# NEW GEOMETRIC CAPABILITIES OF DRAGON

Guy Marleau
Institut de génie nucléaire,
École Polytechnique de Montréal
C.P. 6079, succ. Centre-ville, Montréal,
Québec, CANADA H3C 3A7
email: marleau@meca.polymtl.ca

## ABSTRACT

In this paper we describe the new 2-D and 3-D supercell geometries that can now be treated by the EXCELL module of the code DRAGON. We also present the new pre-homogenization module that can be used to reduce the dimension of the collision probability matrix. Finally we discuss the modifications to the code that were required to extend the use of periodic boundaries to a more general class of geometries.

## I. INTRODUCTION

Because the new computers currently available on the market have very large memory storage capabilities and computing power, problems previously considered too complex to solve with a lattice code are now being requested frequently by users. These new geometries are often needed for code validation purpose where typical experiments provide fine mesh reaction rate distributions inside a cell.[1] In addition, various code users require CANDU simulations that reflect more closely the actual changes in the core which take place during a loss of coolant accident.

In fact, the old version of DRAGON already had extensive geometry modeling capabilities.[2] For instance, it could be used to analyze 2-D and 3-D reactor assemblies of mixed Cartesian-cylindric cells as well as 2-D cluster cell similar to those found in a CANDU reactor. However, there were often restrictions in these geometry definition which seemed somewhat arbitrary. For instance, a 2-D Cartesian assembly containing cells with concentric annular sub-regions had to be defined in such a way that the center of each annulus coincides with the center of the Cartesian cell. We therefore decided to remove from the code a large number of geometric restrictions. As a result it was possible to implement in the EXCELL module of DRAGON a large number of new discretization options which can be useful for a fine mesh treatment of 2-D assembly or 3-D supercell geometry. In addition we have generalized the use of periodic boundary conditions, which was formerly available only for the analysis of 1-D geometry, to 2-D and 3-D geometries.

One problem which became rapidly apparent when such fine mesh discretization schemes are used in DRAGON is that the size of the collision probability matrix which increases as $N^2$ with $N$ the number of regions in the problem often becomes overwhelming.

Accordingly it was deemed useful to write for DRAGON a pre-homogenization module which could be used to reduce the number of independent regions to be treated in any given transport calculation. This module can be used to impose beforehand the same neutron flux distribution in different regions of the fine mesh geometry provided these regions all contain the same mixture. The specific choice of the regions to combine is left to the user and can be based on internal symmetries of the core or on the knowledge that the flux will vary slowly from one region to another.[3, 4]

In this paper we will describe the new geometries that can now be treated using the code DRAGON. We will also comments on the modifications to DRAGON that were required to implement these options in a consistent fashion.[5]

## II. THE DRAGON CODE

The code DRAGON is used to solve the transport equation for the neutron flux distribution inside a reactor cell or assembly.[2] It generally relies on the collision probability method to discretize spatially the multigroup transport equation to the form:[6]

$$\vec{\phi}^g = \tilde{\mathbf{p}}^g_{vv}\vec{Q}^g \tag{1}$$

where the vectors $\vec{\phi}^g$ and $\vec{Q}^g$ each have $N$ components $\phi^g_i$ and $Q^g_i$ which represent respectively the average neutron flux and source in region $V_i$ of the cell. The $N \times N$ collision probability matrix $\tilde{\mathbf{p}}^g_{vv}$ which includes the effect of the boundary conditions is generally evaluated using the relations

$$\tilde{\mathbf{p}}^g_{vv} = \mathbf{p}^g_{vv} + \mathbf{p}^g_{vs}\tilde{\mathbf{p}}^g_{ss}\mathbf{p}^g_{sv} \tag{2}$$

$$\tilde{\mathbf{p}}^g_{ss} = (\mathbf{A}^{-1} - \mathbf{p}^g_{ss})^{-1} \tag{3}$$

where $\mathbf{A} = A_{\alpha\beta}$ is the $\Lambda \times \Lambda$ reflection/transmission matrix giving the relation between the outgoing current at surface $S_\alpha$ and the incoming current at surface $S_\beta$. The collision, leakage and transmission probabilities have components which are given respectively by the relations

$$V_i p_{ij} = \frac{1}{\Sigma_i \Sigma_j}\int d\vec{\Omega}\int d\vec{S}F_{i,j} \tag{4}$$

$$V_i p_{i\beta} = \frac{1}{\Sigma_i}\int d\vec{\Omega}\int d\vec{S}F_{i\beta} \tag{5}$$

$$S_\alpha p_{\alpha\beta} = \int d\vec{\Omega}\int d\vec{S}F_{\alpha,\beta} \tag{6}$$

In the EXCELL module of DRAGON there are two independent options that can be used to evaluate the collision probability matrix $\tilde{\mathbf{p}}^g_{vv}$ for 2-D Cartesian problems, corresponding respectively to the standard and cyclic tracking methods while for 3-D problems, only the standard tracking method is currently available.

For a 2-D problem, the standard tracking method is based on the assumption that the specific geometry is isolated in space. The 4-D integral of Eqs. (4) to (6) can then be reduced to the form:

$$\int d\vec{\Omega}\int d\vec{S} = \frac{1}{2\pi}\int_0^{2\pi} d\varphi \int dh \tag{7}$$

such that for the cases when region $i$ is located at a position in space which is different from $j$ we will have

$$\begin{aligned} F_{i,j} &= \mathrm{Ki}_3[\Sigma_i l_i + \tau_{i+\frac{1}{2},j-\frac{1}{2}} + \Sigma_j l_j] - \mathrm{Ki}_3[\tau_{i+\frac{1}{2},j-\frac{1}{2}} + \Sigma_j l_j] \\ &- \mathrm{Ki}_3[\Sigma_i l_i + \tau_{i+\frac{1}{2},j-\frac{1}{2}}] + \mathrm{Ki}_3[\tau_{i+\frac{1}{2},j-\frac{1}{2}}] \end{aligned} \tag{8}$$

where $l$ and $\tau$ are both functions of $\varphi$ and $h$ as described in Figure 1. If region $i$ and $j$ are physically identical we will rather use:

$$F_{i,i} = 2(\Sigma_i l_i + \mathrm{Ki}_3[\Sigma_i l_i] - \mathrm{Ki}_3[0]) \tag{9}$$

Similar relations can also be obtained for $F_{i,\beta}$ and $F_{\alpha,\beta}$.

After replacing the integration over $\varphi$ and $h$ by simple trapezoidal quadratures thereby defining a specific integration line, DRAGON then undertakes the tracking process. This consists in identifying for a specific integration line the path length $l_i$ associated with each subregion of the problem and the initial and final surfaces intersecting the line. After this information has been generated for each line, the probability integration procedure boils down to a double sum over Bickley-Naylor functions. Once the collision, leakage and transmission probabilities have been evaluated for a region isolated in space, there remains the problem of applying the boundary conditions which involved the evaluation of the matrix

$$\tilde{\mathbf{p}}_{ss}^g = (\mathbf{A}^{-1} - \mathbf{p}_{ss}^g)^{-1} = (\mathbf{I} - \mathbf{A}\mathbf{p}_{ss}^g)^{-1}\mathbf{A}.$$

In the case where only reflection boundary conditions are considered (isotropic reflection at a surface), $\mathbf{A} = \mathbf{I}$ and the inversion procedure represented by the second term of the above equation can be performed using very simple methods because the matrix to invert is diagonally dominant. In the case where one or more surface is associated with a void boundary condition, the problem is slightly more complex. Instead of inverting the full matrix we will consider only the submatrix involving the reflective surfaces (which is again diagonally dominant) since according to the last term of the above equation, the components of $\tilde{\mathbf{p}}_{ss}^g$ associated with void surfaces will vanish identically. Finally, in the case where some of the surfaces are associated with periodic boundary conditions, then the matrix $\mathbf{A}$ is no longer diagonal, and the diagonal dominance of $(\mathbf{A}^{-1} - \mathbf{p}_{ss}^g)^{-1}$ is no longer ensured. Accordingly, an inversion procedure which includes pivoting is required. In the old version of DRAGON, the matrix inversion procedure which was used to compute $\tilde{\mathbf{p}}_{ss}^g$ did not include pivoting and could fail. It was replaced in the new version of DRAGON by a more robust matrix inversion procedure.

It is also possible to use in DRAGON the cyclic tracking method to analyze 2-D Cartesian problems.[7] In this case instead of considering the cell to analyze as being isolated in space, one uses the boundary conditions to unfold the cell to infinity. In the case where only void or albedo boundary conditions are considered this unfolding consists in reflecting each cell along its external surfaces. As a result, the period of the infinite lattice is twice the initial lattice pitch in each Cartesian direction. Since the explicit boundary conditions are already taken into account, one is therefore left with a

direct evaluation of $\tilde{\mathbf{p}}_{vv}^{g}$ using Eq. (4) where the 4-D integration is now replaced by a 3-D integration of the form:

$$\int d\vec{\Omega} \int d\vec{S} = \frac{1}{2\pi} \int_0^{2\pi} d\varphi \int dh \int_0^1 \sqrt{1-u^2} du \tag{10}$$

such that

$$\begin{aligned} F_{i,j} &= \left( \exp\left[ -\frac{\Sigma_i l_i + \tau_{i+\frac{1}{2},j-\frac{1}{2}} + \Sigma_j l_j}{\sqrt{1-u^2}} \right] - \exp\left[ -\frac{\tau_{i+\frac{1}{2},j-\frac{1}{2}} + \Sigma_j l_j}{\sqrt{1-u^2}} \right] \right. \\ &- \left. \exp\left[ -\frac{\Sigma_i l_i + \tau_{i+\frac{1}{2},j-\frac{1}{2}}}{\sqrt{1-u^2}} \right] + \exp\left[ -\frac{\tau_{i+\frac{1}{2},j-\frac{1}{2}}}{\sqrt{1-u^2}} \right] \right) \left( 1 - \exp\left[ -\frac{T_p \tau_p}{\sqrt{1-u^2}} \right] \right)^{-1} \end{aligned} \tag{11}$$

where $T_p$ represents the product of the albedo for each surface crossed by an integration line. Note that in the case where all the surface have void boundary conditions, $T_p = 0$ and Eq. (11) can be integrated analytically over $\mu$ to Eq. (8). The numerical quadrature required in this case is such that all the integration lines selected are periodic, their period corresponding to the optical length $\tau_p$. In the case where periodic boundary conditions are also considered, the simple unfolding of the cell by reflection which was implemented in DRAGON is no longer valid.

We therefore had to modify the tracking procedure of DRAGON to account for the fact that the unfolding by reflection will be used only for reflective or voided surfaces while the unfolding of the cell to infinity will be accounted for by cell translation in a direction normal to the surfaces having periodic boundary conditions. In addition we had to modify the definition of $T_p$ in DRAGON since in the case of albedo boundary conditions a neutron leaving one cell by an external surface would effectively cross only one external surface before entering a different cell. In the case where periodic boundary conditions are considered, two different surface are encountered.

For 3-D calculations, only the standard tracking procedure is currently available. We will therefore consider a cell isolated in space and use a 2-D trapezoidal quadrature for the $dx$ and $dy$ integration while for the angular integration we use a 2-D equal weight angular integration:

$$\int d\vec{\Omega} \int d\vec{S} = \frac{1}{4\pi} \int_0^{4\pi} d^2\Omega \int dx \int dy \tag{12}$$

with

$$\begin{aligned} F_{i,j} &= \exp[-(\Sigma_i l_i + \tau_{i+\frac{1}{2},j-\frac{1}{2}} + \Sigma_j l_j)] \\ &- \exp[-(\tau_{i+\frac{1}{2},j-\frac{1}{2}} + \Sigma_j l_j)] \\ &- \exp[-(\Sigma_i l_i + \tau_{i+\frac{1}{2},j-\frac{1}{2}})] \\ &+ \exp[-(\tau_{i+\frac{1}{2},j-\frac{1}{2}})] \end{aligned} \tag{13}$$

in the case where $i$ and $j$ are located at different spatial position and

$$F_{i,i} = 2(\Sigma_i l_i + \exp[-\Sigma_i l_i] - \exp[0]) \tag{14}$$

if region $i$ and $j$ are physically identical. Here $l$ and $\tau$ are now both functions of $\vec{\Omega}$ and $\vec{S}$ (see Figure 2).

In DRAGON, once the geometry tracking procedure has been completed, each track segment can be normalized. According to Eq. (4) the segment length can be used to evaluate numerically the volume $V_{i,k}^n$ associated with each region and each tracking angle $\Omega_k$ ($\varphi_k$ in 2-D) using the relation:

$$V_{i,k} = \sum_j l_j(\Omega_k)$$

Since $V_{i,k}^n$ will generally differ from $V_i$, each segment length can be corrected using

$$\tilde{l}_j(\Omega_k) = \frac{V_i}{V_{j,k}^n} l_j(\Omega_k)$$

## III. NEW DRAGON GEOMETRIES

One can find in Figure 3 a typical geometry which can be processed by DRAGON. Up to now there were some restrictions on the Cartesian mesh splitting which could be used with such geometries. For example it was possible for a cell containing a $Z$-directed cylinder to subdivide arbitrarily the mesh spacing in the $Z$ direction while in the $X$ and $Y$ directions the maximum Cartesian mesh splitting allowed represented a subdivision in 2 equal parts of the cell in these directions. Another restriction consisted in the fact that only Cartesian cell containing annular regions located at the center of the cell could be processed. Finally, it was also impossible for mixed Cartesian-cylindric cells with a mesh splitting to associate a different mixture to each individual region of the cell.

One of the problems encountered when a finer Cartesian mesh decomposition ($K \times L$) of such a cell is considered consists in determining the exact number of independent regions that will be associated with the cell. The second problem one faces is the evaluation of the exact volume associated with each region. As we noted in the previous section this volume is generally required for track length normalization. However, even if this normalization procedure in DRAGON could be by-passed, using approximate rather than exact volumes affects the value of the collision probability matrix and thereby the flux solution.

In the new version of DRAGON we have tackle these problems in the following way. Each 3-D Cartesian cell containing $N$ concentric cylinders of radius $r^n$ is first projected on a plane perpendicular to these cylinders. Then the resulting 2-D Cartesian cell is discretized according to the users specification and to each Cartesian sub-cell $(k, l)$ located at $x_1^k \leq x \leq x_2^k$ and $y_1^l \leq y \leq y_2^l$ are 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 while in the case where it is located inside the region of radius $r_n$ the volume of region $n + 1$ to $N + 1$ vanishes identically. Finally when the Cartesian mesh intersects a cylinder, one can compute the volume of intersection $\Delta V_n^{k,l}$ between the mesh $(k, l)_n$ and this 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 $y_j^j$.

Defining $V_j^k$ to represent the cylinder surface located to the left of the plane defined by $x_j^k$ we can write

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

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

where $V_j^k = V_j$ when $u_j = x_j^k - x_r$ is selected. Using $u_j = y_j^l - y_r$ in the above equation for $V_j$ results in $V_j^l$ which represents the cylinder surface located below the plane defined by surface $y_j^l$.

In the case where the lines $x = x_i^k$ and $y = y_j^l$ intersect inside the cylinder of radius $r_n$ we obtain

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

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 all the Cartesian regions associated with a 2-D mapping of the 3-D cell have been processed in the same way, the next step will consists 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 can easily be obtained.

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 to generate a DRAGON integration line.

## IV. PRE-HOMOGENIZATION MODULE

In the previous releases of DRAGON the tracking module already included an automatic pre-homogenization module. It was generally used to reduce the number of regions in an assembly in the cases where reflection symmetries were imposed as boundary conditions. The new pre-homogenization module can be used after the tracking of a DRAGON geometry to impose additional symmetries in the flux distribution or to combine independent regions where the flux will be assumed identical. This last option is particularly useful in the case where the number of independent regions in the geometry becomes very large.

This new pre-homogenization module can be used to re-define the fine mesh associated with each cell to a more convenient mesh. As an example, consider the 2-D DRAGON geometry described in Figure 4 which is assumed symmetric under a $\pi$ rotation of the cell. It can easily be transformed to the geometry Figure 5 using the DRAGON pre-homogenization module even if this last geometry cannot be explicitly analyzed using the DRAGON tracking module.

Typically the pre-homogenization module of DRAGON works in the following way. Assuming that a set of $I$ regions $i = N - I + 1$ to $N$ that contain the same mixture all see the same flux $\phi_I$ and source $Q_I^g$ the transport equation can be written in the form:

$$\begin{aligned}\phi_I &= \sum_{i=N-I+1}^{N} \sum_{j=1}^{N-I} \frac{V_i}{V_I} \tilde{p}_{ij}^g Q_j^g \\ &+ \sum_{i=N-I+1}^{N} \sum_{j=N-I+1}^{N} \frac{V_i}{V_I} \tilde{p}_{ij}^g Q_I^g\end{aligned}$$

we can then define for the set of regions $I$ the following collision probabilities:

$$\tilde{p}_{Ij}^g = \sum_{i=N-I+1}^{N} \frac{V_i}{V_I} \tilde{p}_{ij}^g$$

$$\tilde{p}_{II}^g = \sum_{i=N-I+1}^{N} \sum_{j=N-I+1}^{N} \frac{V_i}{V_I} \tilde{p}_{ij}^g$$

We can then write:

$$V_I p_{Ij} = \frac{1}{\Sigma_I \Sigma_j} \int d\vec{\Omega} \int d\vec{S} \sum_{i=N-I+1}^{N} F_{i,j}$$

$$V_I p_{II} = \frac{1}{\Sigma_I \Sigma_I} \int d\vec{\Omega} \int d\vec{S} \sum_{i=N-I+1}^{N} \sum_{j=N-I+1}^{N} F_{i,j}$$

Accordingly, the collision probability for the macro region $I$ can be obtained by summing the function $F_{i,j}$ over all zones $i$ and $j$ included in $I$. Note that $F_{i,j}$ will take different forms depending on the respective location of region $i$ and $j$. In fact, assuming all the region $i > N - I$ are disconnected, then the sum above will be carried out using an

expression for $F_{i,j}$ similar to that given in Eq. (13) except for the terms where $i = j$ where we will use Eq. (14). On the other hand, if some of the regions $i > N - I$ are physically connected, then we can further simplify the integration process. For example, if one assumes that region $N$ and $N - 1$ have a common surface, then each integration line crossing both regions will travel a distance $l_N + l_{N-1}$ inside region I. The contribution of these two regions to the collision probability $p_{II}$ will contain a term $F_{I,I}$ which is given by:

$$F_{I,I} = (F_{N,N} + F_{N,N-1}) + (F_{N-1,N} + F_{N-1,N-1})$$

which in 3-D yields:

$$\begin{aligned} F_{I,I} &= 2(\Sigma_I(l_N + l_{N-1}) \\ &+ \exp[-\Sigma_I(l_N + l_{N-1})] - \exp[0]) \end{aligned}$$

As expected this is the result we would have obtained if a single region of volume $V_I$ had been considered. In DRAGON the pre-homogenization module will therefore affect the tracking file in more than one way. First each fine mesh region number intersected by an integration line in the assembly will be replaced by its associated coarse mesh region number. Then the integration line will be compressed in such a way that a sequence of line segments all associated with the same coarse mesh region number $I$ will be replaced with a single track segment.

## V. NUMERICAL RESULTS

The geometry we will study is illustrated in Figure 6 where 4 large $Z$-directed cylinders containing fuel and a control rod represented by the small $Y$-directed cylinder are all immersed in heavy water moderator. The boundary conditions we will use are total reflection on the top and bottom $z$ faces and periodic boundary conditions in $x$ and $y$. This geometry will be simulated in DRAGON using the coarse mesh model described in Figure 7 where the fuel rods are divided into 3 regions containing respectively the fuel and coolant, the pressure and calandria tube and part of the moderator. Similarly, the adjuster rods will be divided into 3 concentric cylinders. The most external region will be associated with moderator, while the central ring represents the adjuster rod guide tube. The internal region contains the adjuster itself and can be fully inserted or extracted from the assembly. The Cartesian mesh containing the adjuster rod is subdivided into 2 equal parts in the $x$ and $z$ directions. An analysis of this cell using DRAGON results in a problem with 108 independent regions (since DRAGON automatically takes into account the simplifications implied by the symmetry plane in the $z$ direction).

This coarse mesh geometry could have been described explicitly using the old DRAGON tracking procedure provided the moderator properties associated with the cooled and voided cell are identical. However, because the cylindric regions then allowed had to be centered in each Cartesian sub-cell, the current $3 \times 3$ mesh in the $x - y$ plane would have had to be replaced by a $5 \times 5$ mesh in order to reposition the cylinder at the center of a cell. As a result the total number of regions would then have increased by 72.

By using the new pre-homogenization module of DRAGON we can reduced even further the size of the problem to be solved by observing that the geometry is invariant

under a rotation of $\pi$ around the $z$ axis. As a result the size of the coarse mesh problem in the new version of DRAGON can be reduce to 54 independent flux evaluation, to be compared with the 180 independent flux evaluation which would have been required with the old version of DRAGON.

We have also considered the case where the DRAGON calculations are performed using a much finer mesh where the fuel and adjuster cylinders are further subdivided into 7 sub-regions, while the $x$ and $z$ Cartesian mesh surrounding the adjuster rods is subdivided in 8 rather than 2 equal regions. This specific choice of mesh was selected because the variation of flux in the region surrounding the adjuster rod is expected to be much more important than in the moderator region. As a result, a problem involving 764 regions was generated, which after pre-homogenization (resulting from the $\pi$ rotation invariance) was reduced to 382 regions which is only about twice as large as the coarse mesh problem would have required in the old version of DRAGON.

Table 1: Reactivity worth of CANDU adjuster using various DRAGON model

| Model/ Mesh | $K_{\text{eff}}$ | | $\rho$ (mk) | CPU |
|---|---|---|---|---|
| | out | in | out-in | minutes |
| Old/Coarse | 1.1142 | 1.0745 | -33.3 | 345 |
| New/Coarse | 1.1153 | 1.0753 | -33.4 | 280 |
| New/Fine | 1.1108 | 1.0693 | -34.9 | 525 |

Once can find in Table 1 the results we obtained for the reactivity worth of the adjuster using the three models described above. As one can see the old DRAGON model with 180 regions is about equivalent to the new coarse mesh DRAGON model with only 54 regions. On the other hand the use of a fine mesh model in DRAGON indicates that the coarse mesh model is not really converged and that a finer mesh discretization is really necessary to ensure an adequate evaluation of the reactivity worth of this adjuster.

Table 2: Two groups incremental cross sections for adjuster rods

| Cross Section | Coarse mesh $\frac{\Delta\Sigma}{\Sigma}$ (%) | Fine mesh $\frac{\Delta\Sigma}{\Sigma}$ (%) |
|---|---|---|
| $\Sigma_a^1$ | -0.74 | -0.73 |
| $\Sigma_a^2$ | -6.85 | -7.23 |
| $\Sigma_s^{1\to 1}$ | -0.21 | -0.20 |
| $\Sigma_a^{1\to 2}$ | 0.15 | 0.15 |
| $\Sigma_a^{2\to 2}$ | -0.02 | -0.01 |

We also evaluated the 2 groups incremental cross sections resulting from the insertion of the adjuster on the homogenization cell. These are presented in Table 2. As one can see, the incremental cross section associated with the absorption cross section in the thermal group is about 0.4 % larger when the fine mesh model rather than the coarse

mesh model is considered. This shows the importance of adequately discretizing the 3-D supercell model.

## VI. CONCLUDING REMARKS

The use of the new geometry options of DRAGON combined with the pre-homogenization module allows the user to perform a fine mesh discretization of a cell while keeping the number of regions to be analyzed to a minimum. These improvements which have been developed for use in the context of the collision probability method are such that they can readily be used in the module of DRAGON which solved the transport equation using the method of characteristics.[8]

## ACKNOWLEDGMENTS

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

## REFERENCES

[1] B. Arsenault, R. Baril and G. Hotte, *Advances in Supercell Calculation Methods and Comparison with Measurements*, Fifth International Conference on Simulation Methods in Nuclear Engineering, Montréal, Canada, September 8-11, 1996.

[2] G. Marleau, A. Hébert and R. Roy, *New Computational Methods Used in the Lattice Code DRAGON*, Topical Meeting on Advances in Reactor Physics, Charleston, SC, March 8-11 1992; see also A. Hébert, G. Marleau and R. Roy, "Application of the Lattice Code DRAGON to CANDU Analysis", *Trans. Am. Nucl. Soc.*, **72**, 335 (1995); see also G. Marleau, A. Hébert and R. Roy, *A User's Guide for DRAGON*, Report IGE-174 Rev. 3, École Polytechnique de Montréal, Institut de Génie Nucléaire (1997).

[3] R. Roy, A. Hébert and G. Marleau, "A Transport Method for Treating Three-Dimensional Lattices of Heterogeneous Cells", *Nucl. Sci. Eng.*, **101**, 217-225 (1989).

[4] R. Roy, G. Marleau, J. Tajmouati and D. Rozon, "Modeling of CANDU Reactivity Control Devices with the Lattice Code DRAGON", *Ann. nucl. Energy*, **21**, 115-132 (1994).

[5] G. Marleau, *Fine Mesh 3-D Collision Probability Calculations Using the Lattice Code DRAGON*, 1998 International Conference on the Physics of Nuclear Science and Technology, Long Island, New York, October 5-8, 1998.

[6] E.E. Lewis and W.F. Miller, Jr., *Computational Methods of Neutron Transport*, John Wiley and Sons, New York (1984).

[7] R. Roy, "Anisotropic Scattering for Integral Transport Codes. Part 1. Slab Assemblies", *Ann. nucl. Energy*, **17**, 379-388 (1990); see also R. Roy, "Anisotropic Scattering for Integral Transport Codes. Part 2. Cyclic Tracking and its Application to

*XY* Lattices", *Ann. nucl. Energy*, **18**, 511-524 (1991); see also R. Roy, A. Hébert and G. Marleau, "A Cyclic Tracking Procedure for Collision Probability Calculations in 2-D Lattices", *International Topical Meeting on Advances in Mathematics, Computation and Reactor Physics*, pp 2.2.4.1-2.2.4.14, Pittsburgh, PA, April 28 – May 2, 1991.

[8] R. Roy, *The Cyclic Characteristics Method*, 1998 International Conference on the Physics of Nuclear Science and Technology, Long Island, New York, October 5-8, 1998.

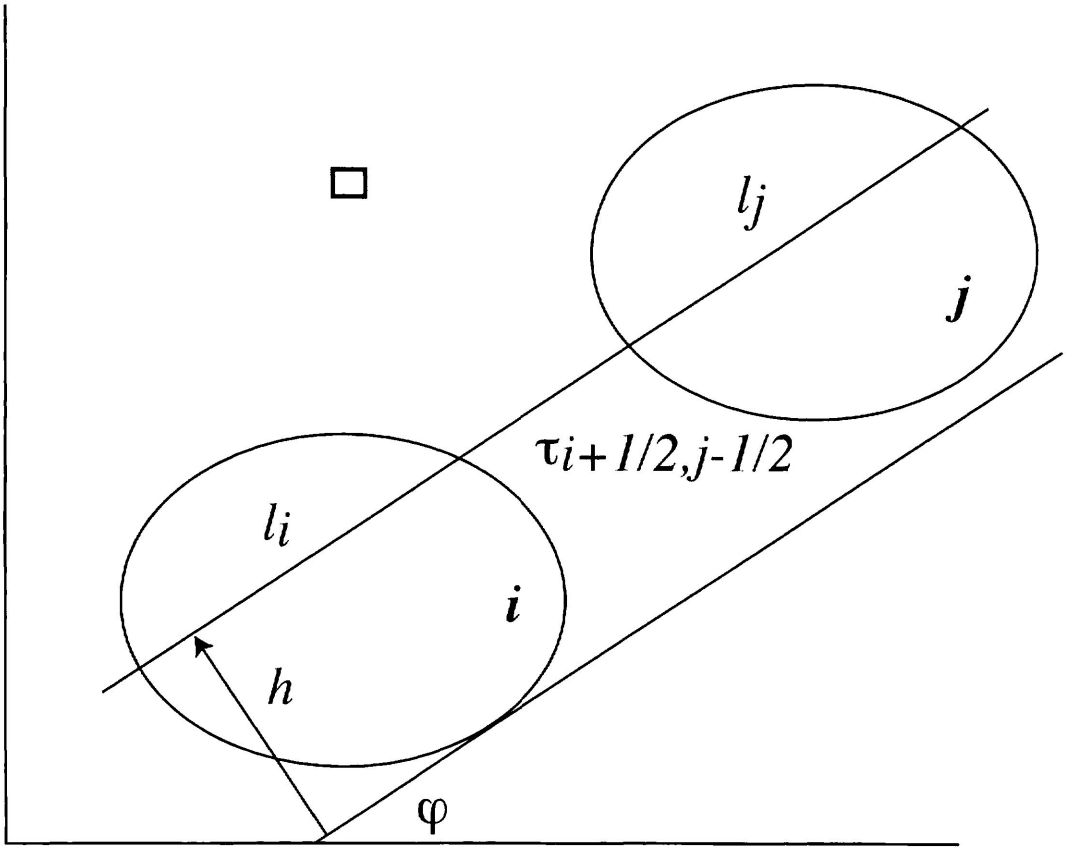


Figure 1: Integration parameters for general 2-D geometry

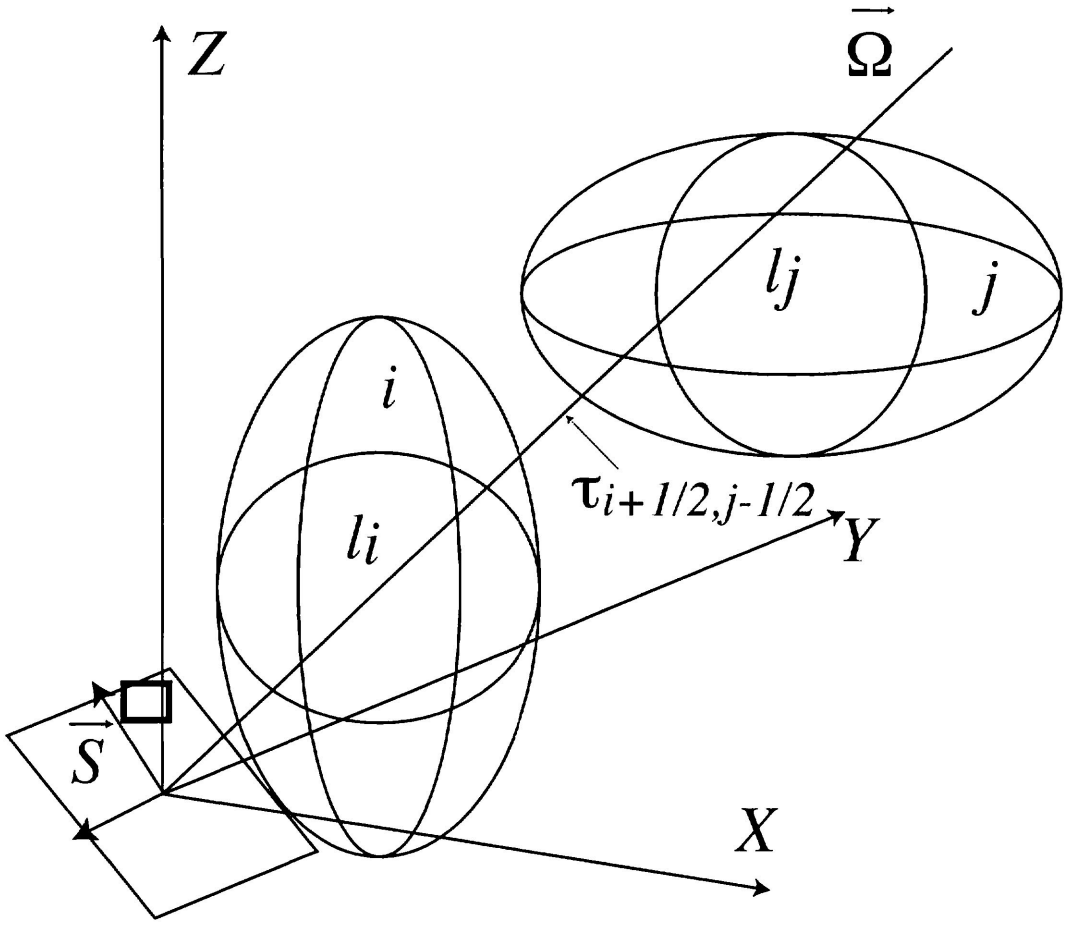


Figure 2: Integration parameters for general 3-D geometry

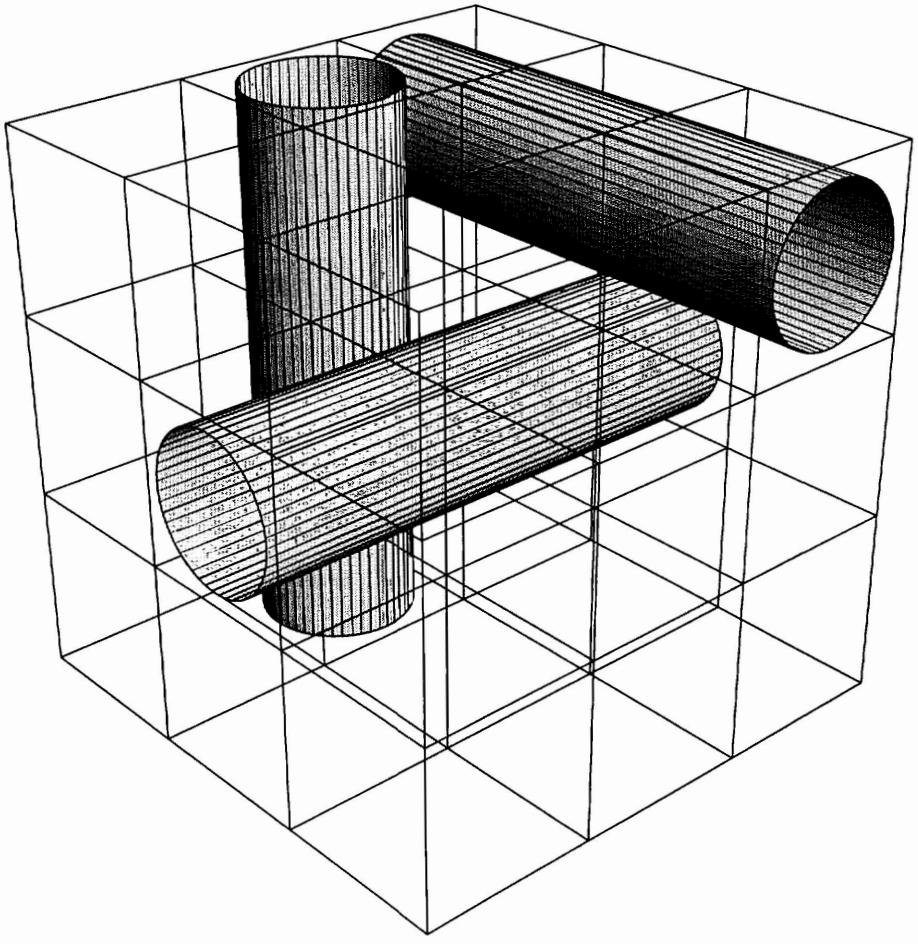

Figure 3: A typical 3-D DRAGON geometry

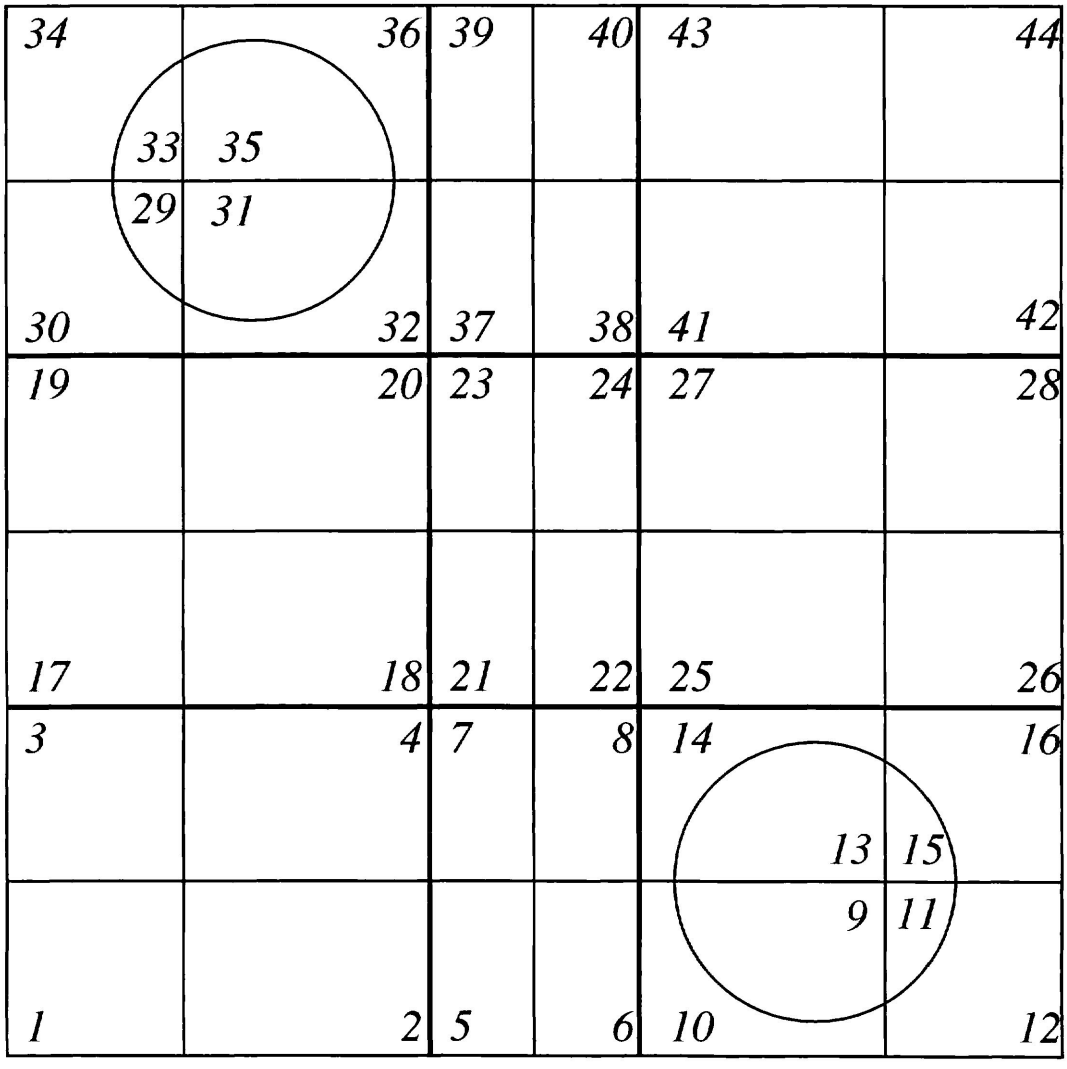


Figure 4: A valid DRAGON 2-D geometry

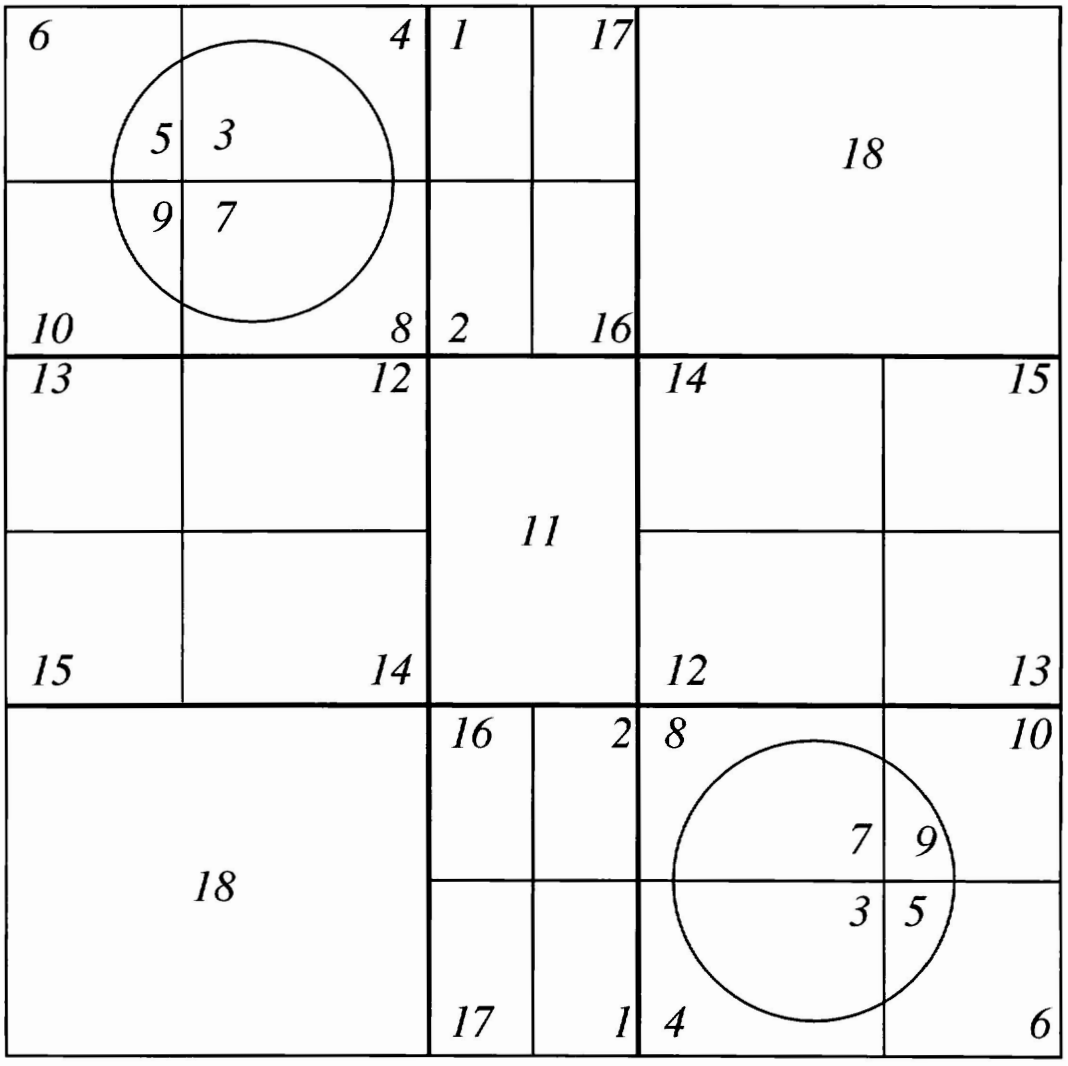


Figure 5: A pre-homogenized DRAGON 2-D geometry

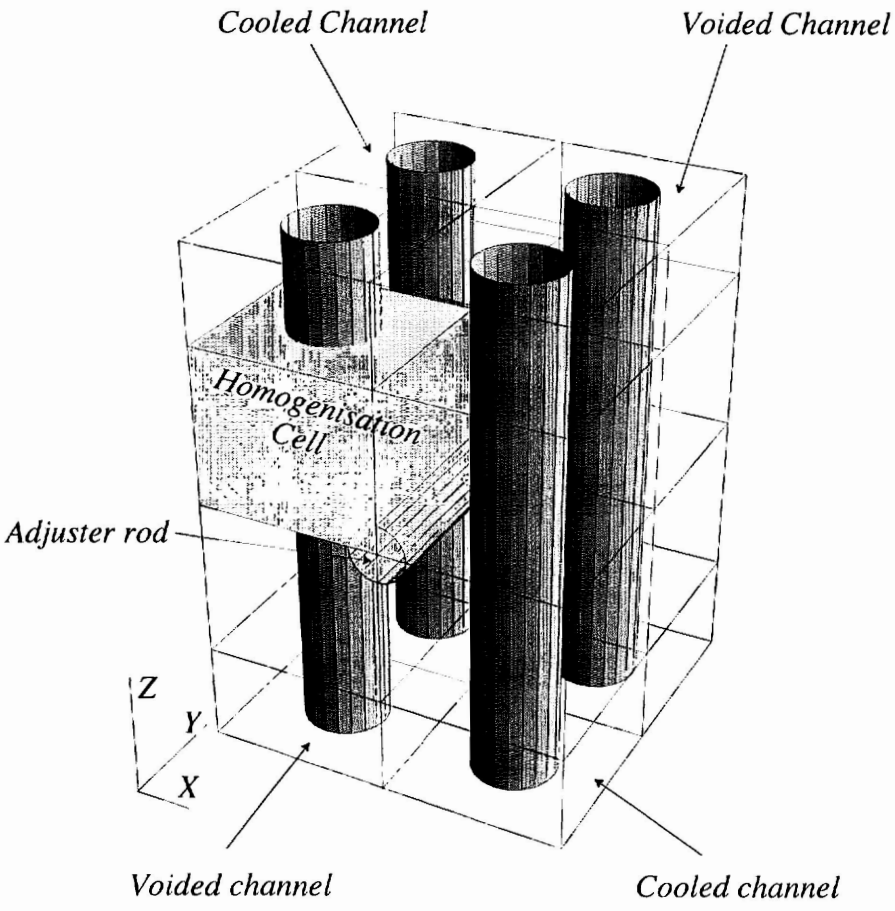


Figure 6: A 3-D CANDU assembly with adjuster rod.

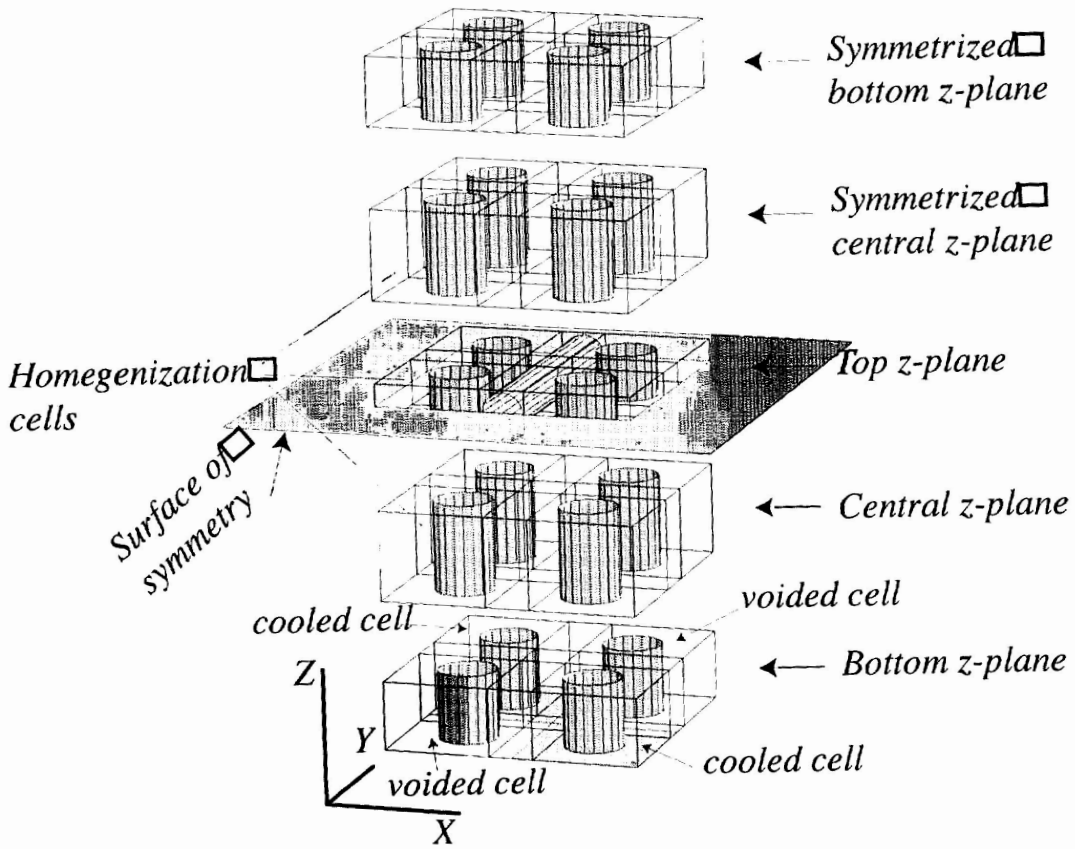


Figure 7: DRAGON coarse mesh model for a CANDU adjuster rod.