

Polynomial axial expansion in the Method of Characteristics for neutron transport in 3D extruded geometries

Laurent Graziano^{1,*}, Simone Santandrea^{1,**}, and Daniele Sciannandrone^{1,***}

¹CEA, Saclay (France)

Abstract. Recently a solver based on the Method Of Characteristics (MOC) for 3D extruded geometries has been developed in the APOLLO3® project [1]. In this method the domain is divided in regions and the Step Characteristics (SC) approximation is used in each of them to represent the solution of the transport problem. Since the biggest degree of heterogeneities is found along the radial direction, the idea proposed in this paper is to keep the SC approximation to compute the solution over the radial plane and to implement a polynomial expansion of the flux along the vertical direction. In fact the results of the previous works [1] show that the flux gradient along the z axis are likely to be represented by a classical polynomial basis. In this paper we weigh up the benefits of using a less refined mesh.

1 Introduction

In the recent years a solver based on the Method of Characteristics (MOC) allowing the treatment of 3D extruded geometries has been developed inside the TDT module of APOLLO3® [1][2] [3]. The standard Step Characteristics (SC) approximation is used and results show an excellent agreement with Monte-Carlo simulations. However a fine mesh refinement is needed to converge, due to strong flux gradients. In this paper an improvement of this method is proposed: the results of the previous work show that the flux gradients are likely to be represented by a polynomial basis along the vertical direction. Since most of the geometrical and physical heterogeneities are radially located, the SC approach is preserved to represent the solution over the radial plane. As a matter of fact the strong irregularities in the geometrical meshes prevent from an efficient use of a polynomial expansion. On the contrary along the axial direction the computational meshes assume a Cartesian shape, well suited for a polynomial representation of sources and fluxes, as shown in Figure 1. A suitable polynomial development in this direction allows us to approximate the strong flux gradients without the help of a large number of axial meshes. The seminal ideas of this method are already given in [3]. They are here developed in a different form.

2 Equation derivation

2.1 Polynomial basis definition

The starting point is the mono-group, steady state neutron transport equation is a geometrical domain D and bound-

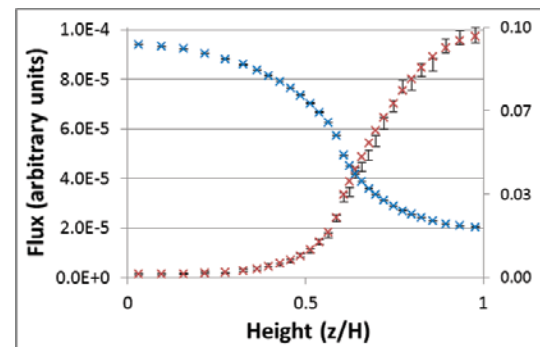


Figure 1: Axial distribution of the flux for a thermal (red) and a fast (blue) group inside the fuel pin. The figure also shows the reference Monte Carlo solution with associated error bars [3].

ary Γ :

$$\begin{aligned} [\vec{\Omega}\nabla + \Sigma_r(\vec{r})] \Psi(\vec{r}, \vec{\Omega}) &= q(\vec{r}, \vec{\Omega}) \quad \vec{r} \in D, \vec{\Omega} \in S_{4\pi} \\ \Psi(\vec{r}, \vec{\Omega}) &= \beta \Psi(\vec{r}, \vec{\Omega}) \quad r \in \Gamma, \vec{\Omega} \in S_{4\pi}^+(\vec{r}) \end{aligned} \quad (1)$$

where $q(\vec{r}, \vec{\Omega})$ is the source term, $S_{4\pi}$ stands for the unit sphere directions, $S_{4\pi}^+(\vec{r})$ is the set of exiting directions from D in the point \vec{r} and β is a function relating entering and exiting direction.

A local polynomial basis can be used to express sources and flux moments:

$$\vec{P}(z_r) = \{z_r^p = \left(\frac{z_r - \bar{z}_r}{\Delta z_r/2}\right)^p, \quad 0 \leq p \leq N_p\} \quad (2)$$

where Δz_r is the height of the given region, $z_r \in [-\Delta z_r/2, \Delta z_r/2]$ is the local coordinate for the r_{th} region,

*e-mail: laurent.graziano@cea.fr
 **e-mail: simone.santandrea@cea.fr
 ***e-mail: daniele.sciannandrone@cea.fr

\bar{z}_r is the value of the axial coordinate at the region center and N_p is the chosen order for the polynomial expansion.

Thanks to this definition $\bar{z}_r \in [-1, 1]$ for every region, independently on the region height. In this way passing from a region to the next one through an horizontal border the polynomial value only change is sign, switching from +1 to -1. This property allows to avoid a certain number of operations during the trajectory sweep, as it will be shown later.

From the local polynomial basis $\vec{P}(\bar{z})$ is possible to retrieve an orthonormalized basis $\vec{\xi}(\bar{z})$ through the Gram-Schmidt orthonormalization algorithm. The coefficients of this algorithm can be used to form a lower triangular matrix that allows to switch from the orthonormal to the polynomial basis:

$$\vec{\xi}(\bar{z}) = \bar{\mathbf{M}}^{-1} \vec{P}(\bar{z}) \quad (3)$$

where:

$$\bar{\mathbf{M}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{10} & m_{11} & 0 & 0 \\ m_{20} & m_{21} & m_{22} & 0 \\ m_{30} & m_{31} & m_{32} & m_{33} \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \text{with} \quad \begin{aligned} m_{pj} &= \frac{\langle P_p, \xi_j \rangle}{\|\xi_p(\bar{z})\|} \\ m_{pp} &= \|\xi_p(\bar{z})\| \end{aligned}$$

The terms $\langle P_p, \xi_j \rangle$ are computed using the geometrical matrix $\bar{\mathbf{P}}$:

$$\bar{\mathbf{P}} = \frac{1}{V_r} \int d\vec{r} \vec{P}(\bar{z}) \otimes \vec{P}(\bar{z}) = \int_{-1}^{+1} d\bar{z} \vec{P}(\bar{z}) \otimes \vec{P}(\bar{z}) \quad (4)$$

which elements read:

$$\bar{P}_{pp'} = \langle P_p, P_{p'} \rangle = \begin{cases} \frac{1}{p+p'+1} & \text{for } p+p' \text{ even} \\ 0 & \text{for } p+p' \text{ odd} \end{cases}$$

and V_r denotes the volume of the r region.

2.2 Source expansion

The source term can now be expanded over the orthonormalized basis:

$$q(\vec{r}, \vec{\Omega}) = \sum_r \theta_r(\vec{r}) \vec{\xi}(\bar{z}_r) \cdot \vec{q}_r(\vec{\Omega}) \quad (5)$$

where θ_r is the characteristic function for the r_{th} region and \vec{q}_r are the orthonormal coefficients of the source moments:

$$\vec{q}_r(\vec{\Omega}) = \sum_n A_n(\vec{\Omega}) \cdot \vec{q}_r^n(\vec{\Omega}) = \sum_n A_n(\vec{\Omega}) \cdot [\Sigma_r^n \vec{\phi}_r^n + \vec{S}_r^n] \quad (6)$$

Σ_r^n is the macroscopic in-group transfer cross section, A_n the real spherical harmonics, $\vec{\phi}_r^n$ the coefficients of the flux moments expansion and \vec{S}_r^n is the external source, accounting for fissions and transfer from other groups. The sum over n stands for:

$$\sum_n A_n(\vec{\Omega}) \vec{q}_r^n(\vec{\Omega}) = \sum_{l=0}^{l=K} \sum_{m=-l}^{m=l} A_l^m(\vec{\Omega}) \vec{q}_{r,l}^m(\vec{\Omega})$$

Where K is the anisotropy order.

Thanks to the orthonormality property, by projecting $q(\vec{r}, \vec{\Omega})$ over the polynomial basis $\vec{\xi}(\bar{z})$ we get:

$$\begin{aligned} \int_{-1}^{+1} d\bar{z} \vec{\xi}(\bar{z}) q(\vec{r}, \vec{\Omega}) &= \int_{-1}^{+1} d\bar{z} \vec{\xi}(\bar{z}) \otimes \vec{\xi}(\bar{z}) \vec{q}_r(\vec{\Omega}) \\ &= \begin{cases} 0 & \text{if } p \neq p' \\ \vec{q}_r(\vec{\Omega}) & \text{if } p = p' \end{cases} \end{aligned}$$

This important property is very useful, as shown later, and justify the choice of a orthonormalized basis.

To switch from the orthonormal source coefficients to the polynomial one the following relation is used, retrieved from the definition (3):

$${}^t(\bar{\mathbf{M}}^{-1} \vec{P}(\bar{z})) \vec{q}_r(\vec{\Omega}) = {}^t(\vec{P}(\bar{z})) \cdot \vec{q}_{r,pol}(\vec{\Omega})$$

From which we get:

$$\vec{q}_{r,pol} = {}^t\bar{\mathbf{M}}^{-1} \cdot \vec{q}_r(\vec{\Omega}) \quad \text{and} \quad \vec{q}_r = {}^t\bar{\mathbf{M}} \cdot \vec{q}_{r,pol}(\vec{\Omega}) \quad (7)$$

where $\vec{q}_{r,pol}(\vec{\Omega})$ is the coefficient vector referring to the polynomial basis $\vec{P}(\bar{z})$.

2.3 Transmission equation

The SC numerical MOC transmission equation obtained from the integral transport equation reads:

$$\Psi_r(t, \vec{\Omega}) = \Psi_r(0, \vec{\Omega}) e^{-\tau(0,t)} + \beta(t, \vec{\Omega}) q_r(t, \vec{\Omega})$$

where $\Psi_r(0, \vec{\Omega})$ is the entering flux at the trajectory beginning, $\Psi_r(t, \vec{\Omega})$ is the flux in a generic point t along the trajectory and:

$$\beta_r(t, \vec{\Omega}) = \left[\frac{1 - e^{-\tau(0,t)}}{\Sigma_r} \right] \quad \tau(0, t) = \int_0^t dt' \Sigma_r$$

are called respectively the *transmission coefficient* and the *optical length*.

The source expansion of Eq.(5) is now used. Eq.(3) allows us to switch to the standard polynomial basis and a local system of coordinates along the trajectory is used to express the \bar{z} component:

$$\bar{z}_r = z_r^{in} - \bar{z}_r + \mu t \quad \mu = \cos(\theta) \quad t \in (0, l_r)$$

z_r^{in} is the z coordinate of the entering point of the trajectory in the r region, θ is the polar component of the $\vec{\Omega}$ direction. The polynomial transmission equation obtained in this way reads:

$$\Psi_r(t, \vec{\Omega}) = \Psi_r(0, \vec{\Omega}) \cdot e^{-\tau(0,t)} + \quad (8)$$

$$\sum_{p=0}^{N_p} \left[\sum_{k=0}^p c_{p,k} P_k(z_r^{in}) \mu^{p-k} \left(\frac{2}{\Delta z_r} \right)^{p-k} E_{p-k}(\tau(t)) \right] \cdot \frac{(\vec{q}_{r,pol}(\vec{\Omega}))_p}{\Sigma_r}$$

where:

$$c_{pk} = \frac{p!}{k!(p-k)!}$$

$$E_{p-k}(\tau(t)) = \frac{1}{\Sigma_r^{(p-k)}} \int_0^{\tau(t)} d\tau' \tau'^{p-k} e^{(\tau'-\tau(t))} \quad (9)$$

and $P_k(\vec{z}_r^{in})$ is defined in (2).

For numerical reasons it is useful to rearrange the transmission equation as follows:

$$\Psi_r(t, \vec{\Omega}) = \Psi_r(0, \vec{\Omega}) \cdot e^{-\tau(0,t)} + \vec{P}(\vec{z}_r^{in}) \cdot \vec{T} \quad (10)$$

where:

$$T_k = \sum_{p=\max(k,1)}^{N_p} F_{p,k} \frac{(\vec{q}_{r,pol}(\vec{\Omega}))_p}{\Sigma_r}$$

$$F_{p,k} = c_{p,k} \mu^{p-k} \left(\frac{2}{\Delta z} \right)^{p-k} E_{p-k}(\tau(t))$$

This final form of the transmission equation allows to minimize the number of floating point operations to be executed on-the-fly during the sweep, for each chord. The term \vec{T} , on the other hand, can be pre-computed during the classification phase. The classification process is widely explained in [3]. In a few words, the chord classification is a procedure that exploits the regularity of extruded geometries to recognize chords with the same length. Starting for a given 2D chord, a given polar angle and moving vertically, every chord that lies on that basic 2D chord and intersect two successive vertical surface will have the same length. Fig. 2 gives a visual interpretation of the concept. These chords intersecting two successive vertical surfaces are referred to as the *V-chords* in [3]. In the same way *H-chords* are defined as chords intersecting two successive horizontal surfaces, and *M-chords*, as chords intersecting a horizontal and then a vertical surface (or vice versa). The advantage of the classification process is that when the same chords is found several times, its coefficient can be computed just once. Since the \vec{T} term is computationally expensive, the possibility to strongly reduce the number of times that it must be computed represent a big advantage.

Computational cost

The transmission sweep is one of the most expensive part of the iterative scheme in computational terms. The transmission equation adapted to the polynomial case derived in this section is clearly more expensive than the classical SC transmission, which asks for one floating point operation per chord (for classified chords). On the other hand the polynomial equation needs $1+N_p$ operations (N_p being the polynomial degree). Moreover, it requires to know at each point during the trajectory sweep the value of the axial coordinate and all its powers. If this information is computed on-the-fly it requires $1+N_p$ additional operations. In addition, even the coefficients that can be pre-computed during the classification phase are much more expensive. For these reason for an equivalent tracking (so the same amount and type of chords) the polynomial solution will be more expensive in computational terms.

2.4 Escape coefficient computation

The escape coefficients $E_{p-k}(\tau)$ can be computed integrating Eq. (9). However, by integrating by parts this equation, a useful recursive relation can be obtained. This allows to compute only the 0_{th} order coefficient and to retrieve the others:

$$E_0(\tau) = 1 - e^{-\tau}$$

$$E_b(\tau) = l^b - \frac{b}{\Sigma_r} E_{b-1}(\tau) \quad 1 \leq b \leq N_p \quad (11)$$

The generic escape coefficient E_b can be expressed as a function of the first one through the relation:

$$E_0(\tau) = 1 - e^{-\tau}$$

$$E_1(\tau) = l - \frac{1}{\Sigma_r} + \frac{1}{\Sigma_r} e^{-\tau}$$

$$E_2(\tau) = l^2 - \frac{2}{\Sigma_r} l + \frac{2}{\Sigma_r^2} - \frac{2}{\Sigma_r^2} e^{-\tau}$$

$$E_3(\tau) = l^3 - \frac{3}{\Sigma_r} l^2 + \frac{3 \cdot 2}{\Sigma_r^2} l - \frac{3 \cdot 2}{\Sigma_r^3} + \frac{3 \cdot 2}{\Sigma_r^3} e^{-\tau}$$

$$E_b(\tau) = \sum_{i=0}^b \left[\frac{b!}{i!} \frac{l^i}{(-\Sigma_r)^{b-i}} \right] - \frac{b!}{(-\Sigma_r)^b} e^{-\tau}$$

Unfortunately the recursive relation Eq. (11) is ill-conditioned for small values of the total cross section. Another possible approach is to start with the highest order coefficient and retrieve the others with the corresponding backward recursive relation. This is a more stable approach for small τ values. In this case the highest order coefficient must be computed. The more efficient way to do this is through the use of tabulated values. Again, small values of the optical length cause big instabilities not only in the use of the recursive relation, but also in the initial computation of the highest order coefficient.

To avoid this problem the strategy adopted is to tabulate a different function. The escape coefficient Eq. (9) is computed as:

$$E_b(\tau) = l^b E_b^T(\tau)$$

$$E_b^T(\tau) = \frac{1}{\tau^b} \int_0^{\tau(t)} d\tau' \tau'^b e^{(\tau'-\tau(t))} \quad (12)$$

The function $E_b^T(\tau)$ happens to be well-conditioned even for small τ values. Integrating (12) by parts a forward and a backward relation can be retrieved:

$$\text{forward} \rightarrow E_b^T(\tau) = 1 - \frac{b}{\tau} E_{b-1}^T(\tau)$$

$$\text{backward} \rightarrow E_b^T(\tau) = \frac{\tau}{b+1} \left[1 - E_{b+1}^T(\tau) \right]$$

Also in this case the backward relation is the more stable for small values of the optical length. A table must be pre-computed for the highest order escape coefficient. Instead of integrating the relation (12) a more efficient approach can be the following: the forward recursive relation is used starting from the 0_{th} order value for large values of the optical length and a Maclaurin series is used to expand the function $e^{-\tau}$ for small τ values.

For large τ :

$$E_0^T(\tau) = 1 - e^{-\tau}$$

$$E_b^T(\tau) = 1 - \frac{b}{\tau} E_{b-1}^T(\tau)$$

For small τ :

$$E_0^T(\tau) = 1 - e^{-\tau} = \tau - \frac{\tau^2}{2} + \frac{\tau^3}{3!} - \dots$$

$$E_1^T(\tau) = 1 - \frac{1}{\tau} E_0^T(\tau) = \frac{\tau}{2} - \frac{\tau^2}{3!} + \frac{\tau^3}{4!} - \dots$$

$$E_2^T(\tau) = 1 - \frac{2}{\tau} E_1^T(\tau) = \frac{2}{3!} \tau - \frac{2}{4!} \tau^2 + \frac{2}{5!} \tau^3 - \dots$$

$$E_3^T(\tau) = 1 - \frac{3}{\tau} E_2^T(\tau) = \frac{3!}{4!} \tau - \frac{3!}{5!} \tau^2 + \frac{3!}{6!} \tau^3 - \dots$$

$$E_b^T(\tau) = \sum_{i=1}^{\infty} (-)^{i+1} \frac{b!}{(b+i)!} \tau^i$$

$$= \frac{\tau}{b+1} \left[1 - \frac{\tau}{b+2} \left[1 - \frac{\tau}{b+3} \left[1 - \frac{\tau}{4} \right] \right] \right]$$

2.5 Balance equation

A balance equation can be obtained by projecting the transport equation (1) over $\vec{\xi}(\vec{z})$:

$$\frac{1}{V_r} \int_{V_r} d\vec{r} \vec{\xi}(\vec{z}) (\vec{\Omega} \nabla \Psi(\vec{r}, \vec{\Omega})) + \frac{\Sigma_r}{V_r} \int_{V_r} d\vec{r} \vec{\xi}(\vec{z}) \Psi(\vec{r}, \vec{\Omega}) = \vec{q}(\vec{r}, \vec{\Omega})$$

where the integral is performed over the r region.

The following definition is given for the orthonormal coefficients of the angular flux:

$$\vec{\Psi}(\vec{r}, \vec{\Omega}) = \frac{1}{V_r} \int_{V_r} d\vec{r} \vec{\xi}(\vec{z}) \Psi(\vec{r}, \vec{\Omega}) \quad (13)$$

The equation is rearranged, the basis switch is done again with Eq. (3) and the following useful property of the polynomial basis $\vec{P}(\vec{z})$ is used:

$$\frac{\partial P_p}{\partial z} = \frac{p}{(\Delta z/2)} P_{p-1}$$

The final balance equation obtained is:

$$\boxed{\Sigma_r \vec{\Psi}(\vec{r}, \vec{\Omega}) = \vec{q}_r(\vec{r}, \vec{\Omega}) - \vec{M}^{-1} \cdot \Delta \vec{J}(\vec{r}, \vec{\Omega}) + \mu \vec{C} \vec{\Psi}(\vec{r}, \vec{\Omega})} \quad (14)$$

where the current term is:

$$\Delta \vec{J}(\vec{r}, \vec{\Omega}) = \frac{1}{V_r} \sum_{\substack{j \in \vec{\Omega}_j \\ t \cap r}} \Delta_r [\vec{P}(z_r^{out}) \Psi_r^{out} - \vec{P}(z_r^{in}) \Psi_r^{in}]$$

Here, $\sum_{j \in \vec{\Omega}_j}$ means that the sum is performed for all the trajectories intersecting the r region parallel to the direction $\vec{\Omega}_j$.

The \vec{C} matrix reads:

$$\vec{C} = \vec{M}^{-1} \cdot \left[\frac{p}{\Delta z/2} \right] \cdot \vec{S} \cdot \vec{M}$$

and the shift matrix \vec{S} is:

$$\vec{S} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots \end{bmatrix}$$

Since the profile of the \vec{C} matrix is lower triangular with zeros on the main diagonal the components of $\vec{\Psi}(\vec{r}, \vec{\Omega})$ can be computed treating the system as a standard lower triangular system of equations, solving starting from the 0 order polynomial component.

2.6 Rebalancing

The tracking strategies used in the Method of Characteristics for three-dimensional geometries are widely explained in [1]. The same strategies apply to the polynomial version, since the same type of geometries are treated.

As already said, the use of a polynomial development leads to a lower numbers of axial meshes. This means regions with bigger volumes that could be integrated, with similar relative numerical errors, using a bigger axial integration parameter. Fig.2 represents the radial and axial integration parameters, respectively Δr and Δs . As supposed, to a bigger Δs value corresponded an equivalently precise volume integration. Unfortunately, the same thing can not be said about the precision of the solution. Releasing the Δs value caused big instability issues, that lead in most of the cases to a divergent solution.

The problem has been identified in a difference between the balance used to compute the flux moments and the one coming from the transmission equation. A *rebalancing* technique has been implemented to force the two balances to coincide. The equations for this technique are retrieved starting by integrating the transmission equation along a chord of length L :

$$L \vec{\Psi}(L, \vec{\Omega}) = \int_0^L dt \Psi_r(t, \vec{\Omega}) = \int_0^L dt \Psi_r(0, \vec{\Omega}) e^{-\tau(0,t)} + \int_0^L dt \sum_{p=0}^{N_p} \left\{ \left[\sum_{k=0}^p c_{p,k} P_k(z^{in}) \mu^{p-k} \left(\frac{2}{\Delta z} \right)^{p-k} E_{p-k}(\tau(t)) \right] \cdot \frac{(\vec{q}_{r,pol}(\vec{\Omega}))_p}{\Sigma_r} \right\}$$

The only terms depending on t are:

$$\int_0^L dt \Psi_r(0, \vec{\Omega}) e^{-\tau(0,t)} = \Psi_r(0, \vec{\Omega}) \left[\frac{1 - e^{-\Sigma L}}{\Sigma} \right]$$

$$\int_0^L dt E_{p-k}(\tau(t)) = \frac{L^{p-k+1}}{p-k+1} - \frac{E_{p-k}(\tau(L))}{\Sigma}$$

So we obtain:

$$\begin{aligned} \Sigma L \bar{\Psi}(L, \vec{\Omega}) &= \Psi_r(0, \vec{\Omega}) - \Psi_r(L, \vec{\Omega}) e^{-\Sigma L} + \\ &- \sum_{p=0}^{N_p} \left\{ \left[\sum_{k=0}^p c_{p,k} P_k(z^{in}) \mu^{p-k} \left(\frac{2}{\Delta z} \right)^{p-k} E_{p-k}(\tau(t)) \right] \cdot \frac{(\vec{q}_{r,pol}(\vec{\Omega}))_p}{\Sigma_r} \right\} \\ &+ \sum_{p=0}^{N_p} \left\{ \left[\sum_{k=0}^p c_{p,k} P_k(z^{in}) \mu^{p-k} \left(\frac{2}{\Delta z} \right)^{p-k} \frac{L^{p-k+1}}{p-k+1} \right] \cdot \frac{(\vec{q}_{r,pol}(\vec{\Omega}))_p}{\Sigma_r} \right\} \\ &= \Psi_r(0, \vec{\Omega}) - \Psi_r(L, \vec{\Omega}) + \sum_{p=0}^{N_p} \left[\gamma_p(t) \cdot (\vec{q}_{r,pol}(\vec{\Omega}))_p \right] \end{aligned}$$

where:

$$\gamma_p(t) = \sum_{k=0}^p c_{p,k} P_k(z^{in}) \mu^{p-k} \left(\frac{2}{\Delta z} \right)^{p-k} \frac{L^{p-k+1}}{p-k+1}$$

Integrating now the equation over the volume of each region we obtain an average angular flux. Multiplying by the real spherical harmonic and integrating over the angular variable we obtain a balance equation for the angular flux moments, only for the constant polynomial value:

$$\begin{aligned} \Sigma V_r \phi_{0,r}^n &= \Sigma \frac{1}{4\pi} \oint d\vec{\Omega} A_n(\vec{\Omega}) \int_{\delta r} d_{2r\perp} L \bar{\Psi}_r(L, \vec{\Omega}) = \\ &\frac{1}{4\pi} \oint d\vec{\Omega} A_n(\vec{\Omega}) \int_{\delta r} d_{2r\perp} (\Psi_r(0, \vec{\Omega}) - \Psi_r(L, \vec{\Omega})) + \\ &\frac{1}{4\pi} \oint d\vec{\Omega} A_n(\vec{\Omega}) \int_{\delta r} d_{2r\perp} \sum_{p=0}^{N_p} \left[\gamma_p(t) \cdot (\vec{q}_{r,pol}(\vec{\Omega}))_p \right] \end{aligned}$$

where the volume integral has been decomposed coherently with the trajectory-based spatial discretization, and computed numerically as:

$$\int_r d\vec{r} = \int_{\delta r} d_{2r\perp} \int_0^L dt \simeq \sum_{\substack{t \parallel \vec{\Omega}_j \\ t \cap r}} \int_0^L dt$$

Eq.(6) and Eq.(7) are now used to expand the source term and to switch to the orthonormal basis, obtaining (written as the numerical equivalent form):

$$\begin{aligned} \Sigma V_r \phi_{0,r}^n &= \Sigma \sum_{\vec{\Omega}_j \in S_N} \omega(\vec{\Omega}_j) A_n(\vec{\Omega}_j) (\Psi_r(0, \vec{\Omega}) - \Psi_r(L, \vec{\Omega})) + \\ &+ \sum_{p=0}^{N_p} \sum_{n'} \left[\sum_{p'=0}^{N_p} \bar{M}_{pp'}^{-1} \sum_{\vec{\Omega}_j \in S_N} \omega(\vec{\Omega}_j) A_n(\vec{\Omega}_j) A_{n'}(\vec{\Omega}_j) \cdot \right. \\ &\left. \sum_{\substack{t \parallel \vec{\Omega}_j \\ t \cap r}} \Delta_t \cdot \gamma_{p'}(t) \right] \cdot q_{p,r}^{n'} \end{aligned}$$

The last term can be expressed in a tensorial form, defining the operator $\bar{\Gamma}$ with dimension $N_p \leftarrow N_p * N_{mom}$:

$$\begin{aligned} (\bar{\Gamma} : \bar{q}_r)_n &= \sum_{p=0}^{N_p} \sum_{n'} \left[\sum_{p'=0}^{N_p} \bar{M}_{pp'}^{-1} \cdot \right. \\ &\left. \sum_{\vec{\Omega}_j \in S_N} \omega(\vec{\Omega}_j) A_n(\vec{\Omega}_j) A_{n'}(\vec{\Omega}_j) \sum_{\substack{t \parallel \vec{\Omega}_j \\ t \cap r}} \Delta_t \cdot \gamma_{p'}(t) \right] \cdot q_{p,r}^{n'} \end{aligned}$$

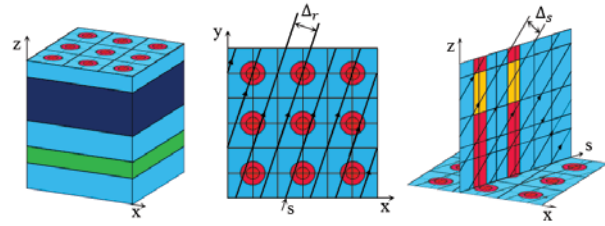


Figure 2: Tracking strategy for 3D axial geometries. On the left the whole geometry, composed by a 2D unstructured geometry extruded along z . In the middle are shown the 2D trajectories, with spacing Δr . Finally on the right each 2D trajectory generate a sz -plane on which the actual 3D trajectory is traced, with axial spacing Δs [1].

where ":" refers to the double tensorial reduction.

This new balance equation can be compared to the standard balance equation, that reads:

$$\begin{aligned} \Sigma V_r \phi_{0,r}^n &= \Sigma \sum_{\vec{\Omega}_j \in S_N} \omega(\vec{\Omega}_j) A_n(\vec{\Omega}_j) (\Psi_r(0, \vec{\Omega}) - \Psi_r(L, \vec{\Omega})) + \\ &+ V_r (\bar{S}_p \cdot \vec{q}_{0,r})_n \end{aligned}$$

where the *Spurious* matrix \bar{S}_p reads:

$$\bar{S}_p = \frac{1}{4\pi} \oint d\vec{\Omega} A_n(\vec{\Omega}) \otimes A_n(\vec{\Omega})$$

A rebalancing equation can be obtained forcing the two equations to give the same result. In particular, since only the last term is different in the two equations, a correction to be applied to the 0_{th} polynomial order transport source can be obtained, in order to let this two terms to coincide:

$$V_r \bar{S}_p \cdot \vec{q}_{0,r} = \bar{\Gamma} : [\bar{q}_r + \delta \vec{q}_{0,r}]$$

from which we obtain:

$$\delta \vec{q}_{0,r} = \bar{\Gamma}_0^{-1} \cdot \left[V_r \bar{S}_p \cdot \vec{q}_{0,r} - \bar{\Gamma} : \bar{q}_r \right]$$

where $\bar{\Gamma}_0$ is:

$$\begin{aligned} (\bar{\Gamma}_0 \cdot \delta \vec{q}_{0,r})_n &= \\ &\sum_{n'} \left[\sum_{\vec{\Omega}_j \in S_N} \omega(\vec{\Omega}_j) A_n(\vec{\Omega}_j) A_{n'}(\vec{\Omega}_j) \sum_{\substack{t \parallel \vec{\Omega}_j \\ t \cap r}} \Delta_t \cdot \gamma_0(t) \right] \cdot \delta q_{0,r}^{n'} \end{aligned}$$

where the \bar{M}^{-1} and his related sum over p' disappeared since the matrix is lower triangular.

3 Results

The polynomial axial MOC method has been validated using the same calculation scheme and the same case of a previous work[3]. This case is a heterogeneous three-dimensional assembly of the innovative Sodium-cooled

Fast Breeder Reactor ASTRID, developed at the CEA in France. The accuracy of the solution is analyzed comparing the results with a reference Monte Carlo simulation obtained with the TRIPOLI4® code, also developed at the CEA.

The two-dimensional section of the ASTRID assembly used in the calculation is shown in Fig. 3a, while Fig. 3b shows the actual computational domain. Axially, the geometry is composed by a fertile part superposed to a fissile section.

The multi-group cross sections have been computed with the Subgroup method of the AUTOP module of APOLLO3®. The self-shielding procedure and the 2D-3D equivalence applied is explained in [2][3]. A 1968 groups discretization of the energy is used for the multi-group calculation.

The presence of two parts with different materials and composition leads to the strong flux gradient, shown in Fig. 1. For this reason the Step approximation needs a big number of axial meshes, in particular in the interface section, and justifies the idea of a polynomial development. Fig. 4 shows the amount of meshes needed both in the Step and in the polynomial case to properly describe the flux gradients. The use of a polynomial development allowed to decrease the meshes number, from 30 to 2.

Although the number of axial meshes decreased of about one order of magnitude, the number of chords did not decreased proportionally (for the same values of the integration parameters described in section 2.6). In fact, the most abundant chords even with 30 planes in this geometry (and probably in most of them) are the *V-chord*, connecting two vertical surfaces. The reason lies in the length of the average 2D chord, smaller than the average plane height. Reducing the number of meshes the number of *M-chords* decreased, while the *H-chords* basically disappeared, leading to a percentage of *V-chords* > 95%. Nevertheless the total number of chords slightly decreased. The *V-chords* became almost the only type. The bigger percentage of *V-chords* in comparison with the step case represent an advantage, since it is possible to pre-compute the transmission coefficients for classified chords. However, since the amount of *V-chords* was already high in the step case and since the number of floating operation per chord in the polynomial case is much higher than in the SC approximation, as explained in section 2.3, the hoped advantage in terms of computational time has not been achieved. Calculations have been performed on a Xeon E5-2680@2.8 GHz with two CPUs sharing 256GB of main memory. Each CPU is composed of 10 cores, with 12GB of memory each.

Table 1 shows a precise comparison of chords type and amount between Step and Polynomial method as well as the reactivity errors and computational times for both methods.

Figs. 5, 6, 7 shows the absolute errors on the absorption and fission reaction rates of the Pu_{94}^{239} , U_{92}^{235} and U_{92}^{238} isotopes in the four fuel pins homogenization zones (external, internal, fissile, fertile), over the 33 homogenization

Table 1: Comparison between the Step and the Polynomial method regarding the chords amount, types and computational time, using the same integration parameters. Since the DP_N acceleration is available for the Step method, but not yet for the Polynomial development, both calculation are performed without acceleration. M stands for million.

	Step	Polynomial
Δr (cm)	0.05	0.05
Δs (cm)	0.2	0.2
Total number of chords	44.79 M	38.06 M
• number of V chords	31.75 M	37,10 M
• number of H chords	1.38 M	0
• number of M chords	11.66 M	0.96 M
Classified chords	73.97 %	97.47 %
Chord reduction	-	-15.03 %
Computational time	14 393 s	20 069 s
Time overhead	-	+39.44 %
Reactivity error (pcm)	-1.38	+9.39

groups. Comparison with TRIPOLI4® are done normalizing each rate to the total absorption reaction rate.

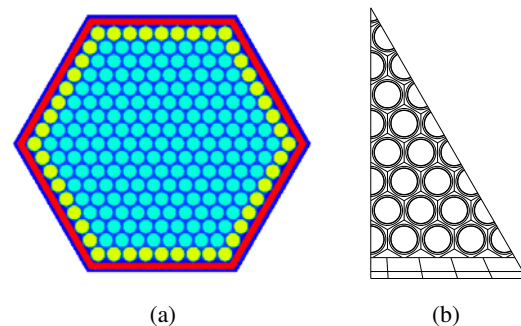


Figure 3: On the left the two dimensional section of the ASTRID assembly with $2\pi/12$ symmetry. The two different colors of the fuel pins represents the two self-shielding areas. On the right the actual computational domain, that corresponds to one twelve of the full assembly.

4 Conclusions and perspectives

The polynomial axial development has been implemented for the 3D Method of Characteristics. Results show a good agreement with the Monte Carlo simulation and with the results obtained in the previous work, where the 3D SC method has been developed[2][3].

The tracking size for the Polynomial case, thanks to a lower amount of meshes, is smaller and the percentage of classified chords is bigger. Nevertheless the computational cost is still higher for the polynomial case. Two factors may come in use to increase the computational performances of the Polynomial method, that we will try to

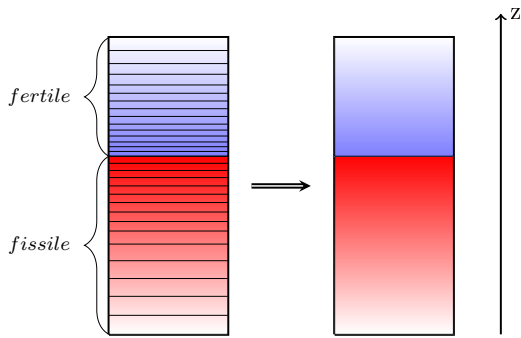


Figure 4: Difference between the Step and the Polynomial axial discretization.

develop. First, the DP_N acceleration technique that will be developed should take advantage of the lower amount of meshes both for the important memory occupation that this method requires, and for the computational cost that strongly depends on the number of unknowns. Secondly, thanks to the bigger axial size of the meshes, an equivalent precise solution should be obtained using a bigger axial integration parameter (Δs) and a significant lower number of chords. This analysis is already in progress, that is why the *rebalancing* technique explained in Sec. 2.6 has been implemented. Tests carried out with bigger Δs seem to show equally precise values in term of reactivity. However we cannot consider these results conclusive, since the reactiv-

ity and the integrated reactions rates proved to be well calculated, even if the axial profile of the flux was not. This phenomena observed while using the Step method is not yet well understood, however it may cause errors for example in an evolution calculation. For this reason we leave the discussion of the possibility to increment the value of the axial integration parameters for the future, when conclusive results will be available.

References

- [1] D. SCIANNANDRONE, S. SANTANDREA, R. SANCHEZ, *Optimized tracking strategies for step MOC calculations in extruded 3D axial geometries*, *Annals of Nuclear Energy* (2015)
- [2] D. SCIANNANDRONE, S. SANTANDREA, R. SANCHEZ, J. F. Vidal et al., *Coupled fine-group three-dimensional flux calculation and subgroups method for a FBR hexagonal assembly*, *Proc. Int. Conf. ANS MC2015, Nashville, Tennessee* (April 19-23, 2015)
- [3] D. SCIANNANDRONE, *Acceleration and higher order schemes of a characteristics solver for the neutron transport equation in 3D axial geometries*, PHD thesis, Université Paris Sud-Orsay (2015)
- [4] P. ARCHIER, J-M PALAU, J-F VIDAL, S. SANTANDREA, D. SCIANNANDRONE, *Validation of the newly implemented 3D TDT-MOC solver of APOLLO3® code on a whole 3D SFR heterogeneous assembly*, *Physor 2016, American Nuclear Society, Sun Valley Idaho, USA May 1-5* (2016)

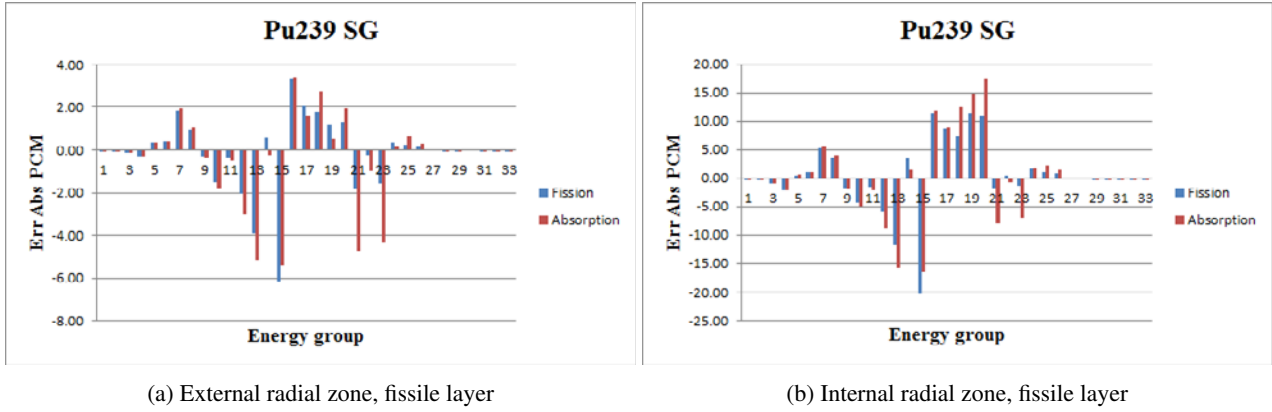


Figure 5: Absolute error over the 33 homogenization groups of the absorption and fission reactions rates for the isotope Pu_{94}^{239} .

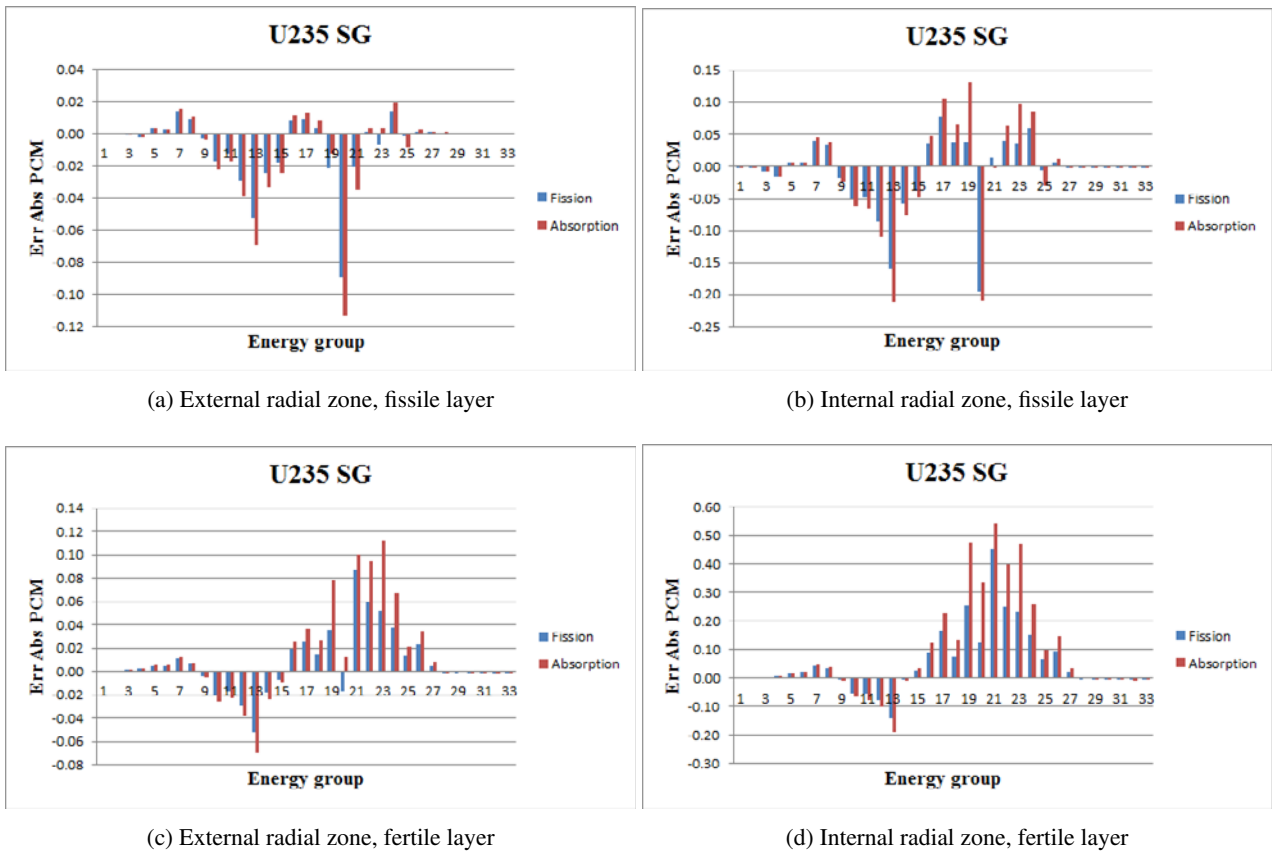
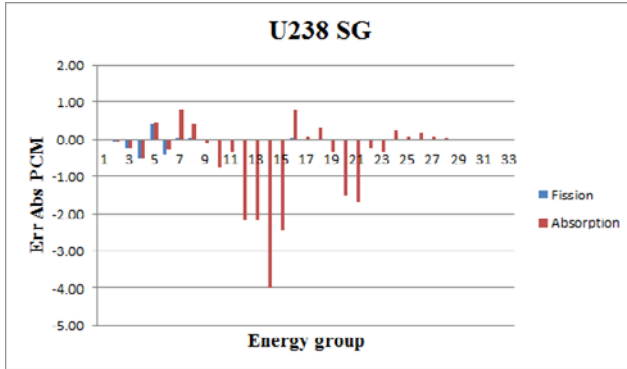
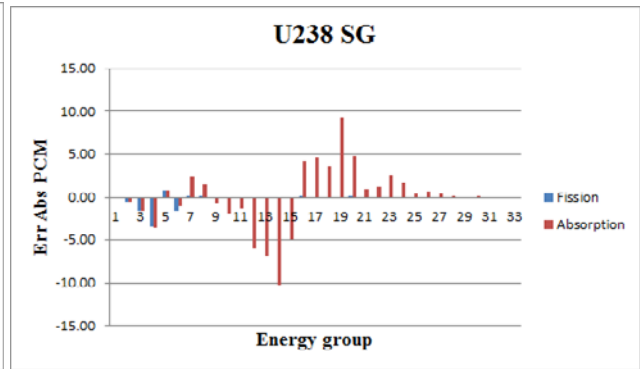


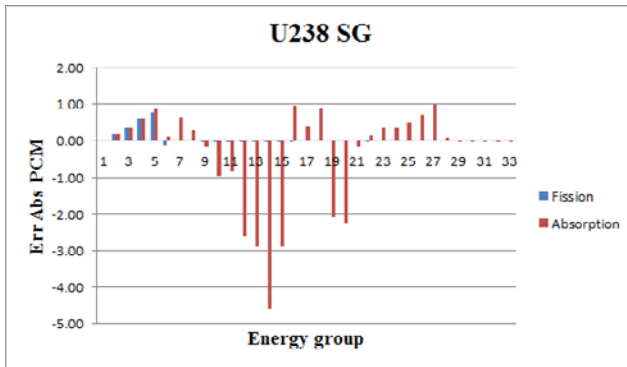
Figure 6: Absolute error over the 33 homogenization groups of the absorption and fission reactions rates for the isotope U_{92}^{235} .



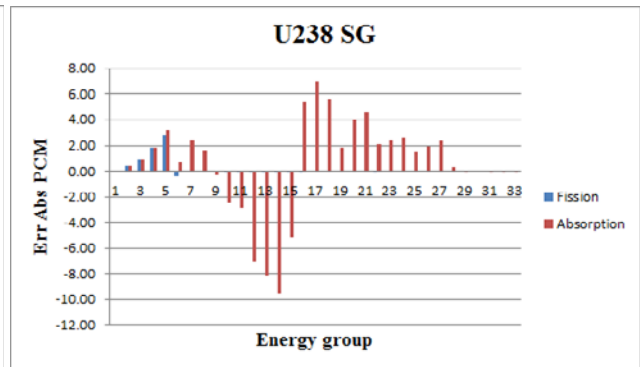
(a) External radial zone, fissile layer



(b) Internal radial zone, fissile layer



(c) External radial zone, fertile layer



(d) Internal radial zone, fertile layer

Figure 7: Absolute error over the 33 homogenization groups of the absorption and fission reactions rates for the isotope U_{92}^{238} .