
ARTICLE

Development of a Finite Volume inter-cell Polynomial Expansion method for the Neutron Diffusion Equation

Álvaro Bernal^{a*}, José E. Román^b, Rafael Miró^a, Damián Ginestar^c and Gumersindo 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.*

^c*Institute for Multidisciplinary Mathematics, Universitat Politècnica de València, Camí de Vera s/n ,46022 Valencia, Spain.*

Acknowledgements

This work has been partially supported by the Spanish Ministerio de Educación Cultura y Deporte under the grant FPU13/01009, the Spanish Ministerio de Ciencia e Innovación under projects ENE2014-59442-P and ENE2012-34585, the Generalitat Valenciana under projects PROMETEOII/2014/008, the Universitat Politècnica de València under project UPPTTE/2012/118, and the Spanish Ministerio de Economía y Competitividad under the project TIN2013-41049-P.

Heterogeneous nuclear reactors require numerical methods to solve the neutron diffusion equation (NDE) to obtain the neutron flux distribution inside them, by discretizing the heterogeneous geometry in a set of homogeneous regions. This discretization requires additional equations at the inner faces of two adjacent cells: neutron flux and current continuity, which imply an excess of equations. The Finite Volume Method (FVM) is suitable

*Corresponding author. Email: rmiro@iqn.upv.es

to be applied to NDE, because it can be easily applied to any mesh and it is typically used in the transport equations due to the conservation of the transported quantity within the volume. However, the gradient and face averaged values in the FVM are typically calculated as a function of the cell averaged values of adjacent cells. So, if the materials of the adjacent cells are different, the neutron current condition could not be accomplished. Therefore, a polynomial expansion of the neutron flux is developed in each cell for assuring the accomplishment of the flux and current continuity and calculating both analytically. In this polynomial expansion, the polynomial terms for each cell were assigned previously and the constant coefficients are determined by solving the eigenvalue problem with SLEPc. A sensitivity analysis for determining the best set of polynomial terms is performed.

neutron diffusion equation; finite volume method; polynomial expansion; steady-state

1. Introduction

The spatial distribution of the neutron flux in nuclear reactors is relevant to assure nuclear reactor safety since it is related with the power. The neutron transport equation could be used to determine it accurately, but it requires high computational resources. In contrast, the neutron diffusion equation, which is a simplification of the neutron transport equation using the Fick's Law [1], requires less computational resources and is suitable to be applied to LWR nuclear reactors.

The neutron diffusion equation contains spatial and time-dependent terms, so the spatial distribution is calculated by transforming it into an eigenvalue problem, explained in section 2.1. Moreover, this equation contains spatial partial derivatives terms and therefore, heterogeneous nuclear reactors require numerical methods to solve the neutron diffusion equation applied to them, by discretizing the heterogeneous geometry in a set of

homogeneous regions.

On the one hand, neutron diffusion equation is an approximation of the neutron transport equation. On the other hand, neutron calculations require a coupled thermalhydraulic-neutronic calculation. In fact, best estimate thermal-hydraulics calculations as Computational Fluid Dynamics (CFD) codes use the Finite Volume Method (FVM), because it can be easily applied to unstructured meshes and is typically used in the transport equations due to the conservation of the transported quantity within the volume [2]. Consequently, the FVM is suitable to be applied to the neutron diffusion equation [3, 4].

Furthermore, the neutron diffusion theory applied to discretized geometries requires additional equations at the inner faces of two adjacent cells: neutron flux and neutron current continuity, which imply an excess of equations. In addition, the neutron current is proportional to the neutron flux gradient, and the proportional constant depends on the material. Since the gradient in the FVM is typically calculated by using the cell-averaged values of the neighbouring cells [5], the gradient is a continuous function of the adjacent cells. Nevertheless, if the materials of the two adjacent cells are different, the neutron current condition will not be accomplished.

In this paper, a polynomial expansion of the neutron flux is developed in each cell, up to the same number of equations as unknowns, for calculating the gradient analytically. In this polynomial expansion, the polynomial terms for each cell were assigned previously and the constant coefficients are determined by solving the eigenvalue problem by means of SLEPc library, because it is appropriate for large and sparse matrices [6, 7]. A sensitivity analysis for determining the best set of polynomial terms is performed.

The outline of the paper is as follows. Section 2 explains the polynomial expansion method for the steady-state 2 energy-group neutron diffusion equation discretized by the Finite Volume Method. Sections 3 describes the reactors used and exhibits the results. Section 4 summarizes the conclusions about this work.

2. Materials and Methods

2.1. Steady-state 2 energy-group neutron diffusion equation discretized by the Finite Volume Method

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.

$$\begin{aligned} \frac{1}{v_1} \frac{d\phi_1(\vec{r}, t)}{dt} = & -\nabla \left(-D_1 \vec{\nabla} \phi_1(\vec{r}, t) \right) - \left(\Sigma_{a,1}(\vec{r}) + \Sigma_{s,1 \rightarrow 2}(\vec{r}) \right) \phi_1(\vec{r}, t) \\ & + (1 - \beta) \left(v \Sigma_{f,1}(\vec{r}) \phi_1(\vec{r}, t) + v \Sigma_{f,2}(\vec{r}) \phi_2(\vec{r}, t) \right) + \sum_i \lambda_i C_i \end{aligned} \quad (1)$$

$$\frac{1}{v_2} \frac{d\phi_2(\vec{r}, t)}{dt} = -\nabla \left(-D_2 \vec{\nabla} \phi_2(\vec{r}, t) \right) - \Sigma_{a,2}(\vec{r}) \phi_2(\vec{r}, t) + \Sigma_{s,1 \rightarrow 2}(\vec{r}) \phi_1(\vec{r}, t)$$

$$\frac{dC_i}{dt} = \beta_i \left(v \Sigma_{f,1}(\vec{r}) \phi_1(\vec{r}, t) + v \Sigma_{f,2}(\vec{r}) \phi_2(\vec{r}, t) \right) - \lambda_i C_i \quad ; \quad i = 1, \dots, 6$$

The steady-state of Equation 1 is accomplished only for certain geometry and nuclear parameters, which are the coefficients of Equation 1. As a result, Equation 1 is transformed into the eigenvalue problem expressed by Equation 2, to attain the steady-state.

$$\begin{aligned} 0 = & -\nabla \left(-D_1 \vec{\nabla} \phi_1(\vec{r}, t) \right) - \left(\Sigma_{a,1}(\vec{r}) + \Sigma_{s,1 \rightarrow 2}(\vec{r}) \right) \phi_1(\vec{r}, t) \\ & + \frac{1}{k} \left(v \Sigma_{f,1}(\vec{r}) \phi_1(\vec{r}, t) + v \Sigma_{f,2}(\vec{r}) \phi_2(\vec{r}, t) \right) \end{aligned} \quad (2)$$

$$0 = -\nabla \left(-D_2 \vec{\nabla} \phi_2(\vec{r}, t) \right) - \Sigma_{a,2}(\vec{r}) \phi_2(\vec{r}, t) + \Sigma_{s,1 \rightarrow 2}(\vec{r}) \phi_1(\vec{r}, t)$$

If one applies the FVM to Equation 2, Equation 3 is obtained [3]. In this equation, the face-averaged values of the neutron flux gradient ($\vec{\nabla} \phi_{g,i,j}$) have to be determined. In the reference mentioned [3], $\vec{\nabla} \phi_{g,i,j}$ is calculated by means of *Arb* [5] algorithm, which calculates it as a weighted sum of the cell averaged values of the neutron flux of the neighbouring cells ($\phi_{g,n}$) as in Equation 4 [3, 5]. Nonetheless, it works well for fine meshes,

but it requires high computational time [3]. Besides, the neutron flux continuity is not applied in this method and the neutron current continuity is imposed by several approximations [3], since they imply an excess of equations in comparison with the unknowns, which are the cell values.

$$\sum_j \left(\frac{S_j}{V_i} u_j (-D_1^i \vec{\nu} \phi_{1,i,j}) \right) + (\Sigma_{a,1}^i + \Sigma_{s,1 \rightarrow 2}^i) \phi_{1,i} = \frac{1}{k} (v \Sigma_{f,1}^i \phi_{1,i} + v \Sigma_{f,2}^i \phi_{2,i})$$

$$\sum_j \left(\frac{S_j}{V_i} u_j (-D_2^i \vec{\nu} \phi_{2,i,j}) \right) + \Sigma_{a,2}^i \phi_{2,i} - \Sigma_{s,1 \rightarrow 2}^i \phi_{1,i} = 0 \quad (3)$$

$$\vec{\nu} \phi_{g,i,j} = \vec{\nu} \phi_{g,j} = \sum_{n \in \ell} k_{n,j}^{grad} \phi_{g,n} \quad (4)$$

2.2. Inter-cells polynomial expansion method

In this paper, a polynomial expansion of the neutron flux for each cell is proposed. Firstly, the number of terms of this expansion must equal the number of equations for each cell. **Figure 1** shows a geometry discretized into 2 cells, where the diffusion equations, boundary conditions, neutron flux continuity and current continuity are applied. In **Figure 1**, one could appreciate that the number of equations for each cell is the number of faces plus one, so this will be the number of terms of this expansion, which is exhibited in Equation 5.

< Figure 1 >

$$\phi_{g,i}(x, y, z) = \sum_{t=1}^{N_{faces}+1} a_{g,i,t} p_t(x, y, z) \quad (5)$$

In this expansion, each polynomial term $(p_t(x, y, z))$ is assumed to be known, and is

defined in Equation 6. In contrast, the coefficients of the expansion ($a_{g,i,t}$) are unknown and will be determined by solving the eigenvalue problem. Since there are infinite polynomial combinations, a sensitivity analysis was done to determine the best polynomial set.

$$p_t(x, y, z) = x^{\alpha_t} y^{\beta_t} z^{\gamma_t} \quad (6)$$

The volume and surface averaged values of the neutron flux are easily calculated because the polynomial terms are known and depend exclusively on the geometry of cells and faces and number of faces, as Equations 7 and 8 show. In addition, the neutron flux gradient can be easily and analytically calculated with Equations 9-12. Then, surface averaged values of the neutron flux gradient are determined by means of Equation 13.

$$\begin{aligned} \phi_{g,i} &= \frac{1}{V_i} \int_{V_i} \phi_{g,i}(x, y, z) dV = \sum_{t=1}^{N_{faces+1}} a_{g,i,t} \frac{1}{V_i} \int_{V_i} p_t(x, y, z) dV \\ &= \sum_{t=1}^{N_{faces+1}} a_{g,i,t} \bar{p}_t^{V_i} \end{aligned} \quad (7)$$

$$\begin{aligned} \phi_{g,i,j} &= \frac{1}{S_j} \int_{S_j} \phi_{g,i}(x, y, z) dS = \sum_{t=1}^{N_{faces+1}} a_{g,i,t} \frac{1}{S_j} \int_{S_j} p_t(x, y, z) dS \\ &= \sum_{t=1}^{N_{faces+1}} a_{g,i,t} \bar{p}_t^{S_{i,j}} \end{aligned} \quad (8)$$

$$\begin{aligned} \vec{\nabla} \phi_{g,i,j}(x, y, z) &= \frac{d\phi_{g,i}(x, y, z)}{dx} u_{ijx} + \frac{d\phi_{g,i}(x, y, z)}{dy} u_{ijy} + \frac{d\phi_{g,i}(x, y, z)}{dz} u_{ijz} = \\ &= \sum_{t=1}^{N_{faces+1}} a_{g,i,t} \left(\frac{dp_t(x, y, z)}{dx} u_{ijx} + \frac{dp_t(x, y, z)}{dy} u_{ijy} \right. \\ &\quad \left. + \frac{dp_t(x, y, z)}{dz} u_{ijz} \right) \end{aligned} \quad (9)$$

$$\frac{dp_t(x, y, z)}{dx} = \alpha_t x^{\alpha_t-1} y^{\beta_t} z^{\gamma_t} \quad (10)$$

$$\frac{dp_t(x, y, z)}{dy} = \beta_t x^{\alpha_t} y^{\beta_t-1} z^{\gamma_t} \quad (11)$$

$$\frac{dp_t(x, y, z)}{dz} = \gamma_t x^{\alpha_t} y^{\beta_t} z^{\gamma_t-1} \quad (12)$$

$$\begin{aligned} \vec{V}\phi_{g,i,j} &= \frac{1}{S_j} \int_{S_j} \vec{V}\phi_{g,i,j}(x, y, z) dS = \\ &= \sum_{t=1}^{N_{faces}+1} a_{g,i,t} \left(u_{ijx} \frac{1}{S_j} \int_{S_j} \frac{dp_t(x, y, z)}{dx} dS \right. \\ &\quad \left. + u_{ijy} \frac{1}{S_j} \int_{S_j} \frac{dp_t(x, y, z)}{dy} dS + u_{ijz} \frac{1}{S_j} \int_{S_j} \frac{dp_t(x, y, z)}{dz} dS \right) \quad (13) \\ &= \sum_{t=1}^{N_{faces}+1} a_{g,i,t} \left(u_{ijx} \frac{\overline{dp_t}^{S_{i,j}}}{dx} + u_{ijy} \frac{\overline{dp_t}^{S_{i,j}}}{dy} + u_{ijz} \frac{\overline{dp_t}^{S_{i,j}}}{dz} \right) \\ &= \sum_{t=1}^{N_{faces}+1} a_{g,i,t} \overline{Vp_t}^{S_{i,j}} \end{aligned}$$

If one substitutes these averaged values in Equation 3, Equation 14 is obtained, which is the discretized neutron diffusion equation. Finally, the face equations are required to complete the system of equations. These equations are the boundary conditions, the neutron flux and current continuity. The boundary conditions most commonly used are the zero flux and the reflective flux, which are discretized in Equations 15 and 16 respectively. The neutron flux and current continuity are discretized in Equations 17 and 18 respectively.

$$\begin{aligned}
& \sum_{t=1}^{Nfaces+1} a_{1,i,t} \left(\frac{-D_1^i}{V_i} \sum_j (S_j \overline{\vec{v}} p_t^{S_{i,j}}) + \bar{p}_t^{V_i} (\Sigma_{a,1}^i + \Sigma_{s,1 \rightarrow 2}^i) \right) \\
&= \frac{1}{\mathbf{k}} \left(\sum_{t=1}^{Nfaces+1} a_{1,i,t} \bar{p}_t^{V_i} \nu \Sigma_{f,1}^i + \sum_{t=1}^{Nfaces+1} a_{2,i,t} \bar{p}_t^{V_i} \nu \Sigma_{f,2}^i \right) \quad (14)
\end{aligned}$$

$$\sum_{t=1}^{Nfaces+1} a_{2,i,t} \left(\frac{-D_2^i}{V_i} \sum_j (S_j \overline{\vec{v}} p_t^{S_{i,j}}) + \bar{p}_t^{V_i} \Sigma_{a,2}^i \right) - \sum_{t=1}^{Nfaces+1} a_{1,i,t} \bar{p}_t^{V_i} \Sigma_{s,1 \rightarrow 2}^i = 0$$

$$\phi_{g,i,j} = 0 = \sum_{t=1}^{Nfaces+1} a_{g,i,t} \bar{p}_t^{S_{i,j}} \quad (15)$$

$$\vec{v} \phi_{g,i,j} = 0 = \sum_{t=1}^{Nfaces+1} a_{g,i,t} \overline{\vec{v}} p_t^{S_{i,j}} \quad (16)$$

$$\phi_{g,i,j} - \phi_{g,i+1,j} = 0 = \sum_{t=1}^{Nfaces+1} (\bar{p}_t^{S_{i,j}} a_{g,i,t} - \bar{p}_t^{S_{i+1,j}} a_{g,i+1,t}) \quad (17)$$

$$\begin{aligned}
& (-D_g^i \vec{v} \phi_{g,i,j}) + (-D_g^{i+1} \vec{v} \phi_{g,i+1,j}) = 0 \\
&= \sum_{t=1}^{Nfaces+1} \left(-D_g^i a_{g,i,t} \overline{\vec{v}} p_t^{S_{i,j}} - D_g^{i+1} a_{g,i+1,t} \overline{\vec{v}} p_t^{S_{i+1,j}} \right) \quad (18)
\end{aligned}$$

2.3. Eigenvalue problem

The generalized eigenvalue problem of Equation 19 is obtained by applying Equations 14-18 to each cell and its faces of the discretized geometry.

$$\begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix} = \frac{1}{\mathbf{k}} \begin{pmatrix} M_{11} & M_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix} \quad (19)$$

L_{11} : $\left(\frac{-D_1^i}{V_i} \sum_j \left(S_j \overline{\vec{v}} p_t^{S_{i,j}} \right) + \bar{p}_t^{V_i} (\Sigma_{a,1}^i + \Sigma_{s,1 \rightarrow 2}^i) \right)$ of Equation 14 and terms of Equations 15-18.

L_{21} : $\bar{p}_t^{V_i} \Sigma_{s,1 \rightarrow 2}^i$ of Equation 14.

L_{22} : $\left(\frac{-D_2^i}{V_i} \sum_j \left(S_j \overline{\vec{v}} p_t^{S_{i,j}} \right) + \bar{p}_t^{V_i} \Sigma_{a,2}^i \right)$ of Equation 14 and terms of Equations 15-18.

M_{11} : $\bar{p}_t^{V_i} \nu \Sigma_{f,1}^i$ of Equation 14.

M_{12} : $\bar{p}_t^{V_i} \nu \Sigma_{f,2}^i$ of Equation 14.

The neutron flux for each energy group (Φ_g) is composed of the neutron flux for each cell ($\Phi_{g,i}$), which consists of each polynomial expansion coefficient ($a_{g,i,t}$). These definitions are shown in Equations 20 and 21.

$$\Phi_g = \begin{pmatrix} \Phi_{g,1} \\ \vdots \\ \Phi_{g,N} \end{pmatrix} \quad (20)$$

$$\Phi_{g,i} = \begin{pmatrix} a_{g,i,1} \\ \vdots \\ a_{g,i,N_{faces}+1} \end{pmatrix} \quad (21)$$

The dimension of the generalized eigenvalue problem of Equation 19 can be reduced by a factor of two. One can appreciate from Equation 19 the relationship between neutron flux for each energy group, expressed in Equation 22. If one uses Equation 22, Equation 19 is transformed into Equation 23, which represents an eigenvalue problem with half dimension of the original one. This equation is solved iteratively by means of the procedure exhibited in Equation 24, 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 of 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 direct methods as it is exhibited in Equations 25 and 26, where vectors y and u are known in the iterative procedure of Equation 24.

$$\Phi_2 = -L_{22}^{-1} L_{21} \Phi_1 \quad (22)$$

$$(L_{11}^{-1} (M_{11} - M_{12} L_{22}^{-1} L_{21})) \Phi_1 = k \Phi_1 \quad (23)$$

$$\begin{aligned} (L_{11}^{-1} (M_{11} - M_{12} L_{22}^{-1} L_{21})) x &= L_{11}^{-1} (M_{11} x - M_{12} L_{22}^{-1} L_{21} x) \\ &= L_{11}^{-1} (v - M_{12} L_{22}^{-1} y) = L_{11}^{-1} (v - M_{12} z) = L_{11}^{-1} (v - w) \\ &= L_{11}^{-1} (u) = t \end{aligned} \quad (24)$$

$$L_{22}^{-1} y = z \rightarrow L_{22} z = y \quad (25)$$

$$L_{11}^{-1} u = t \rightarrow L_{11} t = u \quad (26)$$

3. Results and Discussions

Homogeneous and heterogeneous reactors are simulated to assess the method. The reactors are modelled and meshed by means of *Gmsh* code [8], which is a 3D finite element grid generator with a build-in CAD engine. Different structured and unstructured meshes were used to analyse its sensitivity.

The structured meshes are composed of hexahedra and the unstructured meshes consist of tetrahedra. As a result, the number of polynomial terms for each cell has to be 7 for structured meshes and 5 for unstructured meshes. A sensitivity analysis of different polynomial sets is done, but only a maximum of second order terms are considered, because there are infinite possible combinations for higher order expansions.

The power and eigenvalue errors are used to evaluate the results and are defined in Equations 27 and 28 respectively. SLEPc library can calculate several eigenvalues, in our case 5 eigenvalues (the highest eigenvalues) and eigenvectors were computed. The eigenvalue is k of Equation 23 and the eigenvector is the neutron flux for the first energy group ϕ_1 . The

neutron flux for the second energy group ϕ_2 is calculated with Equation 22. The modal power is a weighted sum of the ϕ_1 and ϕ_2 and it is defined in Equation 29. The results are normalized to attain mean power equals unity, which is defined in Equation 30. In this equation, only cells i with not null power are considered. With the aim of reducing the extension of this paper, the mean power error will be used to assess the power results, and it is defined in Equation 31.

$$Power\ error\ (\%) = \frac{|Power - Power_{reference}|}{Power_{reference}} * 100 \quad (27)$$

$$Eigenvalue\ error\ (pcm) = \frac{|k - k_{reference}|}{k_{reference}} * 10^5 \quad (28)$$

$$Power_i = (\Sigma_{f,1}^i \phi_{1,i} + \Sigma_{f,2}^i \phi_{2,i}) \cdot constant \quad (29)$$

$$Mean\ Power = \frac{\sum_{i=1}^N |Power_i| V_i}{\sum_{i=1}^N V_i} \quad (30)$$

$$Mean\ Power\ Error = \frac{\sum_{i=1}^N Power\ error_i \cdot |Power_i| \cdot V_i}{\sum_{i=1}^N |Power_i| \cdot V_i} \quad (31)$$

3.1. 3D homogeneous reactor

It is a parallelepiped reactor of dimensions 99 cm x 60 cm x 180 cm and is composed of one material, whose diffusion coefficients and cross sections are exhibited in **Table 1**. Regarding the boundary conditions, zero flux condition was imposed in each boundary. With respect to the meshes, 3 structured and 4 unstructured meshes were simulated. The first structured mesh is composed of 3x3x6 hexahedra, the second one is composed of 6x6x12 hexahedra and the third one is composed of 12x12x24 hexahedra. The unstructured meshes are presented in **Figures 2-5**.

< Table 1 >

<Figure 2>

<Figure 3>

<Figure 4>

<Figure 5>

As far as polynomial terms are concerned, there are only several possible combinations giving valid results. For structured meshes, there are 120 polynomial combinations of order 2 composed of 7 monomials, all of them were used, but only one combination gave valid results: $1, x, y, z, x^2, y^2, z^2$. In case of unstructured meshes, there are 252 polynomial sets composed of 5 monomials, all of them were used, but only 3 combinations gave valid results. The first combination is $1, x, y, z, x^2$; the second one is $1, x, y, z, y^2$ and the third one 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 combination gives also valid and accurate results for coarse unstructured meshes: $1, x, y, z, x^2y^2$.

Regarding the reference solution, this reactor has analytical solution. As regards eigenvalues, the analytical eigenvalues are: 0.99391916952, 0.97602952377, 0.94734259138, 0.93778667636, 0.92148598185. With respect to power results, the modal power corresponding to the unstructured mesh 4 and the third combination of polynomial is shown in **Figures 6** and **7** for the first and second eigenvectors. Regarding the power errors, the reference power and power errors are evaluated at $3 \times 3 \times 6$ nodes. Only the Mean Power Error (Equation 31) is exhibited in this paper due to the extension of the results.

Table 2 contains the computational time and eigenvalue errors for the structured meshes. Mean power errors are 0.00 %. First, the computational time has an order of magnitude of seconds, which is appropriate for neutron diffusion codes. Regarding the eigenvalue errors, only the third mesh gives accurate eigenvalue results. As regards the mean power errors, the 3 meshes give null errors.

< Table 2 >

Table 3 contains the computational time and eigenvalue errors and for the unstructured mesh 1. **Table 4** contains the same results for the unstructured mesh 4. Mean power errors are 0.00% for both meshes. In this case, all polynomial combinations give accurate results, but the finer the mesh the more accurate the eigenvalues. One can see high differences of the eigenvalue errors for the unstructured mesh 1, but these differences are negligible for the unstructured mesh 4, so for fine meshes the polynomial set is almost insensitive.

< Table 3 >

< Table 4 >

A sensitivity analysis of the unstructured mesh was performed and one can see the results in **Table 5**, in which the polynomial combination is fixed (the third combination was used). In this table, the higher the number of cells, the more accurate the eigenvalue results.

< Table 5 >

With respect to mean power errors, unstructured meshes 2 and 3 give non zero errors due to their asymmetry. Therefore, these errors are due to the modelling, not to the method. One notes these errors are virtually zero for the unstructured mesh 3, so the asymmetry is negligible for fine meshes and the polynomial sets are almost insensitive.

3.2. Langenbuch reactor

It is a heterogeneous reactor composed of 4 materials, whose cross sections are shown in **Table 6**. **Figures 8** and **9** exhibit its geometry. Zero flux boundary condition was imposed, except at west and south boundaries, where reflective flux condition was used. With respect to the meshes, 3 structured and 4 unstructured meshes were simulated. The structured mesh 1 is composed of 6x6x10 hexahedra, which is shown in **Figures 8** and **9**. The structured mesh 2 is composed of 12x12x20 hexahedra and the third one is composed of 24x24x40 hexahedra. The

unstructured meshes are shown in **Figures 10-13**. Regarding the polynomial sets, the same sets as in section 3.1 were used, because they were the only ones giving valid results.

< Table 6 >

<Figure 8 >

<Figure 9 >

<Figure 10 >

<Figure 11 >

<Figure 12 >

<Figure 13 >

With respect to the reference solution, the nodal diffusion codes PARCS [9] and VALKIN [10, 11] 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 can calculate several eigenvalues. The eigenvalue calculated by PARCS is 0.995088. The 5 eigenvalues which VALKIN obtains are 0.994881227, 0.948210698, 0.911891847, 0.907632411 and 0.877972136. As regards power results, the modal power errors were evaluated at the nodes of **Figures 8** and **9**, without considering the nodes containing the reflector, that is 8 axial levels (2-9) and 5x5 radial nodes. The results corresponding with the first eigenvector are compared with PARCS and the rest of eigenvectors are compared with VALKIN. Modal power distribution corresponding to the unstructured mesh 4 and the third combination of polynomial is shown in **Figures 14** and **15** for the first and second eigenvectors.

< Figure 14 >

<Figure 15 >

Computational time, eigenvalue errors and mean power errors for the structured meshes are shown in **Table 7**. This table exhibits accurate eigenvalue results for the 3 meshes, but the power results are not good enough for the structured mesh 1. However, the structured

mesh 2 and 3 give excellent power results. Moreover, the computational time is about seconds, which is appropriate for diffusion codes.

< Table 7 >

Table 8 shows the results for the unstructured mesh 1 and **Table 9** those for the unstructured mesh 3. By comparing these tables, one can see that the finer the mesh the more accurate the results, but also the higher the computational time. Overall, the unstructured mesh 1 gives accurate results with computational time about seconds. As regards the polynomial combinations, one can see different results for the coarsest mesh, but the results are almost insensitive for the finest mesh.

< Table 8 >

< Table 9 >

Another sensitivity analysis of the unstructured mesh was performed for Langenbuch reactor, and the results are shown in **Table 10**, in which the polynomial combination is fixed (the third combination was used). Another time, the higher the number of cells, the more accurate the eigenvalue results. However, the coarsest mesh gives accurate results and requires few seconds, so it is the best unstructured mesh.

< Table 10 >

On the whole, the best mesh is the structured mesh 2, because it gives accurate results with computational time of 7 seconds.

3.3. Eigenvalue calculation of a commercial PWR

In this section, the results of a commercial PWR is included to show the power of the method developed. However, the whole description of the reactor is omitted due to the extension of the paper, but its geometry is exhibited in **Figure 16**. **Table 11** shows only the first 5 eigenvalues and its comparison against the results obtained with VALKIN, for a real reactor with structured mesh composed of 8194 nodes. The computational time was 17.65

seconds and the eigenvalue errors are about 200 pcm which are acceptable.

< Figure 16 >

< Table 11 >

4. Conclusions

The inter-cell polynomial expansion method improves the solution of the eigenvalue problem of the 2-energy group neutron diffusion equation discretized by the FVM by reducing the computational time, increasing the accuracy and assuring the flux and current continuity.

The method was applied to homogeneous and heterogeneous reactors, obtaining good results in both cases. A sensitivity analysis of the meshes and polynomial sets was performed. In case of structured meshes, there is only one polynomial set of order 2 giving valid results. In case of unstructured meshes, there are 3 polynomial sets of order 2 giving valid results. In addition, the symmetry of the mesh and the polynomial is important to obtain the best results. The finer the mesh, the more accurate the results and more insensitivity with respect to the polynomial set.

In conclusion, this method gives accurate results for structured and unstructured meshes with computational time about seconds. The finer the mesh, the more accurate the results, but also the higher the computational time. However, this method is capable of obtaining accurate results with relatively coarse meshes, and consequently with low computational time.

Regarding future work, other polynomial expansions will be considered as exponential or sine functions and Legendre polynomial. Moreover, more equations could be added to increase the number of terms of the polynomial expansion. The parallelization of both geometry and solver will be performed. With respect to additional nuclear applications, the following steps will be the thermal-hydraulic coupling and the transitory state.

Nomenclature

ϕ_g : Neutron flux 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

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

$\nu\Sigma_{f,g}$: 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

$k_{n,j}^{grad}$: Kernel of the neutron flux of the cell n used to calculate the gradient of the neutron flux at the face j . The gradient is calculated by multiplying this kernel by the neutron flux of the cell n .

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

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

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

\mathbf{k} : Eigenvalue

$\nu\Sigma_{f,g}^i$: 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

$\bar{p}_t^{V_i}$: Volume averaged value of the term t of the polynomial expansion of the flux for the cell i

$\bar{p}_t^{S_{i,j}}$: 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

References

- [1] Stacey WM. Nuclear Reactor Physics. New York: John Wiley & Sons; 2001.
- [2] Hoffmann KA, Chiang ST. Computational Fluid Dynamics, vol. 2. Wichita: Engineering Education System; 2000.
- [3] 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.
- [4] Theler G. Unstructured Grids and the Multigroup Neutron Diffusion Equation. *Science and Technology of Nuclear Installations*. 2013 July; 2013: 26 pages.
- [5] Harvie DJE. An implicit finite volume method for arbitrary transport equations. *ANZIAM Journal*. 2012 March; 52:C1126–C1145.
- [6] 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.
- [7] Román JE, Campos C, Romero E, Tomas A. SLEPc Users Manual. Tech. Rep. DSIC-II/24/02 - Revision 3.5, Universitat Politècnica de València, 2014.
- [8] 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.
- [9] Downar TJ, Xu Y, Kozlowski T, Carlson D. PARCS v2.7. U.S. NRC Core Neutronics Simulator. User Manual. 2006 August; Rockville, Md.
- [10] 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.
- [11] 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.

Figure captions

- Figure 1. Equations applied to a discretized geometry
- Figure 2. Unstructured mesh 1 of the homogeneous reactor
- Figure 3. Unstructured mesh 2 of the homogeneous reactor
- Figure 4. Unstructured mesh 3 of the homogeneous reactor
- Figure 5. Unstructured mesh 4 of the homogeneous reactor
- Figure 6. Power corresponding to 1st eigenvector and the homogeneous reactor
- Figure 7. Power corresponding to 2nd eigenvector and the homogeneous reactor
- Figure 8. Axial plane of Langenbuch reactor
- Figure 9. Frontal planes of Langenbuch reactor
- Figure 10. Unstructured mesh 1 of Langenbuch reactor
- Figure 11. Unstructured mesh 2 of Langenbuch reactor
- Figure 12. Unstructured mesh 3 of Langenbuch reactor
- Figure 13. Unstructured mesh 4 of Langenbuch reactor
- Figure 14. Power corresponding to 1st eigenvector and Langenbuch reactor
- Figure 15. Power corresponding to 2nd eigenvector and Langenbuch reactor
- Figure 16. Commercial PWR reactor

Table captions

- Table 1. Cross sections of the homogeneous reactor
- Table 2. Results of the homogeneous reactor and the structured meshes
- Table 3. Results of the homogeneous reactor and the unstructured mesh 1
- Table 4. Results of the homogeneous reactor and the unstructured mesh 4
- Table 5. Sensitivity analysis of the unstructured mesh of the homogeneous reactor with the third polynomial set
- Table 6. Cross sections of Langenbuch reactor
- Table 7. Results of Langenbuch reactor and the structured meshes
- Table 8. Results of Langenbuch reactor and the unstructured mesh 1
- Table 9. Results of Langenbuch reactor and the unstructured mesh 3
- Table 10. Sensitivity analysis of the unstructured mesh of Langenbuch reactor with the third polynomial set
- Table 11. Results of a commercial PWR