

Document downloaded from:

<http://hdl.handle.net/10251/120345>

This paper must be cited as:

Carreño, A.; Vidal-Ferrándiz, A.; Ginestar Peiro, D.; Verdú Martín, GJ. (2018). Block hybrid multilevel method to compute the dominant lambda-modes of the neutron diffusion equation. *Annals of Nuclear Energy*. 121:513-524. <https://doi.org/10.1016/j.anucene.2018.08.010>



The final publication is available at

<http://doi.org/10.1016/j.anucene.2018.08.010>

Copyright Elsevier

Additional Information

# Block hybrid multilevel method to compute the dominant $\lambda$ -modes of the neutron diffusion equation

A. Carreño<sup>a</sup>, A. Vidal-Ferràndiz<sup>a</sup>, D. Ginestar<sup>b</sup>, G. Verdú<sup>a,\*</sup>

<sup>a</sup>*Instituto Universitario de Seguridad Industrial, Radiofísica y Medioambiental,  
Universitat Politècnica de València, Camino de Vera, s/n, 46022, Valencia*

<sup>b</sup>*Instituto Universitario de Matemática Multidisciplinar,  
Universitat Politècnica de València, Camino de Vera, s/n, 46022, Valencia*

---

## Abstract

The dominant  $\lambda$ -modes associated with a nuclear reactor configuration describe the neutron steady-state distribution and its criticality. Furthermore, they are useful to develop modal methods to study reactor instabilities. Different eigenvalue solvers have been successfully used to obtain such modes, most of them are implemented reducing the original generalized eigenvalue problem to an ordinary one. Thus, it is necessary to solve many linear systems making these methods not very efficient, especially for large problems. In this work, the original generalized eigenvalue problem is considered and two block iterative methods to solve it are studied: the block inverse-free preconditioned Arnoldi method and the modified block Newton method. All of these iterative solvers are initialized using a block multilevel technique. A hybrid multilevel method is also proposed based on the combination of the methods proposed. Two benchmark problems are studied illustrating the convergence and the competitiveness of the methods proposed. A comparison with the Krylov-Schur method and the Generalized Davidson is also included.

*Keywords:* Lambda modes, Neutron diffusion equation, Generalized eigenvalue problem, Multilevel methods, Newton's block method.

---

## 1. Introduction

The neutron diffusion equation is an approximation of the neutron transport equation that assumes that the neutron current is proportional to the gradient of the neutron flux by means of a diffusion coefficient. Given a configuration of a reactor core, its criticality can be forced dividing the neutron production by a positive number,  $\lambda$ , leading to a generalized differential eigenvalue problem known as the  $\lambda$ -modes problem (Stacey, 2007). This problem, considering that the energy is divided into two energy groups and that there is no up-scattering, can be expressed as

$$\mathcal{L}\phi = \frac{1}{\lambda}\mathcal{M}\phi, \quad (1)$$

---

\*Corresponding author

*Email addresses:* amcarsan@iqn.upv.es (A. Carreño), anvifer2@upv.es (A. Vidal-Ferràndiz), dginesta@mat.upv.es (D. Ginestar), gverdu@iqn.upv.es (G. Verdú)

*Preprint submitted to Annals of Nuclear Energy*

August 2, 2018

where the neutron loss operator is

$$\mathcal{L} = \begin{pmatrix} -\vec{\nabla}(D_1\vec{\nabla}) + \Sigma_{a_1} + \Sigma_{12} & 0 \\ -\Sigma_{12} & -\vec{\nabla}(D_2\vec{\nabla}) + \Sigma_{a_2} \end{pmatrix},$$

and the neutron production and the neutron flux are

$$\mathcal{M} = \begin{pmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \\ 0 & 0 \end{pmatrix}, \quad \phi = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}.$$

$\phi_1$  and  $\phi_2$  are neutron fast and thermal flux, respectively. The subindex in each coefficient indicates the corresponding energy group. Thus, the diffusion coefficients are  $D_1$  and  $D_2$ . The absorption cross sections are  $\Sigma_{a_1}$  and  $\Sigma_{a_2}$ . The values of  $\nu\Sigma_{f1}$  and  $\nu\Sigma_{f2}$  are the average number of neutrons produced in each fission multiplied by the fission cross sections. Finally,  $\Sigma_{12}$  is the scattering cross section from group 1 to group 2. The dominant eigenvalue of the problem established in Equation (1), known as the  $k_{\text{eff}}$ , and its corresponding eigenfunction describe the criticality and the steady state neutron distribution inside the core. The next eigenvalues and their corresponding eigenfunctions can be used to develop modal methods to study the time dependent neutron diffusion equation (Miró et al., 2002) and to classify BWR reactor instabilities (March-Leuba & Blaked, 1991). Because of this, the solution of the partial eigenvalue problem associated with the neutron diffusion equation, (1), that is, the computation of the dominant eigenvalues and their corresponding eigenfunctions has a great interest in nuclear reactor physics.

Several methods have been proposed for the spatial discretization of the neutron diffusion equation. Some of the most popular numerical techniques are the Nodal Collocation Method (NCM) Verdú et al. (1994), the Nodal Expansion Method (NEM) (Singh et al., 2014), the Analytical Nodal Method (ANM) (Hébert, 1987), the Finite Volume method (FVM) (Theler, 2013) and the Finite element methods (FEM) (Hébert, 2008) and all of them provide accurate results. In this work, a high order continuous Galerkin Finite Element Method is used, transforming the problem given in Equation (1) into an algebraic generalized eigenvalue problem (see more details in (Vidal-Ferrandiz et al., 2014)).

Different methods such as the subspace iteration method (Verdú et al., 1994; Vidal et al., 1998), the classical Arnoldi method, the Implicit Restarted Arnoldi method (IRAM) (Warsa et al., 2004; Verdú et al., 1999) or, more recently, the Krylov-Schur method (Vidal-Ferrandiz et al., 2014) have been used to solve this kind of algebraic eigenvalue problems. The application of these methods requires to transform the generalized problem into an ordinary eigenvalue problem (for two energy groups) or to apply a shift and invert technique. In both cases it is necessary to solve numerous linear systems associated with large matrices and the convergence of the methods is slow. This problem can be mitigated using the Jacobi-Davidson method, (Verdú et al., 2005), that also makes use of a shift and invert strategy, but it does not need to solve as many linear systems as the previous ones. Moreover, classical methods such as the shifted inverse iteration method for the computation of several eigenvalues make use of a deflation process and this has been shown to have a slow converge when it is compared with Krylov-Schur method (Bernal et al., 2017).

Other methods to solve eigenvalue problems associated with nonsymmetric matrices are the gradient type methods, that do not require solving linear systems. However, if there are clustered or degenerate eigenvalues, these methods may have problems to find all the eigenvalues. In practical situations of reactor analysis, the dominance ratio corresponding to the dominant eigenvalues is often near unity, resulting in a slow convergence. In such cases, block methods with

40 several initial approximated eigenvalues and eigenvectors are an alternative since their conver-  
 45 gence behaviour depends only on the separation of the group of target eigenvalues from the rest  
 of the spectrum. One of these methods is the Generalized Davidson that has been successfully  
 used for the computation of the modes in other approximations of the neutron transport equation  
 such as the multigroup  $SP_N$  equations (Hamilton & Evans, 2015). In this work, an inverse-free  
 Krylov subspace method introduced by Golub in (Golub & Ye, 2002) is used with a block imple-  
 mentation proposed by Quillen in (Quillen & Ye, 2010). This method improves the traditional  
 steepest descent method by expanding the search direction to a Krylov subspace with the advan-  
 tage of better approximation properties offered by Krylov subspaces. Usually, different types of  
 preconditioners are used to accelerate these iterative methods. In our case, the incomplete LU  
 50 preconditioner and a geometric multigrid preconditioner are studied.

On the other hand, Newton's methods have been shown very efficient in the computation of  
 eigenvalues in neutron diffusion theory. The modified block Newton method has been consid-  
 ered to solve the ordinary eigenvalue problem associated with the original generalized problem,  
 (González-Pintor et al., 2011) or directly in the generalized eigenproblem resolution, (Carreño  
 55 et al., 2017a). Furthermore, the use of Jacobian-Free Newton-Krylov methods has been stud-  
 ied using traditional methods as preconditioners (Gill & Azmy, 2009; Knoll et al., 2011; Gill  
 & Azmy, 2011). However, all of these methods are very sensitive to the initial guess and good  
 approximations are needed to initialize them. In this context, hybrid methods have been devel-  
 oped using slow convergence methods to initiate Newton's methods. The Jacobian-free Newton-  
 60 Krylov method with the IRAM has been investigated in (Gill & Azmy, 2009) and in (Mahadevan  
 & Ragusa, 2008).

In our work, a hybrid multilevel method, based on the combination of the preconditioned  
 block inverse-free Krylov method and the generalized modified block Newton method, is also  
 presented. This method uses two spatial meshes, a coarse and a fine mesh. The initial approx-  
 65 imations for the desired eigenvalues and their corresponding eigenfunctions are obtained from  
 the solution of the problem in a coarse mesh, using a block multilevel technique to improve  
 the computational speed to compute the solution of the partial eigenvalue problem in the fine  
 mesh. Other multilevel strategies for the solution of the neutron diffusion equation using a nodal  
 collocation method have been studied in (Ginestar et al., 2001).

70 The structure of the rest of the paper is as follows. In Section 2, the block methods studied  
 to solve the  $\lambda$ -modes problem are presented. In Section 2.1, the block inverse-free preconditioned  
 Arnoldi method is exposed and in Section 2.2 the Modified block Newton method is  
 briefly reviewed. In Section 2.3, the multilevel technique used to initiate the eigenvalue solver  
 is explained. Some numerical results for two different benchmark problems to analyze the block  
 75 methods performance are given in Section 3. In Section 4, a block hybrid method combining  
 an inverse-free block method with Newton's method is proposed and numerical results of the  
 performance of this hybrid method are also presented. Finally, the main conclusions of the paper  
 are summarized in Section 5.

## 2. Block eigenvalue methods

In this section, we present two block algorithms for finding the  $q$  dominant eigenvalues and  
 their corresponding eigenvectors of the generalized eigenvalue problem associated with the  $\lambda$ -  
 modes problem given by the Equation (1). This algebraic problem can be expressed as a generic  
 problem of the form

$$AX = BXA, \tag{2}$$

80 where  $A \in \mathbb{R}^{n \times n}$  and  $B \in \mathbb{R}^{n \times n}$  arise from the discretization of  $\mathcal{M}$  and  $\mathcal{L}$ , respectively. The matrix  $X \in \mathbb{R}^{n \times q}$  has the eigenvectors in its columns and  $\Lambda \in \mathbb{R}^{q \times q}$  is a diagonal matrix whose elements are the first  $q$  dominant eigenvalues. The number  $n$  denotes the number of degrees of freedom in the finite element method used for the spatial discretization of the continuous problem (1). We assume that the eigenvalues and their corresponding eigenvectors are real, even though it this  
 85 seems to be proved only under restrictive conditions (e.g., monoenergetic transport). If not, this theoretical assumption is nevertheless supported by numerical evidence on benchmark problems (Carney et al., 2014).

Note that the structure of the algebraic eigenvalue problem (2) is the same for other approximations of the neutron transport equation, even if the neutron loss and production operators are  
 90 different. Therefore, this method can be applied not only to the diffusion equation but also to other angular approximations of the neutron transport equation, such as the  $SP_N$ , the  $S_N$  (Lewis & Miller, 1984) or the  $P_N$  approximations (Capilla et al., 2005). Furthermore, this block implementation is interesting for the development of modal methods to integrate the transient equations (Miró et al., 2002), since it permits initializing all modes in each time step from the modes  
 95 computed in a previous time step. Perturbation calculations can be also improved using this initialization characteristic.

A block multilevel initialization based on two different meshes to initialize these methods for those cases we can not dispose of initial approximations is also presented.

### 2.1. Block inverse-free preconditioned Arnoldi method (BIFPAM)

100 The block inverse-free preconditioned Arnoldi method was originally presented and analyzed for  $A$  and  $B$  symmetric matrices and  $B$  positive definite (see (Golub & Ye, 2002; Quillen & Ye, 2010)). Nevertheless, we will show that this methodology also works efficiently to compute the  $\lambda$ -modes associated to two benchmark reactor problems, where matrices  $A$  and  $B$  are not symmetric.

We start with the problem for one eigenvalue

$$Ax = \lambda Bx, \quad (3)$$

105 and an initial approximation  $(\lambda_0, x_0)$ .

An iterative method is constructed where a new approximation,  $x_{k+1}$ , is obtained from the  $m$ -order Krylov subspace

$$K_m(A - \lambda_k B, x_k) := \text{span}\{x_k, (A - \lambda_k B)x_k, (A - \lambda_k B)^2 x_k, \dots, (A - \lambda_k B)^m x_k\},$$

by using the Rayleigh-Ritz projection method. The projection can be carried out by forming the projected problem for the generalized eigenvalue problem (3) as

$$Z^T AZU = Z^T BZU\Lambda, \quad (4)$$

where  $Z$  is a basis of  $K_m(A - \lambda_k B, x_k)$ , which is constructed by using the Arnoldi method (Saad, 1992). Problem (4) is small and the dominant eigenvalue  $\Lambda_{1,1}$ , together with its corresponding eigenvector  $u_1$  are computed to obtain the value of  $\lambda_{k+1} = \Lambda_{1,1}$  and its eigenvector  $x_{k+1} = Zu_1$ .

This method can be dealt with through an iteration with a block of vectors that allows computing several eigenvalues simultaneously, improving the convergence. If we are interested in

computing  $q$  eigenvalues of problem (2), we can accelerate the convergence by using the subspace  $\mathcal{K}_m$  with

$$\mathcal{K}_m := \bigcup_{i=1}^q K_m^i(A - \lambda_{k,i}B, x_{k,i}),$$

where  $\lambda_{k,i}$  denotes the  $i$ -th eigenvalue computed in the  $k$ -th iteration and  $x_{k,i}$  its associated eigenvector. One may construct a basis for this subspace by first constructing  $q$  bases, say  $Z_i$ , of the Krylov subspaces  $K_m^i(A - \lambda_{k,i}B, x_{k,i})$ ,  $1 \leq i \leq q$ . Then, the original generalized eigenvalue problem (2) is projected on the union of all bases and it is solved for the first  $q$  dominant eigenvalues to obtain the new eigenvalues and the corresponding eigenvectors.

The basic block inverse-free method (BIFAM) is summarized in Algorithm 1. In this implementation, in contrast to original Algorithm proposed in (Quillen & Ye, 2010), the orthonormalization of the union of subspaces is not considered in order to reuse the matrix-vector multiplications when forming  $A_m$  and  $B_m$  by utilizing  $AZ_i$  and  $BZ_i$ ,  $1 \leq i \leq q$ , which need to be computed in the construction of  $K_m^i$ .

---

**Algorithm 1** Block inverse-free Arnoldi method (BIFAM)

---

**Input:** Matrices  $A$  and  $B$ , initial approximation  $X_0 = [x_{0,1}, \dots, x_{0,q}]$ .

**Output:** Diagonal matrix of eigenvalues  $\Lambda$  and matrix  $V$  with the eigenvectors as its columns.

- 1: Compute  $\lambda_{0,i} = (x_{0,i}^T A x_{0,i}) / (x_{0,i}^T B x_{0,i})$ ,  $1 \leq i \leq q$
  - 2: **for**  $k = 1$  to *maxits* **do**
  - 3:     Obtain the basis  $Z_i$  of  $K_m^i(A - \lambda_{k-1,i}B, x_{k-1,i})$ ,  $1 \leq i \leq q$
  - 4:     Construct  $Z := [Z_1, \dots, Z_q]$
  - 5:     Form projections  $A_m = Z^T A Z$ ,  $B_m = Z^T B Z$
  - 6:     Compute  $q$  dominant eigenpairs  $(\lambda_{k,i}, u_i)$  of  $A_m U = B_m U \Lambda$
  - 7:     Compute  $x_{k,i} = Z u_i$ ,  $1 \leq i \leq q$
  - 8: **end for**
- 

The rate of convergence of this method improves when the dimension of subspace,  $m$ , is increased. However, the computational cost is also considerably increased. In this way, alternatively the BIFAM will be accelerated with an equivalent transformation of the original problem by means of a preconditioner.

The idea of preconditioning is based on a convergence result of the method, presented in (Golub & Ye, 2002) which states that the rate of convergence of the method depends on the spectral distribution of  $C = A - \lambda B$ , being  $\lambda$  the desired eigenvalue.

With an approximate eigenpair  $(\lambda_{i,k}, x_{i,k})$ , we consider for some matrices  $L_{i,k}$ ,  $U_{i,k}$  the transformed eigenvalue problem

$$(L_{i,k}^{-1} A U_{i,k}^{-1})x = \lambda (L_{i,k}^{-1} B U_{i,k}^{-1})x \Leftrightarrow \hat{A}_{i,k}x = \lambda \hat{B}_{i,k}x, \quad (5)$$

which has the same eigenvalues as the original problem. This transformation is called preconditioning. Thus, applying one step of Algorithm 1 to the problem (5), the rate of convergence will be determined by the eigenvalues of

$$\hat{C}_{i,k} := \hat{A}_{i,k} - \lambda_{i,k} \hat{B}_{i,k} = L_{i,k}^{-1} (A - \lambda_{i,k} B) U_{i,k}^{-1}. \quad (6)$$

Different preconditioning transformations can be constructed using different factorizations of the matrix  $A - \lambda_{i,k} B$ . The main goal must be to choose suitably  $L_{i,k}$  and  $U_{i,k}$  to obtain a favorable distribution of the eigenvalues of matrix  $\hat{C}_{i,k}$ .

Moreover, the preconditioned iteration of block inverse-free Arnoldi method (BIFPAM) can be implicitly implemented by constructing a basis for the subspace  $U_{i,k}^{-1}\hat{K}_m^i$ , where

$$\hat{K}_m^i := \text{span}\{U_{i,k}x_{i,k}, \hat{C}_{i,k}U_{i,k}x_{i,k}, \hat{C}_{i,k}^2U_{i,k}x_{i,k}, \dots, \hat{C}_{i,k}^mU_{i,k}x_{i,k}\}.$$

In practice, we use constants  $L_{i,k} = L_{1,1}$  and  $U_{i,k} = U_{1,1}$  obtained from a preconditioner for  $A - \lambda_{1,1}B$ , where  $\lambda_{1,1}$  is a first approximation of the first eigenvalue.

Two preconditioners have been considered in this paper. The first one is a classical incomplete LU factorization with level 0 of fill in which is referred as the ILU preconditioner. The second one is a geometrical multigrid preconditioner (GMG) used to solve linear systems. In particular, it is a V-cycle multigrid method with two meshes (coarse and fine meshes), smoothing with a Gauss-Seidel iterative method, similar to the strategy developed in (Neymeyr, 2003). This preconditioner is implemented using the matrices and operators that have already been defined to initialize the iterative block method as it is explained below. Using the notation introduced in Section 2.3, the application of this preconditioner is summarized in Algorithm 2. In our implementation, the number of iterations for the smoothers,  $r$ , is set to  $r = 3$ . The solution of the linear system associated with the coarse mesh is computed with low precision.

---

**Algorithm 2** Geometric multigrid preconditioner (GMG)

---

**Input:** Vector  $b$ .

**Output:** Vector  $y = M^{-1}b$  con  $M$  preconditioner of  $D = A - \lambda B$ .

- 1: Pre-smooth with  $r$  iterations of Gauss-Seidel on  $Dy = b$  (Initialize the iterative method with  $y = 0$ )
  - 2: Restrict the residual  $r = Dy - b$  to the coarse mesh by  $r_c = \mathcal{R}(r)$
  - 3: Solve  $D_c e_c = r_c$  with  $D_c = A_c - \lambda B_c$
  - 4: Prolongate  $e_c$  by  $e = \mathcal{P}(e_c)$
  - 5: Correct  $y = y + e$
- 

## 2.2. Modified generalized block Newton method (MGBNM)

Another kind of block methods for eigenvalue computations are based on Newton's method. The modified block Newton method for ordinary eigenvalue problems was proposed by Lösche in (Lösche et al., 1998) and it was applied to the neutron diffusion equation in (González-Pintor et al., 2011). In this section, we briefly review an extension of this method for generalized eigenvalue problems given by the authors in (Carreño et al., 2017b).

To apply this method to the problem (2), we assume that the eigenvectors can be factorized as

$$X = ZS, \tag{7}$$

where  $Z^T Z = I_q$ . Then, problem (2) can be rewritten as

$$AX = BX\Lambda \Rightarrow AZS = BZS\Lambda \Rightarrow AZ = BZS\Lambda S^{-1} \Rightarrow AZ = BZK. \tag{8}$$

It is necessary to introduce a biorthogonality condition  $W^T Z = I_q$  in order to determine the problem, where  $W$  is a fixed matrix of rank  $q$ . Newton's method is used to solve the following problem

$$F(Z, \Lambda) := \begin{pmatrix} AZ - BZK \\ W^T Z - I_q \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{9}$$

Thus, a new iterated solution arises as,

$$Z^{(k+1)} = Z^{(k)} - \Delta Z^{(k)}, \quad K^{(k+1)} = K^{(k)} - \Delta K^{(k)}, \quad (10)$$

where  $\Delta Z^{(k)}$  and  $\Delta K^{(k)}$  are solutions of the system that is obtained substituting (10) into (9) and removing second order terms.

The system obtained is coupled, since the matrix  $K^{(k)}$  is not necessarily a diagonal matrix. To decouple the system, the Modified Block Newton method performs two previous steps. The first step consists of an orthogonalization of the columns of the matrix  $Z^{(k)}$  using the modified Gram-Schmidt Orthogonalization. In the second step, a Rayleigh-Ritz projection method for generalized eigenvalue problems is applied (Saad, 1992). More details on the method can be found in (Carreño et al., 2017b).

### 2.3. Multilevel initialization

A block multilevel technique is proposed to obtain good initial approximations of several eigenvectors for the block iterative solvers when we can not dispose of initial approximate solutions. This technique is based on two meshes: the fine mesh, which is the final mesh considered to solve the problem that comes from the spatial discretization and, a coarse mesh, with less number of nodes, constructed from the fine mesh to obtain an initial approximation for the problem. This multilevel method will strongly depend on the geometry of the underlying meshes.

A complete description of the multilevel concepts can be found, for instance, in (McCormick, 1987; Sampath & Biros, 2010; Hackbusch, 2013). However, some basic concepts on multilevel techniques are reviewed.

Let us consider the original algebraic partial eigenvalue problem

$$A^f X^f = B^f X^f \Lambda^f, \quad (11)$$

where  $A^f$  and  $B^f$  are the matrices that arise from the discretization of a given domain  $\Omega$  using the fine mesh  $\Omega^f$ . From this mesh a coarser mesh,  $\Omega^c$ , is constructed and a new eigenvalue problem

$$A^c X^c = B^c X^c \Lambda^c, \quad (12)$$

is considered, where  $A^c$  and  $B^c$  are the matrices associated with the spatial discretization using the mesh  $\Omega^c$ . This algebraic problem has a smaller dimension than the initial one. To assembly matrices  $A^c$  and  $B^c$ , besides the coarsening of the initial spatial discretization, the cross sections must be homogenized. In each coarse cell the value of each cross section  $\Sigma_d$  is computed as a volume average,

$$\Sigma_d = \frac{1}{V_d} \sum_{i=1}^m V_{d_i} \Sigma_{d_i}, \quad (13)$$

where  $d$  is equal to the union of cells  $d_1, \dots, d_m$ , i.e.  $d = \cup_{i=1}^m d_i$  and  $\Sigma_{d_i}$  is the value of the cross section in cell  $d_i$ .  $V_d$  is the volume of the coarse cell  $d$  and  $V_{d_i}$  the volume of the cell  $d_i$ .

To use the eigenvectors  $X^c$  as an initial guess for the problem associated with the finer grid  $\Omega^f$ , we define a linear operator called prolongation operator,  $\mathcal{P}$ , that interpolates vectors defined on the coarse mesh onto the fine mesh. Similarly, to represent the vectors  $X^f$  on the coarse grid,  $\Omega^c$ , we use the restriction operator  $\mathcal{R}$ . Both operators are the standard prolongation and restriction operators considered for finite element methods in other works (Janssen & Kanschat, 2011). They are performed with a stencil computed by interpolation on each coarse mesh cell



by using interpolating polynomials of the same degree as the one used in the high order finite element approximation.

175 The multilevel initialization is structured as follows: the coarse problem (12) is solved for the largest  $q$  eigenvalues and the eigenvectors obtained are projected onto the fine mesh to initialize the block method. This strategy is implemented as it is shown in Algorithm 3.

---

**Algorithm 3** Multilevel initialization

---

- 1: Construct the coarse problem  $A^c X^c = B^c X^c \Lambda^c$
  - 2: Solve  $A^c X^c = \lambda^c B^c X^c$
  - 3: Prolongate  $X_0^f = \mathcal{P}(X^c)$
  - 4: Solve  $A^f X^f = B^f X^f \Lambda^f$  using as initial guess  $X_0^f$  ▷ BLOCK METHOD
- 

### 3. Numerical results of block methods

180 This section studies the competitiveness of the BIFPAM and MGBNM methods on two different problems: a theoretical heterogeneous cuboid reactor and a version of the 3D NEACRP reactor (Finnemann & Galati, 1991). The diffusion equation for all cases has been discretized using the high order finite element method with polynomials of degree 3. The number of eigenvalues computed has been  $q = 4$  for each problem.

185 The solutions of linear systems needed to apply the MGBNM have been computed with the GMRES method using the ILU preconditioner, and a previous reordering of the matrix to minimize the fill-in using the Cuthill-McKee method.

As block methods need an initial approximation of a set of eigenvectors, the multilevel initialization proposed in Section 2.3 is used to compare the block methods performance. However, the multilevel initialization has been also compared with other initialization strategies to study 190 the effectiveness of this technique (Section 3.3).

The stopping criteria for all solvers has been set equal of  $10^{-6}$  in the global residual error, defined as

$$\text{res} = \sqrt{\sum_{i=1, \dots, q} \|Ax_i - \lambda_i Bx_i\|^2},$$

where  $\lambda_i$  is the  $i$ -th eigenvalue and  $x_i$  its associated unitary eigenvector. The criteria chosen is related to the convergence of the eigenvectors because the eigenvalues converge in a faster way..

The methods have been implemented in C++ based on data structures provided by the library Deal.II (Bangerth et al., 2007), PETSc (Balay et al., 2016) and SLEPc (Hernandez et al., 2005). 195 The computer used for the computations has been an Intel® Core™ i7-4790 @3.60GHz×8 processor with 32Gb of RAM running on Ubuntu GNU/Linux 16.04 LTS.

#### 3.1. Cuboid reactor

200 Even though this problem is completely theoretical, it is relevant to analyze the computational improvement of the multilevel scheme proposed using a mesh that can be largely coarsened (8 cells in the fine mesh are joint to form 1 cell in the coarse one). The cuboid reactor has been designed with 8000 different cells with the same size (20 cm in each axis). The dimensions of the reactor are 4 m × 4 m × 4 m. The geometry and the material distribution for one plane are presented in Figure 1. This plane configuration is repeated from the 3rd to the 18th plane. The

205 cells in the rest of planes are assembled with the material 1, modelling the reflector. The nuclear cross sections of each material are given in Table 1. The average number of neutrons produced in each fission,  $\nu$ , is assumed to be constant in each cell and equal to  $\nu = 2.5$ . Zero flux boundary conditions have been assumed to solve the problem.

The fine and coarse meshes considered in this problem are represented in Figure 2. Moreover, the number of cells in each mesh (n. cells) is indicated. The neutron diffusion equation discretization, by using polynomials of degree 3, provides a number of 453 962 degrees of freedom using the fine mesh and 59 582 degrees of freedom with the coarse mesh. The eigenvalues computed are, in decreasing order,  $\lambda_1 = 0.986215$ ,  $\lambda_2 = 0.939683$ ,  $\lambda_3 = 0.935521$  and  $\lambda_4 = 0.935521$ . It means that the spectrum of these modes is very clustered. The third and fourth eigenvalue are equal since they are degenerate due to the reactor radial symmetry (for more details, see 215 (Tommasi et al., 2016)).

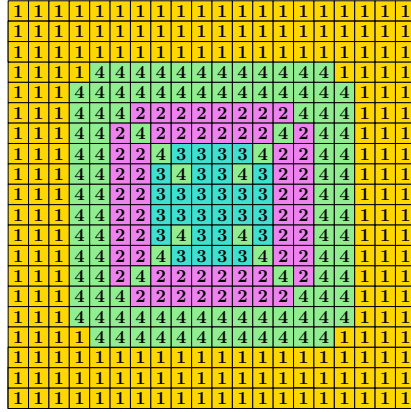


Figure 1: Material distribution of the cuboid reactor.

Table 1: Macroscopic cross sections of the cuboid reactor.

Mat.	$D_1$ (cm)	$D_2$ (cm)	$\Sigma_{a1}$ (cm <sup>-1</sup> )	$\Sigma_{a2}$ (cm <sup>-1</sup> )	$\Sigma_{12}$ (cm <sup>-1</sup> )	$\nu\Sigma_{f1}$ (cm <sup>-1</sup> )	$\nu\Sigma_{f2}$ (cm <sup>-1</sup> )
1	1.6342	2.6400e-01	2.6660e-03	4.9363e-02	2.7596e-02	0.0000	0.0000
2	1.4239	3.5630e-01	1.0952e-02	9.1462e-02	1.7555e-02	6.4777e-03	1.1273e-01
3	1.4239	3.5630e-01	1.0402e-02	8.7662e-02	1.7555e-02	2.5910e-03	4.5093e-02
4	1.42531	3.5055e-01	1.0992e-02	9.9925e-02	1.7177e-02	7.5032e-03	1.3780e-01

220 First, we study the performance of the block inverse-free preconditioner Arnoldi method (BIFPAM). Table 2, shows a comparison of the number of iterations (Iterations) and CPU time needed to solve the problem with the BIFPAM using different dimensions,  $m$ , of the Krylov subspace without preconditioner, with the ILU preconditioner and with the geometric multigrid preconditioner (GMG). We observe that the BIFPAM without preconditioner does not reach the tolerance in the maximum number of iterations set to 100 iterations. If this value (or the value of  $m$ ) increases it can be observed that this method reaches a residual error of  $10^{-7}$  and the rate of convergence improves when the dimension of the Krylov subspace ( $m$ ) is larger. Then, by comparing the results obtained with the BIFPAM with preconditioner, it can be observed that 225 the number of iterations decreases as the dimension increases. However, the number of matrix-

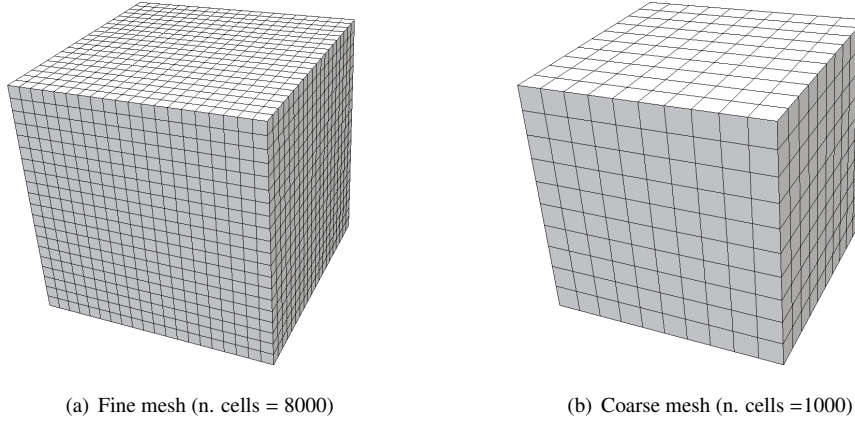


Figure 2: Meshes for 3D cuboid reactor

vector multiplications and applications of the preconditioner needed in each iteration increases too. By using the ILU preconditioner, there are not big differences in the computational times with the Krylov subspace dimensions  $m = 8$  and  $m = 10$ . With the multigrid preconditioner, the computational time does not significantly change when the value of  $m$  is increased. Furthermore, if the preconditioners of BIFPAM are compared, the best results are obtained using the GMG, although there are not big differences with the best results obtained with ILU preconditioner ( $m = 8$ ,  $m = 10$ ). This is because when we apply the GMG preconditioner, there are not big differences in the number of iterations and it is expensive to apply. So, the best results are obtained with  $m = 4$ . With ILU preconditioner, we have the opposite results because, this is a cheaper preconditioner to apply, but the number of iterations with  $m = 6$  differs a lot of the number of iterations needed when  $m = 8$ . Finally, note that the GMG preconditioner works efficiently even when the dimension of the Krylov subspace is  $m = 4$ , which implies to use less computational memory.

The convergence history for the MGBNM and BIFPAM methods, with the most efficient configurations are shown in Table 2, and are represented in the Figure 3. In the MGBNM the subspace dimension used has been the same as the number of eigenvalues requested. In this Figure, we note the sawtooth convergence of the BIFPAM. For the first iterations, this convergence is faster than the MGBNM one, but there is a point where the residual error increases and then continues down. This makes that the MGBNM reaches the required tolerance in a faster way.

Finally, the Krylov-Schur method and the Generalized Davidson method implemented in the SLEPc library (Hernandez et al., 2005) have been analyzed and compared with the techniques proposed in this paper. These methods are not initialized with any approximation to determine which solver works better to obtain the solution in the multilevel initialization. The Krylov-Schur method needs to solve linear systems to transform the generalized eigenvalue problem into an ordinary eigenvalue problem. For this reason, two methods to solve linear system with non-symmetric matrices have been tested: the Generalized Minimal Residual method (GMRES) and the stabilized version of BiConjugate Gradient method (BCGS) by comparing the computational times. Both solvers have been implemented using the PETSc (Balay et al., 2016) library with the

Table 2: Results of iterations and CPU times needed by the BIFPAM to solve the cuboid problem without preconditioner, with the ILU preconditioner and with the GMG preconditioner and several dimensions of the Krylov subspace.

Krylov subspace dimension ( $m$ )	Iterations	CPU time (s)
BIFPAM without preconditioner		
4	100*	558*
6	100*	770*
8	100*	992*
10	100*	1205*
BIFPAM with ILU preconditioner		
4	82	607
6	39	407
8	22	294
10	17	284
BIFPAM with GMG preconditioner		
4	14	261
6	10	282
8	7	264
10	6	279

\* Note: The method does not reach the tolerance in the maximum number of iterations set to 100.

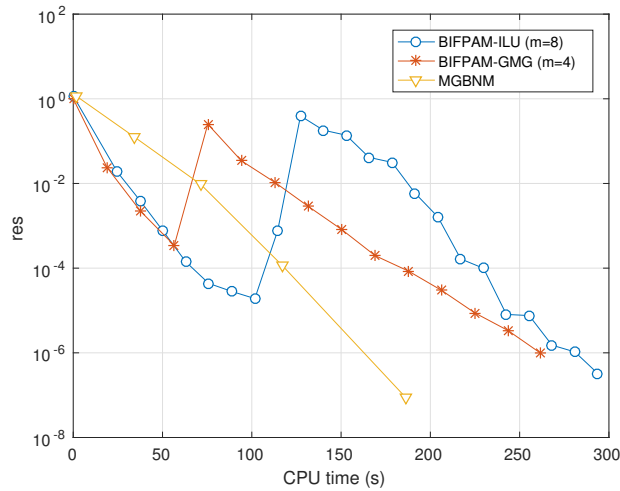


Figure 3: Convergence history for the fourth dominant eigenvalues of the cuboid problem using the MGBNM and the BIFPAM with the ILU and GMG preconditioners.

255 ILU preconditioner, Cuthill-McKee reordering and a tolerance in the residual of  $10^{-8}$ . The efficiency of the Krylov-Schur method is very dependent on the dimension of the Krylov subspace used. Therefore, different values for the Krylov subspace dimension ( $m$ ) are studied. In the same way, the Generalized Davidson method depends on the maximum number of projected vectors

(*mpd*) used. Then, different values of *mpd* were set to test the best configurations. The ILU preconditioner is used with the Davidson method. The number of iterations and the CPU time that the Krylov-Schur method and the Generalized Davidson method need to reach the tolerance ( $10^{-7}$ ) are presented in Table 3.

First, the differences between the linear system solvers used in the Krylov-Schur method are analyzed. By comparing the computational times it can be checked that the GMRES method computes the solution of the linear systems in less time than BCGS method. If, now, we are interested in comparing the dimensions of the subspace for both methods, we observe that the Krylov-Schur method needs a dimension of  $m = 24$  or larger to obtain competitive results, while for the Generalized Davidson, good results are obtained setting a maximum number of projected vector (*mpd*) greater than 16.

Table 3: Results of iterations and CPU times needed by the Krylov-Schur and the Generalized Davidson to solve the cuboid problem.

Dimension	Iterations	CPU time (s)
Krylov-Schur with GMRES		
12	16	283
16	11	271
20	9	279
24	7	266
Krylov-Schur with BCGS		
12	16	325
16	11	334
20	9	322
24	7	312
Generalized Davidson		
8	606	164
12	283	120
16	199	107
20	186	103

### 3.2. NEACRP reactor

The NEACRP benchmark (Finnemann & Galati, 1991) is chosen to compare the proposed methodology with a more realistic benchmark. The reactor core has a radial dimension of 21.606 cm  $\times$  21.606 cm per assembly. Axially the reactor, with the total height of 427.3 cm, is divided into 18 layers with height (from bottom to top): 30.0 cm, 7.7 cm, 11.0 cm, 15.0 cm, 30.0 cm (10 layers), 12.8 cm (2 layers), 8.0 cm and 30.0 cm. The boundary condition is zero flux in the outer reflector surface. The material distribution is shown in Figure 4. The material cross-sections are displayed in Table 4.

Figure 5 shows the fine mesh used for the spatial discretization considered to solve the problem (Figure 5(a)) and the coarse mesh used to initialize the iterative methods (Figure 5(b)). The fine mesh and the coarse mesh considered have 3978 and 1308 cells, respectively. In this way, the problem associated with the fine mesh has 230 120 degrees of freedom and the one associated with the coarse mesh has 78 440 degrees of freedom because cubic polynomials have been used.

The four dominant eigenvalues computed are  $\lambda_1 = 1.00200$ ,  $\lambda_2 = 0.988620$ ,  $\lambda_3 = 0.985406$  and  $\lambda_4 = 0.985406$ , giving a dominance ratio of  $|\lambda_2/\lambda_1| = 0.986647$ . Figure 6 shows the power distribution for the four dominant modes of this reactor.

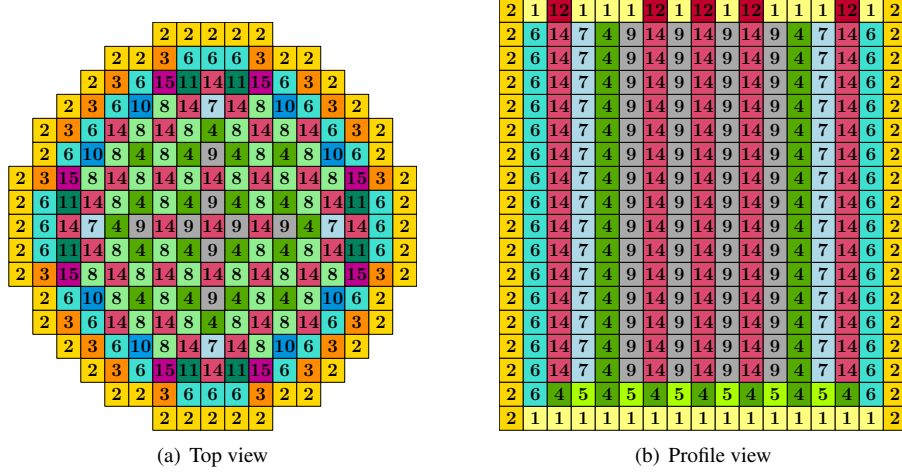


Figure 4: Geometry and distribution of materials of NEACRP reactor.

Table 4: Macroscopic cross section of the NEACRP reactor.

Mat.	$D_1$ (cm)	$D_2$ (cm)	$\Sigma_{a1}$ (cm $^{-1}$ )	$\Sigma_{a2}$ (cm $^{-1}$ )	$\Sigma_{12}$ (cm $^{-1}$ )	$\Sigma_{f1}$ (cm $^{-1}$ )	$\Sigma_{f2}$ (cm $^{-1}$ )	$\nu\Sigma_{f1}$ (cm $^{-1}$ )	$\nu\Sigma_{f2}$ (cm $^{-1}$ )
1	5.9264	8.2289e-01	2.5979e-04	1.7085e-01	2.7988e-02	0.0000	0.0000	0.0000	0.0000
2	1.1276	1.7053e-01	1.1878e-03	1.9770e-01	2.3161e-02	0.0000	0.0000	0.0000	0.0000
3	1.1276	1.7053e-01	1.1878e-03	1.9770e-01	2.0081e-02	0.0000	0.0000	0.0000	0.0000
4	1.4624	3.9052e-01	8.4767e-03	6.2569e-02	1.9686e-02	6.1479e-14	1.1515e-12	5.0150e-03	8.7712e-02
5	1.4637	3.9485e-01	8.8225e-03	6.9978e-02	1.9436e-02	6.9275e-14	1.3685e-12	5.6085e-03	1.0424e-01
6	1.4650	3.9851e-01	9.1484e-03	7.6850e-02	1.9196e-02	7.6811e-14	1.5694e-12	6.1819e-03	1.1954e-01
7	1.4641	4.0579e-01	9.0869e-03	7.7687e-02	1.8526e-02	6.8996e-14	1.3509e-12	5.5830e-03	1.0289e-01
8	1.4642	4.0946e-01	9.1738e-03	8.0302e-02	1.8223e-02	6.8913e-14	1.3433e-12	5.5741e-03	1.0232e-01
9	1.4642	4.1314e-01	9.2596e-03	8.2924e-02	1.7920e-02	6.8817e-14	1.3351e-12	5.5650e-03	1.0169e-01
10	1.4653	4.0919e-01	9.4097e-03	8.4462e-02	1.8288e-02	7.6530e-14	1.5501e-12	6.1564e-03	1.1807e-01
11	1.4655	4.1277e-01	9.4956e-03	8.7030e-02	1.7986e-02	7.6449e-14	1.5419e-12	6.1474e-03	1.1744e-01
12	5.5576	8.7013e-01	2.7375e-03	1.9644e-01	2.4796e-02	0.0000	0.0000	0.0000	0.0000
13	5.6027	8.6371e-01	2.4169e-03	1.9313e-01	2.5209e-02	0.0000	0.0000	0.0000	0.0000
14	1.4389	4.0085e-01	1.0954e-02	8.8157e-02	1.6493e-02	6.0265e-14	1.1145e-12	4.9122e-03	8.4889e-02
15	1.4413	4.0665e-01	1.1578e-02	1.0250e-01	1.6054e-02	7.5335e-14	1.5263e-12	6.0593e-03	1.1626e-01

285 Initially, the numerical results obtained with the BIFPAM by using the ILU and GMG preconditioners and without preconditioning for  $m = 4$  and  $m = 8$  are compared. Similar conclusions to the ones obtained for the cuboid reactor for this problem are obtained. Figure 7 displays the convergence histories for the BIFPAM with the configurations considered. It is observed that the preconditioner improves the rates of convergence and attenuates the oscillations produced in the convergence histories. Between the different preconditioners used, this Figure shows that the ILU preconditioner (with  $m = 8$ ) and the GMG preconditioner (with  $m = 4$ ) reach the tolerance almost at the same time. The reason why the ILU preconditioner works better with higher dimensions and the GMG preconditioner with lower ones is the same as the explanation exposed in the numerical results presented for the cuboid reactor. The use of these preconditioners with 290 the other values considered for the dimension ( $m$ ) are less efficient.

295 The convergence histories of the MGBNM and the BIFPAM with the ILU and GMG pre-

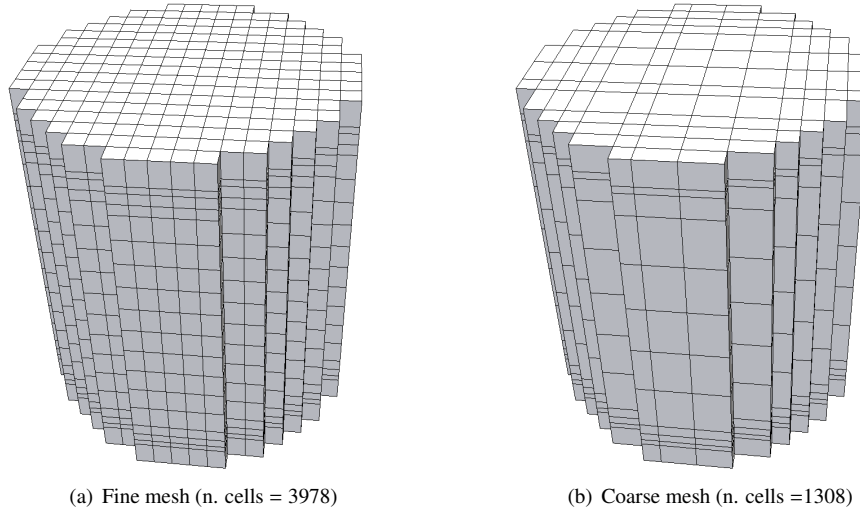


Figure 5: Meshes for NEACRP reactor.

conditioner are compared in Figure 8. For this reactor, it is deduced that the desired tolerance is reached quicker with the MGBNM. We would like to highlight that the convergence behaviour of BIFPAM-ILU is very similar to the one of BIFPAM-GMG and when the residual becomes smaller the convergence of the Newton method becomes faster.

### 3.3. Numerical results to test the multilevel initialization

In this section, we are going to compare the multilevel initialization proposed in the Section 2.3 with other two initialization alternatives for the block methods studied in this paper. The NEACRP reactor has been used to make this study. The Generalized Davidson with  $mpd = 16$  has been used to compute the solution in the coarse mesh, since we have checked in Section 3.1 that this solver (with this configuration and without initialization) converges faster than the Krylov-Schur method. The BIFPAM is used with the GMG preconditioner and  $m = 4$ .

We are interested in obtaining a good initial guess  $X^0 = \{x_1, \dots, x_q\}$  (closer to the solution) for the  $q$  dominant eigenvalues of the  $\lambda$ -modes problem at a computational cost as low as possible. The first initialization choice, that we denote as Random, is to generate the  $q$  vectors using random numbers on the interval  $[-1, 1]$ . The second one is to estimate the  $q$  vectors using the Krylov subspace generated by the matrix  $L^{-1}M$  acting on an initial vector (we use an all-ones vector). The Arnoldi method has been used to obtain this subspace. The dimension of the subspace has been  $m = 10$  for the initialization of 4 eigenvectors. We called to this strategy as Krylov initialization. Before using both initializations, the resulting system of vectors are orthonormalized by using the modified Gram-Schmidt process. Then, the Rayleigh-Ritz algorithm for the generalized eigenvalue problem is applied.

Figure 9 shows the convergence histories for the BIFPAM and the MBGNM initialized in several ways: the multilevel, the Random and the Krylov initialization. In the multilevel initialization times, the CPU time to assemble and solve the eigenvalue problem in the coarse mesh, that is 21s, has been included. In the CPU times obtained with the Krylov initialization, the time

to compute the subspace (18s) has been added. Both graphics reflect that the multilevel initialization, although it takes more time to obtain the initial guess than the other initializations, is a better strategy to initialize the block methods.

#### 325 4. Block hybrid eigenvalue solver

The previous numerical results show that the BIFPAM converges efficiently using the ILU or the GMG preconditioner. However, the method for very accurate approximations exhibits a slower convergence history or the convergence rate decreases after a certain number of iterations. Theoretically, we know that the MGBNM converges quadratically. However, it needs a suitable  
 330 initial guess, otherwise in the first iterations has a slow convergence. Thus, it is proposed to use a hybrid scheme combining both methods. This hybrid scheme is based on using the BIFPAM in the first iterations until a given tolerance of  $res = 10^{-3}$  is reached and then applying the MGBNM. This cut off value for the tolerance is reactor dependent and it has been chosen by observing the convergence history of the methods. The initial guess for this method is computed  
 335 with the multilevel technique. The implementation of this block hybrid multilevel method can be summarized in the Algorithm 4.

---

#### Algorithm 4 Block hybrid multilevel method

---

- 1: Apply the multilevel method to obtain the initial guess
  - 2: **while**  $res \geq 10^{-3}$  **do**
  - 3:     Apply the BIFPAM
  - 4: **end while**
  - 5: **while**  $res \geq tol$  **do**
  - 6:     Apply the MGBNM
  - 7: **end while**
- 

##### 4.1. Numerical results for the hybrid method

In this section, the same benchmark reactors as the ones presented in the Section 3 are used to demonstrate the effectiveness of the hybrid multilevel method with respect to other eigenvalue  
 340 solvers. The BIFPAM has been set with the ILU preconditioner and dimension of the Krylov subspace  $m = 8$ . Even though, similar results are obtained by setting the BIFPAM with the GMG preconditioner and dimension  $m = 4$ .

First, we compare the hybrid scheme with the MGBNM and the BIFPAM with ILU preconditioner. All methods have been initialized with the same initial guess obtained with the multilevel  
 345 technique. Figure 10 displays the results obtained. They show that the hybrid algorithm is the best scheme to compute several eigenvalues of the benchmark problems analyzed. Table 5 shows the number of iterations and computational times that spend the BIFPAM and MGBNM in the hybrid method. It can be observed that we need more iterations of BIFPAM than of the MGBNM but the CPU time needed for each solver changes with the problem.

350 Finally, these methods are compared with the Krylov-Schur method and the Generalized Davidson for the NEACRP reactor. The same conclusions for the cuboid reactor are obtained. The dimension of the Krylov subspace used in the Krylov-Schur method has been  $m = 24$ . The solutions of the linear systems needed in the Krylov-Schur method have computed with GMRES



Table 5: Number of iterations and CPU times of the hybrid method.

Benchmark	Iterations		CPU time (s)	
	BIFPAM	MGBNM	BIFPAM	MGBNM
Cuboid	3	1	50.2	74.3
NEACRP	5	1	38.4	29.9

method, with the ILU preconditioner with a previous reordering of the matrix with the Cuthill-McKee method. The maximum number of projected vectors for the Generalized Davidson is set equal to  $mpd = 16$  and the ILU preconditioner has been used. Furthermore, the Generalized Davidson can be initialized in block, by using the same initialization strategy as for the BIFPAM and the MGBNM. Table 6 displays the total computational times obtained for different numbers of eigenvalues. This CPU time includes the time needed to assemble the matrices and to obtain the initial guess. From this Table, we deduce that the block methods (G. Davidson, BIFPAM, MGBNM and Hybrid) compute the eigenvalues faster than the Krylov-Schur method even in the computation of one eigenvalue. If we compare the methods presented in this work (BIFPAM, MGBNM and Hybrid), we observe that the strategy of using a hybrid method improves the computational times obtained with the methods separately. Finally, it can be observed that the computational times obtained with the Generalized Davidson and the hybrid method, are similar. Note that for 6 eigenvalues, the MGBNM does not reach the tolerance in the maximum number of iterations, but if we use the MGBNM in the hybrid scheme, the method converges. This shows that the methods studied in this work are an interesting alternative for the computation of  $\lambda$ -modes to classical methods such as the Krylov-Schur method that, with the SLEPc implementation, converges a block of eigenvectors separately or the Generalized Davidson method that needs a high dimension of the subspace to be competitive. Nevertheless, there are other implementations of the Krylov-Schur method that allow a treatment of the eigenvalues in block (Baker et al., 2009), but we have not studied them in this paper.

Table 6: Computational times (s) obtained for the NEACRP reactor using the Krylov-Schur method, the Generalized Davidson, the BIFPAM, the MGBNM and the hybrid method for different number of eigenvalues.

n. eigs ( $q$ )	Krylov-Schur	G. Davidson	BIFPAM	MGBNM	Hybrid
1	100	59	57	66	58
2	142	79	134	86	78
3	143	90	169	103	92
4	236	100	126	119	105
5	235	114	157	135	123
6	255	130	184	*	135

\* Note: The method does not converge.

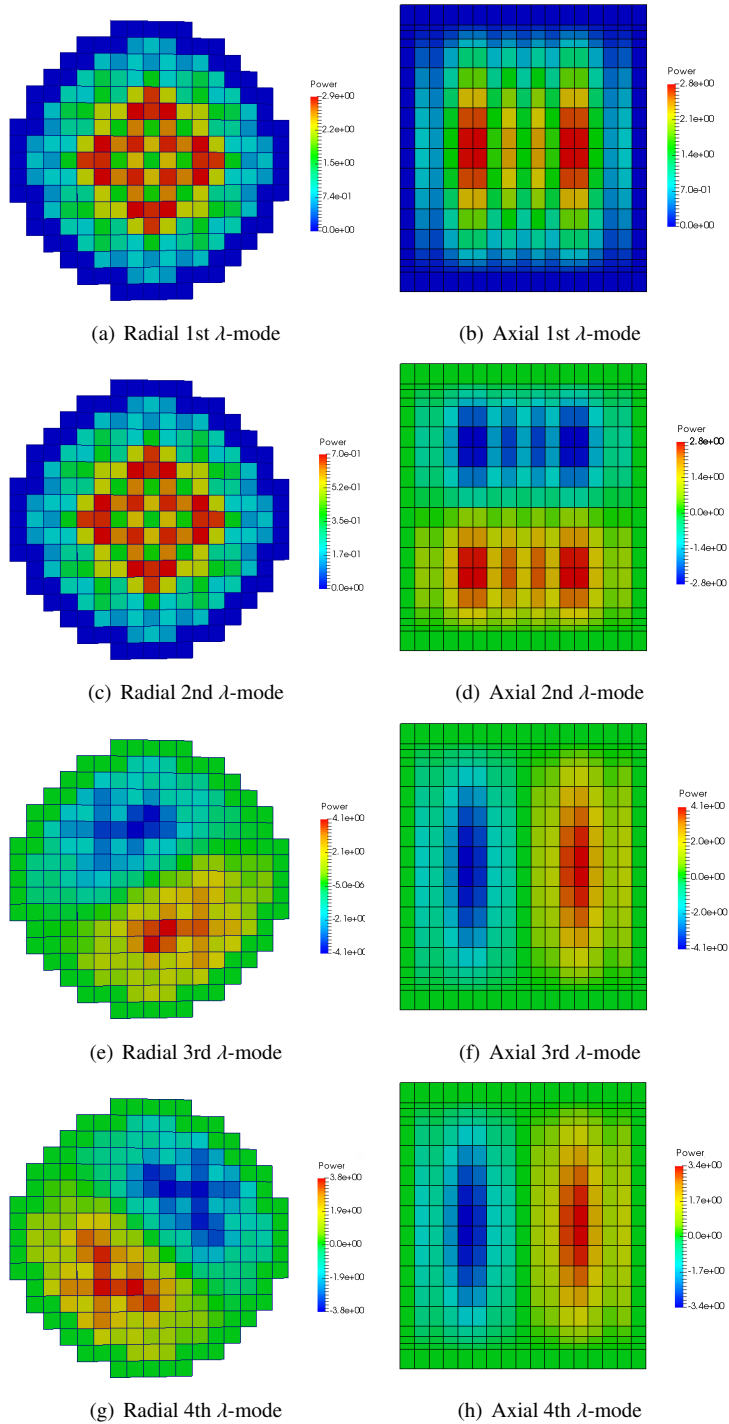


Figure 6: Radial and axial power profiles for NEACRP reactor.

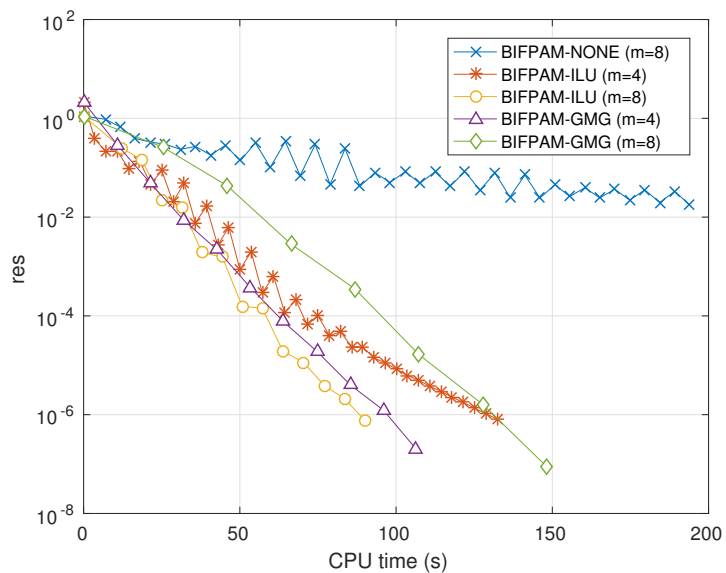


Figure 7: Residual errors against CPU time (s) of the BIFPAM with different configurations for the NEACRP problem.

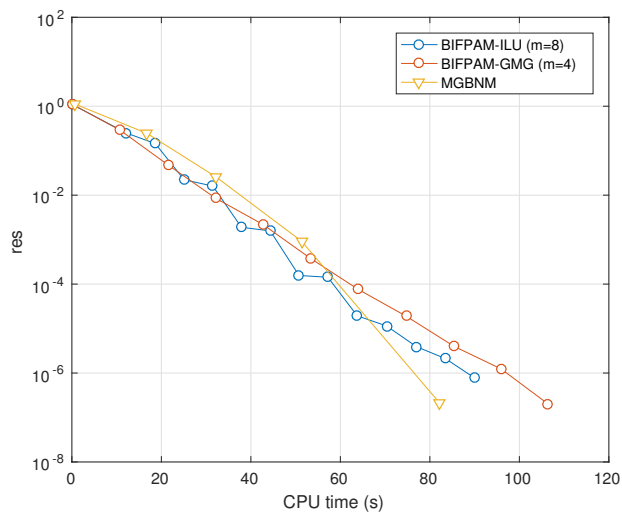


Figure 8: Convergence history for the fourth dominant eigenvalues of the NEACRP problem using the MGBNM and the BIFPAM with the ILU preconditioner.

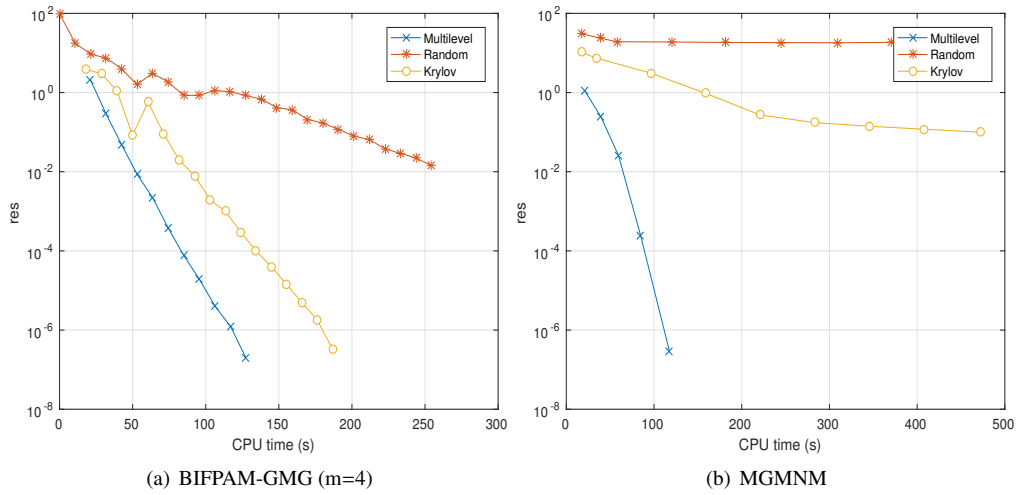


Figure 9: Convergence histories for the BIFPAM and the MGMNM using different initializations for the computation of the  $\lambda$ -modes of the NEACRP problem.

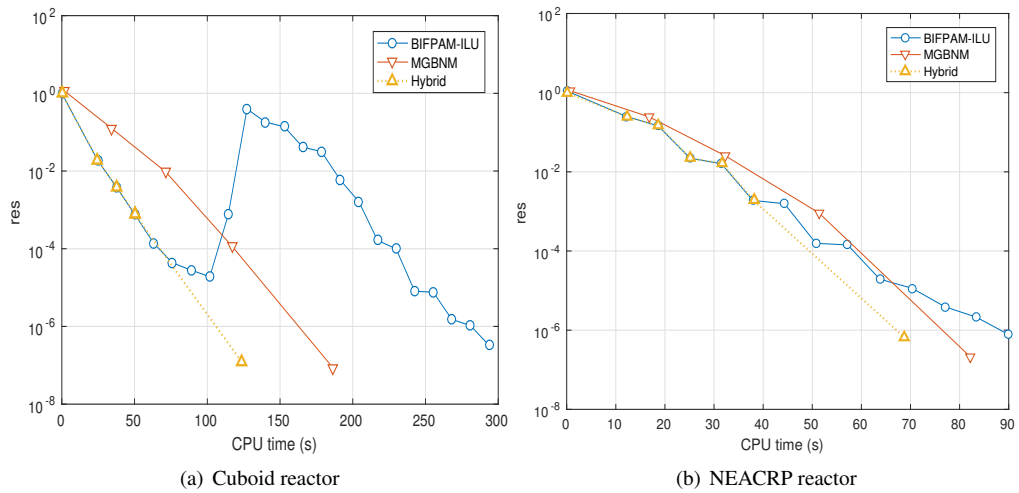


Figure 10: Convergence history of the BIFPAM with ILU preconditioner, the MGBNM and the hybrid method.

## 5. Conclusions

375 In this work, two block methods to solve the  $\lambda$ -modes problem obtained from the discretization of the neutron diffusion equation through a high order finite element method are studied. Both methods are implemented for a generalized eigenvalue problem without transforming it into an ordinary eigenvalue problem to apply them. The first method studied has been the block inverse free preconditioned Arnoldi method (BIFPAM), where the efficiency of using a preconditioner has been studied. In particular, the ILU preconditioner and a geometrical multigrid preconditioner (GMG) have been used. The second method studied has been the modified generalized block Newton method (MGBNM). Moreover, a block multilevel technique has been proposed to initialize the iterative methods for those cases where an initial approximation is not available. The numerical results have shown that both block methods are very efficient to compute several dominant eigenvalues and their corresponding eigenvectors associated with the neutron diffusion equation. The use of a preconditioner (ILU or GMG preconditioner) with the BIFPAM is necessary to be competitive against the MGBNM. According to the initialization proposed, the multilevel scheme is an efficient technique to obtain initial guesses close to the solution. This improves the rate of convergence of the block methods over other kind of initialization strategies. 385 Furthermore, a hybrid scheme is proposed combining these methods that improves the robustness and the computational times with respect to the methods used. Our numerical tests also indicate that the block hybrid method is more efficient than the Krylov-Schur method and it is competitive against the Generalized Davidson method. In further studies, these methods will be used for higher approximations of the neutron transport equation than the neutron diffusion equation and the parallelization of the methods will be undertaken. 390 395

## Acknowledgements

This work has been partially supported by Spanish Ministerio de Economía y Competitividad under projects ENE2017-89029-P, MTM2017-85669-P and BES-2015-072901.

The authors also want to thank the anonymous reviewers for their comments and suggestions that have greatly improved the manuscript. 400

## References

- Baker, C. G., Hetmaniuk, U. L., Lehoucq, R. B., & Thornquist, H. K. (2009). Anasazi software for the numerical solution of large-scale eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)*, 36, 13.
- Balay, S., Abhyankar, S., Adams, M., Brune, P., Buschelman, K., Dalcin, L., Gropp, W., Smith, B., Karpeyev, D., Kaushik, D. et al. (2016). *Petsc users manual revision 3.7*. Technical Report Argonne National Lab.(ANL), Argonne, IL (United States).
- Bangerth, W., Hartmann, R., & Kanschat, G. (2007). deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33, 24/1–24/27.
- Bernal, Á., Hébert, A., Roman, J. E., Miró, R., & Verdú, G. (2017). A Krylov–Schur solution of the eigenvalue problem for the neutron diffusion equation discretized with the Raviart–Thomas method. *Journal of Nuclear Science and Technology*, 54, 1085–1094.
- Capilla, M., Talavera, C., Ginestar, D., & Verdú, G. (2005). A nodal collocation method for the calculation of the lambda modes of the PL equations. *Annals of Nuclear Energy*, 32, 1825–1853.
- Carney, S., Brown, F., Kiedrowski, B., & Martin, W. (2014). Theory and applications of the fission matrix method for continuous-energy monte carlo. *Annals of Nuclear Energy*, 73, 423–431.
- Carreño, A., Vidal-Ferrandiz, A., Ginestar, D., & Verdú, G. (2017a). Multilevel method to compute the lambda modes of the neutron diffusion equation. *Applied Mathematics and Nonlinear Sciences*, 2, 225–236.

- Carreño, A., Vidal-Ferrandiz, A., Ginestar, D., & Verdú, G. (2017b). Spatial modes for the neutron diffusion equation and their computation. *Annals of Nuclear Energy*, *110*, 1010 – 1022.
- 420 Finnemann, H., & Galati, A. (1991). *NEACRP 3-D LWR core transient benchmark, final specification*.
- Gill, D., & Azmy, Y. (2009). A Jacobian-free Newton-Krylov iterative scheme for criticality calculations based on the neutron diffusion equation. In *International Conference on Mathematics, Computational Methods, and Reactor Physics, Saratoga Springs, NY, United States*.
- Gill, D., & Azmy, Y. (2011). Newton's method for solving k-eigenvalue problems in neutron diffusion theory. *Nuclear Science and Engineering*, *167*, 141–153.
- 425 Ginestar, D., Marín, J., & Verdú, G. (2001). Multilevel methods to solve the neutron diffusion equation. *Applied Mathematical Modelling*, *25*, 463–477.
- Golub, G., & Ye, Q. (2002). An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems. *SIAM Journal on Scientific Computing*, *24*, 312–334.
- 430 González-Pintor, S., Ginestar, D., & Verdú, G. (2011). Updating the lambda modes of a nuclear power reactor. *Mathematical and Computer Modelling*, *54*, 1796–1801.
- Hackbusch, W. (2013). *Multi-grid methods and applications* volume 4. Springer Science & Business Media.
- Hamilton, S. P., & Evans, T. M. (2015). Efficient solution of the simplified PN equations. *Journal of Computational Physics*, *284*, 155–170.
- 435 Hébert, A. (1987). Development of the nodal collocation method for solving the neutron diffusion equation. *Annals of Nuclear Energy*, *14*, 527–541.
- Hébert, A. (2008). A Raviart–Thomas–Schneider solution of the diffusion equation in hexagonal geometry. *Annals of Nuclear Energy*, *35*, 363–376.
- Hernandez, V., Roman, J., & Vidal, V. (2005). SLEPC: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Software*, *31*, 351–362.
- 440 Janssen, B., & Kanschä, G. (2011). Adaptive multilevel methods with local smoothing for  $H^1$ - and  $H^{\text{curl}}$ -conforming high order finite element methods. *SIAM Journal on Scientific Computing*, *33*, 2095–2114.
- Knoll, D., Park, H., & Newman, C. (2011). Acceleration of k-eigenvalue/criticality calculations using the Jacobian-free Newton-Krylov method. *Nuclear Science and Engineering*, *167*, 133–140.
- 445 Lewis, E. E., & Miller, W. F. (1984). *Computational methods of neutron transport*. John Wiley and Sons, Inc., New York, NY.
- Lösche, R., Schwetlick, R., & Timmermann, G. (1998). A modified block Newton iteration for approximating an invariant subspace of a symmetric matrix. *Linear Algebra and its Applications*, *275*, 381 – 400.
- Mahadevan, V., & Ragusa, J. (2008). Novel hybrid scheme to compute several dominant eigenmodes for reactor analysis problems. In *International Conference on the Physics of Reactors, Interlaken, Switzerland, September 14–19*.
- 450 March-Leuba, J., & Blaked, E. (1991). A mechanism for out-of-phase power instabilities in boiling water reactors. *Nuclear Science and Engineering*, *107*, 173–179.
- McCormick, S. F. (1987). *Multigrid methods*. SIAM.
- Miró, R., Ginestar, D., Verdú, G., & Hennig, D. (2002). A nodal modal method for the neutron diffusion equation. Application to BWR instabilities analysis. *Annals of Nuclear Energy*, *29*, 1171–1194.
- 455 Neymeyr, K. (2003). Solving mesh eigenproblems with multigrid efficiency. In *Numerical Methods for Scientific Computing. Variational problems and applications. CIMNE, Barcelona*.
- Quillen, P., & Ye, Q. (2010). A block inverse-free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems. *Journal of Computational and Applied Mathematics*, *233*, 1298–1313.
- 460 Saad, Y. (1992). *Numerical methods for large eigenvalue problems*. SIAM.
- Sampath, R. S., & Biros, G. (2010). A parallel geometric multigrid method for finite elements on octree meshes. *SIAM Journal on Scientific Computing*, *32*, 1361–1392.
- Singh, T., Mazumdar, T., & Pandey, P. (2014). NEMSQR: a 3-D multigroup diffusion theory code based on nodal expansion method for square geometry. *Annals of Nuclear Energy*, *64*, 230–243.
- 465 Stacey, W. (2007). *Nuclear reactor physics*. John Wiley & Sons.
- Theiler, G. (2013). Unstructured grids and the multigroup neutron diffusion equation. *Science and Technology of Nuclear Installations*, *2013*.
- Tommasi, J., Maillot, M., & Rimpault, G. (2016). Calculation of higher-order fluxes in symmetric cores-I: Theory. *Nuclear Science and Engineering*, *184*, 174–189.
- 470 Verdú, G., Ginestar, D., Miró, R., & Vidal, V. (2005). Using the Jacobi–Davidson method to obtain the dominant Lambda modes of a nuclear power reactor. *Annals of nuclear energy*, *32*, 1274–1296.
- Verdú, G., Ginestar, D., Vidal, V., & Muñoz-Cobo, J. (1994). 3D  $\lambda$ -modes of the neutron-diffusion equation. *Annals of Nuclear Energy*, *21*, 405–421.
- Verdú, G., Miró, R., Ginestar, D., & Vidal, V. (1999). The implicit restarted Arnoldi method, an efficient alternative to solve the neutron diffusion equation. *Annals of nuclear energy*, *26*, 579–593.
- 475 Vidal, V., Verdú, G., Ginestar, D., & Muñoz-Cobo, J. (1998). Variational acceleration for subspace iteration method.

- Application to nuclear power reactors. *International Journal for Numerical Methods in Engineering*, 41, 391–407.
- Vidal-Ferrandiz, A., Fayez, R., Ginestar, D., & Verdú, G. (2014). Solution of the lambda modes problem of a nuclear power reactor using an  $h$ - $p$  finite element method. *Annals of Nuclear Energy*, 72, 338 – 349.
- 480 Warsa, J., Wareing, T., Morel, J., McGhee, J., & Lehoucq, R. (2004). Krylov subspace iterations for deterministic  $k$ -eigenvalue calculations. *Nuclear Science and Engineering*, 147, 26–42.