ARTICLE

Typeset with jnst.cls <ver.2.1>

Multigroup Neutron Diffusion Equation with the Finite Volume Method in reactors using MOX fuels

Álvaro Bernal[1]*, Jose E Roman[2], Rafael Miró[1] and Gumersindo Verdú[1]

[1]*Institute for Industrial,Radiophysical and Environmental Safety, Universitat Politècnica de València, Valencia 46022, Spain;* [2]*Department of Information Systems and Computations, Universitat Politècnica de València, Valencia 46022, Spain*

The use of mixed oxide (MOX) fuel to partially fill the cores of commercial light water reactors (LWRs) gives rise to a reduction of the radioactive waste and production of more energy. However, the use of MOX fuels in LWRs changes the physics characteristics of the reactor core, since the variation with energy of the cross sections for the plutonium isotopes is more complex than for the uranium isotopes. Although the neutron diffusion theory could be applied to reactors using MOX fuels, more emphasis on treatment of the energy discretization should be placed. This energy discretization could be typically 4-8 energy groups, instead of the standard 2 energy group approach. In this work, the authors developed a finite volume method for discretizing the general multigroup neutron diffusion equation. This method solves the eigenvalue problem by using Krylov projection methods, in which the size of the vectors used for building the Krylov subspace does not depend on the number of energy groups, but it can solve the multigroup formulation with upscattering and fission production terms in several energy groups. The method was applied to MOX reactors for its validation.

Keywords: neutron diffusion equation; finite volume method; multigroup; MOX

1. Introduction

The mixed oxide (MOX) fuel was designed to be used in liquid-metal fast breeder reactors (LMFBRs), as well as in light water reactors (LWRs), if reprocessing and recycling are used. Although it has been used for research purposes, it is more important to utilize it to partially fill the cores of commercial LWRs[1]. Actually, reprocessing and recycling of spent fuels give rise to a reduction of the radioactive waste and production of more energy.

First, the potential energy content of the fissile and fertile isotopes remaining in spent reactor fuel constitutes a substantial fraction of the potential energy content of the initial fuel loading. About 96% of this energy content remains in the discharged spent fuel in the form of uranium, plutonium, and higher-actinide isotopes. Only about 1% of the energy content of the uranium used to produce the fuel is extracted in a typical LWR fuel cycle[2].

Second, all the nuclides that can contribute to the potential radiotoxicity of the spent fuel are retained together, resulting in a large volume of high-level activity waste (HLW). Thus, this large volume has to be stored in geological repositories for hundreds of thousands to millions of years[2].

However, the use of MOX fuels in LWRs changes their physics characteristics, since the variation with energy of the cross sections for the plutonium isotopes is more complex than that of the uranium isotopes[2]. Although the neutron diffusion theory could be applied to reactors with MOX fuels, more emphasis on treatment of the energy discretization should be placed. Thus, multigroup neutron diffusion equation with 4-8 energy groups should be applied to these reactors, instead of the classical 2 energy group approach.

Moreover, numerical methods are required to discretize the neutron diffusion equation applied to heterogeneous reactors. There are lot of methods, but those that can be applied to unstructured grids are more interesting, because they can model any kind of geometry

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

and can be applied to reactors without a homogenization approach. Examples of these methods are the finite element and the finite volume method. The authors are interested in the finite volume method because the transported quantity is conserved in the discretized cells. Several works applied the finite volume method to the neutron diffusion equation[3–5]. Among these works, those developed in[4, 5] can calculate multiple eigenvalues, but the results are accurate in fine meshes. For improving the solution in coarser meshes, the authors developed a polynomial expansion method in[6, 7]. This method works accurately in LWR, but it was applied to the 2 energy group approach, without upscattering and with fission production in the first group.

The goal of this work is to develop an algorithm for calculating multiple eigenvalues of the multigroup neutron diffusion equation and validate it in MOX reactors. The algorithm is based on Krylov subspace methods. The major novelty of this algorithm is that the size of the vectors used for building the Krylov subspace does not depend on the number of energy groups, as can be seen in Equation (31), but it can solve the multigroup formulation with upscattering and fission production terms in several energy groups. Thus, one obtains better computational times with this method in comparison with the solution of the generalized eigenvalue problem. For these reasons, this algorithm is not only an extension of the method developed in[6, 7], but also an improvement. It should be pointed out that this algorithm can be applied to any number of energy groups with upscattering, but there is one restriction: no upscattering to the first energy group.

The outline of the paper is as follows. Section 2 explains the multigroup discretization of the neutron diffusion equation with the finite volume method. Section 3 defines the reactors used for the validation of the method and shows the results. Section 4 summarizes the conclusions about this work.

2. Materials and methods

The steady-state multigroup neutron diffusion equation for the g energy group is written in Equation 1. The neutron current in Equation 1 ($\vec{J}_g$) is calculated by means of Fick's Law, as expressed in Equation 2.

$$\begin{aligned} 0 &= -\nabla \vec{J}_g(\vec{r}) - \left(\Sigma_{a,g}(\vec{r}) + \sum_{\substack{g'=1 \\ g' \neq g}}^{G} \Sigma_{s,g \to g'}(\vec{r}) \right) \phi_g(\vec{r}) + \\ &+ \sum_{\substack{g'=1 \\ g' \neq g}}^{G} \Sigma_{s,g' \to g}(\vec{r}) \phi_{g'}(\vec{r}) + \frac{1}{\mathbf{k}} \chi_g(\vec{r}) \sum_{g'=1}^{G} \nu \Sigma_{f,g'}(\vec{r}) \phi_{g'}(\vec{r}) \end{aligned} \tag{1}$$

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

In Equation (1), $\Sigma_{a,g}$ is the absorption macroscopic cross section of the g energy group, $\Sigma_{s,g \to g'}$ is the scattering macroscopic cross section from the g to g' energy groups, $\nu \Sigma_{f,g}$ is the nu-fission macroscopic cross section of the g energy group, χ_g is the fission spectrum for the g energy group, $\mathbf{k}$ is the eigenvalue, ϕ_g is the neutron flux of the g energy group, $\vec{J}_g$ is the neutron current of the g energy group. In Equation (2), D_g is the diffusion coefficient of the g energy group.

For discretizing the derivatives terms of Equations 1-2, the geometry is divided in cells, containing uniform cross sections. If one applies the finite volume method to Equations 1-2 in the the discretized geometry, one obtains Equations 3-4. In Equation 3, S_j is the area of face j, $J_{g,i,j}$ is the face averaged value of J_g in cell i and on face j, n_f is the number of faces of the cell, V_i is the volume of cell i, $\Sigma^i_{a,g}$ is the value of $\Sigma_{a,g}$ in cell i, $\Sigma^i_{s,g \to g'}$ is the value of $\Sigma_{s,g \to g'}$ in cell i, $\phi_{g,i}$ is cell averaged value of ϕ_g in cell i, χ^i_g is the value of χ_g in cell i, $\nu \Sigma^i_{f,g}$ is the value of $\nu \Sigma_{f,g}$ in cell i. Moreover, additional face equations are required. First, boundary conditions are imposed on boundary faces. Second, the heterogeneous neutron flux continuity and the current continuity should be accomplished at the inner

faces of the discretized geometry. The heterogeneous flux continuity can be calculated by using the Assembly Discontinuity Factors (ADFs) as explained in[7]. These continuity conditions are expressed in Equations 5-6, for the cells i and l, which are adjacent to face j. In these equations, D_g^i is the value of D_g in cell i, $\vec{\nabla}\phi_{g,i,j}$ is the face averaged value of $\vec{\nabla}\phi_g$ in cell i and on face j, $\phi_{g,i,j}$ is the face averaged value of ϕ_g in cell i and on face j, $ADF_{g,i,j}$ is the value of ADF of the g energy group, in cell i and on face j.

$$\sum_{j=1}^{n_f} S_j J_{g,i,j} + V_i \left(\Sigma_{a,g}^i + \sum_{\substack{g'=1 \\ g' \neq g}}^{G} \Sigma_{s,g \to g'}^i \right) \phi_{g,i} - V_i \sum_{\substack{g'=1 \\ g' \neq g}}^{G} \Sigma_{s,g' \to g}^i \phi_{g',i} = \frac{1}{\mathbf{k}} V_i \chi_g^i \sum_{g'=1}^{G} \nu \Sigma_{f,g'}^i \phi_{g',i} \quad (3)$$

$$J_{g,i,j} = -D_g^i \vec{\nabla} \phi_{g,i,j} \quad (4)$$

$$J_{g,i,j} = -J_{g,l,j} \quad (5)$$

$$ADF_{g,i,j} \phi_{g,i,j} = ADF_{g,l,j} \phi_{g,l,j} \quad (6)$$

Equations 3-6 contain face and cell averaged values of the neutron flux and current. These values can be calculated by means of a polynomial expansion method[6, 7]. In this method, the neutron flux in each cell is expressed as a finite sum of polynomial terms, as shown in Equation 7. In this polynomial expansion, the polynomial terms ($p_t(x, y, z)$) for each cell were assigned previously and the coefficients of the expansion ($a_{g,i,t}$) are determined by solving the eigenvalue problem. The authors performed a sensitivity analysis for determining the best set of polynomial terms $x^{\alpha_t} y^{\beta_t} z^{\gamma_t}$, as discussed in Section 3. In the cited reference[6], it is demonstrated that the number of polynomial terms should be equal to the number of faces of the cell plus one. Since the polynomial terms are known, one can easily calculate the cell and face averaged values of the neutron flux and current

with Equations 8-10. In Equation 10, $u_{i,j,x}$, $u_{i,j,y}$ and $u_{i,j,z}$ are the direction cosines of the normal to face j.

$$\phi_{g,i}(x,y,z) = \sum_{t=1}^{n_f+1} a_{g,i,t} p_t(x,y,z) = \sum_{t=1}^{n_f+1} a_{g,i,t} x^{\alpha_t} y^{\beta_t} z^{\gamma_t} \tag{7}$$

$$\phi_{g,i} = \frac{1}{V_i}\int_{V_i} \phi_{g,i}(x,y,z)dV = \sum_{t=1}^{n_f+1} a_{g,i,t}\frac{1}{V_i}\int_{V_i} p_t(x,y,z)dV = \sum_{t=1}^{n_f+1} a_{g,i,t}\overline{p}_t^{V_i} \tag{8}$$

$$\phi_{g,i,j} = \frac{1}{S_j}\int_{S_j} \phi_{g,i}(x,y,z)dS = \sum_{t=1}^{n_f+1} a_{g,i,t}\frac{1}{S_j}\int_{S_j} p_t(x,y,z)dS = \sum_{t=1}^{n_f+1} a_{g,i,t}\overline{p}_t^{S_{i,j}} \tag{9}$$

$$\begin{aligned}
\vec{\nabla}\phi_{g,i,j} &= \frac{1}{S_j}\int_{S_j} \vec{\nabla}\phi_{g,i}(x,y,z)dS \\
&= \sum_{t=1}^{n_f+1} a_{g,i,t}\left(u_{i,j,x}\frac{1}{S_j}\int_{S_j}\frac{dp_t(x,y,z)}{dx}dS + u_{i,j,y}\frac{1}{S_j}\int_{S_j}\frac{dp_t(x,y,z)}{dy}dS \right. \\
&\quad \left. + u_{i,j,z}\frac{1}{S_j}\int_{S_j}\frac{dp_t(x,y,z)}{dz}dS \right) = \sum_{t=1}^{n_f+1} a_{g,i,t}\overline{\vec{\nabla}p_t}^{S_{i,j}}
\end{aligned} \tag{10}$$

For each energy group, one has one Equation 3 for each cell, one boundary condition for each boundary face and two Equations 5-6 for each inner face. However, one can define the boundary conditions and the current continuity (Equation 5) implicitly, as explained in[7]. Thus, the equations are reduced to N_c Equations 3 and N_f Equations 6, where N_c is the number of cells and N_f is the number of inner faces. In addition, the unknowns will be $\phi_{g,i}$, $J_{g,j}$ and $\phi_{g,i,j}$ instead of $a_{g,i,t}$. Now, $J_{g,i,j} = u_{i,j}J_{g,j}$, with $u_{i,j} = \pm 1$. However, $\phi_{g,i,j}$ can be calculated from $\phi_{g,i}$ and $J_{g,j}$. For this purpose, Equations 8-10 are considered to build the linear system defined in Equation 11, in which the matrix will be named I_i. In Equation 11, $F_{g,i,j}$ and $f_{i,j,t}$ are defined in Equations 12-13. So, one can calculate the $a_{g,i,t}$ coefficients by calculating the inverse of I_i as expressed in Equation 14.

$$\begin{bmatrix} \overline{p}_1^{V_i} & \cdots & \overline{p}_{n_f+1}^{V_i} \\ f_{i,1,1} & \cdots & f_{i,1,n_f+1} \\ \vdots & & \vdots \\ f_{i,n_f,1} & \cdots & f_{i,n_f,n_f+1} \end{bmatrix} \begin{bmatrix} a_{g,i,1} \\ \vdots \\ a_{g,i,n_f+1} \end{bmatrix} = \begin{bmatrix} \phi_{g,i} \\ F_{g,i,1} \\ \vdots \\ F_{g,i,n_f} \end{bmatrix} \tag{11}$$

$$F_{g,i,j} = \begin{cases} \phi_{g,i,j} & \text{if face } j \text{ is a boundary face of zero flux condition} \\ -u_{i,j}\vec{\nabla}\phi_{g,i,j} & \text{the rest of cases} \end{cases} \tag{12}$$

$$f_{i,j,t} = \begin{cases} \overline{p}_t^{S_{i,j}} & \text{if face } j \text{ is a boundary face of zero flux condition} \\ -u_{i,j}\overrightarrow{\nabla p_t}^{S_{i,j}} & \text{the rest of cases} \end{cases} \tag{13}$$

$$\begin{bmatrix} a_{g,i,1} \\ \vdots \\ a_{g,i,n_f+1} \end{bmatrix} = I_i^{-1} \begin{bmatrix} \phi_{g,i} \\ F_{g,i,1} \\ \vdots \\ F_{g,i,n_f} \end{bmatrix} \tag{14}$$

If one combines Equations 9 and 14, $\phi_{g,i,j}$ can be calculated with Equation 15, with $X_{i,j,k}$ defined in Equation 16. One can see that each $X_{i,j,k}$ represents the contribution of each unknown to $\phi_{g,i,j}$.

$$\phi_{g,i,j} = X_{i,j,1}\phi_{g,i} + \sum_{t=1}^{n_f} X_{i,j,t+1}F_{g,i,t} \tag{15}$$

$$X_{i,j,k} = \sum_{t=1}^{n_f+1} \overline{p}_t^{S_{i,j}} I_i^{-1}(t,k) \tag{16}$$

If a boundary face has a reflective boundary condition, the current at this face will be zero. However, if the boundary condition is zero flux, the current has to be calculated. For calculating it, one may combine Equations 10 and 14 to obtain Equation 17, which calculates $\vec{\nabla}\phi_{g,i,j}$, with $R_{i,j,k}$ defined in Equation 18. One can see that each $R_{i,j,k}$ represents the contribution of each unknown to $\vec{\nabla}\phi_{g,i,j}$.

$$-\vec{\nabla}\phi_{g,i,j} = R_{i,j,1}\phi_{g,i} + \sum_{t=1}^{n_f} R_{i,j,t+1}F_{g,i,t} \tag{17}$$

$$R_{i,j,k} = -\sum_{t=1}^{n_f+1} \overrightarrow{\vec{\nabla}p_t}^{S_{i,j}} I_i^{-1}(t,k) \tag{18}$$

It is better to multiply Equations 15 and 17 by D_g^i, as expressed in Equations 19-20, because $D_g^i F_{g,i,j}$ is related to the current, as Equation 21 shows.

$$D_g^i\phi_{g,i,j} = X_{i,j,1}D_g^i\phi_{g,i} + \sum_{t=1}^{n_f} X_{i,j,t+1}D_g^i F_{g,i,t} \tag{19}$$

$$J_{g,i,j} = -D_g^i\vec{\nabla}\phi_{g,i,j} = R_{i,j,1}D_g^i\phi_{g,i} + \sum_{t=1}^{n_f} R_{i,j,t+1}D_g^i F_{g,i,t} \tag{20}$$

$$D_g^i F_{g,i,j} = \begin{cases} 0 & \text{if face } j \text{ is a boundary face} \\ J_{g,i,j} = u_{i,j}J_{g,j} & \text{if face } j \text{ is an inner face} \end{cases} \tag{21}$$

Finally, Equations 3 and 6 can be transformed into Equations 22-23, which will be the equations of the eigenvalue problem.

$$\sum_{j=1}^{n_f} S_j u_{i,j} J_{g,j} + V_i \left(\Sigma_{a,g}^i + \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \Sigma_{s,g\to g'}^i \right) \phi_{g,i} - V_i \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \Sigma_{s,g'\to g}^i \phi_{g',i} = \frac{1}{\mathbf{k}} V_i \chi_g^i \sum_{g'=1}^{G} \nu\Sigma_{f,g'}^i \phi_{g',i} \tag{22}$$

$$\begin{aligned} 0 &= ADF_{g,i,j}\phi_{g,i,j} - ADF_{g,l,j}\phi_{g,l,j} \\ &= \frac{ADF_{g,i,j}}{D_g^i}\left(X_{i,j,1}D_g^i\phi_{g,i} + \sum_{t=1}^{n_f} X_{i,j,t+1}D_g^i F_{g,i,t} \right) \\ &\quad -\frac{ADF_{g,l,j}}{D_g^l}\left(X_{l,j,1}D_g^l\phi_{g,l} + \sum_{t=1}^{n_f} X_{l,j,t+1}D_g^l F_{g,l,t} \right) \end{aligned} \tag{23}$$

Next, the matrix form of Equations (22)-(23) is shown in Equation (24) for a simple case: 2 energy groups, cells i and l, face j and boundary conditions of reflective flux.

$$\begin{pmatrix} V_i\left(\Sigma^i_{a,1}+\Sigma^i_{s,1\to2}\right) & 0 & S_j u_{i,j} & -V_i\Sigma^i_{s,2\to1} & 0 & 0 \\ 0 & V_l\left(\Sigma^l_{a,1}+\Sigma^l_{s,1\to2}\right) & S_j u_{l,j} & 0 & -V_l\Sigma^l_{s,2\to1} & 0 \\ ADF_{1,i,j}X_{i,j,1} & -ADF_{1,l,j}X_{l,j,1} & \frac{ADF_{1,i,j}}{D^i_1}X_{i,j,k}u_{i,j}-\frac{ADF_{1,l,j}}{D^l_1}X_{l,j,k}u_{l,j} & 0 & 0 & 0 \\ -V_i\Sigma^i_{s,1\to2} & 0 & 0 & V_i\left(\Sigma^i_{a,2}+\Sigma^i_{s,2\to1}\right) & 0 & S_j u_{i,j} \\ 0 & -V_l\Sigma^l_{s,1\to2} & 0 & 0 & V_l\left(\Sigma^l_{a,2}+\Sigma^l_{s,2\to1}\right) & S_j u_{l,j} \\ 0 & 0 & 0 & ADF_{2,i,j}X_{i,j,1} & -ADF_{2,l,j}X_{l,j,1} & \frac{ADF_{2,i,j}}{D^i_2}X_{i,j,k}u_{i,j}-\frac{ADF_{2,l,j}}{D^l_2}X_{l,j,k}u_{l,j} \end{pmatrix} \begin{pmatrix} \phi_{1,i} \\ \phi_{1,l} \\ J_{1,j} \\ \phi_{2,i} \\ \phi_{2,l} \\ J_{2,j} \end{pmatrix}$$

$$= \frac{1}{\mathbf{k}} \begin{pmatrix} V_i\chi^i_1\nu\Sigma^i_{f,1} & 0 & 0 & V_i\chi^i_1\nu\Sigma^i_{f,2} & 0 & 0 \\ 0 & V_l\chi^l_1\nu\Sigma^l_{f,1} & 0 & 0 & V_l\chi^l_1\nu\Sigma^l_{f,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ V_i\chi^i_2\nu\Sigma^i_{f,1} & 0 & 0 & V_i\chi^i_2\nu\Sigma^i_{f,2} & 0 & 0 \\ 0 & V_l\chi^l_2\nu\Sigma^l_{f,1} & 0 & 0 & V_l\chi^l_2\nu\Sigma^l_{f,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \phi_{1,i} \\ \phi_{1,l} \\ J_{1,j} \\ \phi_{2,i} \\ \phi_{2,l} \\ J_{2,j} \end{pmatrix} \tag{24}$$

If one considers the whole geometry and G energy groups, the eigenvalue problem will be that of Equation (25), in which the authors considered that the first energy group containing upscattering terms is energy group u. In this eigenvalue problem, the eigenvector is composed of vectors Φ_g, which are defined in Equation (26), for each energy group g.

$$\begin{pmatrix} L_{1,1} & 0 & & \cdots & & 0 \\ \vdots & \ddots & \ddots & & & \vdots \\ L_{u-1,1} & \cdots & L_{u-1,u-1} & 0 & \cdots & 0 \\ L_{u,1} & \cdots & \cdots & L_{u,u} & \cdots & L_{u,G} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ L_{G,1} & \cdots & \cdots & L_{G,u} & \cdots & L_{G,G} \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_G \end{pmatrix} = \frac{1}{\mathbf{k}} \begin{pmatrix} M_{1,1} & \cdots & M_{1,G} \\ \vdots & \vdots & \vdots \\ M_{G,1} & \cdots & M_{G,G} \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_G \end{pmatrix} \tag{25}$$

$$\Phi_g = \begin{pmatrix} \phi_{g,1} \\ \vdots \\ \phi_{g,N_c} \\ J_{g,1} \\ \vdots \\ J_{g,N_f} \end{pmatrix} \tag{26}$$

Matrices $M_{g,g'}$ contain the fission terms, which are the same except the fission spectrum (χ_g). Therefore, $M_{g,g'}$ can be expressed as in Equation 27. By using this relation, one can multiply Equation 25 by matrix P, defined in Equation 28, so one obtains the eigenvalue problem of Equation 29. The submatrix I appearing in matrix P is the identity matrix. Matrices $\bar{L}_{g,1}$ are calculated as in Equation 30.

$$M_{g,g'} = diag\left(\frac{\chi_g^1}{\chi_1^1}, \cdots, \frac{\chi_g^{N_c}}{\chi_1^{N_c}}, 0, \cdots, 0\right) \cdot M_{1,g'} = C_g M_{1,g'} \tag{27}$$

$$P = \begin{pmatrix} I & 0 & \cdots & \cdots & 0 \\ -C_2 & I & \ddots & & \vdots \\ -C_3 & 0 & I & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -C_G & 0 & \cdots & 0 & I \end{pmatrix} \tag{28}$$

$$\begin{pmatrix} L_{1,1} & 0 & & \cdots & & 0 \\ \vdots & \ddots & \ddots & & & \vdots \\ \bar{L}_{u-1,1} & \cdots & L_{u-1,u-1} & 0 & \cdots & 0 \\ \bar{L}_{u,1} & \cdots & \cdots & L_{u,u} & \cdots & L_{u,G} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{L}_{G,1} & \cdots & \cdots & L_{G,u} & \cdots & L_{G,G} \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_G \end{pmatrix} = \frac{1}{\mathbf{k}} \begin{pmatrix} M_{1,1} & \cdots & M_{1,G} \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_G \end{pmatrix} \tag{29}$$

$$\bar{L}_{g,1} = L_{g,1} - C_g L_{1,1} \tag{30}$$

From Equation 29, the eigenvalue problem is reduced to the eigenvalue problem of Equation 31, which is solved by using an iterative process, for several eigenpairs. In this iterative process, Φ_1 is the iterative eigenvector and Φ_g for $g > 1$ are calculated with Equations 32-33. Equation 32 is used for calculating sequentially Φ_g, for $g = 2, \ldots, u-1$, because there is no upscattering for these energy groups. Once they are calculated, Equation 33 is used for calculating Φ_g, for $g \geq u$. Then, Φ_g is substituted in the right-hand-side of Equation 31. Equations 32-33 are linear systems and are obtained from Equation 29. It is important to highlight that the inverse of $L_{1,1}$ is not calculated in Equation 31, but instead the linear system is solved in each iteration of the eigensolver.

$$\mathbf{k}\Phi_1 = L_{1,1}^{-1} \sum_{g=1}^{G} M_{1,g}\Phi_g \tag{31}$$

$$L_{g,g}\Phi_g = -\bar{L}_{g,1}\Phi_1 - \sum_{g'=2}^{g-1} L_{g,g'}\Phi_{g'} \quad , \quad 2 \leq g < u \tag{32}$$

$$\begin{pmatrix} L_{u,u} & \cdots & L_{u,G} \\ \vdots & \vdots & \vdots \\ L_{G,u} & \cdots & L_{G,G} \end{pmatrix} \begin{pmatrix} \Phi_u \\ \vdots \\ \Phi_G \end{pmatrix} = \begin{pmatrix} -\bar{L}_{u,1}\Phi_1 - \sum_{g'=2}^{u-1} L_{u,g'}\Phi_{g'} \\ \vdots \\ -\bar{L}_{G,1}\Phi_1 - \sum_{g'=2}^{u-1} L_{G,g'}\Phi_{g'} \end{pmatrix} \tag{33}$$

The eigenvalue problem of Equation 31 is solved with the Krylov-Schur algorithm[8], implemented in the SLEPc library and for several modes. SLEPc, the Scalable Library for Eigenvalue Problem Computations[9,10], is a software library for the solution of large, sparse eigenproblems on parallel computers. It provides projection methods or other methods with similar properties, such as Krylov-Schur or Jacobi-Davidson. SLEPc is built on top of PETSc (Portable, Extensible Toolkit for Scientific Computation)[11] and extends it with all the functionality necessary for the solution of eigenvalue problems, which includes matrix operation and solution of linear systems. PETSc includes several linear systems solvers, such as: Conjugate Gradient, BiConjugate Gradient, GMRES, Generalized Conjugate Residual, BiCGSTAB and Conjugate Gradient Squared. The authors tested these solvers, but the fastest solver for this method was Generalized Minimal Residual (GMRES)[12], using the Additive Schwarz preconditioner. These solvers are based on iterative methods, which require less computational resources than direct methods. However, this is true if the system matrix is well-conditioned; for ill-conditioned matrices it is better to use direct solvers, as discussed in Section 3.1. In particular, the authors used the version 3.7 of both PETSc and SLEPc.

3. Results

The authors tested the method in two MOX benchmarks. The first one is C5G7 MOX Benchmark[13]. In particular, only the 2D benchmark was tested. The second one is PWR

MOX/UO_2 Core Transient Benchmark[14], which is a 3D benchmark; but only the steady state was tested.

Both benchmarks were modeled and meshed by means of Gmsh code[15], version 2.16.0, which is a 3D finite element grid generator with a built-in CAD engine. The C5G7 MOX Benchmark was meshed with an unstructured grid composed of quadrangles and the PWR MOX/UO_2 Benchmark was meshed with unstructured grids composed of hexahedra.

The Power Errors (PE) and Eigenvalue Error (EE) are used to evaluate the results and are defined in Equations 34 and 35. The power for each cell (P_i) is defined in Equation 36. 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 37.

$$PE_i(\%) = \frac{|P_i - P_{i,ref}|}{P_{i,ref}} \cdot 100 \tag{34}$$

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

$$P_i = constant \cdot \sum_{g=1}^{G} \Sigma_{f,g}^{i} \phi_{g,i} \tag{36}$$

$$MPE(\%) = \frac{\sum_i PE_i(\%)|P_i|V_i}{\sum_i |P_i|V_i} \tag{37}$$

The reference results (ref) for the C5G7 MOX Benchmark were obtained with MCNP code[16], which is a Monte Carlo code. These results are given in the mentioned benchmark. On the other hand, for the PWR MOX/UO_2 Benchmark, the reference solution was obtained with TRIVAC code[17], which is a diffusion code. TRIVAC can solve the neutron diffusion equation with several methods, but in this work the authors used the

Nodal Collocation Method[18], with Legendre polynomials of third order. In addition, the authors used a version of TRIVAC based on its version 5, but including an eigensolver based on the SLEPc library[19].

Moreover, five eigenmodes were calculated and all the CPU time values have been obtained on an Intel Core i7-3770 CPU (3.4GHz), with the CentOS 6.8 operating system.

3.1. 2D C5G7

The geometry and composition of this benchmark[13] is shown in Figure 1. The overall dimensions of the 2D configuration of Figure 1 are 64.26 cm × 64.26 cm, while each assembly is 21.42 cm × 21.42 cm. Each fuel assembly is made up of a 17 × 17 lattice of square pin level cells, as shown in Figure 2. The side length of each pin cell is 1.26 cm and all of the fuel pins and guide tubes have a 0.54 cm radius. For this benchmark problem, a single moderator composition is provided for use in all of the pin cells and for use in the water moderator (reflector) surrounding the assemblies. The composition layout for all four assemblies is provided in Figure 3. This benchmark uses 7 energy groups and the cross sections for each material are given in Appendix A in[13]. As regards the ADFs, they have a value of 1.0. The boundary conditions are shown in Figure 1.

[Figure 1 about here.]

[Figure 2 about here.]

[Figure 3 about here.]

This benchmark was modeled with the mesh of Figure 4. A zoom of the mesh is given in Figure 5, which shows the mesh in the pin cells. One can see in Figure 5 that the fuel-clad circle was modeled as a regular hexadecagon. The radius of this polygon (R_p) was calculated to obtain the same area as the circle of radius (R_c) 0.56 cm, so the reaction rates are conserved. The analytical expression for R_p is shown in Equation 38.

[Figure 4 about here.]

[Figure 5 about here.]

$$R_p = R_c \cdot \sqrt{\frac{\pi}{8 \sin\left(\frac{\pi}{8}\right)}} = 1.013 \cdot R_c \tag{38}$$

As the cells of these meshes are quadrangles, the number of faces of each cell (n_f) is 4. Thus, the polynomial expansion is limited to 5, as discussed in[6, 7]. There are infinitely many possible polynomial sets, so the authors restricted the sets to monomials $x^{\alpha_t} y^{\beta_t}$ of order 2, that is, $\alpha_t + \beta_t \leq 2$. There are six 2D monomials of order 2: 1, x, y, x^2, y^2 and xy. Thus, there are six possible 5-combinations of the set composed of these six monomials. The authors tested these six combinations, and only one of them gave valid results:1, x, y, x^2, y^2.

Although there are only reference results for the first mode (fundamental mode), the authors calculated five modes. It is important to highlight that this benchmark was solved by using direct solvers for the linear systems, because the matrices were ill-conditioned, and consequently GMRES solver was slower than a direct solver. In particular, the authors used the LU decomposition of the MUMPS library[20]. The computational time was 34 seconds and the five eigenvalues are: 1.182958, 0.903905, 0.859309, 0.702847 and 0.562519. These eigenvalues corresponds to a quarter of the core, but they are not the five largest ones. For calculating them, the full core should be calculated. Thus, the authors also made the calculation of the full core, which required 2 minutes and 38 seconds, obtaining the following eigenvalues: 1.182958, 1.038667, 0.949189, 0.903905 and 0.859309. The eigenvalue error (EE) was -302.75 pcm, the mean power error (MPE) was 7.33% and the maximum power error was 13.17%. In addition, Figure 6 displays the power errors (PE_i). In this figure, one can see high errors at the boundaries of MOX assemblies. These errors are high, even the minimum error, but it is normal, because the reference solution is obtained with a Monte Carlo method. Although these errors are high for a transport code, the results are acceptable for a diffusion code.

[Figure 6 about here.]

3.2. PWR MOX/UO_2

This case is a PWR reactor composed of 224 fuel assemblies of different compositions and burnup[14]. The assembly pitch is 21.42 cm and the active fuel length is 365.76 cm, which has been modeled with 20 axial levels. The reflector width is also 21.42 cm.

A quarter of the core characteristics is displayed in Figure 7. In this figure, the composition U 4.2 corresponds to UO_2 fuel type and enrichment of 4.2%. By contrast, M 4.0 corresponds to MOX fuel type and enrichment of 4.0%. One can find further details of the composition of these fuels in[14]. The core has uniform fuel composition in axial direction with axial reflector of the same width as the fuel assembly pitch. Thus, the model includes two more axial levels for the reflector, so it is modeled with 22 axial levels.

[Figure 7 about here.]

The cross sections were obtained for 8 energy groups and the following thermal-hydraulic conditions: inlet coolant temperature of 560 K and inlet pressure of 15.5 MPa. From Figure 7, one can conclude that there are 18 different sets of cross sections, so the authors did not include them in this paper due to the extent of the data; yet it is important to highlight that there are upscattering terms in energy groups 6 and 7. As regards the ADFs, they have a value of 1.0.

All boundary conditions for this benchmark are zero flux. The reference results were calculated with TRIVAC code, as mentioned in Section 3. The mesh used in TRIVAC is shown in Figure 8. On the other hand, three meshes were used to validate the method of this work and perform a sensitivity analysis. Mesh 1 is the same as that of Figure 8. Mesh 2 is obtained by generating 2 x 2 x 2 identical hexahedra in each hexahedron of Mesh 1. Mesh 3 is obtained by generating 3 x 3 x 3 identical hexahedra in each hexahedron of Mesh 1.

[Figure 8 about here.]

As the cells of these meshes are hexahedra, the number of faces of each cell (n_f) is 6. Thus, the polynomial expansion is limited to 7, as discussed in[6, 7]. There are infinitely many possible polynomial sets, so the authors restricted the sets to monomials $x^{\alpha_t} y^{\beta_t} z^{\gamma_t}$ of order 2, that is, $\alpha_t + \beta_t + \gamma_t \leq 2$. There are ten 3D monomials of order 2: 1, x, y, z, x^2, y^2, z^2, xy, xz and yz. Thus, there are 120 possible 7-combinations of the set composed of these ten monomials. The authors tested these 120 combinations, and only one of them gave valid results:1, x, y, z, x^2, y^2, z^2.

The computational time for TRIVAC calculation was 1 minute and 38 seconds and the five eigenvalues are: 1.149530, 1.137884, 1.135261, 1.135259 and 1.124268. The computational time for the present method was: 8 seconds for Mesh 1, 1 minute and 18 seconds for Mesh 2, and 5 minutes and 30 seconds for Mesh 3. As regards the results, Table 1 summarizes the results corresponding to the first mode, in which MEA is the maximum power error per assembly; Table 2 shows the axial power errors for the first mode. In addition, only eigenvalue errors for modes 2-5 are shown in Table 3, due to the extent of the results. Finally, Figure 9 displays the assembly power errors corresponding to the first mode and Mesh 3. This figure looks a little bit asymmetrical due to the fact that results obtained with TRIVAC are a little asymmetrical, but they should not be asymmetrical. The reason for this little asymmetry is round-off errors when cell averaged values are calculated.

[Table 1 about here.]

[Table 2 about here.]

[Table 3 about here.]

[Figure 9 about here.]

In conclusion, this method obtains accurate results. Firstly, eigenvalue errors are excellent, even for Mesh 1, with a maximum value of 42.25 pcm for the second mode. Secondly,

power errors are also good, specially for Mesh 3, obtaining a maximum axial power error of 0.86% and a maximum assembly power error of 1.56%. Finally, one can see that the finer the mesh, the better the results. Although the maximum error increases in Mesh 2 in comparison with Mesh 1, the mean power error (MPE) and axial power errors decrease, except those of axial levels 2 and 21.

If one compares the axial power errors of Meshes 1 and 2, one can see that errors of Mesh 1 are two or three times higher (or even more) than those of Mesh 2, except for axial levels 2 and 21. Power errors are bigger next to the reflector. A possible reason might be round-off errors. Round-off errors have more effect in power errors for low power values than for high ones. In this reactor, the power calculated by TRIVAC in axial levels 2 and 11 are 0.2461 and 1.4849. A variation of the power of 0.005 in axial level 2 corresponds to a percentage variation of 2.03%; the same variation in axial level 11 corresponds to a percentage variation of 0.34%. This example justifies the big errors of axial levels 2 and 21. Furthermore, Figure 10 shows that the axial power for all meshes is virtually identical to that of TRIVAC.

[Figure 10 about here.]

4. Conclusions

An efficient method has been developed for solving the multigroup neutron diffusion equation, for any number of energy groups, including upscattering terms and using the finite volume method for discretizing the spatial derivative terms.

The method can be applied to any geometry, since it uses the finite volume method, which can be applied to unstructured grids. Moreover, the method assures the accomplishment of the current continuity and the continuity of the heterogeneous flux, by using ADFs. The method is based on Krylov subspace methods. The major novelty of this method is that the size of the vectors used for building the Krylov subspace does not de-

pend on the number of energy groups, but it can solve the general multigroup formulation with upscattering and fission production terms in several energy groups.

The authors applied the method to two MOX reactors to validate it. The first one is C5G7 MOX Benchmark, with the aim of checking the application to any kind of geometries. This problem used 7 energy-groups, with upscattering terms in groups 4-7. The second one is PWR MOX/UO_2 Core Transient Benchmark, for testing the capabilities of the method in PWR reactors. This problem used 8 energy-groups, with upscattering terms in groups 6-8. The authors also performed a sensitivity analysis of the geometry discretization in PWR MOX/UO_2 Core Transient Benchmark, obtaining excellent results.

As regards future work, the parallelization of both geometry and solver will be performed. In addition, the following nuclear applications will be the transitory state and thermal-hydraulic coupling.

Acknowledgement

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 the project ENE2014-59442-P, the Spanish Ministerio de Economía y Competitividad and the European Fondo Europeo de Desarrollo Regional (FEDER) under the project ENE2015-68353-P (MINECO/FEDER), the Generalitat Valenciana under the project PROMETEOII/2014/008, and the Spanish Ministerio de Economía y Competitividad under the project TIN2016-75985-P.

References

[1] Cochran RG, Tsoulfanidis N.The Nuclear Fuel Cycle: Analysis and Management. American Nuclear Society;1999. Chapter 4, Reactor Fuel Design and Fabrication, 77-104.

[2] Stacey WM. Nuclear Reactor Physics. New York: John Wiley & Sons; 2001. Chapter 6, Fuel Burnup,

195-242.

[3] Ge J, Zhang D, Tian W, Wang K, Qiu S, Su GH. Steady and transient solutions of neutronics problems based on finite volume method (FVM) with a CFD code. Prog. Nucl. Energ. 2015 Nov; 85:366-374.

[4] Theler G. Unstructured grids and the multigroup neutron diffusion equation. Sci. Technol. Nucl. Install. 2013 Jul;2013:26.

[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 Appl. Anal. 2014 Jan;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. J. Nucl. Sci. Technol. 2015 Oct; 53:1212-1223.

[7] Bernal A, Roman JE, Miró R, Verdú G. Assembly Discontinuity Factors for the Neutron Diffusion Equation discretized with the Finite Volume Method. Application to BWR. Ann. Nucl. Energy. 2016 Nov; 97:76-85.

[8] Stewart GW. A Krylov-Schur algorithm for large eigenproblems. SIAM. J. Matrix Anal. Appl. 2002 Mar; 23:601-614.

[9] Hernandez V, Roman JE, Vidal V. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. ACM Trans. Math. Software. 2005 Sep; 31:351-362.

[10] Hernandez V, Roman JE, Vidal V. SLEPc: Scalable Library for Eigenvalue Problem Computations. Lect. Notes Comp. Sci. 2003 Apr; 2565:377-391.

[11] Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Eijkhout V, Gropp WD, Kaushik D, Knepley MG, Curfman L, Rupp K, Smith BF, Zampini S, Zhang H. PETSc Users Manual. Argonne: Argonne National Laboratory; 2016, Report no. ANL-95/11 Rev 3.7.

[12] Saad Y, Schultz MH. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. SIAM J. Sci. and Stat. Comput. 1986 Jul; 7:856-869.

[13] Lewis EE, Smith MA, Tsoulfanidis N, Palmiotti G, Taiwo TA, Blomquist RN. Benchmark Specification for Deterministic 2-D/3-DMOX Fuel Assembly Transport Calculations without Spatial Homogenisation (C5G7 MOX Benchmark). Nuclear Energy Agency; 2003, NEA/NSC/DOC(2003)16.

[14] Kozlowski T, Downar TJ. PWR MOX/UO_2 Core Transient Benchmark. Nuclear Energy Agency; 2006, NEA/NSC/DOC(2006)20.

[15] Geuzaine C, Remacle J-F. Gmsh: a 3-D finite element mesh generator with built.in pre- and post-processing facilities. Int J Numer Methods Eng. 2009;79:1309-1331.

[16] Briesmeister JF. MCNP^{TM} - A General Monte Carlo N-particle Transport Code. Los Alamos National Laboratory; 1997, LA-12625-M.

[17] Hébert A. A user guide for TRIVAC version4. Canada: Institut de génie nucleaire, École Polytechnique de Montréal; 2014, Report no. IGE-2293.

[18] Hébert A. Development of the nodal collocation method for solving the neutron diffusion equation. Ann. Nucl. Energy. 1987 May; 14:527-541.

[19] Bernal A, Hébert A, Roman JE, Miró R, Verdú G. A Krylov-Schur solution of the eigenvalue problem for the neutron diffusion equation discretized with the Raviart-Thomas method. J. Nucl. Sci. Technol. 2017 Jul; 1-10.

[20] Amestoy PR, Duff IS, L'Excellent J-Y. Multifrontal parallel distributed symmetric and unsymmetric solvers. Comput. Methods in Appl. Mech. Eng. 2000 Apr; 184:501-520.

Table 1 Results for the PWR MOX/UO_2 and first mode.

Mesh	$EE(pcm)$	$MEA(\%)$	$MPE(\%)$
1	39.85	3.89	1.55
2	4.45	3.98	1.25
3	2.89	1.56	0.43

Table 2 Axial power errors (%) for the PWR MOX/UO_2 and first mode.

Axial level	Mesh 1	Mesh 2	Mesh 3
21	1.66	2.44	0.86
20	1.43	0.20	0.06
19	0.87	0.09	0.02
18	0.34	0.01	0.00
17	0.06	0.04	0.01
16	0.12	0.06	0.02
15	0.24	0.09	0.03
14	0.32	0.11	0.04
13	0.36	0.12	0.04
12	0.39	0.12	0.04
11	0.39	0.13	0.04
10	0.36	0.12	0.04
9	0.31	0.11	0.04
8	0.23	0.10	0.04
7	0.11	0.07	0.03
6	0.07	0.04	0.01
5	0.35	0.01	0.01
4	0.87	0.09	0.01
3	1.43	0.19	0.06
2	1.67	2.43	0.85

Table 3 Eigenvalue errors for the PWR MOX/UO_2 and modes 2-5.

Mesh	$EE_2(pcm)$	$EE_3(pcm)$	$EE_4(pcm)$	$EE_5(pcm)$
1	42.25	34.63	34.80	36.81
2	0.45	-8.63	-8.55	-17.80
3	1.57	-2.35	-2.23	-5.70

Figure Captions

Figure 1 Core configuration for the 2D C5G7.

Figure 2 Pin cell geometry for the 2D C5G7.

Figure 3 Pin cell compositions for the 2D C5G7.

Figure 4 Mesh for the 2D C5G7.

Figure 5 Zoom of the mesh for the 2D C5G7.

Figure 6 Power errors for the 2D C5G7.

Figure 7 A quarter of the core for the PWR MOX/UO_2.

Figure 8 Mesh for the PWR MOX/UO_2.

Figure 9 Assembly power errors for the PWR MOX/UO_2 and Mesh 3.

Figure 10 Axial power for the PWR MOX/UO_2.

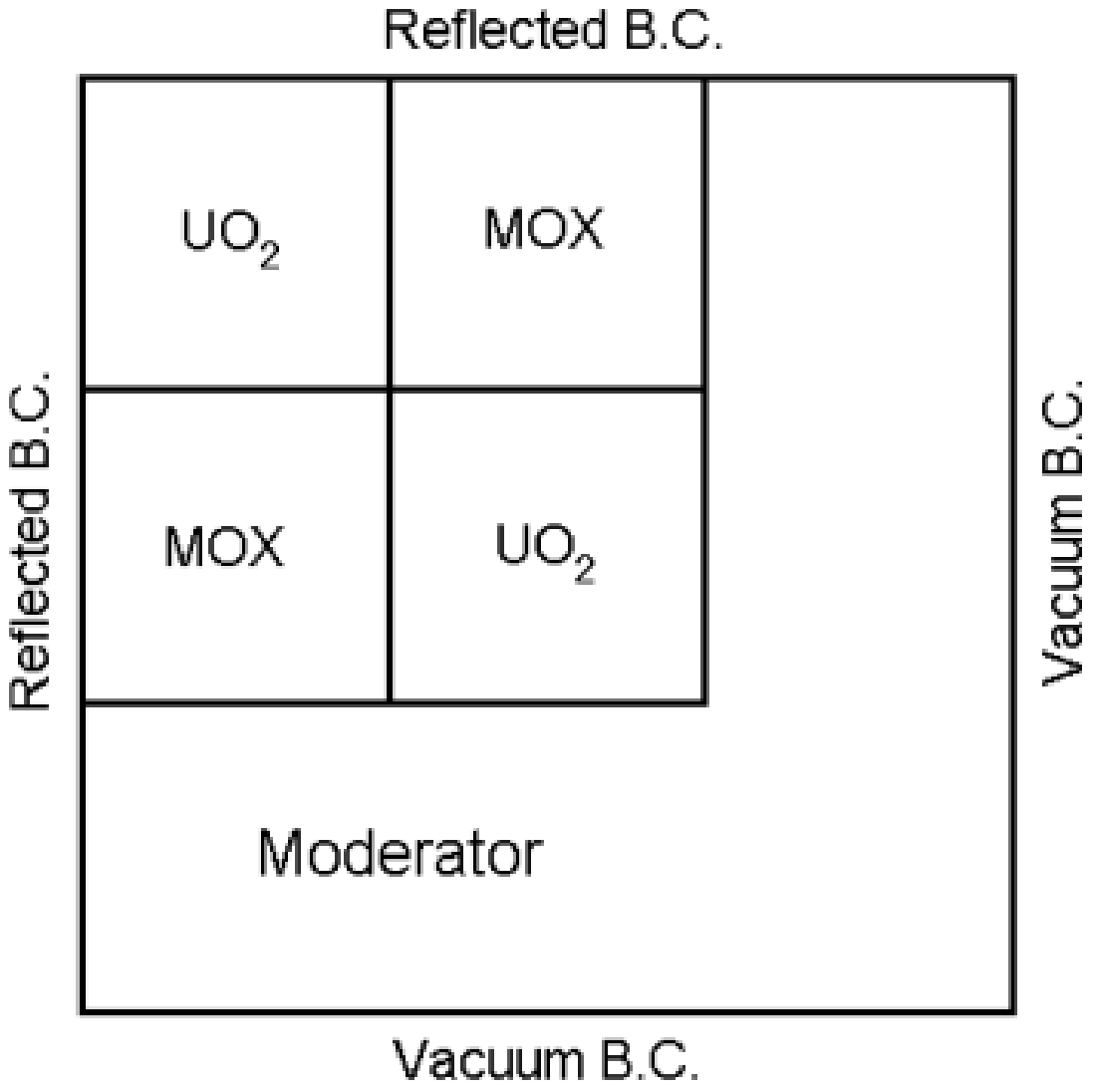

Figure 1 Core configuration for the 2D C5G7.

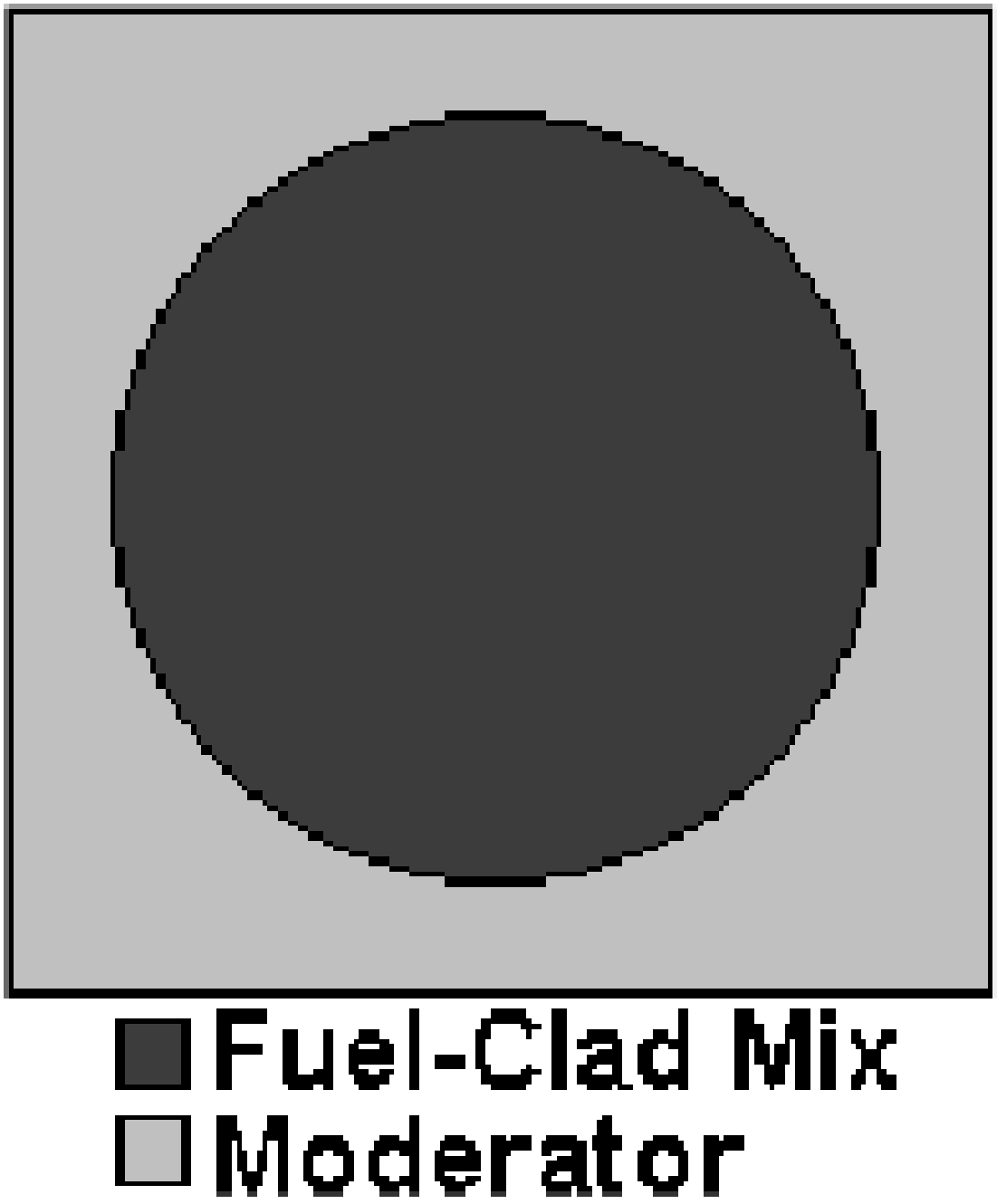

Figure 2 Pin cell geometry for the 2D C5G7.

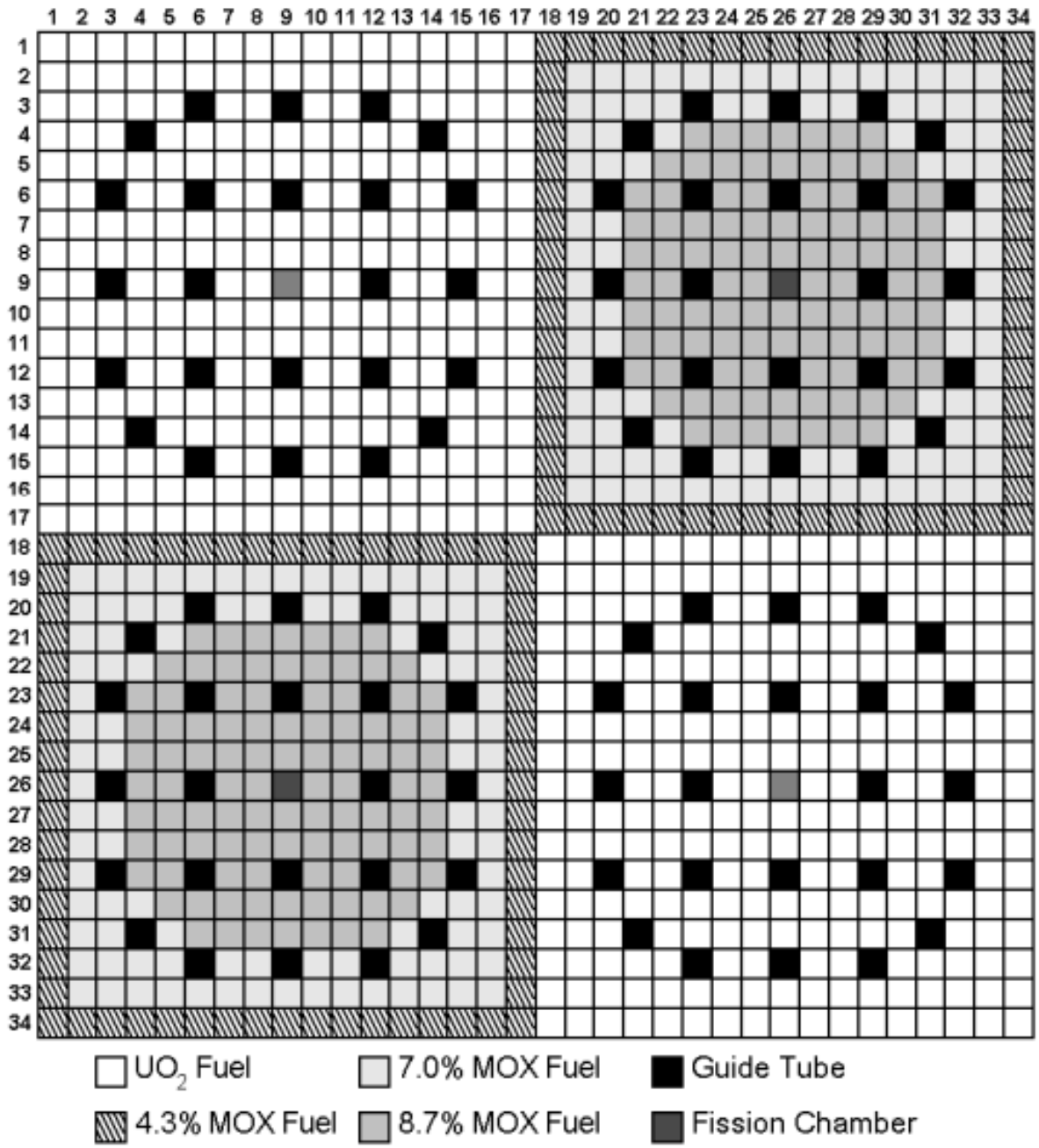

Figure 3 Pin cell compositions for the 2D C5G7.

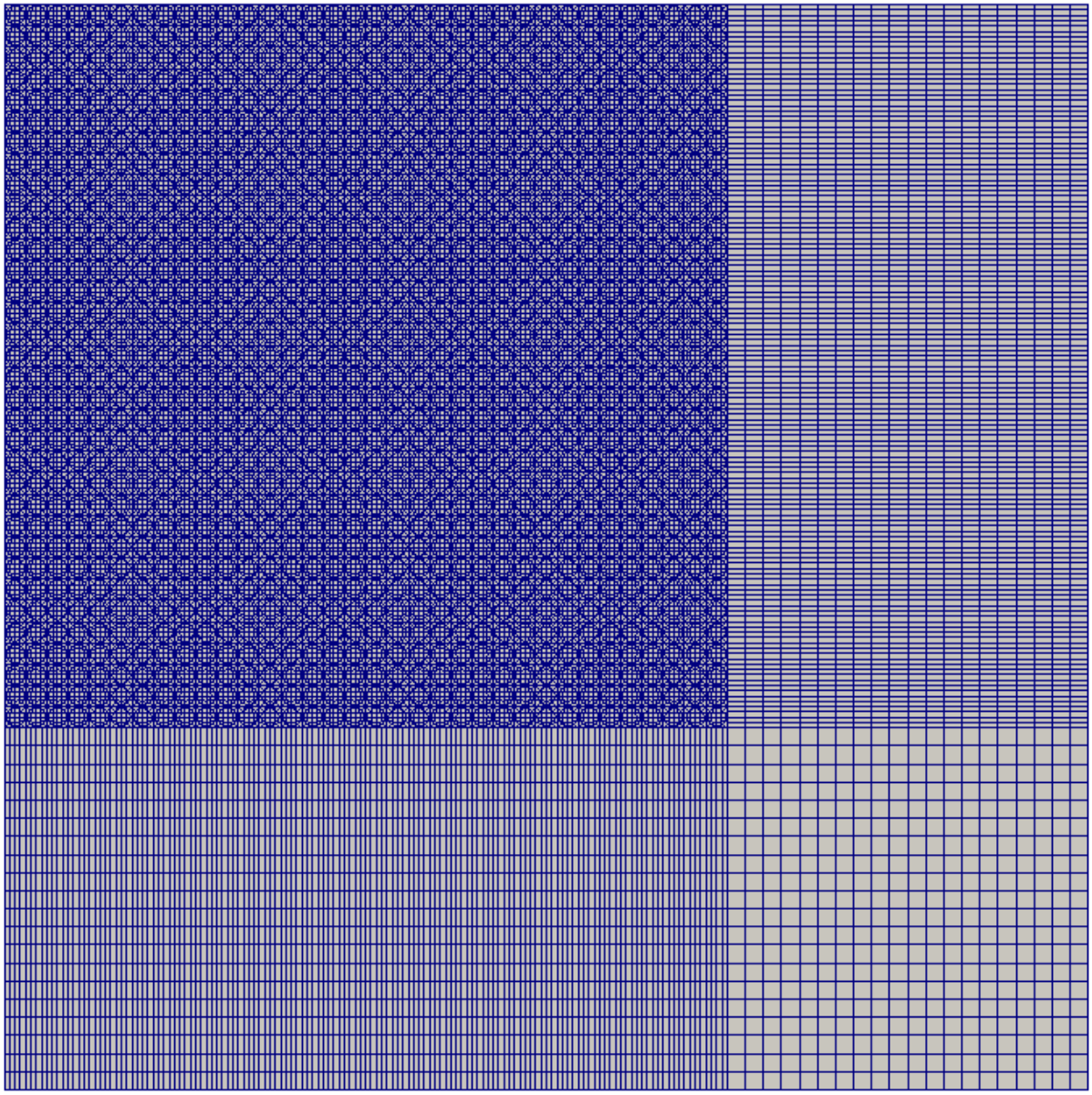

Figure 4 Mesh for the 2D C5G7.

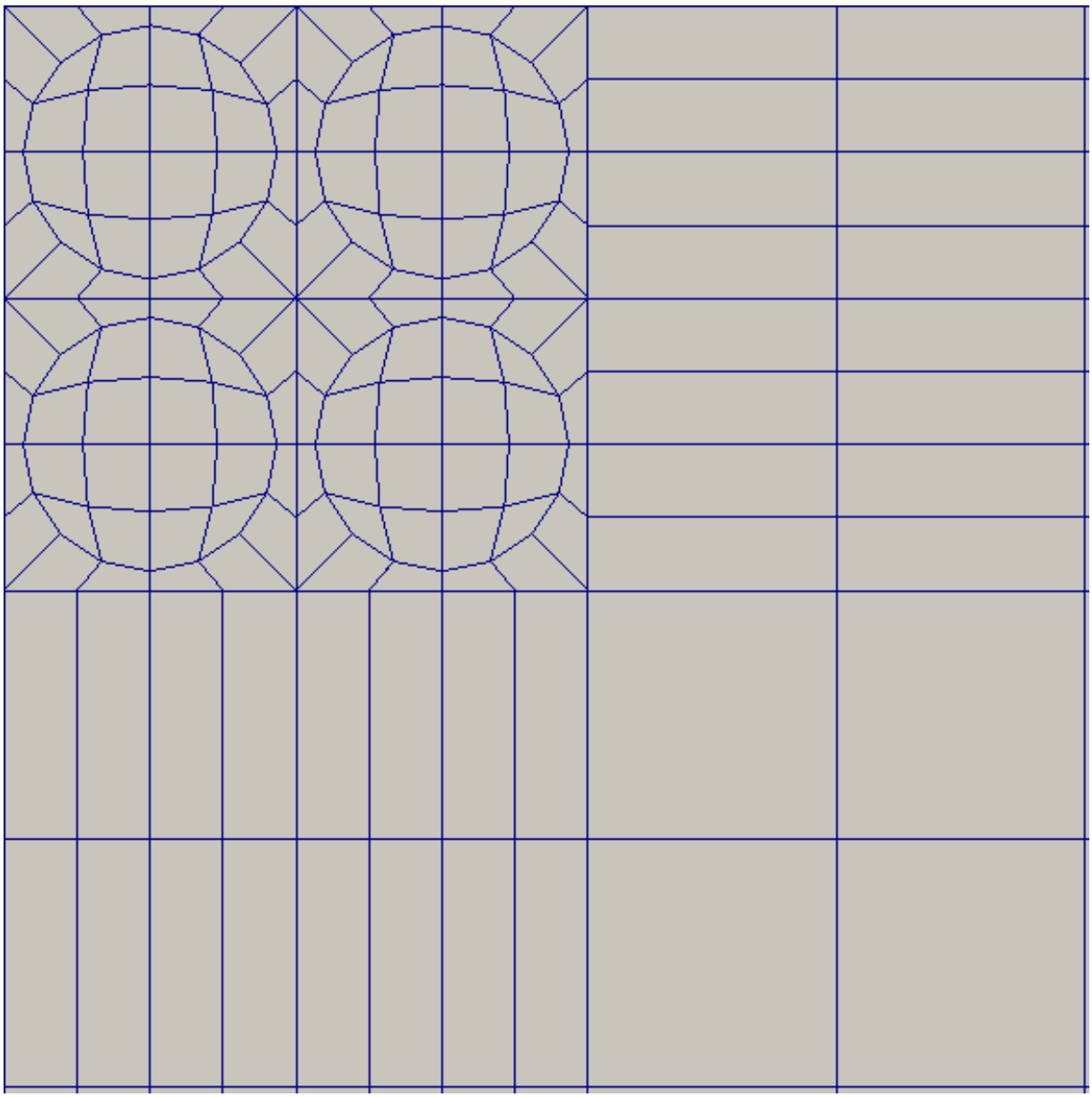

Figure 5 Zoom of the mesh for the 2D C5G7.

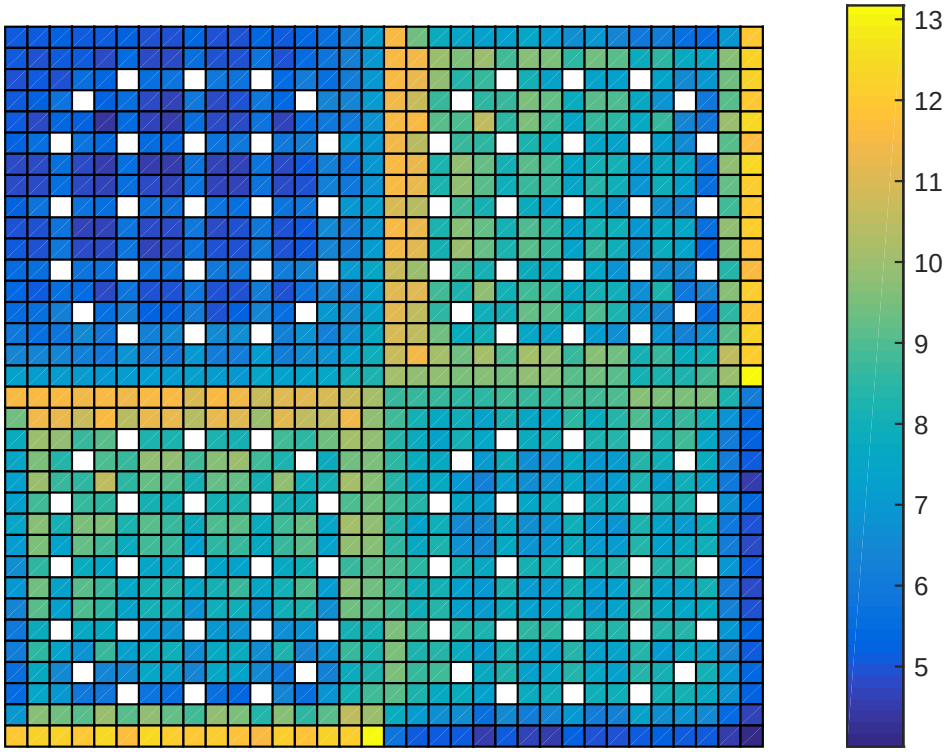

Figure 6 Power errors for the 2D C5G7.

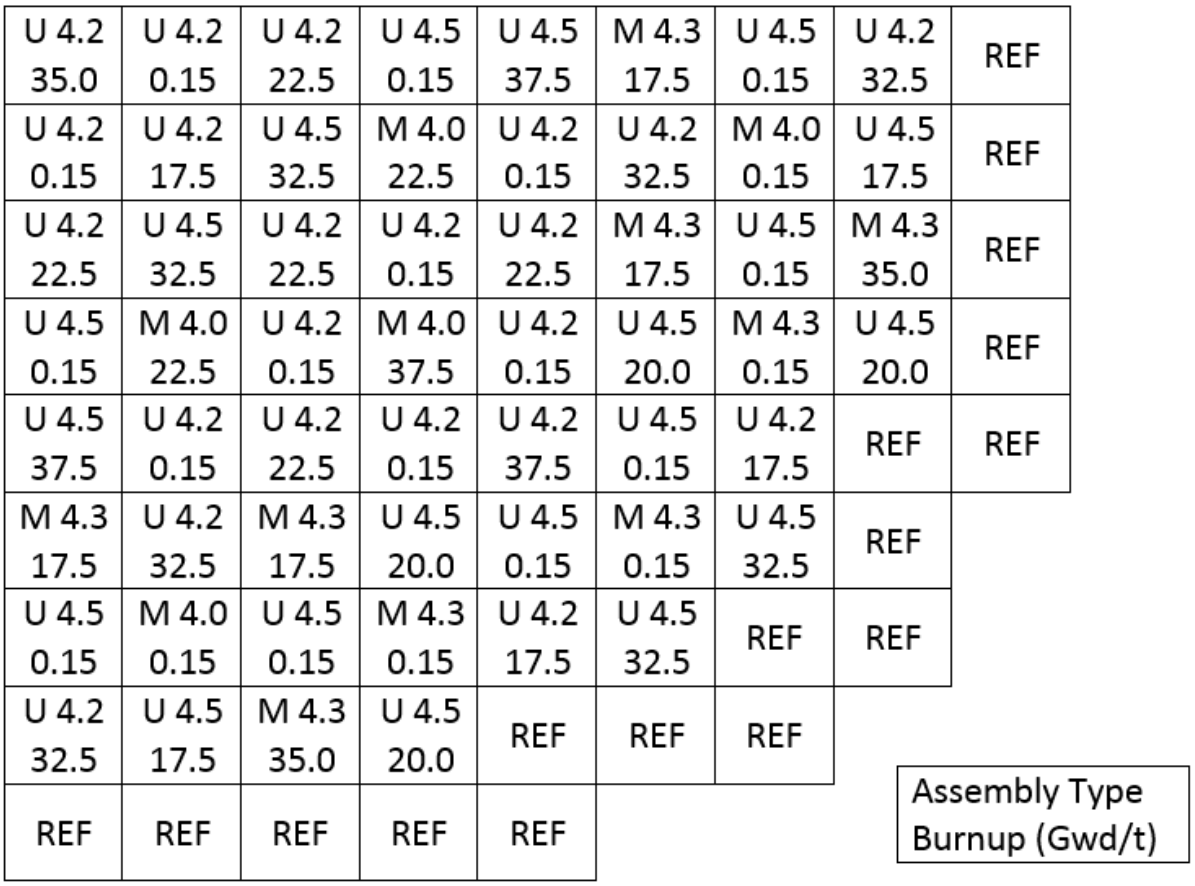

Figure 7 A quarter of the core for the PWR MOX/UO_2.

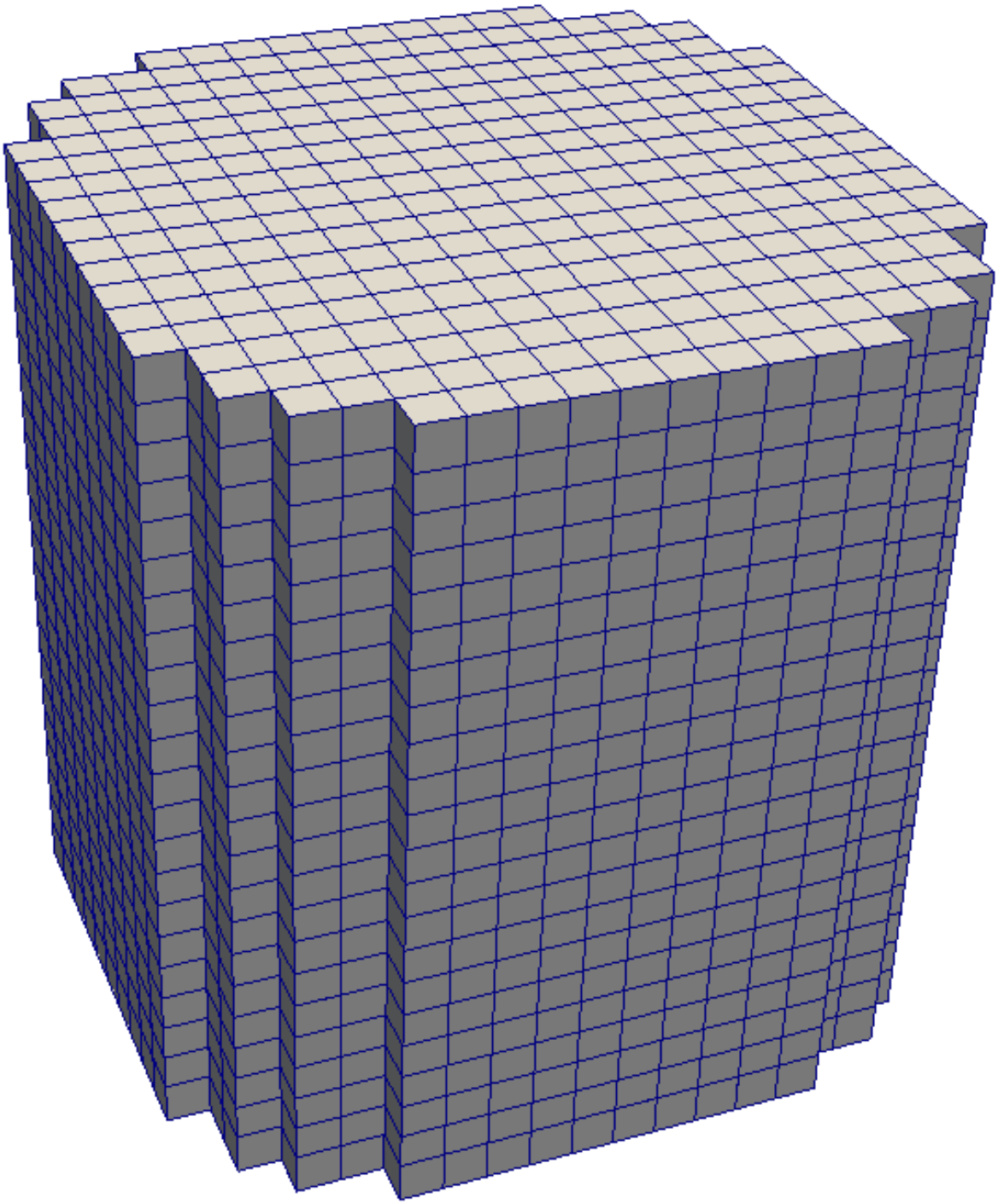

Figure 8 Mesh for the PWR MOX/UO_2.

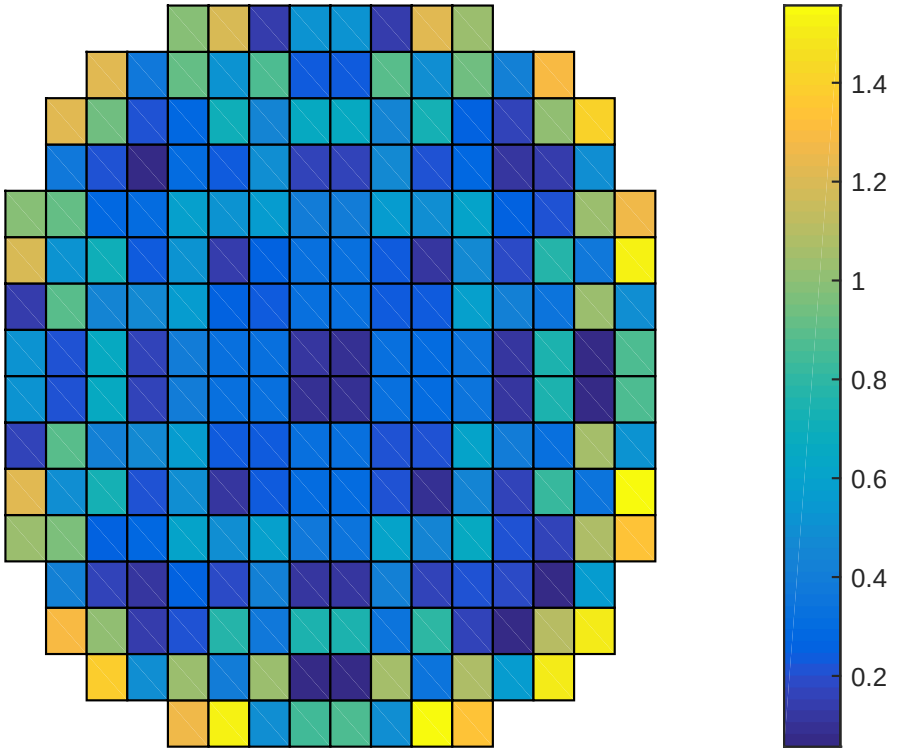

Figure 9 Assembly power errors for the PWR MOX/UO_2 and Mesh 3.

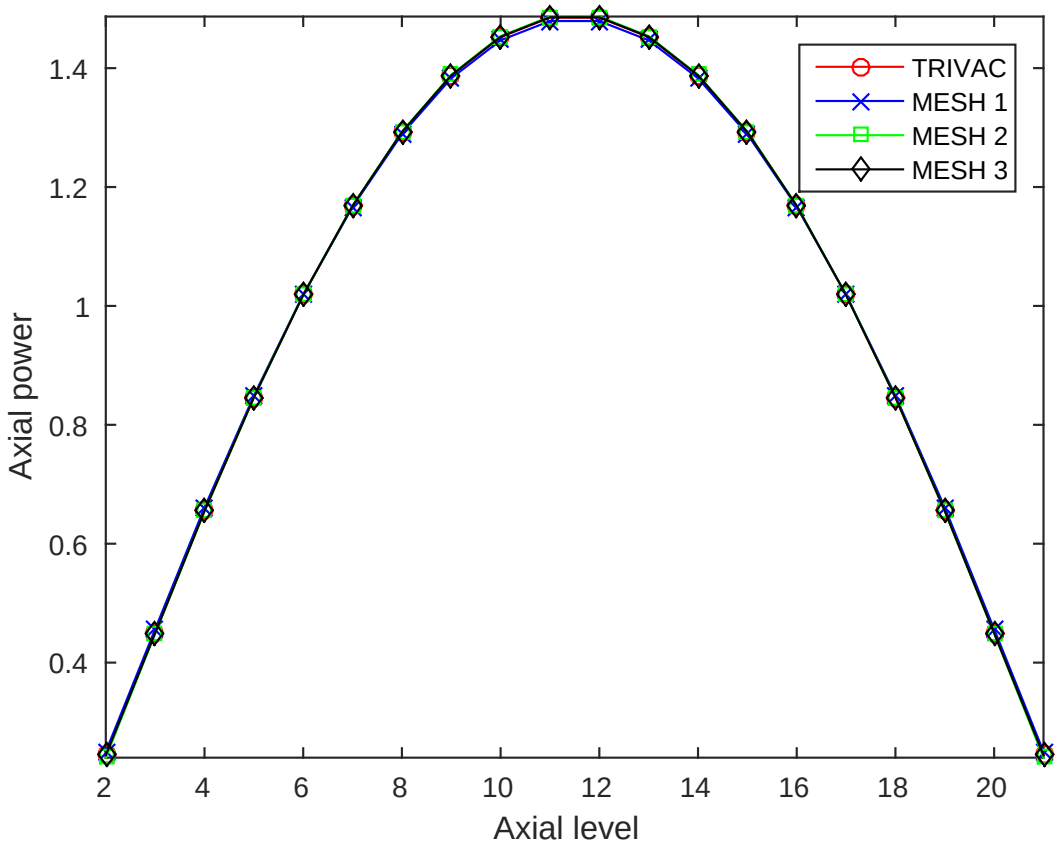

Figure 10 Axial power for the PWR MOX/UO_2.