saturation model. For isotope k at saturation one uses

$$N_k(t) = \frac{1}{\Lambda_k(t)} \left[a\delta(t - t_0) + S_k(t) + b\delta(t - t_f) \right]$$

This is the default option.

`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 will be optimized in the `EVO:` module to ensure that the solution to the depletion equations converges. The default value is 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.[79] This is the default value.

`TIXS` keyword that specifies that time independent cross sections will be used. A time dependent flux distribution will also be considered. This is the default option when no time dependent cross sections are provided.

`TDXS` keyword that specified that time dependent cross sections will be used if available. This is the default option when time dependent cross sections are provided.

`NOEX` keyword that specified that time independent cross sections and fluxed will be used.

`GLOB` keyword to specify that the global energy produced will be taken into account if available. This is the default option.

`NOGL` keyword to specify that only the energy produced in the fuel will be taken into account even if energy production outside the fuel is available.

`EXTR` keyword for power extrapolation when fixed power burnup is selected.

3.11 The `CPO:` module

The `CPO:` module is used to generate a reactor cross-section database in the `COMPO` format that can be used for full core calculation using DONJON.[45] The input specifications for this module are presented in Table 54.

Table 54: Structure **(CPO:)**

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

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.

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).

3.11.1 Data input for module `CPO:`

Table 55: Structure **(desccpo)**

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

`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 directories with full name `NAMDIR` created using

```
WRITE(NAMDIR,'(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 must be extracted before being stored on the reactor database. The microscopic cross sections and concentrations associated with these isotopes will 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

$$\rho(T,P) = \frac{100 \times \rho_{H_2O}(T)\rho_{D_2O}(T)}{P\rho_{H_2O}(T) + (1-P)\rho_{D_2O}(T)}$$

Temperature tabulations for $\rho_{H_2O}(T)$ and $\rho_{D_2O}(T)$ are the same as those of the WIMS-AECL code and are not valid for supercritical water conditions.[35] The input specifications for this module are presented in Table 56.

Table 56: Structure **(INFO:)**

```
INFO: :: (descinfo)
```

(descinfo) structure containing the input data to this module (see Section 3.12.1).

3.12.1 Data input for module `INFO:`

Table 57: Structure **(descinfo)**

```
[ EDIT iprint ]
[ LIB: { DRAGON | MATXS | MATXS2 | WIMSD4 | WIMS | WIMSAECL } FIL: NAMEFIL ]
[ TMP: temp { K | C } ]
[ PUR: purity { WGT% | ATM% } ]
[ CALC DENS WATER >>dens<< ]
[ ENR: enrichment { WGT% | ATM% } ]
```

continued on next page

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

```
[[ 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<< } } ]]
```

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 microscopic cross section library from which the isotopic mass ratio will be recovered.
DRAGON	keyword to specify that the microscopic cross section library is in the DRAGLIB format.
MATXS	keyword to specify that the microscopic cross section library is in the MATXS format of NJOY-II and NJOY-89.
MATXS2	keyword to specify that the microscopic cross section library is in the MATXS format of NJOY-91.
WIMSD4	keyword to specify that the microscopic cross section library is in the WIMS-D4 format.
WIMS	keyword to specify that the microscopic cross section library is in the WIMS-AECL format.
WIMSAECL	keyword to specify that the microscopic cross section library is in the WIMS-AECL format.
FIL:	keyword to specify the name of the file where the microscopic cross section library is stored.
NAMEFIL	character*8 name of the microscopic cross section library from which the mass ratio are recovered.
TMP:	keyword to specify the isotopic temperature.
temp	temperature given in Kelvin (K) or Celsius (C).
PUR:	keyword to specify the heavy water purity, that is fraction of heavy water in a mix of heavy and light water.
purity	heavy water purity in weight percent (WGT%) or atomic percent (ATM%).
ENR:	keyword to specify the fuel enrichment. For UO_2 fuel, this represent the ratio of ^{235}U (concentration or weight) to ^{235}U+^{238}U. For ThO_2 fuel, the ratio of ^{233}U (concentration or weight) to ^{233}U+^{232}Th is considered.
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.

`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.

dens calculated water density.

`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 a given isotope list, the mass is extracted from the library or a calculation process is expected. Once this calculation is has been performed, it is possible to list other isotopes and ask for further calculations. Finally note that the number of output parameters (denoted by >> *param*<<) 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

The `CFC:` module is used to generate a Feedback Model database required for a full core calculation in DONJON.[37–39] The input specifications for this module are presented in Table 58.

Table 58: Structure **(CFC:)**

CFCNAM `:=` `CFC:` [*CFCNAM*]
(*CPONAM*(i), i=1,28) `::` **(desccfc)**

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. These CPO respectively contain the following information

1. Reference data for cell averaged and two group burnup dependent cross sections.
2. Data for cell averaged and two group burnup dependent cross sections cross section at high fuel temperature perturbation.

3. Data for cell averaged and two group burnup dependent cross sections cross section at low fuel temperature perturbation.
4. Data for cell averaged and two group burnup dependent cross sections cross section at high coolant temperature perturbation.
5. Data for cell averaged and two group burnup dependent cross sections cross section at low coolant temperature perturbation.
6. Data for cell averaged and two group burnup dependent cross sections cross section at high moderator temperature perturbation.
7. Data for cell averaged and two group burnup dependent cross sections cross section at low moderator temperature perturbation.
8. Data for cell averaged and two group burnup dependent cross sections cross section at high coolant density perturbation.
9. Data for cell averaged and two group burnup dependent cross sections cross section at low coolant density perturbation.
10. Data for cell averaged and two group burnup dependent cross sections cross section at high moderator density perturbation.
11. Data for cell averaged and two group burnup dependent cross sections cross section at low moderator density perturbation.
12. Data for cell averaged and two group burnup dependent cross sections cross section with perturbed moderator boron concentration.
13. Data for cell averaged and two group burnup dependent cross sections cross section with perturbed moderator purity.
14. Data for cell averaged and two group burnup dependent cross sections cross section with perturbed xenon concentration.
15. Data for cell averaged and two group burnup dependent cross sections cross section with perturbed samarium concentration.
16. Data for cell averaged and two group burnup dependent cross sections cross section with perturbed neptunium concentration.
17. Data for cell averaged and two group burnup dependent cross sections cross section with perturbed fuel temperature and coolant density.
18. Data for cell averaged and two group burnup dependent cross sections cross section with perturbed coolant temperature and density.
19. Data for cell averaged and two group low power burnup dependent cross sections cross section.
20. Data for cell averaged and two group low intermediate burnup dependent cross sections cross section.
21. Data for cell averaged and two group high power burnup dependent cross sections cross section.
22. Reference data for two group burnup dependent moderator cross sections.
23. Data for two group burnup dependent moderator cross sections at high moderator temperature perturbation.
24. Data for two group burnup dependent moderator cross sections at low moderator temperature perturbation.
25. Data for two group burnup dependent moderator cross sections at high moderator density perturbation.

26. Data for two group burnup dependent moderator cross sections at low moderator density perturbation.
27. Data for two group burnup dependent moderator cross sections with perturbed moderator boron concentration.
28. Data for two group burnup dependent moderator cross sections with perturbed 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 59: Structure **(desccfc)**

[`EDIT` *iprint*]
[`INFOR` *TITLE*]
[`DNAME` *RNANE*]
[`PWR` *powerref powerhigh powerint powerlow*]
[`TCOOL` *tcoolref tcoolhigh tcoollow*]
[`TMODE` *tmoderef tmodehigh tmodelow*]
[`TFUEL` *tfuelref tfuelhigh tfuellow*]
[`RHOC` *dcoolref*]
[`RHOM` *dmoderef*]
[`XIR` *pmodref pmodper*]

`EDIT`	keyword used to modify the print level *iprint*.
iprint	index used to control the printing of the module.
`INFOR`	keyword to define the database title.
TITLE	`character*72` title associated with the reactor FMB database.
`DNAME`	keyword to define the database name.
RNAME	`character*12` name of the FBM database.
`PWR`	keyword to define the power levels used for the burnup calculations.
powerref	reference power level (kW).
powerhigh	high power level level (kW).
powerint	intermediate power level level (kW).
powerlow	low power level level (kW).
`TCOOL`	keyword to specify coolant temperature used for reference and perturbed cases.
tcoolref	reference coolant temperature (K).
tcoolhigh	high coolant temperature (K).
tcoollow	low coolant temperature (K).
`TMODE`	keyword to specify moderator temperature used for reference and perturbed cases.
tmoderef	reference moderator temperature (K).
tmodehigh	high moderator temperature (K).
tmodelow	low moderator temperature (K).

TFUEL	keyword to specify fuel temperature used for reference and perturbed cases.
tfuelref	reference fuel temperature (K).
tfuelup	high fuel temperature (K).
tfueldown	low fuel temperature (K).
RHOC	keyword to specify coolant density used for reference calculations.
dcoolref	reference coolant density (g/cm^3).
RHOM	keyword to specify moderator density used reference calculations.
dmoderef	reference moderator density (g/cm^3).
XIR	keyword to specify water purity (D_2O content) used for reference and perturbed cases.
pmodref	reference moderator purity (fraction of D_2O in water).
pmodper	perturbed moderator purity (fraction of D_2O in water).

Other reference and perturbed values are recovered directly from the concentrations and isotope densities stored in the different *CPO*.

3.14 The MRG: module

The MRG: module is used to pre-homogenize a geometry after it has been tracked with the EXCELT: module. This module can also be used for the same purpose for NXT: tracked geometries.[80–82] In addition, NXT: based tracking files can also be partition using this module.

The general specifications for this module are presented in Tables 60 to 62

Table 60: Structure for merging EXCELT: tracks

TRKENEW TFILENEW := MRG: *TRKEOLD TFILEOLD* :: **(descmrg)**

Table 61: Structure for merging NXT: tracks

TRKNNEW := MRG: *TRKNOLD* :: **(descmrg)**

Table 62: Structure for partitioning NXT: tracking files

TFILEMOD TFILEEXT := MRG: *TRKNOLD TFILEOLD* :: **(descextr)**

TRKENEW `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.

TFILENEW `character*12` name of the new `EXCELT:` compatible sequential binary tracking file used to store the tracks lengths after the pre-homogenization process has take place.

TFILEMOD `character*12` name of the new `NXT:` sequential binary tracking file where the lines not-associated with the regions to extract are stored.

TFILEEXT `character*12` name of the new `NXT:` compatible sequential binary tracking file where the lines associated with the regions to extract are stored.

TRKEOLD `character*12` name of the TRACKING data structure that contains region volume and surface area vectors in addition to region identification pointers and other tracking information before the pre-homogenization process.

TFILEOLD `character*12` name of the old sequential binary tracking file used to store the tracks lengths before the pre-homogenization process takes place.

(descmrg) structure containing the input data to this module (see Section 3.14.1).

(descextr) structure containing the input data for track file partitioning process (see Section 3.14.2).

3.14.1 Data input for geometry pre-homogenization

Table 63: Structure **(descmrg)**

[`EDIT` *iprint*]
[`REGI` (*irmrg*(i), i=1,*nreg*)]
[`SURF` (*ismrg*(i), i=1,*nsur*)]

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

iprint index used to control the printing in this module.

`REGI` keyword to specify that regions will be pre-homogenized.

irmrg list of new region numbers associated with old region numbers. Two or more regions can be combined together only if they contain the same mixture. The number *nreg* of region is that printed after the execution of the tracking module.

`SURF` keyword to specify that surfaces will be pre-homogenized.

ismrg list of new surface numbers associated with old surface numbers. Two or more surfaces can be combined together only if they are associated with the same boundary conditions. The number *nsur* of surfaces is that printed after the execution of the tracking module.

3.14.2 Data input for tracking file partitioning

Table 64: Structure **(descextr)**

[EDIT *iprint*]
[EXTR (*iext*(*i*), *i*=1,*nreg*)]

where

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

iprint — index used to control the printing in this module.

EXTR — keyword to specify that the track associated with a specific set of regions will be extracted from the reference tracking file.

iext — list of region numbers for track extraction. The number nreg of region is that printed after the execution of the tracking module.

3.15 The PSP: module

The PSP: module can be used to generate a graphical file in a POSTSCRIPT format (ASCII file) for 2-D geometries that can be analyzed using the EXCELT: and NXT: tracking modules (see Section 3.4). The module PSP: is based on the PSPLOT FORTRAN library from Nova Southeastern University.[83] Since only a few PSPLOT routines were required in DRAGON and because additional routine not present in the original package were needed, the PSPLOT package has been reprogrammed for DRAGON. The POSTSCRIPT files generated by this module can be viewed by several utility programs, such as Ghostview[84] or sent to a printer compatible with this language. The input specifications for this module are presented in Tables 65 and 66.

Table 65: Structure **(PSP:)** for valid EXCELT: geometry

PSGEO := PSP: [*PSGEO*] *GEONAM* [*FLUNAM*] :: **(descpsp)**

Table 66: Structure **(PSP:)** for valid NXT: tracking data structure

PSGEO := PSP: [*PSGEO*] *TRKNAM* [*FLUNAM*] :: **(descpsp)**

PSGEO — character*12 name of the file that contains 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). This option can be used only with geometries that can be processed using the `EXCELT:` module.

TRKNAM `character*12` name of an EXCELL type read-only TRACKING (see Section 3.4). This structure must have been created using the `EXCELT:`, `EXCELL:` or `NXT:` modules.

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 67: Structure **(descpsp)**

[`EDIT` *iprint*]
[`FILL` { `NONE` | `GRAY` | `RGB` | `CMYK` | `HSB` } [`NOCONTOUR`]]
[`TYPE` { `REGION` | `MIXTURE` | `HMIX` | `FLUX` | `MGFLUX` (*icond*(g),$g = 1, N_g$)}]

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

iprint index used to control the printing in this module.

`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 physical mixture.

`HMIX` keyword to specify that different colors or gray levels will be associated with each virtual homogenization mixture (valid only for `NXT:` based tracking data structures).

`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.

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. The number of group condensation entry N_g must satisfy $N_g \leq$*ngroup* where *ngroup* is the total number of groups on the flux data structure.

3.16 The `SAD:` module

The `SAD:` module is used to compute the generalized adjoint fluxes associated with homogenized and condensed cross-sections.[40–44] The input specifications for this module are presented in Table 68.

Table 68: Structure **(SAD:)**

FLUNAM EDINAM `:=` `SAD:` *PIJNAM LIBNAM TRKNAM* `::` **(descsad)**

FLUNAM `character*12` name of the FLUXUNK data structure containing the generalized adjoint fluxes solution.

EDINAM `character*12` name of the EDITION data structure where the homogenized and condensed properties and the generalized adjoint sources will be stored.

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 (see Section 3.4).

(descsad) structure containing the input data to this module (see Section 3.16.1).

3.16.1 Data input for module `SAD:`

Table 69: Structure **(descsad)**

[`EDIT` *iprint*]
[`INIT` { `OFF` | `ON` ((*fluxes*(i, g), i=1,*nregio*), g=1,*ngroup*) }]
[{ `FLX` | `PAF` | `AF` }]
`TYPE` { `N` | `S` | `K` }
[`THER` [*maxthr*] [*epsthr*]]
[`EXTE` [*maxout*] [*epsout*]]

continued on next page

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

[UNKT [*epsunk*]]
[REBA [OFF]]
[ACCE *nlibre naccel*]
[EGPA *epsgpa*]
[CGPA *congpa*]
[SAVE] [NCOR]
[COND { NONE | ({ *icond*(g) | *energy*(g) }, $g = 1, N_g$)] }
[MERG { COMP | NONE | MIX (*imixt*(i), $i = 1, N_m$) | REGI (*ireg*(i), $i = 1, N_r$) }]
[TAKE { MIX (*imixt*(i), $i = 1, N_m$) | REGI (*ireg*(i), $i = 1, N_r$) }]
[SELE { NONE | ALL | TOTAL | TRANC | NUSIGF | NFTOT | SCAD }]

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 see Section 3.7.1.

OFF see Section 3.7.1.

ON see Section 3.7.1.

fluxes see Section 3.7.1.

FLX see Section 3.7.1.

PAF see Section 3.7.1.

AF see Section 3.7.1.

TYPE see Section 3.7.1.

N see Section 3.7.1.

S see Section 3.7.1.

K see Section 3.7.1.

THER see Section 3.7.1.

maxthr see Section 3.7.1.

epsthr see Section 3.7.1.

EXTE see Section 3.7.1.

maxout see Section 3.7.1.

epsout see Section 3.7.1.

UNKT see Section 3.7.1.

epsunk see Section 3.7.1.

REBA see Section 3.7.1.

`OFF`	see Section 3.7.1.
`ACCE`	see Section 3.7.1.
nlibre	see Section 3.7.1.
naccel	see Section 3.7.1.
`EGPA`	see Section 3.7.1.
epsgps	see Section 3.7.1.
`CGPA`	see Section 3.7.1.
congpa	see Section 3.7.1.
`SAVE`	The generalized adjoint sources are saved on EDINAM.
`NCOR`	The correction matrix associated with the high order components of the flux is not computed.
`COND`	see Section 3.9.1.
`NONE`	see Section 3.9.1.
icond	see Section 3.9.1.
energy	see Section 3.9.1.
`MERG`	see Section 3.9.1.
`TAKE`	see Section 3.9.1.
`COMP`	see Section 3.9.1.
`MIX`	see Section 3.9.1.
`REGI`	see Section 3.9.1.
ireg	see Section 3.9.1.
imixt	see Section 3.9.1.
`SELE`	keyword to specify the homogenized and condensed cross sections to take into account for the generalized adjoints computation.
`NONE`	keyword to specify that no cross section will be considered for generalized adjoints computation.
`ALL`	keyword to specify that all the cross sections available will be considered for generalized adjoints computation.
`TOTAL`	keyword to specify that the total cross section will be considered for generalized adjoints computation.
`TRANC`	keyword to specify that the transport correction cross section will be considered for generalized adjoints computation.
`NUSIFG`	keyword to specify that the fission neutron production cross section ($\nu\Sigma_f$) will be considered for generalized adjoints computation.
`NFTOT`	keyword to specify that the fission cross section will be considered for generalized adjoints computation.
`SCAT`	keyword to specify that the scattering cross section will be considered for generalized adjoints computation.

3.17 The PER: module

The PER: module is used to perform generalized perturbation theory calculations in DRAGON.[40–44] The input specifications for this module are presented in Table 70.

Table 70: Structure **(PER:)**

EDINAM := PER: *EDINAM LIBNAM TRKNAM FLUNAM LIBNAP* [*FLUNAP*] :: **(descper)**

EDINAM character*12 name of the EDITION data structure containing the perturbed homogenized and condensed cross sections. *EDINAM* must appears on the RHS after being computed with SAD: for the reference state.

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 (see Section 3.4).

FLUXNAM character*12 name of the FLUXUNK data structure containing the reference flux (see Section 3.16).

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

FLUXNAP character*12 name of the FLUXUNK data structure containing the perturbed flux (see Section 3.7).

(descper) structure containing the input data to this module (see Section 3.17.1).

3.17.1 Data input for module PER:

Table 71: Structure **(descper)**

[SAVE ON NAMREC]
[TYPE DIR | CALC | PER]
[LIN]

SAVE keyword to specify that the results of the perturbative calculations are to be saved on a sub-directory of *EDINAM*.

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

NAMREC character*12 name of the sub-directory of *EDINAM* where the perturbed homogenized and condensed cross sections are to be saved (a MACROLIB).

`TYPE`	keyword to specify the type of perturbation calculations to perform.
`DIR`	keyword to specify that the reference flux is to be used in the perturbation calculations.
`PER`	keyword to specify that the perturbed flux is to be used in the perturbation calculations.
`CALC`	keyword to specify that the perturbed homogenized and condensed cross sections are to be explicitly calculated. In this case, the perturbed flux is required.
`LIN`	keyword to specify that the linear perturbation formulas are used. In this case, the perturbed flux is required. This can be used for sensitivity coefficient calculations.

3.18 The `HST:` module

The `HST:` module is designed to manage a full reactor execution in DONJON using explicit DRAGON calculations for each cell.[46–49] This module can save in an HISTORY data structure the information available in BURNUP data structures generated by DRAGON (see Table 72). It can also read MAP data structures generated by DONJON[45] to prepare the HISTORY data structure for a new series of cell calculations in DRAGON (see Table 73). The HISTORY data structures can also be used to update MAP data structures (see Table 74). Finally, the module `HST:` can be used to create an initial BURNUP data structure that can be used to burn the cell another time step in DRAGON (see Table 75).

Table 72: Updating or creating an HISTORY structure using a BURNUP structure

HISTORY `:=` `HST:` [*HISTORY*] [*BURNUP*] `::`
[**(hstdim)**]
[`GET` **(hstpar)**]
[`CELLID` *icha ibun* [*idfuel*] [`GET` **(hstpar)**]]

Table 73: Updating or creating an HISTORY structure using a MAP structure

HISTORY `:=` `HST:` [*HISTORY*] *MAP* `::`
[**(hstdim)**]
[`GET` **(hstpar)**]

Table 74: Updating a MAP structure using an HISTORY structure

MAP `:=` `HST:` *MAP HISTORY*

Table 75: Creating a BURNUP structure using an HISTORY structure

```
BURNUP := HST: HISTORY ::
[ (hstdim) ]
[ PUT (hstpar) ]
CELLID icha ibun [ PUT {
                        BREFL (hstbrn) (hstpar) AREFL (hstbrn) (hstpar) |
                      [ AREFL ] (hstbrn) (hstpar) }
                 ]
```

The description of the variables and structures presented in Tables 72 to 75 follows.

HISTORY `character*12` name of an HISTORY data structure.

BURNUP `character*12` name of a BURNUP data structure.

MAP `character*12` name of a MAP data structure.

(hstdim) structure containing the dimensions for the HISTORY data structure.

`CELLID` keyword to identify the cell for which history information is to be processed.

icha channel number for which history information is to be processed.

ibun bundle number for which history information is to be processed.

idfuel fuel type number associated with this cell. One can associate to each fuel cell a different fuel type. By default a single fuel type is defined and it fills every fuel cell. Only the initial properties of each fuel type are saved. These properties are used for refueling.

`GET` keyword to specify that the values of the parameters selected in **(brnpar)** will be read from the input stream or CLE-2000 local variables and stored on the HISTORY data structure.

`PUT` keyword to specify that the values of the parameters selected in **(brnpar)** will be read from the HISTORY data structure and transferred to local CLE-2000 variables.

`BREFL` keyword to specify that the information to extract from the HISTORY data base is related to the properties of the cell before refueling takes place.

`AREFL` keyword to specify that the information to extract from the HISTORY data base is related to the properties of the cell after refueling took place.

(hstbrn) structure containing the burnup options.

(hstpar) structure containing the local parameters options.

The **(hstdim)** input structure is required for general dimensioning purpose. It is generally used only when creating the HISTORY data structure. However, the number of global and local parameters used in a HISTORY data structure can be increased at all time. The number of channels, bundles and the refueling scheme must be defined at the creation of the HISTORY data structure. This information can be provided manually or extracted from a MAP data structure. The general form of the **(hstdim)** input is presented in Table 76.

Table 76: Structure **(hstdim)**

```
[ EDIT iprint ]
[ DIMENSIONS [ GLOBAL nglo ] [ LOCAL nloc ] [ BUNDLES nbun bunl ] [ CHANNELS ncha ] ]
```

EDIT	keyword used to modify the print level *iprint*.
iprint	index used to control the printing in this module.
DIMENSIONS	keyword used to indicate that the general dimensioning of the HISTORY data structure will be modified.
GLOBAL	keyword used to modify the number of global parameters on the HISTORY data structure.
nglo	the number of global parameters. The history module will use the maximum between *nglob* and the value, if any, defined on the HISTORY data structure.
LOCAL	keyword used to modify the number of local parameters on the HISTORY data structure.
nloc	the number of local parameters. The history module will use the maximum between *nloc* and the value, if any, defined on the HISTORY data structure.
BUNBLES	keyword used to specify the number of bundles per channel for the reactor model considered in the HISTORY data structure.
nbun	the number of bundles per channel for the reactor model. If *nbun* is different from the value already defined on the HISTORY or MAP data structures, the execution is aborted.
bunl	bundle length in cm. This information is required to compute inital fuel weight.
CHANNELS	keyword used to specify the number of fuel channels for the reactor model considered in the HISTORY data structure.
ncha	the number of fuel channels for the reactor model. If *ncha* is different from the value already defined on the HISTORY or MAP data structures, the execution is aborted.

The **(hstbrn)** serves a unique purpose, mainly to extract from the HISTORY file the information required to process a burnup evaluation in DRAGON using the EVO: module. The information must be stored inside CLE-2000 variables. The general form of this output structure is presented in Table 77.

Table 77: Structure **(hstbrn)**

```
BURN period power
```

BURN	keyword to indicate that burnup information follows.
period	the burnup period (in days) that will be transferred to a real CLE-2000 variable.
power	the power density (in kW/kg) that will be transferred to a real CLE-2000 variable.

The **(hstpar)** is used for two purposes:

1. to define the names of the local and global parameters that are required for the calculations as well as to initialize these parameters;
2. to extract from a HISTORY data structure the values of local and global parameters.

The general form of this structure is presented in Table 78.

Table 78: Structure **(hstpar)**

[[*NAMPAR valpar*]]

where

NAMPAR name of a local or global parameter to process. The parameters specified before the keyword *CELLID* is read will be considered global otherwise they will be considered local.

valpar real value for the local or global parameter to process. In the case where the `GET` option is activated, the history module will extract this parameter from the input data stream. In the case where the `PUT` option is activated, the history module will try to transfer this information into a real CLE-2000 variable.

3.19 The `TLM:` module

The `TLM:` module generates a Matlab[50] `m-file` (`ASCII` file format) that contains the instructions for plotting the tracking lines generated by the `NXT:` (option `LONG` activated in Table 32).[51] The input specifications for this module are presented in Table 79.

Table 79: Structure **(TLM:)**

MFILE `:=` `TLM:` *MFILE TRKNAM TRKFIL* `::` **(desctlm)**

MFILE `character*12` name of the `ASCII` Matlab `m-file` that will contain the instructions for plotting the tracking lines.

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

TRKFIL `character*12` name of the sequential binary tracking file used to store the tracks lengths.

(desctlm) structure describing the type of graphics generated (see Section 3.19.1).

3.19.1 Data input for module `TLM:`

Table 80: Structure **(desctlm)**

```
[ EDIT iprint ]
[ NTPO nplots ]
( {
   REGIONS [ NoPause ] ireg |
   POINTS [ NoPause ] |
   DIRECTIONS [ NoPause ] DIR idir [ PLAN iplan { U iuv | V iuv } ] |
   PLANP [ NoPause ] DIR idir DIST dist [ PLAN iplan ] |
   PLANA [ NoPause ] A a B b [ C c ] D d
   }, iplot=1, nplots )
```

EDIT	keyword used to modify the print level *iprint*.
iprint	index used to control the printing in this module.
NTPO	keyword to specify the number of figures to draw.
nplots	integer value for the number of figures to draw.
REGIONS	keyword to specify that the figure will illustrate only the lines associated with a given region.
POINTS	keyword to specify that the figure will illustrate the intersection points between the lines and the external faces of the geometry.
DIRECTIONS	keyword to specify that the figure will illustrate the lines crossing each region as well as the intersection points between the lines and the external faces of the geometry.
PLANP	keyword to specify that the figure will illustrate the points crossing a plane normal to the line direction.
PLANA	keyword to specify that the figure will illustrate the points crossing an arbitrary surface in 3-D or line in 2-D. The equation for the surface in 3-D is : $$aX + bY + cZ = d$$ while the equation for the line in 2-D is : $$aX + bY = d$$
NoPause	keyword to specify that all the lines must be drawn without Matlab pause. By default, there is a pause after all the points associated with an external surface and all the lines associated with a region are drawn.
ireg	region number for which line illustration will be provided.
DIR	keyword to specify the line direction to draw.
idir	integer value to identify the track direction to draw. In the case where *idir*=0, all the directions will be drawn. A value of *idir*=0 for 2-D geometry is generally acceptable. However, for 3-D geometry the number of lines generated is such that the figure becomes a mess and it is generally more convenient to draw the lines direction per direction.

PLAN	keyword to specify which of the three planes normal to the specified direction in 3-D will be considered for drawing. This plane is defined by the axes $U-V$. Used only for 3-D geometries.
iplan	integer value to identify which of the three planes normal to the specified direction in 3-D will be considered for drawing. The only values permitted are 0, 1, 2 or 3. When a value of 0 is specified (default) all three planes will be drawn. Used only for 3-D geometries.
U	keyword to specify that all the lines in the V axis associated with a position on the U axis will be drawn. Used only for 3-D geometries.
V	keyword to specify that all the lines in the U axis associated with a position on the V axis will be drawn. Used only for 3-D geometries.
iuv	integer value to identify the position on the U or V axis to be drawn. Used only for 3-D geometries.
DIST	keyword to specify the distance between the plane normal and the line direction and the origin.
dist	real or double precision value for the distance of the plane from the origin.
A	keyword to specify the value of *a* for an arbitrary plane or line.
a	real or double precision value *a*.
B	keyword to specify the value of *b* for an arbitrary plane or line.
b	real or double precision value *b*.
C	keyword to specify the value of *c* for an arbitrary plane.
b	real or double precision value *c*.
D	keyword to specify the value of *d* for an arbitrary plane or line.
d	real or double precision value *d*.

3.20 The FMT: module

The utility module FMT: is used to format various data structure to suit the specific user needs. Here two formatting options are available.

1. The SUS3D option where three files are created that respectively contain the integration weights and directions (ASCII), the directional flux (binary or ASCII) and the directional adjoints (binary or ASCII) in a CP or S_N format.[52,85] The input specifications for this option are presented in Table 81.
2. The DIRFLX option where a single file is created that contain the directional flux, adjoints and generalized adjoints. The input specifications for this option are presented in Table 82.

Table 81: Structure **(FMT:)** for SUS3D option

WGTANGL DFLUX DADJOINTS := FMT: *FLUX VOLTRK* ::
[EDIT *iprint*]
SUS3D [{ SN | CP }]

Table 82: Structure **(FMT:)** for DIRFLX option

DAF := FMT: *FLUX VOLTRK* ::
[EDIT *iprint*]
DIRFLX

WGTANGL character*12 name of the ASCII file that will contain the angular weights and directions.

DFLUX character*12 name of the ASCII or BINARY file that will contain the directional flux in a SUS3D compatible format.

DADJOINTS character*12 name of the ASCII or BINARY file that will contain the directional adjoints in a SUS3D compatible format.

DAF character*12 name of the ASCII file that will contain the weights, angular directions and directional flux, adjoints and generalized adjoints in a DIRFLX compatible format (see Appendix A.1 for a description of the format for this file).

FLUX character*12 name of the FLUXUNK data structure to process.

VOLTRK character*12 name of the TRACKING data structure to process.

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

iprint index used to control the printing in this module.

SUS3D keyword to activate the SUS3D processing option.

SN keyword to generate S_N compatible fluxes and adjoints (cell edge values). It is the default value.

CP keyword to generate CP compatible fluxes and adjoints (cell averaged values).

DIRFLX keyword to activate the DIRFLX processing option.

3.21 The ITR: module

The utility module ITR: performs dedicated editing to generate TRIPOLI importances files. The input specifications for this module are presented in Table 83.

Table 83: Structure **(ITR:)**

XMLIMP := FMT: *FLUX VOLTRK LIBNAM* :: **(descitr)**

XMLIMP character*12 name of the XML (ASCII format) file importance that will contain the TRIPOLI importances.

FLUX character*12 name of the FLUXUNK data structure to process.

VOLTRK `character*12` name of the TRACKING data structure to process.

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

(descitr) structure containing the input data to this module (see Section 3.21.1).

3.21.1 Data input for module `ITR:`

Table 84: Structure **(descasm)**

[`EDIT` *iprint*]
[`COND` { `NONE` | ({ *icond*(g) | *energy*(g) }, $g = 1, N_g$)] }
[`MERG` { `COMP` | `NONE` | `MIX` (*imixt*(i), $i = 1, N_m$) | `REGI` (*ireg*(i), $i = 1, N_r$) }]
[`TAKE` { `MIX` (*imixt*(i), $i = 1, N_m$) | `REGI` (*ireg*(i), $i = 1, N_r$) }]
`ORIGINE` (*orpg*(i), $i = 1, 3$)
`REPERE` ((*rep*(i, j), $i = 1, 3$), $j = 1, 3$)
`NBMESH` (*nb*(i), $i = 1, 3$)

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

iprint index used to control the printing of this module.

`NONE` see Section 3.9.1.

`COND` see Section 3.9.1.

icond see Section 3.9.1.

energy see Section 3.9.1.

`MERG` see Section 3.9.1.

`TAKE` see Section 3.9.1.

`COMP` see Section 3.9.1.

`MIX` see Section 3.9.1.

`REGI` see Section 3.9.1.

ireg see Section 3.9.1.

imixt see Section 3.9.1.

`ORIGINE` keyword used to define the absolute origin in space of the cell.

orpg array containing the X, Y and Z positions of the origin.

`REPERE` keyword used to define the three axis direction vectors.

rep	array containing the X, Y and Z components of the $\vec{i}$, $\vec{j}$ and $\vec{j}$ axis vectors.
NBMESH	keyword used to define the number of mesh point in each direction for adjoint weighting.
rep	array containing the number of mesh points in the $\vec{i}$, $\vec{j}$ and $\vec{j}$ directions.

4 EXAMPLES

Here we present a few examples of DRAGON input structures in such a way 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:

$$\Sigma_s(\Omega \cdot \Omega') = \Sigma_s(\mu) = \sum_{l=0}^{L} \frac{(2l+1)}{4\pi} \Sigma_{s,l} P_l(\mu).$$

Since the Legendre polynomials satisfy the following orthogonality conditions:

$$\int_{-1}^{1} d\mu P_l(\mu) P_m(\mu) = \frac{2\delta_{l,m}}{(2l+1)},$$

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})
0	1	0.90	0.80	0.00
	2	0.00	0.70	0.60
	3	0.00	0.30	0.40
1	1	0.09	0.05	0.08
	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   1 1  0.90
       3 3  0.30  0.70  0.80
       2 3  0.40  0.60
       3 3  0.03  0.00  0.09
       2 2  0.07  0.05
       3 3  0.04  0.06 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 12):

  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 13):

  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 14):

  The first annular geometry, namely ANNPIN, can be analyzed using both the JPMT: and EXCELT: tracking modules since the pins in the annular clusters are all located between annular regions. For the geometry ANNSPIN this is no longer the case and only the EXCELT: tracking module can be used. Similarly, the CARPIN geometry can be analyzed using both the JPMT:, EXCELT: and NXT: tracking modules. The geometry CARSPIN, which is based on CARPIN, can only be treated by the EXCELT: and NXT: tracking modules since the pins in the clusters intersect the annular regions defined by the SPLITR option. Finally, for the geometry CAROPIN, which is based also based on CARPIN can only be treated by the NXT: tracking modules because some pins are located outside the last annular regions in the cell.

```
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 ; ;
ANNSPIN := GEO: ANNPIN ::
  SPLITR   3 8 8 ;
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 ; ;
CARSPIN := GEO: CARPIN ::
  SPLITR   3 8 8 ;
CAROPIN := GEO: CARPIN ::
  RADIUS 0.0 0.75 2.75 2.75 ;
```

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 15):

  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 16):

  This geometry can only be analyzed using the `EXCELT:` and `NXT:` 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 17).

  Here we consider an infinite lattice having two types of cells such that

$$\begin{pmatrix} \text{pource}(1) \\ \text{pource}(2) \end{pmatrix} = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix}$$

  and

$$\begin{pmatrix} \text{procel}(1,1) & \text{procel}(1,2) \\ \text{procel}(2,1) & \text{procel}(2,2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1/2 & 1/2 \end{pmatrix}$$

  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 except `NXT:`.

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 **TCM13**.

*4.3.1 **TCM01** – Annular region*

This sample input is used to analyze the annular cell presented in Figure 18. 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.x2m**

```
*----
* 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 0.039
  ;
*----
*  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, 86] The corresponding geometry is shown in Figure 19 where the cell numbers generated by DRAGON are shown.

Input data for test case: **TCM02.x2m**

```
*----
*  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 20). This benchmark was first proposed by Watanabe and Maynard. Akroyd and Riyait used it to analyze the performance of various codes.[18, 19, 87]

Input data for test case: **TCM03.x2m**

```
*----
*  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  2  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  4  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  6  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  8  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 21).

Input data for test case: **TCM04.x2m**

```
*----
*  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 :: ;
SYS2 TRACK2 := 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 ::
  ;
SYS2 TRACK2 := 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 23).[88]

Input data for test case: **TCM05.x2m**

```
*----
*  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.x2m**

```
*----
*  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.x2m**

```
*----
*  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 that contains both a fission and a fixed source.

Input data for test case: **TCM08.x2m**

```
*----
*  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" .
```

*4.3.9 **TCM09** – Solution of a 2-D fission source problem using the `MOCC:` module*

This test case is for a 3×3 Cartesian assembly in 2-D similar to TCM03. It is solved using the method of cyclic characteristics.

Input data for test case: **TCM09.x2m**

```
*----
* TEST CASE TCM09
* MACROSCOPIC CROSS SECTIONS
* FIXED SOURCE PROBLEM
* CARTESIAN 3 X 3 ASSEMBLY
* WATANABE-MAYNARD PROBLEM SIMILAR TO TCM03
*
* REF: R. Roy, "The Cyclic Characteristics Method,"
*      Int. Conf. Physics of Nuclear Science and Technology,
*      Long Island, NY, October 1998, pp. 407-414.
*----
*
LINKED_LIST WATA WAT24 TRACK MACRO FLUX EDITION ;
```

```
SEQ_BINARY WATATRK ;
STRING  PolarAng := "CACB" ;
MODULE GEO: EXCELT: MAC: MOCC: EDI: DELETE: END: ;
* MACROSCOPIC CROSS SECTIONS
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 ;
* GEOMETRIES  ENTERED WITH SYMMETRIES
* WAT24  - 24 X 24 REGIONS
WATA := GEO: :: CAR2D 3 3
      X- DIAG X+ VOID
      MESHX 0.00 1.25 5.00 10.00
      Y- REFL Y+ DIAG
      MESHY 0.00 1.25 5.00 10.00
      MIX     1    3    2
                   3    2
                        2         ;
WAT24 := GEO: WATA ::
      SPLITX  3  9 12
      SPLITY  3  9 12       ;
* SOLUTION FOR WAT24
TRACK WATATRK := EXCELT: WAT24 ::
     TITLE 'TCM03: WATANABE-MAYNARD 24X24 '
     MAXR 300
     TRAK TSPC 12 12.0 ;
FLUX := MOCC: MACRO TRACK WATATRK ::
  <<PolarAng>> TYPE S THER 1.E-6 100 EXTE 1.E-6 100 ;
* UPPER QUADRANT FLUX FOR 24X24
* RESULTS GIVEN IN TABLE 1. (ref. p. 411 )
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 0 0 0 0 0 0 0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
      0 0 0 0 0 0 0 0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
        0 0 0 0 0 0 0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
          0 0 0 0 0 0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
            0 0 0 0 0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
              0 0 0 0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
                0 0 0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
                  0 0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
                    0 0 0   0  0  0  0  0  0  0  0  0  0  0  0
                      0 0   0  0  0  0  0  0  0  0  0  0  0  0
                        0   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
  ;
* FLUX AT X=5.625CM FOR 24X24
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  2  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  4  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  6  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  8  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" .
```

*4.3.10 **TCM10** – Solution of a 2-D fixed source problem using the `MOCC:` module*

This test case is for a 2–D Cartesian assembly that contains a fixed source. It is solved using the method of cyclic characteristics.

Input data for test case: **TCM10.x2m**

```
*
* TEST CASE TCM11
* MACROSCOPIC CROSS SECTIONS
* FIXED SOURCE PROBLEM
* CARTESIAN 2 X 2 ASSEMBLY
* TSAI-LOYALKA SEMI-INFINITE PROBLEM
*
* REF: R. Roy, "The Cyclic Characteristics Method,"
*      Int. Conf. Physics of Nuclear Science and Technology,
*      Long Island, NY, October 1998, pp. 407-414.
*
LINKED_LIST LOYA LOY25 TRACK FLUX EDITION
            MACRO MACRO100 MACRO050 MACRO010 MACRO005 MACRO000 ;
SEQ_BINARY LOYATRK ;
STRING  PolarAng := "CACB" ;
MODULE GEO: EXCELT: MAC: MOCC: EDI: DELETE: END: ;
INTEGER  i n := 1 1 ;
*
* MACROSCOPIC CROSS SECTIONS
MACRO100 := MAC: ::
 NGRO 1 NMIX 2
 READ INPUT
   MIX 1 TOTAL 1.0 SCAT 1 1 1.00 FIXE 1.0
   MIX 2 TOTAL 1.0 SCAT 1 1 1.00 ;
MACRO050 := MAC: ::
 NGRO 1 NMIX 2
 READ INPUT
   MIX 1 TOTAL 1.0 SCAT 1 1 0.50 FIXE 1.0
   MIX 2 TOTAL 1.0 SCAT 1 1 0.50 ;
MACRO010 := MAC: ::
 NGRO 1 NMIX 2
 READ INPUT
   MIX 1 TOTAL 1.0 SCAT 1 1 0.10 FIXE 1.0
   MIX 2 TOTAL 1.0 SCAT 1 1 0.10 ;
MACRO005 := MAC: ::
 NGRO 1 NMIX 2
 READ INPUT
   MIX 1 TOTAL 1.0 SCAT 1 1 0.05 FIXE 1.0
   MIX 2 TOTAL 1.0 SCAT 1 1 0.05 ;
MACRO000 := MAC: ::
 NGRO 1 NMIX 2
 READ INPUT
   MIX 1 TOTAL 1.0 SCAT 1 1 0.00 FIXE 1.0
   MIX 2 TOTAL 1.0 SCAT 1 1 0.00 ;
* GEOMETRIES  ENTERED WITH SYMMETRIES
* LOYA   -  2 X  2 REGIONS
* LOY25  - 25 X 25 REGIONS
LOYA := GEO: :: CAR2D 2 2
      X- REFL X+ VOID
      MESHX 0.00 0.52 1.00
```

```
       Y- REFL Y+ REFL
       MESHY 0.00 0.52 1.00
       MIX     1    2
               2    2      ;
LOY25 := GEO: LOYA ::
       SPLITX  13 12
       SPLITY  13 12       ;
* SOLUTION FOR LOY25
TRACK LOYATRK := EXCELT: LOY25 ::
     TITLE 'TCM03: LOYANABE-MAYNARD 24X24 '
     MAXR 625
     TRAK TSPC 12 100.0 ;
REPEAT
IF i 1 = THEN
  MACRO := MACRO100 ;
ENDIF ;
IF i 2 = THEN
  MACRO := MACRO050 ;
ENDIF ;
IF i 3 = THEN
  MACRO := MACRO010 ;
ENDIF ;
IF i 4 = THEN
  MACRO := MACRO005 ;
ENDIF ;
IF i 5 = THEN
  MACRO := MACRO000 ;
ENDIF ;
FLUX := MOCC: MACRO TRACK LOYATRK ::
  <<PolarAng>> TYPE S ;
* SOLUTION FOR LOY25
* FLUX AT X=Y= 0.50, 0.70 AND 0.98
* SEE TABLE 2. (ref. p. 412)
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 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0

  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0

  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 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   0 0 0 0 0   0 0 0 0 0
  0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
```

```
   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0

   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 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   0 0 0 0 0   0 0 0 0 0
   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0

   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 3
    ;
FLUX MACRO EDITION := DELETE: FLUX MACRO EDITION ;
EVALUATE i := i 1 + ;
UNTIL i 6 = ;
LOYATRK := DELETE: LOYATRK ;
END: ;
QUIT "LIST" .
```

4.3.11 ***TCM11*** – *Comparison of CP and MoC solutions*

This test case is for a 4×4 Cartesian assembly in 2-D. It is solved using the method of cyclic characteristics and the method of collision probabilities using specular (mirror like) boundary conditions.

Input data for test case: **TCM11.x2m**

```
*
* TEST CASE TCM12
* MACROSCOPIC CROSS SECTIONS
* FIXED SOURCE PROBLEM
* CARTESIAN 4 X 4 ASSEMBLY WITH FUEL RODS AND POISON
* KAVENOKY BENCHMARK
*
* REF: R. Roy, "The Cyclic Characteristics Method,"
*      Int. Conf. Physics of Nuclear Science and Technology,
*      Long Island, NY, October 1998, pp. 407-414.
*
STRING Polar_Ang := "CACB"  ;
INTEGER   Nazimuth  :=  8    ;
REAL      DenTrak   := 100.  ;
INTEGER   Nsplit    :=  5    ;
REAL      Tolerance := 5.E-6 ;
LINKED_LIST PWR TRACK MACRO SYS FLUX EDITION ;
SEQ_BINARY  PWRTRK ;
MODULE GEO: EXCELT: MAC: MOCC: EDI: END: DELETE: GREP: ;
REAL ou := 100. ;
REAL
  f1 f2 f3 f8 f9 f10 f11 f12 f13 f14 f15
  v1 v2 v3 v8 v9 v10 v11 v12 v13 v14 v15 ;
```

```
REAL
  r1     r2    r3
               r8    r9    r10
  r11    r12   r13   r14   r15  :=
  5.166  3.699 4.183
               3.178 3.617 0.2847
  2.913  3.441 3.937 3.225 3.673 ;
REAL
  e1     e2    e3
               e8    e9    e10
  e11    e12   e13   e14   e15  :=
  0.11   0.08  0.07
               0.04  0.05  0.002
  0.03   0.03  0.04  0.05  0.05 ;
EVALUATE
  e1      e2      e3
                  e8      e9      e10
  e11     e12     e13     e14     e15   :=
  e1 r1 / e2 r2 / e3 r3 /
                  e8 r8 / e9 r9 / e10 r10 /
  e11 r11 / e12 r12 / e13 r13 / e14 r14 / e15 r15 / ;
EVALUATE
  e1      e2      e3
                  e8      e9      e10
  e11     e12     e13     e14     e15   :=
  e1 ou * e2 ou * e3 ou *
                  e8 ou * e9 ou * e10 ou *
  e11 ou * e12 ou * e13 ou * e14 ou * e15 ou * ;
*----
* Macroscopic XS
*----
MACRO := MAC: ::
  NGRO 1 NMIX 19
  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  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX  4 TOTAL  0.625  SCAT 1 1  0.355  FIXE 0.000
  MIX  5 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX  6 TOTAL  0.625  SCAT 1 1  0.355  FIXE 0.000
  MIX  7 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX  8 TOTAL  0.625  SCAT 1 1  0.355  FIXE 0.000
  MIX  9 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX 10 TOTAL 14.000  SCAT 1 1  0.000  FIXE 0.000
  MIX 11 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX 12 TOTAL  0.625  SCAT 1 1  0.355  FIXE 0.000
  MIX 13 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX 14 TOTAL  0.625  SCAT 1 1  0.355  FIXE 0.000
  MIX 15 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX 16 TOTAL  0.625  SCAT 1 1  0.355  FIXE 0.000
  MIX 17 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
  MIX 18 TOTAL  0.625  SCAT 1 1  0.355  FIXE 0.000
```

```
    MIX 19 TOTAL  1.250  SCAT 1 1  1.242  FIXE 1.000
    ;
  *----
  *  Geometry : PWR - Cartesian 7x7
  *  Tracking : EXCELT
  *----
  PWR := GEO: :: CAR2D 4 4
    X- DIAG  X+ REFL Y- SYME  Y+ DIAG
    CELL   WA  F2  F4  F6
               F8  P10 F12
                   F14 F16
                       F18
    ::: WA := GEO: CAR2D 1 1
      MESHX -0.625  0.625 SPLITX <<Nsplit>>
      MESHY -0.625  0.625 SPLITY <<Nsplit>>
      MIX          1               ;
    ::: F2  := GEO: CARCEL 1
      MESHX -0.625  0.625 SPLITX <<Nsplit>>
      MESHY -0.625  0.625 SPLITY <<Nsplit>>
      RADIUS 0.000 0.450
      MIX         2     3          ;
    ::: F4  := GEO: F2
      MIX         4     5          ;
    ::: F6  := GEO: F2
      MIX         6     7          ;
    ::: F8  := GEO: F2
      MIX         8     9          ;
    ::: P10 := GEO: F2
      MIX        10    11 SPLITR 3 ;
    ::: F12 := GEO: F2
      MIX        12    13          ;
    ::: F14 := GEO: F2
      MIX        14    15          ;
    ::: F16 := GEO: F2
      MIX        16    17          ;
    ::: F18 := GEO: F2
      MIX        18    19          ;
    ;
  TRACK PWRTRK := EXCELT: PWR ::
    MAXR 300 TRAK TSPC <<Nazimuth>> <<DenTrak>>  ;
  FLUX := MOCC: MACRO TRACK PWRTRK ::
    <<Polar_Ang>> TYPE S
    THER <<Tolerance>> 100 EXTE <<Tolerance>> 100 ;
  EDITION := EDI: FLUX MACRO TRACK ::
    EDIT 2 SAVE
    MERGE  MIX  1 2 3 0 0 0 0 4 5 6 7 8 9 10 11 0 0 0 0  ;
   GREP: EDITION ::
    STEP UP 'REF-CASE   1' STEP UP MACROLIB
    GETVAL VOLUME 1 11
    >>v1<< >>v2<< >>v3<< >>v8<< >>v9<< >>v10<<
    >>v11<< >>v12<< >>v13<< >>v14<< >>v15<<
      STEP UP 'GROUP  1/  1'
```

```
      GETVAL FLUX-INTG 1 11
    >>f1<< >>f2<< >>f3<< >>f8<< >>f9<< >>f10<<
    >>f11<< >>f12<< >>f13<< >>f14<< >>f15<<
      STEP DOWN
    STEP DOWN STEP DOWN ;
  EVALUATE f1 f2 f3 f8 f9 f10 f11 f12 f13 f14 f15 :=
                   f1 v1 / r1 - r1 / ou *
                   f2 v2 / r2 - r2 / ou *
                   f3 v3 / r3 - r3 / ou *
                   f8 v8 / r8 - r8 / ou *
                   f9 v9 / r9 - r9 / ou *
                   f10 v10 / r10 - r10 / ou *
                   f11 v11 / r11 - r11 / ou *
                   f12 v12 / r12 - r12 / ou *
                   f13 v13 / r13 - r13 / ou *
                   f14 v14 / r14 - r14 / ou *
                   f15 v15 / r15 - r15 / ou *

                   ;
  * SOLUTION FOR KAVENORY BENCHMARK
  * FLUX VALUES COMPARED TO MONTE-CARLO RESULTS
  * SEE TABLE 3. (ref. p. 412)
  ECHO "DF( 1/ 3)%=" f1  f2  f3  ;
  ECHO "DF( 8/ 9)%="     f8  f9  ;
  ECHO "DF(10/12)%=" f10 f11 f12 ;
  ECHO "DF(13/15)%=" f13 f14 f15 ;
  ECHO "ACCEPT=" f1 ABS e1 <= f2 ABS e2 <= f3 ABS e3 <= ;
  ECHO "ACCEPT="              f8 ABS e8 <= f9 ABS e9 <= ;
  ECHO "ACCEPT=" f10 ABS e10 <= f11 ABS e11 <= f12 ABS e12 <= ;
  ECHO "ACCEPT=" f13 ABS e13 <= f14 ABS e14 <= f15 ABS e15 <= ;
  PWRTRK := DELETE: PWRTRK ;
  END: ;
  QUIT .
```

*4.3.12 **TCM12** - Solution of a 3-D problem using the `MCU:` module*

This test case is for a simplified 3-D Cartesian assembly analyzed using the `EXCELT:`. A collisions probability solution is generated as well as two solutions using the method of characteristics.

Input data for test case: **TCM12.x2m**

```
*----
* TEST CASE TCM12
* MACROSCOPIC CROSS SECTIONS
* 3-D CARTESIAN ASSEMBLY
* USE the 3-D CHARACTERISTICS MODULE MCU:
*  1) TEST1: USE TRACKING FILE
```

```
*  2) TEST2: USE EXCELL TRACKING TYPE
*----
*  Define STRUCTURES and MODULES used
*----
MODULE   MAC: GEO: MCU: END: EXCELT: DELETE: EXCELL: FLU: ASM: ;
LINKED_LIST MACRO GEOM T3D FLUXA PIJMatrix ;
SEQ_BINARY  IntLine        ;
REAL     DenTra := 10.0    ;
INTEGER  AngTra := 4       ;
STRING   Itlm ;
EVALUATE Itlm := "ITLM"    ;
STRING   Prll := "STRD"    ;
INTEGER  Merg ;
EVALUATE Merg  := 0        ;
STRING   Etab  ;
EVALUATE Etab := "OFF"     ;
STRING   jacc ;
EVALUATE jacc  := "JACC"   ;
*  Validate input options
IF Itlm "ITLM" = NOT THEN
   EVALUATE Itlm := " "    ;
ENDIF ;
*
*----
* Macroscopic cross sections
*----
MACRO := MAC: ::
EDIT 0 NGRO 7 NMIX 7 NIFI 1
ENER  1.0E7 1.0E5 1.0E3 10.0 4.0 0.625 0.025 0.001
READ INPUT
  MIX      1
    EFISS    200.0
    NFTOT    7.21206E-03 8.19301E-04 6.45320E-03 1.85648E-02
             1.78084E-02 8.30348E-02 2.16004E-01
    NUSIGF   2.00600E-02 2.02730E-03 1.57060E-02 4.51830E-02
             4.33421E-02 2.02090E-01 5.25711E-01
    CHI      5.87910E-01 4.11760E-01 3.39060E-04 1.17610E-07
             0.00000E+00 0.00000E+00 0.00000E+00
    TOTAL    1.77949E-01 3.29805E-01 4.80388E-01 5.54367E-01
             3.11801E-01 3.95168E-01 5.64406E-01
    SCAT         1     1 1.27537E-01
                 2     2 3.24456E-01 4.23780E-02
                 3     3 4.50940E-01 1.63140E-03 9.43740E-06
                 5     5 1.25250E-04 4.52565E-01 2.67920E-03
                           3.14270E-09 5.51630E-09
                 3     6 1.29680E-03 2.71401E-01 5.56640E-03
                 3     7 8.54580E-03 2.65802E-01 1.02550E-02
                 3     7 2.73080E-01 1.68090E-02 1.00210E-08
  MIX      2
    EFISS    200.0
    NFTOT    8.25446E-03 1.32565E-03 8.42156E-03 3.28730E-02
             1.59636E-02 3.23794E-01 3.62803E-01
```

```
      NUSIGF   2.38140E-02 3.85869E-03 2.41340E-02 9.43662E-02
               4.57699E-02 9.28181E-01 1.04320E+00
      CHI      5.87910E-01 4.11760E-01 3.39060E-04 1.17610E-07
               0.00000E+00 0.00000E+00 0.00000E+00
      TOTAL    1.81323E-01 3.34368E-01 4.93785E-01 5.91216E-01
               4.74198E-01 8.33601E-01 8.53603E-01
      SCAT         1     1 1.30457E-01
                   2     2 3.28428E-01 4.17920E-02
                   3     3 4.58371E-01 1.64360E-03 8.51050E-06
                   5     5 1.76190E-04 4.63709E-01 2.53310E-03
                             2.20170E-09 5.13290E-09
                   3     6 2.27600E-03 2.82313E-01 5.47660E-03
                   3     7 8.86450E-03 2.49751E-01 8.72890E-03
                   3     7 2.59529E-01 1.31140E-02 9.00160E-09

    MIX      3
      TOTAL    1.26032E-01 2.93160E-01 2.84240E-01 2.80960E-01
               3.34440E-01 5.65640E-01 1.17215E+00
      SCAT         1     1 6.61659E-02
                   2     2 2.40377E-01 5.90700E-02
                   3     3 1.83297E-01 5.24350E-02 2.83340E-04
                   5     5 3.73330E-05 7.88511E-02 9.23970E-02
                             2.49900E-04 1.46220E-06
                   6     6 9.17260E-04 9.97372E-02 1.70140E-01
                             6.94460E-03 1.92390E-05 2.06420E-08
                   6     7 4.97920E-02 3.16765E-01 2.06790E-01
                             2.58810E-02 1.08030E-03 2.98750E-06
                   6     7 1.09912E+00 2.38770E-01 2.44780E-02
                             4.92970E-03 2.05670E-04 4.21400E-07
    MIX      4
      TOTAL    1.59206E-01 4.12970E-01 5.90310E-01 5.84350E-01
               7.18000E-01 1.25445E+00 2.65038E+00
      SCAT         1     1 4.44777E-02
                   2     2 2.82334E-01 1.13400E-01
                   3     3 3.45256E-01 1.29940E-01 7.23470E-04
                   5     5 7.14370E-05 9.10284E-02 2.24570E-01
                             6.23400E-04 3.74990E-06
                   6     6 2.21570E-03 1.39138E-01 4.15510E-01
                             1.69990E-02 4.80020E-05 5.31840E-08
                   6     7 1.32440E-01 6.99913E-01 5.11820E-01
                             6.37320E-02 2.64430E-03 7.44860E-06
                   6     7 2.48070E+00 5.37320E-01 6.12290E-02
                             1.21390E-02 5.03440E-04 1.04550E-06
  ;
*-----
* Geometry
*-----
GEOM := GEO: :: CAR3D 4 4 2
 X- DIAG X+ REFL Y+ DIAG Y- SSYM  Z- SSYM  Z+ REFL
   CELL
    C1 C1 C1 C1
       C6 C6 C6
```

```
           C6 C6
              C6
     C7 C7 C7 C7
        C7 C7 C7
           C7 C7
              C7
::: C1 := GEO: CARCELZ 1 1
  MESHX 0.0 1.26  MESHY 0.0 1.26  MESHZ 0.0 1.26
  RADIUS 0.0 0.54  MIX 1 2 ;
::: C6 := GEO: C1  MIX 3 3 ;
::: C7 := GEO: C1  MESHZ 0.0 1.26
    MIX 4 4 ;
;
*-------
* Use EXCELL
*-------
PIJMatrix T3D := EXCELL: GEOM  MACRO ::
         MAXR 400
         TRAK SUBG 1 TISO 4 10.0  ;
FLUXA := FLU: MACRO PIJMatrix T3D ::
         TYPE K ;
FLUXA T3D PIJMatrix  := DELETE: FLUXA T3D PIJMatrix ;
*-------
* Use EXCELT to track then ASM+FLU or MCU
*-------
T3D  IntLine  := EXCELT: GEOM  MACRO ::
           EDIT 0 MAXR 400
           TRAK TISO <<AngTra>> <<DenTra>>      ;
PIJMatrix := ASM:  MACRO T3D  IntLine           ;
FLUXA  := FLU:  PIJMatrix MACRO T3D       ::
              TYPE K                            ;
FLUXA    := DELETE: FLUXA                 ;
FLUXA  := MCU: T3D  IntLine  MACRO ::
                TYPE K
                ETAB <<Etab>>
                CURR DIRT  <<Itlm>> ;
FLUXA T3D IntLine  := DELETE: FLUXA T3D IntLine ;
*-------
* MCU: Use EXCELL tracking option
*-------
FLUXA T3D := MCU: GEOM  MACRO ::
                 TYPE K
                 ETAB <<Etab>>
                 CURR DIRT  <<Itlm>>
                 MAXR 400
                 TRAK TISO <<AngTra>> <<DenTra>> ;
FLUXA T3D  := DELETE: FLUXA T3D  ;
GEOM MACRO := DELETE: GEOM MACRO ;
END: ;
QUIT  .
```

*4.3.13 **TCM13** - Hexagonal assembly with hexagonal cells containing clusters*

This test represents an example of a 2-D hexagonal assembly filled with triangular/hexagonal cells containing clusters (see Figure 22) that can be analyzed with `NXT:`.

Input data for test case: **TCM13.x2m**

```
*----
* TEST CASE TCM13
* MACROSCOPIC CROSS SECTIONS
* FISSION SOURCE PROBLEM
* HEXAGONAL CELL with PINS
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST  MacLib GlobalGeo Tracking Pij Flux ;
SEQ_ASCII    Fig.ps ;
SEQ_BINARY   Lines ;
MODULE       MAC: GEO: NXT: PSP: ASM: FLU: DELETE: END: ;
*----
*  Macroscopic XS
*----
MacLib := MAC: ::
  NGRO 2 NMIX 18 NIFI 1
  READ INPUT
  MIX  1   TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX  2  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX  3  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX  4  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX  5  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX  6  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX  7  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX  8  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX  9  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX 10   TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX 11   TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX 12  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX 13  TOTAL 0.166667 1.111111
    SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
  MIX 14   TOTAL 0.166667 1.111111
```

```
      SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
    MIX 15  TOTAL 0.166667 1.111111
      SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
    MIX 16  TOTAL 0.166667 1.111111
      SCAT   2 2  0.00015 0.126667 2 2 1.10111 0.039
    MIX 17   TOTAL 0.222222 0.833333
      SCAT   1 1  0.19222          2 2 0.75333 0.02
      NUSIGF  0.0  0.170 CHI 1.0 0.0
    MIX 18   TOTAL 0.222222 0.833333
      SCAT   1 1  0.19222          2 2 0.75333 0.02
      NUSIGF  0.0  0.170 CHI 1.0 0.0
    ;
  *----
  *  Geometry :  Hexagonal assembly containing
  *  hexagons with 4 triangular crown and pins
  *----
  GlobalGeo := GEO: ::  HEX 7
    HBC COMPLETE REFL
    CELL C1 C2 C1 C2 C1 C2 C1
    ::: C1 := GEO: HEXT 4
      SIDE 4.0
      MIX
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      CLUSTER ROD1 ROD2
      ::: ROD1 := GEO: TUBE 2 MIX 17 18 NPIN  1 RPIN 0.0000 APIN 0.0000
        RADIUS 0.00000 0.6122 0.6540 ;
      ::: ROD2 := GEO: ROD1   MIX 17 18 NPIN  6 RPIN 1.4885 APIN 0.0000 ;
      ;
    ::: C2 := GEO: HEXT 4
      SIDE 4.0 1.1
      MIX
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
      ;
    ;
  *----
  *  Tracking : NXT
  *  Solution : PIJ
  *----
  Lines Tracking := NXT: GlobalGeo ::
    EDIT 2 LONG NORE TISO 3 10.0 ;
  Fig.ps := PSP: Tracking ;
  Pij := ASM: MacLib Tracking Lines ;
```

```
Flux := FLU: Pij MacLib Tracking :: TYPE K ;
Flux Pij := DELETE:  Flux Pij ;
GlobalGeo Tracking Lines := DELETE: GlobalGeo Tracking Lines ;
END: ;
QUIT "LIST" .
```

4.4 WLUP microscopic cross section examples

The test cases we present here use the `LIB:` module to provide microscopic cross sections taken from a WIMS-D4 format library. We will assume that this library is located in file `iaea`.[34] This file is the result of processing the file `iaea.lib` downloaded form the IAEA WLUP site with the utility `WILLIE.f`.[34] An example of how to process such files for DRAGON can be found in `ftp://ftp.polymtl.ca/pub/nucl/WLUP.tgz`.

Our test cases are numbered successively from **TCWU01** to **TCWU17**.

*4.4.1 **TCWU01** – The Mosteller benchmark*

This benchmark uses both a cartesian 2-D cell with a central annular pin and an equivalent annular cell (see Figure 23).[88] No depletion information is required since the module `EVO:` will not be executed. A comparison between various calculation options is provided here. We first consider an 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: **TCWU01.x2m**

```
*----
*  TEST CASE TCWU01
*  MOSTELLER BENCHMARK: 1-D ANNULAR CELL AND 2-D CARTESIAN CELL
*  WIMSD4 69 GROUPS LIBRARY FILE iaea from WLUP
*
*  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 iaea format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 3 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: iaea
  MIX 1 600.0                          O16      = '6016'     4.61309E-2
    U235     = '2235'   1.66078E-4 1
    U238     = '8238'   2.28994E-2 1
```

```
  MIX 2 600.0
    Zr91     = '91'     3.83243E-2
  MIX 3 600.0
    H1H2O    = '3001'   4.42326E-2   O16H2O   = '6016'     2.21163E-2
    BNat     = '1011'   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 'TCWU01: MOSTELLER BENCHMARK (JPM / SYBIL)'
  MAXR 4 IP01 QUA1 5 ;
LIBRARY := SHI: LIBRARY DISCR1 :: ;
DISCR2 := SYBILT: MOSTELA ::
  TITLE 'TCWU01: 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 'TCWU01: MOSTELLER BENCHMARK (JPM / SYBIL)'
  MAXR 4 IP01 QUA1 5 QUA2 12 5 ;
LIBRARY := SHI: LIBRARY DISCR1 :: ;
DISCR2 := SYBILT: MOSTELC ::
  TITLE 'TCWU01: 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 'TCWU01: 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 'TCWU01: 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 **TCWU02** – 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 24.

Input data for test case: **TCWU02.x2m**

```
*----
*  TEST CASE TCWU02
*  17 X 17 PWR ASSEMBLY WITHOUT POISON
*
*  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 iaea format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 8 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: iaea
  MIX 1 579.9
    H1H2O    = '3001'   4.76690E-2   O16H2O   = '6016'   2.38345E-2
    BNat     = '1011'   2.38103E-5
  MIX 2 579.9                         O16      = '6016'   3.06711E-4
    Cr52     = '52'     7.54987E-5   Fe56     = '2056'   1.47624E-4
    Zr91     = '91'     4.18621E-2
  MIX 3 579.9
    H1H2O    = '3001'   4.65292E-2   O16H2O   = '6016'   2.32646E-2
                                      Cr52     = '52'     4.79927E-5
    Fe56     = '2056'   4.45845E-5   Ni58     = '58'     1.13521E-4
                                      MoNat    = '96'     4.03755E-6
                                      Al27     = '27'     2.35231E-6
    Mn55     = '55'     4.15901E-7
    BNat     = '1011'   2.32761E-5
    Zr91     = '91'     8.92427E-4
  MIX 4 579.9                         O16      = '6016'   2.87335E-4
    Cr52     = '52'     7.07291E-5   Fe56     = '2056'   1.38298E-4
    Zr91     = '91'     3.92175E-2
  MIX 5 579.9
    H1H2O    = '3001'   4.71346E-2   O16H2O   = '6016'   2.35673E-2
                                      Cr52     = '52'     2.24991E-5
    Fe56     = '2056'   2.09013E-5   Ni58     = '58'     5.32188E-5
                                      MoNat    = '96'     1.89281E-6
                                      Al27     = '27'     1.10277E-6
    Mn55     = '55'     1.94976E-7
    BNat     = '1011'   2.35598E-5
    Zr91     = '91'     4.18372E-4
  MIX 6 579.9
    H1H2O    = '3001'   4.71676E-2   O16H2O   = '6016'   2.35838E-2
                                      Cr52     = '52'     2.11122E-5
    Fe56     = '2056'   1.96130E-5   Ni58     = '58'     4.99383E-5
                                      MoNat    = '96'     1.77614E-6
                                      Al27     = '27'     1.03479E-6
    Mn55     = '55'     1.82957E-7
    BNat     = '1011'   2.35753E-5
```

```
      Zr91     = '91'     3.92583E-4
    MIX 7 579.9
      H1H2O    = '3001'   4.72020E-2   O16H2O   = '6016'   2.36010E-2
                                        Cr52     = '52'     1.96591E-5
      Fe56     = '2056'   1.82630E-5   Ni58     = '58'     4.65011E-5
                                        MoNat    = '96'     1.65389E-6
                                        Al27     = '27'     9.63569E-7
      Mn55     = '55'     1.70365E-7
      BNat     = '1011'   2.35914E-5
      Zr91     = '91'     3.65562E-4
    MIX 8 933.6                         O16      = '6016'     4.49355E-2
      U235     = '2235'   7.39237E-4 1
      U238     = '8238'   2.17285E-2 1
    ;
  *----
  *  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 'TCWU02: 17 X 17 MULTICELL PWR BENCHMARK WITHOUT POISON'
  MAXR 400 OLD ;
LIBRARY := SHI: LIBRARY DISCR1 :: ;
DISCR2 := SYBILT: ASSMB ::
  TITLE 'TCWU02: 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 **TCWU03** – An hexagonal assembly*

This test case represents a production calculation of a typical hexagonal control assembly. Its configuration is presented in Figure 25.

Input data for test case: **TCWU03.x2m**

```
*----
*  TEST CASE TCWU03
*  MULTICELL HEXAGONAL ASSEMBLY WITH POISON
*  iaea WLUP Library
*
*  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 iaea format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 11 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: iaea
  MIX 1 579.9
    H1H2O    = '3001'   4.76690E-2   O16H2O   = '6016'   2.38345E-2
    BNat     = '1011'   2.38103E-5
  MIX 2 579.9                         O16      = '6016'   3.06711E-4
    Cr52     = '52'     7.54987E-5   Fe56     = '2056'   1.47624E-4
    Zr91     = '91'     4.18621E-2
  MIX 3 579.9
    H1H2O    = '3001'   4.65292E-2   O16H2O   = '6016'   2.32646E-2
                                      Cr52     = '52'     4.79927E-5
    Fe56     = '2056'   4.45845E-5   Ni58     = '58'     1.13521E-4
                                      MoNat    = '96'     4.03755E-6
                                      Al27     = '27'     2.35231E-6
    Mn55     = '55'     4.15901E-7
    BNat     = '1011'   2.32761E-5
    Zr91     = '91'     8.92427E-4
  MIX 4 933.6                         O16      = '6016'   4.49355E-2
    U235     = '2235'   7.39237E-4 1
    U238     = '8238'   2.17285E-2 1
  MIX 5 579.9                         In115    = '2115'   7.57464E-3
    Cd113    = '2113'   2.62493E-3
    Ag109    = '3109'   4.49188E-2
  MIX 6 579.9                         Cr52     = '52'     1.52702E-2
    Fe56     = '2056'   5.57670E-2   Ni58     = '58'     7.51418E-3
    Mn55     = '55'     8.02943E-4
  MIX 7 579.9
    H1H2O    = '3001'   3.06466E-2   O16H2O   = '6016'   1.53233E-2
    Fe56     = '2056'   5.27485E-5   Cr52     = '52'     2.69769E-5
    BNat     = '1011'   1.53077E-5
    Zr91     = '91'     1.49580E-2
  MIX 8 579.9
    H1H2O    = '3001'   4.65292E-2   O16H2O   = '6016'   2.32646E-2
                                      Cr52     = '52'     4.79927E-5
    Fe56     = '2056'   4.45845E-5   Ni58     = '58'     1.13521E-4
                                      MoNat    = '96'     4.03755E-6
                                      Al27     = '27'     2.35231E-6
    Mn55     = '55'     4.15901E-7
    BNat     = '1011'   2.32761E-5
    Zr91     = '91'     8.92427E-4
  MIX 9 579.9                         O16      = '6016'   2.87335E-4
    Cr52     = '52'     7.07291E-5   Fe56     = '2056'   1.38298E-4
    Zr91     = '91'     3.92175E-2
  MIX 10 579.9
```

```
    H1H2O    = '3001'   4.71346E-2   O16H2O   = '6016'   2.35673E-2
                                      Cr52     = '52'     2.24991E-5
    Fe56     = '2056'   2.09013E-5   Ni58     = '58'     5.32188E-5
                                      MoNat    = '96'     1.89281E-6
                                      Al27     = '27'     1.10277E-6
    Mn55     = '55'     1.94976E-7   BNat     = '1011'   2.35598E-5
    Zr91     = '91'     4.18372E-4
  MIX 11 579.9
    H1H2O    = '3001'   4.71676E-2   O16H2O   = '6016'   2.35838E-2
                                      Cr52     = '52'     2.11122E-5
    Fe56     = '2056'   1.96130E-5   Ni58     = '58'     4.99383E-5
                                      MoNat    = '96'     1.77614E-6
                                      Al27     = '27'     1.03479E-6
    Mn55     = '55'     1.82957E-7
    BNat     = '1011'   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 'TCWU03: MULTICELL HEXAGONAL ASSEMBLY WITH POISON'
  MAXR 400 MAXZ 15000 OLD ;
LIBRARY := SHI: LIBRARY DISCR1 :: ;
DISCR2 := SYBILT: ASSMBH ::
```

```
    TITLE 'TCWU03: 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 ***TCWU04*** – *A Cylindrical cell with burnup*

This test case represents a burnup calculation for the Mosteller annular geometry (see Figure 23).

Input data for test case: **TCWU04.x2m**

```
*----
*  TEST CASE TCWU04
*  iaea WLUP Library
*  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 iaea format WIMSD4
```

```
*  Microscopic cross sections from file iaea format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 3 CTRA WIMS
  DEPL LIB: WIMSD4 FIL: iaea
  MIXS LIB: WIMSD4 FIL: iaea
  MIX 1 600.0                          O16      = '6016'   4.61309E-2
    U235     = '2235'   1.66078E-4 1
    U238     = '8238'   2.28994E-2 1
    U236     = '236'    0.0        1
    Pu239    = '6239'   0.0        1
  MIX 2 600.0
    Zr91     = '91'     3.83243E-2
  MIX 3 600.0
    H1H2O    = '3001'   4.42326E-2   O16H2O   = '6016'   2.21163E-2
    BNat     = '1011'   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 'TCWU04: MOSTELLER BENCHMARK WITH BURNUP'
  EDIT 1 MAXR 3 ;
LIBRARY := SHI: LIBRARY TRACKS :: ;
TRACK := SYBILT: MOSTELA ::
  TITLE 'TCWU04: 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 **TCWU05** – 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 26. 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.[2] This test case uses the embedded DRAGON procedure stored in the `TCWU05Lib.c2m` file.

Input data for test case: **TCWU05.x2m**

```
*----
*  TEST CASE TCWU05
*  CANDU-6 ANNULAR CELL
*  iaea WLUP Library
*  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
*
*----
*  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
  fuel mode ;
MODULE
  GEO: EXCELT: SHI: ASM: FLU: EVO: EDI: CPO:
  DELETE: END: ;
*----
```

```
*  Depletion data from file iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*----
PROCEDURE TCWU05Lib ;
INTEGER   iedit := 1 ;
LIBRARY := TCWU05Lib :: <<iedit>> ;
*----
*  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 'TCWU05: 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 'TCWU05: 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 ;
fuel := COMPO1 ;
COMPO2 := CPO: EDITION ::
  STEP 'EDITMOD' NAME MIXTMOD ;
mode := COMPO2 ;
INTLINF INTLINS := DELETE: INTLINF INTLINS ;
END: ;
QUIT "LIST" .
```

Input data for test case: **TCWU05Lib.c2m**

```
*----
*  Procedure TCWU05Lib
*  Create Library for test CASE TCWU05
*  Calling :
*      LIBRARY := TCWU05Lib :: iedit ;
*  with :
*      LIBRARY = Linked list containing the result of LIB: for
*                TCWU05
*      iprint  = print level for LIB: module
*----
*  Define PARAMETERS,STRUCTURES and MODULES used
*----
PARAMETER  LIBRARY  ::  ::: LINKED_LIST LIBRARY ; ;
MODULE     LIB: DELETE: END: ;
*----
*  Define and read LIB: EDIT option
INTEGER iedit   ;
:: >>iedit<<    ;
*----
*  Depletion data from file iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*----
LIBRARY := LIB: ::
  EDIT <<iedit>>
  NMIX 17 CTRA WIMS
  DEPL LIB: WIMSD4 FIL: iaea
  MIXS LIB: WIMSD4 FIL: iaea
  MIX  1  560.66   0.81212           O16      = '6016'   7.99449E-1
    D2D2O    = '3002'   1.99768E-1   H1H2O    = '3001'   7.83774E-4
  MIX  2  560.66   6.57              Nb93     = '93'     2.50000
    BNat     = '1011'   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     = '2056'   1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '1011'   3.10000E-4
    Zr91     = '91'     9.97100E+1
  MIX  5  345.66   1.082885          O16      = '6016'   7.98895E-1
    D2D2O    = '3002'   2.01016E-1   H1H2O    = '3001'   8.96000E-5
  MIX  6  941.29  10.4375010         O16      = '6016'   1.18473E+1
    Xe135    = '4135'   0.0
    U235     = '2235'   6.27118E-1 1
    U238     = '8238'   8.75256E+1 1
    U236     = '236'    0.0        1
    Pu239    = '6239'   0.0        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     = '2056'   1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
```

```
      BNat     = '1011'   3.10000E-4
      Zr91     = '91'     9.97100E+1
    MIX 11  COMB 10 1.0
    MIX 12  COMB 10 1.0
    MIX 13  COMB 10 1.0
    MIX 14 COMB  1 1.0
    MIX 15 COMB  1 1.0
    MIX 16 COMB  1 1.0
    MIX 17 COMB  1 1.0
    ;
  END: ;
  QUIT "LIST" .
```

4.4.6 ***TCWU06*** *– 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 26) and the supercell containing a stainless steel rod which can be either in the inserted or extracted position (see Figure 21). 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.[2] This test case also uses the embedded DRAGON procedure stored in the `TCWU05Lib.c2m` file.

Input data for test case: **TCWU06.x2m**

```
*----
*  TEST CASE TCWU06
*  CANDU-6 CARTESIAN CELL
*  iaea WLUP Library
*  STAINLESS STELL RODS IN 3D SUPERCELL
*
*----
*  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: ;
*----
*  Depletion data from file iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*----
PROCEDURE TCWU05Lib ;
INTEGER   iedit := 1 ;
LIBRARY := TCWU05Lib :: <<iedit>> ;
*----
*  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 'TCWU06: 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 'TCWU06: 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: iaea
  MIX 4    345.66                    Fe56     = '2056'   6.19027E-2
    Cr52     = '52'     1.56659E-3   Ni58     = '58'     6.83337E-3
    Si29     = '29'     7.79072E-4   C12      = '2012'   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 **TCWU07** – 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 26) using various leakage options. This test case also uses the embedded DRAGON procedure stored in the `TCWU05Lib.c2m` file.

Input data for test case: **TCWU07.x2m**

```
*----
*  TEST CASE TCWU07
*  CANDU-6 CARTESIAN CELL
*  iaea WLUP Library
*  TEST VARIOUS LEAKAGE OPTIONS
*
*----
*  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 ;
*----
*  Depletion data from file iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*----
PROCEDURE TCWU05Lib ;
INTEGER   iedit := 1 ;
LIBRARY := TCWU05Lib :: <<iedit>> ;
*----
*  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 'TCWU07: 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 'TCWU07: 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 'TCWU07: 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 'TCWU07: 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 **TCWU08** – Burnup of an homogeneous cell*

This case illustrates 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: **TCWU08.x2m**

```
*----
*  TEST CASE TCWW08
*  HOMOGENEOUS DEPLETION CASE
*  iaea WLUP Library
*
*  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, TotalTime
*        Timei = initial time
*        Timef = final time
*        TotalTime = Final time reached
*     d) Print variable Iprint
*        = 1 reduced print
*        = 3 full print
*----
REAL
  Power  Delt  Timec Timei Timef TotalTime :=
  600.0  10.0  50.0  0.0   0.0   2000.0 ;
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 iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 1 CTRA WIMS
  DEPL LIB: WIMSD4 FIL: iaea
  MIXS LIB: WIMSD4 FIL: iaea
  MIX 1 300.0                        H1       = '3001'       2.00000E+1
    U235     = '2235'     1.0        1
    U236     = '8238'     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 'TCWW08: 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 TotalTime < 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 **TCWU09** – Testing boundary conditions*

This case tests different boundary conditions for the Mosteller cell (see Figure 23).

Input data for test case: **TCWU09.x2m**

```
*----
*  TEST CASE TCWU09
*  MOSTELLER BENCHMARK FOR 1-D ANNULAR CELL
*  iaea WLUP Library
*  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 iaea format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 3 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: iaea
  MIX 1 600.0                          O16      = '6016'   4.61309E-2
    U235     = '2235'   1.66078E-4 1
    U238     = '8238'   2.28994E-2 1
  MIX 2 600.0
    Zr91     = '91'     3.83243E-2
  MIX 3 600.0
    H1H2O    = '3001'   4.42326E-2   O16H2O   = '6016'   2.21163E-2
    BNat     = '1011'   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 'TCWU09: JPM TRACK MOSTELLER BENCHMARK REFLECTIVE BC '
  MAXR 4 IP01 QUA1 5 ;
LIBRARY := SHI: LIBRARY VOLMAT :: ;
VOLMAT := DELETE: VOLMAT ;
VOLMAT := SYBILT: MOSTELA ::
  TITLE 'TCWU09: 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 'TCWU09: 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 **TCWU10** – Fixed source problem in multiplicative media*

This case verifies the use of a fixed source inside a cell where fission also takes place.

Input data for test case: **TCWU10.x2m**

```
*----
*  TEST CASE TCWU10
*  MOSTELLER BENCHMARK FOR 1-D ANNULAR CELL
*  iaea WLUP Library
*  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 iaea format WIMSD4
*  Fixed source of 1.0E5 in group 6
*----
LIBRARY := LIB: ::
  EDIT 0 NMIX 3  CTRA WIMS
  MIXS LIB: WIMSD4 FIL: iaea
  MIX 1 600.0                          O16      = '6016'   4.61309E-2
    U235     = '2235'   1.66078E-4 1
    U238     = '8238'   2.28994E-2 1
  MIX 2 600.0
    Zr91     = '91'     3.83243E-2
  MIX 3 600.0
    H1H2O    = '3001'   4.42326E-2   O16H2O   = '6016'   2.21163E-2
    BNat     = '1011'   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 'TCWU10: JPM TRACK MOSTELLER BENCHMARK'
```

```
  MAXR 4 IP01 QUA1 5 ;
LIBRARY := SHI: LIBRARY VOLMAT :: ;
VOLMAT := DELETE: VOLMAT ;
VOLMAT := SYBILT: MOSTELA ::
  TITLE 'TCWU10: 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 **TCWU11** – Two group burnup of a CANDU–6 type cell*

This case is similar to **TCWU05** except that the burnup module uses DRAGON generated two groups time dependent microscopic cross sections. This test case also uses the embedded DRAGON procedure stored in the `TCWU05Lib.c2m` file.

Input data for test case: **TCWU11.x2m**

```
*----
*  TEST CASE TCWU11
*  CANDU-6 ANNULAR CELL
*  iaea WLUP Library
*  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
*
*----
*  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 ;
LINKED_LIST
   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 iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*----
PROCEDURE TCWU05Lib ;
INTEGER   iedit := 1 ;
LIBRARY := TCWU05Lib :: <<iedit>> ;
*----
*  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 'TCWU11: 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 'TCWU11: 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 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 **TCWU12** – Mixture composition*

This case illustrates the use of the `INFO:` module of DRAGON (see Section 3.12) as well as the `COMB` option in the module `LIB:` (see Section 3.2).

Input data for test case: **TCWU12.x2m**

```
*----
*  TEST CASE TCWU12
*  iaea WLUP Library
*  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: iaea
  ISO: 3 '3001'   '3002'   '6016'
  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: iaea
  ISO: 3 '2235'    '8238'    '6016'
  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 iaea format WIMSD4
*----
LIBRARY := LIB: ::
  NMIX 8 CTRA WIMS
  MIXS LIB: WIMSD4 FIL: iaea
  MIX  1  <<TempCool>> <<DensCool>>  O16      = '6016'   <<WO16C>>
    D2D2O    = '3002'   <<WD2C>>     H1H2O    = '3001'   <<WH1C>>
  MIX  2  <<TempFuel>> <<DensFuel>>  O16      = '6016'   <<WO16F>>
    U235     = '2235'   <<WU235F>> 1
    U238     = '8238'   <<WU238F>> 1
  MIX  3 COMB 1 0.5 0 0.5
  MIX  4 COMB 1 0.1 2 0.9
  ;
END: ;
QUIT "LIST" .
```

4.4.13 ***TCWU13*** – *Solution by the method of cyclic characteristics*

This case illustrates the use of the `MOCC:` module of DRAGON for a solution by the transport equation by the method of cyclic characteristics. This test case also uses the embedded DRAGON procedure stored in the

`TCWU05Lib.c2m` file.

Input data for test case: **TCWU13.x2m**

```
*----
*  TEST CASE TCWW13
*  CANDU-6 ANNULAR CELL
*  iaea WLUP Library
*  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
*
*----
*  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, TotalTime
*        Timei = initial time
*        Timef = final time
*        TotalTime = Final time reached
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY CANDU6S CANDU6F VOLMATF PIJ FLUX ;
SEQ_BINARY
  INTLINF ;
SEQ_ASCII
  flxxel flxmoc ;
MODULE
  GEO:  SHI: EXCELT: ASM: FLU: MOCC:
  DELETE: END: ;
*----
*  Depletion data from file iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
```

```
*----
PROCEDURE TCWU05Lib ;
INTEGER   iedit := 1 ;
LIBRARY := TCWU05Lib :: <<iedit>> ;
*----
*  Geometry CANDU6S : 13 regions annular cluster for self-shielding
*           CANDU6F : 31 regions annular cluster for transport
*----
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: ::  CARCEL 5
  X- REFL X+ REFL Y- REFL Y+ REFL
  MESHX 0.0 28.375
  MESHY 0.0 28.375
  RADIUS  0.00000 5.16890 5.60320  6.44780 6.58750 14.0
  MIX  1 2 3 4 5 5
  SPLITR  6 1 1 1 10
  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 SPLITR 2 1 ;
  ::: 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 ;
  ;
VOLMATF INTLINF := EXCELT: CANDU6S ::
  EDIT 0 MAXR 100 TRAK TISO 29 20.0 ;
LIBRARY := SHI: LIBRARY VOLMATF INTLINF ::
  EDIT 0 ;
VOLMATF INTLINF := DELETE: VOLMATF INTLINF ;
VOLMATF INTLINF := EXCELT: CANDU6F ::
  TITLE 'TCWW13: CANDU-6 ANNULAR POWER= 31.971 FUEL TEMP= 941.29'
  EDIT 0 MAXR 32 TRAK TSPC 12 10.0  ;
PIJ := ASM: LIBRARY VOLMATF INTLINF :: ;
FLUX := FLU: PIJ LIBRARY VOLMATF ::
  TYPE K ;
flxxel := FLUX ;
FLUX PIJ := DELETE: FLUX PIJ ;
FLUX := MOCC: LIBRARY VOLMATF INTLINF ::
  CACB TYPE K
  THER 1.0E-5 100 EXTE 1.0E-5 100 ;
flxmoc := FLUX ;
FLUX := DELETE: FLUX ;
```

```
INTLINF VOLMATF CANDU6F := DELETE: INTLINF VOLMATF CANDU6F ;
LIBRARY := DELETE: LIBRARY ;
END: ;
QUIT "LIST" .
```

*4.4.14 **TCWU14** – SPH Homogenisation without tracking*

This case illustrates the use of the SPH homogenisation procedure in the EDI: module of DRAGON when a tracking data structure is provided as input. This test case also uses the embedded DRAGON procedure stored in the TCWU05Lib.c2m file.

Input data for test case: **TCWU14.x2m**

```
*----
*  TEST CASE TCWW14
*  CANDU-6 Cartesian CELL
*  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
*
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST
  LIBRARY CANDU6F CANDU6S VOLMATF CANDU6H VOLMATH PIJ FLUX EDITION ;
SEQ_BINARY
  INTLINF INTLINH ;
MODULE
  GEO: EXCELT: LIB: ASM: FLU: EDI: SHI:
  DELETE: END: ;
*----
*  Depletion data from file iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*----
PROCEDURE TCWU05Lib ;
INTEGER   iedit := 1 ;
LIBRARY := TCWU05Lib :: <<iedit>> ;
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 ;
```

```
  ;
VOLMATF INTLINF := EXCELT: CANDU6S ::
  EDIT 0 MAXR 100 TRAK TISO 29 20.0 ;
LIBRARY := SHI: LIBRARY VOLMATF INTLINF ::
  EDIT 0 ;
VOLMATF INTLINF := DELETE: VOLMATF INTLINF ;
CANDU6F := GEO: ::  CARCEL 8
  X- REFL MESHX -14.2875 14.2875 X+ REFL
  Y- REFL MESHY -14.2875 14.2875 Y+ REFL
  RADIUS  0.00000 0.7221626 2.160324 3.600681
          5.168878 5.60320 6.44780 6.587482 14.0
  SPLITR   1  2  2  2  1  1  1  9
  MIX      14  15  16  17
           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 SPLITR 2 1 ;
  ::: ROD2 := GEO: ROD1   MIX  7 11
    NPIN  6 RPIN 1.4885 APIN 0.0000 ;
  ::: ROD3 := GEO: ROD1   MIX  8 12
    NPIN 12 RPIN 2.8755 APIN 0.261799 ;
  ::: ROD4 := GEO: ROD1   MIX  9 13
    NPIN 18 RPIN 4.3305 APIN 0.0 ;
  ;
CANDU6H := GEO: ::  CARCEL 8
  X- REFL MESHX -14.2875 14.2875 X+ REFL
  Y- REFL MESHY -14.2875 14.2875 Y+ REFL
  RADIUS  0.00000 0.7221626 2.160324 3.600681
          5.168878 5.60320 6.44780 6.587482 14.0
  SPLITR   1  2  2  2  1  1  1  9
  MIX      1  2  3  4
           5  6  7  8  8
  ;
VOLMATF INTLINF := EXCELT: CANDU6F ::
  TITLE 'TCWW14 Flux geometry'
  EDIT 0 MAXR 100 TRAK TISO 12 20.0  ;
VOLMATH INTLINH := EXCELT: CANDU6H ::
  TITLE 'TCWW14 Homogenisation geometry'
  EDIT 0 MAXR 100 TRAK TISO 12 20.0  ;
PIJ := ASM: LIBRARY VOLMATF INTLINF :: ;
FLUX := FLU: PIJ LIBRARY VOLMATF ::
  TYPE K ;
EDITION := EDI: FLUX LIBRARY VOLMATF VOLMATH INTLINH ::
  MERGE MIX
  0 5 6 7 8
  1 2 3 4 1
  2 3 4 1 2
  3 4
  SPH MTRK SAVE ;
PIJ FLUX EDITION := DELETE: PIJ FLUX EDITION ;
INTLINF VOLMATF CANDU6F := DELETE: INTLINF VOLMATF CANDU6F ;
```

```
INTLINH VOLMATH CANDU6H := DELETE: INTLINH VOLMATH CANDU6H ;
LIBRARY := DELETE: LIBRARY ;
END: ;
QUIT "LIST" .
```

*4.4.15 **TCWU15** – A CANDU–6 type Cartesian cell with burnup*

This test case is similar to **TCWU05** except that the cell boundary are Cartesian and the `NXT:` tracking module is used. It uses the embedded DRAGON procedure stored in the `TCWU05Lib.c2m` file.

Input data for test case: **TCWU15.x2m**

```
*----
*  TEST CASE TCWU05
*  CANDU-6 ANNULAR CELL
*  iaea WLUP Library
*  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
*
*----
*  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
  fuel mode ;
MODULE
  GEO: NXT: SHI: ASM: FLU: EVO: EDI: CPO:
  DELETE: END: ;
*----
*  Depletion data from file iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*----
PROCEDURE TCWU05Lib ;
INTEGER   iedit := 1 ;
LIBRARY := TCWU05Lib :: <<iedit>> ;
*----
*  Geometry CANDU6S : 13 regions annular cluster for self-shielding
*           CANDU6F : 31 regions annular cluster for transport
*----
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
*----
VOLMATS INTLINS := NXT: CANDU6S ::
  TITLE 'TCWU05: CANDU-6 CARTESIAN POWER= 31.971 FUEL TEMP= 941.29'
  EDIT 0 TRAK TISO 5 10.0 ;
LIBRARY := SHI: LIBRARY VOLMATS INTLINS ::
  EDIT 0 ;
VOLMATF INTLINF := NXT: CANDU6F ::
  TITLE 'TCWU05: CANDU-6 CARTESIAN POWER= 31.971 FUEL TEMP= 941.29'
  EDIT 0 TRAK TISO 5 10.0 ;
```

```
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 ;
fuel := COMPO1 ;
COMPO2 := CPO: EDITION ::
  STEP 'EDITMOD' NAME MIXTMOD ;
mode := COMPO2 ;
INTLINF INTLINS := DELETE: INTLINF INTLINS ;
END: ;
QUIT "LIST" .
```

4.4.16 ***TCWU16*** – *A basic PWR pin cell*

This case represents a basic PWR pin cell. It illustrates the use of the PSP: module.

Input data for test case: **TCWU16.x2m**

```
*---
* A basic PWR pin cel.
* Author : T. Courau
* Date   : 2004
* To test the NXT: and PSP: modules.
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST   GEOMETRY DISCR PIJ SUBGEO ;
SEQ_ASCII     Fig.ps   ;
MODULE GEO: EXCELT: NXT: PSP: DELETE: END: ;
*
*---
* Define & set up parameters
*---
REAL LPitch := 1.26  ;
*---
*  Geometry GEOMETRY : Cartesian 3 region geometry
*---
GEOMETRY := GEO: :: CAR2D 2 2
  X+ REFL   X- DIAG
  Y+ DIAG   Y- SYME
  CELL  C1 C2
           C3
  ::: C1 := GEO: CAR2D 1 1
  MESHX 0.00 <<LPitch>>
  MESHY 0.00 <<LPitch>>
  MIX  1   ;
  ::: C2 := GEO: CAR2D 1 1
  MESHX 0.00 <<LPitch>>
  MESHY 0.00 <<LPitch>>
  MIX  2   ;
  ::: C3 := GEO: CAR2D 1 1
  MESHX 0.00 <<LPitch>>
```

```
  MESHY 0.00 <<LPitch>>
  MIX  3   ;
  ;
DISCR := NXT: GEOMETRY :: EDIT 10
               TISO 4 5.0 ;
Fig.ps := PSP: DISCR  ;
Fig.ps := PSP: Fig.ps DISCR :: TYPE MIXTURE ;
DISCR  := DELETE: DISCR  ;
END: ;
QUIT "LIST" .
```

*4.4.17 **TCWU17** – A 2-D CANDU–6 supercell with control rods*

This test case treats a 2-D CANDU–6 supercell containing fuel clusters and control rods (see Figure 27). The use of the virtual homogenization mixtures defined by `HMIX` is also illustrated. This test case uses the embedded DRAGON procedure stored in the `TCWU17Lib.c2m` file.

Input data for test case: **TCWU17.x2m**

```
*----
* Exemple of the use of HMIX for cell homogenization
* 2-D supercell with fuel clusters based on AECL supercell model
* for G2 SOR and MCA with fuel and reactivity devices parallel
* References
* PREPARED BY   : G. Marleau  on 2013/06/11
*
*----
*  modules and data structures
*----
SEQ_ASCII      MACROLIBF ;
LINKED_LIST    SORINS SORIN TRACK MicLib FLUX EDITION ;
XSM_FILE       ASMPIJ ;
SEQ_ASCII      FigReg.ps FigMix.ps FigHom.ps ;
SEQ_ASCII      HomMix.txt HomHMix.txt ;
SEQ_BINARY     Lines ;
MODULE         GEO: EXCELT: EXCELL: SHI: ASM: LIB: FLU:
               EDI: UTL: DELETE: FREE: END: INFO: CPO: MAC:
               NXT: PSP: ;
PROCEDURE      TCWU17Lib ;
INTEGER   iedit := 1 ;
MicLib := TCWU17Lib :: <<iedit>> ;
*----
* DEFINE GEOMETRY FOR SUPERCELL CALCULATION
*  SORINS  : 2D self-shielding geometry with SHUT-OFF ROD & GT in
*            for annular fuel.
*  SORIN   : 2D transport geometry with SHUT-OFF ROD & GT in
*            for annular fuel.
*----
SORINS := GEO: :: CAR2D 5 3
  EDIT 0
  X- REFL X+ REFL
```

```
Y- REFL Y+ REFL
CELL  M    MXL   MX2  MXR  M
      MY   FXYL  BXY  FXYR MY
      M    MXL   MX2  MXR  M
::: M := GEO: CAR2D 1 1
  MESHX  0.0     7.14375
  MESHY  0.0     7.14375
  HMIX 0
  MIX  5 ;
::: MXL := GEO: CAR2D 2 1
  MESHX -7.14375 0.0 7.14375
  MESHY  0.0     7.14375
  HMIX    0  1
  MIX     5 15 ;
::: MX2 := GEO: CAR2D 1 1
  MESHX -7.14375 7.14375
  MESHY  0.0     7.14375
  HMIX    1
  MIX    15 ;
::: MXR := GEO: CAR2D 2 1
  MESHX -7.14375 0.0 7.14375
  MESHY  0.0     7.14375
  HMIX    1  0
  MIX    15  5 ;
::: MY := GEO: CAR2D 1 2
  MESHY -7.14375 0.0 7.14375
  MESHX  0.0     7.14375
  HMIX    0   0
  MIX     5   5 ;
::: BXY := GEO: CARCEL 2
  MESHX -7.14375 7.14375
  MESHY -7.14375 7.14375
  RADIUS 0.0  6.380  6.530
  HMIX    1      1     1
  MIX    15     15    15
  CLUSTER ROD
  ::: ROD := GEO: TUBE 4
      NPIN 1 RPIN 0.0  APIN  0.0
      RADIUS 0.0 5.4115  5.4877  5.5791  5.6553
      HMIX      1     1      1       1
      MIX      15    14     21      14      ;    ;
::: FXYL := GEO: CARCEL 5 2 1
  MESHX -7.14375 0.0 7.14375
  MESHY -7.14375 7.14375
  RADIUS  0.00000 5.16890 5.60320  6.44780 6.58750  7.00
  HMIX      0       0       0       0       0       0
            1       1       1       1       1       1
  MIX       1       2       3       4       5       5
           11      12      13      14      15      15
  CLUSTER ROD1 ROD2L ROD2R ROD3L ROD3R ROD4L ROD4R
    ::: ROD1  := GEO: TUBE 2 1 2 MIX  6 10 16 20  HMIX 0 0 1 1
                 NPIN  1 RPIN 0.0000 APIN 0.0000
```

```
                    MESHX -0.6540 0.6540
                    MESHY -0.6540 0.0 0.6540
                    RADIUS 0.00000 0.6122 0.6540 ;
      ::: ROD2L := GEO: TUBE 2     MIX  7 10 HMIX 0 0
                    RADIUS 0.00000 0.6122 0.6540
                    NPIN  3 RPIN 1.4885
                    APIN 2.09439510 3.14159265 4.18879020 ;
      ::: ROD2R := GEO: TUBE 2     MIX 17 20 HMIX 1 1
                    RADIUS 0.00000 0.6122 0.6540
                    NPIN  3 RPIN 1.4885
                    APIN -1.04719755 0.0000 1.04719755 ;
      ::: ROD3L := GEO: TUBE 2     MIX  8 10 HMIX 0 0
                    RADIUS 0.00000 0.6122 0.6540
                    NPIN  6 RPIN 2.8755
                    APIN -2.87979327  -2.35619449  -1.83259571
                          1.83259571   2.35619449   2.87979327 ;
      ::: ROD3R := GEO: TUBE 2     MIX 18 20 HMIX 1 1
                    RADIUS 0.00000 0.6122 0.6540
                    NPIN  6 RPIN 2.8755
                    APIN -1.30899694  -0.78539816  -0.26179939
                          0.26179939   0.78539816   1.30899694 ;
      ::: ROD4L := GEO: TUBE 2     MIX  9 10 HMIX 0 0
                    RADIUS 0.00000 0.6122 0.6540
                    NPIN  9 RPIN 4.3305
                    APIN  1.74532925   2.09439510   2.44346095
                          2.79252680   3.14159265   3.49065850
                          3.83972435   4.18879020   4.53785606 ;
      ::: ROD4R := GEO: TUBE 2     MIX 19 20 HMIX 1 1
                    RADIUS 0.00000 0.6122 0.6540
                    NPIN  9 RPIN 4.3305
                    APIN -1.39626340  -1.04719755  -0.69813170
                         -0.34906585   0.0          0.34906585
                          0.69813170   1.04719755   1.39626340 ;
;
::: FXYR := GEO: CARCEL 5 2 1
  MESHX -7.14375 0.0 7.14375
  MESHY -7.14375 7.14375
  RADIUS  0.00000 5.16890 5.60320  6.44780 6.58750  7.00
  HMIX      1       1       1       1       1       1
            0       0       0       0       0       0
  MIX      11      12      13      14      15      15
            1       2       3       4       5       5
  CLUSTER ROD1 ROD2L ROD2R ROD3L ROD3R ROD4L ROD4R
    ::: ROD1  := GEO: TUBE 2 1 2 MIX 16 20  6 10  HMIX 1 1 0 0
                 NPIN  1 RPIN 0.0000 APIN 0.0000
                 MESHX -0.6540 0.6540
                 MESHY -0.6540 0.0 0.6540
                 RADIUS 0.00000 0.6122 0.6540 ;
    ::: ROD2L := GEO: TUBE 2     MIX 17 20 HMIX 1 1
                 RADIUS 0.00000 0.6122 0.6540
                 NPIN  3 RPIN 1.4885
                 APIN 2.09439510 3.14159265 4.18879020 ;
```

```
        ::: ROD2R := GEO: TUBE 2     MIX  7 10 HMIX 0 0
                     RADIUS 0.00000 0.6122 0.6540
                     NPIN  3 RPIN 1.4885
                     APIN -1.04719755 0.0000 1.04719755 ;
        ::: ROD3L := GEO: TUBE 2     MIX 18 20 HMIX 1 1
                     RADIUS 0.00000 0.6122 0.6540
                     NPIN  6 RPIN 2.8755
                     APIN -2.87979327  -2.35619449  -1.83259571
                           1.83259571   2.35619449   2.87979327 ;
        ::: ROD3R := GEO: TUBE 2     MIX  8 10 HMIX 0 0
                     RADIUS 0.00000 0.6122 0.6540
                     NPIN  6 RPIN 2.8755
                     APIN -1.30899694  -0.78539816  -0.26179939
                           0.26179939   0.78539816   1.30899694 ;
        ::: ROD4L := GEO: TUBE 2     MIX 19 20 HMIX 1 1
                     RADIUS 0.00000 0.6122 0.6540
                     NPIN  9 RPIN 4.3305
                     APIN  1.74532925   2.09439510   2.44346095
                           2.79252680   3.14159265   3.49065850
                           3.83972435   4.18879020   4.53785606 ;
        ::: ROD4R := GEO: TUBE 2     MIX  9 10 HMIX 0 0
                     RADIUS 0.00000 0.6122 0.6540
                     NPIN  9 RPIN 4.3305
                     APIN -1.39626340  -1.04719755  -0.69813170
                          -0.34906585   0.0          0.34906585
                           0.69813170   1.04719755   1.39626340 ;
    ;
  ;
  SORIN  := GEO: :: CAR2D 5 3
    EDIT 0
    X- REFL X+ REFL
    Y- REFL Y+ REFL
    CELL  M    MXL   MX2  MXR  M
          MY   FXYL  BXY  FXYR MY
          M    MXL   MX2  MXR  M
    ::: M := GEO: CAR2D 1 1
      MESHX  0.0     7.14375   SPLITX  2
      MESHY  0.0     7.14375   SPLITY  2
      HMIX 0
      MIX  5 ;
    ::: MXL := GEO: CAR2D 2 1
      MESHX -7.14375 0.0 7.14375  SPLITX 3  3
      MESHY  0.0     7.14375   SPLITY 3
      HMIX    0  1
      MIX     5 15 ;
    ::: MX2 := GEO: CAR2D 1 1
      MESHX -7.14375 7.14375   SPLITX 6
      MESHY  0.0     7.14375   SPLITY 3
      HMIX    1
      MIX    15 ;
    ::: MXR := GEO: CAR2D 2 1
      MESHX -7.14375 0.0 7.14375  SPLITX 3  3
```

```
  MESHY  0.0     7.14375   SPLITY 3
  HMIX    1  0
  MIX    15  5 ;
::: MY := GEO: CAR2D 1 2
  MESHY -7.14375 0.0 7.14375   SPLITY 3  3
  MESHX  0.0     7.14375   SPLITX 3
  HMIX    0   0
  MIX     5   5 ;
::: BXY := GEO: CARCEL 2
  MESHX -7.14375 7.14375   SPLITX 6
  MESHY -7.14375 7.14375   SPLITY 6
  RADIUS 0.0  6.380  6.530
  SPLITR  2      2
  HMIX    1      1     1
  MIX    15     15    15
  CLUSTER ROD
  ::: ROD := GEO: TUBE 4
      NPIN 1 RPIN 0.0  APIN  0.0
      RADIUS 0.0 5.4115  5.4877  5.5791  5.6553
      SPLITR    1     1      1       1
      HMIX      1     1      1       1
      MIX      15    14     21      14      ;    ;
::: FXYL := GEO: CARCEL 5 2 1
  MESHX -7.14375 0.0 7.14375   SPLITX 3 3
  MESHY -7.14375 7.14375   SPLITY 6
  RADIUS  0.00000 5.16890 5.60320  6.44780 6.58750  7.00
  HMIX      0       0       0       0       0       0
            1       1       1       1       1       1
  MIX       1       2       3       4       5       5
           11      12      13      14      15      15
  CLUSTER ROD1 ROD2L ROD2R ROD3L ROD3R ROD4L ROD4R
    ::: ROD1  := GEO: TUBE 2 1 2 MIX  6 10 16 20  HMIX 0 0 1 1
                 NPIN  1 RPIN 0.0000 APIN 0.0000
                 MESHX -0.6540 0.6540
                 MESHY -0.6540 0.0 0.6540
                 RADIUS 0.00000 0.6122 0.6540 ;
    ::: ROD2L := GEO: TUBE 2     MIX  7 10 HMIX 0 0
                 RADIUS 0.00000 0.6122 0.6540
                 NPIN  3 RPIN 1.4885
                 APIN 2.09439510 3.14159265 4.18879020 ;
    ::: ROD2R := GEO: TUBE 2     MIX 17 20 HMIX 1 1
                 RADIUS 0.00000 0.6122 0.6540
                 NPIN  3 RPIN 1.4885
                 APIN -1.04719755 0.0000 1.04719755 ;
    ::: ROD3L := GEO: TUBE 2     MIX  8 10 HMIX 0 0
                 RADIUS 0.00000 0.6122 0.6540
                 NPIN  6 RPIN 2.8755
                 APIN -2.87979327  -2.35619449  -1.83259571
                       1.83259571   2.35619449   2.87979327 ;
    ::: ROD3R := GEO: TUBE 2     MIX 18 20 HMIX 1 1
                 RADIUS 0.00000 0.6122 0.6540
                 NPIN  6 RPIN 2.8755
```

```
                  APIN -1.30899694  -0.78539816  -0.26179939
                        0.26179939   0.78539816   1.30899694 ;
      ::: ROD4L := GEO: TUBE 2     MIX  9 10 HMIX 0 0
                  RADIUS 0.00000 0.6122 0.6540
                  NPIN  9 RPIN 4.3305
                  APIN  1.74532925   2.09439510   2.44346095
                        2.79252680   3.14159265   3.49065850
                        3.83972435   4.18879020   4.53785606 ;
      ::: ROD4R := GEO: TUBE 2     MIX 19 20 HMIX 1 1
                  RADIUS 0.00000 0.6122 0.6540
                  NPIN  9 RPIN 4.3305
                  APIN -1.39626340  -1.04719755  -0.69813170
                       -0.34906585   0.0          0.34906585
                        0.69813170   1.04719755   1.39626340 ;
  ;
  ::: FXYR := GEO: CARCEL 5 2 1
    MESHX -7.14375 0.0 7.14375   SPLITX 3 3
    MESHY -7.14375 7.14375   SPLITY 6
    RADIUS  0.00000 5.16890 5.60320  6.44780 6.58750  7.00
    HMIX      1       1       1       1       1       1
              0       0       0       0       0       0
    MIX      11      12      13      14      15      15
              1       2       3       4       5       5
    CLUSTER ROD1 ROD2L ROD2R ROD3L ROD3R ROD4L ROD4R
      ::: ROD1  := GEO: TUBE 2 1 2 MIX 16 20  6 10  HMIX 1 1 0 0
                  NPIN  1 RPIN 0.0000 APIN 0.0000
                  MESHX -0.6540 0.6540
                  MESHY -0.6540 0.0 0.6540
                  RADIUS 0.00000 0.6122 0.6540 ;
      ::: ROD2L := GEO: TUBE 2     MIX 17 20 HMIX 1 1
                  RADIUS 0.00000 0.6122 0.6540
                  NPIN  3 RPIN 1.4885
                  APIN 2.09439510 3.14159265 4.18879020 ;
      ::: ROD2R := GEO: TUBE 2     MIX  7 10 HMIX 0 0
                  RADIUS 0.00000 0.6122 0.6540
                  NPIN  3 RPIN 1.4885
                  APIN -1.04719755 0.0000 1.04719755 ;
      ::: ROD3L := GEO: TUBE 2     MIX 18 20 HMIX 1 1
                  RADIUS 0.00000 0.6122 0.6540
                  NPIN  6 RPIN 2.8755
                  APIN -2.87979327  -2.35619449  -1.83259571
                        1.83259571   2.35619449   2.87979327 ;
      ::: ROD3R := GEO: TUBE 2     MIX  8 10 HMIX 0 0
                  RADIUS 0.00000 0.6122 0.6540
                  NPIN  6 RPIN 2.8755
                  APIN -1.30899694  -0.78539816  -0.26179939
                        0.26179939   0.78539816   1.30899694 ;
      ::: ROD4L := GEO: TUBE 2     MIX 19 20 HMIX 1 1
                  RADIUS 0.00000 0.6122 0.6540
                  NPIN  9 RPIN 4.3305
                  APIN  1.74532925   2.09439510   2.44346095
                        2.79252680   3.14159265   3.49065850
```

```
                          3.83972435   4.18879020   4.53785606 ;
       ::: ROD4R := GEO: TUBE 2     MIX  9 10 HMIX 0 0
                    RADIUS 0.00000 0.6122 0.6540
                    NPIN  9 RPIN 4.3305
                    APIN -1.39626340  -1.04719755  -0.69813170
                         -0.34906585   0.0          0.34906585
                          0.69813170   1.04719755   1.39626340 ;
  ;
;
*----
*  Rod and GT absent
*----
TRACK Lines := NXT: SORINS :: EDIT 5 TISO 40 30.0   ;
MicLib := SHI: MicLib TRACK Lines   ;
TRACK Lines := DELETE: TRACK Lines ;
TRACK Lines := NXT: SORIN :: EDIT 5 TISO 40 30.0   ;
FigReg.ps := PSP: TRACK :: TYPE REGI ;
FigMix.ps := PSP: TRACK :: TYPE MIXT ;
FigHom.ps := PSP: TRACK :: TYPE HMIX ;
ASMPIJ := ASM: MicLib TRACK Lines :: PIJ ;
FLUX := FLU: ASMPIJ MicLib TRACK :: TYPE K ;
EDITION := EDI: FLUX MicLib TRACK ::
  EDIT 3
  COND 0.625
  MERG HMIX
  SAVE ON SORINHMIX ;
HomHMix.txt := EDITION ;
EDITION := DELETE: EDITION ;
EDITION := EDI: FLUX MicLib TRACK ::
  EDIT 3
  COND 0.625
  MERG MIX 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
  SAVE ON SORINMIX ;
HomMix.txt := EDITION ;
EDITION := DELETE: EDITION ;
TRACK FLUX ASMPIJ Lines := DELETE: TRACK FLUX ASMPIJ Lines ;
END: ;
QUIT "LIST" .
```

Input data for test case: **TCWU17Lib.c2m**

```
*----
*  Create Library for test CASE TCWU17.x2m
*  Calling :
*      LIBRARY := TCWU17Lib.x2m :: iedit ;
*  with :
*      LIBRARY = Linked list containing the result of LIB: for
*                TCWU17.x2m
*      iprint  = print level for LIB: module
*----
*  Define PARAMETERS,STRUCTURES and MODULES used
*----
```

```
PARAMETER  LIBRARY  ::  ::: LINKED_LIST LIBRARY ; ;
MODULE     LIB: DELETE: END: ;
*----
*  Define and read LIB: EDIT option
INTEGER iedit   ;
:: >>iedit<<    ;
*----
*  Depletion data from file iaea format WIMSD4
*  Microscopic cross sections from file iaea format WIMSD4
*  All materials are duplicated for left and right cell identification
*  during homogenization
*----
LIBRARY := LIB: ::
  EDIT <<iedit>>
  NMIX 21 CTRA WIMS
  DEPL LIB: WIMSD4 FIL: iaea
  MIXS LIB: WIMSD4 FIL: iaea
  MIX  1  560.66   0.81212           O16      = '6016'   7.99449E-1
    D2D2O    = '3002'   1.99768E-1   H1H2O    = '3001'   7.83774E-4
  MIX  2  560.66   6.57              Nb93     = '93'     2.50000
    BNat     = '1011'   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     = '2056'   1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '1011'   3.10000E-4
    Zr91     = '91'     9.97100E+1
  MIX  5  345.66   1.082885          O16      = '6016'   7.98895E-1
    D2D2O    = '3002'   2.01016E-1   H1H2O    = '3001'   8.96000E-5
  MIX  6  941.29  10.4375010         O16      = '6016'   1.18473E+1
    Xe135    = '4135'   0.0
    U235     = '2235'   6.27118E-1 1
    U238     = '8238'   8.75256E+1 1
    U236     = '236'    0.0        1
    Pu239    = '6239'   0.0        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     = '2056'   1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '1011'   3.10000E-4
    Zr91     = '91'     9.97100E+1
  MIX 11  560.66   0.81212           O16      = '6016'   7.99449E-1
    D2D2O    = '3002'   1.99768E-1   H1H2O    = '3001'   7.83774E-4
  MIX 12  560.66   6.57              Nb93     = '93'     2.50000
    BNat     = '1011'   2.10000E-4
    Zr91     = '91'     9.75000E+1
  MIX 13  345.66   0.0014            He4      = '4'      1.00000E+2
  MIX 14  345.66   6.44              Fe56     = '2056'   1.60000E-1
    Ni58     = '58'     6.00000E-2   Cr52     = '52'     1.10000E-1
    BNat     = '1011'   3.10000E-4
    Zr91     = '91'     9.97100E+1
```

```
    MIX 15  345.66   1.082885          O16      = ’6016’   7.98895E-1
      D2D2O    = ’3002’   2.01016E-1   H1H2O    = ’3001’   8.96000E-5
    MIX 16  941.29  10.4375010         O16      = ’6016’   1.18473E+1
      Xe135    = ’4135’   0.0
      U235     = ’2235’   6.27118E-1 1
      U238     = ’8238’   8.75256E+1 1
      U236     = ’236’    0.0        1
      Pu239    = ’6239’   0.0        1
    MIX 17  COMB 6 1.0
    MIX 18  COMB 6 1.0
    MIX 19  COMB 6 1.0
    MIX 20   560.66   6.44            Fe56     = ’2056’   1.60000E-1
      Ni58     = ’58’     6.00000E-2   Cr52     = ’52’     1.10000E-1
      BNat     = ’1011’   3.10000E-4
      Zr91     = ’91’     9.97100E+1
    MIX 21    345.66                   Fe56     = ’2056’   6.19027E-2
      Cr52     = ’52’     1.56659E-3   Ni58     = ’58’     6.83337E-3
      Si29     = ’29’     7.79072E-4   C12      = ’2012’   1.46552E-4
      Mn55     = ’55’     1.25431E-3
    ;
  END: ;
  QUIT "LIST" .
```

References

[1] G. Marleau, A. Hébert, and R. Roy, "New Computational Methods Used in the Lattice Code DRAGON," *Topical Meeting on Advances in Reactor Physics*, p. 1.177, Charleston, South Carolina (1992).

[2] G. Marleau, R. Roy, and A. Hébert, *DRAGON: A Collision Probability Transport Code for Cell and Supercell Calculations*, Technical Report IGE-157, École Polytechnique de Montréal (1994).

[3] A. Hébert, G. Marleau, and R. Roy, "Application of the Lattice Code DRAGON to CANDU Analysis," *Transaction American Nuclear Society*, **72**, 335 (1995).

[4] A. Hébert, G. Marleau, and R. Roy, *A Description of the Data Structures for DRAGON 3.06*, Technical Report IGE-232 Rev. 5, École Polytechnique de Montréal (2008).

[5] G. Marleau, *The EXCELL Geometries Numbering Scheme in DRAGON*, Technical Report IGE-233, École Polytechnique de Montréal (1997).

[6] G. Marleau, *DRAGON Theory Manual Part 1: Collision Probability Calculations*, Technical Report IGE-236 Rev. 1, École Polytechnique de Montréal (2001).

[7] R. Roy, and A. Hébert, *The GAN Generalized Driver*, Technical Report IGE-158, École Polytechnique de Montréal (2000).

[8] R. Roy, *The CLE-2000 Tool-Box*, Technical Report IGE-163, École Polytechnique de Montréal (1999).

[9] F. E. Drigger, *A Method for Calculating Neutron Absorption and Flux Spectra at Epithermal Energies*, Technical Report AECL-1996, Atomic Energy of Canada Limited (1964).

[10] A. Hoffman, *APOLLO: Code Multigroupe de résolution de l'équation du transport pour les neutrons thermiques et rapides*, Technical Report CEA-N-1610, Commisariat à l'énergie Atomique (1973).

[11] A. Hébert, *Développement de la méthode SPH: Homogénéisation de cellules dans un réseau non uniforme et calcul des paramètres de réflecteur*, Technical Report CEA-N-2209, Commisariat à l'énergie Atomique (1981).

[12] R. Roy, A. Hébert, and G. Marleau, "A Transport Method for Treating Three-Dimensional Lattices of Heterogeneous Cells," *Nuclear Science and Engineering*, **101**, 217–225 (1989).

[13] R. Roy, G. Marleau, J. Tajmouati, and D. Rozon, "Modeling of CANDU Reactivity Control Devices with the Lattice Code DRAGON," *Annals of Nuclear Energy*, **21**, 115–132 (1994).

[14] G. Marleau, *New Geometries Processing in DRAGON: The NXT: Module*, Technical Report IGE-260, École Polytechnique de Montréal (2006).

[15] M. Dahmani, G. Marleau, and E. Varin, "Effect of Burnup on ACR-700 3-D Reactivity Devices Cross Sections," *PHYSOR-2006, American Nuclear Society's Topical Meeting on Reactor Physics*, Vancouver, Canada (2006), (Proceedings available on CD-Rom).

[16] R. Le Tellier, A. Hébert, and G. Marleau, "The Implementation of a 3D Characteristics Solver for the Generation of Incremental Cross Sections for Reactivity Devices in a CANDU Reactor," *PHYSOR-2006, American Nuclear Society's Topical Meeting on Reactor Physics*, Vancouver, Canada (2006), (Proceedings available on CD-Rom).

[17] R. Roy, "Anisotropic Scattering for Integral Transport Codes. Part 1. Slab Assemblies," *Annals of Nuclear Energy*, **17**, 379–388 (1990).

[18] R. Roy, "Anisotropic Scattering for Integral Transport Codes. Part 2. Cyclic Tracking and its Application to XY Lattices," *Annals of Nuclear Energy*, **18**, 511–524 (1991).

[19] R. Roy, G. Marleau, and A. Hébert, "A Cyclic Tracking Procedure for Collision Probability Calculation in 2-D Lattices," *International Topical Meeting Advances in Mathematics, Computations and Reactor Physics*, pp. 2.2.4.1–2.2.4.14, Pittsburgh, PA (1991).

[20] 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, ON (1985).

[21] G. Marleau, and A. Hébert, "Introduction of an Improved Critical Buckling Search in WIMS," *1986 Simulation Symposium on Reactor Dynamics and Plant Control*, Hamilton, ON (1986).

[22] R. Roy, "The Cyclic Characteristics Method," *International Topical Meeting on the Physics of Nuclear Science and Technology*, pp. 407–414, Long Island, NY (1998).

[23] R. Roy, "The Cyclic Characteristics Method with Anisotropic Scattering," *M&C-1999 Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, pp. 27–30, Madrid, Spain (1999), (Proceedings available on CD-Rom).

[24] G. J. Wu, and R. Roy, "A New Characteristics Algorithm for 3D Transport Calculations," *Annals of Nuclear Energy*, **30**, 1–16 (2003).

[25] M. Dahmani, R. Roy, and J. Koclas, "New Computational Methodology for Large 3D Neutron Transport Problems," *PHYSOR-2004 International Meeting on the Physics of Fuel Cycles and Advanced Nuclear Systems*, Chicago, IL (2004), (Proceedings available on CD-Rom).

[26] M. Dahmani, and R. Roy, "Parallel Solver Based on Three-Dimensional Characteristics Method: Design and Performance Analysis," *Nuclear Science and Engineering*, **150**, 155–169 (2005).

[27] A. Hébert, *DRAGON: Description of the DRAGLIB Format*, Technical Report IGE-117, École Polytechnique de Montréal (1991).

[28] R. E. MacFarlane, *TRANSX-CTR: A Code for Interfacing MATXS Cross-Section Libraries to Nuclear Transport Codes for Fusion System Analysis*, Technical Report LA-9863-MS, Los Alamos National Laboratory (1984).

[29] RSIC Data Library Collection, DLC-117, *MATXS7A - 69 Neutron Group Cross Section Library in MATXS* (1985).

[30] P. Vontobel, and S. Pelloni, "New JEF/EFF Based MATXS-Formatted Nuclear Data Libraries," *Nuclear Science and Engineering*, **101**, 298 (1989).

[31] J. R. Askew, F. J. Fayers, and P. B. Kemshell, "A General Description of the Lattice Code WIMS," *Journal of the British Nuclear Energy Society*, **5**, 564 (1966).

[32] C. J. Taubman, *The WIMS 69–Group Library Tape 166259*, Technical Report AEEW-M1324, United Kingdom Atomic Energy Establishment (1975).

[33] J. J. Kim, J. T. Lee, and H. R. Kim, "Generation and Benchmarking of a 69 Group Cross Section Library for Thermal Reactor Applications," *Journal of the Korean Nuclear Society*, **21**, 245 (1989).

[34] WLUP, *Final Stage of the WIMS-D Library Update Project* (2005).

[35] J. V. Donnelly, *WIMS-CRNL, A User's Manual for the Chalk River Version of WIMS*, Technical Report AECL-8955, Atomic Energy of Canada Limited (1986).

[36] R. E. MacFarlane, *TRANSX–2: A Code for Interfacing MATXS Cross-Sectios Libraries to Nuclear Transport Codes*, Technical Report LA-12312-MS, Los Alamos National Laboratory (1992).

[37] M. T. Sissaoui, and G. Marleau, "Application of the Feedback Model for the History Base Using DRAGON," *19^{th} CNS Simulation Symposium*, Hamilton, ON (1995).

[38] M. T. Sissaoui, G. Marleau, and D. Rozon, "Accounting For Actinide Burnup History in CANDU Reactor simulation," *Advances In Nuclear Fuel Management II*, Myrtle beach, SC (1997).

[39] M. T. Sissaoui, G. Marleau, and D. Rozon, "CANDU Reactor Simulations Using the Feedback Model with Actinide Burnup History," *Nuclear Technology*, **125**, 197 (1999).

[40] 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 (2000), (Proceedings available on CD-Rom).

[41] T. Courau, *Application de la théorie des perturbations généralisées aux calculs de cellules utilisant la méthode des probabilités de collision*, Ph.D. thesis, École Polytechnique de Montréal (2001).

[42] T. Courau, and G. Marleau, "Generalized Perturbation Theory in DRAGON: Application to CANDU Cell Calculations," *22nd Annual Canadian Nuclear Society Conference*, Toronto, ON (2001), (Proceedings available on CD-Rom).

[43] T. Courau, and G. Marleau, "Adjoint and Generalized Adjoint Flux Calculations Using the Collision Probability Technique," *Nuclear Science and Engineering*, **141**, 46–54 (2002).

[44] T. Courau, and G. Marleau, "Perturbation Theory for Lattice Cell Calculations," *Nuclear Science and Engineering*, **143**, 19–32 (2003).

[45] E. Varin, A. Hébert, R. Roy, and J. Koclas, *A User's Guide for DONJON*, Technical Report IGE-208 Rev. 2, École Polytechnique de Montréal (2004).

[46] G. Marleau, and E. Varin, *An History Interface Between DRAGON and DONJON: The HST: module*, Technical Report IGE-259 Rev. 1, École Polytechnique de Montréal (2004).

[47] E. Varin, and G. Marleau, "Transport-Diffusion Coupling for CANDU Reactor Core Follow-up," *SNA'2003, International Conference on Supercomputing in Nuclear Applications*, Paris, France (2003), (Proceedings available on CD-Rom).

[48] E. Varin, and G. Marleau, "A First Attempt at a CANDU-6 Core Tracking Using Coupled DRAGON/DONJON Calculations," *6th International Conference on Simulation Methods in Nuclear Engineering*, Montréal, QC (2004), (Proceedings available on CD-Rom).

[49] E. Varin, and G. Marleau, "Validating the History-based Method in DRAGON/DONJON Using G2 Core Follow-up," *26th Annual Conference of the Canadian Nuclear Society*, Toronto, ON (2005), (Proceedings available on CD-Rom).

[50] MATLAB, *The Language of Technical Computing* (2006).

[51] C. Plamondon, *Vérification des lignes dintégration et illustration des géométries DRAGON*, Technical Report IGE-290, École Polytechnique de Montréal (2006).

[52] A. Bidaud, G. Marleau, and E. Nablat, "Nuclear Data Uncertainty Analysis using the coupling of DRAGON with SUSD3D," *M&C 2009*, Saratoga Springs, NY (2009), (Proceedings available on CD-Rom).

[53] C. Camand, *Utilisation de sources et d'adjoints DRAGON pour les calculs TRIPOLI*, M.Sc. thesis, École Polytechnique de Montréal (2012).

[54] A. Kavenoky, *Calcul et utilisation des probabilités de première collision pour les milieux hétérogènes à une dimension*, Technical Report CEA-N-1077, Commisariat à l'énergie Atomique (1969).

[55] A. Hébert, "A Collision Probability Analysis of the Double-Heterogeneity Problem," *Nuclear Science and Engineering*, **115**, 177 (1993).

[56] G. Marleau, R. Roy, and A. Hébert, "An Integral Transport Method for Treating CANDU and GCR Clusters," *Nuclear Science and Engineering*, **18**, 197–205 (1986).

[57] G. Marleau, R. Roy, and A. Hébert, "Analytic Reductions for Transmission and Leakage Probabilities in Finite Tubes and Hexahedra," *Nuclear Science and Engineering*, **104**, 209–216 (1990).

[58] G. Marleau, M. L. Vergain, A. Hébert, and R. Roy, "Computation of the DP1 Collision Probabilities for Spherical and Cylindrical Geometries," *Annals of Nuclear Energy*, **17**, 119–134 (1990).

[59] 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, PA (1991).

[60] G. Marleau, and A. Hébert, "Analysis of Cluster Geometries Using the DP1 Approximation of the $J_{\pm}$ Technique," *Nuclear Science and Engineering*, **111**, 257–270 (1992).

[61] A. Hébert, "Application of a Dual Variational Formulation to Finite Element Reactor Calculations," *Annals of Nuclear Energy*, **20**, 823 (1993).

[62] 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," *Transaction American Nuclear Society*, **99**, 169 (1998).

[63] M. J. Halsall, *CACTUS: A Characteristics Solution to the Neutron Transport Equation in Complicated Geometries*, Technical Report AEEW-R 1291, United Kingdom Atomic Energy Establishment (1980).

[64] B. G. Carlson, *Tables of Equal Weight Quadrature EQ_n Over the Unit Sphere*, Technical Report LA-4734, Los Alamos Scientific Laboratory (1971).

[65] G. Longoni, and A. Haghighat, "Development of New Quadrature Sets with the Ordinate Splitting Technique," *M&C-2001, American Nuclear Society Topical Meeting in Mathematics and Computations*, Salt Lake City, Utah (2001), (Proceedings available on CD-Rom).

[66] R. Sanchez, L. Mao, and S. Santandrea, "Treatment of Boundary Conditions in Trajectory-Based Deterministic Transport Methods," *Nuclear Science and Engineering*, **140**, 23–50 (2002).

[67] A. Hébert, and G. Marleau, "Generalization of the Stamm'ler Method for the Self-Shielding of Resonant Isotopes in Arbitrary Geometries," *Nuclear Science and Engineering*, **108**, 230 (1991).

[68] A. Hébert, "Revisiting the Stamm'ler Self-Shielding Method," *25th Annual Canadian Nuclear Society Conference*, Toronto (2004), (Proceedings available on CD-Rom).

[69] R. Roy, and G. Marleau, "Normalization Techniques for Collision Probability Matrices," *PHYSOR-1990*, pp. IX.40–IX.49, Marseille, France (1990).

[70] E. A. Villarino, R. J. Stamm'ler, A. A. Ferri, and J. J. Casal, "HELIOS: Angularly Dependent Collision Probabilities," *Nuclear Science and Engineering*, **112**, 16–31 (1992).

[71] A. Hébert, "A Consistent Technique for the Pin-by-Pin Homogenization of a Pressurized Water Reactor Assembly," *Nuclear Science and Engineering*, **113**, 227 (1993).

[72] A. Hébert, and A. Mathonnière, "Development of a Third-Generation Superhomogénéisation Method for the Homogenization of a Pressurized Water Reactor Assembly," *Nuclear Science and Engineering*, **115**, 129 (1993).

[73] G. Rimpault, "Algorithmic Features of the ECCO Cell Code for Treating Heterogeneous Fast Reactor Subassemblies," *M&C-1995, International Conference on Mathematics and Computations, Reactor Physics, and Environmental Analyses*, Portland, OR (1995), (Proceedings available on CD-Rom).

[74] 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," *Nuclear Science and Engineering*, **118**, 194 (1994).

[75] I. Petrovic, P. Benoist, and G. Marleau, "A Quasi-Isotropic Reflecting Boundary Condition for the Heterogeneous Leakage Model Tibere," *Nuclear Science and Engineering*, **122**, 151 (1996).

[76] 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, ON (1998).

[77] G. Marleau, "Coherent Anisotropic Scattering Homogenization in the B_1 Leakage Model," *M&C-1999 Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, Madrid, Spain (1999), (Proceedings available on CD-Rom).

[78] A. Hébert, "Development of a Second Generation SPH Technique for the Pin-by-Pin Homogenization of a Pressurized Water Reactor Assembly in Hexagonal Geometry," *Transaction American Nuclear Society*, **71**, 253 (1994).

[79] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The art of scientific computing Second Edition (FORTRAN Version)*, Cambridge University Press, Cambridge, MA (1994).

[80] 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, NY (1998).

[81] G. Marleau, "New Geometric Capabilities of DRAGON," *Nineteenth Annual Conference of the Canadian Nuclear Society*, Toronto, ON (1998).

[82] G. Harrisson, and G. Marleau, "Modeling of a 3-D SCWR unit cell," *32nd Annual conference of the CNS*, Niagara Falls, ON (2011), (Proceedings available on CD-Rom).

[83] Kohler, K. E., *PostScript for Technical Drawings PSPLOT: A FORTRAN-Callable POSTSCRIPT Plotting Library User's Manual* (2000).

[84] Theisen, T. O., *Ghostview: An X11 user interface for Ghostscript* (2000), this program is free software under the term of the GNU general public licence as published by the Free Software Fundation.

[85] I. Kodeli, "Multidimensional Deterministic Nuclear Data Sensitivity and Uncertainty Code System: Method and Application," *Nuclear Science and Engineering*, **138**, 45–66 (2001).

[86] v. Stankovski, "Refinement of the Substructure Method for Integral Transport Calculations," *Nuclear Science and Engineering*, **92**, 255 (1986).

[87] 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," *Annals of Nuclear Energy*, **16**, 1 (1989).

[88] R. D. Mosteller, L. D. Eisenhart, R. C. Little, W. J. Eich, and J. Chao, "Benchmark Calculations for the Doppler Coefficient of Reactivity," *Nuclear Science and Engineering*, **107**, 265–271 (1991).

FIGURES

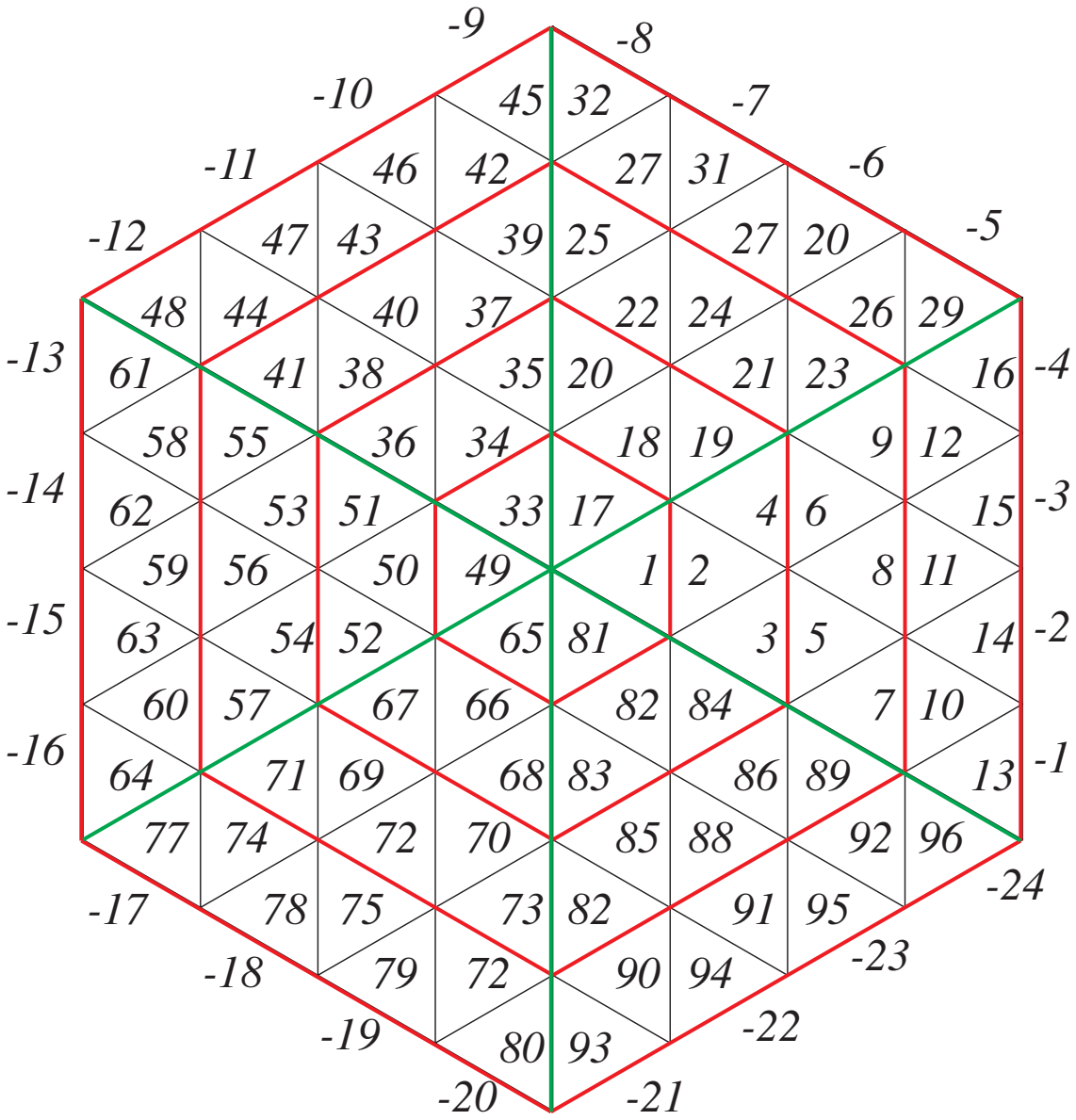

Figure 1: Hexagonal geometry with triangular mesh containing 4 concentric hexagon

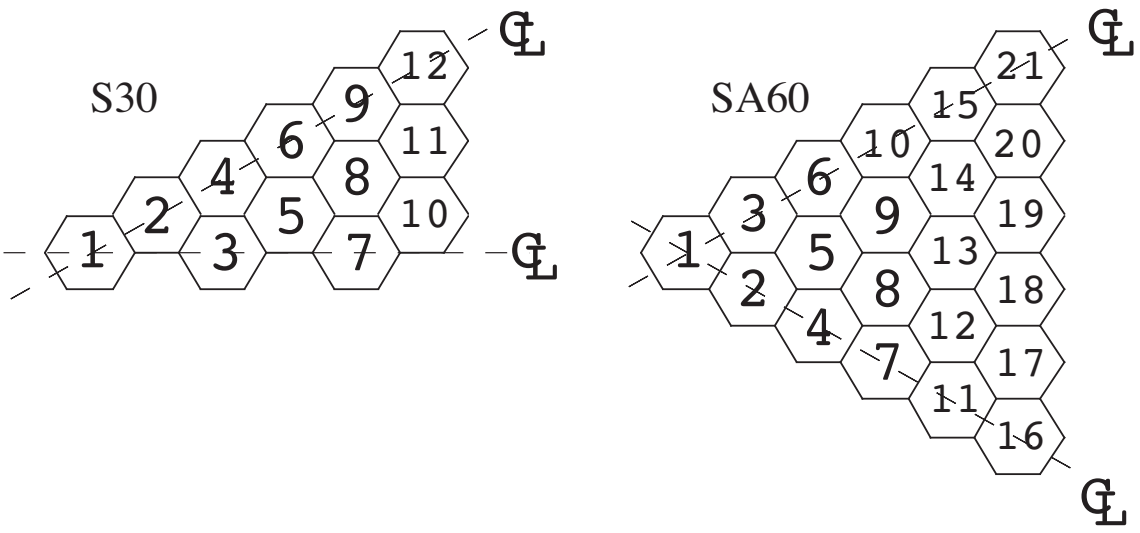

Figure 2: Hexagonal geometries of type S30 and SA60

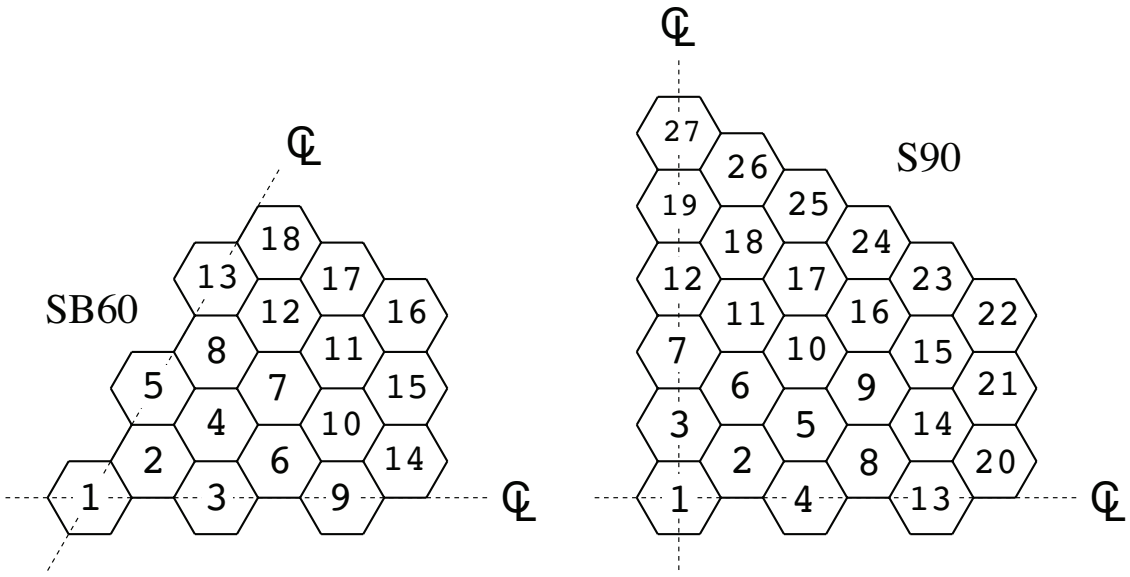

Figure 3: Hexagonal geometries of type SB60 and S90

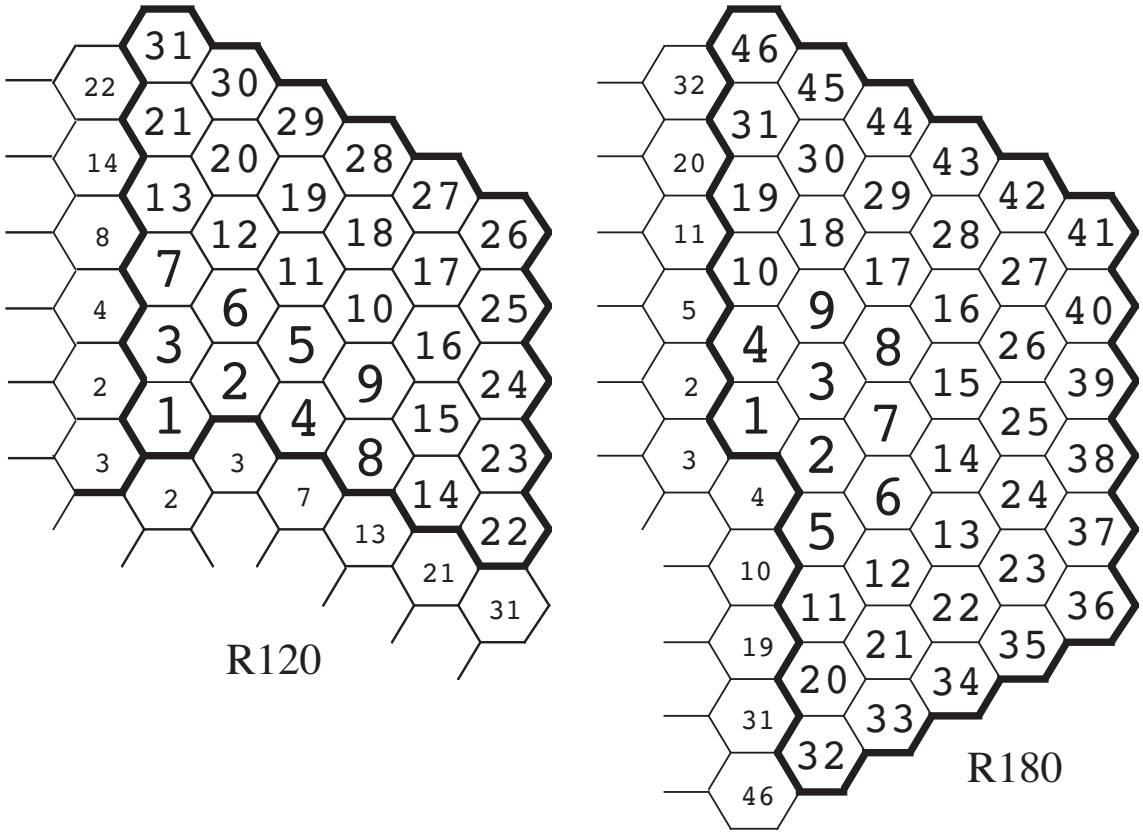

Figure 4: Hexagonal geometries of type R120 and R180

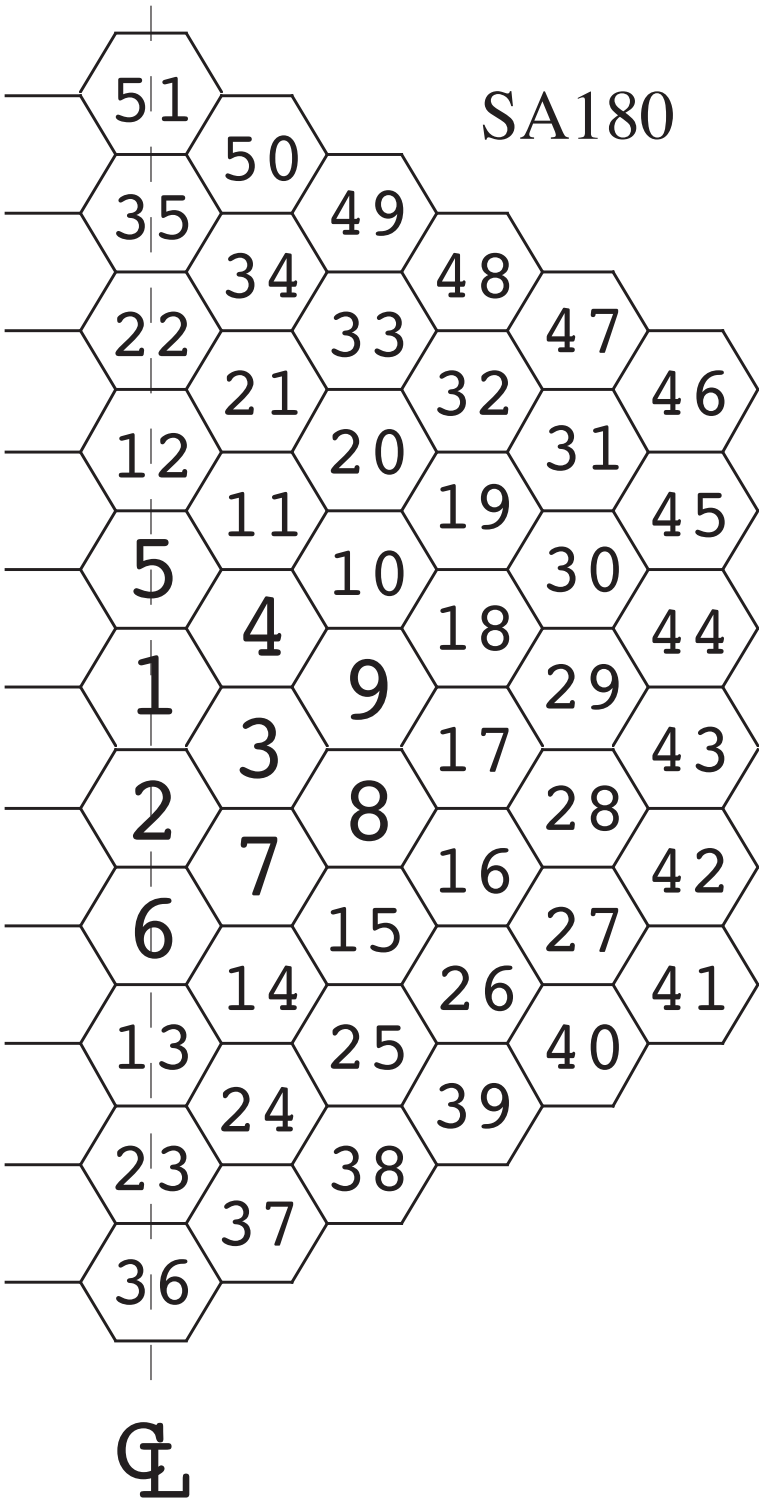

Figure 5: Hexagonal geometry of type SA180

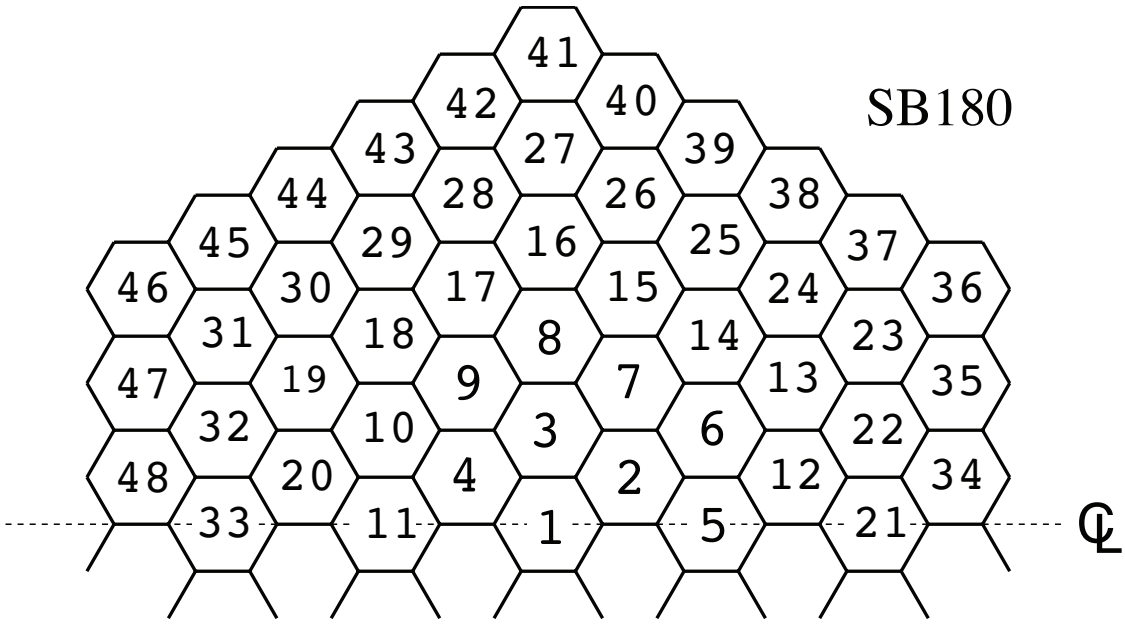

Figure 6: Hexagonal geometry of type SB180

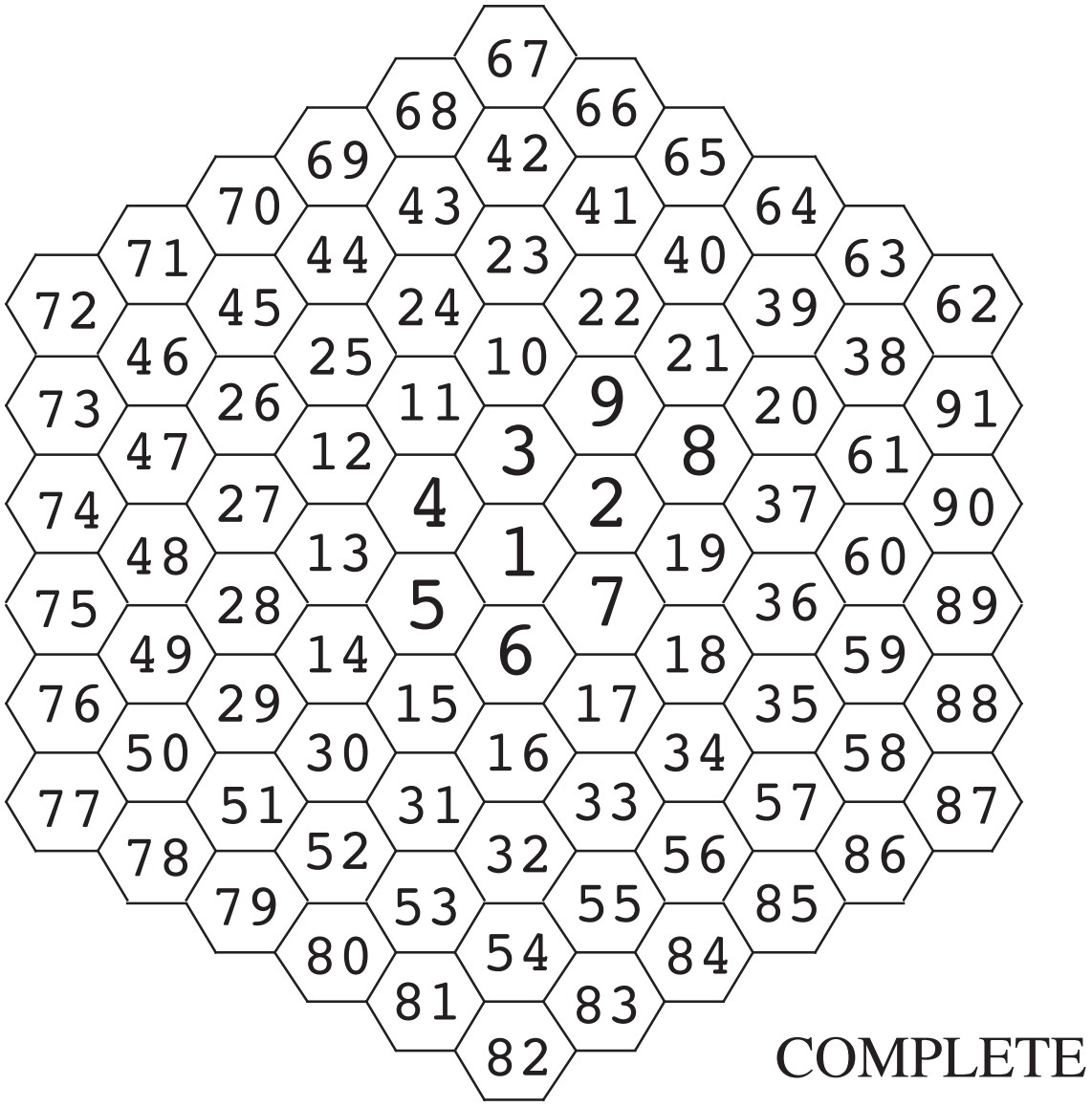

Figure 7: Hexagonal geometry of type COMPLETE

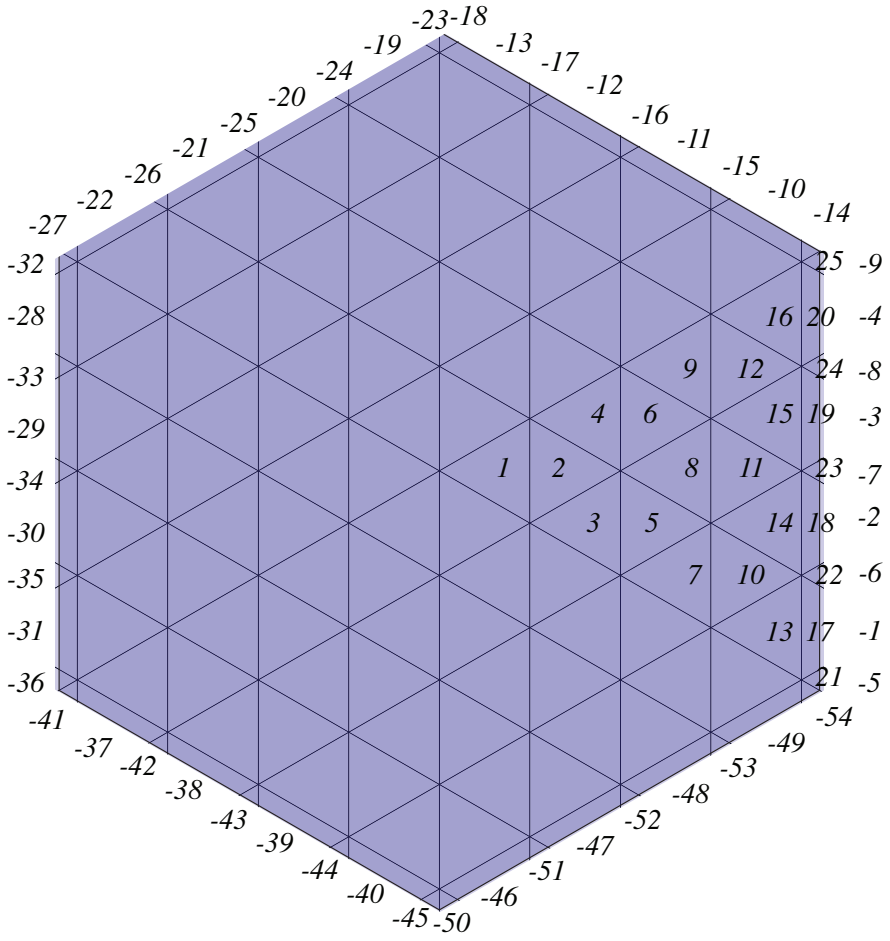

Figure 8: Hexagonal geometry with triangular mesh that extends past the hexagonal boundary

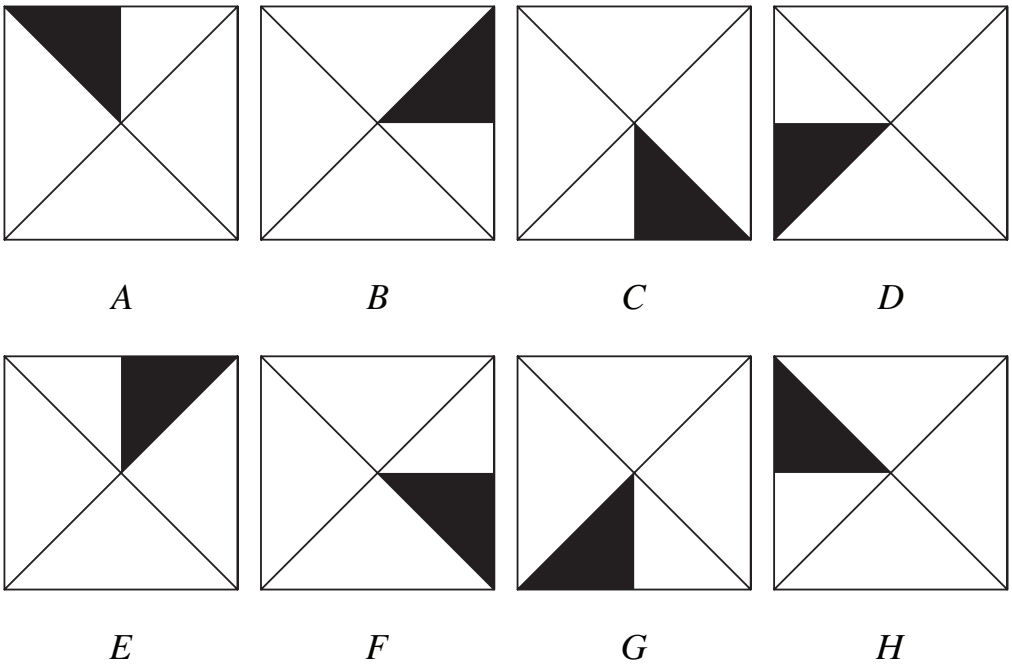

Figure 9: Description of the various rotations allowed for Cartesian geometries

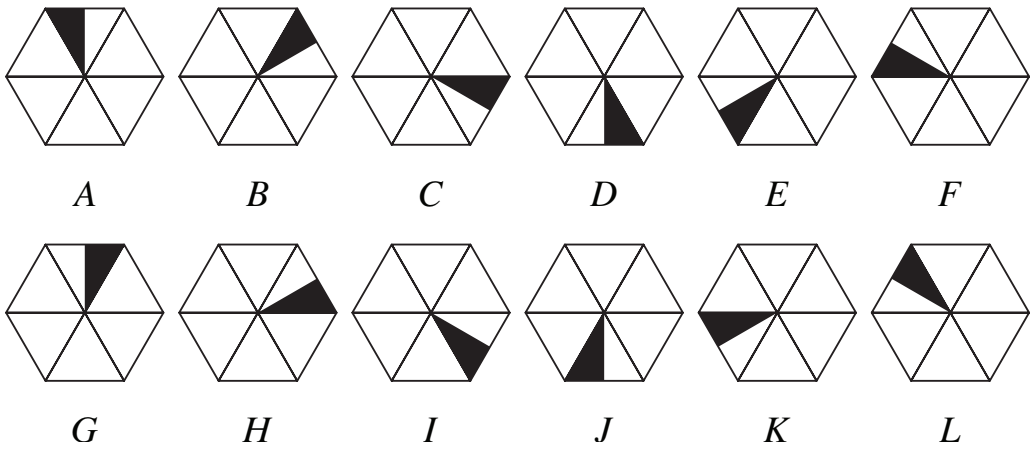

Figure 10: Description of the various rotation allowed for hexagonal geometries

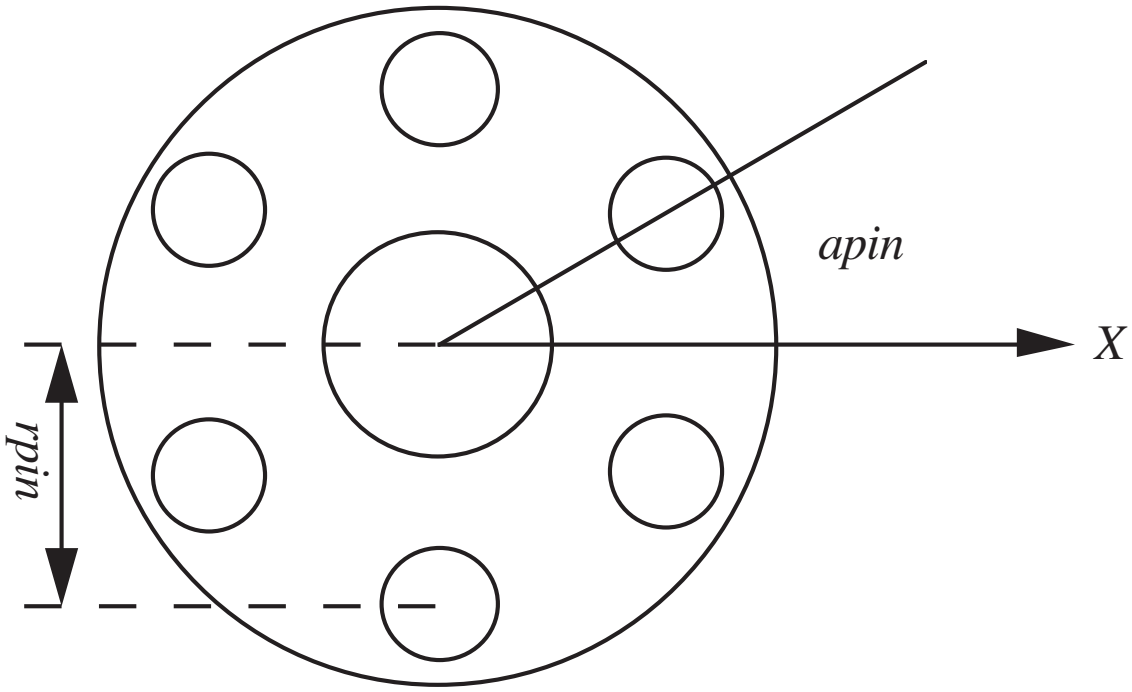

Figure 11: Typical cluster geometry

Figure 12: Slab geometry with mesh splitting

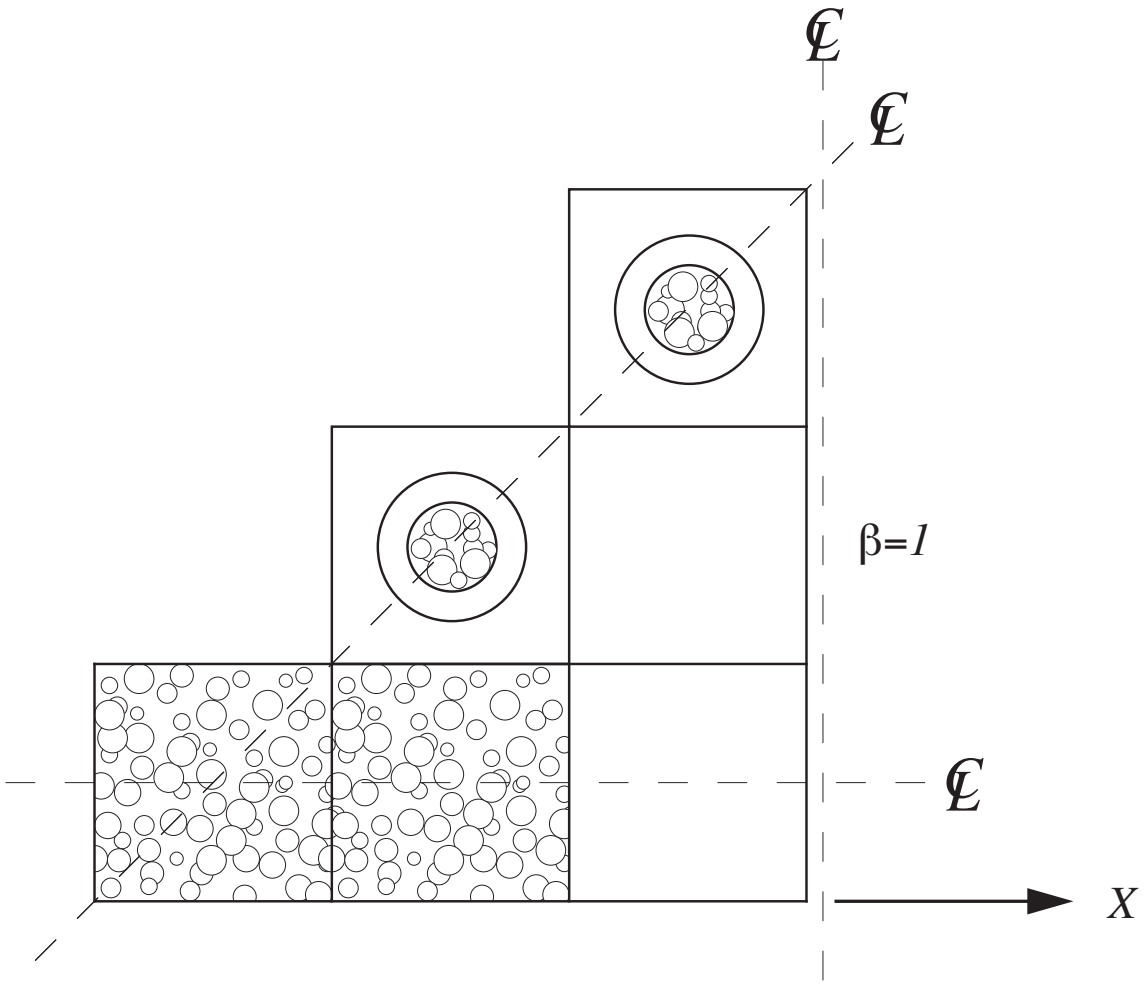

Figure 13: Two dimensional Cartesian assembly containing micro structures

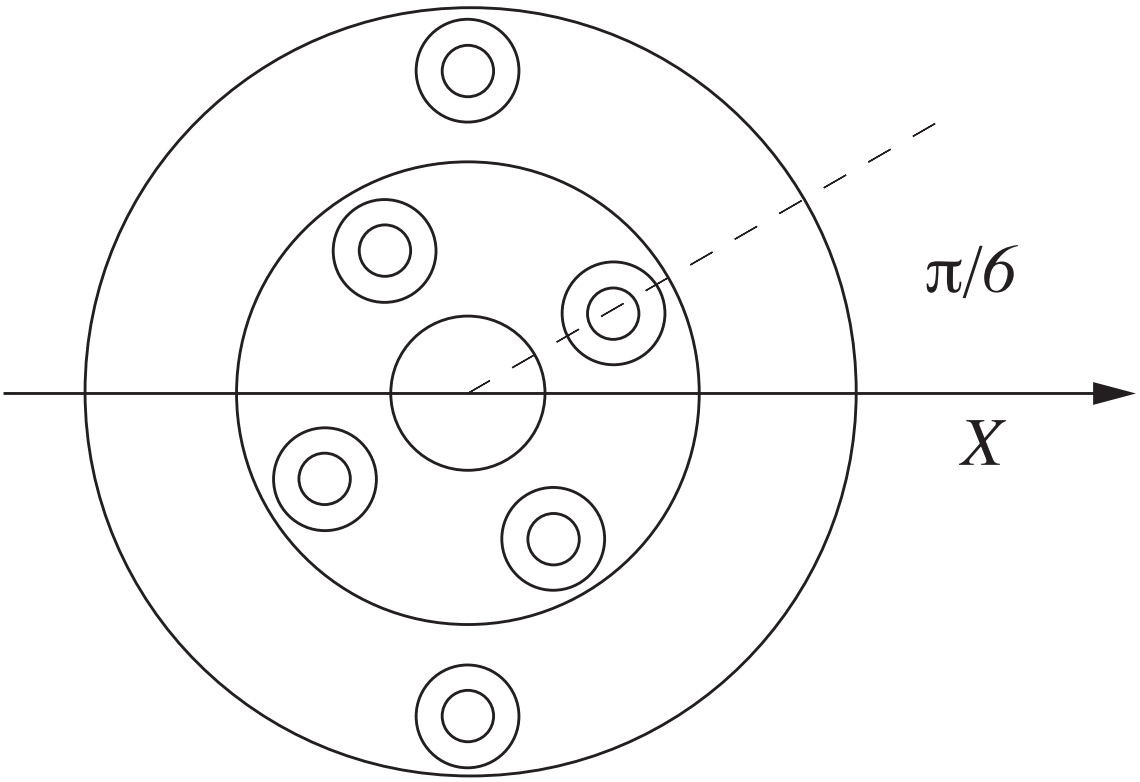

Figure 14: Cylindrical cluster geometry

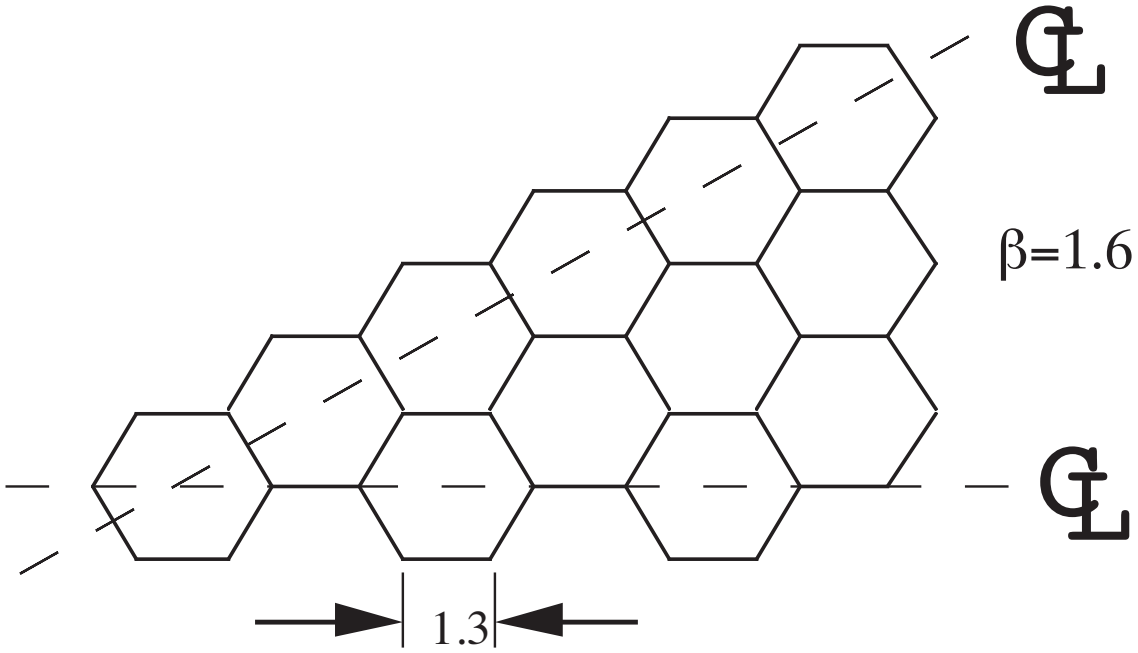

Figure 15: Two dimensional hexagonal geometry

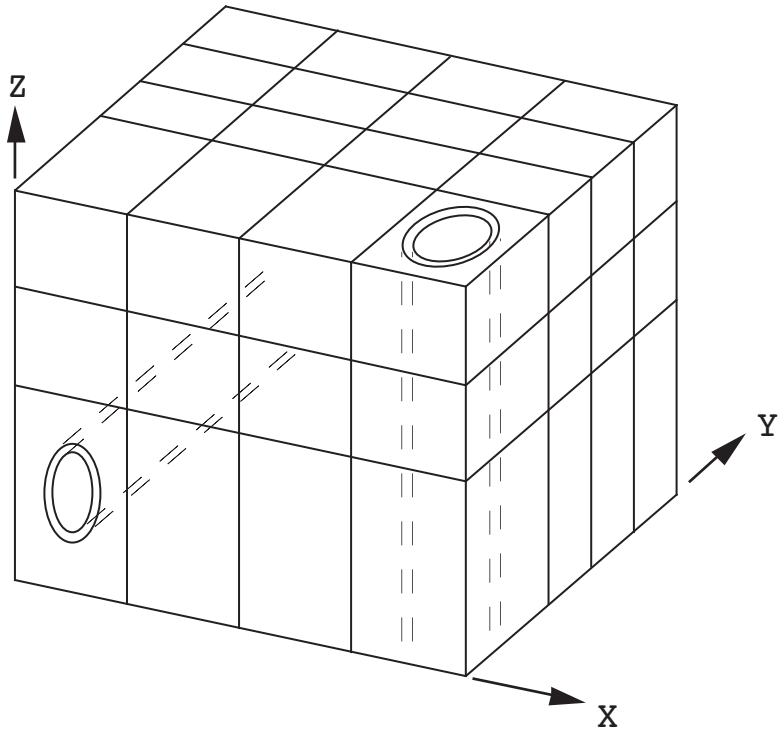

Figure 16: Three dimensional Cartesian supercell

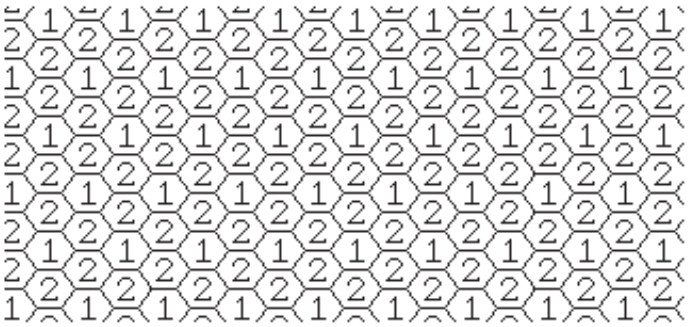

Figure 17: Hexagonal multicell lattice geometry

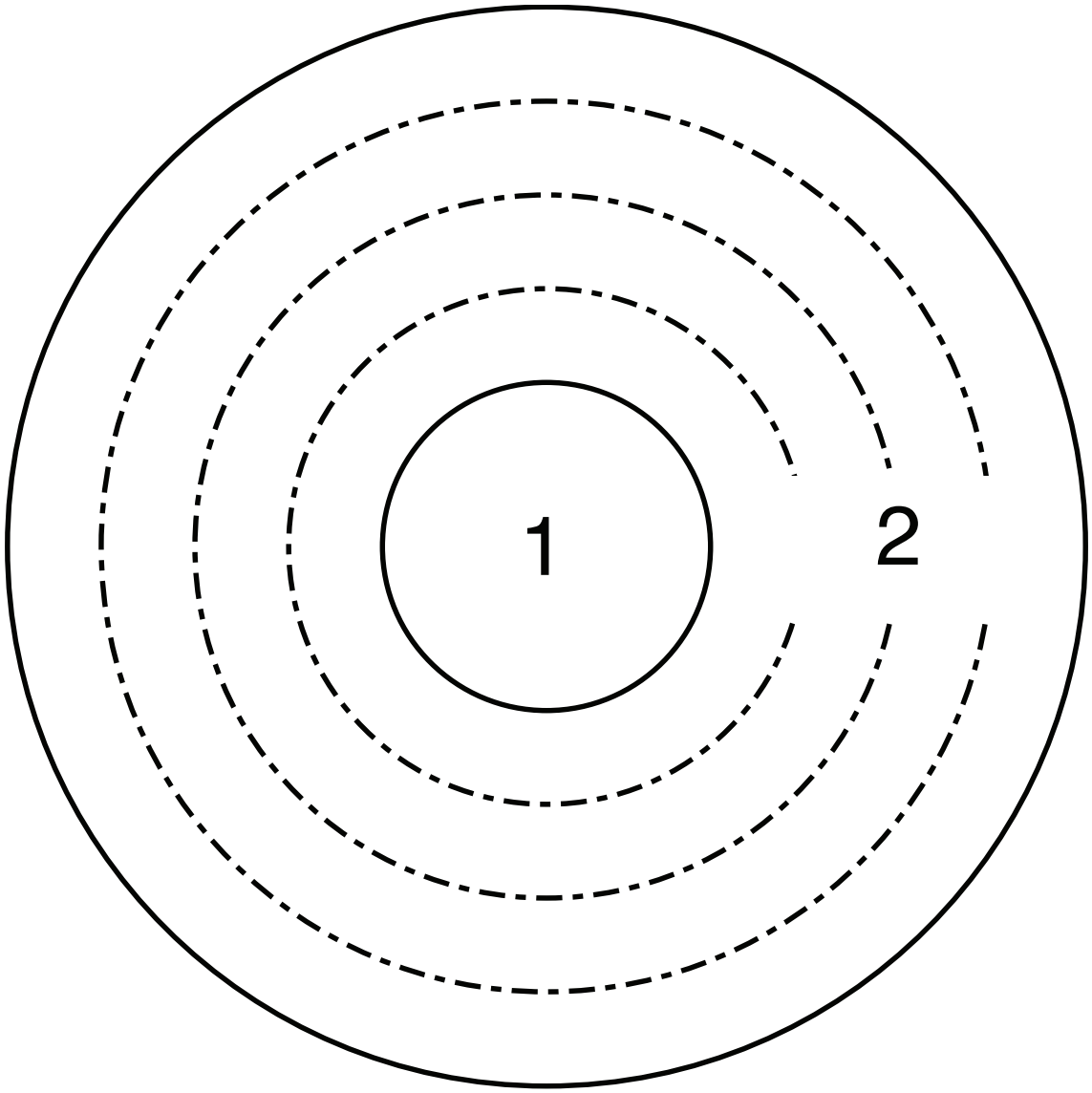

Figure 18: Geometry for test case **TCM01** for an annular cell with macroscopic cross sections.

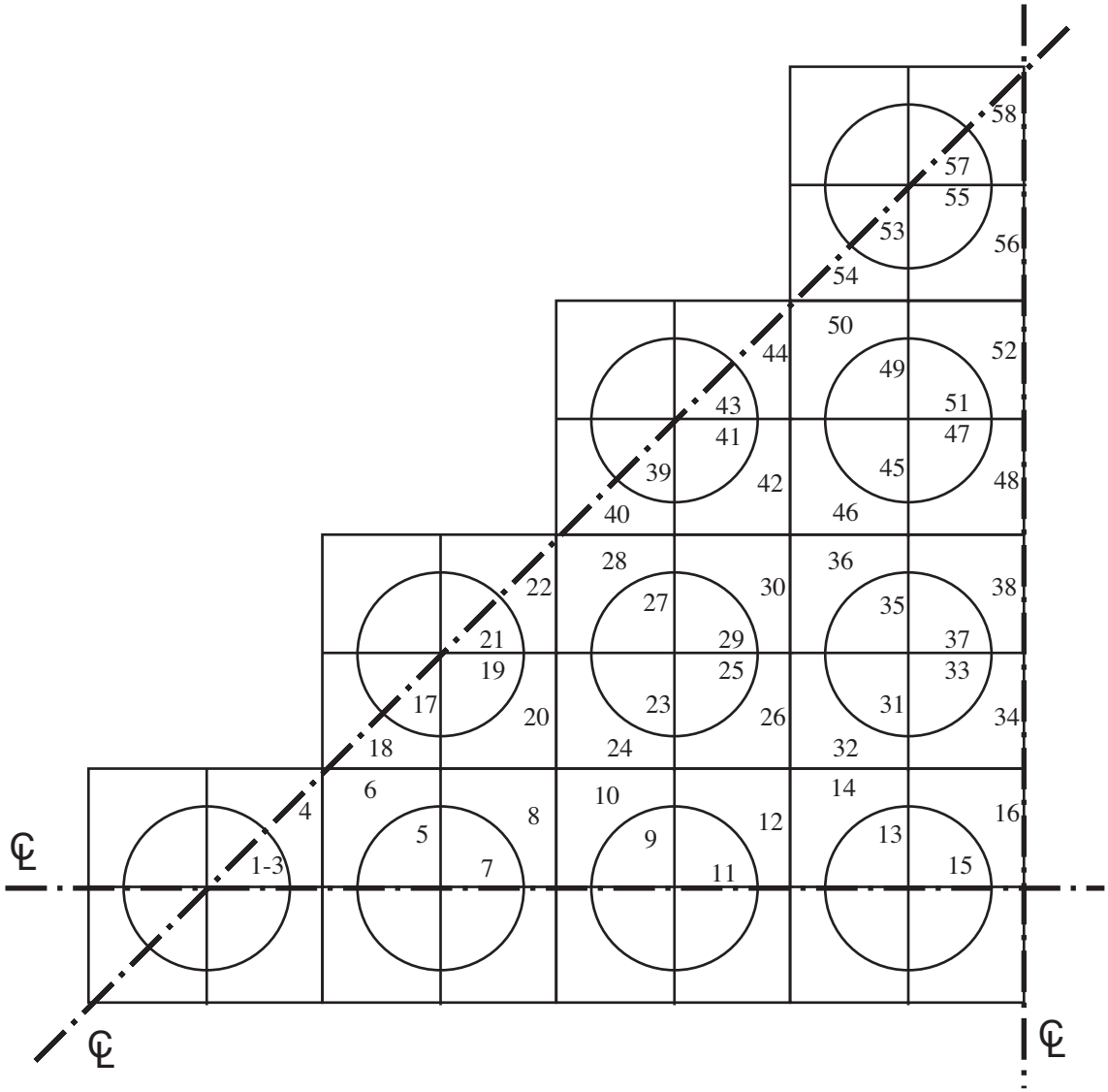

Figure 19: Geometry for test case **TCM02**.

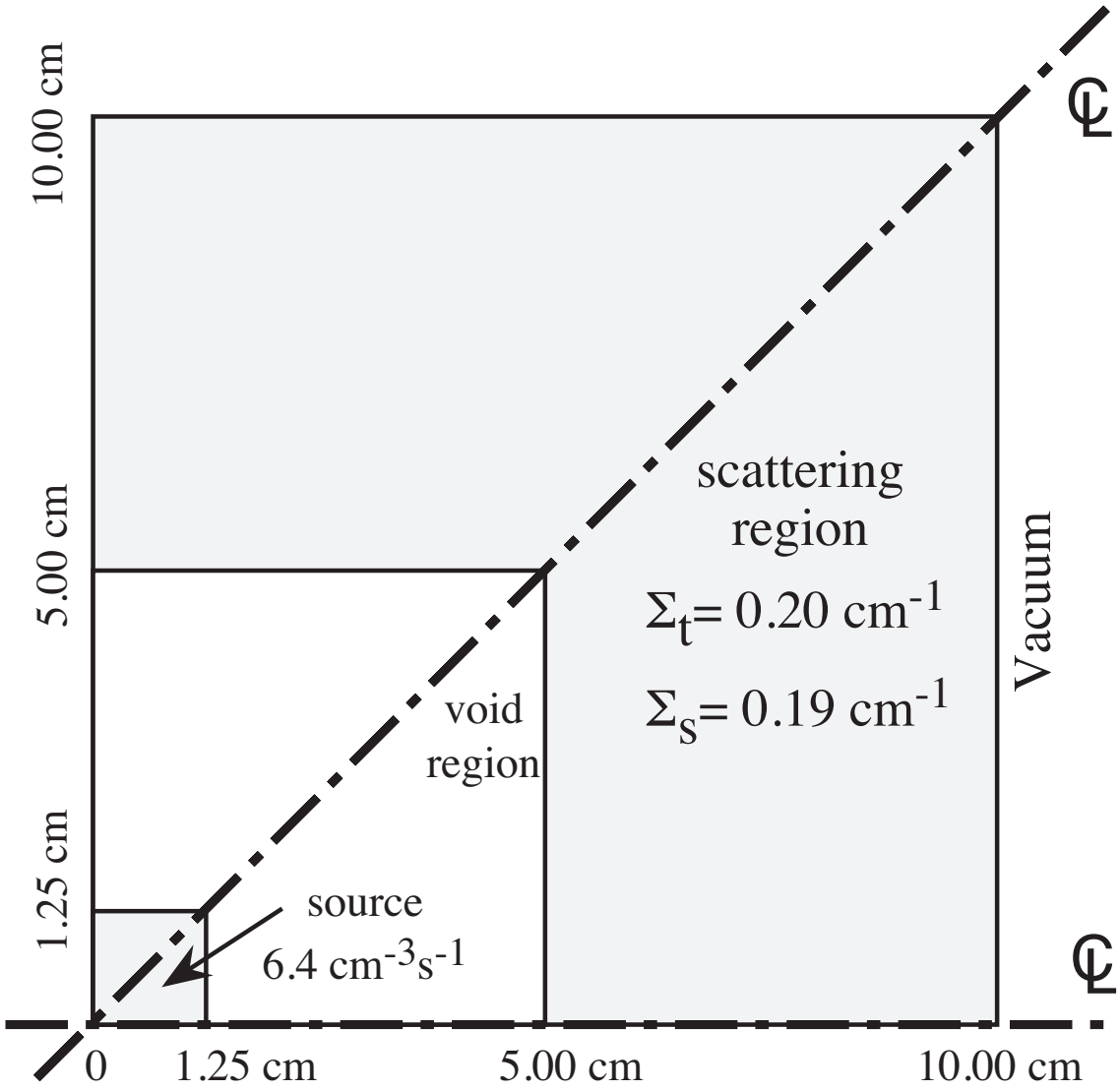

Figure 20: Geometry for test case **TCM03**.

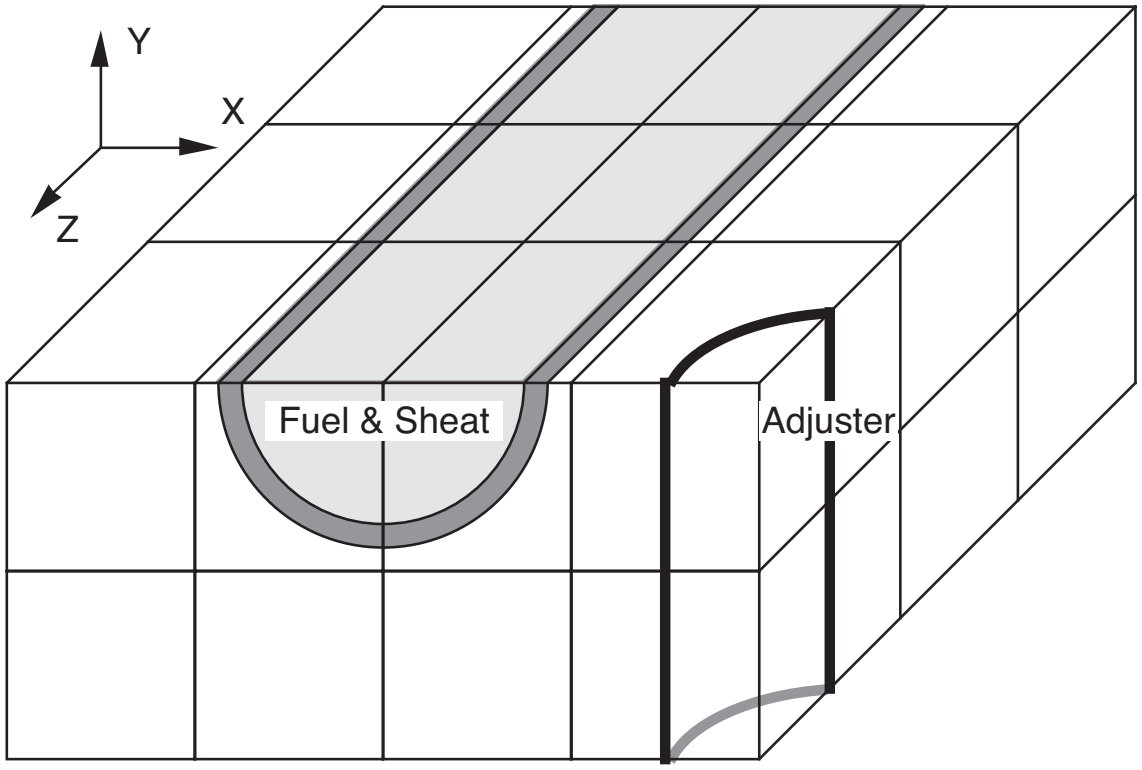

Figure 21: Geometry of the CANDU–6 supercell with stainless steel adjuster rods.

Colored by Region

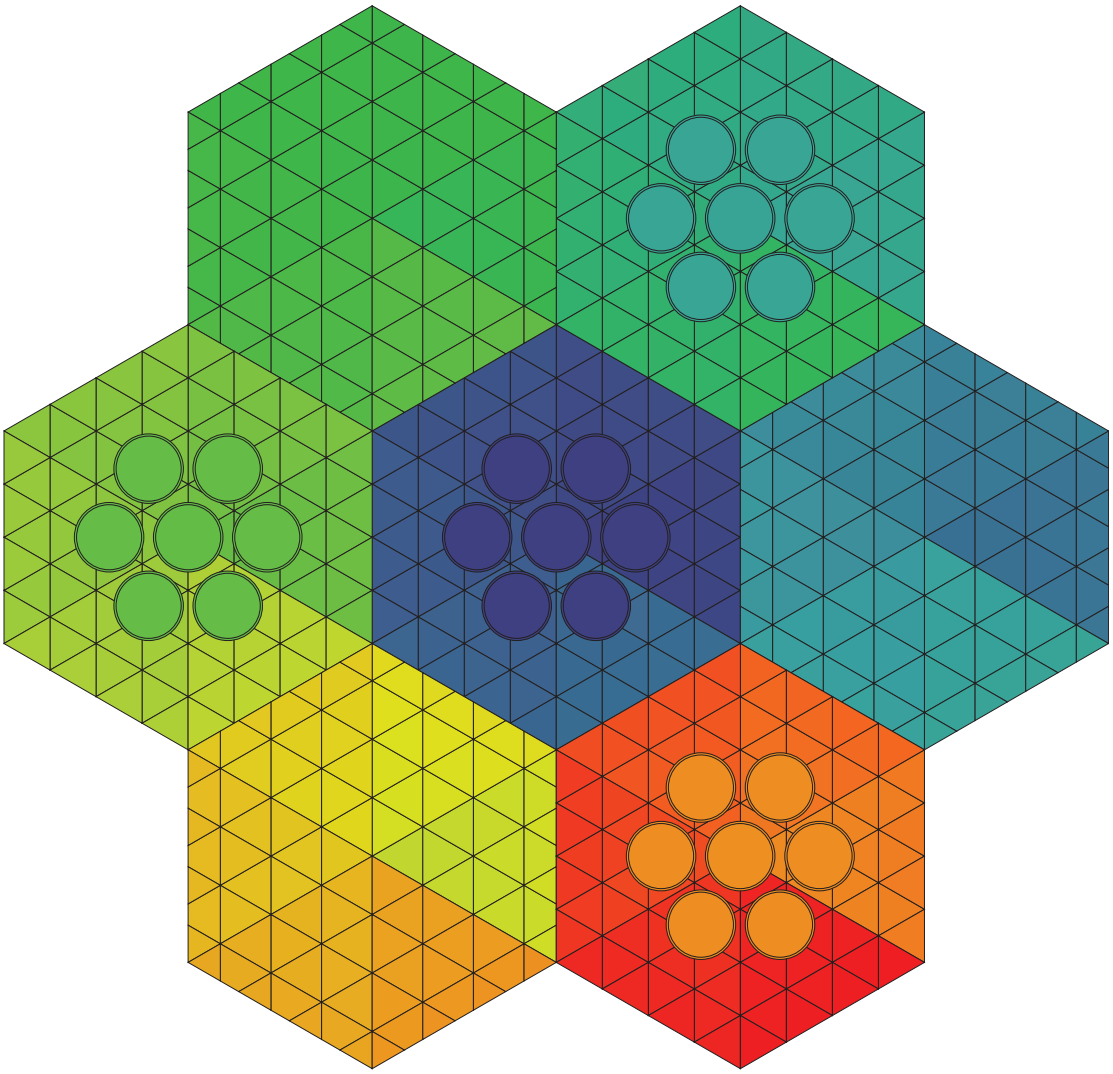

Figure 22: Geometry of the CANDU–6 supercell with stainless steel adjuster rods.

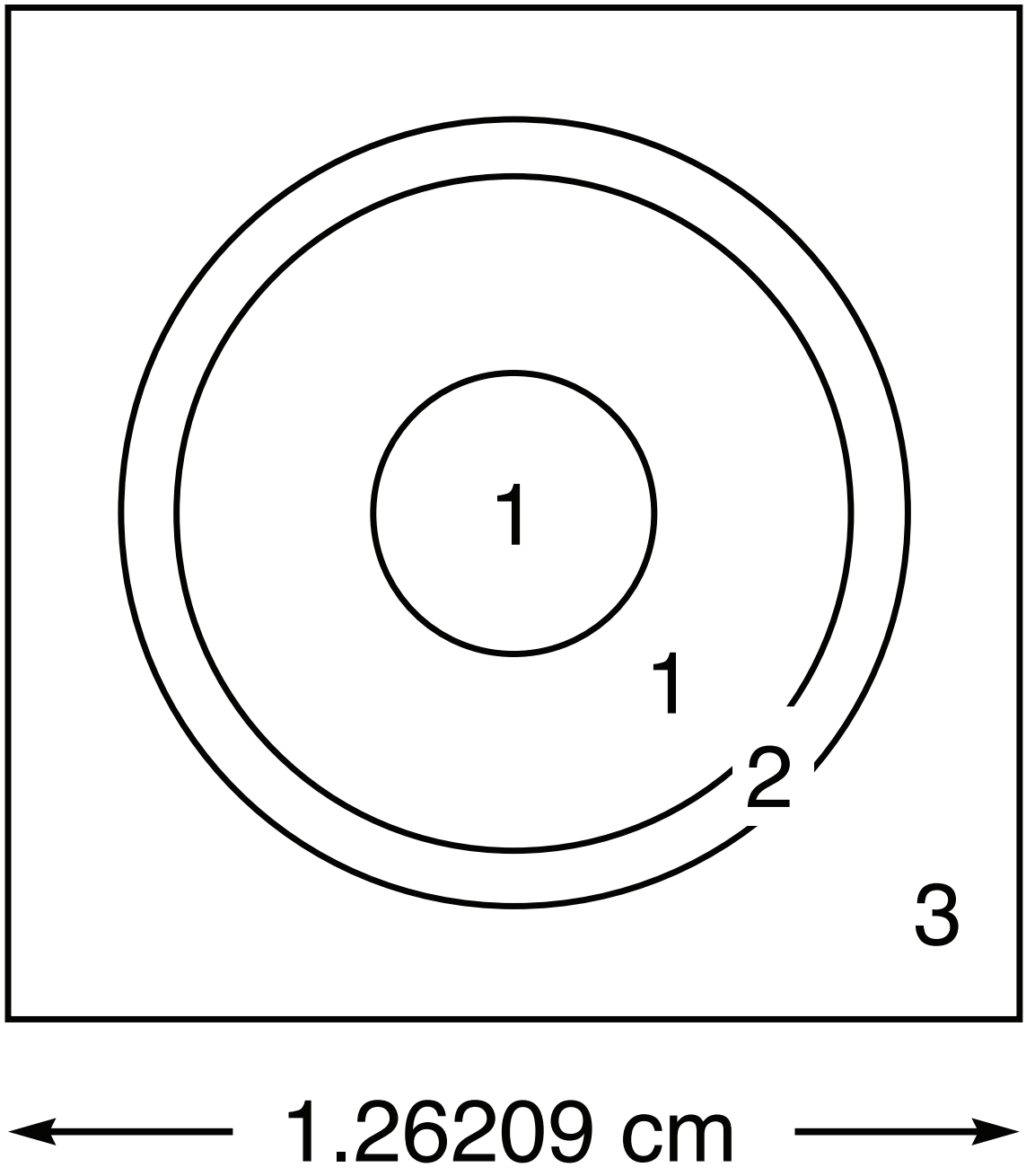

Figure 23: Geometry for the Mosteller benchmark problem used for **TCWU01**.

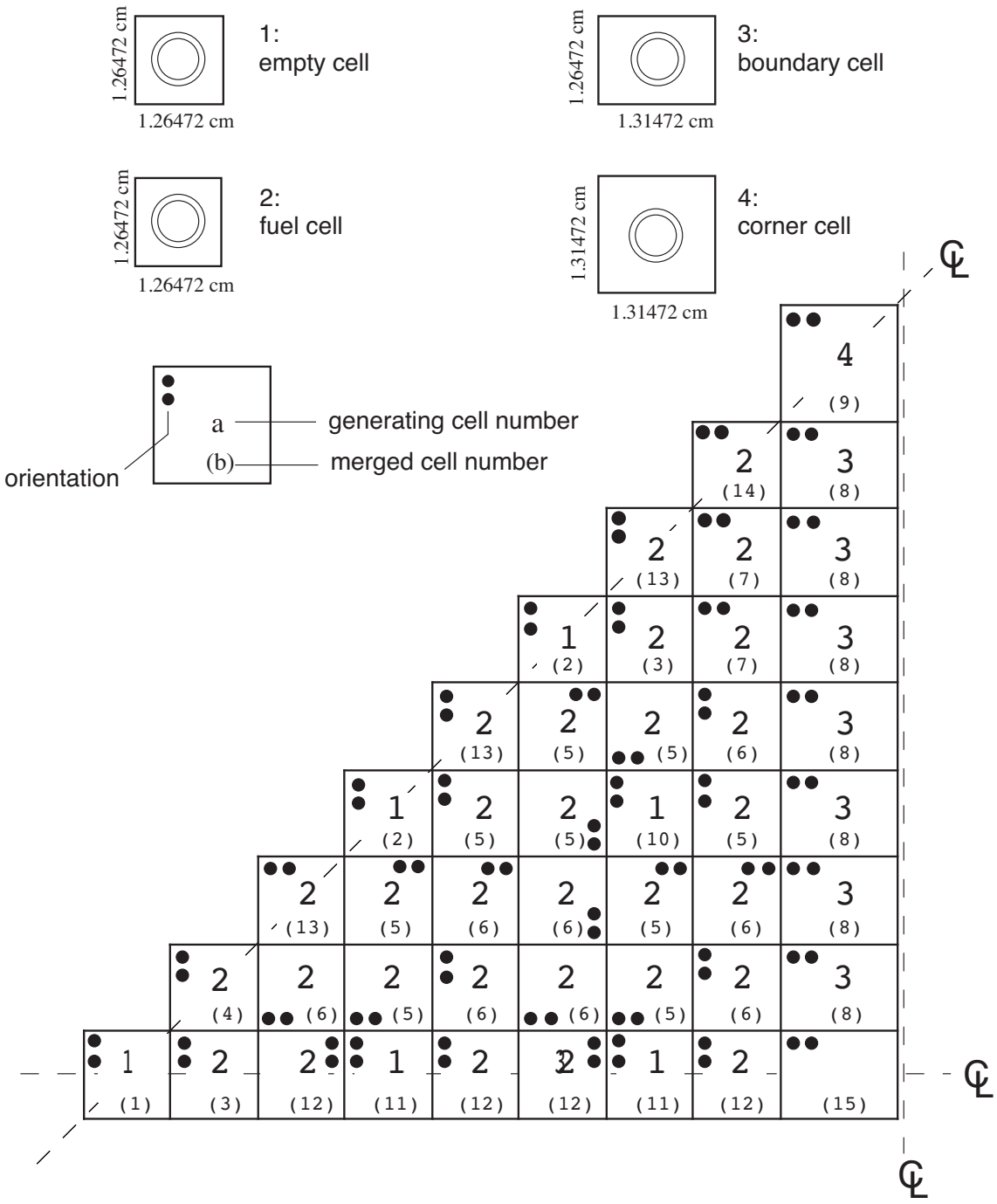

Figure 24: Geometry for test case **TCWU02**.

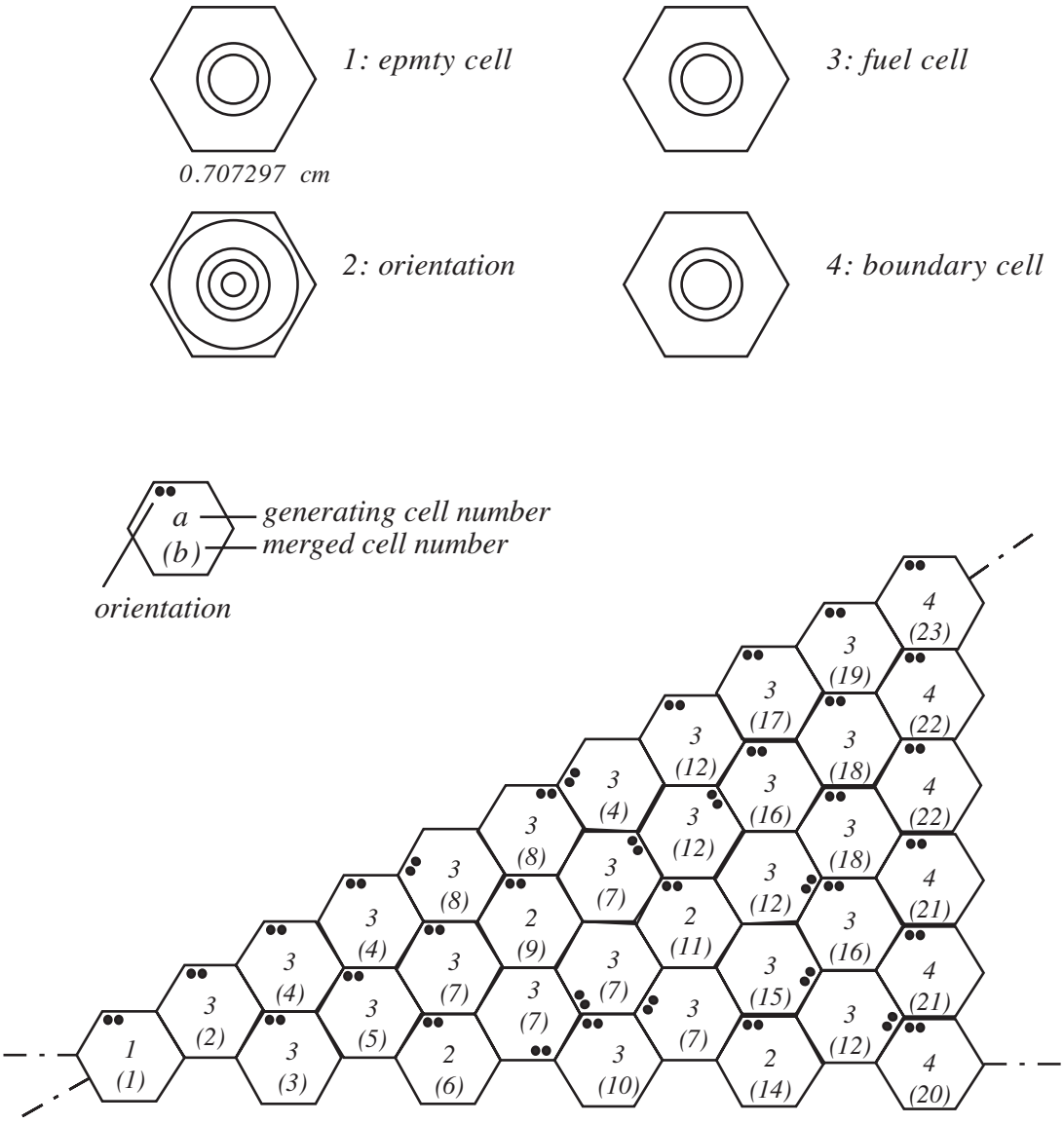

Figure 25: Geometry for test case **TCWU03**.

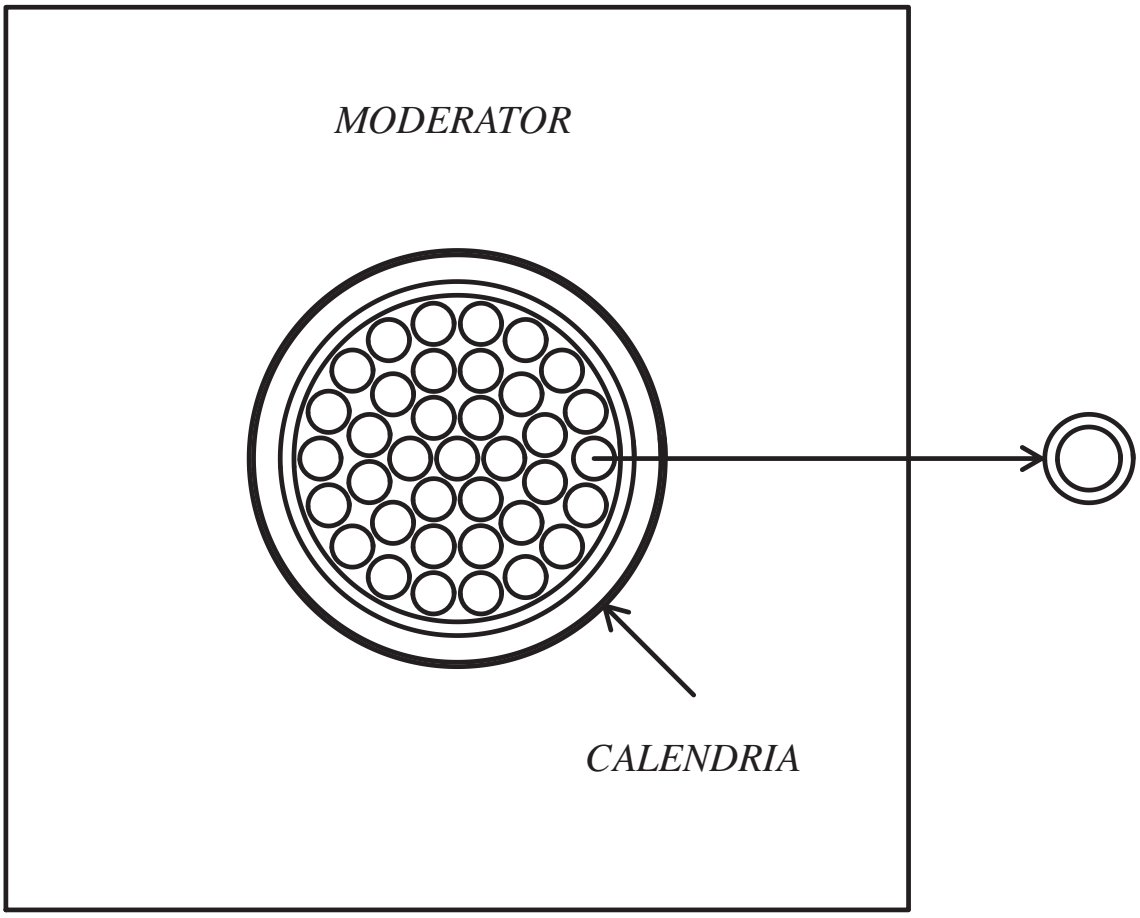

Figure 26: Geometry of the CANDU–6 cell.

Colored by Region

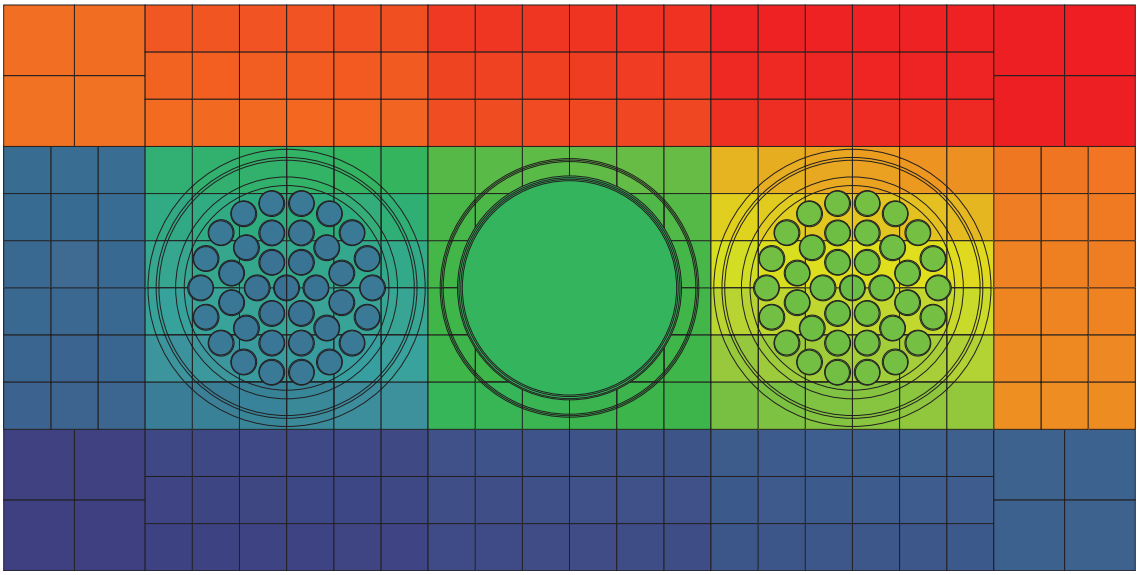

Figure 27: Geometry of 2-D CANDU–6 supercell with control rods.

APPENDICES

Appendix A

Contents of miscellaneous DRAGON files

A.1 Directional fluxes and adjoints output files

The DRAGON directional fluxes and adjoints file has the following format.

```
*----------------------------------------------------------------
*  Fortran instructions to write directional flux and adjoint file
      READ(IPU,1000) NGROUP,NDIM,NANGL,NREG,NFLUX
      READ(IPU,1001) (NAMFLX(IF),IF=1,NFLUX)
      READ(IPU,1002) (WGHT(IQUA),IQUA=1,NANGL)
      READ(IPU,1002) (MU(IQUA),IQUA=1,NANGL)
      READ(IPU,1002) (ETA(IQUA),IQUA=1,NANGL)
      READ(IPU,1002) (VOLUME(IR),IR=1,NREG)
      DO IGROUP=1,NGROUP
        READ(IPU,1002) ((FLUX(IR,IA),IR=1,NREG),IA=1,NANGL)
      ENDDO
      IF(NFLUX .GT. 1) THEN
        DO IGROUP=1,NGROUP
           READ(IPU,1002) ((ADJOINT(IR,IA),IR=1,NREG),IA=1,NANGL)
        ENDDO
      ENDIF
      DO IF=3,NFLUX
        DO IGROUP=1,NGROUP
          READ(IPU,1002) ((GAMMA(IR,IA,IF-2),IR=1,NREG),IA=1,NANGL)
         ENDDO
      ENDDO
 1000 FORMAT(5I10)
 1001 FORMAT(5(A12,2X))
 1002 FORMAT(5E20.10)
*----------------------------------------------------------------
```

Here

IPU	fortran output unit number.
NGROUP	number of energy groups.
NANGL	number of angles N_A at which the angular flux is evaluated.
NREG	number of regions N_R.
NAMFLX	number of flux record. In the case where NFLUX=1, only the directional fluxes are provided. If NFLUX=2, both the directional fluxes and adjoints are provided. Finally for NFLUX=2, the directional fluxes, adjoints as well as generalized adjoints are provided. The type of the generalized adjoints is provided in the record NAME_k.
NAME	name NAME_k of the fluxes record. By definition • NAME_1=FLUX␣␣␣␣␣␣␣␣ • NAME_2=ADJOINT␣␣␣␣␣ • NAME_k for $k > 2$ depends on the generalized adjoint source.

MU	projection of direction on x axis $\mu_\alpha = \cos\varphi_\alpha \sin\theta_\alpha$ with φ_α the azimuthal angle and θ_α the polar angle.
ETA	projection of direction on y axis $\eta_\alpha = \sin\varphi_\alpha \sin\theta_\alpha$ with φ_α the azimuthal angle and θ_α the polar angle.
WGHT	angular weights w_α.
FLUX	directional fluxes $\phi_{i,\alpha}$.
ADJOINT	directional adjoints $\phi^*_{i,\alpha}$.
GAMMA	directional generalized adjoints $\Gamma^*_{l,i,\alpha}$ for type $l = k - 2$.

The scalar fluxes, adjoints and generalized adjoints are then given by

$$\phi_i = \sum_{\alpha=1}^{N_A} w_\alpha \phi_{i,\alpha}$$
$$\phi^*_i = \sum_{\alpha=1}^{N_A} w_\alpha \phi^*_{i,\alpha}$$
$$\Gamma^*_{l,i} = \sum_{\alpha=1}^{N_A} w_\alpha \Gamma^*_{l,i,\alpha}$$

Index