Verification Of Dragon: The NXT Tracking Module

A. Zkiek, G. Marleau
Institut de génie nucléaire, Département de génie physique,
École Polytechnique de Montréal, Montréal, Québec, Canada

Abstract

The version of DRAGON-IST that has been verified for the calculation of the incremental cross sections associated with CANDU reactivity devices is version 3.04Bb that was released in 2001. Since then, various improvements were implemented in the code including the `NXT:` module that can track assemblies of clusters in 2-D and 3-D geometries. Here we will discuss the verification plan for the `NXT:` module of DRAGON, illustrate the verification procedure we selected and present our verification results.

1. Introduction

Lattice codes such as DRAGON [1] and WIMS [2] cannot not remain static for very long intervals of time before minor errors are detected and improved calculation options are required to analyse new types of reactors or to improve the precision of previous reactor analysis. In 2000, a version of DRAGON (DRAGON 3.04Bb) was frozen to become “DRAGON-IST” [3]. This version of the code was then qualified for use by the Canadian nuclear industry for the calculation of the incremental cross sections associated with CANDU reactivity devices [4,5]. A substantial part of this DRAGON qualification dealt with the verification of the modules that were essential for 3-D incremental cross sections evaluation including:

1. the `EXCELT:` module that analyses and tracks 3-D assemblies of Cartesian cells containing annular subregions;
2. the `ASM:` module that performs the multigroup collision probability (CP) integration;
3. the `FLU:` module that solves the CP based transport equation for the multiplication constant and the flux (with or without leakage models);
4. the `EDI:` module that generates the fewgroup condensed and homogenised incremental cross section associated with CANDU reactivity devices.

This means that most of the calculation options programmed in the code cannot be used for CANDU safety studies, including those required for 2-D CANDU cluster analysis for the processing of microscopic cross section libraries with resonance self-shielding and for general burnup calculations.

Since then, minor errors were detected and corrected in the code and a substantial number of improvements were introduced. In fact, over the last 6 years, multiple revision to the code DRAGON were produced including revision C to T of version 3.04 as well two successive major releases namely version 3.05 [6,7] and 3.06 [8]. This later release includes:

- a new tracking module that can process assemblies of 2-D and 3-D pin clusters;

- an improved distributed self-shielding algorithm based on an equivalence method;
- a generalized perturbation theory algorithm that can be used for incremental cross section evaluation;
- the possibility to access NDAS format microscopic cross section libraries similar to those used by WIMS-IST and it successors.

We have recently proposed that DRAGON 3.06 be qualified to become the next IST release and we have already undertaken the process of verifying this version. This implies that

- all the modifications to the code since the last IST release must be documented;
- the verification process that was previously undertaken must be repeated when significant changes implemented in the modules previously verified may affect safety related calculation results;
- a significant number of modules that were left out from the last DRAGON verification exercise will need to be considered.

The goal of this paper is to describe the verification procedure that has been used to ensure that the `NXT:` tracking module of DRAGON that can process assemblies of 2-D and 3-D cluster performs according to its specifications for studies of interest in CANDU reactors analysis. We will also discuss the results of our verification efforts.

In Section 2. we will present the general verification procedure that was selected for the `NXT:` module. This will be followed in Section 3. with a description of the means by which this procedure was effectively implemented and in Section 4. of a description of the verification results. Finally we will conclude.

2. Verification procedure for module **`NXT:`**

The `NXT:` module is a new DRAGON tracking module that can deal with 2-D and 3-D Cartesian assemblies of pin clusters. It has the general structure illustrated in Figure 1.

```
NXT
 |----- NXTGET
 |----- NXTACG
 |----- NXTTCG
```

Figure 1 General structure of the `NXT:` module.

This module contains two main components controlled respectively by the routines `NXTACG` and `NXTTCG`. The first routine is used to analyze the geometry, identify the surfaces and regions and evaluate their areas and volumes while the second routine is used to track the geometry for a specific set of numerical quadrature points (see Figure 2). Since most of the subroutines programmed in this module were verified independently while developing the module most of the information generated during the execution of this module with a large `EDIT` level is in a format compatible with Mathematica [9]. Accordingly, the procedures that were used for verifying the various subroutines as they were developed remain available and can be used in the global verification program. We also developed an independent module, called `TLM:`, that can read a sequential binary file and generate the Matlab [10] instructions for drawing these lines in 2 and 3 dimensions. This module is quite general and will be used to show that the tracking module performs as it should. Finally, a large number of the geometries that can be treated by `NXT:` can also be processed by the `EXCELT:` module which was verified in the previous IST release of DRAGON. Thus we can rely on self-consistency tests for geometries that can be processed using both `EXCELT:` and `NXT:`.

```
NXTACG
|----- NXTBCG
|----- NXTBRT
       |----- NXTETS
|----- NXTCUA
       |----- NXTTRS
|----- NXTCVS
       |----- NXTAVS
|----- NXTGMD
       |----- NXTTPO
              |----- NXTIAA
              |----- NXTIRA
       |----- NXTTRM
|----- NXTMCD
       |----- NXTEGI
       |----- NXTRCS
       |----- NXTRIS
              |----- NXTTRS
       |----- NXTRPS
       |----- NXTSGI
       |----- NXTTPS
       |----- NXTVOL
              |----- NXTPCA
                     |----- NXTIRA
                     |----- NXTIRR
                            |----- NXTPRR
              |----- NXTPCC
                     |----- NXTIAA
                     |----- NXTIRA
                     |----- NXTIRR
                            |----- NXTPRR
                     |----- NXTPRA
                     |----- NXTPRR
              |----- NXTVCA
              |----- NXTVCC
                     |----- NXTIRA
                     |----- NXTVCA
```

```
NXTTCG
|----- NXTQAS
       |----- NXTQEW
       |----- NXTQLC
       |----- NXTQLT
|----- NXTQSC
|----- NXTQSS
|----- NXTTCG
       |----- NXTLCA
       |----- NXTTCR
              |----- NXTLCA
              |----- NXTLCU
              |----- NXTLCY
              |----- NXTLRS
              |----- NXTRTL
|----- NXTTLC
       |----- NXTLCA
       |----- NXTTCR
              |----- NXTLCA
              |----- NXTLCU
              |----- NXTLCY
              |----- NXTLRS
              |----- NXTRTL
|----- NXTTLS
       |----- NXTLCA
       |----- NXTQPS
       |----- NXTTCR
              |----- NXTLCA
              |----- NXTLCU
              |----- NXTLCY
              |----- NXTLRS
              |----- NXTRTL
|----- NXTTNS
```

Figure 2 Structure of the routines `NXTACG` and `NXTTCG`.

This new geometry module also provides the following options which were not available in the `EXCELT:` module: 1) a Cartesian mesh discretization is now supported for cells containing pin clusters; 2) the pins can now be placed inside a pure Cartesian region; 3) the pins in a 3-D cell do not have to cover the global geometry; and finally 4) one can superimpose a Cartesian mesh over each pin in a cluster.

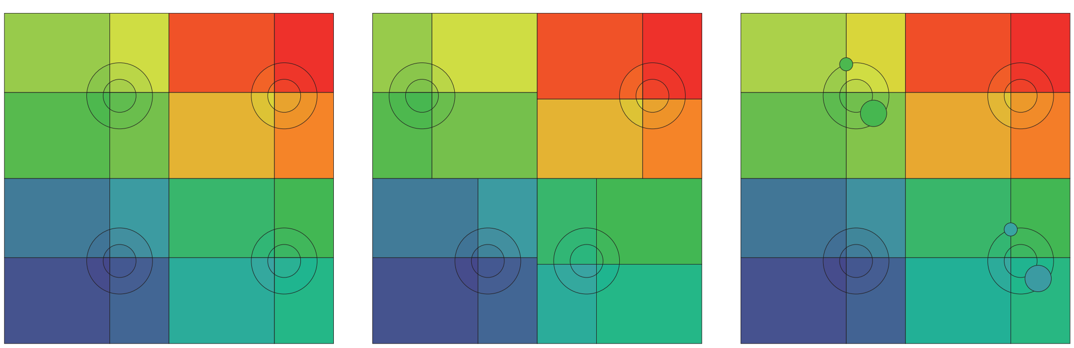
There are also some major differences in the tracking procedure used by the two modules. This is a consequence of the fact that the mesh does not have to match from cell to cell in an assembly. In `EXCELT:`, the complete geometry was tracked in a single sweep for each integration line. In the `NXT:` module, a multilevel tracking procedure is preferred. Accordingly, the global geometry is first tracked and the cells that are crossed by tracks identified. Then the cells are tracked, independently of the pin clusters. Finally the pins in each cluster are processed. The final integration line is then generated by superposing the line segments associated with pins over those associated with the cells that are then combined with the line segments from the remaining cells in the global geometry. A second observation is that cells that are automatically generated based on the symmetries of the geometry are explicitly defined when the `EXCELT:` module is selected. In the `NXT:` module, a cell that appears at multiple locations in the global geometry as a result of symmetry unfolding is only generated once. The symmetries in the geometry (translation, reflection and rotation) are simulated using a translation, reflection and rotation of the integration lines.

The `NXT:` module will be verified using the following procedure. We will first start with a global verification of the geometry analysis routines, controlled by `NXTACG`, based on the contents of the *tracking* data structures (this is the geometry information transferred to the other modules of DRAGON). This will include a comparison of the `NXT:` computed volume and surface area with analytically computed values as well as a comparison of the `NXT:` based *tracking* with an already validated `EXCELT:` *tracking* for identical geometries.

The second step will consist of a verification of the geometry tracking routines, controlled by `NXTTCG`, by examining the contents of the binary tracking file. This will include a comparison of the `NXT:` tracking lines with reference tracking lines generated analytically. Finally, a self-consistency study will also be performed using the `NXTTNS` subroutine to compare the regional volume and surface area evaluated using the binary tracking file and the reference values stored on the *tracking* data structure [11].

3. Implementation of the verification procedure

As we mentioned previously, the verification procedure we have implemented relies mainly on the analysis of the information stored in the DRAGON *tracking* data structure and in the tracking file. Accordingly, the first step consists in defining the 2-D and 3-D models that will catch the main geometry features that can be processed by the code:



(a) Model C2A1G (b) Model C2A1L (c) Model C2A2

Figure 3 Examples of 2-D assemblies that can be processed using the `NXT:` module.

- 2-D and 3-D assemblies of cells with a Cartesian mesh that extends throughout the assembly (a global mesh) and containing annular subregions compatible with the `EXCELT:` tracking module (see Figure 3 (a) for a 2-D example);

- 2-D and 3-D uniform assemblies of cells with a non-uniform Cartesian mesh (a local mesh) containing annular subregions (see Figure 3 (b) for a 2-D example);
- 2-D and 3-D assemblies of cells with a non-uniform Cartesian mesh containing annular subregions and pin clusters (see Figure 3 (c) for a 2-D example and Figure 4 for a 3-D example).

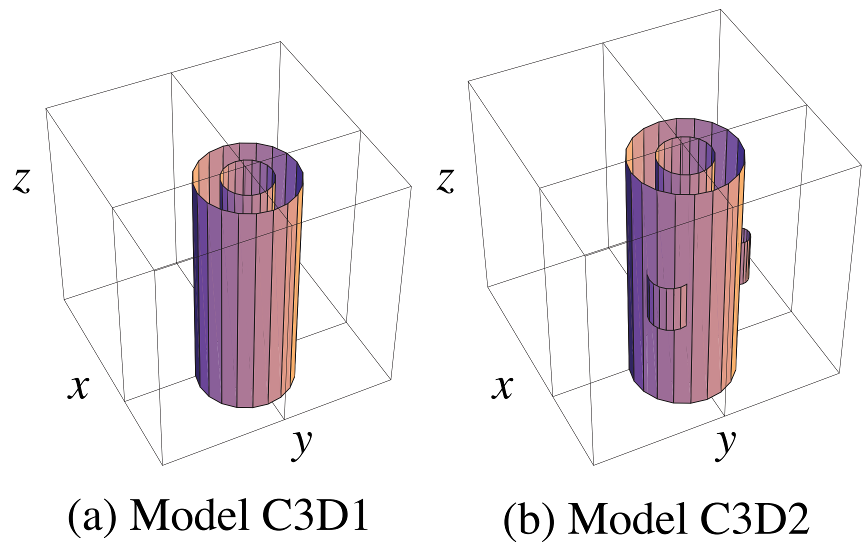


Figure 4 Examples of 3-D cells.

In order to simplify the verification procedure while preserving as much as possible its generality we proceeded in the following way. For the verification of the *tracking* data structure, we first selected the two reference 2-D cells presented in Figure 5. The volumes and area associated with each region and surface in these cells were then computed analytically. We then proceeded to extend these cells to three dimensions (see Figure 4 for a 3-D extension of geometry C2D2), the regional volumes and surface area being an extension of the 2-D results. Finally, using these cells, we built the 2-D assemblies illustrated in Figure 3 and a similar set of assemblies (see Figure 6) for 3-D geometries.

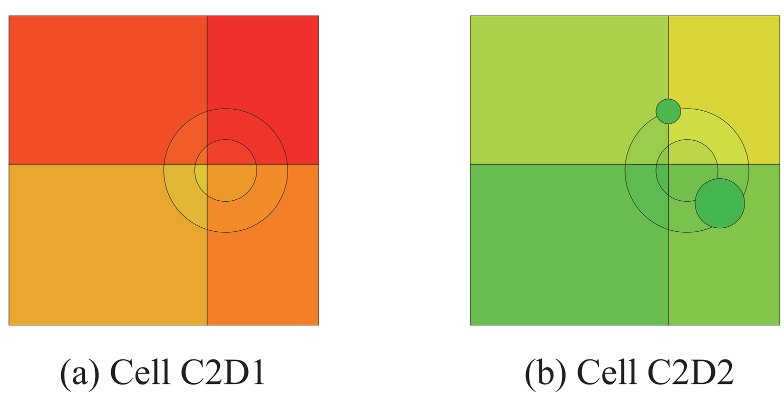


Figure 5 Cell geometries used for 2-D assembly construction.

The advantage of using this method is to use a single analytic evaluation of the regional volumes for two relatively simple geometries that possesses most of the features one will find in more complex geometries. These cells can then be used to generate very complex 2-D and 3-D assemblies that activate most of the routines programmed in the `NXT:` tracking module.

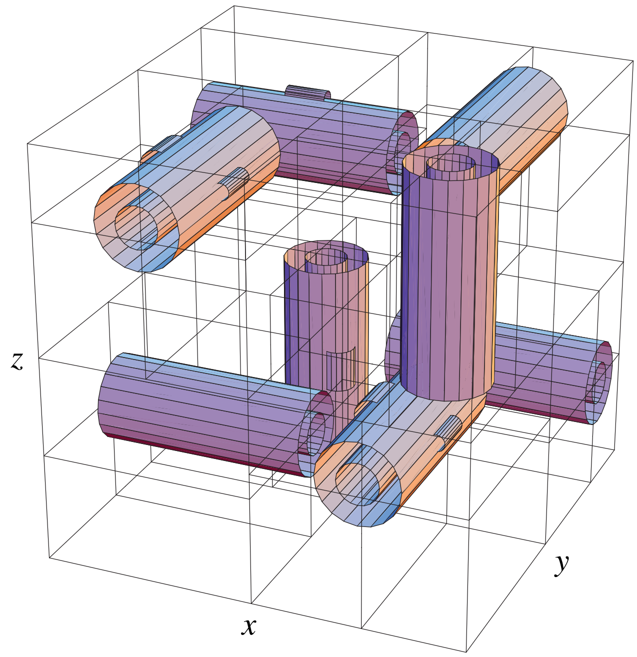


Figure 6 Example of a complex 3-D assembly.

Once the *tracking* data structure has been verified in this fashion, the second step, which consists of verifying the contents of the tracking file, can be undertaken. Here, two different techniques were considered. First, the intersection points between a tracking line, identified by a point (x_0,y_0,z_0) in space and its direction identified by the direction cosines (η,ξ,μ), and each of the surfaces defining the block used to built the geometry are evaluated analytically and compared with the points generated using DRAGON. Then the `TLM:` module is used to read the tracking file and to generate a Matlab-m file that can be used to illustrate these lines in colour in such a way that the region number associated with each segment can be easily identified. We also superposed over these lines an explicit drawing of the geometry (for 2-D geometries only) and the points computed analytically.

Finally, using the intrinsic capabilities of DRAGON we also computed numerically the volume associated with each region, based on the contents of the tracking file and used for segment length normalization, and compared these with the exact volumes. This will provide further proof that the `NXT:` tracking module is performing as expected.

4. Results

Here we will first examine the verification results associated with the geometry analysis part of the `NXT:` module and controlled by the routine `NXTACG`. Then we will illustrate the results we obtained when verifying the tracking file.

4.1 Geometry analysis results

The first geometries we considered are those illustrated in Figure 5. The regional volume computed for geometry C2D1 using the `EXCELT:` and `NXT:` modules are compared with the analytical volumes computed using an external program in Table 1. As we can see, the maximum error is of the order of 0.0001 % and both modules perform very well here. In fact these results are within the expected precision of the code for these types of calculations (single precision dimensioning data).

Region number	Analytical volume	`NXT:` volume	`EXCELT:` volume	Error (%) `NXT:`	Error (%) `EXCELT:`
1	0.0755765398	0.0755765438	0.0755765438	-5.32E-06	-5.32E-06
2	0.4842174988	0.4842174650	0.4842174650	6.98E-06	6.98E-06
3	7.7602059614	7.7602057500	7.7602057500	2.72E-06	2.72E-06
4	0.4164518170	0.4164517220	0.4164517220	2.28E-05	2.28E-05
5	0.7942166361	0.7942165730	0.7942166330	7.94E-06	3.88E-07
6	3.4693315469	3.4693315000	3.4693315000	1.35E-06	1.35E-06
7	0.0362472647	0.0362472981	0.0362472981	-9.21E-05	-9.21E-05
8	0.3838806090	0.3838806750	0.3838807050	-1.72E-05	-2.50E-05
9	7.2598721263	7.2598724400	7.2598724400	-4.32E-06	-4.32E-06
10	0.2571225419	0.2571226060	0.2571226060	-2.49E-05	-2.49E-05
11	0.6938797463	0.6938797830	0.6938797830	-5.29E-06	-5.29E-06
12	3.3689977118	3.3689975700	3.3689978100	4.21E-06	-2.91E-06

Table 1 Comparison of exact and DRAGON computed regional volumes for C2D1.

For the geometry C2D2, which contains annular pins and can only be processed using the `NXT:` module we obtained the results presented in Table 2. Again, we see that the relative errors on the volumes computed with DRAGON reach a maximum of 0.0001 %.

Region number	Analytical volume	Error (%) `NXT:`	Error `T1`(%)	Error `T2`(%)	Error `T3`(%)
1	0.502654825	-5.05914E-06	1.2	-0.77	0.001
2	0.125663706	-5.45901E-06	-3.5	-2.30	-0.008
3	0.075576540	-5.32308E-06	17.6	1.13	0.051
4	0.484217499	6.98033E-06	-1.4	0.13	-0.013
5	7.760205961	2.72415E-06	-0.1	-0.01	0.000
6	0.366199027	2.71956E-05	-0.9	-0.57	-0.011
7	0.419660483	2.24872E-05	0.3	0.50	0.030
8	3.391485666	6.35415E-06	-0.1	0.12	-0.003
9	0.036247265	-9.20787E-05	19.1	3.11	0.039
10	0.359893207	-1.54129E-05	-1.6	-0.13	-0.009
11	7.221027676	-2.41378E-06	-0.2	0.01	0.000
12	0.257122542	-2.4922E-05	-4.1	-0.30	-0.003
13	0.657704638	-2.03435E-06	1.2	-0.02	0.006
14	3.342340967	5.20593E-07	0.5	0.11	0.000

Table 2 Comparison of exact and DRAGON computed regional volumes for C2D2.

Finally we also considered the 2-D assembly C2D1L described in Figure 3. Here the volume of each region is computed independently in DRAGON while for the analytical results we used rely on the symmetry of the geometry. The results we obtained in this case are presented in Table 3 where for each cell, in addition to presenting the relative errors in volume, we also identify the regions in the cell associated with identical volumes. Here, the analytical results are identical to those presented in Table 1 for cell C2D1. Again the performance of the `NXT:` module for region volume evaluation and is very good. Note that this module also computes the area of the external surfaces. For the cases presented above, this represents only line segments and the maximum error observed is below 1.0E-6 %.

Cell 1		Cell 2		Cell 3		Cell 4	
Region	Error (%)	Region	Error (%)	Region	Error (%)	Region	Error (%)
1	-5.32E-06	22	1.52E-04	28	-5.32E-06	43	1.52E-04
2	6.98E-06	23	3.16E-05	29	6.98E-06	44	3.16E-05
3	2.72E-06	24	8.91E-06	30	2.72E-06	45	8.91E-06
4	2.28E-05	19	-6.01E-06	25	2.28E-05	46	-6.01E-06
5	7.94E-06	20	-7.04E-06	26	7.94E-06	47	-7.04E-06
6	1.35E-06	21	-5.57E-06	27	1.35E-06	48	-5.57E-06
7	-9.21E-05	16	1.03E-04	34	-9.21E-05	37	1.03E-04
8	-1.72E-05	17	1.38E-05	35	-1.72E-05	38	1.38E-05
9	-4.32E-06	18	2.29E-06	36	-4.32E-06	39	2.29E-06
10	-2.49E-05	13	-4.83E-05	31	-2.49E-05	40	-4.83E-05
11	-5.29E-06	14	-2.24E-05	32	-5.29E-06	41	-2.24E-05
12	4.21E-06	15	-1.00E-05	33	4.21E-06	42	-1.00E-05

Table 3 Comparison of exact and DRAGON computed regional volumes for C2A1L.

We have repeated the same type of analysis for geometry C3D1 and C3D2 (see Figure 4). In fact, geometry C3D1 is a simple extension to 3-D of C2D1 of height $z = 5$ cm and DRAGON produces, as expected, regional volumes that are the C2D1 volumes multiplied by z to within 0.0001%. In addition, all the surface area for model C3D1 can also be derived from the C2D1 surfaces area and volumes and are compatible with the DRAGON results to within 0.0001 % (slightly larger than for the 2-D case since some surfaces now involves annular regions).

For geometry C3D2, the problem we considered is somewhat more complex. The Cartesian and annular parts of the cell were again extruded to a height of $z = 5$ cm but the cylindrical pins were only extruded by 1 cm and located at a position z in the centre of the cell. As a result the outer surfaces area has the same value as for C3D1. On the other hand, the volume of each region is slightly more complex to evaluate but can again be obtained analytically by taking combinations of the C2D1 and C2D2 volumes. Our results indicate that the DRAGON results remain within 0.0001 % of the analytical predictions. In order to illustrate the behaviour of the `NXT:` module for cells and pins directed in every direction, we also analysed the assembly presented in Figure 6. This represents a combination of C3D1 and C3D2 geometries and the errors on the volume and surfaces area remain within 0.0001 % of the analytical results. Finally, we compared the

`NXT:` results for a 2-D and 3-D CANDU-6 cluster with analytical results. The maximum relative volume error obtained in 2-D is 0.0001 % while for 3-D, the error is slightly higher reaching a value of 0.0004%, which is still within the limit expected from a single precision cell description.

4.2 Tracking results

As we have noted before the contents of the tracking file was verified from two different points of view. First, for selected tracking lines we identified analytically the intersection points between a line and the known geometry and compared the results with those obtained using DRAGON. These results are illustrated in Figure 7 for geometry C2D1 and C2D2. Here, the intersection points computed analytically (illustrated by red dots) are superposed over the integration lines stored in the tracking file. The maximum error on the location of these points is 2 nm.

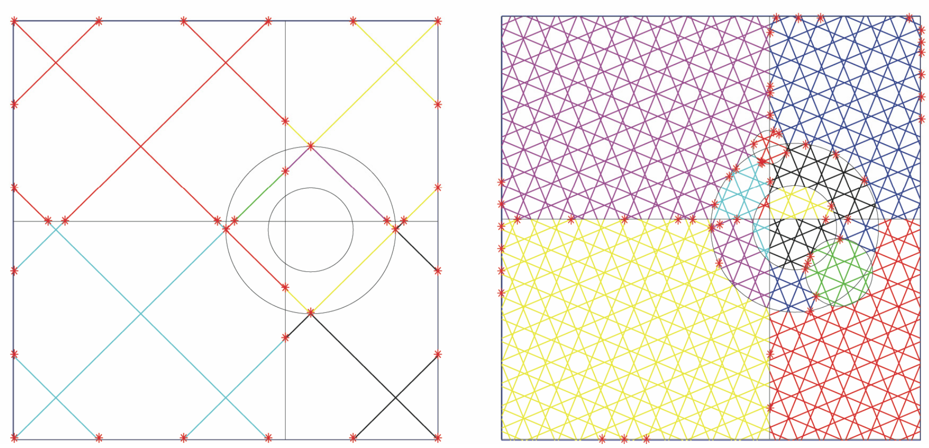


Figure 7 Tracking of C2D1 and C2D2.

The same procedure was also repeated with geometry C3D2 and the result is presented in Figure 8. Here we have illustrated all the points crossing three surfaces perpendicular to the z direction in addition to an arbitrary line in 3-D. As one can see, the regions crossed by the tracks are well identified (the pins appear only for plane z= 2.5 cm as expected) and the position of the analytical intersections points is correct since DRAGON succeeds in locating these points to within 2 nm.

Finally, DRAGON can also evaluate the region volumes using the information available on the tracking file. One can find in Table 2 for geometry C2D1 the numerical errors on these volumes as compared with the analytical volume. Here the tracking options `T1`, `T2` and `T3` represent sets of 2, 4 and 8 angles with track densities of 4.0, 10.0 and 100.0 respectively (`TISO` option). As one can see the errors decrease from a maximum of 19.1 % for the very coarse tracking to 0.05 % for our finest tracking. For the 3-D geometry C3D2, the errors are more important and we reach a maximum error of 49.5 % error for the coarse tracking and 1 % for the fine tracking. This is expected since for 3-D tracking, the density is in line per square cm rather than lines per cm and a 3-D density of 100 line/cm^2 is equivalent to a 2-D density of 10 lines/cm.

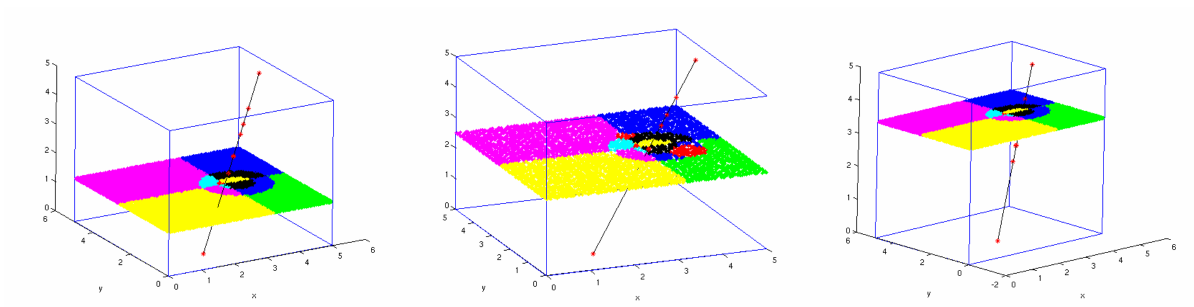


Figure 8 Intersection of tracking with plane at z= 1.5, 2.5 and 3.5 cm for cell C3D2 .

The verification of the tracking file for 2-D and 3-D CANDU-6 clusters geometries has also been performed with similar results. One can find in Figure 9 an illustration of the contents of the tracking file generated for the simulation of liquid zone controllers in a CANDU-6. The left hand

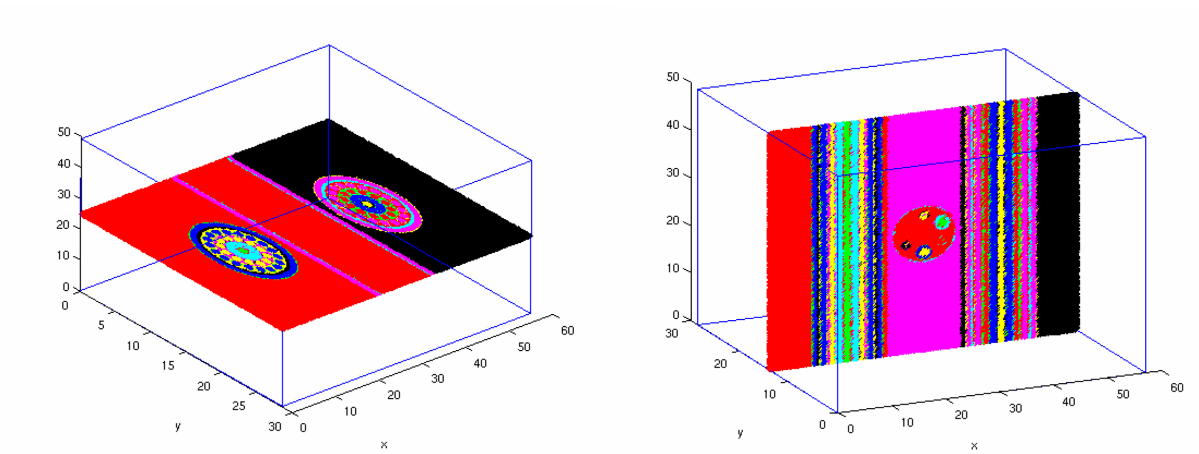


Figure 9 CANDU-6 LZC model.

side figure shows the intersection of the tracking lines with a plane located at $z = 25$ cm while the right hand side picture if for intersection with the plane $y = 14$ cm. We can easily identify from these pictures the location of the two fuel bundles as well as the structure of the liquid zone controller. This indicated that the tracking is performed adequately in DRAGON. This if further confirmed by the fact that the maximum error in the numerical volumes, computed using the tracking file (EQ$_8$ angular quadrature and 100 lines/cm^2) remain lower than 2.2 %.

5. Conclusion

The verification of NXT: has shown that the module performs as expected. The area and volume of surfaces and volumes computed in DRAGON are consistent with our analytical results (with single precision dimensions). The integration lines generated by this module are also correctly identified with the intersection point of a line with each surface defining the lattice located to within a few nm from the predicted analytical value. This precision is further confirmed by the fact that the approximate numerical volumes and surface areas evaluated using the contents of the tracking file are coherent (errors decrease with more refined quadrature sets) with the exact DRAGON predictions.

The next step in our verification plan will be to tackle the LIB: module which is use to generate the macroscopic cross sections associated with various mixture in the cell starting with different types of microscopic cross section libraries, including the latest WIMS-AECL NDAS format library.

6. Acknowledgments

This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada and by the CANDU Owner's Group.

7. References

[1] Marleau, G., Roy, R., Hébert, A., "New Computational Methods used in the Lattice Code DRAGON", *ANS Topical Meeting on Advances in Reactor Physics*, Charleston, South Carolina, March 8-11 (1992).
[2] Donnelly, J.V., WIMS-CRNL, *A User's Manual for the Chalk River Version of WIMS*, Report AECL-8955, Atomic Energy of Canada Limited (1986).
[3] Marleau, G., Hébert, A., Roy, R., "*A User guide for DRAGON 3.04*", Report IGE-174 Revision 5, École Polytechnique de Montréal (2000).
[4] Marleau, G., *DRAGON Verification Manual: Modules Required in the AECL Critical Path*, Report IGE-255, École Polytechnique de Montréal (2001).
[5] Marleau, G., "The Verification of DRAGON: Progress and Lessons Learned", *23rd Annual Canadian Nuclear Society Conference*, Toronto, Ontario, June 2-5 (2002).
[6] Marleau, G., Hébert, A., Roy, R., *A User Guide for DRAGON: Version 3.05C*, Report IGE-174 Rev. 6C, École Polytechnique de Montréal (2006).
[7] Hébert, A., Marleau, G., Roy, R., *A Description of the Data Structures for DRAGON 3.05C*, Report IGE-232 Rev 4C, École Polytechnique de Montréal (2006).
[8] Marleau, G., Hébert, A., Roy, R., *A User Guide for DRAGON: Version 3.06*, to be published, École Polytechnique de Montréal (2007).
[9] Wolfram, S., *The Mathematica Book* , 3rd ed. Wolfram Media/Cambridge University press (1996).
[10] Hanselman, D., Littlefield, B., "Mastering Matlab 6: A Comprehensive Tutorial and References", Prentice Hall (2001).
[11] Marleau, G. "DRAGON Theory Manual Part 1: Collision Probability Calculations", Report IGE-236 Rev. 1, École Polytechnique de Montréal (2001)