

## A USER GUIDE FOR DRAGON VERSION5

G. MARLEAU, A. HÉBERT AND R. ROY

## Copyright Notice for DRAGON

The development of DRAGON is financially supported, 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). The code DRAGON and its users guide are and will remain the property of École Polytechnique de Montréal. The PostScript utility module used in DRAGON is based on PSLOT which is owned by Kevin E. Kohler at the Nova Southeastern University Oceanographic Center in Florida.

Dragon is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

Permission is granted to the public to copy DRAGON without charge. École Polytechnique de Montréal, makes no warranty, express or implied, and assumes no liability or responsibility for the use of DRAGON.

## Acknowledgments

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

## SUMMARY

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

The code DRAGON contains a multigroup iterator conceived to control a number of different algorithms for the solution of the neutron transport equation. Each of these algorithms is presented in the form of a one-group solution procedure where the contributions from other energy groups are included in a source term. The current version of DRAGON contains many such algorithms. The SYBIL option which solves the integral transport equation using the collision probability method for simple one-dimensional (1-D) geometries (either plane, cylindrical or spherical) and the interface current method for 2-D Cartesian or hexagonal assemblies. The EXCELL option which solves the integral transport equation using the collision probability method for general 2-D geometries and for three-dimensional (3-D) assemblies. The MCCG option solves the integro-differential transport equation using the long characteristics method for general 2-D and 3-D geometries.

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

## Contents

## List of Figures

## List of Tables

## 1 INTRODUCTION

The computer code DRAGON is a lattice code designed around solution techniques of the neutron transport equation.<sup>[?]</sup> The DRAGON project results from an effort made at *École Polytechnique de Montréal* to rationalize and unify into a single code the different models and algorithms used in a lattice code.<sup>[?, ?, ?, ?]</sup> One of the main concerns was to ensure that the structure of the code was such that the development and implementation of new calculation techniques would be facilitated. DRAGON is therefore a lattice cell code which is divided into many calculation modules linked together using the GAN generalized driver<sup>[?, ?]</sup>. These modules exchange informations only via well defined data structures.

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

The SYBIL tracking option emulates the main flux calculation option available in the APOLLO-1 code,<sup>[?, ?]</sup> and includes a new version of the EURYDICE-2 code which performs reactor assembly calculations in both rectangular and hexagonal geometries using the interface current method. The option is activated when the SYBILT: module is called.

The EXCELL tracking option is used to generate the collision probability matrices for the cases having cluster, two-dimensional or three-dimensional mixed rectangular and cylindrical geometries.<sup>[?, ?]</sup> A cyclic tracking option is also available for treating specular boundary conditions in two-dimensional rectangular geometry.<sup>[?, ?]</sup> EXCELL calculations are performed using the EXCELT: or NXT: module.

The MCCG tracking option activates the long characteristics solution technique. This implementation uses the same tracking as EXCELL and perform flux integration using the long characteristics algorithm proposed by Igor Suslov.<sup>[?, ?, ?]</sup> The option is activated when both EXCELT: (or NXT:) and MCCGT: modules are called.

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

DRAGON can access directly standard microscopic cross-section libraries in various formats. It has the capability of exchanging macroscopic cross-section libraries with a code such as TRANSX-CTR or TRANSX-2 by the use of GOXS format files.<sup>[?, ?]</sup> The macroscopic cross section can also be read in DRAGON via the input data stream.



## 2 GENERAL STRUCTURE OF THE DRAGON INPUT

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

### 2.1 Data organization

The structure of the input data is independent of the physical or computational characteristics of the host system. The physical characteristics of the input data is a collection of sequential records. These characters are by necessity ASCII characters. The logical organization of an input deck is in the form of a sequential structure of input variables presented in free format. This structure must be located in the first 72 columns of each record in the input stream. Characters located in column 73 and  $\leq 132$  can be used to identify the records and are treated as comments. An input variable can be defined in one of two ways.

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

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

- characters in column 73 and  $\leq 132$  on each record are considered to be comments;
- all the information following the ';' keyword on a record are not considered by the generalized driver;
- each record starting with the characters '\*' is considered to be commented out;
- all the characters on a given record inserted between '(' and ')' are considered to be commented out.

This users guide was written using the following conventions:

- An input structure represents a set of input variables. It is identified by a name in boldface surrounded by parenthesis. For example, the complete DRAGON input deck is represented by the structure **(DRAGON)**;
- A standard DRAGON data structure represents a set records and directory stored in a hierarchical format on a direct access XSM file or in memory via a linked list.<sup>[?]</sup> It is identified by a name in small capital letter. For example, the data structure ASMPIJ contains the multigroup collision probability matrices generated by the ASM: module of DRAGON;
- The variables presented using the typewriter font are character variables used as keywords. For example **GEO:** is the keyword required to activate the geometry reading module of DRAGON.
- The variables in italics are user defined variables. When indexed and surrounded by parenthesis they denote arrays. If they are in lower case they represent either integer type (starting with *i* to *n*) or real type (starting with *a* to *h* or *o* to *z*) variables. If they are in upper case they represent character type variables. For example, *iprint* must be replaced in the input deck by an integer variable, (*energy*(igroup), igroup=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 generally specified. No character variable can exceed 72 character in length.
- The variables or structures surrounded by single square brackets '['] are optional.

- The variables or structures surrounded by double square brackets ‘[[ ]]’ are also optional. However, they can be repeated as many times as required.
- The variables or structures surrounded by braces and separated by vertical bars ‘{ | }’ represents various calculation options available in DRAGON. Only one of these options is permitted.

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

## 2.2 DRAGON Data Structure and Module Declarations

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

Table 1: Structure (**DRAGON**)

```
[ MODULE [[ MODNAME ]] ; ]
[ LINKED_LIST [[ STRNAME ]] ; ]
[ XSM_FILE [[ STRNAME ]] ; ]
[ SEQ_BINARY [[ STRNAME ]] ; ]
[ SEQ_ASCII [[ STRNAME ]] ; ]
[[ (module) ; ]
END: ;
```

where

<b>MODULE</b>	keyword used to specify the list of modules to be used in this DRAGON execution.
<b>MODNAME</b>	<b>character*12</b> name of a DRAGON or utility module. The list of DRAGON module is provided in Section 2.3 while the list of Utility module is described in Section 2.4. By default a ‘ ’ module is always available (see Sections 2.4 and 5.1).
<b>LINKED_LIST</b>	keyword used to specify which data structures will be stored in linked lists.
<b>XSM_FILE</b>	keyword used to specify which data structures will be stored on XSM format files.
<b>SEQ_BINARY</b>	keyword used to specify which data structures will be stored on sequential binary files.
<b>SEQ_ASCII</b>	keyword used to specify which data structures will be stored on sequential ASCII files.
<b>STRNAME</b>	<b>character*12</b> name of a DRAGON data structure. Note that on MVS file names are truncated to 7 characters due to a constraint of this operating system. The list of DRAGON data structure is presented in Section 2.5.
<b>(module)</b>	input specifications for a DRAGON or utility module. For the DRAGON specific modules these input structures will be defined in Section 3. For utility modules, the required structures are described in Section 5.

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

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

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

The input data normally ends with a call to the **END:** module (see Section 5.13). However, the GAN driver will insert automatically the **END:** module, even if it was not provided, upon reaching an end-of-file in the input stream.

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

### 2.3 The DRAGON Modules

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

**MAC:** module used to generate or modify a DRAGON MACROLIB (see Section 2.5) which contains the group ordered macroscopic cross sections for a series of mixture (see Section 3.1). This MACROLIB can be either an independent data structure or it can be included as a substructure in a MICROLIB. The spatial location of these mixtures will be defined using the **GEO:** module (see Section 3.3).

**LIB:** module used to generate or modify a DRAGON MICROLIB (see Section 2.5) that can read a number of different types of microscopic cross-section libraries (see Section 3.2). Each such access requires a double interpolation (temperature, dilution) carried out by a subroutine specifically tailored to each type of library. Currently the formats DRAGLIB<sup>[?]</sup>, WIMS-D4<sup>[?]</sup>, MATXS<sup>[?]</sup>, WIMS-AECL<sup>[?]</sup>, APOLLO<sup>[?, ?]</sup> and NDAS format<sup>[?]</sup> are supported. After having reconstructed the microscopic cross sections for each isotope, they are then multiplied by the isotopic concentrations (particles per  $cm^3$ ) and combined in such a way as to produce an embedded MACROLIB (see Section 2.5). The spatial location of these mixtures will be defined using the **GEO:** module (see Section 3.3).

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

**SYBILT:** the standard tracking module based on 1D collision probability or Interface Current technique (see Section 3.4 and Section 3.4.1).

**EXCELT:** the standard tracking module for 2D and 3D geometries as well as isolated 2D cells containing clusters (see Section 3.4 and Section 3.4.2).

**NXT:** the standard tracking module for 2D or 3D assemblies of cluster (see Section 3.4 and Section 3.4.3).

**SNT:** the discrete ordinates tracking module (see Section 3.4 and Section 3.4.5).

**MCCGT:** the tracking module of the open characteristics flux solver (see Section 3.4 and Section 3.4.4).

**BIVACT:** the 1D/2D diffusion and  $SP_n$  tracking module (see Section 3.4 and Section 3.4.6).

**TRIVAT:** the 1D/2D/3D diffusion and  $SP_n$  tracking module (see Section 3.4 and Section 3.4.7).

SHI:	module used to perform self-shielding calculations using the generalized Stamm'ler method (see Section 3.5).
USS:	module used to perform self-shielding calculations using a subgroup method (see Section 3.6). A method using physical probability tables (cf. Wims-7 and Helios) and the Ribon extended method are available.
ASM:	module which uses the tracking information to generate a multigroup response or collision probability matrix (see Section 3.7).
FLU:	module which uses inner-iteration approach or collision probability matrix to solve the transport equation for the fluxes (see Section 3.8). Various leakage models are available.
EDI:	editing module (see Section 3.9). An equivalence method based on SPH method is available.
EVO:	burnup module (see Section 3.10).
SPH:	<i>supermomogénéisation</i> (SPH) module (see Section 3.11). The SPH: module can also be used to extract a MICROLIB or MACROLIB from a MULTICOMPO or SAPHYB.
INFO:	utility to compute number densities for selected isotopes in materials such as UO <sub>2</sub> or ThUO <sub>2</sub> (see Section 3.13).
COMPO:	multi-parameter reactor database construction module (see Section 3.14).
TLM:	module used to generate a Matlab M-file to obtain a graphics representation of the NXT: tracking lines (see Section 3.15).
M2T:	interface module for transforming a macrolib into a Trimaran/Tripoli multigroup file (see Section 3.16).
CHAB:	cross section perturbation module similar to CHABINT (see Section 3.17).
CP0:	burnup-dependent mono-parameter reactor database construction module (see Section 3.18).
SAP:	multi-parameter reactor database construction module in SAPHYB format (see Section 3.19).
MC:	multigroup Monte-Carlo flux solution module (see Section 3.20).
T:	macrolib transposition operator (see Section 3.21).
DMAC:	construction module for a Generalized Perturbation Theory (GPT) source (see Section 3.22).
SENS:	sensitivity analysis of keff to nuclear data (see Section 3.24).
PSP:	module to generate PostScript images for 2D geometries that can be tracked using the module EXCELT: or NXT: (see Section 3.27).
DUO:	module to perform a perturbative analysis of two systems using the Clio formula and to determine the origins of Keff discrepancies (see Section 3.25).

## 2.4 The Utility Modules

Because the execution of DRAGON is controlled by the GAN generalized driver it can use directly any one of its utility modules. These modules perform the following tasks:

' '	default module used to make an explicit copy of a data structure (see Section 5.1).
UTL:	module used to manipulate a data structure (see Section 5.2).
DELETE:	module used to delete a data structure (see Section 5.3).
BACKUP:	module used to make a backup copy of a child data structure along with its parent (see Section 5.4).
RECOVER:	module used to recover from a backup copy a child data structure along with its parent (see Section 5.5).
ADD:	module used to add two data structures (see Section 5.6).
MPX:	module used to multiply a data structure by a constant (see Section 5.7).
STAT:	module used to compare two data structures (see Section 5.8).
GREP:	module used to locate information on a data structure (see Section 5.9).
FINDO:	module used to find the zero of a tabulated function (see Section 5.11).
END:	module used to terminate an execution controlled by the generalized driver (see Section 5.13).

## 2.5 The DRAGON Data Structures

The transfer of information between the DRAGON execution modules is ensured by well defined data structure. They are generally created or modified directly by one of the modules of DRAGON or by one of the utility modules. Here we will give a brief description of these data structures but a more complete description of their content is also available upon request.<sup>[?]</sup> These data structures are memory-resident or persistent (i.e., XSM-type) objects.

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. It can be created by the MAC:, LIB: and EDI: modules. It can also be modified by the SHI:, USS: and EVO: modules. Such a structure (either stand-alone or as part of a MICROLIB) is also required for a successful execution of the ASM: and FLU: modules.
MICROLIB	a standard data structure used by DRAGON to transfer microscopic and macroscopic cross sections between its modules. It always include a MACROLIB substructure. It can be a stand-alone structure or included into a larger structure, such as an EDITION structure. It can be created by the LIB: and EDI: modules. It can also be modified by the MAC:, SHI:, USS: and EVO: modules.
GEOMETRY	a standard data structure used by DRAGON to transfer the geometry between its modules. It can be a stand-alone structure or included into a larger structure, such as another GEOMETRY structure. It can be created by the GEO: module. Such a structure is also required directly for a successful execution of the tracking modules (SYBILT:, EXCELT: and MCCGT:).
TRACKING	a standard data structure used by DRAGON to transfer the general tracking information between its modules. It is a stand-alone structure. It can be created by the SYBILT:, EXCELT: and MCCGT: modules. Such a structure is also required directly for a successful execution of the ASM: module.

ASMPIJ	a standard data structure used by DRAGON to transfer the multigroup response and collision probability matrices between its modules. It is a stand-alone structure. It is created by the <b>ASM:</b> module. Such a structure is also required directly for a successful execution of the <b>FLU:</b> module.
FLUXUNK	a standard data structure used by DRAGON to transfer the fluxes between its modules. It is a stand-alone structure. It is created by the <b>FLU:</b> module. Such a structure is also required for a successful execution of the <b>EDI:</b> and <b>EVO:</b> modules.
EDITION	a standard data structure used by DRAGON to store condensed and merged microscopic and macroscopic cross sections. It is a stand-alone structure but can contain <b>MACROLIB</b> and <b>MICROLIB</b> substructure. It is created by the <b>EDI:</b> module. Such a structure is also required for a successful execution of the <b>COMPO:</b> module.
BURNUP	a standard data structure used by DRAGON to store burnup informations. It is created by the <b>EVO:</b> module. Such a structure is also required for a successful execution of the <b>COMPO:</b> module.
DRAGLIB	a standard data structure used by DRAGON (input) to recover isotopic-, dilution- and temperature-dependent information, including multigroup microscopic cross sections and burnup data. This is a stand-alone structure that is generally stored on a persistent LCM object. It may be created by the <b>dragr</b> module of <b>NJOY</b> .
CPO	a standard data structure used by DRAGON to store a simplified reactor database. It is a stand-alone structure that must be stored on a linked list or an XSM file. It is created by the <b>CPO:</b> module. It is required for a successful execution of the <b>CFC:</b> module. It can be used by the <b>CRE:</b> module of <b>DONJON</b> .
MULTICOMPO	a standard data structure used by DRAGON (output) to store reactor related information and to classified it using tuples of local and global parameters. This is a stand-alone structure that is generally stored on a persistent LCM object. It is created by the <b>COMPO:</b> module.
SAPHYB	a standard data structure used by <b>APOLLO2</b> and <b>DRAGON</b> (output) to store reactor related information and to classified it using tuples of global parameters. This is a stand-alone structure that is generally stored on a persistent LCM object. It is created by the <b>SAP:</b> module.
FBMXSDB	a standard data structure used by <b>DRAGON</b> 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 <b>CFC:</b> module. It can be used by the <b>AFM:</b> module of <b>DONJON</b> . <sup>[?]</sup>

## 2.6 Main Updates in DRAGON

The frozen version (DRAGON Release 3.06) has seen a large number of changes since the first official release of the code (DRAGON\_960627).

The following DRAGON package (DRAGON Version4) is an evolution of the frozen version, released as an attempt to introduce innovative capabilities:

- The new self-shielding module **USS:** allow increased accuracy and better representation of phenomena such as *distributed self-shielding effects* and *mutual self-shielding effects*.
- The new flux solution solver **MCCG** is an implementation of the long characteristics method proposed by Igor Suslov. This solver is initiated by the new tracking module **MCCGT:**.
- The new flux solution module **FLU:** is a complete rewrite of the outer iteration for the multigroup flux calculation that is now compatible with the method of characteristics and with any other approach requiring inner iterations. The **MOCC:** module is no longer required.

- The burnup module **EVO:** was extended to take into account energy produced by radioactive decay and by reactions other than fission.
- The new module **COMPO:** is used to create and increment a multiparameter reactor database. The module **NCR:** is used to interpolate an existing multiparameter reactor database.
- The flux solution solver **SYBIL** related to 2D assembly calculations was extended to allow sectorization of the cells.
- The method of discrete ordinates is implemented in tracking module **SNT:**.
- The **EXCELL:** module has been removed, but its capability is now implemented using the **XCLL** keyword in **EXCELT:**.
- The **LIB:** module can access NDAS-formatted cross-section libraries.

The current DRAGON package (DRAGON Version5) is a rewrite of the code around the GANLIB5 kernel<sup>[?]</sup>, intended to be 64-bit clean.

### 3 THE DRAGON MODULES

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

#### 3.1 The MAC: module

In DRAGON, the macroscopic cross sections associated with each mixture are stored in a MACROLIB (as an independent data structure or as part of a MICROLIB) which may be generated using one of different ways:

- First, one can use directly the input stream already used for the remaining DRAGON data. In this case, a single macroscopic library is involved.
- The second method is via a GOXS format binary sequential file.<sup>[?]</sup> 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. One can also transfer the macroscopic cross sections to a GOXS format binary file if required. In this case, a single macroscopic library is involved.
- The third input method is through a file which already contains a MACROLIB. In this case, two macroscopic and microscopic libraries are to be combined
- The fourth method consists to update an existing MACROLIB using control-variable data recovered from a L\_OPTIMIZE object.

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

Table 2: Structure (MAC:)

```
{ MACLIB := MAC: [ MACLIB ] :: (descmacinp)
  | MICLIB := MAC: MICLIB :: (descmacinp)
  | MACLIB := MAC: [ MACLIB ] [ OLDLIB ] :: (descmacupd)
  | MACLIB := MAC: MACLIB OPTIM ;
}
```

The meaning of each of the terms above is:

<i>MACLIB</i>	<b>character*12</b> name of a MACROLIB that will contain the macroscopic cross sections. If <i>MACLIB</i> appears on both LHS and RHS, it is updated; otherwise, it is created. If <i>MACLIB</i> is created, all macroscopic cross sections are first initialized to zero.
<i>MICLIB</i>	<b>character*12</b> 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).
<i>OLDLIB</i>	<b>character*12</b> name of a MACROLIB or a MICROLIB which will be used to update or create the <i>MACLIB</i> MACROLIB.
<i>OPTIM</i>	<b>character*12</b> name of a L_OPTIMIZE object. The MACROLIB <i>MACLIB</i> is updated using control-variable data recovered from <i>OPTIM</i> .
(descmacinp)	macroscopic input data structure for this module (see Section 3.1.1).



(**descmacupd**) macroscopic update data structure for this module (see Section 3.1.3).

### 3.1.1 Input structure for module MAC:

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

Table 3: Structure (**descmacinp**)

```
[ EDIT iprint ]
[ NGRO ngroup ]
[ NMIX nmixt ]
[ NIFI nifiss ]
[ DELP ndel ]
[ ANIS naniso ]
[ NADF nadf ]
[ CTRA { NONE | APOL | WIMS | LEAK } ]
[ ALBP nalbp ((albedp(ig,ia),ig=1,ngroup),ia=1,nalbp) ]
[ WRIT GOXSWN ]
[ ENER (energy(jg), jg=1,ngroup +1) ]
[ VOLUME (volume(ibm), ibm=1,nmixt) ]
[ ADD ]
[[ { READ [ (imat(i), i=1,nmixt) ] GOXSRN [ DELE ] | READ INPUT [[ (descxs) ] ] } ]
[[ STEP istep READ INPUT [[ (descxs) ] ] ]
[ NORM ]
;
```

with

EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	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 <i>iprint</i> is greater than or equal to 2. The transfer cross sections will be printed if this parameter is greater than or equal to 3. The normalization of the transfer cross sections will be checked if <i>iprint</i> is greater than or equal to 5.
NGRO	keyword to specify the number of energy groups for which the macroscopic cross sections will be provided. This information is required only if <i>MACLIB</i> is created and the cross sections are taken directly from the input data stream.
<i>ngroup</i>	the number of energy groups used for the calculations in DRAGON. The default value is <i>ngroup</i> =1.
NMIX	keyword used to define the number of material mixtures. This information is required only if <i>MACLIB</i> is created and the cross sections are taken directly from the input data stream or from a GOXS file.
<i>nmixt</i>	the maximum number of mixtures (a mixture is characterized by a distinct set of macroscopic cross sections) the MACROLIB may contain. The default value is <i>nmixt</i> =1.
NIFI	keyword used to specify the maximum number of fissile spectrum associated with each mixture. Each fission spectrum generally represents a fissile isotope. This information

is required only if *MACLIB* is created and the cross sections are taken directly from the input data stream.

<i>nifiss</i>	the maximum number of fissile isotopes per mixture. The default value is <i>nifiss</i> =1.
DELP	keyword used to specify the number of delayed neutron groups.
<i>ndel</i>	the number of delayed neutron groups. The default value is <i>ndel</i> =0.
ANIS	keyword used to specify the maximum level of anisotropy permitted in the scattering cross sections. This information is required only if <i>MACLIB</i> is created and the cross sections are taken directly from the input data stream.
<i>naniso</i>	number of Legendre orders for the representation of the scattering cross sections. The default value is <i>naniso</i> =1 corresponding to the use of isotropic scattering cross sections.
NADF	keyword used to specify the number of averaged fluxes surrounding the geometry and used to compute <i>assembly discontinuity factors</i> (ADF).
<i>nadf</i>	number of averaged fluxes surrounding the geometry.
CTRA	keyword to specify the type of transport correction that should be generated and stored on the MACROLIB. The transport correction is to be subtracted from the total and isotropic ( $P_0$ ) within-group scattering cross sections. A leakage correction, equal to the difference between current- and flux-weighted total cross sections ( $\Sigma_1 - \Sigma_0$ ) is also applied in the APOL and LEAK cases. All the modules that will read this MACROLIB will then have access to transport corrected cross sections. The default is no transport correction when the MACROLIB is created from the input or GOXS files.
NONE	keyword to specify that no transport correction should be used in this calculation.
APOL	keyword to specify that an APOLLO type transport correction based on the linearly anisotropic ( $P_1$ ) scattering cross sections is to be set. This correction assumes that the micro-reversibility principle is valid for all energy groups. $P_1$ scattering information must exist in the MACROLIB.
WIMS	keyword to specify that a WIMS-type transport correction is used. The transport correction is recovered from a record named TRANC. This record must exist in the MACROLIB.
LEAK	A leakage correction is applied to the total and $P_0$ within-group scattering cross sections. No transport correction is applied in this case.
ALBP	keyword used for the input of the multigroup physical albedo array.
<i>nalbp</i>	the maximum number of multigroup physical albedos.
<i>albedp</i>	multigroup physical albedo array.
WRIT	keyword used to write cross section data to a GOXS file. In the case where <i>nifiss</i> >1, this option is invalid.
GOXSWN	<b>character*7</b> name of the GOXS file to be created or updated.
ENER	keyword to specify the energy group limits.
<i>energy</i>	energy (eV) array which define the limits of the groups ( <i>ngroup</i> +1 elements). Generally <i>energy</i> (1) is the highest energy.
VOLUME	keyword to specify the mixture volumes.
<i>volume</i>	volume (cm <sup>3</sup> ) occupied by each mixture.

ADD	keyword for adding increments to existing macroscopic cross sections. In this case, the information provided in <b>(descxs)</b> 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.
<i>imat</i>	array of mixture identifiers to be read from a GOXS file. The maximum number of identifiers permitted is <i>nmixt</i> and the maximum value that <i>imat</i> may take is <i>nmixt</i> . When <i>imat</i> is 0, the corresponding mixture on the GOXS file is not included in the MACROLIB. In the cases where <i>imat</i> 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	<b>character*7</b> name of the GOXS file to be read.
DELE	keyword to specify that the GOXS file is deleted after being read ( <b>Revision 3.03 only</b> ).
INPUT	keyword to specify that mixture cross sections will be read on the input stream.
<b>(descxs)</b>	structure describing the format used for reading the mixture cross sections from the input stream (see Section 3.1.2).
STEP	keyword used to create a perturbation directory.
<i>istep</i>	the index of the perturbation directory.
NORM	keyword to specify that the macroscopic scattering cross sections and the fission spectrum have to be normalized. This option is available even if the mixture cross sections were not read by the MAC: module.

### 3.1.2 Macroscopic cross section definition

Table 4: Structure **(descxs)**

```

MIX [ matnum ]
  [ { NTOTO | TOTAL } (xssigt(jg), jg=1,ngroup) ]
  [ NTOT1 (xssig1(jg), jg=1,ngroup) ]
  [ TRANC (xsstra(jg), jg=1,ngroup) ]
  [ NUSIGF ((xssigf(jf,jg), jg=1,ngroup), jf=1,nifiss) ]
  [ CHI ((xschi(jf,jg), jg=1,ngroup), jf=1,nifiss)]
  [ FIXE (xsfixe(jg), jg=1,ngroup) ]
  [ DIFF (diff(jg), jg=1,ngroup) ]
  [ DIFFX (xdiffx(jg), jg=1,ngroup) ]
  [ DIFFY (xdiffy(jg), jg=1,ngroup) ]
  [ DIFFZ (xdiffz(jg), jg=1,ngroup) ]
  [ NUSIGD (((xssigd(jf,idel,jg), jg=1,ngroup), idel=1,ndel), jf=1,nifiss) ]
  [ CHDL (((xschid(jf,idel,jg), jg=1,ngroup), idel=1,ndel), jf=1,nifiss)]
  [ OVERV (overv(jg), jg=1,ngroup) ]
  [ NFTOT (nftot(jg), jg=1,ngroup) ]
  [ FLUX-INTG (xsint0(jg), jg=1,ngroup) ]

```

continued on next page

Structure (**descxs**)

continued from last page

```
[ FLUX-INTG-P1 (xsint1(jg), jg=1,ngroup) ]
[ H-FACTOR (hfact(jg), jg=1,ngroup) ]
[ SCAT (( nbscat(jl,jg), ilastg(jl,jg),(xsscat(jl,jg,ig),
           ig=1,nbscat(jl,jg) ), jg=1,ngroup), jl=1,naniso) ]
[[ ADF hadf (xadf(jg), jg=1,ngroup) ]]
```

MIX	keyword to specify that the macroscopic cross sections associated with a new mixture are to be read.
<i>matnum</i>	identifier for the next mixture to be read. The maximum value permitted for this identifier is <i>nmixt</i> . When <i>matnum</i> is absent, the mixtures are numbered consecutively starting with 1 or with the last mixture number read either on the GOXS or the input stream.
NTOTO	keyword to specify that the total macroscopic cross sections for this mixture follows.
TOTAL	alias keyword for NTOTO.
<i>xssigt</i>	array representing the multigroup total macroscopic cross section ( $\Sigma^g$ in $\text{cm}^{-1}$ ) associated with this mixture.
NTOT1	keyword to specify that the $P_1$ -weighted total macroscopic cross sections for this mixture follows.
<i>xssig1</i>	array representing the multigroup $P_1$ -weighted total macroscopic cross section ( $\Sigma_1^g$ in $\text{cm}^{-1}$ ) associated with this mixture.
TRANC	keyword to specify that the transport correction macroscopic cross sections for this mixture follows.
<i>xsstra</i>	array representing the multigroup transport correction macroscopic cross section ( $\Sigma_{tc}^g$ in $\text{cm}^{-1}$ ) associated with this mixture.
NUSIGF	keyword to specify that the macroscopic fission cross section multiplied by the average number of neutrons per fission for this mixture follows.
<i>xssigf</i>	array representing the multigroup macroscopic fission cross section multiplied by the average number of neutrons per fission ( $\nu\Sigma_f^g$ in $\text{cm}^{-1}$ ) for all the fissile isotopes associated with this mixture.
CHI	keyword to specify that the fission spectrum for this mixture follows.
<i>xschi</i>	array representing the multigroup fission spectrum ( $\chi^g$ ) for all the fissile isotopes associated with this mixture.
FIXE	keyword to specify that the fixed neutron source density for this mixture follows.
<i>xsfixe</i>	array representing the multigroup fixed neutron source density for this mixture ( $S^g$ in $s^{-1}\text{cm}^{-3}$ ).
DIFF	keyword to specify that the isotropic diffusion coefficient for this mixture follows.
<i>diff</i>	array representing the multigroup isotropic diffusion coefficient for this mixture ( $D^g$ in $\text{cm}$ ).
DIFFX	keyword for input of the $X$ -directed diffusion coefficient.

<i>xdiffx</i>	array representing the multigroup $X$ -directed diffusion coefficient ( $D_x^g$ in cm) for the mixture <i>matnum</i> .
<b>DIFFY</b>	keyword for input of the $Y$ -directed diffusion coefficient.
<i>xdiffy</i>	array representing the multigroup $Y$ -directed diffusion coefficient ( $D_y^g$ in cm) for the mixture <i>matnum</i> .
<b>DIFFZ</b>	keyword for input of the $Z$ -directed diffusion coefficient.
<i>xdiffz</i>	array representing the multigroup $Z$ -directed diffusion coefficient ( $D_z^g$ in cm) for the mixture <i>matnum</i> .
<b>NUSIGD</b>	keyword to specify that the delayed macroscopic fission cross section multiplied by the average number of neutrons per fission for this mixture follows.
<i>xssigd</i>	array representing the delayed multigroup macroscopic fission cross section multiplied by the average number of neutrons per fission ( $\nu\Sigma_f^{g,idel}$ in $\text{cm}^{-1}$ ) for all the fissile isotopes associated with this mixture.
<b>CHDL</b>	keyword to specify that the delayed fission spectrum for this mixture follows.
<i>xschid</i>	array representing the delayed multigroup fission spectrum ( $\chi^{g,idel}$ ) for all the fissile isotopes associated with this mixture.
<b>OVERV</b>	keyword for input of the multigroup average of the inverse neutron velocity.
<i>overv</i>	array representing the multigroup average of the inverse neutron velocity ( $\langle 1/v \rangle_m^g$ ) for the mixture <i>matnum</i> .
<b>NFTOT</b>	keyword for input of the multigroup macroscopic fission cross sections.
<i>nftot</i>	array representing the multigroup macroscopic fission cross section ( $\Sigma_f^g$ ) for the mixture <i>matnum</i> .
<b>FLUX-INTG</b>	keyword for input of the multigroup $P_0$ volume-integrated fluxes.
<i>xsint0</i>	array representing the multigroup $P_0$ volume-integrated fluxes ( $V\phi_0^g$ ) for the mixture <i>matnum</i> .
<b>FLUX-INTG-P1</b>	keyword for input of the multigroup $P_1$ volume-integrated fluxes.
<i>xsint1</i>	array representing the multigroup $P_1$ volume-integrated fluxes ( $V\phi_1^g$ ) for the mixture <i>matnum</i> .
<b>H-FACTOR</b>	keyword to specify that the power factor for this mixture follows.
<i>hfact</i>	array representing the multigroup power factor for this mixture ( $H^g$ in $\text{MeV cm}^{-1}$ ).
<b>SCAT</b>	keyword to specify that the macroscopic scattering cross section matrix for this mixture follows.
<i>nbscat</i>	array representing the number of primary groups <i>ig</i> with non vanishing macroscopic scattering cross section towards the secondary group <i>ig</i> considered for each anisotropy level associated with this mixture.
<i>ilastg</i>	array representing the group index of the most thermal group with non-vanishing macroscopic scattering cross section towards the secondary group <i>ig</i> considered for each anisotropy level associated with this mixture.

<i>xsscat</i>	array representing the multigroup macroscopic scattering cross section ( $\Sigma_{sl}^{ig \rightarrow jg}$ in $\text{cm}^{-1}$ ) from the primary group <i>ig</i> towards the secondary group <i>jg</i> considered for each anisotropy level associated with this mixture. The elements are ordered using decreasing primary group number <i>ig</i> , from <i>ilastg</i> to ( <i>ilastg</i> - <i>nbscat</i> + 1), and an increasing secondary group number <i>jg</i> . Examples of input structures for macroscopic scattering cross sections can be found in Section 7.1.
<b>ADF</b>	keyword to specify that the boundary flux information for this mixture follows.
<i>hadf</i>	character*8 type of a flux surrounding the geometry. The maximum number of types is equal to <i>nadf</i> .
<i>xadf</i>	array representing a multigroup flux of type <i>hadf</i> surrounding the geometry for this mixture.

### 3.1.3 Update structure for operator MAC:

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

Table 5: Structure (**descmacupd**)

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

with

<b>EDIT</b>	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index used to control the printing in this operator. 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 <i>iprint</i> is greater than or equal to 2. The transfer cross sections will be printed if this parameter is greater than or equal to 3. The normalization of the transfer cross sections will be checked if <i>iprint</i> is greater than or equal to 5.
<b>CTRA</b>	keyword to specify the type of transport correction that should be generated and stored on the MACROLIB. All the operators that will read this MACROLIB will then have access to transport corrected cross sections. In the case where the MACROLIB is updated using other MACROLIB or MICROLIB the default is to use a transport correction whenever one of these older data structure requires a transport correction.
<b>OFF</b>	deactivates the transport correction.
<b>MIX</b>	keyword to specify that the macroscopic cross sections associated with a mixture is to be created or updated.
<i>numnew</i>	mixture number to be updated or created on the output MACROLIB.
<i>numold</i>	mixture number on an old MACROLIB or MICROLIB which will be used to update or create <i>numnew</i> on the output macrolib
<b>OLDL</b>	the macroscopic cross sections associated with mixture <i>numold</i> are taken from <i>OLDLIB</i> . This is the default option.

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

### 3.2 The LIB: module

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

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

$MICLIB := LIB: [ MICLIB [ OLDLIB ] ] :: (desclib)$
---

where

<i>MICLIB</i>	<b>character*12</b> name of the MICROLIB that will contain the internal library. If <i>MICLIB</i> appears on both LHS and RHS, it is updated; otherwise, it is created.
<i>OLDLIB</i>	<b>character*12</b> name of a read-only MACROLIB, MICROLIB or BURNUP data structure. In the case where a MACROLIB is considered, it is included directly in the <i>MICLIB</i> before updating it. If it is a second MICROLIB or a BURNUP data structure, the number densities for the isotopes in file <i>MICLIB</i> will be replaced selectively by those found in <i>OLDLIB</i> .
( <b>desclib</b> )	input structure for this module (see Section 3.2.1).

#### 3.2.1 Data input for module LIB:

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

Table 7: Structure (**desclib**)

<pre>[ EDIT <i>iprint</i> ] [ NGRO <i>ngroup</i> ] [ MXIS <i>nmisot</i> ] [ NMIX <i>nmixt</i> ] [ CALENDF <i>ipreci</i> ] [ CTRA { NONE   APOL   WIMS   OLDW   LEAK } ] [ ANIS <i>naniso</i> ] [ ADJ ] [ PROM ] [ { SKIP   INTR   SUBG   PT   PTMC   PTSL   NEWL } ] [ MACR ] [ ADED <i>nedit</i> ( <i>HEDIT</i>(i), i=1,<i>nedit</i> ) ] [ DEPL { LIB: { DRAGON   WIMS   WIMSD4   WIMSAECL   NDAS } FIL: <i>NAMEFIL</i>         LIB: { APLIB2   APXSM } FIL: <i>NAMEFIL</i> (<b>descdeplA2</b>)         <i>ndepl</i> (<b>descdepl</b>) } ] [[ MIXS LIB:    { DRAGON   MATXS   MATXS2   WIMS   WIMSD4   WIMSAECL   NDAS   APLIB1   APLIB2   APXSM }    FIL: <i>NAMEFIL</i> [[ (<b>descmix1</b>) ] ] ]]</pre>
--

It is possible to reset an existing microlib (i.e., *MICLIB* is present in the RHS) and to reprocess all the isotopes from the cross section libraries. In this case, (**desclib**) takes the simplified form:



Table 8: Structure (**desclib**)

```
[ EDIT iprint ]
{ INTR | SUBG | PT | PTMC | PTSL | NEWL } [ MACR ]
MIXS ;
```

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

Table 9: Structure (**desclib**)

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

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

Table 10: Structure (**desclib**)

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

with

EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index used to control the printing in this operator. 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. If $iprint \geq 10$ , the depletion chain is printed in the format of structure ( <b>descdepl</b> ). If $iprint \geq 20$ , the depletion chain is also printed in the format of structure ( <b>descdeplA2</b> ).
MXIS	keyword used to redefine the maximum number of isotopes per mixture.
<i>nmisot</i>	the maximum number of isotopes per mixture. By default up to 300 different isotopes per mixture are permitted.
NMIX	keyword used to define the number of material mixtures. This data is required if <i>MICLIB</i> is created.
<i>nmixt</i>	the maximum number of mixtures (a mixture is characterized by a distinct set of macroscopic cross sections).
CALENDF	keyword to set the accuracy of the CALENDF probability tables.
<i>ipreci</i>	integer set to 1, 2, 3 or 4. The highest the value, the more accurate are the probability tables.
CTRA	keyword to specify the type of transport correction that should be generated and stored on the MICROLIB. The transport correction is to be subtracted from the total and isotropic

( $P_0$ ) within-group scattering cross sections. A leakage correction, equal to the difference between current- and flux-weighted total cross sections ( $\sigma_1 - \sigma_0$ ) is also applied in the APOL, OLDW and LEAK cases. All the operators 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 based on the linearly anisotropic ( $P_1$ ) within-group scattering cross sections is to be set. This correction assumes that the micro-reversibility principle is valid for all energy groups. This type of correction uses $P_1$ scattering information present on the library.
WIMS	This type of correction uses directly a transport-correction provided on the library. Such information is available in WIMSD4 and WIMS-AECL libraries. This is the new recommended option with WIMS-type libraries. <i>This option has no effect on libraries that does not contain transport correction information.</i>
OLDW	keyword to specify that a WIMS type transport correction based on the $P_1$ scattering cross sections is to be set. This correction assumes that the micro-reversibility principle is valid only for groups energies less than 4.0 eV. For the remaining groups a $1/E$ current spectrum is considered in the evaluation of the transport correction. This type of correction uses $P_1$ scattering information present on the library.
LEAK	A leakage correction is applied to the total and $P_0$ within-group scattering cross sections. No transport correction is applied in this case.
ANIS	keyword to specify the maximum level of anisotropy for the scattering cross sections.
<i>naniso</i>	number of Legendre orders for the representation of the scattering cross sections. Isotropic scattering is represented by <i>naniso</i> =1 while <i>naniso</i> =2 represents linearly anisotropic scattering. Generally the linearly anisotropic ( $P_1$ ) scattering contributions are taken into account via the transport correction (see CTRA keyword) in the transport calculation. For $B_1$ or $P_1$ leakage calculations, the linearly anisotropic scattering cross sections are taken into account explicitly. The default value is <i>naniso</i> =2.
ADJ	keyword to specify the production of adjoint macroscopic cross sections. By default, direct cross sections are produced.
PROM	keyword to specify that prompt neutrons are to be considered for the calculation of the fission spectrum. By default, the contribution due to delayed neutrons is considered. This option is only compatible with a MATXS or MATXS2 format library.
SKIP	keyword to recover the user-defined microlib data without processing any library (i.e., without temperature and/or dilution interpolation).
INTR	keyword to perform a temperature and dilution interpolation of the microscopic cross sections present in the libraries. The bin-type cross-section data is not processed. This is the default option.
SUBG	keyword to activate the calculation of the physical probability tables using the temperature-interpolated cross-section data as input. <sup>[?, ?]</sup> The bin-type cross-section data is not processed.
PT	keyword to activate the calculation of the CALENDF-type mathematical probability tables ( <i>without</i> slowing-down correlated weight matrices) using the bin-type cross-section data as input. <sup>[?]</sup> This option is compatible with the Sanchez-Coste self-shielding method and with the subgroup projection method (SPM). <sup>[?]</sup>
PTMC	this option is similar to the PT procedure. Here, the base points of the probability tables corresponding to fission and scattering cross sections and to components of the transfer scattering matrix are also obtained using the CALENDF approach.

PTSL	keyword to activate the calculation of the CALENDF-type mathematical probability tables and slowing-down correlated weight matrices using the bin-type cross-section data as input. <sup>[?]</sup>
NEWL	keyword to activate the calculation of a DRAGLIB library using the temperature-interpolated cross-section data as input. The bin-type cross-section data is processed.
MACR	keyword to force the calculation of the embedded macrolib. By default, the embedded macrolib is computed, <i>except if</i> one of the key words SKIP, INTR, SUBG, PT or NEWL is used.
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: and utility operators.
<i>nedit</i>	number of types of additional cross sections.
<i>HEDIT</i>	<p><b>character*6</b> name of an additional cross-section type. This name also corresponds to vectorial reactions in a MATXS and MATXS2 format library. For example:</p> <p>NWT0/NWT1=<math>P_0/P_1</math> library weight functions.  NTOT0/NTOT1=<math>P_0/P_1</math> neutron total cross sections.  NELAS=Neutron elastic scattering cross sections (MT=2).  NINEL=Neutron inelastic scattering cross sections (MT=4).  NG=Neutron radiative capture cross sections (MT=102).  NFTOT=Total fission cross sections (MT=18).  NUDEL=Number of delayed secondary neutrons (Nu-D / MT=455).  NFSLO=<math>\nu</math>*slow fission cross section.  NHEAT=Heat production cross section.  CHIS/CHID=Slow/delayed fission spectrum.  NF/NNF/N2NF/N3NF=<math>\nu</math>*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,<math>\alpha</math>) transmutation cross sections (MT=103 and 107).</p> <p>By default, DRAGON will always attempt to recover the additional cross sections NG, NFTOT, NHEAT and N2N which are required for the depletion calculations.</p>
DEPL	keyword to specify that the isotopic depletion (burnup) chain is to be read. For a given LIB: execution only one isotopic depletion chain can be read.
MIXS	keyword to specify that the mixture description is to be read. For a given LIB: execution more than one cross-section library can be read.
LIB:	keyword to specify the type of library from which the isotopic depletion chain or microscopic cross section is to be read. It is optional when preceded by the keyword DEPL in which case the isotopic depletion chain is read from the standard input file.
DRAGON	keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the DRAGLIB format.
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 and the microscopic cross sections are in the WIMSD4 format.
WIMS	keyword to specify that the isotopic depletion chain and the microscopic cross sections are in the WIMS-AECL format.

WIMSAECL	keyword to specify that the isotopic depletion chain and the microscopic cross sections are in the WIMS-AECL format.
NDAS	keyword to specify that the isotopic depletion chain and the microscopic cross sections are in the NDAS format, as used in recent versions of WIMS-AECL.
APLIB1	keyword to specify that the microscopic cross sections are in the APOLLO-1 format. There are no depletion chains available for libraries using this format.
APLIB2	keyword to specify that the microscopic cross sections are in the APOLLO-2 direct access format. There are no depletion chains available for libraries using this format. However, fission yields, radioactive decay constants and energy released per fission or radiative capture are recovered from the file. Only versions of the APOLIB-2 libraries subsequent or equal to CEA93-V4 can be processed. The list of isotopes (standard and self-shielded) available in an APOLIB-2 is printed by setting the print flag to a value <i>iprint</i> ≥ 10.
APXSM	keyword to specify that the microscopic cross sections are in the APOLIB-XSM format, the output format of N2A2 utility. There are no depletion chains available for libraries using this format. However, fission yields, radioactive decay constants and energy released per fission or radiative capture are recovered from the file. The list of isotopes (standard and self-shielded) available in an APOLIB-XSM is printed by setting the print flag to a value <i>iprint</i> ≥ 10.
FIL:	keyword to specify the name of the file where is stored the isotopic depletion data.
NAMEFIL	<b>character*64</b> name of the library where the isotopic depletion chain or the microscopic cross sections are stored. Library names in DRAGLIB format are limited to 12 characters. A NDAS library is made of two or more files. These file names must be concatenated in a single <i>NAMEFIL</i> name, using semicolumns as separators. The ASCII index file is always the first, followed by optional patch files, and terminated by the main direct-access binary file. The following sample data line corresponds to a NDAS library without patch:  MIXS LIB: NDAS FIL: E65LIB6.idx:E65LIB6.sdb
<i>ndepl</i>	number of isotopes in the depleting chain.
( <b>descdepl</b> )	input structure describing the depletion chain (see Section 3.2.2).
( <b>descdeplA2</b> )	simplified input structure describing the depletion chain in cases where an APOLIB-2 or APOLIB-XSM file is used (see Section 3.2.2).
MAXS	keyword to specify that the mixture density on <i>MICLIB</i> are to be modified. If <i>OLDLIB</i> is present and ( <b>descmix2</b> ) is absent, a direct one to one correspondence between the isotope on both libraries is assumed. If <i>OLDLIB</i> and ( <b>descmix2</b> ) are present, only the mixture on the library file specified by ( <b>descmix2</b> ) are updated using information from the <i>OLDLIB</i> . If <i>OLDLIB</i> is absent and ( <b>descmix2</b> ) is present, only the mixture on <i>MICLIB</i> specified by ( <b>descmix2</b> ) are updated.
BURN	keyword to specify that the mixture density on <i>MICLIB</i> are to be updated using information taken from <i>OLDLIB</i> . If ( <b>descmix2</b> ) is absent, a direct one to one correspondence between the isotope on <i>OLDLIB</i> and <i>MICLIB</i> is assumed. If ( <b>descmix2</b> ) is present, only the mixture specified by ( <b>descmix2</b> ) are updated using information from <i>OLDLIB</i> .
<i>iburn</i>	burnup step from the burnup file to use. This step must be already present on the burnup file.
<i>tburn</i>	burnup time in days from the burnup file to use. This time step must be already present on the burnup file.

- (**descmix1**) input structure describing the isotopic and physical properties of a given mixture (see Section 3.2.3).
- (**descmix2**) input structure describing perturbations to the isotopic and physical properties of a given mixture (see Section 3.2.3).

Note that it is possible to recompute the embedded macrolib in an existing microlib named **MICRO** by writing

```
MICRO := LIB: MICRO :: MACR MIXS ;
```

### 3.2.2 Depletion data structure

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

Table 11: Structure (**descdepl**)

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

with:

CHAIN	keyword to specify the beginning of the depletion chain.										
NAMDPL	<b>character*12</b> name of an isotope (or isomer) of the depletion chain that appears in the cross-section library.										
izaef	optional six digit integer representing the isotope. The first two digits represent the atomic number of the isotope; the next three indicate its mass number and the last digit indicates the excitation level of the nucleus (0 for a nucleus in its ground state, 1 for an isomer in its first excited state, etc.). For example, $^{238}\text{U}$ in its ground state will be represented by izaef=922380.										
DECAY	indicates that a decay reaction takes place either for production of this isotope or its depletion.										
dcr	radioactive decay constant (in $10^{-8} \text{ s}^{-1}$ ) of the isotope. By default, dcr=0.0.										
reaction	<b>character*6</b> identification of a neutron-induced reaction that takes place either for production of this isotope, its depletion, or for producing energy. Example of reactions are following: <table style="margin-left: 20px;"> <tr> <td>NG</td><td>indicates that a radiative capture reaction takes place either for production of this isotope, its depletion or for producing energy.</td></tr> <tr> <td>N2N</td><td>indicates that the following reaction is taking place:               <math display="block">n + {}^A X_Z \rightarrow 2n + {}^{A-1} X_Z</math> </td></tr> <tr> <td>N3N</td><td>indicates that the following reaction is taking place:               <math display="block">n + {}^A X_Z \rightarrow 3n + {}^{A-2} X_Z</math> </td></tr> <tr> <td>N4N</td><td>indicates that the following reaction is taking place:               <math display="block">n + {}^A X_Z \rightarrow 4n + {}^{A-3} X_Z</math> </td></tr> <tr> <td>NP</td><td>indicates that the following reaction is taking place:               <math display="block">n + {}^A X_Z \rightarrow p + {}^A Y_{Z-1}</math> </td></tr> </table>	NG	indicates that a radiative capture reaction takes place either for production of this isotope, its depletion or for producing energy.	N2N	indicates that the following reaction is taking place: $n + {}^A X_Z \rightarrow 2n + {}^{A-1} X_Z$	N3N	indicates that the following reaction is taking place: $n + {}^A X_Z \rightarrow 3n + {}^{A-2} X_Z$	N4N	indicates that the following reaction is taking place: $n + {}^A X_Z \rightarrow 4n + {}^{A-3} X_Z$	NP	indicates that the following reaction is taking place: $n + {}^A X_Z \rightarrow p + {}^A Y_{Z-1}$
NG	indicates that a radiative capture reaction takes place either for production of this isotope, its depletion or for producing energy.										
N2N	indicates that the following reaction is taking place: $n + {}^A X_Z \rightarrow 2n + {}^{A-1} X_Z$										
N3N	indicates that the following reaction is taking place: $n + {}^A X_Z \rightarrow 3n + {}^{A-2} X_Z$										
N4N	indicates that the following reaction is taking place: $n + {}^A X_Z \rightarrow 4n + {}^{A-3} X_Z$										
NP	indicates that the following reaction is taking place: $n + {}^A X_Z \rightarrow p + {}^A Y_{Z-1}$										

NA	indicates that the following reaction is taking place:
$n + {}^A X_Z \rightarrow {}^4\text{He}_2 + {}^{A-3} X_{Z-2}$	
NFTOT	indicates that a fission is taking place.
energy	energy (in MeV) recoverable per neutron-induced reaction of type <i>reaction</i> . If the energy associated to radiative capture is not explicitly given, it should be added to the energy released per fission. By default, <i>energy</i> =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	branching ratio or production yield expressed in fraction.
NAMPAR	<b>character*12</b> name of the a parent isotope (or isomer) that appears in the cross-section library.
ENDCHAIN	keyword to specify the end of the depletion chain.

If the keyword **APLIB2** or **APXSM** was used in structure (**desclib**), part of the depletion data is recovered from the APOLIB file: the fission yields, the radioactive decay constants and the energy released per fission or radiative capture. Moreover, the following simplified structure is used to provide the remaining depletion data:

Table 12: Structure (**descdeplA2**)

```
CHAIN
[[ NAMDPL [ FROM [[ { DECAY | reaction } yield NAMPAR ] ] ] ] ]
ENDCHAIN
```

In this case, the following rules apply:

- We should provide the names *NAMDPL* of *all* the depleting isotopes (i.e. isotopes with a time-dependent number density), including the pseudo fission products (PFP).
- The fission father reactions (**NFTOT**) are not given.
- The stable isotopes are automatically recovered from the APOLIB file. They are not given in structure (**descdeplA2**).
- An isotope is considered to be stable if it is not present in structure (**descdeplA2**), has no father and no daughter, but can release energy by fission or radiative capture.
- It is possible to truncate the isotope name *NAMDPL* at the underscore. For example, *D20\_3\_P5* can be simply written *D20*.
- Only the radioactive decay constants of the isotopes present in structure (**descdeplA2**) are recovered from the APOLIB file. The radioactive decay constants of the other isotopes are set to zero.

### 3.2.3 Mixture description structure

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

Table 13: Structure (**descmix1**)

```

MIX [ matnum ] {
  [ temp [ denmix ] ] [ NOEV ]
  [[ [ NAMALI = ] NAMISO dens [ { dil | INF } ]
   [ [ CORR ] inrs ] [ DBYE tempd ] [ SHIB NAMS ] [ THER ntfg HINC [ TCOH HCOH ] ]
   [ IRSET { gir | { PT | PTMC | PTSL } } { nir | NONE } ] [ { NOEV | SAT } ] ] ]
  |
  COMB [[ mati relvol ] ] }

```

where:

MIX	keyword to specify the number identifying the next mixture to be read.
<i>matnum</i>	mixture identifier. The maximum value that <i>matnum</i> may have is <i>nmixt</i> . When <i>matnum</i> 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.
<i>temp</i>	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.
<i>denmix</i>	mixture density in $g\ cm^{-3}$ .
<i>NAMALI</i>	<b>character*8</b> alias name for an isotope to be used locally. When the alias name is absent, the isotope name used locally is identical to the first 8-character isotope name on the library.
=	keyword to specify to which isotope in a library is associated the previous alias name.
<i>NAMISO</i>	<b>character*12</b> name of an isotope present in the library which is included in this mixture.
<i>dens</i>	isotopic concentration of the isotope <i>NAMISO</i> in the current mixture in $10^{24}cm^{-3}$ . When the mixture density <i>denmix</i> is specified, the relative weight percentage of each of the isotopes in this mixture is to be provided.
<i>dil</i>	group independent microscopic dilution cross section (in barns) of the isotope <i>NAMISO</i> in this mixture. It is possible to recalculate a group dependent dilution for an isotope by the use of the <b>SHI:</b> operator (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 a dilution of $10^{10}$ barns is to be associated with this isotope. This value represents an infinite dilution (the isotope is present in trace amounts only). It is possible to recalculate a group dependent dilution for an isotope by the use of the <b>SHI:</b> operator (see Section 3.5). In this case, the dilution is only used as a starting point for the self-shielding iterations and has no effect on the final result. If the dilution is not given an infinite dilution is assumed.



CORR	keyword to specify that the resonances of an isotope are correlated with those of other isotopes with the same <i>inrs</i> index. This option is only available with the <i>Ribon extended</i> model <sup>[7]</sup> or with the <i>subgroup projection method</i> (SPM) <sup>[7]</sup> in energy groups where this model is set. If this option is selected for an isotope, it must be set for all isotopes with the same <i>inrs</i> index. By default, the resonances of distinct isotopes are assumed to be uncorrelated.
<i>inrs</i>	index of the resonant region associated with this isotope. By default <i>inrs</i> =0 and the isotope is not a candidate for self-shielding. When <i>inrs</i> ≠0, the isotope can be self-shielded where it is assumed that a given isotope distributed with different concentrations in a number of mixtures and having the same value of <i>inrs</i> will share the same fine flux. Should we wish to self-shield both the clad and the fuel it is important to assign a different <i>inrs</i> number to each. If a single type of fuel is located in different mixture in <i>onion-peel fashion</i> , it is necessary to attribute a single <i>inrs</i> 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.
<i>tempd</i>	absolute temperature (in Kelvin) of the isotope. By default <i>tempd</i> = <i>temp</i> .
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. This option is not required if a MATXS or a DRAGLIB file is used.
NAMS	<b>character*12</b> 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 <i>NAMS</i> is different from <i>NAMISO</i> . This record must be contained in the same library file as record <i>NAMISO</i> .
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	<b>character*6</b> 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 <b>FREE</b> is used to simulate the effects of a gas.
TCOH	keyword to specify that coherent thermalization effects will be taken into account.
HCOH	<b>character*6</b> name of the coherent thermalization effects which will be taken into account. The coherent effects are the <i>vectorial reactions</i> 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.
<i>ntfg</i>	number of energy groups that will be affected by the thermalization effects.
IRSET	keyword to specify an intermediate resonance (IR) approximation or the <i>Ribon extended</i> model for some energy groups. By default, an IR approximation with the value of the Goldstein-Cohen parameter found on the library is used. If no value is found on the library, a statistical (ST) model <sup>[7]</sup> is set in all groups by default.
<i>gir</i>	imposed Goldstein-Cohen IR parameter. A Goldstein-Cohen IR parameter $0 \leq \lambda_g \leq 1$ is set in energy group <i>g</i> . A value of 1.0 stands for a statistical (ST) approximation. A value of 0.0 stands for an infinite mass (IM or WR) approximation.
PT	keyword to enable the calculation of CALENDF-type probability tables in some energy groups. The slowing-down correlated weight matrices are <i>not</i> computed. This type of probability tables is consistent with the Sanchez-Coste self-shielding method and with the subgroup projection method (SPM). <sup>[7]</sup>

PTMC	keyword to enable the calculation of CALENDF-type probability tables, similar to the PT procedure. Here, the base points of the probability tables corresponding to fission and scattering cross sections and to components of the transfer scattering matrix are also obtained using the CALENDF approach.
PTSL	keyword to enable the calculation of CALENDF-type probability tables, consistent with the Ribon extended model, in some energy groups.
nir	the intermediate resonance (IR) approximation or the Ribon extended model is imposed for energy groups with an index equal or greater than <i>nir</i> . A statistical (ST) model is set in other groups.
NONE	keyword to specify that a statistical (ST) model is set in all groups.
NOEV	keyword to force a mixture or a nuclide to be non-depleting (even in cases where it is potentially depleting). Note that the mixture or nuclide keeps its capability to produce energy. By default, the depleting isotopes are automatically regognized as depleting.
SAT	keyword to force a nuclide to be at saturation. By default, the saturation approximation is automatically set as a function of the half life and capture cross sections of the isotope.
COMB	keyword to specify that this mixture is reset with a combination of previously defined mixtures.
mati	number associated with a previously defined mixture. In order to insert some void in a mixture use <i>mati</i> =0. If the mixture is not already defined one assumes that it represents a voided mixture.
relvol	relative volume $V_i$ occupied by mixture <i>mati</i> = <i>i</i> in <i>matnum</i> . Two cases can be considered, namely that where the density $\rho_i$ of each mixture <i>mati</i> is provided along with the weight percent for each isotope $J$ ( $W_i^J$ ) and the case where the explicit concentration $N_i^J$ of each isotope in a <i>mati</i> was provided (it is forbidden to combined two mixtures with different isotopic content description). In the case where the initial mixtures are defined using densities $\rho_i$ , the density ( $\rho_k$ ) and volume ( $V_k$ ) of the final mixture will become:

$$V_k = \sum_i V_i$$

$$\rho_k = \frac{1}{V_k} \sum_i \rho_i V_i$$

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

$$W_{k,J} = \frac{\rho_i V_i W_{i,J}}{\rho_k V_k}$$

When the explicit concentration are given we will use:

$$N_{k,J} = \frac{V_i N_{i,J}}{V_k}$$

There is a very common usage of keyword COMB. In the following example, a new mixture with index 42 is defined in such a way to be identical to an existing mixture with index 25.

```
MIX 42 COMB 25 1.0
```

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-formatted library associated with file DRAGLIB. Assume also that the depletion chain, which is written on the WIMS-AECL format library associated with file WIMSLIB, states that isotope U236 (initially absent in the mixture) can be generated from U235 by neutron capture. Then, one can either specify explicitly from which library file the microscopic cross sections associated with isotope U236 (zero concentration) are to be read, or omit U236 from the mixture description in which case DRAGON will assume that the microscopic cross sections associated with isotope U236 are to be read from the same library as the cross section for isotope U235. Note that the isotopes added automatically will remain at infinite dilution.

If the SHI: module implementing the generalized Stamm'ler method is used, the self-shielding data for an isotope takes the form

```
U235      = U235  5.105E-5 1
```

where the last index indicates the self-shielding region (1 in this case).

If the USS: module implementing the subgroup method is used, additional self-shielding data is required:

- Physical probability tables are used (keyword SUBG). Consider the following data:

```
U235      = U235  5.105E-5 1 IRSET 0.0 81
```

The data “IRSET 0.0 81” indicates that a Goldstein-Cohen parameter  $\lambda_g$  equal to 0.0 is used for all energy groups with an index equal or greater than 81. A value of  $\lambda_g = 1.0$  corresponding to a statistical model is used by default.

- Mathematical probability tables (with slowing-down correlated weight matrices) are used (keyword PTSI) or mathematical probability tables with the subgroup projection method (SPM)<sup>[?]</sup> are used (keyword PT or PTMC). Consider the following data:

```
U235      = U235  5.105E-5 1 IRSET PT 5
```

The Goldstein-Cohen approximation is not used with mathematical (CALENDF) probability tables. The data “IRSET PT 5” indicates that the CALENDF probability tables are used for energy groups with an index equal or greater than 5, *with the exception of the energy groups where no Autolib data is available* and a statistical model (with physical probability tables) is used for energy groups with an index smaller than 5. A statistical model is also imposed in groups where no Autolib data is available.

The following data:

```
U235      = U235  5.105E-5 1 IRSET PT NONE
```

is useful to impose the statistical model (with physical probability tables) in all energy groups. *The option IRSET PT NONE is equivalent to the default behavior obtained using*

```
U235      = U235  5.105E-5 1
```

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

Table 14: Structure (**descmix2**)

MIX matnum [ matold ] [ relden ] [ NAMALI dens ] [ NOEV ]
---

where:

<b>MIX</b>	keyword to specify the number identifying the next mixture to be updated.
<i>matnum</i>	mixture identifier on <i>MICLIB</i> .
<i>matold</i>	mixture identifier on <i>OLDLIB</i> .
<i>relden</i>	relative density of updated mixture. The concentration of each isotope in the mixture is to be multiplied by this factor whether it comes from <i>MICLIB</i> , from <i>OLDLIB</i> or is specified explicitly using <i>dens</i> .
<b>NAMALI</b>	<b>character*8</b> alias name for an isotope on <i>MICLIB</i> to be modified.
<i>dens</i>	isotopic concentration of the isotope <i>NAMISO</i> in the current mixture in $10^{24}cm^{-3}$ . When <i>relden</i> is specified, the isotopic concentration becomes $dens \times relden$ .
<b>NOEV</b>	keyword to force a mixture to be non-depleting (even in cases where it is potentially depleting). Note that the mixture keeps its capability to produce energy.

### 3.3 The GEO: module

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

Table 15: Structure (**GEO:**)

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

where

<b>GEONAM</b>	<b>character*12</b> name of the GEOMETRY created or modified.
<b>OLDGEO</b>	<b>character*12</b> name of a read-only GEOMETRY. The type and all the characteristics of <b>OLDGEO</b> will be copied onto <b>GEONAM</b> before this later geometry is modified.
<b>(descgtyp)</b>	structure describing the geometry type of <b>GEONAM</b> (see Section 3.3.1).
<b>(descgcnt)</b>	structure describing the characteristics of a geometry (see Section 3.3.1).

#### 3.3.1 Data input for module **GEO:**

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

Table 16: Structure (**descgtyp**)

```
{ VIRTUAL |
HOMOGE |
SPHERE lr |
CAR1D lx |
CAR2D lx ly |
CAR3D lx ly lz |
TUBE lr [ lx ly ] |
TUBEX lr { lx | lx ly lz } |
TUBEY lr { ly | lx ly lz } |
TUBEZ lr { lz | lx ly lz } |
RTHETA lr lz |
```

continued on next page

Structure (**descgtyp**)

continued from last page

```

HEX lh |
HEXZ lh lz |
HEXT nhr |
HEXTZ nhr lz |
CARCEL lr [ lx ly ] |
CARCELX lr { lx | lx ly lz } |
CARCELY lr { ly | lx ly lz } |
CARCELZ lr { lz | lx ly lz } |
HEXCEL lr |
HEXCELZ lr lz |
HEXTCEL lr nhr |
HEXTCELZ lr nhr lz |
GROUP lp }

```

Table 17: Structure (**descgcnt**)

```

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

```

where

VIRTUAL	keyword to specify that a virtual geometry description follows. This type of geometry is used to complete an assembly that has irregular boundaries.
HOMOGE	keyword to specify that a infinite homogeneous geometry description follows.
SPHERE	keyword to specify that a spherical geometry (concentric spheres) description follows.
CAR1D	keyword to specify that a one dimensional plane geometry (infinite slab) description follows.
CAR2D	keyword to specify that a two-dimensional Cartesian geometry description follows.
CAR3D	keyword to specify that a three-dimensional Cartesian geometry description follows.
TUBE	keyword to specify that a cylindrical geometry (infinite tubes or cylinders) description follows. This geometry can contain an imbedded $X - Y$ Cartesian mesh.
TUBEX	keyword to specify that a polar $R - X$ cylindrical geometry description follows. This geometry can contain an imbedded $Y - Z$ Cartesian mesh.
TUBEY	keyword to specify that a polar $R - Y$ cylindrical geometry description follows. This geometry can contain an imbedded $Z - X$ Cartesian mesh.

TUBEZ	keyword to specify that a polar $R - Z$ cylindrical geometry description follows. This geometry can contain an imbedded $X - Y$ Cartesian mesh.
RTHETA	keyword to specify that a polar geometry ( $R - \theta$ ) description follows.
HEX	keyword to specify that a two-dimensional hexagonal geometry description follows.
HEXZ	keyword to specify that a three-dimensional hexagonal geometry description follows.
HEXT	keyword to specify a single 2-D hexagonal cell geometry having a triangular mesh. This option is only supported by the <b>NXT</b> : tracking module (see Section 3.4).
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 <b>NXT</b> : tracking module (see Section 3.4).
CARCEL	keyword to specify that a two-dimensional mixed Cartesian cell (concentric tubes surrounded by a rectangle) description follows. The rectangle can now be subdivided into a fine mesh when the <b>EXCEL</b> T: modules is used.
CARCELX	keyword to specify that a three-dimensional mixed Cartesian cell with tubes oriented along the $X$ -axis description follows. The three-dimensional Cartesian cell can now be subdivided into a fine mesh when the <b>EXCEL</b> T: module is used.
CARCELY	keyword to specify that a three-dimensional mixed Cartesian cell with tubes oriented along the $Y$ -axis description follows. The three-dimensional Cartesian cell can now be subdivided into a fine mesh when the <b>EXCEL</b> T: module is used.
CARCELZ	keyword to specify that a three-dimensional mixed Cartesian cell with tubes oriented along the $Z$ -axis description follows. The three-dimensional Cartesian cell can now be subdivided into a fine mesh when the <b>EXCEL</b> T: module is used.
HEXCEL	keyword to specify that a two-dimensional mixed hexagonal cell (concentric tubes surrounded by a hexagon) description follows.
HEXCELZ	keyword to specify that a three-dimensional mixed hexagonal cell with tubes oriented along the $Z$ -axis description follows.
HEXTCEL	keyword to specify a single 2-D hexagonal cell geometry having a triangular mesh and containing concentric annular regions.
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 that a <i>do-it-yourself</i> type geometry description follows.
$l_x$	number of subdivisions along the $X$ -axis (before mesh-splitting).
$l_y$	number of subdivisions along the $Y$ -axis (before mesh-splitting).
$l_z$	number of subdivisions along the $Z$ -axis (before mesh-splitting).
$l_r$	number of cylinders or spherical shells (before mesh-splitting).
$l_h$	number of hexagons in an axial plane (including the virtual hexagon).
$n_{hr}$	number of concentric hexagons in a <b>HEXT</b> , <b>HEXTZ</b> , <b>HEXTCEL</b> or <b>HEXTCELZ</b> cell (see Figure 1). This will lead to an hexagon subdivided into $6N^2$ identical triangles.
$l_p$	number of types of cells (number of cells inside which a distinct flux will be calculated) for a <i>do-it-yourself</i> type geometry.
EDIT	keyword used to modify the print level <i>iprint</i> .

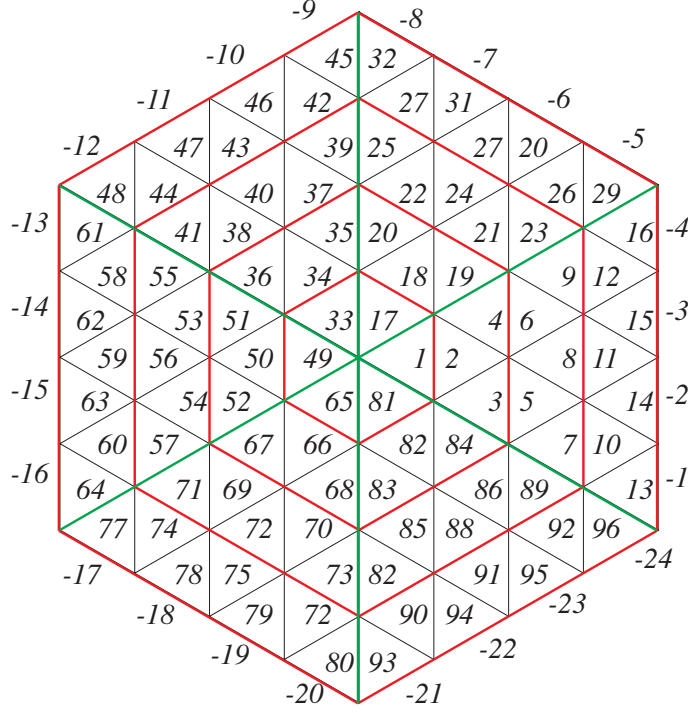


Figure 1: Hexagonal geometry with triangular mesh containing 4 concentric hexagon

<i>iprint</i>	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 allowing the boundary conditions surrounding the geometry to be treated (see Section 3.3.2).
(descSP)	structure allowing the coordinates of a geometry to be described (see Section 3.3.3).
(descPP)	structure allowing material mixtures to be associated with a geometry (see Section 3.3.4).
(descDH)	structure used to specify double-heterogeneity data (see Section 3.3.5).
(descSIJ)	structure used to specify the properties of <i>do-it-yourself</i> geometries (see Section 3.3.6).
<i>SUBGEO</i>	<b>character*12</b> name of the directory that will contain the sub-geometry.
<i>OLDGEO</i>	<b>character*12</b> name of a parallel directory containing an existing sub-geometry. The type and all the characteristics of <i>OLDGEO</i> will be copied onto <i>SUBGEO</i> .

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

### 3.3.2 Boundary conditions

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



Table 18: Structure (**descBC**)

```

[ X- { VOID | REFL | SSYM | DIAG | TRAN | SYME | ALBE { albedo | icode } | ZERO | PI/2 | PI
    | CYLI | ACYL { albedo | icode } } ]
[ X+ { VOID | REFL | SSYM | DIAG | TRAN | SYME | ALBE { albedo | icode } | ZERO | PI
    | CYLI | ACYL { albedo | icode } } ]
[ Y- { VOID | REFL | SSYM | DIAG | TRAN | SYME | ALBE { albedo | icode } | ZERO | PI/2 | PI
    | CYLI | ACYL { albedo | icode } } ]
[ Y+ { VOID | REFL | SSYM | DIAG | TRAN | SYME | ALBE { albedo | icode } | ZERO | PI
    | CYLI | ACYL { albedo | icode } } ]
[ Z- { VOID | REFL | SSYM | TRAN | SYME | ALBE { albedo | icode } | ZERO } ]
[ Z+ { VOID | REFL | SSYM | TRAN | SYME | ALBE { albedo | icode } | ZERO } ]
[ R+ { VOID | REFL | ALBE { albedo | icode } | ZERO } ]
[ HBC { S30 | SA60 | SB60 | S90 | R120 | R180 | SA180 | SB180 | COMPLETE }
{ VOID | REFL | SYME | ALBE { albedo | icode } | ZERO } ]
[ RADS [ ANG ] nrads (xrad(ir), rrad(ir) [, ang(ir) ], ir=1,nrads ) ]

```

where:

- X-/X+** keyword to specify the boundary conditions associated with the negative or positive  $X$  surface of a Cartesian geometry.
- Y-/Y+** keyword to specify the boundary conditions associated with the negative or positive  $Y$  surface of a Cartesian geometry.
- Z-/Z+** keyword to specify the boundary conditions associated with the negative or positive  $Z$  surface of a Cartesian geometry.
- R+** keyword to specify the boundary conditions associated with the outer surface of a cylindrical or spherical geometry.
- HBC** keyword to specify the boundary conditions associated with the outer surface of an hexagonal geometry.
- VOID** keyword to specify that the surface under consideration has zero re-entrant angular flux. This side is an external surface of the domain.
- REFL** keyword to specify that the surface under consideration has a reflective boundary condition. In most DRAGON calculations, this implies white boundary conditions. The main exception to this rule is when cyclic tracking in 2-D is considered and mirror like reflections are considered. A geometry is never unfolded to take into account a **REFL** boundary condition.
- SSYM** keyword to specify that the surface under consideration has a specular (or mirror) reflective boundary condition. The main difference between **REFL** and **SSYM** is that for **SSYM** the cell may be unfolded to take into account the reflection at the boundary.
- DIAG** keyword to specify that the Cartesian surface under consideration has the same properties as that associated with a diagonal through the geometry (see Figure 2). 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**

TRAN	<p>keyword to specify that the surface under consideration is connected to the opposite surface of a Cartesian domain (see Figure 3). This option provides the facility to treat an infinite geometry with translation symmetry. The only combinations of translational symmetry permitted are:</p> <ul style="list-style-type: none"> <li>• Translation along the <math>X</math>-axis X- TRAN X+ TRAN</li> <li>• Translation along the <math>Y</math>-axis Y- TRAN Y+ TRAN</li> <li>• Translation along the <math>Z</math>-axis Z- TRAN Z+ TRAN</li> </ul>
SYME	keyword to specify that the Cartesian surface under consideration is virtual and that a reflection symmetry is associated with the adequately directed axis running through the center of the cells closest to this surface (see Figure 3). Only the hexagonal geometries S30 and SA60 can be surrounded by a SYME boundary condition if a specular condition is to be applied on this boundary.
ALBE	keyword to specify that the surface under consideration has an arbitrary albedo. This side is an external surface of the domain.
<i>albedo</i>	geometric albedo corresponding to the boundary condition ALBE ( <i>albedo</i> >0.0).
<i>icode</i>	index of a physical albedo corresponding to the boundary condition ALBE. The numerical values of the physical albedo are supplied by the operator MAC: (see Section 3.1).
ZERO	keyword to specify that the surface under consideration has a zero-flux boundary condition. This side is an external surface of the domain.
PI/2	keyword to specify that the surface under consideration has a $\pi/2$ rotational symmetry (see Figure 3). The only $\pi/2$ symmetry permitted is related to sides (X- and Y-). This condition can be combined with a translation boundary condition:(PI/2 X- TRAN X+) and/or (PI/2 Y- TRAN Y+) (see Figure 4).
PI	keyword to specify that the surface under consideration has a $\pi$ rotational symmetry (see Figure 3). This keyword is useful for representing a Cartesian checkerboard pattern as shown in Fig. 5.
CYLI	the side under consideration has a zero incoming current boundary condition with a circular correction applied on the Cartesian boundary. This option is only available in the $X$ - $Y$ plane for CAR2D and CAR3D geometries defined for TRIVAC full-core calculations.
ACYL	the side under consideration has an arbitrary albedo with a circular correction applied on the Cartesian boundary. This option is only available in the $X$ - $Y$ plane for CAR2D and CAR3D geometries defined for TRIVAC full-core calculations.
S30	keyword to specify an hexagonal symmetry of one twelfth of an assembly (see Figure 6).
SA60	keyword to specify an hexagonal symmetry of one sixth of an assembly of type A (see Figure 6).
SB60	keyword to specify an hexagonal symmetry of one sixth of an assembly of type B (see Figure 7).
S90	keyword to specify an hexagonal symmetry of one quarter of an assembly (see Figure 7).
R120	keyword to specify a rotation symmetry of one third of an assembly (see Figure 8).

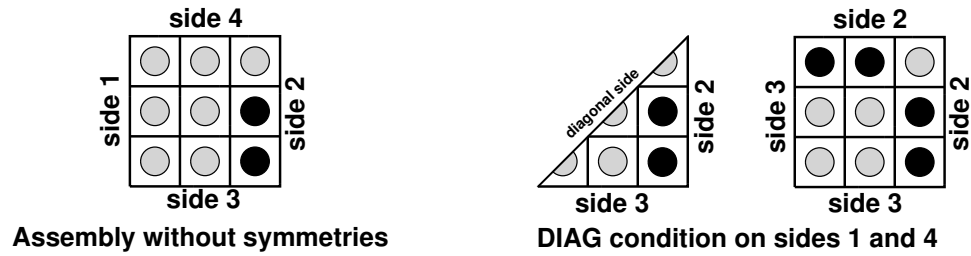


Figure 2: Diagonal boundary conditions in Cartesian geometry

R180	keyword to specify a rotation symmetry of a half assembly (see Figure 8).
SA180	keyword to specify an hexagonal symmetry of half a type A assembly (see Figure 9).
SB180	keyword to specify an hexagonal symmetry of half a type B assembly (see Figure 10).
COMPLETE	keyword to specify a complete hexagonal assembly (see Figure 11).
RADS	This key word is used to specify the cylindrical correction applied in the $X - Y$ plane for CAR2D and CAR3D geometries. <sup>[?]</sup>
ANG	This key word allows the angle (see Figure 12) of the cylindrical notch to be set. By default, no notch is present.
<i>nrads</i>	Number of different corrections along the cylinder main axis (i.e. the $Z$ axis).
<i>xrad(ir)</i>	Coordinate of the $Z$ axis from which the correction is applied.
<i>rrad(ir)</i>	Radius of the real cylindrical boundary.
<i>ang(ir)</i>	Angle of the cylindrical notch. This data is given if and only if the key word <b>ANG</b> is present. $ang(ir) = \frac{\pi}{2}$ by default (i.e. the correction is applied at every angle).

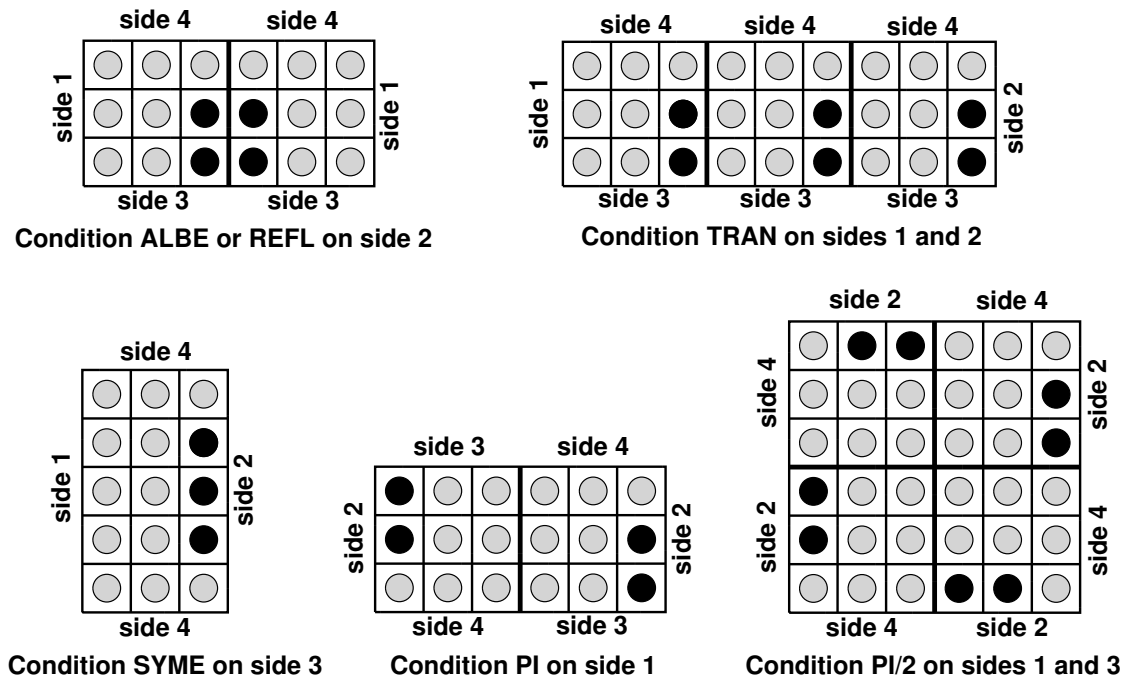


Figure 3: Various boundary conditions in Cartesian geometry

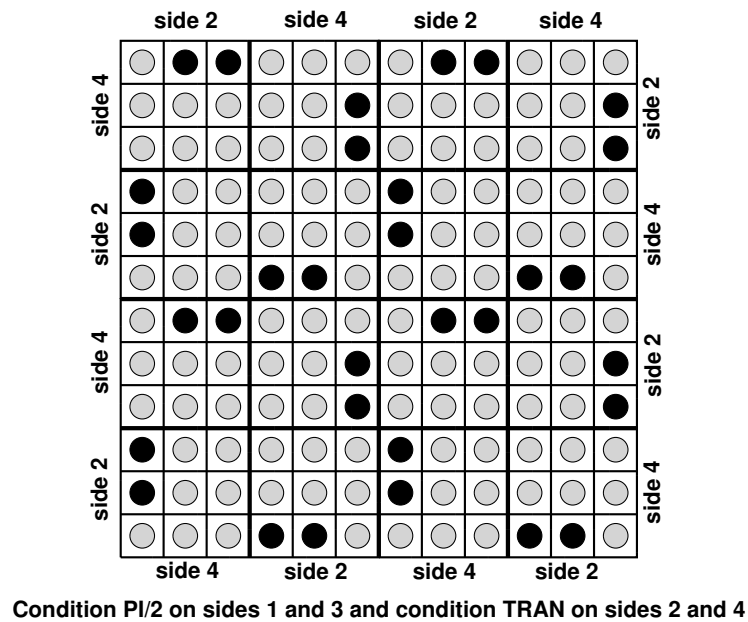
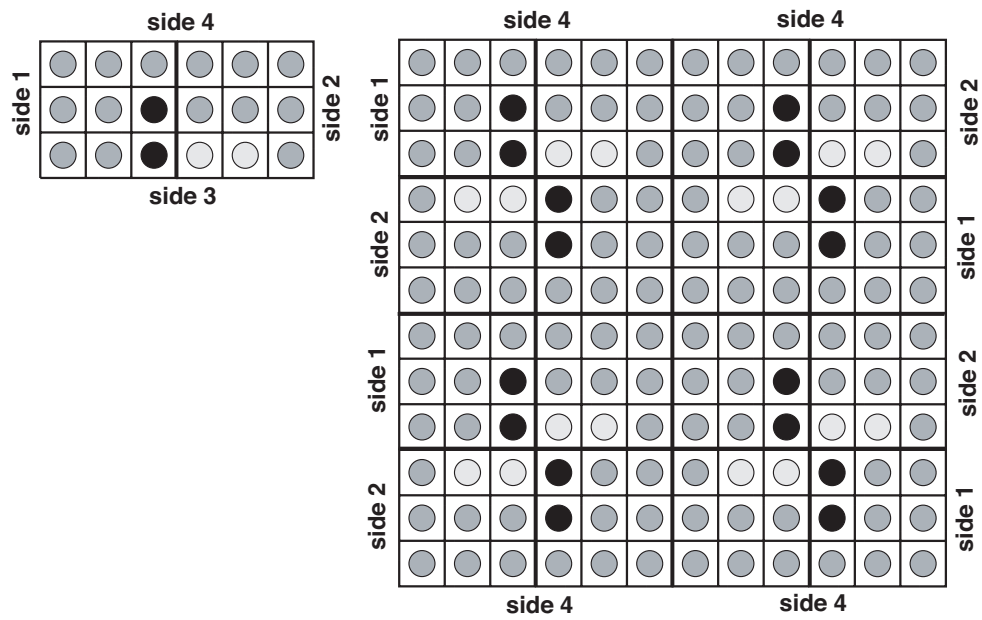


Figure 4: Translation/rotation boundary conditions in Cartesian geometry



Condition PI on side 3 and TRAN on sides 1, 2 and 4

Figure 5: Representing a checkerboard in Cartesian geometry

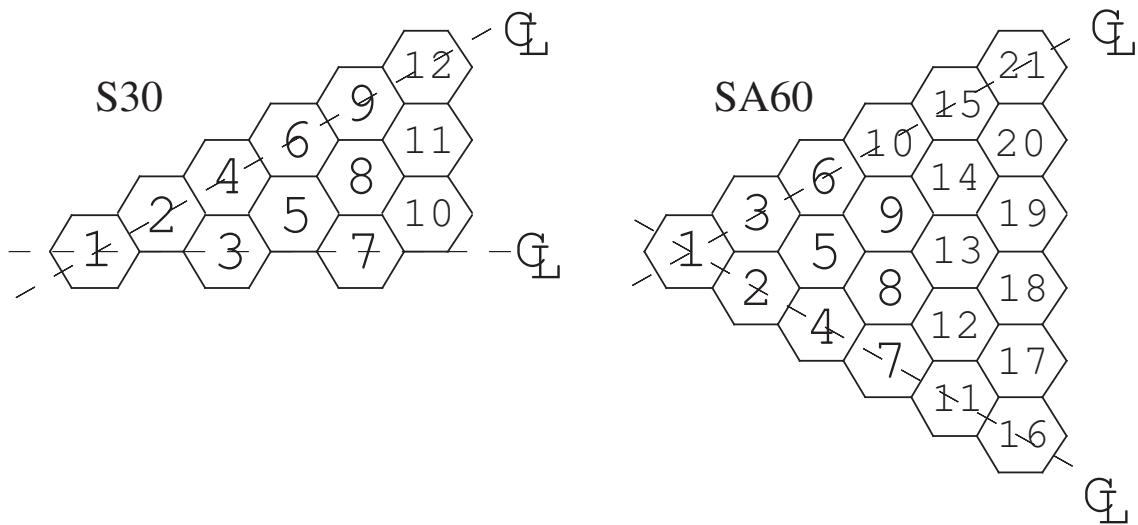


Figure 6: Hexagonal geometries of type S30 and SA60

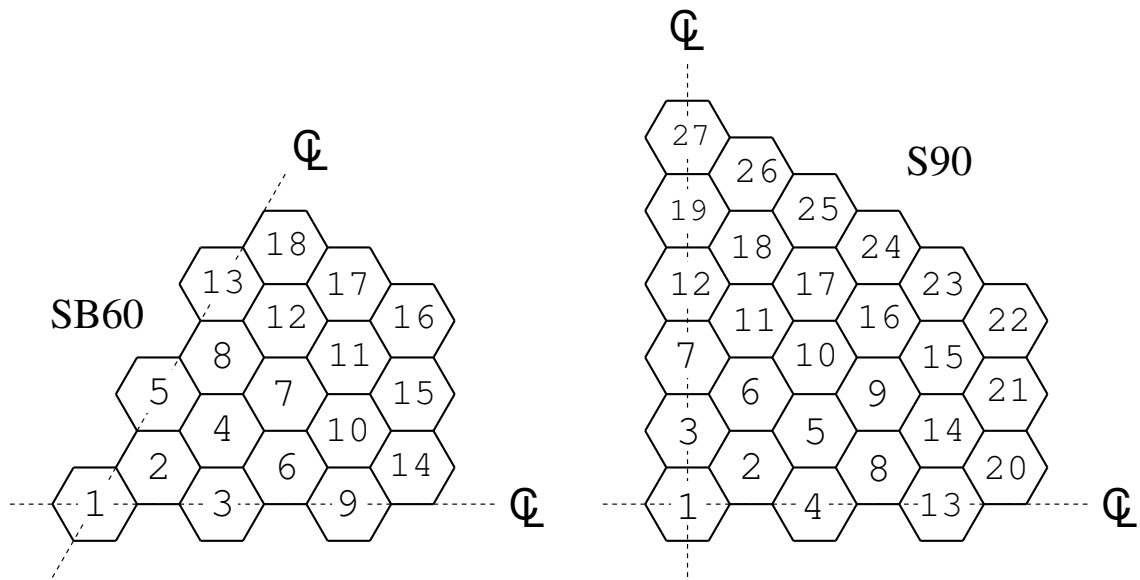


Figure 7: Hexagonal geometries of type SB60 and S90

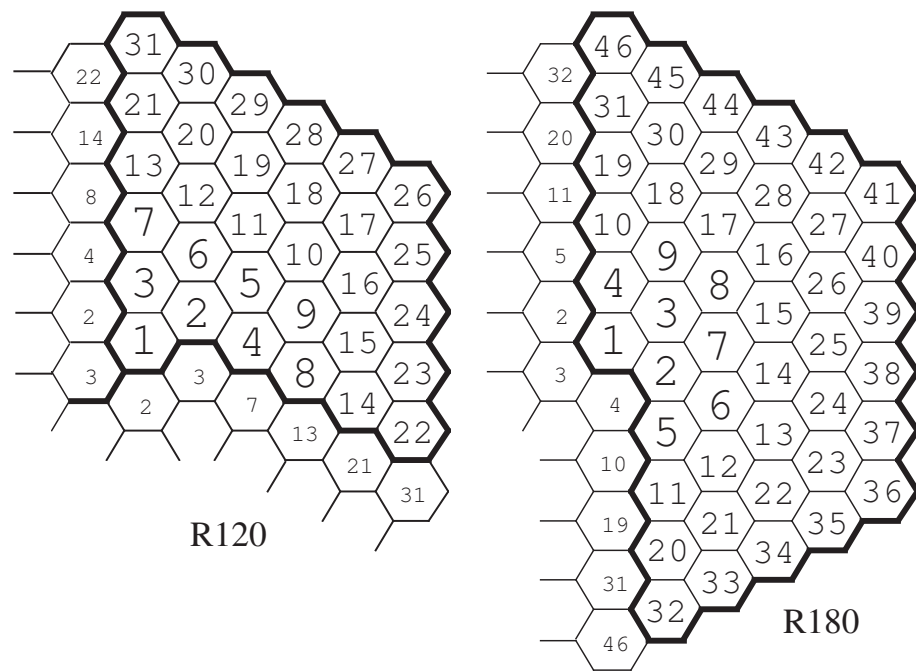


Figure 8: Hexagonal geometries of type R120 and R180

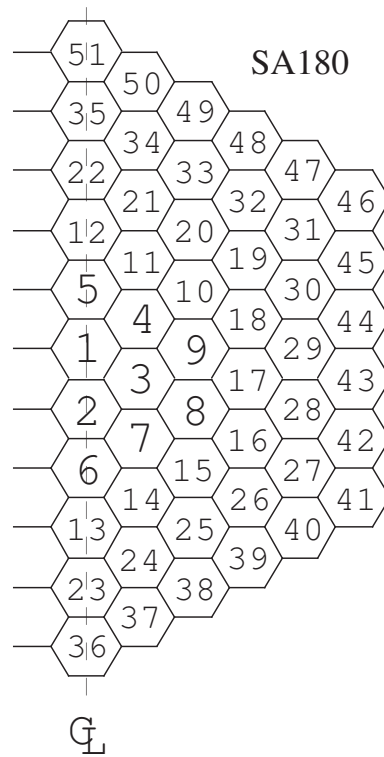


Figure 9: Hexagonal geometry of type SA180

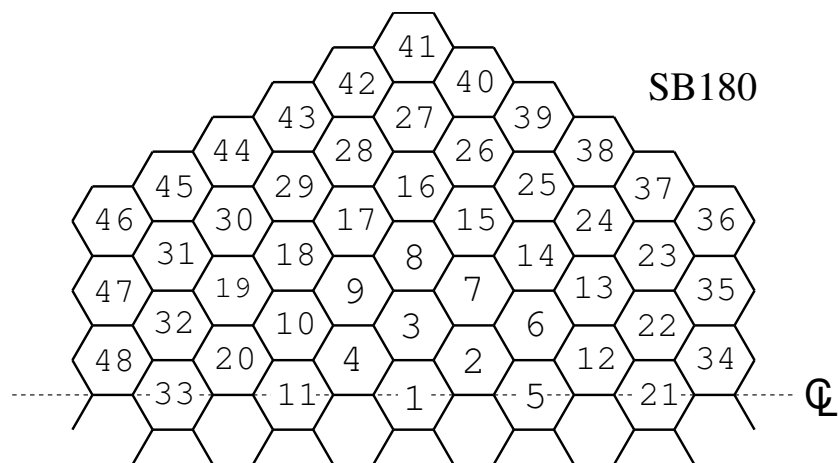


Figure 10: Hexagonal geometry of type SB180

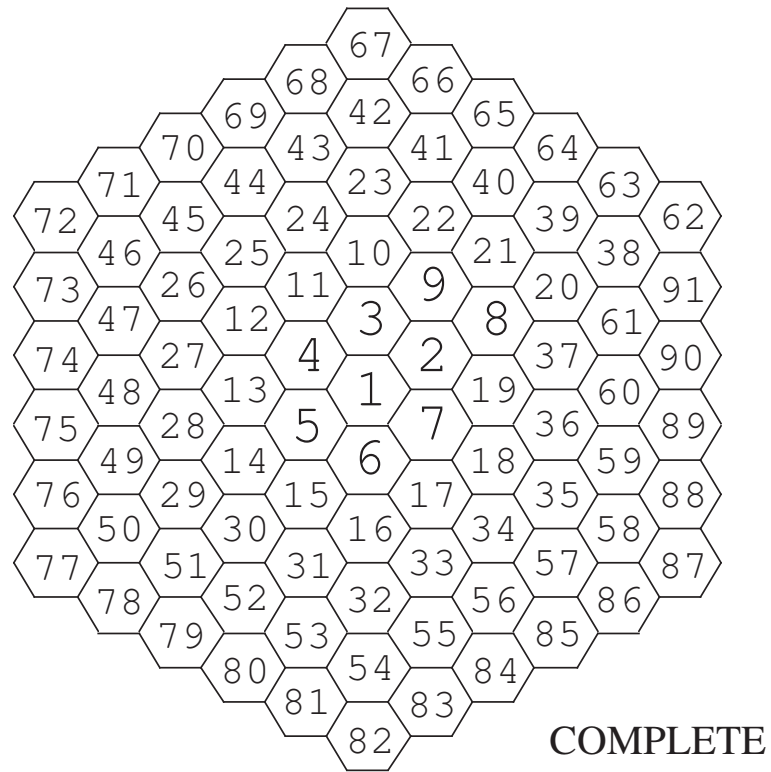


Figure 11: Hexagonal geometry of type COMPLETE

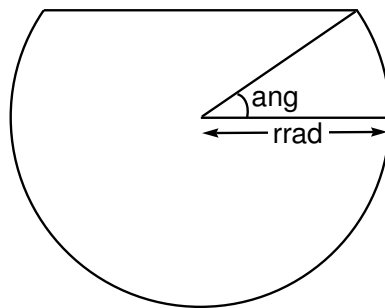


Figure 12: Cylindrical correction in Cartesian geometry



### 3.3.3 Spatial properties of geometry

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

Table 19: Structure (**descSP**)

```
[ MESHX (xxx(i), i=1,lx+1) ]
[ SPLITX (ispltx(i), i=1,lx) ]
[ MESHY (yyy(i), i=1,ly+1) ]
[ SPLITY (isply(i), i=1,ly) ]
[ MESHZ (zzz(i), i=1,lz+1) ]
[ SPLITZ (ispltz(i), i=1,lz) ]
[ RADIUS (rrr(i), i=1,lr+1) ]
[ OFFCENTER (disxyz(i), i=1,3) ]
[ SPLITR (ispltr(i), i=1,lr) ]
[ SECT isect [ jsect ] ]
[ 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 } ]
```

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.
isply	array giving the number of zones that will be considered for each region along the $Y$ -axis. By default, $isply=1$ unless a diagonal symmetry is used in which case $isply=ispltx$ .
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.
<i>ispltz</i>	array giving the number of zones that will be considered for each region along the $Z$ -axis. By default, <i>ispltz</i> =1.
RADIUS	keyword to specify the spatial mesh along the radial direction.
<i>rrr</i>	array giving the radial limits (cm) of the annular regions (cylindrical or spherical) making up the geometry. It is used for the following geometries: TUBE, TUBEZ, SPHERE), CARCEL, CARCELX, CARCELY, CARCELZ, HEXCEL and HEXCELZ. It is important to note that we must have <i>rrr</i> (1)=0.0. The other values of <i>rrr</i> ( <i>i</i> ) in a CARCEL- or HEXCEL-type geometry are defined as shown in Figure 13.
OFFCENTER	keyword to specify that the concentric annular regions in a CARCEL, CARCELX, CARCELY, CARCELZ, TUBE, TUBEZ, TUBEY and TUBEZ geometry can now be displaced with respect to the center of the Cartesian mesh. This option will only be treated when the EXCELT:, NXT: and EXCELL: modules are used.
<i>disxyz</i>	array giving the $x$ ( <i>disxyz</i> (1)), $y$ ( <i>disxyz</i> (2)) and $z$ ( <i>disxyz</i> (3)) displacement (cm) of the concentric annular regions with respect to the center of the Cartesian mesh.
SPLITR	keyword to specify that a mesh splitting of the geometry along the radial direction is to be performed.
<i>ispltr</i>	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, <i>ispltr</i> =1.
SECT	keyword to specify the type of sectorization for a Cartesian or hexagonal cell. In hexagonal geometry, this keyword is expected to be defined near the SIDE keyword. By default, no sectorization is performed.
<i>isect</i>	sectorization index, defined as $isect = \begin{cases} -999 : & \text{non-sectorized cell processed as a sectorized cell} \\ -1 : & \text{\(\times\)-type sectorization} \\ 0 : & \text{non-sectorized cell} \\ 1 : & \text{\(+\)-type sectorization} \\ 2 : & \text{simultaneous \(\times\)- and \(+\)-type sectorization} \\ 3 : & \text{simultaneous \(\times\)- and \(+\)-type sectorization shifted by } 22.5^\circ \\ 4 : & \text{windmill sectorization.} \end{cases}$
<i>jsect</i>	number of embedded tubes that are <i>not</i> sectorized, with $0 \leq jsect \leq lr$ . By default, <i>jsect</i> = 0. Examples of sectorization options are depicted in Figs. 14 and 15.
SIDE	keyword to specify the length of a side of a hexagon.
<i>sideh</i>	length of one side of a hexagon (cm).
<i>hexmsh</i>	triangular mesh for HEXT, HEXTCEL, HEXTZ and HEXTCELZ hexagonal geometries. By default, <i>hexmsh</i> = <i>sideh</i> / <i>nhr</i> . When <i>hexmsh</i> 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 <i>sideh</i> (see Figure 16).
SPLITH	keyword to specify that a triangular mesh splitting level for HEXT, HEXTCEL, HEXTZ and HEXTCELZtype geometries is to be considered. This is valid only if <i>nhr</i> =1.

<i>isplth</i>	value of the triangular mesh splitting. Its use is similar to <i>nh</i> 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.
<i>npins</i>	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.

<i>dpins</i>	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.
<i>rpins</i>	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 <i>rpins</i> , 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.
<i>apins</i>	the angle (radian) of the first pin in the ring (only one value provided for <i>apins</i> , the 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.
<i>xpins</i>	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.
<i>ypins</i>	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.
<i>zpins</i>	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  $L_{\text{zones}}$  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

Figure 13: Definition of the radii in a CARCEL- or HEXCEL-type geometry

$L_x$ ,  $L_y$ ,  $L_z$ ,  $L_r$ ,  $L_h$  and  $L_t$  as:

$$\begin{aligned}
 L_x &= \sum_{i=1}^{l_x} ispltx(i) \\
 L_y &= \sum_{i=1}^{l_y} isply(i) \\
 L_z &= \sum_{i=1}^{l_z} isplz(i) \\
 L_r &= \sum_{i=1}^{l_r} |isplr(i)| \\
 L_h &= lh \\
 L_t &= \begin{cases} 6 \times nhr^2 & \text{if } nhr \leq 1 \\ 6 \times isplth^2 & \text{otherwise} \end{cases}
 \end{aligned}$$

and  $L_{\text{zones}}$  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.

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

- 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.

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

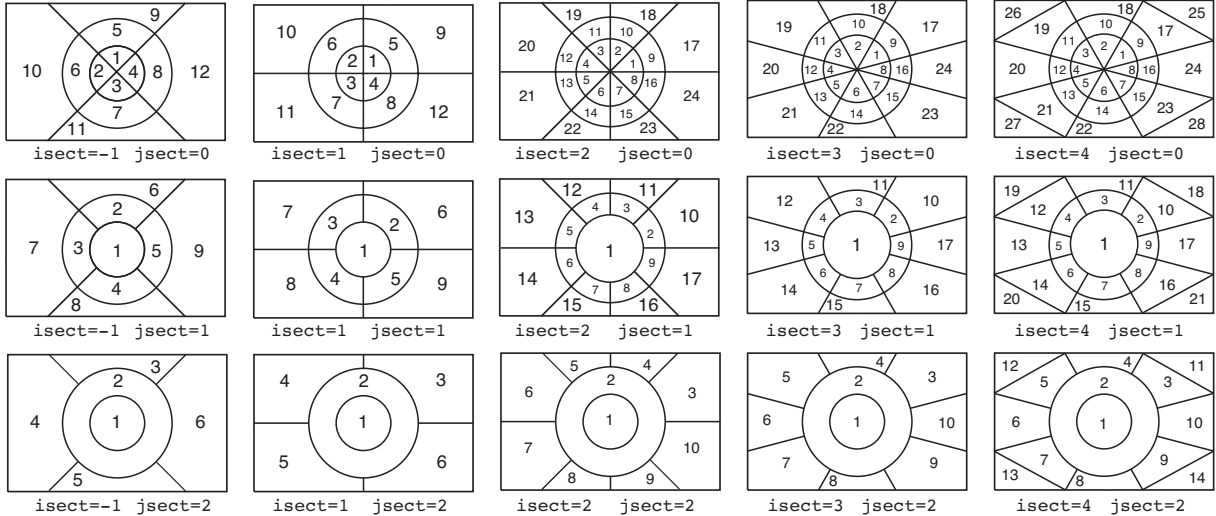


Figure 14: Numerotation of the sectors in a Cartesian cell

### 3.3.4 Physical properties of geometry

In addition to specifying the mixture associated with each region in the geometry, the (**descPP**) structure is also used to provide information on the sub-geometry required in this geometry. For example,

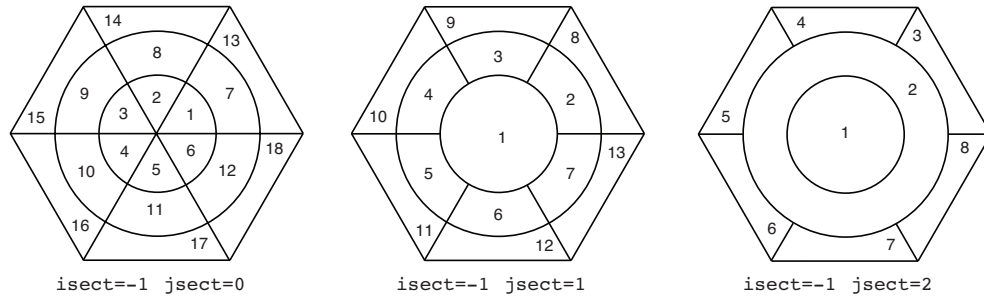


Figure 15: Numerotation of the sectors in an hexagonal cell

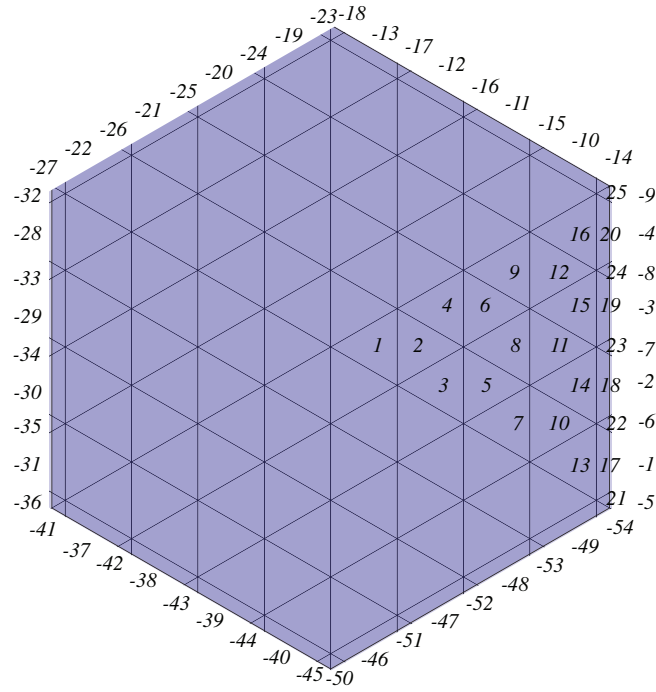


Figure 16: Hexagonal geometry with triangular mesh that extends past the hexagonal boundary

an optional procedure in DRAGON groups together regions so as to reduce the number of unknowns *maxreg* in the flux calculation. In this way, only the merged regions contribute to the cost of the calculation. However, the following points must be considered:

1. All the cells belonging to the same merged region must have the same nuclear properties and dimensions.
2. The grouping procedure is based on the approximation that all the regions belonging to the same

merged region share the same flux.

3. The merging can also take into account region orientation (by a rotation and/or transposition) before they are merged. This procedure facilitates the merging of regions when a **DIAG** or **SYME** boundary condition is used.

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

Table 20: Structure **(descPP)**

```
[ MIX { (imix(i),i=1,nt) [ REPEAT ] |
  [[ PLANE iplan { (imix(i),i=1,lp) | SAME iplan1
    | [[ CROWN { (imix(i),i=1,lc) | ALL jmix | SAME iplan1 } ]]
    | [[ UPTO ic ALL jmix | SAME iplan1 } ]] ] ] }
]
[ HMIX (ihmix(i), i=1,Nt) [ REPEAT ] ]
[ CELL (HCELL(i),i=1,Nt) ]
[ MERGE (imerge(i),i=1,Nt) ]
[ TURN (HTURN(i),i=1,Nt) ]
[ CLUSTER (NAMPIN(i),i=1,Np) ]
[ MIX-NAMES (NAMMIX(i),i=1,maxmix) ]
```

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 mixtures are then given in the following order

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

- **TUBE** geometry.

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

The mixtures are then given in the following order

1. radially outward ( $l = 1, lr$ ) and such that *imix* 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 mixtures are then given in the following order



1. radially outward ( $l = 1, lr$ ) and such that *imix* 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 mixtures are then given in the following order

1. radially outward ( $l = 1, lr$ ) and such that *imix* 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 mixtures are then given in the following order

1. radially outward ( $l = 1, lr$ ) and such that *imix* 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 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 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 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 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 mixtures are then given in the following order

1. radially outward ( $l = 1, lr$ ) and such that *imix* 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 mixtures or 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 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 mixtures or 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 mixtures or 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 mixtures are then given in the following order

1. radially outward ( $l = 1, lr$ ) and such that *imix* 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 mixtures are then given in the following order

1. radially outward ( $l = 1, lr$ ) and such that *imix* 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 mixtures are then given in the following order

1. radially outward ( $l = 1, lr$ ) and such that *imix* 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 mixtures or cells are then given in the order provided in Figure 6 to 11.

- 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  $l$  ( $l = 1, 2 \times nhr - 1$ ) 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 16 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 mixtures are then 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 hexagonal region.

- HEXZ geometry.

$$N_t = lh \times lz$$

The mixtures or cells are then given in the following order

1. according to Figure 6 to 11 for plane  $k$ ;
2. from surface Z- to surface Z+ ( $k = 1, lz$ ).

- 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 16 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  $j = 1, 6$ .

- 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 16 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.

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

- 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 16 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 mixtures are then 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 surface Z- to surface Z+ ( $k = 1, lz$ ).

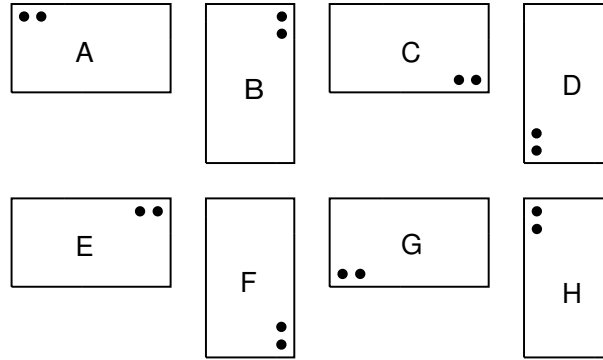


Figure 17: Description of the various rotations allowed for Cartesian geometries

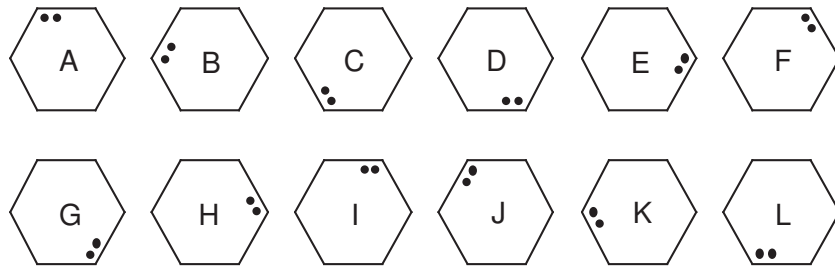


Figure 18: Description of the various rotation allowed for hexagonal geometries

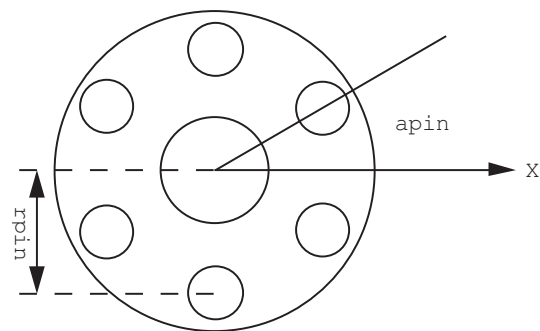


Figure 19: Typical cluster geometry

The inputs associated with this structure have the following meaning:

MIX	keyword to specify the isotopic mixture number or sub-geometry associated with each region inside the geometry. When diagonal symmetries are considered, only the mixture associated with regions inside the symmetrized geometry need to be specified. When a sub-geometry is located inside symmetrized geometry but outside the calculation region it must be declared <i>virtual</i> (for example, the corners of a nuclear reactor).
imix	array of $n_t \leq N_t$ integers or character variables associated with each region. An integer is a mixture number associated with a region $imix \leq maxmix$ (see Sections 3.1 and 3.2). If $imix=0$ , the corresponding volume is replaced by a void region. If $imix$ is a character variable, it is replaced by the corresponding sub-geometry or <i>generating cell</i> . These values must be specified in the following order for most geometries:

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

In the cases where a CARCELX and a TUBEX geometry are defined then we will use

1. radially from the inside out ( $lr + 1$  mixtures for CARCELX and  $lr$  for TUBEX).
2. from surface Y- to surface Y+
3. from surface Z- to surface Z+
4. from surface X- to surface X+

Finally, for a CARCELY and TUBEY geometry are defined the following order is considered:

1. radially from the inside out ( $lr + 1$  mixtures for CARCELY and  $lr$  for TUBEY)
2. from surface Z- to surface Z+
3. from surface X- to surface X+
4. from surface Y- to surface Y+

In the cases where a sectorized cell geometry is defined, *imix* must be defined in each sector, following the order shown in Figure 14 and 15. Also note that *imix* is *not affected* by the values of the mesh-splitting indices *ispltx*, *isplty*, *ispltz* or *ispltr*.

REPEAT	keyword to specify the previous list of mixtures 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 ( $imix(i) = 0$ ).
PLANE	keyword to attribute mixture numbers to each volume inside a single 2-D plane. This option is valid only for 3-D geometries, Cartesian or hexagonal.
iplan	plane number for which material mixture are input.
SAME	keyword to attribute the same material mixture numbers of the <i>iplan1</i> plane to the <i>iplan</i> plane. In hexagonal geometry, it can indicate that the mixture numbers of the current crown of the <i>iplanth</i> plane will be identical to those of the same crown of the <i>iplan1</i> th plane.
iplan1	plane number used as reference to input the current plane or crown(s).
lp	number of volumes in a plane. In Cartesian geometry, $lp = lx * ly$ and in hexagonal geometry, $lp = lh$ .



CROWN	keyword to attribute mixture numbers to each hexagon of a single crown. This option is only valid for COMPLETE hexagonal geometry definition. Each use of the keyword CROWN increases the crown number by 1. So it is not required to give its number, but crowns must be defined from the center to the peripheral regions of a plane.
<i>lc</i>	number of hexagons in the current crown. For the <i>i</i> th crown of a complete hexagonal plane, $lc = (i - 1) * 6$ . The first crown is composed of only one hexagon.
ALL	keyword to specify that the <i>lc</i> material mixture number of the current crown have the same value <i>jmix</i> .
UPTO	keyword to attribute material mixture numbers of the current crown up to the <i>ic</i> one.
<i>ic</i>	number of the last crown in UPTO option. Its value must be greater than equal to the current crown number.
HMIX	keyword to specify the virtual isotopic mixture associated with each region inside the geometry. These virtual mixtures will be produced by homogenization in the EDI: module (see Section 3.9.1).
CELL	keyword to specify the location of the sub-geometry called <i>generating cells</i> in a Cartesian or hexagonal geometry.
HCELL	array of sub-geometry <b>character*12</b> names which will be superimposed upon the current Cartesian geometry. The same sub-geometry may appear in different positions within the global geometry if the material properties and dimensions are identical. The concept of sub-geometry is useful for the interface current method in a SYBIL calculation since the collision probability matrix associated with each sub-geometry is computed independently of its location in the geometry. In general, the neutron fluxes in identical sub-geometry located at different locations will be different even if they are associated with the same collision probability matrix. These sub-geometry names must be specified in the following order: <ol style="list-style-type: none"> <li>1. from surface X- to surface X+</li> <li>2. from surface Y- to surface Y+</li> <li>3. from surface Z- to surface Z+</li> </ol>
MERGE	keyword to specify that some sub-geometries or regions must be merged.
<i>imerge</i>	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 <b>character*1</b> 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 17 while for hexagonal geometries the permitted orientations are shown in Figure 18. For 3-D cells, the same letters can be used to describe the rotation in the <i>X – Y</i> plane. However, an additional – sign can be glued to the 2-D rotation identifier to indicate reflection of the cell along the <i>Z</i> -axis (-A to -L).
CLUSTER	keyword to specify that pin (cylindrical) sub-geometry will be inserted in the geometry (see Figure 19).
NAMPIN	array of cylindrical sub-geometry <b>character*12</b> name representing a pin. This sub-geometry must be of type TUBE, TUBEX, TUBEY or TUBEZ.

- MIX-NAMES** keyword to specify character names to material mixtures. By default, the material mixtures are not named.
- NAMMIX** array of **character\*12** names for the material mixtures.

### 3.3.5 Double-heterogeneity

The structure (**descDH**) provides the possibility to define a stochastic mixture of cylindrical or spherical micro-structures that can be distributed inside *composite mixtures* of the current *macro-geometry*. A composite mixture is represented by a *material mixture index* with a value greater than *maxmix*, the maximum number of real mixtures. Each micro-structure can be composed of many micro-volumes.<sup>[?]</sup>

Table 21: Structure (**descDH**)

```
[ 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) ]
```

where

BIHET	keyword to specify that the current macro-geometry is containing composite mixtures.
TUBE	keyword to specify that the micro-structures are of a cylindrical geometry;
SPHE	keyword to specify that the micro-structures are of a spherical geometry.
nmistr	maximum number of micro-structure types in the composite mixtures. 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 composite mixtures. This is the number of material mixture indices of the macro-geometry with a value $> \text{maxmix}$ .
ns	array giving the number of sub-regions (tubes or spherical shells) in the micro-structures. Each type of micro-structures may contain a different number of micro-volumes.
rs	array giving the radius of the tubes or spherical shells making up the micro-structures. For each type of micro structure $i$ , we will have an initial radius of $rs(1, i) = 0.0$ .
milie	array giving the indices used to defined composite mixtures in the macro-geometry. These composite mixture indices must be $> \text{maxmix}$ .
mixdil	array giving the mixture indices associated with the diluent in each composite mixtures of the macro-geometry. These values must be $\leq \text{maxmix}$ .
fract	array of volumetric concentration ( $V_G/V_R$ ) of each micro-structures (volume $V_G$ ) in a given region (volume $V_R$ ) of the macro-geometry.
mixgr	array giving the mixture index associated with each region of the micro-structures. Note that <i>mixgr</i> should be specified only for the regions of the micro-structure which have a concentration <i>fract</i> $>0$ . These values must be $\leq \text{maxmix}$ .

Examples of geometry definitions can be found in Section [7.2](#).

### 3.3.6 Do-it-yourself geometries

A *do-it-yourself* geometry is an abstract representation of an assembly of arbitrary unit-cells defined in term of their probability of presence and of their probability to have a particular neighbor. Structure (**descSIJ**) is defined as

Table 22: Structure (**descSIJ**)

```
[ POURCE (pcinl(i),i=1,lp) ]
[ PROCEL ((pijcel(i,j),j=1,lp),i=1,lp) ]
```

where

**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.<sup>[?]</sup> This option permits the interactions between different arbitrarily 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 cell type in the lattice such that:

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

**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. If the **POURCE** structure is given without the **PROCEL** structure, a *statistical* approximation is used, as defined in Ref. ?.

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

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

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

Examples of geometry definitions can be found in Section 7.2.

### 3.4 The tracking modules

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

Many different operators are available for tracking in DRAGON. The **SYBILT**: module is used for 1-D geometries (either plane, cylindrical or spherical) and interface current tracking inside heterogeneous blocks. The **EXCELT**: module is used to perform full cell collision probability tracking with isotropic<sup>[?, ?]</sup> or specular<sup>[?, ?]</sup> surface current. The **NXT**: module is an extension of the **EXCELT**: module to more complex geometry including assemblies of clusters in two and three dimensions.<sup>[?]</sup> The **MCCGT**: module is an implementation of the open characteristics method of I. R. Suslov.<sup>[?, ?]</sup> These are the transport tracking modules which can be used everywhere in the code where tracking information needs to be generated. The **SNT**: module is an implementation of the discrete ordinates (or  $S_N$ ) method in 1-D/2-D/3-D geometries. The module **BIVACT**: is used to perform a finite-element (diffusion or  $SP_n$ ) 1-D/2-D tracking which may be required for diffusion synthetic acceleration (DSA) or homogenization purposes.<sup>[?]</sup> The final module **TRIVAT**: is used to perform a finite-element 1-D/2-D/3-D tracking which may be required for DSA or homogenization purposes.<sup>[?]</sup>

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

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

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

- the different rings of a cylindrical or spherical region starting with the inner most after mesh splitting;
- for a cluster regions located in a ring, two different numbering schemes are possible. For the **EXCELT**: module, one first numbers the region inside the pin in the same way as for cylindrical regions and finishes by associating the next region number to the shell of the global geometry which contains this pin. If two cluster types are located in a given ring, they are classified according to increasing  $rp_{in}$  and  $ap_{in}$  and then numbered in this order. Cluster overlapping annular region are numbered before considering the annular regions. For the **NXT**: module, each pin is numbered individually in a Cartesian region according to their ordered in the **CLUSTER** keywords and then the Cartesian regions are numbered sequentially. A description of the explicit numbering of regions and surfaces can be found in report IGE-260.<sup>[?]</sup>
- the zones in ascending order corresponding to the first axial component (normally  $X$ ) after mesh splitting;
- the zones in ascending order corresponding to the second axial component (normally  $Y$ ) after mesh splitting;
- the hexagonal zones corresponding to the order described in Figure 6 to Figure 11.
- the sub-geometry of type **CARCELX**, **CARCELY** and **CARCELZ** are numbered assuming that the third component corresponds to  $X$ ,  $Y$  and  $Z$  respectively.

We should also note that symmetry conditions implicitly force the grouping of certain calculation zones.

All the tracking operators of DRAGON share an identical general tracking data structure defined as

Table 23: Structure (**desctrack**)

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

with

EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index used to control the printing of this operator. The amount of output produced by this tracking operators will vary substantially depending on the print level specified. For example, <ul style="list-style-type: none"> <li>• when <i>iprint</i>=0 no output is produced;</li> <li>• when <i>iprint</i>=1 a minimum amount of output is produced; the main geometry properties are printed (fixed default option);</li> <li>• when <i>iprint</i>≥2 In addition to the information printed when using <i>iprint</i>=1 the zone numbering (zones associated with a flux) is printed;</li> </ul>
TITL	keyword which allows the run title to be set.
<i>TITLE</i>	the title associated with a DRAGON run. This title may contain up to 72 characters. The default when TITL is not specified is no title.
MAXR	keyword which permits the maximum number of regions to be considered during a DRAGON run to be specified.
<i>maxreg</i>	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.
NORE	keyword to specify that the automatic normalization of the integration lines is deactivated.
RENO	keyword to specify the activation of the <i>direction-independent</i> normalization procedure of the integration lines. The normalization factors are <i>not</i> function of the subtracks directions. This option is only valid for modules NXT:, EXCELT: and SALT:. This is the default option for NXT: and SALT: modules.
REND	keyword to specify the activation of the <i>direction-dependent</i> normalization procedure of the integration lines. The normalization factors are function of the subtracks directions. This option is only valid for modules NXT:, EXCELT: and SALT:. This is the default option for EXCELT: module.

### 3.4.1 The SYBILT: tracking module

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

1. The homogeneous geometry HOMOGE.
2. The one-dimensional geometries SPHERE, TUBE and CAR1D.<sup>[?]</sup>
3. The two-dimensional geometries CAR2D and HEX including respectively CARCEL and HEXCEL sub-geometries as well as VIRTUAL sub-geometries.
4.  $S_{ij}$ -type two-dimensional non-standard geometries.<sup>[?]</sup>
5. The double heterogeneity option.<sup>[?]</sup>

The calling specification for this module is:

Table 24: Structure (SYBILT:)

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

where

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

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

(desctrack)    structure describing the general tracking data (see Section 3.4)

(descsybil)    structure describing the transport tracking data specific to SYBILT:.

The SYBILT: specific tracking data in (descsybil) is defined as

Table 25: Structure (descsybil)

```
[ MAXJ maxcur ] [ MAXZ maxint ]
[ HALT ]
[ QUA1 iqua1 ] [ QUA2 iqua2 nsegment ] [ { EQW | GAUS } ]
[ { ROTH | ROT+ | DPO0 | DPO1 } ]
[ { WIGN | ASKE | SANC } ] [ LIGN ] [ RECT ]
[ [ QUAB iquab ] [ { SAPO | HEBE } ] ]
;
```

where

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

<i>maxcur</i>	the maximum number of interface currents surrounding the blocks. The default value is $maxcur = \max(18, 4 \times maxreg)$ for the SYBILT: module.
MAXZ	keyword to specify the maximum amount of memory required to store the integration lines. An insufficiently large value can lead to an execution failure (core dump).
<i>maxint</i>	the maximum amount of memory required to store the integration lines. The default value is $maxint = 10000$ .
HALT	keyword to specify that the program is to be stopped at the end of the geometry calculations. This option permits the geometry inputs to be checked, the number of blocks and interface currents to be calculated, and a conservative estimate of the memory required for storing the tracks to be made for mixed geometries.
QUA1	keyword to specify the one-dimensional integration parameters.
<i>iqua1</i>	number of basis points for the angular integration of the blocks in a one-dimensional geometry. This parameter is not used for CAR1D geometries. If a Gauss-Legendre or Gauss-Jacobi quadrature is used, the values of <i>iqua1</i> allowed are: 1 to 20, 24, 28, 32 or 64. The default value is $iqua1 = 5$ .
QUA2	keyword to specify the two-dimensional integration parameters.
<i>iqua2</i>	number of basis points for the angular integration of the blocks in a two-dimensional geometry appearing during assembly calculations. If a Gauss-Legendre or Gauss-Jacobi formula is used the values allowed for <i>iqua2</i> are: 1 to 20, 24, 28, 32 or 64. The default value is $iqua2 = 3$ and represents the number of angles in $(0, \pi/4)$ for Cartesian geometries and $(0, \pi/6)$ for hexagonal geometries.
<i>nsegment</i>	number of basis points for the spatial integration of the blocks in a two-dimensional geometry appearing during assembly calculations. The values of <i>nsegment</i> allowed are: 1 to 10. The default value is $nsegment = 3$ .
EQW	keyword to specify the use of equal-weight quadrature.
GAUS	keyword to specify the use of the Gauss-Legendre or the Gauss-Jacobi quadrature. This is the default option.
ROTH	keyword to specify that the isotropic ( $DP_0$ ) components of the inter-cell current is used with the incoming current being averaged over all the faces surrounding a cell. The global collision matrix is calculated in an annular model. Only used when 2-d assembly of cells are considered.
ROT+	keyword to specify that the isotropic ( $DP_0$ ) components of the inter-cell current is used. The global collision matrix is calculated in an annular model. Only used when 2-d assembly of cells are considered.
DP00	keyword to specify that the isotropic ( $DP_0$ ) components of the inter-cell current is used. The global collision matrix are computed explicitly. Only used when 2-d assembly of cells are considered.
DP01	keyword to specify that the linearly anisotropic ( $DP_1$ ) components of the inter-cell current are used. This hypothesis implies 12 currents per cell in a cartesian geometry and 18 currents per cell for an hexagonal geometry. Linearly anisotropic reflection is used. Only used when 2-d assembly of cells are considered.
WIGN	keyword to specify the use of a Wigner cylinderization which preserves the volume of the external crown. This applies only in cases where the external surface is annular using the ROTH or ROT+ options. Only used when 2-d assembly of cells are considered. Note that an assembly of rectangular cells having unequal volumes cannot use a Wigner cylinderization.



ASKE	keyword to specify the use of an <i>Askew</i> cylinderization which preserves both the external surface of the cells and the material balance of the external crown (by a modification of its concentration). This applies only in cases where the external surface is annular using the ROTH or ROT+ options. Only used when 2-d assembly of cells are considered. Note that an assembly of rectangular cells having unequal volumes can use an <i>Askew</i> cylinderization.
SANC	keyword to specify the use of a <i>Sanchez</i> cylinderization. This model uses a <i>Wigner</i> cylinderization for computing the collision $P_{ij}$ and leakage $P_{iS}$ probabilities. However, the reciprocity and conservation relations used to compute the incoming $P_{Sj}$ and transmission $P_{SS}$ probabilities are defined in the rectangular cell (with the exact surface). <sup>[?]</sup> This applies where the external surface is annular using the ROTH or ROT+ options. Only used when 2-d assembly of cells are considered. Note that an assembly of rectangular cells having unequal volumes can use a <i>Sanchez</i> cylinderization. This is the default option.
LIGN	keyword to specify that all the integration lines are to be printed. This option should only be used when absolutely necessary because it generates a rather large amount of output. Only used when 2-d assembly of cells are considered.
RECT	keyword to specify that square cells are to be treated as if they were rectangular cells, with the inherent loss in performance that this entails. This option is of purely academic interest.
QUAB	keyword to specify the number of basis point for the numerical integration of each micro-structure in cases involving double heterogeneity (Bihet).
<i>iquab</i>	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 <i>iquab</i> =5.
SAP0	use the Sanchez-Pomraning double-heterogeneity model. <sup>[?]</sup>
HEBE	use the Hebert double-heterogeneity model (default option). <sup>[?]</sup>

### 3.4.2 The EXCELT: tracking module

The calling specification for this module is:

Table 26: Structure (**EXCELT:**)

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

where

<i>TRKNAM</i>	<b>character*12</b> name of the TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information. If <i>TRKNAM</i> also appears on the RHS, the previous tracking parameters will be applied by default on the current geometry.
<i>TRKFIL</i>	<b>character*12</b> name of the sequential binary tracking file used to store the tracks lengths. If <i>TRKFIL</i> does not appear, the keyword <b>XCLL</b> is set automatically. If the user wants to use a tracking file, <i>TRKFIL</i> is required for the <b>EXCELT:</b> module, either on the LHS, on the RHS or on both sides. In the case where <i>TRKFIL</i> appears on both LHS and RHS, the existing tracking file is modified by the module while if <i>TRKFIL</i> appears only on the RHS, the existing tracking file is read but not modified.
<i>GEONAM</i>	<b>character*12</b> name of the GEOMETRY data structure.
( <b>desctrack</b> )	structure describing the general tracking data (see Section 3.4)
( <b>descexcel</b> )	structure describing the transport tracking data specific to <b>EXCELT:</b> .

The **EXCELT:** specific tracking data in (**descexcel**) is defined as

Table 27: Structure (**descexcel**)

```
[ ANIS nanis ]
[ { ONEG | ALLG | XCLL } ]
[ { TREG | TMER } ]
[ { PISO | PSPC [ CUT pcut ] } ]
[ [ QUAB iquab ] [ { SAPO | HEBE } ] ]
[ { PRIX | PRIY | PRIZ } denspr ]
[ { LCMD | OPP1 | OGAU | GAUS | CACA | CACB } [ nmu ] ]
[ TRAK { TISO nanl [ nanl_z ] dens [ dens_z ] [ CORN pcorn ] [ SYMM isymm | NOSY ] |
TSPC [ MEDI ] nanl dens | HALT } ]
;
```

where

<b>ANIS</b>	keyword to specify the order of scattering anisotropy.
<i>nanis</i>	order of anisotropy in transport calculation. A default value of 1 represents isotropic (or transport-corrected) scattering while a value of 2 correspond to linearly anisotropic

scattering. When anisotropic scattering is considered, user should pay attention to the following points:

- the usage of **DIAG**, **SYME**, **SSYM** keywords in the definition of the geometry is forbidden. Indeed, in **EXCELT:/NXT**: tracking procedures, the geometry is “unfolded” according to these symmetries : this is incompatible with the integration of the anisotropic moments of the flux;
- the angular quadratures should be selected paying attention to the restrictions mentioned in this manual in order to ensure the particle conservation.

ONEG	keyword to specify that the tracking is read before computing each group-dependent collision probability or algebraic collapsing matrix (default value if <i>TRKFIL</i> is set). The tracking file is read in each energy group if the method of characteristics (MOC) is used.
ALLG	keyword to specify that the tracking is read once and the collision probability or algebraic collapsing matrices are computed in many energy groups. The tracking file is read once if the method of characteristics (MOC) is used.
XCLL	keyword to specify that the tracking is computed <i>on-demand</i> (it is not stored on a file) and the collision probability matrices are computed in many energy groups. The tracking file <i>TRKFIL</i> should <i>not</i> be provided (default value if <i>TRKFIL</i> is not set).
TREG	keyword to specify that the normalization procedure of the integration lines activated by keywords <b>RENO</b> or <b>REND</b> in Sect. 3.4 is to be performed with respect of the fine volumes as specified in the <b>KEYFLX</b> record of the tracking object. This is the default option.
TMER	keyword to specify that the normalization procedure of the integration lines activated by keywords <b>RENO</b> or <b>REND</b> in Sect. 3.4 is to be performed with respect of the <i>merged volumes</i> as specified in the <b>KEYMRG</b> record of the tracking object.
PISO	keyword to specify that a collision probability calculation with isotropic reflection boundary conditions is required. It is the default option if a <b>TISO</b> type integration is chosen. To obtain accurate transmission probabilities for the isotropic case it is recommended that the normalization options in the <b>ASM</b> : module be used.
PSPC	keyword to specify that a collision probability calculation with specular reflection boundary conditions required; this is the default option if a <b>TSPC</b> type integration is chosen. This calculation is only possible if the file was initially constructed using the <b>TSPC</b> option.
CUT	keyword to specify the input of cutting parameters for the specular integration.
<i>pcut</i>	real value representing the maximum error allowed on the exponential function used for specular collision probability calculations. Tracks will be cut at a length such that the error in the probabilities resulting from this reduced track will be of the order of <i>pcut</i> . By default, there is no cutting of the tracks and <i>pcut</i> =0.0. If this option is used in an entirely reflected case, it is preferable to use the <b>NORM</b> command in the <b>ASM</b> : module.
QUAB	keyword to specify the number of basis point for the numerical integration of each micro-structure in cases involving double heterogeneity (Bihet).
<i>iquab</i>	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 <i>iquab</i> =5.
SAPO	use the Sanchez-Pomraning double-heterogeneity model. <sup>[?]</sup>

HEBE	use the Hebert double-heterogeneity model (default option). <sup>[?]</sup>
PRIX	keyword to specify that a prismatic tracking is considered for a 3D geometry invariant along the $x$ -axis. In this case, the 3D geometry is projected in the $y-z$ plane and a 2D tracking on the projected geometry is performed. This capability is limited to the non-cyclic method of characteristics solver for the time being and a subsequent call to <b>MCCGT</b> : is mandatory.
PRIY	keyword to specify that a prismatic tracking is considered for a 3D geometry invariant along the $y$ -axis. In this case, the 3D geometry is projected in the $z-x$ plane and a 2D tracking on the projected geometry is performed. This capability is limited to the method of characteristics solver for the time being and a subsequent call to <b>MCCGT</b> : is mandatory.
PRIZ	keyword to specify that a prismatic tracking is considered for a 3D geometry invariant along the $z$ -axis. In this case, the 3D geometry is projected in the $x-y$ plane and a 2D tracking on the projected geometry is performed. This capability is limited to the method of characteristics solver for the time being and a subsequent call to <b>MCCGT</b> : is mandatory.
<i>denspr</i>	real value representing the linear track density (in $\text{cm}^{-1}$ ) to be used for the inline construction of 3D tracks from 2D tracking when a prismatic tracking is considered.
LCMD	keyword to specify that optimized (McDaniel-type) polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> This is the default option. The conservation is ensured only for isotropic scattering.
OPP1	keyword to specify that $P_1$ constrained optimized (McDaniel-type) polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic and linearly anisotropic scattering.
OGAU	keyword to specify that Optimized Gauss polar integration angles are to be selected for the method of characteristics. <sup>[?, ?]</sup> The conservation is ensured up to $P_{nmu-1}$ scattering.
GAUS	keyword to specify that Gauss-Legendre polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. The conservation is ensured up to $P_{nmu-1}$ scattering.
CACA	keyword to specify that CACTUS type equal weight polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic scattering.
CACB	keyword to specify that CACTUS type uniformly distributed integration polar angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic scattering.
<i>nmu</i>	user-defined number of polar angles. By default, a value consistent with <i>nanagl</i> is computed by the code. For <b>LCMD</b> , <b>OPP1</b> , <b>OGAU</b> quadratures, <i>nmu</i> is limited to 2, 3 or 4.
TRAK	keyword to specify the tracking parameters to be used.
TISO	keyword to specify that isotropic tracking parameters will be supplied. This is the default tracking option for cluster geometries.
TSPC	keyword to specify that specular tracking parameters will be supplied.
MEDI	keyword to specify that instead of selecting the angles located at the end of each angular interval, the angles located in the middle of these intervals are selected. This is particularly useful if one wants to avoid tracking angles that are parallel to the $X$ - or $Y$ -axis as its is the case when the external region of a <b>CARCEL</b> geometry is voided.

<i>nangl</i>	angular quadrature parameter. For applications involving 3-D cells, the choices are <i>nangl</i> =2, 4, 8, 10, 12, 14 or 16; these angular quadratures $EQ_n$ present a rotational symmetry about the three cartesian axes. For 2-D isotropic applications, any value of <i>nangl</i> $\geq 2$ may be used; equidistant angles will be selected. For 2-D specular applications the input value must be of the form $p + 1$ where $p$ is a prime number (for example $p=7, 11$ , etc.); the choice of <i>nangl</i> = 8, 12, 14, 18, 20, 24, or 30 are allowed. For cluster type geometries the default value is <i>nangl</i> =10 for isotropic cases and <i>nangl</i> =12 for specular cases.
<i>nangl_z</i>	angular quadrature parameter in the axial $Z$ direction. Used only with <i>HEXZ</i> and <i>HEXCELZ</i> geometries.
<i>dens</i>	real value representing the density of the integration lines (in $cm^{-1}$ for 2-D cases and $cm^{-2}$ for 3-D cases). This choice of density along the plan perpendicular to each angle depends on the geometry of the cell to be analyzed. If there are zones of very small volume, a high line density is essential. This value will be readjusted by <i>EXCELT</i> :. In the case of the analysis of a cluster type geometry the default value of this parameter is $5/r_m$ where $r_m$ is the minimum radius of the pins or the minimum thickness of an annular ring in the geometry. If the selected value of <i>dens</i> is too small, some volumes or surfaces may not be tracked.
<i>dens_z</i>	real value representing the density of the integration lines in the axial $Z$ direction. Used only with <i>HEXZ</i> and <i>HEXCELZ</i> geometries.
<i>CORN</i>	keyword to specify that the input of the parameters used to treat the corners for the isotropic integration.
<i>pcorn</i>	maximum distance (cm) between a line and the intersection of $n \geq 2$ external surfaces where track redistribution will take place. Track redistribution will take place if a line comes close to the intersection of $n \geq 2$ external surfaces. In this case the line will be replicated $n$ times, each of these lines being associated with a different external surface, while its weight is reduced by a factor of $1/n$ . This allows for a better distribution of tracks which are relatively close to $n$ external surfaces. By default, there is no treatment of the corners and <i>pcorn</i> =0.0.
<i>SYMM</i>	keyword to specify that the geometry has a rotation symmetry.
<i>isymm</i>	integer value describing the rotation symmetry of the geometry. The fixed default of this parameter is 1.
<i>NOSY</i>	<i>EXCELT</i> : automatically try to take into account geometric symmetries in order to reduce the number of tracks and the CPU time. The <i>NOSY</i> keyword desactivates this automatic capability.
<i>HALT</i>	keyword to specify that the program is to be stopped after the analysis of the geometry, without the explicit tracking being performed.

### 3.4.3 The NXT: tracking module

The calling specification for this module is:

Table 28: Structure (NXT:)

```
[ TRKFIL ] TRKNAM := NXT: [ TRKNAM ] [ GEONAM ] :: (desctrack) (descnxt)
```

where

<i>TRKNAM</i>	<b>character*12</b> name of the TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information. If <i>TRKNAM</i> also appears on the RHS, the previous tracking parameters will be applied by default on the current geometry.
<i>TRKFIL</i>	<b>character*12</b> name of the sequential binary tracking file used to store the tracks lengths. If <i>TRKFIL</i> does not appear, the keyword <b>XCLL</b> is set automatically. If the user wants to use a tracking file, <i>TRKFIL</i> is required.
<i>GEONAM</i>	<b>character*12</b> name of the GEOMETRY data structure.
(desctrack)	structure describing the general tracking data (see Section 3.4)
(descnxt)	structure describing the transport tracking data specific to NXT:.

The NXT: specific tracking data in (descnxt) is defined as

Table 29: Structure (descnxt)

```
[ ANIS nanis ]
[ { ONEG | ALLG | XCLL } ]
[ [ QUAB iquab ] [ { SAPO | HEBE } ] ]
[ { PISO | PSPC [ CUT pcut ] } ]
[ { SYMM isymm | NOSY } ]
[ { GAUS | CACA | CACB | LCMD | OPP1 | OGAU } [ nmu ] ]
{ TISO [ { EQW | GAUS | PNTN | SMS | LSN | QRN } ] nangl dens [ CORN pcorn ]
  | TSPC [ { EQW | MEDI | EQW2 } ] nangl dens }
[ { NOTR | MC } ]
[ NBSLIN nbslin ]
[ LONG ]
[ PRIZ denspr ]
;
```

where

<b>ANIS</b>	keyword to specify the order of scattering anisotropy.
<i>nanis</i>	order of anisotropy in transport calculation. A default value of 1 represents isotropic (or transport-corrected) scattering while a value of 2 correspond to linearly anisotropic

scattering. When anisotropic scattering is considered, user should pay attention to the following points:

- the usage of **DIAG**, **SYME**, **SSYM** keywords in the definition of the geometry is forbidden. Indeed, in **EXCELT:/NXT**: tracking procedures, the geometry is “unfolded” according to these symmetries : this is incompatible with the integration of the anisotropic moments of the flux;
- an angular dependent normalization of the track lengths should be requested in the tracking procedure (**REND** keyword) in order to ensure the particle conservation;
- the angular quadratures should be selected paying attention to the restrictions mentioned in this manual in order to ensure the particle conservation.

ONEG	keyword to specify that the tracking is read before computing each group-dependent collision probability or algebraic collapsing matrix (default value if <i>TRKFIL</i> is set). The tracking file is read in each energy group if the method of characteristics (MOC) is used.
ALLG	keyword to specify that the tracking is read once and the collision probability or algebraic collapsing matrices are computed in many energy groups. The tracking file is read once if the method of characteristics (MOC) is used.
XCLL	keyword to specify that the tracking is computed <i>on-demand</i> (it is not stored on a file) and the collision probability matrices are computed in many energy groups. The tracking file <i>TRKFIL</i> should <i>not</i> be provided (default value if <i>TRKFIL</i> is not set).
QUAB	keyword to specify the number of basis point for the numerical integration of each micro-structure in cases involving double heterogeneity (Bihet).
<i>iquab</i>	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 <i>iquab</i> = 5.
SAPO	use the Sanchez-Pomraning double-heterogeneity model. <sup>[?]</sup>
HEBE	use the Hebert double-heterogeneity model (default option). <sup>[?]</sup>
PISO	keyword to specify that a collision probability calculation with isotropic reflection boundary conditions is required. It is the default option if a <b>TISO</b> type integration is chosen. To obtain accurate transmission probabilities for the isotropic case it is recommended that the normalization options in the <b>ASM</b> : module be used.
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 <b>TSPC</b> type integration is chosen. This calculation is only possible if the file was initially constructed using the <b>TSPC</b> option.
CUT	keyword to specify the input of cutting parameters for the specular collision probability of characteristic integration.
<i>pcut</i>	real value representing the maximum error allowed on the exponential function used for specular collision probability calculations. Tracks will be cut at a length such that the error in the probabilities resulting from this reduced track will be of the order of <i>pcut</i> . By default, the tracks are extended to infinity and <i>pcut</i> = 0.0. If this option is used in an entirely reflected case, it is recommended to use the <b>NORM</b> command in the <b>ASM</b> : module.

SYMM	keyword to specify the level to which the tracking will respect the symmetry of the geometry.
<i>isymm</i>	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 <ul style="list-style-type: none"> <li>• <math>S_x = 1</math> if the <math>X</math> symmetry is to be considered and <math>S_x = 0</math> otherwise.</li> <li>• <math>S_y = 1</math> if the <math>Y</math> symmetry is to be considered and <math>S_y = 0</math> otherwise.</li> <li>• <math>S_z = 1</math> if the <math>Z</math> symmetry is to be considered and <math>S_z = 0</math> otherwise.</li> </ul>
NOSY	keyword to specify the full tracking will take place irrespective of the symmetry of the geometry. This is equivalent to specifying $isymm=0$ .
GAUS	keyword to specify that Gauss-Legendre polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. The conservation is ensured up to $P_{nmu-1}$ scattering.
CACA	keyword to specify that CACTUS type equal weight polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic scattering.
CACB	keyword to specify that CACTUS type uniformly distributed integration polar angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic scattering.
LCMD	keyword to specify that optimized (McDaniel-type) polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> This is the default option. The conservation is ensured only for isotropic scattering.
OPP1	keyword to specify that $P_1$ constrained optimized (McDaniel-type) polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic and linearly anisotropic scattering.
OGAU	keyword to specify that Optimized Gauss polar integration angles are to be selected for the method of characteristics. <sup>[?, ?]</sup> The conservation is ensured up to $P_{nmu-1}$ scattering.
<i>nmu</i>	user-defined number of polar angles. By default, a value consistent with <i>nangl</i> is computed by the code. For LCMD, OPP1, OGAU quadratures, <i>nmu</i> is limited to 2, 3 or 4.
TISO	keyword to specify that isotropic tracking parameters will be supplied. This is the default tracking option for cluster geometries.
TSPC	keyword to specify that specular tracking parameters will be supplied.
EQW	keyword to specify the use of equal weight quadrature. <sup>[?]</sup> The conservation is ensured up to $P_{nangl/2}$ scattering.
GAUS	(after TISO keyword) keyword to specify the use of the Gauss-Legendre quadrature. This option is valid only if an hexagonal geometry is considered.
PNTN	keyword to specify that Legendre-Techbychev quadrature quadrature will be selected. <sup>[?]</sup> The conservation is ensured only for isotropic and linearly anisotropic scattering.
SMS	keyword to specify that Legendre-trapezoidal quadrature quadrature will be selected. <sup>[?]</sup> The conservation is ensured up to $P_{nangl-1}$ scattering.
LSN	keyword to specify the use of the $\mu_1$ -optimized level-symmetric quadrature. The conservation is ensured up to $P_{nangl/2}$ scattering.
QRN	keyword to specify the use of the quadrupole range (QR) quadrature. <sup>[?]</sup>



MEDI	keyword to specify the use of a median angle quadrature in TSPC cases. 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 it is the case when the external region of a CARCEL geometry is voided.
EQW2	keyword to eliminate angles $\phi = 0$ and $\phi = \pi/2$ from the EQW quadrature in TSPC cases.
nangl	angular quadrature parameter. For a 3-D EQW option, the choices are $nangl = 2, 4, 8, 10, 12, 14$ or $16$ . For a 3-D PNTN or SMS option, $nangl$ is an even number smaller than $46$ . <sup>[7]</sup> For 2-D isotropic applications, any value of $nangl$ may be used, equidistant angles will be selected.  For 2-D specular applications the input value must be of the form $p + 1$ where $p$ is a prime number, as proposed in Ref. ?. In this case, the choice of $nangl = 8, 12, 14, 18, 20, 24$ , or $30$ are allowed. For a rectangular Cartesian domain of size $X \times Y$ , the azimuthal angles in $(0, \pi/2)$ interval are obtained from formula $\phi_k = \begin{cases} \arctan\left(\frac{kY}{(p-k)X}\right), & k = 0, 1, 2, \dots, p & \text{if EQW (default)} \\ \arctan\left(\frac{kY}{(2p+2-k)X}\right), & k = 1, 3, 5, \dots, 2p+1 & \text{if MEDI} \\ \arctan\left(\frac{kY}{(p+2-k)X}\right), & k = 1, 2, 3, \dots, p+1 & \text{if EQW2.} \end{cases}$
dens	real value representing the density of the integration lines (in $\text{cm}^{-1}$ for 2-D Cartesian cases and 3-D hexagonal cases and $\text{cm}^{-2}$ for 3-D cases Cartesian 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:.
CORN	keyword to specify that the input of the parameters used to treat the corners for the isotropic integration.
pcorn	maximum distance (cm) between a line and the intersection of $n \geq 2$ external surfaces where track redistributing will take place. Track redistribution will take place if a line comes close to the intersection of $n \geq 2$ external surfaces. In this case the line will be replicated $n$ times, each of these lines being associated with a different external surface, while its weight is reduced by a factor of $1/n$ . This allows for a better distribution of tracks which are relatively close to $n$ external surfaces. By default, there is no treatment of the corners and $pcorn = 0.0$ .
NOTR	keyword to specify that the geometry will not be tracked. This is useful for 2-D geometries to generate a tracking data structure that can be used by the PSP: module (see Section 3.27). One can then verify visually if the geometry is adequate before the tracking process as such is undertaken.
MC	keyword to specify that the geometry will not be tracked and that object TRKNAM will be used with the Monte-Carlo method. This option is similar to NOTR with additional information being added into TRKNAM.
NBSLIN	keyword to set the maximum number of segments in a single tracking line.
nbsl	integer value representing the maximum number of segments in a single tracking line. The default value is $nbsl = 100000$ .
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 3.15).

<i>PRIZ</i>	keyword to specify that a prismatic tracking is considered for a 3D geometry invariant along the $z$ - axis. In this case, the 3D geometry is projected in the $x - y$ plane and a 2D tracking on the projected geometry is performed. This capability is limited to the non-cyclic method of characteristics solver for the time being and a subsequent call to <b>MCCGT</b> : is mandatory.
<i>denspr</i>	real value representing the linear track density (in $\text{cm}^{-1}$ ) to be used for the inline contruction of 3D tracks from 2D tracking when a prismatic tracking is considered.

### 3.4.4 The MCCGT: tracking module

This module *must* follow a call to module EXCELT: or NXT:. Its calling specification is:

Table 30: Structure (MCCGT:)

$TRKNAM := MCCGT: TRKNAM TRKFIL [ GEONAM ] :: (descmccg)$
---

where

- TRKNAM**      **character\*12** name of the TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information. It is provided by EXCELT: or NXT: operator and modified by MCCGT: operator.
- TRKFIL**      **character\*12** name of the sequential binary tracking file used to store the tracks lengths. This file is provided by EXCELT: or NXT: operator and used without modification by MCCGT: operator.
- GEONAM**      **character\*12** name of the optional GEOMETRY data structure. This structure is only required to recover double-heterogeneity data.
- (descmccg)**    structure describing the transport tracking data specific to MCCGT:.

The MCCGT: specific tracking data in (descmccg) is defined as

Table 31: Structure (descmccg)

```
[ EDIT iprint ]
[ { LCMD | OPP1 | OGAU | GAUS | CACA | CACB } [ nmu ] ]
{ DIFC { NONE | DIAG | FULL | ILUO } [ TMT ] [ LEXA ]
  |
  [ [ AAC iaca { NONE | DIAG | FULL | ILUO } [ TMT ] ] [ SCR isr ] [ LEXA ] ]
  [ KRYL ikryl ]
  [ MCU imcu ]
  [ HDD xhdd ]
  [ { SC | LDC } ]
  [ LEXF ]
  [ STIS istis ]
}
[ MAXI nmaxi ]
[ EPSI xepsi ]
[ ADJ ]
;
```

where

- EDIT**              keyword used to modify the print level *iprint*.
- iprint***              index used to control the printing in this operator.

LCMD	keyword to specify that optimized (McDaniel-type) polar integration angles are to be selected for the method of characteristics. <sup>[?]</sup> This is the default option. The conservation is ensured only for isotropic scattering.
OPP1	keyword to specify that $P_1$ constrained optimized (McDaniel-type) polar integration angles are to be selected for the method of characteristics. <sup>[?]</sup> The conservation is ensured only for isotropic and linearly anisotropic scattering.
OGAU	keyword to specify that Optimized Gauss polar integration angles are to be selected for the method of characteristics. <sup>[?, ?]</sup> The conservation is ensured up to $P_{nmu-1}$ scattering.
GAUS	keyword to specify that Gauss-Legendre polar integration angles are to be selected for the method of characteristics. The conservation is ensured up to $P_{nmu-1}$ scattering.
CACA	keyword to specify that CACTUS type equal weight polar integration angles are to be selected for the method of characteristics. <sup>[?]</sup> The conservation is ensured only for isotropic scattering.
CACB	keyword to specify that CACTUS type uniformly distributed integration polar angles are to be selected for the method of characteristics. <sup>[?]</sup> The conservation is ensured only for isotropic scattering.
<i>nmu</i>	user-defined number of polar angles for the integration of the tracks with the method of characteristics for 2D geometries. By default, a value consistent with <i>nangl</i> is computed by the code. For LCMD, OPP1, OGAU quadratures, <i>nmu</i> is limited to 2, 3 or 4.
DIFC	keyword used to specify that only an ACA-simplified transport flux calculation is to be performed (not by default).
LEXA	keyword used to force the usage of exact exponentials in the preconditioner calculation (not by default).
MAXI	keyword to specify the maximum number of inner iterations (or Bi-CGSTAB iterations to solve the ACA-simplified system if DIFC is present).
<i>nmaxi</i>	the maximum number of iterations. The default value is <i>nmaxi</i> =20.
EPSI	keyword to specify the convergence criterion on inner iterations (or ACA-simplified flux calculation if DIFC is present).
<i>xepsi</i>	convergence criterion. The default value is <i>xepsi</i> = $1.0 \times 10^{-5}$ .
AAC	keyword to set the ACA preconditioning of inner/multigroup iterations in case where a transport solution is selected. <sup>[?, ?]</sup>
<i>iaca</i>	0/> 0: ACA preconditioning of inner or multigroup iterations off/on. The default value is <i>iaca</i> =1. If MAXI is set to 1, ACA is used as a rebalancing technique for multigroup-inner mixed iterations and <i>iaca</i> is the maximum number of iterations allowed to solve the ACA system (e.g. 100).
NONE	no preconditioning for the iterative resolution by Bi-CGSTAB of the ACA system.
DIAG	diagonal preconditioning for the iterative resolution by Bi-CGSTAB of the ACA system.
FULL	full-matrix preconditioning for the iterative resolution by Bi-CGSTAB of the ACA system.
ILU0	ILU0 preconditioning for the iterative resolution by Bi-CGSTAB of the ACA system (This is the default option).
TMT	two-step collapsing version of ACA which uses a tracking merging technique while building the ACA matrices.

SCR	keyword to set the SCR preconditioning of inner/multigroup iterations. <sup>[?]</sup>
<i>iscr</i>	0/> 0: SCR preconditioning of inner or multigroup iterations off/on. The default value is <i>iscr</i> =0. If MAXI is set to 1, SCR is used as a rebalancing technique for multigroup-inner mixed iterations and <i>iscr</i> is the maximum number of iterations allowed to solve the SCR system. When anisotropic scattering is considered, SCR provides an acceleration of anisotropic flux moments. If both ACA and SCR are selected ( <i>iscr</i> > 0 and <i>iaca</i> > 0), a two-step acceleration scheme (equivalent to ACA when isotropic scattering is considered) involving both methods is used.
KRYL	keyword to set the Krylov acceleration of inner iterations. <sup>[?]</sup>
<i>ikryl</i>	0: GMRES/Bi-CGSTAB acceleration not used; > 0: dimension of the Krylov subspace in GMRES; < 0: Bi-CGSTAB is used. The default value is <i>ikryl</i> =10.
MCU	keyword used to specify the maximum dimension of the connection matrix for memory allocation.
<i>imcu</i>	The default value is eight (resp. twelve) times the number of volumes and external surfaces for 2D (resp. 3D) geometries.
HDD	keyword to select the integration scheme along the tracking lines.
<i>xhdd</i>	selection criterion:

$$xhdd = \begin{cases} 0.0 & \text{step characteristics scheme} \\ > 0.0 & \text{diamond differencing scheme.} \end{cases}$$

The default value is *xhdd*=0.0 so that the step characteristics method is used.

LEXF	keyword used to force the usage of exact exponentials in the flux calculation (not by default).
SC	keyword used to select the step characteristics (SC) or DD0 diamond differencing approximation. This option is a flat source approximation (default option).
LDC	keyword used to select the linear discontinuous characteristics (LDC) or DD1 diamond differencing approximation. This option is a linear source approximation.
STIS	keyword to select the tracking integration strategy.
<i>istis</i>	0: a direct approach with asymptotical treatment is used; 1: a “source term isolation” approach with asymptotical treatment is used (this technique tends to reduce the computational cost and increase the numerical stability but requires the calculation of angular mode-to-mode self-collision probabilities); -1: an ”MOCC/MCI”-like approach is used (it tends to reduce further more the computational cost as it doesn’t feature any asymptotical treatment for vanishing optical thicknesses). Note that when a zero total cross section is found with <i>istis</i> =-1, it is reset to 1. The default value is <i>istis</i> =1 for $P_{L \leq 3}$ anisotropy and 0 otherwise.
ADJ	keyword to select an adjoint solution of ACA and characteristics systems. A direct solution is set by default.

### 3.4.5 The SNT: tracking module

The SNT: module can process one-dimensional, two-dimensional regular geometries and three-dimensional Cartesian geometries of type CAR1D, TUBE, SPHERE, CAR2D, TUBEZ and CAR3D.

The calling specification for this module is:

Table 32: Structure (SNT:)

$TRKNAM := SNT: [ TRKNAM ] GEONAM :: (desctrack) (descsn)$
--

where

<i>TRKNAM</i>	<b>character*12</b> name of the TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information. If <i>TRKNAM</i> also appears on the RHS, the previous tracking parameters will be applied by default on the current geometry.
<i>GEONAM</i>	<b>character*12</b> name of the GEOMETRY data structure.
( <b>desctrack</b> )	structure describing the general tracking data (see Section 3.4)
( <b>descsn</b> )	structure describing the transport tracking data specific to SNT:.

The SNT: specific tracking data in (**descsn**) is defined as

Table 33: Structure (**descsn**)

$  \begin{aligned}  &[ DIAM \ m ] \\  &SN \ n \ [ SCAT \ iscat ] \ [ \{ DSA \   \ NDSA \} ] \ [ \{ LIVO \ icl1 \ icl2 \   \ NLIVO \} ] \\  &[ GMRES \ nstart ] \ [ NSDSA \ nsdsa ] \ [ MAXI \ maxi ] \ [ EPSI \ epsi ] \ [ QUAD \ iquad ] \\  &[ [ QUAB \ iquab ] \ [ \{ SAPO \   \ HEBE \} ] ] \\  &;  \end{aligned}  $
--

where

( <b>desctrack</b> )	structure describing the general tracking data (see Section 3.4)
DIAM	keyword to fix the spatial approximation order.
<i>m</i>	spatial order. $m = 1$ is used for the classical diamond scheme (default value). $m = 2$ or $m = 3$ is currently available in 1D slab, 2D Cartesian and 3D Cartesian geometries.
SN	keyword to fix the angular approximation order of the flux.
<i>n</i>	order of the $S_N$ approximation (even number).
SCAT	keyword to limit the anisotropy of scattering sources.
<i>iscat</i>	number of terms in the scattering sources. $iscat = 1$ is used for isotropic scattering in the laboratory system. $iscat = 2$ is used for linearly anisotropic scattering in the laboratory system. The default value is set to $n$ .

LIVO	keyword to enable Livolant acceleration method (default value).
<i>icl1</i> , <i>icl2</i>	Numbers of respectively free and accerated iterations in the Livolant method.
NLIVO	keyword to disable Livolant acceleration method.
DSA	keyword to enable diffusion synthetic acceleration using BIVAC or TRIVAC (default value).
NDSA	keyword to disable diffusion synthetic acceleration.
GMRES	keyword to set the GMRES(m) acceleration of the scattering iterations. The default value, equivalent to <i>nstart</i> =0, corresponds to a one-parameter Livolant acceleration. <sup>[?]</sup>
<i>nstart</i>	restarts the GMRES method every <i>nstart</i> iterations.
NSDSA	keyword to set the number if inner flux iterations <i>without</i> DSA in 3D cases if $m \geq 2$ . If DSA is enabled too soon, instabilities and convergence failure can occur in these cases.
<i>nsdsa</i>	number if inner flux iterations <i>without</i> DSA. The default value is <i>nsdsa</i> = 10.
MAXI	Keyword to set the maximum number of inner iterations (or GMRES iterations if activated).
<i>maxi</i>	Maximum number of inner iterations. Default value: 100.
EPSI	Set the convergence criterion on inner iterations (or GMRES iterations if activated).
<i>epsi</i>	Convergence criterion on inner iterations. The default value is $1 \times 10^{-5}$ .
QUAD	keyword to set the type of angular quadrature.
<i>iquad</i>	type of quadrature: = 1: Lathrop-Carlson level-symmetric quadrature; = 2: $\mu_1$ -optimized level-symmetric quadrature (default option in 2D and in 3D); = 3 Snow-code level-symmetric quadrature (obsolete); = 4: Legendre-Chebyshev quadrature (variable number of base points per axial level); = 5: symmetric Legendre-Chebyshev quadrature; = 6: quadruple range (QR) quadrature; <sup>[?]</sup> = 10: product of Gauss-Legendre and Gauss-Chebyshev quadrature (equal number of base points per axial level).
QUAB	keyword to specify the number of basis point for the numerical integration of each micro-structure in cases involving double heterogeneity (Bihet).
<i>iquab</i>	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 <i>iquab</i> =5.
SAPO	use the Sanchez-Pomraning double-heterogeneity model. <sup>[?]</sup>
HEBE	use the Hebert double-heterogeneity model (default option). <sup>[?]</sup>

### 3.4.6 The BIVACT: tracking module

The BIVACT: module can only process 1D/2D regular geometries of type CAR1D, CAR2D and HEX. The geometry is analyzed and a LCM object with signature L\_BIVAC is created with the tracking information.

The calling specification for this module is:

Table 34: Structure (BIVACT:)

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

where

- TRKNAM*      **character\*12** name of the TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information. If *TRKNAM* also appears on the RHS, the previous tracking parameters will be applied by default on the current geometry.
- GEONAM*      **character\*12** name of the GEOMETRY data structure.
- (desctrack)    structure describing the general tracking data (see Section 3.4)
- (descbivac)    structure describing the transport tracking data specific to BIVACT:.

The BIVACT: specific tracking data in (descbivac) is defined as

Table 35: Structure (descbivac)

```
[ { PRIM [ ielem icol [ isplh ] ]
  | DUAL [ ielem icol [ isplh ] ]
  | MCFD isplh } ]
[ { PN | SPN } n [ SCAT [ DIFF ] iscat ] [ VOID nvd ] ]
;
```

where

- (desctrack)    structure describing the general tracking data (see Section 3.4)
- PRIM          keyword to set a primal finite element (classical) discretization.
- DUAL          keyword to set a mixed-dual finite element discretization. If the geometry is hexagonal, a Thomas-Raviart-Schneider method is used.
- MCFD          keyword to set a mesh-centered finite difference discretization in hexagonal geometry.
- ielem*          order of the finite element representation. The values permitted are: 1 (linear polynomials), 2 (parabolic polynomials), 3 (cubic polynomials) or 4 (quartic polynomials). By default *ielem*=1.
- icol*          type of quadrature used to integrate the mass matrices. The values permitted are: 1 (analytical integration), 2 (Gauss-Lobatto quadrature) or 3 (Gauss-Legendre quadrature).



By default  $icol=2$ . The analytical integration corresponds to classical finite elements; the Gauss-Lobatto quadrature corresponds to a variational or nodal type collocation and the Gauss-Legendre quadrature corresponds to superconvergent finite elements.

<i>isplh</i>	type of hexagonal mesh-splitting. This data is given only if the geometry is 2-D hexagonal. The values permitted with the finite difference discretizations are: 1 (full hexagons), 2 for splitting each hexagon into 6 triangles, 3 for splitting each hexagon into 24 triangles, 5 for splitting each hexagon into 96 triangles, 9 for splitting each hexagon into 384 triangles and 17 for splitting each hexagon into 1536 triangles. The values permitted with the Thomas-Raviart-Schneider method are: 1 (full hexagons), $> 1$ for performing a mesh-splitting in $3 \times isplh^2$ losanges per hexagon.
PN	keyword to set a spherical harmonics ( $P_n$ ) expansion of the flux. <sup>[?]</sup> This option is currently limited to 1D and 2D Cartesian geometries.
SPN	keyword to set a simplified spherical harmonics ( $SP_n$ ) expansion of the flux. <sup>[?, ?]</sup> This option is currently available with 1D and 2D Cartesian geometries and with 2D hexagonal geometries.
<i>n</i>	order of the $P_n$ or $SP_n$ expansion (odd number). Set to zero for diffusion theory (default value).
SCAT	keyword to limit the anisotropy of scattering sources.
DIFF	keyword to force using $1/3D^g$ as $\Sigma_1^g$ cross sections. A $P_1$ or $SP_1$ method will therefore behave as diffusion theory.
<i>iscat</i>	number of terms in the scattering sources. $iscat = 1$ is used for isotropic scattering in the laboratory system. $iscat = 2$ is used for linearly anisotropic scattering in the laboratory system. The default value is set to $n + 1$ in $P_n$ or $SP_n$ case.
VOID	key word to set the number of base points in the Gauss-Legendre quadrature used to integrate void boundary conditions if $icol = 3$ and $n \neq 0$ .
<i>nvd</i>	type of quadrature. The values permitted are: 0 (use a $(n+2)$ -point quadrature consistent with $P_n$ theory), 1 (use a $(n+1)$ -point quadrature consistent with $S_{n+1}$ theory), 2 (use an analytical integration of the void boundary conditions). By default $nvd=0$ .

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

- PRIM 1 1 : Linear finite elements;
- PRIM 1 2 : Mesh corner finite differences;
- PRIM 1 3 : Linear superconvergent finite elements;
- PRIM 2 1 : Quadratic finite elements;
- PRIM 2 2 : Quadratic variational collocation method;
- PRIM 2 3 : Quadratic superconvergent finite elements;
- PRIM 3 1 : Cubic finite elements;
- PRIM 3 2 : Cubic variational collocation method;
- PRIM 3 3 : Cubic superconvergent finite elements;
- PRIM 4 2 : Quartic variational collocation method;
- DUAL 1 1 : Mixed-dual linear finite elements;
- DUAL 1 2 : Mesh centered finite differences;

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

### 3.4.7 The TRIVAT: tracking module

The TRIVAT: module is used to perform a TRIVAC-type “tracking” on a 1D/2D/3D regular Cartesian or hexagonal geometry.<sup>[?, ?]</sup> The geometry is analyzed and a LCM object with signature L\_TRIVAC is created with the following information:

- Diagonal and hexagonal symmetries are unfolded and the mesh-splitting operations are performed. Volumes, material mixture and averaged flux recovery indices are computed on the resulting geometry.
- A finite element discretization is performed and the corresponding numbering is saved.
- The unit finite element matrices (mass, stiffness, etc.) are recovered.
- Indices related to an ADI preconditioning with or without supervectorization are saved.

The calling specification for this module is:

Table 36: Structure (TRIVAT:)

*TRKNAM* := TRIVAT: [ *TRKNAM* ] *GEONAM* :: (desctrack) (descTRIVAC)

where

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

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

(desctrack)    structure describing the general tracking data (see Section 3.4)

(descTRIVAC) structure describing the transport tracking data specific to TRIVAT:.

The TRIVAT: specific tracking data in (descTRIVAC) is defined as

Table 37: Structure (descTRIVAC)

```
[ { PRIM [ ielem [ isplh ] ] | DUAL [ ielem icol [ isplh ] ] | MCFD [ ielem [ isplh ] ] | LUMP [ ielem ] } ]
[ SPN n [ SCAT [ DIFF ] iscat ] [ VOID nvd ] ]
[ ADI nadi ]
[ VECT [ iseg ] [ PRTV impv ] ]
;
```

where

(desctrack)    structure describing the general tracking data (see Section 3.4)

PRIM            key word to set a discretization based on the variational collocation method.

DUAL	key word to set a mixed-dual finite element discretization. If the geometry is hexagonal, a Thomas-Raviart-Schneider method is used.
MCFD	key word to set a discretization based on the nodal collocation method. The mesh centered finite difference approximation is the default option and is generally set using MCFD 1. The MCFD approximations are numerically equivalent to the DUAL approximations with $icol=2$ ; however, the MCFD approximations are less expensive.
LUMP	key word to set a discretization based on the nodal collocation method with serendipity approximation. The serendipity approximation is different from the MCFD option in cases with $ielem \geq 2$ . This option is not available for hexagonal geometries.
<i>ielem</i>	order of the finite element representation. The values permitted are: 1 (linear polynomials), 2 (parabolic polynomials), 3 (cubic polynomials) or 4 (quartic polynomials). By default $ielem=1$ .
<i>icol</i>	type of quadrature used to integrate the mass matrices. The values permitted are: 1 (analytical integration), 2 (Gauss-Lobatto quadrature) or 3 (Gauss-Legendre quadrature). By default $icol=2$ . The analytical integration corresponds to classical finite elements; the Gauss-Lobatto quadrature corresponds to a variational or nodal type collocation and the Gauss-Legendre quadrature corresponds to superconvergent finite elements.
<i>isplh</i>	type of hexagonal mesh-splitting. This data is given only if the geometry is 2D or 3D hexagonal. The values permitted with the MCFD option are: 1 (full hexagons), 2 for splitting each hexagon into 6 triangles, 3 for splitting each hexagon into 24 triangles, etc. The values permitted with the PRIM option are: 1 (full hexagons) and 2 for splitting each hexagon into 6 triangles. The values permitted with the Thomas-Raviart-Schneider method are: 1 (full hexagons), $> 1$ for performing a mesh-splitting in $3 \times isplh^2$ losanges per hexagon.
SPN	keyword to set a simplified spherical harmonics ( $SP_n$ ) expansion of the flux. <sup>[?, ?]</sup> This option is available with 1D, 2D and 3D Cartesian geometries and with 2D and 3D hexagonal geometries.
<i>n</i>	order of the $P_n$ or $SP_n$ expansion (odd number). Set to zero for diffusion theory (default value).
SCAT	keyword to limit the anisotropy of scattering sources.
DIFF	keyword to force using $1/3D^g$ as $\Sigma_1^g$ cross sections. A $P_1$ or $SP_1$ method will therefore behave as diffusion theory.
<i>iscat</i>	number of terms in the scattering sources. $iscat = 1$ is used for isotropic scattering in the laboratory system. $iscat = 2$ is used for linearly anisotropic scattering in the laboratory system. The default value is set to $n + 1$ in $P_n$ or $SP_n$ case.
VOID	key word to set the number of base points in the Gauss-Legendre quadrature used to integrate void boundary conditions if $icol = 3$ and $n \neq 0$ .
<i>nvd</i>	type of quadrature. The values permitted are: 0 (use a $(n+2)$ -point quadrature consistent with $P_n$ theory), 1 (use a $(n+1)$ -point quadrature consistent with $S_{n+1}$ theory), 2 (use an analytical integration of the void boundary conditions). By default $nvd=0$ .
ADI	keyword to set the number of ADI iterations at the inner iterative level.
<i>nadi</i>	number of ADI iterations (default: $nadi = 2$ ).
VECT	key word to set an ADI preconditionning with supervectorization. By default, TRIVAC uses an ADI preconditionning without supervectorization.
<i>iseg</i>	width of a vectorial register. <i>iseg</i> is generally a multiple of 64. By default, $iseg=64$ .
PRTV	key word used to set <i>impv</i> .

*impv*            index used to control the printing in supervectorization subroutines. =0 for no print; =1 for minimum printing (default value); Larger values produce increasing amounts of output.

Various finite element approximations can be obtained by combining different values of *ielem* and *isplh*.

### 3.5 The SHI: module

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

Table 38: Structure (**SHI:**)

$MICLIB := SHI: \{ MICLIB \mid OLDLIB \} TRKNAM [ TRKFIL ] :: (descshi)$
--

where

<i>MICLIB</i>	<b>character*12</b> name of the MICROLIB that will contain the microscopic and macroscopic cross sections updated by the self-shielding module. If <i>MICLIB</i> appears on both LHS and RHS, it is updated; otherwise, the internal library <i>OLDLIB</i> is copied into <i>MICLIB</i> and <i>MICLIB</i> is updated.
<i>OLDLIB</i>	<b>character*12</b> name of a read-only MICROLIB that is copied into <i>MICLIB</i> .
<i>TRKNAM</i>	<b>character*12</b> name of the required TRACKING data structure.
<i>TRKFIL</i>	<b>character*12</b> name of the sequential binary tracking file used to store the tracks lengths. This file is given if and only if it was required in the previous tracking module call (see Section 3.4).
( <b>descshi</b> )	structure describing the self-shielding options.

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

#### 3.5.1 Data input for module SHI:

Table 39: Structure (**descshi**)

<pre>[ EDIT <i>iprint</i> ] [ GRMIN <i>lgrmin</i> ] [ GRMAX <i>lgrmax</i> ] [ MXIT <i>imxit</i> ] [ EPS <i>valeps</i> ] [ { LJ   NOLJ } ] [ { GC   NOGC } ] [ NOTR ] [ LEVEL <i>ilev</i> ] [ { PIJ   ARM } ] ;</pre>
--

where

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

<i>iprint</i>	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.
<i>lgrmin</i>	first group number considered during the self-shielding process. By default, <i>lgrmin</i> is set to the first group number containing self-shielding data in the library.
GRMAX	keyword to specify the maximum group number considered during the self-shielding process.
<i>lgrmax</i>	last group number considered during the self-shielding process. By default, <i>lgrmax</i> is set to the last group number containing self-shielding data in the library.
MXIT	keyword to specify the maximum number of iterations during the self-shielding process.
<i>imxit</i>	the maximum number of iterations. The default is <i>imxit</i> =20.
EPS	keyword to specify the convergence criterion for the self-shielding iteration.
<i>valeps</i>	the convergence criterion for the self-shielding iteration. By default, <i>valeps</i> = $1.0 \times 10^{-4}$ .
LJ	keyword to activate the Livolant and Jeanpierre normalization scheme which modifies the self-shielded averaged neutron fluxes in heterogeneous geometries. By default the Livolant and Jeanpierre normalization scheme is not activated.
NOLJ	keyword to deactivate the Livolant and Jeanpierre normalization scheme which modifies the self-shielded averaged neutron fluxes in heterogeneous geometries. This is the default option.
GC	keyword to activate the Goldstein-Cohen approximation in cases where Goldstein-Cohen parameters are stored on the internal library. These parameters may not be available with some libraries (e.g., APLIB1, APLIB2 or MATXS-type libraries). The Goldstein-Cohen parameters can always be imposed using the IRSET keyword of the LIB: module (see Section 3.2). This is the default option.
NOGC	keyword to deactivate the Goldstein-Cohen approximation in cases where Goldstein-Cohen parameters are stored on the internal library.
NOTR	keyword to deactivate the transport correction option for self-shielding calculations (see CTRA in Sections 3.1 and 3.2).
LEVEL	keyword to specify the self-shielding model.
<i>ilev</i>	= 0: original Stamm'ler model (without distributed effects); = 1: use the Nordheim (PIC) distributed self-shielding model <sup>[?] </sup> ; = 2: use both Nordheim (PIC) distributed self-shielding model and Riemann integration method <sup>[?] </sup> . By default, <i>ilev</i> = 0.
PIJ	keyword to specify the use of complete collision probabilities in the self-shielding calculations of SHI:. This is the default option for EXCELT: and SYBILT: trackings. This option is not available for MCCGT: trackings.
ARM	keyword to specify the use of iterative flux techniques in the self-shielding calculations of SHI:. This is the default option for MCCGT: trackings.

### 3.6 The USS: module

The universal self-shielding module in DRAGON, called **USS:**, allows the correction of the microscopic cross sections to take into account the self-shielding effects related to the resonant isotopes. These isotopes are identified as such by the *inrs* parameter, as defined in Section 3.2. The universal self-shielding module is based on the following models:

- The Livolant-Jeanpierre flux factorization and approximations are used to uncouple the self-shielding treatment from the main flux calculation;
- The resonant cross sections are represented using probability tables computed in the **LIB:** module (the keyword **SUBG** or **PTSL** *must* be used). Two approaches can be used to compute the probability tables:
  1. Physical probability tables can be computed using a RMS approach similar to the one used in Wims-7 and Helios.<sup>[?]</sup> In this case, the slowing-down operator of each resonant isotope is represented as a pure ST<sup>[?]</sup>, ST/IR or ST/WR approximation;
  2. Mathematical probability tables<sup>[?]</sup> and slowing-down correlated weight matrices can be computed in selected energy groups using the *Ribon extended* approach.<sup>[?]</sup> In this case, an elastic slowing-down model is used and a mutual self-shielding model is available.
- The resonant fluxes are computed for each band of the probability tables using a subgroup method;
- The flux can be solved using collision probabilities, or using any flux solution technique for which a tracking module is available;
- The resonant isotopes are computed one-a-time, starting from the isotopes with the lower values of index *inrs*, as defined in Section 3.2; If many isotopes have the same value of *inrs*, the isotope with the greatest number of resonant nuclides is self-shielded first. One or many outer iterations can be performed;
- the distributed self-shielded effect is automatically taken into account if different mixture indices are assigned to different regions inside the resonant part of the cell. The rim effect can be computed by dividing the fuel into "onion rings" and by assigning different mixture indices to them.
- a SPH (superhomogénéisation) equivalence is performed to correct the self-shielded cross sections from the non-linear effects related to the heterogeneity of the geometry.

The general format of the data for this module is:

Table 40: Structure (**USS:**)

```
MICLIB := USS: MICLIB_SG [ MICLIB ] TRKNAM [ TRKFIL ] :: (descuss)
```

where

<i>MICLIB</i>	<b>character*12</b> name of the MICROLIB that will contain the microscopic and macroscopic cross sections updated by the self-shielding module. If <i>MICLIB</i> appears on both LHS and RHS, it is updated; otherwise, <i>MICLIB</i> is created.
<i>MICLIB_SG</i>	<b>character*12</b> name of the MICROLIB build by module <b>LIB:</b> and containing probability table information (the keyword <b>SUBG</b> <i>must</i> be used in module <b>LIB:</b> ).
<i>TRKNAM</i>	<b>character*12</b> name of the required TRACKING data structure.



<i>TRKFIL</i>	<b>character*12</b> name of the sequential binary tracking file used to store the tracks lengths. This file is given if and only if it was required in the previous tracking module call (see Section 3.4).
<b>(descuss)</b>	structure describing the self-shielding options.

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

### 3.6.1 Data input for module **USS:**

Table 41: Structure **(descuss)**

```
[ EDIT iprint ]
[ GRMIN lgrmin ] [ GRMAX lgrmax ] [ PASS ipass ] [ NOCO ] [ NOSP ] [ { TRAN | NOTR } ]
[ { PIJ | ARM } ]
[ MAXST imax ]
[ CALC
  [[ REGI suffix [[ isot { ALL | (imix(i),i=1,nmix) } ] ]
  ]]
ENDC ]
;
```

where

<b>EDIT</b>	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	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.
<b>GRMIN</b>	keyword to specify the minimum group number considered during the self-shielding process.
<i>lgrmin</i>	first group number considered during the self-shielding process. By default, <i>lgrmin</i> is set to the first group number containing self-shielding data in the library.
<b>GRMAX</b>	keyword to specify the maximum group number considered during the self-shielding process.
<i>lgrmax</i>	last group number considered during the self-shielding process. By default, <i>lgrmax</i> is set to the last group number containing self-shielding data in the library.
<b>PASS</b>	keyword to specify the number of outer iterations during the self-shielding process.
<i>ipass</i>	the number of iterations. The default is <i>ipass</i> = 2 if <b>MICLIB</b> is created.
<b>NOCO</b>	keyword to ignore the directives set by <b>LIB</b> concerning the mutual resonance shielding model. This keyword has the effect to replace the mutual resonance shielding model in the subgroup projection method (SPM) by a full correlation approximation similar to the technique used in the <b>ECCO</b> code. This keyword can be used to avoid the message

USSIST: UNABLE TO FIND CORRELATED ISOTOPE \*\*\*\*\*.

	that appears with the SPM if the correlated weights matrices are missing in the microlib.
NOSP	keyword to deactivate the SPH equivalence scheme which modifies the self-shielded averaged neutron fluxes in heterogeneous geometries. The default option is to perform SPH equivalence.
TRAN	keyword to activate the transport correction option for self-shielding calculations (see CTRA in Sections 3.1 and 3.2). This is the default option.
NOTR	keyword to deactivate the transport correction option for self-shielding calculations (see CTRA in Sections 3.1 and 3.2).
PIJ	keyword to specify the use of complete collision probabilities in the subgroup and SPH equivalence calculations of USS:. This is the default option for EXCELT: and SYBILT: trackings. This option is not available for MCCGT: trackings.
ARM	keyword to specify the use of iterative flux techniques in the subgroup and SPH equivalence calculations of USS:. This is the default option for MCCGT: trackings.
MAXST	keyword to set the maximum number of fixed point iterations for the ST scattering source convergence.
<i>imax</i>	the maximum number of ST iterations. The default is <i>imax</i> = 50. A non-iterative response matrix approach is available with the subgroup projection method (SPM) by setting <i>imax</i> = 0.
CALC	keyword to activate the simplified self-shielding approximation in which a single self-shielded isotope is shared by many resonant mixtures.
REGI	keyword to specify a set of isotopes and mixtures that will be self-shielded together. All the self-shielded isotopes in this group will share the same 4-digit suffix.
<i>suffix</i>	<b>character*4</b> suffix for the isotope names in this group
<i>isot</i>	<b>character*8</b> alias name of a self-shielded isotope in this group
ALL	keyword to specify that a unique self-shielded isotope will be made for the complete domain
<i>imix</i>	list of mixture indices that will share the same self-shielded isotope
<i>nmix</i>	number of mixtures that will share the same self-shielded isotope
ENDC	end of CALC data keyword

Here is an example of the data structure corresponding to a production case where only  $^{238}\text{U}$  is assumed to show distributed self-shielding effects:

```

LIBRARY2 := USS: LIBRARY TRACK ::
  CALC REGI W1 PU239 ALL
    REGI W1 PU241 ALL
    REGI W1 PU240 ALL
    REGI W1 PU242 ALL
    REGI W1 U235 ALL
    REGI W1 U236 ALL
    REGI W1 PU238 ALL
    REGI W1 U234 ALL
    REGI W1 AM241 ALL
    REGI W1 NP237 ALL
    REGI W1 ZRNAT ALL

```

```

      REGI W1 U238 <<COMB0101>> <<COMB0201>> <<COMB0301>>
                  <<COMB0401>> <<COMB0501>>
      REGI W2 U238 <<COMB0102>> <<COMB0202>> <<COMB0302>>
                  <<COMB0402>> <<COMB0502>>
      REGI W3 U238 <<COMB0103>> <<COMB0203>> <<COMB0303>>
                  <<COMB0403>> <<COMB0503>>
      REGI W4 U238 <<COMB0104>> <<COMB0204>> <<COMB0304>>
                  <<COMB0404>> <<COMB0504>>
      REGI W5 U238 <<COMB0105>> <<COMB0205>> <<COMB0305>>
                  <<COMB0405>> <<COMB0505>>
      REGI W6 U238 <<COMB0106>> <<COMB0206>> <<COMB0306>>
                  <<COMB0406>> <<COMB0506>>

      ENDC ;

```

In this case,  $^{238}\text{U}$  is self-shielded within six distributed regions (labeled W1 to W6) and each of these regions are merging volumes belonging to five different fuel rods. The mixture indices of the 30 resonant volumes belonging to the fuel are CLE-2000 variables labeled <<COMB0101>> to <<COMB0506>>.

### 3.7 The ASM: module

We will now describe the assembly modules which can be used to prepare the group-dependent complete collision probability or the assembly matrices required by the flux solution module of DRAGON. The assembly module **ASM:** is generally called after a tracking module; it recovers tracking lengths and material numbers from the sequential tracking file and then computes the collision probability or group-dependent system matrices under various normalizations. The calling specifications are:

Table 42: Structure (**ASM:**)

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

where

<i>PIJNAM</i>	<b>character*12</b> name of ASMPIJ data structure containing the system matrices. If <i>PIJNAM</i> appears on the RHS, the ( <b>descasm</b> ) information previously stored in <i>PIJNAM</i> is kept.
<i>LIBNAM</i>	<b>character*12</b> name of the MACROLIB or MICROLIB data structure that contains the macroscopic cross sections (see Sections 3.1 and 3.2).
<i>TRKNAM</i>	<b>character*12</b> name of the TRACKING data structure containing the tracking (see Section 3.4).
<i>TRKFIL</i>	<b>character*12</b> name of the sequential binary tracking file used to store the tracks lengths. This file is given if and only if it was required in the previous tracking module call (see Section 3.4).
( <b>descasm</b> )	structure containing the input data to this module (see Section 3.7.1).

#### 3.7.1 Data input for module ASM:

Table 43: Structure (**descasm**)

```
[ EDIT iprint ]
[ { ARM |
  { PIJ | PIJK } [ SKIP ]
  [ { NORM | ALBS } ]
  [ PNOR { NONE | DIAG | GELB | HELI | NONL } ]
} ]
[ ECCO ]
;
```

where

EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index used to control the printing of this module. The amount of output produced by

this tracking module will vary substantially depending on the print level specified.

ARM	keyword to specify that an assembly calculation is carried out without building the full collision probability matrices. This option can only be used for a geometry tracked using the SYBILT: (with EURYDICE-2 option) or MCCGT: module. By default, the PIJ option is used.
PIJ	keyword to specify that the standard scattering-reduced collision probabilities must be computed. This option cannot be used with the MCCGT: module. This is the default option.
PIJK	keyword to specify that both the directional and standard scattering-reduced collision probabilities must be computed. Moreover, the additional directional collision probability matrix can only be used if HETE is activated in Section 3.8. Finally, the PIJK option is only available for 2-D geometries analyzed with the operator EXCELT: with collision probability option. By default, the PIJ option is used.
SKIP	keyword to specify that only the reduced 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:

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

where  $\Sigma_{s0}^{g \rightarrow g}$  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. Consequently, the total number of inner iterations is greatly increased.

NORM	keyword to specify that the scattering-reduced 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 <i>a posteriori</i> a uniform reentrant current.
ALBS	keyword to specify that a consistent Selengut normalization of the scattering-reduced collision probability matrix is to be used both for the flux solution module (see Section 3.8) and in the equivalence calculation (see Section 3.9). This keyword results in storing the scattering-reduced escape probabilities $W_{iS}$ in the record named 'DRAGON-WIS'. 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).
PNOR	keyword to specify that the collision, leakage and escape probability matrices are to be normalized in such a way as to satisfy explicitly the neutron conservation laws. This option compensates for the errors which will arise in the numerical evaluation of these probabilities and may result in non-conservative collision probability matrices. The default option is now HELI while it was formerly GELB ( <b>Revision 3.03</b> ).
NONE	keyword to specify that the probability matrices are not to be renormalized.
DIAG	keyword to specify that only the diagonal element of the probability matrices will be modified in order to insure the validity of the conservation laws.
GELB	keyword to specify that the Gelbard algorithm will be used to normalize the collision probability matrices. <sup>[?]</sup>
HELI	keyword to specify that the Helios algorithm will be used to normalize the collision probability matrices. <sup>[?]</sup>
NONL	keyword to specify that a non-linear multiplicative algorithm will be used to normalize the collision probability matrices. <sup>[?]</sup>

ECCO keyword used to compute the  $P_1$ -scattering reduced collision probability or system matrices required by the ECCO isotropic streaming model. By default, this information is not calculated.

### 3.8 The FLU: module

The FLU: module is used to solve the linear system of multigroup collision probability or response matrix equations in DRAGON. Different types of solution are available, such as fixed source problem, fixed source eigenvalue problem (GPT type) or different types of eigenvalue problems. The calling specifications are:

Table 44: Structure (**FLU:**)

```
FLUNAM := FLU: [ FLUNAM ] PIJNAM LIBNAM TRKNAM [ TRKFIL ]
               [ TRKFLP TRKGPT ] :: (descflu)
```

where

<i>FLUNAM</i>	<b>character*12</b> name of the FLUXUNK data structure containing the solution (L_FLUX signature). If <i>FLUNAM</i> appears on the RHS, the solution previously stored in <i>FLUNAM</i> (flux and buckling) is used to initialize the new iterative process; otherwise, a uniform unknown vector and a zero buckling are used.
<i>PIJNAM</i>	<b>character*12</b> name of the ASMPIJ data structure containing the group-dependent system matrices (L_PIJ signature, see Section 3.7).
<i>LIBNAM</i>	<b>character*12</b> name of the MACROLIB or MICROLIB data structure that contains the macroscopic cross sections (L_MACROLIB or L_LIBRARY signature, see Sections 3.1 and 3.2). Module FLU: is performing a <i>direct</i> or <i>adjoint</i> calculation, depending if the adjoint flag is set to <code>.false.</code> or <code>.true.</code> in the STATE-VECTOR record of the MACROLIB.
<i>TRKNAM</i>	<b>character*12</b> name of the TRACKING data structure containing the tracking (L_TRACK signature, see Section 3.4).
<i>TRKFIL</i>	<b>character*12</b> name of the sequential binary tracking file used to store the tracks lengths. This file is given if and only if it was required in the previous tracking module call (see Section 3.4).
<i>TRKFLP</i>	<b>character*12</b> name of the FLUXUNK data structure containing the unperturbed flux used to decontaminate the GPT solution (L_FLUX signature). This object is mandatory if and only if “TYPE P” is selected.
<i>TRKGPT</i>	<b>character*12</b> name of the GPT data structure containing the GPT fixed sources (L_GPT signature). This object is mandatory if and only if “TYPE P” is selected.
( <b>descflu</b> )	structure containing the input data to this module (see Section 3.8.1).

## 3.8.1 Data input for module FLU:

Table 45: Structure (**descflu**)

```

[ EDIT iprint ]
[ INIT { OFF | ON } ]
TYPE { N | S | P | K [ (descleak) ] | { B | L } (descleak) } ]
[ EXTE [ maxout ] [ epsout ] ]
[ THER [ maxthr ] [ epsthr ] ] [ REBA [ OFF ] ]
[ UNKT [ epsunk ] ]
[ ACCE nlibre naccel ]
;
```

where

EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index used to control the printing of this operator. The amount of output produced by this operator will vary substantially depending on the print level specified.
OFF	keyword to specify that the neutron flux is to be initialized with a flat distribution.
ON	keyword to specify that the initial neutron flux distribution is to be recovered from <i>FLUNAM</i> if present in the RHS arguments. Otherwise, the neutron flux is to be initialized with a flat distribution.
TYPE	keyword to specify the type of solution used in the flux operator.
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 <i>FLUNAM</i> .
S	keyword to specify that a fixed source problem is to be treated. Such problem can also include fission source contributions.
P	keyword to specify that a fixed source eigenvalue problem (GPT type) is to be treated. Such problem includes fission source contributions in addition of GPT sources.
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 sources, if any is present on the MACROLIB or MICROLIB data structure, are 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 ( <b>descBC</b> ) in Section 3.3.2). It is also possible to use an open geometry with VOID boundary conditions provided it is closed by the ASM: module (see Section 3.7.1) using the keywords NORM or ALSB.
L	keyword to specify that a non-multiplicative medium eigenvalue problem is to be treated. The eigenvalue in this case is the critical buckling with vanishing fission cross sections. The buckling eigenvalue has meaning only in the case of a cell without leakages (see the structure ( <b>descBC</b> ) in Section 3.3.2). It is also possible to use an open geometry with



	VOID boundary conditions provided it is closed by the <b>ASM:</b> module (see Section 3.7.1) using the keywords <b>NORM</b> or <b>ALSB</b> .
<b>(descleak)</b>	structure describing the general leakage parameters options (see Section 3.8.2).
<b>EXTE</b>	keyword to specify that the control parameters for the external iteration are to be modified.
<i>maxout</i>	maximum number of external iterations. The fixed default value for a case with no leakage model is $maxout=2 \times n_f - 1$ where $n_f$ is the number of regions containing fuel. The fixed default value for a case with a leakage model is $maxout=10 \times n_f - 1$ .
<i>epsout</i>	convergence criterion for the external iterations. The fixed default value is $epsout=5.0 \times 10^{-5}$ .
<b>THER</b>	keyword to specify that the control parameters for the thermal iterations are to be modified.
<i>maxthr</i>	maximum number of thermal iterations. The fixed default value is $maxthr=2 \times ngroup-1$ (using scattering modified CP) or $maxthr=4 \times ngroup-1$ (using standard CP).
<i>epsthr</i>	convergence criterion for the thermal iterations. The fixed default value is $epsthr=5.0 \times 10^{-5}$ .
<b>UNKT</b>	keyword to specify the flux error tolerance in the outer iteration.
<i>epsunk</i>	convergence criterion for flux components in the outer iteration. The fixed default value is $epsunk=epsthr$ .
<b>REBA</b>	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.
<b>OFF</b>	keyword used to deactivate the flux rebalancing option. When this keyword is absent the flux rebalancing option is reactivated.
<b>ACCE</b>	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.
<i>nlibre</i>	number of free iterations per cycle of $nlibre+naccel$ iterations.
<i>naccel</i>	number of accelerated iterations per cycle of $nlibre+naccel$ iterations. Variational acceleration may be deactivated by using $naccel=0$ .

### 3.8.2 Leakage model specification structure

Without leakage model, the multigroup flux  $\vec{\phi}_g$  of the collision probability method is obtained from equation

$$\vec{\phi}_g = \mathbf{W}_g \vec{Q}_g^* \quad (3.1)$$

where  $\mathbf{W}_g$  is the scattering reduced collision probability matrix and  $\vec{Q}_g^*$  is the fission and out-of-group scattering source. This equation is modified by the leakage model. The leakage models **PNLR**, **PNL**, **SIGS** and **ECCO** can also be used with solutions techniques other than the collision probability method.

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

Table 46: Structure (**descleak**)

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

LKRD	keyword used to specify that the leakage coefficients are recovered from data structure named <i>LIBNAM</i> . The LKRD option is not available with the ECCO and HETE leakage models.
RHS	keyword used to specify that the leakage coefficients are recovered from RHS flux data structure named <i>FLUNAM</i> . The RHS option is not available with the ECCO and HETE leakage models. If the flux calculation is an adjoint calculation, the energy group ordering of the leakage coefficients is permuted.
P0	keyword used to specify that the leakage coefficients are calculated using a $P_0$ model.
P1	keyword used to specify that the leakage coefficients are calculated using a $P_1$ model.
B0	keyword used to specify that the leakage coefficients are calculated using a $B_0$ model. This is the default value when a buckling calculation is required (B).
B1	keyword used to specify that the leakage coefficients are calculated using a $B_1$ model.
BOTR	keyword used to specify that the leakage coefficients are calculated using a $B_0$ model with transport correction.
PNLR	keyword used to specify that the elements of the scattering modified collision probability matrix are multiplied by the adequate non-leakage homogeneous buckling dependent factor. <sup>[?]</sup> . The non-leakage factor $P_{\text{NLR},g}$ is defined as

$$P_{\text{NLR},g} = \frac{\bar{\Sigma}_g - \bar{\Sigma}_{s0,g \leftarrow g}}{\bar{\Sigma}_g - \bar{\Sigma}_{s0,g \leftarrow g} + d_g(B) B^2} \quad (3.2)$$

where transport-corrected total cross sections are used to compute the  $\mathbf{W}_g$  matrix.  $\bar{\Sigma}_{s0,g \leftarrow g}$  is the average transport-corrected macroscopic within-group scattering cross section in group  $g$ , homogenized over the lattice and transport corrected. Eq. (3.1) is then replaced by

$$\vec{\phi}_g = P_{\text{NLR},g} \mathbf{W}_g \vec{Q}_g^* \quad (3.3)$$

PNL	keyword used to specify that the elements of the collision probability matrix are multiplied by the adequate non-leakage homogeneous buckling dependent factor. <sup>[?]</sup> . This is the default option when a buckling calculation is required (B) or a fission source eigenvalue problem (K) with imposed buckling is considered. The non-leakage factor $P_{\text{NL},g}$ is defined as
-----	--

$$P_{\text{NL},g} = \frac{\bar{\Sigma}_g}{\bar{\Sigma}_g + d_g(B) B^2} \quad (3.4)$$

where  $\bar{\Sigma}_g$  is the average transport-corrected macroscopic total cross section in group  $g$ , homogenized over the lattice and transport corrected. Eq. (3.1) is then replaced by

$$\vec{\phi}_g = \mathbf{W}_g \left[ P_{\text{NL},g} \vec{Q}_g^* - (1 - P_{\text{NL},g}) \bar{\Sigma}_{s0,g \leftarrow g} \vec{\phi}_g \right] \quad (3.5)$$

where  $\Sigma_{s0,g \leftarrow g} = \text{diag}\{\Sigma_{s0,i,g \leftarrow g} ; \forall i\}$  and the total cross sections used to compute the  $\mathbf{W}_g$  matrix are also transport-corrected.

It is important to note that the PNLR option reduces to the PNL option in cases where no scattering reduction is performed. Scattering reduction can be avoided in module **ASM**: by setting **PIJ SKIP** (See Section 3.7.1).

**SIGS** keyword used to specify that an homogeneous buckling correction is to be applied on the diffusion cross section ( $\Sigma_s - dB^2$ ). Eq. (3.1) is then replaced by

$$\vec{\phi}_g = \mathbf{W}_g \left[ \vec{Q}_g^* - d_g(B) B^2 \vec{\phi}_g \right] \quad (3.6)$$

where transport-corrected total cross sections are used to compute the  $\mathbf{W}_g$  matrix. This is the so called *DIFFON method* used in the APOLLO-family of thermal lattice codes.

**ALBS** keyword used to specify that an homogeneous buckling contribution is introduced by a group dependent correction of the albedo.<sup>[?]</sup> This leakage model is restricted to the collision probability method. It is then necessary to define the geometry with an external boundary condition of type **VOID** (see Section 3.3.2) and to close the region in module **ASM**: using the **ALBS** option (see Section 3.7.1). Eq. (3.1) is then replaced by

$$\vec{\phi}_g = \mathbf{W}_g \vec{Q}_g^* - [\mathbf{I} + \mathbf{W}_g \Sigma_{s0,g \leftarrow g}] d_g(B) B^2 \gamma \mathbf{P}_{iS,g} \quad (3.7)$$

where  $\mathbf{P}_{iS,g} = \{P_{iS,g} ; i = 1, I\}$  is the array of escape probabilities in the open geometry and where

$$\gamma = \frac{\sum_j V_j \phi_{j,g}}{\sum_j V_j \phi_{j,g} P_{jS,g}} \quad (3.8)$$

**ECCO** keyword used to perform an ECCO-type leakage calculation taking into account isotropic streaming effects. This method introduces an heterogeneous buckling contribution as a group dependent correction to the source term.<sup>[?, ?]</sup> It is then necessary to set the keyword **ECCO** in module **ASM**: (see Section 3.7.1). In the  $P_1$  non-consistent case, Eq. (3.1) is then replaced by

$$\vec{\varphi}_g = \mathbf{W}_g \left( \vec{Q}_g^* - B^2 \frac{i\vec{\mathcal{J}}_g}{B} \right) \quad (3.9)$$

$$\frac{i\vec{\mathcal{J}}_g}{B} = \mathbf{X}_g \left[ \frac{1}{3} \vec{\varphi}_g + \sum_{h \neq g} \Sigma_{s1,g \leftarrow h} \frac{i\vec{\mathcal{J}}_h}{B} \right] \quad (3.10)$$

where  $i\vec{\mathcal{J}}_{j,g}/B$  is the multigroup fundamental current,  $\Sigma_{s1,g \leftarrow h} = \text{diag}\{\Sigma_{s1,i,g \leftarrow h} ; \forall i\}$  and where

$$\mathbf{X}_g = [\mathbf{I} - \mathbf{p}_g \Sigma_{s1,g \leftarrow g}]^{-1} \mathbf{p}_g \quad (3.11)$$

**HETE** keyword used to perform a TIBÈRE-type leakage calculation taking into account anisotropic streaming effects. This method introduces an heterogeneous buckling contribution as a group dependent correction to the source term.<sup>[?, ?]</sup> The heterogeneous buckling contribution is introduced in the  $B_n$  model using directional collision probabilities (**PIJK** method). It is then necessary to set the keyword **PIJK** in module **ASM**: (see Section 3.7.1).

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

R	keyword used to specify that a radial buckling search will be considered assuming an imposed $z$ -direction buckling.
Z	keyword used to specify that a $z$ -direction buckling search will be considered assuming an imposed $x$ -direction and $y$ -direction buckling.
X	keyword used to specify that a $x$ -direction buckling search will be considered assuming an imposed $y$ -direction and $z$ -direction buckling.
Y	keyword used to specify that a $y$ -direction buckling search will be considered assuming an imposed $x$ -direction and $z$ -direction buckling.
BUCK	keyword used to specify the initial (for a buckling eigenvalue problem) or fixed (for a effective multiplication constant eigenvalue problem) buckling.
G	keyword used to specify that the buckling in the $x$ -direction, $y$ -direction and $z$ -direction are to be initialized to $valb2/3$ (floating default).
R	keyword used to specify that the buckling in the $x$ -direction, and $y$ -direction are to be initialized to $valbr2/2$ .
Z	keyword used to specify that the buckling in the $z$ -direction, is to be initialized to $valbz2$ .
X	keyword used to specify that the buckling in the $x$ -direction, is to be initialized to $valbx2$ .
Y	keyword used to specify that the buckling in the $y$ -direction, is to be initialized to $valby2$ .
$valb2$	value of the fixed or initial total buckling in $cm^{-2}$ . The floating default value is $valb2 = valbx2 + valby2 + valbz2.$
$valbr2$	value of the fixed or initial radial buckling in $cm^{-2}$ . The floating default value is $valbr2 = valbx2 + valby2.$
$valbz2$	value of the fixed or initial $z$ -direction buckling in $cm^{-2}$ . The floating default value is $valbz2=0.0\ cm^{-2}$ . If $valb2$ is specified then $valbz2=valb2/3$ .
$valbx2$	value of the fixed or initial $x$ -direction buckling in $cm^{-2}$ . The floating default value is $valbx2=0.0\ cm^{-2}$ . 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 $cm^{-2}$ . The floating default value is $valby2=0.0\ cm^{-2}$ . 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) buckling is to be read from the data structure <i>LIBNAM</i> .

### 3.9 The EDI: module

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

Table 47: Structure (EDI:)

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

where

<i>EDINAM</i>	<b>character*12</b> name of the EDITION data structure (L_EDIT signature) where the edition results will be stored.
<i>LIBNAM</i>	<b>character*12</b> name of the read-only MACROLIB or MICROLIB data structure (L_MACROLIB or L_LIBRARY signature) that contains the macroscopic cross sections (see Sections 3.1 and 3.2).
<i>TRKNAM</i>	<b>character*12</b> name of the read-only TRACKING data structure (L_TRACK signature) containing the tracking (see Section 3.4). <b>Note:</b> If data structures <i>TRKNAM</i> and <i>FLUNAM</i> are not given, a flux is recovered from the MACROLIB present in <i>LIBNAM</i> and used to perform the editions.
<i>FLUNAM</i>	<b>character*12</b> name of the read-only FLUXUNK data structure (L_FLUX signature) containing a transport solution (see Section 3.8).
<i>REFGEO</i>	<b>character*12</b> optional name of the read-only reference GEOMETRY data structure (L_GEOM signature) that was used for the original flux calculation (see Section 3.3).
<i>MACROGEO</i>	<b>character*12</b> optional name of the read-only macro-GEOMETRY data structure (L_GEOM signature) that is saved in <i>EDINAM</i> and can be used in the homogenization process or in the SPH equivalence procedure. In some cases the module EDI: can automatically build a macro-geometry, however it is always possible to specify explicitly the macro-geometry to be saved in <i>EDINAM</i> .
<i>REFPIJ</i>	<b>character*12</b> optional name of the read-only ASMPIJ data structure (L_PIJ signature) that was used for the reference flux calculation (see Section 3.7). Compulsory if keyword ALBS is used in Section 3.9.1.
(descedi)	structure containing the input data to this module (see Section 3.9.1).

#### 3.9.1 Data input for module EDI:

Table 48: Structure (descedi)

```
[ EDIT iprint ]
```

continued on next page

Structure (**descedi**)

continued from last page

```

[ UPS ]
[ MERG { NONE | COMP | GEO | HMIX |
  CELL [ { SYBIL | EXCELL | NXT | DEFAULT | UNFOLD } ] [ REMIX (imixm2(ii),ii=1,nbmix2) ] |
  REGI (iregm(ii),ii=1,nregio) |
  MIX [ (imixm(ii),ii=1,nbmix) ] } ]
[ TAKE {
  REGI (iregt(ii),ii=1,nregio) |
  MIX (imixt(ii),ii=1,nbmix) } ]
[ { POW | P1W_L | P1W_T } ]
[ COND [ { NONE | ( icond(ii), ii=1,ngcond) | ( energy(ii), ii=1,ngcond) } ] ]
[ MICR [ ALLX ] [ ISOTXS [ ASCII ] ] { ALL | RES | nis (HISO(i),i=1,nis) }
  [ REAC nreac (HREAC(i),i=1,nreac) ] ]
[ ACTI [ ISOTXS [ ASCII ] ] { NONE | (imixa(ii),ii=1,nbmix) } ]
[ SAVE [ ON { DIRN | idirn } ] ]
[ PERT ]
[ STAT { ALL | RATE | FLUX | DELS } [ REFE { DIRO | idiro } ] ]
[ NOHF ] [ NBAL ]
[ MAXR maxpts ]
[ { DIRE | PROD } ]
[ MGEO MACGEO ]
[ { NADF | ALBS | JOUT | ADFM |
  [[ ADF TYPE { REGI (ireg(ii),ii=1,iimax) ENDR | MIX (imix(ii),ii=1,iimax) ENDM } ] ] } ]

```

where

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

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

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

**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 which cannot take into account such cross sections.

**NONE** keyword to deactivate the homogeneization or the condensation.

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

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

<i>iregm</i>	array of homogenized region numbers to which are associated the old regions. In the editing routines a value of <i>iregm</i> =0 allows the corresponding region to be neglected.
MIX	keyword to specify that the homogenization of the neutron flux will take place over the following mixtures. Here we must have $\text{nbmix} \leq \text{maxmix}$ where <i>maxmix</i> is the maximum number of mixtures in the macroscopic cross section library.
<i>imixm</i>	array of homogenized region numbers to which are associated the material mixtures. In the editing routines a value of <i>imixm</i> =0 allows the corresponding isotopic mixtures to be neglected. For a mixture in this library which is not used in the geometry one should insert a value of 0 for the new region number associated with this mixture. By default, if MIX is set and <i>imixm</i> is not set, <i>imixm(ii)=ii</i> is assumed.
COMP	keyword to specify that the a complete homogenization is to take place.
GEO	keyword to specify that a geometry equivalence procedure (equigeom) is to be used. Merging indices are automatically computed by comparing the reference geometry <i>RE-FGEO</i> with the macro-geometry <i>MACROGEO</i> . This capability is limited to EXCELL-type reference geometries.
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 3.3.4). 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 3.3.4). 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 the a cell-by-cell homogenization (with or without SPH equivalence) is to take place. The macro-geometry and the merging indices are automatically computed and the macro-geometry named <i>MACRO-GEOM</i> is created on the root directory of <i>EDINAM</i> . This capability is limited to reference geometries previously tracked by EURYDICE (see Section 3.4.1) or NXT (see Section 3.4.3).
SYBIL	the macro-geometry produced by CELL is tracked by SYBILT: module.
EXCELL	the macro-geometry produced by CELL is tracked by EXCELT: module.
NXT	the macro-geometry produced by CELL is tracked by NXT: module.
DEFAULT	the macro-geometry produced by CELL is tracked by another module (default option).
UNFOLD	the macro-geometry produced by CELL is unfolded and tracked with the DEFAULT option. This option is useful with fine power reconstruction techniques.
REMIX	the cell-by-cell homogenization produced by option CELL is further homogenized according to <i>imixm2</i> indices. This option is useful to integrate the assembly gap into the boundary cells. By default, one homogenized region is created for each cell of the macro-geometry.
<i>imixm2</i>	array of rehomogenized region numbers to which are associated the regions indices created <i>after</i> the cell-by-cell homogenization was performed. In the editing routines a value of <i>imixm2</i> =0 allows the corresponding cell region to be neglected. Here, <i>nbmix2</i> is equal to the number of cells in the macro-geometry.
TAKE	keyword to specify that the neutron flux is to be edited over specified regions or mixtures.
REGI	keyword to specify that the editing of the neutron flux will take place over the following regions. Here $\text{nregio} \leq \text{maxreg}$ with <i>maxreg</i> the maximum number of regions for which solutions were obtained.

<i>iregt</i>	regions where the editing will take place. The new region numbers associated with these editing regions are numbered sequentially.
MIX	keyword to specify that the editing of the neutron flux will take place over the following mixtures. Here we must have $\text{nbmix} \leq \text{maxmix}$ where <i>maxmix</i> is the maximum number of mixtures in the macroscopic cross section library.
<i>imixt</i>	mixtures where the editing will take place. Each mixture set here must exist in the reference geometry.
POW	keyword to specify that the $P_1$ information is to be homogenized and condensed using the scalar flux. This is the default option.
P1W_L	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.
P1W_T	keyword to specify that the $P_1$ information is to be homogenized and condensed using the Todorova flux <sup>[?]</sup> , defined as

$$\phi_1(\mathbf{r}, E) = \frac{\phi(\mathbf{r}, E)}{\Sigma_i(E) - \Sigma_{s1,i}(E)}$$

where  $\Sigma_i(E)$  and  $\Sigma_{s1,i}(E)$  are the macroscopic total and  $P_1$  scattering cross sections in the mixture  $i$  containing the point  $\mathbf{r}$ .

COND	keyword to specify that a group condensation of the flux is to be performed.
<i>icond</i>	array of increasing energy group limits that will be associated with each of the <i>ngcond</i> condensed groups. The final value of <i>icond</i> will automatically be set to <i>ngroup</i> while the values of <i>icond</i> > <i>ngroup</i> will be dropped from the condensation. We must have $\text{ngcond} \leq \text{ngroup}$ . By default, if COND is set and <i>icond</i> is not set, all energy groups are condensed together.
<i>energy</i>	array of decreasing energy limits (in eV) that will be associated with each of the <i>ngcond</i> condensed groups. We must have $\text{ngcond} \leq \text{ngroup} + 1$ . 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 fluxes appearing in the regions where the isotopes are present. The resulting nuclear properties are saved on <i>EDINAM</i> when the <i>SAVE</i> keyword is present.
ALLX	keyword used to register the region number of each isotope before merging, in the embedded library. The homogenized information is therefore registered for each isotope in the merging region, as depicted by the formulas below. This procedure is useful to produce particular databases, in order to perform micro-depletion calculations in diffusion with DONJON.
ALL	keyword to specify that all the isotopes present in the homogenized region are to be kept individual and processed.
RES	keyword to specify that all the isotopes present in the homogenized region will be merged as a single residual isotope.
<i>nis</i>	number of isotopes present in the homogenized region to be processed.



<i>HISO</i>	array of <b>character*8</b> isotopes alias names to be processed.
<b>REAC</b>	keyword to specify the reaction names to be included in the output microlib. By default, all available reactions are included in the output microlib.
<i>nreac</i>	number of reactions to be included in the output microlib.
<i>HREAC</i>	array of <b>character*8</b> reaction names to be included in the output microlib.
<b>ACTI</b>	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 $1.0 \times cm^{-3}$ in each region. This keyword is followed by nacti material mixture indices, where $nacti \leq maxmix$ .
<b>NONE</b>	keyword to specify that no isotope present in the homogenized region is to be used as activation data.
<i>imixa</i>	array of material mixture indices which contains the isotopes for which activation data is to be generated. $nmix \leq maxmix$ . Even mixture not used in the geometry can be considered here.
<b>ISOTXS</b>	keyword to specify that the set of microscopic cross section generated by the <b>MICR</b> and <b>ACTI</b> command will also be saved on a microscopic group neutron cross section library in the ISOTXS-IV format. This will generate a file for each final region specified by the <b>TAKE</b> or <b>MERG</b> keyword, numbered consecutively ( <b>IFILE</b> ). The name of the file ( <b>NISOTXS</b> ) is built using the command  WRITE(NISOTXS, '(A6,I6.6)') 'ISOTXS', IFILE
<b>ASCII</b>	keyword to specify that the ISOTXS file is created in ascii format. By default, it is created in binary format.
<b>SAVE</b>	keyword to specify that the fluxes, the macroscopic and microscopic cross sections and the volumes corresponding to homogenized regions are to be saved on <i>EDINAM</i> . A <b>MACROLIB</b> is store on a subdirectory of <i>EDITION</i> .
<b>ON</b>	keyword to specify on which directory of <i>EDINAM</i> this information is to be stored.
<i>DIRN</i>	name of the directory on which the above information is to be stored.
<i>idirn</i>	number associated with a directory of <i>EDINAM</i> on which the above information is to be stored. To each number <i>idirn</i> is associated a directory name <b>CDIRN</b> = 'REF-CASE'//CN where CN is a <b>character*4</b> variable defined by WRITE(CN, '(I4)') <i>idirn</i> .
<b>PERT</b>	keyword to specify that first order perturbations for the microscopic cross sections are to be saved on <i>EDINAM</i> .
<b>STAT</b>	keyword to specify that a comparison between the current and a reference set of reaction rates and/or integrated fluxes is to be performed.
<b>ALL</b>	keyword to specify that the relative differences in the reaction rates and the integrated fluxes are to be printed.
<b>RATE</b>	keyword to specify that the relative differences in the reaction rates are to be printed.
<b>FLUX</b>	keyword to specify that the relative differences in the integrated fluxes are to be printed.
<b>DELS</b>	keyword to specify that the absolute differences in the macroscopic cross section are to be printed.

REFE	keyword to specify the directory of <i>EDINAM</i> where the reference data requires for the comparison is stored. When this keyword is absent, the last reaction rates and integrated fluxes saved on <i>EDINAM</i> are used.
DIRO	name of the directory from which the reference information is taken.
idiro	number associated with an directory of <i>EDINAM</i> on which the reference information is stored. To each number <i>idirn</i> is associated a the directory <code>CDIRN='REF-CASE'//CN</code> where <code>CN</code> is a <code>character*4</code> variable defined by <code>WRITE(CN,'(I4)') idirn</code> .
NOHF	keyword to suppress the calculation and edition of the H-factors (sum of all the cross sections producing energy times the energy produced by each reaction). Note that this calculation may be time-consuming. By default, the H-factors are computed and edited if keyword <code>DEPL</code> and associated data is set in module <code>LIB</code> .
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.
MAXR	keyword to specify the number of components in region-related dynamically allocated arrays. If the default value is not sufficient, an error message is issued.
maxpts	user-defined maximum number of components.
DIRE	use the direct flux to perform homogenization or/and condensation (default value).
PROD	use the product of the direct and adjoint flux to perform homogenization or/and condensation. This option is used only in specialized applications such as in the <code>CLIO</code> perturbative analysis formula. <sup>[?]</sup> The homogenization and condensation equations are presented in Sect. 3.9.3. <b>Note:</b> The <i>FLUNAM</i> object must contain both an adjoint and a direct flux solution.
MGEO	keyword to define the name of the macro-geometry, which must appear among the RHS. The macro-geometry is recovered automatically by interface modules such as <code>COMPO</code> : (see Section 3.14) or manually by a <code>CLE-2000</code> statement such as  <pre>GEONAM := EDINAM :: STEP UP 'MACRO-GEOM' ;</pre> where <code>GEONAM</code> and <code>EDINAM</code> are <code>L_GEOM</code> and <code>L_EDIT</code> LCM objects, respectively.
MACGEO	character*12 name of the macro-geometry.
NADF	keyword to deactivate boundary editions.
ALBS	keyword to specify that the boundary flux is to be obtained from relation $\phi_{\text{surf}} = 4J_{\text{out}}/S$ where $J_{\text{out}}$ is the outgoing interface current. The albedo of the geometry are to be taken into account in the complete homogenization process. Thus the <code>MERG</code> and <code>COMP</code> options must be specified. The boundary fluxes are obtained from a calculation using the collision probabilities. This option requires a geometry with <code>VOID</code> (see Section 3.3.2) external boundary conditions to be closed using <code>ALBS</code> in module <code>ASM</code> : (see Section 3.7.1). <sup>[?]</sup>
JOUT	keyword to specify that the boundary flux is to be obtained from relation $\phi_{\text{surf}} = 4J_{\text{out}}/S$ where $J_{\text{out}}$ is the outgoing interface current. The outgoing interface current is recovered by direct homogenization and condensation of the flux unknown components corresponding to external boundary and used with the current iteration method in <i>Eurydice</i> . This keyword is only compatible if a <code>SYBILT</code> : tracking is used and if keyword <code>ARM</code> is set in module <code>ASM</code> : (see Section 3.7.1).
ADFM	keyword to specify that the ADF information is recovered from <code>macrolib</code> in RHS object <i>LIBNAM</i> . ADF information can be defined as explained in Sect. 3.1.2 of module <code>MAC</code> : and recovered in module <code>EDI</code> : for further processing.

ADF	keyword to specify that boundary editions are required. Averaged fluxes are computed over boundary regions.
TYPE	<b>character*4</b> name of the boundary edit corresponding to regions <i>ireg</i> or mixtures <i>imix</i> . Any user-defined name can be used, but some standard names are recognized by module SPH (see Section 3.11.2). Standard names are: = FD.C: corner flux edition; = FD.B: surface (assembly gap) flux edition; = FD.H: row flux edition. These are the first row of surrounding cells in the assembly.
<i>ireg</i>	index of a region of the reference geometry belonging to boundary edition.
<i>imix</i>	index of a material mixture of the reference geometry belonging to boundary edition.

### 3.9.2 Homogenization and condensation with the flux

The cross sections are homogenized over macro-volumes  $V_{\text{merg}}$  and condensed over macro groups  $E_{\text{merg}}$ . We also use  $V_i$  to identify the subset of  $V_{\text{merg}}$  where the isotope  $i$  is defined. The module EDI: produces the following homogenized/condensed information:

**integrated volume:**

$$\bar{V} = \int_{V_{\text{merg}}} dV$$

**macroscopic cross section of type  $x$ :**

$$\bar{\Sigma}_x = \frac{\int_{V_{\text{merg}}} dV \int_{E_{\text{merg}}} dE \Sigma_x(\mathbf{r}, E) \phi(\mathbf{r}, E)}{\int_{V_{\text{merg}}} dV \int_{E_{\text{merg}}} dE \phi(\mathbf{r}, E)}$$

**number density for isotope  $i$ :**

$$\bar{N}_i = \frac{1}{\bar{V}} \int_{V_i} dV N_i(\mathbf{r})$$

where  $N_i(\mathbf{r})$  is the space-dependent number density of isotope  $i$ .

**neutron flux:**

$$\bar{\phi} = \frac{1}{\bar{V}} \int_{V_{\text{merg}}} dV \int_{E_{\text{merg}}} dE \phi(\mathbf{r}, E)$$

**microscopic cross section of type  $x$  for isotope  $i$ :**

$$\begin{aligned} \bar{\sigma}_{x,i} &= \frac{1}{\bar{N}_i} \frac{\int_{V_i} dV \int_{E_{\text{merg}}} dE N_i(\mathbf{r}) \sigma_{x,i}(\mathbf{r}, E) \phi(\mathbf{r}, E)}{\int_{V_{\text{merg}}} dV \int_{E_{\text{merg}}} dE \phi(\mathbf{r}, E)} \\ &= \frac{1}{\bar{N}_i \bar{\phi} \bar{V}} \int_{V_i} dV \int_{E_{\text{merg}}} dE N_i(\mathbf{r}) \sigma_{x,i}(\mathbf{r}, E) \phi(\mathbf{r}, E) . \end{aligned}$$

### 3.9.3 Homogenization and condensation with the flux and adjoint flux

If the PROD keyword is set in data structure 3.9.1, the adjoint flux is introduced as a weighting function in the homogenization and condensation formulas. In this case, the module EDI: produces the following homogenized/condensed information:

**adjoint neutron flux:**

$$\bar{\phi}^* = \frac{1}{\bar{V}} \int_{V_{\text{merg}}} dV \int_{E_{\text{merg}}} dE \phi^*(\mathbf{r}, E)$$

**microscopic transfer cross section for isotope  $i$ :**

$$\bar{\sigma}_{s,i} = \frac{1}{N_i (\bar{\phi}^*)' \bar{\phi} \bar{V}} \int_{V_i} dV \int_{E'_{\text{merg}}} dE' \int_{E_{\text{merg}}} dE N_i(\mathbf{r}) \sigma_{s,i}(\mathbf{r}, E' \leftarrow E) \phi^*(\mathbf{r}, E') \phi(\mathbf{r}, E)$$

with

$$(\bar{\phi}^*)' = \frac{1}{\bar{V}} \int_{V_{\text{merg}}} dV \int_{E'_{\text{merg}}} dE' \phi^*(\mathbf{r}, E')$$

**microscopic cross section of type  $x \neq \text{f}$  for isotope  $i$ :**

$$\bar{\sigma}_{x,i} = \frac{1}{N_i \bar{\phi}^* \bar{\phi} \bar{V}} \int_{V_i} dV \int_{E_{\text{merg}}} dE N_i(\mathbf{r}) \sigma_{x,i}(\mathbf{r}, E) \phi^*(\mathbf{r}, E) \phi(\mathbf{r}, E)$$

**microscopic  $\nu$  times fission cross section for isotope  $i$ :**

$$\bar{\nu} \bar{\sigma}_{\text{f},i} = \frac{1}{N_i \bar{\phi} \bar{V}} \int_{V_i} dV \int_{E_{\text{merg}}} dE N_i(\mathbf{r}) \nu \sigma_{\text{f},i}(\mathbf{r}, E) \phi(\mathbf{r}, E)$$

**fission spectra for isotope  $i$ :**

$$\bar{\chi}_i = \frac{1}{N_i \bar{\phi}^* \bar{V}} \int_{V_i} dV \int_{E_{\text{merg}}} dE N_i(\mathbf{r}) \chi_i(\mathbf{r}, E) \phi^*(\mathbf{r}, E) \quad .$$

Both the macrolib and microlib information is affected by the adjoint weighting. However, users should be advised that this operation may have some undesirable effects on the fission spectrum normalization. Its use must therefore be limited to specialized applications where the adjoint weighting is theoretically required. This is the case, for example, with the CLIO perturbative analysis method.<sup>[?]</sup>

### 3.10 The EV0: module

The **EV0**: module performs the burnup calculations. The depletion equations for the various isotope of the MICROLIB are solved using the burnup chains also present in the MICROLIB. Both in-core and out-of-core calculations can be considered. For in-core depletion calculations, one assumes linear flux variation over each irradiation period (time stage). The initial (and possibly final) flux distributions are recovered from previous **FLU**: calculations. In-core depletion can be performed at constant flux or constant power (expressed in MW/Tonne of initial heavy elements) but these values can undergo step variations from one time stage to another. All the information required for successive burnup calculation is stored on the PyLCM BURNUP data structure. Thus it is possible at any point in time to return to a previous time step and restart the calculations.

In each burnup mixture of the unit cell, the depletion of  $K$  isotopes over a time stage  $(t_0, t_f)$  follows the following equation:

$$\frac{dN_k}{dt} + N_k(t) \Lambda_k(t) = S_k(t) \quad ; \quad k = 1, K \quad (3.12)$$

with

$$\Lambda_k(t) = \lambda_k + \langle \sigma_{a,k}(t) \phi(t) \rangle, \quad (3.13)$$

$$S_k(t) = \sum_{l=1}^L Y_{kl} \langle \sigma_{f,l}(t) \phi(t) \rangle N_l(t) + \sum_{l=1}^K m_{kl}(t) N_l(t), \quad (3.14)$$

$$\langle \sigma_{x,l}(t) \phi(t) \rangle = \int_0^\infty \sigma_{x,l}(u) \phi(t, u) du \quad (3.15)$$

and

$$\sigma_{x,k}(t, u) \phi(t, u) = \sigma_{x,k}(t_0, u) \phi(t_0, u) + \frac{\sigma_{x,k}(t_f, u) \phi(t_f, u) - \sigma_{x,k}(t_0, u) \phi(t_0, u)}{t_f - t_0} (t - t_0) \quad (3.16)$$

where

- $K$  = number of depleting isotopes
- $L$  = number of fissile isotopes producing fission products
- $N_k(t)$  = time dependant number density for  $k$ -th isotope
- $\lambda_k$  = radioactive decay constant for  $k$ -th isotope
- $\sigma_{x,k}(t, u)$  = time and lethargy dependant microscopic cross section for nuclear reaction  $x$  on  $k$ -th isotope.  $x=a$ ,  $x=f$  and  $x=\gamma$  respectively stands for absorption, fission and radiative capture cross sections
- $\phi(t, u)$  = time and lethargy dependant neutron flux
- $Y_{kl}$  = fission yield for production of fission product  $k$  by fissile isotope  $l$
- $m_{kl}(t)$  = radioactive decay constant or  $\langle \sigma_{x,l}(t) \phi(t) \rangle$  term for production of isotope  $k$  by isotope  $l$ .

Depleting isotopes with  $\Lambda_k(t_0) [t_f - t_0] \geq \text{valexp}$  and  $\Lambda_k(t_f) [t_f - t_0] \geq \text{valexp}$  are considered to be at saturation. They are described by making  $\frac{dN_k}{dt} = 0$  in Eq. (3.12) to obtain

$$N_k(t) = \frac{S_k(t)}{\Lambda_k(t)} \quad ; \quad \text{if } k \text{ is at saturation.} \quad (3.17)$$

If the keyword **SAT** is set, beginning-of-stage and end-of-stage Dirac contributions are added to the previous equation:

$$N_k(t) = \frac{1}{\Lambda_k(t)} [a\delta(t - t_0) + S_k(t) + b\delta(t - t_f)] \quad ; \quad \text{if } k \text{ is at saturation} \quad (3.18)$$

where  $a$  and  $b$  are chosen in order to satisfy the time integral of Eq. (3.12):

$$N_k(t_f^+) - N_k(t_0^-) + \int_{t_0^-}^{t_f^+} N_k(t) \Lambda_k(t) dt = \int_{t_0^-}^{t_f^+} S_k(t) dt \quad (3.19)$$

It is numerically convenient to chose the following values of  $a$  and  $b$ :

$$a = N_k(t_0^-) - \frac{S_k(t_0^+)}{\Lambda_k(t_0^+)} \quad (3.20)$$

and

$$b = \frac{S_k(t_0^+)}{\Lambda_k(t_0^+)} - \frac{S_k(t_f^+)}{\Lambda_k(t_f^+)} \quad (3.21)$$

The numerical solution techniques used in the **EVO**: module are the following. Very short period isotopes are taken at saturation and are solved apart from non-saturating isotopes. If an isotope is taken at saturation, all its parent isotopes, other than fissiles isotopes, are also taken at saturation. Isotopes at saturation can produce daughter isotopes using decay *and/or* neutron-induced reactions.

The lumped depletion matrix system containing the non-saturating isotopes is solved using either a fifth order Cash-Karp algorithm or a fourth order Kaps-Rentrop algorithm<sup>[?]</sup>, taking care to perform all matrix operations in sparse matrix algebra. Matrices  $[m_{kl}(t_0)]$  and  $[m_{kl}(t_f)]$  are therefore represented in diagonal banded storage and kept apart from the yield matrix  $[Y_{kl}]$ . Every matrix multiplication or linear system solution is obtained via the LU algorithm.

The solution of burnup equations is affected by the flux normalization factors. **DRAGON** can perform out-of-core or in-core depletion with a choice between two normalization techniques:

1. Constant flux depletion. In this case, the lethargy integrated fluxes at beginning-of-stage and end-of-stage are set to a constant  $F$ :

$$\int_0^\infty \phi(t_0, u) du = \int_0^\infty \phi(t_f, u) du = F \quad (3.22)$$

2. Constant power depletion. In this case, the power released per initial heavy element at beginning-of-stage and end-of-stage are set to a constant  $W$ .

$$\begin{aligned} \sum_{k=1}^K [\kappa_{f,k} \langle \sigma_{f,k}(t_0) \phi(t_0) \rangle + \kappa_{\gamma,k} \langle \sigma_{\gamma,k}(t_0) \phi(t_0) \rangle] N_k(t_0) &= \\ \sum_{k=1}^K [\kappa_{f,k} \langle \sigma_{f,k}(t_f) \phi(t_f) \rangle + \kappa_{\gamma,k} \langle \sigma_{\gamma,k}(t_f) \phi(t_f) \rangle] N_k(t_f) &= C_0 W \end{aligned} \quad (3.23)$$

where

$$\begin{aligned} \kappa_{f,k} &= \text{energy (MeV) released per fission of the fissile isotope } k \\ \kappa_{\gamma,k} &= \text{energy (MeV) released per radiative capture of isotope } k \\ C_0 &= \text{conversion factor (MeV/MJ) multiplied by the mass of initial heavy elements} \\ &\quad \text{expressed in metric tonnes} \end{aligned}$$

The end-of-stage power is function of the number densities  $N_k(t_f)$ ; a few iterations will therefore be required before the end-of-stage power released can be set equal to the desired value. Note that there is no warranties that the power released keep its desired value at every time during the stage; only the beginning-of-stage and end-of-stage are set.

Whatever the normalisation technique used, DRAGON compute the exact burnup of the unit cell (in MW per tonne of initial heavy element) by adding an additional equation in the depletion system. This value is the local parameter that should be used to tabulate the output cross sections.

The general format of the data which is used to control the execution of the **EVO:** module is the following:

Table 49: Structure (**EVO:**)

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

where

<i>BRNNAM</i>	<b>character*12</b> name of the BURNUP data structure that will contain the depletion history as modified by the depletion module. If <i>BRNNAM</i> appears on both LHS and RHS, it is updated; otherwise, it is created.
<i>MICNAM</i>	<b>character*12</b> name of the MICROLIB containing the microscopic cross sections at save point <i>xts</i> . <i>MICNAM</i> is modified to include an embedded MACROLIB containing the updated macroscopic cross sections at set point <i>xtr</i> . If <i>MICNAM</i> appears on both LHS and RHS, it is updated; otherwise, the internal library <i>OLDMIC</i> is copied in <i>MICNAM</i> and <i>MICNAM</i> is updated. It is possible to assign different MICROLIB to different save points of the depletion calculation. In this case, the microscopic reaction rates will be linearly interpolated/extrapolated between points <i>xti</i> and <i>xtf</i> .
<i>OLDMIC</i>	<b>character*12</b> name of a read-only MICROLIB that is copied in <i>MICNAM</i> .
<i>FLUNAM</i>	<b>character*12</b> name of a read-only FLUXUNK at save point <i>xts</i> . This information is used for in-core depletion cases. This information is not required for out-of-core depletion cases. Otherwise, it is mandatory
<i>TRKNAM</i>	<b>character*12</b> name of a read-only TRACKING constructed for the depleting geometry and consistent with object <i>FLUNAM</i> .
<i>POWNAM</i>	<b>character*12</b> name of a read-only POWER object (generated by DONJON) at save point <i>xts</i> . This information is used for micro-depletion cases.
( <b>descevo</b> )	structure containing the input data to this module (see Section 3.10.1).

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

### 3.10.1 Data input for module EVO:

Table 50: Structure (**descevo**)

```
[ EDIT iprint ]
```

continued on next page

Structure (**descevo**)

continued from last page

```

[ { SAVE xts { S | DAY | YEAR } { FLUX flux | POWR fpower | W/CC apower } | NOSA } ]
[ EPS1 valeps1 ] [ EPS2 valeps2 ] [ { EXPM valexp | SATOFF } ]
[ H1 valh1 ] [ { RUNG | KAPS } ]
[ { TIXS | TDXS } ] [ { NOEX | EXTR } ] [ { NOGL | GLOB } ] [ { NSAT | SAT } ] [ { NODI | DIRA } ]
[ { FLUX_FLUX | FLUX_MAC | FLUX_POW } ] [ { CHAIN | PIFI } ]
[ DEPL { xti xtf | dxt } { S | DAY | YEAR } { COOL | FLUX flux | POWR fpower | W/CC apower | KEEP } ]
[ SET xtr { S | DAY | YEAR } ]
[ MIXB [[ mixbrn ]] ] [ MIXP [[ mixpwr ]] ]
;

```

where

EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	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 current isotopic concentration and the microscopic reaction rates resulting from the last transport calculation will be normalized and stored on <i>BRNNAM</i> in a sub-directory corresponding to a specific time. By default this data is stored at a time corresponding to <i>xti</i> .
NOSA	keyword to specify that the current isotopic concentration and the results of the last transport calculation will not be stored on <i>BRNNAM</i> . By default this data is stored at a time corresponding to <i>xti</i> .
SET	keyword used to recover the isotopic concentration already stored on <i>BRNNAM</i> from a sub-directory corresponding to a specific time. By default this data is recovered from a time corresponding to <i>xtf</i> .
DEPL	keyword to specify that a burnup calculation between an initial and a final time must be performed. In the case where the SAVE keyword is absent, the initial isotopic concentration will be stored on <i>BRNNAM</i> 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 used to update <i>MICNAM</i> .
<i>xti</i>	initial time associated with the burnup calculation. The name of the sub-directory where this information is stored will be given by 'DEPL-DAT'//CNN where CNN is a <b>character*4</b> variable defined by <code>WRITE(CNN,'(I4.4)')</code> INN where INN is an index associated with the time <i>xti</i> . The initial values are recovered from this sub-directory in <i>BRNNAM</i> .
<i>xtf</i>	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 <i>xti</i> input.
<i>dxt</i>	time interval for the burnup calculation. The initial time <i>xti</i> in this case is taken as the final time reached at the last depletion step. If this is the first depletion step, <i>xti</i> = 0.
<i>xts</i>	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 <i>xti</i> input. By default (fixed default) <i>xts</i> = <i>xti</i> .
<i>xtr</i>	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 <i>xti</i> input. By default (fixed default) <i>xtr</i> = <i>xtf</i> .



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 a constant flux burnup calculation is to be performed.
<i>flux</i>	flux expressed in $cm^{-2}s^{-1}$ .
POWR	keyword to specify that a constant fuel power depletion calculation is to be performed. The energy released outside the fuel (e.g., by $(n,\gamma)$ reactions) is <i>not</i> taken into account in the flux normalization, unless the <b>GLOB</b> option is set.
<i>fpower</i>	fuel power expressed in $KW Kg^{-1} = MW tonne^{-1}$ .
W/CC	keyword to specify that a constant assembly power depletion calculation is to be performed. The energy released outside the fuel (e.g., by $(n,\gamma)$ reactions) is always taken into account in the flux normalization.
<i>apower</i>	assembly power density expressed in $W/cm^3$ (Power per unit assembly volume).
KEEP	keyword to specify that the flux is used without been normalized. This option is useful in cases where the flux was already normalized before the call to <b>EVO:</b> module.
EPS1	keyword to specify the tolerance used in the algorithm for the solution of the depletion equations.
<i>valeps1</i>	the tolerance used in the algorithm for the solution of the depletion equations. The default value is $valeps1=1.0 \times 10^{-5}$ .
EPS2	keyword to specify the tolerance used in the search algorithm for a final fixed power (used if the <b>POWR</b> or <b>W/CC</b> option is activated).
<i>valeps2</i>	the tolerance used in the search algorithm for a final fixed power. The default value is $valeps2=1.0 \times 10^{-4}$ .
EXPM	keyword to specify the selection criterion for non-fissile isotopes that are at saturation.
<i>valexp</i>	the isotopes for which $\lambda \times (xtf - xti) \geq valexp$ will be treated by a saturation approximation. Here, $\lambda$ is the sum of the radioactive decay constant and microscopic neutron absorption rate. The default value is $valexp=80.0$ . In order to remove the saturation approximation for all isotopes set <i>valexp</i> to a very large number such as $1.0 \times 10^5$ . On the other way, the saturation approximation can be set for a specific isotope by using the keyword <b>SAT</b> in Sect. 3.2.3 (module <b>LIB:</b> ).
SATOFF	keyword to remove the saturation approximation for all isotopes even if <b>SAT</b> keyword was set in Sect. 3.2.3 (module <b>LIB:</b> ).
H1	keyword to specify an estimate of the relative width of the time step used in the solution of burnup equations.
<i>valh1</i>	relative width of the time step used in the solution of burnup equations. An initial time step of $\Delta_t = valh1 \times (xtf - xti)$ is used. This value is optimized dynamically by the program. The default value is $valh1=1.0 \times 10^{-4}$ .
RUNG	keyword to specify that the solution will be obtained using the 5 <sup>th</sup> order Cash-Karp algorithm.
KAPS	keyword to specify that the solution will be obtained using the 4 <sup>th</sup> order Kaps-Rentrop algorithm. This is the default value.

TIXS	keyword that specified that time independent cross sections will be used. 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 to suppress the linear extrapolation of the microscopic reaction rates in the solution of the burnup equations.
EXTR	keyword to perform a linear extrapolation of the microscopic reaction rates, using the available information preceding the initial time <i>xti</i> . This is the default option.
NOGL	keyword to compute the burnup using the energy released in fuel only. This is the default option.
GLOB	keyword to compute the burnup using the energy released in the complete geometry. This option has an effect only in cases where some energy is released outside the fuel (e.g., due to $(n,\gamma)$ reactions). This option affects both the meaning of <i>fpower</i> (given after the key-word <b>POWR</b> ) and the value of the burnup, as computed by <b>EVO</b> .
NSAT	save the non-saturated initial number densities in the BURNUP object <i>BRNNAM</i> (default value)
SAT	save the saturated initial number densities in the BURNUP object <i>BRNNAM</i>
NODI	select Eq. (3.17) to compute the saturated number densities (default value)
DIRA	select Eq. (3.18) to compute the saturated number densities
FLUX_FLUX	recover the neutron flux from <i>FLUNAM</i> object (default option)
FLUX_MAC	recover the neutron flux from embedded macrolib present in <i>MICNAM</i> or <i>OLDMIC</i> object. This option is useful to deplete in cases where the neutron flux is obtained from a Monte Carlo calculation.
FLUX_POW	recover the neutron flux from the <b>POWER</b> object named <i>POWNAM</i> generated in <b>DONJON</b> . This option is useful in micro-depletion cases. The neutron flux recovered from <i>POWNAM</i> is generally normalized to the power of the full core. It is therefore recommended to use the <b>KEEP</b> option in <b>DEPL</b> data structure.
CHAIN	recover the fission yield data from 'DEPL-CHAIN' directory of <i>MICNAM</i> or <i>OLDMIC</i> object (default option). With this option, the fission yield data is the same in all material mixtures.
PIFI	recover the fission yield data from 'PIFI' and 'PYIELD' records present in isotopic directories of <i>MICNAM</i> or <i>OLDMIC</i> object. With this option, the fission yield data is mixture-dependent. This option is useful in micro-depletion cases.
MIXB	keyword to select depleting material mixtures. By default, all mixtures with depleting isotopes are set as depleting.
<i>mixbrn</i>	indices of depleting material mixtures.
MIXP	keyword to select material mixtures producing power. By default, <ul style="list-style-type: none"> <li>• if <b>MIXB</b> is not set, all mixtures with isotopes producing power are set as producing power</li> <li>• if <b>MIXB</b> is set, the same mixtures <i>mixbrn</i> are set as producing power.</li> </ul>
<i>mixpwr</i>	indices of material mixtures producing power.

### 3.10.2 Power normalization in EVO:

Flux-induced depletion is dependent of the flux or power normalization factor given after key-words **FLUX**, **POWR** or **W/CC**. The depletion steps, given after key-words **SAVE**, **DEPL** or **SET**, are set in time units. Traditionally, the power normalization factor is given in  $MW \text{ tonne}^{-1}$  and the depletion steps are given in  $MWday \text{ tonne}^{-1}$ . However, a confusion appear in cases where some energy is released outside the fuel (e.g., due to  $(n,\gamma)$  reactions).

The accepted rule and default option in **EVO**: is to compute the burnup steps in units of  $MWday \text{ tonne}^{-1}$  by considering only the energy released in fuel (and only the initial mass of the heavy elements present in fuel). However, it is also recommended to provide a normalization power taking into account the *total* energy released in the global geometry. The **GLOB** option can be use to change this rule and to use the energy released in the complete geometry to compute the burnup. However, this is not a common practice, as it implies a non-usual definition of the burnup. A more acceptable solution consists in setting the normalization power in power per unit volume of the complete geometry using the key-word **W/CC**. The value of *apower* can be computed from the linear power  $f_{lin}$  (expressed in  $Mev \text{ s}^{-1} \text{ cm}^{-1}$ ) using:

$$apower = \frac{f_{lin} \cdot 1.60207 \times 10^{-13}}{V_{assmb}} \quad (3.24)$$

where  $V_{assmb}$  is the 2-D lumped volume of the assembly expressed in  $cm^2$ .

The corresponding normalization factor  $f_{burnup}$  in  $MW \text{ tonne}^{-1}$  is given as

$$f_{burnup} = \frac{apower}{D_g F_{power}} \quad (3.25)$$

where  $D_g$  is the mass of heavy elements per unit volume of the complete geometry ( $g \text{ cm}^{-3}$ ) and  $F_{power}$  is the ratio of the energy released in the complete geometry over the energy released in fuel. Numerical values of  $D_g$  and  $f_{power}$  are computed by **EVO**: when the parameter *iprint* is greater or equal to 2. The burnup  $B$  corresponding to an elapsed time  $\Delta t$  is therefore given as

$$B = f_{burnup} \Delta t \quad (3.26)$$

where  $B$  is expressed in  $MWday \text{ tonne}^{-1}$  and  $\Delta t$  is expressed in *day*.

The unit of the reaction rates depends on the normalization applied to the flux. This normalization takes place after the flux calculation, using the **EVO**: module. Here is an example:

```
INTEGER istep := 1 ;
REAL Tend    := 0.0 ;
REAL Fuelpwr := 38.4 ; ! expressed in MW/tonne

BURN MICROLIB := EVO: MICROLIB FLUX TRACKN ::
  EDIT 0
  SAVE <<Tend>> DAY POWR <<Fuelpwr>>
;
```

where **BURN** is the burnup object, **MICROLIB** is the Microlib used to compute the flux, **FLUX** is the flux object and **TRACKN** is the tracking object used to compute the flux. After this call, the record '**FLUX-NORM**' in **BURN** contains a unique real number, equal to the flux normalization factor. If **MICROLIB** is obtained using the **LIB**: module, the **DEPL** keyword with following data must be set (see Section 3.2.1). Unfortunately, the normalization factor is kept aside and is not applied to the flux present in object **FLUX**. In fact, only the advanced post-processing modules **COMPO**: (see Section 3.14) and **SAP**: (see Section 3.19) are making use of this normalization factor.

### 3.11 The SPH: module

The *superhomogénéisation* (SPH) equivalence technique is based on the calculation of a set of *equivalence factors*  $\{\mu_{m,k}, m \in C_m \text{ and } k \in M_k\}$ , where  $C_m$  and  $M_k$  is a macro region and a coarse energy group of a full-core or macro calculation (see Sect. 4.4 of Ref. ?). These equivalence factors are computed in such a way that a macro calculation made over  $C_m$  and  $M_k$  with a simplified transport operator leads to the same leakage and reaction rates as a reference calculation performed without homogenization and with a fine group discretization.

The SPH correction is applied differently, depending on the type of macro-calculation:

- In the case where the macro-calculation is done with the diffusion theory, neutron balance is satisfied if the SPH correction is applied as follows:

$$\nabla \cdot \mathbf{J}_g(\mathbf{r}) + \mu_g \Sigma_g(\mathbf{r}) \frac{\phi_g(\mathbf{r})}{\mu_g} = \frac{\chi_g}{k_{\text{eff}}} \sum_{h=1}^G \mu_h \nu \Sigma_{f,h}(\mathbf{r}) \frac{\phi_h(\mathbf{r})}{\mu_h} + \sum_{h=1}^G \mu_h \Sigma_{s0,g \leftarrow h}(\mathbf{r}) \frac{\phi_h(\mathbf{r})}{\mu_h} \quad (3.27)$$

and

$$\mathbf{J}_g(\mathbf{r}) = -\mu_g D_g(\mathbf{r}) \frac{\nabla \phi_g(\mathbf{r})}{\mu_g}. \quad (3.28)$$

In conclusion:

- Diffusion coefficients and all  $P_0$  cross sections (including the total cross section NTOT0) must be multiplied by  $\mu_g$ .
- Scattering matrix terms  $\Sigma_{s0,g \leftarrow h}(\mathbf{r})$  must be multiplied by  $\mu_h$ .
- Fluxes (such as NWT0 and FLUX-INTG) must be divided by  $\mu_g$ .
- In the case where the macro-calculation is done with the simplified  $P_n$  method, the neutron balance is satisfied if the SPH correction is applied on even parity equations as follows:[?]

$$\mu_g \Sigma_{0,g}(\mathbf{r}) \frac{\phi_{0,g}(\mathbf{r})}{\mu_g} + \nabla \cdot \phi_{1,g}(\mathbf{r}) = \frac{\chi_g}{k_{\text{eff}}} \sum_{h=1}^G \mu_h \nu \Sigma_{f,h}(\mathbf{r}) \frac{\phi_{0,h}(\mathbf{r})}{\mu_h} + \sum_{h=1}^G \mu_h \Sigma_{s0,g \leftarrow h}(\mathbf{r}) \frac{\phi_{0,h}(\mathbf{r})}{\mu_h} \quad (3.29)$$

$$\frac{2\ell}{4\ell+1} \nabla \cdot \phi_{2\ell-1,g}(\mathbf{r}) + \mu_g \Sigma_{0,g}(\mathbf{r}) \frac{\phi_{2\ell,g}(\mathbf{r})}{\mu_g} + \frac{2\ell+1}{4\ell+1} \nabla \cdot \phi_{2\ell+1,g}(\mathbf{r}) = \sum_{h=1}^G \mu_h \Sigma_{s2\ell,g \leftarrow h}(\mathbf{r}) \frac{\phi_{2\ell,h}(\mathbf{r})}{\mu_h} \quad (3.30)$$

and on odd-parity equations as follows:

$$\frac{2\ell+1}{4\ell+3} \nabla \frac{\phi_{2\ell,g}(\mathbf{r})}{\mu_g} + \frac{\Sigma_{1,g}(\mathbf{r})}{\mu_g} \phi_{2\ell+1,g}(\mathbf{r}) + \frac{2\ell+2}{4\ell+3} \nabla \frac{\phi_{2\ell+2,g}(\mathbf{r})}{\mu_g} = \sum_{h=1}^G \frac{\Sigma_{s2\ell+1,g \leftarrow h}(\mathbf{r})}{\mu_g} \phi_{2\ell+1,h}(\mathbf{r}) \quad (3.31)$$

where  $\ell \geq 1$ .

In conclusion:

- All  $P_0$  cross sections (including the total cross section NTOT0 in the even-parity equations) must be multiplied by  $\mu_g$ .
- The total cross section NTOT1 in the odd-parity equations must be divided by  $\mu_g$ .
- Scattering matrix terms  $\Sigma_{s\ell,g \leftarrow h}(\mathbf{r})$  with  $\ell$  even must be multiplied by  $\mu_h$ .
- Scattering matrix terms  $\Sigma_{s\ell,g \leftarrow h}(\mathbf{r})$  with  $\ell$  odd must be divided by  $\mu_g$ .
- Even parity fluxes (such as NWT0 and FLUX-INTG) must be divided by  $\mu_g$ .
- Odd parity fluxes (such as NWT1 and FLUX-INTG-P1) are not modified.

- In the case where the macro-calculation is done in transport theory, but not with a  $P_n$ -type method, the macroscopic total cross section *is not modified*, and the even-odd corrections consistent with the simplified  $P_n$  method are reported to the macroscopic within-group scattering cross sections. They are now corrected as<sup>[?]</sup>

$$\tilde{\Sigma}_{s2\ell, g \leftarrow g}(\mathbf{r}) = \mu_g \Sigma_{s2\ell, g \leftarrow g}(\mathbf{r}) + (1 - \mu_g) \Sigma_{0, g}(\mathbf{r}) \quad (3.32)$$

and

$$\tilde{\Sigma}_{s2\ell+1, g \leftarrow g}(\mathbf{r}) = \frac{\Sigma_{s2\ell+1, g \leftarrow g}(\mathbf{r})}{\mu_g} + \left(1 - \frac{1}{\mu_g}\right) \Sigma_{1, g}(\mathbf{r}) \quad (3.33)$$

where  $\ell \geq 0$ .

Other cross sections and scattering matrix terms are corrected the same way as for the simplified  $P_n$  method.

### 3.11.1 Data input for module SPH:

The **SPH:** module perform a SPH equivalence calculation using information recovered in a macrolib and apply SPH factors to the corresponding EDITION (**L\_EDIT**), MICROLIB (**L\_LIBRARY**), MACROLIB (**L\_MACROLIB**) or SAPHYB (**L\_SAPHYB**) object. This module is also useful to extract a corrected or non-corrected MICROLIB or MACROLIB from the first RHS object. The calling specification is:

Table 51: Structure (**SPH:**)

```
{ EDINEW | LIBNEW | MACNEW | SAPNEW | CPONEW | EDINAM | LIBNAM | MACNAM
  | SAPNAM | CPONAM }
:= SPH: { EDINAM | LIBNAM | MACNAM | SAPNAM | CPONAM }
[ TRKNAM [ TRKFIL ] ] [ FLUNAM ]
:: (descsph)
```

where

<b>EDINEW</b>	<b>character*12</b> name of the new EDITION data structure containing SPH-corrected information (see Section 3.9). In this case, an existing EDITION data structure must appear on the RHS.
<b>LIBNEW</b>	<b>character*12</b> name of the new MICROLIB data structure containing SPH-corrected information (see Section 3.2). In this case, an existing EDITION, MICROLIB or MULTICOMPO data structure must appear on the RHS.
<b>MACNEW</b>	<b>character*12</b> name of the new MACROLIB data structure containing SPH-corrected information (see Section 3.1).
<b>SAPNEW</b>	<b>character*12</b> name of the new SAPHYB data structure containing SPH information (see Section 3.19). In this case, data structure <b>SAPNAM</b> must appear on the RHS.
<b>CPONEW</b>	<b>character*12</b> name of the new MULTICOMPO data structure containing SPH-corrected information (see Section 3.14). In this case, data structure <b>CPONAM</b> must appear on the RHS.
<b>EDINAM</b>	<b>character*12</b> name of the existing EDITION data structure where the edition information is recovered (see Section 3.9).

<i>LIBNAM</i>	<b>character*12</b> name of the existing MICROLIB data structure where the edition information is recovered (see Section 3.2).
<i>MACNAM</i>	<b>character*12</b> name of the existing MACROLIB data structure where the edition information is recovered (see Section 3.1).
<i>SAPNAM</i>	<b>character*12</b> name of the existing SAPHYB data structure where the edition information is recovered (see Section 3.19).
<i>CPONAM</i>	<b>character*12</b> name of the existing MULTICOMPO data structure where the edition information is recovered (see Section 3.14).
<i>TRKNAM</i>	<b>character*12</b> name of the existing TRACKING data structure containing the tracking of the macro-geometry (see Section 3.4). This object is compulsory only if a macro-calculation is to be performed by module SPH:.
<i>TRKFIL</i>	<b>character*12</b> name of the existing sequential binary tracking file used to store the tracks lengths of the macro-geometry. This file is given if and only if it was required in the previous tracking module call (see Section 3.4).
<i>FLUNAM</i>	<b>character*12</b> name of an initialization flux used to start SPH iterations (see Section 3.8). By default, a flat estimate of the flux is used.
<b>(descsph)</b>	structure containing the input data to this module (see Section 3.11.2).

Note: Saphyb files generated by APOLLO2 don't have a signature. If such a Saphyb is given as input to module SPH:, a signature must be included before using it. The following instruction can do the job:

```
Saphyb := UTL: Saphyb :: CREA SIGNATURE 3 = 'L_SA' 'PHYB' ' ' ;
```

### 3.11.2 Specification for the type of equivalence calculation

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.<sup>[?, ?, ?]</sup> This structure is defined as:

Table 52: Structure **(descsph)**

```
[ EDIT iprint ]
[[ STEP { UP NOMDIR | AT index } ]]
[ { IDEM | MACRO | MICRO } ]
[ { OFF | SPRD [ nmerge ngcond (sph(i), i=1, nmerge×ngcond) ] | HOMO | ALBS } ]
[ { PN | SN [ BELL ] } ]
[ { STD | SELE_ALB | SELE_FD | SELE_MWG | SELE_EDF | ASYM mixs } ] [ ARM ]
[ ITER [ maxout ] [ epsout ] ]
[ MAXNB maxnb ]
[ EQUI TEXT4 [ LOCNAM TEXT80 ] ]
[ LEAK b2 ]
```

where

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

<i>iprint</i>	index used to control the printing of this module. The <i>iprint</i> parameter is important for adjusting the amount of data that is printed by this calculation step.
STEP	keyword used to set a specific elementary calculation from the first RHS.
UP	keyword used to select an elementary calculation located in a subdirectory of <i>EDINAM</i> or <i>CPONAM</i> . By default, <ul style="list-style-type: none"> <li>• the sub-directory name stored in record 'LAST-EDIT' is selected if <i>EDINAM</i> is defined at RHS.</li> <li>• the sub-directory 'default' is selected if <i>CPONAM</i> is defined at RHS.</li> </ul>
NOMDIR	name of an existing sub-directory of <i>EDINAM</i> or <i>CPONAM</i> .
AT	keyword used to select the <i>index</i> -th elementary calculation in <i>SAPNAM</i> or <i>CPONAM</i> .
<i>index</i>	index of the elementary calculation.
IDEM	keyword to force the production of a LCM object of the same type as the RHS.
MACRO	keyword to force the production of a macrolib at LHS.
MICRO	keyword to force the production of a microlib at LHS.
OFF	keyword to specify the SPH factors are all set to 1.0, meaning no correction. This keyword is useful to get rid of a SPH correction which have been set previously. By default, the PN or SN option is activated.
SPRD	keyword to specify that the SPH factors are read from input (if <i>nmerge</i> , <i>ngcond</i> and <i>sph</i> are set) or recovered from a RHS object (otherwise).
<i>nmerge</i>	number of regions.
<i>ngcond</i>	number of energy groups.
<i>sph(i)</i>	initial value of each SPH factor in each mixture (inner loop) and each group (outer loop).
HOMO	keyword to specify that the SPH factors are uniform over the complete macro-geometry. This option is generally used with a complete homogenization of the reference geometry, obtained using option MERG COMP. In this case the neutron flux (transport or diffusion) will be uniform, which allows the SPH factors to be obtained (one per macro-group) using a non-iterative strategy. For a given macro-group the SPH factor will be equal to the ratio between the average flux of the region and the surface flux if the SELE option is used otherwise the SPH factor are all set equal to 1.0 (no correction). The SELE option allows an SPH factor equal to the inverse of the discontinuity factor to be calculated.
ALBS	keyword to specify that the albedo of the geometry are to be taken into account in the complete homogenization process. Thus the MERG and COMP options must be specified. The SPH factors are obtained using a transport-transport equivalence based on a calculation using the collision probabilities. This option requires a geometry with VOID (see Section 3.3.2) external boundary conditions to be closed using ALBS in modules ASM: (see Section 3.7.1). <sup>[7]</sup>
PN	keyword to activate a calculation of heterogeneous SPH factors based on a converging series of macro-calculations with the correction strategy of Eqs. (3.27) to (3.31). This is the default option if the macro-calculation is of diffusion, PN or SPN type. A normalization condition must be set if the macro-geometry has no boundary leakage ( <i>fundamental mode</i> condition). If boundary leakage is present, no normalization condition is used but the SPH iterations are difficult to converge in this case.

**SN** keyword to activate a calculation of heterogeneous SPH factors based on a converging series of macro-calculations with the correction strategy of Eqs. (3.32) and (3.33). This is the default option if the macro-calculation is of PIJ, IC, SN or MOC type. A normalization condition must be set if the macro-geometry has no boundary leakage (*fundamental mode* condition). If boundary leakage is present, no normalization condition is used but the SPH iterations are difficult to converge in this case.

**BELL** keyword to activate the Bell procedure to accelerate the convergence of the SPH factors. This feature is currently available with macro-calculations of PIJ type.<sup>[?]</sup>

**STD** keyword to specify the use of flux-volume normalization for the SPH factors (default option). In each macro-group, the macro-fluxes in macro regions  $i$  are normalized using

$$\tilde{\phi}_i = \phi_i \frac{\bar{\phi}_{\text{ref}}}{\bar{\phi}_{\text{mc}}}$$

where  $\bar{\phi}_{\text{ref}}$  is the averaged volumic flux of the reference calculation and  $\bar{\phi}_{\text{mc}}$  is the averaged volumic flux of the macro-calculation. Using this definition, the averaged SPH factor is equal to one.

**SELE\_ALB** keyword to specify the use of Selengut normalization for the SPH factors. It is necessary to know the averaged surfacic flux of the reference calculation. Two possibilities exist:

- We use collision probabilities. We define the reference geometry with **VOID** external boundary conditions (see Section 3.3.2) and to close the region for the collision probability calculations using the **ALBS** option (see Section 3.7.1).
- We perform a flux calculation with the current iteration method in Eurydice. This option is only available if a **SYBILT**: tracking is used and if keyword **ARM** is set in module **ASM**: (see Section 3.7.1).

**SELE\_FD** keyword to specify the use of Selengut normalization for the SPH factors. It is necessary to know the averaged surfacic flux of the reference calculation. This value can be obtained by defining a small region near boundary in the reference geometry and by using the **ADF\_FD-B** data structure in Section 3.9.1.

In each macro-group, the macro-fluxes in macro regions  $i$  are normalized using

$$\tilde{\phi}_i = \phi_i \frac{\phi_{\text{ref}}^{\text{gap}}}{\phi_{\text{mc}}}$$

where  $\phi_{\text{ref}}^{\text{gap}}$  is the averaged surfacic flux of the reference calculation. Using this definition, the averaged SPH factor is equal to

$$\bar{\mu} = \frac{\bar{\phi}_{\text{ref}}}{\phi_{\text{ref}}^{\text{gap}}}.$$

**SELE\_MWG** keyword to specify the use of Selengut *macro calculation water gap* normalization for the SPH factors.<sup>[?]</sup> It is necessary to know the averaged surfacic flux of the reference and that of the *macro* calculations. This reference value can be obtained by defining a small region near boundary in the reference geometry and by using the **ADF\_FD-B** data structure in Section 3.9.1.

In each macro-group, the macro-fluxes in macro regions  $i$  are normalized using

$$\tilde{\phi}_i = \phi_i \frac{\phi_{\text{ref}}^{\text{gap}}}{\phi_{\text{mc}}^{\text{surf}}}$$

where  $\phi_{\text{ref}}^{\text{gap}}$  is the averaged surfacic flux of the reference calculation and  $\phi_{\text{mc}}^{\text{surf}}$  is the averaged surfacic flux of the macro calculation. Using this definition, the averaged SPH factor is equal to

$$\bar{\mu} = \frac{\bar{\phi}_{\text{ref}} \phi_{\text{mc}}^{\text{surf}}}{\phi_{\text{mc}} \phi_{\text{ref}}^{\text{gap}}}.$$



**SELE\_EDF** keyword to specify the use of generalized Selengut normalization for the SPH factors.<sup>[?]</sup> It is necessary to know the averaged surfacic flux and the averaged volumic flux in a row of cells of the reference calculation. The surfacic flux is obtained as with the **SELE** option. The value of the volumic flux in a row of cells is computed using index information from the **ADF\_FD\_H** data structure in Section 3.9.1.

In each macro-group, the macro-fluxes in macro regions  $i$  are normalized using

$$\tilde{\phi}_i = \phi_i \frac{\bar{\phi}_{\text{ref}} \phi_{\text{ref}}^{\text{gap}}}{\bar{\phi}_{\text{mc}} \phi_{\text{ref}}^{\text{row}}}$$

where  $\phi_{\text{ref}}^{\text{gap}}$  is the averaged surfacic flux of the reference calculation and  $\phi_{\text{ref}}^{\text{row}}$  is the averaged volumic flux in a row of cells of the reference calculation. Using this definition, the averaged SPH factor is equal to

$$\bar{\mu} = \frac{\phi_{\text{ref}}^{\text{row}}}{\phi_{\text{ref}}^{\text{gap}}}.$$

<b>ASYM</b>	keyword to specify the use of asymptotic normalization of the SPH factors. The SPH factors in homogenized mixture <i>mixs</i> are set to one in all macro-energy groups.
<i>mixs</i>	index of the homogenized mixture where asymptotic normalization is performed.
<b>ARM</b>	keyword to activate a solution technique other than the collision probability method. Used with the Eurydice solution technique within <b>SYBILT</b> : to activate the current iteration method.
<b>ITER</b>	keyword to specify the main convergence parameters used to control SPH iterations.
<i>maxout</i>	user-defined maximum number of SPH iterations (default value: 200).
<i>epsout</i>	user-defined convergence criterion (default value: $1.0 \times 10^{-4}$ ).
<b>MAXNB</b>	keyword to specify an auxiliary convergence parameter used to control SPH iterations.
<i>maxnb</i>	acceptable number of SPH iterations with an increase in convergence error before aborting (default value: 10).
<b>EQUI</b>	keyword used to select an existing set of SPH factors in <i>SAPNAM</i> or to store a new set of SPH factors in <i>SAPNEW</i> or <i>SAPNAM</i> .
<i>TEXT4</i>	character*4 user-defined keyword of a set of SPH factors. This keyword is related to variable <i>parkey</i> , as defined in Sect. 3.19.1 for a local variable.
<b>LOCNAM</b>	keyword used to define a character*80 name for the set of SPH factors, if this set is created. By default, <i>TEXT80</i> is taken equal to <i>TEXT4</i> .
<i>TEXT80</i>	character*80 user-defined name associated to keyword <i>TEXT4</i> . This name is related to variable <i>parnam</i> , as defined in Sect. 3.19.1 for a local variable.
<b>LEAK</b>	keyword used to introduce leakage in the embedded <b>MACROLIB</b> . This option should only be used for non-regression tests.
<i>b2</i>	the imposed buckling corresponding to the leakage.

### 3.12 The CFC: module

The **CFC:** module is used to generate a Feedback Model database required for a full core calculation in DONJON.<sup>[?]</sup>  The general specifications of this module are:

Table 53: Structure (**CFC:**)

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

where

*CFCNAM*      **character\*12** name of the FBMXSDB data structure containing the Feedback Model reactor database. The reactor database can be updated if *CFCNAM* appears on the RHS.

*CPONAM*      **character\*12** name of read only CPO data structures. There are 28 different CPO data structures required here each containing respectively

1. the reactor reference cross section.
2. cell cross section for the first fuel temperature.
3. cell cross section for the second fuel temperature.
4. cell cross section for the first coolant temperature.
5. cell cross section for the second coolant temperature.
6. cell cross section for the first moderator temperature.
7. cell cross section for the second moderator temperature.
8. cell cross section for the first coolant density.
9. cell cross section for the second coolant density.
10. cell cross section for the first moderator density.
11. cell cross section for the second moderator density.
12. cell cross section for a different concentration of boron.
13. cell cross section for a different moderator purity.
14. cell cross section for a different concentration of xenon.
15. cell cross section for a different concentration of samarium.
16. cell cross section for a different concentration of neptunium.
17. cell cross section for the spectral mixed effect fuel/coolant density.
18. cell cross section for the spectral mixed effect coolant density/temperature.
19. cell cross section for low power history.
20. cell cross section for intermediate power history.
21. cell cross section for high power history.
22. reactor reference moderator cross section.
23. moderator cross section for the first moderator temperature.
24. moderator cross section for the second moderator temperature.
25. moderator cross section for the first moderator density.
26. moderator cross section for the second moderator density.

- 27. moderator cross section for a different concentration of boron.
- 28. moderator cross section for a different moderator purity.

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

### 3.12.1 Data input for module CFC:

Table 54: Structure (**descfc**)

```
[ EDIT iprint ]
[ INFOR TITLE ]
[ DNAME RNAME ]
[ PWR powerref powerup powerint powerdown ]
[ TCOOL tcoolref tcoolup tcooldown ]
[ TMODE tmoderef tmodeup tmodedown ]
[ TFUEL tfuelref tfuelup tfueldown ]
[ RHOC denscool]
[ RHOM densmode]
[ XIR purityref puritydown ]
```

where

EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index used to control the printing of the module.
INFOR	keyword which allows to set the title.
<i>TITLE</i>	<b>character*72</b> title associated to the reactor database generated.
DNAME	keyword to set a specific database name in the data structure.
<i>RNAME</i>	<b>character*12</b> name of the feedback database.
PWR	keyword to specify power used for evolution for power history.
<i>powerref</i>	power value for regular power history ( <i>CPONAM</i> default).
<i>powerup</i>	power value for high power history ( <i>CPONAM</i> 21).
<i>powerint</i>	power value for intermediate power history ( <i>CPONAM</i> 20).
<i>powerdown</i>	power value for low power history ( <i>CPONAM</i> 19).
TCOOL	keyword to specify coolant temperature used for regular evolution and perturbed cases.
<i>tcoolref</i>	normal coolant temperature ( <i>CPONAM</i> default).
<i>tcoolup</i>	high coolant temperature ( <i>CPONAM</i> 4).
<i>tcooldown</i>	low coolant temperature ( <i>CPONAM</i> 5).
TMODE	keyword to specify moderator temperature used for regular evolution and perturbed cases.

<i>tmodef</i>	normal moderator temperature ( <i>CPONAM</i> default).
<i>tmodeup</i>	high moderator temperature ( <i>CPONAM</i> 6 and 23).
<i>tmodedown</i>	low moderator temperature ( <i>CPONAM</i> 7 and 24).
<b>TFUEL</b>	keyword to specify fuel temperature used for regular evolution and perturbed cases.
<i>tfuelref</i>	normal fuel temperature ( <i>CPONAM</i> default).
<i>tfuelup</i>	high fuel temperature ( <i>CPONAM</i> 2).
<i>tfueldown</i>	low fuel temperature ( <i>CPONAM</i> 3).
<b>RHOC</b>	keyword to specify coolant density used for regular evolution.
<i>denscool</i>	normal coolant density ( <i>CPONAM</i> default).
<b>RHOM</b>	keyword to specify moderator density used for regular evolution.
<i>densmode</i>	normal moderator density ( <i>CPONAM</i> default).
<b>XIR</b>	keyword to specify water purity ( $D_2O$ content) used for regular evolution and perturbed cases.
<i>purityref</i>	normal moderator purity (fraction of $D_2O$ in water) ( <i>CPONAM</i> default).
<i>puritydown</i>	perturbed moderator purity (fraction of $D_2O$ in water) ( <i>CPONAM</i> 13 and 28).

Note: Other perturbed values are recovered directly from the concentrations and isotope densities stored in the different *CPONAM*.

### 3.13 The INFO: module

The **INFO:** module is mainly used to compute the number densities for selected isotopes at specific local conditions. The module can also be used to compute the water density  $\rho(T, P)$  according to the assumed temperature  $T$  and purity  $P$ . In that case, the compound water density for a mix of light and heavy water is

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

Temperature tabulations for  $\rho_{H_2O}(T)$  and  $\rho_{D_2O}(T)$  are the same as those of the WIMS-AECL code.

The calling specifications are:

Table 55: Structure (**INFO:**)

<b>INFO: :: (descinfo)</b>
----------------------------

where

**(descinfo)**            structure containing the input data to this module (see Section 3.13.1).

#### 3.13.1 Data input for module **INFO:**

Table 56: Structure (**info**)

<pre>[ EDIT <i>iprint</i> ] [ LIB: { DRAGON   MATXS   MATXS2   WIMSD4   WIMS   WIMSAECL   NDAS   APLIB2   APLIB1 }       FIL: NAMEFIL ] [ TMP: <i>temp</i> { K   C } ] [ PUR: <i>purity</i> { WGT%   ATM% } ] [ CALC DENS WATER &gt;&gt;<i>dens</i>&lt;&lt; ] [ ENR: <i>enrichment</i> { WGT%   ATM% } ] [[ ISO: <i>nbiso</i> (ISONAM(<i>i</i>), <i>i</i>=1,<i>nbiso</i>)    { GET MASS (&gt;&gt;<i>mass</i>(<i>i</i>)&lt;&lt;, <i>i</i>=1,<i>nbiso</i>)   CALC WGT% {      D20 &gt;&gt;<i>nh1</i>&lt;&lt; &gt;&gt;<i>hd2</i>&lt;&lt; &gt;&gt;<i>no16</i>&lt;&lt;        UO2 &gt;&gt;<i>nu5</i>&lt;&lt; &gt;&gt;<i>hu8</i>&lt;&lt; &gt;&gt;<i>no16</i>&lt;&lt;        THO2 &gt;&gt;<i>nth2</i>&lt;&lt; &gt;&gt;<i>nu3</i>&lt;&lt; &gt;&gt;<i>no16</i>&lt;&lt; } } ]]</pre>
--

where

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

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

**LIB:**            keyword to specify the type of library from which the isotopic mass ratio is to be read.

DRAGON	keyword to specify that the isotopic depletion chain or the microscopic cross sections are in the DRAGLIB format.
MATXS	keyword to specify that the microscopic cross sections are in the MATXS format of NJOY-II and NJOY-89 (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 and the microscopic cross sections are in the WIMSD4 format.
WIMS	keyword to specify that the isotopic depletion chain and the microscopic cross sections are in the WIMS-AECL format.
WIMSAECL	keyword to specify that the isotopic depletion chain and the microscopic cross sections are in the WIMS-AECL format.
NDAS	keyword to specify that the isotopic depletion chain and the microscopic cross sections are in the NDAS format, as used in recent versions of WIMS-AECL.
APLIB1	keyword to specify that the microscopic cross sections are in the APOLLO-1 format.
APLIB2	keyword to specify that the microscopic cross sections are in the APOLLO-2 format.
FIL:	keyword to specify the name of the file where is stored the mass ratio data.
NAMEFIL	<b>character*8</b> name of the library where the mass ratio are stored.
TMP:	keyword to specify the isotopic temperature.
<i>temp</i>	temperature given in Kelvin (K) or Celsius (C).
PUR:	keyword to specify the water purity, that is fraction of heavy water in a mix of heavy and light water.
<i>purity</i>	water purity in weight percent (WGT%) or atomic percent (ATM%).
ENR:	keyword to specify the fuel enrichment.
<i>enrichment</i>	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.
<i>nbiso</i>	number of isotopic names used for a calculation (limited to $nbiso \leq 3$ ).
ISONAM	<b>character*12</b> 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 <i>mass</i> .
CALC	keyword to ask the module to compute some parametric values. It returns one value in the output parameter <i>dens</i> .
DENS WATER	set of keywords to recover the water density as a function of its temperature and purity. This option requires the setting of temperature and purity, and it does not affect any given list of isotope names.
WGT% D2O	keywords to recover 3 number densities for a compound mixture of heavy and light water. The isotope list is assumed to contain $^1\text{H}$ , $^2\text{D}$ and $^{16}\text{O}$ . Temperature and purity are supposed to be available. It returns concentration of these isotopes in the output parameters <i>nh1</i> , <i>nd2</i> and <i>no16</i> .

WGT% UO2	keywords to recover 3 number densities for a compound mixture of Uranium oxide. The isotope list is assumed to contain $^{235}\text{U}$ , $^{238}\text{U}$ and $^{16}\text{O}$ . The $^{235}\text{U}$ 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 <i>nu5</i> , <i>nu8</i> and <i>no16</i> .
WGT% THO2	keywords to recover 3 number densities for a compound mixture of Thorium/Uranium oxide. The isotope list is assumed to contain $^{232}\text{Th}$ , $^{233}\text{U}$ and $^{16}\text{O}$ . The $^{233}\text{U}$ 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 <i>nth2</i> , <i>nu3</i> and <i>no16</i> .

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<<*, are recovered as CLE-2000 variables in (**descinfo**). The number of these parameters must be equal to the number of isotopes names given, plus the water density when a command **CALC DENS WATER** is issued.

### 3.14 The COMPO: module

This component of the lattice code is dedicated to the constitution of the reactor database intended to store *all* the nuclear data, produced in the lattice code, that is useful in reactor calculations including fuel management and space-time kinetics. Multigroup lattice calculations are too expensive to be executed dynamically from the driver of the global reactor calculation. A more feasible approach is to create a reactor database where a finite number of lattice calculation results are tabulated against selected *global* and/or *local parameters* chosen so as to represent expected operating conditions of the reactor.

The COMPO: module is used to create and construct a MULTICOMPO object. This object is generally *persistent* and used to collect information gathered from many DRAGON *elementary calculations* performed under various conditions.

For each elementary calculation, the results are recovered from the output of the EDI: module and stored in a list of *homogenized mixture* directories. The EDI: module is responsible for performing condensation in energy, homogenization in space of the microscopic cross sections and constitution of *macroscopic sets* for collecting together many isotopes. All the elementary calculations gathered in a single MULTICOMPO object are characterized by the same number of *homogenized mixtures* and by a specific output energy-group structure.

Figure 20: Organization of a `multicompo` object.

Each elementary calculation is characterized by a tuple of *global* and/or *local parameters*. Global parameters are characteristics of the complete lattice, while local parameters are characteristics of each homogenized mixture. These parameters are of different types, depending on the nature of the study under consideration: type of assembly, power, temperature in a mixture, concentration of an isotope, time, burnup or exposure rate in a depletion calculation, etc. Each step of a depletion calculation represents an elementary calculation. The MULTICOMPO object is often presented as a *multi-parameter reactor database*.

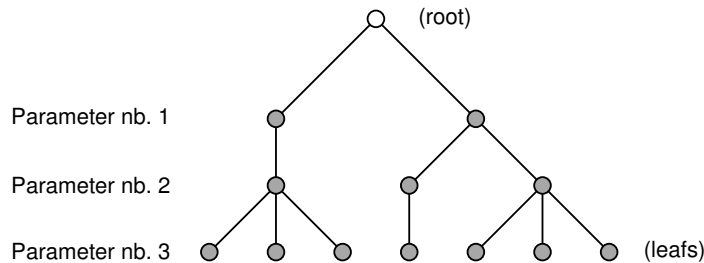


Figure 21: Parameter tree in a MULTICOMPO object

The MULTICOMPO object is organized as shown in Figure 20. The root of the object contains table-of-content information for global and local parameters and two lists of directories. Each component of the first list ('MIXTURES') contains the directory 'TREE' (the parameter tree) and the list ('CALCULATIONS') made of MICROLIB objects. Each component of the second list ('GEOMETRIES') contains the homogenized



geometry of an elementary calculation.

The localization of an elementary calculation is done using a tuple of global and/or local parameters. The elementary calculation indices are stored in a tree with the number of levels equal to the number of global and local parameters. An example of a tree with three parameters is shown in Figure 21. Each node of this tree is associated with the index of the corresponding parameter and with the reference to the daughter nodes if they exist. The number of leaf nodes is equal to the number of elementary calculations stored in the MULTICOMPO object. The index of each elementary calculation is therefore an attribute of each leaf.

In each homogenized mixture component, the **COMPO:** module recovers cross sections for a number of *particularized isotopes* and of a single *macroscopic set*, a collection of the remaining isotopic cross sections weighted by isotopic number densities. Other information is also recovered: multigroup neutron fluxes, isotopic number densities, fission spectrum, delayed neutron data, etc.

A different specification of the **COMPO:** function call is used for creation and construction of the MULTICOMPO object.

- The first specification is used to initialize the MULTICOMPO data structure and to set the choice of global and local parameters.
- A modification call to the **COMPO:** function is performed after each elementary calculation in order to recover output information processed by **EDI:** (condensed and homogenized cross sections) and **EVO:** (burnup dependant values). Global and local parameters can optionally be recovered from MICROLIB objects.
- Another modification call to the **COMPO:** function is used to concatenate a *read-only* MULTICOMPO object into a *master* MULTICOMPO object.

The calling specifications are:

Table 57: Structure (**COMPO:**)

```
{ CPONAM := COMPO: [ CPONAM ] :: (compo_data1)
| CPONAM := COMPO: CPONAM EDINAM [ EDINA2 ] [ BRNNAM ] [ HMIC1 [ HMIC2 ] ]
  :: (compo_data2)
| CPONAM := COMPO: CPONAM CPORHS :: (compo_data3)
| COMPO: CPONAM :: (compo_data4) }
```

where

<i>CPONAM</i>	<b>character*12</b> name of the LCM object containing the <i>master</i> MULTICOMPO data structure.
<i>EDINAM</i>	<b>character*12</b> name of the LCM object (type <b>L_EDIT</b> ) containing the EDITION data structure corresponding to an elementary calculation. This EDITION data structure is containing homogenized and condensed cross-section information. The EDITION data produced by the last call to the <b>EDI:</b> module is used.
<i>EDINA2</i>	<b>character*12</b> name of an optional LCM object (type <b>L_EDIT</b> ) containing the EDITION data structure corresponding to an elementary calculation. This EDITION data structure is containing <i>group form factor</i> information. The EDITION data produced by the last call to the <b>EDI:</b> module is used.
<i>BRNNAM</i>	<b>character*12</b> name of the LCM object (type <b>L_BURNUP</b> ) containing the BURNUP data structure.

<i>HMIC1</i>	<b>character*12</b> name of a MICROLIB (type L-LIBRARY) containing global parameter information.
<i>HMIC2</i>	<b>character*12</b> name of a MICROLIB (type L-LIBRARY) containing global parameter information.
<i>CPORHS</i>	<b>character*12</b> name of the <i>read-only</i> MULTICOMPO data structure. This data structure is concatenated to <i>CPONAM</i> using the <i>compo_data3</i> data structure, as presented in Section 3.14.3. <i>CPORHS</i> must be defined with the same number of energy groups and the same number of homogeneous regions as <i>CPONAM</i> . Moreover, all the global and local parameters of <i>CPORHS</i> must be defined in <i>CPONAM</i> . <i>CPONAM</i> may be defined with <i>global</i> parameters not defined in <i>CPORHS</i> .
<i>compo_data1</i>	input data structure containing initialization information (see Section 3.14.1).
<i>compo_data2</i>	input data structure containing information related to the recovery of an elementary calculation (see Section 3.14.2).
<i>compo_data3</i>	input data structure containing information related to the catenation of a <i>read-only</i> MULTICOMPO (see Section 3.14.3).
<i>compo_data4</i>	input data structure containing information related to the display of a <i>read-only</i> MULTICOMPO (see Section 3.14.4).

### 3.14.1 Initialization data input for module COMPO:

Table 58: Structure (**compo\_data1**)

```
[ EDIT iprint ]
[[ [ STEP UP NAMDIR ]
  [ MAXCAL maxcal ]
  [ COMM [[ HCOM ]] ENDC ]
  [[ PARA PARKEY
    { TEMP HMIC imix | CONC HISO1 HMIC imix | IRRA | FLUB |
      POWR | MASL | FLUX | TIME | VALU { REAL | CHAR | INTE } }
  ]]
  [[ LOCA PARKEY
    { TEMP | CONC HISO2 | IRRA | FLUB | FLUG | POWR | MASL | FLUX }
  ]]
[ ISOT nisp (HISOP(i),i=1,nisp) ]
[ GFF ]
INIT ]]
;
```

where

<b>EDIT</b>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module COMPO: . =0 for no print; =1 for minimum printing (default value).
<b>STEP</b>	keyword used to create the database from a sub-directory named <i>NAMDIR</i> . This capability make possible the creation of a single object with many independent MULTICOMPO structures in it. By default, the database is created on directory ' <b>default</b> '.

UP	keyword used to move up towards a sub-directory of <i>CPONAM</i> .
NAMDIR	create the MULTICOMPO structure in the sub-directory named <i>NAMDIR</i> .
MAXCAL	keyword used to set <i>maxcal</i> .
<i>maxcal</i>	maximum number of elementary calculations to be stored in the MULTICOMPO. <i>maxcal</i> = 10 by default. This maximum size is automatically increased when the number of elementary calculations exceeds the current value of <i>maxcal</i> .
COMM	keyword used to input a general comment for the MULTICOMPO.
HCOM	<b>character*80</b> user-defined comment.
ENDC	end-of-HCOM keyword.
PARA	keyword used to define a single global parameter.
LOCA	keyword used to define a single local parameter.
PARKEY	<b>character*12</b> user-defined keyword associated to a global or local parameter.
HMIC	<b>character*12</b> name of the MICROLIB (type <b>L-LIBRARY</b> ) associated to a global parameter. The corresponding MICROLIB will be required on RHS of the <b>COMPO</b> : call described in Sect. 3.14.2.
<i>imix</i>	index of the mixture associated to a global parameter. This mixture is located in MICROLIB named <i>HMIC</i> .
HISO1	<b>character*8</b> alias name of the isotope associated to a global parameter. This isotope is located in MICROLIB data structure named <i>HMIC</i> .
HISO2	<b>character*8</b> alias name of the isotope associated to a local parameter. This isotope is located in the MICROLIB directory of the EDITION data structure named <i>EDINAM</i> .
TEMP	keyword used to define a temperature (in Kelvin) as global or local parameter.
CONC	keyword used to define a number density as global or local parameter.
IRRA	keyword used to define a burnup (in MWday/Tonne) as global or local parameter.
FLUB	keyword used to define a <i>fuel-only</i> exposure rate (in n/kb) as global or local parameter. The exposure rate is recovered from the <i>BRNNAM</i> LCM object.
FLUG	keyword used to define an exposure rate in global homogenized mixtures (in n/kb) as local parameter. The exposure rate is recovered from the <i>BRNNAM</i> LCM object.
POWR	keyword used to define the power as global or local parameter.
MASL	keyword used to define the mass density of heavy isotopes as global or local parameter.
FLUX	keyword used to define the volume-averaged, energy-integrated flux as global or local parameter.
TIME	keyword used to define the time (in seconds) as global parameter.
VALU	keyword used to define a user-defined quantity as global parameter. This keyword must be followed by the type of parameter.
REAL	keyword used to indicate that the user-defined global parameter is a floating point value.
CHAR	keyword used to indicate that the user-defined global parameter is a <b>character*12</b> value.

INTE	keyword used to indicate that the user-defined global parameter is an integer value.
ISOT	keyword used to select the set of particularized isotopes. By default, all the isotopes available in the EDITION data structure <i>EDINAM</i> are selected.
<i>nisp</i>	number of user-defined particularized isotopes.
<i>HISOP</i>	<b>character*8</b> names of the user-defined particularized isotopes. These names must be present in the EDITION data structure <i>EDINAM</i> .
GFF	keyword used to enable the recovery of group form factor information from EDITION data structure <i>EDINA2</i> .
INIT	keyword used to create the empty structure in the MULTICOMPO.

## 3.14.2 Modification data input for module COMPO:

Table 59: Structure (**compo\_data2**)

```
[ EDIT iprint ]
[ ALLX ]
[ STEP UP { NAMDIR | * } ]
[ ORIG orig ]
[[ PARKEY value ]]
[ MACRO ] [ SET xtr { S | DAY | YEAR } ]
;
```

where

EDIT	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module COMPO: . =0 for no print; =1 for minimum printing (default value).
ALLX	keyword used to register the region number of each isotope before merging. This option is useful if the same keyword has been specified in EDI: before. This allows to perform subsequent depletion calculations, in taking into account different fuel regions in the diffusion calculation.
STEP	keyword used to access the database from a sub-directory named <i>NAMDIR</i> instead of accessing it from the root of <i>CPONAM</i> .
UP	keyword used to move up towards a sub-directory of <i>CPONAM</i> .
<i>NAMDIR</i>	access the MULTICOMPO structure in the sub-directory named <i>NAMDIR</i> .
*	use a sub-directory name identical to the directory in <i>EDINAM</i> where the edition data is coming from.
<i>PARKEY</i>	<b>character*12</b> keyword associated to a user-defined global parameter.
<i>value</i>	floating-point, integer or <b>character*12</b> value of a user-defined global parameter.
ORIG	keyword used to define the father node in the parameter tree. By default, the index of the previous elementary calculation is used.
<i>orig</i>	index of the elementary calculation associated to the father node in the parameter tree.
MACRO	keyword used to recover cross-section information from the macrolib directory in <i>EDINAM</i> . By default, the cross-section information is recovered from the microlib in <i>EDINAM</i> .
SET	keyword used to recover the flux normalization factor already stored on <i>BRNNAM</i> from a sub-directory corresponding to a specific time.
<i>xtr</i>	time associated with the current flux calculation. The name of the sub-directory where this information is stored will be given by 'DEPL-DAT'//CNN where CNN is a <b>character*4</b> variable defined by WRITE(CNN,'(I4.4)') INN where INN is an index associated with the time <i>xtr</i> .
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.

### 3.14.3 Modification (catenate) data input for module COMPO:

Table 60: Structure (**compo\_data3**)

```
[ EDIT iprint ]
[ STEP UP NAMDIR ]
[ ORIG orig ]
[[ PARKEY value ]]
[ WARNING-ONLY ]
;
```

where

EDIT	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module COMPO: . =0 for no print; =1 for minimum printing (default value).
<i>PARKEY</i>	<b>character*12</b> keyword associated to a global parameter that is specific to <i>CPONAM</i> (not defined in <i>CPORHS</i> ).
<i>value</i>	floating-point, integer or <b>character*12</b> value of a global parameter that is specific to <i>CPONAM</i> .
ORIG	keyword used to define the father node in the parameter tree. By default, the index of the previous elementary calculation is used.
<i>orig</i>	index of the elementary calculation associated to the father node in the parameter tree.
WARNING-ONLY	This option is useful if an elementary calculation in <i>CPORHS</i> is already present in <i>CPONAM</i> . If this keyword is set, a warning is send and the <i>CPONAM</i> values are kept, otherwise the run is aborted (default).

### 3.14.4 Display data input for module COMPO:

Table 61: Structure (**compo\_data4**)

```
[ EDIT iprint ]
[ STEP UP NAMDIR ]
[ EDIT iprint ]
DB-STRUC
;
```

where

**EDIT** keyword used to set *iprint*.

*iprint* index used to control the printing in module **COMP0**:.  $<2$  for MUPLET display only (default value) and parameters values are presented at the end,  $\geq 2$  for the parameter value display for each calculation.

**DB-STRUC** **character\*12** keyword used to display the content of the *CPONAM* object for the *NAMDIR* directory.

### 3.15 The TLM: module

The TLM: module has been designed to generate a Matlab **m-file** (in an ASCII format) that contains the instructions for plotting the tracking lines generated by the NXT: module (EDIT -1000 option).<sup>[?]</sup> The TLM: module is activated using the following list of commands:

Table 62: Structure (TLM:)

$MFILE := TLM: MFILE TRKNAM TRKFIL :: (desctlm)$
--

where

<i>MFILE</i>	<b>character*12</b> name of the ASCII Matlab <b>m-file</b> data structure that will contain the instructions for plotting the tracking lines.
<i>TRKNAM</i>	<b>character*12</b> name of the TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information.
<i>TRKFIL</i>	<b>character*12</b> name of the sequential binary tracking file used to store the tracks lengths. <sup>[?]</sup>
<b>(desctlm)</b>	structure describing the type of graphics generated (see Section 3.15.1).

#### 3.15.1 Data input for module TLM:

Table 63: Structure (desctlm)

<pre>[ EDIT <i>iprint</i> ] [ NTPO <i>nplots</i> ] ( {   POINTS [ NoPause ]     DIRECTIONS [ NoPause ] DIR <i>idir</i> [ PLAN <i>iplan</i> { U <i>iuv</i>   V <i>iuv</i> } ]     PLANP [ NoPause ] DIR <i>idir</i> DIST <i>dist</i> [ PLAN <i>iplan</i> ]     PLANA [ NoPause ] A <i>a</i> B <i>b</i> [ C <i>c</i> ] D <i>d</i> } , <i>iplot</i>=1, <i>nplots</i> )</pre>
---

where

EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index used to control the printing in this module. It must be set to 0 if no printing on the output file is required.
NTPO	keyword to specify the number of figures to draw.
<i>nplots</i>	integer value for the number of figures to draw.



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.
DIR	keyword to specify line direction to draw.
<i>idir</i>	integer value to identify the track direction to draw. In the case where <i>idir</i> =0, all the directions will be drawn. A value of <i>idir</i> =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.
<i>iplan</i>	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 the 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 the all the lines in the $U$ axis associated with a position on the $V$ axis will be drawn. Used only for 3-D geometries.
<i>iuu</i>	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 to the line direction and the origin.
<i>dist</i>	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 $d$ .

### 3.16 The M2T: module

This component of the lattice code is dedicated to the generation of an ASCII file with the Apotrim specification using MACROLIB data. Such a file is useful to transfer multigroup and macroscopic cross-section data toward a Moret calculation.

The calling specifications are:

Table 64: Structure (**M2T:**)

```
APTRIM := M2T: [ APTRIM ] MLIB :: (M2T_data)
```

where

<i>APTRIM</i>	<b>character*12</b> name of an ASCII file with the Apotrim specification. If <i>APTRIM</i> appears on the RHS, new information is appended to the existing Apotrim file.
<i>MLIB</i>	<b>character*12</b> name of a MACROLIB (type L_MACROLIB) object.
<i>M2T_data</i>	input data structure containing specific data (see Section 3.16.1).

#### 3.16.1 Data input for module M2T:

Table 65: Structure (**M2T\_data**)

```
[ EDIT iprint ]
[ PN nl ] [ TRAN ] [ NOMA ]
[[ MIX hmix [ FROM imixold ] [ BURN bup ] [ TEMP tval ] ENDMIX ]]
;
```

where

<b>EDIT</b>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module M2T:.. =0 for no print; =1 for minimum printing (default value).
<b>PN</b>	keyword used to set the Legendre order of the scattering transfers written on the Apotrim file.
<i>nl</i>	Legendre order. By default, <i>nl</i> = 0 corresponding to an isotropic collision in LAB.
<b>TRAN</b>	keyword used to set a transport correction on cross sections written on the Apotrim file.
<b>NOMA</b>	keyword used to avoid writing the energy mesh on the Apotrim file. This option is useful to catenate additional mixture information on an existing Apotrim file. By default, the energy mesh is written on the Apotrim file.
<b>MIX</b>	keyword used to set <i>hmix</i> .

<i>hmix</i>	<b>character*20</b> name of the mixture to be written on the Apotrim file.
<b>BURN</b>	keyword used to set the burnup of a mixture.
<i>bup</i>	burnup of a mixture. By default, <i>bup</i> = 0.0.
<b>TEMP</b>	keyword used to set the temperature of a mixture.
<i>tval</i>	temperature of a mixture in Celsius. By default, <i>tval</i> = 0.0 °C.
<b>FROM</b>	keyword used to set the index of the mixture in the MACROLIB object.
<i>imixold</i>	index of the mixture that is recovered in the MACROLIB object. By default, <i>imixold</i> = 1.
<b>ENDMIX</b>	end of specification keyword for the material mixture.

Here is an example of the creation of an Apotrim file named **APOTR** with a Hansen-Roach energy mesh created from a XMAS 172-group flux calculation. The Apotrim file is created from three LCM objects **FLUX**, **LIBRARY2** and **TRACK** containing the flux, the XMAS-formatted microlib and the tracking.

```
LINKED_LIST TRACK LIBRARY2 FLUX MAC2 EDIT ;
SEQ_ASCII APOTR ;
...
EDIT := EDI: LIBRARY2 TRACK FLUX :: EDIT 3
*      Hansen-Roach energy mesh follows
      COND 12 17 21 27 33 42 50 60 66 76 84 95 123 140 155 172
      MERGE MIX 1 1 1 1 1 1 2 3 3
      SAVE ON 'EDITCDAT 1' ;
MAC2 := EDIT :: STEP UP 'EDITCDAT 1' STEP UP 'MACROLIB' ;
APOTR := M2T: MAC2 :: EDIT 3 TRAN MIX FUEL FROM 1 ENDMIX
      MIX CLAD FROM 2 ENDMIX
      MIX COOLANT FROM 3 ENDMIX ;
```

### 3.17 The CHAB: module

This component of the lattice code is dedicated to the modification of cross section information in a MICROLIB.

The calling specifications are:

Table 66: Structure (**CHAB:**)

```
{ MICRO1 | DRAGLIB1 } := CHAB: { MICRO1 | MICRO2 | DRAGLIB2 } :: (CHAB_data)
```

where

<i>MICRO1</i>	<b>character*12</b> name of a MICROLIB (type L_LIBRARY) object that is created or modified by CHAB:.
<i>DRAGLIB1</i>	<b>character*12</b> name of a DRAGLIB (type L_DRAGLIB) object that is created by CHAB:.
<i>MICRO2</i>	<b>character*12</b> name of a MICROLIB (type L_LIBRARY) object open in read-only mode.
<i>DRAGLIB2</i>	<b>character*12</b> name of a DRAGLIB (type L_DRAGLIB) object open in read-only mode.
<i>CHAB_data</i>	input data structure containing specific data (see Section 3.17.1).

#### 3.17.1 Data input for module CHAB:

Table 67: Structure (**CHAB\_data**)

```
[ EDIT iprint ]  
[[ MODI TYPSEC igm TO igp { VALE [[ val ]] | CONS value | PLUS value | MULT value } HISOT ]] ;
```

where

<b>EDIT</b>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module CHAB:. =0 for no print; =1 for minimum printing (default value).
<b>MODI</b>	keyword used to define a modification of a nuclear reaction belonging to a given isotope.
<i>TYPSEC</i>	<b>character*8</b> name of an existing nuclear reaction chosen among the following values: ' <b>NTOTO</b> ' Total cross section. ' <b>NG</b> ' Radiative capture cross section. The total (' <b>NTOTO</b> ') cross section is modified accordingly. ' <b>NA</b> ' ( $n, \alpha$ ) cross section. The total (' <b>NTOTO</b> ') cross section is modified accordingly. ' <b>NP</b> ' ( $n, p$ ) cross section. The total (' <b>NTOTO</b> ') cross section is modified accordingly. ' <b>ND</b> ' ( $n, d$ ) cross section. The total (' <b>NTOTO</b> ') cross section is modified accordingly. ' <b>NT</b> ' ( $n, t$ ) cross section. The total (' <b>NTOTO</b> ') cross section is modified accordingly.

	<p>'CAPT' Capture cross sections. Each present reaction of capture (<b>NG</b>, <b>NA</b>, <b>NP</b>, <b>ND</b>, <b>NT</b>) are taken into account. The total ('NTOTO') cross section is modified accordingly. Only the keyword <b>MULT</b>, indicating a multiplication of the all cross sections, is available.</p> <p>'NELAS' Elastic scattering cross section. The scattering ('SIGS00' and 'SCAT00') and total ('NTOTO') cross sections are modified accordingly.</p> <p>'NINEL' Inelastic scattering cross section. The scattering ('SIGS00' and 'SCAT00') and total ('NTOTO') cross sections are modified accordingly.</p> <p>'N2N' (<math>n,2n</math>) cross section. The scattering ('SIGS00' and 'SCAT00') and total ('NTOTO') cross sections are modified accordingly.</p> <p>'N3N' (<math>n,3n</math>) cross section. The scattering ('SIGS00' and 'SCAT00') and total ('NTOTO') cross sections are modified accordingly.</p> <p>'N4N' (<math>n,4n</math>) cross section. The scattering ('SIGS00' and 'SCAT00') and total ('NTOTO') cross sections are modified accordingly.</p> <p>'SIGS00', 'SIGS01', <b>etc.</b> Scattering cross section. The total ('NTOTO') cross section is modified accordingly.</p> <p>'SCAT00', 'SCAT01', <b>etc.</b> Differential scattering cross section. The total ('NTOTO') cross section is modified accordingly.</p> <p>'NUSIGF' <math>\nu</math> times the fission cross section. The fission ('NFTOT') and total ('NTOTO') cross sections are modified accordingly.</p> <p>'NFTOT' Fission cross section. The <math>\nu</math> times fission ('NUSIGF') and total ('NTOTO') cross sections are modified accordingly.</p> <p>'NU' Number of neutrons emitted per fission. The <math>\nu</math> times fission ('NUSIGF') cross section is modified accordingly.</p> <p>'CHI' Fission spectrum. The resulting spectrum is normalized.</p>
<i>igm</i>	lower energy group index of the energy domain where the modification is taking place.
<i>igp</i>	upper energy group index of the energy domain where the modification is taking place.
<b>VALE</b>	keyword indicating a replacement of all values in the above energy domain by different values.
<i>val</i>	group-dependent real variable used as replacement value. We expect $igp-igm+1$ components.
<b>CONS</b>	keyword indicating a replacement of all values in the above energy domain by <i>value</i> .
<b>PLUS</b>	keyword indicating that <i>value</i> is added to all values in the above energy domain.
<b>MULT</b>	keyword indicating a multiplication of all values in the above energy domain by <i>value</i> .
<i>value</i>	real variable used to modify the nuclear reaction.
<b>HISOT</b>	<b>character*8</b> or <b>character*12</b> name of the isotope to modify. If <b>HISOT</b> is a <b>character*8</b> value, all <b>character*12</b> isotope names prefixed by <b>HISOT</b> are modified.

### 3.18 The CP0: module

The CP0: module is used to generate the reactor cross-section database in Version3 format to be used in a full core calculation using DONJON. This type of database is only parametrized in burnup (or irradiation). The calling specifications are:

Table 68: Structure (CP0:)

$CPONAM := CP0: [ CPONAM ] EDINAM [ BRNNAM ] :: (descppo)$
--

where

<i>CPONAM</i>	<b>character*12</b> name of the CPO data structure containing the reactor database. Additional contributions can be included in the reactor cross-section database if <i>CPONAM</i> appears on the RHS.
<i>EDINAM</i>	<b>character*12</b> name of the read-only EDITION data structure.
<i>BRNNAM</i>	<b>character*12</b> 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.
( <b>descppo</b> )	structure containing the input data to this module (see Section 3.18.1).

#### 3.18.1 Data input for module CP0:

Table 69: Structure (descppo)

<pre>[ EDIT <i>i</i>print ] [ B2 ] [ NOTR ] { STEP NOMDIR   BURNUP PREFIX } [ [ EXTRACT { ALL   NEWNAME (OLDNAME(<i>i</i>), <i>i</i>=1,niext) } ] ] [ NAME NDIR ] [ { GLOB   LOCA } ]</pre>
---

where

<b>EDIT</b>	keyword used to modify the print level <i>i</i> print.
<i>i</i> print	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.
<b>B2</b>	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.
<b>NOTR</b>	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 <i>EDINAM</i> via the SAVE option in the EDI: module is to be transferred to <i>CPONAM</i> .
NOMDIR	<b>character*12</b> name of the specific cross section directory to be treated.
BURNUP	keyword to specify that a chain of cross section directory stored in <i>EDINAM</i> via the SAVE option in the EDI: module will be transferred to <i>CPONAM</i> .
PREFIX	<b>character*8</b> 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  <pre>WRITE(NAMDIR, '(A8,I4)') PREFIX,nb</pre> where <b>nb</b> is an integer greater than 0 indicating the depletion step index.
EXTRACT	keyword to specify that the contribution of some isotopes to the macroscopic cross sections associated with each homogenized mixture should be extracted before being stored on the reactor database. The microscopic cross sections and concentrations associated with these isotopes should also be generated and stored on the reactor database.
ALL	keyword to specify that all the isotopes processed using the MICR option of the EDI: module should be extracted from the macroscopic cross sections associated with each homogenized mixture.
NEWNAME	<b>character*12</b> name under which a given set of extracted isotope will be stored on the reactor database.
OLDNAME	array of <b>character*8</b> 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 <i>NDIR</i> ='COMPO~~~'.
NDIR	<b>character*8</b> prefix for the name of the sub-directory. The complete name is constructed by the concatenation of <i>NDIR</i> with a four digit integer value.
GLOB	keyword to specify that global parameters are used to index the database (default option). A global parameter is defined over the complete calculation domain.
LOCA	keyword to specify that local parameters are used to index the database. A local parameter is defined over each homogenization mixture.





- The first specification is used to initialize the SAPHYB data structure as a function of the MICROLIB used in the reference calculation. Optionnally, the homogenized geometry is also provided. The initialization call is also used to set the choice of global parameters, local variables, particularized isotopes, macroscopic sets and selected reactions.
- A modification call to the **SAP:** function is performed after each elementary calculation in order to recover output information processed by EDI: (condensed and homogenized cross sections) and EVO: (burnup dependant values). Global parameters and local variables can optionnally be recovered from MICROLIB objects. The EDI: calculation is generally performed with option MICR ALL.

The calling specifications are:

Table 70: Structure (**SAP:**)

```
{ SAPNAM := SAP: [ SAPNAM ] [ HMIC ] :: (saphyb_data1)
  | SAPNAM := SAP: SAPNAM EDINAM [ BRNNAM ] [ HMIC1 [ HMIC2 ] ] [ FLUNAM ]
  :: (saphyb_data2)
  | SAPNAM := SAP: SAPNAM SAPRHS :: (saphyb_data3) }
```

where

<i>SAPNAM</i>	<b>character*12</b> name of the LCM object containing the <i>master</i> SAPHYB data structure.
<i>HMIC</i>	<b>character*12</b> name of the reference MICROLIB (type L_LIBRARY) containing the microscopic cross sections.
<i>EDINAM</i>	<b>character*12</b> name of the LCM object (type L_EDIT) containing the EDITION data structure corresponding to an elementary calculation. The EDITION data produced by the last call to the EDI: module is used.
<i>BRNNAM</i>	<b>character*12</b> name of the LCM object (type L_BURNUP) containing the BURNUP data structure. This object is compulsory if one of the following parameters is used: IRRR, FLUB and/or TIME.
<i>HMIC1</i>	<b>character*12</b> name of a MICROLIB (type L_LIBRARY) containing global parameter information.
<i>HMIC2</i>	<b>character*12</b> name of a MICROLIB (type L_LIBRARY) containing global parameter information.
<i>FLUNAM</i>	<b>character*12</b> name of the reference FLUX (type L_FLUX). By default, the reference flux is not recovered and not written on the SAPHYB.
<i>SAPRHS</i>	<b>character*12</b> name of the <i>read-only</i> SAPHYB data structure. This data structure is concatenated to <i>SAPNAM</i> using the <i>saphyb_data3</i> data structure, as presented in Section 3.19.3. <i>SAPRHS</i> must be defined with the same number of energy groups and the same number of homogeneous regions as <i>SAPNAM</i> . Moreover, all the global and local parameters of <i>SAPRHS</i> must be defined in <i>SAPNAM</i> . <i>SAPNAM</i> may be defined with <i>global</i> parameters not defined in <i>SAPRHS</i> .
<i>saphyb_data1</i>	input data structure containing initialization information (see Section 3.19.1).
<i>saphyb_data2</i>	input data structure containing information related to the recovery of an elementary calculation (see Section 3.19.2).
<i>saphyb_data3</i>	input data structure containing information related to the catenation of a <i>read-only</i> SAPHYB (see Section 3.19.3).

## 3.19.1 Initialization data input for module SAP:

Table 71: Structure (**saphyb\_data1**)

```

[ EDIT iprint ]
[ NOML nomlib ]
[ COMM [[ comment ]] ENDC ]
[[ PARA parnam parkey
  { TEMP micnam imix | CONC isonam1 micnam imix | IRRR | FLUB |
    PUIS | MASL | FLUX | TIME | VALE { FLOT | CHAI | ENTI } }
]]
[[ LOCA parnam parkey
  { TEMP | CONC isonam2 | IRRR | FLUB | FLUG | PUIS | MASL | FLUX | EQUI }
]]
[ ISOT { TOUT | MILI imil | [ FISS ] [ PF ] [ (HNAISO(i), i=1,Niso) ] } ]
[[ MACR HNAMAC { TOUT | REST } ]]
[ REAC (HNAREA(i), i=1,Nreac) ]
[ NAME (HNAMIX(i), i=1,Nm) ]
;
```

where

EDIT	key word used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module SAP: . =0 for no print; =1 for minimum printing (default value).
NOML	key word used to input a user-defined name for the SAPHYB. This information is mandatory if the Saphyb is to be read by the Lisaph module of Cronos.
<i>nomlib</i>	<b>character*80</b> user-defined name.
COMM	key word used to input a general comment for the SAPHYB.
<i>comment</i>	<b>character*80</b> user-defined comment.
ENDC	end-of-comment key word.
PARA	keyword used to define a single global parameter.
LOCA	keyword used to define a single local variable (a local variable may be a single value or an array of values).
<i>parnam</i>	<b>character*80</b> user-defined name of a global parameter or local variable.
<i>parkey</i>	<b>character*4</b> user-defined keyword associated to a global parameter or local variable.
<i>micnam</i>	<b>character*12</b> name of the MICROLIB (type <b>L.LIBRARY</b> ) associated to a global parameter. The corresponding MICROLIB will be required on RHS of the SAP: call described in Sect. 3.19.2.
<i>imix</i>	index of the mixture associated to a global parameter. This mixture is located in MICROLIB named <i>micnam</i> .
<i>isonam1</i>	<b>character*8</b> alias name of the isotope associated to a global parameter. This isotope is located in MICROLIB data structure named <i>micnam</i> .

<i>isonam2</i>	<b>character*8</b> alias name of the isotope associated to a local variable. This isotope is located in the MICROLIB directory of the EDITION data structure named <i>EDINAM</i> .
TEMP	keyword used to define a temperature (in °C) as global parameter or local variable.
CONC	keyword used to define a number density as global parameter or local variable.
IRRA	keyword used to define a burnup (in MWday/Tonne) as global parameter or local variable.
FLUB	keyword used to define a <i>fuel-only</i> exposure rate (in n/kb) as global parameter or local variable. The exposure rate is recovered from the <i>BRNNAM</i> LCM object.
FLUG	keyword used to define an exposure rate in global homogenized mixtures (in n/kb) as local variable. The exposure rate is recovered from the <i>BRNNAM</i> LCM object.
PUIS	keyword used to define the power as global parameter or local variable.
MASL	keyword used to define the mass density of heavy isotopes as global parameter or local variable.
FLUX	keyword used to define the volume-averaged, energy-integrated flux as global parameter or local variable.
TIME	keyword used to define the time (in seconds) as global parameter.
EQUI	keyword used to define the SPH equivalence factors as local variable. A set of SPH factors can be defined as local variables. Note that the cross sections and fluxes stored in the SAPHYB are <i>not</i> SPH corrected.
VALE	keyword used to define a user-defined quantity as global parameter. This keyword must be followed by the type of parameter.
FLOT	keyword used to indicate that the user-defined global parameter is a floating point value.
CHAI	keyword used to indicate that the user-defined global parameter is a <b>character*12</b> value.
ENTI	keyword used to indicate that the user-defined global parameter is an integer value.
ISOT	keyword used to select the set of particularized isotopes.
TOUT	keyword used to select all the available isotopes in the reference MICROLIB named <i>HMIC</i> as particularized isotopes.
MILI	keyword used to select the isotopes in the reference MICROLIB named <i>HMIC</i> from a specific mixture as particularized isotopes.
<i>imil</i>	index of the mixture where the particularized isotopes are recovered.
FISS	keyword used to select all the available fissile isotopes in the reference MICROLIB named <i>HMIC</i> as particularized isotopes.
PF	keyword used to select all the available fission products in the reference MICROLIB named <i>HMIC</i> as particularized isotopes.
<i>HNAISO(i)</i>	<b>character*12</b> user-defined isotope name. $N_{iso}$ is the total number of explicitly-selected particularized isotopes.
MACR	keyword used to select a type of macroscopic set. A maximum of two macroscopic sets is allowed.
<i>HNAMAC</i>	<b>character*8</b> user-defined name of the macroscopic set.

TOUT	keyword used to select all the available isotopes in the macroscopic set.																																								
REST	keyword used to remove all the particularized isotope contributions from the macroscopic set.																																								
REAC	keyword used to select the set of nuclear reactions.																																								
HNAREA(i)	<p><b>character*4</b> name of a user-selected reaction. <math>N_{\text{reac}}</math> is the total number of selected reactions. <i>HNAREA(i)</i> is chosen among the following values:</p> <table> <tr><td>TOTA</td><td>Total cross sections</td></tr> <tr><td>TOP1</td><td>Total <math>P_1</math>-weighted cross sections</td></tr> <tr><td>ABSO</td><td>Absorption cross sections</td></tr> <tr><td>SNNN</td><td>Excess cross section due to (n,xn) reactions</td></tr> <tr><td>FISS</td><td>Fission cross section</td></tr> <tr><td>CHI</td><td>Steady-state fission spectrum</td></tr> <tr><td>NUFI</td><td><math>\nu\Sigma_f</math> cross sections</td></tr> <tr><td>ENER</td><td>Energy production cross section, taking into account all energy production reactions</td></tr> <tr><td>EFIS</td><td>Energy production cross section for (n,f) reaction only</td></tr> <tr><td>EGAM</td><td>Energy production cross section for (n,<math>\gamma</math>) reaction only</td></tr> <tr><td>FUIT</td><td><math>B^2</math> times the leakage coefficient</td></tr> <tr><td>SELF</td><td>within-group <math>P_0</math> scattering cross section</td></tr> <tr><td>DIFF</td><td>scattering cross section for each available Legendre order. These cross sections are <i>not</i> multiply by the <math>2\ell + 1</math> factor.</td></tr> <tr><td>PROF</td><td>profile of the transfer cross section matrices (i.e. position of the non-zero element in the transfer cross section matrices)</td></tr> <tr><td>TRAN</td><td>transfer cross section matrices for each available Legendre order. These cross sections are multiply by the <math>2\ell + 1</math> factor.</td></tr> <tr><td>CORR</td><td>transport correction. Note that the cross sections stored in the SAPHYB are <i>not</i> transport corrected.</td></tr> <tr><td>STRD</td><td>STRD cross sections used to compute the diffusion coefficients</td></tr> <tr><td>NP</td><td>(n,p) production cross sections</td></tr> <tr><td>NT</td><td>(n,t) production cross sections</td></tr> <tr><td>NA</td><td>(n,<math>\alpha</math>) production cross sections</td></tr> </table>	TOTA	Total cross sections	TOP1	Total $P_1$ -weighted cross sections	ABSO	Absorption cross sections	SNNN	Excess cross section due to (n,xn) reactions	FISS	Fission cross section	CHI	Steady-state fission spectrum	NUFI	$\nu\Sigma_f$ cross sections	ENER	Energy production cross section, taking into account all energy production reactions	EFIS	Energy production cross section for (n,f) reaction only	EGAM	Energy production cross section for (n, $\gamma$ ) reaction only	FUIT	$B^2$ times the leakage coefficient	SELF	within-group $P_0$ scattering cross section	DIFF	scattering cross section for each available Legendre order. These cross sections are <i>not</i> multiply by the $2\ell + 1$ factor.	PROF	profile of the transfer cross section matrices (i.e. position of the non-zero element in the transfer cross section matrices)	TRAN	transfer cross section matrices for each available Legendre order. These cross sections are multiply by the $2\ell + 1$ factor.	CORR	transport correction. Note that the cross sections stored in the SAPHYB are <i>not</i> transport corrected.	STRD	STRD cross sections used to compute the diffusion coefficients	NP	(n,p) production cross sections	NT	(n,t) production cross sections	NA	(n, $\alpha$ ) production cross sections
TOTA	Total cross sections																																								
TOP1	Total $P_1$ -weighted cross sections																																								
ABSO	Absorption cross sections																																								
SNNN	Excess cross section due to (n,xn) reactions																																								
FISS	Fission cross section																																								
CHI	Steady-state fission spectrum																																								
NUFI	$\nu\Sigma_f$ cross sections																																								
ENER	Energy production cross section, taking into account all energy production reactions																																								
EFIS	Energy production cross section for (n,f) reaction only																																								
EGAM	Energy production cross section for (n, $\gamma$ ) reaction only																																								
FUIT	$B^2$ times the leakage coefficient																																								
SELF	within-group $P_0$ scattering cross section																																								
DIFF	scattering cross section for each available Legendre order. These cross sections are <i>not</i> multiply by the $2\ell + 1$ factor.																																								
PROF	profile of the transfer cross section matrices (i.e. position of the non-zero element in the transfer cross section matrices)																																								
TRAN	transfer cross section matrices for each available Legendre order. These cross sections are multiply by the $2\ell + 1$ factor.																																								
CORR	transport correction. Note that the cross sections stored in the SAPHYB are <i>not</i> transport corrected.																																								
STRD	STRD cross sections used to compute the diffusion coefficients																																								
NP	(n,p) production cross sections																																								
NT	(n,t) production cross sections																																								
NA	(n, $\alpha$ ) production cross sections																																								
NAME	<p>key word used to define mixture names. By default, mixtures names are of the form <i>HNAMIX(i)</i>, where</p> <pre>WRITE(HNAMIX(I), '(3HMIX,I5.5)') I</pre>																																								
HNAMIX(i)	Character*20 user-defined mixture name. $N_m$ is the number of mixtures.																																								

### 3.19.2 Modification data input for module SAP:

Table 72: Structure (**saphyb\_data2**)

```
[ EDIT iprint ]
[ CRON ]
[[ parkey value ]]
[ ORIG orig ]
[ SET xtr { S | DAY | YEAR } ]
;
```

where

<b>EDIT</b>	key word used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module <b>SAP</b> :. =0 for no print; =1 for minimum printing (default value).
<b>CRON</b>	key word used to force the kinetics data to be placed into the <b>divers</b> directory. By default, the kinetics data is placed in the <b>cinetique</b> directory of each mixture subdirectory. The <b>CRON</b> option can only be used if the Saphyb contains a unique mixture. This option is mandatory if the Saphyb is to be read by the Lisaph module of Cronos.
<i>parkey</i>	<b>character*4</b> keyword associated to a user-defined global parameter.
<i>value</i>	floating-point, integer or <b>character*12</b> value of a user-defined global parameter.
<b>ORIG</b>	key word used to define the father node in the global parameter tree. By default, the index of the previous elementary calculation is used.
<i>orig</i>	index of the elementary calculation associated to the father node in the global parameter tree.
<b>SET</b>	keyword used to recover the flux normalization factor already stored on <i>BRNNAM</i> from a sub-directory corresponding to a specific time.
<i>xtr</i>	time associated with the current flux calculation. The name of the sub-directory where this information is stored will be given by 'DEPL-DAT'//CNN where CNN is a <b>character*4</b> variable defined by <b>WRITE(CNN,'(I4)')</b> INN where INN is an index associated with the time <i>xtr</i> .
<b>S</b>	keyword to specify that the time is given in seconds.
<b>DAY</b>	keyword to specify that the time is given in days.
<b>YEAR</b>	keyword to specify that the time is given in years.

### 3.19.3 Modification (catenate) data input for module **SAP**:

Table 73: Structure (**saphyb\_data3**)

```
[ EDIT iprint ]
[ ORIG orig ]
[[ parkey value ]]
[ WARNING-ONLY ]
;
```

where

<b>EDIT</b>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module <b>SAP</b> :. =0 for no print; =1 for minimum printing (default value).
<i>parkey</i>	<b>character*4</b> keyword associated to a user-defined global parameter.
<i>value</i>	floating-point, integer or <b>character*12</b> value of a user-defined global parameter.

ORIG	keyword used to define the father node in the parameter tree. By default, the index of the previous elementary calculation is used.
<i>orig</i>	index of the elementary calculation associated to the father node in the parameter tree.
WARNING-ONLY	This option is useful if an elementary calculation in <i>SAPRHS</i> is already present in <i>SAPNAM</i> . If this keyword is set, a warning is send and the <i>SAPNAM</i> values are kept, otherwise the run is aborted (default).

### 3.20 The MC: module

This component of the lattice code is dedicated to the Monte-Carlo solution of the transport equation in multigroup approximation.

The calling specifications are:

Table 74: Structure (MC:)

```
OUTMC [ TRACK ] := MC: [ OUTMC ] TRACK { MICRO | MACRO } :: (MC_data)
```

where

<i>OUTMC</i>	<b>character*12</b> name of a MONTE-CARLO (type L_MC) object open in modification or creation mode.
<i>TRACK</i>	<b>character*12</b> name of a NXT: TRACKING (type L_TRACK) object open in read-only or modification mode. Object <i>TRACK</i> must be constructed with option MC activated (see Section 3.4.3). Opening <i>TRACK</i> in modification mode is useful to add tracking information to be plotted with module PSP: (see Section 3.27).
<i>MICRO</i>	<b>character*12</b> name of a MICROLIB (type L_LIBRARY) object open in read-only mode. The information on the embedded macrolib is used.
<i>MACRO</i>	<b>character*12</b> name of a MACROLIB (type L_MACROLIB) object open in read-only mode.
<i>MC_data</i>	input data structure containing specific data (see Section 3.20.1).

#### 3.20.1 Data input for module MC:

Table 75: Structure (MC\_data)

```
[ EDIT iprint ]
KCODE nsrck ikz kct
[ SEED iseed ] [ N2N ]
[ TALLY
    [ MERG { COMP | NONE |
            REGI (iregm(ii),ii=1,nregio) |
            MIX [ (imixm(ii),ii=1,nbmix) ] } ]
    [ COND [ { NONE | ( icond(ii), ii=1,ngcond) } ] ]
ENDT ]
;
```

where

<i>EDIT</i>	keyword used to set <i>i</i> print.
<i>i</i> print	index used to control the printing in module MC:. =0 for no print; =1 for minimum printing (default value); =100 to add free-path information in object <i>TRACK</i> (must be open in modification mode in that case).



KCODE	keyword used to define the power iteration settings.
nsrck	number of neutrons generated per cycle
ikz	number of inactive cycles
kct	number of active cycles
SEED	keyword used to set the initial seed integer for the random number generator. By default, the seed integer is set from the processor clock.
iseed	initial seed integer
N2N	keyword used to enable an explicit treatment of $(n, 2n)$ reactions. In this case, N2N cross sections are expected to be available in the macrolib. By default, $(n, 2n)$ reactions are taken into account implicitly by the correction on scattering cross sections.
TALLY	keyword used to define a tally (macrolib and effective multiplication factor). Using "TALLY ENDT" construct permits to obtain a virtual collision estimation of the effective multiplication factor <i>without</i> estimation of the macrolib information.
NONE	keyword to deactivate the homogeneization or the condensation.
MERG	keyword to specify that the neutron flux is to be homogenized over specified regions or mixtures.
REGI	keyword to specify that the homogenization of the neutron flux will take place over the following regions. Here $n_{regio} \leq maxreg$ with <i>maxreg</i> the maximum number of regions for which solutions were obtained.
iregm	array of homogenized region numbers to which are associated the old regions. In the editing routines a value of <i>iregm</i> =0 allows the corresponding region to be neglected.
MIX	keyword to specify that the homogenization of the neutron flux will take place over the following mixtures. Here we must have $nbmix \leq maxmix$ where <i>maxmix</i> is the maximum number of mixtures in the macroscopic cross section library.
imixm	array of homogenized region numbers to which are associated the material mixtures. In the editing routines a value of <i>imixm</i> =0 allows the corresponding isotopic mixtures to be neglected. For a mixture in this library which is not used in the geometry one should insert a value of 0 for the new region number associated with this mixture. By default, if MIX is set and <i>imixm</i> is not set, <i>imixm(ii)=ii</i> is assumed.
COMP	keyword to specify that the a complete homogenization is to take place.
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 ngcond condensed groups. The final value of <i>icond</i> will automatically be set to <i>ngroup</i> while the values of <i>icond</i> > <i>ngroup</i> will be dropped from the condensation. We must have $ngcond \leq ngroup$ . By default, if COND is set and <i>icond</i> is not set, all energy groups are condensed together.
ENDT	keyword used to terminate the definition of a tally.

### 3.21 The T: module

A MACROLIB object can be defined directly using module **MAC:** (see Section 3.1) or as part of a MICROLIB object using module **LIB:** (see Section 3.2). It is possible to transpose a MACROLIB using the module **T:**. Transposition consists in

- renumbering the energy groups from thermal to fast
- transposing the transfer matrices (**SCAT**) so that the primary and secondary energy group indices are permuted
- storing **NUSIGF** information in **CHI** and storing **CHI** information in **NUSIGF**.

A transposed MACROLIB object permits to make adjoint flux calculations.

The general format of the data for the **T:** module is the following:

Table 76: Structure (**T:**)

*MACLIB1* := **T:** { *MACLIB2* | *LIBRARY* } ;

where

<i>MACLIB1</i>	<b>character*12</b> name of a the transposed MACROLIB
<i>MACLIB2</i>	<b>character*12</b> name of a the original MACROLIB
<i>LIBRARY</i>	<b>character*12</b> name of a the original MICROLIB containing an embedded MACROLIB.

### 3.22 The DMAC: module

This module is used to set fixed sources that can be used in the right hand term of an adjoint fixed source eigenvalue problem. This type of equation appears in generalized perturbation theory (GPT) applications. The fixed sources set in **DMAC:** are corresponding to the gradient of a reference macrolib with respect to homogenization and condensation of the cross-section information. The gradient of a cross section  $\Sigma(\mathbf{r}) = \text{col}\{\Sigma_1(\mathbf{r}), \Sigma_2(\mathbf{r})\}$  with respect to homogenization and condensation is defined as

$$\nabla P\{\phi(\zeta); \mathbf{r}\} = P\{\phi(\mathbf{r})\} \begin{bmatrix} \frac{\Sigma_1(\mathbf{r})}{\langle \Sigma, \phi \rangle} - \frac{1}{\langle \phi \rangle} \\ \frac{\Sigma_2(\mathbf{r})}{\langle \Sigma, \phi \rangle} - \frac{1}{\langle \phi \rangle} \end{bmatrix}$$

where the homogenized and condensed cross section is an homogeneous functional of the flux defined as

$$P\{\phi(\mathbf{r})\} = \frac{\langle \Sigma, \phi \rangle}{\langle \phi \rangle} \quad .$$

Each fixed source  $\nabla P\{\phi(\zeta); \mathbf{r}\}$  is orthogonal to the flux  $\phi(\mathbf{r})$ .

The calling specifications are:

Table 77: Structure (**DMAC:**)

*SOURCE* := **DMAC:** *FLUX* { *MICRO* | *MACRO* } *TRACK* :: (**DMAC\_data**)

where

<i>SOURCE</i>	<b>character*12</b> name of a FIXED SOURCES (type <b>L_GPT</b> ) object open in creation mode. This object contains a set of adjoint fixed sources corresponding to different macro-regions, macro-groups and cross-section types present in the reference macrolib.
<i>FLUX</i>	<b>character*12</b> name of a reference FLUX (type <b>L_FLUX</b> ) object open in read-only mode.
<i>MICRO</i>	<b>character*12</b> name of a reference MICROLIB (type <b>L_LIBRARY</b> ) object open in read-only mode. The information on the embedded macrolib is used.
<i>MACRO</i>	<b>character*12</b> name of a reference MACROLIB (type <b>L_MACROLIB</b> ) object open in read-only mode.
<i>TRACK</i>	<b>character*12</b> name of a reference TRACKING (type <b>L_TRACK</b> ) object open in read-only mode.
<i>DMAC_data</i>	input data structure containing specific data (see Section <a href="#">3.22.1</a> ).

#### 3.22.1 Data input for module DMAC:

Table 78: Structure (**DMAC\_data**)

```
[ EDIT iprint ]
[ RATE
  [ MERG { COMP | NONE |
```

continued on next page

Structure (DMAC\_data)

continued from last page

```

      REGI (iregm(ii),ii=1,nregio) |
      MIX [ (imixm(ii),ii=1,nbmix) ] } ]
    [ COND [ { NONE | ( icond(ii), ii=1,ngcond) } ] ]
  ENDR ]
;
```

where

EDIT	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module DMAC: . =0 for no print; =1 for minimum printing (default value).
RATE	keyword used to define the homogenization and condensation limits.
NONE	keyword to deactivate the homogeneization or the condensation.
MERG	keyword to specify that the neutron flux is to be homogenized over specified regions or mixtures.
REGI	keyword to specify that the homogenization of the neutron flux will take place over the following regions. Here $nregio \leq maxreg$ with <i>maxreg</i> the maximum number of regions for which solutions were obtained.
<i>iregm</i>	array of homogenized region numbers to which are associated the old regions. In the editing routines a value of <i>iregm</i> =0 allows the corresponding region to be neglected.
MIX	keyword to specify that the homogenization of the neutron flux will take place over the following mixtures. Here we must have $nbmix \leq maxmix$ where <i>maxmix</i> is the maximum number of mixtures in the macroscopic cross section library.
<i>imixm</i>	array of homogenized region numbers to which are associated the material mixtures. In the editing routines a value of <i>imixm</i> =0 allows the corresponding isotopic mixtures to be neglected. For a mixture in this library which is not used in the geometry one should insert a value of 0 for the new region number associated with this mixture. By default, if MIX is set and <i>imixm</i> is not set, <i>imixm</i> (ii)=ii is assumed.
COMP	keyword to specify that the a complete homogenization is to take place.
COND	keyword to specify that a group condensation of the flux is to be performed.
<i>icond</i>	array of increasing energy group limits that will be associated with each of the ngcond condensed groups. The final value of <i>icond</i> will automatically be set to <i>ngroup</i> while the values of <i>icond</i> > <i>ngroup</i> will be dropped from the condensation. We must have $ngcond \leq ngroup$ . By default, if COND is set and <i>icond</i> is not set, all energy groups are condensed together.
ENDR	keyword used to terminate the definition of the homogenization and condensation.

### 3.23 The DREF: module

This module is used to set fixed sources that can be used in the right hand term of an adjoint fixed source eigenvalue problem. This type of equation appears in generalized perturbation theory (GPT) applications. The fixed sources set in **DREF:** are corresponding to the gradient of the RMS functional which is a measure of the discrepancy between actual and reference (or target) power distributions. The actual power distribution is recovered from a *MICRO* or *MACRO* object. The reference power distribution is recovered from a *MICREF* or *MACREF* object.

Actual power values are defined as

$$P_i\{\phi(r)\} \equiv \langle H, \phi \rangle_i = \int_0^\infty dE \int_{V_i} d^3r H(\mathbf{r}, E) \phi(\mathbf{r}, E)$$

where the power factors  $H(\mathbf{r}, E)$  and fluxes  $\phi(\mathbf{r}, E)$  are recovered from **H-FACTOR** and **FLUX-INTG** records in a **MACROLIB** object.

The RMS error on power distribution is an homogeneous functional of the flux defined as

$$F\{\phi(r)\} = \sum_i \left( \frac{\langle H, \phi \rangle_i}{\langle H, \phi \rangle} - \frac{P_i^*}{\sum_j P_j^*} \right)^2$$

where the reference (or target) powers  $P_i^*$  are obtained from the full-core reference transport calculation.

The gradient of functional  $F\{\phi(r)\}$  is a  $G$ -group function of space defined as

$$\nabla F\{\phi(\zeta); \mathbf{r}\} = \frac{2}{\langle H, \phi \rangle} \sum_i \left( \frac{\langle H, \phi \rangle_i}{\langle H, \phi \rangle} - \frac{P_i^*}{\sum_j P_j^*} \right) \left( \delta_i(\mathbf{r}) - \frac{\langle H, \phi \rangle_i}{\langle H, \phi \rangle} \right) \begin{bmatrix} H_1(\mathbf{r}) \\ H_2(\mathbf{r}) \\ \vdots \\ H_G(\mathbf{r}) \end{bmatrix}$$

where  $\delta_i(\mathbf{r}) = 1$  if  $\mathbf{r} \in V_i$  and  $= 0$  otherwise.

Each fixed source  $\nabla F\{\phi(\zeta); \mathbf{r}\}$  is orthogonal to the flux  $\phi(\mathbf{r})$ .

The calling specifications are:

Table 79: Structure (**DREF:**)

```
SOURCE := DREF: FLUX TRACK { MICRO | MACRO } { MICREF | MACREF }
[ :: [ EDIT iprint ] [ RMS >>RMS_VAL<< ] ] ;
```

where

<b>SOURCE</b>	<b>character*12</b> name of a <b>FIXED SOURCES</b> (type <b>L_GPT</b> ) object open in creation mode. This object contains the adjoint fixed source corresponding to the RMS error on power distribution.
<b>FLUX</b>	<b>character*12</b> name of the actual <b>FLUX</b> (type <b>L_FLUX</b> ) object open in read-only mode.
<b>TRACK</b>	<b>character*12</b> name of the actual <b>TRACKING</b> (type <b>L_TRACK</b> ) object open in read-only mode.
<b>MICRO</b>	<b>character*12</b> name of the actual <b>MICROLIB</b> (type <b>L_LIBRARY</b> ) object open in read-only mode. The information on the embedded macrolib is used.
<b>MACRO</b>	<b>character*12</b> name of the actual <b>MACROLIB</b> (type <b>L_MACROLIB</b> ) object open in read-only mode.

<i>MICREF</i>	<b>character*12</b> name of reference (or target) MICROLIB (type L_LIBRARY) object open in read-only mode. The information contained in the embedded macrolib is used to compute $P_i^*$ values.
<i>MACREF</i>	<b>character*12</b> name of reference (or target) MACROLIB (type L_MACROLIB) object open in read-only mode. This information is used to compute $P_i^*$ values.
<i>EDIT</i>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module DREF:.. =0 for no print; =1 for minimum printing (default value).
<i>RMS</i>	keyword used to recover the RMS error on power distribution in a CLE-2000 variable.
<i>RMS_VAL</i>	<b>character*12</b> CLE-2000 variable name in which the extracted RMS value will be placed.

### 3.24 The SENS: module

This module is used to perform an explicit sensitivity analysis of  $k_{eff}$  to nuclear data represented by the cross sections.<sup>[?]</sup> The calculations are performed using adjoint-based first-order-linear perturbation theory and require the adjoint flux (see Section 3.8). The sensitivity coefficients are stored in a *SDF* text file that is compatible with the JAVAPENO module of SCALE<sup>[?]</sup> (this compatibility is achieved via a slight modification of the *rdragon* execution script). An example of modification is presented in the file `sens.save` from the *non regression testcase sens.x2m*.

The calling specifications are:

Table 80: Structure (SENS:)

```
SENS.sdf := SENS: FLUNAM ADJ.FLUNAM TRKNAM MACRO :: (SENS_data)
```

where

<i>SENS.sdf</i>	<b>character*12</b> name of a SDF file object that is created by SENS:.
<i>FLUNAM</i>	<b>character*12</b> name of the required FLUX (type L_FLUX) object open in read-only mode.
<i>ADJ.FLUNAM</i>	<b>character*12</b> name of the required ADJOINT FLUX (type L_FLUX) object open in read-only mode.
<i>TRKNAM</i>	<b>character*12</b> name of the required TRACKING (type L_TRACK) object open in read-only mode.
<i>MACRO</i>	<b>character*12</b> name of the required MACROLIB (type L_MACROLIB) object open in read-only mode.
<i>SENS_data</i>	input data structure containing specific data (see Section 3.24.1).

#### 3.24.1 Data input for module SENS:

Table 81: Structure (SENS\_data)

```
[ EDIT iprint ]
[ ANIS nanis ]
;
```

where

<b>EDIT</b>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module SENS:. =0 for no print; =1 for minimum printing (default value).
<b>ANIS</b>	keyword used to specify the level <i>naniso</i> of anisotropy permitted in the calculation.
<i>nanis</i>	number of Legendre orders for the representation of the scattering cross sections and the anisotropy of the flux. The default value is <i>nanis</i> =1 corresponding to the use

of isotropic scattering cross sections and integrated flux. The number of Legendre orders used for the sensitivity calculations is the lowest between *nanis* and the level of anisotropy available in the *MACRO* data.



### 3.25 The DUO: module

This module is used to perform a perturbative analysis of two systems in fundamental mode conditions using the Clio formula and to determine the origins of Keff discrepancies.

The calling specifications are:

Table 82: Structure (**DUO:**)

DUO: <i>MICLIB1 MICLIB2</i> :: ( <b>DUO_data</b> )
--

where

<i>MICLIB1</i>	<b>character*12</b> name of the first MICROLIB (type <b>L_LIBRARY</b> ) object open in read-only mode.
<i>MICLIB2</i>	<b>character*12</b> name of the second MICROLIB (type <b>L_LIBRARY</b> ) object open in read-only mode.
<i>DUO_data</i>	input data structure containing specific data (see Section <a href="#">3.25.1</a> ).

#### 3.25.1 Data input for module DUO:

Note that the input order must be respected.

Table 83: Structure (**DUO\_data**)

<pre>[ EDIT <i>iprint</i> ] [ ENERGY ] [ ISOTOPE ] [ MIXTURE ] [ REAC   [[ <i>reac</i> [ PICK &gt;&gt; <i>deltaRho</i> &lt;&lt; ] ]] ENDREAC ] ;</pre>
--

where

<b>EDIT</b>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module DUO: . =0 for no print; =1 for minimum printing (default value).
<b>ENERGY</b>	keyword used to perform a perturbation analysis as a function of the energy group indices.
<b>ISOTOPE</b>	keyword used to perform a perturbation analysis as a function of the isotopes present in the geometry.
<b>MIXTURE</b>	keyword used to perform a perturbation analysis as a function of the mixtures indices.
<b>REAC</b>	keyword used to perform a perturbation analysis for specific nuclear reactions.

**reac** **character\*8** name of a nuclear reaction  $\sigma_x$ . The reactivity effect is computed using the formula

$$\delta\lambda_x = \frac{(\phi_1^*)^\top \delta \mathbb{S}_x \phi_2}{(\phi_1^*)^\top \mathbb{P}_2 \phi_2}. \quad (3.34)$$

where  $\mathbb{S}_x$  is a matrix containing the contributions of the reaction  $\sigma_x$ . The other symbols are defined in Sect. 3.25.2. Examples of reaction names are:

**NTOTO**: total cross section

**NG**: radiative capture cross section

**N2N**: (n,2n) cross section

**NFTOT**: fission cross section

**NELAS**: elastic scattering cross section

**SCAT00**: scattering matrix

**NUSIGF**: dyadic product of the fission spectrum times  $\nu$  fission cross section

**LEAK**: neutron leakage

The balance relation for the global reactivity effect is

$$\delta\lambda = \delta\lambda_{\text{NTOTO}} - \delta\lambda_{\text{SCAT00}} - \frac{\delta\lambda_{\text{NUSIGF}}}{K_{\text{eff}}} + \delta\lambda_{\text{LEAK}} \quad (3.35)$$

where  $K_{\text{eff}}$  is the effective multiplication factor.

**PICK** keyword used to recover the delta-rho discrepancy for reaction *reac* in a CLE-2000 variable.

**deltaRho** **character\*12** CLE-2000 variable name in which the extracted delta-rho discrepancy will be placed.

**ENDREAC** keyword used to indicate that no more nuclear reactions will be analysed.

### 3.25.2 Theory

The module **DUO**: is an implementation of the CLIO perturbative analysis method, as introduced in Ref. ?. This method is useful for comparing two similar systems in fundamental mode conditions. It is based on fundamental mode balance equations that must be satisfied by the direct and adjoint solutions of each of the two systems. The balance equation of the first system is written

$$\mathbb{L}_1 \phi_1 - \lambda_1 \mathbb{P}_1 \phi_1 = \mathbf{0} \quad \text{and} \quad \mathbb{L}_1^\top \phi_1^* - \lambda_1 \mathbb{P}_1^\top \phi_1^* = \mathbf{0} \quad (3.36)$$

where

$\mathbb{L}_1$  = absorption (total plus leakage minus scattering) reaction rate matrix

$\mathbb{P}_1$  = production (nu times fission) reaction rate matrix

$\lambda_1$  = one over the effective multiplication factor

$\phi_1$  = direct multigroup flux in each mixture of the geometry

$\phi_1^*$  = adjoint multigroup flux in each mixture of the geometry.

Similarly, the balance equation of the second system is written

$$\mathbb{L}_2 \phi_2 - \lambda_2 \mathbb{P}_2 \phi_2 = \mathbf{0}. \quad (3.37)$$

Next, we write

$$\mathbb{L}_2 = \mathbb{L}_1 + \delta \mathbb{L} \quad \mathbb{P}_2 = \mathbb{P}_1 + \delta \mathbb{P}, \quad \phi_2 = \phi_1 + \delta \phi \quad \text{and} \quad \lambda_2 = \lambda_1 + \delta \lambda. \quad (3.38)$$

Substituting Eq. (3.38) into Eq. (3.37), we write

$$\mathbb{L}_1 \phi_1 + \mathbb{L}_1 \delta \phi + \delta \mathbb{L} \phi_2 - [\lambda_1 \mathbb{P}_1 \phi_1 + \lambda_1 \mathbb{P}_1 \delta \phi + (\lambda_2 \mathbb{P}_2 - \lambda_1 \mathbb{P}_1) \phi_2] = \mathbf{0}. \quad (3.39)$$

Following the guideline from Ref. ?, we subtract Eq. (3.36) from Eq. (3.39) to obtain

$$(\mathbb{L}_1 - \lambda_1 \mathbb{P}_1) \delta \phi = (-\delta \mathbb{L} + \lambda_2 \mathbb{P}_2 - \lambda_1 \mathbb{P}_1) \phi_2 \quad (3.40)$$

Next, we left-multiply this matrix system by a row vector equal to  $(\phi_1^*)^\top$ , in order to make the LHS vanishing. This operation is written

$$(\phi_1^*)^\top (\mathbb{L}_1 - \lambda_1 \mathbb{P}_1) \delta \phi = (\phi_1^*)^\top (-\delta \mathbb{L} + \lambda_2 \mathbb{P}_2 - \lambda_1 \mathbb{P}_1) \phi_2 = 0 \quad (3.41)$$

because

$$(\phi_1^*)^\top (\mathbb{L}_1 - \lambda_1 \mathbb{P}_1) = \mathbf{0}^\top \quad (3.42)$$

in term of Eq. (3.36).

Using the relation  $\lambda_2 \mathbb{P}_2 - \lambda_1 \mathbb{P}_1 = \delta \lambda \mathbb{P}_2 + \lambda_1 \delta \mathbb{P}$ , Eq. (3.41) can be rewritten as

$$(\phi_1^*)^\top (-\delta \mathbb{L} + \delta \lambda \mathbb{P}_2 + \lambda_1 \delta \mathbb{P}) \phi_2 = 0 \quad (3.43)$$

so that

$$\delta \lambda = \frac{(\phi_1^*)^\top (\delta \mathbb{L} - \lambda_1 \delta \mathbb{P}) \phi_2}{(\phi_1^*)^\top \mathbb{P}_2 \phi_2}. \quad (3.44)$$

Equation 3.44 is *not* a first order perturbation approximation of  $\delta \lambda$ ; it is an *exact* expression of it. Its numerator is used to obtain every component of  $\delta \lambda$  in term of energy group, isotope, mixture and/or nuclear reaction.

### 3.26 The S2M: module

This module is used to extract macroscopic cross-section data from a Matlab-formatted ASCII file generated by the SERPENT Monte Carlo code (see Ref. ?) and to convert it to the MACROLIB format.

The calling specifications are:

Table 84: Structure (**S2M:**)

```
MACRO := S2M: matlab.m :: (S2M_data)
```

where

<i>MACRO</i>	<b>character*12</b> name of the required MACROLIB (type <b>L_MACROLIB</b> ) object that is created by <b>S2M:</b> .
<i>matlab.m</i>	<b>character*12</b> name of a ASCII file, open in read-only mode, containing Matlab-formatted SERPENT information.
<i>S2M_data</i>	input data structure containing specific data (see Section 3.26.1).

#### 3.26.1 Data input for module S2M:

Table 85: Structure (**S2M\_data**)

```
[ EDIT iprint ]
[ IDX idx ]
[ B1 ]
;
```

where

<b>EDIT</b>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module <b>S2M:</b> . =0 for no print; =1 for minimum printing (default value).
<b>IDX</b>	keyword used to specify the occurrence index of a flux calculation in the SERPENT output file. This index generally refers to the burnup step.
<i>idx</i>	occurrence index.
<b>B1</b>	keyword used to specify that diffusion coefficients and other fundamental-mode information are to be recovered from the SERPENT output file.

### 3.27 The PSP: module

The **PSP:** module is used to generate a graphical file in a PostScript ASCII format for a DRAGON 2-D geometry which can be analyzed using the **EXCELT:** or **NXT:** tracking module (see Sections. 3.4.2 and 3.4.3). The module **PSP:** is based on the **PSPLOT** Fortran library from Nova Southeastern University.<sup>[?]</sup> Since only a few **PSPLOT** routines were required and because additional PostScript routine not present in the original package were needed, the routines have been completely readapted to DRAGON. These routines are no longer machine dependent. The PostScript files generated by DRAGON can be viewed by any PostScript viewer, such as Ghostview<sup>[?]</sup> or sent to a printer compatible with this language. In DRAGON the **PSP:** module is activated using the following list of commands:

Table 86: Structure (**PSP:**)

$PSGEO := \text{PSP: } PSGEO \{ GEONAM \mid TRKNAM \} [ FLUNAM ] :: (\text{descpsp})$
---

where

<i>PSGEO</i>	<b>character*12</b> name of the file that will contain the graphical description in a POSTSCRIPT format. This file must have a sequential ASCII format.
<i>GEONAM</i>	<b>character*12</b> name of a read-only GEOMETRY (see Section Section 3.3).
<i>TRKNAM</i>	<b>character*12</b> name of an NXT or EXCELL type read-only TRACKING (see Sections. 3.4.2 and 3.4.3).
<i>FLUNAM</i>	<b>character*12</b> name of an optional read-only FLUXUNK (see Section 3.8). It is required only if a flux mapping plot is requested.
<b>(descpsp)</b>	structure containing the input data to this module (see Section 3.27.1).

#### 3.27.1 Data input for module PSP:

Table 87: Structure (**descpsp**)

$\begin{aligned} &[ \text{EDIT } i\text{print} ] \\ &[ \text{FILL } \{ \text{NONE} \mid \text{GRAY} \mid \text{RGB} \mid \text{CMYK} \mid \text{HSB} \} [ \text{NOCONTOUR} ] ] \\ &[ \text{TYPE } \{ \text{REGION} \mid \text{MIXTURE} \mid \text{FLUX} \mid \text{MGFLUX} \} ] \end{aligned}$
--

where

<b>EDIT</b>	keyword used to modify the print level <i>i</i> print.
<i>i</i> print	index used to control the printing in this module. It must be set to 0 if no printing on the output file is required.
<b>FILL</b>	keyword to specify the drawing options.
<b>NONE</b>	keyword to specify that only region contour are to be drawn.

GRAY	keyword to specify that the regions will be filled with various levels of gray.
RGB	keyword to specify that the regions will be filled with various colors taken using the RGB color scheme.
CMYK	keyword to specify that the regions will be filled with various colors taken using the CMYK color scheme.
HSB	keyword to specify that the regions will be filled with various colors taken using the HSB color scheme. This is the default option.
NOCONTOUR	keyword to specify that the contour lines delimiting each region will not be drawn.
TYPE	keyword to specify the type of graphics generated.
REGION	keyword to specify that different colors or gray levels will be associated with each region. This is the default option.
MIXTURE	keyword to specify that different colors or gray levels will be associated with each mixture.
FLUX	keyword to specify that the group integrated flux is to be drawn.
MGFLUX	keyword to specify that the group flux is to be drawn.

## 4 THE SALOME-RELATED MODULES

A few modules have been introduced in DRAGON Version5 in order to facilitate the processing of geometries originating from the Geometry module of SALOME.<sup>[?]</sup> The methods presented in this section have been initially developed at CEA SERMA and integrated in the TDT code.<sup>[?, ?]</sup> In the course of year 2001, a subset of these methods have been integrated into a development version of DRAGON under the terms of its LGPL license as a prototyping exercise of the DESCARTES operation.<sup>[?]</sup>

The track generator **SALT**: is a direct descendent of this prototyping exercise. Later, we have extracted the 5000 lines of Fortran-90 code responsible for the track calculation and have rewritten them in a way consistent with the **NXT**: tracking methodology and with the DRAGON architecture.

### 4.1 The G2S: module

The module **G2S**: is used to compute the SALOME-formatted surfacic elements corresponding to a gigogne geometry. The general format of the input data for the **G2S**: module is the following:

Table 88: Structure (**G2S**:)

```
[ SURFIL ] [ PSFIL ] := G2S: { SURFIL | GEONAM } :: (G2S_data)
```

where

<i>SURFIL</i>	<b>character*12</b> name of the SALOME-formatted sequential ASCII file used to store the surfacic elements of the geometry.
<i>PSFIL</i>	<b>character*12</b> name of the sequential ASCII file used to store a postscript representation of the geometry corresponding to <i>SURFIL</i> or <i>GEONAM</i> .
<i>GEONAM</i>	<b>character*12</b> name of the <i>read-only</i> GEOMETRY data structure. This structure may be build using the operator <b>GEO</b> : (see Section 3.3).
<i>G2S_data</i>	input data structure containing specific data (see Section 4.1.1).

#### 4.1.1 Data input for module G2S:

Table 89: Structure (**G2S\_data**)

```
[ EDIT iprint ]
[ { DRAWNOD | DRAWMIX } ]
;
```

where

<b>EDIT</b>	keyword used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module <b>G2S</b> :. =0 for no print; =1 for minimum printing (default value).

DRAWNOD	keyword used to print the region indices on the LHS postscript plot <i>PSFIL</i> . By default, no indices are printed.
DRAWMIX	keyword used to print the material mixture indices on the LHS postscript plot <i>PSFIL</i> . By default, no indices are printed.



## 4.2 The G2MC: module

The module **G2MC:** is used to compute the SERPENT-, TRIPOLI4-, or MCNP-formatted surfacic elements corresponding to a SALOME-formatted or gigogne geometry. The general format of the input data for the **G2MC:** module is the following:

Table 90: Structure (**G2MC:**)

*MCFIL* [ *PSFIL* ] := **G2MC:** { *SURFIL* | *GEONAM* } ;

where

<i>MCFIL</i>	<b>character*12</b> name of the SERPENT-, TRIPOLI4- or MCNP-formatted sequential ASCII file used to store the surfacic elements of the geometry. A SERPENT file is produced if the file name has extension ".sp". A TRIPOLI4 file is produced if the file name has extension ".tp". Otherwise, a MCNP file is produced. This file is to be included in the complete dataset of a Monte Carlo code.
<i>PSFIL</i>	<b>character*12</b> name of the sequential ASCII file used to store a postscript representation of the geometry corresponding to <i>GEONAM</i> .
<i>SURFIL</i>	<b>character*12</b> name of the <i>read-only</i> SALOME-formatted sequential ASCII file used to store the surfacic elements of the geometry.
<i>GEONAM</i>	<b>character*12</b> name of the <i>read-only</i> GEOMETRY data structure. This structure may be build using the operator <b>GEO:</b> (see Section 3.3).

### 4.3 The SALT: tracking module

The **SALT:** module can process general 2-D geometries. It is used to compute the tracking information requested in the method of collision probabilities or in the method of characteristics.

The calling specification for this module is:

Table 91: Structure (**SALT:**)

```
TRKNAM TRKFIL := SALT: SURFIL [ GEONAM ] :: (desctrack) (descsalt)
```

where

<b>TRKNAM</b>	<b>character*12</b> name of the SALT TRACKING data structure that will contain region volume and surface area vectors in addition to region identification pointers and other tracking information.
<b>TRKFIL</b>	<b>character*12</b> name of the sequential binary tracking file used to store the tracks lengths.
<b>SURFIL</b>	<b>character*12</b> name of the sequential ASCII file used to store the surfacic elements of the geometry. This file may be build using the operator <b>G2S:</b> (see Section 4.1) or recovered from SALOME.
<b>GEONAM</b>	<b>character*12</b> name of the GEOMETRY data structure containing the double heterogeneity (Bihet) data.
<b>(desctrack)</b>	structure describing the general tracking data (see Section 3.4)
<b>(descsalt)</b>	structure describing the transport tracking data specific to <b>SALT:</b> .

All information for the modelization used can be found in ?. The **SALT:** specific tracking data in **(descsalt)** is defined as :

Table 92: Structure (**descsalt**)

```
[ ANIS nanis ]
[ { ONEG | ALLG } ]
[ [ QUAB iquab ] [ { SAPO | HEBE } ] ]
[ { PISO | PSPC [ CUT pcut ] } ]
[ { GAUS | CACA | CACB | LCMD | OPP1 | OGAU } [ nmU ] ]
{ TISO [ { EQW | GAUS | PNTN | SMS | LSN | QRN } ] nangl dens
  | TSPC [ { MEDI | EQW2 } ] nangl dens }
[ CORN pcorn ]
[ NOTR ]
[ NBSLIN nbslin ]
[ LONG ]
;
```

where

**ANIS** keyword to specify the order of scattering anisotropy.

<i>nanis</i>	order of anisotropy in transport calculation. A default value of 1 represents isotropic (or transport-corrected) scattering while a value of 2 correspond to linearly anisotropic scattering.
ONEG	keyword to specify that the tracking is read before computing each group-dependent collision probability or algebraic collapsing matrix (default value if <i>TRKFIL</i> is set). The tracking file is read in each energy group if the method of characteristics (MOC) is used.
ALLG	keyword to specify that the tracking is read once and the collision probability or algebraic collapsing matrices are computed in many energy groups. The tracking file is read once if the method of characteristics (MOC) is used.
QUAB	keyword to specify the number of basis point for the numerical integration of each micro-structure in cases involving double heterogeneity (Bihet).
<i>iquab</i>	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 <i>iquab</i> = 5.
SAPO	use the Sanchez-Pomraning double-heterogeneity model. <sup>[?]</sup>
HEBE	use the Hebert double-heterogeneity model (default option). <sup>[?]</sup>
PISO	keyword to specify that a collision probability calculation with isotropic reflection boundary conditions is required. It is the default option if a <i>TISO</i> type integration is chosen. To obtain accurate transmission probabilities for the isotropic case it is recommended that the normalization options in the <i>ASM:</i> module be used.
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 <i>TSPC</i> type integration is chosen. This calculation is only possible if the file was initially constructed using the <i>TSPC</i> option.
CUT	keyword to specify the input of cutting parameters for the specular collision probability of characteristic integration.
<i>pcut</i>	real value representing the maximum error allowed on the exponential function used for specular collision probability calculations. Tracks will be cut at a length such that the error in the probabilities resulting from this reduced track will be of the order of <i>pcut</i> . By default, the tracks are extended to infinity and <i>pcut</i> = 0.0. If this option is used in an entirely reflected case, it is recommended to use the <i>NORM</i> command in the <i>ASM:</i> module.
GAUS	keyword to specify that Gauss-Legendre polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. The conservation is ensured up to $P_{nmu-1}$ scattering.
CACA	keyword to specify that CACTUS type equal weight polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic scattering.
CACB	keyword to specify that CACTUS type uniformly distributed integration polar angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic scattering.
LCMD	keyword to specify that optimized (McDaniel-type) polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> This is the default option. The conservation is ensured only for isotropic scattering.

OPP1	keyword to specify that $P_1$ constrained optimized (McDaniel-type) polar integration angles are to be selected for the polar quadrature when a prismatic tracking is considered. <sup>[?]</sup> The conservation is ensured only for isotropic and linearly anisotropic scattering.
OGAU	keyword to specify that Optimized Gauss polar integration angles are to be selected for the method of characteristics. <sup>[?, ?]</sup> The conservation is ensured up to $P_{nmu-1}$ scattering.
<i>nmu</i>	user-defined number of polar angles. By default, a value consistent with <i>nangl</i> is computed by the code. For LCMD, OPP1, OGAU quadratures, <i>nmu</i> is limited to 2, 3 or 4.
TISO	keyword to specify that isotropic tracking parameters will be supplied. This is the default tracking option for cluster geometries.
TSPC	keyword to specify that specular tracking parameters will be supplied.
EQW	keyword to specify the use of equal weight quadrature. <sup>[?]</sup> The conservation is ensured up to $P_{nangl/2}$ scattering.
GAUS	(after TISO keyword) keyword to specify the use of the Gauss-Legendre quadrature. This option is valid only if an hexagonal geometry is considered.
PNTN	keyword to specify that Legendre-Techbychev quadrature quadrature will be selected. <sup>[?]</sup> The conservation is ensured only for isotropic and linearly anisotropic scattering.
SMS	keyword to specify that Legendre-trapezoidal quadrature quadrature will be selected. <sup>[?]</sup> The conservation is ensured up to $P_{nangl-1}$ scattering.
LSN	keyword to specify the use of the $\mu_1$ -optimized level-symmetric quadrature. The conservation is ensured up to $P_{nangl/2}$ scattering.
QRN	keyword to specify the use of the quadrupole range (QR) quadrature. <sup>[?]</sup>
MEDI	keyword to specify the use of a median angle quadrature in TSPC cases. For a rectangular Cartesian domain of size $X \times Y$ , the azimuthal angles in $(0, \pi/2)$ interval are obtained from formula

$$\phi_k = \tan^{-1} \frac{kY}{(2p+2-k)X}, \quad k = 1, 3, 5, \dots, 2p+1.$$

EQW2	keyword to specify the use of a standard cyclic quadrature without angles $\phi = 0$ and $\phi = \pi/2$ in TSPC cases. For a rectangular Cartesian domain of size $X \times Y$ , the azimuthal angles in $(0, \pi/2)$ interval are obtained from formula
------	--

$$\phi_k = \tan^{-1} \frac{kY}{(p+2-k)X}, \quad k = 1, 2, 3, \dots, p+1.$$

This is the default option.

<i>nangl</i>	angular quadrature parameter. For a 3-D EQW option, the choices are <i>nangl</i> = 2, 4, 8, 10, 12, 14 or 16. For a 3-D PNTN or SMS option, <i>nangl</i> is an even number smaller than 46. <sup>[?]</sup> For 2-D isotropic applications, any value of <i>nangl</i> may be used, equidistant angles will be selected.
--------------	--

For 2-D specular applications the input value must be of the form  $p+1$  where  $p$  is a prime number, as proposed in Ref. ?. In this case, the choice of *nangl* = 8, 12, 14, 18, 20, 24, or 30 are allowed.

<i>dens</i>	real value representing the density of the integration lines (in $\text{cm}^{-1}$ for 2-D Cartesian 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 SALT:.
-------------	---

CORN	keyword to specify the meaningful distance (cm) between a tracking line and a surfacic element.
<i>pcorn</i>	meaningful distance (cm) between a tracking line and a surfacic element. By default, $pcorn = 1.0 \times 10^{-5}$ cm.
NOTR	keyword to specify that the geometry will not be tracked. This is useful for 2-D geometries to generate a tracking data structure that can be used by the <b>PSP:</b> module (see Section 3.27). One can then verify visually if the geometry is adequate before the tracking process as such is undertaken.
NBSLIN	keyword to set the maximum number of segments in a single tracking line.
<i>nbsl</i>	integer value representing the maximum number of segments in a single tracking line. The default value is $nbsl = 100000$ .
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 <b>TLM:</b> module (see Section 3.15).

## 5 THE UTILITY MODULES

DRAGON contains a number of utility modules used to perform tasks not related to reactor physics. These modules are also available to any code built around the GAN generalized driver.<sup>[?]</sup>

### 5.1 The equality module

This module is used to duplicate a LCM object. The calling specifications are:

Table 93: Structure (**equality**)

```
NAME1 := [ NAME1 ] NAME2
[ :: [ EDIT iprint ] [ { OLD | SAP } ] [[ STEP { UP NOMDIR | AT index } ]] ] ;
```

<i>NAME1</i>	<b>character*12</b> name of the output data structure. It can be a LCM object (either memory-resident or XSM-based), a sequential binary file or a sequential ASCII file. If <i>NAME1</i> is a LCM object and if it appears on both sides, it is erased and refilled with the contents of <i>NAME2</i> .
<i>NAME2</i>	<b>character*12</b> name of the input data structure. It can be a LCM object (either memory-resident or XSM-based), a sequential binary file or a sequential ASCII file.
EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	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.
OLD	keyword used to import/export a LHS sequential ASCII file in 1995 LCM specification. By default, the up-to-date specification is used.
SAP	keyword used to import/export a LHS sequential ASCII file in Saphyr LCM specification.
STEP	keyword used to move in the LCM object hierarchy of <i>NAME2</i> before making the copy.
UP	keyword used to move up towards a sub-directory of <i>NAME2</i> of the active directory.
<i>NOMDIR</i>	copy the information located in the sub-directory named <i>NOMDIR</i> .
AT	keyword used to move up towards a component in an heterogeneous list of <i>NAME2</i> .
<i>index</i>	copy the information located in the <i>index</i> -th component of the heterogeneous list.

If both the RHS and LHS are LCM objects (either memory-resident or XSM-based), a single copy is performed. A memory-resident LCM object can be created from an XSM file or an XSM file can be created from a memory-resident LCM object. If the LHS is a sequential file and the RHS is a LCM object, an export is performed. The export format is either binary or ASCII. If the LHS is a LCM object and the RHS is a sequential file, an import is performed. The case where both the LHS and the RHS are sequential files is not supported.

## 5.2 The UTL: module

The UTL: module is used to perform utility actions on a LCM object. The calling specifications are:

Table 94: Structure (UTL:)

```
[ NAME1 := ] UTL: [ NAME1 ] ::
[[ DIR ]] [[ STEP { UP NOMDIR | AT index | DOWN | ROOT } ]]
[[ IMPR { BLOCK | index } { ileni | * } ]]
[[ CREA { BLOCK | index } [ ilenc1 ] ilenc2 = { (valc(i),i=ilenc1,ilenc2) | (ivalc(i),i=ilenc1,ilenc2)
| (hvalc(i),i=ilenc1,ilenc2) | (dvalc(i),i=ilenc1,ilenc2) } ]]
[[ DEL BLOCK ]]
[[ MULT { BLOCK | index } flott ]]
[[ { COPY | STAT { REL | ABS } | ADD } NOMREF NOMALT ]]
[ DUMP ] ;
```

<i>NAME1</i>	<b>character*12</b> name of the LCM object that will be treated by the utility module.
DIR	keyword used to print the active directory content.
STEP	keyword used to move in the LCM object hierarchy.
UP	keyword used to move up towards a sub-directory of the active directory.
NOMDIR	name of the sub-directory to which we wish to head.
AT	keyword used to move towards a component in an heterogeneous list of <i>NAME1</i> .
<i>index</i>	access the information located in the <i>index</i> -th component of the heterogeneous list.
DOWN	keyword to return to the sub-directory containing the active directory.
ROOT	keyword to return to the root directory of the LCM object.
IMPR	keyword to print the complete contents or part of the record <i>BLOCK</i> or component <i>index</i> located on the current directory.
MULT	keyword to multiply each element of a block or sub-directory in the active directory by a real constant. If <i>BLOCK</i> is a sub-directory, only floating point information contained in it is multiplied.
CREA	keyword used to create a block of information on the current directory.
DEL	keyword used to delete a block of information on the current directory.
<i>BLOCK</i>	name of the block or sub-directory selected.
<i>ileni</i>	maximum number of elements that the user wishes to print. A value of <i>ileni</i> =0 is permitted.
*	keyword, indicates that all the elements of a block will be printed. In a realistic case, the number of elements contained in a block may be rather large; this option must therefore be used with caution.
<i>ilenc1</i>	index of the first element included in the block. Can only be set if block <i>BLOCK</i> already exists. By default, <i>ilenc1</i> = 1.
<i>ilenc2</i>	index of the last element included in the block.

=	keyword, indicates that the input values will follow.
<i>valc</i>	real vector containing the information to be written in the record <i>BLOCK</i> .
<i>ivalc</i>	integer vector containing the information to be written in the record <i>BLOCK</i> .
<i>hvalc</i>	<b>character*4</b> array containing the information to be written in the record <i>BLOCK</i> .
<i>dvalc</i>	double precision array containing the information to be written in the record <i>BLOCK</i> .
<i>flott</i>	constant by which a block or sub-directory will be multiplied.
COPY	keyword used to copy an existing record or sub-directory onto a new record or sub-directory.
ADD	keyword used to add the contents of two records or two sub-directories. If <i>NOMREF</i> and <i>NOMALT</i> are two sub-directories, only the floating point information contained in them is added. The result is written into <i>NOMALT</i> .
STAT	keyword used to compare the contents of two records.
REL	the relative differences are printed.
ABS	the absolute differences are printed.
<i>NOMREF</i>	name of the reference block.
<i>NOMALT</i>	name of the block which may possibly be modified during the ADD and COPY operations.
DUMP	Dump the active directory of and its sub-directories to the printer.



### 5.3 The DELETE: module

This module is used to delete one or many LCM objects. The calling specifications are:

Table 95: Structure **(DELETE:)**

```
[[ NAME1 ]] := DELETE: [[ NAME1 ]] [ :: EDIT iprint ] ;
```

*NAME1*                **character\*12** name of a LCM object.

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

*iprint*                index used to control the printing of this module. Set to 1 to print the name of the deleted objects.

The names of the LCM object should be present on both the LHS and the RHS. A LCM object named **PARENT** can be deleted using the following command:

```
PARENT := DELETE: PARENT ;
```

#### 5.4 The BACKUP: module

This module is used to copy one or many LCM objects (memory-resident or XSM-based), along with all of its parent to a backup LCM object. The backup data structure can be stored on a single LCM object (either memory-resident or XSM-based), a sequential binary file or a sequential ASCII file. The calling specifications are:

Table 96: Structure (**BACKUP:**)

```
NAME1 := BACKUP: [ NAME1 ] [[ NAME2 ]] [ : :
[ EDIT iprint ]
[[ STEP { UP NOMDIR | AT index } ]] ;
```

<i>NAME1</i>	<b>character*12</b> name of LCM object used as a backup media.
<i>NAME2</i>	<b>character*12</b> name of LCM object to be transfer to the backup LCM object. This LCM object must be in a memory-resident or XSM-based format.
EDIT	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	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.
STEP	keyword used to move in the LCM object hierarchy of <i>NAME2</i> before making the backup.
UP	keyword used to move up towards a sub-directory of <i>NAME2</i> of the active directory.
<i>NOMDIR</i>	backup the information into the sub-directory named <i>NOMDIR</i> .
AT	keyword used to move up towards a component in an heterogeneous list of <i>NAME2</i> .
<i>index</i>	backup the information into the <i>index</i> -th component of the heterogeneous list.

If *NAME1* appears only on the LHS, it is created. If *NAME1* appears on both the LHS and the RHS, it is updated.

### 5.5 The RECOVER: module

This module is used to recover from a backup LCM object (see Section 5.4) one or many LCM objects (memory-resident or XSM-based). The calling specifications are:

Table 97: Structure **(RECOVER:)**

```
[[ NAME1 ]] := RECOVER: NAME2 [[ NAME1 ]] [ ::
[ EDIT iprint ]
[[ STEP { UP NOMDIR | AT index } ]] ] ;
```

<i>NAME1</i>	<b>character*12</b> name of the LCM objects that are to be recovered.
<i>NAME2</i>	<b>character*12</b> name of a backup LCM object.
<b>EDIT</b>	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	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.
<b>STEP</b>	keyword used to move in the LCM object hierarchy of <i>NAME2</i> before making the recover.
<b>UP</b>	keyword used to move up towards a sub-directory of <i>NAME2</i> of the active directory.
<i>NOMDIR</i>	recover the information located in the sub-directory named <i>NOMDIR</i> .
<b>AT</b>	keyword used to move up towards a component in an heterogeneous list of <i>NAME2</i> .
<i>index</i>	recover the information located in the <i>index</i> -th component of the heterogeneous list.

If *NAME1* appears only on the LHS, it is created. If *NAME1* appears on both the LHS and the RHS, it is replaced by the information located on the backup media.

### 5.6 The ADD: module

This module is used to add the floating point information contained of the two LCM object located on the RHS. The result is stored in a third output LCM object. The calling specifications are:

Table 98: Structure (**ADD:**)

$NAME1 := \text{ADD: } NAME2 \text{ } NAME3 ;$
--

<i>NAME1</i>	<b>character*12</b> name of a LCM object which contains the final information.
<i>NAME2</i>	<b>character*12</b> name of a LCM object which contains the first part of the initial information. One can use <i>NAME2=NAME1</i> .
<i>NAME3</i>	<b>character*12</b> name of a LCM object which contains the second part of the initial information.

### 5.7 The MPX: module

This module is used to multiply the floating point information contained in a LCM object located on the RHS by a user-defined real number. The result is stored in a second output LCM object. The calling specifications are:

Table 99: Structure (MPX:)

*NAME1* := MPX: *NAME2* :: *real* ;

<i>NAME1</i>	<b>character*12</b> name of a LCM object which contains the final information.
<i>NAME2</i>	<b>character*12</b> name of a LCM object which contains the the initial information. One can use <i>NAME2=NAME1</i> .
<i>real</i>	real number used as a multiplication factor.

### 5.8 The **STAT:** module

This module is used to compare the floating point information contained in two different LCM object. The calling specifications are:

Table 100: Structure (**STAT:**)

<b>STAT:</b> <i>NAME1</i> <i>NAME2</i> ;
--

*NAME1*            **character\*12** name of the reference LCM object.

*NAME2*            **character\*12** name of a compared LCM object.

## 5.9 The GREP: module

The GREP: module is used to extract a single value from a LCM object. The calling specifications are:

Table 101: Structure (**GREP:**)

```

GREP: NAME3 ::
[ EDIT iprint ]
[[ STEP { UP NOMDIR | AT index } ]]
[[ TYPE { BLOCK | index } >>itype<< ]]
[[ { GETVAL | MAXVAL | MINVAL | INDMAX | INDMIN | MEAN }
    { BLOCK | index } index1 [ { { index2 | * } [ index3 ] | NVAL { neval | * } } ]
    [[ >>value<< ]]
]] ;

```

<i>NAME3</i>	<b>character*12</b> name of the LCM object from which extractions will be performed.
<b>EDIT</b>	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index set to 1 to enable printing in this module. By default, <i>iprint</i> = 0.
<b>STEP</b>	keyword used to move in the LCM object hierarchy.
<b>UP</b>	keyword used to move up towards a sub-directory of the active directory.
<i>NOMDIR</i>	name of the sub-directory or heterogeneous list to which we wish to head.
<b>AT</b>	keyword used to move up towards a component in the heterogeneous list.
<i>index</i>	access the information located in the <i>index</i> -th component of the heterogeneous list.
<i>BLOCK</i>	name of the record which will be analyzed by the <b>GREP:</b> utility.
<i>index</i>	index of the record which will be analyzed by the <b>GREP:</b> utility.
<b>TYPE</b>	keyword used to get the LCM type of record <i>BLOCK</i> .
<i>itype</i>	type of block <i>BLOCK</i> or list component <i>index</i> (= 1: integer; = 2: real; = 3: character; = 4: double precision; = 5: logical; = 10: list; = 99: undefined).
<b>GETVAL</b>	keyword used to get values from an existing record. The receiving CLE-2000 variables are assumed to be of the same type as the picked values (all CLE-2000 types are supported).
<b>MAXVAL</b>	keyword used to get the maximum value of an existing record. The receiving CLE-2000 single variable is assumed to be of the same type as the picked maximum (valid for integer, real and double precision types).
<b>MINVAL</b>	keyword used to get the minimum value of an existing record. The receiving CLE-2000 single variable is assumed to be of the same type as the picked minimum (valid for integer, real and double precision types).
<b>INDMAX</b>	keyword used to get the index (position inside the block) of the maximum value of an existing record. The receiving CLE-2000 single variable is assumed of an integer type (valid for integer, real and double precision blocks).

INDMIN	keyword used to get the index (position inside the block) of the minimum value of an existing record. The receiving CLE-2000 single variable is assumed of an integer type (valid for integer, real and double precision blocks).
MEAN	keyword used to get the mean value of an existing record. The receiving CLE-2000 single variable is assumed to be of the same type as the computed mean (valid only for real and double precision types).
<i>index1</i>	the first element number in record <i>BLOCK</i> to be considered.
<i>index2</i>	the last element in record <i>BLOCK</i> to be considered. If <i>index2</i> is absent only element <i>index1</i> will be considered.
*	the search will extend to the last element in the record <i>BLOCK</i> .
<i>index3</i>	specifies the stride between values to be extracted between <i>index1</i> and <i>index2</i> . By default, a stride of 1 is assumed.
NVAL	keyword used to specify the number of elements to be extracted from the specified record.
<i>neval</i>	the number of elements to be extracted from the the specified record. If the record contains <b>character</b> information, elements <i>index1</i> to <i>index1+neval-1</i> are extracted.

The output parameters, denoted as  $\gg value \ll$ , are recovered as CLE-2000 variables in the module data located after the `:: keyword`.



## 5.10 The MSTR module

This module is used to create user-defined structures. In particular, it can be used to store and retrieve user variables in a structure or copy specific records from different structures to a single one so that the user can have an easy access to the information he wants from a CLE-2000 procedure. The calling specifications are:

Table 102: Structure (**MSTR:**)

```
[ STRUCT := ] MSTR: [ STRUCT ] [[ EXTSTR ]] ::
[ EDIT iprint ]
[ TYPE type ]
[[ CD [ilcm:]path ]]
[[ GET nbelem [ indexfirst [ increment ] ] [ilcm:]path]recname [[ >> VAR.IN << ]] ]]
[[ PUT nbelem [ indexfirst [ increment ] ] [ilcm:]path]recname [[ value ]] ]]
[[ CP nbelem [ indexfirst [ increment ] ] [ilcm1:]path1]recname1 [ilcm2:] [path2]recname2 ]]
;
```

<i>STRUCT</i>	<b>character*12</b> name of the user-defined LCM object in creation, modification or read-only mode depending on the requested actions.
<i>EXTSTR</i>	<b>character*12</b> name of existing LCM object from which information will be retrieved.
<i>EDIT</i>	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	index used to control the printing of this module.
<i>TYPE</i>	keyword used to modify the structure signature.
<i>type</i>	string containing the user-defined signature, limited to 12 characters.
<i>CD</i>	keyword for introducing a UNIX-like command to change the active directory of the structures.
<i>ilcm</i>	integer defining the structure index according to its position in the LHS or RHS list of parameters. By default, equal to 1 (i.e. <i>STRUCT</i> is affected by the <i>CD</i> command).
<i>path</i>	string containing the UNIX-like path (relative or absolute) of the directory to access. Note that if the directory does not exist and that the structure is in creation/modification mode, it is created.
<i>recname</i>	string containing the record name. Note that if this record does not exist and that the structure is in creation/modification mode, it is created.  For example, 2:/dir/rec refers to the record <i>rec</i> in the directory <i>dir</i> of the second structure in the calling specifications of the module.
<i>GET</i>	keyword for introducing the action of retrieving variables from the structure.
<i>PUT</i>	keyword for introducing the action of storing variables in the structure.
<i>CP</i>	keyword for introducing the UNIX-like action of copying some elements from one record (defined by <i>[ilcm1:]path1recname1</i> ) to another ( <i>[ilcm2:]path2recname2</i> ).
<i>nbelem</i>	integer defining the number of elements to store/retrieve/copy.
<i>indexfirst</i>	integer defining the index of the first element to store/retrieve/copy. By default, equal to 1.

<i>increment</i>	integer defining the stride in the record between the values to be stored/retrieved/copied. By default, equal to 1.
<i>VAR_IN</i>	<b>character*12</b> CLE-2000 variable name in which the extracted value will be placed. It is expected that the number of values extracted and the number (and types) of variables agree.
<i>value</i>	value to be stored. The first one defines the record type and all the values should be of the same type.

### 5.11 The FIND0: module

The FIND0: module is used to find the root of a function using the Brent's method. This procedure assumes that the zero is bracketed in an interval given in the input using the two first points, and that the function used is continuous in this interval. The calling specifications are:

Table 103: Structure (FIND0:)

```
L0 := FIND0: [ L0 ] ::
{ [ DEBUG ] [ ITMAX itmax ] [ TOL tol ] POINT X x1 Y y1 POINT X x2 Y y2 | Y y3 }
>>lFlag<< >>rRoot<< ;
```

<i>L0</i>	<b>character*12</b> names of the FIND0 LCM object (type L_0) that will contain all information necessary for the zero-finding procedure. If <i>L0</i> appears on both sides, it is updated; otherwise, it is created.
DEBUG	keyword used to edit the content of most variables in FIND0; used only for debugging purposes.
ITMAX	keyword used to specify the maximum number of iterations that will be allowed for the zero-finding procedure. The procedure will abort if the number of iterations goes beyond this maximum value.
<i>itmax</i>	the maximum number of iterations. Default value: 100.
TOL	keyword used to specify the tolerance on the zero to be found.
<i>tol</i>	tolerance. Default value: 1.E-5.
POINT	keyword used to specify that the next point will be given.
X	keyword used to specify that an abscissa will be given.
Y	keyword used to specify that an ordinate will be given.
<i>x1</i>	the first abscissa value.
<i>y1</i>	the first ordinate value.
<i>x2</i>	the second abscissa value.
<i>y2</i>	the second ordinate value.
<i>y3</i>	in the case we are in an update mode, only a new ordinate value is given.
<i>lFlag</i>	CLE-2000 logical variable in writable mode. The value returned is <i>true</i> if the new guessed root is within <i>tol</i> , <i>false</i> otherwise.
<i>rRoot</i>	CLE-2000 real variable in writable mode. The value returned is the last guess for the root.

Note that the zero-finding procedure has an initial mode where *NAME1*, *NAME2* and *NAME3* are created. In the initialization process, the two points specifying the interval must be given, and it is expected that  $y1 \times y2 < 0$ . In the updated mode, there is no need to put back the abscissa of the next point because it is expected to be the last real value that was generated by the procedure. This explains why you will only input Y *y3*.

The FIND0 specification is used to store intermediate values needed by the zero-finding procedure. There are no directories in this object, and it is created and updated only by the FIND0: module. To understand the content of the object, it is possible, using the labels given for every block, to refer to Brent's algorithm.<sup>[?]</sup>

### 5.12 The **ABORT:** module

This module is used to abort the overall calculation, calling the **XABORT()** subroutine from the Ganlib.

Table 104: Structure (**ABORT:**)

<b>ABORT:</b> ;
-----------------

### 5.13 The **END:** module

This module is used to delete all the memory-resident LCM objects, to close all the remaining local files and to return from a procedure or to stop the run. The calling specifications are:

Table 105: Structure (**END:**)

<b>END:</b> ;
---------------

## 6 THE MPI MODULES

DRAGON contains two modules that enables MPI capabilities. These modules are also available to any code built around the GAN generalized driver<sup>[?]</sup>, provided that it is properly compiled.

### 6.1 The DRVMPI: module

This module is a utility module related to MPI. It is mostly used to know the rank of the node running the current script. The calling specifications are:

Table 106: Structure (**DRVMPI:**)

```
[ NAME := ] DRVMPI: :: [ EDIT iprint ] [ WORLD-SIZE >>ncpu<< ] [ MY-ID >>rank<< ]
[[ SETLOOP { B0 | B1 } len >>beg<< >>end<< ]]
[ ALLREDUCE { SUM | PROD | MAX | MIN } operand >>result<< ]
[ TIME >>dTime<< ] [ BARRIER ] ;
```

<i>NAME</i>	<b>character*12</b> name of a dummy data structure to be possibly used as <i>NAME2</i> in the SNDMPI: module. It can be a linked list or an XSM file.
<b>EDIT</b>	keyword used to modify the print level <i>iprint</i> .
<i>iprint</i>	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.
<b>WORLD-SIZE</b>	keyword used to recover <i>ncpu</i> .
<i>ncpu</i>	total number of nodes in the MPI environment.
<b>MY-ID</b>	keyword used to recover <i>rank</i> .
<i>rank</i>	rank of the node that is running the script.
<b>SETLOOP</b>	keyword used to partition the set $\{0 \cdots len-1\}$ or $\{1 \cdots len\}$ equitably over the nodes. The result $\{beg \cdots end\}$ is different for each node.
<b>B0</b>	keyword used to set the initial position to 0.
<b>B1</b>	keyword used to set the initial position to 1.
<i>len</i>	set length.
<i>beg</i>	beginning of the set for the current script. Must be an integer variable.
<i>end</i>	end of the set for the current script. Must be an integer variable.
<b>ALLREDUCE</b>	keyword used to make a computation over all the <i>operand</i> and to store the result in <i>result</i> . Problems can be encountered in the 64-bits version.
<b>SUM</b>	keyword used to make <b>ALLREDUCE</b> perform a summation.
<b>PROD</b>	keyword used to make <b>ALLREDUCE</b> perform a multiplication.
<b>MAX</b>	keyword used to make <b>ALLREDUCE</b> find the maximum over all <i>operand</i> .
<b>MIN</b>	keyword used to make <b>ALLREDUCE</b> find the maximum over all <i>operand</i> .
<i>operand</i>	operand in the <b>ALLREDUCE</b> calculation. Cannot be a string or a logical value.

<i>result</i>	result of the <b>ALLREDUCE</b> calculation. Must be a variable of the same type as <i>operand</i> .
<b>TIME</b>	keyword used to recover <i>dTime</i> .
<i>dTime</i>	time in seconds since an arbitrary time in the past. Must be a double precision variable.
<b>BARRIER</b>	keyword used to stop the calculation until every node has reach this barrier.

*NAME* is always empty. What matters is that *NAME* is no more only declared, it now exists after the call of **DRVMPI:** module.

The output parameters, denoted as  $\gg value \ll$ , are recovered as CLE-2000 variables in the module data located after the **::** keyword.

## 6.2 The **SNDMPI:** module

This module is used to send or receive a linked list or an XSM file from one node to another one thanks to MPI. It is possible to send a linked list into an XSM file and vice versa. The module is blocked until the message is sent or received. The calling specifications are:

Table 107: Structure (**SNDMPI:**)

```
[ NAME1 := ] SNDMPI: [ NAME2 ] :: [ EDIT iPrint ] FROM iFrom TO { iTo | ALL }
[ ITEM from  $\gg to \ll$  ] ;
```

<i>NAME1</i>	<b>character*12</b> name of the data structure that will be received. It can be a linked list or an XSM file.
<i>NAME2</i>	<b>character*12</b> name of the data structure that will be sent. It can be a linked list or an XSM file. Since on the RHS, it has to exist even for receiving scripts. In this case it is recommended to create an empty data structure <i>NAME2</i> by calling the <b>DRVMPI:</b> module.
<b>EDIT</b>	keyword used to modify the print level <i>iPrint</i> .
<i>iPrint</i>	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.
<b>FROM</b>	keyword used to set <i>iFrom</i> .
<i>iFrom</i>	rank of the node from which <i>NAME2</i> has to be read.
<b>TO</b>	keyword used to set <i>iTo</i> .
<i>iTo</i>	rank of the node where <i>NAME1</i> has to be written.
<b>ALL</b>	keyword to make every node receive <i>NAME2</i> except the node <i>iFrom</i> .
<b>ITEM</b>	keyword used to send the value <i>from</i> to <i>to</i> either on <i>iTo</i> or <b>ALL</b> nodes. <i>NAME1</i> and <i>NAME2</i> are optional since they will be ignored.
<i>from</i>	value to send. Can be an integer, a real, a double or a logical.
<i>to</i>	variable used to receive <i>from</i> value. Must match <i>from</i> type.

## 7 EXAMPLES

We will now present a few examples of DRAGON input structures in such as to clarify and illustrate some of the options presented in Sections 3 and 5. These examples are *non-regression testcases* used to ensure that existing capabilities of Dragon are not lost with subsequent updates of the code. They make use of two assert procedures defined in CLE-2000 language, `assertS` and `assertV`, defined in Section ?? . These procedures are generally not used in computational schemes.

### 7.1 Scattering cross sections

In DRAGON, the angular dependence of the scattering cross section is expressed in a Legendre series expansion of the form:

$$\Sigma_s(\Omega \cdot \Omega') = \Sigma_s(\mu) = \sum_{l=0}^L \left( \frac{(2l+1)}{4\pi} \right) \Sigma_{s,l} P_l(\mu).$$

Since the Legendre polynomials satisfy the following orthogonality conditions:

$$\int_{-1}^1 d\mu P_l(\mu) P_m(\mu) = \left( \frac{2\delta_{l,m}}{(2l+1)} \right),$$

we will have

$$\Sigma_{s,l} = \int_{-1}^1 d\mu \int_0^{2\pi} d\varphi \Sigma_s(\mu) P_l(\mu) = 2\pi \int_{-1}^1 d\mu \Sigma_s(\mu) P_l(\mu).$$

Let us now consider the following three-group (*ngroup=3*) isotropic and linearly anisotropic scattering cross sections (*L=naniso=2*) given by:

<i>l</i>	<i>g</i>	$\Sigma_{s,l}^{g \rightarrow 1} \text{ (cm}^{-1}\text{)}$	$\Sigma_{s,l}^{g \rightarrow 2} \text{ (cm}^{-1}\text{)}$	$\Sigma_{s,l}^{g \rightarrow 3} \text{ (cm}^{-1}\text{)}$
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  (* L=0 *) 1 1 (* 3->1 *)      (* 2->1 *)      (* 1->1 *) 0.90
          3 3 (* 3->2 *) 0.30 (* 2->2 *) 0.70 (* 1->2 *) 0.80
          2 3 (* 3->3 *) 0.40 (* 2->3 *) 0.60 (* 1->3 *)
SCAT  (* L=1 *) 3 3 (* 3->1 *) 0.03 (* 2->1 *) 0.00 (* 1->1 *) 0.09
          2 2 (* 3->2 *)      (* 2->2 *) 0.07 (* 1->2 *) 0.05
          3 3 (* 3->3 *) 0.04 (* 2->3 *) 0.06 (* 1->3 *) 0.08
```

### 7.2 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 23):

This geometry can be analyzed using a 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 ;
```





Figure 23: Slab geometry with mesh-splitting

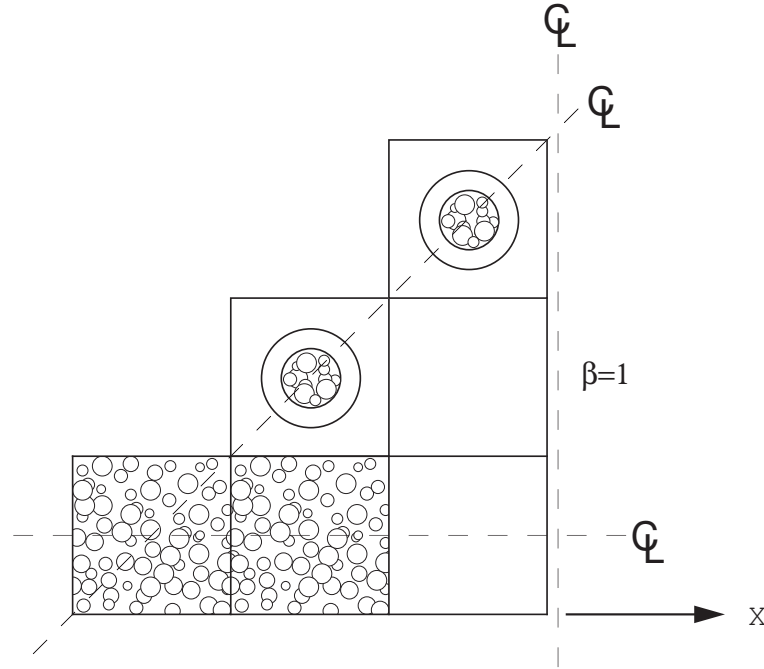


Figure 24: Two-dimensional Cartesian assembly containing micro-structures

- 2-D Cartesian geometry containing micro-structures (see figure Figure 24):

This geometry can be analyzed only using SYBILT: tracking modules:

```
CARNSG := GEO: :: CAR2D 3 3
X- DIAG X+ REFL Y- SYME Y+ DIAG
MIX  C1  C1  C2
      C3  C2
      C3

BIHET SPHE (*NG=*) 2 (* NMILG= *) 2 (* SPHERICAL MICRO-STRUCTURE *)
(* NS= *) 3 3
(* M-S-1 *) 0.0 0.1 0.2 0.3 (* M-S 2 *) 0.0 0.2 0.4 0.5
(* COMPOSITE MIXTURES *) 4 5
(* MIXTURES SURROUNDING M-S *) 1 1
(* COMPOSITE MIXTURE 4 FRACT *) 0.4 0.0
(* REAL MIXTURE CONTENT M-S-1 *) 3 1 3
(* COMPOSITE MIXTURE 5 FRACT *) 0.2 0.1
(* REAL MIXTURE CONTENT M-S-1 *) 1 2 1
(* REAL MIXTURE CONTENT M-S-2 *) 2 3 1
:: C1 := GEO: CAR2D 1 1 (* HOMOGENEOUS CELL WITH M-S *)
  MESHX 0.0 1.45 MESHY 0.0 1.45 MIX 4 ;
:: C2 := GEO: C1 (* HOMOGENEOUS CELL WITHOUT M-S *)
  MIX 1 ;
:: C3 := GEO: CARCEL 2 (* CELL WITH M-S TUBE *)
```

```

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 25):

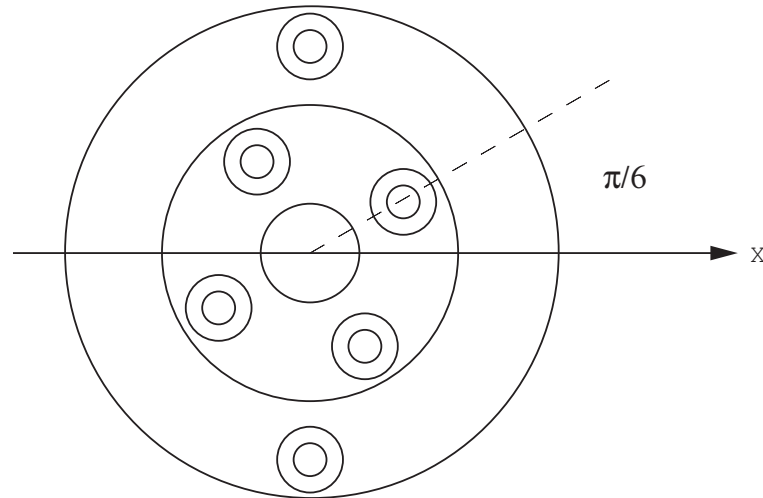


Figure 25: Cylindrical cluster geometry

The first two geometries, namely **ANNPIN** and **CARPIN** can be analyzed using a **EXCELT**: tracking modules since the pins in the clusters are all located between annular region. For the last two geometries, **ANNSPIN** and **CARSPIN**, which are based on **ANNPIN** and **CARPIN** respectively, they only be treated by the **EXCELT**: tracking modules since the pins in the clusters intersect the annular regions defined by the **SPLITR** option. This later option which was selected to ensure a uniform thickness of 0.25 cm for each the annular region in the final geometries.

```

ANNPIN := GEO: :: TUBE 3
R+ REFL RADIUS 0.0 0.75 2.75 4.75
MIX 2 1 3
CLUSTER C1 C2
::: C1 := GEO: TUBE 2
MIX 2 4 RADIUS 0.0 0.3 0.6
NPIN 4 RPIN 1.75 APIN 0.523599 ;
::: C2 := GEO: C1
NPIN 2 RPIN 3.75 APIN 1.570796 ;
;
CARPIN := GEO: :: CARCEL 3
X- REFL X+ REFL Y- REFL Y+ REFL
MESHX 0.0 10.0 MESHY -5.0 5.0
RADIUS 0.0 0.75 2.75 4.75
MIX 2 1 3 3
CLUSTER C1 C2
::: C1 := GEO: TUBE 2
MIX 2 4 RADIUS 0.0 0.3 0.6
NPIN 4 RPIN 1.75 APIN 0.523599 ;
::: C2 := GEO: C1
NPIN 2 RPIN 3.75 APIN 1.570796 ;
;
ANNSPIN := GEO: ANNPIN ::

```

```

SPLITR  3 8 8 ;
CARSPIN := GEO: CARPIN ::
SPLITR  3 8 8 ;

```

Note that even if MESHX and MESHY differ in CARPIN, the annular regions and pins will still be localized with respect to the center of the cell located at  $(x, y) = (5.0, 0.0)$  cm.

- 2-D hexagonal geometry (see Figure 26):

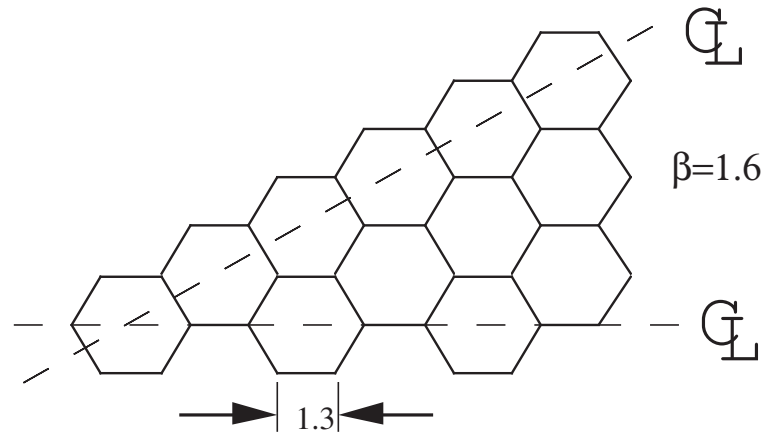


Figure 26: Two-dimensional hexagonal geometry

This geometry can be analyzed using the 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 27):

This geometry can only be analyzed using the EXCELT: tracking modules:

```

SUPERCCELL := GEO: :: CAR3D 4 4 3
X- REFL  X+ REFL
Y- REFL  Y+ REFL
Z- REFL  Z+ REFL
MIX  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 ;

```

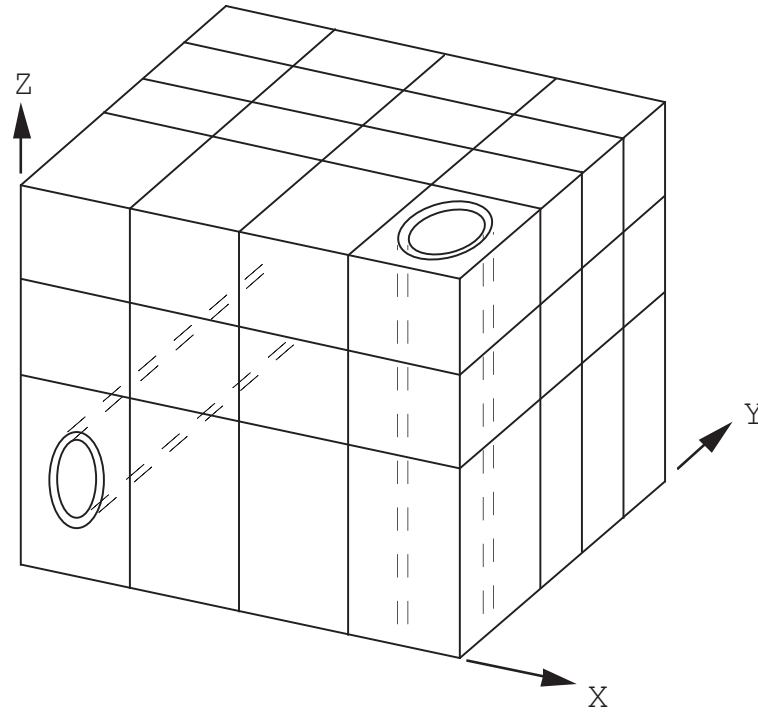


Figure 27: Three-dimensional Cartesian super-cell

```

::: 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 28).

Here we are considering an infinite lattice having two types of cells such that

$$\begin{pmatrix} \text{pource}(1) \\ \text{pource}(2) \end{pmatrix} = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \text{procel}(1,1) & \text{procel}(1,2) \\ \text{procel}(2,1) & \text{procel}(2,2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1/2 & 1/2 \end{pmatrix}$$

This lattice, can be represented either in a *do-it-yourself* type geometry (HEXDIY) or directly (HEXDIR):

```

HEXDIY := GEO: :: GROUP 2
POURCE 0.3333333 0.6666667
PROCEL 0.0      1.0
        0.5      0.5
MIX     C1 C2
::: C1 := GEO: TUBE 1
  RADIUS 0.0 1.1822093 MIX 1 ;

```

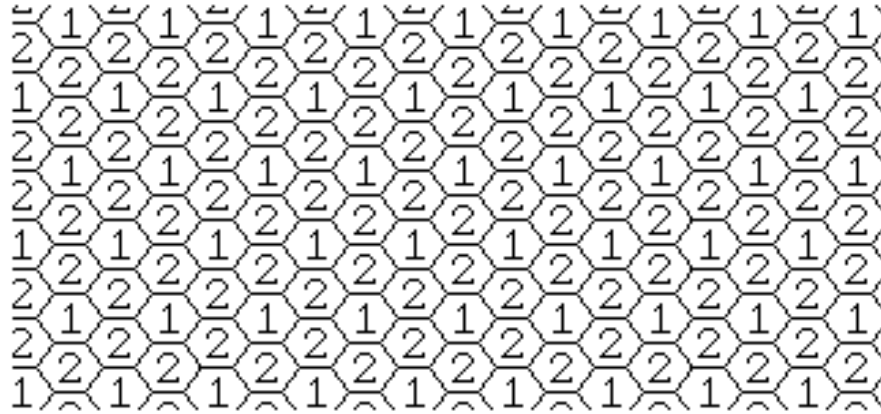


Figure 28: Hexagonal multicell lattice geometry

```

::: 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 the **SYBILT**: tracking module, while the second lattice can be analyzed using all the tracking modules of **DRAGON**.

### 7.3 MATXS7A microscopic cross-section examples

The test cases we will consider here use the **LIB:** module to specify that the cross sections will be taken from a MATXS7A 69 groups microscopic cross-sections library. We will assume that this library is located in file **MATXS7A**.

#### 7.3.1 (TCXA01) – The Mosteller benchmark.

The typical input data required to analyze this benchmark<sup>[?]</sup> with DRAGON is of the form:

The input deck begins with declarations for the linked lists and the interface files and the various modules used for this DRAGON execution. Any word not declared is considered as a keyword.

The **LIB:** module is used to interpolate the microscopic cross sections in absolute temperature and dilution and to produce group-ordered macroscopic cross sections. We use the MATXS format 69 groups microscopic cross section library named '**MATXS7A**'.<sup>[?]</sup> Each mixture at a given absolute temperature (in Kelvin) is defined in terms of MATXS isotope names (**U235**, **U238**, **O16**, etc.). In this case, the number density (in  $10^{24}$  particules per cubic centimeter) for each isotope is provided. Resonant region indices and the type of thermal scattering approximation used with the 42 thermal groups (free gas or H<sub>2</sub>O molecular model) is also specified. Only MATXS type libraries require the thermalization model to be set.

The **GEO:** module is used to define the geometry. Here two types of geometry are considered, **MOSTELA** a 1-D annular geometry and **MOSTELC** a 2-D Cartesian geometry. These geometries are defined before knowing the type of discretization or numerical treatment that will follow. For **MOSTELA** the first line indicates that the geometry has circular boundaries and that it contains three concentric annular subregions. The boundary conditions (reflection), the annular radii and the mixture index corresponding to each region of the cell are given successively. For **MOSTELC** the first line indicates that this geometry has 2-D Cartesian boundaries containing three subregions, two of which are annular. The boundary conditions (reflection on each side), the annular radii, the external side widths and the mixture index corresponding to each region of the cell are given successively.

Four cases are then considered. First we will analyse the annular geometry using the **SYBILT:** module for flux calculation. The **DISCR** and **ddstracking** structures are thereby generated. The **SHI:** module uses microscopic cross section data contained in the **LIBRARY** and tracking information contained in '**DISCR**' and '**TRACKS**' in order to compute the actual dilution of each resonant isotope (**U235** and **U238**) and to perform a new interpolation in the MATXS file. Dilutions are only computed for the energy groups with resonance data present on the library; the other groups are assumed to stay at infinite dilution.

For the second case we will analyse the Cartesian geometry using the again the **SYBILT:** tracking module for self shielding calculations and the **SYBILT:** module for flux calculation. The **DISCR** and **TRACKING** structures are thereby generated.

Four cases are then considered. First we will analyse the annular geometry using the **SYBILT:** tracking module allows the geometry named '**MOSTEL**' to be discretized by the full CP tracking algorithm. A new tracking file (sequential binary) is created and named '**TRACKS**', together with a **TRACKING1** structure named '**DISCR**'. A periodic tracking (with 12 angles and 20.0 tracks per cm) is considered here.

The **ASM:** module uses macroscopic cross section data contained in the embedded **MACROLIB** of '**LIBRARY**' and tracking information contained in '**DISCR**' and '**TRACKS**' in order to compute the reduced and scattering modified collision probability matrices for each of the 69 energy groups. We have not used the important capability of DRAGON to use a different tracking to perform self-shielding and flux calculations.

The **FLU:** module uses macroscopic cross section data contained in '**LIBRARY**' (recovered from the dependency tree) and CPs contained in '**CP**' in order to compute the neutron flux for each of the 69 energy groups. The transport equation is solved for the effective multiplication factor without buckling or leakage model.

Next, the **EDI:** module performs spatial homogenization (the cross sections are smeared over the complete cell) and coarse energy group condensation. The first coarse energy group contains the micro-groups 1 to 27; the second coarse energy group contains the remaining micro-groups.