A New Characteristics Algorithm for 3D Transport Calculations

G.J.Wu and R. Roy
Institut de génie nucléaire, École Polytechnique de Montréal
P.O.Box 6079, Station CV, Montréal, Québec, Canada H3C 3A7

Abstract

In this paper, we present recent developments based on a characteristics method applied for solving general 3D geometries in the case of isotropic boundary conditions. Assuming isotropic sources and scattering, this characteristics solver involves the calculation of the region-to-region angular flux by scanning the tracking file containing the integration lines. The scalar flux is computed by collecting all mean angular fluxes in terms of the entering angular flux and the source of the region. At the boundary, the entering angular fluxes are linked to the emerging angular fluxes by isotropic albedos. The transport solution is similar to the one obtained by the standard collision probability method. The main advantage of this treatment is to get rid of collision probability matrices which have a dimensionality of the size of the square of the number of regions. For multigroup calculations, the rebalancing scheme was enhanced to take into account the external currents for any value of the albedos. Numerical comparisons are also presented in order to show the accuracy of the characteristics method as compared to the standard collision probability treatment for 3D supercells in the lattice code DRAGON. Several calculations for the incremental cross sections of adjusters and liquid zone controllers show that the characteristics results are accurate for the usual supercell calculations in a CANDU reactor.

1 Introduction

The usual deterministic method in DRAGON[1] for solving the 3D neutron transport equation in supercell geometry is the collision probability technique. In the EXCELT module, a sequential tracking file is constructed to cover all integration lines necessary to cover the 3D Cartesian domain; this is done by covering the supercell by arrays of parallel lines at several angles. The ASM module uses this tracking file to compute the collision, escape and transmission (corresponding to region-to-region, region-to-surface and surface-to-surface respectively) probabilities for every energy group. These probability matrices are then algebraically reduced by eliminating the currents for the multigroup system. In the newer EXCELL module, the tracking file does not have to be stored; this gives even faster solutions to 3D transport problems. As computing time and storage requirements increase as the square of the number of regions, the collision probability techniques can become prohibitive for large problems. However, we will now see that this is not the only way to solve these problems.

The aim of the present work is to combine the best features of the above methods, without the need to use matrices. As in every method of characteristics (MOC), the differential form of the Boltzmann equation will be used.[2] This differential equation will then be solved after following the "characteristics" of the system, which correspond with the tracking lines. Characteristics multigroup solvers are used since the seventies to obtain accurate solutions for 2D lattice cell problems. The CACTUS code (now part of the WIMSE package) was one of the first flexible and well-organized software to perform lattice calculations.[3] CACTUS uses the cyclic tracking technique where a characteristics line is followed after mirror reflexions or periodic replications up to a point where it coms back to its origin. Using this technique, there is no need to take into account the value of the currents at the boundary. Based on similar ideas, a new module called MOCC was also developed in DRAGON to treat lattice cell problems with specular boundary conditions.[4] This technology was found very accurate with respect to several benchmark problems, and we expect that similar accurate results could be obtained for 3D problems in the coming years. Nevertheless, the treatment of specular boundary conditions is slower than the usual simplified isotropic returns for surfaces at the external boundary. Recent attempts to use simplified boundary conditions between cells or at the external boundary have lead to

various other well-known codes. In the recent versions of CASMO (CASMO-4), the standard transmission probability treatment has been replaced by a characteristics solver (named KRAM).[5] Using a similar formalism, another characteristics solver was also implemented in the GTRAN2 code.[6] The CRX code was also based on the MOC and can be used to solve rectangular and hexagonal 2D lattices.[7] All these codes are still for 2D lattice cell calculations.

In CANDU reactors, we need to be able to solve 3D supercell problems where solid reactivity mechanisms are placed perpendicularly to the fuel channels. The advantage of the MOC approach over the collision probability one is that it requires less memory for large problems with many regions. In the next section of this paper, we will describe the MOC method. Section 3 will be devoted to the presentation of the multigroup iteration scheme. Numerical results are given in Section 4; these results include the study of incremental cross sections for stainless-steel adjuster rods and liquid zone controllers in a CANDU. It is shown that the results obtained by the new MCI module are consistent with the results of EXCELL. Conclusions and description of future work on that field are given in Section 5. An appendix is also provided in order to appreciate the various comparisons done with collision probabilities under various normalizations.

2 The characteristics formalism

2.1 generation of 3D tracks

Assuming a finite domain V split into homogeneous regions, each region having a volume V_j, the average (one-group) flux Φ_j is given by:

$$\begin{aligned} V_j\Phi_j &= \int_{V_j} d^3r \int_{4\pi} d^2\Omega\ \Phi(\vec{r},\hat{\Omega}) \\ &= \int_{\Upsilon} d^4T \int_{-\infty}^{+\infty} dt\ \chi_{V_j}(\vec{T},t)\ \Phi(\vec{p}+t\hat{\Omega},\hat{\Omega}) \end{aligned} \tag{1}$$

A characteristics line $\vec{T}$ (tracking line) is determined by its orientation (solid angle $\hat{\Omega}$) along with a reference starting point $\vec{p}$ for the line. To cover the Υ domain, Monte Carlo codes will use random generators. In most deterministic codes, a quadrature set of solid angles is selected (in DRAGON, an

EQ_n angular quadrature set sustained by uniform weights is currently used) and the starting point $\vec{p}$ is chosen by scanning the plane $\pi_{\hat{\Omega}}$ perpendicular to the selected direction $\hat{\Omega}$; the d^4T element is then composed of a $d^2\Omega$ solid angle element times the corresponding plane element d^2p. In the above, the variable t refers to the local coordinates on the tracking line and the function $\chi_{V_j}(.,.)$ is defined as 1 if the tracking line passes through the region j, and 0 otherwise.

For a chosen line $\vec{T} = (\hat{\Omega}, \vec{p})$, the required local data is a collection of segment lengths L_k and numbers N_k for each region encountered along the line. In order to later simplify the equations, these values will be ordered in the reverse travelling direction, ie $\{L_K, L_{K-1}, \ldots L_2, L_1, L_0\}$. Crossing points between regions and their corresponding angular fluxes are defined as:

$$\begin{aligned} \vec{r}_{k+1} &= \vec{r}_k - L_k\hat{\Omega} \\ \phi_k(\vec{T}) &= \Phi(\vec{r}_k, \hat{\Omega}) \end{aligned} \tag{2}$$

where $\vec{r}_0$ is the exit point of line $\vec{T}$ from the domain V and $\vec{r}_{K+1}$ the enter point. For each segment of the line, we define the integrated angular flux as:

$$L_k\,\bar{\phi}_k(\vec{T}) = \int_0^{L_k} dt\ \Phi(\vec{r}_{k+1} + t\hat{\Omega}, \hat{\Omega}) \tag{3}$$

Using this definition, the integrated flux becomes:

$$V_j\Phi_j = \int_{\Upsilon} d^4T \sum_k \delta_{jN_k}\, L_k\, \bar{\phi}_k(\vec{T}) \tag{4}$$

where δ is the usual Kronecker symbol, and where the summation of k runs over all integers. All characteristic lines are accepted, but only the contributions of segments crossing region j are added together. For a fixed solid angle $\hat{\Omega}$, we theoretically have that:

$$V_j = \int_{\pi_{\hat{\Omega}}} d^2p \sum_k \delta_{jN_k}\, L_k \tag{5}$$

Unfortunately, the estimated volumes $V_j'(\hat{\Omega})$ do not generally agree with the true volumes V_j. In deterministic codes, the segment lengths are usually renormalized to preserve the true volumes; this is done by multiplying L_k by the angular-dependent factor $V_j/V_j'(\hat{\Omega})$.

2.2 solving along characteristics

From (3), one can remark that only averaged angular flux $\bar{\phi}_k$ is needed to compute the region averaged flux Φ_j. So in this section, we will consider just one segment of a fixed tracking line $\vec{T}$, the index k is dropped for simplification.

Assuming an isotropic source of Q neutrons/cm^3/sec, the one-group neutron transport equation may be written:

$$\frac{d\phi(\vec{r}_0 + s\hat{\Omega}, \hat{\Omega})}{ds} + \Sigma_t(\vec{r}_0 + s\hat{\Omega})\phi(\vec{r} + s\hat{\Omega}, \hat{\Omega}) = \frac{Q(\vec{r}_0 + s\hat{\Omega})}{4\pi} \tag{6}$$

where $\vec{r}_0$ is a starting point, s the distance measured from $\vec{r}_0$ on the "characteristic" line which started from $\vec{r}_0$ and prolongate in the neutron direction $\hat{\Omega}$, ϕ is the angular flux along this line and Σ_t is the transport-corrected total macroscopic cross section.

For one line segment of length L and constant properties Σ_t and Q, we may integrate equation (6) along the line and obtain[3]:

$$\phi_{\text{out}} = \phi_{\text{in}} e^{-\Sigma_t L} + \frac{Q}{4\pi\Sigma_t}(1 - e^{-\Sigma_t L}) \tag{7}$$

where $\phi_{\text{in}} = \phi(\vec{r}_0, \hat{\Omega})$ is the inward value of angular flux at $s = 0$, $\phi_{\text{out}} = \phi(\vec{r}_0 + L\hat{\Omega}, \hat{\Omega})$ the outward value at $s = L$.

When $\Sigma_t \neq 0$, the average value in the segment of angular flux along the line can be given by:

$$\bar{\phi}(\hat{\Omega}) = \left(\phi_{\text{in}} - \frac{Q}{4\pi\Sigma_t}\right)\frac{1 - e^{-\tau_L}}{\tau_L} + \frac{Q}{4\pi\Sigma_t} \tag{8}$$

where $\tau_L = \Sigma_t L$, and when $\Sigma_t = 0$:

$$\bar{\phi}(\hat{\Omega}) = \phi_{\text{in}} = \phi_{\text{out}} \tag{9}$$

The outward flux ϕ_{out} of one segment also serves as inward flux ϕ_{in} in the next segment. If the inward flux of the first segment of one line $\vec{T}$ is known, a segment-by-segment calculation will be done to compute the outward flux by (7) and the averaged flux by (8) or (9) according to the region propreties.

For an iterative scheme, if only the isotropic reflection is considered for example, the outward flux of the domain will be summed at the end of each inner iteration on every surface in order to obtain the outward current, and the last one, after being multiplied by will be used as inward current for the next inner iteration after multiplying by the albedo factor.

3 Multigroup iteration scheme

To resolve a multigroup critical problem with this characteristic method, one should need the characteristic integration lines, provided by the DRAGON tracking module EXCELT. A flat flux is supposed to start the iterative resolution scheme. The last includes two levels of iteration: the inner loop and the outer loop(see Figure 1). As explained in the last section, each inner iteration corresponds with a sweep along all characteristics lines with the updated incoming currents of the last iteration.

In the inner loop of a perfect reflection transport calculation, a rebalancing scheme is usually performed in order to accelerate the inner iterations[12]. In our case, this rebalancing scheme is generalized to be suitable for any value of the albedo. For showing that, one can rewrite the equation (6) in his conventional multigroup form:

$$\begin{aligned} \hat{\Omega}\cdot\nabla\phi^g(\vec{r},\hat{\Omega}) \;+\; & \Sigma_t(\vec{r})\phi^g(\vec{r},\hat{\Omega}) = S^g(\vec{r}) \\ S^g(\vec{r}) \;=\; & \sum_{g'}\Sigma_s^{gg'}(\vec{r})\phi^{g'}(\vec{r}) + \frac{\chi^g(\vec{r})}{\text{Keff}}\sum_{g'}\nu\Sigma_f^{g'}(\vec{r})\phi^{g'}(\vec{r}) \end{aligned} \qquad (10)$$

One can supposed that the whole field V is divided into I regions V_i in which the properties Σ_t^g, $\Sigma_s^{gg'}$, and $\nu\Sigma_f^g$ are constants, and the surface ∂V into K sub-surfaces S_k. By integrating the above equation (10) over the whole domain and all directions, one obtains:

$$\sum_k\left(J^g_{\text{out},k} - J^g_{\text{in},k}\right) + \sum_i\left(\Sigma^g_{t,i}V_i\phi^g_i - \sum_{g'}\Sigma^{gg'}_{s,i}V_i\phi^{g'}_i\right) = \sum_i\chi^g_iF_iV_i \qquad (11)$$

where ϕ_i is the region averaged scalar flux and F_i is given by:

$$F_i = \frac{1}{\text{Keff}}\sum_{g'}\nu\Sigma^{g'}_{f,i}\phi^{g'}_i \qquad (12)$$

An average rebalancing factor α^g is computed for each unconverged energy group such that the rebalanced fluxes $\bar{\phi}^g_i$, and the rebalanced currents $\bar{J}^g_{\text{in},k}$, $\bar{J}^g_{\text{out},k}$ defined as:

$$\bar{\phi}^g_i = \alpha^g\phi^g_i \qquad \bar{J}^g_{\text{in},k} = \alpha^gJ^g_{\text{in},k} \qquad \bar{J}^g_{\text{out},k} = \alpha^gJ^g_{\text{out},k}$$

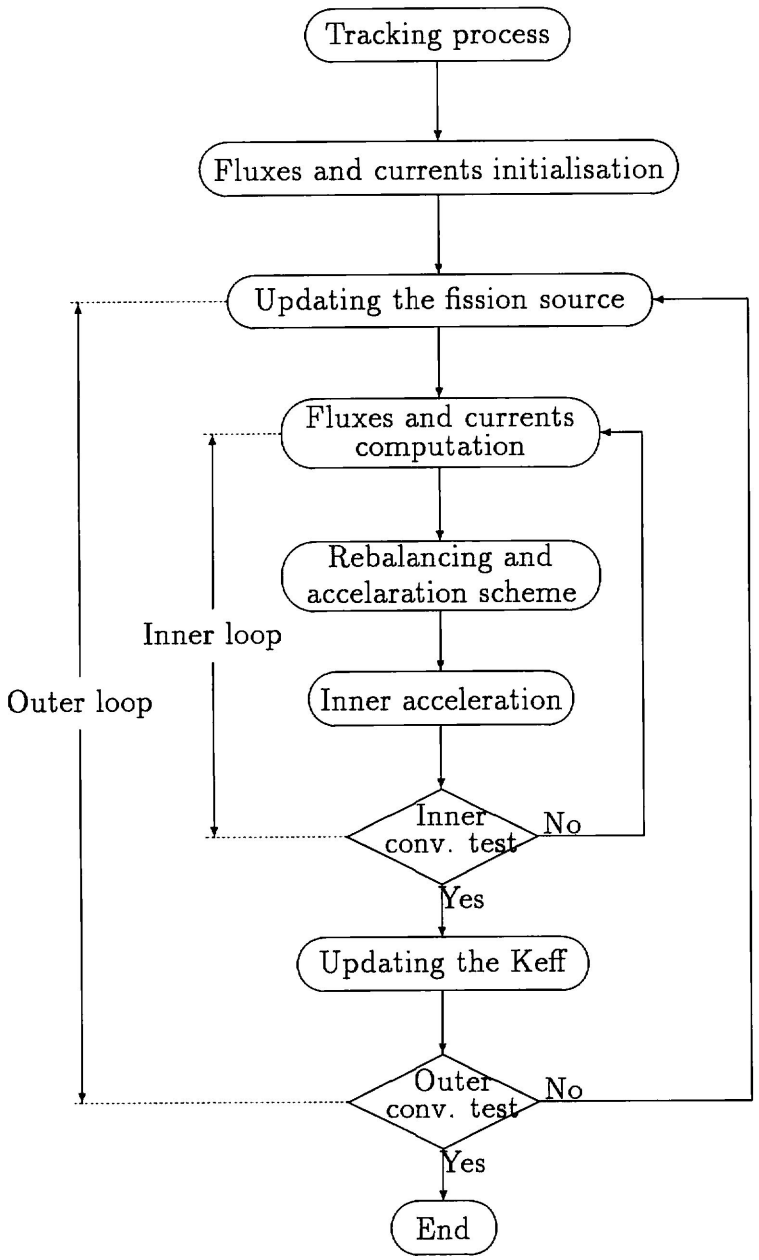

Figure 1: Iteration scheme for the MCI multigroup solver

satisfy equation (11). In fact, the $\bar{J}^g_{\text{in},k}$ is replaced by $\beta_k \bar{J}^g_{\text{out},k}$ since the inward current, calculated in the last iteration, is "older" than the fluxes and the outward current which are obtained at the current iteration. So we obtain the following system:

$$\left(\sum_k J^g_{\text{out},k}(1-\beta_k) + \sum_i \left(\Sigma^g_{t,i} - \Sigma^{gg'}_{s,i}\right) V_i \phi^g_i\right) \alpha^g$$
$$- \sum_i \sum_{g'} \Sigma^{gg'}_{s,i} V_i \phi^{g'}_i \alpha^{g'} = \sum_i \left(\chi^g_i F_i V_i + \sum_{g''} \Sigma^{gg''}_{s,i} V_i \phi^{g''}_i\right) \qquad (13)$$

where g' denotes the unconverged energy groups and g'' the converged ones.

The rebalanced fluxes and currents are then updated by an acceleration scheme before to be used in the next inner iteration. The inner loop will be ended when the inner convergence criteria are verified.

When the inner loop end, for the effective multiplicative factor calculation (TYPE K), the Keff is computed and the outer convergence criteria are checked. The fission source will be updated by the new Keff and the new fluxes values, and then a new inner loop will be started. This outer iteration will continue until the outer criteria are verified.

DRAGON supports also the bucking calculation (TYPE B), the bucking value is automatically searched after each inner loop in order to maintain the Keff to an imposed value which is usually the unit. The usual strategy used to obtained consistent reaction rates for diffusion models is to perform a critical buckling search. In that case, several options are possible:

- either use a homogeneous B_0 or B_1 model for leakage coefficients;
- the DB^2 correction factor can be included either on the left side (PNL calculation) or on the right side (SIGS calculation) of the transport equation.

The module MCI was developed to perform all these calculations, and can be used to generate nuclear properties: adirectional diffusion coefficients and macroscopic cross sections. We will now show that the results obtained by MCI are accurate with respect to the standard calculation scheme in DRAGON.

4 Numerical results

We consider now some 3D Gentilly-2 supercells. Each of these supercells is composed of two horizonal fuel bundles and one vertical adjuster with a symmetry factor 4.

Tracking is performed by EXCELT module of the lattice code DRAGON with isotropic reflection option. A EQ_4 angular quadrature and a 2.5 lines/cm^2 tracking density on the perpendicular plane were used.

Transport equation is then solved by a critical buckling search with a first order leakage treatment B1. Homogenization and condensation processes are then made with the resulting fluxes. Properties are condensed to 2 energy groups keeping the small up-scattering effect from the thermal to the fast group.

Six types of stainless steel adjuster rods are considered and two calculations have been performed for each adjuster rod type: one with the rod inside the supercell and one without the rod. The variations of the cross sections properties can be found in the Table 1-6.

The collision probability matrices are usually normalized in such a way as to satisfy exactly the neutron conservation laws. The following normalization schemes (see Appendix) are available in DRAGON :

- Gelbard algorithm (GELB) : used by default in DRAGON, but which may drive to negative probabilities;
- Non-linear multiplicative algorithm (NONL) : more expensive, no negative probabilities, usually more precise than Gelbard algorithm;
- Additive algorithm (HELI) : gives almost the same precision than NONL algorithm.

Each supercell calculation in this paper are performed four times: one with the characteristics method, the three others with the collision probabilities method but associated with different normalization algorithm. Results obtained with NONL option serve as reference, and that with GELB and HELI option are compared with the reference just like that of the characteristics method. The relative error for a calculated value of the incremental cross-section, presented in the results tables, is defined as follows:

$$\frac{\Delta\Sigma_{\mathrm{cal}} - \Delta\Sigma_{\mathrm{ref}}}{\Delta\Sigma_{\mathrm{ref}}} \times 100 \tag{14}$$

In the Table 1-6, one can see that the HELI scheme gives almost the same results than the reference, and the characteristics method gives more precise results than to the GELB scheme (which is the default option in DRAGON code). It's particularly interesting to note that there seems to be a systematic offset of the GELB scheme for $\Delta\Sigma_{s0}^{2\leftarrow 2}$ which range from 6% upto 10%; the offset pratically disapears when using either the HELI scheme or our new MCI module.

We have also performed several calculations for the liquid zone controllers (ZCU) whose geometry is more complex. It depends on the number of feeder tubes and scavenge tubes in the various control zones of the reactor. Because of the different number of scavenge and feeder tubes in various zones of a reactor, three types of ZCU must be modelled:

- Type 10: with 1 small scavenge tubes and no feeder,
- Type 21: with 2 small scavenge tubes and 1 large feeder,
- Type 32: with 3 small scavenge tubes and 2 large feeder,

A 6-region cylinderization was performed where the central and the fifth regions can be either voided or filled of water. For each type of the above ZCU, all these two cases are considered in the paper. On account of the complexity of the ZCU geometry, a EQ_6 angular quadrature and a 10.0 lines/cm^2 tracking density on the perpendicular plane are necessary. The computing procedure is similar than that for the adjuster supercells. The results of the Table 7 (Filled of water) and of the Table 8 (Voided of water) show that the characteristics method is usually more accurate than the standard Pij method (DRAGON using the GELB normalization). One can also remark that, for several types of $\Delta\Sigma$, sometimes the GELB normalization presents better approximations, but the absolute value of that $\Delta\Sigma$ in the concerned case is quite small (usually $\Delta\Sigma$ values $< 1.0^{-5}$), this effect is therefore not significant. Consequently, the MCI solution is comparable to the best available results obtained with the collision probability modules of DRAGON.

5 Conclusion and future work

The isotropic characteristics method presented above offers accurate solutions to 3D transport problems in heterogeneous geometries. The module MCI involves much simpler programming than the standard collision probability techniques used in DRAGON. Moreover, because there are no matrix treatment, the number of words necessary to perform a 3D calculation is greatly reduced. With such a characteristics solver, the size of the transport problem can thus be quite larger than with CP techniques. This module will be particularly useful and reliable for doing supercell calculations in CANDU reactors. In its current state, the MCI module supports most of the options included in DRAGON and is therefore available to successfully perform both Keff and critical buckling searches.

Some work is left to be done in order to get faster multigroup solutions. For relatively small problems with a limited number of regions, the CPU times observed with MCI are similar the ones of the EXCELL module (which is highly optimized). However, for larger problems, the MCI solution is slowed down by the current convergence. An obvious option would be to parallelize the multigroup treatment. Nevertheless, some research will be done to accelerate further the sequential version. In the coming years, a similar development will be done for 2D transport problems with isotropic boundary conditions, and the important aspect of rebuilding a self-shielding module also based on the characteristics methos will be considered. Moreover, the possibility of generating directional diffusion coefficients could also be implemented in the extended MCI module. It is expected that this characteristics solver will supersede the use of collision probabilities once fully integrated and validated in DRAGON.

Acknowledgments

This work has been carried out partly with the help of a grant from the Natural Science and Engineering Research Council of Canada.

Incremental cross-section	Char. method results	Char. Err.(%)	GELB Err.(%)	HELI Err.(%)
$\Delta\Sigma_t^1$	7.13050E-04	0.0418	0.1338	0.0042
$\Delta\Sigma_a^1$	1.25001E-05	0.0932	-0.0438	0.0000
$\Delta\Sigma_{s0}^{1\leftarrow 1}$	6.31483E-04	0.0244	0.2082	-0.0037
$\Delta\Sigma_{s0}^{1\leftarrow 2}$	1.69197E-07	0.0983	-1.0053	-0.0003
$\Delta\Sigma_t^2$	3.21597E-04	0.0556	0.7696	-0.0093
$\Delta\Sigma_a^2$	2.68039E-04	0.0951	-0.8081	-0.0002
$\Delta\Sigma_{s0}^{2\leftarrow 1}$	6.90471E-05	0.1167	-0.5688	0.0003
$\Delta\Sigma_{s0}^{2\leftarrow 2}$	5.33835E-05	-0.1184	8.7230	-0.0382

Table 1: Adjuster $\Delta\Sigma$ values for BCAINT

Incremental cross-section	Char. method results	Char. Err.(%)	GELB Err.(%)	HELI Err.(%)
$\Delta\Sigma_t^1$	5.81741E-04	0.0359	0.1076	0.0000
$\Delta\Sigma_a^1$	1.02519E-05	0.1718	-0.0398	0.0000
$\Delta\Sigma_{s0}^{1\leftarrow 1}$	5.14031E-04	0.0259	0.1787	-0.0032
$\Delta\Sigma_{s0}^{1\leftarrow 2}$	1.39506E-07	0.1271	-0.9299	0.0004
$\Delta\Sigma_t^2$	2.62022E-04	0.0569	0.6259	0.0114
$\Delta\Sigma_a^2$	2.22334E-04	0.1206	-0.7474	0.0004
$\Delta\Sigma_{s0}^{2\leftarrow 1}$	5.74859E-05	0.1197	-0.5220	-0.0004
$\Delta\Sigma_{s0}^{2\leftarrow 2}$	3.95438E-05	-0.3773	8.2295	-0.0570

Table 2: Adjuster $\Delta\Sigma$ values for BCAOUT

Incremental cross-section	Char. method results	Char. Err.(%)	GELB Err.(%)	HELI Err.(%)
$\Delta\Sigma_t^1$	1.31831E-03	0.0430	0.0701	0.0023
$\Delta\Sigma_a^1$	2.21216E-05	0.0959	-0.1744	-0.0011
$\Delta\Sigma_{s0}^{1\leftarrow 1}$	1.19216E-03	0.0309	0.1778	0.0008
$\Delta\Sigma_{s0}^{1\leftarrow 2}$	2.86756E-07	0.0970	-1.2463	-0.0006
$\Delta\Sigma_t^2$	5.43445E-04	0.0823	0.4061	0.0000
$\Delta\Sigma_a^2$	4.25977E-04	0.1139	-1.2117	0.0001
$\Delta\Sigma_{s0}^{2\leftarrow 1}$	1.04006E-04	0.1466	-1.1082	-0.0021
$\Delta\Sigma_{s0}^{2\leftarrow 2}$	1.17180E-04	-0.0334	6.2990	-0.0087

Table 3: Adjuster $\Delta\Sigma$ values for BCBINT

Incremental cross-section	Char. method results	Char. Err.(%)	GELB Err.(%)	HELI Err.(%)
$\Delta\Sigma_t^1$	1.06514E-03	0.0336	0.1064	0.0000
$\Delta\Sigma_a^1$	1.83062E-05	0.2014	-0.1071	-0.0006
$\Delta\Sigma_{s0}^{1\leftarrow 1}$	9.53219E-04	0.0291	0.2031	0.0009
$\Delta\Sigma_{s0}^{1\leftarrow 2}$	2.42814E-07	0.1021	-1.1340	0.0003
$\Delta\Sigma_t^2$	4.61280E-04	0.0970	0.6402	0.0000
$\Delta\Sigma_a^2$	3.72622E-04	0.0976	-1.0099	-0.0001
$\Delta\Sigma_{s0}^{2\leftarrow 1}$	9.36326E-05	0.0958	-0.8192	0.0004
$\Delta\Sigma_{s0}^{2\leftarrow 2}$	8.84104E-05	0.0922	7.6030	-0.0217

Table 4: Adjuster $\Delta\Sigma$ values for BCCINT

Incremental cross-section	Char. method results	Char. Err.(%)	GELB Err.(%)	HELI Err.(%)
$\Delta\Sigma_t^1$	3.86864E-04	0.0308	0.1618	0.0000
$\Delta\Sigma_a^1$	6.90389E-06	0.1080	0.0236	0.0000
$\Delta\Sigma_{s0}^{1\leftarrow 1}$	3.39511E-04	0.0228	0.2160	-0.0071
$\Delta\Sigma_{s0}^{1\leftarrow 2}$	9.50436E-08	0.0670	-0.8618	0.0000
$\Delta\Sigma_t^2$	1.78576E-04	0.0835	0.8685	0.0000
$\Delta\Sigma_a^2$	1.54355E-04	0.0697	-0.6110	0.0000
$\Delta\Sigma_{s0}^{2\leftarrow 1}$	4.04578E-05	0.0775	-0.3437	-0.0017
$\Delta\Sigma_{s0}^{2\leftarrow 2}$	2.41129E-05	0.0355	10.2027	-0.0984

Table 5: Adjuster $\Delta\Sigma$ values for BCCOUT

Incremental cross-section	Char. method results	Char. Err.(%)	GELB Err.(%)	HELI Err.(%)
$\Delta\Sigma_t^1$	5.90682E-04	0.0353	0.1009	0.0000
$\Delta\Sigma_a^1$	1.04083E-05	0.2242	-0.0493	-0.0011
$\Delta\Sigma_{s0}^{1\leftarrow 1}$	5.22047E-04	0.0184	0.1776	-0.0033
$\Delta\Sigma_{s0}^{1\leftarrow 2}$	1.41487E-07	0.1163	-0.9398	-0.0017
$\Delta\Sigma_t^2$	2.65568E-04	0.0224	0.5949	0.0112
$\Delta\Sigma_a^2$	2.25227E-04	0.0879	-0.7652	-0.0002
$\Delta\Sigma_{s0}^{2\leftarrow 1}$	5.81993E-05	0.1010	-0.5444	-0.0040
$\Delta\Sigma_{s0}^{2\leftarrow 2}$	4.02073E-05	-0.3943	8.1105	0.0201

Table 6: Adjuster $\Delta\Sigma$ values for BCDINT

Incremental cross-section	Char. method results	Char. Err.(%)	GELB Err.(%)	HELI Err.(%)
$\Delta\Sigma_t^1$	1.99492E-02	0.0486	-0.0756	-0.0002
	1.72636E-02	0.0509	-0.1510	0.0000
	1.48840E-02	0.0549	-0.1625	-0.0002
$\Delta\Sigma_a^1$	2.43045E-05	-0.2327	-2.1758	-0.0005
	2.12097E-05	-0.1403	-2.1278	-0.0006
	1.85197E-05	-0.3389	-2.2973	0.0006
$\Delta\Sigma_{s0}^{1\leftarrow 1}$	1.86061E-02	0.0474	-0.0656	-0.0001
	1.60927E-02	0.0503	-0.1408	0.0000
	1.38677E-02	0.0541	-0.1523	-0.0002
$\Delta\Sigma_{s0}^{1\leftarrow 2}$	-2.20621E-07	0.1742	-0.7137	0.0008
	-1.84629E-07	0.1050	-0.9682	0.0004
	-1.53766E-07	0.2199	-0.9929	0.0009
$\Delta\Sigma_t^2$	1.13716E-01	0.0712	-0.5081	0.0001
	1.02633E-01	0.0628	-0.5204	-0.0003
	9.12975E-02	0.0682	-0.5221	-0.0002
$\Delta\Sigma_a^2$	6.48049E-04	0.0599	-0.3967	0.0001
	5.84884E-04	0.0501	-0.4086	-0.0003
	5.20361E-04	0.0479	-0.4142	-0.0001
$\Delta\Sigma_{s0}^{2\leftarrow 1}$	1.31886E-03	0.0706	-0.1765	-0.0003
	1.14967E-03	0.0672	-0.2568	-0.0002
	9.97729E-04	0.0745	-0.2601	-0.0002
$\Delta\Sigma_{s0}^{2\leftarrow 2}$	1.13068E-01	0.0712	-0.5087	0.0001
	1.02049E-01	0.0629	-0.5210	-0.0003
	9.07773E-02	0.0684	-0.5227	-0.0002

Table 7: $\Delta\Sigma$ values of ZCU (Filled of water)

Incremental cross-section	Char. method results	Char. Err.(%)	GELB Err.(%)	HELI Err.(%)
$\Delta\Sigma_t^1$	-1.01640E-02	0.0173	0.1478	0.0000
	-7.73776E-03	0.0185	0.0890	0.0000
	-5.63818E-03	0.0169	0.0545	0.0000
$\Delta\Sigma_a^1$	-5.16954E-06	-2.0038	-0.2361	-0.0022
	-1.87347E-06	-4.6115	0.3912	0.0059
	1.06602E-06	13.9639	-3.0989	-0.0124
$\Delta\Sigma_{s0}^{1\leftarrow 1}$	-9.78485E-03	0.0171	0.1482	0.0000
	-7.50553E-03	0.0189	0.0892	-0.0003
	-5.53369E-03	0.0172	0.0543	-0.0003
$\Delta\Sigma_{s0}^{1\leftarrow 2}$	9.63423E-08	0.4906	-0.4615	-0.0002
	6.25603E-08	0.4162	-0.6357	-0.0038
	3.48246E-08	1.0519	-0.9956	-0.0011
$\Delta\Sigma_t^2$	-1.59801E-02	0.0312	-0.1229	0.0000
	-9.57009E-03	0.0374	-0.0931	0.0000
	-2.94524E-03	0.0283	0.0941	0.0000
$\Delta\Sigma_a^2$	2.68328E-05	0.3780	-1.1863	0.0017
	6.71591E-05	0.1375	-0.4374	0.0000
	1.08667E-04	0.1176	-0.2832	-0.0004
$\Delta\Sigma_{s0}^{2\leftarrow 1}$	-3.73974E-04	0.0568	0.1494	-0.0003
	-2.30375E-04	0.0599	0.0766	0.0009
	-1.05559E-04	0.1165	0.0200	-0.0003
$\Delta\Sigma_{s0}^{2\leftarrow 2}$	-1.60071E-02	0.0319	-0.1247	0.0000
	-9.63731E-03	0.0383	-0.0955	0.0000
	-3.05395E-03	0.0324	0.0808	0.0000

Table 8: $\Delta\Sigma$ values of ZCU (Voided of water)

Appendix. Review of PIJ normalizations.

To account for possible quadrature errors in the computation of collision probabilities, several normalization schemes were implemented for the PIJ doors of DRAGON. In the particular case of the EXCELL case, we end up with symmetric matrices of the shape:

$$\mathbf{Q} = \begin{pmatrix} T & E^T \\ E & C \end{pmatrix} \tag{15}$$

where the collision, escape and transmission probabilities obey some symmetry rules given by:

$$\begin{aligned} C_{ij} = V_i p_{ij} &= V_j p_{ji} = C_{ji} \\ E_{i\beta} = V_i P_{i\beta} &= \frac{S_\beta}{4} p_{\beta i} = E_{\beta i} \\ T_{\alpha\beta} = \frac{S_\alpha}{4} P_{\alpha\beta} &= \frac{S_\beta}{4} P_{\beta\alpha} = T_{\beta\alpha} \end{aligned} \tag{16}$$

Group-dependent conservation relations are normally given by the system:

$$\mathbf{Q}\ \vec{s} = \vec{g} \tag{17}$$

where the two vectors are defined by:

$$\vec{s} = \begin{pmatrix} 1 \\ \Sigma \end{pmatrix} \qquad \vec{g} = \begin{pmatrix} V \\ S/4 \end{pmatrix} \tag{18}$$

with $1, \Sigma, V$ and S containing ones, the transport-corrected cross sections, the volumes and the surfaces respectively.

Unfortunately, these conservation laws are not always **numerically** satisfied. With normalization schemes, we can obtain a new symmetric matrix $\bar{\mathbf{Q}} = \mathbf{Q}$ (normalized) which will satisfy (17). We are thus interested in constructing normalization schemes so that:

$$\bar{\mathbf{Q}}\ \vec{s} = \vec{g} \tag{19}$$

We will now revise three such normalization schemes.

A.1 homogeneous normalization (of Gelbard)

This method uses a correction for probability matrices trying to preserve the homogeneous limit.[10] Defining:

$$\bar{g} = \sum_k s_k \, g_k \quad \text{et} \quad \bar{R} = \frac{1}{\bar{g}} \sum_k s_k \, R_k \tag{20}$$

this method leads to the following definition of normalized coefficients:

$$\bar{Q}_{lm} = Q_{lm} + \frac{\{g_l R_m + g_m R_l - g_l g_m \bar{R}\}}{\bar{g}} \tag{21}$$

This is the current default normalisation scheme in DRAGON; in the ASM module, the appropriate keyword is GELB. It is suitable to normalize even void regions, but it could also yield to some negative unphysical values $\bar{Q}_{lm} < 0$.

A.2. multiplicative normalization

The idea here is to impose weights $w_l \approx 1$ that will perturbe the system symmetrically to yield new coefficient of the form:

$$\bar{Q}_{lm} = w_l w_m Q_{lm} \tag{22}$$

Insertion of this relation into (19) leads to the following non-linear system:

$$F(w) = w_l \sum_k w_k Q_{lk} = g_l \tag{23}$$

which is solved by a Newton-Raphson method. This is probably the best scheme; it always gives positive probabilities. However, it is not often used because of its computing cost. This option is supported in DRAGON under the keyword NONL.

A.3. additive normalization (Villarino)

A few years ago, a new scheme was developed in the code HELIOS in order to symplify the last scheme, while keeping its nice properties.[11] The idea is to use new weights $z_l \approx \frac{1}{2}$ to perturbe the system in the following way:

$$\bar{Q}_{lm} = \{z_l + z_m\} Q_{lm} \tag{24}$$

After insertion into (19), it can be easily shown that Gauss-Seidel iterations are very efficient to provide the converged solutions. Thus, we define initial values of the weigths $z_l^{(0)} = \frac{1}{2}$, and then compute the successive iterations as:

$$z_l^{(n+1)} = \frac{g_l - \{\sum_{m<l} Q_{lm} s_m z_m^{(n+1)} + \sum_{m>l} Q_{lm} s_m z_m^{(n)}\}}{Q_{ll} s_l + \sum_m Q_{lm} s_m} \tag{25}$$

and this scheme can be easily accelerated by a dynamic parameter. This scheme was implemented in the newer version of DRAGON under the keyword HELI. It generally gives results as accurated as the NONL scheme at an affordable cost.

References

[1] G. Marleau, A. Hébert and R. Roy, "A User's Guide for DRAGON", Report IGE-174 Rev. 3, École Polytechnique de Montréal (December 1997).

[2] J. R. Askew, "A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries", Report AEEW-M 1108, United Kingdom Atomic Energy Establishment, Winfrith (1972).

[3] M. J. Halsall, "CACTUS, A Characteristics Solution to the Neutron Transport Equation in Complicated Geometries," Report AEEW-R 1291, United Kingdom Atomic Energy Establishment, Winfrith (1980).

[4] R. Roy, "The Cyclic Characteristics Method", Int. Conf. Physics of Nuclear Science and Technology, Long Island, October 5–8, 1998.

[5] D. Knott, M. Edenius, "Validation of the CASMO-4 Transport Solution", Int. COnf. Math. Methods and Supercomputing in Nuclear Applications, Karlsruhe, April 19–23, 1993.

[6] L. Goldberg, J. Vujic, A. Leonard, R. Stachowski, "The Characteristics Method in General Geometry", *Trans. Am. Nucl. Soc.*, **73**, 173-174 (1995).

[7] S. G. Hong and N. Z. Cho, " CRX: A Code for Rectangular and Hexagonal Lattices Based on the Method of Characteristics, " *Ann. Nucl. Energy* **25**, 545-565 (1998).

[8] R. Roy *et al*, "A Transport Method for Treating Three-Dimensional Lattices of Heterogeneous Cells", *Nucl. Sc. Eng.*, **101**, 217-229 (1989).

[9] R. Roy *et al*, "Modelling of CANDU Reactivity Control Devices with the Lattice Code DRAGON", *Ann. Nucl. Energy*, **21**, 115-132 (1994).

[10] E. M. Gelbard, "Refinements in Computation of Collision Probabilities", Int. Mtg. on Adv. in Nuclear Engineering and Computational Methods, Knoxville (Arpil 1989).

[11] E. A. Villarino *et al*, "HELIOS: Angularly Dependent Collision Probabilities", *Nucl. Sc. Eng.*, **112**, 16-31 (1992).

[12] R. Roy. Théorie des probabilités de collision. Technical Report IGE-235, Institut de Génie Nucléaire, École Polytechnique de Montréal, March 1998.