Assembly Discontinuity Factors for the Neutron Diffusion Equation discretized with the Finite Volume Method. Application to BWR

A. Bernal[a], J.E. Roman[b], R. Miró[a] and G. Verdú[a]

[a] *Institute for Industrial, Radiophysical and Environmental Safety (ISIRYM), Universitat Politècnica de València, Camí de Vera s/n ,46022 Valencia, Spain;*

[b] *Department of Information Systems and Computation, Universitat Politècnica de València, Camí de Vera s/n ,46022 Valencia, Spain.*

abernal@iqn.upv.es, jroman@dsic.upv.es, rmiro@iqn.upv.es, gverdu@iqn.upv.es

Abstract

The neutron flux spatial distribution in Boiling Water Reactors (BWRs) can be calculated by means of the Neutron Diffusion Equation (NDE), which is a space- and time-dependent differential equation. In steady state conditions, the time derivative terms are zero and this equation is rewritten as an eigenvalue problem. In addition, the spatial partial derivatives terms are transformed into algebraic terms by discretizing the geometry and using numerical methods. As regards the geometrical discretization, BWRs are complex systems containing different components of different geometries and materials, but they are usually modelled as parallelepiped nodes each one containing only one homogenized material to simplify the solution of the NDE. There are several techniques to correct the homogenization in the node, but the most commonly used in BWRs is that based on Assembly Discontinuity Factors (ADFs). As regards numerical methods, the Finite Volume Method (FVM) is feasible and suitable to be applied to the NDE. In this paper, a FVM based on a polynomial expansion method has been used to obtain the matrices of the eigenvalue problem, assuring the accomplishment of the ADFs for a BWR. This eigenvalue problem has been solved by means of the SLEPc library.

Keywords: **assembly discontinuity factor; boiling water reactor; neutron diffusion equation; finite volume method; eigenvalue problem**

1. Introduction

Boiling Water Reactors (BWR) are multi-physics systems where the energy released by nuclear fission is transformed into thermal energy. Since the fission rate can be determined by means of the neutron flux, the determination of the neutron flux distribution inside nuclear reactors is crucial to obtain the power distribution in a nuclear reactor core.

The neutron flux spatial distribution in BWR can be calculated by means of stochastic or deterministic methods. The latter ones solve the integral-differential neutron transport equation and they require typically less computational resources than the former ones. The deterministic method most widely used in Reactor Physics Analysis is the neutron diffusion theory, which is a simplification of the neutron transport theory based on Fick's Law, as discussed by many authors, such as Stacey [1], due to its lowest computational resources. Nevertheless, the neutron diffusion theory applied to Nuclear Power Plants also has to cope with partial differential equations in heterogeneous media, and consequently numerical methods are required, since the analytical solution cannot be calculated in 3D. First, the time-dependent Neutron Diffusion Equation (NDE) is transformed into an eigenvalue problem to solve the neutron flux spatial distribution, as discussed in Section 2. Then, the spatial partial derivatives terms are transformed into algebraic terms by using numerical methods, as discussed also in Section 2.

Moreover, BWR are complex systems containing different components of different geometries and materials. Since the coefficients of the NDE, diffusion coefficients and cross-sections, depend on the materials, BWR are usually modelled as parallelepiped nodes each one containing only one homogenized material to simplify the solution of the NDE. It is usual that each of these nodes represents one fuel assembly, this homogenization should be based on assembly homogenization techniques [2]. However, using homogenization of cross sections and diffusion coefficients in BWR requires techniques to correct this homogenization in order to accurately solve the NDE [2]. One of these techniques most commonly used in BWR is that of the Assembly Discontinuity Factors (ADFs) [2, 3], which are the ratio of the surface flux obtained from heterogeneous composition to that obtained from homogeneous composition. By means of this technique, the continuity of the heterogeneous flux has to be accomplished, which implies the discontinuity of the homogeneous flux if ADF is different from one.

As far as numerical methods are concerned, the most popular numerical techniques used to solve the NDE give accurate results in structured meshes. However, the application of these methods in unstructured meshes dealing with complex geometries is not straightforward and it may cause problems of stability and convergence of the solution, as discussed by Hoffmann and Chiang [4]. In contrast, the Finite Volume Method (FVM) is easily applied to unstructured meshes. In addition, the application of the FVM to the NDE is feasible, as discussed by Bernal et al. [5, 6] and Theler [7].

In this paper, the FVM based on a polynomial expansion method [6] has been used to discretize the steady state of the NDE to obtain the matrices of the eigenvalue problem. This eigenvalue problem has been solved by means of the SLEPc library, which is appropriate to solve eigenvalue problems in which the associated matrices are sparse, such as those arising after the discretization of partial differential equations [8, 9].

This polynomial expansion method can accurately solve nuclear reactors in coarse meshes [6]. However, two modifications have been performed in order to solve the neutronics in a BWR. First, the accomplishment of the continuity of the heterogeneous flux at the faces of the fuel assemblies, which is imposed by means of the ADF. Second, the reduction of the number of equations used by this method, due to the high number of nodes used to model BWRs. Nevertheless, this reduction should not decrease the accuracy of the results.

The outline of the paper is as follows. Section 2 explains the method applying the ADF to the NDE equation discretized with the FVM. Section 3 describes the characteristics of the BWR used to assess the methodology and their results. Finally, Section 4 summarizes the conclusions about the results.

2. Material and methods

The time-dependent multigroup neutron diffusion approximation most widely used for commercial nuclear reactors is that of 2-energy groups [1], exhibited in Equation (1). In this equation, the neutron current can be calculated by using Fick’s Law, expressed in Equation (2).

$$\begin{aligned}
\frac{1}{v_1}\frac{d\phi_1(\vec{r},t)}{dt} &= -\nabla\left(\vec{J}_1(\vec{r},t)\right) - \left(\Sigma_{a,1}(\vec{r}) + \Sigma_{s,1\rightarrow 2}(\vec{r})\right)\,\phi_1(\vec{r},t) \\
&\quad + (1-\beta)\left(\upsilon\Sigma_{f,1}(\vec{r})\,\phi_1(\vec{r},t) + \upsilon\Sigma_{f,2}(\vec{r})\,\phi_2(\vec{r},t)\right) + \sum_i \lambda_i\, C_i \\
\frac{1}{v_2}\frac{d\phi_2(\vec{r},t)}{dt} &= -\nabla\left(\vec{J}_2(\vec{r},t)\right) - \Sigma_{a,2}(\vec{r})\,\phi_2(\vec{r},t) + \Sigma_{s,1\rightarrow 2}(\vec{r})\,\phi_1(\vec{r},t) \\
\frac{dC_i}{dt} &= \beta_i\left(\upsilon\Sigma_{f,1}(\vec{r})\,\phi_1(\vec{r},t) + \upsilon\Sigma_{f,2}(\vec{r})\,\phi_2(\vec{r},t)\right) - \lambda_i\, C_i \quad ; \quad i = 1, \dots, 6
\end{aligned} \tag{1}$$

$$\vec{J}_g(\vec{r},t) = -D_g\vec{\nabla}\phi_g(\vec{r},t) \tag{2}$$

In steady state conditions, the time-dependent terms are zero and Equation (1) is rewritten as the eigenvalue problem expressed by Equation (3). If one applies the FVM to Equation (3, 2), Equations (4, 5) are obtained [5].

$$\begin{aligned}
0 &= -\nabla\left(\vec{J}_1(\vec{r})\right) - \left(\Sigma_{a,1}(\vec{r}) + \Sigma_{s,1\rightarrow 2}(\vec{r})\right)\,\phi_1(\vec{r}) \\
&\quad + \frac{1}{\boldsymbol{k}}\left(\upsilon\Sigma_{f,1}(\vec{r})\,\phi_1(\vec{r},t) + \upsilon\Sigma_{f,2}(\vec{r})\,\phi_2(\vec{r})\right) \\
0 &= -\nabla\left(\vec{J}_2(\vec{r})\right) - \Sigma_{a,2}(\vec{r})\,\phi_2(\vec{r}) + \Sigma_{s,1\rightarrow 2}(\vec{r})\,\phi_1(\vec{r})
\end{aligned} \tag{3}$$

$$\begin{aligned}
\sum_j \left(S_j u_{i,j}\, J_{1,i,j}\right) + \left(\Sigma^{\mathrm{i}}_{a,1} + \Sigma^{\mathrm{i}}_{s,1\rightarrow 2}\right)\phi_{1,i}V_i &= \frac{1}{\boldsymbol{k}}\left(\upsilon\Sigma^{\mathrm{i}}_{f,1}\,\phi_{1,i} + \upsilon\Sigma^{\mathrm{i}}_{f,2}\,\phi_{2,i}\right)V_i \\
\sum_j \left(S_j u_{i,j}\, J_{2,i,j}\right) + \Sigma^{\mathrm{i}}_{a,2}\phi_{2,i}V_i - \Sigma^{\mathrm{i}}_{s,1\rightarrow 2}\,\phi_{1,i}V_i &= 0
\end{aligned} \tag{4}$$

$$J_{g,i,j} = -D^{\mathrm{i}}_g\vec{\nabla}\phi_{g,i,j} \tag{5}$$

Additional face equations are required to complete the set of equations. These equations are the boundary conditions for the boundary faces, and the heterogeneous neutron flux continuity and the current continuity for the inner faces. The boundary conditions most commonly used are the zero flux and the reflective flux, which are shown in Equations (6, 7), respectively. The neutron current continuity is shown in Equation (8), for the adjacent cells *i* and *l*. The heterogeneous neutron flux condition is calculated in Equation (9), for the adjacent cells *i* and *l*, by using the ADFs definition, which is expressed in Equation (10).

$$\phi_{g,i,j} = 0 \tag{6}$$

$$0 = \vec{\nabla}\phi_{g,i,j} = -J_{g,i,j}/D_g^{\mathrm{i}} \tag{7}$$

$$J_{g,i,j} = -J_{g,l,j} \tag{8}$$

$$0 = \phi_{g,i,j}^{\mathrm{het}} - \phi_{g,l,j}^{\mathrm{het}} = ADF_{g,i,j}\ \phi_{g,i,j} - ADF_{g,\mathrm{l},j}\ \phi_{g,l,j} \tag{9}$$

$$ADF_{g,i,j} = \phi_{g,i,j}^{\mathrm{het}}/\phi_{g,i,j} \tag{10}$$

In order to calculate the surface and volume averaged values of the neutron flux and current, a polynomial expansion method is used [6]. By means of this method [6], the neutron flux in each cell is expressed as a finite sum of polynomial terms $\left(p_t(x,y,z)\right)$, which are supposed to be known and their coefficients $\left(a_{g,i,t}\right)$ are determined by solving the eigenvalue problem. This polynomial expansion is shown in Equation (11). In the cited reference [6] it is demonstrated that the number of polynomial terms should be the number of faces of the cell plus one. Since the polynomial terms are known, one can easily calculate the volume and surface averaged values of the neutron flux and current with Equations (12-14).

$$\phi_{g,i}(x,y,z) = \sum_{t=1}^{Nfaces+1} a_{g,i,t}\ p_t(x,y,z) = \sum_{t=1}^{Nfaces+1} a_{g,i,t}\ x^{\alpha_t} y^{\beta_t} z^{\gamma_t} \tag{11}$$

$$\begin{aligned}\phi_{g,i} &= \frac{1}{V_i}\int_{V_i} \phi_{g,i}(x,y,z)\ dV = \sum_{t=1}^{Nfaces+1} a_{g,i,t}\ \frac{1}{V_i}\ \int_{V_i} p_t(x,y,z)\ dV \\ &= \sum_{t=1}^{Nfaces+1} a_{g,i,t}\ \overline{p_t}^{V_i}\end{aligned} \tag{12}$$

$$\begin{aligned}\phi_{g,i,j} &= \frac{1}{S_j}\int_{S_j} \phi_{g,i}(x,y,z)\ dS = \sum_{t=1}^{Nfaces+1} a_{g,i,t}\ \frac{1}{S_j}\ \int_{S_j} p_t(x,y,z)\ dS \\ &= \sum_{t=1}^{Nfaces+1} a_{g,i,t}\ \overline{p_t}^{S_{i,j}}\end{aligned} \tag{13}$$

$$
\begin{aligned}
\vec{\nabla}\phi_{g,i,j} &= \frac{1}{S_j}\int_{S_j} \vec{\nabla}\phi_{g,i,j}(x,y,z)\, dS = \\
&= \sum_{t=1}^{Nfaces+1} a_{g,i,t} \left(u_{ijx}\ \frac{1}{S_j} \int_{S_j} \frac{dp_t(x,y,z)}{dx}\, dS \right. \\
&\left. + u_{ijy}\ \frac{1}{S_j} \int_{S_j} \frac{dp_t(x,y,z)}{dy}\, dS + u_{ijz}\ \frac{1}{S_j} \int_{S_j} \frac{dp_t(x,y,z)}{dz}\, dS \right) = \\
&= \sum_{t=1}^{Nfaces+1} a_{g,i,t}\ \overline{\vec{\nabla} p_t}^{S_{i,j}}
\end{aligned}
\tag{14}
$$

The number of equations for each energy group with the method explained in [6] is the number of cells times the number of faces of each cell plus one. Thus, reactors with large number of nodes, such as BWRs, will be modelled with a large number of equations, and consequently it will require a high computational time. With the aim of accelerating the calculation, the number of equations will be reduced to the number of cells plus the total number of inner faces, which will reduce up to 60%, and the unknowns will be $\phi_{g,i}$ for each cell *i* and $J_{g,i,j}$ for each inner face *j*. The idea is to implicitly define the boundary conditions and the current continuity, and therefore the only equations will be the diffusion equations in each cell and the heterogeneous flux continuity in each inner face.

In order to define the current continuity implicitly, only one unknown current per each inner face *j* will be considered. For each inner face *j*, whose adjacent cells are *i* and *l*, the direction of its unknown current will be from cell *i* to cell *l*, thus $J_{g,i,j}$ will be the unknown value for inner face *j*, and $J_{g,l,j}$ will be substituted by $-J_{g,i,j}$, and consequently the current condition will be accomplished.

To change the unknowns, Equation (15) is considered from the polynomial expansion, with the variables defined in Equations (16, 17). By means of Equation (15), one can obtain the coefficients $a_{g,i,t}$ as a weighted sum of the volume averaged values and surface averaged values of the neutron flux or gradient of the neutron flux, by calculating the inverse of the matrix of Equation (15). The matrix of Equation (15) will be named I_i. Although the calculation of the inverse of a matrix is not recommended, in this case it is appropriate, because the largest matrix size will be the number of faces of one cell plus one.

$$
\begin{bmatrix}
\overline{p_1}^{V_i} & & \overline{p_{Nfaces+1}}^{V_i} \\
f_{i,1,1} & \cdots & f_{i,1,Nfaces+1} \\
\vdots & & \vdots \\
f_{i,Nfaces,1} & & f_{i,Nfaces,Nfaces+1}
\end{bmatrix}
\begin{bmatrix}
a_{g,i,1} \\ \vdots \\ a_{g,i,Nfaces+1}
\end{bmatrix}
=
\begin{bmatrix}
\phi_{g,i} \\ F_{g,i,1} \\ \vdots \\ F_{g,i,Nfaces}
\end{bmatrix}
\tag{15}
$$

$$
F_{g,i,j} = \begin{cases} \phi_{g,i,j} & , if\ face\ j\ is\ a\ boundary\ face\ of\ zero\ flux\ condition \\ -u_{i,j}\vec{\nabla}\phi_{g,i,j} & , the\ rest\ of\ cases \end{cases}
\tag{16}
$$

$$f_{i,j,t} = \begin{cases} \overline{p_t}^{S_{i,j}} & , if\ face\ j\ is\ a\ boundary\ face\ of\ zero\ flux\ condition \\ -u_{i,j}\,\overline{\vec{\nabla} p_t}^{S_{i,j}} & , the\ rest\ of\ cases \end{cases} \tag{17}$$

If one substitutes Equation (15) in Equation (13), the surface averaged value of the neutron flux is obtained in Equation (18), which uses the factor defined in Equation (19). Likewise, the surface averaged value of the gradient of the neutron flux is calculated in Equation (20), which uses the factor defined in Equation (21). Then, if one multiplies the diffusion coefficient of the cell *i* in Equations (18, 20), one obtains Equations (22, 23), in which the boundary conditions and the current continuity can be defined implicitly as expressed by Equation (24). Taking into account Equation (25) of the unitary vectors of the currents, the current continuity expressed in Equation (8) will be accomplished.

$$\phi_{g,i,j} = DXS_{i,j,1}\,\phi_{g,i} + \sum_{t=1}^{Nfaces} DXS_{i,\mathrm{j},t+1}\,F_{g,i,t} \tag{18}$$

$$DXS_{i,j,k} = \sum_{t=1}^{Nfaces+1} \overline{p_t}^{S_{i,j}}\, I_i^{-1}(t,k) \tag{19}$$

$$-\vec{\nabla}\phi_{g,i,j} = CUR_{i,j,1}\,\phi_{g,i} + \sum_{t=1}^{Nfaces} CUR_{i,\mathrm{j},t+1}\,F_{g,i,t} \tag{20}$$

$$CUR_{i,j,k} = -\sum_{t=1}^{Nfaces+1} \overline{\vec{\nabla} p_t}^{S_{i,j}}\, I_i^{-1}(t,k) \tag{21}$$

$$D_g^{\mathrm{i}}\phi_{g,i,j} = DXS_{i,j,1}\,D_g^{\mathrm{i}}\phi_{g,i} + \sum_{t=1}^{Nfaces} DXS_{i,\mathrm{j},t+1}\,D_g^{\mathrm{i}}F_{g,i,t} \tag{22}$$

$$J_{g,i,j} = -D_g^{\mathrm{i}}\vec{\nabla}\phi_{g,i,j} = CUR_{i,j,1}\,D_g^{\mathrm{i}}\phi_{g,i} + \sum_{t=1}^{Nfaces} CUR_{i,\mathrm{j},t+1}\,D_g^{\mathrm{i}}F_{g,i,t} \tag{23}$$

$$D_g^{\mathrm{i}}F_{g,i,j} = \begin{cases} 0 & , if\ face\ j\ is\ a\ boundary\ face \\ u_{i,j}\,J_{g,i,j} & , if\ face\ j\ is\ an\ inner\ face \end{cases} \tag{24}$$

$$u_{i,j} = -u_{l,j} = 1 \tag{25}$$

In addition, the set of Equations (4) can be transformed into the set of Equations (26), which will be the equations of the eigenvalue problem applied to each cell. In this equation, the eigenvector of each cell and group of energy will be $D_g^{\mathrm{i}} \phi_{g,i}$. As regards the neutron current for the boundary faces with zero flux condition, it will be calculated by means of Equation (23). Finally, the equations of the eigenvalue problem applied to each inner face is Equation (27), which is obtained by substituting Equation (22) in Equation (9); the eigenvector of each inner face and group will be the neutron current.

$$\begin{gathered}\sum_j \left(S_j u_{i,j} J_{1,i,j}\right) + V_i \frac{\left(\Sigma_{a,1}^{\mathrm{i}} + \Sigma_{s,1\to 2}^{\mathrm{i}}\right)}{D_1^{\mathrm{i}}} D_1^{\mathrm{i}} \phi_{1,i} = \frac{1}{\boldsymbol{k}} \left(V_i \frac{\upsilon \Sigma_{f,1}^{\mathrm{i}}}{D_1^{\mathrm{i}}} D_1^{\mathrm{i}} \phi_{1,i} + V_i \frac{\upsilon \Sigma_{f,2}^{\mathrm{i}}}{D_2^{\mathrm{i}}} D_2^{\mathrm{i}} \phi_{2,i} \right) \\ \sum_j \left(S_j u_{i,j} J_{2,i,j}\right) + V_i \frac{\Sigma_{a,2}^{\mathrm{i}}}{D_2^{\mathrm{i}}} D_2^{\mathrm{i}} \phi_{2,i} - V_i \frac{\Sigma_{s,1\to 2}^{\mathrm{i}}}{D_1^{\mathrm{i}}} D_1^{\mathrm{i}} \phi_{1,i} = 0\end{gathered} \tag{26}$$

$$\begin{aligned}0 = ADF_{g,i,j}\, \phi_{g,i,j} - ADF_{g,l,j}\, \phi_{g,l,j} \\ = \frac{ADF_{g,i,j}}{D_g^{\mathrm{i}}} \left(DXS_{i,j,1}\, D_g^{\mathrm{i}} \phi_{g,i} + \sum_{t=1}^{Nfaces} DXS_{i,\mathrm{j},t+1}\, D_g^{\mathrm{i}} F_{g,i,t} \right) \\ - \frac{ADF_{g,l,j}}{D_g^{\mathrm{l}}} \left(DXS_{l,j,1}\, D_g^{\mathrm{l}} \phi_{g,l} + \sum_{t=1}^{Nfaces} DXS_{l,\mathrm{j},t+1}\, D_g^{l} F_{g,l,t} \right)\end{aligned} \tag{27}$$

Therefore, the eigenvalue problem expressed in Equation (28) can be obtained from Equations (26, 27), whose eigenvector is defined in Equation (29). Moreover, this eigenvalue problem can be transformed into a standard eigenvalue problem as that expressed in Equation (30), whose size is half dimension the original one. This eigenvalue problem is solved by using the SLEPc library, with Krylov-Schur method. In this calculation, the eigenvalue problem is solved by the iterative process shown in Equation (31), where *x* is the initial guess of Φ_1 in the iterative method and *y, z, w, v, u* and *t* are vectors of the same dimension as Φ_1. It is important to highlight that the inverse of matrices L_{22} and L_{11} are not calculated, but the vectors *z* and *t* are determined by solving linear systems of equations with iterative methods as it is exhibited in Equations (32, 33), where vectors *y* and *u* are known in the iterative procedure of Equation (31). In particular, the linear systems of Equations (32, 33) are solved by using the GMRES method and preconditioning the matrices with the Additive Schwarz Method (ASM) [9-11].

$$\begin{pmatrix} \mathrm{L}_{11} & 0 \\ \mathrm{L}_{21} & \mathrm{L}_{22} \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix} = \frac{1}{\boldsymbol{k}} \begin{pmatrix} \mathrm{M}_{11} & \mathrm{M}_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix} \tag{28}$$

$$\Phi_g = \begin{pmatrix} D_g^1 \phi_{g,1} & \cdots & D_g^{\mathrm{Ncells}} \phi_{g,Ncells} & J_{g,1} & \cdots & J_{g,Ninnerfaces} \end{pmatrix}^T \tag{29}$$

$$\left(L_{11}^{-1} \left(\mathrm{M}_{11} - \mathrm{M}_{12}\, \mathrm{L}_{22}^{-1}\, \mathrm{L}_{21} \right) \right) \Phi_1 = \boldsymbol{k}\, \Phi_1 \tag{30}$$

$$\begin{aligned}\left(\mathrm{L}_{11}^{-1}\left(\mathrm{M}_{11} - \mathrm{M}_{12}\,\mathrm{L}_{22}^{-1}\,\mathrm{L}_{21}\right)\right) x &= \mathrm{L}_{11}^{-1}\left(\mathrm{M}_{11}\, x - \mathrm{M}_{12}\,\mathrm{L}_{22}^{-1}\,\mathrm{L}_{21} x\right) \\ &= \mathrm{L}_{11}^{-1}\left(v - \mathrm{M}_{12}\,\mathrm{L}_{22}^{-1}\, y\right) = \mathrm{L}_{11}^{-1}\left(v - \mathrm{M}_{12}\, z\right) \\ &= L_{11}^{-1}\left(v - w\right) = \mathrm{L}_{11}^{-1}\, u = t\end{aligned} \qquad (31)$$

$$L_{22}^{-1}\, y = z \rightarrow L_{22}\, z = y \qquad (32)$$

$$L_{11}^{-1}\; u = t \rightarrow L_{11}\, t = u \qquad (33)$$

3. Results

A commercial BWR reactor was simulated to validate the method, whose geometry is displayed in Figure 1. This reactor is composed of 624 fuel assemblies and 503 different compositions and has been modelled with 19980 nodes, which are cubes whose side length is 15.24 cm. Each node has different values of moderator density and fuel temperature. These values and the corresponding cross sections were obtained by applying the SIMTAB methodology [12] developed by the UPV together with Iberdrola Ingeniería y Construcción (Iberinco), for the following conditions: 92% of full power, 66.3% of flow and burnup of 11.196 Gwd/MT. Two configurations of the control rods were considered: the first one with all rods out (Case 1) and the second one displayed in Figures 2 and 3 (Case 2).

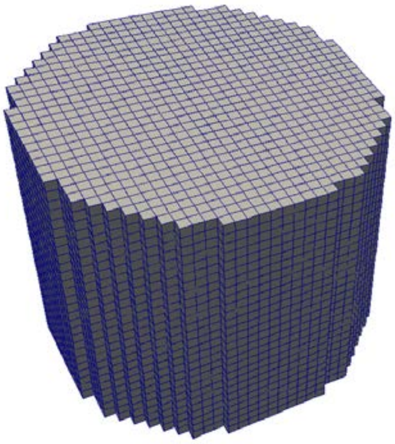

Figure 1. BWR

The number of cross sections for this reactor is too high for giving their values in this paper. Nevertheless, the authors tested a simple reactor in section 3.1, providing all its cross section and ADF, so one can reproduce the calculations of this simple case.

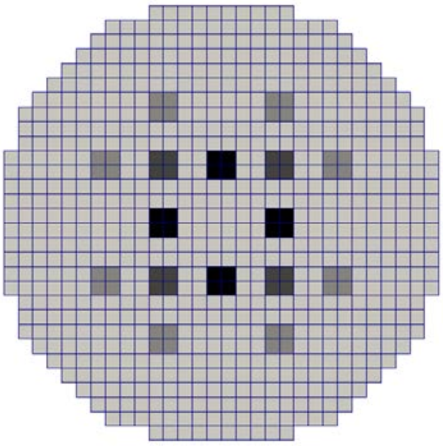

Figure 2. Radial position of the control rods

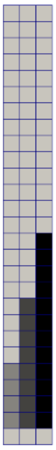

Figure 3. Axial position of the control rods

Both reactors are modelled and meshed by means of *Gmsh* code [13], which is a 3D finite element grid generator with a built-in CAD engine. Four meshes composed of hexahedra and one composed of tetrahedra were used to analyze the sensitivity of the proposed method.

As regards the polynomial terms, the number of polynomial terms for each cell has to be 7 for meshes composed of hexahedra and 5 for the mesh composed of tetrahedra, as explained in [6]. The sensitivity analysis of different polynomial sets was already done in [6], which demonstrated that there were only few sets of polynomial terms giving valid results, considering only a maximum of second order terms. In case of meshes composed of hexahedra there is only one set giving valid results: $1, x, y, z, x^2, y^2, z^2$. In case of meshes composed of tetrahedra, there are 3 sets of second order giving valid results. The first set (Set 1) is $1, x, y, z, x^2$; the second one (Set 2) is $1, x, y, z, y^2$ and the third one (Set 3) is $1, x, y, z, z^2$. In addition, the authors tested other combinations with terms of higher orders and they noted that the following set (Set 4) gives also valid and accurate results for meshes composed of tetrahedra [6]: $1, x, y, z, x^2y^2$.

The Power Errors (PE_i) and Eigenvalue Error (EE) are used to evaluate the results and are defined in Equations (34, 35). The SLEPc library can calculate several eigenfunctions, in our case 5 eigenvalues (the highest eigenvalues) and eigenvectors were computed. The eigenvalue is denoted as $\boldsymbol{k}$ in Equation (30). The relative modal power for each cell i (P_i) is defined in Equation (36). The results are normalized to attain Mean Power (MP) equals unity, which is defined in Equation (37), in which only cells i with not null power are considered. With the aim of reducing the extension of this paper, the Mean Power Error (MPE) will be used to assess the power results, and it is defined in Equation (38). With respect to the reference solution, the nodal diffusion codes PARCS [14] and VALKIN [15, 16] were used. On the one hand, PARCS is the reference code in neutron diffusion field, but it only calculates one eigenvalue. On the other hand, VALKIN uses a modal and nodal collocation method. Basically, in the nodal collocation method, Legendre's polynomials are used to expand the neutron flux in each node of the reactor. So, we used polynomial terms of third order. Moreover, it can calculate several eigenvalues because it is a modal method. It is important to highlight that these codes give accurate results, but they are not an analytical solution. The results are evaluated at the nodes of the reactors.

$$PE_i(\%) = \frac{\left|P_i - P_{i_{reference}}\right|}{P_{i_{reference}}} \cdot 100 \tag{34}$$

$$EE(pcm) = \frac{\boldsymbol{k} - \boldsymbol{k}_{reference}}{\boldsymbol{k}_{reference}} \cdot 10^5 \tag{35}$$

$$P_i = \left(\Sigma^{\mathrm{i}}_{f,1}\phi_{1,i} + \Sigma^{\mathrm{i}}_{f,2}\phi_{2,i}\right) \cdot constant \tag{36}$$

$$MP = \frac{\sum_{i=1}^{Ncells} |P_i|\, V_i}{\sum_{i=1}^{N} V_i} \tag{37}$$

$$MPE(\%) = \frac{\sum_{i=1}^{Ncells} PE_i(\%) |P_i|\, V_i}{\sum_{i=1}^{N} |P_i|\, V_i} \tag{38}$$

3.1. Simple test

The authors simulated a simple reactor composed of six materials, which are similar to the materials of the BWR used to validate this method. Figure 4 displays the geometry of this reactor, which consists of 1170 nodes, each one of the following dimensions: 20 cm x 20 cm x 20 cm. The reactor is divided into 10 axial levels, the bottom one composed of material 4 and the top one composed of material 5. The material distribution of the rest axial levels is shown in Figures 5 and 6. Both figures show the material distribution of a quarter of this reactor from the second axial level to the ninth. Figure 6 does not show material 3 with the aim of displaying the distribution of material 1. The cross sections and ADFs of the materials are exhibited in Table 1. In this simple reactor, the same ADF for the north, east, south and west faces is used.

This reactor was modelled and simulated with 5 different meshes. Mesh 1 is the structured mesh shown in Figure 5. Mesh 2 is obtained by generating 2x2x2 identical hexahedra in each hexahedron of Mesh 1. Mesh 3 is obtained by generating 3x3x3 identical hexahedra in each hexahedron of Mesh 1. Mesh 4 is obtained by generating 4x4x4 identical hexahedra in each hexahedron of Mesh 1. Mesh 5 is composed of 28080 tetrahedra (24 tetrahedra in each hexahedron of Mesh 1) and shown in Figure 7. The same polynomial sets as described in Section 3 were used. The boundary conditions are zero flux and 5 eigenvalues were calculated.

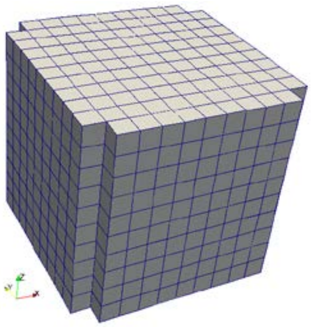

Figure 4. Simple reactor

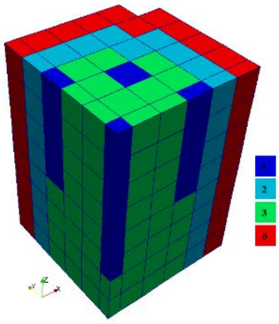

Figure 5. One quarter of the simple reactor

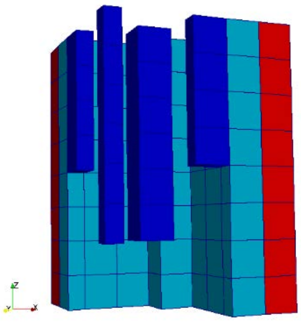

Figure 6. One quarter of the simple reactor, without material 3

Table 1. Cross sections and ADFs of the simple reactor

Material	D_1 (cm)	D_2 (cm)	$\Sigma_{a,1}$ (cm^{-1})	$\Sigma_{a,2}$ (cm^{-1})	$\upsilon\Sigma_{f,1}$ (cm^{-1})	$\upsilon\Sigma_{f,2}$ (cm^{-1})	$\Sigma_{s,1\rightarrow2}$ (cm^{-1})	ADF$_1$	ADF$_2$
1	1.384393	0.340606	0.006168	0.032008	0.002802	0.032638	0.021623	0.906938	1.245334
2	1.403033	0.338023	0.007356	0.054724	0.004650	0.076756	0.019370	0.904180	1.509856
3	1.396900	0.337844	0.007383	0.061280	0.005254	0.075934	0.019822	0.915384	1.610692
4	1.230190	0.265137	0.001713	0.034154	0.0	0.0	0.028561	1.0	1.0
5	1.779660	0.373901	0.000735	0.013441	0.0	0.0	0.029508	1.0	1.0
6	1.519700	0.247519	0.000399	0.010641	0.0	0.0	0.044684	1.0	1.0

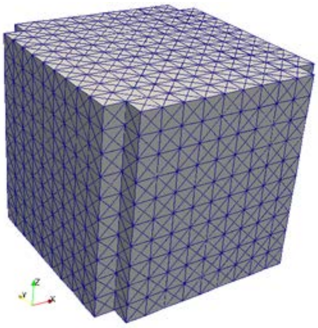

Figure 7. Mesh 5 of the simple reactor

This reactor was also modelled with Mesh 1 and simulated by PARCS and VALKIN. The eigenvalue calculated by PARCS is 1.041186 and those calculated by VALKIN are 1.040374, 1.010489, 1.010489, 0.990549 and 0.979611.

The computational time required for Meshes 1-4 was: 1, 3, 13 and 35 seconds, respectively. The time for Mesh 5 was 11 seconds and was the same for the four polynomial sets. As regards the results, Table 2 summarizes the results corresponding to the first eigenmode. In this table, $\mathcal{MEA}$ is the maximum power error per assembly. Regarding the other eigenmodes, the eigenvalue errors are exhibited in Table 3. Moreover, Tables 4 and 5 show the axial power errors corresponding to the first eigenmode, with respect to VALKIN and PARCS, respectively. Finally, Figures 8 and 9 display the assembly power errors corresponding to the first eigenmode and Mesh 3, with respect to VALKIN and PARCS, respectively.

Table 2. Results of the simple reactor and first eigenmode

	VALKIN			PARCS		
	EE (pcm)	**$\mathcal{MEA}$ (%)**	**MPE (%)**	**EE (pcm)**	**$\mathcal{MEA}$ (%)**	**MPE (%)**
Mesh 1	78.26	1.98	2.1	0.19	2.98	2.19
Mesh 2	-50.85	2.78	1.16	-128.82	5.16	2.09
Mesh 3	-14.56	0.89	0.43	-92.56	3.32	1.38
Mesh 4	11.19	0.3	0.2	-66.83	2.17	0.94
Mesh 5. Set 1	51.54	0.96	1.01	-26.52	2.17	1.26
Mesh 5. Set 2	51.6	0.96	1.01	-26.46	2.17	1.27
Mesh 5. Set 3	23.18	1.84	0.53	-54.85	3.83	1.1
Mesh 5. Set 4	49.16	1.18	0.97	-28.89	1.89	1.23

Table 3. Results of the simple reactor and eigenmodes 2-5

	EE-2 (pcm)	**EE-3 (pcm)**	**EE-4 (pcm)**	**EE-5 (pcm)**
Mesh 1	71.63	71.63	9.72	25.23
Mesh 2	-96.16	-96.23	-84.61	-136.12
Mesh 3	-29.58	-29.59	-26.7	-40.97
Mesh 4	14.48	14.5	10.03	21.24
Mesh 5. Set 1	97.02	76.1	114.17	129.45
Mesh 5. Set 2	96.98	76.36	114.32	129.61
Mesh 5. Set 3	67.44	67.21	38.84	125.07
Mesh 5. Set 4	72.31	71.75	114.94	106.08

It can be concluded that the finer the structured mesh, the lower the errors. Although Table 2 shows that the maximum error per assembly ($\mathcal{MEA}$) increases in Mesh 2, Table 4 probes that the axial power errors decrease in this mesh. As regards the polynomial sets used in Mesh 5, one can see in Table 2 that Sets 1 and 2 give better assembly results than Sets 3 and 4. However, the axial power errors obtained with Set 3 are lower than those obtained with Sets 1 and 2. Overall, the results obtained with structured meshes are better than those obtained with unstructured meshes. Meshes 3 and 4 have more cells than Mesh 5, so one may conclude this is the reason. Nevertheless, Mesh 2 and Mesh 5 have similar results, but he number of cells of Mesh 5 is 3 times greater than that of Mesh 2. Thus, one obtains better results not only by increasing the number of cells, but also by increasing the number of polynomial terms.

Table 4. Axial power errors (%) of the simple reactor with respect to VALKIN

Axial level	**Mesh 1**	**Mesh 2**	**Mesh 3**	**Mesh 4**	**Mesh 5. Set 1**	**Mesh 5. Set 2**	**Mesh 5. Set 3**	**Mesh 5. Set 4**
9	1.15	1.98	0.81	0.2	2.19	2.19	0.01	2.11
8	1.85	0.86	0.38	0.16	1.01	1.01	0.31	0.97
7	1.89	0.72	0.31	0.16	0.92	0.93	0.31	0.91
6	1.64	0.41	0.19	0.13	0.66	0.66	0.24	0.66
5	1.06	0.12	0.06	0.07	0.3	0.3	0.09	0.31
4	0.19	0.41	0.16	0.02	0.34	0.34	0.05	0.31
3	2.12	0.96	0.41	0.18	1.26	1.26	0.3	1.24
2	7.67	1.93	0.91	0.65	3.08	3.08	0.96	3.08

Table 5. Axial power errors (%) of the simple reactor with respect to PARCS

Axial level	**Mesh 1**	**Mesh 2**	**Mesh 3**	**Mesh 4**	**Mesh 5. Set 1**	**Mesh 5. Set 2**	**Mesh 5. Set 3**	**Mesh 5. Set 4**
9	2.13	2.95	1.79	1.18	3.16	3.16	1	3.08
8	2.08	1.09	0.61	0.4	1.24	1.24	0.55	1.2
7	1.95	0.78	0.37	0.22	0.98	0.98	0.37	0.97
6	1.57	0.34	0.12	0.05	0.58	0.58	0.17	0.58
5	0.89	0.06	0.12	0.1	0.12	0.12	0.09	0.13
4	0.41	0.63	0.39	0.25	0.56	0.56	0.27	0.54
3	2.33	1.17	0.62	0.39	1.47	1.47	0.5	1.45
2	7.19	1.47	0.45	0.19	2.62	2.62	0.5	2.62

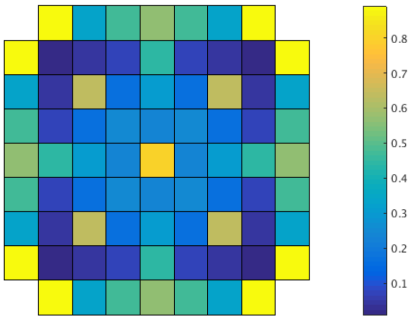

Figure 8. Assembly power errors (%) of the simple reactor with respect to VALKIN

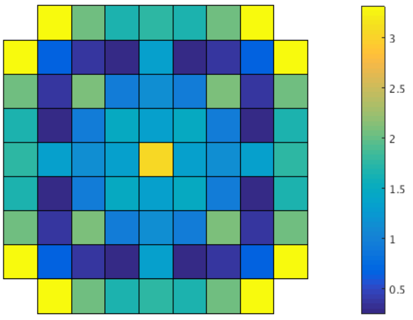

Figure 9. Assembly power errors (%) of the simple reactor with respect to PARCS

In addition, one can see appreciable differences of the results with respect to PARCS and VALKIN in Table 2 and in Figures 8 and 9. It seems that PARCS is less accurate, because the results of the finer structured meshes converges to those obtained with VALKIN.

3.2. Case 1: BWR with all rods out

A BWR was also modelled with 5 different meshes. The first mesh of hexahedra (Mesh 1) is the same as that of Figure 1. The second mesh of hexahedra (Mesh 2) is obtained by generating 2x2x2 identical hexahedra in each hexahedron of Mesh 1. The third mesh of hexahedra (Mesh 3) is obtained by generating 3x3x3 identical hexahedra in each hexahedron of Mesh 1. The last mesh of hexahedra (Mesh 4) is obtained by generating 4x4x4 identical hexahedra in each hexahedron of Mesh 1. The mesh composed of tetrahedra (Mesh 5) is shown in Figure 10.

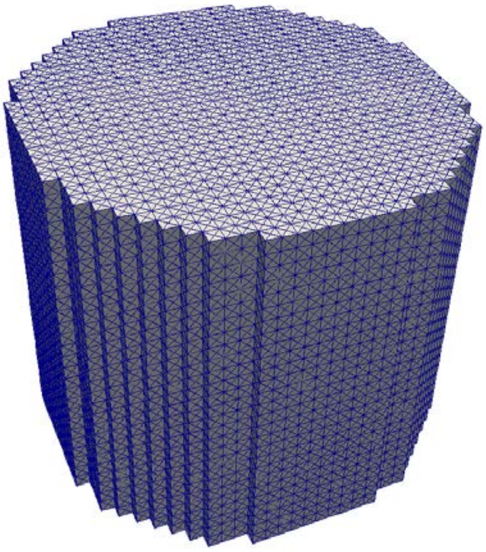

Figure 10. Mesh 5

The number of cells and faces of the different meshes are displayed in Table 6. The last column of this table shows the reduction of the size of the matrices in comparison with the matrices used in method explained in [6], due to the implicit definition of the current continuity and boundary conditions. Not only are these matrices of lower size, but they also have better quality, in spite of using the same polynomial expansion and equations. The quality of the matrices can be quantified by the condition number. Table 7 shows the condition number of both the coefficient matrix of the unpreconditioned linear system, $\mathrm{K}\left(\mathrm{L}_{\mathrm{gg}}\right)$, and the preconditioned one, $\mathrm{K}\left(M^{-1}\mathrm{L}_{\mathrm{gg}}\right)$, where M^{-1} represents the matrix of the preconditioner (single-domain ASM in this case). Moreover, the condition number was calculated as the ratio of the largest singular value to the smallest one, which requires high computational time for large matrices. Actually, if the matrix is large and ill-conditioned, the calculation of the condition number could not converge. Therefore, Table 7 shows only the condition number for Mesh 1, which is similar for the other meshes, but is quickly calculated.

The importance of the condition number arises in the possibility of using iterative methods for solving linear systems. In the method explained in [6], direct methods have been used owing to the high condition number, which requires higher computational time than iterative methods. However, iterative methods can be used by means of this method, and the calculation can be accelerated by using a preconditioner, which is Additive Schwarz Method (ASM) in this implementation.

Table 6. Geometrical elements of the meshes

Mesh	Number of cells	Number of inner faces	Number of boundary faces	(Size matrix)/(Size matrix in [6])
1	19980	57580	4720	0.55
2	159840	470080	18880	0.56
3	539460	1587140	42480	0.57
4	1278720	3798400	75520	0.57
5	479520	949600	18880	0.60

Table 7. Condition number of the matrices of Case 1 and Mesh 1

Matrix	$\mathrm{K}(\mathrm{L}_{\mathrm{gg}})$ with the method used in [6]	$\mathrm{K}(\mathrm{L}_{\mathrm{gg}})$ with this method	$\mathrm{K}(M^{-1}\mathrm{L}_{\mathrm{gg}})$ with this method
L_{11}	7654.40	107.80	1.56
L_{22}	1648.51	116.24	1.10

The eigenvalue calculated by PARCS is 1.021782 and those calculated by VALKIN are: 1.021729, 1.010299, 1.010038, 1.004688 and 0.998619. PARCS and VALKIN require a computational time about seconds and the computational time of this method is exhibited in Table 8. Table 9 summarizes the comparison of the results of the first eigenmode with respect to VALKIN and PARCS. In this table, the results converge to those obtained with VALKIN as the structured mesh becomes finer. However, the results do not converge to the results obtained by PARCS. Thus, it seems that PARCS is less accurate than VALKIN. As regards the results of the other eigenmodes, Table 10 exhibits the eigenvalue errors with respect to VALKIN. This table shows that the finer the structured mesh, the more accurate the results; but it also shows little differences of the polynomial sets for Mesh 5. Nevertheless, Tables 9 and 11 show appreciable differences of the power results among the polynomial sets. Table 11 shows the axial power errors of the first eigenmode. Particularly, Table 11 shows better axial power results for Set 3 because of the polynomial terms z^2; but Table 9 shows a higher assembly error, because Set 3 lacks the polynomial terms x^2 and y^2.

Table 8. Computational time (min:s) of Case 1

Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5. Set 1	Mesh 5. Set 2	Mesh 5. Set 3	Mesh 5. Set 4
0:17	3:15	10:53	34:27	16:38	17:21	13:12	16:1

Table 9. Results of Case 1 and first eigenmode

	VALKIN			PARCS		
	EE (pcm)	$\mathcal{MEA}$ (%)	MPE (%)	EE (pcm)	$\mathcal{MEA}$ (%)	MPE (%)
Mesh 1	4.6	8.45	1.55	-0.59	7.34	3.43
Mesh 2	-4.26	5.04	0.60	-9.44	5.38	3.08
Mesh 3	-2.04	2.03	0.24	-7.22	4.87	2.97
Mesh 4	-0.08	0.61	0.10	-5.27	4.61	2.96
Mesh 5. Set 1	6.9	4.29	0.79	1.71	4.97	3.01
Mesh 5. Set 2	6.87	4.21	0.79	1.69	4.98	3.01
Mesh 5. Set 3	2.34	7.33	0.91	-2.84	6.23	3.17
Mesh 5. Set 4	5.87	1.28	0.57	0.69	4.53	2.93

Table 10. Results of Case 1 and eigenmodes 2-5

	EE-2 (pcm)	EE-3 (pcm)	EE-4 (pcm)	EE-5 (pcm)
Mesh 1	11.14	10.75	9.55	11.17
Mesh 2	-5.60	-5.68	-11.43	-8.77
Mesh 3	-2.43	-2.47	-4.80	-3.43
Mesh 4	0.48	0.46	-1.05	-1.00
Mesh 5. Set 1	13.54	11.45	3.23	18.06
Mesh 5. Set 2	11.54	13.47	3.22	18.12
Mesh 5. Set 3	4.44	4.46	-8.01	4.86
Mesh 5. Set 4	11.42	11.31	6.39	17.91

Table 11. Axial power errors (%) of Case 1 with respect to VALKIN

Axial level	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5. Set 1	Mesh 5. Set 2	Mesh 5. Set 3	Mesh 5. Set 4
26	1.53	0.84	0.32	0.15	1.71	1.71	1.88	1.19
25	4.19	1.87	0.81	0.41	2.67	2.67	2.00	2.00
24	2.82	1.14	0.51	0.24	1.50	1.50	1.91	0.87
23	2.58	1.09	0.43	0.19	1.44	1.44	1.85	0.82
22	2.86	1.14	0.48	0.22	1.61	1.58	1.78	1.00
21	2.53	0.96	0.40	0.18	1.35	1.35	1.64	0.81
20	2.39	0.89	0.36	0.17	1.32	1.32	1.53	0.81
19	2.35	0.83	0.34	0.15	1.25	1.25	1.40	0.79
18	2.24	0.77	0.31	0.14	1.21	1.21	1.28	0.80
17	1.94	0.64	0.25	0.11	1.04	1.04	1.10	0.68
16	1.78	0.54	0.22	0.10	0.93	0.93	0.96	0.64
15	1.72	0.52	0.21	0.10	0.91	0.91	0.80	0.67
14	1.39	0.37	0.15	0.08	0.71	0.71	0.64	0.52
13	1.16	0.29	0.12	0.06	0.60	0.60	0.48	0.48
12	1.00	0.24	0.10	0.06	0.52	0.52	0.32	0.44
11	0.71	0.12	0.06	0.04	0.35	0.35	0.16	0.34
10	0.43	0.03	0.02	0.03	0.22	0.22	0.01	0.25
9	0.18	0.05	0.01	0.01	0.07	0.07	0.16	0.15
8	0.10	0.12	0.05	0.01	0.07	0.07	0.32	0.05
7	0.47	0.23	0.09	0.03	0.27	0.27	0.48	0.10
6	0.91	0.35	0.14	0.06	0.52	0.52	0.63	0.31
5	1.46	0.48	0.20	0.09	0.86	0.86	0.75	0.63
4	1.74	0.58	0.24	0.12	1.23	1.23	0.85	0.99
3	2.10	0.20	0.09	0.11	1.04	1.04	0.98	0.80
2	8.77	2.72	1.18	0.63	3.05	3.05	1.23	2.80

Finally, Figures 11 and 12 display the assembly power errors corresponding to the first eigenmode and Mesh 3, with respect to VALKIN and PARCS, respectively. In these figures, one can conclude that PARCS may not be very accurate, as the distibution errors are lower in the nodes next to the reflector.

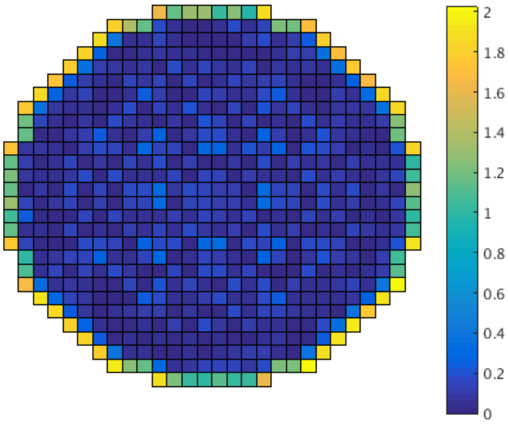

Figure 11. Assembly power errors (%) of Case 1 with respect to VALKIN

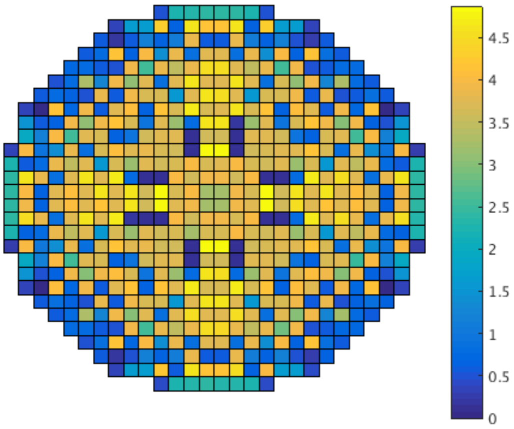

Figure 12. Assembly power errors (%) of Case 1 with respect to PARCS

3.3. Case 2: BWR with rods

The same results as those of Section 3.2 are shown in this section. The eigenvalue calculated by PARCS is 1.009466 and those calculated by VALKIN are: 1.009375, 0.998101, 0.997857, 0.995140 and 0.986519. The computational time for this case is summarized in Table 12. Table 13 summarizes the comparison of the results of the first eigenmode with respect to VALKIN and PARCS. In this table, the results converge to those obtained with VALKIN as the structured mesh becomes finer, but this convergence is worse than that of Section 3.2. In addition, PARCS seems to be less accurate than VALKIN in this case too. Regarding the results of the other eigenmodes, Table 14 displays the eigenvalue errors with respect to VALKIN. From this table, one concludes the same ideas as Case 1. On the other hand, Table 15 shows the axial power errors of the first eigenmode, which are lower than assembly errors of Table 13. Moreover, the highest assembly power errors for the finest structured meshes are not located next to the reflector, as opposed to Case 1. This is shown in Figure 13, which exhibits the assembly power errors for Mesh 3 and first eigenmode with respect to VALKIN. This figure shows that the highest assembly power errors are located near the rodded nodes, and the reason could be the high differences of the assembly discontinuity factors.

Table 12. Computational time (min:s) of Case 2

Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5. Set 1	Mesh 5. Set 2	Mesh 5. Set 3	Mesh 5. Set 4
0:10	2:49	11:56	37:49	17:35	19:51	19:40	16:37

Table 13. Results of Case 2 and first eigenmode

	VALKIN			PARCS		
	EE (pcm)	**$\mathcal{MEA}$ (%)**	**MPE (%)**	**EE (pcm)**	**$\mathcal{MEA}$ (%)**	**MPE (%)**
Mesh 1	4.90	10.83	2.87	4.09	9.51	3.73
Mesh 2	3.78	6.63	1.76	12.77	5.45	3.07
Mesh 3	0.56	5.05	1.37	8.44	4.80	2.87
Mesh 4	3.60	4.86	1.26	5.39	4.46	2.84
Mesh 5. Set 1	6.53	5.77	1.81	2.47	4.99	3.02
Mesh 5. Set 2	6.52	5.76	1.85	2.47	5.03	3.02
Mesh 5. Set 3	2.43	9.16	2.35	6.56	7.77	3.36
Mesh 5. Set 4	7.28	4.63	1.46	1.71	4.44	2.86

Table 14. Results of Case 2 and eigenmodes 2-5

	EE-2 (pcm)	**EE-3 (pcm)**	**EE-4 (pcm)**	**EE-5 (pcm)**
Mesh 1	31.97	17.04	30.95	30.89
Mesh 2	6.00	7.52	5.12	1.06
Mesh 3	10.00	3.52	6.31	4.62
Mesh 4	13.82	0.31	8.56	9.93
Mesh 5. Set 1	24.72	11.72	24.16	27.78
Mesh 5. Set 2	25.05	11.25	24.03	27.79
Mesh 5. Set 3	19.91	6.71	21.52	18.52
Mesh 5. Set 4	23.40	9.89	20.39	26.55

Table 15. Axial power errors (%) of Case 2 with respect to VALKIN

Axial level	**Mesh 1**	**Mesh 2**	**Mesh 3**	**Mesh 4**	**Mesh 5. Set 1**	**Mesh 5. Set 2**	**Mesh 5. Set 3**	**Mesh 5. Set 4**
26	1.64	1.10	0.67	0.55	1.82	1.82	2.36	1.16
25	4.42	2.18	1.18	0.78	2.69	2.69	2.53	1.99
24	3.05	1.45	0.84	0.62	1.53	1.53	2.42	0.84
23	2.84	1.40	0.82	0.60	1.49	1.49	2.36	0.82
22	3.11	1.47	0.85	0.62	1.65	1.63	2.29	1.00
21	2.79	1.28	0.75	0.57	1.40	1.40	2.17	0.81
20	2.65	1.21	0.72	0.54	1.36	1.36	2.04	0.81
19	2.58	1.13	0.68	0.52	1.30	1.29	1.90	0.79
18	2.43	1.05	0.62	0.48	1.22	1.22	1.72	0.77
17	2.08	0.87	0.53	0.42	1.02	1.02	1.51	0.63
16	1.82	0.73	0.45	0.36	0.87	0.87	1.27	0.54
15	1.61	0.63	0.38	0.30	0.78	0.78	1.02	0.53
14	1.05	0.36	0.23	0.20	0.45	0.45	0.70	0.28
13	0.65	0.19	0.12	0.10	0.31	0.31	0.34	0.25
12	0.22	0.03	0.02	0.02	0.06	0.06	0.06	0.10
11	0.39	0.30	0.18	0.13	0.27	0.27	0.50	0.12
10	1.06	0.58	0.35	0.26	0.61	0.61	0.95	0.33
9	1.66	0.84	0.50	0.38	0.86	0.86	1.43	0.46
8	2.34	1.12	0.68	0.52	1.23	1.23	1.94	0.69
7	3.17	1.44	0.86	0.65	1.63	1.63	2.46	0.96
6	4.11	1.82	1.07	0.80	2.13	2.13	2.98	1.33
5	5.06	2.14	1.25	0.95	2.53	2.52	3.51	1.60
4	5.74	2.44	1.42	1.09	3.11	3.10	3.94	2.09
3	6.33	2.18	1.36	1.14	3.07	3.07	4.28	2.00
2	13.18	4.73	2.46	1.66	5.04	5.04	4.67	3.94

In this case, PARCS also shows high differences with respect to VALKIN. Figure 14 displays the assembly power errors of Mesh 3 and the first eigenmode with respect to PARCS, which are completely different from those of Figure 13.

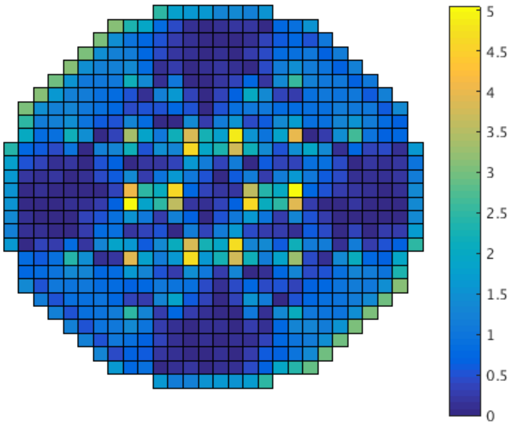

Figure 13. Assembly power errors (%) of Case 2 with respect to VALKIN

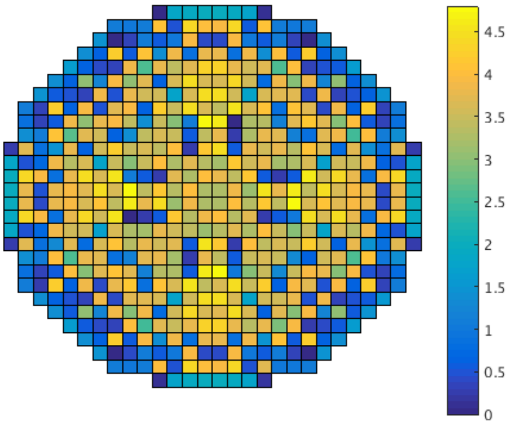

Figure 14. Assembly power errors (%) of Case 2 with respect to PARCS

4. Conclusions

A finite volume method has been developed to solve the eigenvalue problem of the neutron diffusion equation, assuring the continuity of the heterogeneous flux by using ADF.

This method is based on a polynomial expansion method of the neutron flux in each cell of the discretized geometry. This expansion is a finite sum of basic polynomial terms, up to the number of faces of each cell plus one. The final eigenvalue problem assures the accomplishment of the neutron diffusion equation, current continuity, heterogeneous flux continuity and boundary conditions. However, the current continuity and boundary conditions are defined implicitly to reduce the size of the matrices.

The method has been applied to a BWR to evaluate its capabilities. Two cases have been simulated, without and with control rods. Different meshes composed of hexahedra and tetrahedra were used to discretize the simulated BWR. The case without control rods gives excellent results, even with the coarsest mesh. The case with control rods gives good results for Mesh 4, but there are appreciable differences in the nodes next to the control rods.

As regards the future work, another polynomial expansions and more terms will be included by adding more equations. The parallelization of both geometry and solver will be performed. With respect to additional nuclear applications, the following step will be the multigroup formulation and the transitory state.

Acknowledgements

This work has been partially supported by the Spanish Ministerio de Eduación Cultura y Deporte under the grant FPU13/01009, the Spanish Ministerio de Ciencia e Innovación under projects ENE2014-59442-P, the Spanish Ministerio de Economía y Competitividad and the European Fondo Europeo de Desarrollo Regional (FEDER) under project ENE2015-68353-P (MINECO/FEDER), the Generalitat Valenciana under projects PROMETEOII/2014/008, the Universitat Politècnica de València under project UPPTE/2012/118, and the Spanish Ministerio de Economía y Competitividad under the project TIN2013-41049-P.

Nomenclature

ϕ_g: Neutron flux of the g energy group

$\vec{J}_g$: Neutron current of the g energy group

v_g: Neutron velocity of the g energy group

D_g: Neutron diffusion coefficient of the g energy group

$\Sigma_{a,g}$: Absorption macroscopic cross-section of the g energy group

$\Sigma_{s,1\to 2}$: Scattering macroscopic cross-section from the first to the second energy group

$\upsilon\Sigma_{f,g}$: Nu-fission macroscopic cross-section of the g energy group

β_i: Fraction of delayed neutrons of the precursors of group *i*

β: Total fraction of delayed neutrons

λ_i: Decay constant of the precursors of group *i*

C_i: Concentration of delayed neutrons precursors of group *i*

S_j: Area of the face *j*

V_i: Volume of the cell *i*

D_g^i: Neutron diffusion coefficient of the g energy group for the cell *i*

$\phi_{g,i}$: Neutron flux of the g energy group for the cell *i*

$\phi_{g,i,j}$: Neutron flux of the g energy group for the face *j*, corresponding to the cell *i*

$\phi_{g,i,j}^{\mathrm{het}}$: Heterogeneous neutron flux of the g energy group for the face *j*, corresponding to the cell *i*

$ADF_{g,i,j}$: Assembly Discontinuity Factor of the g energy group for the face *j*, corresponding to the cell *i*

$\Sigma_{a,g}^{\mathrm{i}}$: Absorption macroscopic cross-section of the g energy group for the cell *i*

$\Sigma^{\mathrm{i}}_{s,1\rightarrow 2}$: Scattering macroscopic cross-section from the first to the second energy group for the cell *i*

$\boldsymbol{k}$: Eigenvalue

$\upsilon\Sigma^{\mathrm{i}}_{f,g}$: Nu-fission macroscopic cross-section of the *g* energy group for the cell *i*

$Nfaces$: Number of faces of each cell

$a_{g,i,t}$: Coefficient multiplying the term *t* of the polynomial expansion of the flux of *g* energy group for the cell *i*

$p_t(x,y,z)$: Term *t* of the polynomial expansion of the flux

$\overline{p_t}^{V_i}$: Volume averaged value of the term *t* of the polynomial expansion of the flux for the cell *i*

$\overline{p_t}^{S_{i,j}}$: Surface averaged value of the term *t* of the polynomial expansion of the flux for the cell *i* and its face *j*

$\overline{\vec{\nabla} p_t}^{S_{i,j}}$: Surface averaged value of the gradient of the term *t* of the polynomial expansion of the flux for the cell *i* and its face *j*

u_{ijx} : X-component of the unit vector which is normal to face *j* and in the outgoing direction of cell *i*

u_{ijy} : Y-component of the unit vector which is normal to face *j* and in the outgoing direction of cell *i*

u_{ijz} : Z-component of the unit vector which is normal to face *j* and in the outgoing direction of cell *i*

$u_{i,j}$: Its value is +1 or -1 depending on the normal to face *j* and the outgoing direction of cell *i*

References

[1] Stacey WM. Nuclear Reactor Physics. New York: John Wiley & Sons; 2001.

[2] Smith, KS (1986). Assembly homogenization techniques for light water reactor analysis. Progress in Nuclear Energy, 17(3), 303-335.

[3] Verdú G, Ginestar D, Miró R, Vidal V. Using the Jacobi-Davidson method to obtain the dominant lambda modes of a nuclear power reactor. Annals of Nuclear Energy, 32, pp. 1274-1296 (23), 2005.

[4] Hoffmann KA, Chiang ST. Computational Fluid Dynamics, vol. 2. Wichita: Engineering Education System; 2000.

[5] Bernal A, Miró R, Ginestar D, Verdú G. Resolution of the Generalized Eigenvalue Problem in the Neutron Diffusion Equation Discretized by the Finite Volume Method. Abstract and Applied Analysis. 2014 January; 2014:15.

[6] Bernal A, Roman JE, Miró R, Ginestar D, Verdú G. Development of a Finite Volume inter-cell Polynomial Expansion method for the Neutron Diffusion Equation. Journal of Nuclear Science and Technology. 2015. doi: 10.1080/00223131.2015.1102661

[7] Theler G. Unstructured Grids and the Multigroup Neutron Diffusion Equation. Science and Technology of Nuclear Installations. 2013 July; 2013: 26 pages.

[8] Hernández V, Román JE, Vidal V. SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems. ACM Transactions on Mathematical Software. 2005 September; 31:351–362.

[9] Román JE, Campos C, Romero E, Tomas A. SLEPc Users Manual. Tech. Rep. DSIC-II/24/02 - Revision 3.6, Universitat Politècnica de València, 2015.

[10] Balay S et al. PETSc Users Manual. ANL-95/11 - Revision 3.6, Argonne National Laboratory, 2015.

[11] Xiao-Chuan Cai, Marcus Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM Journal on Scientific Computing, 21(2), 792-797 (1999).

[12] Miró R, Verdú G, Barrachina T, Roselló O. Parameterization of nuclear cross-sections for coupled neutronic-thermalhydraulic codes. Proc. PHYSOR-2006; 2006 Sep 10-14; Vancouver (Canada).

[13] Geuzaine C, Remacle J-F. Gmsh: a 3-D finite element mesh generator with built.in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering. 2009 May; 79 1309:1311.

[14] Downar TJ, Xu Y, Kozlowski T, Carlson D. PARCS v2.7. U.S. NRC Core Neutronics Simulator. User Manual. 2006 August; Rockville, Md.

[15] Verdú G, Ginestar D, Vidal V, Muñoz-Cobo JL. 3D λ-modes of the neutron-diffusion equation. Annals of Nuclear Energy. 1994 July; 21 405-421.

[16] Miró R, Ginestar D, Verdú G, Hennig D. A nodal modal method for the neutron diffusion equation. Application to BWR instabilities analysis. Annals of Nuclear Energy. 2002 July; 29 1171:1194.