Vidal-Ferràndiz, A.; Carreño, A.; Ginestar Peiro, D.; Verdú Martín, GJ. (2020). A block Arnoldi method for the SPN equations. International Journal of Computer Mathematics. 97(1-2):341-357. https://doi.org/10.1080/00207160.2019.1602768

A Block Arnoldi Method for the SP_N Equations

A. Vidal-Ferràndiz[a], A. Carreño[a], D. Ginestar[b] and G. Verdú[a]

[a]Instituto Universitario de Seguridad Industrial: Radiofísica y Medioambiental, Universitat Politècnica de València, València, Spain;
[b]Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, València, Spain.

ARTICLE HISTORY
Compiled April 6, 2019

ABSTRACT
The simplified spherical harmonics equations are a useful approximation to the stationary neutron transport equation. The eigenvalue problem associated with them is a challenging problem from the computational point of view. In this work, we take advantage of the block structure of the involved matrices to propose the block inverse-free preconditioned Arnoldi method as an efficient method to solve this eigenvalue problem. For the spatial discretization, a continuous Galerkin finite element method implemented with a matrix-free technique is used to keep reasonable memory demands. A multilevel initialization using linear shape functions in the finite element method is proposed to improve the method convergence. This initialization only takes a small percentage of the total computational time. The proposed eigenvalue solver is compared to the standard power iteration method, the Krylov-Schur method and the generalized Davidson method. The numerical results show that it reduces the computational time to solve the eigenvalue problem.

KEYWORDS
Generalized eigenvalue problem; Neutron transport, Multilevel method; Block inverse-free preconditioned Arnoldi method; Generalized Davidson Method.

1. Introduction

The stationary distribution of neutrons inside a nuclear power reactor is described by the neutron transport equation [21]. This equation depends on the position of neutrons, their energy and their direction vector. This makes the neutron transport equation very challenging to solve for general geometries. One of the most common approximations to solve this equation is the neutron diffusion equation and its higher order approximations to the neutron transport as the simplified spherical harmonics equations (SP_N).

The stationary neutron transport equation and its approximations can be expressed as a generalized eigenvalue problem, of the form,

$$\mathcal{L}\phi = \frac{1}{\lambda}\mathcal{M}\phi, \tag{1}$$

where $\mathcal{L}$ is the neutron loss operator, $\mathcal{M}$ is the neutron production operator, λ is

CONTACT A. Vidal-Ferràndiz. Email: anvifer2@upv.es

the eigenvalue of the problem and ϕ is its corresponding eigenvector related with the neutron flux. The dominant eigenvalue and its corresponding eigenfunction describe the steady state neutron distribution inside the reactor core. Next eigenvalues and their corresponding eigenfunctions are also important to study time dependent problems and reactor instabilities.

For the spatial discretization of the problem a continuous Galerkin finite element method is considered. The finite element method is implemented making use of a matrix-free strategy for the non-diagonal blocks of the matrices [15]. In this way, these block matrices are not stored in the memory and matrix-vector products are computed on the fly by using a cell-based interface. This strategy reduces enormously the memory used by the matrix elements and can reduce matrix-vector multiplication times in some computer architectures [25]. The memory cost of storing the matrices is, in general, very large and it depends on the properties of the problem, in other words, it depends on the finite element order, the number of cells, the number of energy groups and the number of moment equations taken into account. However, the advantage of storing the matrices is that the most efficient preconditioners can be applied. In this way, some balance between memory usage and algorithm efficiency has to be found.

Different methods such as the power iteration method, the Implicit Restarted Arnoldi Method (IRAM) or more recently the Krylov-Schur method have been used to solve this kind of eigenvalue problems [16, 23, 24]. The application of these methods require to transform the generalized problem into an ordinary eigenvalue problem, applying a shift and invert technique. In the solution process, it is necessary to solve numerous linear systems. These systems are very large and thus, these methods require many computational resources.

In this work, an inverse-free Krylov subspace method introduced in [10] is used with a block implementation proposed by Quillen and Ye in [18], called block inverse-free preconditioned Arnoldi method (BIFPAM). Other block methods based on the Arnoldi method can be considered such as the Augmented Block Householder Arnoldi Method (AHBEIGS) [2]. The BIFPAM method improves the traditional steepest descent method by expanding the search direction to a Krylov subspace with the advantage of the better approximation properties offered by Krylov subspaces. The BIFPAM does not need to solve any linear system and it can be preconditioned to improve its convergence properties. Another eigenvalue problem solver with these properties is the generalized Davidson method (GD) [17] used for the neutron transport equation in [11, 12].

Usually, different types of preconditioners are used to accelerate these iterative schemes [1, 14, 27]. In our case, a block Gauss-Seidel preconditioner is implemented.

To ensure the convergence of the BIFPAM solver a good initial approximation is required. To reach the desired accuracy a high order finite element method should be used with typically polynomial basis functions of degree 2 or 3. The proposed initial approximation for the desired eigenvalues and their corresponding eigenfunctions is obtained from the solution of the problem with linear finite elements using a multilevel technique. Furthermore, this multilevel technique can accelerate other eigenvalue solvers such as the power iteration method and the generalized Davidson method, as it will be shown through this work.

The structure of the rest of the paper is as follows. In Section 2, the $\mathrm{SP_N}$ approximation to the neutron transport equation is presented. In Section 3, the block inverse-free preconditioned Arnoldi method and the way how it is preconditioned is exposed. Numerical results for the analysis of the block method are given in Section 4.1. Finally, the conclusions of the paper are summarized in Section 5.

2. Simplified $\mathbf{P_N}$ equations

We consider the eigenvalue problem associated with the multi-group, steady-state, neutron transport equation in one-dimensional slab geometry [21],

$$\left(\mu\frac{\mathrm{d}}{\mathrm{d}x}+\Sigma_t^g(x)\right)\psi^g(x,\mu)=\sum_{g'=1}^{G}\int_{-1}^{1}\Sigma_s^{gg'}(x,\mu_0)\psi^{g'}(x,\mu')\mathrm{d}\mu'$$
$$+\frac{1}{\lambda}\sum_{g'=1}^{G}\frac{\chi^g(x)}{2}\nu\Sigma_f^{g'}(x)\int_{-1}^{1}\psi^{g'}(x,\mu')\mathrm{d}\mu', \qquad (2)$$
$$g=1,\ldots,G, \qquad x\in[0,\,L_t]$$

where G is the number of energy groups considered, μ is the cosine director of the incident neutrons and μ_0 is the change of cosine director due to scattering collisions. $\Sigma_t^g(x)$, $\nu\Sigma_f^g(x)$ and $\Sigma_s^{gg'}(x,\mu_0)$ are the total, production and scattering cross sections for energy group g, respectively, and $\chi^g(x)$ is the fission spectrum. The dominant eigenvalue (the largest in magnitude) of the problem (2), k_{eff}, is the multiplicative factor of the system and measures its criticality.

The spherical harmonics approximation to the neutron transport equation in one-dimensional geometries assumes that the angular dependence of both the neutron flux distribution and the scattering cross-section can be expanded in terms of $N+1$ Legendre polynomials, $P_n(\mu)$, in the following form

$$\psi^g(x,\mu)=\sum_{n=0}^{N}\frac{2n+1}{2}\phi_n^g(x)\,P_n(\mu)\,, \qquad (3)$$

$$\Sigma_s^{gg'}(x,\mu_0)=\sum_{n=0}^{N}\frac{2n+1}{2}\Sigma_{sn}^{gg'}(x)\,P_n(\mu_0)\,. \qquad (4)$$

Inserting equations (3) and (4) in equation (2), the $\mathrm{P_N}$ equations can be expressed in matrix notation [12] as

$$\frac{\mathrm{d}\,\Phi_1}{\mathrm{d}x}+\Sigma_0\Phi_0=\frac{1}{\lambda}F\Phi_0\,, \qquad (5)$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{n}{2n+1}\Phi_{n-1}+\frac{n+1}{2n+1}\Phi_{n+1}\right)+\Sigma_n\Phi_n=0\,, \qquad \text{for } n=1,\,\ldots,\,N\,, \qquad (6)$$

where,

$$\Sigma_n = \begin{pmatrix} \Sigma_t^1 - \Sigma_{sn}^{11} & -\Sigma_{sn}^{12} & \cdots & -\Sigma_{sn}^{1G} \\ \vdots & \vdots & \ddots & \vdots \\ -\Sigma_{sn}^{G1} & -\Sigma_{sn}^{G2} & \cdots & \Sigma_t^G - \Sigma_{sn}^{GG} \end{pmatrix},$$

$$F = \begin{pmatrix} \chi^1\nu\Sigma_f^1 & \chi^1\nu\Sigma_f^2 & \cdots & \chi^1\nu\Sigma_f^G \\ \vdots & \vdots & \ddots & \vdots \\ \chi^G\nu\Sigma_f^1 & \chi^G\nu\Sigma_f^2 & \cdots & \chi^G\nu\Sigma_f^G \end{pmatrix}, \quad \Phi_n = \begin{pmatrix} \phi_n^1 \\ \vdots \\ \phi_n^G \end{pmatrix}.$$

The $\mathrm{P_N}$ equations (5) and (6) constitute a set of $N+1$ equations with $N+2$ unknowns. This problem is usually solved setting the derivative of the highest order moment to zero $\frac{\mathrm{d}}{\mathrm{d}x}\Phi_{N+1} = 0$. It must be noted that in many nuclear applications, as usual static reactor calculations solved with the $\mathrm{SP_3}$ equations the scattering moments above 1 are not required to attain sufficient accuracy [12]. Thus, Σ_n is a diagonal matrix for $n > 1$. In some applications the scattering cross section is assumed to be isotropic as in the case of C5G7 benchmark [20]. Moreover, it exists relatively little coupling from fast to thermal groups that yields to Σ_0 matrix to be predominately lower triangular.

As the multidimensional $\mathrm{P_N}$ approximation is a complicated set of equations, Gelbard [9] showed that one dimensional $\mathrm{P_N}$ equations could be extended to multidimensional geometries substituting the one-dimensional derivatives by a multidimensional gradient in equations (5) and (6). This ad-hoc approximation, called Simplified $\mathrm{P_N}$ ($\mathrm{SP_N}$), gives accurate results compared to the computational cost necessary to solve the problem. Theoretical basis for the $\mathrm{SP_N}$ equations were provided in [5], showing that these equations are high-order asymptotic solutions of the transport equation when diffusion theory is the leading-order approximation in the dominant direction. $\mathrm{SP_N}$ equations are much simpler than the multidimensional $\mathrm{P_N}$ equations [6] and can be easily implemented using numerical methods suited for diffusion-like equations.

By using a linear change of variables, equations (5) and (6) can be expressed as a system of $G \times (N+1)/2$ second order elliptic, diffusive-like, equations for the even moments [12]. For example, the set of $\mathrm{SP_3}$ equations is expressed as

$$-\vec{\nabla}\left(D\vec{\nabla}U\right) + SU = \frac{1}{\lambda}MU\,, \tag{7}$$

where the effective diffusion matrix, D, the absorption matrix, S, and the fission matrix, M, are given by

$$D = \begin{pmatrix} \frac{1}{3}{\Sigma_1}^{-1} & 0 \\ 0 & \frac{1}{7}{\Sigma_3}^{-1} \end{pmatrix}, \qquad S_{ij} = \sum_{n=0}^{1} c_{ij}^{(n)}\Sigma_n\,, \qquad M_{ij} = c_{ij}^{(0)}F\,,$$

and the following linear change of variables has been applied,

$$U = \begin{pmatrix} u_0 \\ u_2 \end{pmatrix} = \begin{pmatrix} \Phi_0 + 2\Phi_2 \\ 3\Phi_0 + 4\Phi_2 \end{pmatrix}. \tag{8}$$

Finally, the coefficients matrices, $c^{(0)}$ and $c^{(1)}$ are defined as

$$c^{(0)} = \begin{pmatrix} 1 & -\frac{2}{3} \\ -\frac{2}{3} & \frac{4}{9} \end{pmatrix}, \qquad c^{(1)} = \begin{pmatrix} 0 & 0 \\ 0 & \frac{5}{9} \end{pmatrix}. \tag{9}$$

2.1. Finite element method and matrix-free strategy

For the spatial discretization of the problem (7), a high-order continuous Galerkin Finite Element Method (FEM) is used, implemented with the help of `deal.II` library [3]. The discretization transforms the problem (1) into an algebraic generalized eigenvalue problem of the form

$$Ax = \lambda Bx, \tag{10}$$

where A takes into account the production operator $\mathcal{M}$ and B is the discretization of the neutron loss operator $\mathcal{L}$. More details about the finite element spatial discretization are explained in [24] and in [25]. In general, these matrices are not symmetric.

Nevertheless, as we are solving an energy multigroup problem for the $\mathrm{SP_N}$ equations we can take advantage of the block structure of the matrices A and B, where each block is symmetric. Most of the blocks in the lower part of matrix A and far from the diagonal in B are zero. For example, the matrices of the C5G7 benchmark in $\mathrm{SP_1}$ case studied in Section 4.1 have the following structure,

$$\begin{pmatrix} A_{11} & \cdots & A_{17} \\ A_{21} & \cdots & A_{27} \\ A_{31} & \cdots & A_{37} \\ A_{41} & \cdots & A_{47} \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \lambda \begin{pmatrix} B_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ B_{21} & B_{22} & 0 & 0 & 0 & 0 & 0 \\ B_{31} & B_{32} & B_{33} & 0 & 0 & 0 & 0 \\ B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & 0 & 0 \\ 0 & 0 & 0 & B_{54} & B_{55} & B_{56} & 0 \\ 0 & 0 & 0 & 0 & B_{65} & B_{66} & B_{67} \\ 0 & 0 & 0 & 0 & 0 & B_{76} & B_{77} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}. \tag{11}$$

It must be noted that this structure is not the same for all reactors and it depends on the material composition of the nuclear core. The matrices A and B have $G \times (N+1)/2$ blocks each row and column. The dimension of each block depends on the number of cells of the spatial discretization, the polynomial degree of the finite element method and the type of boundary conditions applied. For instance, for the C5G7 benchmark solved with a mesh of 28900 cells, polynomials of degree 2 and vacuum boundary conditions; the block size is 116009.

A matrix-free strategy for the blocks of the matrix A and for the non-diagonal blocks of B is developed. The diagonal block matrices of B are stored in sparse format to permit the computation of an incomplete LU factorization of these blocks. In this way, matrix-vector products are computed on the fly in a cell-based interface. For instance, we can consider that a finite element Galerkin approximation that leads to the block matrix $A_{b,b}$ takes a vector u as input and computes the integrals of the operator multiplied by trial functions, and the output vector is v. The operation can be expressed as a sum of K cell-based operations,

$$v = A_{b,b}u = \sum_{k=1}^{K} P_k^T A_{b,b}^k P_k u, \tag{12}$$

where P_k denotes the matrix that defines the location of cell-related degrees of freedom in the global vector and $A^k_{b,b}$ denotes the submatrix of $A_{b,b}$ on cell k. This sum is optimized through *sum-factorization*. Details about the implementation are explained in [15]. This strategy minimizes the memory used by the matrix elements and it can improve matrix-vector multiplication computational times [25].

3. Block inverse-free preconditioned Arnoldi method

The block inverse-free preconditioned Arnoldi method (BIFPAM) was originally presented and analyzed for A and B symmetric matrices and B definite positive [10, 18]. The elements of the matrices related to these problems depend on the reactor composition and it is not possible to prove theoretically the character of the matrices or apply some characterization properties in a general case described in [4]. Although the matrices A and B are not symmetric, we assume that the associated eigenvalues and their corresponding eigenvectors are real and positive. This statement has only been proved under restrictive conditions (e.g. monoenergetic transport). However, if not theoretically, this assumption is supported by numerical evidence on benchmark problems and it is studied for more realistic problems in [7]. We will show that this method also works efficiently for the λ-modes problem associated with the $\mathrm{SP_N}$ equations for the C5G7 problem.

We start with problem (10) for one eigenvalue

$$Ax = \lambda Bx,$$

the goal is maximizing the Rayleigh quotient

$$\lambda(x) := \frac{x^{\mathrm{T}} Ax}{x^{\mathrm{T}} Bx}, \tag{13}$$

where x is on a certain subspace and it is assumed that $x^T Bx$ never vanishes.

By following the steepest descent method and starting from an initial approximation (λ_0, x_0), the approximate eigenvector x_{k+1} in the k-th iteration can be computed as

$$x_{k+1} = x_k + \alpha r_k,$$

where r_k is the residual error $r_k = (A - \lambda_k B)x_k$, α is computed for maximizing the Rayleigh-Ritz quotient (13) and λ_k is the approximation of the eigenvalue λ in the k-th iteration, computed as

$$\lambda_k = \frac{x_k^{\mathrm{T}} Ax_k}{x_k^{\mathrm{T}} Bx_k}.$$

This can also be considered as the Rayleigh-Ritz projection method. For that, we obtain a basis Z of the Krylov subspace

$$K_1(A - \lambda_k B,\ x_k) := \mathrm{span}\{x_k,\ (A - \lambda_k B)x_k\},$$

and construct the projected eigenvalue problem

$$Z^{\mathrm{T}}AZU = Z^{\mathrm{T}}BZU\Lambda, \tag{14}$$

which is a 2×2 generalized eigenvalue problem. Then, the largest eigenvalue $\Lambda_{1,1}$ and 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 approach can be extended by computing a basis Z of the m-order Krylov subspace

$$K_m(A - \lambda_k B,\ x_k) := \mathrm{span}\{x_k,\ (A - \lambda_k B)x_k,\ (A - \lambda_k B)^2 x_k,\ \ldots,\ (A - \lambda_k B)^m x_k\}.$$

Arnoldi method is used to construct the basis of K_m.

Furthermore, a block version of this method can be constructed that allows to compute several eigenvalues simultaneously. Thus, if we are interested on computing q eigenvalues of problem (10), we can accelerate the convergence by using the subspace $\mathcal{K}_m$ with

$$\mathcal{K}_m := \bigcup_{i=1}^{q} K_m^i(A - \lambda_{k,i}B,\ x_{k,i}), \tag{15}$$

where $\lambda_{k,i}$ denotes the i-th eigenvalue computed in the k-th iteration, $x_{k,i}$ its associated eigenvector and K_m^i is the Krylov subspace associated with eigenvector $x_{k,i}$.

One may construct a basis for this subspace by first constructing q bases, say Z_i for $1 \leq i \leq q$ of the q Krylov subspaces $K_m^i(A - \lambda_{k,i}B,\ x_{k,i})$, $1 \leq i \leq q$. Then, following with a similar strategy as the one followed to compute only one eigenvalue, the original generalized eigenvalue problem (10) is projected onto the union of the q bases and the corresponding projected problem is solved for the first q dominant eigenvalues to obtain the new eigenvalues and their corresponding eigenvectors.

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

3.1. Preconditioning

Golub in [10] proved that the rate of convergence of the block inverse-free Arnoldi method depends on the spectral distribution of $C = A - \lambda B$, with λ the desired eigenvalue. Then, the idea of preconditioning is to construct an equivalent problem so that when we apply the block inverse-free Arnoldi method to the new problem, the new matrix associated to the equivalent problem, $\hat{C}$, has a better spectral distribution.

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

$$(L_{i,k}^{-1}AU_{i,k}^{-1})\hat{x} = \lambda(L_{i,k}^{-1}BU_{i,k}^{-1})\hat{x} \Leftrightarrow \hat{A}_{i,k}\hat{x} = \lambda\hat{B}_{i,k}\hat{x}, \tag{16}$$

which has the same eigenvalues as the original problem. Note that the relation between the eigenvector of eigenvalue problem (10), x_k, and the corresponding approximate eigenvector for the transformed problem (16), $\hat{x}_{i,k}$, is $\hat{x}_{i,k} = U_{i,k}x_{i,k}$. This transformation is called preconditioning. Thus, when the block inverse-free Arnoldi method to the

problem (16) is applied, the rate of convergence will be determined by the eigenvalues of $\hat{C}_{i,k} := \hat{A}_{i,k} - \lambda_{i,k}\hat{B}_{i,k}$. Different preconditioning transformations can be constructed by using different factorizations of the matrix $C_{i,k}$. The main goal must be to choose suitably $L_{i,k}$ and $U_{i,k}$ to obtain a favourable distribution of the eigenvalues of matrix $\hat{C}_{i,k}$.

This preconditioned iteration of the block inverse-free preconditioned Arnoldi method (BIFPAM) can be implemented without explicitly forming the transformed problem $\hat{C}_{i,k}$. So, to obtain a basis of K_m^i, we need to construct a basis $\hat{Z}_{i,k}$ for the subspace

$$\hat{K}_m^i := \text{span}\{\hat{x}_{i,k}, \hat{C}_{i,k}\hat{x}_{i,k}, \hat{C}_{i,k}^2\hat{x}_{i,k}, \ldots, \hat{C}_{i,k}^m\hat{x}_{i,k}\},$$

and then compute $U_{i,k}^{-1}\hat{Z}_{i,k}$; or equivalently, we can directly obtain a basis of K_m^i which is obtained premultiplying the vectors of $\hat{K}_m^i$ by $U_{i,k}^{-1}$ as,

$$U_{i,k}^{-1}\hat{K}_m^i := \text{span}\{x_{i,k}, (L_{i,k}U_{i,k})^{-1}C_{i,k}x_{i,k}, \ldots, \left((L_{i,k}U_{i,k})^{-1}C_{i,k}\right)^m x_{i,k}\}. \tag{17}$$

The BIFPAM is summarized in Algorithm 1.

Algorithm 1 Block inverse-free preconditioned Arnoldi method (BIFPAM).

Input: Matrices A and B, initial approximation $X_0 = [x_{0,1}, \ldots, x_{0,p}]$.
Output: Diagonal matrix of eigenvalues Λ and matrix V with the eigenvectors as its columns.

1: Compute $\lambda_{0,i} = (x_{0,i}^T A x_{0,i})/(x_{0,i}^T B x_{0,i})$, $1 \le i \le p$
2: **for** $k = 1$ to $maxits$ **do**
3: Obtain the basis Z_i of $K_m^i(A - \lambda_{k-1,i}B,\ x_{k-1,i})$, $1 \le i \le q$ using the Eq. (17)
4: Construct $Z := [Z_1,\ \ldots,\ Z_q]$
5: Form projection $A_m = Z^T A Z$, $B_m = Z^T B Z$
6: Compute q dominant eigenpairs $(\lambda_{k,i}, u_i)$ of $A_m U = B_m U \Lambda$
7: Compute $x_{k,i} = Z u_i$, $1 \le i \le q$
8: **end for**

It is observed that only $(L_{i,k}U_{i,k})^{-1}$ is needed to premultiply the vectors when the matrix $C_{i,k} = A - \lambda_{i,k}B$ is applied. In order words, we only need a method to multiply a preconditioner of the matrix $C_{i,k}$ by a vector and we can use different preconditioners that do not necessarily come from the factorization of a matrix [8]. Although in the papers [8, 10], preconditioners of the matrix C_k are used, in this work we have numerically observed for our kind of algebraic problems (that come from the discretization of some approximation of the neutron transport equation) that using only preconditioner of matrix B works efficiently and it is not necessary to construct a preconditioner for the whole matrix C_k. This permits using a block preconditioner (by using the block structure of matrix B), without assembling any additional matrix, with the advantage that the blocks of this matrix are symmetric and positive definite. This causes an improvement in the implementation of the Algorithm. In particular, we choose the block Gauss-Seidel preconditioner [19] as shown in Algorithm 2.

Algorithm 2 Block Gauss-Seidel preconditioner.

Input: Matrix B and vector $x = [x_1; \ldots; x_{Bl}]$.
Output: Vector $y = [y_1; \ldots; y_{Bl}]$, result of applying the preconditioner of B to x.

```
for b = 1 to Bl do
    if It was not obtained before then
        Obtain the ILU factorization of submatrix B_bb, L, U.
    end if
    sum = x_b
    for c = 1 to b do
        Compute sum = sum − B_{b,c} y_c
    end for
    Compute y_b = (LU)^{-1} sum
end for
```

3.2. Power Iteration Method

The eigenvalue problem (10) is traditionally solved using the power iteration method (PI) with a combination of inner and outer iterations [16, 26]. The inner iterations solve the following set of of linear systems

$$B_{bb}\, x_b^{(i)} = Q_b^{(i-1)}, \qquad b = 1,\ \ldots,\ Bl = G \times (N+1)/2, \tag{18}$$

where $Q_b^{(i-1)}$ is generated by using the updated solution for the up-diagonal terms and the previous solution for down-diagonal terms. In order words, a Gauss Seidel iteration is embedded in the construction of $Q_b^{(i-1)}$.

$$Q_b^{(i-1)} = \sum_{c=1}^{b-1} B_{bc} x_c^{(i)} + \sum_{c=b+1}^{Bl} B_{bc} x_c^{(i-1)} + \frac{1}{\lambda^{(i-1)}} \sum_{c=1}^{Bl} A_{bc} x_c^{(i-1)}. \tag{19}$$

The outer iteration reads as

$$\lambda^{(i)} = \lambda^{(i-1)} \frac{x^{(i)\,T} B x^{(i)}}{x^{(i)\,T} B\, x^{(i-1)}}. \tag{20}$$

Power iteration will converge to the eigenvalue of largest magnitude, $\lambda_1 = k_{\text{eff}}$. If more than one eigenvalue is requested a deflation technique should be used. In other words, it can be computed one eigenvector at a time while decontaminating the subspace of the computed eigenvector. However, the deflation technique has a very slow convergence. The convergence rate for the fundamental eigenvalue is determined by the dominance ratio $\delta = |\lambda_2|/|\lambda_1|$, where λ_2 is the next largest eigenvalue in magnitude [26]. Convergence of the power iteration method slows as $\delta \to 1.0$.

Another method used to compare the performance of the BIPFAM is the Krylov-Schur method which was studied in [22]. The generalized Davidson method (GD) [17] is also used to this end. Both methods are implemented using the library SLEPc [13]. To make a fair comparative both methods were preconditioned with the same block Gauss-Seidel preconditioner developed in Algorithm 2.

4. Numerical Results

4.1. 2D C5G7

We start analyzing the performance of the proposed BIFPAM method, by studying the two dimensional version of the C5G7 fuel assembly benchmark introduced by the Nuclear Energy Agency (NEA) [20]. The benchmark has been analyzed with several deterministic codes and a very precise solution was obtained as reference using the Monte Carlo method.

The configuration of the benchmark consists of a nuclear reactor core with MOX and UO_2 square fuel assemblies surrounded by a moderator region, as it is shown in Figure 1. Each fuel assembly consists of 17×17 square fuel pin cells of size 1.26 cm. Each pin cell is made of a circular fuel region of radius 0.54 cm surrounded by moderator. This geometry is also shown in Figure 1. A very detailed explanation of the benchmark with all the remaining specifications of the benchmark is given in [20].

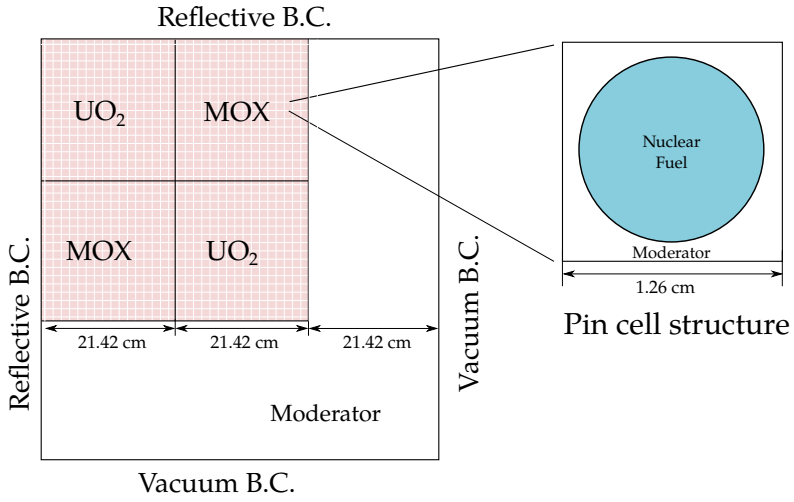
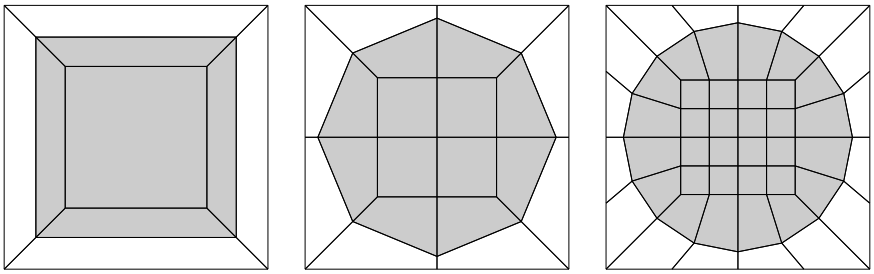


Figure 1.: The core configuration for the 2D C5G7 benchmark problem.

Figure 2 shows the proposed meshes used to discretize the circular region depending on the radial refinement parameter r_r. All meshes maintain the area of the fuel region (in grey color) to provide a more accurate model of the problem. Tables 1 and 2 show the results for the fundamental eigenvalue and its corresponding eigenvector associated with diffusion theory (SP_1) and the SP_3 equations for different meshes depending on the finite element polynomial degree p and the mesh refinement parameter, r_r. Errors for the eigenvalue are given in pcm by $\Delta\lambda_1 = 10^5|\lambda_1 - \lambda_{\text{ref}}|$ where $\lambda_{\text{ref}} = 1.186550$ is the reference eigenvalue given by the benchmark authors. To assess the eigenvector, the following collective per cent error measures were selected: average pin power per cent error (AVG) and mean relative pin power per cent error (MRE) of the pin power



(a) $r_r = 0$ (b) $r_r = 1$ (c) $r_r = 2$

Figure 2.: Meshes considered for the pin structure.

per cent error distribution [20],

$$\text{AVG} = \frac{1}{N_c} \sum_{i=1}^{N_c} |e_i|, \tag{21}$$

$$\text{MRE} = \frac{\sum_{i=1}^{N_c} |e_i| \, p_i}{\sum_{i=1}^{N_c} p_i}, \tag{22}$$

where N_c is the number of fuel pin cells and e_i is the calculated per cent error for the i-th pin neutron power, p_i. It can be seen that the results are spatially converged for $r_r = 1$ and $p = 2$. Furthermore, it is observed that the SP_3 equations improve the accuracy of the results with respect to the SP_1 mainly for the eigenvector. Performance results for the different proposed eigensolvers will be studied for these optimal discretization parameters.

Table 3 displays the residual error and the performance of the BIFPAM solver using $r_r = 1$ and $p = 2$ for different initialization strategies. The first initialization choice, that we denote as Random, is to generate the initial vector using random numbers on the interval $[-1,\ 1]$. The second one is to get the initial guess from the Krylov subspace generated by the matrix $L^{-1}M$ acting on a vector of ones. The Arnoldi method has been used to obtain this subspace. We called to this strategy Krylov initialization. After using both strategies, 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 [8]. Third, we propose a multilevel initialization by previously solving the problem with linear finite elements $p = 1$ and then this solution is interpolated to the problem with the required polynomial degree p. Moreover, for the SP_3 case we can also solve the initialization problem, the problem with $p = 1$, by using the SP_1 approximation. In this last case, the initial guess for u_0 of equation (8) comes from the interpolation of the SP_1 solution and u_2 is set to a zero vector. The SP_1 and SP_3 with $p = 1$ initialization problems are solved with the GD method until a tolerance of 1×10^{-4}.

The results conclude that the proposed multilevel initialization (SP_1, $p = 1$) minimizes the initialization error taking only 6 seconds. It must be noted that the multilevel SP_3 initialization does not improve the initialization error of the multilevel SP_1 initialization to solve the SP_3 problem. Then, it does not worth the computational overhead. Furthermore, the Krylov initialization takes more CPU time than the SP_1, $p = 1$ initialization and it gets one order less accuracy. Random initialization does not spend

any CPU time to obtain the initial guess but this initialization has a large error.

Tables 4 and 5 show the performance of the different eigenvalue solvers for SP_1 and SP_3 problems with $r_r = 1$ and $p = 2$. The initial guess is obtained from the multilevel SP_1 initialization. It must be noted that the SP_1 problem uses about 1200 Mb of RAM memory and the SP_3 case uses about 2500 Mb. The numerical results show that the proposed BIFPAM method with initialization is more efficient than the other methods studied for one eigenvalue calculations, even though this problem is advantageous to the power iteration method as $\delta = 0.763$.

These tables also show that the proposed multilevel initialization is also convenient for the power iteration method and the generalized Davidson method. The 6 seconds of initialization are included in the CPU times in Tables 4 and 5.

Figure 3 shows the residual norm, $\hat{r}_k$, at the k-th iteration of the eigenvalue solver,

$$\hat{r}_k = \frac{\|Ax_k - \lambda_1 Bx_k\|}{|\lambda_1|} \tag{23}$$

for the first eigenvalue in the SP_1 problem with $p = 2$ and $r_r = 1$. Figure 4 also shows the convergence history of $\hat{r}_k$ for the SP_3 problem with $p = 2$ and $r_r = 1$. In these Figures it can be noted that the proposed initialization reduces the starting residual norm but it does not change the convergence ratio of the methods.

Table 1.: Accuracy results for SP_1.

r	**p**	**Number of Cells**	**Number of DoFs**	**Eigenvalue** λ_1	$\Delta\lambda_1$	**AVG** (%)	**MRE** (%)
0	1	11 849	83 664	1.185 11	144	2.26	1.88
0	2	11 849	333 207	1.185 12	143	2.65	2.02
0	3	11 849	748 636	1.185 11	144	2.26	1.88
1	1	28 900	203 735	1.183 81	274	1.52	1.25
1	2	28 900	812 063	1.183 35	320	1.43	1.21
1	3	28 900	1 824 991	1.183 30	325	1.42	1.21
2	1	78 608	553 119	1.183 73	282	1.46	1.24
2	2	78 608	2 206 743	1.183 26	329	1.42	1.21
2	3	78 608	4 960 879	1.183 25	330	1.42	1.21

Table 2.: Accuracy results for SP_3.

r	**p**	**Number of Cells**	**Number of DoFs**	**Eigenvalue** λ_1	$\Delta\lambda_1$	**AVG** (%)	**MRE** (%)
0	1	11 849	167 328	1.185 40	115	1.69	1.46
0	2	11 849	666 414	1.183 75	280	0.85	0.74
1	1	28 900	407 470	1.183 57	298	0.86	0.72
1	2	28 900	1 624 126	1.182 61	394	0.72	0.65
2	1	78 608	1 106 238	1.183 47	308	0.81	0.72

Table 3.: Performance results of different initialization procedures to compute 1 eigenvalue with BIFPAM solver by using $r_r = 1$ and $p = 2$.

Eq.	**Init.**	**Init. Error**	**Init. Time** (s)	**Total Time** (s)
SP_1	Random	5.04	0	55
	Krylov	2.0×10^{-2}	7	51
	SP_1, $p = 1$	2.1×10^{-3}	6	25
SP_3	Random	5.06	0	102
	Krylov	2.3×10^{-2}	24	104
	SP_1, $p = 1$	2.5×10^{-3}	6	65
	SP_3, $p = 1$	2.6×10^{-3}	11	71

Table 4.: Performance results for different eigenvalue solvers for SP_1 problem with $r_r = 1$ and $p = 2$.

Solver Method	**Init**	**n eigs.**	**CPU Time** (s)	**n its**
Power Iteration	No	1	157	44
Power Iteration	Yes	1	81	20
BIFPAM	No	1	55	5
BIFPAM	Yes	1	25	2
Generalized Davidson	No	1	68	14
Generalized Davidson	Yes	1	46	8
Krylov-Schur	No	1	1646	16
BIFPAM	Yes	4	180	4
Generalized Davidson	Yes	4	220	42
Krylov-Schur	No	4	1955	19

Table 5.: Performance results for different eigenvalue solvers for SP_3 problem with $r_r = 1$ and $p = 2$.

Solver Method	**Init**	**n eigs.**	**CPU Time** (s)	**n its**
Power Iteration	No	1	274	52
Power Iteration	Yes	1	160	26
BIFPAM	No	1	102	5
BIFPAM	Yes	1	65	3
Generalized Davidson	No	1	150	17
Generalized Davidson	Yes	1	101	12
Krylov-Schur	No	1	2729	16
BIFPAM	Yes	4	329	4
Generalized Davidson	Yes	4	403	49
Krylov-Schur	No	4	4749	28

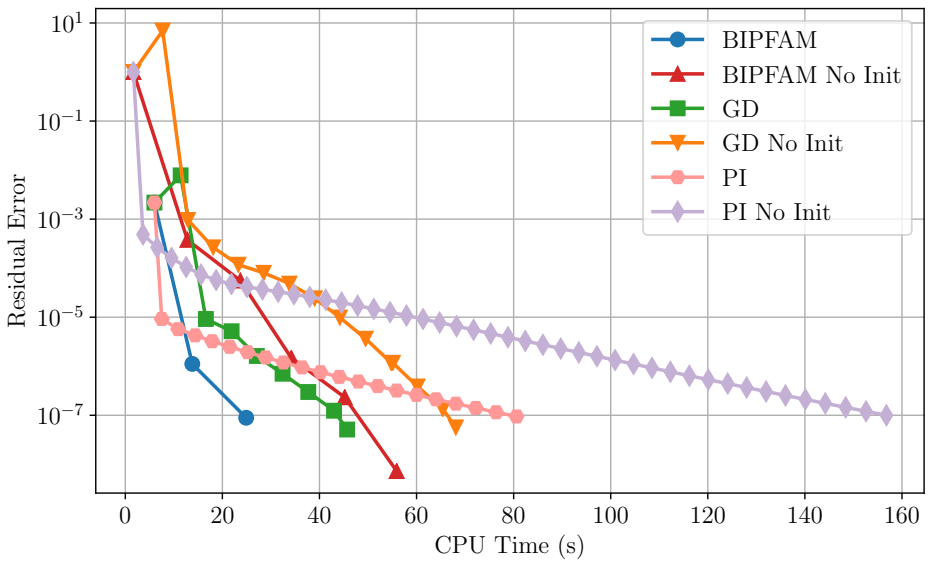


Figure 3.: Convergence histories for the first eigenvalue for SP_1 problem with $r_r = 1$ and $p = 2$.

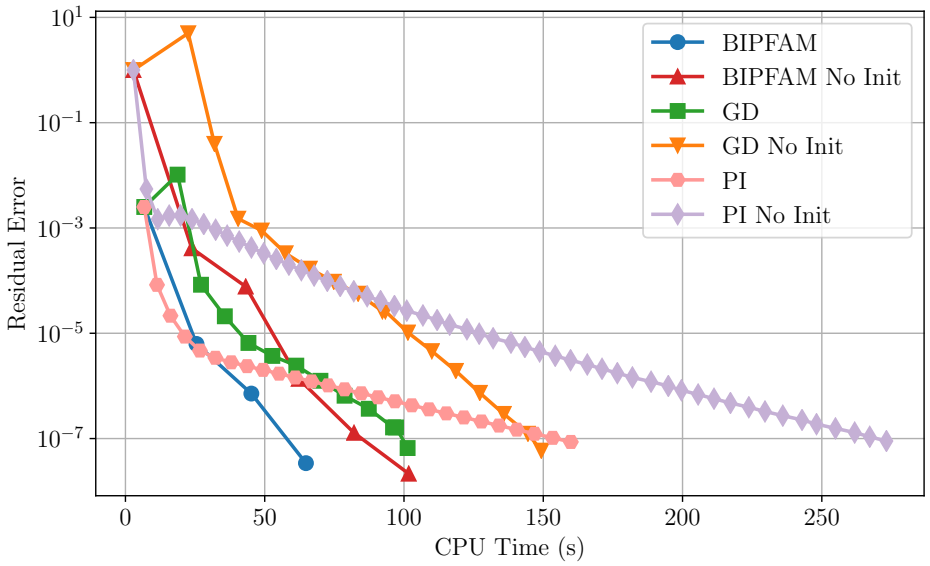


Figure 4.: Convergence histories for the first eigenvalue for SP_3 problem with $r_r = 1$ and $p = 2$.

4.2. Modified 2D C5G7

To solve a more interesting problem from the computational point of view, the 2D C5G7 problem has been modified to increase the dominance ratio from $\delta \simeq 0.76$ to $\delta \simeq 0.95$. The modification has been achieved by increasing the pin size from 1.26 cm to 2.0 cm while maintaining the fuel radius.

Tables 6 and 7 show the performance of the different eigenvalue solvers for SP_1 and SP_3 problems with $r_r = 1$ and $p = 2$ for this benchmark. Again, numerical results show that the proposed BIFPAM method with initialization is more efficient than the other methods studied for one eigenvalue calculations. Compared to the previous benchmark, some more iterations are needed to solve problem for all eigenvalue solvers. This tendency is stronger in the power iteration method where the computational times and the number of iterations are approximately duplicated due to the dominance factor closer to 1.0. Figures 5 and 6 show the residual norm, for the first eigenvalue in the SP_1 and SP_3 problems with $p = 2$ and $r_r = 1$.

Table 6.: Performance results for different eigenvalue solvers for SP_1 problem with $r_r = 1$ and $p = 2$ for the modified 2D-C5G7.

Solver Method	**Init**	**n eigs.**	**CPU Time** (s)	**n its**
Power Iteration	No	1	229	119
Power Iteration	Yes	1	151	72
BIFPAM	No	1	46	8
BIFPAM	Yes	1	32	5
Generalized Davidson	No	1	55	22
Generalized Davidson	Yes	1	40	14
Krylov-Schur	No	1	1174	24
BIFPAM	Yes	4	147	6
Generalized Davidson	Yes	4	156	59
Krylov-Schur	No	4	1805	37

Table 7.: Performance results for different eigenvalue solvers for SP_3 problem with $r_r = 1$ and $p = 2$ for the modified 2D-C5G7.

Solver Method	**Init**	**n eigs.**	**CPU Time** (s)	**n its**
Power Iteration	No	1	417	115
Power Iteration	Yes	1	231	59
BIFPAM	No	1	111	9
BIFPAM	Yes	1	78	6
Generalized Davidson	No	1	123	25
Generalized Davidson	Yes	1	94	18
Krylov-Schur	No	4	2153	24
BIFPAM	Yes	4	323	7
Generalized Davidson	Yes	4	348	66
Krylov-Schur	No	4	3331	37

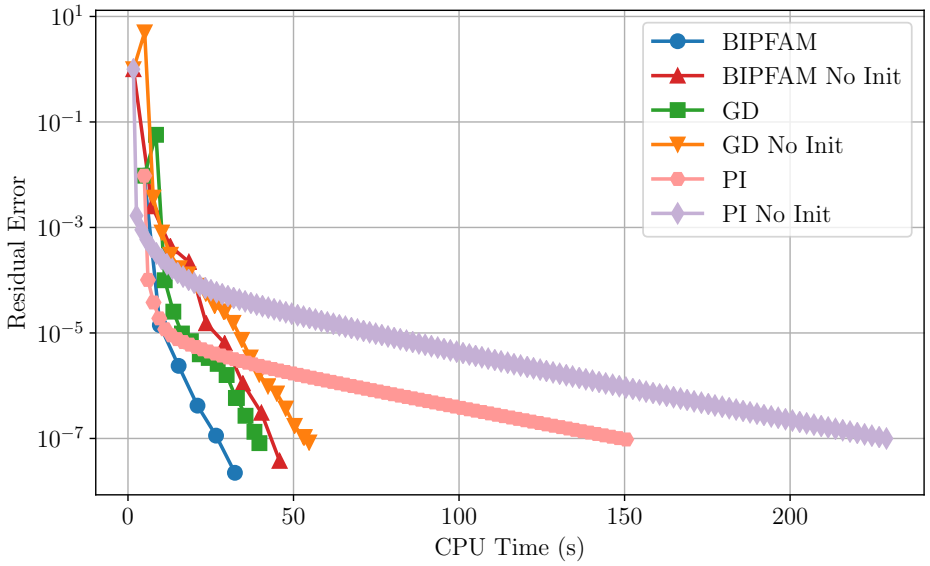


Figure 5.: Convergence histories for the first eigenvalue for SP_1 problem with $r_r = 1$ and $p = 2$ for the modified 2D-C5G7.

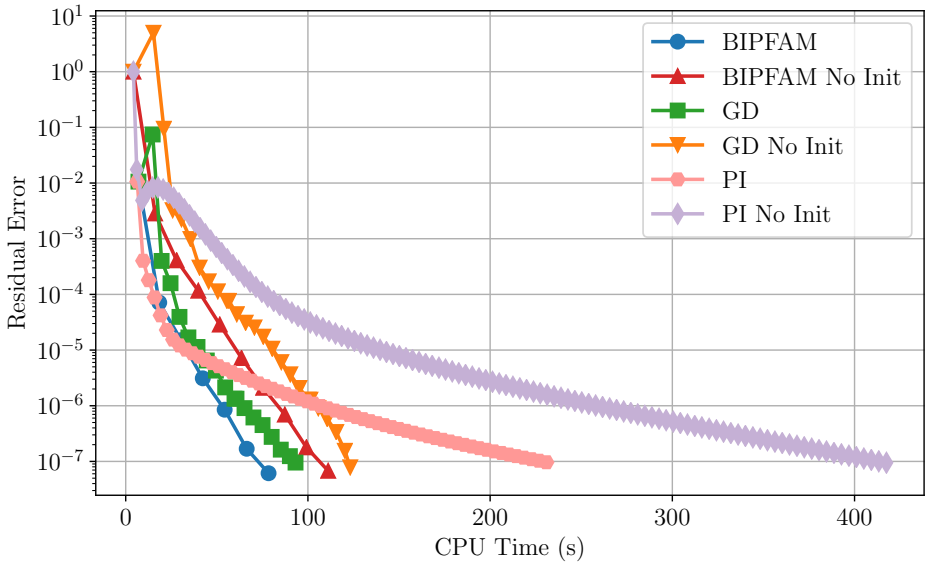


Figure 6.: Convergence histories for the first eigenvalue for SP_3 problem with $r_r = 1$ and $p = 2$ for the modified 2D-C5G7.

4.3. Three dimensional C5G7

Furthermore, the 3D-C5G7 problem [20] has been solved to test the performance of the proposed solver in a challenging three dimensional problem. The dominance ratio for this benchmark in diffusion theory is $\delta \approx 0.767$.

Since this problem has the same radial configuration as the two dimensional version, the same discretization is used in this direction $r_r = 1$ and $p = 2$. Table 8 shows the accuracy results for the fundamental eigenvalue and its corresponding eigenvector for SP_1 and the SP_3 equations for different axial refinement parameters, r_a. It can be seen that it is need $r_a = 2$ to obtain spatially converged results. In this way, the error in the pin power is less than 1% and about 400 pcm in the k_{eff} for the SP_3 equations. It must be noted that the computational time has been increased from a few minutes for the two dimensional version to several hours for the three dimensional case.

Tables 9 and 10 show the performance of the different eigenvalue solvers for SP_1 and SP_3 problems with $r_a = 1$, $r_r = 1$ and $p = 2$ for this benchmark. Again, numerical results show that the proposed BIFPAM method with initialization is more efficient than the other methods studied for one eigenvalue calculations. Figures 7 and 8 show the historic residual norm, for the first eigenvalue in the SP_1 and SP_3 problems.

Table 8.: Accuracy results for 3D-C5G7 benchmark.

Eq.	$\mathbf{r_a}$	$\mathbf{r_r}$	**p**	**Number of Cells**	**Number of DoFs**	**Eigenvalue** λ_1	$\Delta\lambda_1$	**AVG** (%)	**MRE** (%)
SP_1	0	1	2	132 496	8 964 375	1.140 10	298	4.26	3.35
	1	1	2	264 992	16 407 055	1.138 76	432	1.98	1.60
	2	1	2	529 984	31 292 415	1.138 19	489	1.42	1.19
	3	1	2	1 059 968	61 063 135	1.138 13	495	1.41	1.17
SP_3	0	1	2	132 496	17 928 750	1.140 67	241	4.40	3.51
	1	1	2	264 992	32 814 110	1.139 36	372	1.93	1.54
	2	1	2	529 984	62 584 830	1.138 80	428	0.98	0.84
	3	1	2	1 059 968	122 126 270	1.138 70	438	0.87	0.77

Table 9.: Performance results for different eigenvalue solvers for SP_1 problem with $r_a = 1$, $r_r = 1$ and $p = 2$ for the 3D-C5G7.

Solver Method	**Init**	**n eigs.**	**CPU Time** (hr)	**n its**
Power Iteration	No	1	14.9	52
Power Iteration	Yes	1	8.8	27
BIFPAM	No	1	5.2	5
BIFPAM	Yes	1	3.5	3
Generalized Davidson	No	1	6.3	21
Generalized Davidson	Yes	1	4.7	12
BIFPAM	Yes	4	21.0	4
Generalized Davidson	Yes	4	23.8	52

Table 10.: Performance results for different eigenvalue solvers for SP_3 problem with $r_a = 1$, $r_r = 1$ and $p = 2$ for the 3D-C5G7.

Solver Method	**Init**	**n eigs.**	**CPU Time** (hr)	**n its**
Power Iteration	No	1	28.3	60
Power Iteration	Yes	1	16.3	34
BIFPAM	No	1	10.9	6
BIFPAM	Yes	1	7.4	4
Generalized Davidson	No	1	21.2	24
Generalized Davidson	Yes	1	10.3	13
BIFPAM	Yes	4	40.6	4
Generalized Davidson	Yes	4	45.3	55

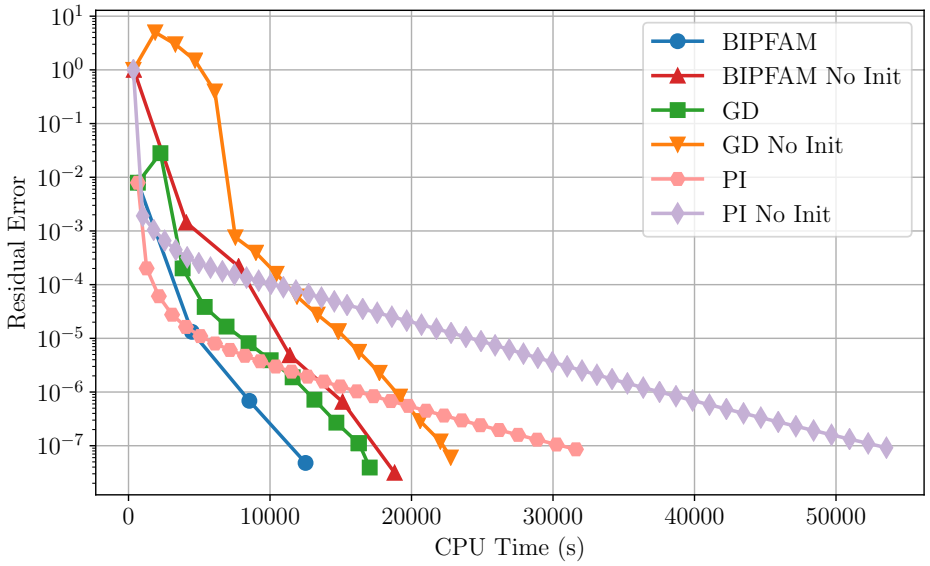


Figure 7.: Convergence histories for the first eigenvalue for SP_1 problem with $r_a = 1$, $r_r = 1$ and $p = 2$ for the 3D-C5G7.

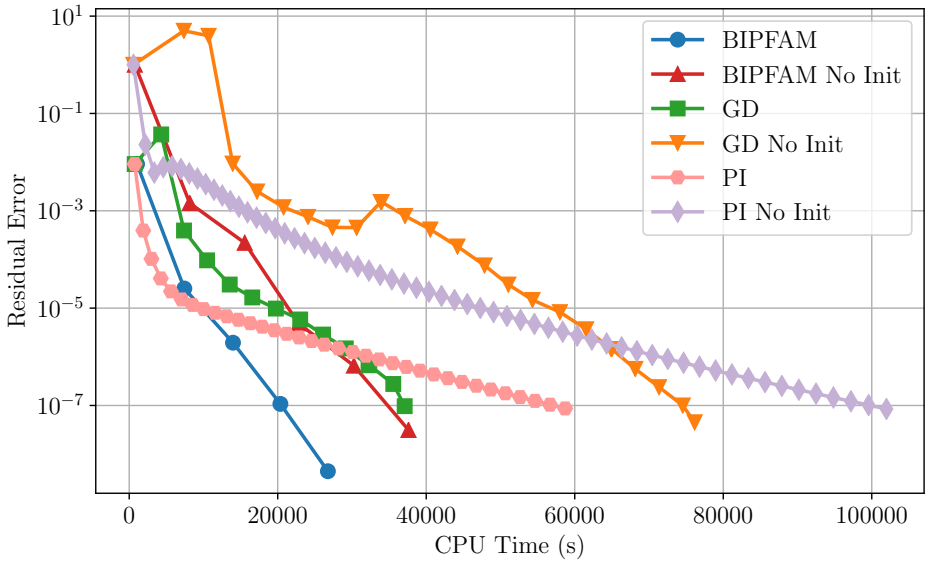


Figure 8.: Convergence histories for the first eigenvalue for SP_3 problem with $r_a = 1$, $r_r = 1$ and $p = 2$ for the 3D-C5G7.

5. Conclusions

In this work, the block inverse-free preconditioned Arnoldi method is proposed to solve the eigenvalue problem associated with the SP_N equations. The SP_N equations have been proven to be a useful approximation to the neutron transport equation especially for full core nuclear reactor calculations. To discretize the problem a high-order finite element method with a matrix-free technique is used. We take advantage of the block structure of the problem to propose the BIFPAM method to solve the eigenvalue problem. An initialization, that only takes a small percentage of the total CPU time to solve the problem, by using linear shape functions in the FEM is proposed in order to improve the convergence of the methods. The proposed BIFPAM method has been compared with the standard power iteration method, the Krylov-Schur method and the generalized Davidson method. The numerical results show that it reduces the computational time to solve the eigenvalue problem. A great advantage can be obtained when more than one eigenvalue is requested. The proposed initialization is also convenient for the power iteration method and the generalized Davidson method.

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. Moreover, it has been supported by the Cátedra of the CSN Vicente Serradell.

References

[1] P. Arbenz, U.L. Hetmaniuk, R.B. Lehoucq, and R.S. Tuminaro, *A comparison of eigensolvers for large-scale 3d modal analysis using amg-preconditioned iterative methods*, Int. J. Numer. Meth. Eng. 64 (2005), pp. 204–236.

[2] J. Baglama, *Augmented block householder arnoldi method*, Linear Algebra and its Applications 429 (2008), pp. 2315–2334.

[3] W. Bangerth, R. Hartmann, and G. Kanschat, *deal.ii – a general-purpose object-oriented finite element library*, ACM Trans. Math. Softw. 33 (2007).

[4] P. Binding, B. Najman, and Q. Ye, *A variational principle for eigenvalues of pencils of hermitian matrices*, Integr. Equat. Operat. Theor. 35 (1999), pp. 398–422.

[5] P.S. Brantley and E.W. Larsen, *The simplified P3 approximation*, Nucl. Sci. Technol. 134 (2000), pp. 1–21.

[6] M. Capilla, C. Talavera, D. Ginestar, and G. Verdú, *Application of a nodal collocation approximation for the multidimensional PL equations to the 3D takeda benchmark problems*, Ann. Nucl. Energy 40 (2012), pp. 1–13.

[7] S. Carney, F. Brown, B. Kiedrowski, and W. Martin, *Theory and applications of the fission matrix method for continuous-energy monte carlo*, Ann. Nucl. Energy 73 (2014), pp. 423–431.

[8] A. Carreño, A. Vidal-Ferràndiz, D. Ginestar, and G. Verdú, *Block hybrid multilevel method to compute the dominant λ-modes of the neutron diffusion equation*, Ann. Nucl. Energy 121 (2018), pp. 513–524.

[9] E.M. Gelbard, *Application of spherical harmonics methods to reactor problems*, Tech. Rep. WAPD-BT-20, Bettis Atomic Power Laboratory, 1960.

[10] G.H. Golub and Q. Ye, *An inverse free preconditioned krylov subspace method for symmetric generalized eigenvalue problems*, SIAM J. Sci. Comput. 24 (2002), pp. 312–334.

[11] S.P. Hamilton, T.M. Evans, G.G. Davidson, S.R. Johnson, T.M. Pandya, and A.T. Godfrey,

Hot zero power reactor calculations using the insilico code, J. Comput. Phys 314 (2016), pp. 700–711.

[12] S.P. Hamilton and T.M. Evans, *Efficient solution of the simplified PN equations*, J. Comput. Phys 284 (2015), pp. 155–170.

[13] V. Hernandez, J.E. Roman, and V. Vidal, *SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems*, ACM Trans. Math. Softw. 31 (2005), pp. 351–362.

[14] A.V. Knyazev, *Preconditioned eigensolvers – an oxymoron?*, Electron. Trans. Numer. Anal 7 (1998), pp. 104–123.

[15] M. Kronbichler and K. Kormann, *A generic interface for parallel cell-based finite element operator application*, Computers & Fluids 63 (2012), pp. 135–147.

[16] E.E. Lewis and W.F.J. Miller, *Computational Methods of Neutron Transport*, John Wiley & Sons, Ltd., New York, USA, 1984.

[17] R.B. Morgan and D.S. Scott, *Generalizations of davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. and Stat. Comput. 7 (1986), pp. 817–825.

[18] P. Quillen and Q. Ye, *A block inverse-free preconditioned krylov subspace method for symmetric generalized eigenvalue problems*, J. Comput. Appl. Math. 233 (2010), pp. 1298–1313.

[19] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, USA, 2003.

[20] M.A. Smith, E.E. Lewis, and B.C. Na, *Benchmark on deterministic transport calculations without spatial homogenisation – A 2-D/3-D MOX Fuel Assembly Benchmark (C5G7 MOX Benchmark)*, Tech. Rep. NEA/NSC/DOC(2003)16, OECD/NEA, 2003.

[21] W.M. Stacey, *Nuclear Reactor Physics*, 2nd ed., John Wiley & Sons, Weinheim, Germany, 2007.

[22] G.W. Stewart, *A krylov–schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl. 23 (2002), pp. 601–614.

[23] G. Verdú, D. Ginestar, V. Vidal, and J. Muñoz-Cobo, *3d lambda-modes of the neutron-diffusion equation*, Ann. Nucl. Energy 21 (1994), pp. 405–421.

[24] A. Vidal-Ferràndiz, R. Fayez, D. Ginestar, and G. Verdú, *Solution of the lambda modes problem of a nuclear power reactor using an h–p finite element method*, Ann. Nucl. Energy 72 (2014), pp. 338–349.

[25] A. Vidal-Ferràndiz, S. González-Pintor, D. Ginestar, G. Verdú, and C. Demazière, *Schwarz type preconditioners for the neutron diffusion equation*, J. Comput. Appl. Math. 309 (2017), pp. 563–574.

[26] J.S. Warsa, T.A. Wareing, J.E. Morel, J.M. McGhee, and R.B. Lehoucq, *Krylov subspace iterations for deterministic k-eigenvalue calculations*, Nucl. Sci. Technol. 147 (2004), pp. 26–42.

[27] K. Wu, Y. Saad, and A. Stathopoulos, *Inexact newton preconditioning techniques for large symmetric eigenvalue problems*, Electron. Trans. Numer. Anal. 7 (1998), pp. 202–214.