Calculation of non-fundamental Modes in TRIVAC5 with SLEPc

A. Bernal[*], A. Hébert[†], J.E. Roman[★], R. Miró[*] and G. Verdú[*]

[*]*Institute for Industrial, Radiophysical and Environmental Safety, Universitat Politècnica de València, Camí de Vera s/n, Valencia, Spain, 46022*
[†]*Institut de Génie Nucléaire, École Polytechnique de Montréal, Montréal H3C 3A7, Canada*
[★]*Department of Information Systems and Computations, Universitat Politècnica de València, Camí de Vera s/n, Valencia, Spain, 46022*
abernal@iqn.upv.es, alain.hebert@polymtl.ca, jroman@dsic.upv.es, rmiro@iqn.upv.es, gverdu@iqn.upv.es

Abstract - *TRIVAC is a computer code intended to compute the neutron flux in a fractional or in a full core representation of a nuclear reactor. The multigroup and multidimensional form of the diffusion equation or simplified P_n equation is first discretized to produce a consistent matrix system. This system matrices are suitable for solving the eigenvalue problem with the preconditioned power method, which 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 and instabilities of nuclear reactors. So, effective and fast methods are required for solving eigenvalue problems.The most effective methods are those based on Arnoldi method and projection methods, such as Krylov-Schur. Currently, the state of the art for calculating eigenvalue problems is the SLEPc library, which is s 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. In this work, the Krylov-Schur method of SLEPc has been applied to the neutron diffusion equation.*

I. INTRODUCTION

TRIVAC [1] is a computer code intended to compute the neutron flux in a fractional or in a full core representation of a nuclear reactor. The multigroup and multidimensional form of the diffusion equation or simplified P_n equation is first discretized to produce a consistent matrix system.

The current implementation of TRIVAC allows the discretization of 1-D geometries (slab and cylindrical), 2-D geometries (Cartesian, cylindrical and hexagonal) and 3-D geometries (Cartesian and hexagonal). Many discretization techniques are available, including mesh corner or mesh centered finite difference methods, collocation techniques of various order, such as the nodal collocation method [2], and finite element methods based on a primal or dual functional formulation [3].

The matrix system constitutes an eigenvalue problem which is subsequently solved using iterative techniques [4] (inverse or preconditioned power method with ADI preconditioning [5]) and sparse matrix algebra techniques (triangular factorization). These techniques are very fast and optimized for the calculation of the fundamental mode.

The calculation of the non-fundamental modes or harmonics is performed in TRIVAC with the Hotelling deflation technique. This technique is a decontamination operation of the power method so as to converge to non- fundamental harmonics of the eigenvalue problem. The algorithm implemented in TRIVAC consists in computing one harmonic at a time, while decontaminating the inverse system matrix from the previously calculated harmonics, including the fundamental solution [4]. This technique may work for obtaining the first dozen harmonics and their associated adjoint problem, but its convergence rate is not as fast as that of the fundamental mode. Moreover, it has been checked that the Hotelling deflation technique cannot calculate more than three harmonics for commercial nuclear reactors, or not the largest ones.

Knowledge of these harmonics is important for modal analysis and instabilities and fluctuations of nuclear reactors [6, 7]. Furthermore, instabilities calculation may require the calculation of a lot of harmonics (a minimum of five). Therefore, fast, effective and accurate methods have been included in TRIVAC.

The most effective methods for solving the eigenvalue problems are those based on projection methods such as Arnoldi. There are several software and libraries containing the algorithm of these methods, which have been widely used. Currently, the state of the art for calculating eigenvalue problems is the SLEPc library.

SLEPc, the Scalable Library for Eigenvalue Problem Computations, is a software library for the solution of large, sparse eigenproblems on parallel computers. It can be used for the solution of linear eigenvalue problems formulated in either standard or generalized form, both Hermitian and non-Hermitian, with either real or complex arithmetic, as well as other related problems [8]. SLEPc focuses on sparse problems, such as those arising after the discretization of partial differential equations. It provides projection methods or other methods with similar properties, such as Krylov-Schur or Jacobi-Davidson [9]. SLEPc is built on top of PETSc (Portable, Extensible Toolkit for Scientific Computation) and extends it with all the functionality necessary for the solution of eigenvalue problems [10].

This work is a preliminary stage, and it has been applied to the diffusion equation, but not to the simplified P_n equations. It has been applied to 3D geometries, either Cartesian or hexagonal. When the authors wrote this paper, we implemented SLEPc only for the variational collocation method and the nodal collocation method, but now is also implemented for the Raviart-Thomas method. Thus, the work is focused on the variational and nodal collocation methods.

The outline of the paper is as follows. Section II. describes the theory and methods discretizing the neutron diffusion equation and solving the eigenvalue problem. Section III. describes the reactor used for testing the method and shows the results. Section IV. summarizes the conclusions.

II. THEORY

1. The nodal collocation method

This section explains the nodal collocation method for the one energy group approach, but it can be extrapolated to any energy group approach. The one energy group 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, $\boldsymbol{J}(r)$ 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. Equations (1) and (2) can be combined to form Equation (3).

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

$$\boldsymbol{J}(r) = -D(r)\boldsymbol{\nabla}\phi(r) \tag{2}$$

$$\begin{aligned} -\frac{\partial}{\partial x}D_x(x,y,z)\frac{\partial\phi}{\partial x} - \frac{\partial}{\partial y}D_y(x,y,z)\frac{\partial\phi}{\partial y} - \frac{\partial}{\partial z}D_z(x,y,z)\frac{\partial\phi}{\partial z} \\ +\Sigma_r(x,y,z)\phi(x,y,z) = S(x,y,z) \end{aligned} \tag{3}$$

The domain can be discretized in elements, such as parallelepipeds, with uniform nuclear properties in each one. For each element e, its nuclear properties will be: D_{xe}, D_{ye}, D_{ze} and Σ_{re}. In addition, the Cartesian coordinates (x,y,z) of the element will be transformed into local coordinates (u, v, w) corresponding to a unitary cube of reference, ranging from $-1/2$ to $1/2$. This transformation is expressed in Equations (4)-(6), in which $\Delta x_e = x_{m+1/2} - x_{m-1/2}$, $\Delta y_e = y_{n+1/2} - y_{n-1/2}$ and $\Delta z_e = z_{p+1/2} - z_{p-1/2}$. By means of this transformation, Equation (3) is transformed into Equation (7), where $V_e = \Delta x_e \Delta y_e \Delta z_e$.

$$u = \frac{1}{\Delta x_e}\left[x - \frac{1}{2}\left(x_{m+1/2} + x_{m-1/2}\right)\right] \tag{4}$$

$$v = \frac{1}{\Delta y_e}\left[y - \frac{1}{2}\left(y_{n+1/2} + y_{n-1/2}\right)\right] \tag{5}$$

$$w = \frac{1}{\Delta z_e}\left[z - \frac{1}{2}\left(z_{p+1/2} + z_{p-1/2}\right)\right] \tag{6}$$

$$\begin{aligned} -\frac{\Delta y_e \Delta z_e}{\Delta x_e}D_{x,e}\frac{\partial^2\phi_e}{\partial x^2} - \frac{\Delta x_e \Delta z_e}{\Delta y_e}D_{y,e}\frac{\partial^2\phi_e}{\partial y^2} \\ -\frac{\Delta x_e \Delta y_e}{\Delta z_e}D_{z,e}\frac{\partial^2\phi_e}{\partial z^2} + V_e\Sigma_{re}\phi_e(u,v,w) = V_e S_e(u,v,w) \end{aligned} \tag{7}$$

Over each element e, $\phi_e(x,y,z)$ and $S_e(x,y,z)$ are expanded with the first $K+1$ orthonormal Legendre polynomials ($P_k(u)$,$k = 0, K$) in the interval $[-1/2, 1/2]$. These expansions are shown in Equations (8) and (9). If one substitutes these expansions in Equation (7), multiplies it by the weight function $P_{l_1}(u)P_{l_2}(v)P_{l_3}(w)$ and integrates this product over the element e, one obtains Equation (10).

$$\phi_e(u,v,w) = \sum_{k_1=0}^{K}\sum_{k_2=0}^{K}\sum_{k_3=0}^{K}\phi_e^{k_1,k_2,k_3}P_{k_1}(u)P_{k_2}(v)P_{k_3}(w) \tag{8}$$

$$S_e(u,v,w) = \sum_{k_1=0}^{K}\sum_{k_2=0}^{K}\sum_{k_3=0}^{K}S_e^{k_1,k_2,k_3}P_{k_1}(u)P_{k_2}(v)P_{k_3}(w) \tag{9}$$

$$\begin{aligned} -\Delta y_e \Delta z_e F_{e,x}^{k_1,k_2,k_3} - \Delta x_e \Delta z_e F_{e,y}^{k_1,k_2,k_3} - \Delta x_e \Delta y_e F_{e,z}^{k_1,k_2,k_3} \\ +V_e\Sigma_{re}\phi_e^{k_1,k_2,k_3} = V_e S_e^{k_1,k_2,k_3} \end{aligned} \tag{10}$$

Terms $F_{e,x}^{k_1,k_2,k_3}$, $F_{e,y}^{k_1,k_2,k_3}$ and $F_{e,z}^{k_1,k_2,k_3}$ are defined in Equations (11)-(13). These equations contain the functions $\phi_{e,x}^{k2,k3}(u)$, $\phi_{e,y}^{k1,k3}(v)$, $\phi_{e,z}^{k1,k2}(w)$ and $L_k(f(u))$, which are defined in Equations (14)-(17). In Equation (17), F_l are the Legendre expansion coefficients of the function $f(u)$.

$$F_{e,x}^{k_1,k_2,k_3} = \frac{D_{x,e}}{\Delta x_e}L_{k_1}\left\{\phi_{e,x}^{k2,k3}(u)\right\} \tag{11}$$

$$F_{e,y}^{k_1,k_2,k_3} = \frac{D_{y,e}}{\Delta y_e}L_{k_2}\left\{\phi_{e,y}^{k1,k3}(v)\right\} \tag{12}$$

$$F_{e,z}^{k_1,k_2,k_3} = \frac{D_{z,e}}{\Delta z_e}L_{k_3}\left\{\phi_{e,z}^{k1,k2}(w)\right\} \tag{13}$$

$$\phi_{e,x}^{k2,k3}(u) = \sum_{k=0}^{K}\phi_e^{k,k_2,k_3}P_k(u) \tag{14}$$

$$\phi_{e,y}^{k1,k3}(v) = \sum_{k=0}^{K}\phi_e^{k_1,k,k_3}P_k(v) \tag{15}$$

$$\phi_{e,z}^{k1,k2}(w) = \sum_{k=0}^{K}\phi_e^{k_1,k_2,k}P_k(w) \tag{16}$$

$$\begin{aligned} L_k\left\{f(u)\right\} &= \int_{-1/2}^{1/2} du P_k(u)\frac{\partial^2}{\partial u^2}f(u) \\ &= \sqrt{2k+1}\left\{(-1)^{k+1}\left[k(k+1)f\left(\frac{-1}{2}\right) + \frac{\partial}{\partial u}f\left(\frac{-1}{2}\right)\right]\right. \\ &\quad -\left[k(k+1)f\left(\frac{1}{2}\right) - \frac{\partial}{\partial u}f\left(\frac{-1}{2}\right)\right] \\ &\quad \left. +\sum_{l=0}^{k-2}\left[1+(-1)^{k+l}\right]\sqrt{2l+1}\left[k(k+1) - l(l+1)\right]F_l\right\} \end{aligned} \tag{17}$$

Equation (10) contains surface quantities, because $F_{e,x}^{k_1,k_2,k_3}$, $F_{e,y}^{k_1,k_2,k_3}$ and $F_{e,z}^{k_1,k_2,k_3}$ contain the following terms: $\phi_{e,x}^{k2,k3}(\pm 1/2)$, $\phi_{e,y}^{k1,k3}(\pm 1/2)$, $\phi_{e,z}^{k1,k2}(\pm 1/2)$ and their derivatives. To couple these surface quantities with the Legendre coefficients $\phi_e^{k_1,k_2,k_3}$, the continuity conditions of the flux and the current are considered. If one considers the interfaces of the element e, as Figure 1 shows, the continuity conditions at the interface of element e and e_1 are defined in Equations (18) and (19).

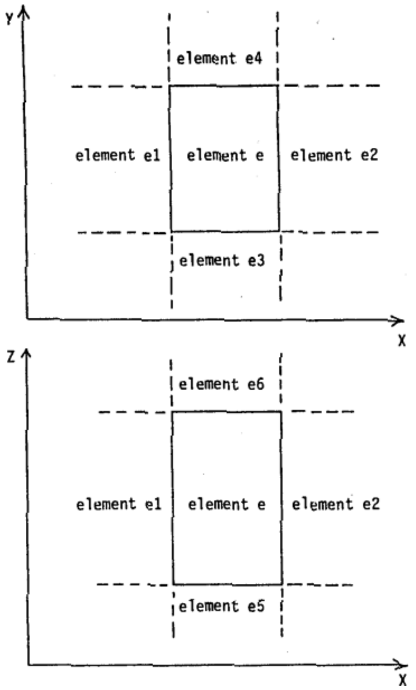

Fig. 1. Position and numbering of the neighboring elements.

$$\phi_{e_1}\left(\frac{1}{2}, v, w\right) = \phi_e\left(-\frac{1}{2}, v, w\right) \tag{18}$$

$$\frac{D_{x,e_1}}{\Delta x_{e_1}}\frac{\partial}{\partial u}\phi_{e_1}\left(\frac{1}{2}, v, w\right) = \frac{D_{x,e}}{\Delta x_e}\frac{\partial}{\partial u}\phi_e\left(-\frac{1}{2}, v, w\right) \tag{19}$$

Multiplying these equations by the weight function $P_{k_2}(v)P_{k_3}(w)$ and integrating over the interface, Equations (20) and (21) are obtained. Similar conditions can be obtained for the rest interfaces of the element e.

$$\phi_{e_1,x}^{k_2,k_3}\left(\frac{1}{2}\right) = \phi_{e,x}^{k_2,k_3}\left(-\frac{1}{2}\right) \tag{20}$$

$$\frac{D_{x,e_1}}{\Delta x_{e_1}}\frac{\partial}{\partial u}\phi_{e_1,x}^{k_2,k_3}\left(\frac{1}{2}\right) = \frac{D_{x,e}}{\Delta x_e}\frac{\partial}{\partial u}\phi_{e,x}^{k_2,k_3}\left(-\frac{1}{2}\right) \tag{21}$$

Now, $F_{e,x}^{k_1,k_2,k_3}$, $F_{e,y}^{k_1,k_2,k_3}$ and $F_{e,z}^{k_1,k_2,k_3}$ can be calculated by means of the Legendre coefficients $\phi_e^{k_1,k_2,k_3}$. To do that, one has to substitute Equation (17) in Equations (11)-(13) and apply the continuity conditions of Equations (20) and (21). After some algebra, Equations (22)-(24) are obtained. Further details of the development of these expressions can be found in [2].

$$F_{e,x}^{k,k_2,k_3} = \sum_{l=0}^{K-1}\left(A_{e,x}^{k,l;K}\phi_{e_1}^{l,k_2,k_3} - B_{e,x}^{k,l;K}\phi_e^{l,k_2,k_3} + C_{e,x}^{k,l;K}\phi_{e_2}^{l,k_2,k_3}\right) \tag{22}$$

$$F_{e,y}^{k_1,k,k_3} = \sum_{l=0}^{K-1}\left(A_{e,y}^{k,l;K}\phi_{e_3}^{k_1,l,k_3} - B_{e,y}^{k,l;K}\phi_e^{k_1,l,k_3} + C_{e,y}^{k,l;K}\phi_{e_4}^{k_1,l,k_3}\right) \tag{23}$$

$$F_{e,z}^{k_1,k_2,k} = \sum_{l=0}^{K-1}\left(A_{e,z}^{k,l;K}\phi_{e_5}^{k_1,k_2,l} - B_{e,z}^{k,l;K}\phi_e^{k_1,k_2,l} + C_{e,z}^{k,l;K}\phi_{e_6}^{k_1,k_2,l}\right) \tag{24}$$

Equations (22)-(24) contain the terms $A_{e,\alpha}^{k,l;K}$, $B_{e,\alpha}^{k,l;K}$ and $C_{e,\alpha}^{k,l;K}$, which are defined in Equations (25)-(27), where $\alpha = x, y, z$.

$$\begin{aligned} A_{e,\alpha}^{k,l;K} &= \frac{(-1)^k}{2K(K+1)}\sqrt{2k+1}\sqrt{2l+1}\,[K(K+1) \\ &\quad -k(k+1)]\,[K(K+1) - l(l+1)]\,W_{e,\alpha}^{-} \end{aligned} \tag{25}$$

$$\begin{aligned} B_{e,\alpha}^{k,l;K} &= B_{e,\alpha}^{l,k;K} = \frac{\sqrt{2k+1}\sqrt{2l+1}}{K(K+1)}\left\{\frac{D_{\alpha,e}}{\Delta\alpha_e}\left[1+(-1)^{k+l}\right][K(K+1)\right. \\ &\quad -k(k+1)]\,l(l+1) + \frac{1}{2}\,[K(K+1) - k(k+1)]\,[K(K+1) \\ &\quad \left. -l(l+1)]\left[(-1)^{k+l}W_{e,\alpha}^{-} + W_{e,\alpha}^{+}\right]\right\},\quad \text{if } k \geq l \end{aligned} \tag{26}$$

$$\begin{aligned} C_{e,\alpha}^{k,l;K} &= \frac{(-1)^l}{2K(K+1)}\sqrt{2k+1}\sqrt{2l+1}\,[K(K+1) \\ &\quad -k(k+1)]\,[K(K+1) - l(l+1)]\,W_{e,\alpha}^{+} \end{aligned} \tag{27}$$

These equations contain the centered finite difference coupling factors $W_{e,\alpha}^{\pm}$, which are defined in Equations (28)-(33). In cases of boundary elements with zero flux boundary condition, the coupling factor at that boundary face will be $W_{e,\alpha}^{\pm} = \frac{2D_{\alpha,e}}{\Delta\alpha_e}$. However, if the boundary condition is reflective, the coupling factor at that boundary will have a value of 0.

$$W_{e,x}^{-} = W_{e_1,x}^{+} = 2D_{x,e}D_{x,e_1}/(\Delta x_e D_{x,e_1} + \Delta x_{e_1} D_{x,e}) \tag{28}$$

$$W_{e,x}^{+} = W_{e_2,x}^{-} = 2D_{x,e}D_{x,e_2}/(\Delta x_e D_{x,e_2} + \Delta x_{e_2} D_{x,e}) \tag{29}$$

$$W_{e,y}^{-} = W_{e_3,y}^{+} = 2D_{y,e}D_{y,e_3}/(\Delta y_e D_{y,e_3} + \Delta y_{e_3} D_{y,e}) \tag{30}$$

$$W_{e,y}^{+} = W_{e_4,y}^{-} = 2D_{y,e}D_{y,e_4}/(\Delta y_e D_{y,e_4} + \Delta y_{e_4} D_{y,e}) \tag{31}$$

$$W^-_{e,z} = W^+_{e_5,z} = 2D_{z,e}D_{z,e_5}/(\Delta z_e D_{z,e_5} + \Delta z_{e_5} D_{z,e}) \tag{32}$$

$$W^+_{e,z} = W^-_{e_6,z} = 2D_{z,e}D_{z,e_6}/(\Delta z_e D_{z,e_6} + \Delta z_{e_6} D_{z,e}) \tag{33}$$

On the other hand, for the general G energy groups approach, the source of Equation (10) for a g energy group is expressed as in Equation (34).

$$S^{k_1,k_2,k_3}_{g,e} = \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \Sigma_{s,g'\to g}\phi^{k_1,k_2,k_3}_{g',e} + \frac{1}{k_{eff}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}\phi^{k_1,k_2,k_3}_{g',e} \tag{34}$$

If one applies this method to the whole geometry, one obtains an eigenvalue problem as that of Equation (35). In this equation, $\mathbb{A}$ contains the leakage, removal and upscattering terms; $\mathbb{B}$ contains the fission terms; and Φ is the eigenvector containing the Legendre coefficients of the flux for each energy group g and element e: $\phi^{k_1,k_2,k_3}_{g,e}$.

$$\mathbb{A}\Phi = \frac{1}{k_{eff}}\mathbb{B}\Phi \tag{35}$$

2. Variational collocation method

In this method, the diffusion equation and its continuity and boundary conditions are rewritten into a variational formulation, in which the diffusion equation is a stationary point of a related functional in a Sobolev space. Many variational formulations exist, but the method implemented in TRIVAC [3, 4] is based on the primal functional of Equation (36), where $\phi(r) \in H_{\partial V_0}(V)$. The Sobolev space $H_{\partial V_0}(V)$ contains functions and first derivatives that are L^2 integrable over the domain V. In addition, this Sobolev space contains functions which are equal to zero over the surfaces with zero flux boundary conditions (∂V_0).

$$\begin{aligned} F_a\{\phi\} &= \frac{1}{2}\int_V d^3r\{\nabla\phi(r)\cdot[D(r)\nabla\phi(r)] + \Sigma_r(r)\phi(r)^2 \\ &\quad -2\phi(r)S(r)\} + \frac{1}{4}\int_{\partial V_\beta} d^2r\frac{1-\beta(r)}{1+\beta(r)}\phi(r)^2 \end{aligned} \tag{36}$$

A stationary point of this functional is defined by Equation (37), which is developed in Equation (38). Now, the finite element method is used for finding a stationary point of Equation (38), which will not be the exact solution, but the best approximation (in the variational sense) of the diffusion equation.

$$\delta_{\delta\phi}F_a = \lim_{\epsilon\to 0}\left\{\frac{d}{d\epsilon}F_a\{\phi(r) + \epsilon\delta\phi(r)\}\right\} = 0 \tag{37}$$

$$\begin{aligned} \delta_{\delta\phi}F_a &= \int_V d^3r\{\nabla\delta\phi(r)\cdot[D(r)\nabla\phi(r)] + \Sigma_r(r)\delta\phi(r)\phi(r) \\ &\quad -\delta\phi(r)S(r)\} + \frac{1}{2}\int_{\partial V_\beta} d^2r\frac{1-\beta(r)}{1+\beta(r)}\delta\phi(r)\phi(r) = 0 \end{aligned} \tag{38}$$

The finite element method is based on an expansion of the dependent variables, $\phi(r)$ and $\delta\phi(r)$, into a linear combination of polynomial trial functions defined over subvolumes. These trial functions are known a priori and the corresponding coefficients will be found by using Equation (38).

The finite element method can be applied to various types of subvolumes or elements. Cartesian and hexagonal elements are the most widely used in reactor physics for full-core calculations.The domain is first partitioned into elements over which the nuclear properties are assumed to be uniform. 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.

The expansion of $\phi(r)$ and $\delta\phi(r)$ proposed in this method uses tensorial products of Lagrange polynomials. Polynomial expressions of the Lagrange polynomials $L_k(u)$ defined over $-1/2 \leq u \leq 1/2$ are given in Appendix A in [3] and in [4]. An order K expansion of ϕ and $\delta\phi$ in each element e is shown in Equations (39) and (40).

$$\phi_e(u,v,w) = \sum_{k1=0}^{K}\sum_{k2=0}^{K}\sum_{k3=0}^{K}\phi_e^{k_1,k_2,k_3}L_{k_1}(u)L_{k_2}(v)L_{k_3}(w) \tag{39}$$

$$\delta\phi_e(u,v,w) = \sum_{k1=0}^{K}\sum_{k2=0}^{K}\sum_{k3=0}^{K}\delta\phi_e^{k_1,k_2,k_3}L_{k_1}(u)L_{k_2}(v)L_{k_3}(w) \tag{40}$$

Since S is a function of ϕ, as it is defined in Equation (34), it can be expanded with the same functions. In particular, the source can be expressed as a sum of the fission terms and the non-fission terms, such as: $S(r) = \frac{1}{k_{eff}}S_f(r) + S_u(r)$. If one substitutes these expansions in Equation (38), one can rewrite it as in Equation (41), whose terms are the integrals defined in Equations (42)-(43). These integrals can be obtained analytically, but it is better to use numerical methods for suppressing off-diagonal terms of the system matrices. In fact, TRIVAC uses a Gauss-Lobatto quadrature for this purpose. By means of this method, one can obtain the matrices of the eigenvalue problem of Equation (35).

$$\delta_{\delta\phi}F_a = \langle\delta\phi, A\phi\rangle + \frac{1}{k_{eff}}\langle\delta\phi, B\phi\rangle = 0 \tag{41}$$

$$\begin{aligned} \langle\delta\phi, A\phi\rangle &= \int_V d^3r\{\nabla\delta\phi(r)\cdot[D(r)\nabla\phi(r)] + \Sigma_r(r)\delta\phi(r)\phi(r)\} \\ &\quad + \int_V d^3r\delta\phi(r)S_u(r) + \frac{1}{2}\int_{\partial V_\beta} d^2r\frac{1-\beta(r)}{1+\beta(r)}\delta\phi(r)\phi(r) \end{aligned} \tag{42}$$

$$\langle\delta\phi, B\phi\rangle = \int_V d^3r\delta\phi(r)S_f(r) \tag{43}$$

3. The Hotelling deflation technique

The Hotelling deflation technique is a decontamination operation of the power method so as to converge to non-fundamental harmonics of the eigenvalue problem [4]. 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 (35), the inverse power recurrence of Equation (44) 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 (45), that includes the decontamination operation that purges the components that the current iteration $\Phi_l^{(n+1)}$ has in the direction of already converged eigenvectors. In Equation (45), Φ_j^* is the adjoint eigenvector. Equations (46)-(48) demonstrate the convergence. Equation (46) develops Equation (45) by considering $\Phi_l^{(n)}$ as a linear combination of the modes. If one performs m iterations of Equation (46), one obtains Equation (47), which converges to Φ_l because of Equation (48).

$$\Phi_l^{(n+1)} = \frac{1}{k_l^{(n)}}\mathbb{A}^{-1}\mathbb{B}\Phi_l^{(n)} \tag{44}$$

$$\Phi_l^{(n+1)} = \frac{1}{k_l^{(n)}}\left(\mathbb{A}^{-1} - \sum_{j=1}^{l-1} k_j \frac{\Phi_j \cdot \Phi_j^*}{\Phi_j^* \cdot \mathbb{B}\Phi_j}\right)\mathbb{B}\Phi_l^{(n)} \tag{45}$$

$$\begin{aligned}
\Phi_l^{(n+1)} &= \frac{1}{k_l^{(n)}}\left(\mathbb{A}^{-1} - \sum_{j=1}^{l-1} k_j \frac{\Phi_j \cdot \Phi_j^*}{\Phi_j^* \cdot \mathbb{B}\Phi_j}\right)\mathbb{B}\sum_{i=1}^{L} c_i^{(n)}\Phi_i \\
&= \frac{1}{k_l^{(n)}}\left(\sum_{i=1}^{L} c_i^{(n)}\mathbb{A}^{-1}\mathbb{B}\Phi_i - \sum_{j=1}^{l-1} k_j \frac{\Phi_j \cdot \Phi_j^*}{\Phi_j^* \cdot \mathbb{B}\Phi_j}\sum_{i=1}^{L} c_i^{(n)}\mathbb{B}\Phi_i\right) \\
&= \frac{1}{k_l^{(n)}}\left(\sum_{i=1}^{L} c_i^{(n)}\mathbb{A}^{-1}\mathbb{B}\Phi_i - \right. \\
&\quad \left. - \sum_{j=1}^{l-1} k_j \frac{\Phi_j}{\Phi_j^* \cdot \mathbb{B}\Phi_j}\sum_{i=1}^{L} c_i^{(n)}\Phi_j^*\mathbb{B}\Phi_i\right) \\
&= \frac{1}{k_l^{(n)}}\left(\sum_{i=1}^{L} c_i^{(n)}k_i\Phi_i - \sum_{j=1}^{l-1} k_j\Phi_j c_j^{(n)}\right) \\
&= \sum_{i=l}^{L} \frac{k_i}{k_l^{(n)}} c_i^{(n)}\Phi_i
\end{aligned} \tag{46}$$

$$\Phi_l^{(n+m)} = \sum_{i=l}^{L}\left(\frac{k_i}{k_l^{(n)}}\right)^m c_i^{(n)}\Phi_i \tag{47}$$

$$\lim_{m\to\infty}\left(\frac{k_i}{k_l^{(n)}}\right)^m = 0 \qquad \text{if } k_l^{(n)} > k_i \; \forall i \geq l+1 \tag{48}$$

4. The Krylov-Schur method

The Krylov-Schur method is an Arnoldi method which uses an implicit restart based on a Krylov-Schur decomposition [11].

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 (49). 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, λ_i , x_i , $i = 1, \cdots, m$, 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.

$$\mathcal{K}_m(\mathcal{A}, x_0) = span\left\{x_0, \mathcal{A}x_0, \mathcal{A}^2x_0, \cdots, \mathcal{A}^{m-1}x_0\right\} \tag{49}$$

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 ($V_m^TV_m = I_m$), one concludes that the pair (λ_i,V_my_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 current eigenvector approximations, 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; this d might be an integer lower than m. 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 [11].

In this work, the authors have applied the Krylov-Schur algorithm implemented in SLEPc [8, 9] to the matrices obtained with TRIVAC [1]. 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 (50). In this case, the eigenvalue problem is defined in Equation (51), where ϕ_1 is the iterative eigenvector and ϕ_g, for $g > 1$, are calculated with Equation (52). The second type includes any upscattering and fission production, and is defined in Equation (53). 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 solvers. The authors have tried different methods implemented in PETSc [10] and the fastest method was Generalized Minimal Residual (GMRES) [12] using the Additive Schwarz preconditioner.

$$\begin{pmatrix} \mathbb{A}_{1,1} & & 0 \\ \vdots & \ddots & \\ -\mathbb{A}_{G,1} & \cdots & \mathbb{A}_{G,G} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_G \end{pmatrix} = \frac{1}{k} \begin{pmatrix} \mathbb{B}_{1,1} & & \mathbb{B}_{1,G} \\ & & \\ & 0 & \end{pmatrix} \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_G \end{pmatrix} \tag{50}$$

$$k\phi_1 = \mathbb{A}_{1,1}^{-1} \sum_{g=1}^{G} \mathbb{B}_{1,g}\phi_g = \mathcal{A}\phi_1 \tag{51}$$

$$\phi_g = \mathbb{A}_{g,g}^{-1} \sum_{i=1}^{g-1} \mathbb{A}_{g,i}\phi_i \tag{52}$$

$$k\Phi = \mathbb{A}^{-1}\mathbb{B}\Phi = \mathcal{A}\Phi \tag{53}$$

III. RESULTS AND ANALYSIS

1. PWR

The new capabilities of TRIVAC were tested in a PWR reactor. This reactor is a representation of the core of the Graveline nuclear plant in France. It was modeled with 8959 nodes, distributed in a Cartesian mesh of 17x17x31. Each of these nodes are cubes of 21.5 cm in length. The geometry model is shown in Figure 1. The thermal power was set at 2775 MW. As regards the cross sections, they are obtained for a critical condition of boron concentration and thermal- hydraulic properties in each reactor node. The thermal- hydraulics properties of each assembly were adjusted to the level of power released in each of them.

As regards the multrigroup energy approach, the two energy group approach was used, considering upscattering terms. The two numerical methods described in Section I. were used: nodal collocation method and a variational collocation method. In both methods, the authors used an order 2 for the polynomial expansion. The boundary conditions are zero flux.

Twelve modes were calculated and the Krylov-Schur solution is compared with the Hotelling deflation technique. Only the eigenvalues are shown in this work due to the extent of the results. Thus, the authors performed an analysis of eigenvalues and computational times. All the CPU time values have been obtained on an Intel Core i7-3770 CPU (3.4GHz), with the CentOS 6.8 operating system. For evaluating the eigenvalues, the authors used the eigenvalue error defined in Equation (54).

$$EE(pcm) = \frac{k - k_{ref}}{k_{ref}} \cdot 10^5 \tag{54}$$

2. Results with the nodal collocation method

The eigenvalues obtained with SLEPc are: 1.000085, 0.995865, 0.992194, 0.992194, 0.990316, 0.987578, 0.987578, 0.982266, 0.982197, 0.982197, 0.979307 and 0.978510. By contrast, the Hotelling deflation technique only could calculate the following 8 eigenvalues: 1.000084, 0.995867, 0.990319, 0.992188, 0.982505, 0.987307, 0.971998 and 0.992193.

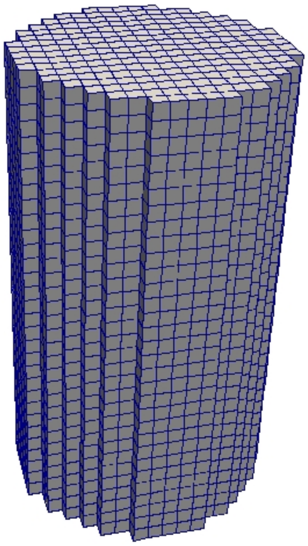

Fig. 2. PWR reactor.

One can appreciate that the first two eigenvalues are the same for both methods, but the third eigenvalue calculated by the Hotelling deflation technique corresponds to the fifth eigenvalue calculated by SLEPc. The Hotelling deflation technique cannot calculate accurately the third and fourth modes because they belong to a multiple eigenvalue (with algebraic multiplicity equal to 2 in this case). This problem is inherent to the power iteration (with or without deflation), because convergence depends on the ratio of magnitude of the eigenvalue being calculated and the next one. A workaround would be to use a block version of the power iteration (also known as simultaneous iteration). Krylov methods such as those implemented in SLEPc do not suffer from this drawback.

If one considers the Hotelling deflation technique as the reference solution, the following eigenvalue errors are obtained, for the first two modes: 0.08 and -0.22 pcm. If one calculates the eigenvalue error of the fifth mode calculated with SLEPc and the third mode calculated with the Hotelling deflation technique, one obtains a value of -0.34 pcm, which is accurate, but the third mode calculated with the Hotelling deflation technique is not the real one. So, it can be concluded that SLEPc calculates accurately the eigenvalues. Moreover, time results are shown in Table I, from which one can draw two conclusions. First, SLEPc is faster than the Hotelling deflation technique. Second, the increase in time with respect to the number of modes calculated is not proportional. In fact, this increase in time is lower than the proportional one. Finally, the fast flux of modes 5, 9 and 12 are displayed in Figures 3-5, to show the flux profile of them.

3. Results with the variational collocation method

The eigenvalues obtained with SLEPc are: 1.000263, 0.996062, 0.992392, 0.992392, 0.990508, 0.987802, 0.987802, 0.982454, 0.982415, 0.982414, 0.979686 and 0.978517. By contrast, the Hotelling deflation technique only could calculate the following 8 eigenvalues: 1.000262,

Number of modes	SLEPc	Hotelling deflation
1	4 s	3 s
2	6 s	12 s
3	8 s	35 s
5	12 s	1 min 18 s
12	20 s	-

TABLE I. Time results for the nodal collocation method.

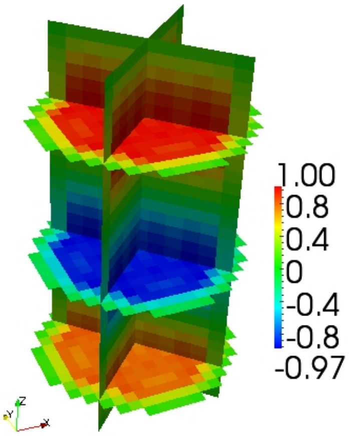

Fig. 3. Fast Flux of mode 5.

0.996065, 0.990509, 0.982457, 0.992386, 0.972812, 0.987143 and 0.992392.

Same conclusions as in the previous subsection are obtained. The Hotelling deflation technique fails to determine the multiple eigenvalues. As regards the eigenvalue errors, the following values are obtained for the first two modes: 0.1 and -0.3 pcm. The time results for this method is shown in Table II.

Number of modes	SLEPc	Hotelling deflation
1	4 s	3 s
2	6 s	13 s
3	8 s	20 s
5	12 s	1 min 5 s
12	20 s	-

TABLE II. Time results for the variational collocation method.

IV. CONCLUSIONS

A new capability has been included in TRIVAC Version5, which can calculate fast and accurately several non-fundamental modes.

The calculation of the non-fundamental modes is performed with the Krylov-Schur algorithm of SLEPc. The implementation can be applied to the neutron diffusion equation, but not to the simplified P_n equations. Although this new capability can be applied to any numerical discretization of TRIVAC, the work was focused only on the nodal collocation method and the variational collocation method.

The method was validated in a PWR with a 2 energy group approach, including upscattering terms. In this validation, the Krylov-Schur algorithm was compared with the Hotelling deflation technique, which was already included in TRIVAC Version5. Two major conclusions are obtained. First, this work shows the incapability of the Hotelling deflation technique to calculate accurately multiple eigenvalues. Second, it was proved that the Krylov-Schur algorithm is faster than the Hotelling deflation technique for calculating non-fundamental modes.

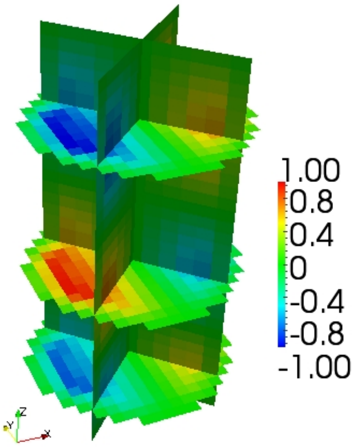

Fig. 4. Fast Flux of mode 9.

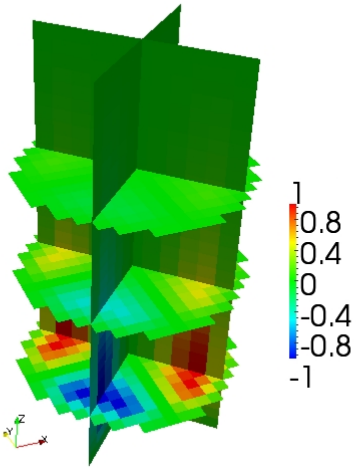

Fig. 5. Fast Flux of mode 12.

As regards future work, the Krylov-Schur algorithm will be applied to the simplified P_n equations.

V. ACKNOWLEDGMENTS

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. A. HÉBERT, "A user guide for TRIVAC version4," Tech. Rep. IGE-2293, Institut de Génie Nucléaire, École Polytechnique de Montréal (2014).
2. A. HÉBERT, "Development of the nodal collocation method for solving the neutron diffusion equation," *Annals of Nuclear Energy*, **14**, *10*, 527–541 (1987).
3. A. HÉBERT, "Application of a dual variational formulation to finite element reactor calculations," *Annals of Nuclear Energy*, **20**, *12*, 823–845 (1993).
4. A. HÉBERT, *Applied Reactor Physics*, Presses Internationales Polytechnique, 2nd ed. (2016).
5. A. HÉBERT, "Preconditioning the Power Method for Reactor Calculations," *Nuclear Science and Engineering*, **94**, *1*, 1–11 (1986).
6. G. VERDÚ, D. GINESTAR, V. VIDAL, and J. MUÑOZ-COBO, "3D λ-modes of the neutron-diffusion equation," *Annals of Nuclear Energy*, **21**, *7*, 405–421 (1994).
7. R. MIRÓ, D. GINESTAR, G. VERDÚ, and D. HENNIG, "A nodal modal method for the neutron diffusion equation. Application to BWR instabilities analysis," *Annals of Nuclear Energy*, **29**, *10*, 1171–1194 (2002).
8. V. V. HERNANDEZ V., ROMAN J.E., "SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems," *ACM Transactions on Mathematical Software*, **31**, *3*, 351–362 (2005).
9. V. V. HERNANDEZ V., ROMAN J.E., "SLEPc: Scalable library for eigenvalue problem computations," *Lecture Notes in Computer Science*, **2565**, 373–391 (2003).
10. S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN, V. EIJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, K. RUPP, B. F. SMITH, S. ZAMPINI, H. ZHANG, and H. ZHANG, "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.7, Argonne National Laboratory (2016).
11. G. STEWART, "A Krylov-Schur algorithm for large eigenproblems," *SIAM Journal on Matrix Analysis and Applications*, **23**, *3*, 601–614 (2002).
12. Y. SAAD and M. H. SCHULTZ, "GMRES - A Generalized Minimal Residual algorithm for solving nonsymmetric linear-systems," *SIAM Journal on Scientific and Statistical Computing*, **7**, *3*, 856–869 (1986).