

**TECHNICAL REPORT  
IGE-260**



**New Geometries Processing in DRAGON:  
The NXT: Module**

G. MARLEAU

Institut de génie nucléaire  
Département de génie physique  
École Polytechnique de Montréal  
December 2005

## CONTENTS

CONTENTS . . . . .	ii
LIST OF TABLES . . . . .	iii
LIST OF FIGURES . . . . .	iv
1 INTRODUCTION . . . . .	1
2 GEOMETRY CONSIDERATIONS . . . . .	2
2.1 Assembly level . . . . .	4
2.2 Cell level . . . . .	8
2.3 Pin level . . . . .	9
3 GEOMETRY ANALYSIS . . . . .	10
3.1 Region and surface identification for pure geometries . . . . .	10
3.1.1 Pure Cartesian cells . . . . .	10
3.1.2 Pure annular and cylindrical cells . . . . .	12
3.1.3 Cartesian cell with annular subregions . . . . .	16
3.1.4 Annular cell with Cartesian subregions . . . . .	21
3.2 Pins in cells . . . . .	26
3.3 Cells in assemblies . . . . .	27
4 TRACKING IN THE NXT MODULE OF DRAGON . . . . .	29
4.1 Region Volume and Surface Area Evaluation . . . . .	29
4.1.1 Cartesian Region . . . . .	29
4.1.2 Cylindrical Region . . . . .	30
4.1.3 Mixed Cartesian/Cylindrical Region . . . . .	30
4.2 Geometry Description . . . . .	30
4.3 Volume Evaluation . . . . .	33
4.4 Generation of the Integration Lines . . . . .	35
4.4.1 Cartesian Geometry . . . . .	35
4.4.2 Cylindrical Geometry . . . . .	36
REFERENCES . . . . .	39
APPENDICES . . . . .	40
A ANALYTICAL VOLUME EVALUATION . . . . .	41
A.1 Rectangle . . . . .	41
A.2 Concentric Circles . . . . .	41
A.3 Overlapping Circles . . . . .	42
A.4 Circles Overlapping a Rectangle . . . . .	44
B PROGRAMMING GUIDE FOR THE NXT MODULE . . . . .	47
B.1 Structure of the program . . . . .	47
B.2 Use of each routine . . . . .	49
C CONTENTS OF THE DRAGON BINARY TRACKING FILE . . . . .	52
D CONTENTS OF A /TRACKING/ DIRECTORY . . . . .	55
D.1 The main directory . . . . .	55
D.2 The EXCELL records and sub-directories . . . . .	56

**LIST OF TABLES**

1	Main records in /TRACKING/ . . . . .	55
2	Additional records for EXCELL tracking . . . . .	58
3	Global geometry records in /NXTRecords/ . . . . .	59
4	Cell $i = 1$ records in /NXTRecords/ . . . . .	63
5	Pin $i = 1$ records in /NXTRecords/ . . . . .	66
6	Global geometry records in /NXTRecords/ . . . . .	68

## LIST OF FIGURES

1	Example of a multilevel 2-D geometry. . . . .	2
2	Global 2-D geometry. . . . .	2
3	Cells inserted in global geometry. . . . .	3
4	Pins superimposed on cells . . . . .	3
5	Geometry with 2-D (left) and 3-D (right) rectangular boundaries. . . . .	3
6	Central cell and diagonal symmetries . . . . .	4
7	2-D (left) and 3-D (right) geometries that can be processed by the <code>NXT:</code> module. . . . .	5
8	Example of a 2-D geometry with diagonal and mirror symmetry . . . . .	7
9	Unfolded assembly with cell contents and <code>TURN</code> option . . . . .	8
10	Example of pure Cartesian cells in 2-D (left) and 3-D (right). . . . .	11
11	Examples of annular geometry . . . . .	14
12	Examples of Cartesian geometry with annular subregions . . . . .	18
13	Examples of annular geometry with Cartesian subregions . . . . .	24
14	Annular pins in <code>CARCEL</code> geometries. . . . .	27
15	<i>X</i> directed cylindrical pins in <code>CARCELX</code> geometries. . . . .	28
16	Numbering a multiregion geometry in <code>DRAGON</code> . . . . .	31
17	Volume of a rectangle . . . . .	41
18	Volume of concentric annular circles . . . . .	42
19	Volume of overlapping circles . . . . .	42
20	Translation and rotation of overlapping circles . . . . .	43
21	Volume for the overlap of a rectangle with an annular region . . . . .	45
22	Decomposition of a rectangle circle region overlap . . . . .	45
23	Volume for the overlap of a rectangle with two non concentric annular regions . . . . .	46
24	Example of an assembly of hexagons (left) and external faces identification for an hexagon . . . . .	61

## 1 INTRODUCTION

The tracking modules of DRAGON are used to perform two successive tasks, namely:<sup>[1-4]</sup>

1. to analyze the geometry provided and generate the information required for the tracking procedure to take place;
2. to generate, for a specific geometry, the integration lines resulting from a specific numerical quadrature.

The result of the first task is the TRACKING data structure where one can find the records that are essential for all the other calculations in DRAGON involving the geometry, including the regional volumes, the mixture associated with each region, the flux index cross referencing the elements in the flux/current array with elements in the volume array and the boundary conditions in the form of an albedo array. Depending on the type of geometry and on the specific tracking module considered, one can also find in this data structure additional records that can be used to rebuilt the geometry or associate a graphical image with the geometry. The result of the second task is the creation of the tracking file that contains all the information required to process the numerical quadrature specified in the tracking module.<sup>[3]</sup> The format of this tracking file does not depend on the tracking module selected even if the explicit contents of the file does. For example, the NXT: tracking file generated for a given geometry will in general differ from the information generated using the EXCELT: module for the same geometry.

The NXT: tracking module retains most of the properties of the tracking procedures already implemented in DRAGON, namely it contains a geometry analysis procedure as well as an integration line generation procedure. However, these procedures are programmed in such a way that they can be called by other modules of DRAGON, therefore making DRAGON more modular. The NXT: module has also been developed using a structure parallel to that used in the EXCELT: module (justifying the name NXT for New Excell Tracking). The main difference is the rationalisation of the information stored in the TRACKING data structure. The new records in the TRACKING data structure associated with the NXT: module are defined in such a way that they can accomodate new types of geometry as well as additional geometry levels, thereby simplifying the process of code maintenance and updating. In this report we will describe the NXT: tracking procedure which, in the long term, could replace the EXCELT: tracking procedure that has been in use for the last 20 years.

The primary goal of this report is to describe the NXT: tracking procedure. This will include:

- the theory manual for the NXT: module.
- the programmer's guide for the NXT: module;
- the user guide for the NXT: module;
- a description of the TRACKING data structure generated by NXT: including a presentation of the contents of the /NewExcellTrk/ subdirectory included in this data structure;

## 2 GEOMETRY CONSIDERATIONS

The `NXT:` module, while being quite general, can only process a limited number of geometry out of the full set that one can define using the `DRAGON GEO:` module. The main constraints on the geometries that can be currently analyzed by this module are the following :

1. A maximum of three sub-levels of geometry will be processed (see Figure 1 for a 2-D exemple of such a geometry construction). The levels are classified using the following hierarchy by the `NXT:` tracking module:

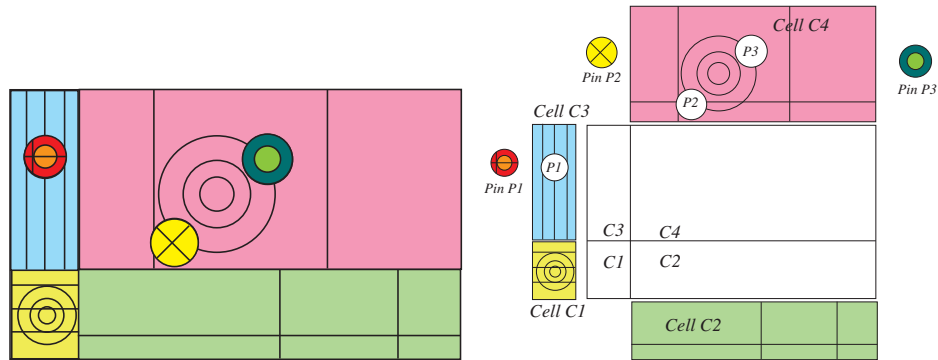


Figure 1: Example of a multilevel 2-D geometry. Explicit geometry (left) and description in terms of cells and pins (right).

- (a) Main assembly level that will be filled with cell and to which are associated the boundary conditions (see Figure 2).

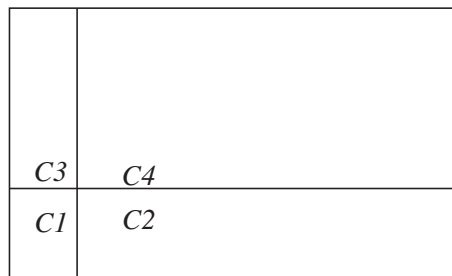


Figure 2: Global 2-D geometry.

- (b) Intermediate level defining the cells that will be used to fill the assembly (see Figure 3). In the current version of `NXT:` the cells used to define these assemblies cannot themselves contain cell sub-assemblies.
- (c) Optional upper level that corresponds to the cell geometries added using the `CLUSTER` keyword (see Figure 3). The cell geometry on this level covers the region defined by lower level geometries.

In the case where the global geometry is not an assembly (a single cell) it is automatically generated by `NXT:` based on the cell properties.

2. Limited to 2-D and 3-D geometries having a rectangular boundary (see Figure 5). This means that each geometry in 2-D must be located between 2 lines parallel to the  $y$  axis and two lines parallel to the  $x$  axis.

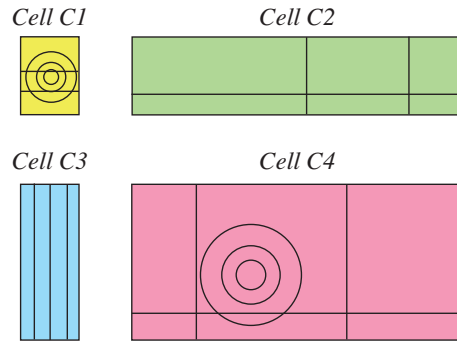


Figure 3: Cells inserted in global geometry.

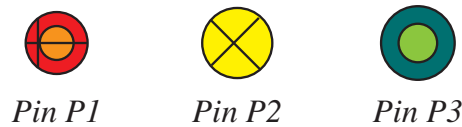


Figure 4: Pins superimposed on cells

In 3-D, the geometry must be located between 2 plane normal to the  $x$  axis, 2 plane normal to the  $y$  axis and finally 2 plane normal to the  $z$  axis .

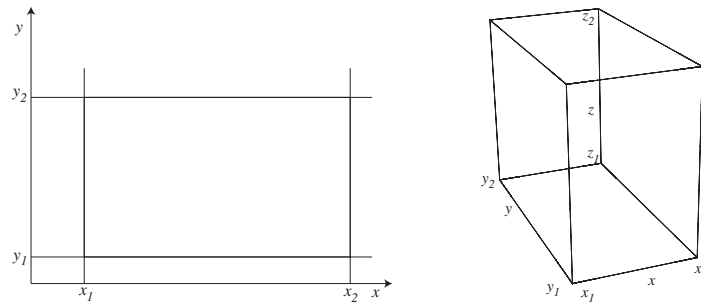


Figure 5: Geometry with 2-D (left) and 3-D (right) rectangular boundaries.

3. The boundary conditions are applied on the external faces of the geometry with three exceptions:
  - SYME, a mirror symmetry applied at the center of the cell (single cell geometry) or at the center of the cells closest to the direction specified for an assembly geometry (see Figure 6).
  - SSME, a mirror symmetry applied at the boundary of the cell or the assembly.
  - DIAG which is applied on a  $x = y$  diagonal passing through the center of the cell or the assembly (see Figure 6). The combination X- and Y+ means that the regions under and to the right of the diagonal are tracked while for the combination X+ and Y- the regions over and to the left of the diagonal are tracked.

The external surfaces associated with a geometry are also generated assuming that the pin surfaces associated with an external boundary cover completely the external surfaces associated with the cell geometries.

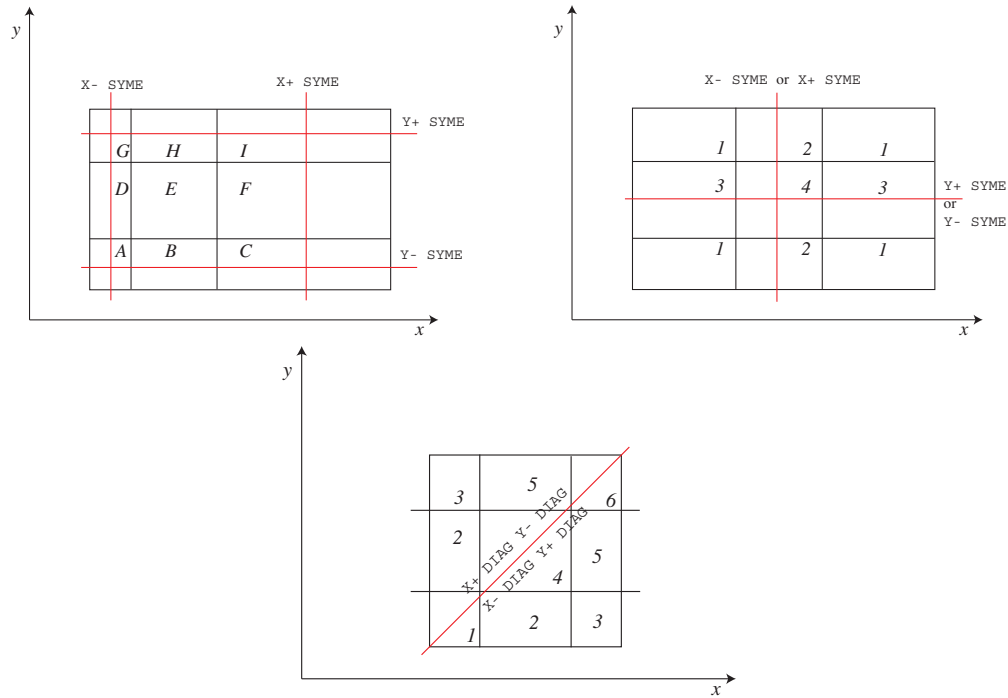


Figure 6: Central cell and diagonal symmetries. Symmetry SYME as applied for assembly of cells (top left) or for individual cells (top right) symmetry DIAG for square cell or assembly (bottom).

Similarly, the cell surfaces associated with an external boundary are superimposed over the assembly surfaces.

Typical 2-D and 3-D geometries that can be processed by the NXT: module are illustrated in Figure 7. For the 2-D geometry, each color is associated with a different region number. One can immediately see the hierarchy of region numbering with namely the pins hiding the cells hiding the assembly level. This hierarchy is also used for 3-D geometry both for the region and external surface numbering. However, the cells and pins surfaces not in contact with an external surface are never considered.

### 2.1 Assembly level

As already mentioned, at the main assembly level the geometry must have a purely Cartesian structure. This level can be built directly by the user using the CAR2D and CAR3D geometry types and filling each region in the geometry with cells as proposed in the following 2-D and 3-D examples

```
Geo2DA := GEO: :: CAR2D 1 2
CELL C1 C2
X- REFL X+ REFL Y- REFL Y+ REFL
::: C1 := GEO: CAR2D 2 1
MESHX <<X1>> <<X2>> <<X3>> MESHY <<Y1>> <<Y2>>
MIX 1 2 ;
::: C2 := GEO: CARCEL 2 1 1
MESHX <<X1>> <<X3>> MESHY <<Y2>> <<Y3>>
RADIUS 0.0 <<R1>> <<R2>>
MIX 3 4 5 ;
;
```



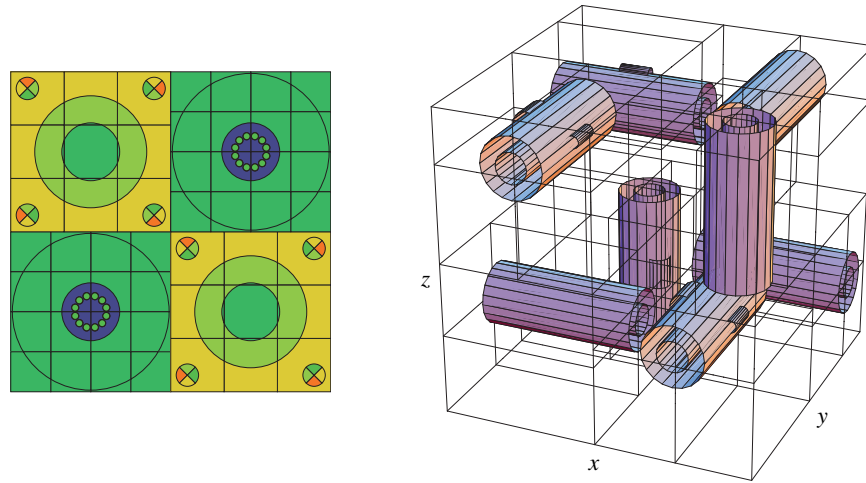


Figure 7: 2-D (left) and 3-D (right) geometries that can be processed by the NXT: module.

```

Geo3DA := GEO: :: CAR3D 2 2 2
CELL C1 C2 C3 C4 C5 C6 C7 C8
X- REFL X+ REFL Y- REFL Y+ REFL Z- REFL Z+ REFL
::: C1 := GEO: CAR3D 1 1 1
    MESHX <<X1>> <<X2>> MESHY <<Y1>> <<Y2>> MESHZ <<Z1>> <<Z2>>
    MIX 1 ;
::: C2 := GEO: CARCELZ 1 1 1 1
    MESHX <<X2>> <<X3>> MESHY <<Y1>> <<Y2>> MESHZ <<Z1>> <<Z2>>
    RADIUS 0.0 <<R1>>
    MIX 12 2 ;
[ ... ]
;

```

The exact dimensions of the assembly are extracted from the dimensions of the individual cells, which must be defined in such a way as to form a uniform Cartesian mesh in the  $X$ ,  $Y$  and  $Z$  directions.

In the case where the assembly is made up of a single cell, as in

```

Geo2DC := GEO: :: CARCEL 2 1 1
X- REFL X+ REFL Y- REFL Y+ REFL
MESHX <<X1>> <<X3>> MESHY <<Y2>> <<Y3>>
RADIUS 0.0 <<R1>> <<R2>>
MIX 3 4 5
;
Geo3DC := GEO: :: CAR3D 2 2 2
X- REFL X+ REFL Y- REFL Y+ REFL Z- REFL Z+ REFL
MESHX <<X1>> <<X2>> <<X3>>
MESHY <<Y1>> <<Y2>> <<Y3>>
MESHZ <<Z1>> <<Z2>> <<Z3>>
MIX 1 2 3 4 5 6 7 8
;

```

the NXT: module will automatically generate, using the information associated with the cell, the required CAR2D or CAR3D assembly that will be filled with the cell specified. For the 2-D and 3-D cases above, NXT: will work as if it was seeing the following equivalent geometries:

```

Geo2DCR := GEO: :: CAR2D 1 1
X- REFL X+ REFL Y- REFL Y+ REFL
CELL Geo2DC
::: Geo2DC := GEO: CARCEL 2 1 1
MESHX <<X1>> <<X3>> MESHY <<Y2>> <<Y3>>
RADIUS 0.0 <<R1>> <<R2>>
MIX 3 4 5 ;
;
Geo3DCR := GEO: :: CAR3D 1 1 1
X- REFL X+ REFL Y- REFL Y+ REFL Z- REFL Z+ REFL
CELL Geo3DC
::: Geo3DC := GEO: CAR3D 2 2 2
MESHX <<X1>> <<X2>> <<X3>>
MESHY <<Y1>> <<Y2>> <<Y3>>
MESHZ <<Z1>> <<Z2>> <<Z3>>
MIX 1 2 3 4 5 6 7 8 ;
;

```

A consequence of this reconstruction feature of NXT: is that in this case the cell geometry options are limited to CAR2D or CARCEL in 2-D and to CAR3D, CARCELX, CARCELY or CARCELZ in 3-D.

This procedure is straightforward for the cases where only the REFL, VOID and ALBE boundary conditions are used. If the first level geometry is a cell, the reconstruction process described above remains valid and the symmetry is applied directly to the cell of interest. On the other hand, if the first main level geometry is an assembly, two problem arise when the SYME, SYMM, and DIAG boundary are used:

1. These symmetries are used to simplify the input file and the GEOMETRY data structure thereby created. This means that the GEOMETRY data structure does not explicitly represent the geometry that will be treated.
2. The final geometry provided in the GEOMETRY data structure does not necessarily have the purely Cartesian mesh required by the NXT: module (primarily when diagonal boundary conditions are selected but also for the SYME geometry which may be cutting cylindrical regions in half).

In order to illustrate this problem let us consider the 2-D examples of Figure 8 generated using the following instructions in DRAGON:

```

Geo2DADSY := GEO: :: CAR2D 2 2
CELL C1 C2 C3
X- DIAG X+ REFL Y- SYME Y+ DIAG
::: C1 := GEO: CAR2D 1 1
MESHX <<X1>> <<X2>> MESHY <<X1>> <<X2>>
MIX 1 ;
::: C2 := GEO: CARCEL 2 1 1
MESHX <<X1>> <<X3>> MESHY <<X1>> <<X2>>
RADIUS 0.0 <<R1>> <<R2>>
MIX 3 4 5 ;
::: C3 := GEO: CARCEL 2 2 2
MESHX <<X1>> <<X2>> <<X3>> MESHY <<X1>> <<X2>> <<X3>>
RADIUS 0.0 <<R1>> <<R2>>
MIX 6 7 8 9 10 11 9 10 11 12 13 14 ;
;
Geo2DADSS := GEO: :: CAR2D 2 2
CELL C1 C2 C3
X- DIAG X+ REFL Y- SYME Y+ DIAG
::: C1 := GEO: CAR2D 1 1

```

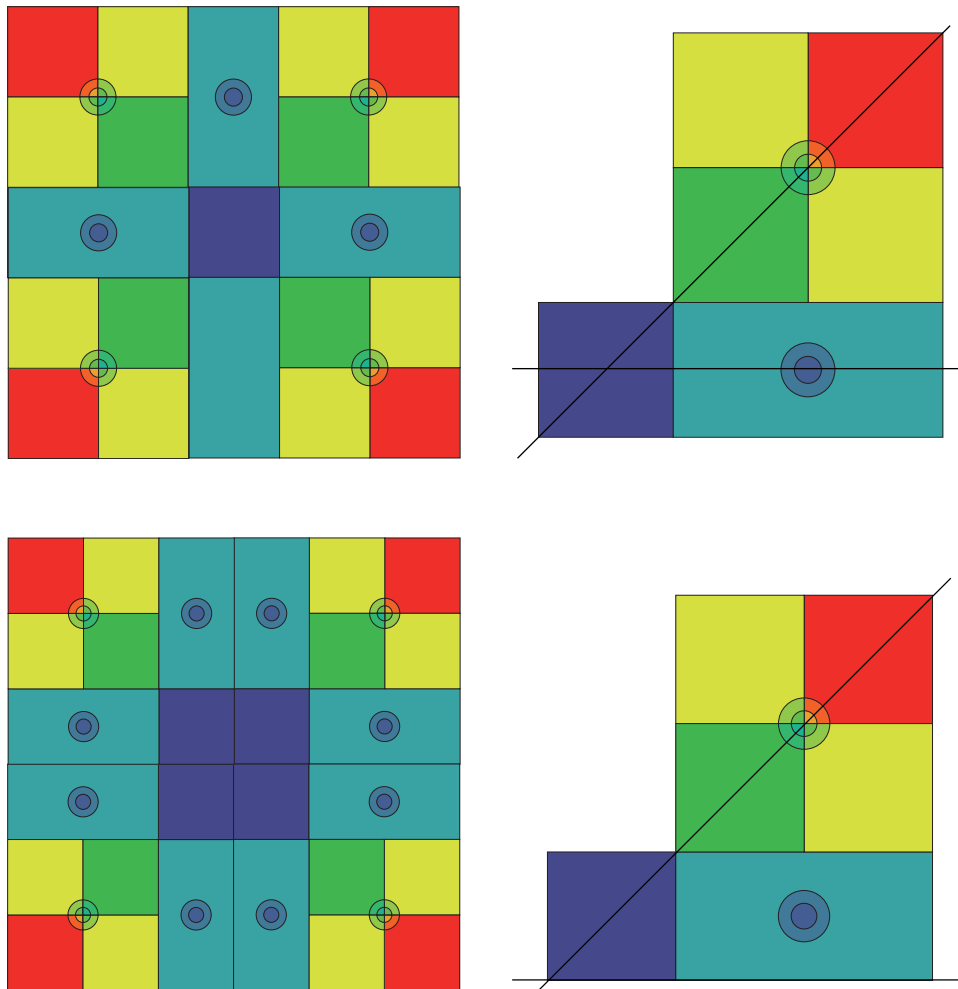


Figure 8: Example of a 2-D geometry with diagonal and mirror symmetry. Here the mirror symmetry are taken with respect to the  $X$  and  $Y$  axis passing through the center (top figures with SYME symmetry) or located at the bottom (bottom figure with SSYM symmetry) of the geometry. The figure on the left is the exact geometry and that to the right is the geometry provided as input to DRAGON.

(C3,E)	(C2,F)	(C3,A)	
(C2,E)	(C1,A)	(C2,A)	
(C3,C)	(C2,B)	(C3,G)	

(C3,E)	(C2,D)	(C2,F)	(C3,A)
(C2,E)	(C1,E)	(C1,A)	(C2,A)
(C2,C)	(C1,A)	(C1,G)	(C2,G)
(C3,C)	(C2,H)	(C2,B)	(C3,G)

Figure 9: Unfolded assembly with cell contents and TURN option for Geo2DADSY (left) and Geo2DADSS (right).

```

MESHX <<X1>> <<X2>> MESHY <<X1>> <<X2>>
MIX 1 ;
::: C2 := GEO: CARCEL 2 1 1
MESHX <<X1>> <<X3>> MESHY <<X1>> <<X2>>
RADIUS 0.0 <<R1>> <<R2>>
MIX 3 4 5 ;
::: C3 := GEO: CARCEL 2 2 2
MESHX <<X1>> <<X2>> <<X3>> MESHY <<X1>> <<X2>> <<X3>>
RADIUS 0.0 <<R1>> <<R2>>
MIX 6 7 8 9 10 11 9 10 11 12 13 14 ;
i

```

For such geometries, the NXT: module first unfolds the geometry according to the instruction, generating a  $3 \times 3$  assembly for geometry Geo2DADSY and a  $4 \times 4$  assembly for geometry Geo2DADSS and then fills these assemblies with the adequately rotated and reflected cells (see Figure 9) where the rotation indices (A to H) are defined in the DRAGON users manual.<sup>[2]</sup> It also test if each cells possesses the intrinsic symmetry (cell C1 for example) required by the geometry description.

## 2.2 Cell level

Two types of cells can be used to fill 2-D assemblies:

- CAR2D, that describes a rectangular cell.
- CARCEL, a rectangular cell that contains an embedded set of concentric annular regions. The center of these annular region coincides with the center of the cell unless specified otherwise (keyword OFFCENTER). In addition, the annular regions must all be located within the boundary of the rectangle.

The explicit rectangular submesh is provided using the MESHX and MESHY keywords for both the CAR2D and CARCEL. In addition the explicit radial submesh for a CARCEL geometry is specified using the RADIUS keyword. Implicit submeshing is also possible at the tracking level with the use of the keywords SPLITX (*X* submesh), SPLITY (*Y* submesh) and SPLITR (radial submesh).

The four types of cells that can be used to fill 3-D assemblies are:

- CAR3D, that describes a rectangular parallelepiped.

- CARCELX, that describes a rectangular parallelepiped that contains an embedded set of concentric cylindrical regions parallel to the  $X$  axis that extend over the full  $X$  range of the cell.
- CARCELY, that describes a rectangular parallelepiped that contains an embedded set of concentric cylindrical regions parallel to the  $Y$  axis that extend over the full  $Y$  range of the cell.
- CARCELZ, that describes a rectangular parallelepiped that contains an embedded set of concentric cylindrical regions parallel to the  $Z$  axis that extend over the full  $Z$  range of the cell.

The explicit rectangular submesh is provided using the MESHX, MESHY and MESHZ keywords for both these geometries while the radial submesh for the last three geometries is specified using the RADIUS keyword. As for 2-D cells, implicit submeshing is also possible at the tracking level with the use of the keywords SPLITX ( $X$  submesh), SPLITY ( $Y$  submesh), SPLITZ ( $Z$  submesh) and SPLITR (radial submesh).

### 2.3 Pin level

The only of pin geometry permitted for 2-D cells is a TUBE that describes a cell with circular boundaries. In 3-D, TUBEX, TUBEY or TUBEZ pin geometry that represents cylindrical cells in the  $X$ ,  $Y$  and  $Z$  directions are generally permitted with the following restrictions:

- The direction of all the pins in a cell must be identical.
- The cylindrical regions associated with a cell, if any, must be identical to that of the pins that are inserted in this cell.

Finally all the pins (both in 2-D and 3-D) must be fully enclosed in the cell.

The radial mesh associated with a tube and specified using the RADIUS can be refined implicitly at the tracking level using the SPLITR option. Similarly cylinders in the  $X$ ,  $Y$  and  $Z$  directions will be defined using the (MESHX, SPLITX), (MESHY, SPLITY) and (MESHZ, SPLITZ) pairs of keywords. Note that is also possible to superimposed a Cartesian mesh on the radial mesh using the following options

- MESHX, SPLITX, MESHY and SPLITY for TUBE geometries.
- MESHY, SPLITY, MESHZ and SPLITZ for TUBEX geometries.
- MESHZ, SPLITZ, MESHX and SPLITX for TUBEY geometries.
- MESHX, SPLITX, MESHY and SPLITY for TUBEZ geometries.

### 3 GEOMETRY ANALYSIS

Here, we will describe successively how DRAGON identifies the region and surfaces associated with a multilevel geometry (an assembly) similar to those described in Section 2. One may recall that such assemblies are always based the superposition of geometries containing cells (keyword `CELL`), pin clusters (keyword `PIN`) or mixtures (keyword `MIX`). Here we will consider the case of first pure geometries (Section 3.1) which are defined as a geometry that is filled only with mixtures. These can be defined using the following keywords:

- `CAR2D` for a 2-D Cartesian cell.
- `CAR3D` for a 3-D Cartesian cell.
- `CARCEL` for a 2-D Cartesian geometry with embedded annular regions,
- `CARCELX` for a 3-D Cartesian geometry with embedded cylindrical regions directed along the  $X$  axis.
- `CARCELY` for a 3-D Cartesian geometry with embedded cylindrical regions directed along the  $Y$  axis.
- `CARCELZ` for a 3-D Cartesian geometry with embedded cylindrical regions directed along the  $Z$  axis.
- `TUBE` for an annular 2-D geometry which may contain an embedded 2-D Cartesian mesh.
- `TUBEX` for an  $X$  directed 3-D cylindrical geometry which may contain an embedded 3-D Cartesian mesh.
- `TUBEY` for a  $Y$  directed 3-D cylindrical geometry which may contain an embedded 3-D Cartesian mesh
- `TUBEZ` for a  $Z$  directed 3-D cylindrical geometry which may contain an embedded 3-D Cartesian mesh

In Section 3.2 we will study the effect of pin clusters superposition on these geometries. Finally in Section 3.3 we will study how these pure geometry are combined to create an assembly.

#### 3.1 Region and surface identification for pure geometries

##### 3.1.1 Pure Cartesian cells

Pure Cartesian cells (see Figure 10) in 2-D and 3-D may be created using the following DRAGON input data structures

- 2-D Cartesian cell

```
Geometry := GEO: CAR2D  n_x n_y
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MIX ((m_{i+n_x(j-1)}, i = 1, n_x), j = 1, n_y) ;
```

- 3-D Cartesian cell

```
Geometry := GEO: CAR3D  n_x n_y n_z
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MESHZ (z_k, k = 0, n_z)
MIX (((m_{i+n_x(j-1)+n_y(k-1)}), i = 1, n_x), j = 1, n_y) k = 1, n_z) ;
```

where  $x_i$ ,  $y_j$  and  $z_k$  are the local position of the lines (in 2-D) or planes (3-D) defining the subregions that can be found in the cell along the  $X$ ,  $Y$  and  $Z$  direction,  $n_x + 1$ ,  $n_y + 1$  and  $n_z + 1$  representing the number of such lines or planes in each direction. Note that for simplicity we will assume that the 2-D geometry is a 3-D geometry with  $n_z = 1$  and

$$\begin{aligned} z_0 &= 0 \\ z_1 &= 1 \end{aligned}$$

even though the 3-D extension is in principle from  $-\infty$  to  $\infty$ . This choice is in fact dictated by the fact that 2-D volumes are in fact identical to 3-D volumes of regions with a height  $z_1 - z_0 = 1$ .

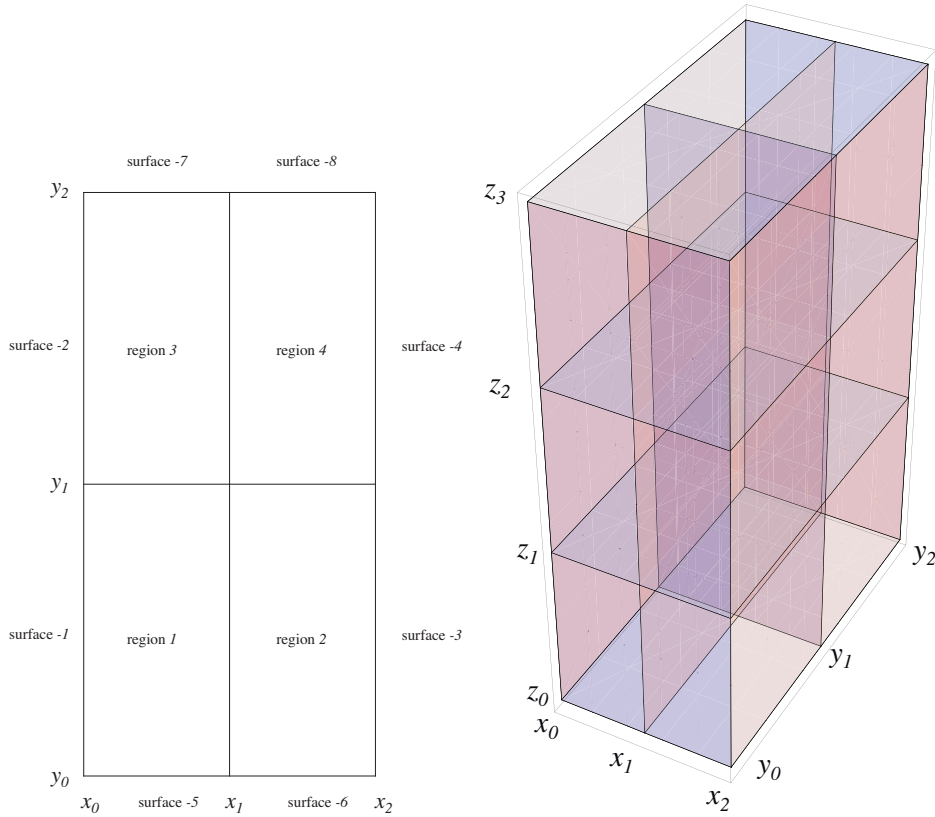


Figure 10: Example of pure Cartesian cells in 2-D (left) and 3-D (right).

Such a Cartesian cell should contain

$$N_r = n_x n_y n_z$$

subregions, the mixture associated with these subregions being provided by  $m_l$ . In DRAGON, each subregion  $l$  with volume  $V_l$  (see Appendix A.1) of a cell can be associated with a position  $(i, j, k)$  in the Cartesian mesh according to (see Figure 10 for an example in 2-D):

$$l = i + n_x (j - 1 + n_y (k - 1)) \quad (3.1)$$

$$V_l = (x_i - x_{i-1})(y_j - y_{j-1})(z_k - z_{k-1}) \quad (3.2)$$

this notation being similar to that used in the input data structure to associate a mixture with a subregion.

The numbering  $\ell$  (negative values) of the surfaces with area  $S_\ell$  follows the following algorithm:

1.  $n_{x,s} = n_y n_z$  surfaces at  $x = x_0$

$$\ell = -(j + n_y(k - 1)) \quad (3.3)$$

$$S_\ell = (y_j - y_{j-1})(z_k - z_{k-1}) \quad (3.4)$$

2.  $n_{x,s}$  surfaces at  $x = x_{n_x}$

$$\ell = -(j + n_y(k - 1)) - n_{x,s} \quad (3.5)$$

$$S_\ell = (y_j - y_{j-1})(z_k - z_{k-1}) \quad (3.6)$$

3.  $n_{y,s} = n_z n_x$  surfaces at  $y = y_0$

$$\ell = -(k + n_z(i - 1)) - 2n_{x,s} \quad (3.7)$$

$$S_\ell = (z_k - z_{k-1})(x_i - x_{i-1}) \quad (3.8)$$

4.  $n_{y,s}$  surfaces at  $y = y_{n_y}$

$$\ell = -(k + n_z(i - 1)) - 2n_{x,s} - n_{y,s} \quad (3.9)$$

$$S_\ell = (z_k - z_{k-1})(x_i - x_{i-1}) \quad (3.10)$$

5.  $n_{z,s} = n_x n_y$  surfaces at  $z = z_0$

$$\ell = -(i + n_x(j - 1)) - 2n_{x,s} - 2n_{y,s} \quad (3.11)$$

$$S_\ell = (x_i - x_{i-1})(y_j - y_{j-1}) \quad (3.12)$$

6.  $n_{z,s}$  surfaces at  $z = z_{n_z}$

$$\ell = -(i + n_x(j - 1)) - 2n_{x,s} - 2n_{y,s} - n_{z,s} \quad (3.13)$$

$$S_\ell = (x_i - x_{i-1})(y_j - y_{j-1}) \quad (3.14)$$

for a total of

$$\begin{aligned} N_S &= 2n_{x,s} + 2n_{y,s} + 2n_{z,s} \\ &= 2n_y n_z + 2n_z n_x + 2n_x n_y \end{aligned} \quad (3.15)$$

surfaces.

### 3.1.2 Pure annular and cylindrical cells

Pure annular (2-D) and cylindrical (3-D) cells (see Figure 11) may be created using the following DRAGON input data structures

- 2-D annular cell

```
Geometry := GEO: TUBE n_r
MESH (r_g, g = 0, n_r)
MIX (m_g, g = 1, n_r) ;
```

containing  $N_r = n_r$  subregions.

- 3-D X directed cylinders cell



```

Geometry := GEO: TUBEX  n_r n_x
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MIX ((m_{g+n_r(i-1)}, g = 1, n_r), i = 1, n_x) ;

```

that contains  $N_r = n_r n_x$  subregions.

- 3-D  $Y$  directed cylinders cell

```

Geometry := GEO: TUBEY  n_r n_y
MESHR (r_g, g = 0, n_r)
MESHY (y_j, j = 0, n_y)
MIX ((m_{g+n_r(j-1)}, g = 1, n_r), j = 1, n_y) ;

```

that contains  $N_r = n_r n_y$  subregions.

- 3-D  $Z$  directed cylinders cell

```

Geometry := GEO: TUBEZ  n_r n_z
MESHR (r_g, g = 0, n_r)
MESHZ (z_k, k = 0, n_z)
MIX ((m_{g+n_r(k-1)}, g = 1, n_r), k = 1, n_z) ;

```

that contains  $N_r = n_r n_z$  subregions.

where  $r_i$  is the radius of the  $n_r$  concentric annular or cylindrical regions in the cell such that  $r_0 = 0.0$ . For 3-D cylinders,  $x_i$ ,  $y_j$  and  $z_k$  are respectively the local position of the planes normal to the cylindrical axis defining the extent of the  $n_x$ ,  $n_y$  or  $n_z$  cylinders. Note that for simplicity we will assume that the 2-D geometry is a 3-D geometry with  $n_z = 1$  and an extension

$$z_0 = 0$$

$$z_1 = 1$$

even though the 3-D extension is in principle from  $-\infty$  to  $\infty$ . The mixture associated with these subregions are provided provided by  $m_l$ .

For the 2-D annular cells, each subregion  $l$  with volume  $V_l$  can be associated with a position  $g$  in the radial mesh according to (see Appendix A.2):

$$l = g \tag{3.16}$$

$$V_g = \pi(r_g - r_{g-1})^2 \tag{3.17}$$

For 3-D cylindrical cells we will use

- $X$  directed cylinders

$$l = g + n_r(i - 1) \tag{3.18}$$

$$V_l = \pi(r_g - r_{g-1})^2(x_i - x_{i-1}) \tag{3.19}$$

- $Y$  directed cylinders

$$l = g + n_r(j - 1) \tag{3.20}$$

$$V_l = \pi(r_g - r_{g-1})^2(y_j - y_{j-1}) \tag{3.21}$$

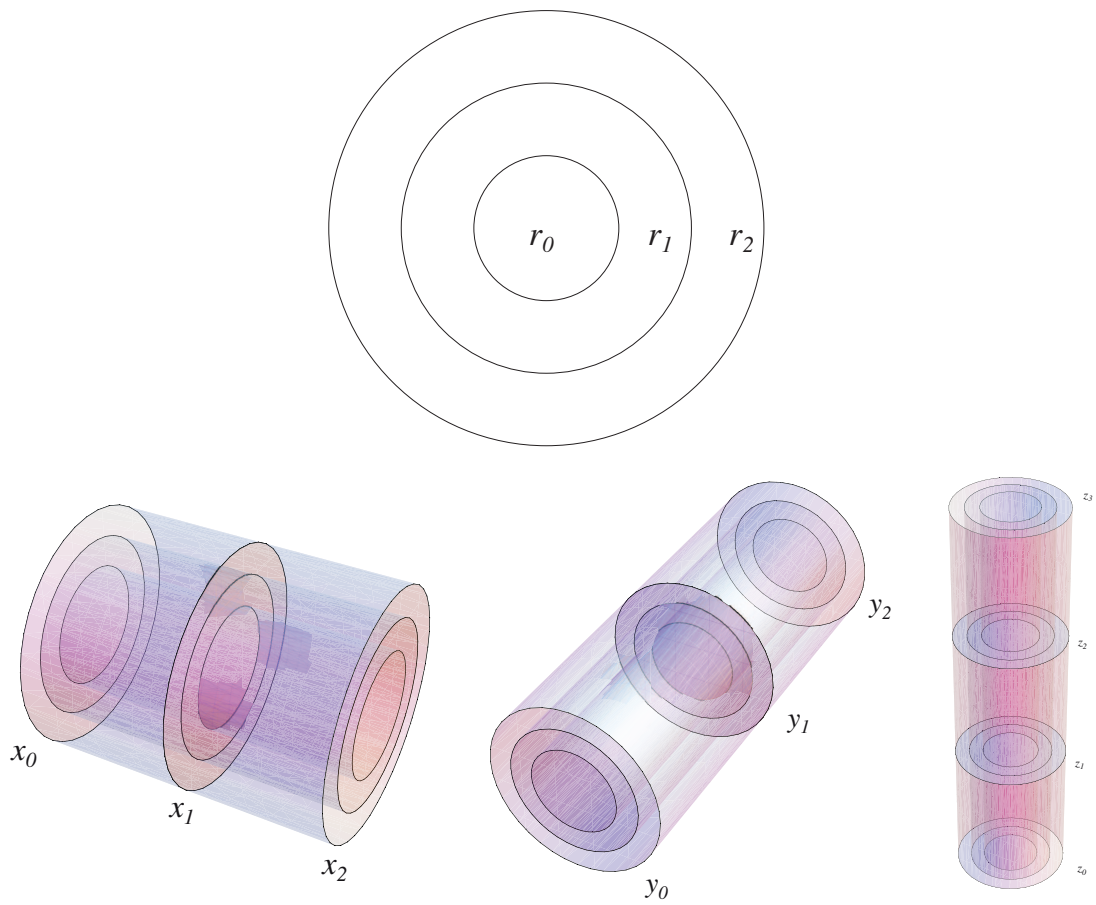


Figure 11: Examples of annular geometry: TUBE (top), TUBEX (bottom left), TUBEY (bottom center) and TUBEZ (bottom right) geometries.

- $X$  directed cylinders

$$l = g + n_r(k - 1) \quad (3.22)$$

$$V_l = \pi(r_g - r_{g-1})^2(z_k - z_{k-1}) \quad (3.23)$$

The volume ordering being consistent with the mixture ordering.

In 2-D, a single outer surface  $\ell = -1$  will be considered with area

$$S_1 = 2\pi r_{n_r} \quad (3.24)$$

while for 3-D cylinders we will use

- $X$  directed cylinders

1.  $n_r$  surfaces at  $x = x_0$

$$\ell = -g \quad (3.25)$$

$$S_\ell = \pi(r_g - r_{g-1})^2 \quad (3.26)$$

2.  $n_r$  surfaces at  $x = x_{n_x}$

$$\ell = -g - n_r \quad (3.27)$$

$$S_\ell = \pi(r_g - r_{g-1})^2 \quad (3.28)$$

3.  $n_x$  radial surfaces (in the  $Y - Z$  plane) at  $r = r_{n_r}$

$$\ell = -i - 2n_r \quad (3.29)$$

$$S_\ell = 2\pi r_{n_r}(x_i - x_{i-1}) \quad (3.30)$$

for a total of  $N_S = 2n_r + n_x$  surfaces.

- $Y$  directed cylinders

1.  $n_r$  surfaces at  $y = y_0$

$$\ell = -g \quad (3.31)$$

$$S_\ell = \pi(r_g - r_{g-1})^2 \quad (3.32)$$

2.  $n_r$  surfaces at  $y = y_{n_y}$

$$\ell = -g - n_r \quad (3.33)$$

$$S_\ell = \pi(r_g - r_{g-1})^2 \quad (3.34)$$

3.  $n_y$  radial surfaces (in the  $Z - X$  plane) at  $r = r_{n_r}$

$$\ell = -j - 2n_r \quad (3.35)$$

$$S_\ell = 2\pi r_{n_r}(y_j - y_{j-1}) \quad (3.36)$$

for a total of  $N_{S,Y} = 2n_r + n_y$  surfaces.

- $Z$  directed cylinders

1.  $n_r$  surfaces at  $z = z_0$

$$\ell = -g \quad (3.37)$$

$$S_\ell = \pi(r_g - r_{g-1})^2 \quad (3.38)$$

2.  $n_r$  surfaces at  $z = z_{n_z}$

$$\ell = -g - n_r \quad (3.39)$$

$$S_\ell = \pi(r_g - r_{g-1})^2 \quad (3.40)$$

3.  $n_z$  radial surfaces (in the  $X - Y$  plane) at  $r = r_{n_r}$

$$\ell = -k - 2n_r \quad (3.41)$$

$$S_\ell = 2\pi r_{n_r}(z_k - z_{k-1}) \quad (3.42)$$

for a total of  $N_{S,Z} = 2n_r + n_z$  surfaces.

### 3.1.3 Cartesian cell with annular subregions

Cartesian 2-D cells containing annular subregions and Cartesian 3-D cells with cylindrical (3-D) subregions (see Figure 12) may be created using the following DRAGON input data structures

- 2-D Cartesian cell with annular mesh

```
Geometry := GEO: CARCEL  n_r n_x n_y
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MIX (((m_{g+(n_r+1)(i-1+n_x(j-1))}, g = 1, n_r + 1), i = 1, n_x), j = 1, n_y) ;
```

containing a maximum of  $N_r = (n_r + 1)n_x n_y$  subregions. Note that the a value of  $(g, i, j)$  with  $g \leq n_r$  correspond to the part of an annular ring located between  $r_g$  and  $r_{g-1}$  that is included in the Cartesian region identified by  $(i, j)$  while a set  $(g, i, j)$  with  $g = n_r + 1$  corresponds to the part of a Cartesian region identified by  $(i, j)$  completely outside the annular ring of radius  $r_{n_r}$ . Some of these regions may not exists when  $g \leq n_r$  since the intersection of the ring and the Cartesian region may vanish. In this case, the a mixture number is still required on input even if it will not be used in the cell description.

- 3-D Cartesian cells containing  $X$  directed cylinders

```
Geometry := GEO: CARCELX  n_r n_x n_y n_z
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MESHZ (z_k, k = 0, n_z)
MIX (((((m_{g+(n_r+1)(j-1+n_y(k-1+n_z(i-1))}), g = 1, n_r+1), j = 1, n_y), k = 1, n_z), i = 1, n_x) ;
```

that contains a maximum of  $N_r = (n_r + 1)n_x n_y n_z$  subregions. Note that the a value of  $(g, i, j, k)$  with  $g \leq n_r$  correspond to the part of an cylindrical ring located between  $r_g$  and  $r_{g-1}$  that is included in the Cartesian region identified by  $(i, j, k)$  while a set  $(g, i, j, k)$  with  $g = n_r + 1$  corresponds to the part of the Cartesian region identified by  $(i, j, k)$  completely outside the cylindrical ring of radius  $r_{n_r}$ . Some of these regions may not exists when  $g \leq n_r$  since the intersection of the ring and the Cartesian region may vanish. In this case, the a mixture number is still required on input even if it will not be used in the cell description.

- 3-D Cartesian cells containing  $Y$  directed cylinders

```

Geometry := GEO: CARCELY  n_r n_x n_y n_z
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MESHZ (z_k, k = 0, n_z)
MIX (((((m_{g+(n_r+1)(k-1+n_z(i-1+n_x(j-1))}), g = 1, n_r+1), k = 1, n_z), i = 1, n_x), j = 1, n_y) ;

```

that contains a maximum of  $N_r = (n_r + 1)n_x n_y n_z$  subregions. The same comment on mixtures input as that provided for Cartesian cells with  $X$  directed cylinders remains valid here.

- 3-D Cartesian cells containing  $Z$  directed cylinders

```

Geometry := GEO: CARCELZ  n_r n_x n_y n_z
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MESHZ (z_k, k = 0, n_z)
MIX (((((m_{g+(n_r+1)(i-1+n_x(j-1+n_y(k-1))}), g = 1, n_r + 1), i = 1, n_x), j = 1, n_y), k = 1, n_z) ;

```

that contains a maximum of  $N_r = (n_r + 1)n_x n_y n_z$  subregions. Again some mixtures may not be used in the calculations as described for Cartesian cells with  $X$  directed cylinders.

Here  $r_i$  is the again radius of the  $n_r$  concentric annular or cylindrical regions in the cell such that  $r_0 = 0.0$ . For 3-D cylinders,  $x_i$ ,  $y_j$  and  $z_k$  are respectively the local position of the planes normal to the cylindrical axis defining the extent of the  $n_x$ ,  $n_y$  or  $n_z$  cylinders. Note that for simplicity we will again assume that the 2-D geometry is a 3-D geometry with  $n_z = 1$  and  $z_0 = 0$  and  $z_1 = 1$  even though the 3-D extension is in principle from  $-\infty$  to  $\infty$ .

For the 2-D CARCEL geometries, each possible subregion  $l$  of the cell can be associated with a position  $(g, i, j)$  according to (see Appendix A.2):

$$l = g + (n_r + 1)(i - 1 + n_x(j - 1)) \quad (3.43)$$

for a maximum of  $N_r$  regions. The volume  $V_l$  of these regions can be evaluated using the procedures described in Appendix A. Note that some of these volumes may turn out to vanish since not all the radial regions will have an intersection with each of the Cartesian region. Thus, the final region identification will correspond to  $\{l_c\}$ , a compressed version of the set  $\{l\}$  that contains  $N_V$  terms where the elements with vanishing volume have been removed:

```

lc=0
DO l=1, N_{r}
  IF (V_{l} .NE. 0) THEN
    lc=lc+1
    V_{lc}=V_{l}
    m_{lc}=m_{l}
  ENDF
ENDDO
N_{V}=lc

```

where the mixture identification vector is compressed in a fashion similar to  $V_l$ .

For the 3-D CARCELX, CARCELY and CARCELZ geometries, a similar process is used where each possible subregion  $l$  of the cell can be now associated with a position  $(g, i, j, k)$ . Here we use:

$$l = g + (n_r + 1)(i - 1 + n_x(j - 1 + n_y(k - 1))) \quad (3.44)$$

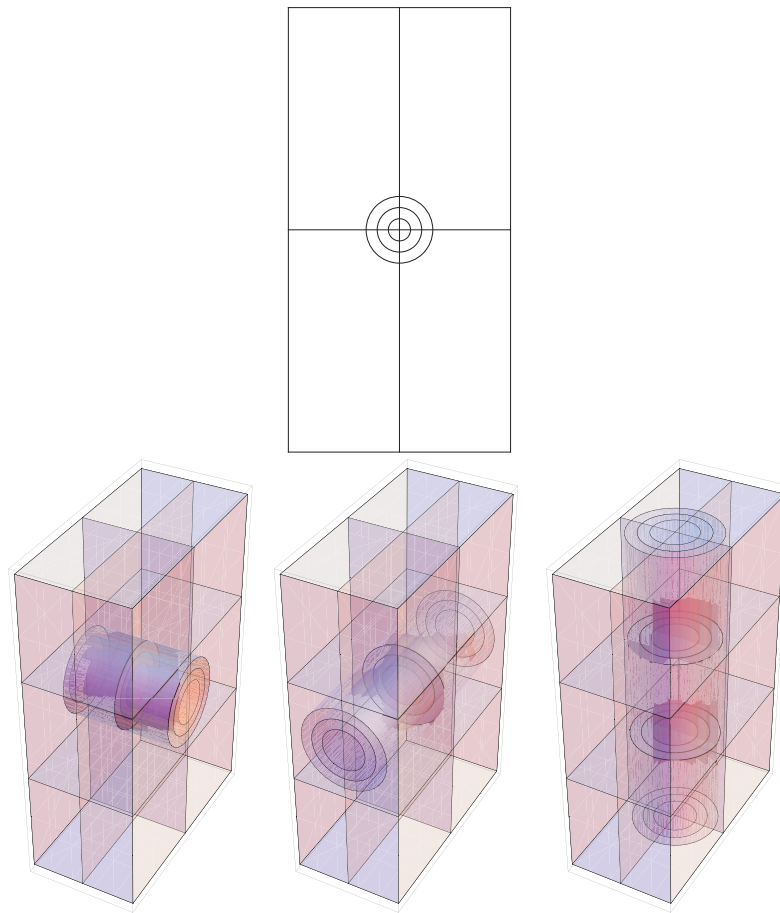


Figure 12: Examples of Cartesian geometry with annular subregions: CARCEL (top), CARCELX (bottom left), CARCELY (bottom center) and CARCELZ (bottom right) geometries.

for a maximum of  $N_r$  regions. The global ordering here is different from that used for the mixture ordering. The volumes  $V_l$  can again be evaluated using the procedures described in Appendix A. As for the CARCEL geometry, some of the volumes may turn out to vanish. Thus, the final region identification will correspond to  $\{l_c\}$ , a compressed version of the set  $\{l\}$  that contains  $N_V$  terms where the elements with vanishing volume can be removed using the procedure described above.

In 2-D, the outer surfaces  $\ell$  are numbered according to the following procedure

1.  $n_y$  surfaces at  $x = x_0$

$$\ell = -j \quad (3.45)$$

$$S_\ell = (y_j - y_{j-1}) \quad (3.46)$$

2.  $n_y$  surfaces at  $x = x_{n_x}$

$$\ell = -j - n_y \quad (3.47)$$

$$S_\ell = (y_j - y_{j-1}) \quad (3.48)$$

3.  $n_x$  surfaces at  $y = y_0$

$$\ell = -i - 2n_y \quad (3.49)$$

$$S_\ell = (x_i - x_{i-1}) \quad (3.50)$$

4.  $n_x$  surfaces at  $y = y_{n_y}$

$$\ell = -i - 2n_y - n_x \quad (3.51)$$

$$S_\ell = (x_i - x_{i-1}) \quad (3.52)$$

for a total of

$$N_S = 2n_y + 2n_x \quad (3.53)$$

surfaces.

For 3-D Cartesian geometries with embedded cylindrical regions, the surfaces are numbered according to the following procedure

- CARCELX

1. A maximum of  $n_{r,x} = (n_r + 1)n_y n_z$  surfaces at  $x = x_0$

$$\ell = -g - (n_r + 1)(j - 1 + n_y(k - 1)) \quad (3.54)$$

with  $S_\ell$  evaluated using the procedure of Appendix A.

2. A maximum of  $n_{r,x}$  surfaces at  $x = x_{n_x}$

$$\ell = -g - (n_r + 1)(j - 1 + n_y(k - 1)) - n_{r,x} \quad (3.55)$$

with  $S_\ell$  evaluated using the procedure of Appendix A.

3.  $n_x n_z$  surfaces at  $y = y_0$

$$\ell = -k - n_z(i - 1) - 2n_{r,x} \quad (3.56)$$

$$S_\ell = (x_i - x_{i-1})(z_k - z_{k-1}) \quad (3.57)$$

4.  $n_x n_z$  surfaces at  $y = y_{n_y}$

$$\ell = -k - n_z(i - 1) - 2n_{r,x} - n_x n_z \quad (3.58)$$

$$S_\ell = (x_i - x_{i-1})(z_k - z_{k-1}) \quad (3.59)$$

5.  $n_x n_y$  surfaces at  $z = z_0$

$$\ell = -i - n_x(j - 1) - 2n_{r,x} - 2n_x n_z \quad (3.60)$$

$$S_\ell = (x_i - x_{i-1})(y_j - y_{j-1}) \quad (3.61)$$

6.  $n_x n_y$  surfaces at  $z = z_{n_z}$

$$\ell = -i - n_x(j - 1) - 2n_{r,x} - 2n_x n_z - n_x n_y \quad (3.62)$$

$$S_\ell = (x_i - x_{i-1})(y_j - y_{j-1}) \quad (3.63)$$

for a maximum of

$$N_S = 2(n_{r,x} + n_x(n_z + n_y)) \quad (3.64)$$

surfaces.

- CARCELY

1.  $n_y n_z$  surfaces at  $x = x_0$

$$\ell = -j - n_y(k - 1) \quad (3.65)$$

$$S_\ell = (y_j - y_{j-1})(z_k - z_{k-1}) \quad (3.66)$$

2.  $n_y n_z$  surfaces at  $x = x_{n_x}$

$$\ell = -j - n_y(k - 1) - n_y n_z \quad (3.67)$$

$$S_\ell = (y_j - y_{j-1})(z_k - z_{k-1}) \quad (3.68)$$

3. A maximum of  $n_{r,y} = (n_r + 1)n_x n_z$  surfaces at  $y = y_0$

$$\ell = -g - (n_r + 1)(k - 1 + n_z(i - 1)) - 2n_y n_z \quad (3.69)$$

with  $S_\ell$  evaluated using the procedure of Appendix A.

4. A maximum of  $n_{r,y}$  surfaces at  $y = y_{n_y}$

$$\ell = -g - (n_r + 1)(k - 1 + n_z(i - 1)) - 2n_y n_z - n_{r,y} \quad (3.70)$$

with  $S_\ell$  evaluated using the procedure of Appendix A.

5.  $n_x n_y$  surfaces at  $z = z_0$

$$\ell = -i - n_x(j - 1) - 2n_{r,y} - 2n_y n_z \quad (3.71)$$

$$S_\ell = (x_i - x_{i-1})(y_j - y_{j-1}) \quad (3.72)$$

6.  $n_x n_y$  surfaces at  $z = z_{n_z}$

$$\ell = -i - n_x(j - 1) - 2n_{r,y} - 2n_y n_z - n_x n_y \quad (3.73)$$

$$S_\ell = (x_i - x_{i-1})(y_j - y_{j-1}) \quad (3.74)$$



for a maximum of

$$N_S = 2(n_{r,x} + n_y(n_z + n_x)) \quad (3.75)$$

surfaces.

- CARCELZ

1.  $n_y n_z$  surfaces at  $x = x_0$

$$\ell = -j - n_y(k - 1) \quad (3.76)$$

$$S_\ell = (y_j - y_{j-1})(z_k - z_{k-1}) \quad (3.77)$$

2.  $n_y n_z$  surfaces at  $x = x_{n_x}$

$$\ell = -j - n_y(k - 1) - n_y n_z \quad (3.78)$$

$$S_\ell = (y_j - y_{j-1})(z_k - z_{k-1}) \quad (3.79)$$

3.  $n_x n_z$  surfaces at  $y = y_0$

$$\ell = -k - n_z(i - 1) - 2n_y n_z \quad (3.80)$$

$$S_\ell = (x_i - x_{i-1})(z_k - z_{k-1}) \quad (3.81)$$

4.  $n_x n_z$  surfaces at  $y = y_{n_y}$

$$\ell = -k - n_z(i - 1) - 2n_y n_z - n_x n_z \quad (3.82)$$

$$S_\ell = (x_i - x_{i-1})(z_k - z_{k-1}) \quad (3.83)$$

5. A maximum of  $n_{r,z} = (n_r + 1)n_x n_y$  surfaces at  $z = z_0$

$$\ell = -g - (n_r + 1)(i - 1 + n_x(j - 1)) - 2n_y n_z - 2n_x n_z \quad (3.84)$$

with  $S_\ell$  evaluated using the procedure of Appendix A.

6. A maximum of  $n_{r,z}$  surfaces at  $z = z_{n_z}$

$$\ell = -g - (n_r + 1)(i - 1 + n_x(j - 1)) - 2n_y n_z - 2n_x n_z - n_{r,z} \quad (3.85)$$

with  $S_\ell$  evaluated using the procedure of Appendix A.

The maximum number of surfaces in this case is

$$N_S = 2(n_{r,x} + n_z(n_x + n_y)) \quad (3.86)$$

Again some of these surfaces will vanish and the final index associated with the surfaces will be compressed using a procedure similar to that used for the volumes.

### 3.1.4 Annular cell with Cartesian subregions

Annular (2-D) and cylindrical (3-D) cells (see Figure 13) with Cartesian subregions may be created using the following DRAGON input data structures

- 2-D annular cell

```

Geometry := GEO: TUBE  n_r n_x n_y
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
OFFCENTER x_c y_y MIX (((m_{g+n_r(i-1+n_x(j-1))}, g = 1, n_r), i = 1, n_x), j = 1, n_y) ;

```

containing a maximum of  $N_r = n_r n_x n_y$  subregions. Here the conditions:

$$\begin{aligned}
 x_0 &< -r_{n_r} + \frac{x_0 + x_{n_x}}{2} + x_c \\
 x_{n_x} &> r_{n_r} + \frac{x_0 + x_{n_x}}{2} + x_c \\
 y_0 &< -r_{n_r} + \frac{y_0 + y_{n_y}}{2} + y_c \\
 y_{n_y} &> r_{n_r} + \frac{y_0 + y_{n_y}}{2} + y_c
 \end{aligned}$$

must be satisfied.

- 3-D  $X$  directed cylinders cell

```

Geometry := GEO: TUBEX  n_r n_x n_y n_z
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MESHZ (z_k, k = 0, n_z)
OFFCENTER x_c y_y z_c MIX (((((m_{g(n_r(j-1+n_y(k-1+n_z(i-1)))}, g = 1, n_r), j = 1, n_y), k = 1, n_z), i = 1, n_x) ;

```

that contains a maximum of  $N_r = n_r n_x n_y n_z$  subregions. Here the conditions:

$$\begin{aligned}
 y_0 &< -r_{n_r} + \frac{y_0 + y_{n_y}}{2} + y_c \\
 y_{n_y} &> r_{n_r} + \frac{y_0 + y_{n_y}}{2} + y_c \\
 z_0 &< -r_{n_r} + \frac{z_0 + z_{n_z}}{2} + z_c \\
 z_{n_z} &> r_{n_r} + \frac{z_0 + z_{n_z}}{2} + z_c
 \end{aligned}$$

must be satisfied.

- 3-D  $Y$  directed cylinders cell

```

Geometry := GEO: TUBEY  n_r n_x n_y n_z
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MESHZ (z_k, k = 0, n_z)
OFFCENTER x_c y_y z_c MIX (((((m_{g+n_r(k-1+n_z(i-1+n_x(j-1)))}, g = 1, n_r), k = 1, n_z), i = 1, n_x), j = 1, n_y) ;

```

that contains a maximum of  $N_r = n_r n_x n_y n_z$  subregions. Here the conditions:

$$\begin{aligned} x_0 &< -r_{n_r} + \frac{x_0 + x_{n_x}}{2} + x_c \\ x_{n_x} &> r_{n_r} + \frac{x_0 + x_{n_x}}{2} + x_c \\ z_0 &< -r_{n_r} + \frac{z_0 + z_{n_z}}{2} + z_c \\ z_{n_z} &> r_{n_r} + \frac{z_0 + z_{n_z}}{2} + z_c \end{aligned}$$

must be satisfied.

- 3-D Z directed cylinders cell

```

Geometry := GEO: TUBEZ  n_r n_x n_y n_z
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MESHZ (z_k, k = 0, n_z)
OFFCENTER x_c y_y z_c MIX (((m_{g+n_r(i-1+n_x(j-1+n_y(k-1)))}, g = 1, n_r), i = 1, n_x), j =
1, n_y), k = 1, n_z) ;

```

that contains a maximum of  $N_r = n_r n_x n_y n_z$  subregions. Here the conditions:

$$\begin{aligned} x_0 &< -r_{n_r} + \frac{x_0 + x_{n_x}}{2} + x_c \\ x_{n_x} &> r_{n_r} + \frac{x_0 + x_{n_x}}{2} + x_c \\ y_0 &< -r_{n_r} + \frac{y_0 + y_{n_y}}{2} + y_c \\ y_{n_y} &> r_{n_r} + \frac{y_0 + y_{n_y}}{2} + y_c \end{aligned}$$

must be satisfied.

Only the part of each Cartesian region totally inside the annular or cylindrical region will be considered. The mixture associated with these subregions are provided provided by  $m_l$ .

For the 2-D TUBE geometries, each possible subregion  $l$  of the cell can be associated with a position  $(g, i, j)$  according to:

$$l = g + n_r(i - 1 + n_x(j - 1)) \quad (3.87)$$

for a maximum of  $N_r$  regions. The volume  $V_l$  of these regions can be evaluated using the procedures described in Appendix A. Note that some of these volumes may turn out to vanish since not all the radial regions will have an intersection with each of the Cartesian region. Thus, the final region identification will correspond to  $\{l_c\}$ , a compressed version of the set  $\{l\}$  that contains  $N_V$  terms where the elements with vanishing volume have been removed:

```

l_c=0
DO l=1, N_{r}
  IF(V_{l} .NE. 0) THEN
    l_c=l_c+1
    V_{l_c}=V_{l}
    m_{l_c}=m_{l}
  ENDIF
ENDDO

```

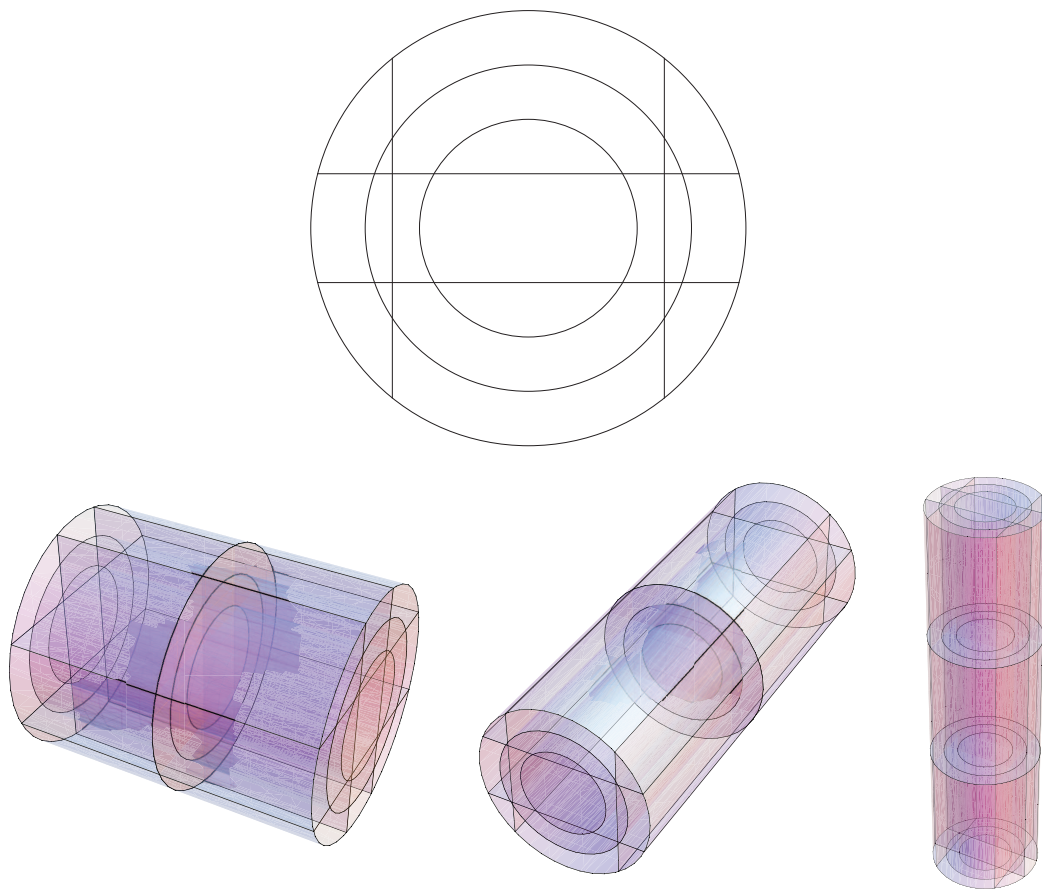


Figure 13: Examples of annular geometry with Cartesian subregions: TUBE (top), TUBEX (bottom left), TUBEY (bottom center) and TUBEZ (bottom right).

```

    ENDDIF
  ENDDO
  N_{V}=1c

```

where the mixture identification vector is compressed in a fashion similar to  $V_l$ .

For the 3-D TUBEX, TUBEY and TUBEZ geometries, a similar process is used where each possible subregion  $l$  of the cell can be now associated with a position  $(g, i, j, k)$ . Here we use:

$$l = g + n_r(i - 1 + n_x(j - 1 + n_y(k - 1))) \quad (3.88)$$

for a maximum of  $N_r$  regions. The global ordering here is different from that used for the mixture ordering. The volumes  $V_l$  can again be evaluated using the procedures described in Appendix A. As for the TUBE geometry, some of the volumes may turn out to vanish. Thus, the final region identification will correspond to  $\{l_c\}$ , a compressed version of the set  $\{l\}$  that contains  $N_V$  terms where the elements with vanishing volume can be removed using the procedure described above.

In 2-D, the outer surfaces  $\ell$  are numbered according to the following procedure

$$\ell = -i - n_x(j - 1) \quad (3.89)$$

for a maximum of  $n_y n_x$  radial surfaces at  $r_{n_r}$  (for example in Figure 13, there is no outer surface associated with the Cartesian region  $(i, j) = (2, 2)$ ).

For 3-D Cylindrical geometries with embedded Cartesian regions, the surfaces are numbered according to the following procedure

- TUBEX

1. A maximum of  $n_{r,x} = n_r n_y n_z$  surfaces at  $x = x_0$

$$\ell = -g - n_r(j - 1 + n_y(k - 1)) \quad (3.90)$$

with  $S_\ell$  evaluated using the procedure of Appendix A.

2. A maximum of  $n_{r,x}$  surfaces at  $x = x_{n_x}$

$$\ell = -g - n_r(j - 1 + n_y(k - 1)) - n_{r,x} \quad (3.91)$$

with  $S_\ell$  evaluated using the procedure of Appendix A.

3. A maximum of  $n_x n_y n_z$  radial surfaces at  $r = r_{n_r}$

$$\ell = -i - n_x(j - 1 + n_y(k - 1)) - 2n_{r,x} \quad (3.92)$$

for a maximum of

$$N_S = 2n_{r,x} + n_x n_y n_z \quad (3.93)$$

surfaces.

- TUBEY

1. A maximum of  $n_{r,y} = n_r n_x n_z$  surfaces at  $y = y_0$

$$\ell = -g - n_r(i - 1 + n_z(k - 1)) \quad (3.94)$$

2. A maximum of  $n_{r,y}$  surfaces at  $y = y_{n_y}$

$$\ell = -g - n_r(i - 1 + n_x(k - 1)) - n_{r,y} \quad (3.95)$$

3. A maximum of  $n_x n_y n_z$  radial surfaces at  $r = r_{n_r}$

$$\ell = -i - n_x(j - 1 + n_y(k - 1)) - 2n_{r,x} \quad (3.96)$$

for a maximum of

$$N_S = 2n_{r,y} + n_x n_y n_z \quad (3.97)$$

surfaces.

- TUBEZ

1. A maximum of  $n_{r,z} = n_r n_x n_y$  surfaces at  $z = z_0$

$$\ell = -g - n_r(i - 1 + n_x(j - 1)) \quad (3.98)$$

2. A maximum of  $n_{r,z}$  surfaces at  $z = z_{n_z}$

$$\ell = -g - n_r(i - 1 + n_x(j - 1)) - n_{r,z} \quad (3.99)$$

3. A maximum of  $n_x n_y n_z$  radial surfaces at  $r = r_{n_r}$

$$\ell = -i - n_x(j - 1 + n_y(k - 1)) - 2n_{r,x} \quad (3.100)$$

for a maximum of

$$N_S = 2n_{r,z} + n_x n_y n_z \quad (3.101)$$

surfaces.

Again some of these surfaces will vanish and the final index associated with the surfaces will be compressed using a procedure similar to that used for the volumes.

### 3.2 Pins in cells

Pin cells can be inserted in a pure geometry using the command CLUSTER. A 2-D example can be found in Figure 14 corresponding to

```

Geometry := GEO: CARCEL  n_r n_x n_y
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MIX (((m_{g+(n_r+1)(i-1+n_x(j-1))}, g = 1, n_r + 1), i = 1, n_x), j = 1, n_y)
CLUSTER GeoPin
  ::: GeoPin := GEO: TUBE  n_{r,l} n_{x,l} n_{y,l}
  MESHR (r_{g,l}, g = 0, n_{r,l})
  MESHX (x_{i,l}, i = 0, n_{x,l})
  MESHY (y_{j,l}, i = 0, n_{y,l})
  MIX (((m_{g+n_r(i-1+n_x(j-1))}, g = 1, n_{r,l}), i = 1, n_{x,l}), j = 1, n_{y,l})
  NPIN 2 RPIN r_p APIN theta_p ;
;

```

where two pin geometry GeoPin are inserted in the cell Geometry at positions

$$(x_1, y_1) = \left( r_p \cos(\theta_p) + \frac{x_0 + x_{n_x}}{2}, r_p \sin(\theta_p) + \frac{y_0 + y_{n_y}}{2} \right)$$

$$(x_2, y_2) = \left( r_p \cos(\theta_p + \pi) + \frac{x_0 + x_{n_x}}{2}, r_p \sin(\theta_p + \pi) + \frac{y_0 + y_{n_y}}{2} \right)$$

Similarly, the 3-D example illustrated in Figure 15 corresponds to

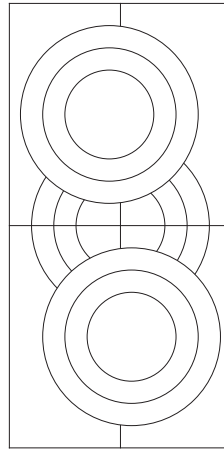


Figure 14: Annular pins in CARCEL geometries.

```

Geometry := GEO: CARCELX  n_r n_x n_y n_z
MESHR (r_g, g = 0, n_r)
MESHX (x_i, i = 0, n_x)
MESHY (y_j, j = 0, n_y)
MESHZ (z_k, k = 0, n_z)
MIX (((m_g+(n_r+1)(j-1+n_y(k-1+n_z(i-1))), g = 1, n_r + 1), j = 1, n_y), k = 1, n_z), i = 1, n_x)
CLUSTER GeoPin
  ::: GeoPin := GEO: TUBEX  n_r,p n_x,p n_y,p n_z,p
  MESHR (r_g,p, g = 0, n_r,p)
  MESHX (x_i,p, i = 0, n_x,p)
  MESHY (y_j,p, i = 0, n_y,p)
  MESHZ (z_k,p, k = 0, n_z,p)
  MIX (((m_g+(n_r)(j-1+n_y(k-1+n_z(i-1))), g = 1, n_r), j = 1, n_y), k = 1, n_z), i = 1, n_x)
  NPIN 2 RPIN r_p i
i

```

The main feature of the pure geometry that are inserted in a cell using this command is the fact that the pins hide completely the part of the cell geometry that is located completely inside the limits of the pins (see Figure 14 for a 2-D example)

### 3.3 Cells in assemblies

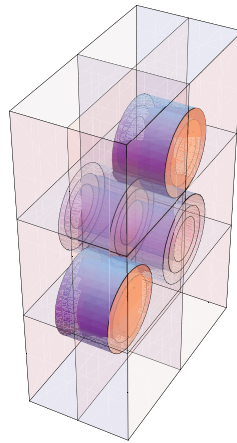


Figure 15:  $X$  directed cylindrical pins in CARCELX geometries.



## 4 TRACKING IN THE NXT MODULE OF DRAGON

### 4.1 Region Volume and Surface Area Evaluation

#### 4.1.1 Cartesian Region

The subroutine that is used to evaluate the region volume and surface area in this case is NXTVCA. Depending on the value of main direction parameter IDIRC specified, the local  $x$ ,  $y$  and  $z$  ordering of axis may differ from the reference  $X$ ,  $Y$  and  $Z$  axis. In fact we will have:

- $x = X, y = Y, z = Z$  when IDIRC=1;
- $x = Y, y = Z, z = X$  when IDIRC=2;
- $x = Z, y = X, z = Y$  when IDIRC=3.

The region number  $1 \leq n \leq N_x N_y N_z$  associated with position  $i, j, k$  in the  $x, y, z$  mesh will be computed using

$$n = i + (j + kN_y)N_x \quad (4.1)$$

and will have a volume given by

$$V_n = \Delta_x \Delta_y \Delta_z \quad (4.2)$$

where for a 3-D geometry we have

$$\Delta_x = (x_i - x_{i-1}) \quad (4.3)$$

$$\Delta_y = (y_j - y_{j-1}) \quad (4.4)$$

$$\Delta_z = (z_k - z_{k-1}) \quad (4.5)$$

For a 2-D  $x - y$  geometry we will assume that  $\Delta_z = 1$  while for  $y - z$  and  $z - x$  geometries we will use  $\Delta_x = 1$  and  $\Delta_y = 1$  respectively. Finally for 1-D  $x$ ,  $y$  and  $z$  geometries we will select successively  $\Delta_y = \Delta_z = 1$ ,  $\Delta_z = \Delta_x = 1$  and  $\Delta_x = \Delta_y = 1$ .

Note that we will use a similar definition of  $\Delta_x$ ,  $\Delta_y$  and  $\Delta_z$  for surface area calculation:

$$S_m = \Delta_x \Delta_y \Delta_z \quad (4.6)$$

For example, in 1-D geometry we will use  $\Delta_z = \Delta_y = \Delta_x = 1$ . For a 2-D  $x - y$  geometry we will consider  $\Delta_z = \Delta_x = 1$  and  $\Delta_z = \Delta_y = 1$  respectively for surfaces parallel to the  $y$  and  $x$  axis. Similar relations will also be used for 2-D  $y - z$  and  $z - x$  geometries. Finally, for 3-D geometries we will use successively  $\Delta_x = 1$ ,  $\Delta_y = 1$  and  $\Delta_z = 1$  for surfaces parallel to the  $y - z$ ,  $z - x$  and  $x - y$  planes. The surface are then numbered according to:

- $m = j + N_y k$  for location  $(j, k)$  in the  $y - z$  plane located at  $x_0$  ( $i = -1$ );
- $m = j + N_y k + N_y N_z$  for location  $(j, k)$  in the  $y - z$  plane located at  $x_{N_x}$  ( $i = -2$ );
- $m = k + N_z i + 2N_y N_z$  for location  $(k, i)$  in the  $z - x$  plane located at  $y_0$  ( $j = -1$ );
- $m = k + N_z i + N_z N_x + 2N_y N_z$  for location  $(k, i)$  in the  $z - x$  plane located at  $y_{N_y}$  ( $j = -2$ );
- $m = i + N_x j + 2(N_y N_z + N_z N_x)$  for location  $(i, j)$  in the  $x - y$  plane located at  $z_0$  ( $k = -1$ );
- $m = i + N_x j + 2(N_y N_z + N_z N_x) + 2N_x N_y$  for location  $(i, j)$  in the  $x - y$  plane located at  $z_{N_z}$  ( $k = -2$ ).

### 4.1.2 Cylindrical Region

### 4.1.3 Mixed Cartesian/Cylindrical Region

## 4.2 Geometry Description

Here we will describe the various procedures in the NXT module to track multilevel geometries. The notation we will use is the following:

- $x_i^\gamma$  is the  $x$ -directed spatial mesh for subgeometry  $\gamma$ . This vector contains  $i = 0, N_x^\gamma$  elements classified in increasing value of  $x$  and stored in record  $d\text{mesh}x(\gamma)$ .
- $y_j^\gamma$  is the  $y$ -directed spatial mesh for subgeometry  $\gamma$ . This vector contains  $j = 0, N_y^\gamma$  elements classified in increasing value of  $y$  and stored in record  $d\text{mesh}y(\gamma)$ .
- $z_k^\gamma$  is the  $z$ -directed spatial mesh for subgeometry  $\gamma$ . This vector contains  $k = 0, N_z^\gamma$  elements classified in increasing value of  $z$  and stored in record  $d\text{mesh}z(\gamma)$ .
- $r_l^\gamma$  is the radial mesh for subgeometry  $\gamma$ . This vector contains  $l = 1, N_r^\gamma$  elements classified in increasing value of  $r$ . For 3-D geometries, one must also specify the direction  $d$  of the axis of the cylinder with  $d = x$  for a  $x$ -directed cylinder,  $d = y$  for a  $y$ -directed cylinder and  $d = z$  for a  $z$ -directed cylinder. For 2-D geometries, the circles are assumed to lie in the  $x - y$  plane (direction  $d = z$ ). This vector contains  $l = 1, N_r^\gamma$  elements classified in increasing value of  $r$  and stored in record  $d\text{mesh}r(\gamma)$ .
- $(r_{-1}^\gamma, r_{-2}^\gamma)$  is the position of the center of the cylinders in a plane normal to the direction  $d$ . In the case where  $d = x$ , the point  $(r_{-1}^\gamma, r_{-2}^\gamma) = (y, z)$  represents a position in the  $Y - Z$  plane. For  $d = y$ , the point  $(r_{-1}^\gamma, r_{-2}^\gamma) = (z, x)$  represents a position in the  $Z - X$  plane while for  $d = z$ , the point  $(r_{-1}^\gamma, r_{-2}^\gamma) = (x, y)$  represents a position in the  $X - Y$  plane. This information corresponds to the elements  $l = -1, -2$  of record  $d\text{mesh}r(\gamma)$ .
- $I_{lijk}^v$  identifies the region located in

$$x_{i-1}^\gamma < x < x_i^\gamma \quad y_{j-1}^\gamma < y < y_j^\gamma \quad z_{k-1}^\gamma < z < z_k^\gamma \quad r_{l-1}^\gamma < r < r_l^\gamma$$

Note that the regions associated with a position such that

$$x_{i-1}^\gamma < x < x_i^\gamma \quad y_{j-1}^\gamma < y < y_j^\gamma \quad z_{k-1}^\gamma < z < z_k^\gamma \quad r_{N_r^\gamma}^\gamma < r$$

is identified by  $I_{l'ijk}^v$  where  $l' = N_r^\gamma + 1$ . This region is located inside the Cartesian mesh but outside the annular region. This is the case of region  $I_{3111}^v$  in Figure 12. Similarly,

$$\begin{array}{llllll} x < x_0^\gamma & y_{j-1}^\gamma < y < y_j^\gamma & z_{k-1}^\gamma < z < z_k^\gamma & r_{l-1}^\gamma < r < r_l^\gamma & \rightarrow I_{l0jk}^v \\ x_{N_x^\gamma}^\gamma < x & y_{j-1}^\gamma < y < y_j^\gamma & z_{k-1}^\gamma < z < z_k^\gamma & r_{l-1}^\gamma < r < r_l^\gamma & \rightarrow I_{li'jk}^v \\ x_{i-1}^\gamma < x < x_i^\gamma & y < y_0^\gamma & z_{k-1}^\gamma < z < z_k^\gamma & r_{l-1}^\gamma < r < r_l^\gamma & \rightarrow I_{li0k}^v \\ x_{i-1}^\gamma < x < x_i^\gamma & y_{N_y^\gamma}^\gamma < y & z_{k-1}^\gamma < z < z_k^\gamma & r_{l-1}^\gamma < r < r_l^\gamma & \rightarrow I_{lij'k}^v \\ x_{i-1}^\gamma < x < x_i^\gamma & y_{j-1}^\gamma < y < y_j^\gamma & z < z_0^\gamma & r_{l-1}^\gamma < r < r_l^\gamma & \rightarrow I_{lij0}^v \\ x_{i-1}^\gamma < x < x_i^\gamma & y_{j-1}^\gamma < y < y_j^\gamma & z_{N_z^\gamma}^\gamma < z & r_{l-1}^\gamma < r < r_l^\gamma & \rightarrow I_{lij'k'}^v \end{array}$$

with  $i' = N_x^\gamma + 1$ ,  $j' = N_y^\gamma + 1$  and  $k' = N_z^\gamma + 1$  as illustrated in Figure 16.

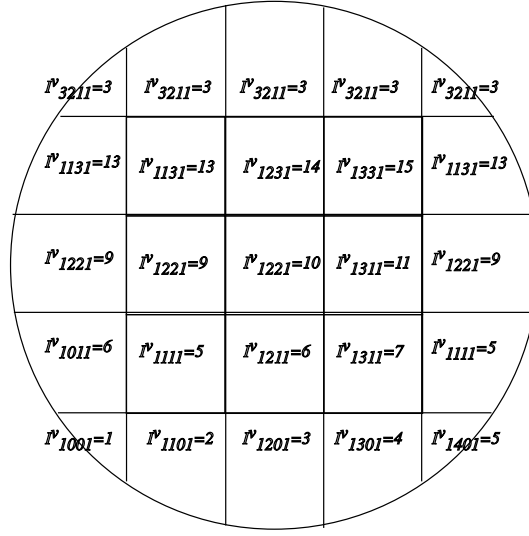


Figure 16: Numbering a multiregion geometry in DRAGON.

- $S_{-,jkl}^x$  identify the region located on the plane  $x = X_0^{g,x}$  with (record INDX-X):

$$\begin{aligned} X_{j-1}^{g,y} &< y < X_j^{g,y} \\ X_{k-1}^{g,z} &< z < X_k^{g,z} \\ R_{l-1}^{g,d} &< r < R_l^{g,d} \end{aligned}$$

- $S_{+,jkl}^x$  identify the region located on the plane  $x = X_{N_x^g}^{g,x}$  with (record INDX+X):

$$\begin{aligned} X_{j-1}^{g,y} &< y < X_j^{g,y} \\ X_{k-1}^{g,z} &< z < X_k^{g,z} \\ R_{l-1}^{g,d} &< r < R_l^{g,d} \end{aligned}$$

- $S_{-,ikl}^y$  identify the region located on the plane  $y = X_0^{g,y}$  with (record INDX-Y):

$$\begin{aligned} X_{i-1}^{g,x} &< x < X_i^{g,x} \\ X_{k-1}^{g,z} &< z < X_k^{g,z} \\ R_{l-1}^{g,d} &< r < R_l^{g,d} \end{aligned}$$

- $S_{+,ikl}^y$  identify the region located on the plane  $y = X_{N_y^g}^{g,y}$  with (record INDX+Y):

$$\begin{aligned} X_{i-1}^{g,x} &< x < X_i^{g,x} \\ X_{k-1}^{g,z} &< z < X_k^{g,z} \\ R_{l-1}^{g,d} &< r < R_l^{g,d} \end{aligned}$$

- $S_{-,ijl}^z$  identify the region located on the plane  $z = X_0^{g,z}$  with (record INDX-Z):

$$\begin{aligned} X_{i-1}^{g,x} &< x < X_i^{g,x} \\ X_{j-1}^{g,y} &< y < X_j^{g,y} \\ R_{l-1}^{g,d} &< r < R_l^{g,d} \end{aligned}$$

- $S_{+,ijl}^z$  identify the region located on the plane  $z = X_{N_z^g}^{g,z}$  with (record INDX+Z):

$$\begin{aligned} X_{i-1}^{g,x} &< x < X_i^{g,x} \\ X_{j-1}^{g,y} &< y < X_j^{g,y} \\ R_{l-1}^{g,d} &< r < R_l^{g,d} \end{aligned}$$

- $S_{+,ijk}^r$  identify the region located on the plane  $r = X_{N_r^g}^{g,d}$  with (record INDX+R):

$$\begin{aligned} X_{i-1}^{g,x} &< x < X_i^{g,x} \\ X_{j-1}^{g,y} &< y < X_j^{g,y} \\ X_{k-1}^{g,z} &< z < X_k^{g,z} \end{aligned}$$

Note that for one can extract from geometry  $g = 1$ , the maximum extension of the geometry in all the directions, namely:

$$\begin{aligned} X_0^{1,x} &< x < X_{N_x^1}^{1,x} \\ X_0^{1,y} &< y < X_{N_y^1}^{1,y} \\ X_0^{1,z} &< z < X_{N_z^1}^{1,z} \\ R_0^{1,n} &< r < R_{N_r^1}^{1,n} \end{aligned}$$

Note that for the case where  $d = z$ , the external boundary will be cylindrical and the limiting condition on  $r$  will be

$$(x - C_1^{g,z})^2 + (y - C_2^{g,z})^2 = r < (R_{N_r^1}^{1,z})^2$$

for a  $z$  directed cylinder and with a similar logic for  $x$  and  $y$  directed cylinders.

Similarly, the maximum extension of each sub-geometry  $g$  will be:

$$\begin{aligned} X_0^{g,x} &< x < X_{N_x^g}^{g,x} \\ X_0^{g,y} &< y < X_{N_y^g}^{g,y} \\ X_0^{g,z} &< z < X_{N_z^g}^{g,z} \\ R_0^{g,d} &< r < R_{N_r^g}^{g,d} \end{aligned}$$

Now each tracking line in DRAGON is represented by a position vector  $\vec{r}_o = (r_o^x, r_o^y, r_o^z)$  through which the tracking line passes ( $r_o^x$ ,  $r_o^y$  and  $r_o^z$  are the  $x$ ,  $y$ , and  $z$  components of  $\vec{r}_o$ ) and a direction vector  $\vec{\mu} = (\mu^x, \mu^y, \mu^z)$  where  $\mu^x$ ,  $\mu^y$  and  $\mu^z$  are the cosine of the angle between the track and the  $x$ ,  $y$  and  $z$  axes respectively. Any point on this line can then be described in the following parametric form:

$$\vec{r} = r_o^x + l\vec{\mu}$$

where  $-\infty < l < \infty$ .

Since each region in space can be specified by a series of bounding surfaces, determining if a line crosses a region is equivalent to determining if a line intersect some of the surfaces bounding the region. In fact, for Cartesian, hexagonal and cylindrical regions either zero or two (one for the line entering the region and one for the line leaving the region) intersection between the tracking line and the bounding surfaces can be found. Now let us see how one can determine the location of these intersection points for various type of geometries.

### 4.3 Volume Evaluation

In the new version of DRAGON we have tackle these problems in the following way.

- For an assembly of Cartesian 3-D cells denoted  $(i, j, k)$  and located at:

$$x_{i-1} \leq x \leq x_i \quad y_{j-1} \leq y \leq y_j \quad z_{k-1} \leq z \leq z_k$$

the regional volumes are given by

$$V_{ijk} = (x_i - x_{i-1})(y_j - y_{j-1})(z_k - z_{k-1})$$

These volumes are computed using routine NXTVCA.

- For sets of  $x$ ,  $y$  or  $z$  directed concentric 3-D cylinders centered at  $(y_c, z_c)$ ,  $(z_c, x_c)$  and  $(x_c, y_c)$  respectively and denoted by  $(i, l)$ ,  $(j, l)$  or  $(k, l)$  we will have :

$$\begin{aligned} (i, l) &\rightarrow r_{l-1} \leq r \leq r_l & x_{i-1} &\leq x \leq x_i \\ (j, l) &\rightarrow r_{l-1} \leq r \leq r_l & y_{j-1} &\leq y \leq y_j \\ (k, l) &\rightarrow r_{l-1} \leq r \leq r_l & z_{k-1} &\leq z \leq z_k \end{aligned}$$

the volumes are given by

$$\begin{aligned} V_{il} &= \pi(r_l - r_{l-1})(r_l + r_{l-1})(x_i - x_{i-1}) \\ V_{jl} &= \pi(r_l - r_{l-1})(r_l + r_{l-1})(y_j - y_{j-1}) \\ V_{kl} &= \pi(r_l - r_{l-1})(r_l + r_{l-1})(z_k - z_{k-1}) \end{aligned}$$

This volume is computed using routine NXTVCY.

- For sets of  $x$ ,  $y$  or  $z$  directed concentric 3-D cylinders centered at  $(y_c, z_c)$ ,  $(z_c, x_c)$  and  $(x_c, y_c)$  respectively and denoted by  $(i, l)$ ,  $(j, l)$  or  $(k, l)$  overlapping a Cartesian mesh denoted by  $(i, j, k)$  the following volume evaluation procedure is considered in routine NXTVCC:

1. Call the routine NXTVCA to compute the volumes of the Cartesian regions.
2. For each concentric cylinders  $l$  starting with the outermost, find the volume of intersection  $\Delta v_l$  between a projection of the cylinder and of the 3-D Cartesian  $(i, j, k)$  in a 2-D plane normal to the cylinder axis. This volume intersection is computed using routine NXTIRA.
3. Remove  $\Delta v_l$  from  $V_{ijkl+1}$  and store in  $V_{ijkl}$ .
4. Return to 2 and repeat the process until  $\Delta v_l = 0$  or  $l = 1$ .

The volumes should satisfy:

$$V_{ijk} = \sum_{l=1}^{L+1} V_{ijkl}$$

- For an annular pin covering a set of concentric 3-D cylinders centered at  $(y_c, z_c)$ ,  $(z_c, x_c)$  and  $(x_c, y_c)$  respectively and denoted by  $(i, l)$ ,  $(j, l)$  or  $(k, l)$  overlapping a Cartesian mesh denoted by  $(i, j, k)$  the following volume evaluation procedure is considered in routine NXTPCC:

A 3-D Cartesian cell containing  $N$  concentric cylinders of radius  $r^n$  was first projected on a plane perpendicular to these cylinders. Then the resulting 2-D Cartesian cell was discretized according to the users specification and to each Cartesian sub-cell  $(i, j, k)$  located at

$$x_1^k \leq x \leq x_2^k \quad y_1^l \leq y \leq y_2^l$$

were associated  $(N + 1)$  2-D regions of identical volume  $V^{k,l} = (x_1^k - x_2^k)(y_1^l - y_2^l)$ , region 1 to  $N$  representing respectively the  $N$  concentric cylinders centered at  $(x_r, y_r)$ . Then starting with the most outer cylinder one determines whether the Cartesian region  $(k, l)$  is located totally outside or inside the specific cylinder or intersects it.

In the case where the Cartesian region is located outside the cylinder of outer radius  $r_n$ , the volume of region 1 to  $n$  vanishes identically. In the case where it is located inside the region of radius  $r_n$  then the volume of region  $n + 1$  to  $N + 1$  vanishes identically.

Finally when the Cartesian cell intersects the  $n^{\text{th}}$  cylinder, one can compute the volume of intersection  $\Delta V_n^{k,l}$  between the mesh  $(k, l)_n$  and the cylinder using the relation:

$$\Delta V_n^{k,l} = V_{2,2}^{k,l} - V_{2,1}^{k,l} - V_{1,2}^{k,l} + V_{1,1}^{k,l}$$

where the volume  $V_{i,j}^{k,l}$  represents the intersection between the cylinder and the plane located to the left of surface  $x_i^k$  and below  $y_j^l$ .

Here for simplicity we will compute  $V_{i,j}^{k,l}$  in terms of  $V_j^k$  which represents the cylinder surface located to the left of the plane defined by  $x_j^k$  and  $V_j^l$  which represents the cylinder surface located below the plane defined by surface  $y_j^l$ . These can be obtained using the relation

$$V_j = \begin{cases} 0 & \text{for } u_j < -r_n \\ \pi r_n^2 & \text{for } u_j > r_n \\ \alpha_j r_n^2 + u_j \sqrt{r_n^2 - (u_j)^2} & \text{otherwise} \end{cases}$$

$$\alpha_j = \arccos\left(-\frac{u_j}{r_n}\right)$$

where  $V_j^k = V_j$  when

$$u_j = x_j^k - x_r$$

is selected, while  $V_j^l = V_j$  if

$$u_j = y_j^l - y_r$$

is used.

In the case where the point of intersection of line  $x_i^k$  and  $y_j^l$  is located inside the cylinder of radius  $r_n$ , namely

$$u_{i,j}^{k,l} = \sqrt{(x_i^k - x_r)^2 + (y_j^l - y_r)^2} < r_n$$

$V_{i,j}^{k,l}$  is given by:

$$V_{i,j}^{k,l} = \frac{1}{2} (V_i^k - V_j^l) + u_i^k u_j^l + \frac{1}{4} \pi r_n^2$$

In all the other cases, depending on the location of the various planes with respect to the center of the cylinder we will use:

$$V_{i,j}^{k,l} = \begin{cases} 0 & \text{if } u_i^k < 0 \text{ and } u_j^l < 0 \\ V_j^l & \text{if } u_i^k < 0 \text{ and } u_j^l > 0 \\ V_i^k & \text{if } u_i^k > 0 \text{ and } u_j^l < 0 \\ V_i^k + V_j^l - \pi r_n^2 & \text{if } u_i^k > 0 \text{ and } u_j^l > 0 \end{cases}$$

Once  $\Delta V_n^{k,l}$  has been computed, we can redefine the volume of regions  $n + 1$  and  $n$  as:

$$\begin{aligned} V_{n+1} &= V_{n+1} - \Delta V_n^{k,l} \\ V_n &= \Delta V_n^{k,l} \end{aligned}$$

and proceed to cylinder  $n - 1$ .

Once all the Cartesian regions associated with a specific 2-D mapping of the 3-D cell have been processed in the same way, the next step will consist in extracting from the set of  $K \times L \times (N + 1)$  regions those which will have a non-vanishing 2-D volume. Then the explicit volume of each region in a 3-D plane will be obtained using:

$$V_n^m = V_n(z_2^m - z_1^m)$$

assuming this plane is located at

$$z_1^m \leq z \leq z_2^m$$

We also associate to each region a 4 dimensional pointer  $I(x, y, z, r)$  which identifies the location of this region inside the  $(x, y, z, r)$  mesh assuming that for regions located inside a Cartesian mesh but outside the cylinders the radial pointer will be set to 0.

After each region has been identified in this fashion, the explicit tracking procedure can begin. Because DRAGON already embodied a quite general tracking procedure the implementation of the new geometric options was quite simple. Each tracking line is first defined by a direction  $\vec{\Omega}$  and a starting point  $(x_s, y_s, z_s)$  located outside the 3-D assembly. Then for each of the Cartesian directions, one locates the 3-D intersection point  $(x_i, y_i, z_i)$  between the integration line and the various Cartesian planes perpendicular to the specific direction the line may encounter. Similarly, for each possible cylinder in the assembly one determines if it can intersect the integration line and the two locations at which this intersection occurs  $(x_{\pm}, y_{\pm}, z_{\pm})$ . The distance between the starting point and each intersection point is then computed using

$$D_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2}$$

and the  $D_i$  are classified by increasing value. It is then simple to identify each track segment with a specific region number using the 4 dimensional pointer  $I(x, y, z, r)$  and to generate a DRAGON integration line.

#### 4.4 Generation of the Integration Lines

##### 4.4.1 Cartesian Geometry

We will first consider the 3-D case where each region is defined by the following 6 surfaces:

$$r_1^x = X_0^{g,x} \quad \text{with} \quad X_0^{g,y} < r_1^y < X_{N_y^g}^{g,y} \quad \text{and} \quad X_0^{g,z} < r_1^z < X_{N_z^g}^{g,z} \quad (4.7)$$

$$r_2^x = X_{N_x^g}^{g,x} \quad \text{with} \quad X_0^{g,y} < r_2^y < X_{N_y^g}^{g,y} \quad \text{and} \quad X_0^{g,z} < r_2^z < X_{N_z^g}^{g,z} \quad (4.8)$$

$$r_3^y = X_0^{g,y} \quad \text{with} \quad X_0^{g,x} < r_3^x < X_{N_x^g}^{g,x} \quad \text{and} \quad X_0^{g,z} < r_3^z < X_{N_z^g}^{g,z} \quad (4.9)$$

$$r_4^y = X_{N_y^g}^{g,y} \quad \text{with} \quad X_0^{g,x} < r_4^x < X_{N_x^g}^{g,x} \quad \text{and} \quad X_0^{g,z} < r_4^z < X_{N_z^g}^{g,z} \quad (4.10)$$

$$r_5^z = X_0^{g,z} \quad \text{with} \quad X_0^{g,x} < r_5^x < X_{N_x^g}^{g,x} \quad \text{and} \quad X_0^{g,y} < r_5^y < X_{N_y^g}^{g,y} \quad (4.11)$$

$$r_6^z = X_{N_z^g}^{g,z} \quad \text{with} \quad X_0^{g,x} < r_6^x < X_{N_x^g}^{g,x} \quad \text{and} \quad X_0^{g,y} < r_6^y < X_{N_y^g}^{g,y} \quad (4.12)$$

The first step is to find the value of  $l$  associated with each face using:

$$\begin{aligned} l_1^x &= (r_1^x - r_o^x) / \mu^x & l_2^x &= (r_2^x - r_o^x) / \mu^x \\ l_3^y &= (r_3^y - r_o^y) / \mu^y & l_4^y &= (r_4^y - r_o^y) / \mu^y \\ l_5^z &= (r_5^z - r_o^z) / \mu^z & l_6^z &= (r_6^z - r_o^z) / \mu^z \end{aligned}$$

The next step is to locate the other two components of the vector  $r$  associated with the distances  $l_i^j$  defined above:

$$\begin{aligned} r_1^y &= l_1^x \mu^y + r_o^y & r_1^z &= l_1^x \mu^z + r_o^z & r_2^y &= l_2^x \mu^y + r_o^y & r_2^z &= l_2^x \mu^z + r_o^z \\ r_3^z &= l_3^y \mu^z + r_o^z & r_3^x &= l_3^y \mu^x + r_o^x & r_4^z &= l_4^y \mu^z + r_o^z & r_4^x &= l_4^y \mu^x + r_o^x \\ r_5^x &= l_5^z \mu^x + r_o^x & r_5^y &= l_5^z \mu^y + r_o^y & r_6^x &= l_6^z \mu^x + r_o^x & r_6^y &= l_6^z \mu^y + r_o^y \end{aligned}$$

and to test if these points can be located can be located on one of the six faces of the parallelepiped. In order for a line to cross a region, or two surfaces, at least two of the six set of relations (Eqs. (4.7) to (4.12)) must be satisfied. One can also classify the surfaces by looking at the distance travelled along the  $\mu$  direction, namely  $l_j^d$ , and ordering them in a monotonically increasing order.

For the 2-D case, we will consider only the following 4 relations

$$r_1^x = X_0^{g,x} \quad \text{with} \quad X_0^{g,y} < r_1^y < X_{N_y^g}^{g,y} \quad (4.13)$$

$$r_2^x = X_{N_x^g}^{g,x} \quad \text{with} \quad X_0^{g,y} < r_2^y < X_{N_y^g}^{g,y} \quad (4.14)$$

$$r_3^y = X_0^{g,y} \quad \text{with} \quad X_0^{g,x} < r_3^x < X_{N_x^g}^{g,x} \quad (4.15)$$

$$r_4^y = X_{N_y^g}^{g,y} \quad \text{with} \quad X_0^{g,x} < r_4^x < X_{N_x^g}^{g,x} \quad (4.16)$$

which are required to delimit a region. The first step is to find the value of  $l$  associated with each face using:

$$\begin{aligned} l_1^x &= (r_1^x - r_o^x) / \mu^x & l_2^x &= (r_2^x - r_o^x) / \mu^x \\ l_3^y &= (r_3^y - r_o^y) / \mu^y & l_4^y &= (r_4^y - r_o^y) / \mu^y \end{aligned}$$

The next step is to evaluate the other component of the vector  $r$  (2-D vector) associated with the distances  $l_i^j$ :

$$\begin{aligned} r_1^y &= l_1^x \mu^y + r_o^y & r_2^y &= l_2^x \mu^y + r_o^y \\ r_3^x &= l_3^y \mu^x + r_o^x & r_4^x &= l_4^y \mu^x + r_o^x \end{aligned}$$

In order for a line to cross a region, or 2 surfaces, at least two of the four relations (Eqs. (4.13) to (4.16)) must be satisfied. Finally, one can classify the surfaces by looking at the distance travelled along the  $\mu$  direction, namely  $l_j^d$ , and ordering them in a monotonically increasing order.

#### 4.4.2 Cylindrical Geometry

Here we will consider independently the three types of 3-D cylinders that are permitted in DRAGON:

1. A cylinder with axis in the  $x$  direction and bounding surfaces:

$$(r_1^y - C_1^{g,x})^2 + (r_1^z - C_2^{g,x})^2 = (R_{N_r^1}^{1,x})^2 \quad \text{with} \quad X_0^{g,x} < r_1^x < X_{N_x^g}^{g,x} \quad (4.17)$$

$$r_2^x = X_0^{g,x} \quad \text{with} \quad (r_2^y - C_1^{g,x})^2 + (r_2^z - C_2^{g,x})^2 < (R_{N_r^1}^{1,x})^2 \quad (4.18)$$

$$r_3^x = X_{N_x^g}^{g,x} \quad \text{with} \quad (r_3^y - C_1^{g,x})^2 + (r_3^z - C_2^{g,x})^2 < (R_{N_r^1}^{1,x})^2 \quad (4.19)$$

First we will evaluate the distance  $l_i^x$  along the cylinder axis from our starting point to one of the bottom ( $r_2^x$ ) or top ( $r_3^x$ ) faces of the cylinder.

$$l_2^x = (r_2^x - r_o^x) / \mu^x \quad l_3^x = (r_3^x - r_o^x) / \mu^x$$

which can be used to locate completely the intersection between the tracking line with an infinite plane perpendicular to this direction.

$$\begin{aligned} (r_2^x, r_2^y, r_2^z) &= (r_2^x, l_2^x \mu^y + r_o^y, l_2^x \mu^z + r_o^z) \\ (r_3^x, r_3^y, r_3^z) &= (r_3^x, l_3^x \mu^y + r_o^y, l_3^x \mu^z + r_o^z) \end{aligned}$$

The line will cross the bottom ( $i = 2$ ) or the top ( $i = 3$ ) of the cylinder if:

$$(r_i^y - C_1^{g,x})^2 + (r_i^z - C_2^{g,x})^2 < (R_{N_r^1}^{1,x})^2$$



The second step consist in locating, if possible, an intersection between the tracking line and a cylinder of infinite extension in the  $x$  direction. The problem can be simplified if one considers only the projection of the cylinder in the  $y, z$  plane. If one defines  $\varphi_{\perp}$  as the angle between a projection of the line in the  $y, z$  plane and the  $y$  axis we will have:

$$\varphi_{\perp} = \arccos(\mu_y / \sqrt{1 - (\mu_x)^2})$$

and we can now define  $\tilde{\mu}_y$  and  $\tilde{\mu}_z$  as follows:

$$\begin{aligned}\tilde{\mu}_y &= \mu_y / \sqrt{1 - (\mu_x)^2} \\ \tilde{\mu}_z &= \mu_z / \sqrt{1 - (\mu_x)^2}\end{aligned}$$

Now we need to determine if a line with direction cosines  $(\tilde{\mu}_y, \tilde{\mu}_z)$  passing by the point  $(r_o^y, r_o^z)$  can cross the circle of radius  $R$  centered at  $(C_1^{g,x}, C_2^{g,x})$ . The simplest way to deal with this problem is to use 2 successive change of coordinates:

- Translate the axis in such a way that the origin of our system of coordinates is the center of the circle:

$$\begin{aligned}(\tilde{r}_o^y, \tilde{r}_o^z) &= (r_o^y, r_o^z) - (C_1^{g,x}, C_2^{g,x}) \\ (\tilde{C}_1^{g,x}, \tilde{C}_2^{g,x}) &= (0, 0)\end{aligned}$$

- Rotate the axis in such a way that the tracking line is parallel to the second axis:

$$\begin{pmatrix} \hat{r}_o^y \\ \hat{r}_o^z \end{pmatrix} = \begin{pmatrix} \tilde{\mu}_y & -\tilde{\mu}_z \\ \tilde{\mu}_z & \tilde{\mu}_y \end{pmatrix} \begin{pmatrix} \tilde{r}_o^y \\ \tilde{r}_o^z \end{pmatrix}$$

- The line will intersect the circle only if  $-R \leq \hat{r}_o^z \leq R$ , in which case

$$\hat{r}_i^y = \pm \sqrt{R^2 - (\hat{r}_o^z)^2}$$

- Rotate and translate back to the original system of coordinates:

$$\begin{pmatrix} r_i^y \\ r_i^z \end{pmatrix} = \begin{pmatrix} \tilde{\mu}_y & \tilde{\mu}_z \\ -\tilde{\mu}_z & \tilde{\mu}_y \end{pmatrix} \begin{pmatrix} \hat{r}_i^y \\ \hat{r}_o^z \end{pmatrix} + \begin{pmatrix} C_1^{g,x} \\ C_2^{g,x} \end{pmatrix}$$

- Find the value of  $r_i^x$  associated with this point:

$$r_i^x = r_i^z \frac{\mu_x}{\mu_y}$$

- The cylinder is crossed in the cases where:

$$X_0^{g,x} < r_i^x < X_{N_z^g}^{g,x}$$

2. A cylinder with axis in the  $y$  direction and bounding surfaces:

$$(r_1^z - C_1^{g,y})^2 + (r_1^x - C_2^{g,y})^2 = (R_{N_r^1}^{1,y})^2 \quad \text{with} \quad X_0^{g,y} < r_1^y < X_{N_y^g}^{g,y} \quad (4.20)$$

$$r_2^y = X_0^{g,y} \quad \text{with} \quad (r_2^z - C_1^{g,y})^2 + (r_2^x - C_2^{g,y})^2 < (R_{N_r^1}^{1,y})^2 \quad (4.21)$$

$$r_3^y = X_{N_z^g}^{g,y} \quad \text{with} \quad (r_3^z - C_1^{g,y})^2 + (r_3^x - C_2^{g,y})^2 < (R_{N_r^1}^{1,y})^2 \quad (4.22)$$

First we will evaluate the distance  $l_i^x$  along the cylinder axis from our starting point to one of the bottom ( $r_2^x$ ) or top ( $r_3^x$ ) faces of the cylinder.

$$l_2^x = (r_2^x - r_o^x) / \mu^x \quad l_3^x = (r_3^x - r_o^x) / \mu^x$$

which can be used to locate completely the intersection between the tracking line with an infinite plane perpendicular to direction this direction.

$$\begin{aligned}(r_2^x, r_2^y, r_2^z) &= (r_2^x, l_2^x \mu^y + r_o^y, l_2^x \mu^z + r_o^z) \\ (r_3^x, r_3^y, r_3^z) &= (r_3^x, l_3^x \mu^y + r_o^y, l_3^x \mu^z + r_o^z)\end{aligned}$$

The line will cross the bottom ( $i = 2$ ) or the top ( $i = 3$ ) of the cylinder if:

$$(r_i^y - C_1^{g,x})^2 + (r_i^z - C_2^{g,x})^2 < (R_{N_r}^{1,x})^2$$

3. A cylinder with axis in the  $z$  direction and bounding surfaces:

$$(r_1^x - C_1^{g,z})^2 + (r_1^y - C_2^{g,z})^2 = (R_{N_r}^{1,z})^2 \quad \text{with} \quad X_0^{g,z} < r_1^z < X_{N_g}^{g,z} \quad (4.23)$$

$$r_2^z = X_0^{g,z} \quad \text{with} \quad (r_2^x - C_1^{g,z})^2 + (r_2^y - C_2^{g,z})^2 < (R_{N_r}^{1,z})^2 \quad (4.24)$$

$$r_3^z = X_{N_g}^{g,z} \quad \text{with} \quad (r_3^x - C_1^{g,z})^2 + (r_3^y - C_2^{g,z})^2 < (R_{N_r}^{1,z})^2 \quad (4.25)$$

First we will evaluate the distance  $l_i^x$  along the cylinder axis from our starting point to one of the bottom ( $r_2^x$ ) or top ( $r_3^x$ ) faces of the cylinder.

$$l_2^x = (r_2^x - r_o^x) / \mu^x \quad l_3^x = (r_3^x - r_o^x) / \mu^x$$

which can be used to locate completely the intersection between the tracking line with an infinite plane perpendicular to direction this direction.

$$\begin{aligned}(r_2^x, r_2^y, r_2^z) &= (r_2^x, l_2^x \mu^y + r_o^y, l_2^x \mu^z + r_o^z) \\ (r_3^x, r_3^y, r_3^z) &= (r_3^x, l_3^x \mu^y + r_o^y, l_3^x \mu^z + r_o^z)\end{aligned}$$

The line will cross the bottom ( $i = 2$ ) or the top ( $i = 3$ ) of the cylinder if:

$$(r_i^y - C_1^{g,x})^2 + (r_i^z - C_2^{g,x})^2 < (R_{N_r}^{1,x})^2$$

The first step consist of evaluating the distance  $l_i^j$  along the cylinder axis with direction  $j$  from our starting point to one of the bottom ( $r_2^j$ ) or top ( $r_3^j$ ) faces of the cylinder.

$$\begin{aligned}l_2^x &= (r_2^x - r_o^x) / \mu^x & l_3^x &= (r_3^x - r_o^x) / \mu^x \\ l_2^y &= (r_2^y - r_o^y) / \mu^y & l_3^y &= (r_3^y - r_o^y) / \mu^y \\ l_2^z &= (r_2^z - r_o^z) / \mu^z & l_3^z &= (r_3^z - r_o^z) / \mu^z\end{aligned}$$

which can be used to locate completely the intersection the tracking line with an infinite plane perpendicular to direction  $j$ .

$$\begin{aligned}(r_2^x, l_2^x \mu^y + r_o^y, l_2^x \mu^z + r_o^z) & \quad (r_3^x, l_3^x \mu^y + r_o^y, l_3^x \mu^z + r_o^z) \\ (l_2^y \mu^x + r_o^x, r_2^y, l_2^y \mu^z + r_o^z) & \quad (l_3^y \mu^x + r_o^x, r_3^y, l_3^y \mu^z + r_o^z) \\ (l_2^z \mu^x + r_o^x, l_2^z \mu^y + r_o^y, r_2^z) & \quad (l_3^z \mu^x + r_o^x, l_3^z \mu^y + r_o^y, r_3^z)\end{aligned}$$

## REFERENCES

- [1] G. Marleau, R. Roy, and A. Hébert, *A User Guide for DRAGON 3.06*, Technical Report IGE-174 Rev. 7, École Polytechnique de Montréal (2007).
- [2] A. Hébert, G. Marleau, and R. Roy, *A Description of the Data Structures for DRAGON 3.06*, Technical Report IGE-232 Rev. 5, École Polytechnique de Montréal (2008).
- [3] G. Marleau, *DRAGON Theory Manual Part 1: Collision Probability Calculations*, Technical Report IGE-236 Rev. 1, École Polytechnique de Montréal (2001).
- [4] G. Marleau, *DRAGON Programmer's Manual*, Technical Report IGE-251, École Polytechnique de Montréal (2001).

# APPENDICES

## APPENDIX A

### ANALYTICAL VOLUME EVALUATION

figures/ Here we will only be dealing with combinations of 3-D rectangles and 3-D cylinders or 2-D rectangles and circles. One restriction in DRAGON is that only 3-D cylinders within a cell can intersect and that these cylinders must all have parallel axes that are co-linear with the  $x$ ,  $y$  or  $z$  directions. One consequence of this observation is that all the volume for 3-D geometry can be computed using the surfaces associated with a 2-D projection of the cell along the axis of the cylinders multiplied by the height of the cell or cylinder. Accordingly, we will concentrate on the problem of evaluating the surfaces associated with each region possible when rectangles and circles intersect with a maximum of 2 non-concentric circles being permitted in DRAGON.

#### A.1 Rectangle

A 2-D Cartesian cell is identified by a point  $(x, y)$  that satisfies (see Figure 17):

$$x_1 \leq x \leq x_2 \quad y_1 \leq y \leq y_2 \quad (\text{A.1})$$

where  $x_1$  and  $x_2$  are the locations with respect to the origin of the left and right boundary while  $y_1$  and  $y_2$  are the locations with respect to the origin of the bottom and top boundary. The 2-D volume (3-D surface) of this rectangle is given by:

$$V^C = (x_2 - x_1)(y_2 - y_1)$$

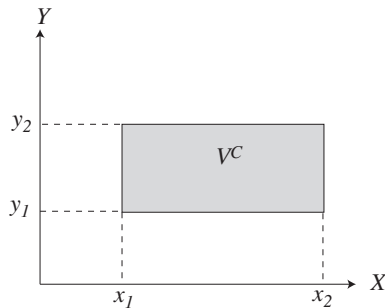


Figure 17: Volume of a rectangle

#### A.2 Concentric Circles

Concentric annular rings are identified by the position of their center  $(R, \theta)$  with respect to the origin and the inner  $r_1$  and outer radius  $r_2$  of the rings (see Figure 18)

$$r_1 \leq r \leq r_2$$

The 2-D volume of this ring is:

$$V^R = \pi(r_2^2 - r_1^2)$$

The volume of a circle of radius  $r$  can be obtained from the above using  $r_2 = r$  and  $r_1 = 0$ .

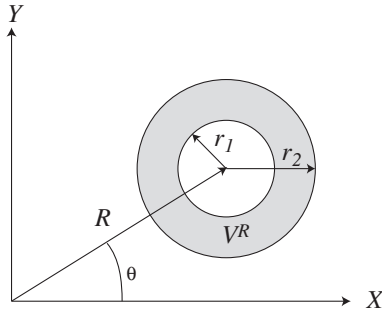


Figure 18: Volume of concentric annular circles

### A.3 Overlapping Circles

The 2-D volume  $V^A$  associated with the intersection of two non-concentric circles overlapping is somewhat more difficult to obtain. Here we will assume for simplicity that each circle  $i$  ( $i = 1, 2$ ) is identified by  $(R_i, \theta_i, r_i)$  as illustrated in Figure 19. The first step is then to locate the center  $(x_{i,c}, y_{i,c})$  of each circle  $i$

$$(x_{i,c}, y_{i,c}) = (R_i \cos \theta_i, R_i \sin \theta_i)$$

the equation for the line delimiting each circle being

$$(x - x_{i,c})^2 + (y - y_{i,c})^2 = r_i^2$$

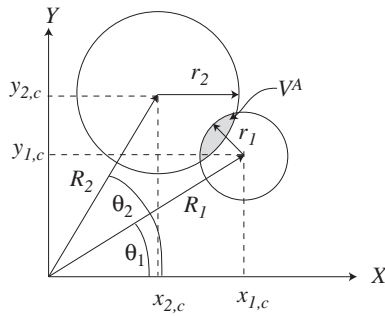


Figure 19: Volume of overlapping circles

The first step in evaluating the surface of intersection between circle 1 and 2 is to find the intersection points, if any between the two lines defining the circle. This can be readily obtained in the following way. First, one displaces the system origin in such a way that it is located at the center of circle 1 ( $x \rightarrow x + x_{1,c}$  and  $y \rightarrow y + y_{1,c}$ ) (see Figure 20). The equations for the line defining each circle in the translated system of reference is

$$\begin{aligned} x^2 + y^2 &= r_1^2 \\ (x - x_{2,d})^2 + (y - y_{2,d})^2 &= (r_2)^2 \end{aligned}$$

with

$$\begin{aligned} x_{2,d} &= (x_{2,c} - x_{1,c}) \\ y_{2,d} &= (y_{2,c} - y_{1,c}) \end{aligned}$$

Every point on the first circle must then satisfy:

$$y^2 = r_1^2 - x^2 \tag{A.2}$$

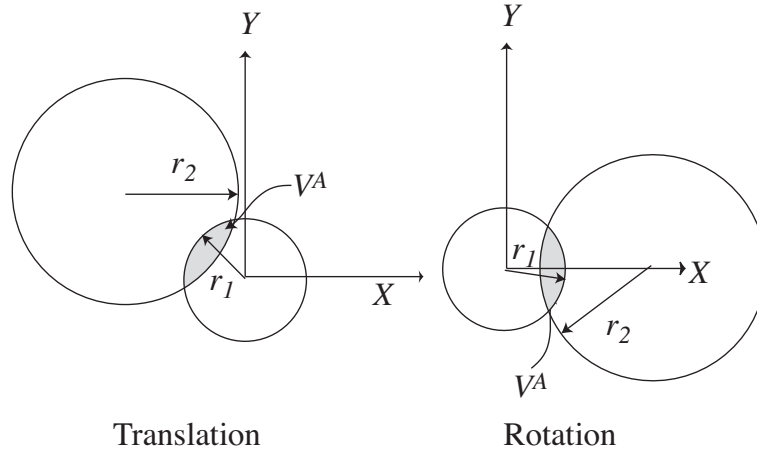


Figure 20: Translation and rotation of overlapping circles

this equation being invariant under rotation of the axis with respect to the origin of circle 1. The second step consists in rotating the axis in such a way that the center of the circle 2 is located on the  $x$  axis (see Figure 20). Since circle 2 is located at position  $(x_{2,d}, y_{2,d})$  with respect to the origin of circle 1, we will have:

$$R_{2,d} = \sqrt{(x_{2,d})^2 + (y_{2,d})^2}$$

$$\theta_{2,d} = \arctan\left(\frac{y_{2,d}}{x_{2,d}}\right)$$

Note that for  $|y_{2,d}/x_{2,d}| > 1$ , one generally uses

$$\theta_{2,d} = \operatorname{arccotan}\left(\frac{x_{2,d}}{y_{2,d}}\right)$$

for increased precision when  $x_{2,d}$  becomes small with respect to  $y_{2,d}$ .

After a rotation of the axis by  $-\theta_{2,d}$ , the equation for circle 2 becomes:

$$(x - R_{2,d})^2 + y^2 = r_2^2 \quad (\text{A.3})$$

Circles 1 and 2 then intersect only if

$$R_{2,d} < r_1 + r_2$$

Using Eq. (A.2) in Eq. (A.3) one obtains

$$(x - R_{2,d})^2 + r_1^2 - x^2 = r_2^2$$

which has for solution:

$$x_s = \frac{R_{2,d}^2 - r_2^2 - r_1^2}{2R_{2,d}} \quad (\text{A.4})$$

The intersection points are then given by  $(x_s, -y_s)$  and  $(x_s, y_s)$  with

$$y_s = \sqrt{r_1^2 - x_s^2} \quad (\text{A.5})$$

Now for circle 1, these two points cover an angular sector of width

$$\alpha_1 = 2 \arctan \left( \frac{y_s}{x_s} \right)$$

while for circle 2 the angular sector covered is

$$\alpha_2 = 2 \arctan \left( \frac{y_s}{x_s - R_{2,d}} \right)$$

still being careful to use the arccotan instead when either  $x_s$  or  $x_s - R_{2,d}$  are much smaller than  $y_s$ .

Using this information, the 2-D volume of the intersection region can be computed in the following way. It is the sum the part of the angular sector of circle 1 to the left of the line define by  $x_s$  given by:

$$V^{1,l} = \frac{\alpha_1}{2} r_1^2 - x_s y_s$$

and the part of the angular sector of circle 2 to the right of the line define by  $x_s$ :

$$V^{2,r} = \pi r_2^2 - \left( \frac{\alpha_2}{2} r_2^2 - (x_s - R_{2,d}) y_s \right)$$

for a total of

$$V^A = V^{1,l} + V^{2,r}$$

#### A.4 Circles Overlapping a Rectangle

First let us consider the intersection of a rectangle defined as in Eq. (A.1) and a circle defined by  $(R, \theta, r)$  or  $(x_c, y_c, r)$  as illustrated in Figure 21:

$$(x_c, y_c) = (R \cos \theta, R \sin \theta)$$

The first step here consists of determining whether the rectangle is located totally outside or inside the circle or intersects it.

In the case where the rectangle is totally located outside the cylinder the volume of intersection of these two regions vanishes identically. If the Cartesian region is an outer cell, the surface area of each face is just the length of the lines composing the sides of the Cartesian cell. In the case where the rectangle is located totally inside the circle, the volume of intersection is  $V^C$  while if the circle is located totally inside the circle the volume of intersection is  $V^R$  and the surface area is the length of the circle, namely  $2\pi r$  (for a circle corresponding to an outer boundary). Finally when some of the faces defining the rectangle intersect the circle, one can compute the volume of intersection  $V^O$  between the rectangle and the circle using

$$V^O = V_{2,2} - V_{2,1} - V_{1,2} + V_{1,1}$$

where the volume  $V_{i,j}$  represents the intersection between the cylinder and the plane located to the left of surface  $x_i$  and below  $y_j$  (see Figure 22 for example).

Here for simplicity we will compute  $V_{i,j}$  in terms of  $V_j$  which represents either the cylinder surface located to the left of the plane defined by  $x_j$  or the cylinder surface located below the plane defined by surface  $y_j$ . These can be obtained using the relation

$$V_j = \begin{cases} 0 & \text{for } u_j < -r \\ \pi r^2 & \text{for } u_j > r \\ \alpha_j r^2 + u_j \sqrt{r^2 - (u_j)^2} & \text{otherwise} \end{cases}$$

$$\alpha_j = \arccos \left( -\frac{u_j}{r} \right)$$



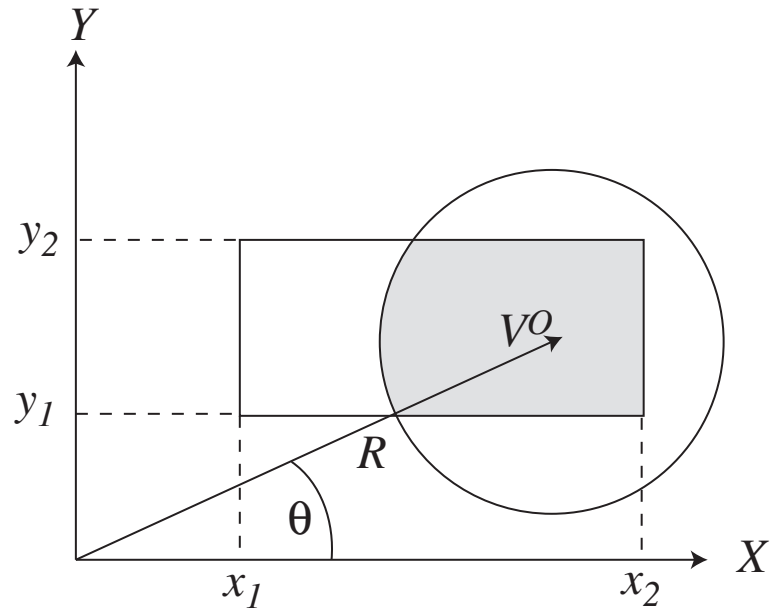


Figure 21: Volume for the overlap of a rectangle with an annular region

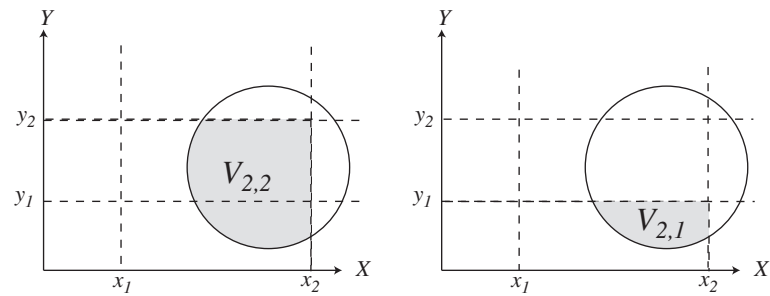


Figure 22: Decomposition of a rectangle circle region overlap

where

$$u_j = x_j - x_c$$

for the surface to the left of  $x_j$  or

$$u_j = y_j - y_c$$

for the surface below  $y_j$ .

In the case where the point of intersection of lines  $x_i$  and  $y_j$  is located inside the cylinder of radius  $r$ , namely

$$u_{i,j} = \sqrt{(x_i - x_c)^2 + (y_j - y_c)^2} < r$$

$V_{i,j}$  is given by:

$$V_{i,j} = \frac{1}{2} (V_i - V_j) + u_i u_j + \frac{1}{4} \pi r^2$$

For all the other cases, depending on the location of the various planes with respect to the center of the cylinder we will use:

$$V_{i,j} = \begin{cases} 0 & \text{if } u_i < 0 \text{ and } u_j < 0 \\ V_j & \text{if } u_i < 0 \text{ and } u_j > 0 \\ V_i & \text{if } u_i > 0 \text{ and } u_j < 0 \\ V_i + V_j - \pi r^2 & \text{if } u_i > 0 \text{ and } u_j > 0 \end{cases}$$

When Cartesian regions intersect annular cell, only the surfaces associated with the annular boundary can be outer boundaries (TUBE). In this case, the area correspond to

$$S_{\text{radial}} = r(\theta_{\text{max}} - \theta_{\text{min}})$$

where  $\theta_{\text{max}}$  is the maximum angle covered by the circular arc and  $\theta_{\text{min}}$  the minimum angle.

Finally, for the case where a rectangle is intersected with 2 non concentric annular regions as illustrated in Figure 23 the volume evaluation process to be considered is a combination of rectangle/circles intersections (see above) and circle/circle intersections (see Appendix A.3).

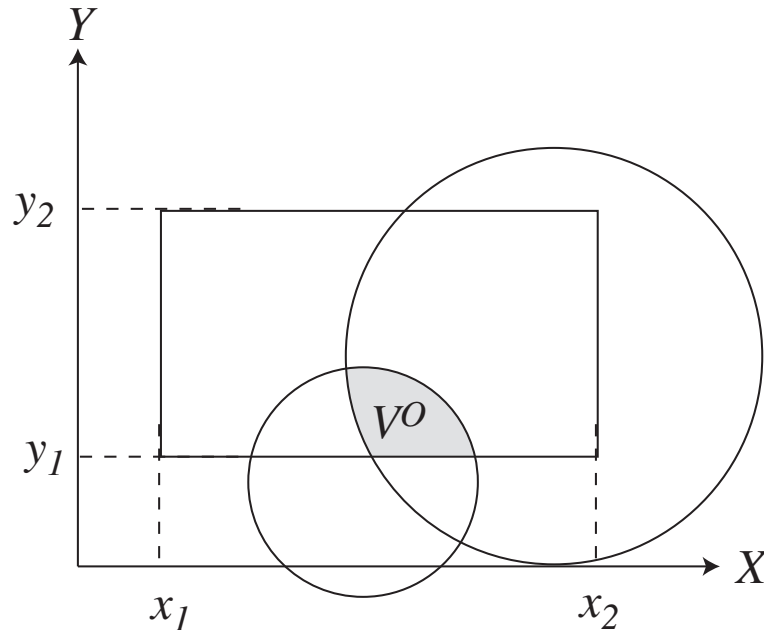


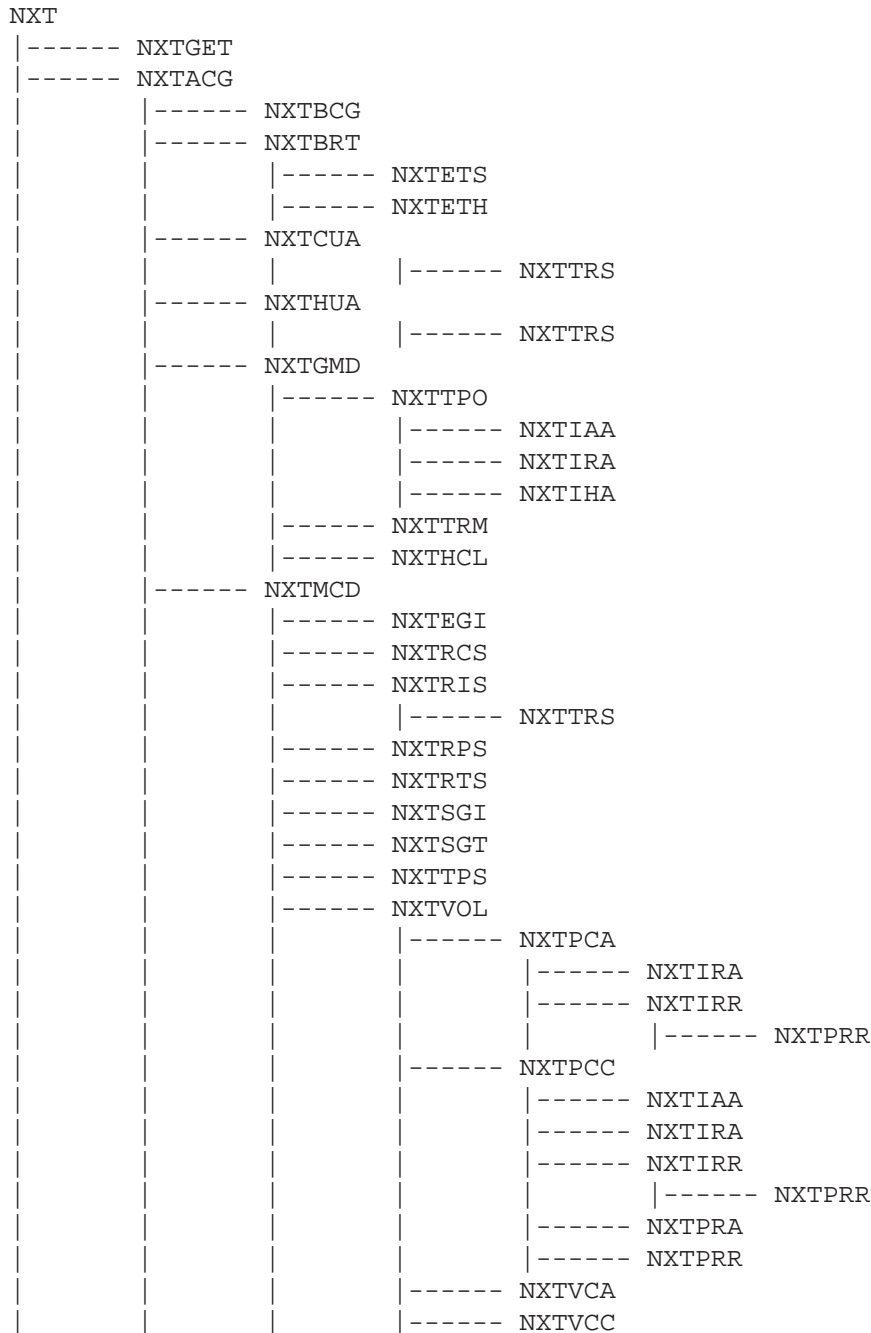
Figure 23: Volume for the overlapp of a rectangle with two non concentric annular regions

## APPENDIX B

### PROGRAMMING GUIDE FOR THE NXT MODULE

#### B.1 Structure of the program

Structure of the NXT tracking driver: STRNXT.tex



```

----- NXTVCA
----- NXTIRA
----- NXTTLO
----- NXTPHC
----- NXTIAA
----- NXTITA
----- NXTPRA
----- NXTVHT
----- NXTPHT
----- NXTVHC
----- NXTCVS
----- NXTAVS
----- NXTAGM
----- NXTTCG
----- NXTXYZ
----- NXTQAC
----- XELTSA
----- XELTSW
----- NXTQAS
----- NXTQEW
----- NXTQLC
----- NXTQLT
----- NXTLSN
----- NXTQRN
----- NXTQSC
----- NXTQSS
----- NXTTLC
----- NXTLCA
----- NXTLHA
----- NXTTCR
----- NXTLCA
----- NXTLCU
----- NXTLCY
----- NXTLHA
----- NXTLHT
----- NXTLRH
----- NXTLRS
----- NXTRTL
----- NXTTLS
----- NXTLCA
----- NXTLHA
----- NXTQPS
----- NXTTCR
----- NXTLCA
----- NXTLCU
----- NXTLCY
----- NXTLHA
----- NXTLHT
----- NXTLRH
----- NXTLRS
----- NXTRTL
----- NXTCVM

```

```

|          |----- NXTSQD
|          |----- NXTTNS

```

Structure of EXCELP inline driver for NXT: STRNXTEXCELP.tex

```

EXCELP
|----- NXTTGC
|          |----- NXTLCA
|          |----- NXTTCR
|----- NXTTGS
|          |----- NXTLCA
|          |----- NXTTCR

```

## B.2 Use of each routine

The NXT : routines perform the following tasks:

- NXT is the main subroutine of the module NXT.
- NXTACG is the main routine to analyze a geometry.
- NXTAGM applies general merge vector to geometry and creates the MATCOD, VOLUME and KEYFLX vectors.
- NXTAVS adds current cell information to global surfaces and volumes for geometry.
- NXTBCG reads boundary conditions and symmetries and verify for consistency.
- NXTBRT builds surface reflection/transmission coupling array.
- NXTCUA creates the array for testing the geometry in a Cartesian assembly for internal symmetries and unfolding the assembly according to these symmetries.
- NXTCVM compresses arrays VOLSUR and MATALB according to KEYMRG and save on tracking file.
- NXTCVS computes final volumes and surfaces.
- NXTEGI extracts cell or pin geometry information.
- NXTETH builds equivalent surface array for translational symmetry in triangular hexagons.
- NXTETS builds equivalent surface array for translational symmetry in Cartesian geometries.
- NXTGET reads NXT : input data.
- NXTGMD evaluates global mesh for assembly.
- NPTHCL locate spatial position of hexagon in assembly of cells.
- NXTHUA creates the array for testing the geometry in an hexagonal assembly for internal symmetries and unfolding the assembly according to these symmetries.
- NXTIAA computes the volume of intersection between two TUBE.
- NXTIHA finds the intersection between an hexagonal region and an annular pin.
- NXTIRA finds the intersection between a rectangular region and an annular pin.
- NXTIRR finds intersection between a rectangular region and a Cartesian pin.

- NXTITA compute the volume of intersection between a 2D triangle and an annular pin.
- NXTLCA tracks a Cartesian 2-D or 3-D geometry.
- NXTLCU merges two sets of tracks.
- NXTLCY tracks an annular geometry.
- NXTLHA tracks an hexagonal assembly in 2-D or 3-D geometry.
- NXTLHT track a triangular hexagon in 2-D or 3-D geometry.
- NXTLRH stores line segments in tracking vector for hexagonal geometry with global region and surface identification.
- NXTLRS stores line segments in tracking vector with global region and surface identification.
- NXTLSN defines level-symmetric (type 2) quadrature angles.
- NXTMCD creates a multicell description for the geometry.
- NXTPCA removes volumes or surfaces of the overlapping pins.
- NXTPCC removes from the volumes or surfaces associated with an annular/hexagonal 2-D or 3-D geometry the volumes or surfaces of the overlapping pins.
- NXTPHC removes from the volumes or surfaces associated with an annular/hexagonal 2-D or 3-D geometry the volumes or surfaces of the overlapping pins.
- NXTPHT Remove from the volumes or surfaces associated with a hexagonal 2D or 3D geometry the volumes or surfaces of the overlapping pins.
- NXTPRA compute the volume of intersection between a CARCEL and a TUBE centered at the origin.
- NXTPRR finds rectangle representing the intersection of two rectangles.
- NXTQAC defines quadrature angles for cyclic tracking.
- NXTQAS defines quadrature angles for a given standard tracking option.
- NXTQEW defines  $EQ_N$  quadrature angles.
- NXTQLC defines Legendre-Chebyshev quadrature angles.
- NXTQLT defines Sanchez-Mao-Santandrea (Legendre-Trapezoidal) quadrature angles.
- NXTQPS generates directions defining the planes normal to a solid angle.
- NXTQRN defines quadruple range (QR) quadrature angles.
- NXTQSC defines spatial quadrature for cyclic tracking.
- NXTQSS defines standard spatial quadrature.
- NXTRCS renumbers cell surfaces.
- NXTRIS rotates geometry according to reference turn.
- NXTRPS renumbers pin cluster surfaces.
- NXTRTS rotate hexagon with triangles according to reference turn and test, and verify that it satisfies intrinsic symmetries.

- NXTRTL rotates tracking line according to reference turn.
- NXTSGI discretizes geometry according to splitting options for Cartesian geometries.
- NXTSGT discretizes geometry according to splitting options for hexagonal geometries.
- NXTSQD saves double precision quadrature parameters on tracking file.
- NXTTCG is the main routine to track a geometry.
- NXTTCR tracks a cell rotated according to its explicit position in the assembly.
- NXTTLC generates the cyclic tracking lines for a geometry.
- NXTTLO locate triangle position for hexagons with triangles.
- NXTTLS generates the standard tracking lines for a geometry
- NXTTNS normalizes tracking lines and save on tracking data structure.
- NXTTPO tests that cluster pins do not overlapp.
- NXTTPS tests if pins satisfy required symmetry.
- NXTTRM determines the final mesh of a cell after turn.
- NXTTRS applies Cartesians symmetry to TURN factors.
- NXTVCA computes volume and surfaces for Cartesian geometry.
- NXTVCC computes volumes for a mixed CARCEL geometry.
- NXTVHC computes the volume and area associated with each region or surface for a annular/hexagon with triangular mesh in 2D or 3D geometry.
- NXTVOL computes regional volumes.
- NXTXYZ finds the global cell limits.

In addition the following lines are called by the EXCELP routine when inline tracking is requested:

- NXTTGC generate a specific cyclic tracking line for a geometry (inline tracking).
- NXTTGS generate a specific standard tracking line for a geometry (inline tracking).

## APPENDIX C

## CONTENTS OF THE DRAGON BINARY TRACKING FILE

The tracking file generated by the NXT module is compatible with EXCELT and stored in a sequential binary format. It contains the information required to evaluate the collision probabilities associated with a geometry or to solve the transport equation using the method of characteristics. It can be decoded using the following FORTRAN instructions:

```

      SUBROUTINE RDTRAK(IUNIT)
*-----
*   IUNIT is fortran file unit number
*-----
*   Dimensioning of variables
*-----
      IMPLICIT NONE
      INTEGER IUNIT, NCOMNT, NBTR, IFMT
      CHARACTER CHEAD*4, COMNT*80
      INTEGER NDIM, LTRK, NREG, NSOUT, NALBG, NCOR, NANGL, MAXSUB, MXSEG
      INTEGER NSUB, NSEG, IA
      DOUBLE PRECISION WEIGHT
      INTEGER ICOM, II, JJ, IDLINE(4)
*-----
*   Allocatable arrays
*-----
      REAL, ALLOCATABLE, DIMENSION(:) :: VOLSUR
      REAL, ALLOCATABLE, DIMENSION(:) :: GALBED, DENSTY
      REAL, ALLOCATABLE, DIMENSION(:, :) :: ANGLE
      INTEGER, ALLOCATABLE, DIMENSION(:) :: MATALB, ICODE,
      > IANGL, NRSEG
      DOUBLE PRECISION, ALLOCATABLE, DIMENSION(:) :: SEGLEN
      DOUBLE PRECISION, ALLOCATABLE, DIMENSION(:, :) :: BOLINE
*-----
*   Read header and comments
*-----
      READ(IUNIT) CHEAD, NCOMNT, NBTR, IFMT
      DO ICOM=1, NCOMNT
         READ(IUNIT) COMNT
      ENDDO
*-----
*   Read general information
*-----
      READ(IUNIT) NDIM, LTRK, NREG, NSOUT, NALBG, NCOR, NANGL, MAXSUB, MXSEG
      ALLOCATE(VOLSUR(-NSOUT:NREG), MATALB(-NSOUT:NREG), ICODE(NALBG),
      > GALBED(NALBG), ANGLE(NDIM, NANGL), DENSTY(NANGL))
      READ(IUNIT) (VOLSUR(II), II=-NSOUT, NREG)
      READ(IUNIT) (MATLAB(II), II=-NSOUT, NREG)
      READ(IUNIT) (ICODE(II), II=1, NALBG)
      READ(IUNIT) (GALBED(II), II=1, NALBG)
      READ(IUNIT) ((ANGLE(II, JJ), II=1, NDIM), JJ=1, NANGL)
      READ(IUNIT) (DENSTY(JJ), JJ=1, NANGL)

```



```

*-----
*   Read tracks information
*-----
      ALLOCATE( IANGL( MAXSUB ), NRSEG( NSEG ) )
      ALLOCATE( SEGLN( NSEG ) )
      ALLOCATE( BOLINE( NDIM, MAXSUB ) )
100  CONTINUE
      IF( IFMT .EQ. 1 ) THEN
          READ( IUNIT, END=105 ) NSUB, NSEG, WEIGHT,
          > ( IANGLE( IA ), IA=1, NSUB ),
          > ( NRSEG( II ), II=1, NSEG ),
          > ( SEGLN( II ), II=1, NSEG ),
          > ( IDLINE( IRA ), IRA=1, 4 ),
          > ( ( BOLINE( IRA, II ), IRA=1, NDIM ), II=1, NSUB )
      ELSE
          READ( IUNIT, END=105 ) NSUB, NSEG, WEIGHT,
          > ( IANGLE( IA ), IA=1, NSUB ),
          > ( NRSEG( II ), II=1, NSEG ),
          > ( SEGLN( II ), II=1, NSEG )
      ENDIF
      GO TO 100
105  CONTINUE
      DEALLOCATE( BOLINE, SEGLN, NRSEG, IANGL )
      DEALLOCATE( DENSTY, ANGLE, GALBED, ICODE, MATALB, VOLSUR )
      RETURN
      END

```

where

IUNIT	FORTTRAN unit associated with this file.
CHEAD	keyword to identify the tracking file. The file is a valid DRAGON tracking file if CHEAD=\$TRK
NCOMNT	Number of comment records.
NBTR	Total number of tracks if known, otherwise NBTRK=0.
IFMT	track format where <ul style="list-style-type: none"> <li>• IFMT=0 for a short file where only the line segments are saved.</li> <li>• IFMT=1 for a long file where the starting location of each line segment is saved (useful for drawing purposes).</li> </ul>
COMNT	comment lines.
NDIM	Dimension of problem (2 for 2-D geometry and 3 for 3-D geometry).
LTRK	Type of tracking: <ul style="list-style-type: none"> <li>• LTRK=0 or isotropic tracking.</li> <li>• LTRK=1 for cyclic (specular) tracking.</li> </ul>
NREG	Number of regions.
NSOUT	Number of outer surfaces.

NALBG	Number of geometric albedos.
NCOR	Number of initial and final surfaces that can be crossed by a track.
NANGL	Number of track direction angles considered in the integration.
MAXSUB	Maximum number of subtracks in a line. A subtrack corresponds to the line segments associated with an inner and an outer surface. For isotropic tracking, a single subtrack is associated with each line. For cyclic (specular) tracking a new subtrack is started after a line intersects a face.
MXSEG	Maximum number of line segments per track.
VOLSUR	Surface-volume vector.
MATALB	Surface direction and region material identification vector.
ICODE	Albedo number associated with a face. Negative values for ICODE refers to a geometric albedo while positive values refers to a physical albedo.
GALBED	Geometric albedos.
ANGLE	Tracking angle directions. For 2-D geometry they represent the cosine and sine of the tracking angles respectively while for 3-D geometries, they represent the 3 director cosines associated with the track direction.
DENSTY	Density associated with each tracking angle. For 2-D geometries, this is a linear density while for 3-D geometries it is a surface track density.
NSUB	Number of subtracks for this line.
NSEG	Number of segments for this line.
WEIGHT	Integration weight factor associated with this track.
IANGL	Angle number for subtracks.
NRSEG	Surface (negative) and region (positive) numbers crossed by track. When ISPEC=0, the first and the last elements of this vector are associated with the external surfaces, all the other elements being associated with region numbers. When ISPEC=1, NRSEG starts and finishes with a surface number. In addition the surface numbers will be mixed with the region numbers in the remaining elements of the vector NRSEG.
SEGLEN	Length of segment crossing a region. Elements of the vector SEGLEN associated with surfaces are set to 1.0 for isotropic scattering and 0.5 for specular tracking.
IDLINE	line type identifier.
BOLINE	begining of subtrack origin.

## APPENDIX D

### CONTENTS OF A /TRACKING/ DIRECTORY

This directory contains the information generated after the geometry has been analyzed by the NXT tracking module of DRAGON.

#### D.1 The main directory

The records presents in Table 1 are always present in the /TRACKING/ directory.

Table 1: Main records in /TRACKING/

Name	Type	Units	Comment
SIGNATURE_	C*12		parameter SIGNA containing the signature of the data structure.
STATE-VECTOR	I(40)		array $\mathcal{S}_i^t$ containing various parameters that are required to describe the data structure.
TRACK-TYPE_	C*12		parameter TRKT containing the name of the tracking model used to create the structure.
TITLE_	C*72		parameter TITLE containing the title associated with the tracking structure.
MATCOD_	I( $\mathcal{S}_1^t$ )		array $M_i$ containing the physical (real) mixture numbers associated with flux regions in the geometry.
HOMMATCOD_	I( $\mathcal{S}_1^t$ )		array $H_i$ containing the virtual (homogenization) mixture numbers associated with flux regions in the geometry.
KEYFLX_	I( $\mathcal{S}_1^t$ )		array $I_i$ containing the index positions, for the array containing the transport unknowns, where the volume averaged flux are stored.
VOLUME_	R( $\mathcal{S}_1^t$ )	cm <sup>3</sup>	array $V_i$ containing the volumes of flux regions in the geometry.

The signature for this data structure is SIGNA=L\_TRACK\_ . The tracking model TRKT can take the following values:

$$\text{TRKT} = \begin{cases} \text{EXCELL} & \text{for tracking by the EXCELT: , EXCELL: or NXT: modules;} \\ \text{SYBIL} & \text{for tracking by the SYBILT: module;} \\ \text{JPM} & \text{for tracking by the JPMT: module.} \end{cases}$$

The array  $\mathcal{S}_i^t$  contains the following information:

- $\mathcal{S}_1^t = N_r$  is the number of independent flux regions in the problem.
- $\mathcal{S}_2^t = N_u$  is the number of independent transport unknowns in the problem.

- $S_3^t = I_L$  is the leakage flag where:

$$I_L = \begin{cases} 0 & \text{indicates that leakage is present;} \\ 1 & \text{indicates that no leakage is present.} \end{cases}$$

- $S_4^t = S_7^G = M_m$  is the maximum number of mixtures (physical and virtual) used in the problem.
- $S_5^t = N_s$  is the number of independent external surfaces in the problem.
- $S_6^t$  is the anisotropy level for flux calculations in the problem where:

$$S_6^t = \begin{cases} 0 & \text{only the isotropic flux components are considered;} \\ 1 & \text{the isotropic and linearly anisotropic flux components are considered.} \end{cases}$$

In addition to the above records, the main /TRACKING/ directory also contains information that is specific to each tracking module. The contents of the  $S_i^t$  vector for  $i \geq 7$  will also depend on the specific tracking module selected. This information will be described in Section D.2.

## D.2 The EXCELL records and sub-directories

When TRKT=EXCELL the following elements of  $S_i^t$  are also defined.

- $S_7^t$  is the specific EXCELL tracking procedure considered where:

$$S_7^t = \begin{cases} 1 & \text{for tracking Cartesian assemblies using EXCELT: or EXCELL:;} \\ 2 & \text{for tracking hexagonal assemblies using EXCELT:;} \\ 3 & \text{for tracking 2-D cluster cells using EXCELT:;} \\ 4 & \text{for tracking 2-D and 3-D Cartesian and hexagonal assemblies with clusters using NXT: .} \end{cases}$$

- $S_8^t$  is the track normalization flag where:

$$S_8^t = \begin{cases} -1 & \text{direction dependent track normalization to merged volumes;} \\ 0 & \text{global track normalization to merged volumes;} \\ 1 & \text{no normalization.} \end{cases}$$

The default is  $S_8^t = 0$ . The option  $S_8^t = -1$  can only be activated using the RENO keyword in the NXT:, EXCELT: and EXCELL: modules.

- $S_9^t$  is the tracking type where:

$$S_9^t = \begin{cases} 0 & \text{means that a standard tracking procedure was considered;} \\ 1 & \text{means that a cyclic tracking procedure was considered.} \end{cases}$$

- $S_{10}^t$  is the type of boundary conditions to be used for the collision probability calculations where:

$$S_{10}^t = \begin{cases} 0 & \text{isotropic (white) boundary conditions are considered;} \\ 1 & \text{mirror-like (specular) boundary conditions are considered.} \end{cases}$$

Mirror-like boundary conditions ( $S_{10}^t = 1$ ) can be used only if a cyclic tracking procedure is considered ( $S_9^t = 1$ ).

- $\mathcal{S}_{11}^t = N_\Omega$  is the order of the azimuthal (2-D) or solid (3-D) angular quadrature. For 2-D geometry, the order of the azimuthal quadrature represents:
  - $N_\Omega$  equal sectors (trapezoidal quadrature) in the  $[0, \pi]$  range when the EXCELT: module is used for Cartesian assemblies;
  - $N_\Omega$  equal sectors (trapezoidal quadrature) in the  $[0, \pi]$  range when the EXCELT: module is used for hexagonal geometries;
  - $N_\Omega$  equal sectors (trapezoidal quadrature) in the  $[0, \max(\mathcal{S}_{12}^t, 2)\pi]$  range when the EXCELT: module is used for cluster geometries;
  - $N_\Omega$  equal sectors in the  $[0, \pi/2]$  range and  $N_\Omega$  equal sectors in the  $[\pi/2, \pi]$  range (trapezoidal quadrature) when the NXT: module is used.

For 3-D geometry, the order of the solid angle quadrature represent

- $\mathcal{S}_{16}^t = N_\Omega(N_\Omega + 2)/8$  directions in each quadrant ( $0 \leq \varphi \leq \pi/2$  and  $0 \leq \theta \leq \pi/2$ ) for the  $EQ_{N_\Omega}$  quadrature (NXT:, EXCELT: and EXCELL: modules for Cartesian assemblies);
  - $\mathcal{S}_{16}^t = 3 \times N_\Omega(N_\Omega + 2)/8$  directions in each quadrant ( $0 \leq \varphi \leq \pi/2$  and  $0 \leq \theta \leq \pi/2$ ) for the  $LC_{N_\Omega}$  quadrature (NXT: module for Cartesian assemblies);
  - $\mathcal{S}_{16}^t = 3 \times N_\Omega^2/2$  directions in each quadrant ( $0 \leq \varphi \leq \pi/2$  and  $0 \leq \theta \leq \pi/2$ ) for the  $LT_n$  quadrature (NXT: module for Cartesian assemblies);
  - $N_\Omega$  equal sectors (trapezoidal quadrature) in the  $[0, 2\pi]$  range when the EXCELT: module is used for hexagonal geometries;
- $\mathcal{S}_{12}^t$  is the angular symmetry factor.
  - $\mathcal{S}_{13}^t$  is the polar angle quadrature type where:

$$\mathcal{S}_{13}^t = \begin{cases} 1 & \text{for a Gauss quadrature;} \\ 2 & \text{for a CACTUS type A quadrature;} \\ 3 & \text{for a CACTUS type B quadrature;} \\ 4 & \text{for an optimized Bickley quadrature.} \end{cases}$$

- $\mathcal{S}_{14}^t$  is the polar angle quadrature order.
- $\mathcal{S}_{15}^t$  is the azimuthal (2-D) or solid (3-D) angle quadrature type where:

$$\mathcal{S}_{15}^t = \begin{cases} 1 & \text{for a } EQ_{N_\Omega} \text{ (3-D) or trapezoidal (2-D) quadrature;} \\ 2 & \text{for a Gauss quadrature (2-D hexagonal geometries in EXCELT:);} \\ 3 & \text{for a median angle quadrature;} \\ 4 & \text{for a } LC_{N_\Omega} \text{ 3-D quadrature;} \\ 5 & \text{for a } LT_{N_\Omega} \text{ 3-D quadrature.} \end{cases}$$

- $\mathcal{S}_{16}^t$  is the number of directions for the azimuthal (2-D) or solid (3-D) angle quadrature.
- $\mathcal{S}_{17}^t$  is the maximum number of tracking points on a line.
- $\mathcal{S}_{18}^t$  is the maximum length of a track.
- $\mathcal{S}_{19}^t$  is the total number of tracks generated.
- $\mathcal{S}_{20}^t$  is the total number of track directions processed.

The records presented in Table 2 will also be present on the main level of a /TRACKING/ directory when the EXCELT:, EXCELL: or NXT: tracking modules are used.

Table 2: Additional records for EXCELL tracking

Name	Type	Units	Comment
EXCELTRACKOP	R(40)		array $\mathcal{R}_i$ containing additional tracking parameters.
ICODE_	I(6)		array $I_{\beta,k}$ containing the surface albedo index (geometric albedos are used if $I_{\beta,k} < 0$ while physical albedos (see /MACROLIB/ directory) are used when $I_{\beta,k} > 0$ ).
ALBEDO_	R(6)		array $\beta_{g,k}$ containing the geometric surface albedo (used only if $I_{\beta,k} \geq 0$ ).
EXCELL_	Dir		directory containing additional EXCELT: and EXCELL: records for the cases where $\mathcal{S}_7^t = 1$ or $\mathcal{S}_7^t = 3$ . <b>Present only if <math>\mathcal{S}_7^t &lt; 4</math>.</b>
NXTRecords_	Dir		directory containing additional NXT: records. <b>Present only if <math>\mathcal{S}_7^t = 4</math>.</b>

The record  $\mathcal{R}_i$  contains the following information:

- $\mathcal{R}_1$  is the maximum error allowed on the exponential function.
- $\mathcal{R}_2$  is the user requested tracking density in  $\text{cm}^{-1}$  and in  $\text{cm}^{-2}$  respectively for 2-D and 3-D geometries.
- $\mathcal{R}_3$  is the maximum distance in cm between an integration line and a surface.
- $\mathcal{R}_4$  is the computed tracking density in  $\text{cm}^{-1}$  and in  $\text{cm}^{-2}$  respectively for 2-D and 3-D D geometries (used only if  $\mathcal{S}_7^t = 4$ ).
- $\mathcal{R}_5$  is the computed line spacing in cm (used only if  $\mathcal{S}_7^t = 4$ ).
- $\mathcal{R}_6$  is the weight of the spatial quadrature (used only if  $\mathcal{S}_7^t = 4$ ).
- $\mathcal{R}_7$  is the minimal radius of the circle (2-D) or sphere (3-D) containing the geometry (used only if  $\mathcal{S}_7^t = 4$ ).
- $\mathcal{R}_8$  is the  $x$  position of the center of the minimal circle (2-D) or sphere (3-D) containing the geometry (used only if  $\mathcal{S}_7^t = 4$ ).
- $\mathcal{R}_9$  is the  $y$  position of the center of the minimal circle (2-D) or sphere (3-D) containing the geometry (used only if  $\mathcal{S}_7^t = 4$ ).
- $\mathcal{R}_{10}$  is the  $z$  position of the center of the minimal circle (2-D) or sphere (3-D) containing the geometry (used only if  $\mathcal{S}_7^t = 4$ ).

The /NXTRecords/ directory contains the information required to track the geometry using the NXT: module once it has been analyzed. The contents of this directory is presented in Table 3.

Table 3: Global geometry records in /NXTRecords/

Name	Type	Units	Comment
G00000001DIM	I(40)		array $N_i^{GG}$ containing the dimensioning information required to rebuild the assembly.
G00000001CUF	I(2, $N_5^{GG}$ )		array $D_{i,j}^{GG}$ containing the assembly description of the geometry in terms of cells and rotations. The first element ( $i = 1$ ) identifies the cell number while the second element identifies the cell rotation.
G00000001CIS	I(4, $N_4^{GG}$ )		array $S_{i,j}^{GG}$ containing the cell intrinsic symmetry properties. A value of 1 indicates that a center cell reflexion symmetry is present while a value of 0 indicates that the symmetry is not considered (see below for a more complete description of this array).
G00000001CFE	I(0 : 10, $N_4^{GG}$ )		array $F_{i,j}^{GG}$ containing the assembly external surface identification index (see below for a more complete description of this array).
G00000001SMX	D(0 : $N_{13}^{GG}$ )	cm	array $x^{GG}$ containing the $x$ -directed mesh for the cell assembly in a Cartesian or Cylindrical geometry and the $x$ position of the cell center for an hexagonal assembly (see below for more explanations).
G00000001SMY	D(0 : $N_{14}^{GG}$ )	cm	array $y^{GG}$ containing the $y$ -directed mesh for the cell assembly in a Cartesian or Cylindrical geometry and the $y$ position of the cell center for an hexagonal assembly (see below for more explanations).
G00000001SMZ	D(0 : $N_{15}^{GG}$ )	cm	array $z^{GG}$ containing the $z$ -directed mesh for the cell assembly (see below for more explanations).
G00000001SMR	D(0 : 1)	cm	<b>Present only if <math>N_1^{GG} = 3</math>.</b> the radius $r_1^{GG}$ of the outer assembly boundary (see below for more explanations).
KEYMRG_	I( $-N_{23}^{GG} : N_{22}^{GG}$ )		<b>Present only if <math>N_2^{GG} = 1</math>.</b> array $MRG_i$ containing the merged surface and region number associated with each individual surfaces and regions in this geometry.
MATALB_	I( $-N_{23}^{GG} : N_{22}^{GG}$ )		array containing the albedo number associated with each surface and the physical mixture number associated with each region in this geometry.
HOMMATALB_	I( $-N_{23}^{GG} : N_{22}^{GG}$ )		array containing the albedo number associated with each surface and the virtual (homogenization) mixture number associated with each region in this geometry.
SAreaRvolume	D( $-N_{23}^{GG} : N_{22}^{GG}$ )		array containing the area ( $S_\alpha$ in cm for 2-D and $\text{cm}^2$ for 3-D problems) and volume ( $V_i$ $\text{cm}^2$ for 2-D and $\text{cm}^3$ for 3-D problems) associated with each surface and region in this geometry.

The dimensioning vector for the global geometry contains the following information:

- $N_1^{GG}$  number of dimensions for the problem.
- $N_2^{GG}$  type of boundary. A value of 0 indicates a Cartesian geometry, a value of 1 indicates a cylindrical geometry and a value of 2 an hexagonal geometry.
- $N_3^{GG}$  first direction to process in the analysis. For cylinder, this is the direction of the first axis of the plane normal to the cylinder axis. For Cartesian and hexagonal geometries a value of 1 ( $x$ -axis) is selected by default.
- $N_4^{GG}$  number of cells in the original geometry (before unfolding).
- $N_5^{GG}$  number of cells in the geometry after the original geometry is unfolded according to the symmetries.
- $N_6^{GG}$  diagonal symmetry flag. A value of 0 indicates that this symmetry is not used. A value of  $-1$  indicates that the symmetry is used for the  $x_- = y_+$  plane and a value of 1 that the symmetry is used for the  $x_+ = y$  plane.
- $N_7^{GG}$  flag to identify symmetries with respect to the  $x$ -axis ( $x_-$  or  $x_+$ ). A value of 0 indicates that no symmetry is present,  $N_7^{GG} = \pm 1$  is for a SYME symmetry at the  $x_{\pm}$  plane,  $N_7^{GG} = \pm 2$  represents a SSYM symmetry at the  $x_{\pm}$  plane and  $N_7^{GG} = 3$  implies a translation symmetry is the  $x$  direction ( $x_- = x_+$ ).
- $N_8^{GG}$  flag to identify symmetries with respect to the  $y$ -axis ( $y_-$  or  $y_+$ ). A value of 0 indicates that no symmetry is present,  $N_7^{GG} = \pm 1$  is for a SYME symmetry at the  $y_{\pm}$  plane,  $N_7^{GG} = \pm 2$  represents a SSYM symmetry at the  $y_{\pm}$  plane and  $N_7^{GG} = 3$  implies a translation symmetry is the  $y$  direction ( $y_- = y_+$ ).
- $N_9^{GG}$  flag to identify symmetries with respect to the  $z$ -axis ( $z_-$  or  $z_+$ ). A value of 0 indicates that no symmetry is present,  $N_7^{GG} = \pm 1$  is for a SYME symmetry at the  $z_{\pm}$  plane,  $N_7^{GG} = \pm 2$  represents a SSYM symmetry at the  $z_{\pm}$  plane and  $N_7^{GG} = 3$  implies a translation symmetry is the  $z$  direction ( $z_- = z_+$ ).
- $N_{10}^{GG}$  number of  $x$  mesh subdivisions or hexagons in the original geometry.
- $N_{11}^{GG}$  number of  $y$  mesh subdivisions or hexagons in the original geometry.
- $N_{12}^{GG}$  number of  $z$  mesh subdivisions in the original geometry.
- $N_{13}^{GG}$  number of  $x$  mesh subdivisions or hexagons in the unfolded geometry.
- $N_{14}^{GG}$  number of  $y$  mesh subdivisions or hexagons in the unfolded geometry.
- $N_{15}^{GG}$  number of  $z$  mesh subdivisions in the unfolded geometry.
- $N_{16}^{GG}$  maximum number cells required to represent this geometry.
- $N_{17}^{GG}$  maximum number of region for this geometry.
- $N_{18}^{GG}$  total number of clusters in this geometry.
- $N_{19}^{GG}$  maximum number of pins in this geometry.
- $N_{20}^{GG}$  maximum dimensions of any mesh array for a cell in this geometry.
- $N_{21}^{GG}$  maximum dimensions of any mesh array for a pin in this geometry.
- $N_{22}^{GG}$  number of external surfaces for this geometry.
- $N_{23}^{GG}$  number of regions for this geometry.
- $N_{24}^{GG}$  maximum number of external surfaces in a sub-geometry included in this geometry.
- $N_{25}^{GG}$  maximum number of regions in a sub-geometry included in this geometry.



The indexing of array  $S_{i,j}^{GG}$  for the axis of symmetry is as follows

1. Cartesian assemblies:

- $i = 1$  refers to a reflexion of the geometry on a plane normal the  $x$ -axis;
- $i = 2$  refers to a reflexion of the geometry on a plane normal the  $y$ -axis;
- $i = 3$  refers to a reflexion of the geometry on the plane  $x = y$ ;
- $i = 4$  refers to a reflexion of the geometry on a plane normal the  $z$ -axis.

2. Hexagonal assemblies (symmetries not yet programmed).

- $i = 1$  refers to a reflexion of the geometry on a plane normal the  $u$ -axis;
- $i = 2$  refers to a reflexion of the geometry on a plane normal the  $v$ -axis;
- $i = 3$  refers to a reflexion of the geometry on the plane  $w$ ;
- $i = 4$  refers to a reflexion of the geometry on a plane normal the  $z$  axis.

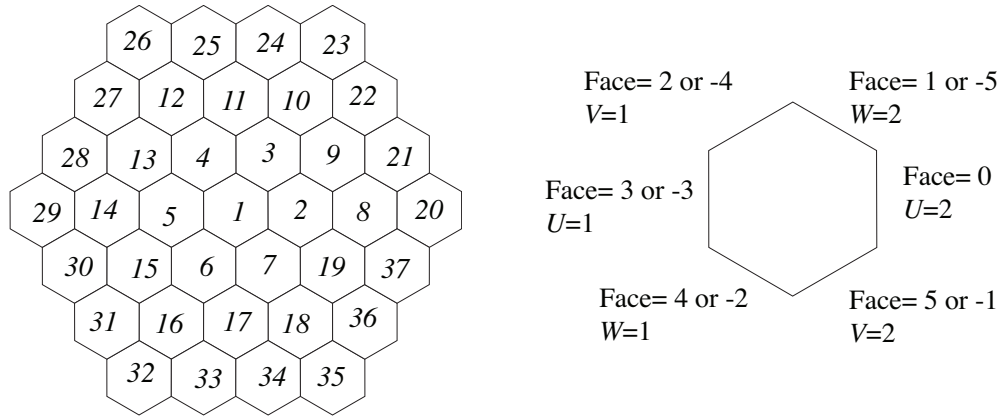


Figure 24: Example of an assembly of hexagons (left) and external faces identification for an hexagon

The indexing of array  $F_{i,j}^{GG}$  for external surface identification is as follows. First  $F_{0,j}^{GG}$  represents the number of times the cell appears in the geometry after it has been unfolded. For  $i > 0$ ,  $F_{i,j}^{GG}$  can take the following values

$$F_{i,j}^{GG} = \begin{cases} 1 & \text{surface associated with direction } i \text{ of cell } j \text{ is an external boundary of the assembly} \\ 0 & \text{surface associated with direction } i \text{ of cell } j \text{ is not an external boundary of the assembly} \end{cases}$$

with the following planes associated with different values of  $i$ :

1. Cartesian assemblies:

- $i = 1$  surfaces on the  $x_-$  plane for cell  $j$ ;
- $i = 2$  surfaces on the  $x_+$  plane for cell  $j$ ;
- $i = 3$  surfaces on the  $y_-$  plane for cell  $j$ ;
- $i = 4$  surfaces on the  $y_+$  plane for cell  $j$ ;
- $i = 5$  surfaces on the  $z_-$  plane for cell  $j$ ;
- $i = 6$  surfaces on the  $z_+$  plane for cell  $j$ .

2. Hexagonal assemblies (see Figure 24):

- $i = 1$  surfaces on the  $u_-$  plane for cell  $j$ ;
- $i = 2$  surfaces on the  $u_+$  plane for cell  $j$ ;
- $i = 3$  surfaces on the  $v_-$  plane for cell  $j$ ;
- $i = 4$  surfaces on the  $v_+$  plane for cell  $j$ ;
- $i = 5$  surfaces on the  $z_-$  plane for cell  $j$ ;
- $i = 6$  surfaces on the  $z_+$  plane for cell  $j$ ;
- $i = 9$  surfaces on the  $w_-$  plane for cell  $j$ ;
- $i = 10$  surfaces on the  $w_+$  plane for cell  $j$ .

The arrays  $x^{\text{GG}}$ ,  $y^{\text{GG}}$ ,  $z^{\text{GG}}$  and  $r^{\text{GG}}$  contain the following information:

1. Cartesian assemblies:

- $x_{i-1}^{\text{GG}}$  and  $x_i^{\text{GG}}$  are the lower and upper  $x$  limits of mesh element  $i$  ( $i = 1, n^x$ );
- $y_{j-1}^{\text{GG}}$  and  $y_j^{\text{GG}}$  are the lower and upper  $y$  limits of mesh element  $j$  ( $j = 1, n^y$ );
- $z_{k-1}^{\text{GG}}$  and  $z_k^{\text{GG}}$  are the lower and upper  $z$  limits of mesh element  $k$  ( $k = 1, n^z$ ).

2. Hexagonal assemblies (see Figure 24):

- $x_0^{\text{GG}} = h$  is the width of one face of the hexagon and  $x_i^{\text{GG}}$  is the position in  $x$  of the center of cell  $i$  in the assembly;
- $y_0^{\text{GG}} = h$  is the width of one face of the hexagon and  $y_j^{\text{GG}}$  is the position in  $y$  of the center of cell  $j$  in the assembly;
- $z_{k-1}^{\text{GG}}$  and  $z_k^{\text{GG}}$  are the lower and upper  $z$  limits of mesh element  $k$  ( $k = 1, n^z$ ).

As we noted above, the global geometry is always an assembly containing cells. For each cell  $i$  in this assembly, several records will be generated in the /NXTRecords/ directory. These records are identified using a FORTRAN CHARACTER\*12 variable as follows

```

INTEGER          I
CHARACTER*12     NAMREC
CHARACTER*3      NREC
WRITE (NAMREC, '(A1,I8.8,A3)') 'C',I,NREC

```

where the variable NREC can take the following values:

- DIM for dimensioning information;
- SMR for the radial mesh description;
- SMX for the  $x$ -directed mesh description;
- SMY for the  $y$ -directed mesh description;
- SMZ for the  $z$ -directed mesh description;
- MIX for physical mixture description;
- HOM for virtual mixture description;
- VSE for areas and volumes results;
- VSI for local surfaces and regions identification;
- RID for final region numbering;

- SID for final surface numbering
- PNT for pin contents description;
- PIN for pins location.

In Table 4, a description of the additional /NXTRecords/ records associated with cell  $i = 1$  can be found.

Table 4: Cell  $i = 1$  records in /NXTRecords/

Name	Type	Units	Comment
C00000001DIM	I(40)		array $N_j^{GC}$ containing the dimensioning information required to rebuilt the cell.
C00000001SMR	D( $N_2^{GC}$ )	cm	array $r_j^{GC}$ containing the cell radial mesh description.
C00000001SMX	D( $N_3^{GC}$ )	cm	array $x_j^{GC}$ containing the cell $x$ -directed mesh description.
C00000001SMY	D( $N_4^{GC}$ )	cm	array $y_j^{GC}$ containing the cell $y$ -directed mesh description.
C00000001SMZ	D( $N_5^{GC}$ )	cm	array $z_j^{GC}$ containing the cell $z$ -directed mesh description.
C00000001MIX	I( $N_6^{GC}$ )		array $M_j^{GC}$ containing the cell physical mixture for each region.
C00000001HOM	I( $N_6^{GC}$ )		array $H_j^{GC}$ containing the cell virtual mixture for each region.
C00000001VSE	D( $-N_9^{GC} : N_8^{GC}$ )		array $SV_j^{GC}$ containing surface area $j$ ( $SV_{-j}^{GC} = S_j^{GC}$ in cm for 2-D and cm <sup>2</sup> for 3-D problems) and regional volumes $j$ ( $SV_j^{GC} = V_j^{GC}$ in cm <sup>2</sup> for 2-D and cm <sup>3</sup> for 3-D problems).
C00000001VSI	I(5, $-N_9^{GC} : N_8^{GC}$ )		array $VSI_{k,j}^{GC}$ containing the location of a surface ( $j < 0$ ) and a region ( $j > 0$ ). See below for a more complete description of this array.
C00000001RID	I( $N_8^{GC}$ )		index array $RID_j^{GC}$ associating local and global region numbering.
C00000001SID	I( $N_9^{GC}$ )		index array $SID_{j,i}^{GC}$ associating local and global outer surface numbering.
C00000001PNT	I(3, $N_{18}^{GC}$ )		array $PC_{k,j}^{GC}$ containing the cell pin contents.
C00000001PIN	D( $-1 : 4, N_{18}^{GC}$ )		array $p_{k,j}^{GC}$ containing the location of the pins in cell.

Note that the record names above are built using the following FORTRAN instructions:

```
WRITE(NAMREC, '(A1, I8.8, A3)') 'C', i, NAMEXT
```

The cell dimensioning array  $N_i^{GC}$  for cell  $i$  contains the following information:

- $N_1^{GC}$  cell geometry type;
- $N_2^{GC}$  dimensions of the radial mesh array;
- $N_3^{GC}$  dimensions of the  $x$ -directed mesh array;
- $N_4^{GC}$  dimensions of the  $y$ -directed mesh array;

- $N_5^{\text{GC}}$  dimensions of the  $z$ -directed mesh array;
- $N_6^{\text{GC}}$  dimensions of the mixture record;
- $N_7^{\text{GC}}$  geometry level (1 for cell);
- $N_8^{\text{GC}}$  number of regions in the cell before symmetry considerations;
- $N_9^{\text{GC}}$  number of surfaces in the cell before symmetry considerations;
- $N_{10}^{\text{GC}}$  number of regions in the cell after symmetry considerations;
- $N_{11}^{\text{GC}}$  number of surfaces in the cell after symmetry considerations;
- $N_{12}^{\text{GC}}$  first global region number for cell;
- $N_{13}^{\text{GC}}$  last global region number for cell;
- $N_{14}^{\text{GC}}$  first global surface number for cell;
- $N_{15}^{\text{GC}}$  last global surface number for cell;
- $N_{16}^{\text{GC}}$  number of pin cluster geometries in cell;
- $N_{17}^{\text{GC}}$  first pin cluster geometry associated with cell;
- $N_{18}^{\text{GC}}$  total number of pins in cell;
- $N_{19}^{\text{GC}}$  number of times this cell is used in the global cell.

while the remaining elements are not used.

The array  $x_j^{\text{GC}}$  contains the following information:

- $x_{-1}^{\text{GC}}$  contains the displacement of the center of the cylindrical region with respect to the center of the Cartesian mesh in direction  $x$ . This center is located at:

$$x_c = \frac{x_{n^x}^{\text{GC}} + x_0^{\text{GC}}}{2}$$

where we have used  $n^x = N_3^{\text{GC}}$ .

- $x_{j-1}^{\text{GC}}$  and  $x_j^{\text{GC}}$  are the lower and upper  $x$  limits of mesh element  $j$  ( $j = 1, n^x$ ).

The array  $y_j^{\text{GC}}$  contains the following information:

- $y_{-1}^{\text{GC}}$  contains the displacement of the center of the cylindrical region with respect to the center of the Cartesian mesh in direction  $y$ . This center is located at:

$$y_c = \frac{y_{n^y}^{\text{GC}} + y_0^{\text{GC}}}{2}$$

where we have used  $n^y = N_4^{\text{GC}}$ .

- $y_{j-1}^{\text{GC}}$  and  $y_j^{\text{GC}}$  are the lower and upper  $y$  limits of mesh element  $j$  ( $j = 1, n^y$ ).

The array  $z_j^{\text{GC}}$  contains the following information:

- $z_{-1}^{\text{GC}}$  contains the displacement of the center of the cylindrical region with respect to the center of the Cartesian mesh in direction  $z$ . This center is located at:

$$z_c = \frac{z_{n^z}^{\text{GC}} + z_0^{\text{GC}}}{2}$$

where we have used  $n^z = N_5^{\text{GC}}$ .

- $z_{j-1}^{\text{GC}}$  and  $z_j^{\text{GC}}$  are the lower and upper  $z$  limits of mesh element  $j$  ( $j = 1, n^z$ ).

The array  $r_j^{\text{GC}}$  contains the following information:

- $r_{-1}^{\text{GC}} = r_0^{\text{GC}} = 0$ .
- $r_{j-1}^{\text{GC}} \leq r \leq r_j^{\text{GC}}$  describes the position in  $r$  of mesh element  $j$  ( $j = 1, N_2^{\text{GC}}$ ).

The array  $p_j^{\text{GC}}$  contains the following information:

- $p_{-1}^{\text{GC}}$  is the angular position of  $z$ -,  $x$ - or  $y$ -directed pin with respect to the  $x$ ,  $y$  or  $z$  axis.
- $p_0^{\text{GC}}$  is the radial position of  $z$ -,  $x$ - or  $y$ -directed pin with respect to the  $x - y$ ,  $y - z$  or  $z - x$  center of the cell where the pin is located.
- $p_1^{\text{GC}}$  is the height of a  $x$ -directed pin.
- $p_2^{\text{GC}}$  is the height of a  $y$ -directed pin.
- $p_3^{\text{GC}}$  is the height of a  $z$ -directed pin.
- $p_4^{\text{GC}}$  is the outer radius of the pin.

The surface and volume identification array  $\text{VSI}_{k,j}^{\text{GC}}$  contains the following information

- $\text{VSI}_{1,j}^{\text{GC}}$  identify the  $x$  or  $u$  mesh position of the region or volume. For a  $-x$  or  $-u$  surface, a value of  $-1$  is used while for a  $+x$  or  $+u$  surface, a value of  $-2$  is specified.
- $\text{VSI}_{2,j}^{\text{GC}}$  identify the  $y$  or  $v$  mesh position of the region or volume. For a  $-y$  or  $-v$  surface, a value of  $-1$  is used while for a  $+y$  or  $+v$  surface, a value of  $-2$  is specified.
- $\text{VSI}_{3,j}^{\text{GC}}$  identify the  $z$  mesh position of the region or volume. For a  $-z$ , a value of  $-1$  is used while for a  $+z$  surface, a value of  $-2$  is specified.
- $\text{VSI}_{4,j}^{\text{GC}}$  identify the radial mesh position of the region or volume. Regions outside the external boundary are assigned the value  $N_2^{\text{GC}} + 1$ .
- $\text{VSI}_{5,j}^{\text{GC}}$  identify the  $w$  mesh position of the region or volume. For a  $-w$  surface, a value of  $-1$  is used while for a  $+w$  surface, a value of  $-2$  is specified.

In Table 5, a description of the additional /NEXTRECORDS/ records associated with pin  $i = 1$  can be found. These records are identified using a procedure similar to that used for cell records, namely

```

INTEGER          I
CHARACTER*12     NAMREC
CHARACTER*3      NREC
WRITE(NAMREC, '(A1,I8.8,A3)') 'P', I, NREC

```

where the variable NREC can take the same values as for cell records, except for NREC=PNT and NREC=PIN which are now forbidden.

Table 5: Pin  $i = 1$  records in /NXTRecords/

Name	Type	Units	Comment
P00000001DIM	I(40)		array $N_j^{\text{GP}}$ containing the dimensioning information required to rebuilt the pin.
P00000001SMR	D( $N_2^{\text{GP}}$ )	cm	array $r_j^{\text{GP}}$ containing the pin radial mesh description.
P00000001SMX	D( $N_3^{\text{GP}}$ )	cm	array $x_j^{\text{GP}}$ containing the pin $x$ -directed mesh description.
P00000001SMY	D( $N_4^{\text{GP}}$ )	cm	array $y_j^{\text{GP}}$ containing the pin $y$ -directed mesh description.
P00000001SMZ	D( $N_5^{\text{GP}}$ )	cm	array $z_j^{\text{GP}}$ containing the pin $z$ -directed mesh description.
P00000001MIX	I( $N_6^{\text{GP}}$ )		array $M_j^{\text{GP}}$ containing the pin physical mixture for each region.
P00000001HOM	I( $N_6^{\text{GP}}$ )		array $H_j^{\text{GP}}$ containing the pin virtual mixture for each region.
P00000001VSE	D( $-N_9^{\text{GP}} : N_8^{\text{GP}}$ )		array $SV_j^{\text{GP}}$ containing surface area $j$ ( $SV_{-j}^{\text{GP}} = S_j^{\text{GP}}$ in $\text{cm}^2$ for 2-D and $\text{cm}^2$ for 3-D problems) and regional volumes $j$ ( $SV_j^{\text{GP}} = V_j^{\text{GP}}$ in $\text{cm}^3$ for 2-D and $\text{cm}^3$ for 3-D problems).
P00000001VSI	I(5, $-N_9^{\text{GP}} : N_8^{\text{GP}}$ )		array $VSI_{k,j}^{\text{GP}}$ containing the location of a surface ( $j < 0$ ) and a region ( $j > 0$ ). The notation used here is similar to that selected for $VSI_{k,j}^{\text{GC}}$ .
P00000001RID	I( $N_8^{\text{GP}}$ )		index array $RID_j^{\text{GP}}$ associating local and global region numbering.
P00000001SID	I( $N_9^{\text{GP}}$ )		index array $SID_{j,i}^{\text{GP}}$ associating local and global outer surface numbering.

The pin dimensioning array  $N^{\text{GP}}$  contains the following information:

- $N_{1,1}^{\text{GP}}$  pin geometry type;
- $N_{2,2}^{\text{GP}}$  dimensions of the radial mesh array;
- $N_{3,3}^{\text{GP}}$  dimensions of the  $x$ -directed mesh array;
- $N_{4,4}^{\text{GP}}$  dimensions of the  $y$ -directed mesh array;
- $N_{5,5}^{\text{GP}}$  dimensions of the  $z$ -directed mesh array;
- $N_{6,6}^{\text{GP}}$  dimensions of the mixture record;
- $N_{7,7}^{\text{GP}}$  geometry level (2 for pins);
- $N_{8,8}^{\text{GP}}$  number of regions in the pin before symmetry considerations;
- $N_{9,9}^{\text{GP}}$  number of surfaces in the pin before symmetry considerations;
- $N_{10,10}^{\text{GP}}$  number of regions in the pin after symmetry considerations;
- $N_{11,11}^{\text{GP}}$  number of surfaces in the pin after symmetry considerations;

- $N_{12}^{\text{GP}}$  first global region number for pins in cluster;
- $N_{13}^{\text{GP}}$  last global region number for pins in cluster;
- $N_{14}^{\text{GP}}$  first global surface number for pins in cluster;
- $N_{15}^{\text{GP}}$  last global surface number for pins in cluster;
- $N_{16}^{\text{GP}}$  first pin cluster geometry for pins in cluster.
- $N_{17}^{\text{GP}}$  total number of pins in cluster.

while the remaining elements are not used. The array  $x_j^{\text{GP}}$  contains the following information:

- $x_{-1}^{\text{GP}}$  contains the displacement of the center of the cylindrical region with respect to the center of the Cartesian mesh in direction  $x$ . This center is located at:

$$x_c = \frac{x_{n^x}^{\text{GP}} + x_0^{\text{GP}}}{2}$$

where we have used  $n^x = N_3^{\text{GP}}$ .

- $x_{j-1}^{\text{GP}}$  and  $x_j^{\text{GP}}$  are the lower and upper  $x$  limits of mesh element  $j$  ( $j = 1, n^x$ ).

The array  $y_j^{\text{GP}}$  contains the following information:

- $y_{-1}^{\text{GP}}$  contains the displacement of the center of the cylindrical region with respect to the center of the Cartesian mesh in direction  $y$ . This center is located at:

$$y_c = \frac{y_{n^y}^{\text{GP}} + y_0^{\text{GP}}}{2}$$

where we have used  $n^y = N_4^{\text{GP}}$ .

- $y_{j-1}^{\text{GP}}$  and  $y_j^{\text{GP}}$  are the lower and upper  $y$  limits of mesh element  $j$  ( $j = 1, n^y$ ).

The array  $z_j^{\text{GP}}$  contains the following information:

- $z_{-1}^{\text{GP}}$  contains the displacement of the center of the cylindrical region with respect to the center of the Cartesian mesh in direction  $z$ . This center is located at:

$$z_c = \frac{z_{n^z}^{\text{GP}} + z_0^{\text{GP}}}{2}$$

where we have used  $n^z = N_5^{\text{GP}}$ .

- $z_{j-1}^{\text{GP}}$  and  $z_j^{\text{GP}}$  are the lower and upper  $z$  limits of mesh element  $j$  ( $j = 1, n^z$ ).

The array  $r_j^{\text{GP}}$  contains the following information:

- $r_{-1}^{\text{GP}} = r_0^{\text{GP}} = 0$ .
- $r_{j-1}^{\text{GP}} \leq r \leq r_j^{\text{GP}}$  describes the position in  $r$  of mesh element  $j$  with  $j = 1, N_2^{\text{GP}}$ .

Finally the `/NXTRECORDS/` directory also contains records associated with global identification of the surfaces and volumes as illustrated in Table 6.

Table 6: Global geometry records in /NXTRecords/

Name	Type	Units	Comment
TrackingDnsA	$D(S_{20}^t)$	cm	array $D_i$ containing the spatial spacing for each track direction.
TrackingDirC	$D(N_1^{GG}, S_{20}^t)$		array $\alpha_{j,i}$ containing the director cosine for axis $j$ for each track direction.
TrackingOrig	$D(N_1^{GG}, N_p, S_{20}^t)$	cm	array $L_{k,j,i}$ containing the origin in space ( $k = 1, N_1^{GG}$ ) and the direction of the normal plan for each plane $j$ and track direction $i$ .
TrackingWgtD	$D(S_{20}^t)$		array $W_i$ containing the integration weight for each track direction.
VTNormalize_	$D(N_{22}^{GG})$		array $R_i$ containing the ratio of the analytical and numerical volume for each region.
VTNormalizedD	$D(N_{22}^{GG}, S_{20}^t)$		array $R_i$ containing the ratio of the analytical and numerical volume for region $i$ for each track direction. <b>Present only if <math>S_8^t = -1</math>.</b>
KEYMRG_	$I(-N_{23}^{GG}, N_{22}^{GG})$		array $MRG_i$ containing the global merging index.
MATALB_	$I(-N_{23}^{GG}, N_{22}^{GG})$		array containing the albedo number associated with each surface and the physical mixture number associated with each region.
HOMMATALB_	$I(-N_{23}^{GG}, N_{22}^{GG})$		array containing the albedo number associated with each surface and the virtual mixture number associated with each region.
SARvvolume	$D(-N_{23}^{GG}, N_{22}^{GG})$		array containing the area ( $S_\alpha$ in cm for 2-D and $\text{cm}^2$ for 3-D problems) and volumes ( $V_i$ $\text{cm}^2$ for 2-D and $\text{cm}^3$ for 3-D problems) of each external surface and region in the geometry.