An integral multi-domain DP_N operator as acceleration tool for the method of characteristics in unstructured meshes

Simone Santandrea

AN INTEGRAL MULTI-DOMAIN DP_N OPERATOR AS ACCELERATION TOOL FOR THE METHOD OF CHARACTERISTICS IN UNSTRUCTURED MESHES

Simone Santandrea
Commissariat à l'Energie Atomique
Direction de l'Energie Nucléaire
Services d'Etudes de Réacteurs et de Mathématiques Appliquées
CEA de Saclay, DM2S/SERMA
91191 Gif-sur-Yvette cedex, France

simone.santandrea@cea.fr

Total number of pages : 29

Number of figures : 2

Number of tables : 3

ABSTRACT

The paper presents the recent developments of the acceleration techniques for the Method Of Characteristics (MOC) in the code APOLLO2. The main contribution concerns the introduction of a multi-domain DP_N technique where all regions belonging to the same macro-domain are coupled by an integral operator that is strictly equivalent to the MOC. Different macro-domains are coupled via currents that are defined with the DP_N formalism. This new Integral DP_N (IDP_N) operator is build by using transmission and escape probabilities factors that respect symmetries/anti-symmetries and complementary properties that are enforced to preserve the physics of the problem and to save computational effort. These factors are computed using the numerical tracking of the MOC operator. The paper presents results on realistic assembly calculations that demonstrate the effectiveness of the IDP_N operator as a synthetic acceleration tool.

1 INTRODUCTION

In this paper we introduce an evolution of the DP_N operator currently used in the two dimensional MOC module of the APOLLO2 code [1] (called TDT) to accelerate the free iterations for the MOC. [2,3,4,5,6] Much work has been dedicated in recent years to the theme of acceleration techniques for the MOC. The reason for all this interest is the capability of the MOC to solve unstructured geometry problems without using any homogenization procedure. As with every method that uses the source iteration algorithm, the MOC can be very slow for collision dominated problems. Therefore acceleration techniques are essential in order to permit a wider practical application of the MOC. The DP_N method, originally introduced by Sanchez and Chetaine [7], is the operator to be used in the framework of the synthetic acceleration procedure for TDT. Generalizations of this technique have been recently introduced in [8,9] to improve the physical approximations of the initial formulation and to speed up the solution procedure. Further improvements are presented here and results are shown to illustrate the speed up factors of the TDT code (which is currently used for industrial calculations as a flux solver of the APOLLO2 [1] code system).

The principal contribution of this paper is a new way of using the DP_N acceleration. Much alike the classical Interface Current (IC) [10] treatment of the Collision Probability Methods (CPM), we show that the MOC admits a similar Integral DP_N (IDP_N) treatment. In this approach one subdivides the system into macro-domains in each of which one applies an integral formalism that does not introduce any supplementary approximation with respect to the MOC. The coupling between regions belonging to different macro-domains is done via interface DP_N currents that result from some simplifying assumptions on the interface flux anisotropy. These assumptions are the only approximations of the method with respect to the MOC. One can expect this operator to behave better than the classic DP_N operator, where the DP_N approximations were made at the interfaces between every region and not only at the interfaces between macro-domains. The main differences with respect to the standard IC method are that our method is not based on the CPM but on the MOC, that the interface approximations are given by the DP_N method and that the resulting equations have a response-matrix form that is different from that of the IC one. Since the numerical derivation of the IDP_N method is consistent with the MOC, one can expect that this method will be a good acceleration tool. We have not investigated, as it has been done for the IC method, the behavior of the IDP_N method as a stand-alone operator. Only its capability as a means to accelerate the MOC free iterations

is considered here. Moreover, even if the IDP_N method theory is valid also for three dimensional unstructured geometries, our practical implementations are limited to two dimensional systems, due to the actual capabilities of TDT.

The paper is organized as follows. In section 2 we recall the basis of the MOC and introduce some notation for the following sections. In section 3 the IDP_N synthetic operator is introduced and its detailed properties are discussed together with the computational algorithm. In section 4 the external synthetic acceleration is revised and discussed. Section 5 is devoted to the analysis and discussion of results of calculations for a realistic assembly. Conclusions follow. Appendix A and B are dedicated to the description of improvements for the treatment of the equations for the boundary conditions, and to show many symmetries/anti-symmetries and reciprocity relations of the IDP_N. We also demonstrate that this operator is robust in the limit of nearly voided media.

2 THE METHOD OF CHARACTERISTICS

The DP_N acceleration equations are used as a low-order operator for the acceleration of the MOC. [7,8,9] The equations are obtained by the same projective technique that is used to derive the transport MOC equations but over different representation functions. We present here the transport MOC equations and in the next section the low-order DP_N equations. Details can be found in [11,12].

The method of characteristics provides an iterative solution for the discrete ordinate formulation of the transport equation in a domain composed of unstructured homogeneous regions. The MOC equations are based on two approximations. The first consists of introducing a region-based flat spatial representation for the source term :

$$q(\mathbf{r},\mathbf{\Omega}) \sim \vec{A}(\mathbf{\Omega}) \cdot \left[\Sigma_{s,i} \vec{\phi}_i + \vec{S}_i \right], \mathbf{r} \in \text{cell}\, i, \tag{1}$$

where $\vec{A}(\mathbf{\Omega}) = \left\{ A^{\rho}(\mathbf{\Omega}) \right\}$ is a vector of real spherical harmonics, $\Sigma_{s,i} = diag\left\{ \Sigma_{s,i}^{\rho} \right\}$ contains for each region the scattering coefficients associated to each spherical harmonic and $\vec{S}_i$ denotes the external source. The components of vector $\vec{\phi}_i$ are the angular flux moments

$$\vec{\phi}_i = \frac{1}{4\pi} \int_{(4\pi)} d\mathbf{\Omega}\, \vec{A}(\mathbf{\Omega}) \psi_i(\mathbf{\Omega}), \tag{2}$$

where $\psi_i(\mathbf{\Omega})$ is the region averaged angular flux. All angular integrals are calculated with the discrete-ordinates quadrature formula.

The second approximation is to use an expansion for the angular flux on region boundaries. For unstructured meshes one uses a set of parallel trajectories for each discrete ordinate direction $\mathbf{\Omega}$. The angular flux in direction $\mathbf{\Omega}$ is then assumed to be constant over the cross sectional area associated with each trajectory,

$$\psi(\mathbf{r},\mathbf{\Omega}) \sim \sum_{t\|\Omega} \theta_t(\mathbf{r}_\perp)\psi_t(z,\mathbf{\Omega}), \quad \mathbf{r} = \mathbf{r}_\perp + z\mathbf{\Omega}, \tag{3}$$

and it is computed analytically along the trajectory. In this formula z and $\mathbf{r}_\perp$ are, respectively, the coordinate in direction Ω and the coordinates over a plane orthogonal to Ω and $\theta_t(\mathbf{r}_\perp)$ is the characteristic function associated to the cross sectional area of trajectory *t*.

The MOC equations comprise a balance equation and a transmission equation. Only the latter concerns us here. The transmission equation, which results from the analytical integration along each trajectory, gives the angular flux exiting the trajectory in terms of the incoming value and of the internal source. The homogeneity of the region as well as the flat source approximation are used to obtain:

$$\psi_{+,i}^{(n)}(t) = e^{-\Sigma_i R_i(t)}\psi_{-,i}^{(n)}(t) + \underbrace{\frac{1-e^{-\Sigma_i R_i(t)}}{\Sigma_i}}_{E(\Sigma_i R_i)} q_i^{(n-1)}(\mathbf{\Omega}), \tag{4}$$

where *n* is the iteration index, $R_i(t)$ is the length of trajectory *t* within region *i* and $\mathbf{\Omega}$ its direction.

A transport iteration consists of a sweep over all the trajectories crossing the domain. This provides the fluxes leaving the domain and, via use of the region balance equations, the updated values of the region averaged angular fluxes. The latter are then used in equation (2) to update the region averaged flux moments that are compared to the previous ones to check convergence. If the iteration has not converged, the sources are updated via Eq. (1) and the transport iteration is repeated. Details of the full MOC equations can be found in [2,12].

3 THE INTEGRAL DP_N SYNTHETIC OPERATOR

A technique similar to the MOC is used to derive the DP_N acceleration equations but with approximations (1) and (3) replaced by low-order ones. The source region expansion is as in (1) but with a

smaller number of angular moments. The second approximation is more severe. The region boundary is decomposed into surfaces, a surface α being typically the common interface with a neighboring region, and a spatially constant angular approximation is assumed for the boundary fluxes:

$$\psi_{\pm}(\mathbf{r}_s,\mathbf{\Omega}) \sim \vec{A}(\mathbf{\Omega})\vec{\psi}_{\alpha^{\pm}} \ , \mathbf{r}_s \in S_\alpha , \tag{5}$$

where S_α stands for a generic surface and $\vec{\psi}_{\alpha^{\pm}}$ indicates exiting (+) or entering (-) surface averaged angular moments. The vector $\vec{A}(\mathbf{\Omega})$ contains all the spherical harmonic of polar order $k \le N$. The index "*N*" indicates that a "Double P_N" spherical harmonics expansion is used in (5) (one per side), this gives its name to the acceleration. Together with equation (5) the following currents are defined:

$$J^{\rho}_{\alpha^{\pm}} = \int_{S_\alpha} dS \int_{(2\pi^{\pm})} d\mathbf{\Omega} \left|\mathbf{\Omega}\bullet n\right| A^{\rho}(\mathbf{\Omega})\psi_{\pm}(\mathbf{r}_s,\mathbf{\Omega}) = \sum_{\upsilon} A^{\rho\upsilon}_{\alpha^{\pm}}\psi^{\upsilon}_{\alpha^{\pm}}, \tag{6}$$

where coefficient *A* are defined below in (9). We note that for the lowest moment this quantity is exactly the neutron current.

The first equation type of the DP_N synthetic method expresses the current exiting a surface α in function of the interface fluxes of neighboring regions. This is done by using the same set of tracking trajectories used for the MOC. In the standard DP_N, method approximation (5) is applied to the surfaces of all regions for all entering fluxes. This allows writing a transmission equation that couples all $\vec{\psi}_{\pm,\alpha}$ quantities. In this work we consider a partition of the system into macro-domains composed of homogeneous regions and we apply (5) to only the surfaces between macro-domains. We use the integral transport equation over a given trajectory "t". The angular flux exiting a region contained in macro-domain I is:

$$\psi^{+}_{i} = e^{-\tau(i,\partial)}\psi^{ex}_{-} + \sum_{j\in I, j<i}\left(1-e^{-\tau_j}\right)\frac{q_j}{\Sigma_j}e^{-\tau_{ij}}, \tag{7}$$

where $\tau(i,\partial)$ is the optical length between the boundary of the macro-domain *I* and the entering boundary of region i along trajectory t, τ_j the optical length over a region j, Σ_j the total cross section in region j and τ_{ij} the optical distance between the entering point in region i and the exiting point of region j. Here we consider a point as "exiting" or "entering" with respect to the given trajectory. Figure 1 gives a sketch of the situation. Using the projection defined in (6) over an arbitrary surface, and applying the DP_N approximation (1) to the macro-domain interface entering fluxes, gives:

$$J^{\rho}_{\alpha^+} = \sum_{\upsilon,\beta\in\partial I} T^{\rho\upsilon}_{\alpha^+\beta^-}\psi^{\upsilon}_{\beta^-} + \sum_{\upsilon,i\in I} E^{\rho\upsilon}_{\alpha^+,i} q^{\upsilon}_i = T\vec{\psi}^{-} + E\,\vec{q}, \tag{8}$$

where the source q^{υ}_i has been defined implicitly in (1) and E and T are escape and transmission coefficients respectively. The sum over β is done on region surfaces that belong to the surface ∂I of the macro-domain I, and the sign "+/-" indicates that one has to consider the entering/exiting current. Equation (8) gives us an expression for all currents appearing in the system, regardless of their position with respect to macro-interfaces. The only approximation in (8) is to assume a reduced DP_N flux anisotropy in the entering flux. In order to allow coupling between macro-domains, another DP_N expansion is assumed for the currents exiting a macro-domain interface, that is to say:

$$J^{\rho}_{\alpha^+} = \sum_{\upsilon} A^{\rho\upsilon}_{\alpha^+}\psi^{\upsilon}_{\alpha,+}\ . \tag{8a}$$

The definitions and properties of the coefficients A,E and T are quite similar to those of the standard DP_N method but, in order to stress some important differences, we recall them here. The matrix coefficients A_α appearing in (6) and (8a), the transmission and escape coefficients in equation (8), are:

$$\begin{aligned}
A^{\rho\upsilon}_{\alpha^+} &= \frac{1}{4\pi}\int_{\alpha} dS \int_{(2\pi^+)} d\boldsymbol{\Omega}\,|\boldsymbol{\Omega}\cdot n|\,A^{\rho}(\boldsymbol{\Omega})A^{\upsilon}(\boldsymbol{\Omega}),\\
T^{\rho\upsilon}_{\alpha^+\beta^-} &= \frac{1}{4\pi}\int_{\alpha} dS \int_{(\beta^-\to\alpha^+)} d\boldsymbol{\Omega}\,|\boldsymbol{\Omega}\cdot n|\,A^{\rho}(\boldsymbol{\Omega})A^{\upsilon}(\boldsymbol{\Omega})\,e^{-\tau_{\alpha\beta}},\\
E^{\rho\upsilon}_{\alpha^+,i} &= \frac{1}{4\pi}\int_{\alpha} dS \int_{(i\to\alpha^+)} d\boldsymbol{\Omega}\,|\boldsymbol{\Omega}\cdot n|\,A^{\rho}(\boldsymbol{\Omega})A^{\upsilon}(\boldsymbol{\Omega})\,e^{-\tau_{\alpha,(i)}}E(\tau_i),
\end{aligned} \tag{9}$$

where $(\beta\to\alpha)$ (respectively $(i\to\alpha)$) indicates that the integration domain has to include only the trajectories going from surface β to surface α (respectively from region i to surface α), $\tau_{\alpha\beta}$ is the optical distance between α and β, $\tau_{\alpha,(i)}$ is the optical distance along direction $\boldsymbol{\Omega}$ between the exiting side from region "i" and the surface α, and, finally, $E(R_i)$ is defined in (4) has been used. It is important to note that for consistency all these coefficients are computed using the MOC trajectories for the integral over surface α. Moreover it is easy to see that the coefficients T and E obey symmetries/anti-symmetries and complementary relations. By "complementary" relation we mean a relation that links the two set of coefficients (E and T). The symmetry/anti-symmetry relation for the T coefficients are:

$$T^{\rho\upsilon}_{\alpha^+\beta^-} = T^{\upsilon\rho}_{\alpha^+\beta^-} = s_{\upsilon}s_{\rho}T^{\rho\upsilon}_{\beta^+\alpha^-}, \tag{10}$$

where s_ρ stands for the parity of the harmonic ρ. It is important to stress how (10) can be used to greatly

reduce the computational effort necessary to obtain the transmission coefficients. Actually, instead of computing the coefficients *T* by using (9) and sweeping over all angles, we strongly reduce the integration domain thanks to (10), and compute, sweeping the track, the following integrals:

$$\hat{T}^{\rho\upsilon}_{\alpha^+\beta^-} = \frac{1}{4\pi}\int_{\alpha} dS \int_{\substack{(\beta^-\to\alpha^+)\\ \rho\le\upsilon \\ \mathbf{\Omega}\in[0,\pi]\times[0,\pi/2]}} d\mathbf{\Omega}\left|\mathbf{\Omega}\cdot n\right| A^{\rho}\left(\mathbf{\Omega}\right) A^{\upsilon}\left(\mathbf{\Omega}\right) e^{-\tau_{\alpha\beta}}, \tag{11}$$

where the conditions listed below the integral sign mean that we are considering only the lowest diagonal angular harmonics and the angular directions belonging to the first two octants. Thanks to (10) it is easy to see that:

$$T^{\rho\upsilon}_{\alpha^+\beta^-} = \hat{T}^{\rho\upsilon}_{\alpha^+\beta^-} + s_{\upsilon}s_{\rho}\hat{T}^{\rho\upsilon}_{\beta^+\alpha^-}, \tag{11a}$$

for ρ<υ, while for ρ>υ the transmission terms are obtained directly from (10). Calculations made with (11) imply the use of less than half of the trajectories that would be needed were we to use (9). A set of complementary relations are also valid for the escape coefficients:

$$\mathbf{\Sigma}_i E^{\rho\upsilon}_{\alpha^+,i} = \sum_{\beta\in i}\left[\left(T^{\rho\upsilon}_{\alpha^+\beta^+} - T^{\rho\upsilon}_{\alpha^+\beta^-}\right)\delta_{\beta\notin\partial I} - \delta_{\beta\in\partial I}T^{\rho\upsilon}_{\alpha^+\beta^-} + \delta_{\alpha,\beta}A^{\rho\upsilon}_{\alpha^+}\right]. \tag{12}$$

Equation (12) expresses the escape coefficients from a region "*i*" through a surface α as a sum over all the surfaces of the region. Three terms inside the sum must be distinguished. The first one, in which appears the symbol $\delta_{\beta\notin\partial I}$, comes from the surfaces that are completely internal to the macro-region *I*. The second one has to be included when the boundary surface β belongs to the border of the macro-region *I*. The last one has to be considered when β coincides with α. Expression (12) gives the escape coefficients only in terms of the transmission ones. Therefore, the transmission coefficients are the only quantities directly computed from tracking.

Equation (8) must be completed to include the boundary conditions, that is to say, by expressing the entering boundary currents in terms of the exiting ones. A detailed description of these equations can be found in [11]. In this paper we introduce an important improvement for the boundary currents equations that is detailed in appendix A. This treatment is common to both the standard DP_N and IDP_N methods.

Since the source *q* contains flux moments, supplementary equations are needed to close the problem. These are obtained from the DP_N region balance relations [9]:

$$\sum_{\alpha,\upsilon} \frac{\left(J^{\upsilon}_{\alpha^+} - J^{\upsilon}_{\alpha^-}\right)}{\Sigma_i} + V_i \phi_i^{\rho} = \frac{V_i}{\Sigma_i} \sum_{\upsilon} B^{\rho\upsilon} q_i^{\upsilon}. \tag{13}$$

Here B is an operator that takes into account the lack of numerical orthogonality for the spherical harmonic functions and is defined as:

$$B^{\rho\upsilon} = \frac{1}{4\pi} \int_{(4\pi)} d\Omega \, A^{\rho}(\Omega) A^{\upsilon}(\Omega). \tag{13a}$$

Equation (13) becomes ill-conditioned for voided or weakly absorbent regions and is therefore cast into a more stable form. The algebraic manipulations are parallel to the ones used in the standard DP_N method [9], but many substantial differences appear due to the full coupling of the fluxes for regions belonging to the same macro-region. Another benefit of the procedure is to reduce the size of the problem to be solved iteratively. In fact, the iteration problem then deals only with the surface currents.

Substituting the expression (8) of the currents into formula (13), one obtains the following system per macro-region I :

$$M_I \vec{\phi}_I = I_I \vec{\psi}_I^{-} + G_I \vec{q}_{ext}, \tag{14}$$

where the expressions for matrices I, G, M are given in appendix B. It is important to note that the coefficients of the "incoming" matrix I can be directly obtained from the escape coefficients E given by (12), as detailed in equation (B2). This relation allows further computational savings. The matrix G of (14) is defined in (B3a) as $G = \frac{BV}{\Sigma} - E,$ where the operator E can be defined as a function of the transmission factors as showed in (B5), and B is the matrix (13a). An analytical definition of the operator G is given in appendix B (formulas (B7) and (B7a)), and there we show also that this operator is well defined also for nearly voided media. Moreover we also prove that:

$$G_{ij}^{\rho\upsilon} = G_{ij}^{\upsilon\rho} = s^{\rho} G_{ji}^{\rho\upsilon} s^{\upsilon}. \tag{15}$$

Equation (15) enables us to reduce the computational effort by adopting a strategy similar to the one sketched in (11) and (11a) for the transmission factors.

The numerical counterpart of (B7) and (B7a) in the framework of the MOC is straightforward, and it is the way we have chosen to compute the operator E. The other possibility to compute E_I, the escape matrix for a given macro-region I, is described in appendix B, and makes use of reciprocity relationships between E_I

and the transmission factors between all surfaces of *I*. We have opted for the first way since the storage required by E_I is by far lower than the one required by the transmission factors. System (14) is then used to express the flux moments as:

$$\vec{\phi} = \hat{I}\vec{\psi}^{-} + \hat{G}\vec{q}_{ext}, \tag{14a}$$

where $\hat{I} = M^{-1}I$ and $\hat{G} = M^{-1}G$. These expressions are inserted back into the transmission equation (8) through the source term (1). This gives:

$$\vec{\psi}^{+} = A^{-1}\left(T + E\Sigma_s\hat{I}\right)\vec{\psi}^{-} + A^{-1}E\left(\Sigma_s\hat{G} + I_d\right)\vec{q}_{ext}, \tag{16}$$

where $\Sigma_{s,i} = diag\left\{\Sigma_{s,i}^{\rho}\right\}$ and I_d is the identity matrix. It is worth noting that the matrix $T+E\Sigma_s\hat{I}$, that appears in (16), is sign-symmetric (as shown in appendix B), a property that is exploited to reduce the computational effort. Equation (16) is the final iterative form used to solve the problem and, since it is written for currents between macro-regions, it has a reduced dimension with respect to the standard DP_N method. This means that problem (16) should be faster to solve than its equivalent in the standard DP_N method. Once the currents have been obtained by solving (16), volume fluxes are obtained using (14a).

All transmission and escape coefficients can be shown to be bounded. For the transmission coefficients, this fact is clear from their definition (9). On the contrary, to show boundedness for the escape coefficients, it is necessary to develop and implement numerically an asymptotic expansion of equation (12). In fact, it is clear that expression (12) is greatly ill-conditioned for nearly voided systems. This development is given in appendix B.

Problem (16) has the same response-matrix form as the standard DP_N operator. In [9] it has been shown that a numerous set of preconditioning and optimization techniques can be used for this problem. This is an important point for our acceleration technique since all previous work, done in the framework of the standard DP_N operator, can be re-used for the IDP_N acceleration.

All the geometry partitions that we have used in this article have been obtained by the Spectral Bisection Algorithm (SBA) [13], that is a standard way to compute domain decompositions.

4 EXTERNAL SYNTHETIC ACCELERATIONS

In this work we have also applied the new IDP_N acceleration as a multigroup acceleration technique to improve the rate of convergence in multigroup calculations. Until now, most of the work that has been done to accelerate the external iterations used some non-linear strategy, [6] some classical linear extrapolations (like the Chebyshev one) or some Krylov subspace techniques. [14] Recently, Suslov [15] proposed a linear algorithm using a mono-group synthetic operator in the external framework. We implemented this idea in APOLLO2 with slight modifications, and used the standard DP_N and the new IDP_N operator to test its efficiency. In this section we describe the algorithm and discuss implementation details.

The multigroup external equations are driven by fission and, written with the iteration index, read :

$$\left(L - H^d\right)\psi^{(p,j)} = H^u\psi^{(p,j-1)} + \frac{1}{\lambda^{(p-1)}}F\psi^{(p-1)}, \tag{17}$$

where L is the one-group transport operator, H^d is the group-to-group and slowing down scattering operator, H^u is the up-scattering operator and F is the fission operator. The upper index p denotes the external iteration index, while j is the thermal iteration index. Denoting by J_{tot} the number of thermal iterations per external sweep, it is assumed in (17) that $\psi^{(p)} = \psi^{(p,J_{tot})}$ for every external iteration p. One can then write the converged solution as

$$\psi^{(\infty)} = \psi^{(p,j)} + f^{(p,j)}, \tag{18}$$

where

$$\left(L - H\right)\psi^{(\infty)} = \frac{F}{\lambda}\psi^{(\infty)}. \tag{19}$$

We substitute formula (18) in (19), taking into account (17), to obtain:

$$\left(L - H^s\right) f^{(p,j)} = H^u f^{(p,j)} + \frac{F}{\lambda} f^{(p,j)} + H^u\Delta\psi + F\left(\frac{\psi^{(p,j)}}{\lambda} - \frac{\psi^{(p-1)}}{\lambda^{(p-1)}}\right), \tag{20}$$

where $\Delta\psi = \psi^{(p,j)} - \psi^{(p,j-1)}$ is the difference between two consecutive transport iterates. Moreover, λ is the new eigenvalue obtained as the result of the multigroup correction problem (20). Problem (20) is an eigenvalue problem but with the presence of an external inhomogeneous term. An accelerated external iteration consists of a free MOC external iteration followed by one synthetic multigroup solution correction

given by (20). This last problem is solved using once again the power iteration solution strategy, that is to say by fixing the terms coming from fission and compute a multigroup solution with the help of "*j*" thermal iterations. The new estimated eigenvalue for the next iteration is given by the usual condition:

$$\lambda^{(k)} == \frac{\left\| F\left(\psi^{(p,j)} + f^{(p,j),k}\right)\right\|}{\left\| F\left(\psi^{(p,j)} + f^{(p,j),k-1}\right)\right\|},$$

where the index *k* refers to the external iterations necessary to solve the problem (20).

5 RESULTS AND COMPARISONS

In this section we will prove the effectiveness of the IDP_N acceleration, presenting some results related to the Atrium Benchmark. [16] Here we will compute a version of this benchmark that makes use of a sixteen group P_0 transport corrected cross section library that has been obtained by a previous self-shielding phase in APOLLO2. [17] The Atrium benchmark is a severe realistic assembly calculation, that has strong transport effects and slow convergence rate. The spatial computational mesh used for our calculation is shown in figure 2, and it consists of 3560 regions. A full description of the case can be found in [16], and we here only insist on some basic features. The Atrium benchmark represents a BWR assembly. The fuel and gadolinium rods are surrounded by an external as well as an internal box, where homogeneous water flows. As it is typical for these reactors, liquid water concentration varies inside the assembly. In fact, for the considered version of the benchmark, we have a 0% vapor fraction inside both the internal and the external box of the assembly, while we have a 40% vapor fraction for the water in direct contact with fuel rods. Fuel rods are characterized by three different enrichments, and strong flux heterogeneities appear next to gadolinium rods, that are distinguishable in figure 2 by the very fine mesh refinement necessary to represent flux gradients. We used the transport code APOLLO2 where all the developments we present here are now implemented. The tracking data used to compute the benchmark are shown in table 1. We used a product angular formula with sixteen uniform azimutal angles, and two optimal Bickley polar angles. [18] The transversal spatial step is 0.025 cm. This results in a total of 643,617 horizontal intersections with computational regions.

A first comparison deals with two calculations that have used a DP_1 or IDP_1 multigroup initialization and a thermal synthetic acceleration, a procedure that is also discussed in [9]. Table 2 gives the performances

of the old DP_1 operator and of the new IDP_1 one. Free iterations are also shown in order to illustrate the efficiency of the acceleration methods. All the results in the table, with the exception of the free iterations ones, were obtained using a mesh grid that was constructed using the recursive SBA. [13] For the standard DP_1 method, the chosen grid is used to make a two-level multigrid acceleration method [9], while for the IDP_1 operator we use this same grid to compute macro-domain integral probabilities. We have always used the adapted ILU0 preconditioning technique [9] to speed up the DP_N and IDP_N equations. Due to their great impact over the total computational time the building times are also shown. All computing times are expressed in seconds on a 2.8Ghz 512Kb cache PC.

Building times strongly depend on the preliminary manipulations required: algebraic elimination process for the standard DP_1 method, and computation of the transmission probabilities for the IDP_1 method. As the number of regions per macro-region increases, the building time also increases, but the burden of the accelerated iterative resolution strongly decreases. Therefore the appropriate number of regions to be included in each macro-region should be optimized. This phenomenon was already described in [9] and we will not insist on it here. Our numerical tests show that a good number of regions per macro one for the Atrium benchmark is 70, so we always retained the grid mesh resulting from the recursive SBA with this fixed maximum number of regions. It is important to stress that, while for the standard DP_N method the grid mesh has no physical impact, for the IDP_N it should be beneficial to take advantage of a "physical" grid, that is a grid with reduced flux anisotropy on grid interfaces. This is the case for example for the classical IC method where the grid is given by the region decomposition of the assembly. In fact one can suppose that flux anisotropy is reduced at region interfaces. In this work we will always use the "numerical" grid given by the SBA, even if it certainly does not guarantee the best result.

As one can see from table 2, the best time with consistent tracking, 26.4 sec., is obtained for the IDP_1 operator. The standard DP_1 operator spends a lot of time in the initial building phase, and requires 151 internal transport iterations. In spite of the non-"physical" SBA grid mesh, the spectral behavior of the IDP_1 operator is quite better since it converges with only 102 iterations. The thermal synthetic acceleration, used with the IDP_1 method, reduces the number of total transport iterations by almost a factor of 40 with respect to free iterations. Unfortunately, the IDP_1 acceleration (as well as the DP_1 one) requires an expensive building phase that accounts for half of the total time; this explains why the accelerated calculation is only 15 times faster than the free iterations.

In order to reduce the expensive building phase, we introduced a low-order tracking described at the third row of table 1. The low-order tracking is now used to compute a synthetic IDP_1 operator: this choice entails a lower building cost but can cause a poorer convergence rate. If one compares the data for the reference and low-order tracking, one can see that the number of tracks that are used to build the IDP_1 operator is now decreased by a factor of four with respect to the reference case. We will call this approach "inconsistent" since it is generally required that the acceleration operator be obtained in a "consistent" way with respect to transport, in order to avoid problems of poor convergence or even of divergence of the accelerated algorithm. Using an inconsistent acceleration can therefore lead to lack of robustness. At this time we do not suggest a method to prevent these drawbacks. In the results we show, and more generally on the basis of our experience, we can state that the behavior of the acceleration is not degraded too much if one maintains the low-order tracking formula at an acceptable level.

The new IDP_1 results are given in fourth row of table 2. The use of a low-order tracking should entail poorer IDP_1 performances. The results in table 2 show that this effect is very limited, since the total number of iterations goes from 102 to 107. On the contrary, thanks to the faster building time, the total computational time decreases from 26.4 to 23.6 seconds. Note that while most of the IDP_1 building time is directly proportional to tracking, the standard DP_1 building time strongly depends on the geometrical structure of the problem, and weakly on tracking. The use of a low-order tracking has practically no effect on the standard DP_1 operator, and therefore we do not present inconsistent DP_1 results in table 2.

A last batch of results (table 3) deals with the impact of the external synthetic acceleration. We used this algorithm in conjunction with the standard DP_N operator or the new IDP_N operator. Both have been computed inconsistently with respect to transport: we used the low-order tracking described by the third row of table 1. In table 3, one can see that the external acceleration is very effective for the synthetic IDP_1 operator, since it reduces the total number of internal transport iterations from 106 to 48. Unfortunately this yields only a slight time reduction, the total computational time goes from 23 to 22.3 seconds. This is due to the fact that the synthetic external acceleration "transfers" a great amount of work from the transport to the synthetic operator. And in fact the total number of synthetic operator problems to be solved is multiplied by more than a factor two, with respect to the cases presented in table 2.

The only possibility to further reduce the computational time is to solve the IDP_N problem faster, without increasing building times. We suppose that this could be possible by adopting a full multilevel algorithm (and not a purely two-level one as in this paper). We will address this issue in future works.

6 CONCLUSIONS

In this paper a new DP_N synthetic operator has been introduced. Its main approximation consists in supposing reduced flux anisotropy over a limited number of surfaces. In fact, after having subdivided the system into some macro-regions, the IDP_N operator supposes limited flux anisotropy only over those surfaces that divide macro-regions. Thanks to this preliminary hypothesis, and by making use of the basic MOC equations, one can find an integral operator whose unknowns are the flux components over macro-interfaces. Many symmetries/anti-symmetries and complementary relations help to minimize the computational effort, and taking into account important physical properties in the numerical operator. Calculations made with APOLLO2 on the Atrium Benchmark show that the IDP_1 method is more efficient than its standard counterpart, and that it can be retained as an efficient acceleration tool. The preliminary time necessary to build the operator is reduced and the spectral performance is improved. The behavior of the new IDP_N operator in conjunction with the external synthetic acceleration approach has also been evaluated. Even if the overall computational time is reduced, further improvements of the IDP_N operator are necessary to make this reduction significant. In particular, we suggest adopting a true multilevel algorithm that will help to reduce building and solution times. We will address this issue in future works.

APPENDIX A: BOUNDARY CONDITIONS TREATMENT

In this section we give a detailed description of the boundary current treatment. The complete form of the boundary current equations has been given elsewhere. [8] What we describe here is an improvement in the treatment of these equations for the DP_N problem. We shall show that for all local boundary conditions (i.e. all boundary conditions where the neutrons exiting the system re-enter through the same surface), exiting currents can be eliminated from the iterative part of the problem with almost no computational and memory cost. Let us write the transmission problem (8) in the following manner:

$$\begin{cases} \psi^+ = T\psi^- + T_\partial \psi_\partial^- + E_i S_i \\ \psi_\partial^- = \beta \psi_\partial^+ \end{cases}, \qquad \text{(A1)}$$

where we have distinguished the transmission contributions coming from internal currents "ψ^-" and from external currents "ψ_∂^-". Each entering external current can be written as

$$\vec{\psi}_\alpha^- = \beta \vec{\psi}_{t(\alpha)}^+ . \tag{A2}$$

For local boundary conditions $t(\alpha)=\alpha$, but in general surfaces do not coincide. For both cases equation (A1) can be re-written as

$$\psi^+ = T\psi^- + \tilde{T}_\partial \psi_\partial^+ + E_i\, S_i \tag{A3}$$

where $\tilde{T}_\partial = T_\partial \beta$. The system of equations (A3) is a reduced form of the initial system (A1): it contains less unknowns and equations but shares the same profile for the remaining equations. This means that contrarily to standard elimination processes, where fill-in always appears, here no elements are added to the initial operator. In the case of no local boundary conditions, the further elimination of the boundary exiting currents appearing in (A3) would cause fill-in and, for this reason, we decided not to eliminate them. For local boundary conditions the situation is different. In fact one can extract the boundary external currents from (A3), and write them as

$$\vec{\psi}_{\partial,+}^{(i)} = T^{(i)}\vec{\psi}_- + \sum_{j\neq i} \tilde{T}_\partial^{(i,j)} \vec{\psi}_{\partial,+}^{(j)} + E^{(i)}\vec{q}\,, \tag{A4}$$

where $\vec{\psi}_{\partial,+}^{(i)}$ is the outgoing external modes related to the i-th external surface, $T^{(i)}$ is the transmission matrix giving the contribution of internal currents to the i-th external boundary current, $\tilde{T}_\partial^{(i,j)}$ is the transmission matrix describing the link between the j-th external surface and the i-th one and $E^{(i)}\vec{q}$ describes the escape. It is clear that the only case when $\tilde{T}_\partial^{(i,j)}$ is different from zero is when the i-th and the j-th surface share the same region. For convenience we will suppress from now on the tilde notation over the T_∂ symbol since each external transmission term will always be post-multiplied by the albedo operator. The objective is to show that building the formal inversion $\left(I - \sum_{i,j} \tilde{T}_\partial^{(i,j)} \right)^{-1}$ is not computationally expensive and can be done without any fill-in in the LU decomposition necessary to compute this inverse. To see this, let us eliminate the first external current from the system (A4), to obtain

$$\underbrace{\left(I - T_{\partial}^{(i,1)} T_{\partial}^{(1,i)}\right)}_{D_i^1} \vec{\psi}_{\partial,+}^{(i)} = \left[T^{(i)} + T_{\partial}^{(i,1)} T^{(1)}\right] \vec{\psi}_{-} + \sum_{\substack{j \neq i \\ j>1}} \left[T_{\partial}^{(i,1)} T_{\partial}^{(1,j)} + T_{\partial}^{(i,j)}\right] \vec{\psi}_{\partial,+}^{(j)} + \left[E^{(i)} + T_{\partial}^{(i,1)} E^{(1)}\right] \vec{q}\,, \tag{A5}$$

which is valid for all i>1. Rearranging these equations to a unit diagonal gives:

$$\vec{\psi}_{\partial,+}^{(i)} = T_1^{(i)} \vec{\psi}_{-} + \sum_{\substack{j \neq i \\ j>1}} T_{\partial,1}^{(i,j)} \vec{\psi}_{\partial,+}^{(j)} + E_1^{(i)} \vec{q}, \quad \forall i > 1\,, \tag{A6}$$

where

$$T_1^{(i)} = \left(D_i^1\right)^{-1} \left(T^{(i)} + T_{\partial}^{(i,1)} T^{(1)}\right),\ T_{\partial,1}^{(i,j)} = \left(D_i^1\right)^{-1} \left(T_{\partial}^{(i,1)} T_{\partial}^{(1,j)} + T_{\partial}^{(i,j)}\right) \text{ and } \ E_1^{(i)} = \left(D_i^1\right)^{-1} \left(E^{(i)} + T_{\partial}^{(i,1)} E^{(1)}\right).$$

System (A6) is formally similar to system (A4), with the exception that the number of unknowns is reduced by one in all equations but the first, and that the profile of the operator has been correspondingly reduced since each coupling term due to the first external current has been eliminated from all equations. A similar reasoning can be applied to all the other currents, so that at the final step one obtains an upper diagonal operator. Before the l-th step one has to eliminate the l-th current from the equations greater than l. After the elimination of this current from the i-th equation of the system (where clearly i>l) this equation reads

$$\vec{\psi}_{\partial,+}^{(i)} = T_l^{(i)} \vec{\psi}_{-} + \sum_{\substack{j \neq i \\ j>l}} T_{\partial,l}^{(i,j)} \vec{\psi}_{\partial,+}^{(j)} + E_l^{(i)} \vec{q} \tag{A7}$$

where, after defining $D_i^l = I - T_{\partial}^{(i,l)} T_{\partial}^{(l,i)}$, one has

$$T_l^{(i)} = \left(D_i^l\right)^{-1} \left(T_{l-1}^{(i)} + T_{\partial,l-1}^{(i,l)} T_{l-1}^{(l)}\right), \tag{A8a}$$

$$T_{\partial,l}^{(i,j)} = \left(D_i^l\right)^{-1} T_{\partial,l-1}^{(i,j)}\,. \tag{A8b}$$

Equations (A8a-b) allow the calculation of all matrices $T_l^{(i)}$ from all matrices at the preceding step ($T_{l-1}^{(i)}$). As before, the elimination process gives rise only to the cancellation of boundary terms, without any fill-in. The remaining step is to complete the inversion of the original $I - \sum_{i,j} \tilde{T}_{\partial}^{(i,j)}$ matrix by substituting back, from the last until the first, the new computed equations. Let us call N_{bd} the total number of boundary equations in (A4). After the N_{bd} steps described above, one has an upper diagonal operator where the last equation gives

the N_{bd} -th external current in function only of internal unknowns and collision sources. The back-substitution of this equation into the (N_{bd} -1)-th one allows again to express the (N_{bd} -1)-th external current in function only of internal currents and collision sources.

This back-substitution process can be easily repeated, until each external current is expressed in function only of internal currents and collision sources. In fact when treating the j-th equation in the backward procedure, one can use the following situation:

$$\begin{aligned} &\vec{\psi}_{\partial,+}^{(j)} = T_j^{(j)}\vec{\psi}_- + \sum_{k>j} T_{\partial,i}^{(j,k)}\vec{\psi}_{\partial,+}^{(k)} + E_j^{(j)}\vec{q} \\ &for\ i > j \quad \vec{\psi}_{\partial,+}^{(i)} = T_{i,back}^{(i)}\vec{\psi}_- + E_{i,back}^{(i)}\vec{q} \end{aligned} \tag{A9}$$

where one can note that all equations beyond the j-th one have already been diagonalised. The back-substitution of these equations into the j-th one gives:

$$\vec{\psi}_{\partial,+}^{(j)} = \underbrace{\left[T_j^{(j)} + \sum_{i>j} T_{\partial,i}^{(j,i)} T_{i,back}^{(i)} \right]}_{T_{j,back}^{(j)}} \vec{\psi}_- + \underbrace{\left[E_j^{(j)} + \sum_{i>j} T_{\partial,i}^{(j,i)} E_{i,back}^{(i)} \right]}_{E_{j,back}^{(j)}} V\vec{q}\,,$$

which gives the recursive definition of the back matrices. Finally the system (A4) is reduced to a system of the form:

$$\vec{\psi}_{\partial,+}^{(i)} = T_{back}^{(i)}\vec{\psi}_- + E_{back}^{(i)}\vec{q}\,, \tag{A10}$$

where all external elements have disappeared. Eqs (A10) are substituted into (A4) to completely eliminate the boundary currents from the problem and finally obtain:

$$\vec{\psi}^+ = (T + \sum_{\partial} T_{\partial}^{+} T_{back}^{\partial,-})\vec{\psi}_- + (\sum_{\partial} T_{\partial}^{+} E_{back}^{\partial} + E^{(i)})\vec{q}\,,$$

where the sums over ∂ is done on boundary currents. The balance equations have also to be consistently modified. In fact the original balance equations (15a), written without the hat sign, can be split in the following manner:

$$\vec{\phi} = H\vec{\psi}^- + GV\vec{q} = H_\partial \beta \vec{\psi}_\partial^+ + H\vec{\psi}^- + GV\vec{q}\,, \tag{A11}$$

where H_∂ stands for the boundary part of the balance operator *H*. By making use of (A10) one obtains:

$$\vec{\phi} = \left(H_\partial \beta \hat{T} + H \right)\vec{\psi}^- + \left(G + H_\partial \beta \hat{E} \right) V\vec{q}\,,$$

where $\hat{T}$ and $\hat{E}$ are the transmission and escape operators appearing in (A10).

APPENDIX B: MATRIX COEFFICIENTS OF THE INTEGRAL DP$_N$ METHOD AND ASYMPTOTIC LIMITS OF ESCAPE PROBABILITIES

In this appendix we give some numerical details about the DP_N equations. First of all we detail the expression of matrices *M*, *G* and *H* appearing in (14). Next we consider the vacuum limit for the escape coefficients in equation (9).

Substituting (8) into (13) yields:

$$J^{\rho}_{\alpha^+} - J^{\rho}_{\alpha^-} = \begin{cases} \sum_{\beta\in\partial I,\upsilon} \left(T^{\rho\upsilon}_{\alpha^+,\beta^-} - T^{\rho\upsilon}_{\alpha^-,\beta^-}\right)\psi^{\upsilon}_{\beta^-} + \sum_{j\in I,\upsilon}\left(E^{\rho\upsilon}_{\alpha^+,j} - E^{\rho\upsilon}_{\alpha^-,j}\right)q^{\upsilon}_j & \text{if } \alpha \notin \partial I, \\ \sum_{\beta\in\partial I,\upsilon} T^{\rho\upsilon}_{\alpha^+,\beta^-}\psi^{\upsilon}_{\beta^-} - \sum_{\upsilon} A^{\rho\upsilon}_{\alpha^-}\psi^{\upsilon}_{\alpha^-} + \sum_{j\in I,\upsilon} E^{\rho\upsilon}_{\alpha^+,j} q^{\upsilon}_j & \text{if } \alpha \in \partial I. \end{cases} \tag{B1}$$

From equation (B1) we derive the following expression for the leakage term of the balance equation:

$$\begin{aligned} \sum_{\alpha\in i}\frac{J^{\rho}_{+,\alpha} - J^{\rho}_{-,\alpha}}{\Sigma_i} &= \sum_{\beta\in\partial I,\upsilon} \underbrace{\left\{\frac{1}{\Sigma_i}\sum_{\alpha\in i}\left[\delta_{\alpha\notin\partial I}\left(T^{\rho\upsilon}_{\alpha^+,\beta} - T^{\rho\upsilon}_{\alpha^-,\beta}\right) + \delta_{\alpha\in\partial I}\left(T^{\rho\upsilon}_{\alpha^+,\beta} - \delta_{\alpha\beta}A^{\rho\upsilon}_{\alpha^-}\right)\right]\right\}}_{(I_I)^{\rho\upsilon}_{i,\beta^-}}\psi^{\upsilon}_{\beta^-} \\ &+ \sum_{j\in I,\upsilon} \underbrace{\left\{\frac{1}{\Sigma_i}\sum_{\alpha\in i,i\in I}\left[\delta_{\alpha\notin\partial I}\left(E^{\rho\upsilon}_{\alpha^+,j} - E^{\rho\upsilon}_{\alpha^-,j}\right) + \delta_{\alpha\in\partial I}E^{\rho\upsilon}_{\alpha^+,j}\right]\right\}}_{(E_I)^{\rho\upsilon}_{ij}} q^{\upsilon}_j . \end{aligned} \tag{B2}$$

Equation (B2) implicitly defines the matrix *I* of equation (15), and another auxiliary matrix E_I used to construct the matrices *G* and *M* of equation (14). The incoming matrix *I* can be better re-written in function of the escape coefficients (12):

$$\begin{aligned} I^{\rho\upsilon}_{i,\beta^-} &= -\frac{1}{\Sigma_i}\sum_{\alpha\in i}\left[\delta_{\alpha\notin\partial I}\left(T^{\rho\upsilon}_{\alpha^+,\beta^-} - T^{\rho\upsilon}_{\alpha^-,\beta^-}\right) + \delta_{\alpha\in\partial I}\left(T^{\rho\upsilon}_{\alpha^+,\beta^-} - \delta_{\alpha\beta}A^{\rho\upsilon}_{\alpha^-}\right)\right] = \\ &-s^{\rho}s^{\upsilon}\frac{1}{\Sigma_i}\sum_{\alpha\in i}\left[\delta_{\alpha\notin\partial I}\left(T^{\rho\upsilon}_{\beta^+,\alpha^-} - T^{\rho\upsilon}_{\beta^+,\alpha^+}\right) + \delta_{\alpha\in\partial I}\left(T^{\rho\upsilon}_{\beta^+,\alpha^-} - \delta_{\alpha\beta}A^{\rho\upsilon}_{\alpha^+}\right)\right] = -s^{\rho}s^{\upsilon}E^{\rho\upsilon}_{\beta^+,i}, \end{aligned} \tag{B3}$$

where (10) has been used together with the fact that $A^{\rho\upsilon}_{\alpha^+} = s^{\rho}s^{\upsilon}A^{\rho\upsilon}_{\alpha^-}$. (B3) represents a complementary property between the coefficients of the incoming matrix *I* and escape coefficients *E*. Re-inserting (B1) into the balance equation (13), one obtains the following expression for the matrix M_I of equation (15) :

$$\left(M_I\right)^{\rho\upsilon}_{ij} = V_i\delta_{ij}\delta_{\rho\upsilon} + [\left(E_I\right)^{\rho\upsilon}_{ij} - \frac{B^{\rho\upsilon}V_j}{\Sigma_i}\delta_{ij}]\Sigma^{\upsilon}_{s,j}, \tag{B4}$$

where $i,j \in I$. Finally, the matrix G that appears in (15) is given by:

$$G_I = \frac{B\vec{V}}{\Sigma_i} - E_I, \tag{B4a}$$

where E_I is defined in (B1) and B in (13). Equations (B3) and (10) allow us to show that the matrix $T+E\Sigma_s\hat{I}$ appearing in (16) is sign-symmetric that is to say that it satisfies the relation ${}^t\left(T + E\Sigma_s\hat{I}\right) = s\left(T + E\Sigma_s\hat{I}\right)s$. Indeed, first note that (B3) and (10) can be written in matrix form as ${}^tT = sTs$ and ${}^tE = -sIs$. Moreover let us suppose that $M_I\Sigma_I^{-1}$ (and then also $\Sigma_I M_I^{-1}$) is sign-symmetric. This implies that

$${}^t\left(T + E\Sigma_s M_I^{-1} I\right) = s\left(T + E\,\Sigma_s M_I^{-1} I\right)s, \tag{B5}$$

which is the expected result. We still need to demonstrate that $M_I\Sigma_I^{-1}$ is sign-symmetric. This can equivalently be shown for E_I (thanks to (B4)). Therefore, inserting the definition of $E_{\alpha i}^{\rho\upsilon}$ given by (12) inside the expression (B2) of E_I , one obtains:

$$\begin{aligned} E_{ij}^{\rho\upsilon}\Sigma_j = \sum_{\substack{\alpha\in i \\ \beta\in j}} [& \delta_{\alpha\notin\partial I}\delta_{\beta\notin\partial I}\left(T_{\alpha^+,\beta^+}^{\rho\upsilon} - T_{\alpha^+,\beta^-}^{\rho\upsilon} + T_{\alpha^-,\beta^-}^{\rho\upsilon} - T_{\alpha^-,\beta^+}^{\rho\upsilon}\right) + \\ & \delta_{\alpha\notin\partial I}\delta_{\beta\in\partial I}\left(T_{\alpha^-,\beta^-}^{\rho\upsilon} - T_{\alpha^+,\beta^-}^{\rho\upsilon}\right) + \delta_{\alpha\in\partial I}\delta_{\beta\notin\partial I}\left(T_{\alpha^+,\beta^+}^{\rho\upsilon} - T_{\alpha^+,\beta^-}^{\rho\upsilon}\right) \\ & \delta_{\alpha\notin\partial I}\delta_{\alpha\beta}\left(A_{\alpha^+}^{\rho\upsilon} - A_{\alpha^-}^{\rho\upsilon}\right) - \delta_{\alpha\in\partial I}\delta_{\beta\in\partial I}T_{\alpha^+,\beta^-}^{\rho\upsilon}\,]. \end{aligned} \tag{B6}$$

Interchanging i with j in the left hand side amounts to interchanging α with β in the right hand side. Using (10), one then sees that each term in (B6) is transformed either into itself multiplied by $s^\rho s^\upsilon$, as it is the case for the terms into the third line of (B6), either in other terms of the same row multiplied by $s^\rho s^\upsilon$, as it is the case for the terms in the first two lines. We will find this result in another way in formula (B8). Equations (B4) and (B8) can be used to reduce the computational effort as well as the memory storage, since they show that only halves of the matrices E_I and $T+E\Sigma_s\hat{I}$ have to be computed and stored.

Thanks to (B6) an analytical definition of operator G can be given:

$$G_{ij}^{\rho\upsilon} = -E_{ij}^{\rho\upsilon} = \frac{1}{4\pi}\int_{(j\to i)} d\Omega \int_{\partial V_i} dr_s \left|\Omega\cdot n\right| A^\rho(\Omega) A^\upsilon(\Omega) \sum_{\substack{l\in T(r_s,\Omega)\cap V_i \\ m\in T(r_s,\Omega)\cap V_j}} E(\tau_l) e^{-\tau_{lm}} E(\tau_m) \quad i\neq j. \tag{B7}$$

In equation (B7) the symbol $(j\to i)$ stands for the angular directions going from region j to region i, the

symbol $T(r,\Omega)$ stands for the set of all the chords intersected by the trajectories that pass through the point r with angle Ω, so that $T(r,\Omega)\cap V_i$ ($T(r,\Omega)\cap V_j$) stands for the subset of chords that belong to region i (or region j) and symbol E is defined in (4). Therefore the sum in (B7) takes into accounts the case of the intersections with non-convex regions. The optical chord lengths τ_l, τ_m and τ_{lm} are related respectively to the chord in region i, to the chord in region j and to the distance between the exiting point of segment l and the entering point of segment m. An analogous expression can be written for the diagonal terms:

$$G_{ii}^{\rho\upsilon} = \frac{B^{\rho\upsilon}V_i}{\Sigma_i} - E_{ii}^{\rho\upsilon} = \frac{1}{4\pi}\int_{\partial V_i} dr_s \int_{(2\pi^+)} d\Omega \left|\Omega\cdot n\right| A^{\rho}(\Omega) A^{\upsilon}(\Omega)\bullet$$
$$\{ \sum_{l\in T(r_s,\Omega)\cap V_i} [l - E(\tau_l)]/\Sigma_i + \sum_{\substack{l,m\in T(r_s,\Omega)\cap V_i \\ l\neq m}} E(\tau_l)e^{-\tau_{lm}}E(\tau_m) \quad \}, \tag{B7a}$$

where the second sum inside parentheses comes again from non-convex regions. Inspecting (B7) and (B7a), one can readily realize the symmetric role of indexes i and j as well as of the angular indexes. This assures us that

$$G_{ij}^{\rho\upsilon} = G_{ij}^{\upsilon\rho} = s^{\rho} G_{ji}^{\rho\upsilon} s^{\upsilon}. \tag{B8}$$

Next, we show that the escape coefficient for the source in Eq. (9) remains bounded at the vacuum limit. From the expression for $E_{\alpha i}^{\rho\upsilon}$ we obtain

$$\lim_{\Sigma_i\to 0} E_{\alpha i}^{\rho\upsilon} \to \frac{1}{4\pi}\int_{\alpha_i} dS \sum_{\beta\in i}\int_{(\beta\to\alpha)} d\Omega \left|\Omega\cdot\mathbf{n}\right| A^{\rho}(\Omega) A^{\upsilon}(\Omega) R_i\, e^{-\tau_{\alpha\beta}} + O\left(\Sigma_i\right), \tag{B9}$$

where we used $\lim_{\Sigma_i\to 0}\frac{1-e^{-\Sigma_i R_i}}{\Sigma_i} \to R_i - \frac{\Sigma_i R_i^2}{2} + O\left(\Sigma_i^2\right)$. Formula (B9) shows that the escape coefficients remain bounded, but unfortunately does not give any complementary relation. This means that, in case of nearly voided media a new integral, given by (B9), must be computed.

Another term that must be treated specifically at the vacuum limit is the diagonal term of operator G in equation (B3a). One can show that this diagonal term has the following limit

$$\lim_{\Sigma_i\to 0} G_{ii}^{\rho\upsilon} \to \frac{1}{4\pi}\int_{\partial V_i} dS \int_{(2\pi^+)} d\Omega \left|\Omega\cdot\mathbf{n}\right| A^{\rho}(\Omega) A^{\upsilon}(\Omega)\bullet$$
$$\{ \sum_{k\in T(r_s,\Omega)\cap V_i} [l_k^2/2 + O(\Sigma_i)] + \sum_{\substack{l,m\in T(r_s,\Omega)\cap V_i \\ l\neq m}} E(\tau_l)e^{-\tau_{lm}}E(\tau_m) \quad \},$$

that is to be computed explicitly for the nearly voided cases.

REFERENCES

1. Sanchez, R., Mondot J., Stankovski, Z., Cossic, A., Zmijarevic, I., 1988. APOLLO II: A user-oriented, portable, modular code for multigroup transport assembly calculations. Nuclear Science and Engineering 100, 352-362.
2. Halsall, M.J., 1980. CACTUS, a Characteristics Solution to the Neutron Transport Equation in Complicated Geometries. AEEW-R 1291, Atomic Energy Establishment, Winfrith, Dorchester, Dorset, United Kingdom.
3. Smith, K., 2000. "CASMO-4 characteristic methods for two-dimensional PWR and BWR core calculations". Transactions of the American Nuclear Society, 83, 294-296.
4. Dahmani M., R. Le Tellier, Roy R. and Hebert A., 2005. An efficient preconditioning technique using Krylov subspace methods for 3D characteristics solvers. Annals of Nuclear Energy 32, 876-896.
5. Hong S.G., Cho N.Z., 1998. CRX: a code for rectangular and hexagonal lattices based one the method of characteristics. Annals of Nuclear Energy 25 (8), 547-565.
6. Nam Zim Cho, 2005. Fundamentals and recent developments of reactor physics methods, Nuclear Engineering and Technology, 37, 25-78.
7. Sanchez, R., Chetaine, A., 2000. A Synthetic acceleration for a two dimensional characteristic method in unstructured meshes. Nuclear Science and Engineering 136, 122-139.
8. Santandrea S. and Sanchez R., 2002. Acceleration Techniques for the characteristics method in unstructured meshes. Annals of Nuclear Energy, 29, 323-352.
9. Santandrea S. and Sanchez R., 2005. Analysis and improvements of the DP_N acceleration technique for the method of characteristics in unstructured meshes. Annals of Nuclear Energy, 32, 163-193.
10. Sanchez, R., Mao, L., 1997. New development in the Interface-Current Module TDT. Proc. Joint Int. Conf. Mtg. Mathematical Methods and Supercomputing for Nuclear Applications. Saratoga, New York, October 5-9, 1997, p. 29, American Nuclear Society.
11. Santandrea S., 2001. Qualification Physique et amélioration du schéma numérique d'une méthode des caractéristiques pour l'équation de transport des neutrons, PHD dissertation, Université d'Evry Val d'Essonne, France.
12. Sanchez, R., Mao, L., Santandrea S., 2002. Treatment of boundary condition in trajectory based deterministic transport methods. Nuclear Science and Engineering 140, 23-50.

13. Saad, Y., 1996. Iterative Methods for Sparse Linear Systems. PWS Publishing Company, Boston, Massachusetts.
14. Warsa J.S., Wareing T.A., Morel J.E. and McGhee J.M., 2004. Krylov Subspace Iterations for Deterministic k-Eigenvalue Calculations. Nuclear Science and Engineering, 147, 26-42.
15. Suslov, I., 2002. “An algebraic collapsing acceleration method for the acceleration of the inner (scattering) iterations in long characteristics transport theory”. Proceeding of the international conference on Supercomputing in Nuclear Application, (SNA 2003) Paris 22-24 September 2003.
16. AEN-NEA, Nuclear Energy Agency, OECD 2003. Physics of Plutonium Recycling. VolumeVII: BWR MOX Benchmark-Specification and Results.
17. Zmijarevic, I., 2005. Personal Communication.
18. Leonard, A., Mc Daniel, C.T., 1995. Optimal Polar Angles and Weights for the Characteristics Method. Transactions of the American Nuclear Society, 73, 172-173.

FIGURE 1

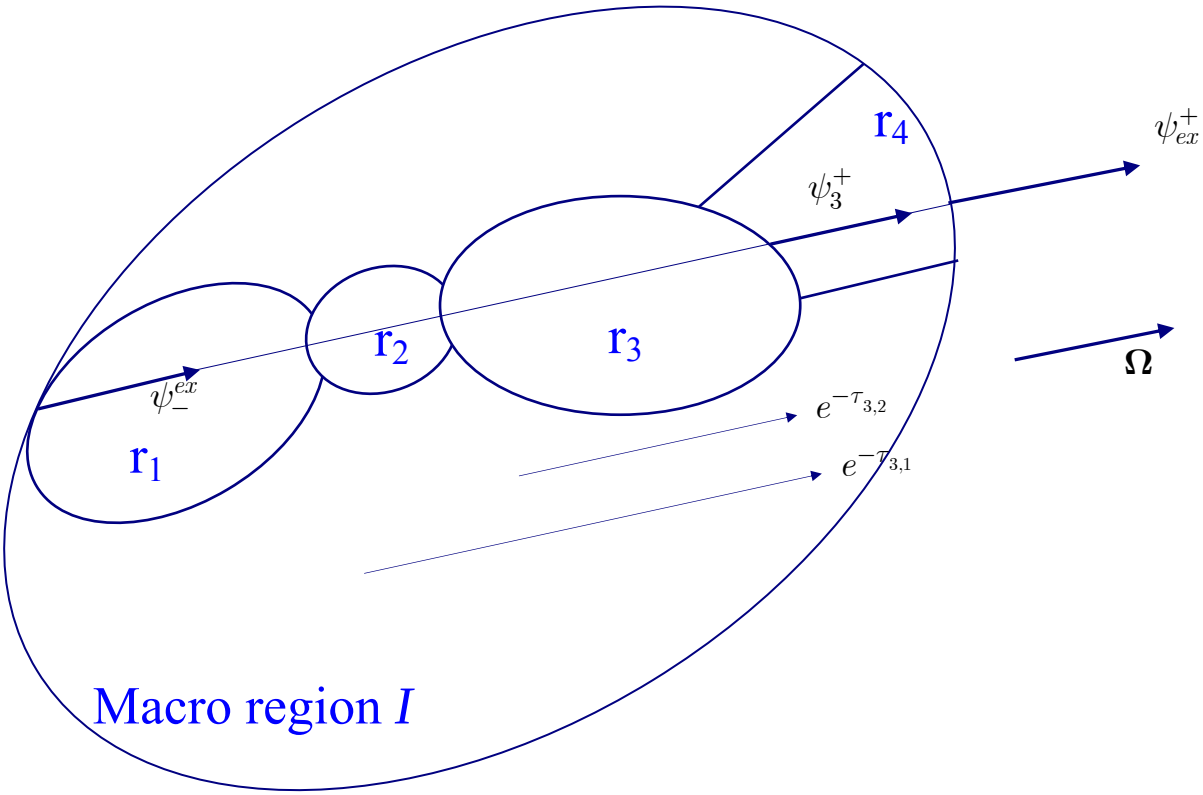

Figure 1: Sketch for the transport equation (7). The macro-region "I" contains many micro regions "r". For a trajectory along direction $\boldsymbol{\Omega}$, many intersections have to be considered from the entering to the exiting point. In order to express each exiting angular flux along this trajectory, the exponential of the optical length between these regions must be computed. For example, to express ψ_3^+ thanks to equation (7), a sum over the preceding collision sources (weighted with $e^{-\tau_{3,2}}$ and $e^{-\tau_{3,1}}$) and over the entering flux must be done.

FIGURE 2

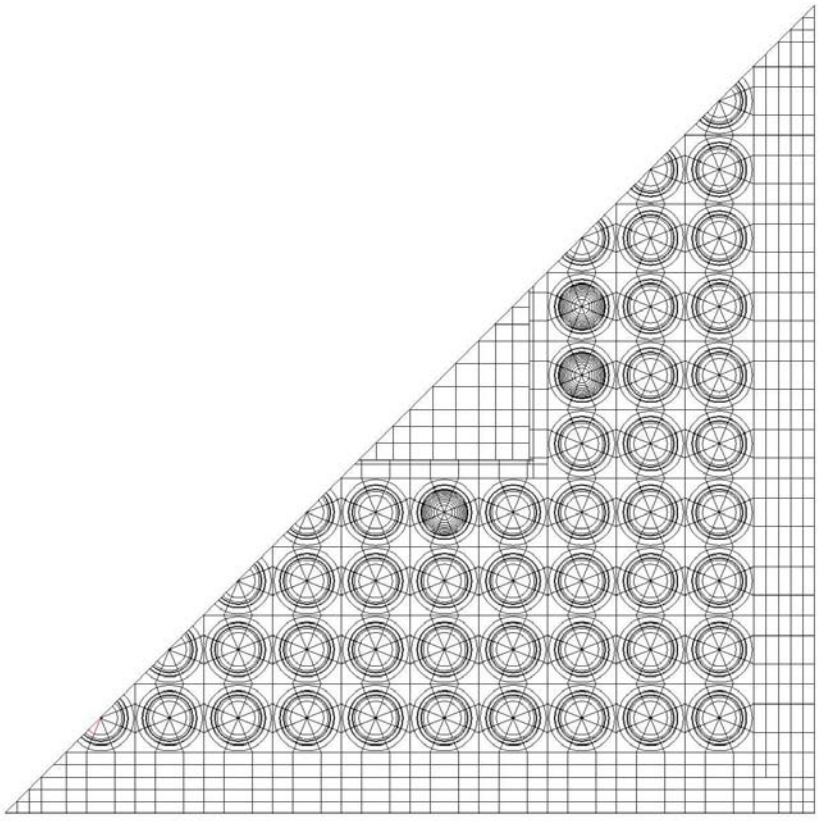

Figure 2: Computational mesh for the BWR atrium assembly.

TABLE 1

trajectories	tracks	azimuthal angles	polar angles	Transversal Integration step
162	643,617	16 (uniform)	2 (Bickley)	0.025cm
143	162,610	10 (uniform)	2 (Bickley)	0.05 cm

Table 1: Tracking parameters for the calculation of the Atrium assembly with cyclic trajectories. The number of tracks is the total number of intersections of trajectories with regions. The second and the third rows gives, respectively, the data of the reference tracking and of the DP_N low-order one.

TABLE 2

	building time	Total time	Nb. External it.	Nb. Internal it.
DP_1 Consistent tracking	22.7	41.4	7	151
IDP_1 Consistent tracking	12.8	26.4	5	102
IDP_1 Inconsistent tracking	9.44	23.6	5	107
Free Iterations		412	13	3848

Table 2: Results for the Atrium assembly calculation using the standard DP_1 operator and the new integral IDP_1 operator. All cases are DP_N-initialized eigenvalue calculations. The computational times are expressed in seconds. The reference k-effective is 1.13811.

TABLE 3

	building time	Total time	Nb. External it.	Nb. Internal it.
DP_1	19.5 sec	40.3	6	96
IDP_1	9.4 sec	22.3	3	48
Free Iterations		412	13	3848

Table 3: Results for the Atrium assembly calculation using the standard DP_1 operator and the new integral IDP_1 operator with double tracking. All cases are DP_N-initialized eigenvalue calculations and external iterations are accelerated with the external synthetic algorithm. The computational times are expressed in seconds. The reference k-effective is 1.13811.

Figure Captions

Figure 1: Sketch for the transport equation (7). Macro-region "I" contains many micro regions "r". For a trajectory along direction Ω many intersections have to be considered from the entering to the exiting point. In order to express each exiting angular flux along this trajectory the exponential of the optical length between these regions must be computed. For example to express ψ_3^{out} thank to equation (7) a sum over the preceding collision sources, weighted with $e^{-\tau_{3,2}}$ and $e^{-\tau_{3,1}}$, and over the entering flux must be done.

Figure 2: Computational mesh for the BWR Atrium assembly.