

Titre: Application of the augmented block Householder Arnoldi method to the calculation of non-fundamental modes of the diffusion equation
Title:

Auteurs: Alain Hébert
Authors:

Date: 2021

Type: Article de revue / Article

Référence: Hébert, A. (2021). Application of the augmented block Householder Arnoldi method to the calculation of non-fundamental modes of the diffusion equation. Annals of Nuclear Energy, 151, 8 pages.
Citation: <https://doi.org/10.1016/j.anucene.2020.107912>

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/9268/>
PolyPublie URL:

Version: Version officielle de l'éditeur / Published version
Révisé par les pairs / Refereed

Conditions d'utilisation: CC BY-NC-ND
Terms of Use:

Document publié chez l'éditeur officiel

Document issued by the official publisher

Titre de la revue: Annals of Nuclear Energy (vol. 151)
Journal Title:

Maison d'édition: Elsevier
Publisher:

URL officiel: <https://doi.org/10.1016/j.anucene.2020.107912>
Official URL:

Mention légale:
Legal notice:



Technical note

Application of the augmented block Householder Arnoldi method to the calculation of non-fundamental modes of the diffusion equation



Alain Hébert

Polytechnique Montréal, C.P. 6079 succ. "Centre-Ville", Montréal Qc. H3C 3A7, Canada

ARTICLE INFO

Article history:

Received 25 July 2020

Received in revised form 13 September 2020

Accepted 28 September 2020

Available online 9 November 2020

Keywords:

Reactor physics

Arnoldi method

Full-core calculations

TRIVAC5 code

ABSTRACT

The determination of non-fundamental modes of the diffusion equation is required for computing CANDU reactor power distribution from analysis of in-core detector readings. They are also important for understanding subcritical mode instabilities occurring in boiling water reactors. The legacy method for computing these modes is the Hotelling deflation technique based on bi-harmonic decontamination. However, the Hotelling technique becomes unstable as the number of modes increase or as their eigenvalues become closer. Effective and fast alternatives are provided with Implicit Arnoldi Restarted Methods (IRAM). Among them, we investigated the Krylov–Schur method available in the SLEPc library, and we are proposing a custom implementation of the augmented block Householder Arnoldi (ABHA) method, similar to the Open Source implementation of Prof. James Baglama. In our work, the ABHA method is applied to the neutron diffusion equation, discretized with the Raviart–Thomas and Raviart–Thomas–Schneider methods or with the mesh-centered finite difference method.

© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

The flux modes are referred to the λ -harmonics of the diffusion equation or to the simplified P_n equation. The 13 dominant modes are required for computing CANDU reactor power distribution from analysis of in-core detector readings (Levine and Diamond, 1972; Xia, 2012). They can also be used for performing *modal space-time kinetics simulations* of the core dynamics using applications such as the SMOKIN code system (Gold and Wight, 1990). The capability to generate time-averaged flux modes for CANDU-PHWR is currently provided by the MONIC code using the legacy Hotelling deflation technique (Kugler, 1976). This technique is a decontamination process of the power method so as to converge on non-fundamental modes of the eigenvalue problem, as explained by Hébert (2016) in Sect. C.2.3. An exercise is currently undertaken to develop a new generation of reactor physics codes, known as Industrial Standard Toolset (IST), for supporting operation of CANDU-PHWR across the world. The capability of generating flux modes is one of many requirements for the new IST.

Knowledge of flux modes is also important for performing stability analyses in Boiling Water Reactors (BWR). A mechanism has been identified where operation of a BWR results in *out-of-phase* power oscillations (March-Leuba and Blakeman, 1991). In this instability mode, the reactor power established a self-

sustained oscillation of large amplitude in which half the core increased power while the other half decreased. The resulting average power, however, remained essentially constant during these oscillations. In those cases, the unstable subcritical modes dominates the reactor response, thus providing an explanation for the observed out-of-phase oscillations. The representation of these instabilities requires the knowledge of a minimum of five modes.

The finite-element flux calculation code TRIVAC5 includes several numerical approach for solving the diffusion equation and the simplified P_n equation, such as the mesh-centered finite difference approximation (MCFD), the Raviart–Thomas and the Raviart–Thomas–Schneider finite element methods (Hébert, 1987; Hébert, 1993; Hébert, 2008). It is worth mentioning that TRIVAC5 uses the same Raviart–Thomas discretization technique as COCAGNE (Hoareau et al., 2008), the flux solution module embedded in ODYSSEE, the new reactor physics platform conjointly developed by EDF and Framatome. Flux solution support in the full-core simulation code DONJON5 is based on finite-difference and finite-element techniques available in TRIVAC5 (Hébert, 2016).

TRIVAC5 currently uses a MCFD discretization approach with Hotelling deflation, similar to the MONIC code, for computing flux modes in a CANDU-PHWR. After years of use, we observe that the Hotelling technique become unstable as the number of modes increase or as their eigenvalues become closer. The advantage of this technique is the use of the accelerated power method, but its convergence rate is not as fast as that of the fundamental mode. In addition,

E-mail address: alain.hebert@polymtl.ca

this technique cannot calculate those modes belonging to a multiple eigenvalue, because convergence depends on the magnitude of the eigenvalue ratio being calculated and the next one.

Several authors are proposing more effective methods than the Hotelling deflation technique for calculating non-fundamental flux modes. Effective and fast alternatives to Hotelling deflation are provided with Implicit Arnoldi Restarted Methods (IRAM). The original IRAM implementation was contributed by [Lehoucq et al. \(1998\)](#) in the legacy ARPACK package. A few years later, [Stewart \(2001\)](#) proposed the Krylov-Schur method which is mathematically equivalent to the IRAM implementation, but offers two practical advantages. First, it is easier to deflate converged Ritz vectors; second, the potential forward instability of the QR algorithm is avoided. The Krylov-Schur method is now used in Matlab, SciPy ([Virtanen et al., 2020](#)) and in the SLEPc library ([Hernandez et al., 2005](#)). More recently, two block implementations of the Krylov-Schur method were presented by [Baglama \(2008\)](#) and [Baker et al. \(2009\)](#).

Our first investigation of the Krylov-Schur method, based on the SLEPc library, was conducted in collaboration with the Universitat Politècnica de València and was presented by [Bernal et al. \(2017\)](#). In this paper, we are proposing a custom implementation of the augmented block Householder Arnoldi (ABHA) method, similar to the Open Source implementation of [Baglama \(2008\)](#). Both IRAM approaches were implemented in TRIVAC5 and were compared to the legacy Hotelling deflation technique on a set of benchmarks and on a time-averaged CANDU6 representation.

2. Implementation details

The discretization of the multigroup diffusion equation in TRIVAC5 is producing a generalized matrix eigenvalue system of the form

$$\left(\mathbb{A} - \frac{1}{\lambda_\ell} \mathbb{B} \right) \mathbf{v}_\ell = \mathbf{0}; \quad \ell = 1, L \quad (1)$$

where \mathbb{A} and \mathbb{B} are non-symmetric real matrices, λ_ℓ is the ℓ -th eigenvalue and \mathbf{v}_ℓ is the corresponding eigenvector. The right-hand side $\mathbf{0}$ is the zero-vector (whose components are zero). The non-symmetry of matrices \mathbb{A} and \mathbb{B} is due to the discretization process that is generally performed for $G > 1$ energy groups. The order L of these matrices is equal to the product of the number of energy groups times the number of flux unknowns per group. Eq. (1) has L eigen-solutions or flux modes, each of them corresponding to root λ_ℓ of the characteristic equation, written as

$$\det \left(\mathbb{A} - \frac{1}{\lambda_\ell} \mathbb{B} \right) = 0; \quad \ell = 1, L. \quad (2)$$

We are concerned about the *dominant flux modes*, i. e., those with maximum eigenvalues λ_ℓ . At most, a dozen of them need to be calculated. If the reactor geometry has symmetries, some eigenvalues may be degenerated (i.e., $\lambda_k = \lambda_\ell$ if $k \neq \ell$). The *fundamental*

solution refers to the first harmonics of Eq. (1). $K_{\text{eff}} = \lambda_1$ is the *effective multiplication factor* and $\Phi = \mathbf{v}_1$ is the discretized *particle flux*. Only the fundamental solution corresponds to a positive particle flux defined over the domain. As stated by [Reuss \(2008\)](#) in Section 5.2.8 of his reference textbook, a fundamental solution is never degenerated. The fundamental problem is therefore written

$$\left(\mathbb{A} - \frac{1}{K_{\text{eff}}} \mathbb{B} \right) \Phi = \mathbf{0}. \quad (3)$$

Matrix \mathbb{A} generally exhibits a block structure similar to the example depicted in [Fig. 1](#), where the diagonal blocks are symmetric and the $\mathbb{A}_{g,g}$ blocks are positive definite. The complete matrix system can be written in a block structure, each block representing specific values of the primary and secondary energy group indices.

The basic strategy for finding the first m dominant eigensolutions of Eq. (1) is the *block inverse power method*, an iterative algorithm written as

$$\begin{aligned} \mathbb{Q}^{(0)} &= [\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}, \dots, \mathbf{v}_m^{(0)}] \text{ given} \\ \mathbb{Z}^{(k+1)} &= \mathbb{A} \mathbb{Q}^{(k)} \text{ if } k \geq 0 \\ \mathbb{Q}^{(k+1)} \mathbb{R}^{(k+1)} &= \mathbb{Z}^{(k+1)} \text{ (economy size QR factorization)} \\ \mathbb{D} &= \text{diag}(\mathbb{R}^{(k+1)}) \text{ at convergence} \end{aligned} \quad (4)$$

where $\mathbb{Q}^{(0)}$ is a non-zero initial estimate of the dominant eigenvectors \mathbb{A} is the *iterative matrix* defined as

$$\mathbb{A} = \mathbb{A}^{-1} \mathbb{B}. \quad (5)$$

and where \mathbb{D} is a diagonal matrix containing the m dominant eigenvalues at convergence.

The IRAM (Implicit Restarted Arnoldi Method) algorithm was proposed by [Sorensen \(1992\)](#) as a Krylov subspace method to compute dominant eigensolutions and to speed up iterations in solving large and sparse nonsymmetric eigenvalues systems similar to Eq. (1). A legacy implementation of the IRAM is provided by the ARPACK package ([Lehoucq et al., 1998](#)) as a set of Application Programming Interfaces (API) based on LAPACK ([Anderson et al., 1999](#)) and contributed by Rice University and Argonne National Laboratory.

In practical applications of the IRAM, the matrix product $\mathbb{A} \mathbb{Q}^{(k)}$ is performed repetitively, being responsible for most of the CPU resources used in the flux mode calculation. Matrices \mathbb{A} and $\mathbb{Q}^{(k)}$ have size $L \times L$ and $L \times m$, respectively. The iterative matrix order L can exceed a million in some cases, making the reconstruction of \mathbb{A} in memory unfeasible. Matrix $\mathbb{Q}^{(k)}$ can yet be stored, as the value of m is generally small. Matrix \mathbb{A} is made of an assembly of groupwise submatrices, each of them being sparse; they are stored in sparse storage mode. The IRAM implementation strategy consists to program a *callback function* `atv(b)` defined as

$$\mathbf{x} := \text{atv}(\mathbf{b}) \iff \mathbf{x} = \mathbb{A} \mathbf{b} \quad (6)$$

$$\begin{bmatrix} \mathbb{A}_{11} & & & & \\ -\mathbb{A}_{21} & \mathbb{A}_{22} & & & \\ & -\mathbb{A}_{32} & \mathbb{A}_{33} & & \\ & -\mathbb{A}_{42} & -\mathbb{A}_{43} & \mathbb{A}_{44} & -\mathbb{A}_{45} \\ & & & -\mathbb{A}_{54} & \mathbb{A}_{55} \end{bmatrix} \begin{bmatrix} v_{1,\ell} \\ v_{2,\ell} \\ v_{3,\ell} \\ v_{4,\ell} \\ v_{5,\ell} \end{bmatrix} = \frac{1}{\lambda_\ell} \begin{bmatrix} \mathbb{B}_{11} & \mathbb{B}_{12} & \mathbb{B}_{13} & \mathbb{B}_{14} & \mathbb{B}_{15} \\ \mathbb{B}_{21} & \mathbb{B}_{22} & \mathbb{B}_{23} & \mathbb{B}_{24} & \mathbb{B}_{25} \\ & & & & \\ & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} v_{1,\ell} \\ v_{2,\ell} \\ v_{3,\ell} \\ v_{4,\ell} \\ v_{5,\ell} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Fig. 1. A multigroup partitioning example with $G = 5$.

and computing the product $\mathbb{A}\mathbf{b}$, without an inversion of matrix \mathbb{A} , using iterative techniques based on *sparse matrix algebra*. The `atv` function is applied on each column of matrix $\mathbb{Q}^{(k)}$ at each iteration.

The application of the IRAM algorithm involves the transformation of a generalized eigenvalue problem into an ordinary eigenvalue problem, the latter been solved with the inverse power method. If the condition number of matrix \mathbb{B} is high, it is necessary to converge the solutions of the callback function with very high precision to obtain a solution of the generalized eigenvalue problem with acceptable precision. Such a constraint doesn't exist with methods based on the preconditioned power method such as the *symmetrical variational acceleration technique* (SVAT) and used by the Hotelling deflation technique (Hébert, 1986). Unfortunately, the SVAT is unsuccessful at computing higher harmonics of the eigenvalue problem. The available implementation of the IRAM algorithm is only compatible with the inverse power method, requiring convergence of the linear system $\mathbb{A}\mathbf{x} = \mathbb{B}\mathbf{b}$ at each call of the `atv` function. Failure to converge the solution of this linear system may result in wrong eigensolution. Practical implementations of the IRAM algorithm are based on an iterative solution of this linear system with the introduction of a *preconditioning matrix* $\mathbb{M} \simeq \mathbb{A}^{-1}$. Available preconditioning options are the Jacobi, Gauss–Seidel, SSOR or ADI preconditioning matrices.

We made two implementations of the IRAM over the recent years. These implementations differ in the choice of the Krylov subspace orthogonal iteration strategy used to speedup algorithm (4) and in the way the call-back function `atv` was programmed.

2.1. The Krylov–Schur method with SLEPc

The first IRAM implementation in TRIVAC5 was developed by Bernal et al. (2017) as a collaboration with the Universitat Politècnica de València. This implementation was based on the Krylov–Schur method available in SLEPc (Hernandez et al., 2005), a software library for the solution of large-scale sparse eigenvalue problems on parallel computers. It is an extension of PETSc (Balay et al., 2020) and can be used for linear eigenvalue problems in either standard or generalized form, with real or complex arithmetic. All the linear algebra routines and sparse matrix storage support required by the Krylov–Schur method is provided by the SLEPc powerful API.

The Krylov–Schur method was not committed in the production version of DRAGON5/DONJON5. All numerical results presented in this study were obtained with the original version of Bernal et al. (2017). SLEPc offers many possibilities to optimize the implementation of the IRAM solver, but we chose to keep the 2017 settings for consistency reasons.

2.2. The augmented block Householder Arnoldi method

We are providing a simple implementation of the IRAM algorithm based on the *augmented block Householder Arnoldi* (ABHA) method that makes use of Householder reflections to maintain orthogonality and where restarting is accomplished by augmentation of the Krylov subspace with Schur vectors (Baglama, 2008). Our implementation of the ABHA method takes the form of a custom 465-line Fortran-90 module named `ALBEIGS.f90` and of a 304-line Fortran callback function named `FLDTMX.f`. All linear algebra routines and sparse matrix storage support are provided by the UTILIB5 library, already available in the official DRAGON5/DONJON5 distribution and used by the legacy Hotelling decontamination technique. Among utility linear algebra tools recovered from the Utilib library are the Cholesky $L - D - L^T$ factorization, the economy size QR factorization, and a powerful full-matrix eigensolver based on the *shifted Hessenberg QR algorithm* as pro-

posed by Robles (2017). ARPACK and LAPACK are not used. The new IRAM capabilities are not adding software configuration complexity. The ABHA method was committed in production version 5.0.7 of DRAGON5/DONJON5.

An interesting capability of the ABHA method is worth mentioning. The block implementation makes possible the initialization of each specific eigenvector to improve computation efficiency. By default, initialization is made with a triangular matrix with ones on the upper part and zeros on the lower part. The fundamental eigenvector is therefore initialized with ones by default.

2.3. Implementation of the callback function in the ABHA method

The callback function used by the ABHA method with a TRIVAC discretization is based on a *preconditioned iterative process*. All matrix operations are performed in sparse matrix algebra using the UTILIB5 library. The sparse matrix algebra techniques used in TRIVAC5 are specific to the finite-difference, nodal collocation (Hébert, 1987), Raviart–Thomas (Hébert, 1993) and Raviart–Thomas–Schneider (Hébert, 2008) finite-element approximations. We are assuming that the diagonal blocks $\mathbb{A}_{g,g}$ can be expressed as an *alternating direction implicit* (ADI) splitting and that all other blocks $\mathbb{A}_{g,h}$ ($g \neq h$) and $\mathbb{B}_{g,h}$ are diagonal (Hébert, 1986). An ADI split of block $\mathbb{A}_{g,g}$ is written

$$\mathbb{A}_{g,g} = \mathbb{U} + \mathbb{P}_x \mathbb{X} \mathbb{P}_x^T + \mathbb{P}_y \mathbb{Y} \mathbb{P}_y^T + \mathbb{P}_z \mathbb{Z} \mathbb{P}_z^T \quad (7)$$

where

\mathbb{U} = matrix containing the diagonal elements of $\mathbb{A}_{g,g}$.

$\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ = symmetrical matrices containing the nondiagonal elements of $\mathbb{A}_{g,g}$.

$\mathbb{P}_x, \mathbb{P}_y, \mathbb{P}_z$ = permutation matrices that ensure a minimum bandwidth for matrices \mathbb{X}, \mathbb{Y} and \mathbb{Z} .

The corresponding preconditioning matrix can be defined as

$$\mathbb{M}_{g,g} = (\mathbb{P}_z \tilde{\mathbb{Z}}^{-1} \mathbb{P}_z^T) \mathbb{U} (\mathbb{P}_y \tilde{\mathbb{Y}}^{-1} \mathbb{P}_y^T) \mathbb{U} (\mathbb{P}_x \tilde{\mathbb{X}}^{-1} \mathbb{P}_x^T) \quad (8)$$

where

$$\tilde{\mathbb{X}} = \mathbb{X} + \mathbb{P}_x^T \mathbb{U} \mathbb{P}_x, \quad \tilde{\mathbb{Y}} = \mathbb{Y} + \mathbb{P}_y^T \mathbb{U} \mathbb{P}_y \quad (9)$$

and

$$\tilde{\mathbb{Z}} = \mathbb{Z} + \mathbb{P}_z^T \mathbb{U} \mathbb{P}_z. \quad (10)$$

In the case of an hexagonal tridimensional domain, the ADI split of Eq. (7) is generalized as

$$\mathbb{A} = \mathbb{U} + \mathbb{P}_w \mathbb{W} \mathbb{P}_w^T + \mathbb{P}_x \mathbb{X} \mathbb{P}_x^T + \mathbb{P}_y \mathbb{Y} \mathbb{P}_y^T + \mathbb{P}_z \mathbb{Z} \mathbb{P}_z^T. \quad (11)$$

Matrices $\mathbb{W}, \mathbb{X}, \mathbb{Y}$ and \mathbb{Z} are diagonal banded matrices where all the components, located within their external profile, are non-zero. This type of preconditioning is efficient as inverses $\tilde{\mathbb{W}}^{-1}, \tilde{\mathbb{X}}^{-1}, \tilde{\mathbb{Y}}^{-1}$ and $\tilde{\mathbb{Z}}^{-1}$ don't need to be explicitly calculated. Any linear system of the form $\mathbb{X}\mathbf{x} = \mathbf{S}$ can be efficiently solved using a Cholesky factorization approach. The ADI splitting approach based on Eqs. (7) or (11) is not producing any fill-in. It is only available in the Utilib5 library and should not be confused with the incomplete Cholesky factorization IC(0), a more general preconditioning technique available in ARPACK and SLEPc.

In the particular case where the up-scattering blocks vanish, i.e., if $\mathbb{A}_{g,h} = \mathbb{O}$ for all group indices $h > g$, the product $\mathbf{x} = \mathbb{A}\mathbf{b}$ in the callback function can be evaluated in a recursive way, using

$$\begin{aligned} \mathbf{x}_1 &= \mathbb{A}_{1,1}^{-1} \sum_{h=1}^G \mathbb{B}_{1,h} \mathbf{b}_h \\ \mathbf{x}_g &= \mathbb{A}_{g,g}^{-1} \left(\sum_{h=1}^{g-1} \mathbb{A}_{g,h} \mathbf{x}_h + \sum_{h=1}^G \mathbb{B}_{g,h} \mathbf{b}_h \right) \quad \text{if } g = 2, G. \end{aligned} \quad (12)$$

Using the ADI partitioning of Eqs. (7)–(10), any groupwise multiplication of the form $\mathbf{x}_g = \mathbb{A}_{g,g}^{-1} \mathbf{S}_g$ can be replaced by an iterative algorithm written as

$$\begin{aligned} \mathbf{x}_g^{(0)} & \text{ given} \\ \mathbf{x}_g^{(j+1)} &= \mathbf{x}_g^{(j)} + \mathbb{M}_g (\mathbf{S}_g - \mathbb{A}_{g,g} \mathbf{x}_g^{(j)}) \quad \text{if } j \geq 0. \end{aligned} \quad (13)$$

One option is to perform a fixed number J of iterations, sufficient for convergence. One also has the option to accelerate the fixed-point algorithm (13) using the *Livolt acceleration* method or the *GMRES* (Generalized Minimum RESidual) algorithm, as presented in Sect. C.1.3 of Hébert (2016).

If up-scattering is present, an additional level of thermal iterations should be added so as to converge on the up-scattering reaction rates. Note that the IRAM is an inverse power method, not a preconditioned power method similar to the symmetric variational acceleration technique (SVAT) introduced by Hébert (1986). If the thermal iterations are not correctly converged, the eigenvalue spectrum will come out in error.

3. Numerical results

All the flux mode calculation techniques have been implemented in the finite-element code TRIVAC5 (Hébert, 1987) and validated on a set of simple two-group diffusion theory 3-D benchmarks. A time-averaged CANDU6 representation is also investigated. Only the eigenvalues are shown in this work because the same TRIVAC5 discretization is used with every flux mode calculation technique. If the eigenvalue spectrum matches, the eigenvectors will do so. Agreement on reaction rate distributions was verified for the first IAEA-3D benchmark but not reported. Agree-

ment between Hotelling deflation and ABHA methods on any benchmark can be reproduced with the Open-Source version of the code. All CPU times reported in this Section are obtained on a 8-Core Intel Core i9, 2.4 GHz laptop (MacBookPro15,3).

3.1. The IAEA-3D benchmark

The IAEA-3D benchmark (Argonne Code Center, 1977) is a simplified representation of a PWR with one-eighth symmetry in Cartesian geometry, as depicted in Fig. 2. Although Fig. 2 shows one-eighth symmetry of this reactor, the modal calculation was performed for the whole core, after unfolding the one-eighth benchmark geometry. Vacuum boundary conditions are set at axial levels $z = 0$ and $z = 380$ cm. A Cartesian mesh of $17 \times 17 \times 5$ was used. The length of each mesh in X and Y directions are 20 cm. The axial mesh sizes are: 20, 130, 130, 80, and 20 cm. Cross-section data for this benchmark is provided in Section 5 of Hébert (2020). A detailed convergence study of this Cartesian geometry with Raviart–Thomas finite elements was presented by Hébert (1993). Here, we selected parabolic ($K = 2$) Raviart–Thomas finite elements with a Gauss–Legendre (superconvergent) quadrature. The eigenvalue spectrum obtained using the three approaches is presented in Table 1. The eigenvalues of modes 2, 3, 5, 6, 8 and 12 cannot be computed without unfolding the one-eighth benchmark geometry. Duplicate eigenvalues are due to the benchmark symmetry. We observe the ineffectiveness of the legacy Hotelling decontamination technique on this benchmark. The Hotelling technique is unable to find duplicate modes, fail to find some of them and suffers from convergence difficulties. Many of the Krylov–Schur modes presented in Table 1 were found by Bernal et al. (2017), with the exception of those resulting from the unfolding of the domain. The CPU time required for obtaining the modes is

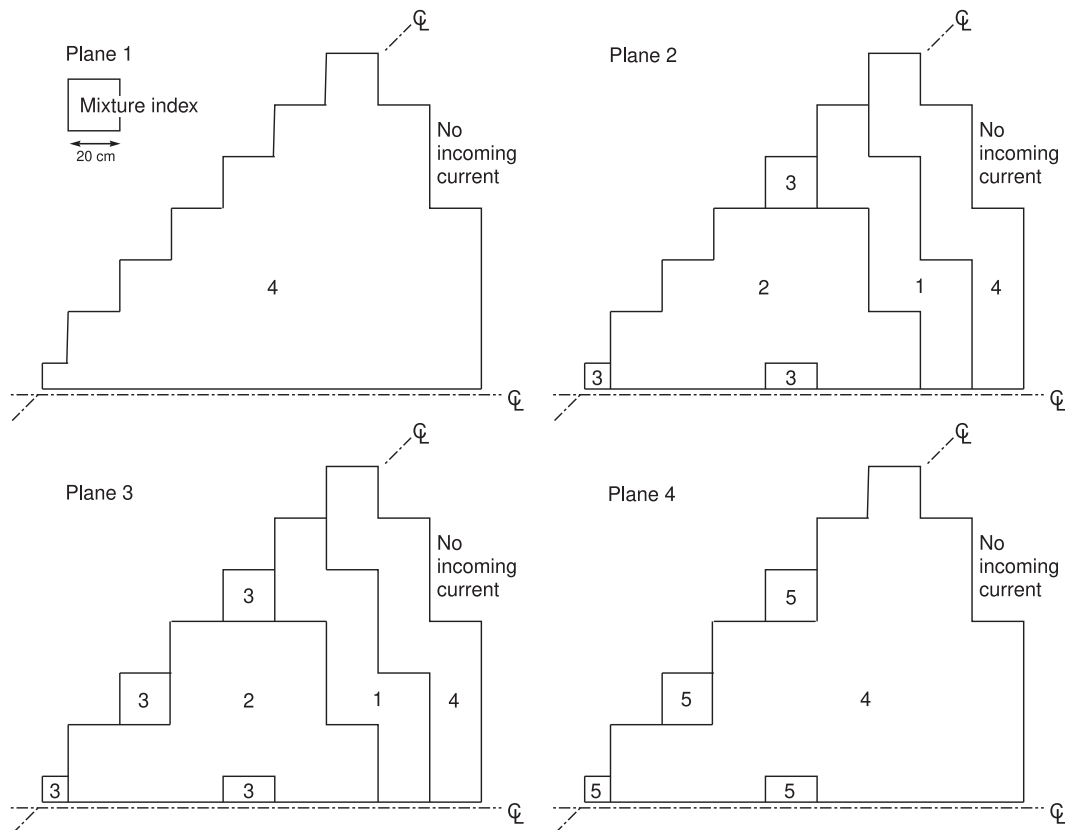


Fig. 2. Description of the IAEA-3D benchmark.

Table 1
Eigenvalue spectrum of the IAEA-3D benchmark (parabolic polynomials).

mode	Hotelling	Krylov–Schur	ABHA
1	1.029287	1.029285	1.029285
2	1.017120	1.017120	1.017120
3	(failure)	1.017120	1.017120
4	1.014651	1.014651	1.014651
5	1.003770	1.003771	1.003771
6	(failure)	1.003771	1.003771
7	(failure)	1.002558	1.002558
8	(failure)	0.997148	0.997148
9	0.991518	0.991518	0.991518
10	(failure)	0.990508	0.990508
11	0.986883	0.986882	0.986882
12	(failure)	0.984233	0.984232

120 s for the Hotelling technique, 13.3 s for the Krylov–Schur method and 10.4 s for the ABHA method.

The numerical results presented in Table 1 are not spatially converged. We have redone the same numerical exercise using cubic ($K = 3$) Raviart–Thomas finite elements in Table 2. In this case, the CPU time required for obtaining the modes is 150 s for the Hotelling technique, 44.6 s for the Krylov–Schur method and 26.7 s for the ABHA method.

3.2. The VVER-1000 benchmark

The VVER-1000 3D benchmark (Chao and Shatilla, 1995) is a simplified representation of a *vodo-vodyanoi energetichesky reaktor* (VVER) defined by Chao and Shatilla (1995) and depicted in Fig. 3. The core symmetry is one-sixth core cyclic, radially. The reflector region outside the fuel assemblies is not explicitly modeled and is assumed to be represented by a radial albedo $\beta = 0.125$. An axial albedo $\beta = 0.15$ is used, as proposed in the benchmark definition. The axial mesh sizes is 20 cm. The total height of the core is 200 cm and the eccentric control rod (mixture 4) in the third ring of hexagons are inserted axially for 100 cm. Two-group diffusion coefficients and cross sections are provided in Table 3 for five different material mixtures.

We first present verification results about TRIVAC5 convergence on the VVER-1000 benchmarks obtained using the preconditioned power method with a symmetric variational acceleration technique (SVAT) as presented by Hébert (1986). Radial mesh-splitting is the number of regions per hexagon in finite-difference cases and the number of lozenges per hexagon in the Raviart–Thomas–Schneider cases. N_{tot} is the number of unknowns per energy group. Statistics on power distribution accuracies are given by ϵ_{max} and $\bar{\epsilon}$, as defined by Hébert (2008). The finite-difference and Raviart–Thomas–Schneider results are presented in Tables 4 and 5, respectively. We note the low-order convergence of finite difference approaches for VVER problems. Numerical results

Table 2
Eigenvalue spectrum of the IAEA-3D benchmark (cubic polynomials).

mode	Hotelling	Krylov–Schur	ABHA
1	1.028982	1.028980	1.028980
2	1.017052	1.017052	1.017052
3	(failure)	1.017052	1.017052
4	1.014952	1.014952	1.014952
5	1.004151	1.004151	1.004151
6	(failure)	1.004151	1.004151
7	(failure)	1.002960	1.002961
8	(failure)	0.996990	0.996992
9	0.994732	0.994732	0.994733
10	(failure)	0.991208	0.991208
11	(failure)	0.990664	0.990665
12	(failure)	0.984512	0.984510

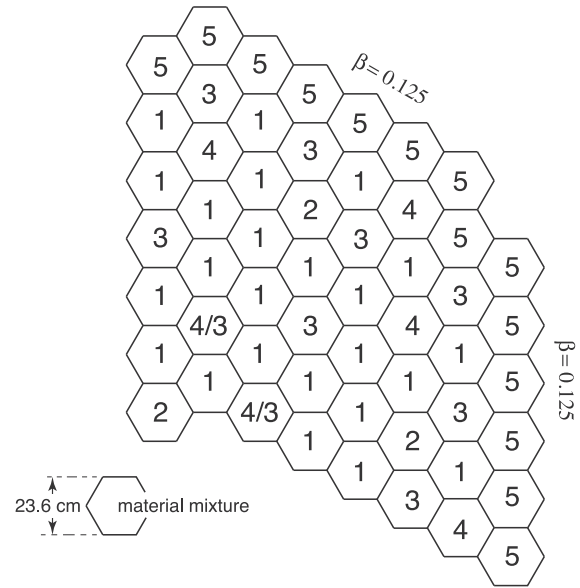


Fig. 3. Description of the VVER-1000 3D benchmark.

demonstrate the capability of high-order polynomial expansions for obtaining accurate results.

Next, we obtain the eigenvalue spectrum for a discretization based on parabolic ($K = 2$) Raviart–Thomas–Schneider finite elements, with 3 lozenges per hexagon and with an analytical integration. The symmetric variational acceleration method (SVAT) used by the Hotelling technique was deactivated so as to help the deflation process, at the cost of increasing number of iterations. The eigenvalue spectrum obtained using the three approaches is presented in Table 6. An unexpected observation is the occurrence of conjugate complex pairs of eigenvalues in the flux mode spectrum. The same complex pairs were obtained with both the Krylov–Schur and ABHA methods, in spite of their dissimilar implementations. The existence of complex conjugate pairs is not a fundamental issue of this study but we do believe that this issue deserves further investigations. The Hotelling technique was unable to find these complex modes. The CPU time required for obtaining the modes is 1500 s for the Hotelling technique, 57.9 s for the Krylov–Schur method and 84.8 s for the ABHA method.

3.3. The VV1K3D benchmark

The VVER-1000 3D benchmark of Section 3.2 was modified by Bernal et al. (2017) so as to remove the complex eigenvalues from the flux mode spectrum. The modification, known as VV1K3D, consists to define a non-symmetrical assembly layout, as depicted in Fig. 4 where the non-symmetric assemblies with respect to a rotational symmetry are highlighted. The albedo boundary conditions are replaced by zero-flux conditions. The cross-section data remains the same and we used the same discretization as for the VVER-1000 3D benchmark. The eigenvalue spectrum obtained using the three approaches is presented in Table 7. Krylov–Schur results have already been presented by Bernal et al. (2017) and are reproduced here for completeness. We observe that making the reactor layout slightly non-symmetrical is sufficient to obtain 12 distinct eigenvalues. Deactivation of the SVAT enable the Hotelling technique to find the complete spectrum. The CPU time required for obtaining the modes is 2400 s for the Hotelling technique, 52.4 s for the Krylov–Schur method and 77.8 s for the ABHA method.

Table 3

Cross-section data for the hexagonal VVER-1000 3D benchmark.

Mixture	Group	D^g (cm)	Σ_t^g (cm ⁻¹)	$\nu\Sigma_f^g$ (cm ⁻¹)	$\Sigma_{s0}^{g \rightarrow g+1}$ (cm ⁻¹)
1	1	1.3832	2.48836×10^{-2}	4.81619×10^{-3}	1.64977×10^{-2}
	2	0.386277	6.73049×10^{-2}	8.46154×10^{-2}	
2	1	1.38299	2.62865×10^{-2}	4.66953×10^{-3}	1.47315×10^{-2}
	2	0.389403	8.10328×10^{-2}	8.52264×10^{-2}	
3	1	1.39522	2.45662×10^{-2}	6.04889×10^{-3}	1.56219×10^{-2}
	2	0.386225	8.44801×10^{-2}	1.19428×10^{-1}	
4	1	1.39446	2.60117×10^{-2}	5.91507×10^{-3}	1.40185×10^{-2}
	2	0.387723	9.89671×10^{-2}	1.20497×10^{-1}	
5	1	1.39506	2.46141×10^{-2}	6.40256×10^{-3}	1.54981×10^{-2}
	2	0.384492	8.93878×10^{-2}	1.29281×10^{-1}	

Table 4Finite-differences VVER-1000 3D benchmark calculations.^(a)

Type of quadrature	Mesh-splitting				Δk_{eff} (pcm)	ϵ_{max} (%)	$\bar{\epsilon}$ (%)	CPU time (s)
	radial	axial	N_{tot}	k_{eff}				
Mesh-corner	1	5 + 5	5722	1.005621	-106.0	38.5	12.6	0.2
finite differences	6	10 + 10	14495	1.007725	104.4	15.8	6.5	0.6
Mesh-centered	1	5 + 5	1630	1.013683	3402.0	700.2	49.2	0.8
finite differences	6	10 + 10	22820	1.007724	104.3	13.3	4.3	1.2
	24	15 + 15	122250	1.006910	22.9	4.5	1.5	8.7
	54	20 + 20	358600	1.006773	9.2	2.3	0.8	65.0
	96	25 + 25	790550	1.006736	5.4	1.4	0.5	209.5

^(a) The reference effective multiplication factor is $k_{\text{eff}} = 1.006681$.**Table 5**Raviart–Thomas–Schneider VVER-1000 3D benchmark calculations.^(a)

Type of quadrature	Poly. order	Mesh-splitting				Δk_{eff} (pcm)	ϵ_{max} (%)	$\bar{\epsilon}$ (%)	CPU time (s)
		radial	axial	N_{tot}	k_{eff}				
Analytical	1	3	5 + 5	20499	1.006724	4.3	7.2	2.3	0.6
		12	10 + 10	160236	1.006621	-6.1	2.9	1.0	11.2
	2	3	5 + 5	140676	1.006601	-8.1	1.5	0.6	5.5
		12	10 + 10	1110384	1.006668	-1.4	0.3	0.1	112.0
	3	3	5 + 5	448551	1.006672	-0.9	0.2	0.1	24.2
		12	10 + 10	3554604	1.006681	≈ 0	≈ 0	≈ 0	616.8
Gauss-Lobatto	1	3	5 + 5	20499	1.009862	318.0	25.2	8.1	0.5
		12	10 + 10	160236	1.007397	71.6	9.0	2.9	8.2
	2	3	5 + 5	140676	1.006581	-10.1	2.3	0.9	5.9
		12	10 + 10	1110384	1.006652	-2.9	0.6	0.2	105.0
	3	3	5 + 5	448551	1.006656	-2.6	0.5	0.2	29.6
		12	10 + 10	3554604	1.006681	-0.1	≈ 0	≈ 0	614.2
Gauss-Legendre	1	3	5 + 5	20499	1.005209	-147.3	7.6	2.4	1.4
		12	10 + 10	160236	1.006289	-39.2	1.5	0.6	6.1
	2	3	5 + 5	140676	1.006641	-4.1	0.6	0.2	8.3
		12	10 + 10	1110384	1.006681	-0.1	≈ 0	≈ 0	71.6
	3	3	5 + 5	448551	1.006691	0.9	0.1	≈ 0	62.8
		12	10 + 10	3554604	1.006681		Reference		595.1

^(a) The reference effective multiplication factor is $k_{\text{eff}} = 1.006681$.

3.4. The time-averaged CANDU6 case

The fourth numerical experiment is a legacy CANDU6 representation recovered from the non-regression data directory of the official DONJON5 distribution, version 5.0.7 (Hébert et al., 2014). This dataset is located at position `Version5/Donjon/data/candu6_iram.x2m`. The Cartesian mesh is $26 \times 26 \times 12$ and the number of energy groups is $G = 2$.

Flux modes of a CANDU-PHWR are computed in time-averaged conditions. The time-average model (Rouben, 2007) is formulated to take into account the spatial distribution of properties over time, arising from different fuel irradiation in different fuel bundles. As opposed to the axially-uniform model, the time-average model

does not average lattice properties along a channel. Instead, it takes into account the fact that individual fuel channels are refuelled once in a while, and that fuel bundles remain in a certain location in the channel until the channel is refuelled. It models the effect of the axial refuelling scheme (e.g., 8-bundle-shift, 4-bundle-shift, etc.) used in each channel on the average lattice property at each bundle location in the channel. For any fuel bundle position $\{c, b\}$ in core and any cross section of type x , we can write the time-average macroscopic cross section $\bar{\Sigma}_{x,c,b}$ during the bundle's residence time at that location using

$$\bar{\Sigma}_{x,c,b} = \frac{1}{\omega_{\text{out},c,b} - \omega_{\text{in},c,b}} \int_{\omega_{\text{in},c,b}}^{\omega_{\text{out},c,b}} d\omega \Sigma_{x,c,b}(\omega)$$

Table 6
Eigenvalue spectrum of the VVER-1000 3D benchmark.

mode	Hotelling	Krylov–Schur	ABHA
1	1.006601	1.006601	1.0066001
2	0.988090	$0.988091 + 9.966 \times 10^{-7}i$	$0.988090 + 1.041 \times 10^{-6}i$
3	(failure)	$0.988091 - 9.966 \times 10^{-7}i$	$0.988090 - 1.041 \times 10^{-6}i$
4	0.967791	0.967791	0.967791
5	0.964047	$0.964048 + 2.237 \times 10^{-6}i$	$0.964047 + 2.275 \times 10^{-6}i$
6	(failure)	$0.964048 - 2.237 \times 10^{-6}i$	$0.964047 - 2.275 \times 10^{-6}i$
7	0.951447	$0.951447 + 9.170 \times 10^{-7}i$	$0.951447 + 9.308 \times 10^{-7}i$
8	(failure)	$0.951447 - 9.170 \times 10^{-7}i$	$0.951447 - 9.308 \times 10^{-7}i$
9	0.948055	0.948056	0.948055
10	0.945470	0.945470	0.945470
11	(failure)	$0.932268 + 4.673 \times 10^{-7}i$	$0.932268 + 4.435 \times 10^{-7}i$
12	(failure)	$0.932268 - 4.673 \times 10^{-7}i$	$0.932268 - 4.435 \times 10^{-7}i$

where $\omega_{in,c,b}$ and $\omega_{out,c,b}$ are the entry and exit burnups of fuel bundles in fuel channel c and bundle position b (with $1 \leq b \leq 12$). The burnup-dependent macroscopic cross sections are recovered from the multi-parameter reactor database generated by the lattice code (Hébert, 2016).

The 3D Cartesian geometry was discretized using mesh-centered finite-differences (MCFD), as presented in Section 5.2.2 of Hébert (2016). Again, the symmetric variational acceleration method (SVAT) used by the Hotelling technique was deactivated so as to help the deflation process, at the cost of increasing number of iterations. The eigenvalue spectrum obtained using the three approaches is presented in Table 8. The CPU time required for

Table 7
Eigenvalue spectrum of the VV1K3D benchmark.

mode	Hotelling	Krylov–Schur	ABHA
1	1.005450	1.005450	1.005450
2	0.987368	0.987368	0.987369
3	0.987360	0.987360	0.987360
4	0.968519	0.968519	0.968521
5	0.964399	0.964399	0.964399
6	0.963050	0.963050	0.963050
7	0.954741	0.954743	0.954745
8	0.948942	0.948942	0.948943
9	0.948283	0.948283	0.948283
10	0.946417	0.946414	0.946414
11	0.934308	0.934309	0.934310
12	0.930107	0.930121	0.930122

Table 8
Eigenvalue spectrum of the time-averaged CANDU6 case.

mode	Hotelling	Krylov–Schur	ABHA
1	1.000000	1.000000	1.000000
2	0.986041	0.986041	0.986041
3	0.985432	0.985432	0.985432
4	0.976302	0.976303	0.976303
5	0.961844	0.961843	0.961843
6	0.961471	0.961476	0.961476
7	0.959934	0.959944	0.959944
8	0.958942	0.958932	0.958932
9	0.944784	0.944783	0.944783
10	0.934379	0.934380	0.934380
11	0.934255	0.934256	0.934256
12	0.931808	0.931812	0.931812
13	(failure)	0.931478	0.931478

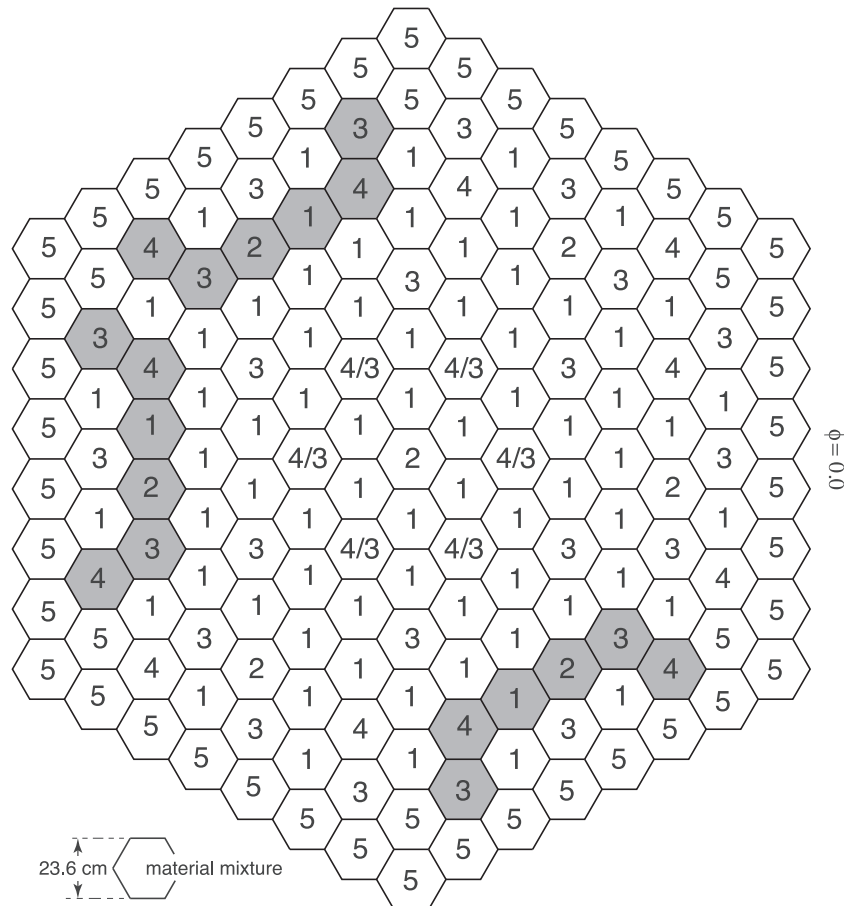


Fig. 4. Description of the VV1K3D benchmark.

obtaining the modes is 639 s for the Hotelling technique, 2.8 s for the Krylov–Schur method and 6.7 s for the ABHA method.

4. Conclusion

Application of the Krylov–Schur and ABHA methods in TRIVAC5, as an alternative to the legacy Hotelling deflation technique has proven more effective in terms of computer resources and more reliable in terms of stability. Both IRAM methods are equally stable end equivalent from a computational point of view. The Krylov–Schur implementation of Bernal et al. (2017) was based on packages SLEPc, PETSc, ARPACK, LAPACK and BLAS, representing over a million lines of code, a mixture of Fortran code written in the seventies with modern C++ code. The proposed ABHA implementation is only based on two new subroutines linked with the UTILIB5 library, containing $\approx 10,000$ lines of code. The UTILIB5 library contains the linear algebra tools already used in the codes TRIVAC5, DRAGON5 and DONJON5. Our custom implementation of the ABHA method is adding a limited increase in configuration complexity and has limited effect on ease of maintenance, quality assurance (QA) and licensing issues. This new capability is now included in the Open Source distribution of DONJON5, version 5.0.7.

CRediT authorship contribution statement

Alain Hébert: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by a grant from the Natural Science and Engineering Research Council of Canada.

References

- Anderson, E. et al., 1999. LAPACK User's Guide, Third Edition, SIAM, Philadelphia. See the home page at <http://www.netlib.org/lapack/>.
- Argonne Code Center, 1977. "Benchmark Problem Book, ANL-7416, Supp. 2, ID11-A2, Argonne National Laboratory.
- Baglama, J., 2008. Augmented block Householder Arnoldi method. *Linear Algebra Appl.* 429 (10), 2315–2334.
- Balay, S. et al., 2020. "PETSc Users Manual," Argonne National Laboratory, Report ANL-95/11 – Revision 3.13. See the home page at <https://www.mcs.anl.gov/petsc>.
- 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 Trans. Math. Software (TOMS)*, 36(3) Article 13.
- 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. *J. Nucl. Sci. Technol.* 54 (10), 1085–1094.
- Chao, Y.A., Shatilla, Y.A., 1995. Conformal Mapping and Hexagonal Nodal Methods – II: Implementation of the ANC-H Code. *Nucl. Sci. Eng.* 121, 210–225.
- Gold, M., Wight, A., 1990. SMOKIN – A Family of Codes for Reactor Space-Time Neutronics Calculations Based on Modal Kinetics – Theory Manual, Report No. 90133, Nuclear Safety Department, Ontario Hydro.
- Hébert, A., 1986. Preconditioning the power method for reactor calculations. *Nucl. Sci. Eng.* 94, 1–11.
- Hébert, A., 1987. TRIVAC, a modular diffusion code for fuel management and design applications. *Nucl. J. Canada* 1 (4), 325–331.
- Hébert, A., 1987. Development of the nodal collocation method for solving the neutron diffusion equation. *Ann. Nucl. Energy* 14, 527.
- Hébert, A., 1993. Application of a dual variational formulation to finite element reactor calculations. *Ann. Nucl. Energy* 20, 823–845.
- Hébert, A., 2008. A Raviart–Thomas–Schneider solution of the diffusion equation in hexagonal geometry. *Ann. Nucl. Energy* 35, 363–376.
- Hébert, A., Sekki, D., Chambon, R., 2014. "A user guide for DONJON version5, Polytechnique Montréal, Report IGE-344. See the home page at <http://www.polymtl.ca/merlin/>.
- Hébert, A., 2016. Applied Reactor Physics, Second Edition, Presses Internationales Polytechnique, ISBN 978-2-553-01698-1, 396 p., Montréal, Canada.
- Hébert, A., 2016. "DRAGON5 and DONJON5, the contribution of École Polytechnique de Montréal to the SALOME platform, *Ann. Nucl. Energy*, 87, 12–20.
- Hébert, A., 2020. A user guide for TRIVAC version5, Polytechnique Montréal, Report IGE-369. See the home page at <http://www.polymtl.ca/merlin/>.
- Hernandez, V., Roman, J.E., Vidal, V., 2005. SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems, *ACM Trans. Math. Softw.*, 31(3) 351–362. See the home page at <https://slepc.upv.es>.
- Hoareau, F., Laugier, F., Couyras, D., 2008. Elements of Validation of microscopic depletion for the Future EDF Calculation Scheme Based on APOLLO2 and COCAGNE Codes, *Proc. of ICAPP '08, Anaheim, CA USA, June 8–12*.
- Kugler, G., 1976. An Iterative Procedure for Calculating the Higher Harmonics of the Diffusion Equation – the MONIC Code, Technical Report, Atomic Energy of Canada Limited.
- Lehoucq, R.B., Sorensen D.C., Yang, C., 1998. ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, Society for Industrial and Applied Mathematics. See the home page at <https://www.caam.rice.edu/software/ARPACK/>.
- Levine, M.M., Diamond, D.J., 1972. Reactor power distribution from analysis of in-core detector readings. *Nucl. Sci. Eng.* 47, 415–420.
- March-Leuba, J., Blakeman, E.D., 1991. A mechanism for out-of-phase power instabilities in boiling water reactors. *Nucl. Sci. Eng.* 107, 173–179.
- red Reuss, P., 2008. Neutron Physics, EDP Sciences, ISBN 978-2-7598-00041-4, 669 p., Les Ulis, France.
- Robles, G.E., 2017. Implementing the QR algorithm for efficiently computing matrix eigenvalues and eigenvectors, Final Degree Dissertation in Mathematics, Universidad del País Vasco, Spain.
- Rouben, B., 2007. Review of the CANDU Time-Average Model and Calculations, in: *Proceedings of the 28th Annual Conference of the Canadian Nuclear Society*, Saint John, New Brunswick, Canada, June 3–6.
- Sorensen, D.C., 1992. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.* 13, 357–385.
- Stewart, G.W., 2001. A Krylov–Schur algorithm for large eigenproblems. *SIAM J. Matrix Anal. Appl.* 23, 601–614.
- Virtanen, P. et al., 2020. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods, in press.
- Xia, L., 2012. Development of 3-D Neutronic Kinetic Model and Control for CANDU Reactors (Ph.D. thesis), The University of Western Ontario. See the home page at <https://ir.lib.uwo.ca/etd/898>.