

Document downloaded from:

<http://hdl.handle.net/10251/101845>

This paper must be cited as:



The final publication is available at

<https://doi.org/10.1080/00223131.2017.1344577>

Copyright Taylor & Francis

Additional Information

A Krylov-Schur solution of the eigenvalue problem for the neutron diffusion equation discretized with the Raviart-Thomas method

Álvaro Bernal^{1*}, Alain Hébert², Jose E Roman³, Rafael Miró¹ and Gumersindo Verdú¹

¹*Institute for Industrial, Radiophysical and Environmental Safety, Universitat Politècnica de València, Valencia 46022, Spain;* ²*Institut de Génie Nucléaire, École Polytechnique de Montréal, Montréal H3C*

3A7, Canada; ³*Department of Information Systems and Computations, Universitat Politècnica de València, Valencia 46022, Spain*

Mixed-dual formulations of the finite element method were successfully applied to the neutron diffusion equation, such as the Raviart-Thomas method in Cartesian geometry and the Raviart-Thomas-Schneider in hexagonal geometry. Both methods obtain system matrices which are suitable for solving the eigenvalue problem with the preconditioned power method. This method is very fast and optimized, but only for the calculation of the fundamental mode. However, the determination of non-fundamental modes is important for modal analysis, instabilities and fluctuations of nuclear reactors. So, effective and fast methods are required for solving eigenvalue problems. The most effective methods are those based on Krylov subspaces projection combined with restart, such as Krylov-Schur. In this work, a Krylov-Schur method has been applied to the neutron diffusion equation, discretized with the Raviart-Thomas and Raviart-Thomas-Schneider methods.

Keywords: *neutron diffusion equation; finite element method; Krylov-Schur; Raviart-Thomas; reactor physics*

1. Introduction

The neutron diffusion equation is the easiest and fastest way to determine the neutron flux distribution inside nuclear reactors, but it requires the use of numerical methods. A lot of methods have been applied to the diffusion equation, but the finite element method has two interesting properties. First, it can be applied to any geometry. Second, it is a consistent discretization technique[1]. As regards finite element formulations, mixed-dual formulations are particularly interesting, since they have many of the characteristics of nodal techniques, which are more computationally efficient than classical finite elements formulation. These mixed-dual formulations were successfully applied to the neutron diffusion equation, such as the Raviart-Thomas method in Cartesian geometry[1] and the Raviart-Thomas-Schneider in hexagonal geometry[2]. Both methods obtain system matrices compatible with the Alternating Direction Implicit (ADI) procedure, which made them suitable for solving the eigenvalue problem with the preconditioned power method[3]. These methods are implemented in TRIVAC code.

TRIVAC[4] is a computer code intended to compute the neutron flux in a fractional or in a full core representation of a nuclear reactor. It can solve the multigroup and multi-dimensional form of the diffusion equation or simplified Pn equation. Moreover, it allows the discretization of 1-D geometries (slab and cylindrical), 2-D geometries (Cartesian, cylindrical and hexagonal) and 3-D geometries (Cartesian and hexagonal).

The preconditioned power method implemented in TRIVAC is very fast and optimized, but only for the calculation of the fundamental mode. However, the determination of non-fundamental modes is important for modal analysis, instabilities and fluctuations in nuclear reactors[5, 6]. In fact, stability analysis calculation may require the calculation of several modes (a minimum of five). Therefore, fast, effective and accurate methods are required. On the one hand, the current version of TRIVAC uses the Hotelling deflation

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

technique for calculating these modes. This technique is a decontamination operation of the power method so as to converge on non-fundamental modes of the eigenvalue problem[7]. The advantage of this technique is the use of the optimized power method, but its convergence rate is not as fast as that of the fundamental mode. In addition, this technique cannot calculate those modes belonging to a multiple eigenvalue, because convergence depends on the magnitude of the eigenvalue ratio being calculated and the next one[7].

Although several eigenvalue problems can be defined, the eigenvalue problem most commonly used for commercial reactors is the λ -eigenvalue problem. The α -eigenvalue problem is also relevant, but for sub-critical systems, like accelerator driven sub-critical systems (ADS). Several authors used more effective methods than the Hotelling deflation technique, for calculating non-fundamental modes of these eigenvalue problems. Examples of these methods are the sub-space iteration method (SSI) and the implicit restarted Arnoldi method (IRAM).

Döring et al. [8], Verdú et al. [5] and Modak and Jain [9] introduced SSI for calculating the λ -eigenvalue problem of the neutron diffusion equation discretized with different methods: finite element method (FEM), nodal collocation method (NCM) and finite difference method (FDM). Singh et al. [10] applied SSI to the neutron diffusion equation for the α -eigenvalue problem. Other works applied SSI to the α -eigenvalue problem of the transport equation using discrete ordinates, but with different spatial methods: Gupta and Mudak used diamond differences (DD) [11] and Kópházi and Lathouwers used FEM [12].

Verdú et al. applied IRAM to the neutron diffusion equation for the α -eigenvalue problem [13]. This method was also used in the α -eigenvalue problem of the transport equation discretized with different methods: Lathouwers [14] used spherical harmonics and FEM, Kópházi and Lathouwers [12] used discrete ordinates and FEM. As it was mentioned

earlier, Kópházi and Lathouwers tested both methods IRAM and SSI in [12], and they concluded that the first one was superior in terms of computational time. Other works applied IRAM to λ -eigenvalue problem: Warsa et al. [15] solved the transport equation and Verdú et al. [16] solved the diffusion equation discretized with NCM.

IRAM belongs to a class of methods called Krylov subspace methods. There are several software and libraries containing the algorithm of Krylov subspace methods, which have been widely used, like ARPACK [17]. Several neutron codes used ARPACK for calculating the eigenvalue problem, such as DALTON [18]. Currently, the state of the art for calculating eigenvalue problems is the SLEPc library. SLEPc, the Scalable Library for Eigenvalue Problem Computations [19, 20], 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) [21] and extends it with all the functionality necessary for the solution of eigenvalue problems, which includes matrix operation and solution of linear systems.

In this work, a Krylov-Schur method has been applied to the λ -eigenvalue problem of the neutron diffusion equation, discretized with the Raviart-Thomas and Raviart-Thomas-Schneider method. An algorithm has been developed combining the Krylov-Schur method from SLEPc and the Raviart-Thomas and Raviart-Thomas-Schneider method from TRIVAC, extending the capabilities of it. The novelty of this work is not only the use of SLEPc in TRIVAC, but also an analysis of four issues. First, the condition number of the system matrices obtained with the Raviart-Thomas method. Second, the preconditioners and linear system solvers of PETSc applied to these matrices. Third, the type of integration used in the Raviart-Thomas method of TRIVAC and its effect on the condition number. Fourth, a comparison of the convergence of the Hotelling deflation technique and the Krylov-Schur method.

The outline of the paper is as follows. Section 2 describes the theory and methods discretizing the neutron diffusion equation and solving the eigenvalue problem. Section 3 defines the reactors used for validating the method and shows the results. Section 4 summarizes the conclusions.

2. Theory

2.1. Dual variational formulation

The one-speed formulation of the steady state neutron diffusion equation and the Fick law can be written as in Equations (1) and (2). In these equations, $\phi(r)$ is the neutron flux, $\vec{J}(r)$ is the neutron current, $D(r)$ is the third-order diagonal matrix containing directional diffusion coefficients, $\Sigma_r(r)$ is the removal cross section, and $S(r)$ is the fission and out-of-group scattering source.

$$\nabla \cdot \vec{J}(r) + \Sigma_r(r)\phi(r) = S(r) \quad (1)$$

$$\vec{J}(r) = -D(r)\vec{\nabla}\phi(r) \quad (2)$$

Boundary conditions are applied on the surface surrounding the domain (∂V), which is split into three components: the surface with a zero flux boundary condition (∂V_0), the surface with a reflective boundary condition (∂V_1) and the surface with a P_1 albedo boundary condition (∂V_β). This last boundary condition is expressed in Equation (3), where $\vec{N}(r)$ is the normal unit vector pointing out of ∂V_β and $\beta(r)$ is the albedo at r .

$$\left[D(r)\vec{\nabla}\phi(r) \right] \cdot \vec{N}(r) + \frac{1}{2} \frac{1 - \beta(r)}{1 + \beta(r)} \phi(r) = 0 \quad \text{if } r \in \partial V_\beta \quad (3)$$

A dual variational formulation exists such that the diffusion equation is a stationary point of dual functionals in a Sobolev space. A stationary point of these functionals in a

polynomial space will be used as a numerical approximation to the diffusion equation. In this formulation, the neutron current $\vec{J}(r)$ is forced to be continuous across any surface element in the domain, as explained by Lautard in[22]. Trial functions used to represent the neutron flux can therefore be discontinuous across some boundary in the domain. This functional is written in Equation (4), where $\phi(r) \in L^2(V)$ and $\vec{J}(r) \in H_{\partial V_1}(div; V)$, a Sobolev space defined in Equation (5).

$$F_b \left\{ \phi, \vec{J} \right\} = \frac{1}{2} \int_V d^3r \left\{ -\vec{J}(r) \cdot \left[D(r)^{-1} \vec{J}(r) \right] + 2\phi(r) \nabla \cdot \vec{J}(r) + \Sigma_r(r) \phi(r)^2 - 2\phi(r) S(r) \right\} - \int_{\partial V_\beta} d^2r \frac{1 + \beta(r)}{1 - \beta(r)} \times \left[\vec{J}(r) \cdot \vec{N}(r) \right]^2 \quad (4)$$

$$H_{\partial V_1}(div; V) = \left\{ v; v \in [L^2(V)]^3 \quad \text{and} \quad \nabla \cdot v \in L^2(V) \right\} \quad (5)$$

This Sobolev space contains functions with L^2 -integrable components over domain V , which possess an L^2 -integrable divergence over domain V and a scalar product $\vec{J}(r) \cdot \vec{N}(r)$ equal to zero on ∂V_1 for a unit vector $\vec{N}(r)$ normal to ∂V_1 . If a function $\vec{J}(r)$ is an element of this Sobolev space, the function $\vec{J}(r)$ has continuous components normal to any interface in V .

The functional of Equation (4) has a stationary point defined by the set of Equations (6). From this set, Equations (7) and (8) are obtained.

$$\begin{aligned} \delta_{\delta \vec{J}} F_b &= \lim_{\epsilon \rightarrow 0} \left\{ \frac{d}{d\epsilon} F_b \left\{ \phi(r), \vec{J}(r) + \epsilon \delta \vec{J}(r) \right\} \right\} = 0 \\ \delta_{\delta \phi} F_b &= \lim_{\epsilon \rightarrow 0} \left\{ \frac{d}{d\epsilon} F_b \left\{ \phi(r) + \epsilon \delta \phi(r), \vec{J}(r) \right\} \right\} = 0 \end{aligned} \quad (6)$$

$$\begin{aligned} \delta_{\delta \vec{J}} F_b &= \int_V d^3r \left\{ -\delta \vec{J}(r) \cdot \left[D(r)^{-1} \vec{J}(r) \right] + \phi(r) \nabla \cdot \delta \vec{J}(r) \right\} \\ &\quad - \int_{\partial V_\beta} d^2r \frac{1 + \beta(r)}{1 - \beta(r)} \left[\vec{J}(r) \cdot \vec{N}(r) \right] \left[\delta \vec{J}(r) \cdot \vec{N}(r) \right] = 0 \end{aligned} \quad (7)$$

$$\delta_{\delta\phi} F_b = \int_V d^3r \delta\phi(r) \left\{ \nabla \cdot \vec{J}(r) + \Sigma_r(r)\phi(r) - S(r) \right\} = 0 \quad (8)$$

Equations (7) and (8) are the stationary conditions and the ones used with the Raviart-Thomas method. These conditions are to be distinguished from the forced conditions, such as the reflective boundary condition on ∂V_1 or the $\vec{J}(r) \cdot \vec{N}(r)$ continuity, which are imposed on the trial functions.

2.2. The Raviart-Thomas method

The Raviart-Thomas method is a finite element method based on a search of a stationary point $[\phi(r), \vec{J}(r)]$ of Equations (7) and (8) in a Raviart-Thomas polynomial subspace of $L^2(V) \times H_{\partial V_1}(\text{div}; V)$. This stationary point will not be the exact solution, but the optimal approximation (in the variational sense) of the diffusion equation, which is an element of this subspace.

The finite element method can be applied to various types and forms of subvolumes or elements. Cartesian and hexagonal elements are the most widely used in reactor physics for full-core calculations. A Cartesian domain is first partitioned into parallelepipeds over which the nuclear properties are assumed to be uniform. An hexagonal domain is partitioned into lozenges and a Piola transformation is performed, as depicted in Figure 1 and described by Hébert in[2]. The reference finite element is a unit cube with $-1/2 \leq u \leq 1/2$, $-1/2 \leq v \leq 1/2$ and $-1/2 \leq w \leq 1/2$. A polynomial basis is defined over each element by using full tensorial products of 1D polynomials up to a given order.

[Figure 1 about here.]

The Raviart-Thomas polynomial basis proposed in this work uses tensorial products of Legendre polynomials for representing the neutron flux over the reference element. Polynomial expressions of the Legendre polynomials $\tilde{P}_k(u)$ defined over $-1/2 \leq u \leq 1/2$ are given in Appendix A in[2]. An order K expansion of ϕ and $\delta\phi$ is shown in Equations

(9) and (10).

$$\phi(u, v, w) = \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} \sum_{k_3=0}^{K-1} \phi_{k_1, k_2, k_3} \tilde{P}_{k_1}(u) \tilde{P}_{k_2}(v) \tilde{P}_{k_3}(w) \quad (9)$$

$$\delta\phi(u, v, w) = \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} \sum_{k_3=0}^{K-1} \delta\phi_{k_1, k_2, k_3} \tilde{P}_{k_1}(u) \tilde{P}_{k_2}(v) \tilde{P}_{k_3}(w) \quad (10)$$

The expansion of the currents proposed in this work uses tensorial products of Lagrange and normalized Legendre polynomials, in order to be consistent with the choice made for the flux and force the continuity of $\vec{J}(r) \cdot \vec{N}(r)$. Polynomial expressions of the Lagrange polynomials $L_k(u)$ defined over $-1/2 \leq u \leq 1/2$ are also given in Appendix A in[2]. An order K expansion of J_u , δJ_u and J_v is shown in Equations (11)-(13). Similar expressions are obtained for δJ_v , J_w and δJ_w .

$$J_u(u, v, w) = \sum_{k_1=0}^K \sum_{k_2=0}^{K-1} \sum_{k_3=0}^{K-1} J_{u, k_1, k_2, k_3} L_{k_1}(u) \tilde{P}_{k_2}(v) \tilde{P}_{k_3}(w) \quad (11)$$

$$\delta J_u(u, v, w) = \sum_{k_1=0}^K \sum_{k_2=0}^{K-1} \sum_{k_3=0}^{K-1} \delta J_{u, k_1, k_2, k_3} L_{k_1}(u) \tilde{P}_{k_2}(v) \tilde{P}_{k_3}(w) \quad (12)$$

$$J_v(u, v, w) = \sum_{k_1=0}^{K-1} \sum_{k_2=0}^K \sum_{k_3=0}^{K-1} J_{v, k_1, k_2, k_3} \tilde{P}_{k_1}(u) L_{k_2}(v) \tilde{P}_{k_3}(w) \quad (13)$$

Equations (7) and (8) can be written as Equations (14) and (15). More explanations about the integration of these terms can be found in[1, 2], for Cartesian and hexagonal geometry. In this work, the authors used the following methods of integration: analytical integration, Gauss-Lobatto quadrature and Gauss-Legendre quadrature. More details of these two quadratures can be found in Appendix A in[2].

$$\delta_{\delta\vec{J}}F_b = \langle \delta\vec{J}, A\vec{J} \rangle + \langle \delta\vec{J}, R\phi \rangle = 0 \quad (14)$$

$$\delta_{\delta\phi}F_b = \langle \delta\phi, R\vec{J} \rangle + \langle \delta\phi, T\phi \rangle - \langle \delta\phi, S\phi \rangle = 0 \quad (15)$$

If one considers the G -group energy approach, the source distribution for each energy group g is calculated as in Equation (16). In this equation, $\phi_i(r)$ is the i -energy group flux, $\Sigma_{s,i \rightarrow g}(r)$ is the scattering cross section from i to g energy group, $\nu\Sigma_{f,i}(r)$ is the production cross section due to fission of i -energy group flux, $\chi_g(r)$ is the fission spectrum for the g energy group and k_{eff} is the multiplication factor.

$$S_g(r) = \sum_{\substack{i=1 \\ i \neq g}}^G \Sigma_{s,i \rightarrow g}(r)\phi_i(r) + \frac{1}{k_{eff}}\chi_g(r) \sum_{i=1}^G \nu\Sigma_{f,i}(r)\phi_i(r) \quad (16)$$

Equations (15) and (14) can be written in matrix form as in Equation (17) or (18), for each g energy group. In Equation (18), Φ_g is the combination of ϕ_g and \vec{J}_g as shown in Equation (19). Finally, if one considers all the energy groups, one obtains the eigenvalue problem expressed in Equation (20), with \mathbb{A} , \mathbb{B} and Φ defined in Equations (21)-(23).

$$\begin{pmatrix} T_g & R^T \\ R & A_g \end{pmatrix} \begin{pmatrix} \phi_g \\ \vec{J}_g \end{pmatrix} = \sum_{\substack{i=1 \\ i \neq g}}^G \begin{pmatrix} S_{s,i \rightarrow g} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \phi_i \\ \vec{J}_i \end{pmatrix} + \frac{1}{k_{eff}} \sum_{i=1}^G \begin{pmatrix} S_{f,g,i} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \phi_i \\ \vec{J}_i \end{pmatrix} \quad (17)$$

$$\mathbb{A}_{g,g}\Phi_g = \sum_{\substack{i=1 \\ i \neq g}}^G \mathbb{A}_{g,i}\Phi_i + \frac{1}{k_{eff}} \sum_{i=1}^G \mathbb{B}_{g,i}\Phi_i \quad (18)$$

$$\Phi_g = \begin{pmatrix} \phi_g \\ \vec{J}_g \end{pmatrix} \quad (19)$$

$$\mathbb{A}\Phi = \frac{1}{k_{eff}}\mathbb{B}\Phi \quad (20)$$

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{1,1} & -\mathbb{A}_{1,2} & \cdots & -\mathbb{A}_{1,G} \\ -\mathbb{A}_{2,1} & \mathbb{A}_{2,2} & \ddots & -\mathbb{A}_{2,G} \\ \vdots & \ddots & \ddots & \vdots \\ -\mathbb{A}_{G,1} & -\mathbb{A}_{G,2} & \cdots & \mathbb{A}_{G,G} \end{pmatrix} \quad (21)$$

$$\mathbb{B} = \begin{pmatrix} \mathbb{B}_{1,1} & \cdots & \mathbb{B}_{1,G} \\ \vdots & \vdots & \vdots \\ \mathbb{B}_{G,1} & \cdots & \mathbb{B}_{G,G} \end{pmatrix} \quad (22)$$

$$\Phi = \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_G \end{pmatrix} \quad (23)$$

The eigenvalue problem of Equation (20) has been solved efficiently by Hébert in[1,2], by using a symmetric variational acceleration technique (SVAT). This techniques is a preconditioned power method explained in[3], which uses an ADI scheme for Cartesian and hexagonal geometries. The Cartesian scheme is explained in[1] and the hexagonal one in[2].

2.3. *The Hotelling deflation technique*

The Hotelling deflation technique is a decontamination operation of the power method so as to converge on non-fundamental harmonics of the eigenvalue problem[7]. A basic algorithm will be explained in the context of the inverse power method, which consist in computing one harmonic at a time, while decontaminating \mathbb{A}^{-1} from the previously calculated harmonics.

To compute eigenvalues and eigenvectors of the eigenvalue problem of Equation (20), the inverse power recurrence of Equation (24) is employed. This recurrence is used only for the first eigenvalue; once the fundamental mode has converged, subsequent modes are obtained with a modified recurrence, Equation (25), that includes the decontamination

operation that purges the components that the current iterate $\Phi_l^{(n+1)}$ has in the direction of already converged eigenvectors.

$$\Phi_l^{(n+1)} = \frac{1}{k_l^{(n)}} \mathbb{A}^{-1} \mathbb{B} \Phi_l^{(n)} \quad (24)$$

$$\Phi_l^{(n+1)} = \frac{1}{k_l^{(n)}} \left(\mathbb{A}^{-1} - \sum_{j=1}^{l-1} k_j \frac{\Phi_j \cdot \Phi_j^*}{\Phi_j^* \cdot \mathbb{B} \Phi_j} \right) \mathbb{B} \Phi_l^{(n)} \quad (25)$$

2.4. The Krylov-Schur method

The Krylov-Schur method is an Arnoldi method which uses an implicit restart based on a Krylov-Schur decomposition[23].

The method of Arnoldi is a Krylov-based projection method that computes an orthonormal basis of the Krylov subspace of order m associated with matrix \mathcal{A} and initial vector x_0 . This Krylov subspace is given in Equation (26). Projection methods for eigenvalue problems are intended for computing a partial eigensolution, that is, given a square matrix \mathcal{A} of order N , the objective is to compute a small number of eigenpairs, $\lambda_i, x_i, i = 1, \dots, m$, with $m \ll N$. The Arnoldi method computes not only this orthonormal basis (V_m), but also the projected matrix H at the same time and in an efficient and numerically stable way.

$$\mathcal{K}_m(\mathcal{A}, x_0) = \text{span} \{x_0, \mathcal{A}x_0, \mathcal{A}^2x_0, \dots, \mathcal{A}^{m-1}x_0\} \quad (26)$$

This projection method calculates the eigenvalue problem $Hy_i = \theta_i y_i$, of order m , instead of $\mathcal{A}x_i = \lambda_i x_i$, of order N . Taken into account that ($H = V_m^T \mathcal{A} V_m$) and ($V_m^T V_m = I_m$), one concludes that the pair $(\lambda_i, V_m y_i)$ can be taken as an approximation of the eigenpair (λ_i, x_i) of matrix \mathcal{A} . This method will converge very fast, if the initial vector x_0 is rich in the direction of the wanted eigenvectors, which is usually not the case. So,

many iterations may be required, which implies a growth in storage requirements and computational time. A solution for this problem is to stop after some iterations and restart the method, by using a new initial vector computed from the recently obtained spectral approximations.

Different approaches can be used for the restart: explicit and implicit. Explicit algorithms calculate the initial vector as a linear combination of the computed eigenvectors, but it is difficult to choose the appropriate parameters. Implicit algorithms combine the Arnoldi process with the implicitly shifted QR algorithm, in which an m -step Arnoldi factorization is compacted into an $(m - d)$ -step Arnoldi factorization, which retains the relevant eigeninformation of the large factorization. The implementation of the implicit restart in a numerically stable way is difficult, but it is solved by using a Krylov-Schur decomposition. More information about this decomposition can be found in[23].

In this work, the authors have applied the Krylov-Schur algorithm implemented in SLEPc[19, 20] to the Raviart-Thomas method of TRIVAC[4]. The system matrices of Equation (20) are obtained with the Raviart-Thomas method and two types of eigenvalue problems are considered. The first type, without upscattering and producing only neutrons in the first energy group from fissions, such as Equation (27). In this case, the eigenvalue problem is defined in Equation (28), where ϕ_1 is the iterative eigenvector and ϕ_g , for $g > 1$, are calculated with Equation (29). The second type includes any upscattering and fission production, and is defined in Equation (30). It is important to highlight that the inverse of the matrices ($\mathbb{A}_{g,g}$ or \mathbb{A}) are not calculated, but linear systems are solved: $x = \mathbb{A}^{-1}b \rightarrow \mathbb{A}x = b$. These linear systems are solved by using iterative or direct solvers, depending on the condition number of the matrices. Iterative solvers are faster than direct solvers, if the matrices are not ill-conditioned. In both cases, the authors have tried different methods implemented in PETSc[21]. As regards iterative solvers, the fastest methods were Generalized Minimal Residual (GMRES) and BiConjugate Gradient

Stabilized (BiCGSTAB), using the incomplete LU preconditioner. With respect to direct solvers, the authors used LU factorization implemented in MUMPS solver.

The use of iterative or direct solvers depends on the condition number and size of the matrices. For well-conditioned and large matrices, iterative solvers are faster and require less memory resources than direct solvers. By contrast, for matrices with bad condition numbers, iterative solvers might not converge, so one should use direct solvers. As a rule of thumb, one should use iterative solvers at first time, but one should use direct solvers if the iterative ones do not converge.

$$\begin{pmatrix} \mathbb{A}_{1,1} & & 0 \\ \vdots & \ddots & \\ -\mathbb{A}_{G,1} & \cdots & \mathbb{A}_{G,G} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_G \end{pmatrix} = \frac{1}{k} \begin{pmatrix} \mathbb{B}_{1,1} & & \mathbb{B}_{1,G} \\ & 0 & \end{pmatrix} \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_G \end{pmatrix} \quad (27)$$

$$k\phi_1 = \mathbb{A}_{1,1}^{-1} \sum_{g=1}^G \mathbb{B}_{1,g} \phi_g = \mathcal{A}\phi_1 \quad (28)$$

$$\phi_g = \mathbb{A}_{g,g}^{-1} \sum_{i=1}^{g-1} \mathbb{A}_{g,i} \phi_i \quad (29)$$

$$k\Phi = \mathbb{A}^{-1} \mathbb{B}\Phi = \mathcal{A}\Phi \quad (30)$$

3. Results

The authors evaluated the method in two reactors: one with Cartesian geometry and another with hexagonal geometry. The reactor defined in Cartesian geometry is IAEA 3D[24]. The reactor in hexagonal geometry is VV1K3D, which is a Water-Water Energetic Reactor (VVER) mockup. Both reactors do not include upscattering terms, so this section evaluates the solution of the eigenvalue problem of Equation (28). The authors assessed the solution of the eigenvalue problem of Equation (30) in a commercial PWR with

upscattering terms, but they do not include these results due to the extent of them.

Twelve modes were calculated for each case. The Krylov-Schur solution is compared with the Hotelling deflation technique. Only the eigenvalues are shown in this work due to two reasons. The first is to reduce the extent of this paper. The second is that the Raviart-Thomas method is the same in both eigen-solvers, so if the eigenvalues match each other, the eigenvectors will do too.

A sensitivity analysis was performed for the integration type and polynomial order. As mentioned in Section 2.2., the following types of integration were used: analytical integration, Gauss-Lobatto quadrature and Gauss-Legendre quadrature.

The authors also tested several linear system solvers of PETSc. The fastest methods were GMRES and BiCGSTAB, using the incomplete LU preconditioner. They produced the same results and the computational time is similar.

All the CPU time values reported in this work have been obtained on an Intel Core 2 Duo CPU P8700 (2.53GHz) with the CentOS 6.8 operating system.

3.1. IAEA 3D

The IAEA 3D reactor[24] is a simplified representation of a PWR with one-eighth symmetry in Cartesian geometry, as depicted in Figure 2. Although Figure 2 shows one-eighth symmetry of this reactor, the modal calculation was performed for the whole core. Vacuum boundary conditions are set at axial levels $z = 0$ and $z = 380$ cm. A Cartesian mesh of $9 \times 9 \times 4$ was used. The length of each mesh in X and Y direction is 20 cm. The length of the meshes in Z direction is: 20, 260, 80 and 20 cm. Table 1 shows the cross sections of this reactor.

[Figure 2 about here.]

[Table 1 about here.]

Time results are shown in Table 2. In this table, K is the polynomial order, $A_{g,g}$ size

is the size of the system matrices $\mathbb{A}_{g,g}$ and t_i is the time for calculating i eigenvalues. In addition, no entry in t_i means that the calculation could not be performed. Eigenvalue results are displayed in Table 3. In this table, k_i is the i -eigenvalue. Likewise, no entry in k_i means that the calculation could not be performed. This table only shows the first three eigenvalues.

The authors decided to give only the 12 largest eigenvalues calculated for one integration type and polynomial order; in particular, for the analytical integration and polynomial order 3. The twelve eigenvalues calculated with the Krylov-Schur method for this case are: 1.028910, 1.014819, 1.002363, 0.994373, 0.990582, 0.990280, 0.977587, 0.971420, 0.965838, 0.956518, 0.943951 and 0.938522.

[Table 2 about here.]

[Table 3 about here.]

From Table 2, one can draw three conclusions. First, the computational times are similar for the calculation of three eigenvalues or less. Second, the computational time for the calculation of five or twelve eigenvalues is higher for the Hotelling deflation technique, because of the poor convergence of this technique. In fact, the authors had to increase the inner iterations for this technique to calculate five or more eigenvalues, which increased the computational time. Third, the Hotelling deflation technique only calculated the twelve eigenvalues for two cases. From Table 3, one can see that the third eigenvalue might differ in some cases. This is due to the fact that this Hotelling deflation technique could not calculate the largest eigenvalues in the correct order. Therefore, this technique does not guarantee the calculation of the largest eigenvalues.

Another important issue is the quality of the system matrices. This matters for using iterative methods for solving linear systems. The quality of the matrices for this purpose can be evaluated with the condition number, but its calculation may be costly for large and ill-conditioned matrices. This calculation can be easily performed for the IAEA 3D

reactor, but not for the others presented in this work. The condition number for IAEA 3D are shown in Table 4. The condition numbers of this table are not good, but the matrices are small, so the linear systems can be calculated with iterative methods. In cases with large matrices, the condition numbers will be even worse, so one may use direct solvers instead of iterative ones.

[Table 4 about here.]

Moreover, the authors checked that the bi-orthogonal property $\langle \Phi_i^*, \mathbb{B}\Phi_j \rangle = \delta_{i,j} \cdot C_{i,j}$ is preserved. In the previous expression, $\delta_{i,j}$ is the Kronecker delta and $C_{i,j}$ is a constant. Table 5 shows the values of $\langle \Phi_i^*, \mathbb{B}\Phi_j \rangle$ for the case with analytical integration and polynomial order 3. In this table, the values are displayed in matrix form, in which each row corresponds to index i and each column corresponds to index j .

[Table 5 about here.]

3.2. *VV1K3D*

VV1K3D is a Water-Water Energetic Reactor (VVER) mockup. It is composed of 1690 hexagonal prisms, distributed in 10 axial levels of 20 cm in length. All the hexagonal prisms are regular and their flat-to-flat distance is 23.6 cm. A cross section of the reactor is displayed in Figure 3, in which each number represents an assembly type. Assemblies from 1 to 5 are composed of materials from 1 to 5, respectively. Composition of assembly 6 varies with the axial level: in the first five axial levels is composed of material 4 and in the last ones is composed of material 3. The cross sections of the 5 materials and two energy group are shown in Table 6[25]. Boundary conditions are zero flux for all boundaries.

[Figure 3 about here.]

[Table 6 about here.]

Time and eigenvalues results are shown in Tables 7 and 8, respectively. Time results of Table 7 correspond to the calculation with iterative solvers. Table 7 shows good time

results for the analytical integration and Gaus-Lobatto quadrature, but not for the Gauss-Legendre quadrature. For the Gauss-Legendre quadrature, calculation with direct solvers is faster than calculation with iterative solvers, as it is shown in Table 9.

For the analytical integration, polynomial order 2 and 3 lozenges per hexagon, the 12 largest eigenvalues calculated with the Krylov-Schur method are: 1.005450, 0.987368, 0.987360, 0.968519, 0.964399, 0.963050, 0.954743, 0.948942, 0.948283 and 0.946414, 0.934309, 0.930121.

[Table 7 about here.]

[Table 8 about here.]

[Table 9 about here.]

From Tables 7 and 8, one can draw similar conclusions as those of IAEA3D. In this case, the computational times are similar for the calculation of two eigenvalues or less. As in the previous reactor, the authors increased the inner iterations for the Hotelling deflation technique to calculate three or more eigenvalues, which increased the computational time. In addition, it is important to highlight that the Hotelling deflation technique only calculated the twelve eigenvalues in one case. Finally, Table 8 shows that the third eigenvalue differs in all cases. Therefore, the Hotelling deflation technique might not calculate the largest eigenvalues in the correct order, and consequently cannot guarantee the calculation of the largest eigenvalues.

4. Conclusions

A Krylov-Schur method has been applied to the neutron diffusion equation discretized with the Raviart-Thomas method.

The method was validated in two reactors, one in Cartesian geometry and another in hexagonal. Twelve eigenvalues were calculated in each reactor. The method was compared with the Hotelling deflation technique and the major conclusion is: the Krylov-Schur

method guarantees the calculation of the twelve largest eigenvalues, but not the Hotelling deflation technique.

The authors performed a sensitivity analysis of different type of integration and polynomial order. In this analysis, direct and iterative solvers were used to evaluate the calculation time in ill-conditioned cases. Iterative solvers require less memory resources than direct ones and might be faster in some cases. However, system matrices of some reactors in hexagonal geometry with the Gauss-Legendre quadrature may be ill-conditioned, so iterative solvers could not calculate the linear systems.

As regards future work, additional preconditioners should be developed for accelerating the calculation. Moreover, the Krylov-Schur method will be applied to the simplified P_n equation.

Acknowledgement

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 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, the Universitat Politècnica de València under the project UPPTE/2012/118, and the Spanish Ministerio de Economía y Competitividad under the project TIN2016-75985-P.

References

- [1] Hébert A. Application of a dual variational formulation to finite element reactor calculations. Ann. Nucl. Energy. 1993 Dec; 20:823-845.
- [2] Hébert A. A Raviart-Thomas-Schneider solution of the diffusion equation in hexagonal geometry.

- Ann. Nucl. Energy. 2008 Mar; 35:363-376.
- [3] Hébert A. Preconditioning the Power Method for Reactor Calculations. Nucl. Sci. Eng. 1986 Sep; 94:1-11.
- [4] Hébert A. A user guide for TRIVAC version4. Canada: Institut de génie nucléaire, École Polytechnique de Montréal; 2014, Report no. IGE-2293.
- [5] Verdú G, Ginestar D, Vidal V, Muñoz-Cobo JL. 3D λ -modes of the neutron-diffusion equation. Ann. Nucl. Energy. 1994 Jul; 21:405-421.
- [6] Miró R, Ginestar D, Verdú G, Hennig D. A nodal modal method for the neutron diffusion equation. Application to BWR instabilities analysis. Ann. Nucl. Energy. 2002 Jul; 29:1171-1194.
- [7] Hébert A. Applied Reactor Physics. Second edition. Montréal: Presses Internationales Polytechnique; 2016. p.368-369.
- [8] Döring MG, Kalkkuhl JC, Schröder W. Subspace Iteration for Nonsymmetric Eigenvalue Problems Applied to the λ -Eigenvalue Problem. Nucl. Sci. Eng. 1993 Nov; 115:244-252.
- [9] Modak RS, Jain VK. Sub-space iteration scheme for the evaluation of λ -modes of finite-differenced multi-group neutron diffusion equations. Ann. Nucl. Energy. 1996 Feb; 23:229-237.
- [10] Singh KP, Modak RS, Degweker SB, Kanchhi Singh. Iterative schemes for obtaining dominant alpha-modes of the neutron diffusion equation. Ann. Nucl. Energy. 2009 Aug; 36:1086-1092.
- [11] Gupta A, Modak RS. Evaluation of dominant time-eigenvalues of neutron transport equation by Meyer's sub-space iterations. Ann. Nucl. Energy. 2011 Jul; 38:1680-1686.
- [12] Kópházi J, Lathouwers D. Three-dimensional transport calculation of multiple alpha modes in sub-critical systems. Ann. Nucl. Energy. 2012 Dec; 50:167-174.
- [13] Verdú G, Ginestar D, Roman JE, Vidal V. 3D Alpha Modes of a Nuclear Power Reactor. J. Nucl. Sci. Technol. 2010 May; 47:501-514.
- [14] Lathouwers D. Iterative computation of time-eigenvalues of the neutron transport equation. Ann. Nucl. Energy. 2003 Nov; 30:1793-1806.
- [15] Warsa JS, Wareing TA, Morel JE, McGhee JM, Lehoucq RB. Krylov subspace iterations for deter-

- ministic k-eigenvalue calculations. Nucl. Sci. Eng. 2004 May; 147:26-42.
- [16] Verdú G, Miró R, Ginestar D, Vidal V. The implicit restarted Arnoldi method, an efficient alternative to solve the neutron diffusion equation. Ann. Nucl. Energy. 1999 May; 26:579-593.
- [17] Lehoucq R, Sorensen D, Yang C. ARPACK Users' Guide. Society for Industrial and Applied Mathematics; 1998.
- [18] Boer B, Lathouwers D, Kloosterman J-L, van der Hagen T, Strydom G. Validation of the DALTON-THERMIX code system with transient analyses of the HTR-10 and application to the PBMR. Nucl. Technol. 2010 May; 170:306-321.
- [19] 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.
- [20] Hernandez V, Roman JE, Vidal V. SLEPc: Scalable Library for Eigenvalue Problem Computations. Lect. Notes Comp. Sci. 2003 Apr; 2565:377-391.
- [21] Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Eijkhout V, Groppe 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.
- [22] Lautard, JJ. La méthode nodale de CRONOS: MINOS. Approximation par des éléments finis mixtes-duaux. France: Commissariat à l'Énergie Atomique; 1984, Report no. CEA-N-2763.
- [23] Stewart GW. A Krylov-Schur algorithm for large eigenproblems. SIAM. J. Matrix Anal. Appl. 2002 Mar; 23:601-614.
- [24] Argonne Code Center: Benchmark Problem Book. Argonne National Laboratory; 1977, Report ANL-7416, Suppl. 2.
- [25] Chao YA, Shatilla YA. Conformal mapping and hexagonal nodal methods-II: Implementation in the ANC-H Code. Nucl. Sci. Eng. 1995 Apr; 121:210-225.

Table 1 Cross section data for IAEA 3D

Mixture	Group	$D_g (cm)$	$\Sigma_{g,r} (cm^{-1})$	$\nu\Sigma_{f,g} (cm^{-1})$	$\Sigma_{s,g \rightarrow g+1} (cm^{-1})$
1	1	1.5	0.03	0.0	0.02
	2	0.4	0.08	0.135	
2	1	1.5	0.03	0.0	0.02
	2	0.4	0.085	0.135	
3	1	1.5	0.03	0.0	0.02
	2	0.4	0.13	0.135	
4	1	2.0	0.04	0.0	0.04
	2	0.3	0.01	0.0	
5	1	2.0	0.04	0.0	0.04
	2	0.3	0.055	0.0	

Table 2 Time results (seconds) of IAEA 3D

Integration	K	$\mathbb{A}_{g,g}$ size	Hotelling					Krylov-Schur				
			t_1	t_2	t_3	t_5	t_{12}	t_1	t_2	t_3	t_5	t_{12}
Analytical	1	1449	1	1	1	2		1	1	1	1	1
	2	7176	1	1	3	9	28	1	1	1	2	2
	3	19251	1	2	9	29		2	3	4	5	7
Gauss-Lobatto	1	1449	1	1	1	1	7	1	1	1	1	1
	2	7176	1	1	3	11		1	1	1	2	2
	3	19251	1	2	6	20		2	3	4	4	7
Gauss-Legendre	1	1449	1	1	2			1	1	1	1	1
	2	7176	1	1	3	10		2	2	2	3	4
	3	19251	1	2	9	275		5	6	8	9	12

Table 3 Eigenvalues of IAEA 3D

Integration	K	Hotelling			Krylov-Schur		
		k_1	k_2	k_3	k_1	k_2	k_3
Analytical	1	1.027948	1.009954	0.987201	1.027949	1.009953	0.997614
	2	1.028483	1.013580	0.989918	1.028480	1.013579	1.000138
	3	1.028916	1.014823	0.994372	1.028910	1.014819	1.002363
Gauss-Lobatto	1	1.033093	1.026220	1.008005	1.033096	1.026221	1.008005
	2	1.028077	1.013743	0.993119	1.028069	1.013745	0.998752
	3	1.028841	1.014759	0.994679	1.028841	1.014755	1.001911
Gauss-Legendre	1	1.027215	0.994432	0.989400	1.027216	1.007120	0.994432
	2	1.029287	1.014653	0.991522	1.029285	1.014651	1.002558
	3	1.028981	1.014956	0.994730	1.028980	1.014952	1.002960

Table 4 Condition number of $\mathbb{A}_{g,g}$ of IAEA 3D

Integration	K	Condition number	
		$\mathbb{A}_{1,1}$	$\mathbb{A}_{2,2}$
Analytical	1	509.2	6876.8
	2	394.9	3296.6
	3	966.2	3388.0
Gauss-Lobatto	1	478.0	7286.9
	2	347.6	3085.6
	3	675.6	3122.0
Gauss-Legendre	1	3150.6	11413.0
	2	961.4	3499.7
	3	24410.0	5108.7

Table 5 Bi-orthogonal property for IAEA 3D

7.1E+02	1.6E-11	6.5E-12	1.7E-11	9.4E-13	1.5E-11	1.8E-12	2.0E-12	1.1E-12	3.3E-12	3.5E-13	6.5E-12
1.9E-12	6.4E+02	6.8E-12	3.0E-12	1.5E-12	1.3E-12	5.1E-12	4.9E-12	4.5E-12	5.7E-13	4.2E-13	3.3E-12
3.0E-13	3.5E-11	6.7E+02	5.3E-11	4.4E-11	7.9E-11	1.4E-11	5.8E-12	1.0E-11	3.6E-12	1.8E-12	6.3E-12
1.6E-11	2.7E-11	3.9E-11	5.8E+02	1.0E-10	2.3E-10	1.5E-11	8.9E-13	3.1E-11	7.5E-12	2.9E-12	7.1E-12
2.2E-13	5.6E-12	1.2E-11	4.1E-11	6.0E+02	4.4E-10	1.9E-11	1.4E-11	2.4E-11	1.4E-11	9.8E-13	7.3E-12
1.8E-11	6.4E-12	8.0E-11	2.4E-10	1.8E-09	6.3E+02	2.0E-11	5.5E-12	1.3E-11	1.1E-11	2.8E-12	6.7E-12
4.4E-12	6.3E-13	1.9E-12	1.5E-11	2.6E-11	2.8E-11	5.7E+02	1.1E-10	3.6E-11	8.8E-12	2.6E-13	2.3E-12
4.7E-12	3.2E-12	7.6E-12	9.6E-12	9.2E-12	1.6E-12	4.7E-11	5.6E+02	1.6E-11	4.2E-11	1.0E-11	7.1E-12
1.2E-12	3.4E-12	1.4E-11	2.2E-11	1.5E-11	2.5E-12	3.9E-11	8.3E-12	5.6E+02	1.3E-11	2.2E-12	1.3E-11
2.3E-12	4.0E-12	7.5E-12	2.5E-13	1.2E-11	1.5E-11	2.0E-12	2.7E-11	3.3E-11	4.9E+02	8.1E-12	3.8E-11
1.6E-12	5.1E-13	1.6E-13	3.0E-14	9.2E-12	6.1E-12	6.5E-13	2.1E-11	2.5E-12	2.7E-12	5.5E+02	5.1E-12
4.8E-12	2.4E-12	5.3E-12	6.0E-13	2.0E-12	9.9E-13	1.2E-11	1.4E-11	1.4E-11	1.6E-11	1.6E-11	6.3E+02

Table 6 Cross section data for VV1K3D

Mixture	Group	$D_g (cm)$	$\Sigma_{g,r} (cm^{-1})$	$\nu\Sigma_{f,g} (cm^{-1})$	$\Sigma_{s,g \rightarrow g+1} (cm^{-1})$
1	1	1.38320	2.48836E-2	4.81619E-3	1.64977E-2
	2	3.86277E-1	6.73049E-2	8.46154E-2	
2	1	1.38299	2.62865E-2	4.66953E-3	1.47315E-2
	2	3.89403E-1	8.10328E-2	8.52264E-2	
3	1	1.39522	2.45662E-2	6.04889E-3	1.56219E-2
	2	3.86225E-1	8.44801E-2	1.19428E-1	
4	1	1.39446	2.60117E-2	5.91507E-3	1.40185E-2
	2	3.87723E-1	9.89671E-2	1.20497E-1	
5	1	1.39506	2.46141E-2	6.40256E-3	1.54981E-2
	2	3.84492E-1	8.93878E-2	1.29281E-1	

Table 7 Time results (seconds) of VV1K3D

Integration	K	L	$\mathbb{A}_{g,g}$ size	Hotelling					Krylov-Schur				
				t_1	t_2	t_3	t_5	t_{12}	t_1	t_2	t_3	t_5	t_{12}
Analytical	1	3	21237	1	5	114	145		2	3	4	4	7
		12	84048	4	19	80	102	470	10	21	22	25	49
	2	3	145788	8	45	254	343		21	46	49	56	109
Gauss-Lobatto	1	3	21237	1	5	20	26		2	3	3	4	6
		12	84048	4	19	58	75		8	17	17	21	37
	2	3	145788	8	43				18	39	41	48	89
Gauss-Legendre	1	3	21237	2	6	230	326		5	10	10	12	20
		12	84048	4	19	176	257		47	103	115	170	271
	2	3	145788	13	70				94	219	224	235	488

Table 8 Eigenvalues of VV1K3D

Integration	K	L	Hotelling			Krylov-Schur		
			k_1	k_2	k_3	k_1	k_2	k_3
Analytical	1	3	1.005633	0.987859	0.972650	1.005633	0.987868	0.987707
		12	1.005399	0.987405	0.969676	1.005404	0.987439	0.987349
	2	3	1.005450	0.987368	0.969726	1.005450	0.987368	0.987360
Gauss-Lobatto	1	3	1.008769	0.992112	0.979504	1.008769	0.992128	0.991786
		12	1.006633	0.988897	0.974677	1.006625	0.988829	0.988791
	2	3	1.005448	0.987465		1.005448	0.987466	0.987410
Gauss-Legendre	1	3	1.004073	0.985430	0.964917	1.004073	0.985508	0.985425
		12	1.004793	0.986571	0.967143	1.004806	0.986726	0.986507
	2	3	1.005477	0.987287		1.005478	0.987333	0.987287

Table 9 Time results (seconds) of VV1K3D, using direct solvers in the Krylov-Schur method

Integration	K	L	t_1	t_2	t_3	t_5	t_{12}
Gauss-Legendre	1	3	3	3	4	7	10
	1	12	20	28	28	34	53
	2	3	65	95	95	118	162

Figure Captions

Figure 1 The Piola transformation.

Figure 2 IAEA 3D reactor.

Figure 3 Assembly distribution in VV1K3D reactor.

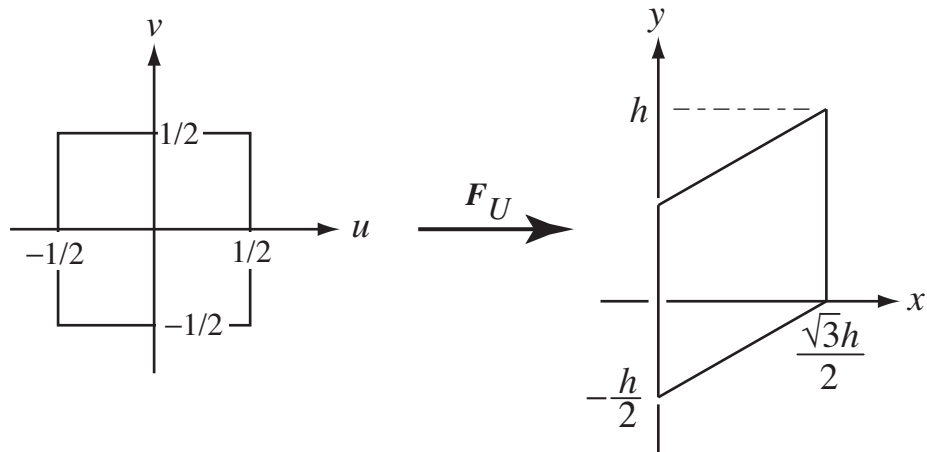


Figure 1 The Piola transformation.

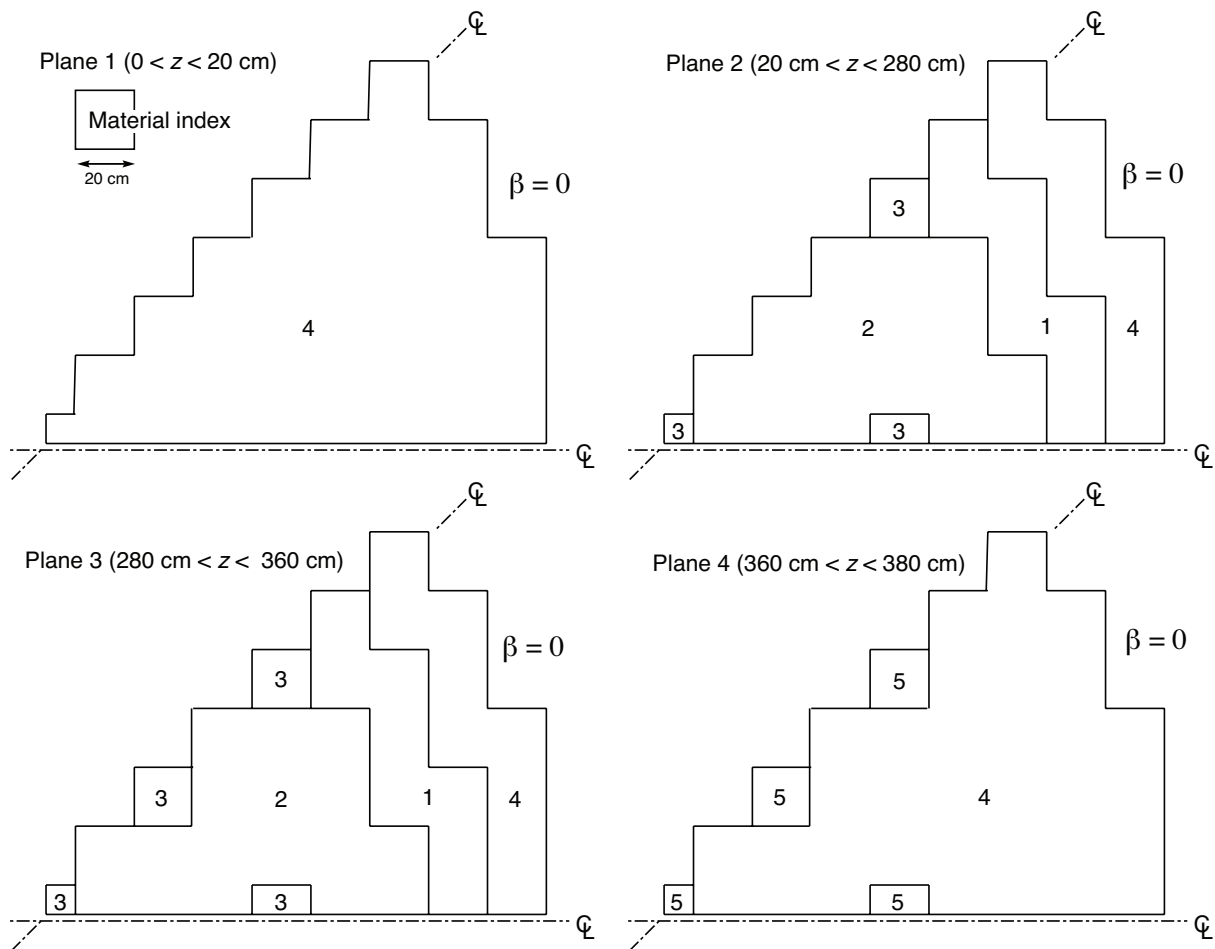


Figure 2 IAEA 3D reactor.

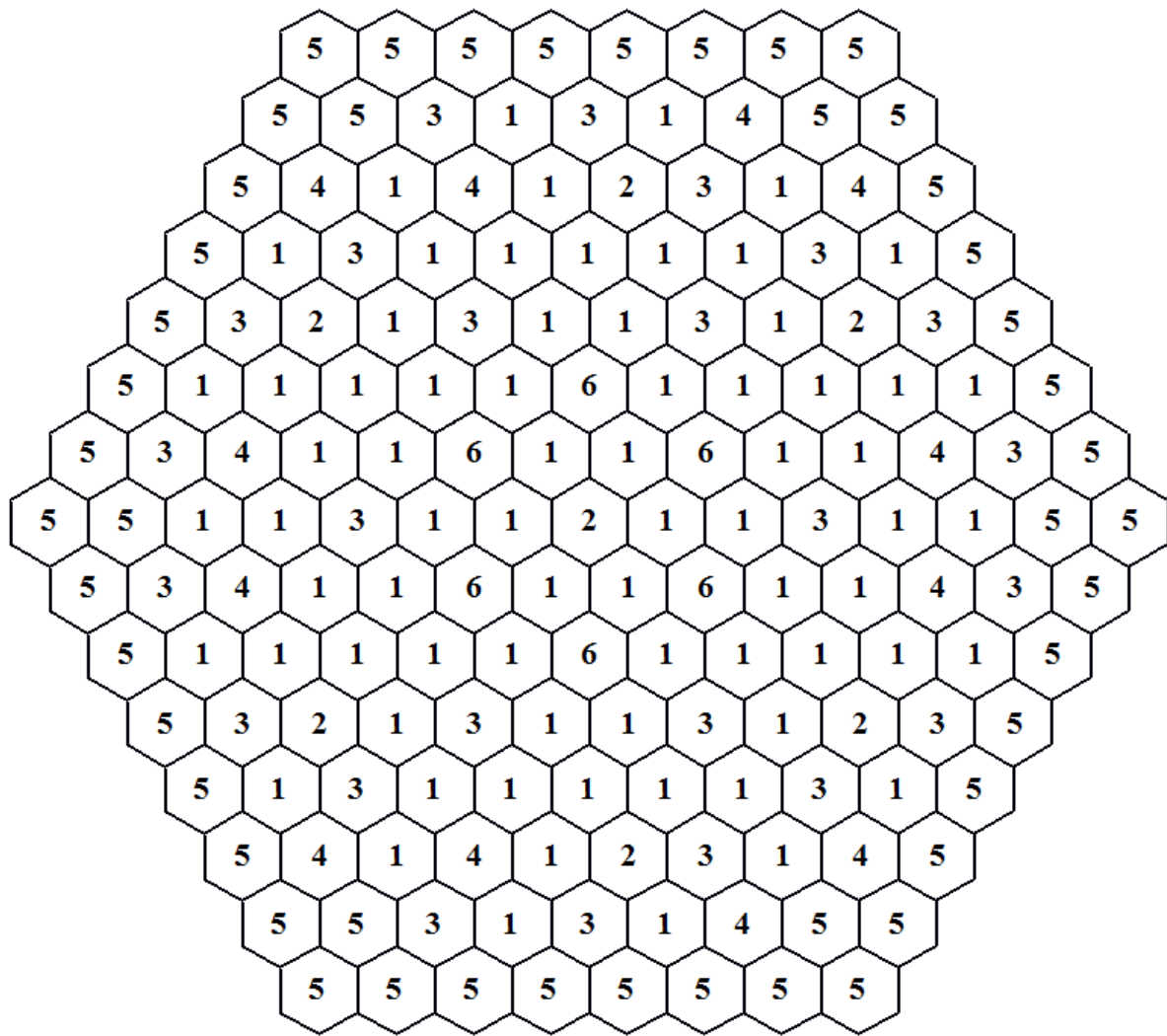


Figure 3 Assembly distribution in VV1K3D reactor.