

# Numerical solution of two-energy-group neutron noise diffusion problems with fine spatial meshes

Antonios G. Mylonakis, Paolo Vinai, Christophe Demaziere

*Chalmers University of Technology,  
Department of Physics  
Division of Subatomic and Plasma Physics  
SE-412 96 Gothenburg, Sweden*

---

## Abstract

The paper presents the development of a strategy for the fine-mesh full-core computation of neutron noise in nuclear reactors. Reactor neutron noise is related to fluctuations of the neutron flux induced by stationary perturbations of the properties of the system. Its monitoring and analysis can provide useful insights in the reactor operations. The model used in the work relies on the neutron diffusion approximation and requires the solution of both the criticality (eigenvalue) and neutron noise equations. A high-resolution spatial discretization of the equations is important for an accurate evaluation of the neutron noise because of the strong gradients that may arise from the perturbations. Considering the size of a nuclear reactor, the application of a fine mesh generates large systems of equations which can be challenging to solve. Then, numerical methods that can provide efficient solutions for these kinds of problems using a reasonable computational effort, are investigated. In particular the power method accelerated with the Chebychev or JFNK-based techniques for the eigenvalue problem, and GMRES with the Symmetric Gauss-Seidel, ILU, SPAI preconditioners for the solution of linear systems, are evaluated with the computation of neutron noise in the case of localized perturbations in 1-D and 2-D simplified reactor cores and in a 3D realistic reactor core.

*Keywords:* reactor neutron noise, k-eigenvalue problem, GMRES, nonlinear acceleration, JFNK, SPAI preconditioner

---

## 1. Introduction

Neutron noise in power nuclear reactors consists of fluctuations of the neutron flux. These fluctuations are small in comparison with the main trend of the neutron flux and they are induced by perturbations such as vibrations of reactor components, oscillations of coolant temperature or density, etc. The perturbations which act as neutron noise sources can be problematic for the operation of the nuclear reactors and their effects need to be monitored and analyzed. For this purpose one should determine the transfer function of the reactor which describes the system response to any possible perturbation. The reactor transfer function can also be used for diagnostic purposes, i.e. to identify the neutron noise source knowing the effect on the measured neutron flux (Williams (1974), Thie (1981), Pázsit and Demaziere (2010)). Consequently, the computation of the transfer function of nuclear reactors is of primary importance.

As for the simulations of the static flux in a nuclear reactor, computational capabilities for reactor neutron noise rely on the neutron transport equation. The solution of this equation is challenging for these kinds of problems because a nuclear reactor core is a very heterogeneous and large system. Then a typical, simplified approach for the full-reactor core computation of neutron noise is the diffusion approximation.

Most existing neutron diffusion-based simulators for neutron noise applications use finite-difference methods for the spatial discretization of the equations, direct methods for the solution procedure, and relatively coarse representations of the physical system. This approach can nevertheless be inadequate for neutron noise simulations. In fact neutron noise sources are often highly localized in space and induce in their vicinity strong gradients of the neutron flux. For an accurate prediction of this behavior, a very fine spatial discretization is therefore required.

The application of fine computational meshes leads to large systems of algebraic equations, whose solution with direct methods is questionable or even prohibitive in terms of computer memory and running time.

The objective of the work presented in this paper, is to investigate a strategy that can be used for the solution of the diffusion-based neutron noise equations over fine meshes with a feasible computational cost. Then, different numerical methods have been implemented and evaluated over three neutron noise problems based on a 1-D and a 2-D homogeneous reactor, and a 3-D realistic reactor.

The structure of the paper is as follows. Section 2 presents the governing equations of the problem. Section 3 discusses the challenges associated to the numerical solution of the neutron noise equations. Section 4 deals with the numerical methods studied in the current work. Section 5 presents the results for three test cases. In Section 6, some conclusions are drawn.

## 2. Governing equations

The two-energy group time-dependent neutron diffusion equations along with the equation for one family of delayed neutron precursors read as:

$$\begin{aligned} \frac{1}{\nu_1} \frac{\partial}{\partial t} \phi_1(\mathbf{r}, t) = \nabla \cdot [D_1(\mathbf{r}) \nabla \phi_1(\mathbf{r}, t)] + [(1 - \beta)v\Sigma_{f,1}(\mathbf{r}, t) - \Sigma_{\alpha,1}(\mathbf{r}, t) - \Sigma_r(\mathbf{r}, t)]\phi_1(\mathbf{r}, t) \\ + (1 - \beta)v\Sigma_{f,2}(\mathbf{r}, t)\phi_2(\mathbf{r}, t) + \lambda C(\mathbf{r}, t) \end{aligned} \quad (1)$$

$$\frac{1}{\nu_2} \frac{\partial}{\partial t} \phi_2(\mathbf{r}, t) = \nabla \cdot [D_2(\mathbf{r}) \nabla \phi_2(\mathbf{r}, t)] + \Sigma_r(\mathbf{r}, t)\phi_1(\mathbf{r}, t) - \Sigma_{\alpha,2}(\mathbf{r}, t)\phi_2(\mathbf{r}, t) \quad (2)$$

$$\frac{\partial}{\partial t} C(\mathbf{r}, t) = \beta v\Sigma_{f,1}(\mathbf{r}, t)\phi_1(\mathbf{r}, t) + \beta v\Sigma_{f,2}(\mathbf{r}, t)\phi_2(\mathbf{r}, t) - \lambda C(\mathbf{r}, t) \quad (3)$$

where the macroscopic removal cross-section is defined as:

$$\Sigma_r(\mathbf{r}, t) = \Sigma_{s,1 \rightarrow 2}(\mathbf{r}, t) - \frac{\Sigma_{s,2 \rightarrow 1}(\mathbf{r}, t)\phi_2(\mathbf{r}, t)}{\phi_1(\mathbf{r}, t)} \quad (4)$$

Indices 1 and 2 denote the fast and the thermal energy group. Eqs (1 & 2) are the balance equations of fast and thermal neutrons, respectively. Eq. (3) is the neutron precursor balance equation.  $\Sigma_r(\mathbf{r}, t)$  describes at once the net neutron flow from the fast to the thermal energy group taking into account the neutron down- and upper-scattering. In Eqs (1 & 2), the neutron diffusion coefficients ( $D$ ) are considered as time-independent. When solving for a steady-state, critical reactor, there exists an equilibrium between the production and the decay of the precursors ( $\frac{\partial}{\partial t} C(\mathbf{r}, t) = 0$ ). Then the above system of equations takes the form:

$$\nabla \cdot [D_1(\mathbf{r}) \nabla \phi_1(\mathbf{r})] - \Sigma_{\alpha,1}(\mathbf{r})\phi_1(\mathbf{r}) - \Sigma_r(\mathbf{r})\phi_1(\mathbf{r}) = -v\Sigma_{f,2}(\mathbf{r})\phi_2(\mathbf{r}) - v\Sigma_{f,1}(\mathbf{r})\phi_1(\mathbf{r}) \quad (5)$$

$$\nabla \cdot [D_2(\mathbf{r}) \nabla \phi_2(\mathbf{r}, t)] + \Sigma_r(\mathbf{r}, t)\phi_1(\mathbf{r}, t) - \Sigma_{\alpha,2}(\mathbf{r}, t)\phi_2(\mathbf{r}, t) = 0 \quad (6)$$

If the system is not critical, a steady-state solution can still be obtained by re-normalizing the fission source with a factor  $k$ . The matrix form of Eqs (5 & 6) when a  $k$  normalization factor is used, is written as follows:

$$\left\{ \nabla \cdot \begin{bmatrix} D_1(\mathbf{r}) & 0 \\ 0 & D_2(\mathbf{r}) \end{bmatrix} \nabla + \begin{bmatrix} -\Sigma_{\alpha,1}(\mathbf{r}) - \Sigma_r(\mathbf{r}) & 0 \\ \Sigma_r(\mathbf{r}) & -\Sigma_{\alpha,2}(\mathbf{r}) \end{bmatrix} \right\} \times \begin{bmatrix} \phi_1(\mathbf{r}) \\ \phi_2(\mathbf{r}) \end{bmatrix} = \quad (7)$$

$$\frac{1}{k} \begin{bmatrix} -v\Sigma_{f,1}(\mathbf{r}) & -v\Sigma_{f,2}(\mathbf{r}) \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} \phi_1(\mathbf{r}) \\ \phi_2(\mathbf{r}) \end{bmatrix} \quad (8)$$

To obtain the Neutron Noise Diffusion Equations (NNDEs) all the time-dependent terms of Eqs (1 - 3), generically expressed as  $X(\mathbf{r}, t)$ , are split into their mean values  $X_0(\mathbf{r})$  (corresponding to the steady-state configuration of the system) and their fluctuations  $\delta X(\mathbf{r}, t)$  around their mean values so that they are written as:

$$X(\mathbf{r}, t) = X_0(\mathbf{r}) + \delta X(\mathbf{r}, t) \quad (9)$$

Considering small fluctuations compared to the mean values and stationary processes, and neglecting the second order terms (linear theory), the two-energy-group NNDEs in the frequency domain are obtained after a temporal Fourier transform:

$$\left\{ \nabla \cdot \begin{bmatrix} D_1(\mathbf{r}) & 0 \\ 0 & D_2(\mathbf{r}) \end{bmatrix} \nabla + \Sigma_{dyn}^{crit}(\mathbf{r}, \omega) \right\} \times \begin{bmatrix} \delta\phi_1(\mathbf{r}, \omega) \\ \delta\phi_2(\mathbf{r}, \omega) \end{bmatrix} = \phi_r(\mathbf{r})\delta\Sigma_r(\mathbf{r}, \omega) + \phi_a(\mathbf{r}) \begin{bmatrix} \delta\Sigma_{a,1}(\mathbf{r}, \omega) \\ \delta\Sigma_{a,2}(\mathbf{r}, \omega) \end{bmatrix} + \phi_f^{crit}(\mathbf{r}, \omega) \begin{bmatrix} \delta\nu\Sigma_{f,1}(\mathbf{r}, \omega) \\ \delta\nu\Sigma_{f,2}(\mathbf{r}, \omega) \end{bmatrix} \quad (10)$$

where the complex components  $\Sigma_{dyn}^{crit}$  and  $\phi_f^{crit}$  as well as the real ones  $\phi_r$  and  $\phi_a$  are defined in Appendix A. As the RHS factors  $\phi_f^{crit}$ ,  $\phi_r$  and  $\phi_a$  imply, the solution of the neutron noise problem requires the computation of the steady-state neutron fluxes from Eq. (8).

The spatial discretization is based on the approach used by (Demaziere, 2011a). Accordingly, the nuclear reactor core is described with  $N$  discrete volumes (nodes) over which the governing equations are spatially averaged. The diffusion operator is discretized with a finite-difference scheme together with a suitable relationship between the surface averaged neutron currents and the node-averaged scalar neutron fluxes. The discretization scheme is summarized in Appendix B.

### 3. Challenges of solving fine-mesh neutron noise problems

The simulation of neutron noise with the model introduced in Section 2 consists of two steps: first, the solution of Eq. (8) and then the solution of Eq. (10). After the spatial discretization, Eq. (8) can be summarized into the matrix form:

$$\mathbf{A}_{crit} \Phi_{crit} = \frac{1}{k} \mathbf{F} \Phi_{crit} \quad (11)$$

or after rearrangement:

$$[\mathbf{A}_{crit}^{-1} \mathbf{F}] \Phi_{crit} = k \Phi_{crit} \quad (12)$$

where  $\mathbf{A}_{crit}$  and  $\mathbf{F}$  are real matrices. The normalization factor  $k$  and the vector  $\Phi_{crit}$  represent the eigenvalues and the eigenvectors of matrix  $[\mathbf{A}_{crit}^{-1} \mathbf{F}]$ , respectively. Only the largest  $k$  and its associated eigenvector are of interest because they correspond to the effective multiplication factor ( $k_{eff}$ ) and to the static neutron flux (i.e. the only eigenvector having the same sign throughout the entire system). The standard algorithm for solving the eigenvalue problem is the Power iteration or Power Method (PM) (Lewis and Miller (1984)) and it is presented in Alg. (1).

---

#### Algorithm 1 Power Method

---

```

 $\Phi_{crit}^{(1)}, k^{(1)}$  given
for  $l \geq 1$ 
 $\Phi_{crit}^{(l+1)} = \frac{1}{k^{(l)}} \mathbf{A}_{crit}^{-1} \mathbf{F} \Phi_{crit}^{(l)}$ 
 $k^{(l+1)} = \frac{(\mathbf{F} \Phi_{crit}^{(l)}, \mathbf{F} \Phi_{crit}^{(l)})}{(\mathbf{A}_{crit} \Phi_{crit}^{(l)}, \mathbf{F} \Phi_{crit}^{(l)})}$ 
end

```

---

The main advantage of PM is that it guarantees the convergence to the largest eigenvalue and its associated eigenvector (“dominant” or “fundamental” eigenpair) and at the same time is very simple. For the generic iteration  $l$ , the vector of unknowns  $\Phi_{crit}^{(l+1)}$  is obtained from the solution of the linear system  $\mathbf{A}_{crit} \Phi_{crit}^{(l+1)} = \frac{1}{k^{(l)}} \mathbf{F} \Phi_{crit}^{(l)}$ , where  $k^{(l)}$  and  $\Phi_{crit}^{(l)}$  come from the previous iteration. Thus PM gives rise to a sequence of linear systems.

Once the static neutron fluxes for the critical reactor configuration are known, Eq. (10) can be solved. The discretized form of the latter equation can be written as:

$$\mathbf{A}_{noise} \Phi_{noise} = \mathbf{S}_{noise} \quad (13)$$

where  $\mathbf{A}_{noise}$  is the complex coefficient matrix derived from the LHS of Eq. (10) and the vector  $\mathbf{S}_{noise}$  is the RHS of Eq. (10).

The numerical efficiency of PM depends on two factors: its convergence speed and the computational effort required from solving the linear system at each iteration. The convergence speed is related to the spectrum of the iterative matrix  $\mathbf{A}_{crit}^{-1}\mathbf{F}$ . More specifically, it is determined by the Dominance Ratio (DR) of the iterative matrix which is the ratio between the second largest and the largest (dominant) eigenvalue ( $k_1/k_0$ ); when  $DR \approx 1$ , PM converges slowly. In nuclear reactor applications, it often happens that DR is greater than 0.95. Under this condition, PM might be too slow making the analysis of full-core configurations discretized with fine meshes even impossible.

For both the criticality and noise problems, the applied discretization algorithm generates sparse systems that in case of 3-D configurations with fine meshes are also large. The solution of these linear systems with direct methods may require a burdensome computational effort because of different reasons. First, the inversion and decomposition of the coefficient matrix generates less sparse matrices (the so-called fill-in phenomenon) imposing high storage requirements. Second, direct methods usually scale poorly with the size of the system in terms of operation counts (Benzi (2002)) increasing the computational time. Besides, the construction of the large coefficient matrices may be time-consuming and thus needs special attention.

To summarize, this work aims to solve fine-mesh 3-D full-core neutron noise diffusion problems with a reasonable computational cost. This induces the following challenges:

- Fast construction of the coefficient matrices.
- Acceleration of PM for the eigenvalue calculation.
- Efficient solution of the linear systems.

In this context, a reasonable computational cost is such that the simulations can be performed in no more than a few hours with no use of parallel computing and no more than 10 GB of RAM.

#### 4. Strategy for solving fine-mesh neutron noise problems

In this section the strategy and methods used to solve numerically the overall neutron noise problem over a fine mesh are introduced. The coefficient matrices and their construction are first discussed. Two methods are considered in order to accelerate PM when applied to Eq. (11), namely the Chebyshev method and a method based on the Jacobian Free Newton Krylov technique. The solution of the linear systems within PM as well as of Eq. (13) for determining the neutron noise, are obtained from the application of GMRES. For the improvement of the performance of the scheme, three preconditioners are combined with GMRES, i.e. Symmetric Gauss-Seidel (SGS), ILU and SPAI.

##### 4.1. Construction of the matrices

The matrices  $\mathbf{A}_{crit}$  and  $\mathbf{A}_{noise}$  that appear in Eqs (11 & 13) are large sparse banded matrices of dimension  $2N \times 2N$  where  $N$  is the number of spatial nodes and 2 is the number of the energy groups. For the matrix construction the so-called ‘‘coordinate format’’ is used. This format stores the diagonals in three vector-arrays of known length: one containing the values of the nonzero entries, one integer array containing their row indices and another integer array containing their column indices. When all diagonals have been computed, the matrix is assembled and stored in a compressed sparse matrix format. Following this strategy, one operates on vector-arrays of known length during the construction phase, avoiding any dynamic allocation of a compressed sparse matrix that would reduce the speed of the process. The matrix  $\mathbf{F}$  consists strictly of two diagonals. Thus, the storage of the coordinates of its elements is not required. The structure of the matrices is reported in Table (1).

Table 1: Structure of the problem matrices  $A_{crit}$ ,  $A_{noise}$  and  $F$ .

Diagonal	Value	Model dimens.	Comment
$A_{crit,(i=j)}$	$-\left(\sum_{\mathbb{N}=x,y,z} a_{g,n}^{\mathbb{N}}\right) - \Sigma_{\alpha,1,n} - \Sigma_{r,n}$ for $i \leq N$ $-\left(\sum_{\mathbb{N}=x,y,z} a_{g,n}^{\mathbb{N}}\right) - \Sigma_{\alpha,2,n}$ for $i > N$	1,2,3-D	main diagonal
$A_{crit,(j=i+1)}$	$-b_{g,n}^x$	1,2,3-D	upper
$A_{crit,(j=i-1)}$	$-c_{g,n}^x$	1,2,3-D	lower
$A_{crit,(j=i-N)}$	$\Sigma_{r,n}$	1,2,3-D	upper
$A_{crit,(j=i+N_y)}$	$-b_{g,n}^y$	2,3-D	upper, bunch of diag.
$A_{crit,(j=i-N_y)}$	$-c_{g,n}^y$	2,3-D	lower, bunch of diag.
$A_{crit,(j=i+N_z)}$	$-b_{g,n}^z$	3-D	upper
$A_{crit,(j=i-N_z)}$	$-c_{g,n}^z$	3-D	lower
$A_{noise,(i=j)}$	$-\left(\sum_{\mathbb{N}=x,y,z} a_{g,n}^{\mathbb{N}}\right) - \Sigma_{1,n}^{crit}(\omega)$ for $i \leq N$ $-\left(\sum_{\mathbb{N}=x,y,z} a_{g,n}^{\mathbb{N}}\right) - \left(\Sigma_{a,2,n} + \frac{i\omega}{v_2}\right)$ for $i > N$	1,2,3-D	main
$A_{noise,(j=i+N)}$	$\frac{v\Sigma_{f,2,n}}{k_{eff}} \left(1 - \frac{i\omega\beta}{i\omega + \lambda}\right)$	1,2,3-D	upper
$A_{noise,(j=i+1)}$	$-b_{g,n}^x$	1,2,3-D	upper
$A_{noise,(j=i-1)}$	$-c_{g,n}^x$	1,2,3-D	lower
$A_{noise,(j=i+N_y)}$	$-b_{g,n}^y$	2,3-D	upper, bunch of diag.
$A_{noise,(j=i-N_y)}$	$-c_{g,n}^y$	2,3-D	lower, bunch of diag.
$A_{noise,(j=i+N_z)}$	$-b_{g,n}^z$	3-D	upper
$A_{noise,(j=i-N_z)}$	$-c_{g,n}^z$	3-D	lower
$F_{(i=j)}$	$-v\Sigma_{f,1,n}$ for $i \leq N$ $0$ for $i > N$	1,2,3-D	main
$F_{(j=i+N)}$	$-v\Sigma_{f,2,n}$	1,2,3-D	upper

## 4.2. Solution of the large eigenvalue problem

Once the matrices are built, the next step is to solve Eq. (11). As mentioned in Section 3, PM is applied. The method is known to often converge slowly, so the implementation of an acceleration technique is important. In nuclear reactor criticality calculations, PM is often accelerated with the technique of the Chebyshev polynomials. In this work the Chebyshev method is compared with an alternative, i.e. the method of nonlinear acceleration based on the Jacobian Free Newton Krylov (JFNK) algorithm. The latter being a generic Newton-based method, allows to obtain quadratic convergence rates. Exploring the performance of this option is of interest since JFNK-based algorithms have attractive features (see discussion in Section 4.2.2) and their implementation for reactor physics problems is an active research topic. Various JFNK-based algorithms for reactor criticality analysis have been proposed. The version used has been selected because of its easy adaptation to PM and the satisfactory performances reported in literature.

### 4.2.1. Chebyshev acceleration of the PM solver

The utilization of Chebyshev polynomials is a traditional method for accelerating PM (Saad (2011)) and it was first proposed for reactor criticality problems by Ferguson and Derstine (1977). The method of Chebyshev acceleration updates the neutron flux estimate of a PM iteration as a linear combination of the solutions of some previous iterations. The coefficients of the linear combination are determined from the minimization of the estimation error with the Chebyshev polynomials. Various alternatives of the method are available; the one implemented in this work is described by Hebert (1985). After some initial free PM iterations a sequence of Chebyshev acceleration cycles starts. Within each cycle, the neutron flux estimates are extrapolated from the previous estimates using the calculated coefficients. The method also requires DR estimates that are updated at the end of each acceleration cycle. A simplified presentation of the algorithm is given in Alg. (2). For completeness, a detailed description of the Chebyshev acceleration algorithm used in this work is presented in Appendix C. Hereinafter, Chebyshev accelerated PM will be referred as PM-Cheb.

---

**Algorithm 2** PM-Cheb: Simplified description

---

```
for  $l \geq 1$  (PM iteration)
  PM updates flux
  PM updates  $k$ 
  ...
  if free PM iteration
    Update DR estimate (Eq. (C.1))
  elseif Cheb acceleration cycle is on
    Cheb calculates the extrapolation factors (Eqs (C.3 - C.6))
    Cheb updates flux extrapolating flux estimates from previous iterations (Eq. (C.2))
    if last iteration of the current Cheb acceleration cycle
      Update DR estimate (Eq. (C.11))
    end
  end
end
...
end
```

---

#### 4.2.2. Nonlinear acceleration of the PM solver

A second option is tested for the acceleration of PM. It is based on a JFNK approach and follows the work of Gill and Asmy (2009). The method allows PM to be easily encapsulated into a Newton scheme without the need for any modifications. In addition, Gill and Asmy (2009) showed that it can provide higher performance in multigroup neutron diffusion problems, than other Newton-based approaches.

The basis of the method is that PM may be seen as a fixed point iteration of the form:

$$\mathbf{u}^{(l+1)} = f(\mathbf{u}^{(l)}) \quad (14)$$

The vector of the unknowns  $\mathbf{u}$  contains the flux vector and the eigenvalue:

$$\mathbf{u} = \begin{bmatrix} \Phi_{crit} \\ k \end{bmatrix} \quad (15)$$

While the ordinary PM updates the flux vector  $\Phi_{crit}$  and the eigenvalue separately, in two steps, Eq. (14) considers both components as inseparable parts of the solution vector  $\mathbf{u}$ . The fixed point iteration  $f$  is denoted as:

$$f(\mathbf{u}) = \begin{bmatrix} \frac{1}{k} \mathbf{A}^{-1} \mathbf{F} \Phi_{crit} \\ \bar{k}(\mathbf{u}) \end{bmatrix} \quad (16)$$

where  $\bar{k}(\mathbf{u})$  represents the updating procedure for  $k$ . A non-linear system is defined by setting:

$$\mathbf{r}(\mathbf{u}) \equiv \mathbf{u} - f(\mathbf{u}) = \mathbf{0} \quad (17)$$

where  $\mathbf{r}$  is the vector of the nonlinear residuals. A linearized form of Eq. (17) is obtained using the Newton method:

$$\mathbf{J}(\mathbf{u}^{(l)}) \delta \mathbf{u}^{(l)} = -\mathbf{r}(\mathbf{u}^{(l)}) \quad (18)$$

$$\mathbf{u}^{(l+1)} = \mathbf{u}^{(l)} + \delta \mathbf{u}^{(l)}, \text{ for } l = 1, 2, \dots \quad (19)$$

where  $\mathbf{J}(\mathbf{u}) = \frac{\partial \mathbf{r}(\mathbf{u})}{\partial \mathbf{u}}$  is the Jacobian matrix, and  $\delta \mathbf{u}$  is the Newton correction vector. The Newton iteration algorithm is shown in Alg. (3).

A JFNK method is used to avoid the high computational cost due to the construction of the Jacobian matrix  $\mathbf{J}$ , and is summarized in Alg. (4). The matrix-vector products  $\mathbf{J}\mathbf{v}$  required in the Krylov method, are approximated as follows:

$$\mathbf{J}\mathbf{v} = \frac{\mathbf{r}(\mathbf{u} + \varepsilon\mathbf{v}) - \mathbf{r}(\mathbf{u})}{\varepsilon} \quad (20)$$

where  $\mathbf{v}$  is the Krylov vector and  $\varepsilon$  is a small perturbation parameter. Many options exist for the calculation of  $\varepsilon$ . In this work the following formula (Knoll and Keys, 2004) is used:

$$\varepsilon = \frac{1}{(2N + 1) \|\mathbf{v}\|_2} \sum_{i=1}^{2N+1} d|u_i| + d \quad (21)$$

where  $d$  is the square root of machine roundoff,  $(2N + 1)$  is the dimension of the linear system.

In the current scheme, an inexact JFNK variation is applied. The convergence of the linear system within each JFNK iteration  $l$  is monitored with the following criterion:

$$\|\mathbf{J}(\mathbf{u}^{(l)})\delta\mathbf{u}^{(l)} + \mathbf{r}(\mathbf{u}^{(l)})\| \leq \gamma^{(l)} \|\mathbf{r}(\mathbf{u}^{(l)})\| \quad (22)$$

where the residual norm is reduced by the factor  $\gamma$ . The procedure is inexact because Eq. (18) is not solved tightly using a very small  $\gamma$ . Instead, it starts with larger values which are progressively reduced. This strategy aims to avoid the phenomenon of ‘‘oversolving’’. This practically implies that a very accurate Krylov solution when the Newton one is far from convergence can cause negative effects to the convergence of the global solution. Discussion about this matter can be found in Shadid et al. (1997) and Tuminaro et al. (2002). In this work,  $\gamma$  is experimentally selected to be constant and equal to  $10^{-1}$ . However, more sophisticated ways able to vary dynamically  $\gamma$ , exist (Eisenstat and Walker (1996)).

Gill and Asmy (2009) reported that PM-JFNK converges to an eigenpair other than the dominant one when an initial guess close to that eigenpair is provided. In order to avoid this issue, a few free PM iterations are performed before starting PM-JFNK.

As discussed above, the implementation of this acceleration scheme only needs a JFNK solver in which the non-linear function is evaluated by the single PM iterations. Another advantage is that the preconditioner for the original PM can still be used within the evaluations of the non-linear function. For the solution of the linear system of Eq. (18), no preconditioning is included. The development of a preconditioner for this problem is not trivial and it is a subject for future research. In fact  $\mathbf{J}$  is a function of  $\mathbf{u}$  and changes with the Newton iterations. In addition,  $\mathbf{J}$  is not explicitly constructed in a Jacobian-free approach, so it is not available. Hereafter, PM accelerated with JFNK is referred to as PM-JFNK.

#### 4.2.3. Outer/inner iteration

The PM and PM-Cheb solvers consist of one level of outer/inner iterations. The term *outer* refers to the iterative process of updating  $\Phi_{crit}$  and  $k$  and the term *inner* refers to the iterations for the solution of the linear system.

PM-JFNK consists of two levels of outer/inner iterations. The first level consists of the Newton (outer) iterations needed for the convergence of the solution of the non-linear system, and the Krylov (inner) iterations for the solution of the linear system of Eq. (18) (see (Alg. (4))). The second level consists of outer iterations used for updating  $\Phi_{crit}$  and  $k$  within the non-linear function evaluations  $f$ , and inner ones for solving the linear systems within  $f$ .

Hereinafter, the terms *outer/inner* iterations are used for PM, PM-Cheb and the second level of PM-JFNK, and the terms *Newton/Krylov* correspond to the first level of PM-JFNK.

#### 4.2.4. Convergence criteria

The following convergence criteria are employed for PM (Duderstadt and Hamilton (1976), Hebert (1986), Gill and Asmy (2009)):

$$|k^{(l+1)} - k^{(l)}| < \epsilon_1 \quad (23)$$

---

**Algorithm 3** PM-JFNK: Newton iteration

---

$\mathbf{u}^{(1)}$  given  
for  $l \geq 1$  (Newton iteration)  
   $\mathbf{r}^{(l)} = \mathbf{u}^{(l)} - f(\mathbf{u}^{(l)})$  for  $l \geq 1$   
   $\delta \mathbf{u}^{(l)} = -\mathbf{J}^{-1}(\mathbf{u}^{(l)}) \cdot \mathbf{r}^{(l)}$  (Krylov solver)  
   $\mathbf{u}^{(l+1)} = \mathbf{u}^{(l)} + \delta \mathbf{u}^{(l)}$   
end

---

---

**Algorithm 4** PM-JFNK: Matrix-vector products

---

for  $l \geq 1$  (Newton iteration)  
  ...  
   $\mathbf{r}(\mathbf{u}^{(l)}) = \mathbf{u}^{(l)} - f(\mathbf{u}^{(l)})$   
  for  $l_K \geq 1$  (Krylov iteration)  
    ...  
    Krylov asks for  $\mathbf{J}\mathbf{v}^{(l_K)}$   
    Perturb the solution estimate:  $\mathbf{u}_{per}^{(l_K)} = \mathbf{u}^{(l)} + \varepsilon \mathbf{v}^{(l_K)}$   
    Perform a single PM iteration to calculate  $f(\mathbf{u}_{per}^{(l_K)})$   
    Calculate:  $\mathbf{r}(\mathbf{u}_{per}^{(l_K)}) = \mathbf{u}_{per}^{(l_K)} - f(\mathbf{u}_{per}^{(l_K)})$   
    Calculate the matrix-vector product:  $\mathbf{J}\mathbf{v}^{(l_K)} = \frac{\mathbf{r}(\mathbf{u}_{per}^{(l_K)}) - \mathbf{r}(\mathbf{u}^{(l)})}{\varepsilon}$   
  ...  
  end  
  ...  
end

---

$$\frac{\max_i (|\Phi_{crit,i}^{(l+1)} - \Phi_{crit,i}^{(l)}|)}{\max_i (|\Phi_{crit,i}^{(l)}|)} < \epsilon_2 \quad (24)$$

$$\max_i \left( \left| \frac{\Phi_{crit,i}^{(l+1)} - \Phi_{crit,i}^{(l)}}{\Phi_{crit,i}^{(l)}} \right| \right) < \epsilon_3 \quad (25)$$

In the case of slow convergence the magnitude of the differences between successive iterations may be small even though the error of the solution is still large. For this reason, a residual-based criterion is also used:

$$\| k^{(new)} \mathbf{A}_{crit} \Phi_{crit}^{(new)} - \mathbf{F} \Phi_{crit}^{(new)} \|_2 < \epsilon_4 \quad (26)$$

where the subscript 2 indicates the Euclidean norm. Eq. (26) checks the residual error obtained by entering the current estimate of the solution in the matrix equation to be solved.

### 4.3. Solution of large linear systems

For the current framework, linear systems need to be solved at different stages. Linear systems arise from each iteration of PM because of Eq. (11). If the PM-JFNK scheme is applied, additional ones are due to Eq. (18). Finally, the solution of the linear system Eq. (13) is required for the calculation of the neutron noise. These linear systems consist of sparse matrices and their size may be large when a fine mesh is used for the spatial discretization of the equations. Then iterative linear solvers are a more preferable choice since the storage requirements and the number of operations are less demanding in comparison to direct methods. On the other hand, they are often slow or even fail to converge, so preconditioning is necessary. The iterative GMRES method is selected and discussed below together with 3 preconditioners, namely SGS, ILU and SPAI.



#### 4.3.1. GMRES method

For the solution of large, sparse and non-symmetric linear systems such as the ones involving the matrices  $\mathbf{A}_{crit}$  and  $\mathbf{A}_{noise}$ , the Krylov-class Generalized Minimal RESidual (GMRES) method (Saad and Schultz (1986)) is selected. The general idea behind GMRES is that a chain of orthonormal vectors can be generated to construct the basis of a Krylov subspace  $\mathbf{K}_j$ :

$$\mathbf{K}_j = \text{span}(\mathbf{r}_0^{linear}, \mathbf{A}\mathbf{r}_0^{linear}, \mathbf{A}^2\mathbf{r}_0^{linear}, \dots, \mathbf{A}^{j-1}\mathbf{r}_0^{linear}) \quad (27)$$

where  $\mathbf{r}_0^{linear} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$  is the linear residual and  $\mathbf{x}_0$  is the initial guess of  $\mathbf{x}$  for the linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . The solution of the linear system is constructed iteratively as a linear combination of the Krylov vectors (Eq. (27)) and can be written as:

$$\mathbf{x}_j = \mathbf{x}_0 + \sum_{i=0}^{j-1} \beta_i(\mathbf{A})^i \mathbf{r}_0^{linear} \quad (28)$$

where  $j$  is the Krylov iteration index and the scalars  $\beta_i$  are calculated to minimize the residual.

The use of matrix-vector products  $(\mathbf{A})^i \mathbf{r}_0^{linear}$  by GMRES is advantageous when, for instance, the method is combined with the PM-JFNK algorithm described in Section 4.2.2. The estimation of these products can in fact replace the construction of the Jacobian matrix. In addition, the property is also helpful because it allows the implementation of the preconditioners in the form of matrix-vector products, a necessary feature when large linear systems are solved. The cost of one GMRES iteration grows with the number of iterations. Therefore, since this work targets large linear systems, GMRES is restarted after a number, say  $n$ , of iterations, with  $\mathbf{x}_n$  as the initial guess. For restarted GMRES the notation GMRES( $n$ ) is used.

#### 4.3.2. Preconditioning

The purpose of preconditioning (Benzi, 2002) is to improve the spectral properties of the coefficient matrix and thus the performance of the linear solver. Having a clustered spectrum of eigenvalues around unity often results in rapid convergence. If  $\mathbf{A}\mathbf{x} = \mathbf{b}$  is a generic linear system and  $\mathbf{P}$  is a non-singular matrix that approximates  $\mathbf{A}^{-1}$ , then the left preconditioned form is:

$$\mathbf{P}\mathbf{A}\mathbf{x} = \mathbf{P}\mathbf{b} \quad (29)$$

Generally, one can classify the preconditioners into physics-based and algebraic. The former are problem-dependent and thus very efficient. However, they may require a significant developmental effort. The latter may be less efficient but at the same time they are general-purpose and easily applicable requiring only information contained in  $\mathbf{A}$ . This work focuses on algebraic preconditioners.

A reasonable initial choice is the SGS preconditioner  $\mathbf{P}_{SGS}$ . The coefficient matrix  $\mathbf{A}$  of the linear system is decomposed as:  $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$  where  $\mathbf{L}$  is a strictly lower triangular matrix,  $\mathbf{U}$  is a strictly upper triangular matrix and  $\mathbf{D}$  is a diagonal matrix containing the main diagonal of  $\mathbf{A}$ . From the split an incomplete factorization is derived:  $\mathbf{A} \approx (\mathbf{D} + \mathbf{L})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{U})$ . Based on this factorization,  $\mathbf{P}_{SGS}$  is defined as:

$$\mathbf{P}_{SGS} \equiv ((\mathbf{D} + \mathbf{L})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{U}))^{-1} = (\mathbf{D} + \mathbf{U})^{-1}\mathbf{D}(\mathbf{D} + \mathbf{L})^{-1} \quad (30)$$

The main advantage of  $\mathbf{P}_{SGS}$  is that, having  $\mathbf{A}$ , the operation  $\mathbf{f} = \mathbf{P}_{SGS}\mathbf{w}$  can be replaced by two linear systems and one matrix-vector multiplication, i.e.  $(\mathbf{D} + \mathbf{L})\mathbf{z} = \mathbf{w}$ ,  $\mathbf{c} = \mathbf{D}\mathbf{z}$  and  $(\mathbf{D} + \mathbf{U})\mathbf{f} = \mathbf{c}$ . Such linear systems are solved fast performing forward/backward substitution of variables because of the triangular form of the coefficient matrices. Thus, the fact that GMRES works with matrix-vector products, offers a great advantage because  $\mathbf{P}_{SGS}$  can be applied to vectors.

The second choice is the Incomplete LU factorization (ILU) preconditioner  $\mathbf{P}_{ILU}$ . ILU is based on the approximation  $\mathbf{A} \approx \mathbf{L}\mathbf{U}$  where  $\mathbf{L}$  is a lower triangular and  $\mathbf{U}$  is an upper triangular matrix. Then  $\mathbf{P}_{ILU}$  is defined as:

$$\mathbf{P}_{ILU} \equiv (\mathbf{L}\mathbf{U})^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1} \quad (31)$$

Its implementation in GMRES has the same advantage that was pointed out for the case of SGS. ILU preconditioners usually suffer from the fill-in phenomenon. When large linear systems are solved, as in this case, the fill-in

phenomenon may create memory problems canceling the advantage that a sparse  $\mathbf{A}$  offers. However, if a part of fill-in is discarded during the factorization process, simple but still powerful preconditioners can be generated. The no fill-in ILU factorization or ILU(0) which preserves the sparsity of the matrix  $\mathbf{A}$ , is thus selected in this work. If other versions than ILU(0) are desired, then their applicability to large systems may become questionable because of the difficulty in parallelization (Benzi, 2002).

The last option is the SPAI (SParse Approximate Inverse) preconditioner  $\mathbf{P}_{SPAI}$  developed by Grote and Huckle (1997). The central idea relies on the minimization of a suitable objective function to create an approximation of  $\mathbf{A}^{-1}$ . Since the perfect right preconditioner  $\mathbf{P}$  would satisfy the relationship  $\mathbf{AP} = \mathbf{I}$  with  $\mathbf{I}$  the identity matrix,  $\mathbf{P}_{SPAI}$  is built by minimizing the Frobenius norm of the difference between  $\mathbf{I}$  and  $\mathbf{AP}_{SPAI}$ :

$$\|\mathbf{I} - \mathbf{AP}_{SPAI}\|_F^2 \quad (32)$$

The objective function can be decoupled as:

$$\|\mathbf{I} - \mathbf{AP}_{SPAI}\|_F^2 = \sum_{j=1}^n \|\mathbf{e}_j - \mathbf{Ap}_j\|_F^2 \quad (33)$$

where  $\mathbf{e}_j$  is the  $j$ th column of the identity matrix and  $\mathbf{p}_j$  is the  $j$ th column of matrix  $\mathbf{P}_{SPAI}$ . Thus minimizing Eq. (32) is equivalent to minimizing the sum of  $n$  independent individual least-square problems. This is useful for parallel implementations. In this work, an adaptive version of SPAI is employed. In fact, the minimization is stopped when the 2-norm of each column of  $\mathbf{I} - \mathbf{AP}_{SPAI}$  is less than a predefined tolerance  $\epsilon_{SPAI}$ . Therefore, the quality and the construction cost can be controlled by tuning  $\epsilon_{SPAI}$ . For this reason, the notation SPAI( $\epsilon_{SPAI}$ ) is used.

Although the computational cost of SPAI preconditioner can be high, its performance can be powerful. Since the criticality problem of Eq. (11) generates a sequence of linear systems with the same coefficient matrix, an initial investment to an expensive but also efficient preconditioner seems attractive. Additionally, because GMRES can not be parallelized easily, a part of the burden of the linear system solution may be transferred to an efficient parallel preconditioner. In this work, SPAI is constructed with a third-party software (Grote and Hagemann (2006)). Since that version is limited to real numbers, this preconditioner is used only for the criticality problem.

## 5. Numerical results

The performance of the methods discussed in Section 4 is evaluated with three test cases respectively associated with a 2-D one-region homogeneous reactor core, a 1-D one-region homogeneous reactor core, and a 3-D Pressurized Water Reactor (PWR) fully-heterogeneous core. In all the cases, a localized, fluctuating perturbation is introduced in the system. The first case of a 2-D one-region homogeneous reactor core is relevant because a semi-analytical solution is available. The criticality and neutron noise solvers are investigated separately in order to highlight their specific needs and features. After a satisfactory fine mesh is found, the different solvers are compared. The second case of a 1-D one-region homogeneous reactor core is used to compare the behavior of GMRES with respect to the criticality and neutron noise calculations. The third case of a 3-D full reactor core allows to study the developed strategy when applied to a realistic configuration. The analysis also shows how a coarse-mesh approximation of the noise source affects the neutron noise predictions. The criticality solvers are tested with two different convergence criteria: a set of regular criteria where all tolerances ( $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ , see Section 4.2.4) are equal to  $10^{-6}$  and a set of tight criteria where all tolerances are equal to  $10^{-10}$ . A flat flux distribution and  $k = 1$  are taken as initial guess for all criticality calculations. In the case of PM-JFNK, two free PM iterations are used at the beginning in order to ensure the convergence to the dominant eigenpair.

### 5.1. 2-D one-region homogeneous nuclear reactor core

This test case considers a 2-D one-region system near to criticality ( $k \approx 1$ ). The 2-D configuration is obtained from a cylindrical reactor core, by suppressing the axial dimension and taking a circular slice. The diameter of the system is set to  $2R=301$  cm. The homogeneous macroscopic cross-sections, the diffusion coefficients and the

Table 2: Macroscopic cross-sections and diffusion coefficients.

$\Sigma_{a,1} (cm^{-1})$	$\Sigma_{a,2} (cm^{-1})$	$v\Sigma_{f,1} (cm^{-1})$	$v\Sigma_{f,2} (cm^{-1})$	$\Sigma_r (cm^{-1})$	$D_1 (cm)$	$D_2 (cm)$
0.0115	0.1019	0.0057	0.1425	0.0151	1.4376	0.3723

Table 3: Dynamic parameters.

$\beta_{eff} (-)$	$l (s^{-1})$	$\nu_1 (cm/s)$	$\nu_2 (cm/s)$
0.00535	0.0851	18230400	413067

dynamic nuclear parameters are representative of a PWR core, and are given in Tables (2 & 3). The noise source is a point-like fluctuation of the macroscopic removal cross-section ( $\Sigma_r$ ) at the center of the core with a frequency of 1 Hz. The reference semi-analytical critical and noise solutions have been derived by Demaziere and Andhill (2005).

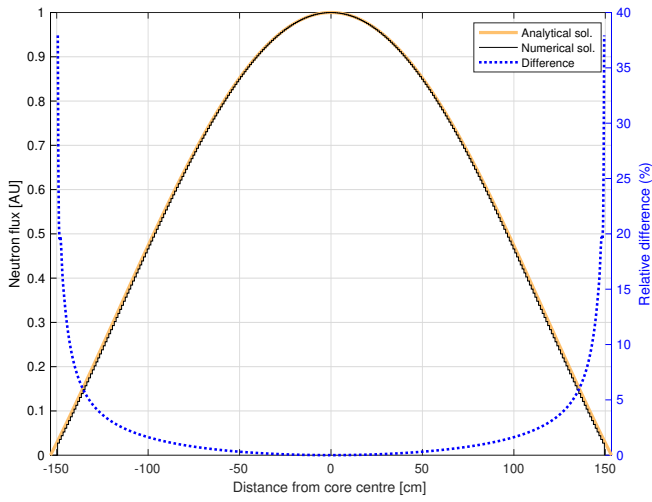
### 5.1.1. Criticality problem

The three criticality solvers based on PM, PM-Cheb and PM-JFNK are applied to the case of the 2-D one-region system. For the solution of the linear systems of Eq. (11), GMRES is preconditioned with SGS, ILU(0) and SPAI(0.2) for all PM versions. The spatial domain is discretized according to a relatively coarse Cartesian mesh, with  $301 \times 301$  nodes, where  $\Delta x = 1$  cm and  $\Delta y = 1$  cm. This computational grid leads to a system of  $2N = 140,962$  equations (2 being the number of neutron energy groups, see Section 2). The results of the different criticality solvers are in good agreement with the semi-analytical solution. The same value, i.e.  $k=0.99395$ , of the dominant eigenvalue is estimated. As an example, Fig. (1) shows the comparison between the neutron flux distribution calculated with PM-JFNK preconditioned with SPAI(0.2) and the analytical one. The numerical solution reproduces the analytical solution satisfactorily. The significant discrepancy that exists at the boundary of the system is attributed to approximation of the circular boundary with spatial discretization.

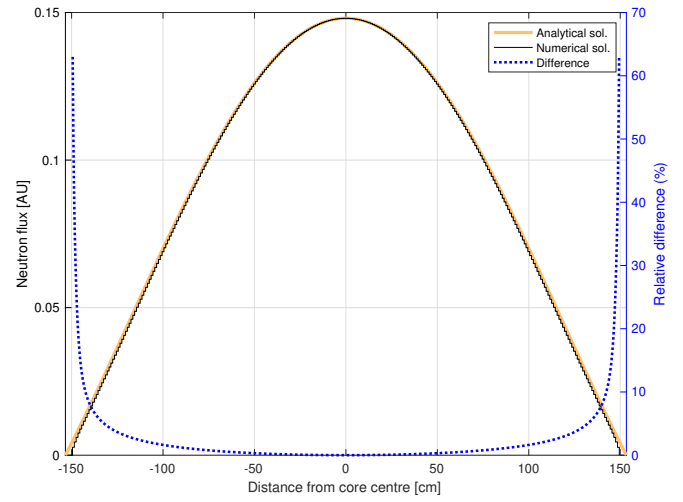
The numerical performances of the 3 solvers coupled to each of the 3 preconditioners are reported in Table (4). The tight convergence criteria are applied. GMRES(30) is used with a tolerance selected empirically equal to  $10^{-9}$  so that the linear solution is accurate enough to allow the convergence of PM. Given the same preconditioner for all the solvers, PM-Cheb is the faster option. As regards the preconditioning of GMRES, SPAI(0.2) is the most efficient option. Fig. (2) provides details of the behavior of the preconditioners for the case of the simple PM solver, at the first iteration. ILU(0) and SGS also accelerate the calculations significantly, although the gain is smaller. For the analysis of the residual error with outer/Newton iterations, the example of the three versions of PM using ILU(0) is depicted in Fig. (3a). The decrease of the residual error is very slow for the free PM, while PM-JFNK converges rapidly. However PM-JFNK requires a high computational time because of its two outer/inner iteration levels (Section 4.2.3). Fig. (3b) emphasizes this aspect, showing how the cost of each Newton iteration is sufficiently high to make PM-JFNK slower than PM-Cheb.

### 5.1.2. Neutron noise problem

After the criticality calculation, the neutron noise solution is obtained with the same  $301 \times 301$  mesh. Two versions of the solver for the linear system of Eq. (13) are tested. They rely on GMRES preconditioned with SGS and ILU(0), respectively. The comparison between the neutron noise predicted by GMRES together with ILU(0) and the analytical solution shows a large discrepancy close to the location of the neutron noise source, because of the coarse mesh (see Fig. (4)). More specifically, the maximum relative difference in the amplitude of the fast and thermal neutron noise is close to 25% and 12%, respectively. Regarding the phase, the difference is up to 10% in both the cases of fast and thermal noise. Therefore, a finer mesh is selected for the criticality and noise calculations. The new mesh has  $903 \times 903$  nodes, being the nodes 9 times smaller than the previous ones. This corresponds to a system of equations with  $2N = 1,268,658$  (i.e. a factor of 9 larger than the coarser one). The improvement in the prediction of the neutron noise is shown in Fig. (5). The new mesh reproduces more satisfactorily the sharp shape



(a) Fast neutron flux.



(b) Thermal neutron flux.

Figure 1: Comparison of the PI-JFNK (SPAI(0.2)) coarse-mesh numerical solution with the analytical solution for the 2-D one-region criticality problem.

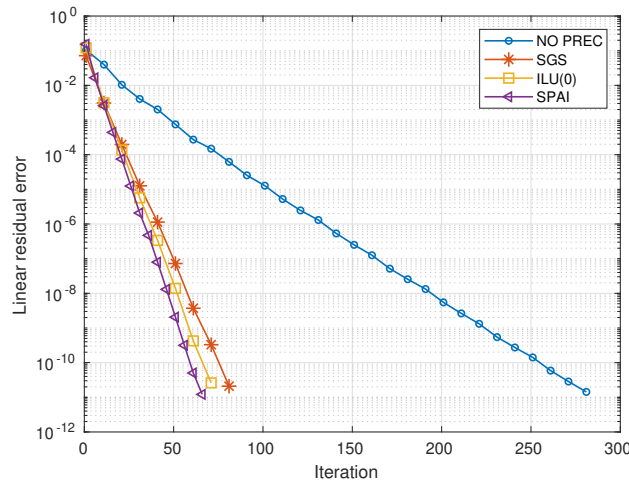
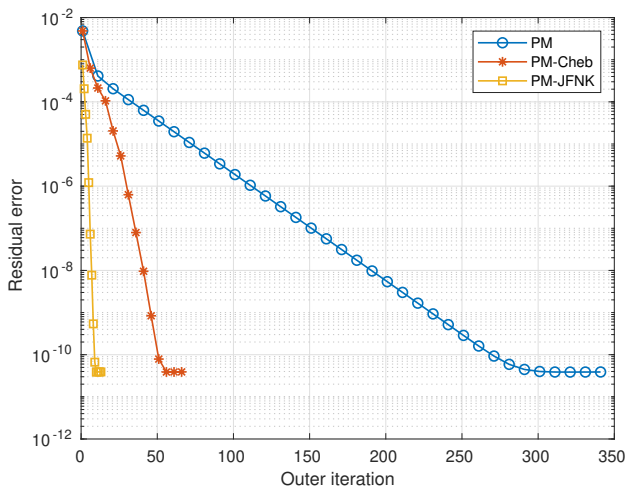
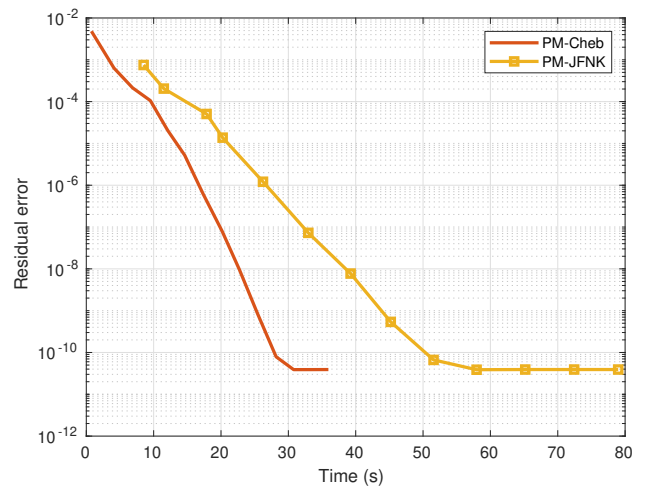


Figure 2: GMRES convergence at the first criticality iteration for the 2-D one-region test case.



(a) Residual error with iteration.



(b) Residual error with time.

Figure 3: Residual error of the criticality calculation for 2-D one-region test case (with ILU(0)).

Table 4: Comparison of different criticality solvers for the 2-D test case with tight convergence criteria.

Solver	Precondit.	Newton (Krylov) iter.	Outer iter.	Inner iter.	Time (s)
PM	SGS	-	346	19318	416
PM	ILU(0)	-	346	16773	203
PM	SPAI(0.2)	-	346	15580	193
PM-Cheb	SGS	-	66	3781	102
PM-Cheb	ILU(0)	-	66	3270	55
PM-Cheb	SPAI(0.2)	-	66	3028	49
PM-JFNK	SGS	13 (74)	114	6522	162
PM-JFNK	ILU(0)	13 (72)	112	5548	101
PM-JFNK	SPAI(0.2)	13 (72)	112	5133	73

of the neutron noise distribution near the location of the noise source. The differences in the amplitude of the fast and thermal neutron noise are reduced to less than 12% and 4.5%. Similarly, the differences in phase decrease, i.e. being less than 2.5%. The findings of this paragraph suggest that the computation of neutron noise requires a finer mesh than the one of the static neutron flux, in the vicinity of the noise source.

The numerical performance of the methods is shown in Fig. (6). ILU(0) leads to a faster convergence of GMRES, which is obtained in 80 min. In the case of the  $903 \times 903$  mesh and tight convergence criteria, the criticality calculation using PM-Cheb with ILU(0) takes 37 minutes. In other words, the solution of the single neutron noise linear system is more time-consuming than the solution of the sequence of linear systems generated from the PM iteration. In fact, the linear systems of PM needs significantly less GMRES iterations compared to the noise problem. This finding is investigated in the next section.

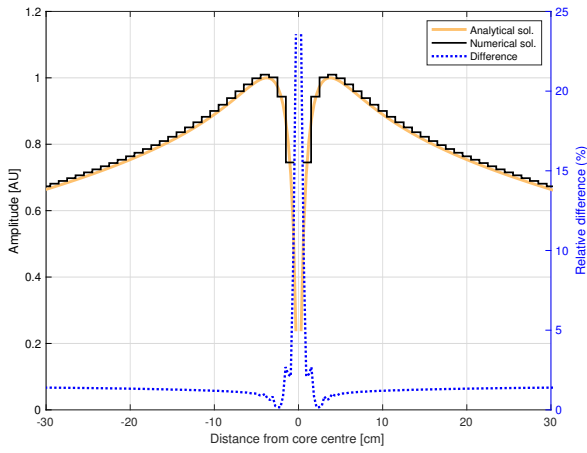
## 5.2. 1-D one-region reactor core: GMRES applied to criticality and neutron noise calculations

As mentioned above, the linear systems in the criticality problem need less GMRES iterations than the one associated with the neutron noise calculation. Then the spectral properties of  $\mathbf{A}_{crit}$  are expected to better favor convergence as compared to the spectral properties of  $\mathbf{A}_{noise}$ . The condition number of the matrix  $\mathbf{A}_{crit}$  of the 2-D problem is estimated as  $4.4697 \times 10^3$ , whereas the one of the matrix  $\mathbf{A}_{noise}$  as  $1.7663 \times 10^6$  confirming the hypothesis. The large difference is a first explanation, and indicates that GMRES coupled with the same type of preconditioner cannot solve both the criticality and the neutron noise problem with the same effort.

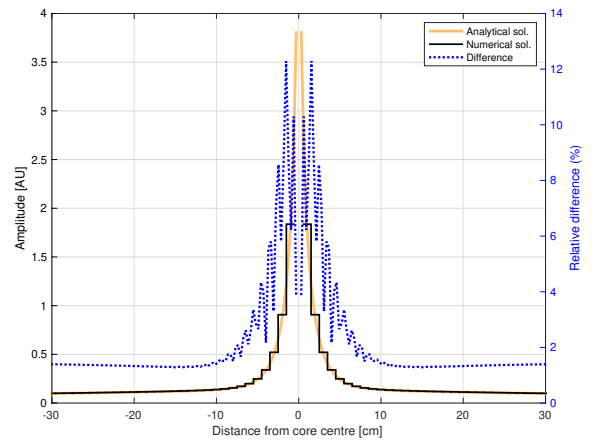
The next step is to study the eigenvalue distribution of the preconditioned  $\mathbf{A}_{crit}$  and  $\mathbf{A}_{noise}$  using ILU, as an example. For this purpose, a 1-D one-region homogeneous reactor core is considered (Demaziere, 2011c) so that the matrices are relatively small and easy to manipulate. The system is an infinite slab with width equal to  $2a = 300\text{cm}$ . The homogeneous macroscopic cross-sections and diffusion coefficients are chosen to be representative of a PWR (Tables (2 & 3)). The noise source is a point-like 1 Hz perturbation of the macroscopic removal cross-section ( $\Sigma_r$ ) located at  $x = -50\text{cm}$ . A mesh with 301 nodes and  $\Delta x = 1\text{ cm}$  is used leading to  $2N=602$ .

The eigenvalue distributions of the criticality matrix  $\mathbf{A}_{crit}$  and the noise matrix  $\mathbf{A}_{noise}$  both preconditioned with ILU(0), are shown in Fig. (7a). ILU(0) leads the eigenvalues of  $\mathbf{A}_{crit}$  to concentrate close to 1, which is the desired situation. On the other hand, the eigenvalues of  $\mathbf{A}_{noise}$  are spread away from unity. This is the reason for the poorer behavior of GMRES when applied to the computation of neutron noise.

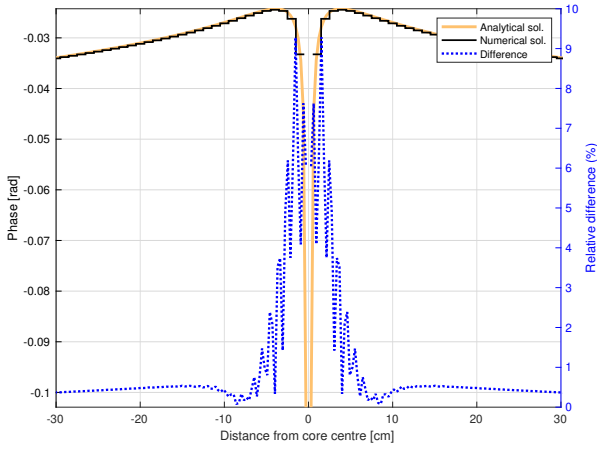
In order to have a closer distribution of the eigenvalues, a more accurate and denser ILU preconditioner is investigated. Therefore the ILU(0) is compared with another version of ILU where the fill-in is controlled with a drop tolerance, namely the Crout version of ILU (denoted as ILUC). The following drop tolerances are used:  $10^{-5}$ ,  $10^{-7}$  and  $10^{-10}$ . Figs (7) shows that the reduction in drop tolerance makes progressively the eigenvalue



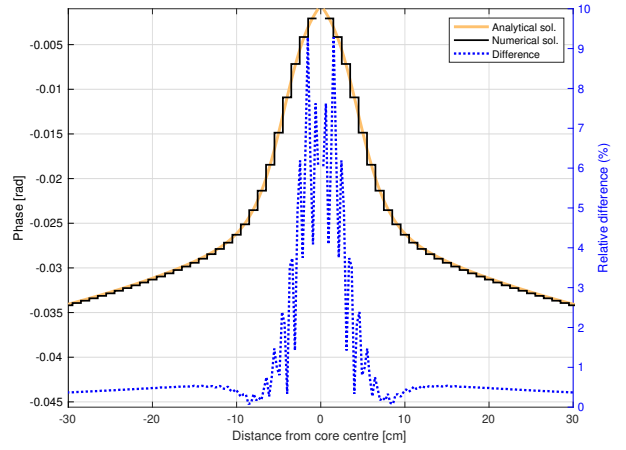
(a) Amplitude of fast neutron noise.



(b) Amplitude of thermal neutron noise.

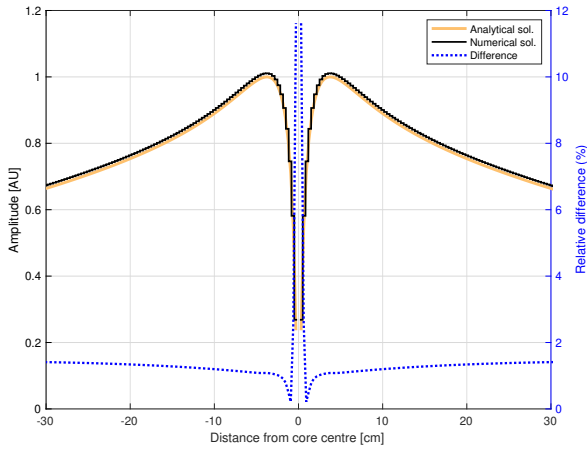


(c) Phase of fast neutron noise.

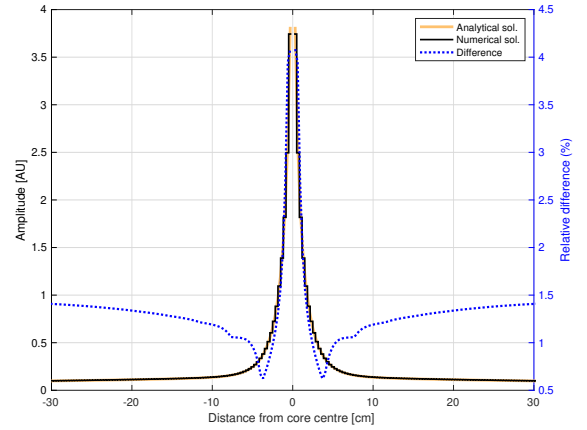


(d) Phase of thermal neutron noise.

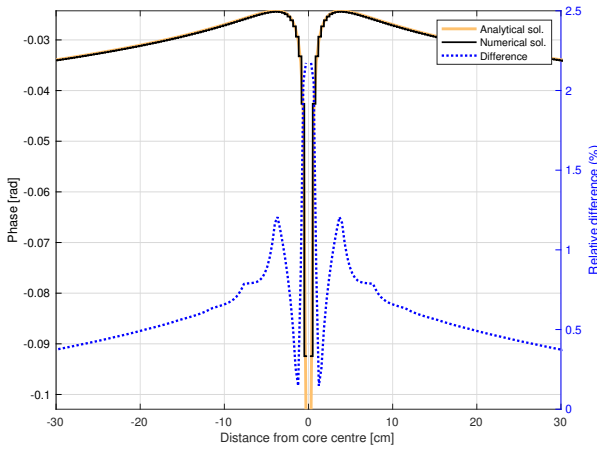
Figure 4: Comparison of the coarse-mesh numerical solution with the analytical solution for the 2-D one-region test case.



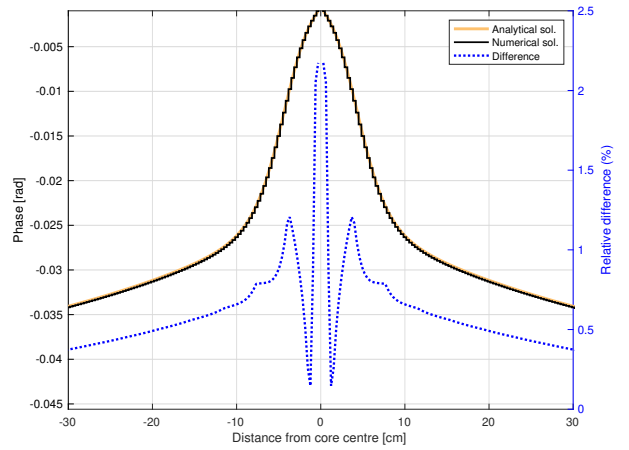
(a) Amplitude of fast neutron noise.



(b) Amplitude of thermal neutron noise.



(c) Phase of fast neutron noise.



(d) Phase of thermal neutron noise.

Figure 5: Comparison of the fine-mesh numerical solution with the analytical solution for the 2-D one-region noise problem.

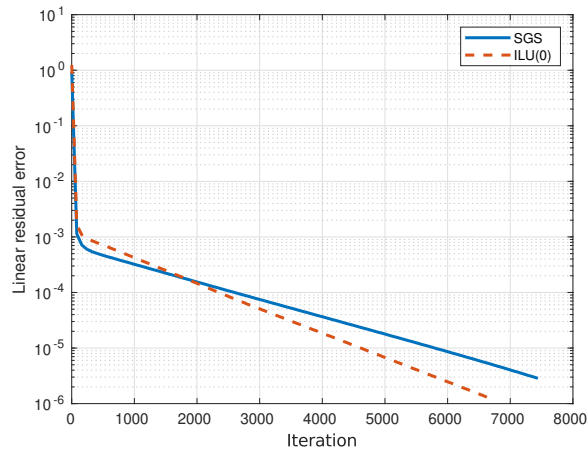
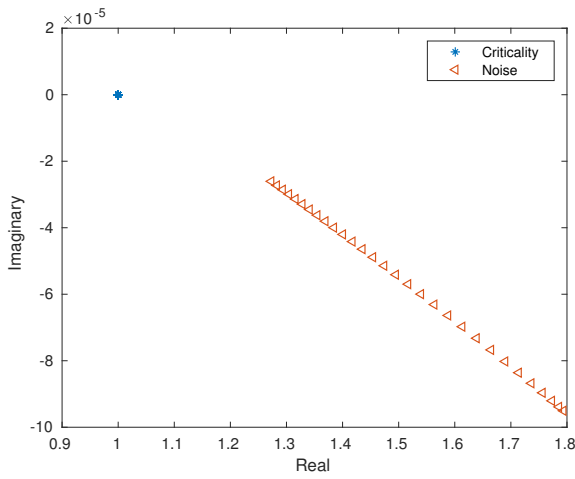
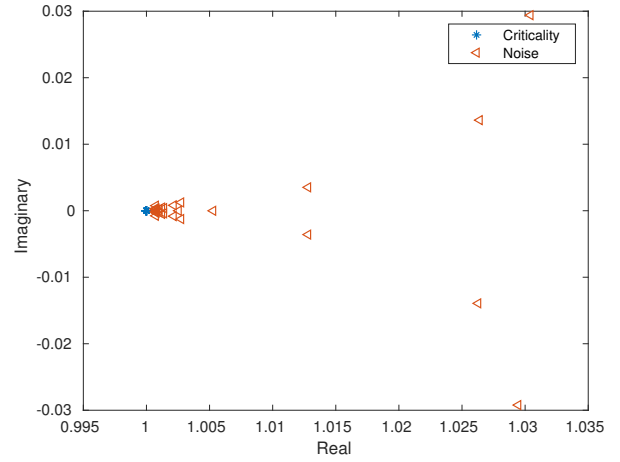


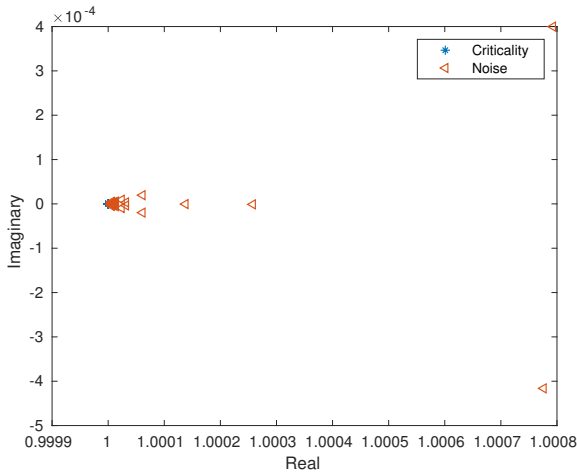
Figure 6: GMRES convergence of noise solver for the 2-D one-region test case.



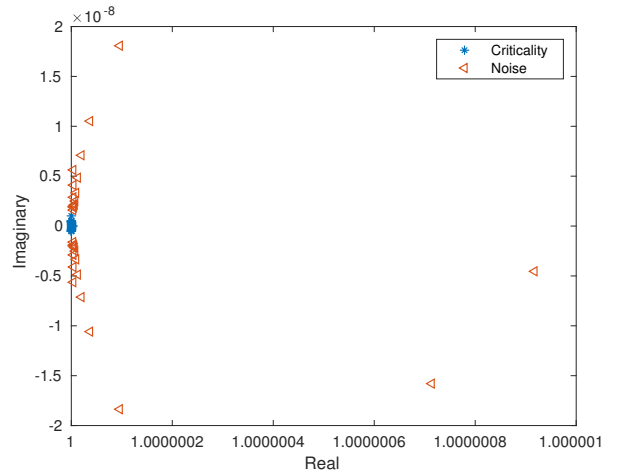
(a) ILU(0) (no fill-in)



(b) ILUC with droptol= $10^{-5}$  (fill-in=58733)



(c) ILUC with droptol= $10^{-7}$  (fill-in=93259).



(d) ILUC with droptol= $10^{-10}$  (fill-in=134470).

Figure 7: Eigenvalue distribution of the criticality and noise preconditioned coefficient matrices.

spectrum as compact as the one of the criticality computation. Then, the performance of each preconditioner is tested on solving the noise problem. As illustrated in Fig. (8) by the variation of the residual error with the number of iterations, a drop tolerance of  $10^{-10}$  for ILUC provides a performance that is similar to the ILU(0) in the criticality calculations. However, such a drop tolerance for ILUC is very tight and not feasible for realistic reactor applications because it implies very high fill-in. These findings indicate that, although the noise problems studied in this work are solved with a reasonable computational cost with the selected methods, there is a significant potential for further improvements.

### 5.3. 3-D heterogeneous PWR core

A 3-D heterogeneous full core near to criticality with diameter  $2R = 344$  cm and axial height  $H = 396.24$  is considered. The noise source is a high-frequency localized absorber of variable strength, i.e. a perturbation in the macroscopic thermal absorption cross-section ( $\Sigma_{a,2}$ ) with a frequency of 1 kHz. The perturbation is assumed to be distributed over a volume with dimension  $0.8958$  cm  $\times$   $0.8958$  cm  $\times$   $7.62$  cm. Two Cartesian meshes are used. The first mesh, denoted as coarse, has a node size of  $\Delta x = \Delta y = 2.6875$  cm, and  $\Delta z = 7.62$  cm. According to this resolution, the spatial distribution of the neutron noise source will be approximated over a larger computational volume. The second mesh, specified as fine, has a smaller node size of  $\Delta x = \Delta y = 0.8958$  cm and  $\Delta z = 7.62$  cm, so it allows to define exactly the perturbation in space. The aim is twofold: first, to test the applicability of the solvers to realistic problems; second, to study the effect of the spatial resolution of the mesh on the modelling of highly localized neutron noise sources.



### 5.3.1. Criticality problem

The three criticality solvers based on PM, PM-Cheb and PM-JFNK are applied. For the solution of the linear systems of Eq. (11), GMRES is preconditioned with ILU(0) and SPAI(0.4) for PM-Cheb and PM-JFNK, and only with SPAI(0.4) in case of non-accelerated PM. For this test case a larger  $\epsilon_{spai}$  is used (0.4 instead of 0.2) in order to mitigate the computational effort of SPAI's construction. The spatial domain is discretized with the coarse mesh since as observed in Section 5.1, the criticality calculation does not require a very fine mesh. This discretization leads to a system of  $2N = 1,252,328$  equations.

The numerical performance of the methods when regular convergence criteria are used is reported in Table (5). GMRES(30) is applied with a tolerance equal to  $10^{-6}$  for PM with SPAI and  $10^{-5}$  for all the other cases. The GMRES tolerance is selected empirically to provide an inner solution that is accurate enough to the outer iteration. PM-Cheb with SPAI(0.4) is the faster option. PM-JFNK is slower than PM-Cheb, which agrees with the findings for the 2-D problem. PM without acceleration is not able to converge in 700 outer iterations, confirming its limited capability. PM-JFNK with SPAI(0.4) is slower than PM-JFNK with ILU(0) and a possible reason is that SPAI has about 5 times more non-zero entries than ILU(0). Hence, SPAI implies more expensive matrix-vector operations which may counterbalance the lower number of inner iterations. The results of the calculations performed with tight convergence criteria are reported in Table (6). In this set of simulations GMRES(50) is used with a tolerance chosen to be  $10^{-10}$ . The main finding is that PM-JFNK is the only method converging within 700 outer iterations.

### 5.3.2. Neutron noise problem

After the criticality analysis, the computation of noise follows. In order to study the effect of the mesh, two computations are performed: the first one estimates the neutron noise according to the coarse mesh with  $128 \times 128 \times 52$  nodes; and the second one (involving a new criticality calculation) is based on the fine  $384 \times 384 \times 52$  mesh which leads to  $2N = 11,261,952$ . As discussed in Section 5.1, the application of GMRES together with the ILU(0) preconditioner provides satisfactory results for the calculation of the neutron noise in the 2-D reactor core. Then the same solver with tolerance equal to  $10^{-10}$  is also used for the solution of the neutron noise linear system in the case of the 3-D realistic reactor core.

Figs (10a & 10b) show a comparison of the neutron noise amplitude normalized to the maximum value of the fine-mesh fast neutron noise. There is a good agreement of the two solutions except in the vicinity of the noise source where some discrepancy exists. This is more remarkable in the case of the thermal noise where the relative difference reaches 80%. These results suggest that a fine mesh is required in the close neighborhood of the noise source. A coarse resolution of the mesh does not allow an accurate modeling of the spatial distribution of the neutron noise source, and thus leads to a poor evaluation of the spatial variation of the induced noise close to the perturbation. On the other hand, the noise in the region lying farther can be predicted accurately even with a coarser mesh. Regarding the phase of the neutron noise, Figs (10c & 10d) illustrate that there is not any notable discrepancy between the two meshes. The computational time of the fine case is 2.6 h for the criticality calculation and 43 min for the noise problem with no more than 10 Gb of computer memory showing that realistic noise problems can be solved with a reasonable computational cost.

Table 5: Comparison of different criticality solvers for the 3-D full-core test case with regular convergence criteria.

Method	Precondit.	Newton (Krylov) iter.	Outer iter.	Inner iter.	Time (s)
PM	SPAI(0.4)	-	700 (not converg.)	10505	2034
PM-Cheb	ILU(0)	-	137	1919	313
PM-Cheb	SPAI(0.4)	-	122	1586	292
PM-JFNK	ILU(0)	11 (140)	174	2436	385
PM-JFNK	SPAI(0.4)	11 (140)	174	2262	429

Table 6: Comparison of different criticality solvers for the 3-D full-core test case with tight convergence criteria.

Method	Precondit.	Newton (Krylov) iter.	Outer iter.	Inner iter.	Time (s)
PM-Cheb	ILU(0)	-	700 (not converg.)	19620	4313
PM-Cheb	SPAI(0.4)	-	700 (not converg.)	18200	4232
PM-JFNK	ILU(0)	21 (243)	307	8611	1779
PM-JFNK	SPAI(0.4)	37 (344)	456	11856	2621

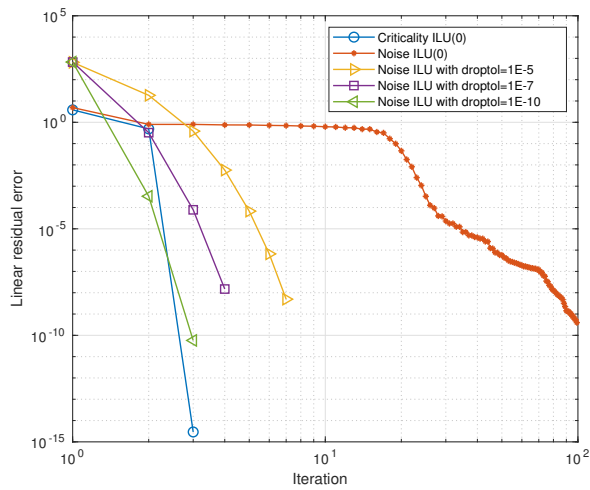


Figure 8: GMRES convergence with progressively higher ILU fill-in for the noise problem, and ILU(0) for the criticality problem.

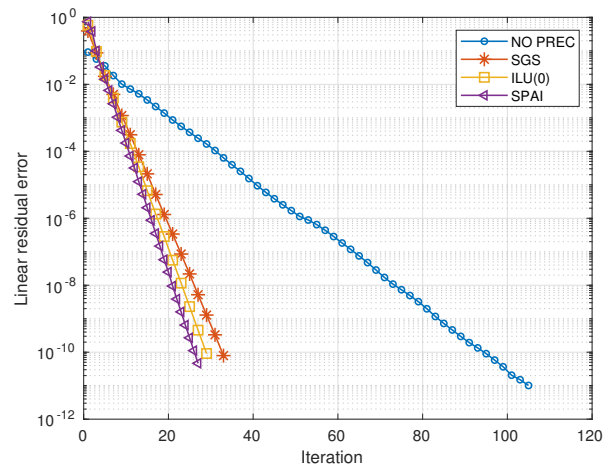
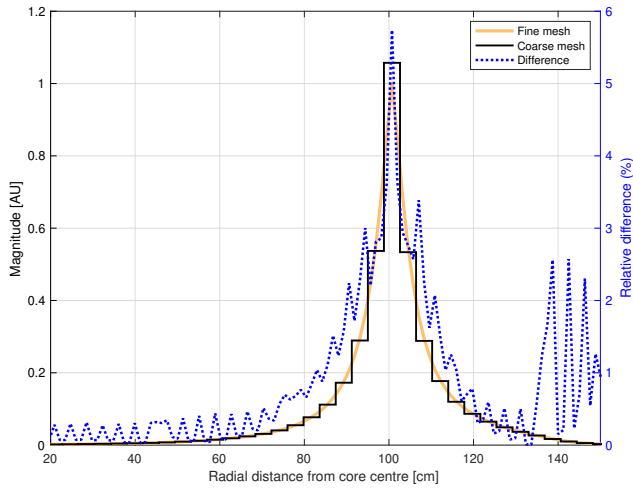
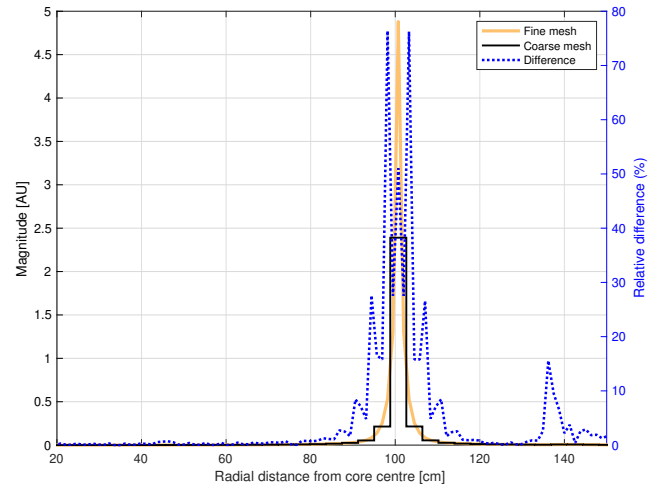


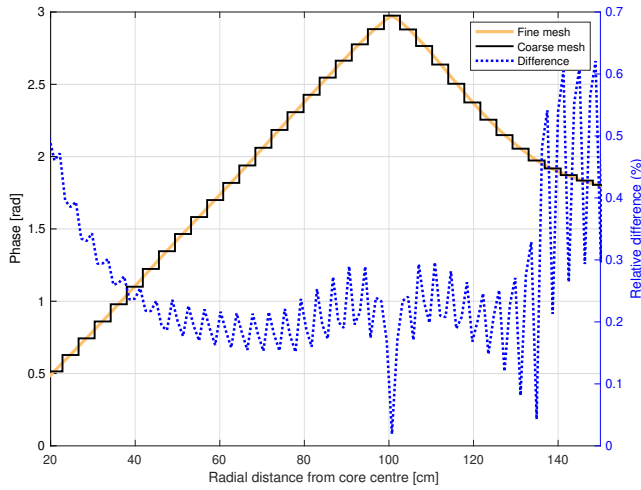
Figure 9: GMRES convergence at the first criticality iteration for the 3-D test case.



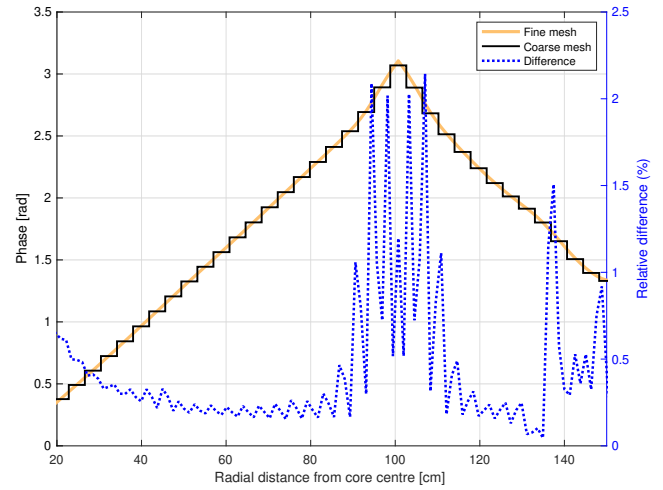
(a) Amplitude of fast neutron noise.



(b) Amplitude of thermal neutron noise.



(c) Phase of fast neutron noise.



(d) Phase of thermal neutron noise.

Figure 10: Comparison of the neutron noise between the fine and the coarse mesh.

## 6. Conclusion

In this paper, a strategy for solving fine mesh realistic neutron noise problems in nuclear reactors is presented. The neutron noise model relies on the diffusion approximation of the neutron transport equation and requires the solution of both the criticality (eigenvalue) and neutron noise equations in the frequency domain. The equations are spatially discretized according to a finite-difference method.

The Power Method is used for the criticality calculation and it is accelerated with two alternatives: the Chebyshev polynomials method and a non-linear technique based on the Jacobian-Free Newton Krylov approach. The restarted GMRES was selected for solving both the linear systems within the iterations of PM and the final linear system representative of the neutron noise problem, given a prescribed stationary perturbation. To improve the performance of the algorithm, GMRES was preconditioned with SGS, ILU and SPAI for the criticality part and with SGS and ILU for the neutron noise part of the problem.

The numerical analysis of the developed solvers is carried out for neutron noise examples in 1-D and 2-D homogeneous reactor cores and in a 3-D heterogeneous reactor core. A fine spatial discretization of the systems is applied for an accurate evaluation of the sharp gradients of the neutron noise that arise from the localized character of the perturbation. From the results of the comparative study, the following conclusions are drawn. For the criticality calculation, although the scheme that combines PM-Cheb and GMRES preconditioned with SPAI can be a very efficient option, it does not always converge. In these cases, the solver that consists of PM-JFNK and GMRES preconditioned with ILU or SPAI seems an advisable option because it can reach a converged solution even in the case of realistic reactor applications with very tight convergence criteria. This highlights the potential of JFNK-based techniques for the criticality problem. As regards the choice between ILU and SPAI, the computation of the first one is very cheap and fast, counterbalancing the somewhat higher reduction in number of iterations that SPAI achieves.

GMRES with ILU also shows a satisfactory behavior in the calculation of neutron noise. Nevertheless, the computational effort in the 2-D test-case was more demanding for the noise part of the problem than for the criticality part. This is due to the structure of the matrices involved. In particular, the eigenvalues of the neutron noise matrix may be widely spread affecting negatively the performance of GMRES. Such an outcome suggests that specific preconditioners should be investigated for neutron noise simulations.

Last but not least, it was shown that a fine mesh is required in the vicinity of the noise source to reproduce properly the strong gradients. In the regions lying farther, a coarser mesh is adequate. Further research on numerical schemes that allow to apply non-uniform meshes, being finer close to the neutron noise source and coarser elsewhere, could lead to a strong reduction of the computational effort without compromising the accuracy.

## ACKNOWLEDGEMENT

The research leading to these results has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 754316.

## Appendix A. Definition of $\Sigma_{dyn}^{crit}$ , $\phi_f^{crit}$ , $\phi_r$ and $\phi_a$

In the neutron noise diffusion Eq. (10),  $\Sigma_{dyn}^{crit}$ ,  $\phi_r$ ,  $\phi_a$  and  $\phi_f^{crit}$  are defined as follows. The matrix  $\Sigma_{dyn}^{crit}$  is:

$$\Sigma_{dyn}^{crit}(\mathbf{r}, \omega) \equiv \begin{bmatrix} -\Sigma_1^{crit}(\mathbf{r}, \omega) & \frac{v\Sigma_{f,2,0}(\mathbf{r})}{k_{eff}} \left(1 - \frac{i\omega\beta}{i\omega + \lambda}\right) \\ \Sigma_{r,0}(\mathbf{r}) & -\left(\Sigma_{a,2,0}(\mathbf{r}) + \frac{i\omega}{v_2}\right) \end{bmatrix} \quad (\text{A.1})$$

where

$$\Sigma_1^{crit}(\mathbf{r}, \omega) \equiv \Sigma_{a,1,0}(\mathbf{r}) + \frac{i\omega}{v_1} + \Sigma_{r,0}(\mathbf{r}) - \frac{v\Sigma_{f,1,0}(\mathbf{r})}{k_{eff}} \left(1 - \frac{i\omega\beta}{i\omega + \lambda}\right) \quad (\text{A.2})$$

The matrix  $\phi_f^{crit}$  is:

$$\phi_f^{crit}(\mathbf{r}, \omega) \equiv \begin{bmatrix} -\frac{\phi_{1,0}(\mathbf{r})}{k_{eff}}(1 - \frac{\omega\beta}{\omega+\lambda}) & -\frac{\phi_{2,0}(\mathbf{r})}{k_{eff}}(1 - \frac{\omega\beta}{\omega+\lambda}) \\ 0 & 0 \end{bmatrix} \quad (\text{A.3})$$

The matrix  $\phi_a$  is:

$$\phi_a(r) \equiv \begin{bmatrix} \phi_{1,0}(\mathbf{r}) & 0 \\ 0 & \phi_{2,0}(\mathbf{r}) \end{bmatrix} \quad (\text{A.4})$$

The column vector  $\phi_r$  is:

$$\phi_r(r) \equiv \begin{bmatrix} \phi_{1,0}(\mathbf{r}) \\ -\phi_{1,0}(\mathbf{r}) \end{bmatrix} \quad (\text{A.5})$$

## Appendix B. Spatial discretization of the diffusion operator

The spatial discretization scheme for Eqs. (8 & 10), is the same as the one used in (Demaziere, 2011a). Accordingly, a nuclear reactor consists of adjacent volumes or nodes. Given a three-dimensional Cartesian coordinate system, the generic node  $n$  can be identified by three indices (I,J,K) which are associated with the x-, y-, z-directions, respectively. The equations derived in Section (2) are discretized with respect to these nodes and spatially-averaged on each of them. For the sake of simplicity, the possible dependence on frequency is dropped. The notation and convention used through this paper are illustrated in Figs (B.11 & B.12).

Being  $\Sigma$  a generic macroscopic cross-section and  $\phi$  the scalar neutron flux, the quantities averaged over the node  $n$ , for the  $g$ -th energy group, are defined as:

$$\Sigma_{g,n} = \frac{\frac{1}{V_n} \int_{V_n} \Sigma_g(\mathbf{r}) \phi_g(\mathbf{r}) d\mathbf{r}}{\phi_{g,n}} \quad (\text{B.1})$$

$$\phi_{g,n} = \frac{1}{V_n} \int_{V_n} \phi_g(\mathbf{r}) d\mathbf{r} \quad (\text{B.2})$$

where  $V_n$  represents the volume of the node  $n$ . Considering that  $J_g(\mathbf{r}) = -D_g(\mathbf{r}) \nabla \phi_g(\mathbf{r})$ , using a spatial discretization scheme based on finite differences, and assuming that the scalar neutron flux in the middle of the nodes is equal to the node-averaged scalar neutron flux (box-scheme), the node-averaged streaming terms can be approximated as:

$$\frac{1}{V_n} \int_{V_n} \nabla \cdot [D_g(\mathbf{r}) \nabla \phi_g(\mathbf{r})] d\mathbf{r} = - \sum_{\aleph=x,y,z} \frac{J_{g,n}^{\aleph} - J_{g,n-1}^{\aleph}}{\Delta \aleph} = - \sum_{\aleph=x,y,z} (a_{g,n}^{\aleph} \phi_{g,n} + b_{g,n}^{\aleph} \phi_{g,n+1} + c_{g,n}^{\aleph} \phi_{g,n-1}) \quad (\text{B.3})$$

In this equation,  $\aleph$  represents the direction  $x$ ,  $y$ , or  $z$ ; and  $\Delta \aleph$  is the node width in the  $\aleph$ -direction. The subscripts  $n-1$  and  $n+1$  are related to the nodes on the two sides of the node  $n$  along one of the  $\aleph$ -directions (see Fig. (B.12)). The surface averaged net current  $J_{g,n}$ , is defined as:

$$J_{g,n}^{\aleph} = \frac{1}{\Delta \aleph \cdot \Delta \varphi} \int_{-\Delta \aleph/2}^{\Delta \aleph/2} \int_{-\Delta \varphi/2}^{\Delta \varphi/2} J_g(\mathbf{r}_{\aleph}) \cdot \mathbf{n}_{\aleph}^+ d\aleph d\varphi \quad (\text{B.4})$$

where  $\mathbf{n}^+$  is the outward normal relative to node  $n$ , and  $\mathbf{r}_{\aleph}$  represents the position of any point belonging to the surface normal. The boundary conditions of the system are the Marshak's ones, which in the case of multigroup diffusion theory, read as:

$$J_g(\mathbf{r}_{\aleph}) \cdot \mathbf{n}_{\aleph} = \frac{1}{2} \phi_g^b \quad (\text{B.5})$$

where  $r_{\aleph}$  represents a spatial point on the boundary with  $n_{\aleph}$  being the outward normal to the boundary, and  $\phi_g^b$  represents the scalar neutron flux at the boundary. The derivation of the coupling coefficients is given by Demaziere (2011b). The expressions of the coupling coefficients  $a_{g,n}^{\aleph}$ ,  $b_{g,n}^{\aleph}$  and  $c_{g,n}^{\aleph}$  are summarized in Table (B.1).

Table B.1: Coupling coefficients.

$a_{g,n}^{\mathcal{N}}$	$b_{g,n}^{\mathcal{N}}$	$c_{g,n}^{\mathcal{N}}$
if the node n-1 does not exist $\frac{2D_{g,n}D_{g,n+1}}{(\Delta\mathcal{N})^2(D_{g,n}+D_{g,n+1})} + \frac{1/2}{\Delta\mathcal{N} + \frac{(\Delta\mathcal{N})^2}{4D_{g,n}}}$	$-\frac{2D_{g,n}D_{g,n+1}}{(\Delta\mathcal{N})^2(D_{g,n}+D_{g,n+1})}$	$0$
if both nodes n-1 and n+1 exist $\frac{2D_{g,n-1}D_{g,n}}{(\Delta\mathcal{N})^2(D_{g,n-1}+D_{g,n})} + \frac{2D_{g,n}D_{g,n+1}}{(\Delta\mathcal{N})^2(D_{g,n}+D_{g,n+1})}$	$-\frac{2D_{g,n}D_{g,n+1}}{(\Delta\mathcal{N})^2(D_{g,n}+D_{g,n+1})}$	$-\frac{2D_{g,n-1}D_{g,n}}{(\Delta\mathcal{N})^2(D_{g,n-1}+D_{g,n})}$
if the node n-1 does not exist $\frac{2D_{g,n-1}D_{g,n}}{(\Delta\mathcal{N})^2(D_{g,n-1}+D_{g,n})} + \frac{1/2}{\Delta\mathcal{N} + \frac{(\Delta\mathcal{N})^2}{4D_{g,n}}}$	$0$	$-\frac{2D_{g,n-1}D_{g,n}}{(\Delta\mathcal{N})^2(D_{g,n-1}+D_{g,n})}$

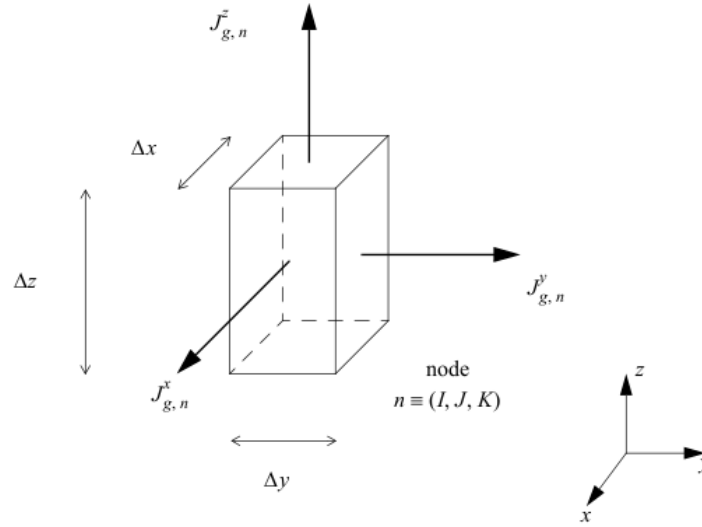


Figure B.11: Principles and conventions used for the spatial discretization of a node  $n$ .

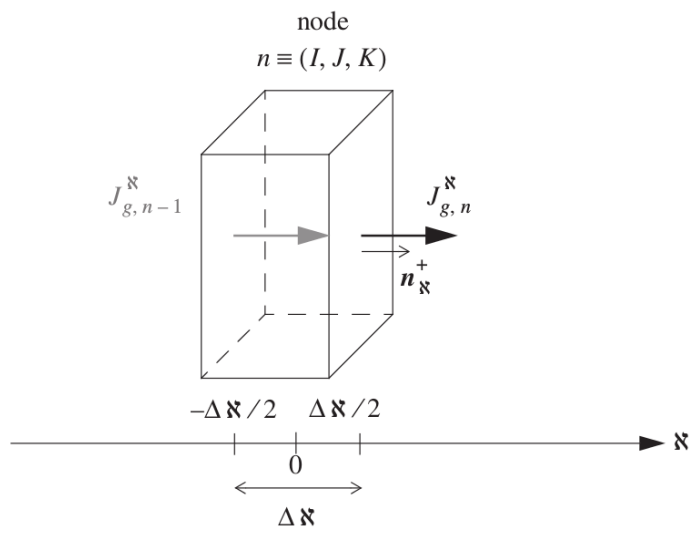


Figure B.12: Generic notations relative to a node  $n$  used in the spatial discretization along direction  $\xi$ .

## Appendix C. Chebyshev acceleration method

Various alternatives of the Chebyshev method are available for the acceleration of PM. The Chebyshev method implemented in this work is described in Hebert (1985), and it is based on a two-parameter formula with cyclical updating of DR. In particular, the two parameters used to control the acceleration procedure, are the initial number  $N$  of free PM iterations, and the number  $M$  of accelerated PM iterations per acceleration cycle. The scheme consists of 3 steps.

In step 1, a set of  $N$  free PM iterations are performed (see Alg. (1)). A DR estimate is calculated at the end of the  $l$ -th PM iteration with the formula:

$$DR^{(l+1)} = \left( \frac{(\mathbf{FR}^{(l)}, \mathbf{FR}^{(l)})}{(\mathbf{FR}^{(l-1)}, \mathbf{FR}^{(l-1)})} \right)^{1/2} \quad (\text{C.1})$$

where  $\mathbf{R}^{(l)} = \Phi^{(l+1)} - \Phi^{(l)}$ . At the end of the  $N$ -th iteration the method checks if  $DR^{(N+1)} > 0.5$ . If this condition is satisfied, the method sets  $N^* = N$  and moves to step 2. If not, it sets  $N = N + 1$  and another free PM iteration is performed.

In step 2, an acceleration cycle is performed including  $M$  extrapolated PM iterations:

$$\Phi^{(N^*+m+1)} = \Phi^{(N^*+m)} + \alpha^{(m)}[\mathbf{R}^{(N^*+m)} + \beta^{(m)}(\Phi^{(N^*+m)} - \Phi^{(N^*+m-1)})] \quad (\text{C.2})$$

where  $m = 1, \dots, M$  denotes the extrapolated iterations. The extrapolation factors are defined as:

$$\alpha^{(1)} = \frac{2}{2 - DR^{(N^*+1)}} \quad (\text{C.3})$$

$$\beta^{(1)} = 0 \quad (\text{C.4})$$

$$\alpha^{(m)} = \frac{4}{DR^{(N^*+1)}} \left( \frac{\cosh[(m-1)\gamma]}{\cosh[m\gamma]} \right) \quad (\text{C.5})$$

$$\beta^{(m)} = \left( 1 - \frac{DR^{(N^*+1)}}{2} \right) - \frac{1}{\alpha^{(m)}} \quad (\text{C.6})$$

where

$$\gamma = \cosh^{-1} \left( \frac{2}{DR^{(N^*+1)}} - 1 \right) \quad (\text{C.7})$$

DR is estimated at the beginning of each cycle and it is kept constant within the cycle.

In step 3, The DR estimate is updated at the end of the acceleration cycle by applying the error reduction factor  $E$  and the theoretical error reduction factor  $E^*$  so that:

$$E = \left( \frac{(\mathbf{FR}^{(N^*+M)}, \mathbf{FR}^{(N^*+M)})}{(\mathbf{FR}^{(N^*+1)}, \mathbf{FR}^{(N^*+1)})} \right)^{1/2} \quad (\text{C.8})$$

and

$$E^* = \left( C_{M-1} \left[ \frac{2 - DR^{(N^*+1)}}{DR^{(N^*+1)}} \right] \right)^{-1} \quad (\text{C.9})$$

In Eq. (C.9),  $C_{M-1}$  denotes the Chebyshev polynomial of degree  $M - 1$ :

$$C_{M-1} = \cosh[(M-1)\cosh^{-1}x] \quad (\text{C.10})$$

Hence DR is updated with the formula:

$$DR^{(N^*+M+1)} = \frac{DR^{(N^*+1)}}{2} \left[ \cosh \left( \frac{\cosh^{-1}(E/E^*)}{M-1} \right) + 1 \right] \quad (\text{C.11})$$

Finally the method sets  $N^* = N^* + M$  and moves to step 2 in order to start a new acceleration cycle. In this work each cycle is prolonged doubling the number of iterations since it was noticed that it has a positive effect to the acceleration of the convergence.



## References

- M. Benzi. Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, 182: 418–477, 2002.
- C. Demaziere. CORE SIM: A multi-purpose neutronic tool for research and education. *Annals of Nuclear Energy*, 38:2698–2718, 2011a.
- C. Demaziere. Description of the models and algorithms used in the CORE SIM neutronic tool. Technical Report CTH-NT-241, Chalmers University of Technology, Department of Applied Physics, Division of Nuclear Engineering, 2011b.
- C. Demaziere. Validation and demonstration of the CORE SIM neutronic tool. Technical Report CTH-NT-242, Chalmers University of Technology, Department of Applied Physics, Division of Nuclear Engineering, 2011c.
- C. Demaziere and G. Andhill. Identification and localization of absorbers of variable strength in nuclear reactors. *Annals of Nuclear Energy*, 32(8):812–842, 2005.
- J. Duderstadt and L. J. Hamilton. *Nuclear Reactor Analysis*. New York, 1976.
- S. Eisenstat and H. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM Journal of Scientific Computing*, 17:16–32, 1996.
- D. R. Ferguson and K. L. Derstine. Optimized iteration strategies and data management considerations for fast reactor finite difference diffusion theory codes. *Nuclear Science and Engineering*, 64:593–604, 1977.
- D. Gill and Y. Asmy. A Jacobian-Free Newton-Krylov iterative scheme for criticality calculations based on the neutron diffusion equation. In *Int. Conf. on Mathematics, Computational Methods and Reactor Physics*, Saratoga Springs, New York, May 3-7 2009.
- M. Grote and M. Hagemann. *SPAI - SParse Approximate Inverse Preconditioner*, March 2006.
- M. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18: 883, 1997.
- A. Hebert. Variational principles and convergence acceleration strategies for the neutron diffusion equation. *Nuclear Science and Engineering*, 91:414–427, 1985.
- A. Hebert. Preconditioning the power method for reactor calculations. *Nuclear Science and Engineering*, 94:1–11, 1986.
- D. Knoll and D. Keys. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193:357–397, 2004.
- E. Lewis and W. Miller. *Computational methods of neutron transport*. John Wiley and Sons, 1984.
- I. Pazsit and C. Demaziere. Noise Techniques in Nuclear Systems, In: *The Handbook of Nuclear Engineering*. vol. 3, Springer, Boston, MA, 2010.
- Y. Saad. *Numerical methods for large eigenvalue problems*. Society of Industrial and Applied Mathematics, 2nd edition, 2011.
- Y. Saad and M. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.
- J. Shadid, R. Tuminaro, and H. Walker. An inexact Newton method for fully coupled solution of the Navier-Stokes equations with heat and mass transport. *J. Comput. Phys*, 137:155–185, 1997.

J. Thie. *Power Reactor Noise*. American Nuclear Society, La Grange Park, IL, USA, 1981.

R. Tuminaro, H. Walker, and J. Shadid. On backtracking failure in Newton-GMRES methods with a demonstration for the Navier-Stokes equations. *J. Comput. Phys.*, 180:549–558, 2002.

M. Williams. *Random Processes in Nuclear Reactors*. Pergamon Press, Oxford, UK, 1974.

## Nomenclature

$\beta$	Delayed neutron fraction
$\mathbf{r}$	Vector of spatial coordinates
$\iota$	Imaginary unit
$\lambda$	Decay constant of delayed neutron precursor
$\nu$	Neutron speed
$\omega$	Angular frequency
$\phi$	Neutron flux
$\Sigma_\alpha$	Absorption macroscopic cross-section
$\Sigma_f$	Fission macroscopic cross-section
$\Sigma_r$	Removal macroscopic cross-section
$\Sigma_{s,1\rightarrow 2}$	Downscattering macroscopic cross-section
$\Sigma_{s,2\rightarrow 1}$	Upscattering macroscopic cross-section
$\nu$	Average number of neutrons released per fission
$C$	Delayed neutron precursor density
$D$	Neutron diffusion coefficient
$E$	Neutron energy
$k_{eff}$	Effective multiplication factor