CORTEX

Research and Innovation Action (RIA)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 754316.

Start date : 2017-09-01 Duration : 48 Months
http://cortex-h2020.eu

Modelling of the neutron flux response to vibrating fuel assemblies

Authors : Mr. Antoni VIDAL-FERRÀNDIZ (UPV), Christophe DEMAZIÈRE (Chalmers), Abdelhamid DOKHANE (PSI), Damián GINESTAR (UPV), Alexander KNOSPE (TUD), Matthias KUENTZEL (GRS), Antonios MYLONAKIS (Chalmers), Yann PÉRIN (GRS), Carsten LANGE (TUD), Gumersindo VERDÚ (UPV), Vasudha VERMA (PSI), Paolo VINAI (Chalmers)

CORTEX - Contract Number: 754316

Project officer: Marco Carbini

Document title	Modelling of the neutron flux response to vibrating fuel assemblies
Author(s)	Mr. Antoni VIDAL-FERRÀNDIZ, Christophe DEMAZIÈRE (Chalmers), Abdelhamid DOKHANE (PSI), Damián GINESTAR (UPV), Alexander KNOSPE (TUD), Matthias KUENTZEL (GRS), Antonios MYLONAKIS (Chalmers), Yann PÉRIN (GRS), Carsten LANGE (TUD), Gumersindo VERDÚ (UPV), Vasudha VERMA (PSI), Paolo VINAI (Chalmers)
Number of pages	111
Document type	Deliverable
Work Package	WP01
Document number	D1.3
Issued by	UPV
Date of completion	2020-12-15 09:34:32
Dissemination level	Public

Summary

This deliverable shows different types of methodologies dedicated to the study of the influence of the mechanical vibrations of fuel assemblies and the core barrel on the neutron flux in nuclear reactors. These mechanical vibrations play a crucial role in reactor neutron noise, i.e. the small, stationary fluctuations of the neutron flux around the average value that occur in nuclear reactors and that can be used for core monitoring and diagnostics. The methodologies presented in the current document employ the diffusion approximation to simulate the influence of the mechanical vibrations in the full core, using time-domain or frequency-domain approaches. Some methodologies also include thermal hydraulic feedback, mechanical models and neutron transport homogenization results inside their computations. Also, different ways to model the neutron noise source related with the mechanical vibrations are presented. Numerical results show coherent results between different strategies and known experimental data.

Approval

Date	By
2020-12-15 10:00:05	Mr. Paolo VINAI (Chalmers)
2020-12-15 10:03:35	Pr. Christophe DEMAZIERE (Chalmers)

Table of Contents

This project has received funding from the Euratom research and training programme 2014-2018 under the grant agreement no.754316. The content in this report reflects only the views of the authors. The European Commission is not responsible for any use that may be made of the information it contains.

Index of Tables

Table of Figures

Abbreviations

APSD	Auto Power Spectral Density
BOC	Beginning Of Cycle
BWR	Boiling Water Reactor
CFT	Continuous Fourier Transform
COLIBRI	CROCUS Oscillator for Lateral Increase Between U-metal Rods and Inner zone
CORTEX	Core monitoring techniques and experimental validation and demonstration
CV	Coefficient of Variation
DoFs	Degrees of Freedom
ECCP	Electronic Collaborative Content Platform
EOC	End Of Cycle
FA	Fuel Assembly
FEM	Finite Element Method
FED	Finite Element Degree
FFT	Fast Fourier Transform
FSI	Fluid Structure Interaction
GCSM	General Control and Simulation Module
GMRES	Generalized Minimal Residual method
GRS	Gesellschaft für Anlagen- und Reaktorsicherheit
HZDR	Helmholtz-Zentrum Dresden-Rossendorf
ILU	Incomplete Lower-Upper decomposition
JFNK	Jacobian Free Newton Krylov
MOC	Middle Of Cycle
ORNL	Oak Ridge National Laboratory
PARCS	the Purdue Advanced Reactor Core Simulator
PSI	Paul Scherrer Institute
PWR	Pressurized Water Reactor
RPV	Reactor Pressure Vessel
SCALE	Standardized Computer Analyses for Licensing Evaluation
SGS	Symmetric Gauss-Seidel
SSP	Studsvik Scandpower
TRITON	Transport Rigor Implemented with Time-dependent Operation for Neutronic depletion
UPV	Universitat Politècnica de València

Summary

This deliverable shows different types of methodologies dedicated to the study of the influence of the mechanical vibrations of fuel assemblies and the core barrel on the neutron flux in nuclear reactors. These mechanical vibrations play a crucial role in reactor neutron noise, i.e. the small, stationary fluctuations of the neutron flux around the average value that occur in nuclear reactors and that can be used for core monitoring and diagnostics.

The methodologies presented in the current document employ the diffusion approximation to simulate the influence of the mechanical vibrations in the full core, using time-domain or frequency-domain approaches. Some methodologies also include thermal hydraulic feedback, mechanical models and neutron transport homogenization results inside their computations. Also, different ways to model the neutron noise source related with the mechanical vibrations are presented. Numerical results show coherent results between different strategies and known experimental data.

1 Introduction

In the last decades, computational capabilities have improved enormously. More accurate nuclear simulation codes can be used in the field of nuclear reactor physics. However, a three-dimensional neutron transport solver, without any type of approximation, will require between 10^{17} and 10^{21} degrees of freedom, since the related neutron transport equation yields in a phase space of dimension seven: three for the location, three for the velocity (two for the angle and one for the energy) and one for the time [1].

To make reactor core simulations feasible with an effort that does not require high performance computing, different approximations of the system and of the time-dependent neutron transport equations can be introduced. The use of assembly-wise homogenized regions combined with the diffusion approximation allows to obtain fast and reliable neutronic simulations. In this way, the diffusion approximation using homogenized cross sections has been the main approach for the work discussed in this deliverable of the CORTEX project.

The deliverable is dedicated to the study of the influence of the mechanical vibrations of Fuel Assemblies (FAs) on the neutron flux in nuclear reactors. These mechanical vibrations are among the main reasons that induce reactor neutron noise, i.e. the small, stationary fluctuations of the neutron flux around the average value that occur in nuclear reactors. To model this effect, different codes are used in the present work, namely *FEMFFUSION*, *PARCS*, *FEMFFUSION-FD*, *CORESIM+*, *SIMULATE3/SIMULATE-3K*, *DYN3D* and *QUABOX-CUBBOX*. Table 1 shows a comparison of the main characteristics of these codes and the associated methodologies for the modelling of the effect of the neutron noise source.

The codes PARCS, *SIMUTLATE3/SIMULATE3K*, *DYN3D* and *QUABOX-CUBBOX* are general nuclear suites for the simulation of a wide variety of dynamic conditions inside an operating nuclear reactor, and they have been adapted to be able to model FA vibrations. Other codes have been developed specifically to simulate the effects of neutron noise sources, namely *FEMFFUSION-FD* and *CORESIM+*.

Some codes use a time-domain integration of the diffusion approximation varying the cross sections along the transient. Other codes like *CORESIM+* and *FEMFFUSION-FD* rely on a first-order perturbation approximation in the frequency-domain to model the effect of the vibrating assembly. This last methodology has been widely applied in the past to model small oscillating fluctuations in simplified problems [2,3].

The methodologies may differ for the modelling of the neutron noise source. One approach is to treat the cross section change provoked by the mechanical vibrations as a volume averaged change of the nuclear cross sections. In other words, the cross sections are changed interpolating the values of the homogenised region that moves inside other homogenised region. Another approach uses neutron transport codes such as *CASMO* or *SERPENT* to re-homogenize the regions according to the different positions of the fuel assembly during the vibration. In this way, they build a water gap model that identifies each position of the moving FA with a cross section. Unlike the first approach, the second one is not linear.

Table 1: Codes and methodologies used for fuel assembly vibrations.

Code	Domain	Neutron Equation	Cross-Section modelling of the noise source	Numerical examples	Organization
FEMFFUSION	Time Domain	Diffusion	Volume Averaged	2D FA vibrations	UPV
PARCS	Time Domain	Diffusion	Volume Averaged	2D FA vibrations	UPV
FEMFFUSION-FD	Frequency Domain	Diffusion	Volume Averaged	2D FA vibrations	UPV
CORESIM+	Frequency Domain	Diffusion	Volume Averaged	FA vibrations and Core Barrel vibrations	Chalmers
SIMULATE3 / SIMULATE-3K	Time Domain	Diffusion	Water gap model CASMO-5	FA vibrations and Core barrel vibrations	PSI
DYN3D	Time domain	Diffusion	Water gap Model CASMO-5	FA vibrations and Core barrel vibrations	TUD
QUABOX-CUBBOX	Time domain	Diffusion	Water gap Model SCALE	Core barrel vibrations	GRS

The rest of this deliverable is structured according to the contributions of the different partners involved in the work. Section 2 summarizes the efforts of Universitat Politècnica de València (UPV), using FEMFFUSION, FEMFFUSION-FD and PARCS. Section 3 discusses the development and application of CORE SIM+ that was carried out by Chalmers University of Technology. Section 4 presents the methodology based on SIMULATE3/SIMULATE-3K that was employed by Paul Scherrer Institute (PSI). Sections 5 explains the modelling strategy followed by Technische Universität Dresden (TUD) and the results obtained from DYN3D simulations. Section 6 explains the work performed at Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) with the code QUABOX/CUBBOX. Section 7 provides the main conclusions.

2 Modelling strategy at UPV

The mechanical vibrations of core internals as fuel assemblies cause oscillations in the neutron flux that require, in some circumstances, to operate nuclear power plants at a reduced power level. This work simulates and analyses the changes of the neutron flux throughout a nuclear core due to the oscillation of a single fuel assembly without considering thermal-hydraulic feedback. The amplitude of the fuel assembly vibration is bounded to a few millimetres and this implies the use of fine meshes and accurate numerical solvers due to the different scales of the problem. The results of the simulations show a main oscillation of the neutron flux with the same frequency as the fuel assembly vibration along with other harmonics at multiples of the vibration frequency much smaller in amplitude. Also, this work compares time-domain analysis and frequency-domain analysis of the mechanical vibrations. Numerical results show a close match between these two approaches for the fundamental frequency.

The methodology used at Universitat Politècnica de València (UPV) for the vibrating assembly simulations is based on the use of the neutron diffusion equation in the approximation of two energy groups. The spatial discretization of this equation is done by means of a high-order finite element method that allows to use different kind of meshes and thus study nuclear reactors with rectangular and hexagonal geometries.

2.1 Introduction

In the area of core monitoring, detection of reactor perturbations gives the possibility to take proper actions before such problems lead to safety concerns or impact plant availability. The CORTEX project [4], aims at developing an innovative core monitoring technique that allows detecting anomalies in nuclear reactors, such as excessive vibrations of core internals, flow blockage and coolant inlet perturbations. The technique is based on using the fluctuations of the neutron flux around its steady state value, known as neutron noise, recorded by in-core and ex-core detectors. The main benefit of this method is that it can be applied on-line without disrupting plant operation. Furthermore, the method is non-intrusive, i.e. no additional perturbation needs to be introduced.

A correlation between the stiffness of fuel assemblies (FA) and neutron noise levels was experimentally demonstrated [5], indicating the direct effect of FA vibrations onto the neutron noise. The work discussed in this Section assesses the capability of time-domain and frequency-domain simulations to study the spatial dependence of the neutron noise caused by FA vibrations. Also, it evaluates the magnitude and spatial shape of the neutron noise when a single FA is vibrating. Reactor core instrumentation is capable of detecting a small periodic variation at a specific frequency and it can be differentiated from the random noise in the detectors' signals (white noise) [6], [7]. The detection of neutron noise nevertheless requires a specific signal processing, both using hardware and software: removal of the mean value, amplification, anti-aliasing filters, and analog-to-digital conversion. The analysis of neutron noise is then most easily and most efficiently performed in the frequency-domain, where peaks in the spectra can be identified and subsequently analysed. FA vibrations has received some particular interest in the CORTEX project, with the COLIBRI experiments in the CROCUS reactor [8]. These experiments mimic fuel assembly vibrations by displacing fuel rods in an oscillatory manner. These experiments also demonstrate that such vibrations induce a neutron noise measurable by the core instrumentation.

The present contribution simulates the movement of a FA as a sinusoidal displacement of the boundary between adjacent homogenized regions. Other studies considered FA vibrations as random modification of the size of the water gaps which surrounds the FA of interest [9]. Time-domain and frequency-domain comparisons of different perturbations have been reported lately in [10], [11], [12] .

FA vibrations have been studied from a mechanical point of view in [13] and [6]. Natural frequencies range from 0.8 Hz to 24.5 Hz depending on the idealized form of bearing and their building materials. The amplitude of the vibration ranges can reach up to 1 mm. However, there can be amplitudes greater than 1 mm in case of synchronous motions. This work is based on a single FA vibration of 1 Hz and 1 mm of amplitude as a typical example.

This section is organized as follows. First, the time-domain analysis is explained in Section 2.2. Section 2.3 describes the frequency-domain analysis and the usual approximation to model a mechanical FA vibration. Then, numerical results for a two-dimensional model are given in Section 2.4 to compare the proposed methodologies.

2.2 Time-domain analysis

2.2.1 FEMFFUSION – A finite element approach

One possible approach to simulate transients associated with moving parts inside the reactor core is to use the time dependent neutron diffusion equation, which is an approximation of the neutron transport equation widely used to study the behaviour of nuclear reactors. The neutron diffusion equation assumes that the neutron current is proportional to the gradient of the neutron flux by means of a diffusion coefficient. This approximation is analogous to Fick's law in species diffusion and to Fourier's law in heat transfer. Although a transport-based solution without spatial homogenization would be required to properly catch the effect of local noise sources, some studies revealed that a diffusion-based solution correctly represents a transport-based solution a few mean free paths away from the noise source and from strong heterogeneities [14].

Particularly, the two energy groups approximation of this equation is considered, assuming that fission neutrons are born in the fast group and there is no up-scattering [15]. This model is of the form

$$[\mathrm{v}^{-1}]\frac{\partial \Phi}{\partial t} + \mathcal{L}\Phi = (1-\beta)\mathcal{M}\Phi + \sum_{k=1}^{K} \lambda_k \chi \mathcal{C}_k\,, \tag{1}$$

$$\frac{\partial \mathcal{C}_k}{\partial t} = \beta_k \left[\nu\Sigma_{f1}\ \nu\Sigma_{f2}\right]\Phi - \lambda_k \mathcal{C}_k, \qquad k = 1, \dots, K \tag{2}$$

where K is the number of delayed neutron precursors groups considered and the matrices are defined as

$$[\mathrm{v}^{-1}] = \begin{pmatrix} \frac{1}{\mathrm{v}_1} & 0 \\ 0 & \frac{1}{\mathrm{v}_2} \end{pmatrix}, \quad \mathcal{L} = \begin{pmatrix} -\vec{\nabla}\cdot\left(D_1\vec{\nabla}\right) + \Sigma_{a1} + \Sigma_{12} & 0 \\ -\Sigma_{12} & -\vec{\nabla}\cdot\left(D_2\vec{\nabla}\right) + \Sigma_{a2} \end{pmatrix},$$

$$\mathcal{M} = \begin{pmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \\ 0 & 0 \end{pmatrix}, \quad \Phi = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}, \quad \chi = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

where ϕ_1 and ϕ_2 are the fast and thermal neutron fluxes, respectively. The diffusion constants D_g and the cross-sections Σ_{12}, Σ_{ag} and $\nu\Sigma_{fg}$, with $g = 1, 2$, depend on the reactor materials, that is, they are position and time dependent functions. The parameter β_k is the yield of delayed neutrons in the k-th precursors group and λ_k is the corresponding decay constant. Both coefficients are related to the delayed neutron precursor decay. Other quantities have their usual notation, see for example [15].

For the spatial discretization of the neutron diffusion equation a high order *hp*-finite element method has been used [16], so that reactors with any kind of geometry can be modelled. The main characteristic of this method is that allows to increase the accuracy of the solution refining the spatial mesh (*h*-adaptivity) and also increasing the degree of the polynomial expansions used in the finite element method (*p*-adaptivity) allowing to obtain solutions to the problem with high accuracy in a reasonable amount of time. The *h-p* finite element method has been implemented using the open source finite elements library *deal.II* [17]. With the help of the library, the proposed code is dimension independent and can manage different cell sizes and different types of finite elements [18]. In order to solve the algebraic problems resulting from the discretizations, the numerical libraries *PETSc* [19], and *SLEPc* [20], have been used.

Different methods have been proposed for the time discretization of the time-dependent neutron diffusion equation [21]. Standard methods use backward difference formulas [22]. These methods, for each time step, need to solve a system of linear equations, which is large and sparse. Preconditioned iterative methods are used to solve these systems [23], [24]. Other kinds of methods such as modal methods [25], [26] or the quasi-static method [27] have been also used in nuclear engineering.

The finite element method (FEM) code developed at UPV to solve the time-dependent neutron diffusion equation with vibrating elements, is an option of a general neutronic code, *FEMFFUSION* [28]. FEMFFUSION is an open-source C++ neutronic code that solves the multigroup neutron transport equation using the diffusion approximation or the SP_N approximation. The code uses the continuous Galerkin finite element method to be able to deal with any type of geometry and any problem dimension (1D, 2D and 3D problems). It works on top of deal.II library [17], which provides supporting and advances in the finite element method. It also helps to include state-of-the-art techniques in partial differential equation resolution, linear algebra and computer science.

The main features of FEMFFUSION are:

- Open-source software (released under the terms of the GNU GPL version 3).
- Use of the FEM to solve the multigroup neutron diffusion (or SPN) equations.
- Use of matrix-free technique to maintain reasonable memory demands.
- Possibility to use or easily implement a variety of eigenvalue solvers and preconditioners.
- Valid for all type of geometries, rectangular, hexagonal, pin-level and unstructured.
- Possibility to import unstructured grids form GMsh.
- Capacity to solve problems in 1D, 2D and 3D.
- Can solve for either the direct or the adjoint flux and several eigenpairs.
- Output provide the effective multiplication factor k_{eff}, the map of the averaged neutron power per assembly and of the fluxes. Also standard .vtk files are provided among other selected outputs.
- Interact with high-quality open-source libraries: deal.II, PETSc, SLEPc, Sundials...
- Easy interface with plotting and post-processing tools (MatLab, ParaView, Matplotlib...).
- Well documented and easy to extend to related problems.

Also, there is an option to solve the time dependent neutron diffusion equation that allows treating neutron noise problems and other neutronic transients. A new feature (FEMFFUSION-FD) has been included that allows to study neutron noise problems in the frequency-domain as Section 2.3 explains. The code is openly available at https://bitbucket.org/Zonni/femffusion/src/master/.

2.2.1.1 Spatial discretization

For a given transient analysis in a reactor core, usually, a static configuration of the reactor is considered as initial condition. Associated with the time dependent neutron diffusion equation, (1) and (2), there is the generalized eigenvalue problem:

$$\mathcal{L}\Phi = \frac{1}{\lambda}\mathcal{M}\Phi\,. \tag{3}$$

This problem is known as the *Lambda Modes problem* for a given configuration of the reactor core [29]. The fundamental eigenvalue (the largest one) is called the *k*-effective of the reactor core. This eigenvalue and its corresponding eigenfunction describe the steady state neutron distribution in the core. In this way, the calculation of the stationary neutron flux distribution is the first step for any transient analysis. To solve the problem defined by (1) and (2), and the problem defined by (**3**), a spatial discretization of the equations has to be selected. In this work, a high order Galerkin finite element method [30] is used.

The weak formulation of this equation is obtained by pre-multiplying by a test function $\left(\varphi^T = (\varphi_1, \quad \varphi_2)\right)$ and integrating over the domain Ω that corresponds to the reactor core volume,

$$\begin{aligned}&\int_\Omega (\varphi_1 \quad \varphi_2)\begin{pmatrix} -\vec{\nabla}\left(D_1\vec{\nabla}\right) + \Sigma_{a1} + \Sigma_{12} & 0 \\ -\Sigma_{12} & -\vec{\nabla}\left(D_2\vec{\nabla}\right) + \Sigma_{a2}\end{pmatrix}\begin{pmatrix}\phi_1 \\ \phi_2\end{pmatrix} \mathrm{d}V = \\ &= \frac{1}{\lambda}\int_\Omega (\varphi_1 \quad \varphi_2)\,(\nu\Sigma_{f1} \quad \nu\Sigma_{f2})\begin{pmatrix}\phi_1 \\ \phi_2\end{pmatrix} \mathrm{dV}\,.\end{aligned} \tag{4}$$

The vectorial identity, $\vec{\nabla}\cdot\left(u\vec{\nabla}v\right) = \left(\vec{\nabla}u\right)\cdot\left(\vec{\nabla}v\right) + u\left(\vec{\nabla}\cdot\vec{\nabla}v\right)$, is applied and expression (4) is rewritten as

$$\begin{aligned}&\int_\Omega \vec{\nabla}\varphi_1 D_1\vec{\nabla}\phi_1 dV - \int_\Omega \vec{\nabla}\cdot\left(\varphi_1 D_1\vec{\nabla}\phi_1\right) dV + \int_\Omega \varphi_1(\Sigma_{a1} + \Sigma_{12})\phi_1 dV \\ &+ \int_\Omega \vec{\nabla}\varphi_2 D_2\vec{\nabla}\phi_2 \mathrm{d}\,V - \int_\Omega \vec{\nabla}\cdot\left(\varphi_2 D_2\vec{\nabla}\phi_2\right) dV + \int_\Omega \varphi_2\Sigma_{a2}\phi_2 dV - \int_\Omega \varphi_2\Sigma_{12}\phi_1 dV = \\ &= \frac{1}{\lambda}\left(\int_\Omega \varphi_1\nu\Sigma_{f1}\phi_1 dV + \int_\Omega \varphi_1\nu\Sigma_{f2}\phi_2 \mathrm{d}\,V\right).\end{aligned} \tag{5}$$

Using Gauss Divergence theorem $\left(\int_\Omega \vec{\nabla}\cdot\vec{F}\,dV = \int_\Gamma \vec{F}d\vec{S}\right)$ to eliminate second order derivatives,

$$\begin{aligned}&\int_\Omega d\varphi_1 D_1\vec{\nabla}\phi_1 dV - \int_\Gamma \varphi_1 D_1\vec{\nabla}\phi_1 d\vec{S} + \int_\Omega \varphi_1(\Sigma_{a1} + \Sigma_{12})\phi_1 dV \\ &+ \int_\Omega \vec{\nabla}\varphi_2 D_2\vec{\nabla}\phi_2 dV - \int_\Gamma \varphi_2 D_2\vec{\nabla}\phi_2 d\vec{S} + \int_\Omega \varphi_2\Sigma_{a2}\phi_2 dV - \int_\Omega \varphi_2\Sigma_{12}\phi_1 \mathrm{d}\,V \\ &= \frac{1}{\lambda}\left(\int_\Omega \varphi_1\nu\Sigma_{f1}\phi_1 dV + \int_\Omega \varphi_1\nu\Sigma_{f2}\phi_2 dV\right),\end{aligned} \tag{6}$$

is obtained, where Γ is the boundary of the domain defining the reactor core.

Finally, the reactor domain Ω is divided into cell subdomains $\Omega_e, (e = 1, \dots, N_t)$ where it is assumed that the nuclear cross sections remain constant. Γ_e is also defined as the corresponding subdomain surface which is part of the reactor frontier Γ. Equation (6) is rewritten as

$$\begin{aligned}&\sum_{e=1}^{N_t}\left(D_1\int_{\Omega_e}\vec{\nabla}\varphi_1\vec{\nabla}\phi_1 dV - D_1\int_{\Gamma_e}\varphi_1\vec{\nabla}\phi_1 d\vec{S} + (\Sigma_{a1}+\Sigma_{12})\int_{\Omega_e}\varphi_1\phi_1 dV\right.\\ &+D_2\int_{\Omega_e}\vec{\nabla}\varphi_2\vec{\nabla}\phi_2 dV - D_2\int_{\Gamma_e}\varphi_2\vec{\nabla}\phi_2 d\vec{S} + \Sigma_{a2}\int_{\Omega_e}\varphi_2\phi_2 dV\\ &\left.-\Sigma_{12}\int_{\Omega_e}\varphi_2\phi_1 dV\right) = \frac{1}{\lambda}\sum_{e=1}^{N_t}\left(\nu\Sigma_{f1}\int_{\Omega_e}\varphi_1\phi_1 dV + \nu\Sigma_{f2}\int_{\Omega_e}\varphi_1\phi_2 dV\right).\end{aligned}\tag{7}$$

There are several surface integrals over the boundary of the subdomains (Γ_e) that rely on the boundary conditions [16]. The solution ϕ_g is approximated through a trial solution assumed as a sum of shape functions N_{gj} multiplied by their corresponding nodal values $\tilde{\phi}_{gj}$.

$$\phi_g \approx \sum_j N_{gj}\tilde{\phi}_{gj}\,. \tag{8}$$

Introducing expression (8) in (7) and eliminating redundant coefficients to obtain continuous solutions (see, for example, [31] for more details) in terms of global coefficients, the procedure leads to an algebraic eigenvalue problem of the form

$$L\begin{pmatrix}\tilde{\phi}_1\\ \tilde{\phi}_2\end{pmatrix} = \frac{1}{\lambda}\,M\begin{pmatrix}\tilde{\phi}_1\\ \tilde{\phi}_2\end{pmatrix}, \tag{9}$$

where the matrices elements are given by

$$\begin{aligned}L_{ij} &= \sum_{e=1}^{N_t}\left(D_1\int_{\Omega_e}\vec{\nabla}N_{1i}\vec{\nabla}N_{1j}dV - D_1\int_{\Gamma_e}N_{1i}\vec{\nabla}N_{1j}d\vec{S} + (\Sigma_{a1}+\Sigma_{12})\int_{\Omega_e}N_{1i}N_{1j}dV\right.\\ &\left.+D_2\int_{\Omega_e}\vec{\nabla}N_{2i}\vec{\nabla}N_{2j}dV - D_2\int_{\Gamma_e}N_{2i}\vec{\nabla}N_{2j}d\vec{S} + \Sigma_{a2}\int_{\Omega_e}N_{2i}N_{2j}dV - \Sigma_{12}\int_{\Omega_e}N_{2i}N_{1j}dV\right),\\ M_{ij} &= \sum_{e=1}^{N_t}\left(\nu\Sigma_{f1}\int_{\Omega_e}N_{1i}N_{1j}dV + \nu\Sigma_{f2}\int_{\Omega_e}N_{1i}N_{2j}dV\right).\end{aligned}\tag{10}$$

These integrals only have non-zero value when the shape functions N_i and N_j belong to the same cell, therefore highly sparse global matrices are obtained. More details on the spatial discretization used can be found in [16]. To solve the algebraic eigenvalue problem (9) a Krylov-Schur method is used from the *SLEPc* library [20]. The generalized eigenvalue problem is reduced to an ordinary eigenvalue problem of the form,

$$\mathrm{L}_{11}^{-1}(\mathrm{M}_{11}+\mathrm{M}_{12}\mathrm{L}_{22}^{-1}\mathrm{L}_{21})\tilde{\phi}_1 = \lambda\,\tilde{\phi}_1, \tag{11}$$

which is solved for the eigenvalue of maximum magnitude (k_{eff}) and its corresponding eigenvector. In this way, for each matrix-vector product it is necessary to solve two linear systems associated with L_{11} and L_{22}, to avoid the calculation of their inverse matrices. These systems are solved by means of an iterative scheme as the preconditioned Conjugate Gradient method [32]. Particularly, a Cuthill-McKee reordering is performed to reduce the bandwidth of the matrices, together with an incomplete Cholesky factorization used for the preconditioning.

Other possibilities of solving the partial eigenvalue problem (9), without the necessity of reducing it to an ordinary eigenvalue problem are presented in [33] and [34].

2.2.1.2 Time discretization

Once the spatial discretization has been selected, a discrete version of the time dependent neutron diffusion equation is solved. Since the system of ordinary differential equations resulting from the spatial discretization of the neutron diffusion equations is stiff, implicit methods are necessary. Particularly, a first order backward method is used [22]. This method is needed at each time step to solve a large system of linear equations.

Once the spatial discretization is performed, the semi-discrete two energy groups time dependent neutron diffusion equation together with the neutron precursors concentration equations are of the form

$$[\tilde{\mathrm{v}}^{-1}]\frac{d\Phi}{dt} + L\Phi = (1-\beta_{\mathrm{eff}})M\Phi + \sum_{k=1}^{K}\lambda_k X\, C_k\,, \tag{12}$$

$$P\frac{dC_k}{dt} = \beta_k(M_{11}\quad M_{12})\Phi - \lambda_k P C_k\,, \qquad k = 1, \dots, K, \tag{13}$$

where L and M are the matrices obtained from the spatial discretization of operators $\mathcal{L}$ and $\mathcal{M}$. Φ and C_k are the vector of coefficients of the neutron flux and the precursors concentration in terms of the polynomials used in the FEM. M_{11}, M_{12} are the non-zero block matrices of the matrix M. The matrices X and $[\tilde{\mathrm{v}}^{-1}]$ are defined as

$$X = \begin{pmatrix}\mathrm{P}\\ 0\end{pmatrix}, \qquad [\tilde{\mathrm{v}}^{-1}] = \begin{pmatrix}P\,\mathrm{v}_1^{-1} & 0\\ 0 & P\,\mathrm{v}_2^{-1}\end{pmatrix},$$

where the matrix P is the mass matrix of the spatial discretization, which appears due to the fact that the polynomial basis used in the spatial discretization is not orthogonal. The matrix elements of P are given by

$$P_{ij} = \sum_{e=1}^{N_t}\int_{\Omega_e} N_i N_j dV\,. \tag{14}$$

The time discretization of the precursor's equations (13), is done using a one-step implicit finite differences scheme. To obtain this scheme, we make the change of function

$$PC_k = e^{-\lambda_k t}B_k\,, \tag{15}$$

obtaining

$$\frac{dB_k}{dt} = e^{\lambda_k t}\beta_k(M_{11}\quad M_{12})\widetilde{\Phi}(t). \tag{16}$$

Integrating between t_n and t,

$$B_k(t) = B_k^n + \int_{t_n}^{t} e^{\lambda_k\tau}\beta_k(M_{11}\quad M_{12})\widetilde{\Phi}(\tau)\,d\tau. \tag{17}$$

Making use of the change (17), C_k^{n+1} can be expressed as

$$PC_k^{n+1} = e^{-\lambda_k\Delta t_n}PC_k^n + e^{-\lambda_k t_{n+1}\int_{t_n}^{t_{n+1}} e^{\lambda_k\tau}\,\beta_k(M_{11}\quad M_{12})\widetilde{\Phi}(\tau)\,d\tau}\,. \tag{18}$$

where $\Delta t = t_{n+1} - t_n$. The term $(M_{11}\quad M_{12})\,\widetilde{\Phi}(t)$ inside the integral is approximated by its value at the instant t_{n+1} obtaining

$$PC_k^{n+1} = PC_k^n e^{-\lambda_k\Delta t} + \frac{\beta_k}{\lambda_k}\left(1 - e^{-\lambda_k\Delta t}\right)(M_{11}^{n+1}\quad M_{12}^{n+1})\,\Phi^{n+1}.$$

In the same way, Euler's backward method is used in equation (12) obtaining,

$$[\tilde{v}^{-1}]\frac{1}{\Delta t}\left(\tilde{\Phi}^{n+1}-\tilde{\Phi}^{n}\right)+L^{n+1}\tilde{\Phi}^{n+1}=(1-\beta_{\text{eff}})M^{n+1}\tilde{\Phi}^{n+1}+\sum_{k=1}^{K}\lambda_k\,XC_k^{n+1}.$$

Taking into account equation (18), equation (12) is rewritten as the system of linear equations:

$$T^{n+1}\Phi^{n+1}=R^{n}\Phi^{n}+\sum_{k=1}^{K}\lambda_k e^{-\lambda_k\Delta t}XC_k^{n}\ , \qquad (19)$$

where the matrices are defined as,

$$T^{n+1}=\frac{1}{\Delta t}[\tilde{v}^{-1}]+L^{n+1}-\hat{a}\,M^{n+1},$$

$$R^{n}=\frac{1}{\Delta t}[\tilde{v}^{-1}]=\frac{1}{\Delta t}\begin{pmatrix}P\,v_1^{-1} & 0\\ 0 & P\,v_2^{-1}\end{pmatrix},$$

and the coefficient $\hat{a}$ is

$$\hat{a}=1-\beta_{\text{eff}}+\sum_{k=1}^{K}\beta_k\left(1-e^{\lambda_k\Delta t}\right)\ ,$$

where $\beta_{\text{eff}}=\sum_{k=1}^{K}\beta_k$. This system of equations must be solved for each time step. The preconditioned BICGSTAB method [32] has been chosen to solve these systems and the preconditioner used has been the incomplete LU preconditioner together with a reordering of the matrix to decrease the fill-in. This system of equations is large and sparse and must be solved for each new time step with a high accuracy, e.g. $\text{tol}=10^{-12}$.

The time step Δt is maintained fixed during all the transient. Small time steps are usually used to solve the neutron diffusion equation when the $[v^{-1}]\frac{\partial\Phi}{\partial t}$ term of equation (12) must be accurately represented. However, this term is important only when the neutron flux changes at least one decade over a millisecond. This is not the case of neutron noise analysis calculations. In this way, a time step of $\Delta t=10^{-2}\ s$ has been proven to give accurate results, although UPV has used a time step of $\Delta t=10^{-4}\ s$ to ensure the convergence of the results.

2.2.1.3 Fuel Assembly vibration simulation in the time-domain

The simulation of the assembly vibration is done by changing in time the homogenized cross-sections of the corresponding materials using a simple volume averaged method. This method for a 1D geometry is schematized in Figure 1.

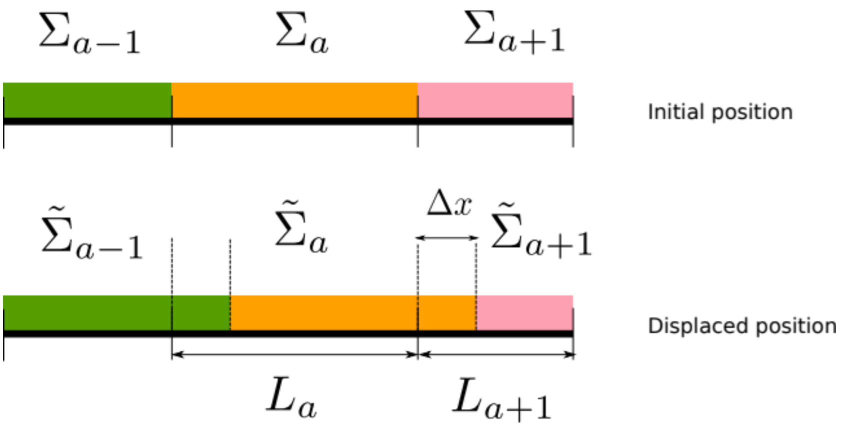


Figure 1: Schematic disposition for a 1D displacement of a vibrating assembly.

The movement of the assembly is taken into account changing the value of the homogenized cross-sections accordingly to equations (20), (21) and (22):

$$\tilde{\Sigma}_{\mathrm{a}-1} = \Sigma_{a-1}, \tag{20}$$

$$\tilde{\Sigma}_{a+1} = \frac{1}{L_a}\big(\Sigma_{a-1}\Delta x + \Sigma_a(L_a - \Delta x)\big), \tag{21}$$

$$\tilde{\Sigma}_{a+1} = \frac{1}{L_{a+1}}\big(\Sigma_a\Delta x + \Sigma_{a+1}(L_{a+1} - \Delta x)\big). \tag{22}$$

A similar model is considered for 2D and 3D rectangular geometries.

Finally, to compare the results obtained from the time-domain analysis with the ones obtained with the frequency-domain methodology we define the neutron noise in the time-domain as:

$$\delta\phi(r,\, t) = \phi(r,\, t) - \phi(r,\, 0). \tag{23}$$

Then, we can apply a Fourier transform to the neutron noise in the time-domain,

$$\delta\phi(r,\, \omega) = \mathcal{F}[\delta\phi(r,\, t)] = \int_{-\infty}^{\infty} \exp(-i\omega t)\, \delta\phi(\boldsymbol{r},\, \omega)\mathrm{d}t\,. \tag{24}$$

This transform is calculated with the Fast Fourier Transform algorithm (FFT) over the time-dependent results using a sampling time of 10^{-2} s and scaled accordingly to be able to compare the discrete Fourier transform results with the continuous Fourier transform results of Section 2.3. A summary of the methodology employed in *FEMFFUSION* for the analysis of transients in the time-domain is shown in Figure 2.

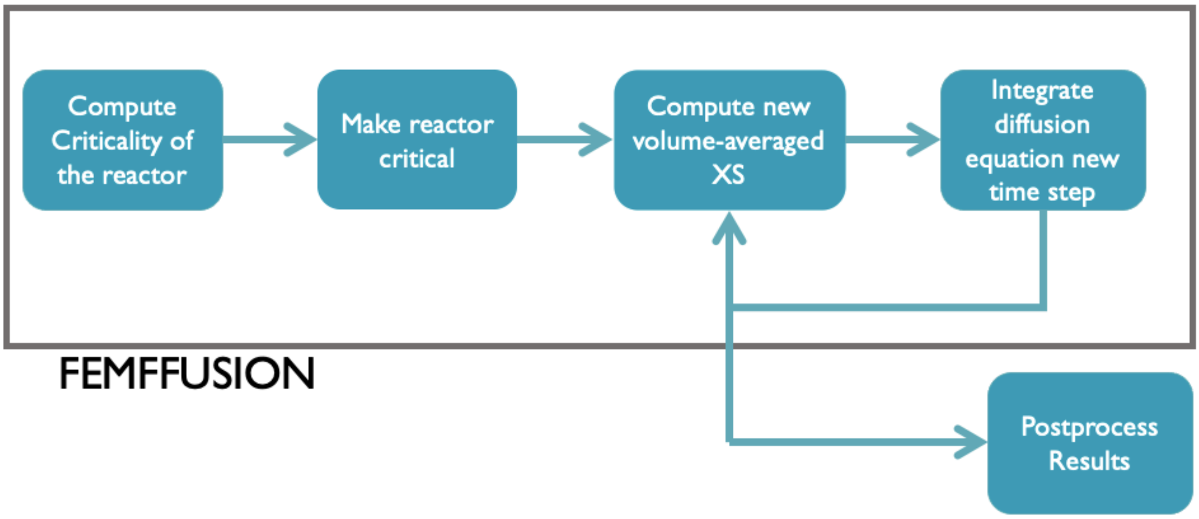

Compute Criticality of the reactor
Make reactor critical
Compute new volume-averaged XS
Integrate diffusion equation new time step
FEMFFUSION
Postprocess Results

Figure 2: Scheme of the methodology employed in FEMFFUSION.

The time-domain neutron simulator developed in this project, *FEMFFUSION*, is applied in different benchmark problems in Section 2.4. In Section 2.4.1, *FEMFFUSION* is compared against the frequency-domain noise code *CORE SIM* [35] for a basic one-dimensional example of a vibrating fuel assembly. In Section 2.4.2 the time-domain approach of *FEMFFUSION* is applied to a more realistic two-dimensional model and compared with the frequency-domain approach of *CORE SIM*.

Section 2.4.3 compares and verifies the time-domain approach of *FEMFFUSION* against the frequency-domain branch of this project, *FEMFFUSION-FD*. In this Section a generic absorber of variable strength perturbation is studied in a three-dimensional hexagonal reactor to show the application of a non-Cartesian mesh. Section 2.4.4 compares the time-domain solution of FEMFFUSION against the time-domain code, *PARCS*. All these examples allow verifying the results obtained with the time-domain code, *FEMFFUSION,* when simulating the noise response to FA vibrations.

2.2.2 *PARCS – A finite difference approach*

The core simulator PARCS [36] has been used to calculate the neutron noise induced by different types of perturbations. For this purpose, the code was modified so that user-defined time-dependent cross sections can be provided.

First, this optional methodology is selected using the label `CUSTOMXS input_file.xs,` in the main input file of PARCS. The name `input_file.xs` refers to the file where the user's cross sections are stored. An example of the main input file where the CUSTOMXS label is activated is shown in Table 2. This file can contain any type of perturbations to the critical conditions of the reactor: control-rod movements, change of the temperature in the reactor, fuel assembly vibrations, core barrel vibrations, etc. An example of the cross-section file `input_file.xs` is shown in Table 3. These files for the case of a FA vibration are written with the help of python scripts as the one shown in Table 13, for the COLIBRI experiments in the CROCUS reactor. The results of the analysis performed for this reactor will be presented in Section 2.4.4.

The PARCS code reads the initial cross sections according to the code in Table 14. Then, PARCS calculates the multiplicative factor and the steady-state neutron flux as usual. After that, the fission cross sections are changed to ensure that the reactor is initially critical in the time-dependent numerical simulation, as follow:

$$\nu\Sigma_{f1}^{*}(\vec{r},t) = \frac{\nu\Sigma_{f1}(\vec{r},t)}{k_{eff}}, \qquad \nu\Sigma_{f2}^{*}(\vec{r},t) = \frac{\nu\Sigma_{f2}(\vec{r},t)}{k_{eff}}.$$

Once the cross-sections are corrected with the k-effective, a new set of custom cross sections is used as input for the time-dependent neutron diffusion equation. Then, this equation is numerically integrated as PARCS usually does. Once the next time step is solved, the cross sections are updated from the values read from the input file for the following time step as Table 15 shows at file XSecFdbkM.f90. This procedure is repeated until the time-domain simulation is finished. Finally, the time-domain results are postprocessed to transform them into the frequency-domain using a FFT with external python scripts shown in Table 16.

Table 2: PARCS main input file for the two-dimensional CROCUS reactor where the CUSTOMXS tag is activated.

```
!******************************************************************************
CASEID '2D_CROCUS/2D_CROCUS_local_2'
!******************************************************************************
CNTL
      core_power   1.0
      TRANSIENT   T
      TH_FDBK F F
!                    input  iteration     planar                 adj
!                     edit       table      power       pin     reac
      print_opt          F          F          F          F          F
!                     fdbk        flux     planar
!                      rho     precurs       flux        Xe      T/H
      print_opt          F          F          T          F          F
      print_opt          F          F          F          F          F
!                                print_xs
      print_opt          F          F          F          F          F
!******************************************************************************
PARAM
 conv_ss     1.e-16 1.e-15 1.e-15 1.e-15   !epseig epsl2 epslinf epstf
 nodal_kern fdm
!******************************************************************************
XSEC
!******************************************************************************
 DNP_NGRP 8
 KIN_COMP 1 1 -1569
 DNP_BETA   2.23169E-04 1.08754E-03 6.38720E-04 1.41699E-03 2.43167E-03 8.08202E-04
6.70806E-04 2.53562E-04
 DNP_LAMBDA 1.24667E-02 2.82917E-02 4.25244E-02 1.33042E-01 2.92467E-01 6.66488E-01
1.63478E+00  3.55460E+00
 CUSTOMXS '2D_CROCUS/2D_CROCUS_local_2.xs'
 NEUT_VELO 16442122  295045
!******************************************************************************
GEOM
 file '2D_CROCUS/2D_CROCUS_local_2.gm'
!******************************************************************************
TRAN
 time_step 3.0 0.01 !tend delt0 tswitch texpand
 conv_tr 1e-15 1e-15  1e-15   1e-15 !epsr2  epsl2t  epslinft  epstft
 sum_edit F F F F
.
```

Table 3: Example of .xs file to insert custom cross section in PARCS. The cross sections that change are defined on the row indexes 9 and 10.

```
TIME_STEPS 4
N_COMP 20
TIME 0.0
1 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
2 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
3 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
4 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
5 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
6 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
7 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
8 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
9  0.2844 0.00876 0.00637629 0.00637629 0.03056 1.59262 0.06543 0.07288776 0.07288776
10 0.2844 0.00876 0.00637629 0.00637629 0.03056 1.59262 0.06543 0.07288776 0.07288776
11 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
12 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
13 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
14 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
15 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
16 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
17 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
18 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
19 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
20 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
TIME 0.01
1 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
2 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
3 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
4 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
5 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
6 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
7 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
8 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
9  0.2816 0.00463 0.00317380 0.00317380 0.04532 1.89876 0.04208 0.03627614 0.03627614
10 0.2816 0.00463 0.00317380 0.00317380 0.04532 1.89876 0.04208 0.03627614 0.03627614
11 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
12 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
13 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
14 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
15 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
16 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
17 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
18 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
19 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
20 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
TIME 0.02
1 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202-06 3.3202e-06
2 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
3 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
4 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
5 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
6 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
7 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
8 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
9  0.2844 0.00874 0.00635930 0.00635930 0.03064 1.59424 0.06531 0.07269346 0.07269346
10 0.2844 0.00874 0.00635930 0.00635930 0.03064 1.59424 0.06531 0.07269346 0.07269346
11 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
12 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
```

```
13 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
14 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
15 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
16 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
17 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
18 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
19 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
20 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
TIME 0.03
1 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
2 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
3 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
4 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
5 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
6 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
7 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
8 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
9 0.28162 0.00466 0.00319467 0.00319467 0.04522 1.89677 0.04224 0.03651474 0.03651474
10 0.2816 0.00466 0.00319467 0.00319467 0.04522 1.89677 0.04224 0.03651474 0.03651474
11 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
12 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
13 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
14 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
15 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
16 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
17 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
18 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
19 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
20 0.2788 0.00054 9.3632e-07 9.3632e-07 0.05994 2.20208 0.01895 3.3202e-06 3.3202e-06
```

Figure 3 shows the methodology employed in the PARCS code to solve the time-dependent diffusion equation with a custom set of time dependent cross-sections.

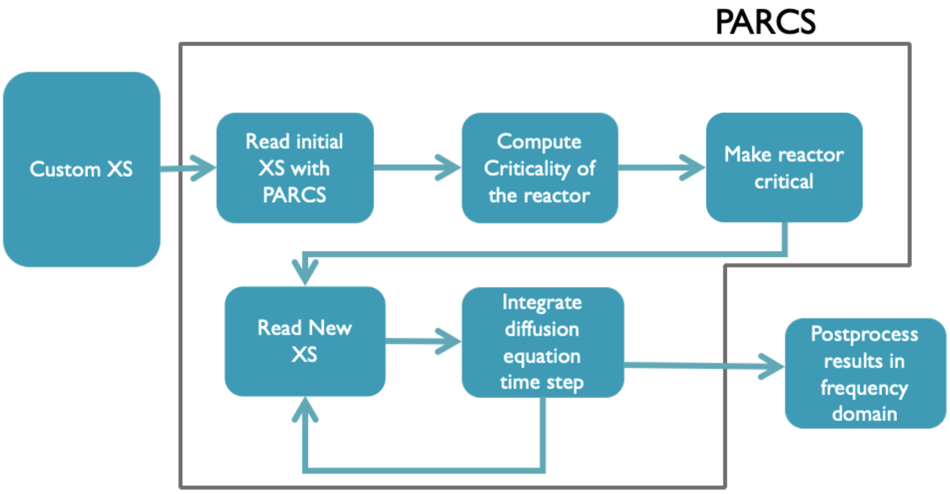

PARCS
Custom XS
Read initial XS with PARCS
Compute Criticality of the reactor
Make reactor critical
Read New XS
Integrate diffusion equation time step
Postprocess results in frequency domain

Figure 3: Scheme of the methodology employed in PARCS.

In Section 2.4.4, The experiments of the first COLIBRI campaign performed in the CROCUS reactor [8] were simulated to validate the *PARCS* approach against the time-domain *FEMFFUSION* calculations. Numerical results show a close agreement between the two time-domain strategies and the frequency-domain strategy. This agreement validates the strategies based on the two energy groups diffusion approximation.

2.3 Frequency-domain analysis

The neutron diffusion equation in the frequency-domain has been already successfully used to study other neutron noise sources [37] [38] [39]. This section discusses the frequency-domain tool developed by UPV based on FEMFFUSION and called FEMFFUSION-FD. This tool uses the usual two energy groups diffusion approximation in the frequency-domain in the first-order neutron noise approximation. This tool is similar to CORESIM [35] and CORESIM+ that will be described in Section 3.

The first-order neutron noise theory is based on separating every time-dependent term, expressed as $U(\vec{r},t)$, into their mean value, U_0, and their fluctuation around their mean value, δU_0 as

$$U(\vec{r},t) = U_0(\vec{r}) + \delta U(\vec{r},t). \tag{25}$$

The fluctuations are assumed to be small compared to the mean values. This allows to neglect second-order terms. Also, the fluctuations of the diffusion coefficients are neglected, so $\delta D_g = 0$ is assumed. Then, a Fourier transform is applied to the neutron diffusion equation. Thus, the first-order neutron noise equation can be written as [35]:

$$-\vec{\nabla}\cdot\left(D\vec{\nabla}\delta\phi(\vec{r},\,\omega)\right) + \Sigma_{\text{dyn}}\,\delta\phi(\vec{r},\,\omega) = \delta S(\vec{r},\,\omega) \tag{26}$$

The perturbation source term $\delta S(\vec{r},\,\omega)$ is given by the frequency-domain changes in the cross-sections:

$$\delta S(\vec{r},\,\omega) = \begin{pmatrix}\delta S_1(\vec{r},\,\omega)\\ \delta S_2(\vec{r},\,\omega)\end{pmatrix} = [\phi_s]\,\delta\Sigma_{12} + [\phi_a]\begin{pmatrix}\delta\Sigma_{a1}\\ \delta\Sigma_{a2}\end{pmatrix} + \frac{1}{k_{eff}}[\phi_f]\begin{pmatrix}\delta\nu\Sigma_{f1}\\ \delta\nu\Sigma_{f2}\end{pmatrix} \tag{27}$$

where

$$D = \begin{pmatrix}D_1 & 0\\ 0 & D_2\end{pmatrix}, \qquad \Sigma_{dyn} = \begin{pmatrix}-\Sigma_1 & \nu\Sigma_{f2}\left(1-\dfrac{i\omega\,\beta_{\text{eff}}}{i\omega+\lambda_{\text{eff}}}\right)\\ \Sigma_{12} & \left(\Sigma_{a2}+\dfrac{i\omega}{\mathrm{v}_2}\right)\end{pmatrix}$$

$$[\phi_s] = \begin{pmatrix}-\phi_1\\ \phi_1\end{pmatrix}, \qquad [\phi_a] = \begin{pmatrix}\phi_1 & 0\\ 0 & \phi_2\end{pmatrix},$$

$$[\phi_f] = \left(1-\frac{i\omega\,\beta_{\text{eff}}}{i\omega+\lambda_{\text{eff}}}\right)\cdot\begin{pmatrix}\phi_1 & \phi_2\\ 0 & 0\end{pmatrix},$$

$$\Sigma_1 = \Sigma_a + \frac{i\omega}{\mathrm{v}_1} + \Sigma_{12} - \nu\Sigma_{f1}\left(1-\frac{i\omega\,\beta_{\text{eff}}}{i\omega+\lambda_{\text{eff}}}\right).$$

The quantities ϕ_1 and ϕ_2 are the steady state fast and thermal neutron fluxes. The effective constants for the delayed neutron precursor, β_{eff} and λ_{eff}, can be calculated as

$$\beta_{\text{eff}} = \sum_{k=1}^{K}\beta_k\,, \qquad \lambda_{\text{eff}} = \frac{\beta_{\text{eff}}}{\sum_{k=1}^{K}\frac{\beta_k}{\lambda_k}}.$$

The neutron noise equation is a partial differential equation with complex coefficients that has to be solved after the static solution is obtained. The related static eigenvalue problem must be solved with the same spatial discretization as the frequency-domain neutron noise equation to get coherent results.

Applying the continuous Galerkin finite element discretization to equation (26) and (27) leads to an algebraic linear system of equation with the following block structure

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \delta\tilde{\phi}_1 \\ \delta\tilde{\phi}_2 \end{pmatrix} = \begin{pmatrix} \delta S_1 \\ \delta S_2 \end{pmatrix}, \tag{28}$$

where $\delta\tilde{\Phi} = \left[\delta\tilde{\phi}_1,\, \delta\tilde{\phi}_2\right]^T$ are the algebraic vectors of weights associated with the fast and thermal neutron noise fluxes. The matrices elements of the different blocks are given by:

$$\begin{aligned}
[A_{11}]_{ij} &= \sum_{e=1}^{N_e} \left(D_1 \int_{\Omega_e} \vec{\nabla} N_i \vec{\nabla} N_j \, dV - D_1 \int_{\Gamma_e} N_i \vec{\nabla} N_j \, d\vec{S} + \left[\Sigma_{\mathrm{dyn}}\right]_{11} \int_{\Omega_e} N_i N_j \, dV \right), \\
[A_{12}]_{ij} &= \sum_{e=1}^{N_e} \left[\Sigma_{\mathrm{dyn}}\right]_{12} \int_{\Omega_e} N_i N_j \, dV, \\
[A_{21}]_{ij} &= \sum_{e=1}^{N_e} \left[\Sigma_{\mathrm{dyn}}\right]_{21} \int_{\Omega_e} N_i N_j \, dV, \\
[A_{22}]_{ij} &= \sum_{e=1}^{N_e} \left(D_2 \int_{\Omega_e} \vec{\nabla} N_i \vec{\nabla} N_j \, dV - D_2 \int_{\Gamma_e} N_i \vec{\nabla} N_j \, d\vec{S} + \left[\Sigma_{\mathrm{dyn}}\right]_{22} \int_{\Omega_e} N_i N_j \, dV \right), \\
[S_1]_{ij} &= \sum_{e=1}^{N_e} \int_{\Omega_e} \delta S_1(\vec{r}, \omega) \; N_i N_j \, dV, \\
[S_2]_{ij} &= \sum_{e=1}^{N_e} \int_{\Omega_e} \delta S_2(\vec{r}, \omega) \; N_i N_j \, \mathrm{d}V.
\end{aligned} \tag{29}$$

It must be noted that the frequency-domain approach must solve a linear system in the complex domain, but this system is only solved once, for the specific frequency of the noise source. On the other hand, the time-domain approach must solve a linear system at each time step, resulting in a large amount of linear systems resolutions for a typical transient. The complex linear system is solved with the help of the *PETSc* library [19] using the GMRES solver [32], to obtain $\delta\tilde{\phi}_1$ and $\delta\tilde{\phi}_2$ which are complex quantities. The results are presented in terms of the fast and thermal neutron noise amplitudes, $|\delta\tilde{\phi}_1|$ and $|\delta\tilde{\phi}_2|$, and the neutron noise phase, $\arg(\delta\tilde{\phi}_1)$ and $\arg(\delta\tilde{\phi}_2)$, corresponding to the argument of the complex number in the $[-180°, 180°]$ interval. Sometimes, it is convenient to represent the relative neutron noise amplitude defined as

$$|\delta\tilde{\phi}_1|_{REL} = \frac{|\delta\tilde{\phi}_1|}{\tilde{\phi}_1}, \qquad |\delta\tilde{\phi}_2|_{REL} = \frac{|\delta\tilde{\phi}_2|}{\tilde{\phi}_2},$$

where $\tilde{\phi}_1$ and $\tilde{\phi}_2$ are respectively the steady-state fast and thermal neutron fluxes previously solved using the same spatial discretization.

2.3.1 Fuel assembly vibration simulation in the frequency-domain

There are different models to describe the vibration of a FA in the frequency-domain. In this work, we used the ϵ/D model, in which the vibration is converted into a spatially localized cross-section perturbation [41].

Assuming a nodal representation of the system (i.e. each fuel assembly is replaced by homogenized regions), one oscillating FA can be modelled as two moving interfaces between homogeneous materials. For the sake of illustration, we consider hereafter two adjacent homogeneous regions and the corresponding possible displacement of the boundary between those regions, as shown in Figure 4. The cross section Σ_α, at the interface $x = b$ between two material regions, can be described as

$$\Sigma_\alpha(x) = \big(1 - \mathcal{H}(x-b)\big)\,\Sigma_\alpha^I + \mathcal{H}(x-b)\,\Sigma_\alpha^{II}, \tag{30}$$

where $\mathcal{H}$ is the unit step function, Σ_α^I and Σ_α^{II} are the cross sections at region I and II, respectively.

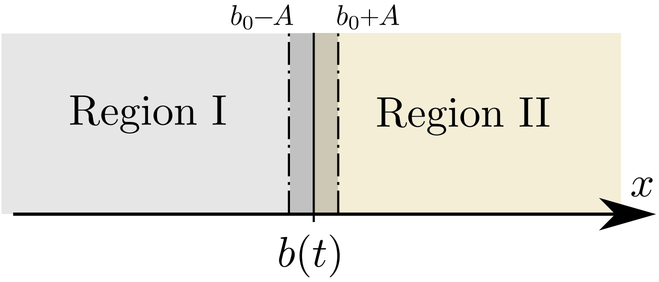


Figure 4: Vibrating interface between two regions.

A moving interface as $b(t) = b_0 + A\sin\big(\omega_p t\big)$, results in:

$$\Sigma_\alpha(x,t) = (1 - \mathcal{H}\big(x - b_0 - A\sin\big(\omega_p\, t\big)\big)\,\Sigma_\alpha^I + \mathcal{H}\big(x - b_0 - A\sin\big(\omega_p\, t\big)\big)\,\Sigma_\alpha^{II}. \tag{31}$$

Using the first order Taylor expansion around $x - b_0$, the cross-section perturbation can be expressed as:

$$\delta\Sigma_\alpha(x,t) = (\Sigma_\alpha^I - \Sigma_\alpha^{II})A\sin\big(\omega_p\, t\big)\,\delta(x - b_0), \tag{32}$$

and, in the frequency-domain, the perturbation can be written as follows,

$$\delta\Sigma_\alpha(x,\omega) = -i\pi A(\Sigma_\alpha^I - \Sigma_\alpha^{II})\delta(x-b_0)\left(\delta\big(\omega-\omega_p\big) - \delta\big(\omega+\omega_p\big)\right). \tag{33}$$

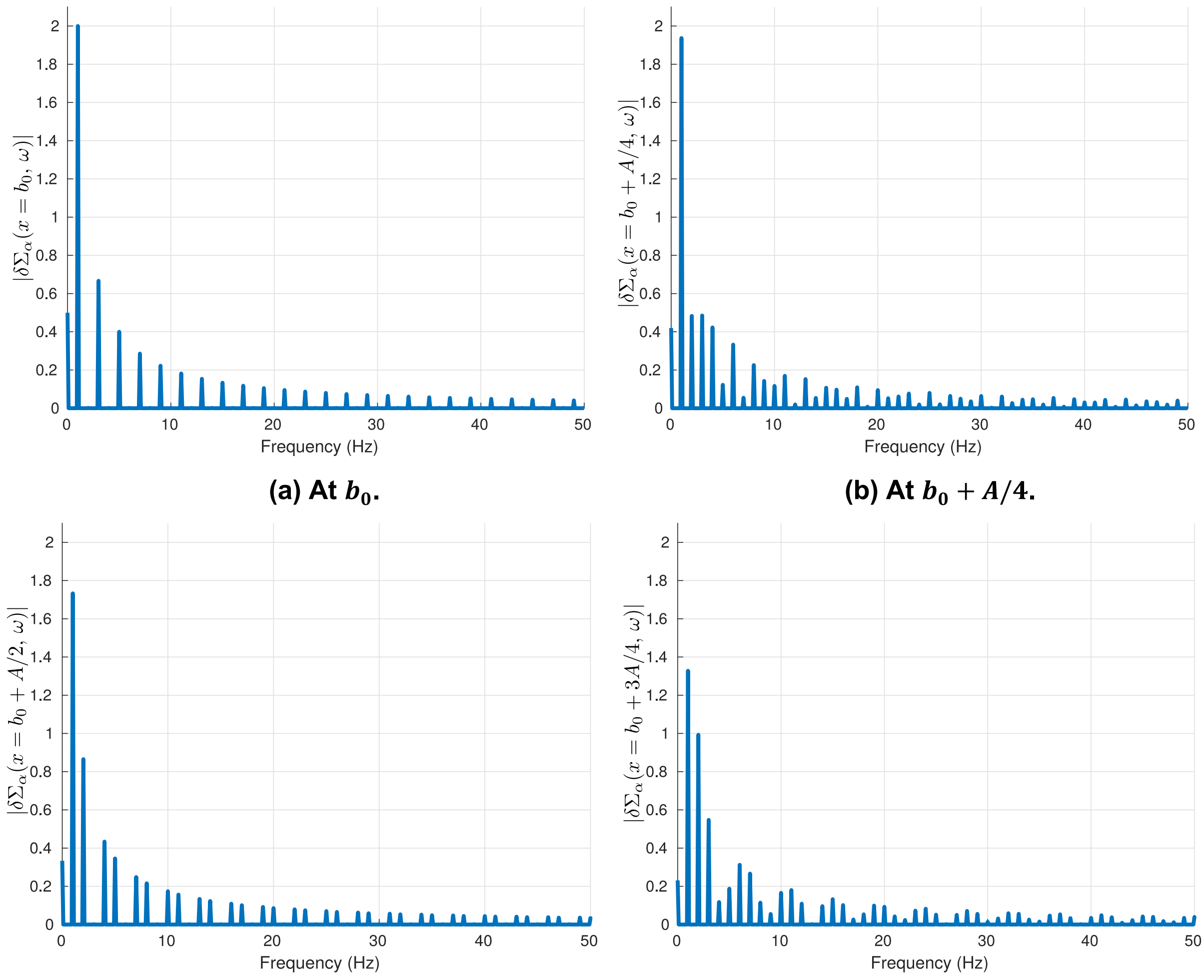
If the perturbation is introduced node-wise, one could assume that the perturbed region is $x \in [b_0 - A, b_0 + A]$ at angular frequency ω_p, with an average value of $\delta\Sigma_\alpha = -1/2i\pi(\Sigma_\alpha^I - \Sigma_\alpha^{II})$ in the perturbed region. If the spatial mesh used does not match the perturbed region, the perturbation must be scaled accordingly. In [42] we can find the full analytical expression for the model of a vibrating FA in the frequency-domain. This model highlights the fact that a monochromatic displacement introduces polychromatic perturbations of the homogenized cross-sections on which the boundary is moving. This is due to the fact that those homogenized regions are perturbed only during parts of the vibration period.

To test the accuracy of the first order approximation, a numerical fast Fourier transform of a time-dependent cross sections $\Sigma_\alpha(x,t)$, is calculated. The numerical FFT was obtained for a one-dimensional moving interface perturbation with $\Sigma_\alpha^I - \Sigma_\alpha^{II} = 1$ and 1 Hz. Figure 5 shows the spectrum of the perturbation at different spatial points using the FFT. The amplitude of the cross-section perturbation is maximum at 1 Hz and the amplitudes at other frequencies are smaller.

Figure 6 shows the absolute value of the amplitude of the perturbation, $|\delta\Sigma_{\alpha}|$, at 1 Hz, 2 Hz, 3 Hz and 4 Hz for the numerical FFT and the first order approximation, i.e., the amplitude of the perturbation at ω_p, $2\omega_p$, $3\omega_p$ and $4\omega_p$. The main frequency of the cross-section perturbation is 1 Hz and corresponds to the frequency of the FA oscillation. In this way, the first order approximation only uses the perturbation at this frequency to model the FA vibration. The first order approximation has the same perturbation integral as the FFT, and it can be introduced using only one node.

The higher harmonics corresponding to frequencies of 2 Hz, 3 Hz and 4 Hz are usually not taken into account in frequency-domain codes. These higher harmonics require additional calculations as the frequency-domain codes solve one frequency each time. Here we study the contributions of those higher harmonics solving the time-dependent problem with *FEMFFUSION*. In this way, the time-domain approach does not decompose the perturbation and does not require any assumption in the perturbation source. In [42], more results corresponding to the higher harmonics of the noise source are given.

Section 2.4.3 studies a generic absorber of variable strength perturbation inside a hexagonal reactor. This noise problem is solved with the time-domain approach and the frequency-domain option of *FEMFFUSION, FEMFFUSION-FD.* In this way, Section 2.4.3 validates the developed frequency-domain noise simulator with FEM that allows to model any type of geometry.



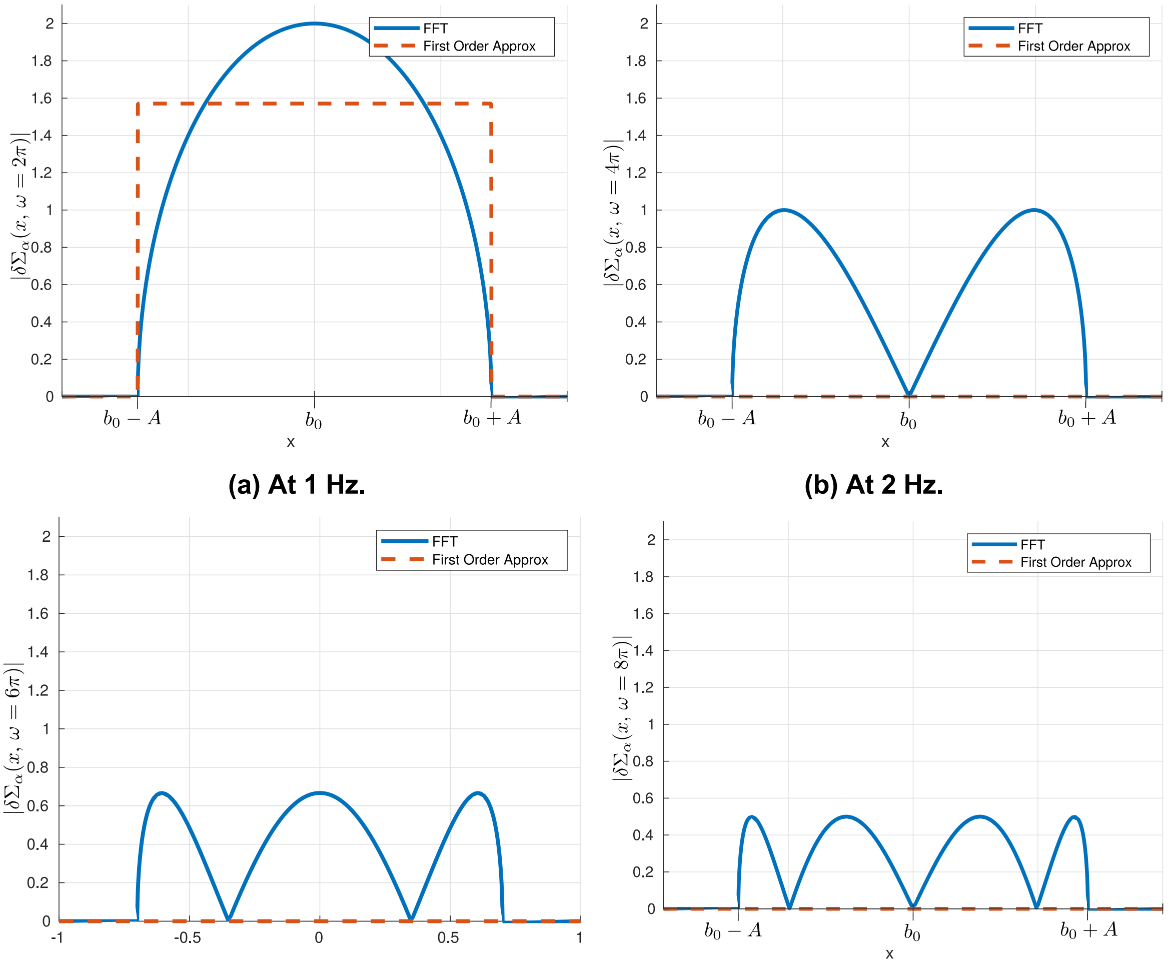
(a) At b_0.

(b) At $b_0 + A/4$.

(c) At $b_0 + A/2$.

(d) At $b_0 + 3A/4$.

Figure 5: Spectrum of the numerical FFT at different spatial points.



(a) At 1 Hz.

(b) At 2 Hz.

(c) At 3 Hz.

(d) At 4 Hz.

Figure 6: Cross-section perturbation amplitudes at different frequencies.

2.4 Comparison between strategies used at UPV

This Section compares the three approaches developed by UPV along with the frequency-domain code *CORE SIM* [35]. In Section 2.4.1, *FEMFFUSION* is compared against *CORE SIM* for a basic one-dimensional example of a vibrating fuel assembly. In Section 2.4.2, the same codes are compared for a more realistic two-dimensional reactor. Section 2.4.3 verifies the time-domain approach of *FEMFFUSION* against the frequency-domain branch of this project, *FEMFFUSION-FD* in a three-dimensional hexagonal reactor to show the possibilities in a non-Cartesian mesh. Section 2.4.4 compares the time-domain solution of *FEMFFUSION* against the time-domain code *PARCS* [36]. All these examples verify the developed strategies.

2.4.1 One-dimensional benchmark

In order to test the numerical tools developed for the FA vibration analysis, a simple one-dimensional benchmark is defined. The benchmark is composed of 11 assemblies 25 cm wide where the vibrating assembly is placed in the middle of the reactor as Figure 7 shows. The cross sections are defined in Table 4 and zero flux boundary conditions are imposed on the left and right frontiers of the reactor. Kinetic data are shown in Table 5 where one group of delayed neutron precursors is used. The problem is made critical before starting the time dependent calculation by dividing $\nu\Sigma_{fg}$ by the previously calculated multiplicative factor of the problem, i.e. $k_{\text{eff}} = 0.9792500$.

The movement of the central assembly is defined as:

$$x_i(t) = x_{i0} + A\ \sin\big(\omega_p t\big), \tag{34}$$

where $x_i(t)$ represents the position of the vibrating assembly along time, originally placed in x_{i0}.

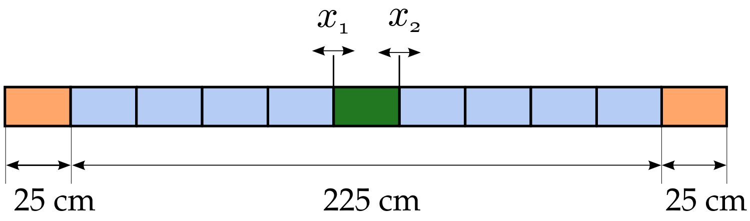


Figure 7: Geometry of the one-dimensional benchmark.

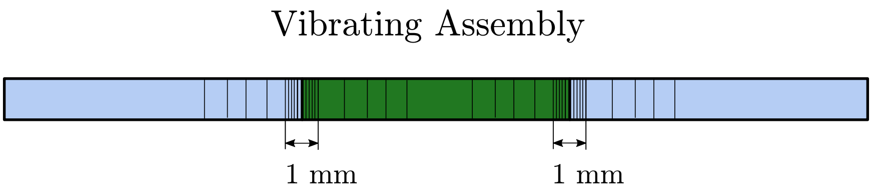


Figure 8: Refined mesh near the vibrating fuel assembly.

Table 4: Cross sections of the materials of the one-dimensional benchmark.

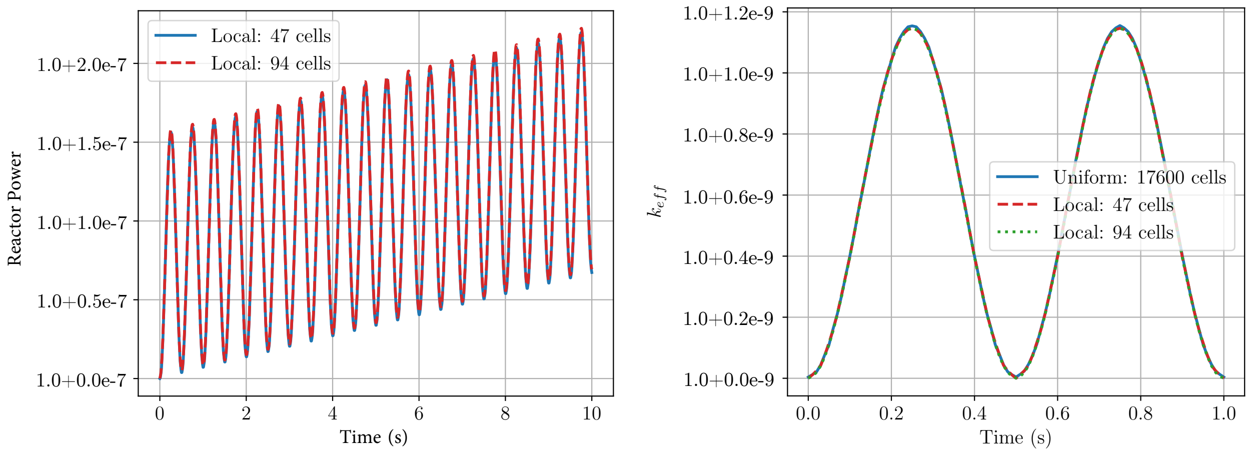
Material	g	D_g (cm)	Σ_{ag} (1/cm)	$\nu\Sigma_{fg}$ (1/cm)	Σ_{fg} (1/cm)	Σ_{12} (1/cm)
Fuel	1	1.40343	1.17659e-2	5.62285e-3	2.20503e-3	1.60795e-2
	2	0.32886	1.07186e-1	1.45865e-1	5.90546e-2	
Vibrating Assembly	1	1.40343	1.17659e-2	5.60285e-3	2.19720e-3	1.60795e-2
	2	0.32886	1.07186e-1	1.45403e-1	5.88676e-2	
Reflector	1	0.93344	2.81676e-3	0.00000e+0	0.00000e+0	1.08805e-2
	2	0.25793	8.87200e-2	0.00000e+0	0.00000e+0	

Table 5: Kinetic neutron data for 1D benchmark.

β_{eff}	λ_{eff} (1/s)	v_1 (cm/s)	v_2 (cm/s)
0.0065	0.0784130	1.25e7	2.5e5

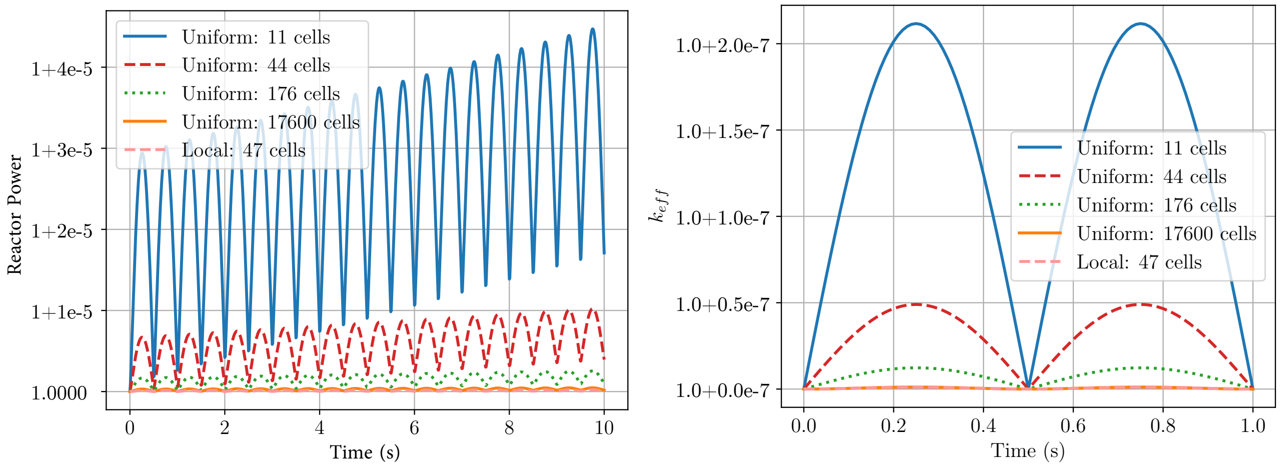
Figure 9a shows the relative power evolution for an oscillation of $A = 1$ mm of amplitude and a frequency of $\omega_p/2\pi = 1$ Hz along 10 periods calculated with the time-domain code FEMFFUSION. A sinusoidal evolution in the relative power can be seen with a small amplitude, around 7.87e-8 and with a constant increment along time. This increment is caused because the reactor becomes slightly supercritical when the central assembly moves from its starting position. This is explained by the fact that this assembly is in the middle of the reactor and this assembly is less reactive than the surrounding assemblies. Figure 9b displays the static multiplication factor through the positions travelled during one period. The change in the k_{eff} is less that 1.2e-9. Figures 10a and 10b present the evolution of the power and of the multiplication factor, respectively, when changing the spatial mesh. Since those quantities are integral quantities and since the moving assembly is located in the centre of a symmetric core, these figures also show that the sinusoidal global results present twice the frequency of the mechanical FA oscillation. In other words, the global reactor results of the assembly moving to the right are equal to the results when the assembly is moving to the left provoking a 2 Hz oscillation in the power and the multiplicative factor. However, the space dependent neutron noise has the same frequency as the mechanical FA vibration. The behaviour of the total power is analogous with the one studied analytically in a point kinetic reactor model in [43] and [44] from a sinusoidal change in reactivity where the linear increase of the power is caused by the delayed neutrons. However, thermal-hydraulic feedback will eliminate the slow increment in the total power because, in an operating nuclear reactor, the temperature coefficients of reactivity are negative [15] and the increment in the power is very small.

In Figure 9a and Figure 9b, two non-equidistant meshes are compared to each other. In both cases, a locally refined mesh around the moving FA is used. The coarsest mesh uses 47 cells with the following sizes $[4*25.00, 24.5, 4*0.1, 10*0.02, 4*0.1, 24.0, 4*0.1, 10*0.02, 4*0.1, 24.5, 4*25.00]$ cm. A detail of the refined mesh near the vibrating assembly is shown in Figure 8. This model of refinement is based on the initial assembly configuration adding 20 cells where the movement of the assembly is located and 4 transition cells at each side of this area to make a smooth change of the refinement level. The other locally refined mesh uses 94 cells where each of the previous cells is split into two cells. Also, a uniform mesh with 17600 cells is utilized for the sake of comparisons. All computations are calculated with polynomials of degree 5 in the finite element method. The results obtained for all the three meshes are essentially identical. The results with 47 cells mesh can thus be considered as spatially converged. The refinement model will still be valid when the oscillation amplitude is increased, and the size of the cells is changed accordingly. Figure 10 shows the evolution of the reactor power and k_{eff} using a smaller number of uniform spatial mesh cells. When using too coarse meshes, the computed solution deviates significantly from the converged solution. These figures indicate the necessity of using local refinements around the oscillating assembly and accurate numerical solvers to correctly integrate the time dependent neutron diffusion equation.



(a) Reactor power along 10 periods. **(b) Multiplication factor along one period.**

Figure 9: Global results for the 1D benchmark.



(a) Reactor power along 10 periods. **(b) Multiplication factor along one period.**

Figure 10: Comparison of results with different refinement levels. Only the local discretization with 47 cells and the uniform discretization with 17600 cells give spatially converged results.

Figure 11 shows the power evolution for different oscillation amplitudes from 0.3 mm to 3 mm while the frequency is fixed to 1 Hz for a mesh with 47 cells. The size of the mesh cells has been adapted to the oscillation amplitude ensuring that the obtained results are spatially converged. Obviously, as the oscillation amplitude increases, its effect in the power increases. Figure 12 represents the spatial dependence of the neutron flux disturbance around the steady-state value at four different simulation times during the first period of oscillation. It can also be verified that the neutron noise response to the vibration disturbance is basically a sinusoidal function at the FA oscillation frequency.

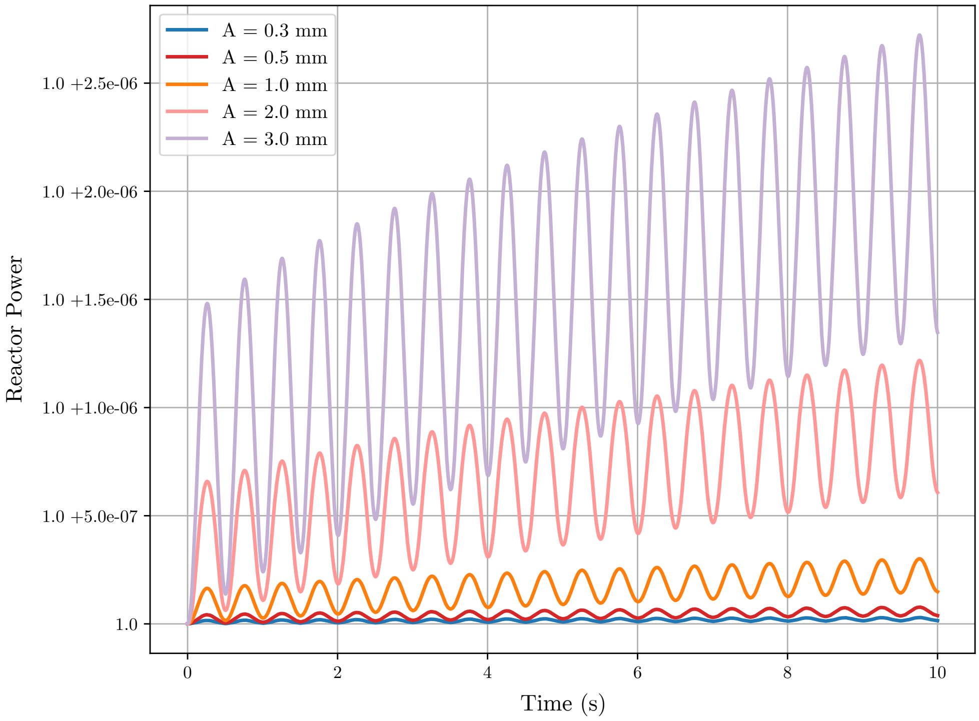
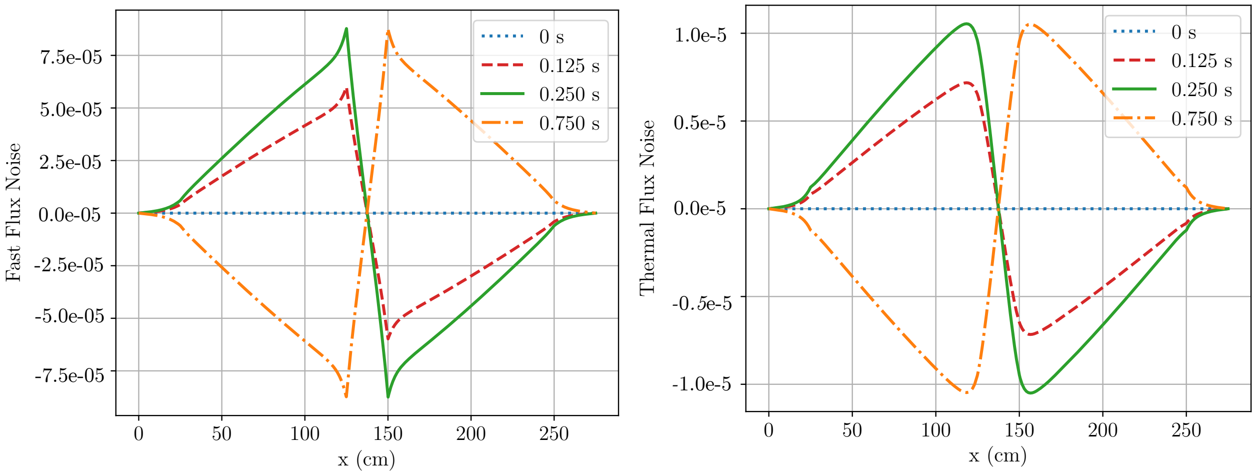


Figure 11: Total power evolution for different oscillation amplitudes.

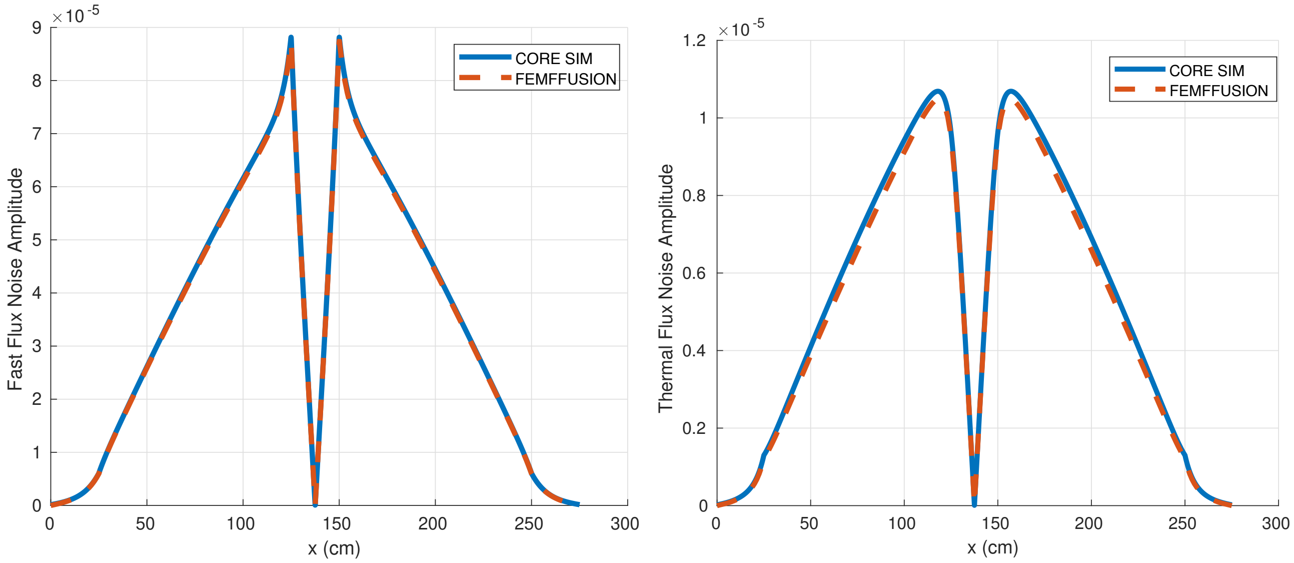


(a) Fast flux noise.

(b) Thermal flux noise.

Figure 12: Spatial evolution of the neutron noise at different times.

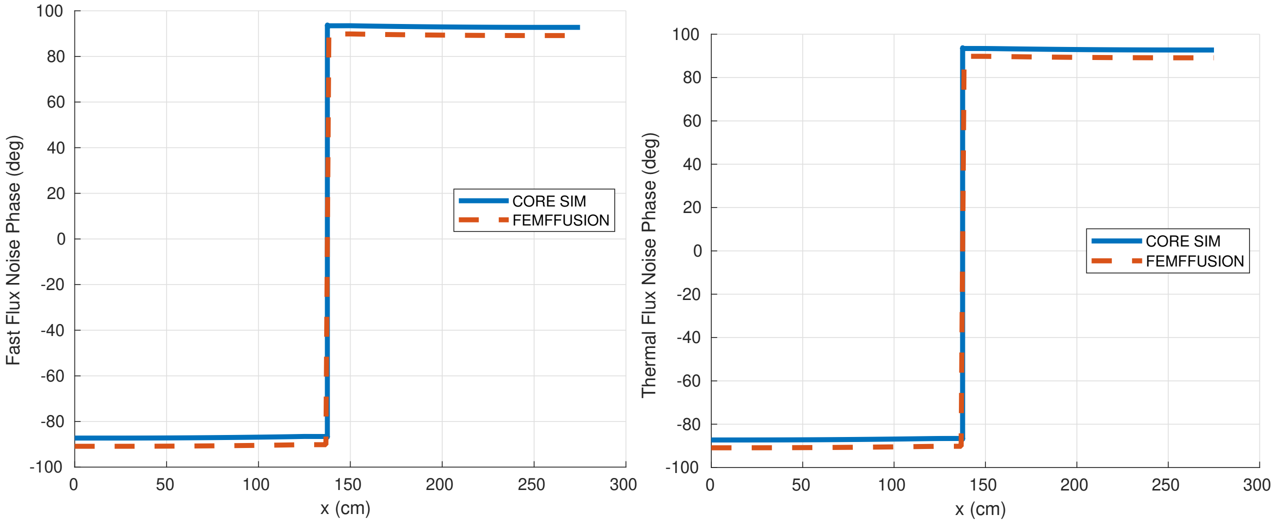
Figure 13 shows a comparison of the amplitude of neutron flux noise for the fast and thermal groups between the time-domain code *FEMFFUSION* and the frequency-domain code *CORE SIM*. These figures look similar to the maximum amplitude over all simulation times because the principal response of the neutron noise is a sinusoidal function at the FA vibration frequency. Figure 14 displays the results for the phase. A close agreement is observed between the time-domain and the frequency-domain methodologies.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 13: Neutron noise amplitude comparison in the 1D benchmark.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 14: Neutron noise phase comparison in the 1D benchmark.

2.4.2 Two-dimensional BIBLIS benchmark

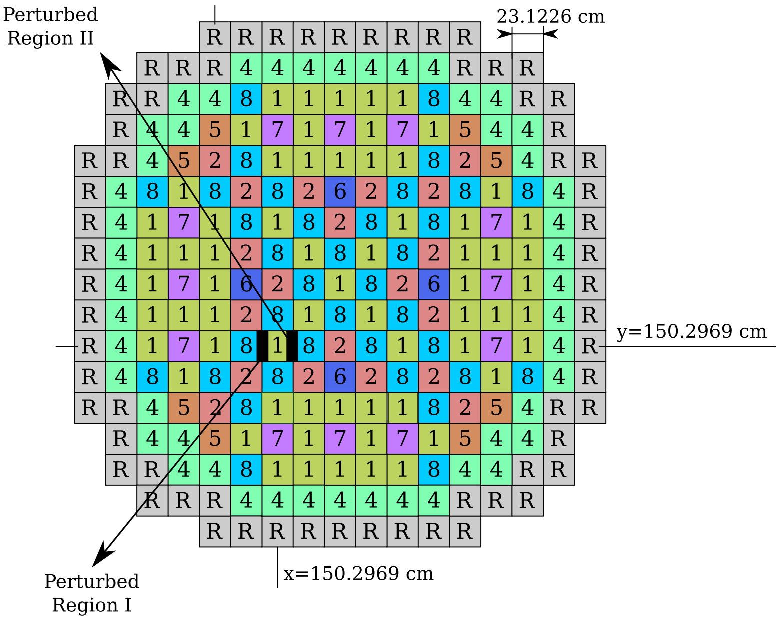
The BIBLIS 2D benchmark is selected to compare the frequency and time-domain analysis for a vibrating FA. This is a classical two-group neutron diffusion problem taken as a benchmark for different numerical codes. The material and geometry definition of the benchmark can be found in [45]. The problem is made critical before starting the time-dependent calculation by dividing $\nu\Sigma_{fg}$ by k_{eff}.

The assembly in the position $(6,6)$ is selected to be oscillating along the x direction as

$$x_i(t) = x_{i0} + A\ \sin(\omega_p t), \tag{35}$$

where $x_i(t)$ represents each position of the vibrating assembly along time, originally placed at x_{i0}. The selected amplitude of the movement is 1 mm and the frequency of the vibration is 1 Hz, i.e. $\omega_p = 2\pi$. The perturbation for the time-domain calculation is inserted by changing the cross sections at each time step and modelling the movement of the FA by volume-averaged cross sections. For the frequency-domain calculation, the perturbation is inserted according to the first order approximation as explained in Section 2.2.1.3. In this way, the perturbation is inserted cell wise in the adjacent cells next to the moving FA as shown in Figure 15. The value of the cross sections is given in Table 6 and the value of the perturbed cross sections in Table 7 following the formula, $\delta\Sigma_\alpha = -i\frac{\pi}{2}(\Sigma_\alpha^I - \Sigma_\alpha^{II})$. The dynamic data are shown in Table 8. Two perturbed regions are then defined: one on the left and one on the right, respectively, of the moving boundary of the vibrating FA. These regions have the same cross-section perturbation amplitude but a phase lag of $180°$.

Due to the different scales of the problem, a fine mesh needs to be used to accurately solve the system. In the time-domain analysis, a refined spatial mesh in the surroundings of the moving FA with 869 cells and cubic polynomials in the FEM are used. In the frequency-domain analysis, a uniform mesh of 4624 cells is employed. If these fine meshes are not used, the effect of the FA vibration could be overestimated [12].



				R	R	R	R	R	R	R	R	R				
		R	R	R	4	4	4	4	4	4	4	R	R	R		
	R	R	4	4	8	1	1	1	1	1	8	4	4	R	R	
	R	4	4	5	1	7	1	7	1	7	1	5	4	4	R	
R	R	4	5	2	8	1	1	1	1	1	8	2	5	4	R	R
R	4	8	1	8	2	8	2	6	2	8	2	8	1	8	4	R
R	4	1	7	1	8	1	8	2	8	1	8	1	7	1	4	R
R	4	1	1	1	2	8	1	8	1	8	2	1	1	1	4	R
R	4	1	7	1	6	2	8	1	8	2	6	1	7	1	4	R
R	4	1	1	1	2	8	1	8	1	8	2	1	1	1	4	R
R	4	1	7	1	8	1	8	2	8	1	8	1	7	1	4	R
R	4	8	1	8	2	8	2	6	2	8	2	8	1	8	4	R
R	R	4	5	2	8	1	1	1	1	1	8	2	5	4	R	R
	R	4	4	5	1	7	1	7	1	7	1	5	4	4	R	
	R	R	4	4	8	1	1	1	1	1	8	4	4	R	R	
		R	R	R	4	4	4	4	4	4	4	R	R	R		
				R	R	R	R	R	R	R	R	R				

Figure 15: Materials and perturbed regions of the 2D BIBLIS reactor.

Table 6: Static cross sections of the BIBLIS 2D reactor.

Material	g	D_g	Σ_{ag}	$\nu\Sigma_{fg}$	Σ_{12}
		(cm)	(1/cm)	(1/cm)	(1/cm)
1	1	1.4360	0.0095042	0.0058708	0.017754
	2	0.3635	0.0750580	0.0960670	
2	1	1.4366	0.0096785	0.0061908	0.017621
	2	0.3636	0.0784360	0.1035800	
3	1	1.3200	0.0026562	0.0000000	0.023106
	2	0.2772	0.0715960	0.0000000	
4	1	1.4389	0.0103630	0.0074527	0.017101
	2	0.3638	0.0914080	0.1323600	
5	1	1.4381	0.0100030	0.0061908	0.017290
	2	0.3665	0.0848280	0.1035800	
6	1	1.4385	0.0101320	0.0064285	0.017192
	2	0.3665	0.0873140	0.1091100	
7	1	1.4389	0.0101650	0.0061908	0.017125
	2	0.3679	0.0880240	0.1035800	
8	1	1.4393	0.0102940	0.0064285	0.017027
	2	0.3680	0.0905100	0.1091100	

Table 7: Perturbation cross sections of the BIBLIS 2D reactor.

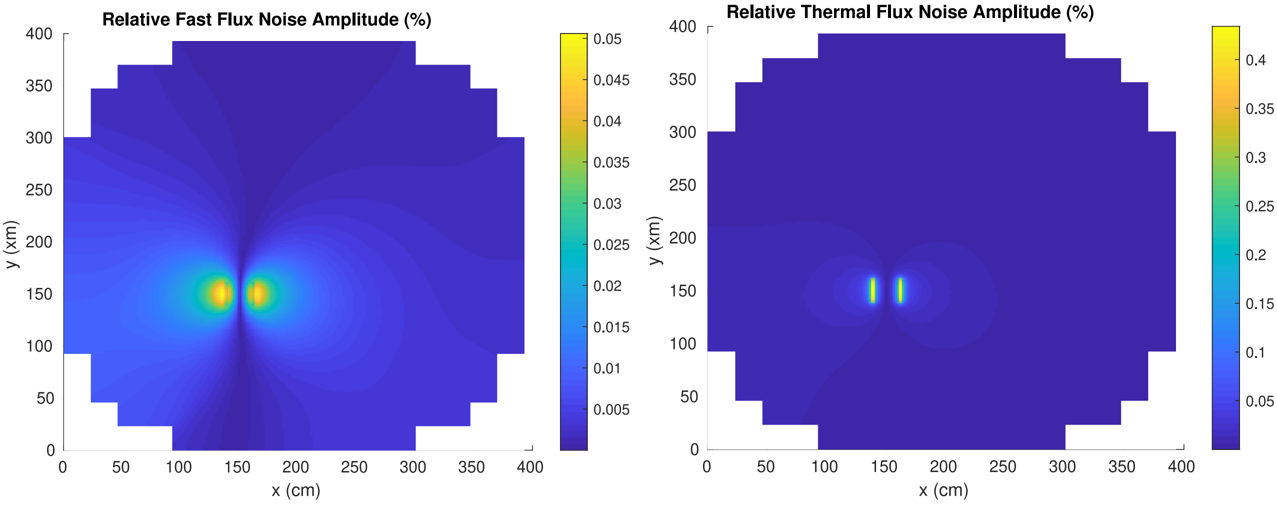
Material	g	δD_g	$\delta\Sigma_{ag}$	$\nu\delta\Sigma_{fg}$	$\delta\Sigma_{12}$
		(cm)	(1/cm)	(1/cm)	(1/cm)
Perturbed Region I	1	0.0000000	-0.0009668i	-0.0003642i	+0.0009330i
	2	0.0000000	-0.0190000i	-0.0085000i	
Perturbed Region II	1	0.0000000	+0.0009668i	+0.0003642i	-0.0009330i
	2	0.0000000	+0.0190000i	+0.0085000i	
Non-perturbed Regions	1	0.0000000	+0.0000000	+0.0000000	+0.0000000
	2	0.0000000	+0.0000000	+0.0000000	

Table 8: Kinetic neutron data for the 2D BIBLIS benchmark.

λ_1 (1/s)	λ_2 (1/s)	λ_3 (1/s)	λ_4 (1/s)	λ_5 (1/s)	λ_6 (1/s)	λ_{eff} (1/s)	v_1 (cm/s)
0.0127	0.0317	0.115	0.311	1.40	3.87	0.0784130	1.25×10^7
β_1	β_2	β_3	β_4	β_5	β_6	β_{eff}	v_2 (cm/s)
0.000247	0.0013845	0.001222	0.0026455	0.000832	0.000169	0.0065	2.5×10^5

Figure 16 shows the spatial distribution of the amplitude of the relative neutron noise in the frequency-domain calculated with the time-dependent code *FEMFFUSION* at 1 Hz. In this figure, the relative neutron noise is represented in percentage of the steady-state neutron flux. It can be seen that the induced fast neutron noise has an influence on larger scales through the reactor compared to the thermal noise, due to the larger mean free path of fast neutrons. On the other hand, the thermal noise is localized in the surroundings of the oscillating FA. Two clear peaks can be observed in these Figures at the position of the moving interfaces where the cross sections change along the FA movement.

When a perturbation in a critical system without thermal-hydraulic feedback is introduced (even a sinusoidal perturbation), the system becomes unstable, since k_{eff} deviates from unity. Nevertheless, this change in criticality will be attenuated by the thermal-hydraulic feedback that has a stabilizing effect. As Figure 17 shows that maximum static reactivity is only $\rho = (k_{\text{eff}} - 1)/k_{\text{eff}} \simeq 1.5 \times 10^{-8}$. Nevertheless, it is possible to introduce some kind of average kinetic eigenvalue to mitigate this effect [46]. This correction was not implemented in this work because of the smallness of the perturbation in reactivity. If the simulation were longer or the perturbation greater, this kind of treatment would be necessary.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 16: Noise amplitudes for the 2D BIBLIS reactor at 1 Hz.

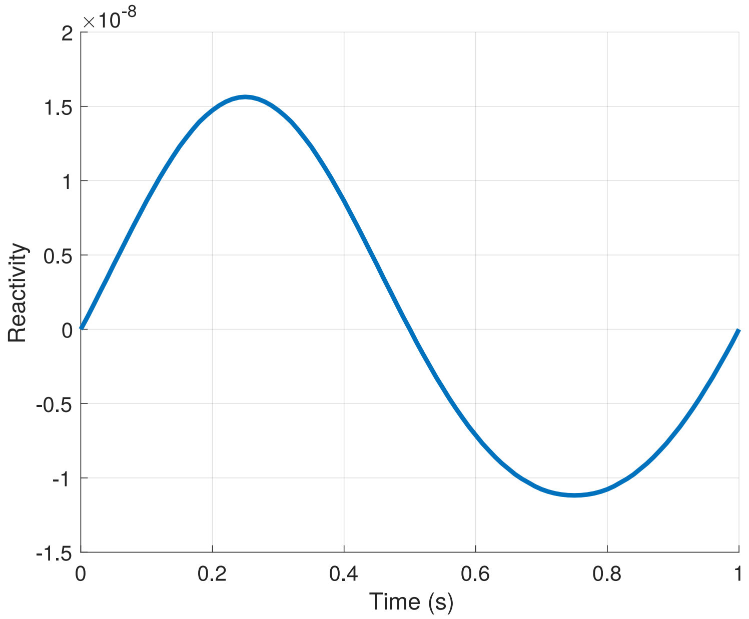
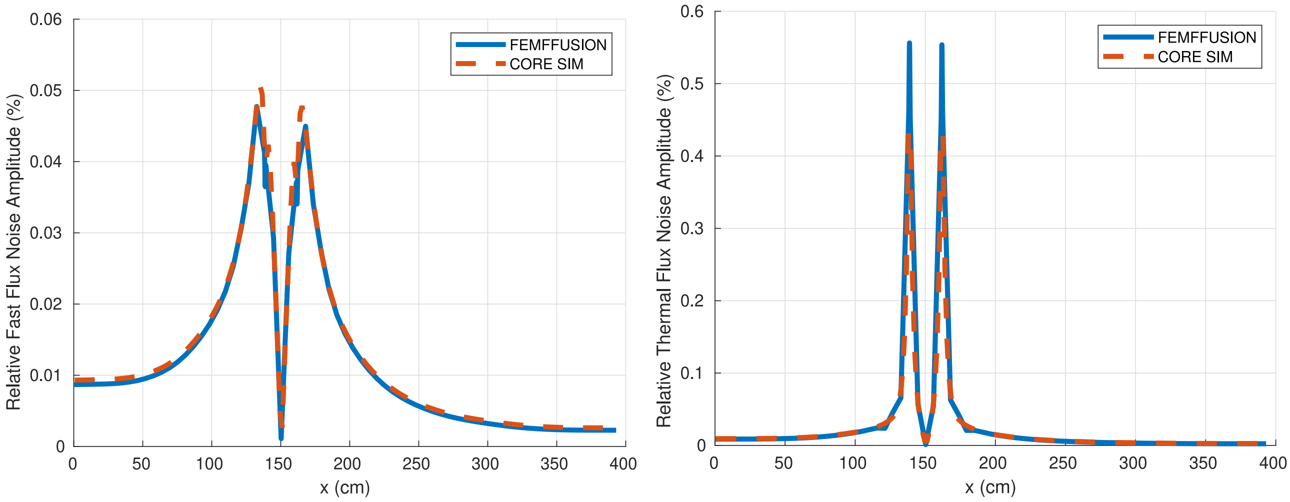


Figure 17: Variation of the reactivity for the 2D BIBLIS reactor.

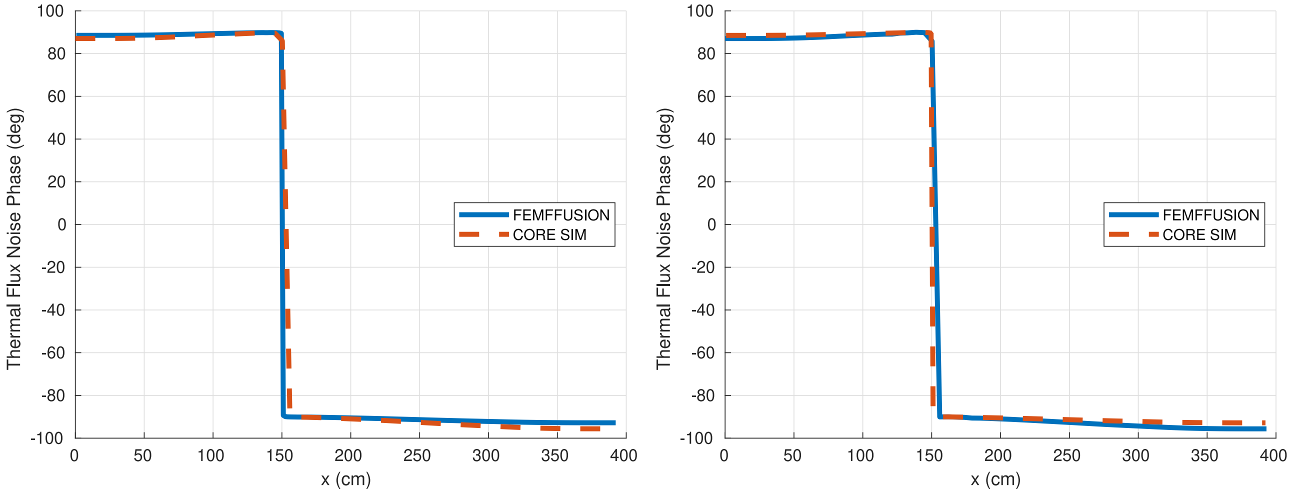
Figure 18 shows a comparison of the relative amplitude of neutron flux noise for the fast and thermal groups along $y = 150.2969$ cm, obtained with the *FEMFFUSION* code and the *CORE SIM* code, in other words, between the time-domain and the frequency-domain approaches. Figure 19 gives a comparison of the phase of the neutron noise obtained with the two codes. A close agreement is observed for both the amplitude and the phase of the neutron noise for the vibration frequency at 1 Hz, especially far from the FA vibration. We can observe that the vibrating FA produces a phase change from $90°$ to $-90°$ in the induced noise.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 18: Relative noise amplitude comparison for 2D BIBLIS reactor in *y* = 150.2969 cm at 1 Hz.



(a) Fast flux noise. **(b) Thermal flux noise.**

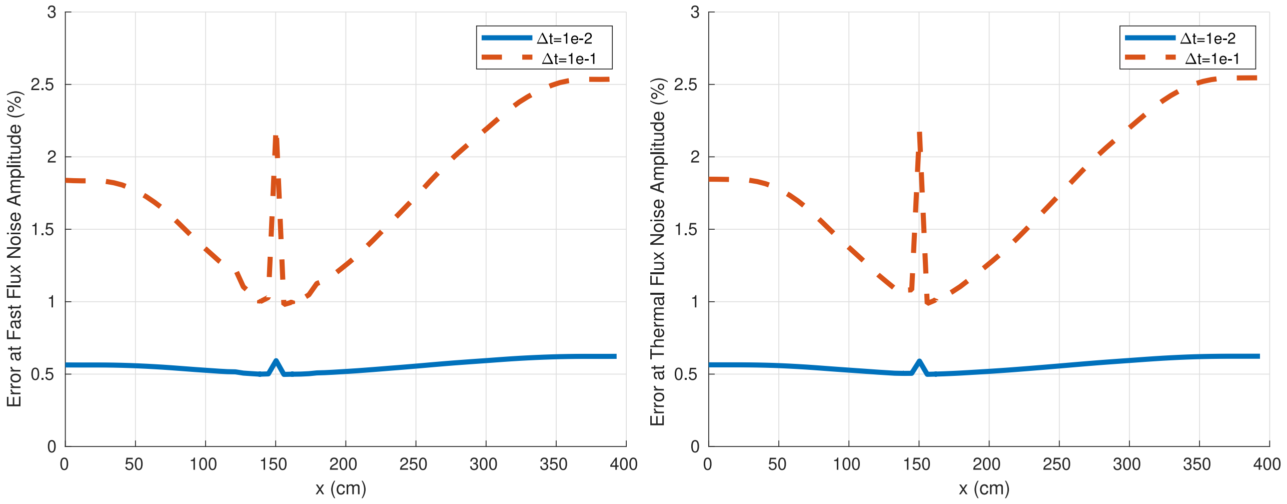
Figure 19: Phase comparison for 2D BIBLIS reactor in *y* = 150.2969 cm at 1 Hz.

Figure 20 shows the relative error in the noise amplitude along the line $y = 150.2969$ cm when different time steps are used. This figure demonstrates that a time step Δt of 10^{-2} s is small enough to accurately reproduce the noise amplitude results.

The relative difference between the frequency-domain and the time-domain approaches is quantified in Figure 21. The relative differences are small in regions far away from the perturbation, although the relative differences at the perturbation location are comparably bigger. This difference is mainly caused by the spatial discretization error. The FEM approach is known to better capture steep gradients, as is the case close to the noise source.

Figure 22 displays the amplitude for the second harmonic of the neutron noise, at 2 Hz, calculated with the time-dependent code. It can be observed that this noise is sharper than the noise associated with the fundamental frequency and its amplitude much smaller than the 1 Hz component. In this figure, it can be seen that there are two drops of noise amplitude in the middle of the moving interfaces as the ones shown in the frequency analysis of the cross-section perturbation in Figure 6. Figure 23 shows the amplitude of the neutron noise for the third harmonic at 3 Hz. These higher harmonics have less influence on the total neutron noise because their amplitudes are much smaller than the fundamental harmonic of the noise at the FA vibration frequency. The contributions of the harmonic frequencies (which are multiples of the fundamental frequency) come from the fact that the perturbation source has different frequencies. In the time-domain simulations moreover, the harmonic frequencies are also due to the explicit modelling of the non-linearities. However, the harmonics following the first one are much smaller and cancel out in adjacent regions. This causes that these harmonics have a very small noise amplitude. Thus, the neutron noise at the FA frequency is the main component of the neutron noise and must be the one used to detect anomalies in the operation of a nuclear reactor.

It should also be pointed out that the analysis of plant data is typically performed using frequency spectra of neutron detectors, i.e. the effect of vibrations is thus resolved at different frequencies. If only the fundamental frequency is considered, the present work demonstrates that estimating the effect of a FA vibration can be equally well carried out using modelling in the time-domain or in the frequency-domain (using a first order approximation).



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 20: Relative error in noise amplitude at different time steps for 2D BIBLIS reactor in *y* = 150.2969 at 1 Hz.

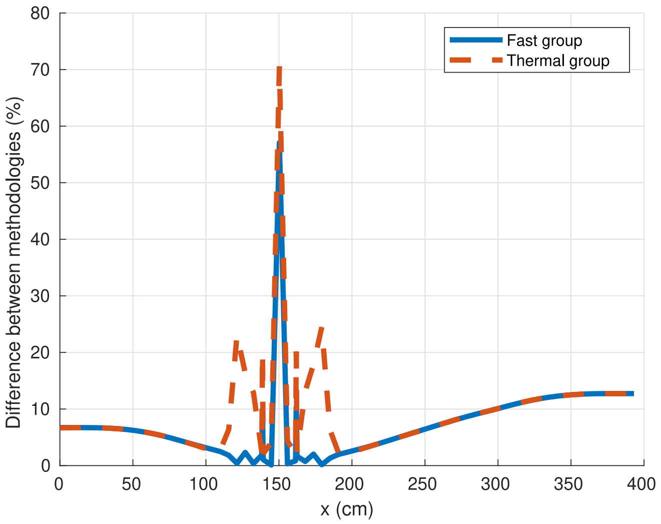
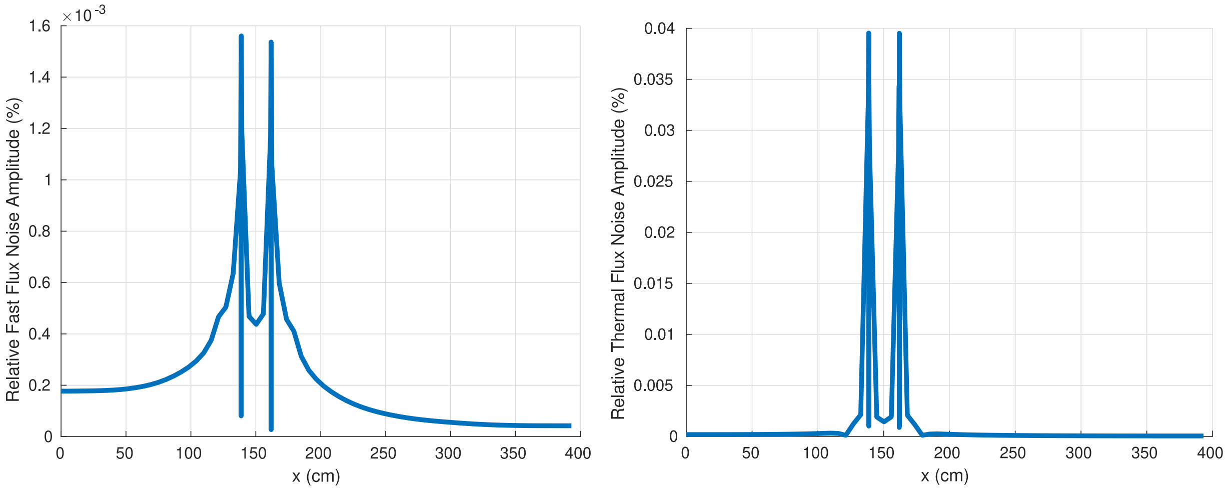


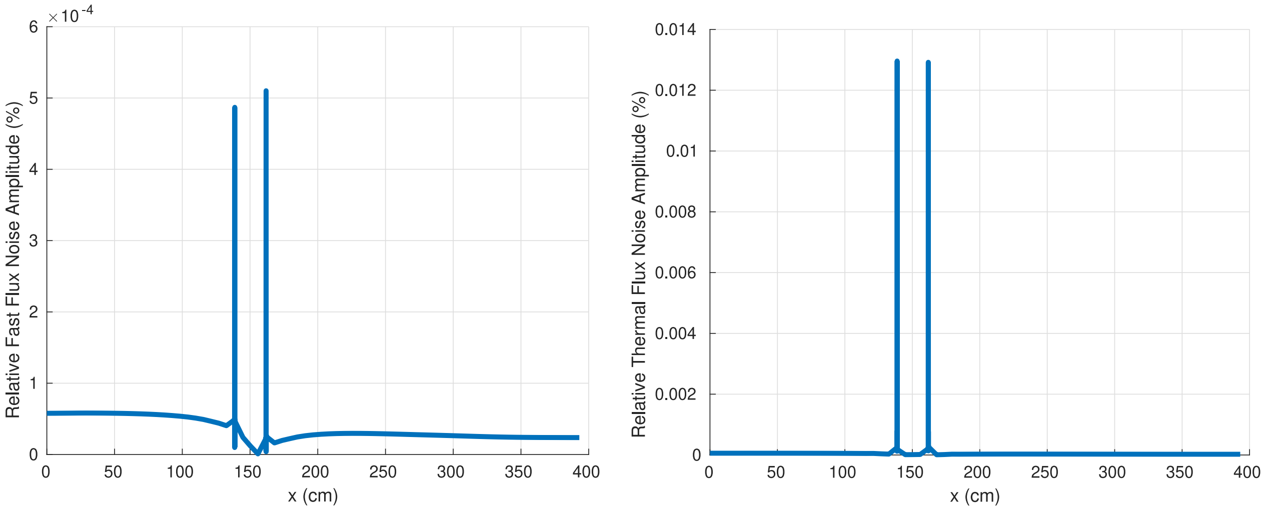
Figure 21: Relative difference in noise amplitude between frequency-domain and time-domain methodologies for 2D BIBLIS reactor in *y* = 150.2969 cm at 1 Hz.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 22: Relative noise amplitudes in *y* = 150.2969 cm at 2 Hz.

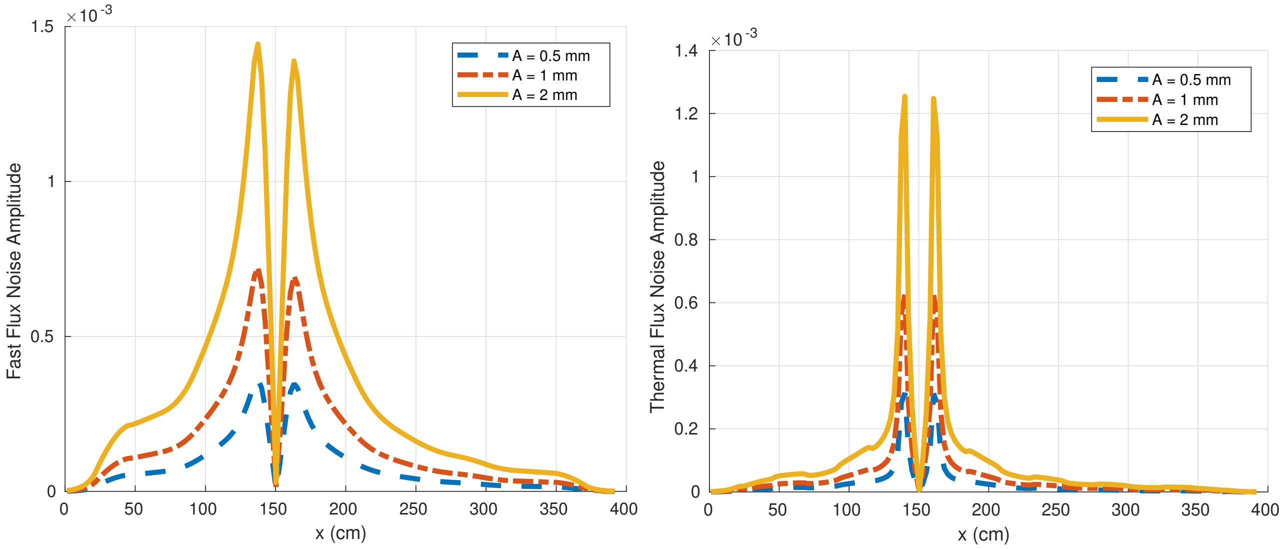


(a) Fast flux noise. **(b) Thermal flux noise.**

Figure 23: Relative noise amplitudes in *y* = 150.2969 cm at 3 Hz.

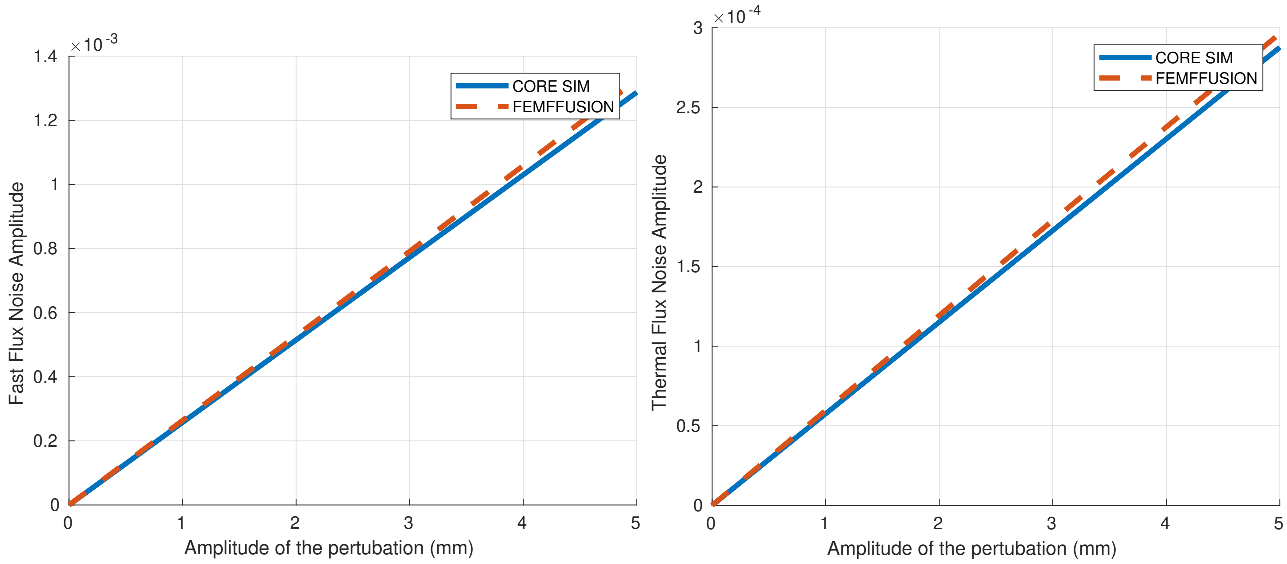
Figure 24 shows the fast and the thermal neutron noise magnitudes for three different oscillation amplitudes calculated with *CORE SIM*. Figure 25 displays the neutron noise amplitude associated with the fast and thermal fluxes for different vibrations amplitudes at the neutron detector, situated at $(x = 104.0517,\ y = 150.2969)$ cm. These results are calculated with *FEMFFUSION* and *CORE SIM* codes. In these figures, a proportional dependency of the neutron noise magnitude with the amplitude of the FA vibration amplitude can be observed.

Figure 26 shows the fast and the thermal neutron noise magnitudes for three different oscillation frequencies computed with *CORE SIM*. Even though the selected vibration frequencies are not realistic values for mechanical vibrations of FA, the neutron noise results are similar. Figure 27 displays the neutron noise magnitude against the vibration frequency at the position of the neutron detector calculated with *CORE SIM* with 1 group of precursors of delayed neutrons, and *FEMFFUSION* with 1 group of delayed neutron precursors (1 gdnp) and 6 groups of delayed neutron precursors (6 gdnp), respectively. This figure is similar to the reactor transfer function shown in [47] where the usual mechanical FA vibration frequencies fall in the so-called *plateau region*, defined as the frequency range $[\lambda_{\text{eff}},\ \beta_{\text{eff}}/\Lambda_0]$. This indicates that with respect to the amplitude of the neutron noise, the frequency-dependence of the reactor response tends to follow the one of point-kinetics. As it can be seen for usual FA vibration frequencies, the neutron noise is basically independent of the vibration frequency. Since the effect of non-linearities is negligible, no resonance effect is predicted. The inclusion of thermal-hydraulic feedback might nevertheless modify the frequency response of the system. Also, a close match between the time-domain and the frequency-domain methodologies is obtained. Only at low frequencies, i.e. at frequencies smaller than 0.5 Hz, a discrepancy between calculations with different number of delayed neutron precursors groups can be observed.



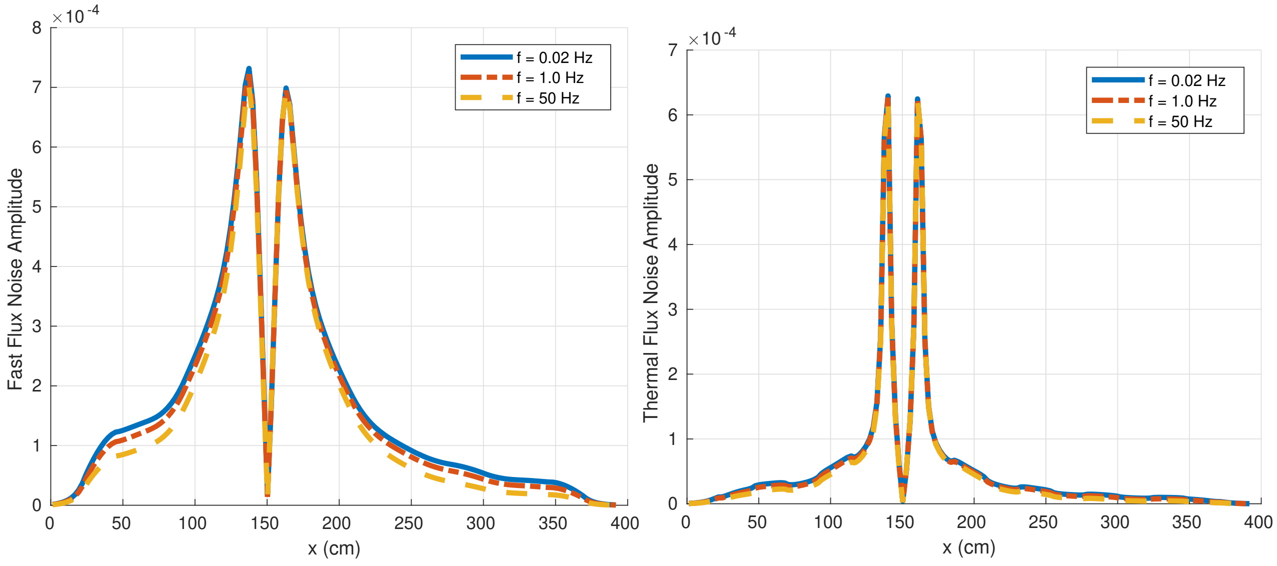
(a) Fast flux noise. **(b) Thermal flux noise.**

Figure 24: Noise magnitude comparison for different amplitudes of vibrations in the 2D BIBLIS reactor at y=150.2969 cm computed with CORE SIM.



(a) Fast flux noise. **(b) Thermal flux noise.**

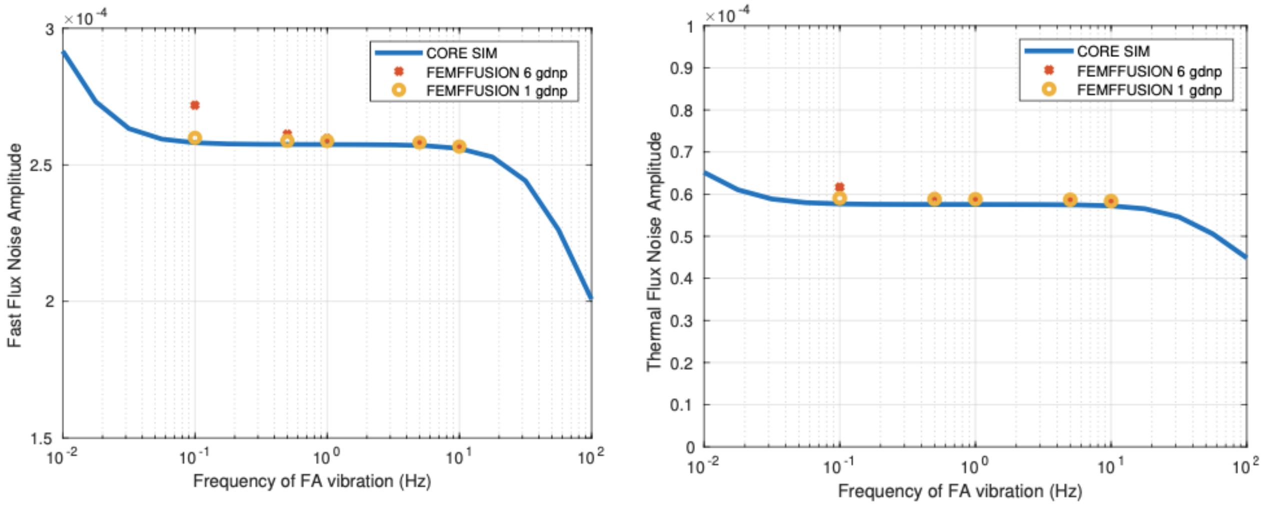
Figure 25: Noise magnitude comparison for different amplitudes of vibrations in the 2D BIBLIS reactor at the detector position.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 26: Noise magnitude comparison for different frequencies of vibrations in the 2D BIBLIS reactor at y=150.2969 cm computed with CORE SIM.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 27: Noise amplitude comparison between CORESIM and FEMFFUSION at different frequencies.

These benchmarks presented the modelling of the effect of the mechanical vibration of fuel assemblies onto neutron noise. The problem combined different spatial scales, e.g. a 1 mm oscillation of a fuel assembly that measures about 20 cm in radial size in a reactor of some metres long. This implies that we need to work with a very high precision in the spatial discretization and in the tolerances given to the solvers, especially in the time-domain.

Numerical results show that the main oscillation in the neutron flux is obtained at the same frequency as the one of the assembly vibrations. Neutron noise at multiples of the mechanical vibration are also seen in time-dependent calculations but with much less amplitude. Numerical results show a close match between these two approaches at the FA vibration frequency.

Higher harmonics are less important to the total neutron noise because their amplitude is much smaller than the fundamental harmonic. Furthermore, the neutron noise corresponding to those higher harmonics has a much larger spatial decay away from the perturbation. It thus means that unfolding the location of the vibrating FA from distant neutron detectors is far more challenging than using the neutron noise at the fundamental frequency.

2.4.3 Three-dimensional hexagonal VVER-1000 benchmark

As a case of study for a hexagonal reactor, a typical hexagonal VVER-1000 reactor core is considered [48]. This benchmark has a 1/12 reflective symmetry but, as the inserted perturbation is not symmetrical, the whole reactor must be solved. The core is composed of 163 fuel assemblies surrounded by 54 reflector cells. Figure 28 shows the materials layout of the core. The fuel assembly pitch is 23.6 cm, and the active height is 355 cm. Therefore, the total height is 426 cm including 35.5 cm thick reflectors in the upper and the lower part of the core. The reactor is discretized into 24 planes, each one of 17.75 cm thick. Vacuum boundary conditions are assumed for this problem.

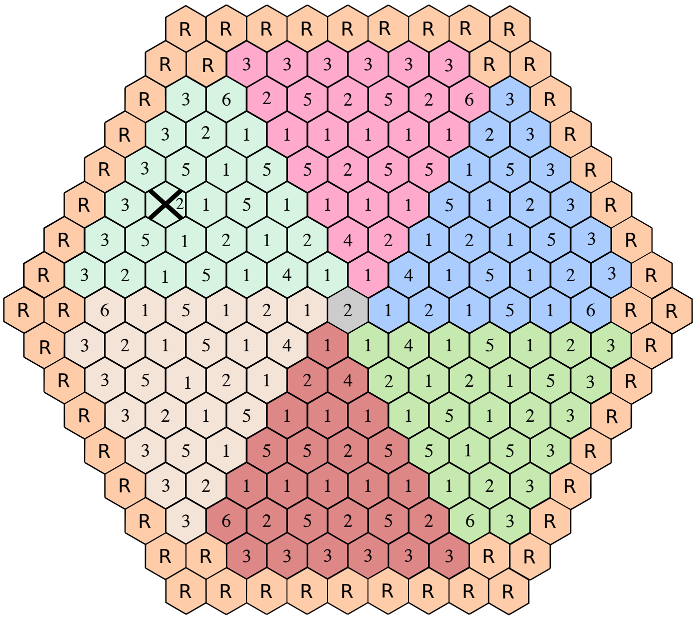

R R R R R R R R R
R R 3 3 3 3 3 3 R R
R 3 6 2 5 2 5 2 6 3 R
R 3 2 1 1 1 1 1 1 2 3 R
R 3 5 1 5 5 2 5 5 1 5 3 R
R 3 2 1 5 1 1 1 1 5 1 2 3 R
R 3 5 1 2 1 2 4 2 1 2 1 5 3 R
R 3 2 1 5 1 4 1 1 4 1 5 1 2 3 R
R R 6 1 5 1 2 1 2 1 2 1 5 1 6 R R
R 3 2 1 5 1 4 1 1 4 1 5 1 2 3 R
R 3 5 1 2 1 2 4 2 1 2 1 5 3 R
R 3 2 1 5 1 1 1 1 5 1 2 3 R
R 3 5 1 5 5 2 5 5 1 5 3 R
R 3 2 1 1 1 1 1 1 2 3 R
R 3 6 2 5 2 5 2 6 3 R
R R 3 3 3 3 3 3 R R
R R R R R R R R R

Figure 28: Material layout of the VVER-1000 benchmark.

A generic absorber of variable strength is inserted in the fuel assembly marked with a cross (×) in Figure 28, on the plane 12. Such a noise source corresponds to a 10% perturbation of the cross sections Σ_{a1} and Σ_{a2}, i.e. $\delta\Sigma_{a1} = 1.06731 \times 10^{-3}$ and $\delta\Sigma_{a2} = 8.85869 \times 10^{-3}$. Kinetic data of this benchmark problem is shown in Table 9.

Table 10 and Table 11 show the convergence of the solution depending on the methodology employed and the polynomial degree used in the FEM shape functions (FED). To compare the solutions, we have defined the following error indicators:

$$\Delta k_{eff} = k_{eff} - k_{eff}^{*}, \tag{36}$$

$$\varepsilon_g = 100 \times \frac{1}{N_c} \sum_{c=1}^{N_c} \frac{\phi_{c,g} - \phi_{c,g}^{*}}{\phi_{c,g}^{*}} \ \%, \qquad g = 1,\, 2. \tag{37}$$

$$\zeta_g = 100 \times \frac{1}{N_c}\sum_{c=1}^{N_c} \frac{\left|\delta\phi_{c,g}\right| - \left|\delta\phi^*_{c,g}\right|}{\left|\delta\phi^*_{c,g}\right|} \ \%, \qquad g = 1, 2. \tag{38}$$

$$\eta_g = 100 \times \frac{1}{N_c}\sum_{c=1}^{N_c} \frac{\arg\left(\delta\phi_{c,g}\right) - \arg\left(\delta\phi^*_{c,g}\right)}{\arg\left(\delta\phi^*_{c,g}\right)} \ \%, \qquad g = 1,\ 2. \tag{39}$$

where the values with * represent the reference results that are extracted with a very accurate time-domain calculation obtained with $\mathrm{FED} = 5$ and $\Delta t = 10^{-4}$. The quantities $\phi^*_{a,g}$ and $\delta\phi^*_{a,g}$ are the steady state mean flux and the average noise flux, respectively, at the hexagonal cell c. The parameter N_c is the number of hexagonal cells in the reactor.

Table 9: Kinetic data for the 3D VVER-1000 benchmark problem.

$\boldsymbol{\beta}_{\mathbf{eff}}$	$\boldsymbol{\lambda}_{\mathbf{eff}}$	$\mathbf{v_1}$	$\mathbf{v_2}$
	(1/s)	(cm/s)	(cm/s)
0.0065	0.0767	1.8230E+7	4.1306E+5

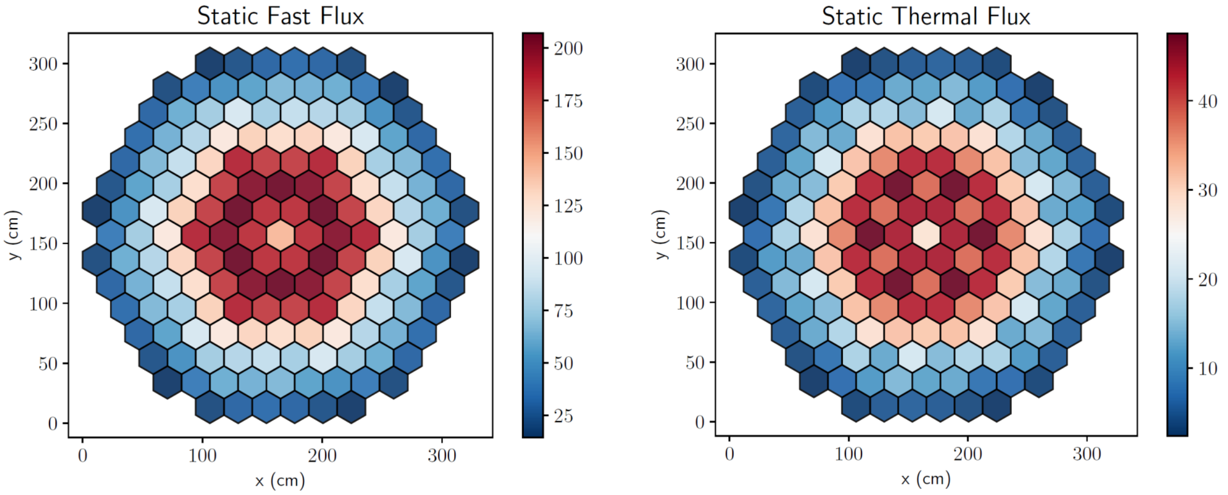
Table 10: Convergence table for the steady-state 3D VVER-1000 reactor.

Methodology	FED	Number of DoFs	k_{eff}	Δk_{eff}	ε_1	ε_2
				(pcm)	(%)	(%)
Time- and Frequency-domain	1	35150	1.02603	2036	4.60	6.06
Time- and Frequency-domain	2	265286	1.00830	263	0.29	0.45
Time- and Frequency-domain	3	877898	1.00595	28	0.11	0.15

Table 11: Convergence table for the noise of 3D VVER-1000 reactor.

Methodology	FED	ζ_1	ζ_2	η_1	η_2
		(%)	(%)	(%)	(%)
Frequency-domain	1	10.51	10.58	0.03	0.03
Frequency-domain	2	1.58	1.58	0.04	0.04
Frequency-domain	3	0.42	0.42	0.04	0.04
Time-domain	1	11.63	11.70	0.02	0.02
Time-domain	2	1.23	1.25	0.00	0.00
Time-domain	3	0.19	0.20	0.00	0.00

Figure 29 represents the average plane of the static assembly flux values for the steady state solution using $\mathrm{FED} = 3$. Table 10 shows the static results for the VVER-1000 benchmark for different FED. As we use the same code to calculate the steady-state of the reactor, the results for the frequency-domain and time-domain methodologies are exactly identical. Only $\mathrm{FED} = 2$ and $\mathrm{FED} = 3$ provide accurate results for this benchmark.



(a) Fast neutron flux.

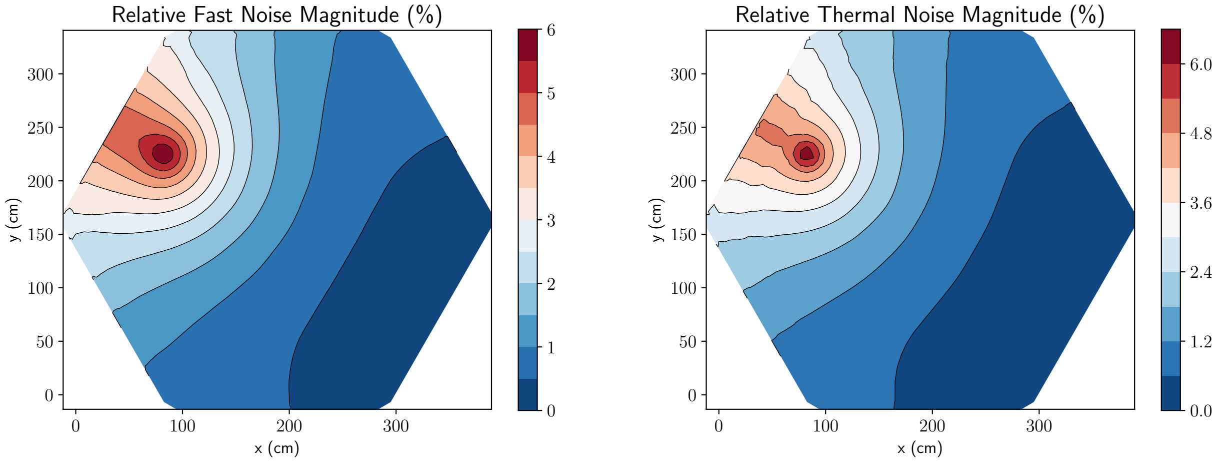
(b) Thermal neutron flux.

Figure 29: Steady state neutron flux at the 3D VVER reactor.

Figure 30 presents the relative noise magnitude for both the fast and the thermal fluxes. Figure 31 displays the neutron noise phase obtained with $\mathrm{FED} = 3$ and with the frequency-domain approach. The results show that the thermal neutron noise is mostly localized while the fast neutron noise has influence over a wider region. Also, for this perturbation, the phase of the neutron noise is similar throughout the entire reactor. Table 11 shows the neutron noise results comparison for the frequency-domain and time-domain methodologies employed and the FED ranging from 1 to 3. This Table shows that the differences between the frequency-domain and time-domain methodologies using the same FED are small, validating both methodologies. Also, calculations with linear shape functions do not provide accurate enough results.

These results verify both the frequency-domain methodology and the time-domain methodology against a generic absorber of variable strength in a selected location in a three-dimensional hexagonal reactor.

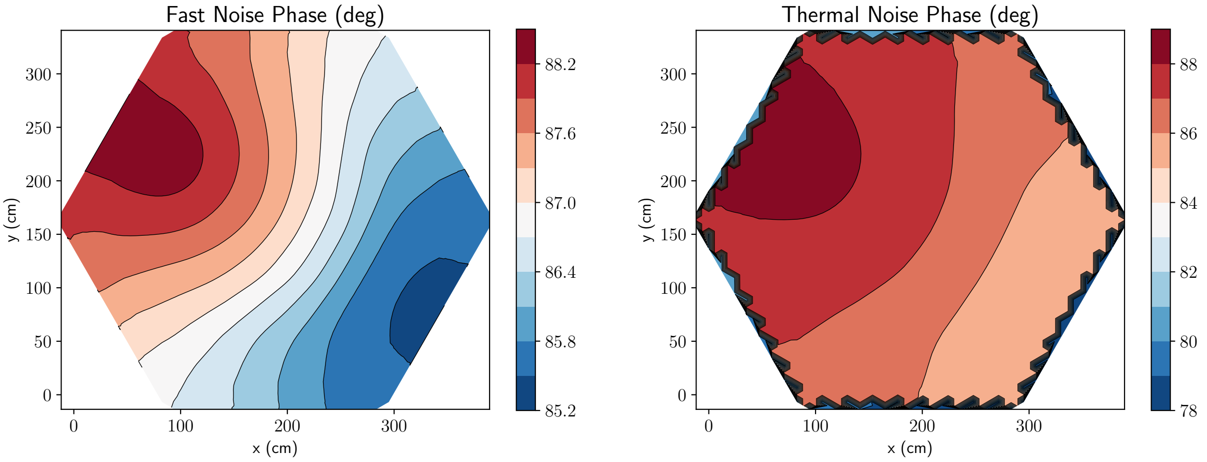
FEMFFUSION-FD has been used to provide several sets of simulated data for the VVER-440 and VVER-1000 reactors considered in the CORTEX project. More information about these sets and their use for training machine learning algorithms is provided in the Deliverable D4.4 of the CORTEX project [49].



(a) Relative fast flux noise.

(b) Relative thermal flux noise.

Figure 30: Relative noise amplitudes at the 3D VVER reactor.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 31: Noise phase at the midplane of 3D VVER reactor.

2.4.4 Two-dimensional CROCUS reactor

The CROCUS reactor is an experimental zero-power nuclear reactor situated at EPFL [50]. The nuclear reactor core is composed of uranium-based fuel rods contained in an aluminium vessel. The aluminium vessel is filled with demineralized light water to serve as both a neutron moderator and a neutron reflector. The reactor was used to carry out the COLIBRI experiments [8], in which a cluster of fuel pins was vibrated via a mechanical device. Figure 32 shows an image of the CROCUS reactor.

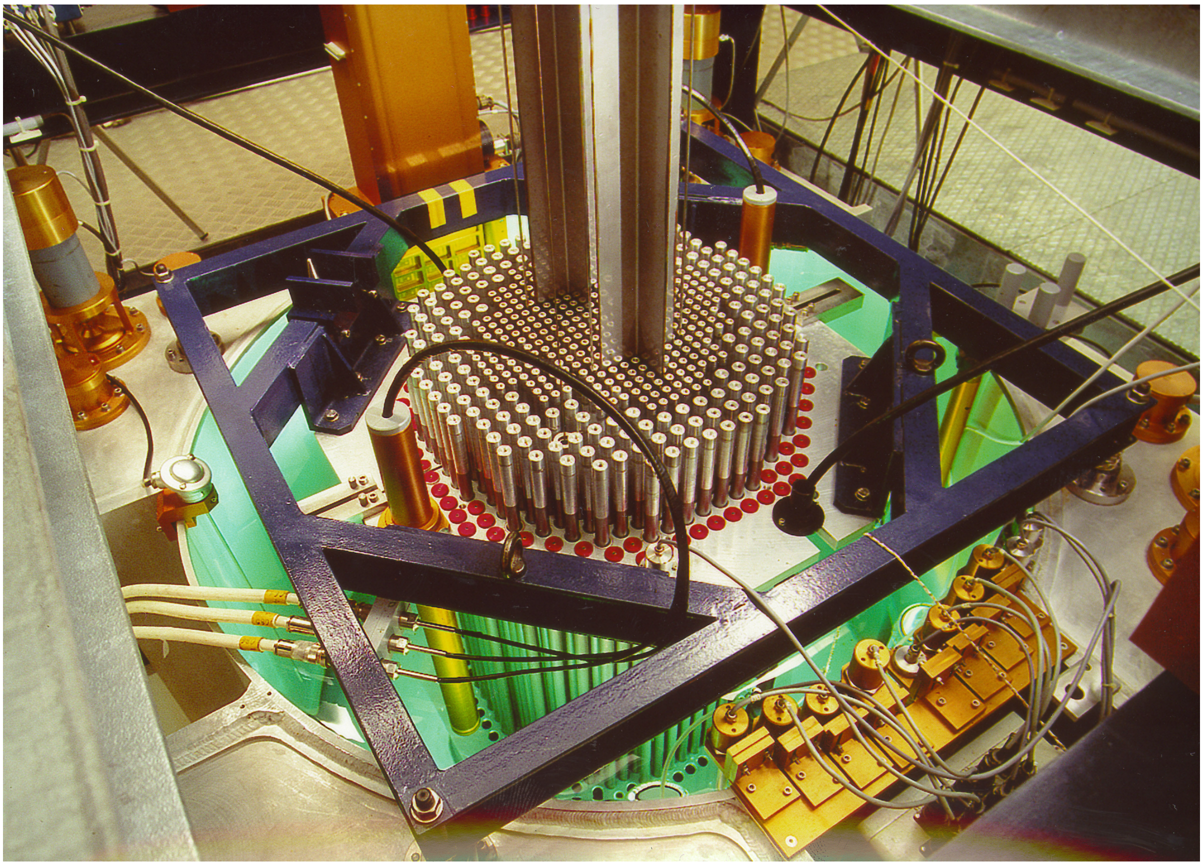


Figure 32: The CROCUS reactor (courtesy of EPFL).

This reactor is modelled using a two-dimensional mesh. Cross sections for the diffusion approximation and kinetic data can be found in [8]. Two grids were proposed:

- A uniform refined mesh with a 44×44 cells, each cell of $2.917\ \mathrm{cm} \times\ 2.917\ \mathrm{cm}$. The uniform mesh is shown in Figure 33.
- A locally refined mesh around the region where the fuel rods vibrate, perpendicular to the movement of the fuel rods. Then the original cell is composed of 96×44 cells. The locally refined mesh is displayed in Figure 34.

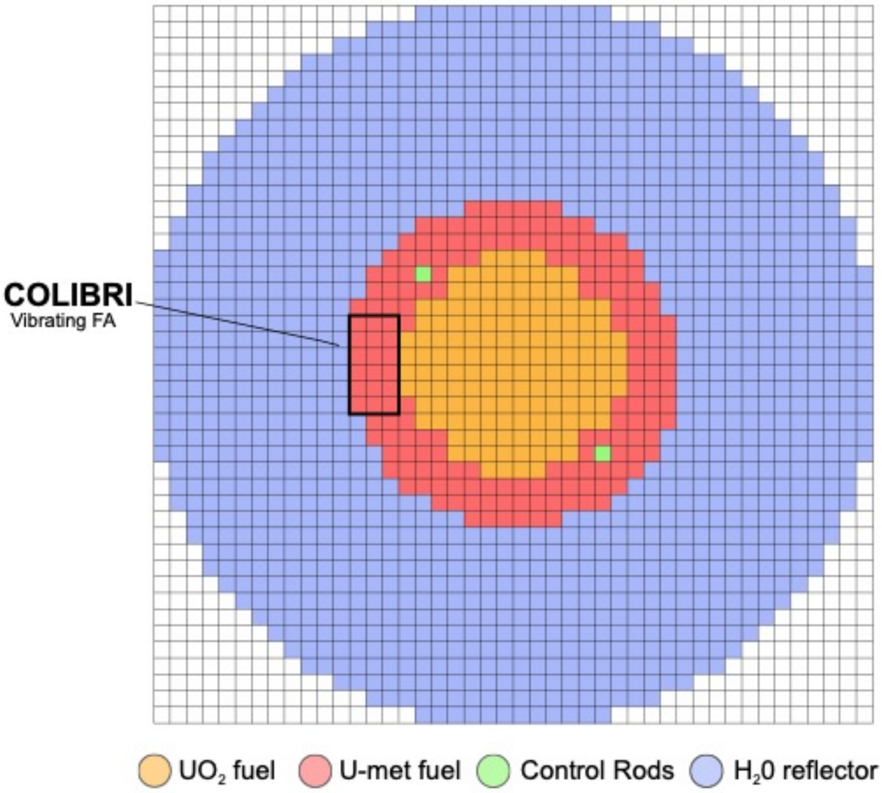


Figure 33: Uniform refined mesh for the CROCUS reactor.

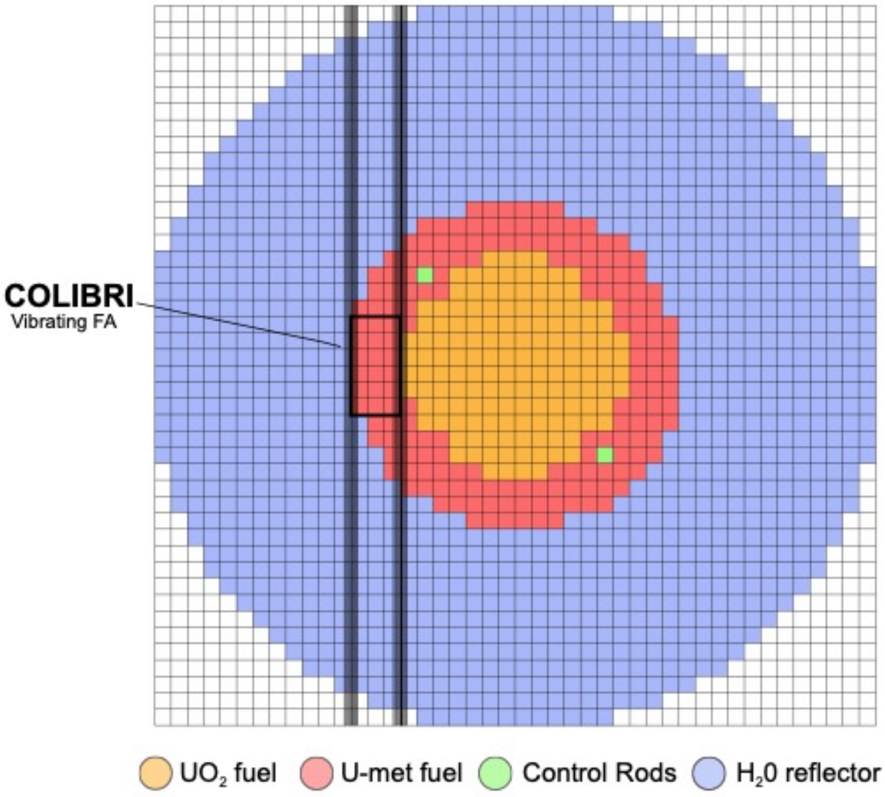


Figure 34: Locally refined mesh around the vibrating cluster of fuel rods, for the CROCUS reactor

Figure 35 shows the static flux for the midline of the CROCUS reactor ($y = 64.17$ cm) computed with both the FEMFFUSION and PARCS time-domain codes. A close agreement can be seen between both strategies.

Next, we show some simulation results for the first campaign of the COLIBRI experiments. In particular, experiment 12 is studied. In this experiment, the vibration amplitude is 2 mm and the frequency is 0.1 Hz. Figure 36 shows the neutron flux noise, $\delta\phi(\vec{r},t) = \phi(\vec{r},t) - \phi(\vec{r},0)$, at 0.0 s, 1.25 s, 2.5 s and 7.5 s at the centre line of the reactor $y = 64.17$ cm, calculated over the uniform refined mesh, with PARCS and FEMFFUSION. Figure 37 shows the neutron flux noise at the same times and the same centre line using the refined mesh around the vibrating region. Far from the vibrating cluster of fuel rods, the neutron noise results seem to be similar and, also, they are similar to the static flux.

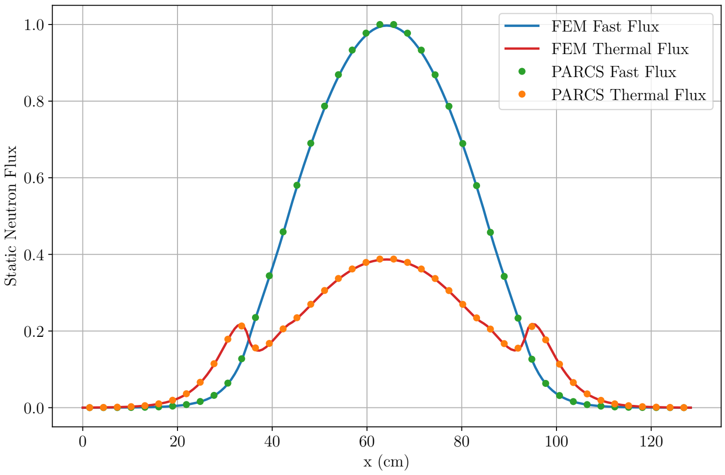
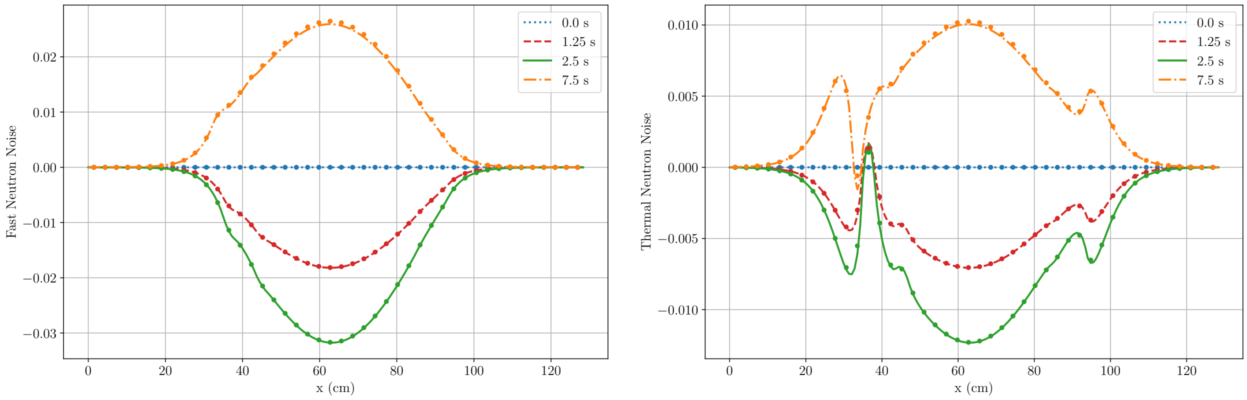
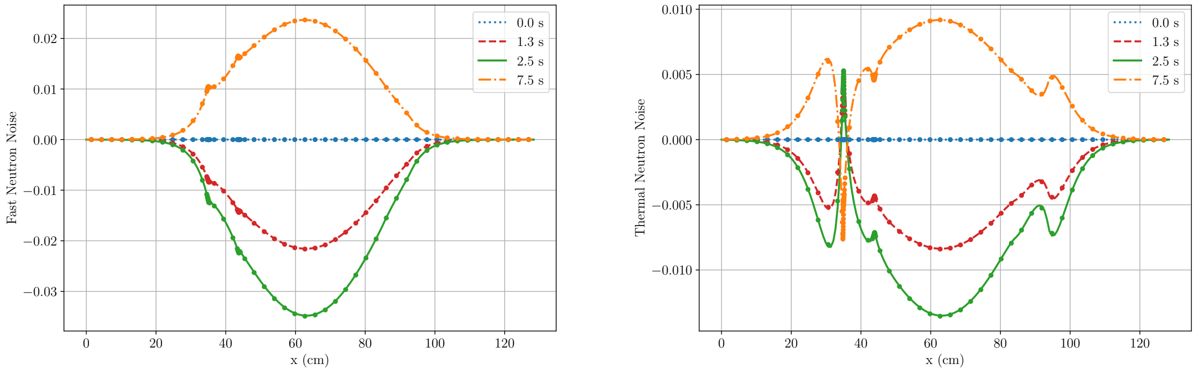


Figure 35: Static flux for the midline in the CROCUS refined reactor.



(a) Fast flux noise. **(b) Thermal flux noise.**

Figure 36: Neutron noise at different times with a uniform mesh; line: FEMFFUSION and dots: PARCS.



(a) Fast flux noise. **(b) Thermal flux noise.**

Figure 37: Neutron noise at different times with a locally refined mesh; line: FEMFFUSION and dots: PARCS.

Then, the time-domain results are transformed to the frequency-domain using the FFT algorithm. Figure 38 shows the spectrum of the neutron noise at the centre of the CROCUS reactor. This figure demonstrates that the neutron noise is mostly monochromatic, in other words, the perturbation only causes neutron noise of the same frequency as the perturbation, at least, for this reactor and when using the diffusion approximation. So, from now to the end of this section, only the perturbation frequency will be taken into account.

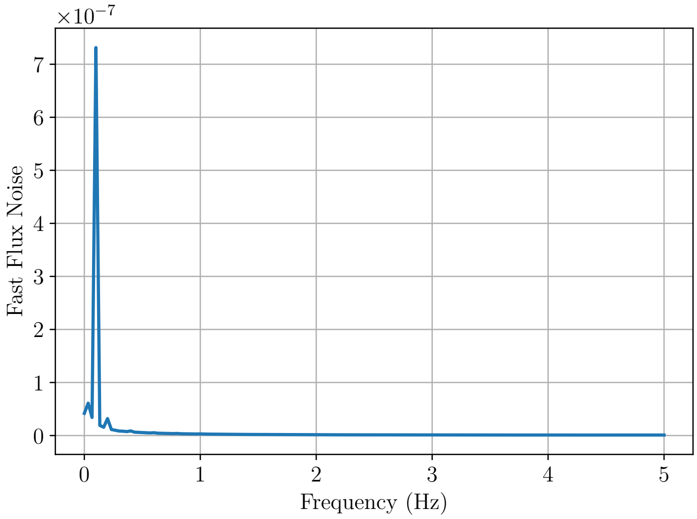
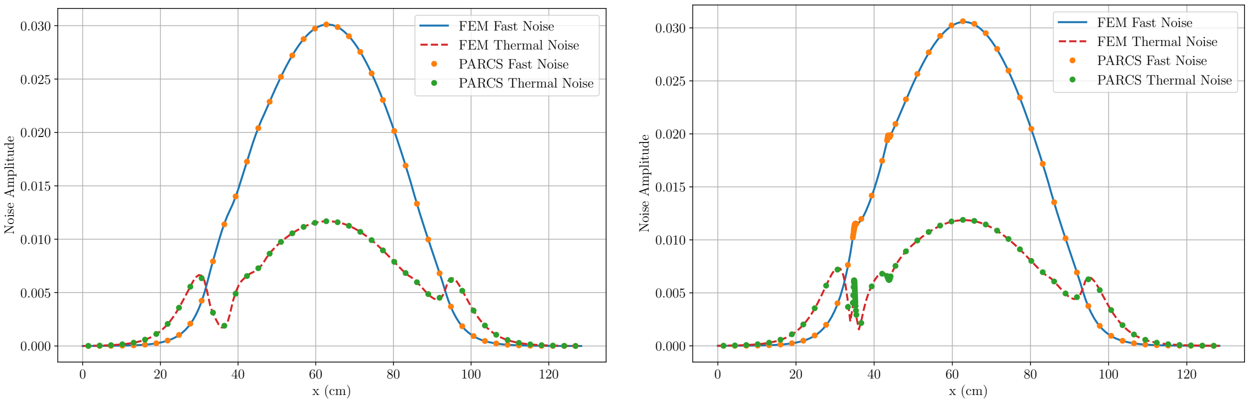


Figure 38: Spectrum of the neutron noise at the centre of the CROCUS reactor.

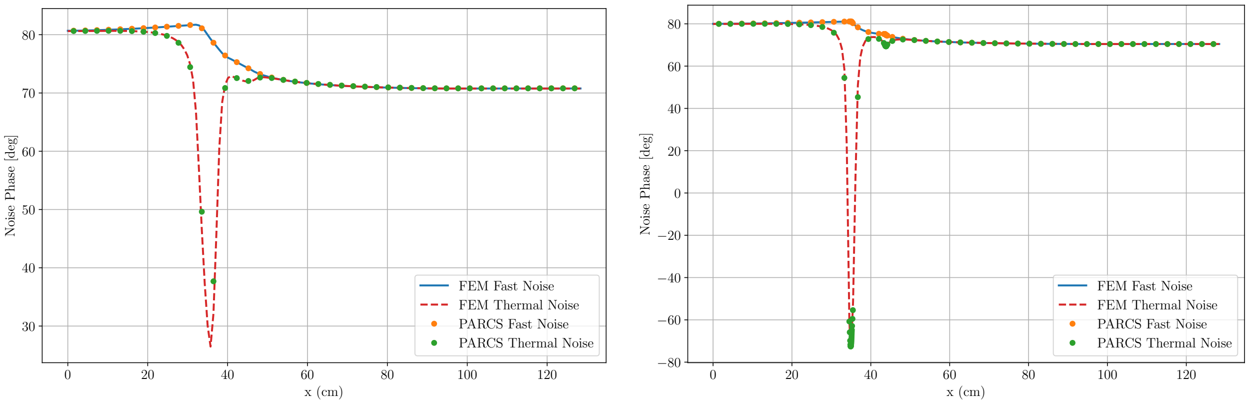
Figure 39 presents the noise amplitude for the midline of the CROCUS reactor for both the uniform refined mesh and the locally refined mesh. The PARCS and the FEMFFUSION codes display a very close agreement in both grids. Also, these figures show that far away from the perturbation the noise amplitude is similar for both the uniform and the local refined mesh. Also, far away from the perturbation these figures are similar to the steady-state neutron flux. This indicates that the calculated neutron noise away from the noise source behaves in a point kinetic way. In the surroundings of the vibrating assembly the neutron noise presents important differences, mainly in the location of the vibrating perturbation, where some fast change in neutron noise is observed. This behaviour is known to be inaccurately captured by the neutron diffusion approximation. In this way, the results in the surrounding of vibrating fuel rods are expected not to be accurate in diffusion simulations and higher order transport simulations must be performed. However, far from the vibrating fuel rods, the results are validated.

Figure 40 shows the noise phase for the midline of the CROCUS reactor for both the uniform and the refined grids computed with PARCS and FEMFUSION. The agreement between both time-domain codes is very good. The figures show that the change in the phase of the neutron noise is quite small far from the vibrating fuel rods. Also, a significant difference can be found in the surrounding of the vibrating fuel rods.



(a) Uniform refinement. **(b) Local refinement.**

Figure 39: Noise amplitude at the midline of the CROCUS reactor during experiment 12.

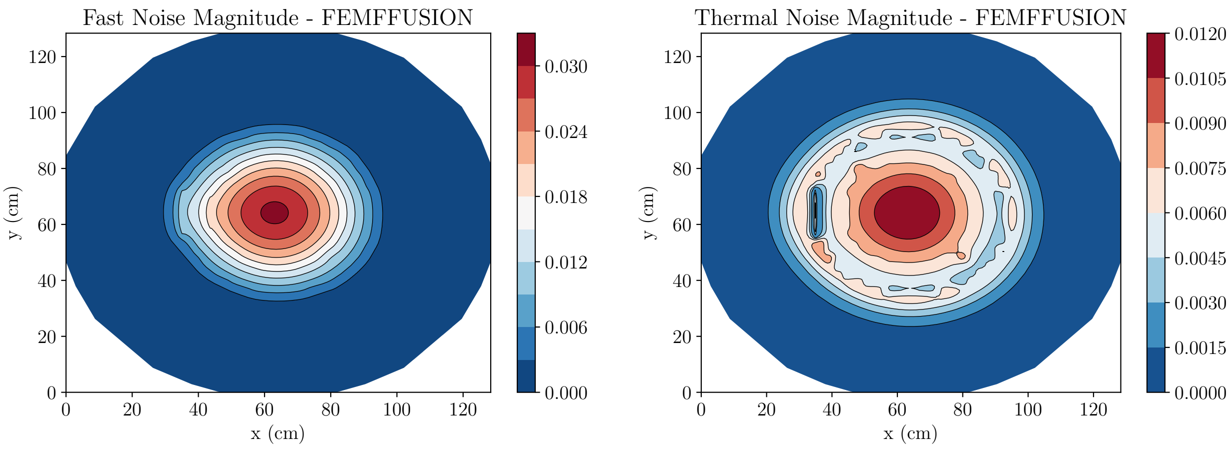


(a) Uniform refinement. **(b) Local refinement.**

Figure 40: Noise phase at the midline of the CROCUS reactor during experiment 12.

Figure 41 shows the neutron noise amplitude in the midplane of the CROCUS reactor using the FEMFFUSION code. On the other hand, Figure 42 displays the neutron noise phase in the midplane of the CROCUS reactor using the same code.

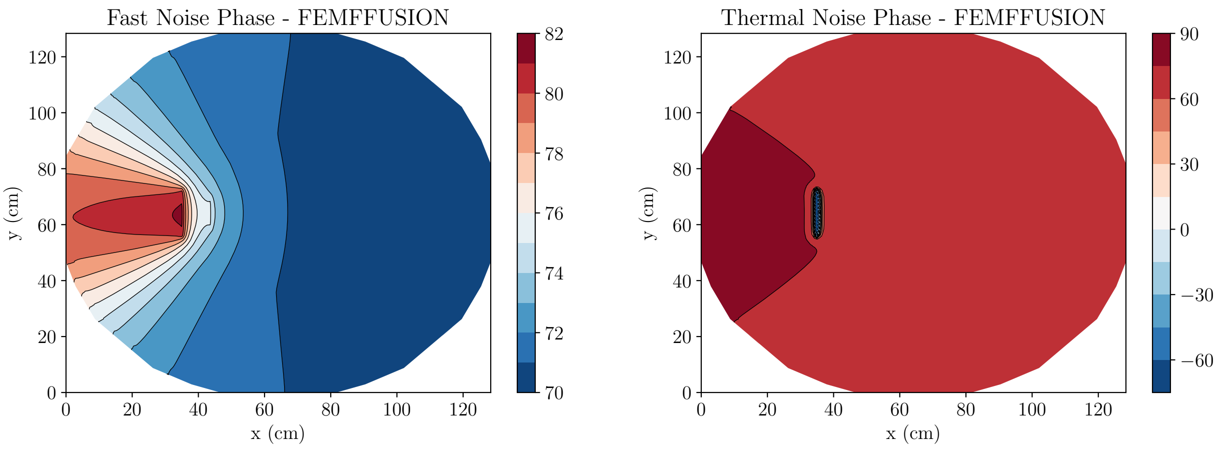
The results of the CROCUS reactor will be compared with the experimental results at the detectors position. Figure 43 shows the time-dependent thermal neutron flux noise at the position of the detectors. The simulation results and the experimental results will be compared in future deliverables.



(a) Fast flux noise.

(b) Thermal flux noise.

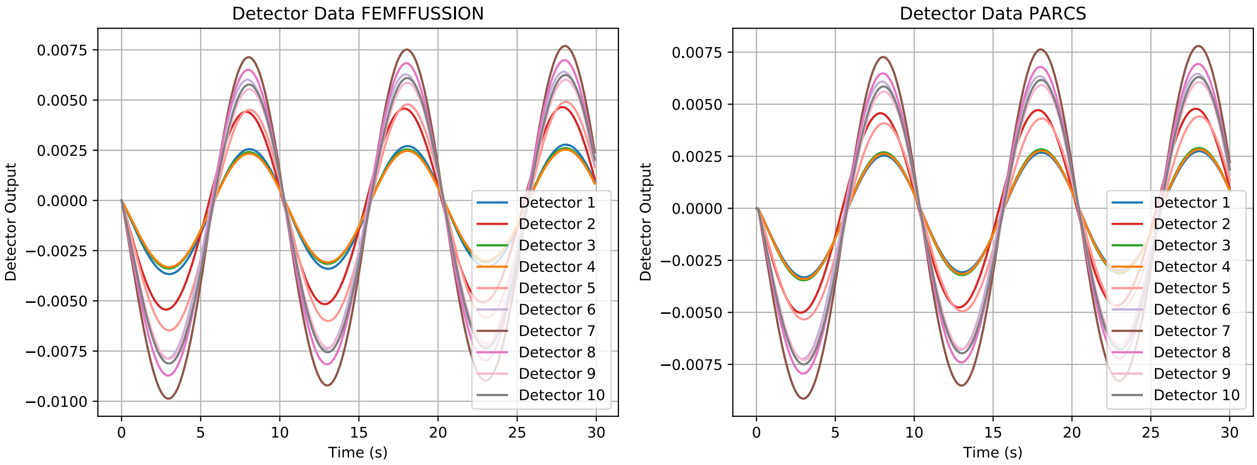
Figure 41: Noise amplitude in the CROCUS reactor computed with FEMFFUSION.



(a) Fast flux noise.

(b) Thermal flux noise.

Figure 42: Noise phase in the CROCUS reactor computed with FEMFFUSION.



(a) FEMFFUSION.

(b) PARCS.

Figure 43: Computed thermal neutron noise at detector locations.

3 Modelling strategy at Chalmers

The computational tool *CORE SIM+* is developed at Chalmers University of Technology to model the reactor transfer function and simulate scenarios induced by neutron noise sources such as vibrations of fuel assemblies in a nuclear core. The simulator relies on the previous experience of CORE SIM [35] and its new improved numerical scheme allows a more accurate modelling of neutron noise problems and more efficient numerical performances. Such a scheme can make use of uniform or non-uniform meshes for the spatial discretization of the neutron balance equations, and acceleration techniques and preconditioning for increasing the convergence rate of the solution.

The section is structured as follows. The governing equations of neutron noise and the numerical methods used in *CORE SIM+* are presented. The modelling of the neutron noise sources that is needed for *CORE SIM+*, is introduced together with the 2 possible strategies for the calculation of the neutron noise, i.e. the direct simulation and the method of the Green's function. The verification and validation work for *CORE SIM+* is summarized. The simulation of 2 neutron noise scenarios induced by vibrations in a Pressurized Water Reactor (PWR) are discussed, namely the case of the vibration of a fuel assembly and the case of a pendular vibration of the core barrel represented as a collective oscillation of the fuel assemblies.

3.1 Governing equations

The tool *CORE SIM+* is based on a two-energy group diffusion model. The calculation scheme first requires the solution of the criticality problem associated with the system under study. After the critical neutron flux is determined, the neutron noise is obtained from the dynamic equations in the frequency-domain.

For the calculation of the critical neutron flux $\boldsymbol{\phi}$ in a nuclear reactor, *CORE SIM+* uses the 3-dimensional, 2-energy group neutron diffusion equations:

$$\begin{aligned}\left\{\nabla\cdot\begin{bmatrix}D_1(\boldsymbol{r}) & 0\\ 0 & D_2(\boldsymbol{r})\end{bmatrix}\nabla+\begin{bmatrix}-\Sigma_{a,1}(\boldsymbol{r})-\Sigma_r(\boldsymbol{r}) & 0\\ \Sigma_r(\boldsymbol{r}) & -\Sigma_{a,2}(\boldsymbol{r})\end{bmatrix}\right\}\times\begin{bmatrix}\phi_1(\boldsymbol{r})\\ \phi_2(\boldsymbol{r})\end{bmatrix}\\ =\frac{1}{k}\begin{bmatrix}-\nu\Sigma_{f,1}(\boldsymbol{r}) & -\nu\Sigma_{f,2}(\boldsymbol{r})\\ 0 & 0\end{bmatrix}\times\begin{bmatrix}\phi_1(\boldsymbol{r})\\ \phi_2(\boldsymbol{r})\end{bmatrix}.\end{aligned} \tag{40}$$

The simulator also solves the associated adjoint problem which is defined as:

$$\begin{aligned}\left\{\nabla\cdot\begin{bmatrix}D_1(\boldsymbol{r}) & 0\\ 0 & D_2(\boldsymbol{r})\end{bmatrix}\nabla+\begin{bmatrix}-\Sigma_{a,1}(\boldsymbol{r})-\Sigma_r(\boldsymbol{r}) & 0\\ \Sigma_r(\boldsymbol{r}) & -\Sigma_{a,2}(\boldsymbol{r})\end{bmatrix}^T\right\}\times\begin{bmatrix}\phi_1^\dagger(\boldsymbol{r})\\ \phi_2^\dagger(\boldsymbol{r})\end{bmatrix}\\ =\frac{1}{k^\dagger}\begin{bmatrix}-\nu\Sigma_{f,1}(\boldsymbol{r}) & -\nu\Sigma_{f,2}(\boldsymbol{r})\\ 0 & 0\end{bmatrix}^T\times\begin{bmatrix}\phi_1^\dagger(\boldsymbol{r})\\ \phi_2^\dagger(\boldsymbol{r})\end{bmatrix}.\end{aligned} \tag{41}$$

Considering a neutron noise source in the critical system, the induced neutron noise $\boldsymbol{\delta\phi}$ is evaluated with the following dynamic equations in the frequency-domain:

$$\left\{\nabla\cdot\begin{bmatrix}D_1(\boldsymbol{r}) & 0\\ 0 & D_2(\boldsymbol{r})\end{bmatrix}\nabla+\Sigma_{dyn}^{crit}(\boldsymbol{r},\omega)\right\}\times\begin{bmatrix}\delta\phi_1(\boldsymbol{r},\omega)\\ \delta\phi_2(\boldsymbol{r},\omega)\end{bmatrix}=\begin{bmatrix}S_1(\boldsymbol{r},\omega)\\ S_2(\boldsymbol{r},\omega)\end{bmatrix}. \tag{42}$$

The neutron noise source on the right-hand side of equation (42) is modelled as small fluctuations of the macroscopic cross sections, i.e.

$$\begin{bmatrix} S_1(\boldsymbol{r},\omega) \\ S_2(\boldsymbol{r},\omega) \end{bmatrix} = \boldsymbol{\phi}_r(\boldsymbol{r})\delta\Sigma_r(\boldsymbol{r},\omega) + \boldsymbol{\phi}_a(\boldsymbol{r}) \begin{bmatrix} \delta\Sigma_{a,1}(\boldsymbol{r},\omega) \\ \delta\Sigma_{a,2}(\boldsymbol{r},\omega) \end{bmatrix} + \boldsymbol{\phi}_f^{crit}(\boldsymbol{r},\omega) \begin{bmatrix} \delta\nu\Sigma_{f,1}(\boldsymbol{r},\omega) \\ \delta\nu\Sigma_{f,2}(\boldsymbol{r},\omega) \end{bmatrix}. \tag{43}$$

The notation used in equations (40) to (43) is standard and the details of it can be found in, e.g., [51].

3.2 Numerical methods

The physical system is discretized using a rectilinear grid that is selected before the calculation and is kept fixed for all the analyses. The choice of a rectilinear grid allows the generation of non-uniform meshes. Thus, it has the advantage that a finer resolution can be specified for regions where perturbations are very localized and the gradient of the neutron flux is strong, while coarser cells are used for those regions where the spatial variation of the system properties and of the neutron flux is less remarkable. Accordingly, computational effort is saved without compromising the accuracy.

The 1D grid shown in Figure 44, in which the cell/node n is of size Δx_n, is used to illustrate the discretization of equations (40) to (43). The scalar neutron flux for the energy group g is averaged over the node as:

$$\phi_{g,n} = \frac{1}{\Delta x_n}\int_{\Delta x_n} \phi_g(x)dx \tag{44}$$

The macroscopic cross sections are node-averaged as:

$$\Sigma_{g,n} = \frac{1}{\Delta x_n}\frac{\int_{\Delta x_n} \Sigma_g(x)\phi_g(x)dx}{\phi_{g,n}} \tag{45}$$

The diffusion operator is approximated using a finite difference method, i.e.

$$\begin{aligned} \frac{1}{\Delta x_n}\int_{\Delta x_n} \nabla\cdot\left[D_g(x)\nabla\phi_g(x)\right]dx &= -\frac{\left(D_g\nabla\phi_g\right)_{n+\frac{1}{2}} - \left(D_g\nabla\phi_g\right)_{n-\frac{1}{2}}}{\Delta x_n} \\ &= -\left(a_{g,n}\phi_{g,n} + b_{g,n}\phi_{g,n+1} + c_{g,n}\phi_{g,n-1}\right) \end{aligned} \tag{46}$$

The coupling coefficients $a_{g,n}$, $b_{g,n}$, $c_{g,n}$ depend on the boundary conditions. In *CORE SIM+*, Marshak and reflective Boundary Conditions (BC) are available, which read as:

$$\text{Marshak BC:} \quad \left(D_g\nabla\phi_g\right)_b \cdot \boldsymbol{n}_x^+ = \frac{1}{2}\phi_g^b, \tag{47}$$

$$\text{Reflective BC:} \quad \left(D_g \nabla \phi_g\right)_b \cdot \boldsymbol{n}_x^+ = 0, \tag{48}$$

The vector $\boldsymbol{n}_x^+$ identifies the outward normal to the boundary and ϕ_g^b is the scalar neutron flux at the boundary. The expression of the coupling coefficients for a non-uniform mesh is given in [51].

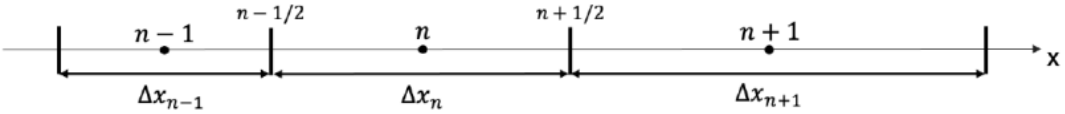


Figure 44: Example of 1D non uniform rectilinear grid.

After the spatial discretization of the equations, the static problem of equation (40), the adjoint problem of equation (41) and the neutron noise problem of equations (42) – (43) can be respectively rewritten in matrix form as:

$$\boldsymbol{A}_{crit}\boldsymbol{\Phi}_{crit} = \frac{1}{k}\boldsymbol{F}\boldsymbol{\Phi}_{crit}, \tag{49}$$

$$\boldsymbol{A}_{crit}^{\dagger}\boldsymbol{\Phi}_{crit}^{\dagger} = \frac{1}{k^{\dagger}}\boldsymbol{F}^{\dagger}\boldsymbol{\Phi}_{crit}^{\dagger}, \tag{50}$$

$$\boldsymbol{A}_{noise}\boldsymbol{\delta\Phi}_{noise} = \boldsymbol{S}_{noise}. \tag{51}$$

In equations (49) to (51), the coefficient matrices are large, sparse and banded, and their size is $2N \times 2N$ since the number of nodes of the spatial grid is N and the neutron energy groups are 2. The matrices are constructed using the coordinate format. 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.

Numerical methods suitable for a flexible neutron noise simulator that allow fast convergence rates, were investigated and discussed in [51]. Here the methods used in *CORE SIM+* are summarized. The steady-state system given in equation (49) and the adjoint system in (50) are eigenvalue problems and three options are available in *CORE SIM+* for their solution. The first method is the standard non-accelerated Power Method (PM). The second option is PM accelerated with Chebyshev polynomials [52]. The third alternative is PM combined with a nonlinear acceleration based on the Jacobian Free Newton Krylov (JFNK) algorithm, as proposed by Gill and Azmy [53]. In reactor static calculations, Mylonakis et al. showed that the JFNK-accelerated PM can meet tight convergence criteria which are not always feasible for the Chebyshev accelerated PM [51].

Linear systems are generated from each iteration of PM when solving the eigenvalue problems. In addition, the neutron noise problem requires the solution of the linear system given by equation (51). As linear solver, *CORE SIM+* applies the iterative Generalized Minimal RESidual (GMRES) method. The acceleration of the convergence is obtained from a preconditioner, that can be chosen between the Symmetric Gauss-Seidel (SGS) preconditioner and the Incomplete LU with zero fill-in – ILU(0) preconditioner. Externally constructed preconditioners can also be provided as inputs to the solver.

3.3 Modelling of neutron noise sources

The tool *CORE SIM+* allows the simulation of neutron noise induced by different types of perturbations. These perturbations are described in terms of fluctuations of macroscopic neutron cross sections according to (43) and can reproduce the absorber of variable strength, the vibration of fuel assemblies, the vibration of the core barrel, the vibration of control rods, and perturbations that are transported by the coolant flow along the axial direction of the core. A detailed discussion on the modelling of these noise sources is available in [54]. A summary is provided below.

3.3.1 Absorber of variable strength

An absorber of variable strength is a point-like source and can be described by the Dirac delta function with respect to space. Therefore, the right-hand side of equation (42) is given by:

$$S_g(\boldsymbol{r},\omega) = \delta\left(\boldsymbol{r} - \boldsymbol{r}_p\right). \tag{52}$$

The coordinate $\boldsymbol{r}_p$ identifies the position of the perturbation.

The neutron noise induced in the energy group g' at a generic position $\boldsymbol{r}$ by an absorber of variable strength for the energy group g and located at position $\boldsymbol{r}_p$, corresponds to the component $G_{g\to g'}\left(\boldsymbol{r},\boldsymbol{r}_p,\omega\right)$ of the Green's function, see section 3.4.

3.3.2 Fuel assembly vibration

The vibration of a fuel assembly is modelled using the so-called ϵ/d approximation [2]. Accordingly, the macroscopic cross sections are perturbed with δ functions at the interfaces between the vibrating fuel assembly and the surrounding regions. The schematic in Figure 45 shows a one-dimensional configuration with three homogenized fuel assemblies, of which the fuel assembly II vibrates in the x-direction. In this case, the fluctuation of the generic cross section $\Sigma_{\alpha,g}$ in the frequency-domain is given as:

$$\begin{aligned}\delta\Sigma_{\alpha,g}(x,z,\omega) = {} & h(z)\epsilon_x(\omega)\delta(x-a_0)\left[\Sigma_{\alpha,g,I} - \Sigma_{\alpha,g,II}\right] \\ & + h(z)\epsilon_x(\omega)\delta(x-b_0)\left[\Sigma_{\alpha,g,II} - \Sigma_{\alpha,g,III}\right].\end{aligned} \tag{53}$$

The quantity $\epsilon_x(\omega)$ is the dimensionless displacement of the vibrating fuel assembly along x, $h(z)$ is the axial shape of the perturbation, a_0 is the equilibrium position of the boundary between the fuel assemblies I and II, and b_0 is the equilibrium position of the boundary between the fuel assemblies II and III. Equation (53) is introduced in (43) so that the proper noise source is built for the evaluation of the neutron noise.

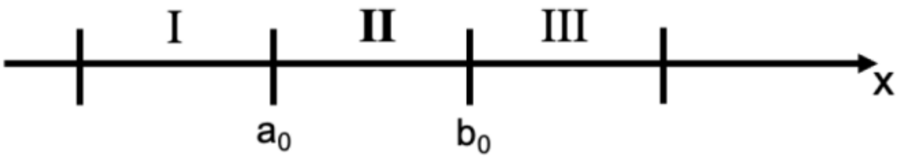


Figure 45: Schematic of 3 fuel assemblies with fuel assembly II vibrating in the x-direction and fuel assembly I and III being fixed.

3.3.3 Core barrel pendular vibration

The vibration of the core barrel may occur according to different modes. Only the pendular mode of the core barrel vibration is considered. It consists of a relative oscillation of the active fuel core with respect to the reflector, so it is equivalent to a collective movement of all the fuel assemblies. This type of core barrel vibration is modelled by introducing a perturbation at the boundary between the active core and reflector regions. The perturbations for all the outer fuel assemblies are computed with equation (53). The second term on the right-hand side of (53) is zero for the fuel assemblies located at the left boundary between the core and the reflector, whereas the first term is zero for the fuel assemblies located at the right boundary.

3.3.4 Control rod vibration

For the simulation of the neutron noise induced by the vibration of a control rod cluster in the core, the so-called weak absorber model developed by Pázsit is used [2]. The neutron noise is thus provided by combining the Green's function and the static flux through the relation:

$$\delta\phi_g(\boldsymbol{r},\omega) = \gamma\epsilon(\omega)\nabla_{x_0,y_0}\int\limits_{z'>z_0} G_{2\to g}(\boldsymbol{r},x_0,y_0,z',\omega)\phi_2(x_0,y_0,z')dz' \tag{54}$$

The Green's function $G_{2\to g}(\boldsymbol{r},x_0,y_0,z',\omega)$ represents the neutron noise at position $\boldsymbol{r}=(x,y,z)$ in group g induced by Dirac perturbations specified in the thermal energy group, at radial position (x_0,y_0), along the entire axial length of the control rod.

3.3.5 Perturbations transported by the coolant flow in the axial direction

Fluctuating perturbations transported by the coolant flow along the z-direction, from the bottom to the top of the core, are modelled as variations of the macroscopic removal cross section, i.e.

$$\delta\Sigma_r(x,y,z,\omega) = \begin{cases} 0 & \text{if } (x,y) \neq (x_0,y_0) \\ 0 & if\ (x,y) \neq (x_0,y_0)\ and\ z < z_0 \\ \delta\Sigma_r(x_0,y_0,z_0,\omega)\exp\left[-\dfrac{i\omega(z-z_0)}{u_c}\right] & if\ (x,y) = (x_0,y_0)\ and\ z \geq z_0 \end{cases} \tag{55}$$

The quantity u_c is the axial velocity of the coolant. Equation (55) is used in (43) to define the source term for the neutron noise calculation.

3.4 Calculation of neutron noise

Once the model of the source term has been built, two different strategies for the calculation of the neutron noise can be followed with *CORE SIM+*. The first strategy is to provide the model of the neutron noise source to equation (42) via equation (43) and then calculate the neutron noise directly from (42). The second strategy is to first determine the Green's functions $G_{g\to g'}(\boldsymbol{r},\boldsymbol{r}',\omega)$ by solving equation (42) where the source term for the energy group g is replaced with the Dirac delta function and the source term for the other energy group is equal to zero. Then the neutron noise is evaluated from the convolution of the Green's function and the noise source, i.e.

$$\begin{bmatrix}\delta\phi_1(\boldsymbol{r},\omega)\\ \delta\phi_2(\boldsymbol{r},\omega)\end{bmatrix} = \begin{bmatrix}\int_V [G_{1\to1}(\boldsymbol{r},\boldsymbol{r}',\omega)S_1(\boldsymbol{r}',\omega) + G_{2\to1}(\boldsymbol{r},\boldsymbol{r}',\omega)S_2(\boldsymbol{r}',\omega)]d^3\boldsymbol{r}'\\ \int_V [G_{1\to2}(\boldsymbol{r},\boldsymbol{r}',\omega)S_1(\boldsymbol{r}',\omega) + G_{2\to2}(\boldsymbol{r},\boldsymbol{r}',\omega)S_2(\boldsymbol{r}',\omega)]d^3\boldsymbol{r}'\end{bmatrix} \quad (56)$$

The advantage of the second method is that the Green's function is calculated just only one time and then can be used to reconstruct the neutron noise induced by any perturbation.

3.5 Verification and validation of the tool

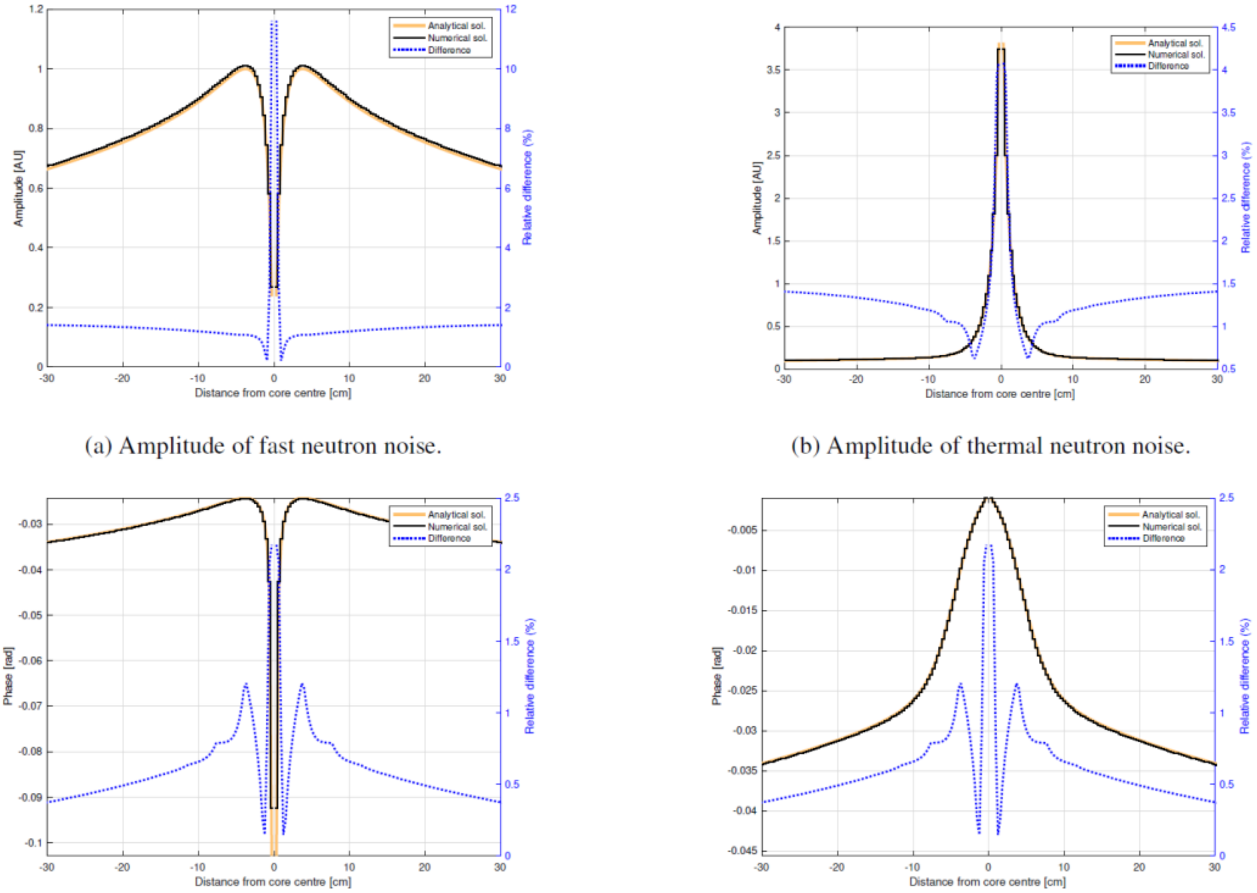
The tool *CORE SIM+* was verified on different neutron noise problems defined in 2-D and 3-D configurations. For the validation of *CORE SIM+*, neutron noise experiments carried out in the research reactors CROCUS (at EPFL) and AKR-2 (at TUD), were simulated. Most of this work is published, e.g., see [51] and [55]. Representative examples are reported below.

3.5.1 Two-dimensional one-region homogeneous nuclear reactor core

A 2-D one-region system near to criticality is considered. 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 dynamic nuclear parameters are representative of a PWR core, see details in [51]. The noise source is a point-like fluctuation of the macroscopic removal cross-section Σ_r at the center of the core with a frequency of $1\ Hz$. For the numerical solution, a fine spatial grid of 903×903 nodes, is applied. The reference semi-analytical critical and noise solutions is derived by Demazière and Andhill [37].

The numerical and the semi-analytical solutions of the neutron noise problem are compared in Figure 46. The overall agreement shows that the *CORE SIM+* calculation is correct. In the region close to the neutron noise source, the relative differences are larger, i.e. about 12% for the amplitude of the fast neutron noise, about 4% for the amplitude of the thermal neutron noise, about 2% for the phase of the noise. This is expected as the chosen spatial discretization might be unable to accurately simulate sharp neutron flux gradients.

The analysis of the numerical performance of the GMRES solver for the neutron noise simulation, shows that the preconditioner ILU(0) requires less iterations to reach tight convergence criteria than the preconditioner SGS, see Figure 47.



(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 46: Benchmarking of the numerical solution against the analytical solution for the 2-D one-region test case.

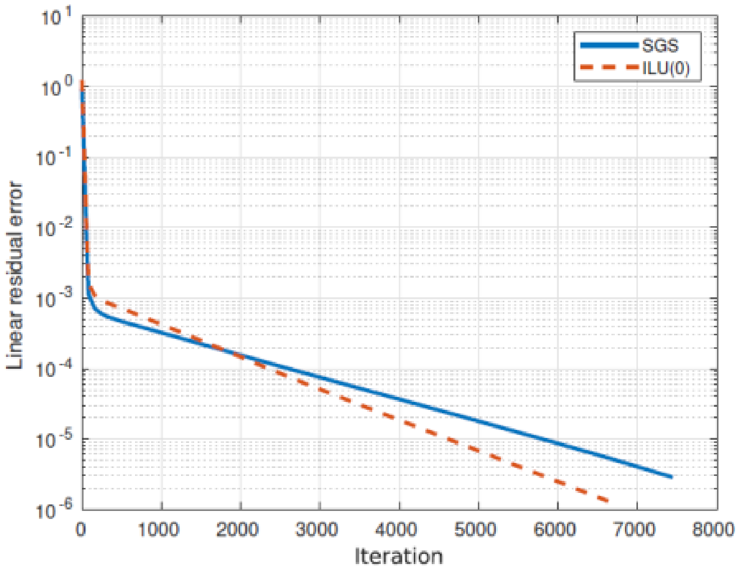


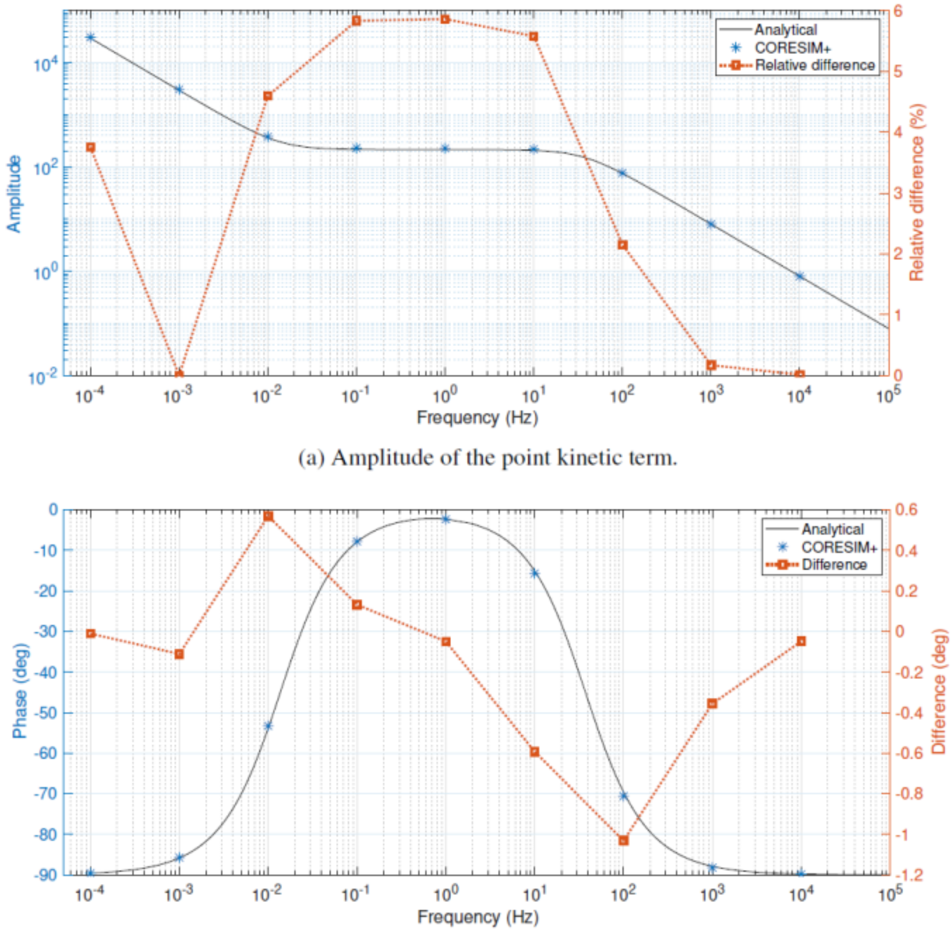
Figure 47: Convergence of the GMRES noise solver using SGS and ILU(0) preconditioners for the 2-D one-region test case.

3.5.2 Verification of the point-kinetic response of a 3-D heterogeneous PWR to a neutron noise source

The methodology described by Demazière et al. [47] is used for a verification test on *CORE SIM+*. Accordingly, the point-kinetic component of the response of a critical reactor to a perturbation is extracted from the induced neutron noise and is thereafter compared with the analytical one. The analytical point-kinetic component is derived from the point reactor kinetics equations (e.g., see [56]).

The test configuration is a MOX/UO2 core of a four-loop Westinghouse PWR [57]. The 3-dimensional spatial domain is discretized using a Cartesian mesh in which the radial grid is 32×32 nodes and the axial levels are 34. The grid for the active core region is $30 \times 30 \times 32$ nodes and the remaining external layers of nodes are for the modelling of the reflector. Marshak boundary conditions are applied at the periphery of the system. The size of each elementary node, assumed to be spatially homogeneous, is $10.71\ cm$ in both the x- and the y-direction and $11.43\ cm$ in the axial direction. A point-like fluctuation of the thermal absorption cross-section $\Sigma_{a,2}$ is prescribed at the grid location $(16,16,17)$.

The frequency-dependence of the point-kinetic zero-power transfer function is shown in Figure 48 and compared to its analytical expression. The point-kinetic term extracted from the CORE SIM+ simulations follows very satisfactory the expected analytical expression in the whole frequency range. A variation of the amplitude over a broad range of values and a variation of phase of more than 80 degrees are well reproduced. The maximum deviation is below 6% for the amplitude and about 1 degree for the phase.



(a) Amplitude of the point kinetic term.

(b) Phase of the point kinetic term.

Figure 48: Comparison between computed and analytical amplitude (a) and phase (b) of the point-kinetic zero-power reactor transfer function.

3.5.3 Simulation of neutron noise experiments in CROCUS

The tool *CORE SIM+* was used to simulate neutron noise experiments that were carried out in the research reactors CROCUS (EPFL), within the CORTEX project. The detailed discussion of this work is reported in [55]. The neutron noise experiments are part of the program COLIBRI and investigate the system response to the forced vibration of a group of fuel rods, see [8] and [58].

For the calculations, a 3-D model of the CROCUS reactor core is used. The top view of the reactor core (where the fuel rods moved in the COLIBRI experiments, are highlighted) and the related [8] *CORE SIM+* simulation grid are shown in Figure 49. The model consists of 3 homogenized regions and the mesh is much finer in the COLIBRI region so that the effect of the oscillation can be reproduced in a more accurate manner. The variation of the macroscopic cross sections induced by the movement of the fuel rods is modelled using the ϵ/d approximation, see section 3.3.2.

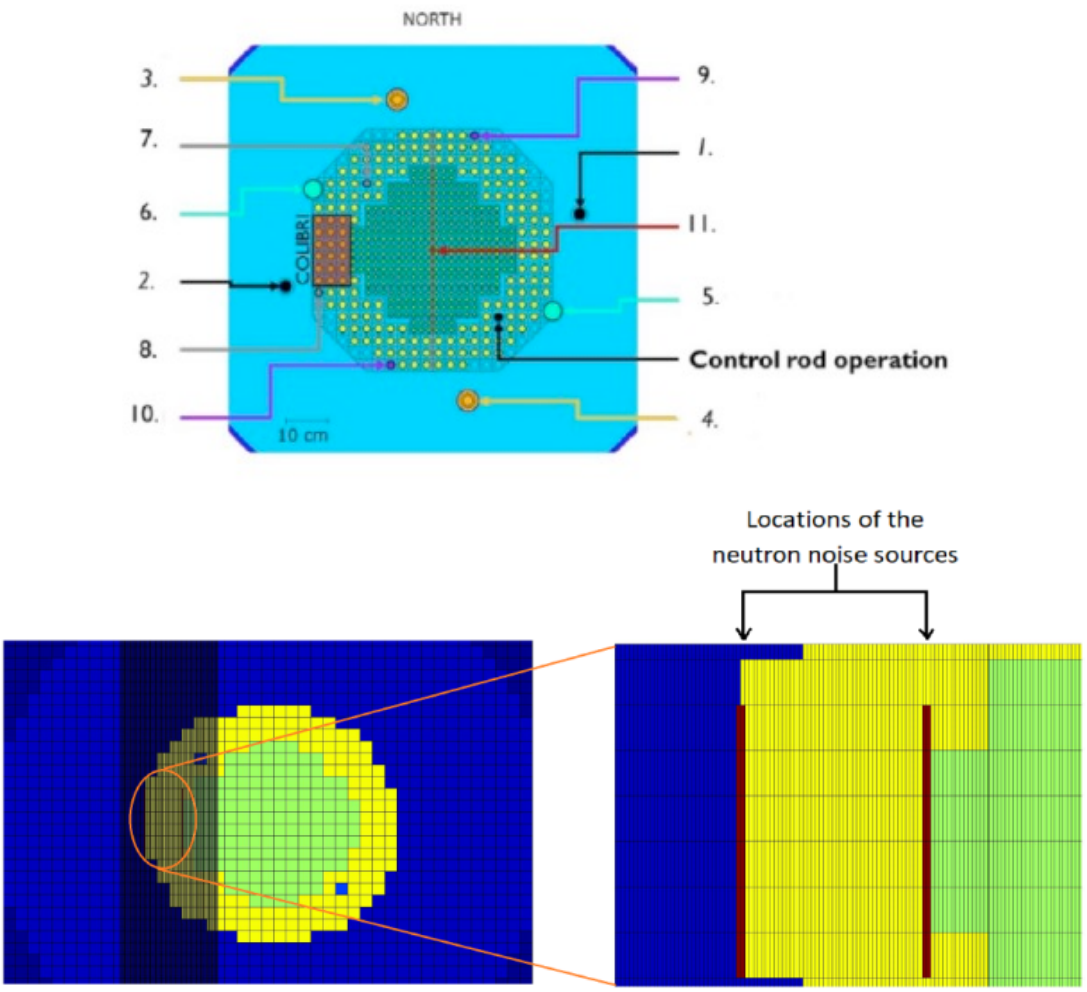


Figure 49: Schematic of the CROCUS reactor for the COLIBRI experiments (top, courtesy of EPFL) and the *CORE SIM*+ model (bottom).

The experimental Cross-Power Spectral Density (CPSD) for a pair of detectors i and j, is estimated from the signals of the 2 detectors according to the Welch's method, see [55]. The predicted CPSD is estimated from the static neutron flux ϕ_0 and the neutron noise $\delta\phi$ calculated with *CORE SIM*+ at the locations of the detectors, i.e.

$$CPSD_{i,j} = \left(\frac{\delta\phi}{\phi_0}\right)_i \left(\frac{\delta\phi}{\phi_0}\right)_j^* \qquad (57)$$

In the equation above, the superscript $*$ symbolizes complex conjugate quantities.

The comparison between experimental and predicted CPSDs for the test No. 12 of the first COLIBRI campaign is given in Figure 50, as an example of this validation work. In the experiment the amplitude of the vibration is +/- 2 mm and the frequency is 1 Hz. The numbering and location of the detectors are shown in Figure 49. The CPSDs are evaluated between each of the detectors and detector 5 and are normalized using the CPSD of the pair of detectors 6 and 5. For the pairs of detectors 10-5, 9-5 and 5-5, the calculated points fall within the error bars associated with the experimental points. The *CORE SIM*+ calculation also reproduced the in-phase behaviour of the detectors, although the differences can reach about 20 degrees.

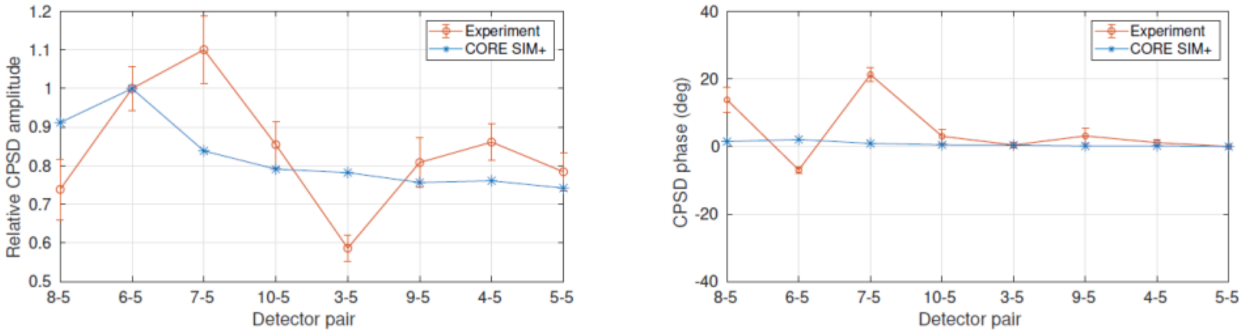


Figure 50: Experiment 12 in the first COLIBRI campaign: comparison between *CORE SIM+* results and experimental data; relative CPSD amplitude (left) and CPSD phase (right).

3.5.4 Simulation of neutron noise experiments in AKR-2

Neutron noise experiments were carried out in the research reactor AKR-2 (TUD), within the CORTEX project, using a rotating absorber or a vibrating absorber, see [8]. The tool *CORE SIM+* was used to simulate one experiment for each type. The schematic of the AKR-2 together with the arrangement of the detectors and the spatial discretization of the reactor used for the *CORE SIM+* calculations, are shown in Figure 51.

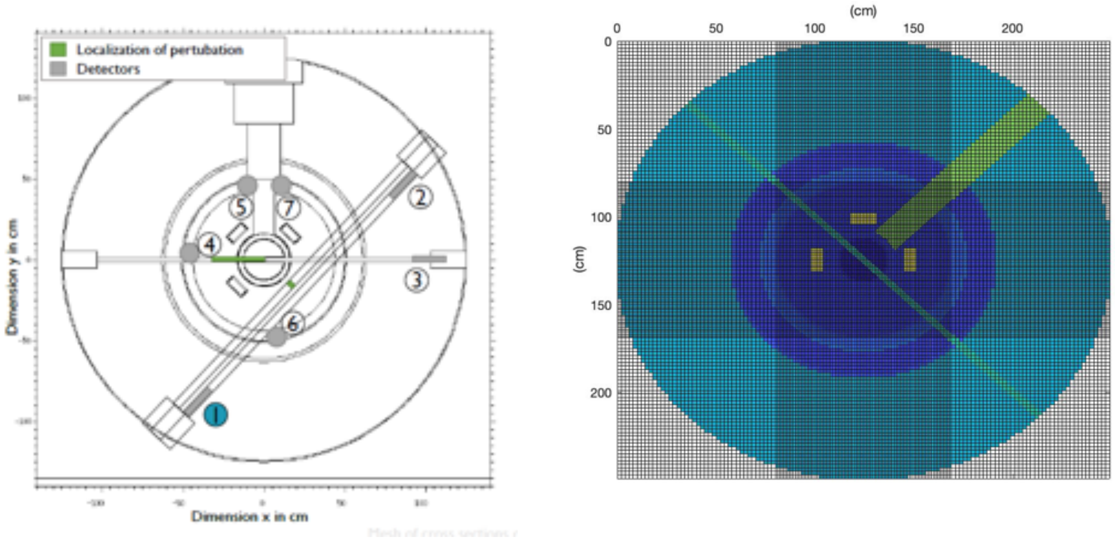


Figure 51: Schematic of the AKR-2 reactor including the detectors 1 to 7 (left, courtesy of TUD) and the CORE SIM model (right).

For the case of a rotating absorber, the experiment No. 7 was selected. A neutron absorber is introduced in the core and is rotated so that an absorber of variable strength is reproduced. The rotational frequency is equal to 1 Hz. After the analysis of the experiment, only detectors 1, 2 and 4 are considered reliable.

For the case of a vibrating absorber, the experiment No. 22 was chosen. A neutron absorber is introduced in the core and is moved in order to mimic a vibrating perturbation. The movement is arranged in a manner that the absorber is moved in one direction for 1.0 second, kept still for 0.1 s, moved back for 1.0 s, kept still 0.1 s, and so on. Only the detectors 1, 2, 3 and 4 are considered reliable.

Experiments and predictions are compared in terms of CPSDs, similarly to the approach used in section 3.5.3. The results are summarized in Figure 52 for the rotating absorber experiment No. 7 and in Figure 53 for the vibrating absorber experiment No. 22. The CPSDs are evaluated between each of the detectors and detector 1 and are normalized with respect to the CPSD of the pair of detectors 2 and 1. Although the experimental relative CPSD amplitudes vary with respect to the detectors and thus the distance from the neutron noise source, the *CORE SIM+* calculations provide no differences between the various detectors. This outcome might indicate issues when using diffusion theory to model the AKR-2 reactor which is characterized by a small size, a complex geometry, and strong heterogeneities in the material composition.

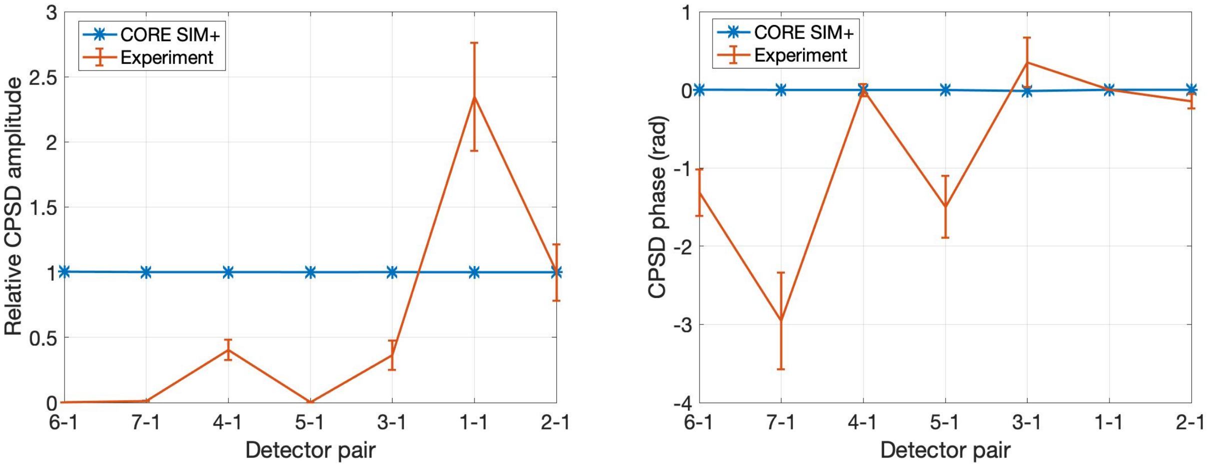


Figure 52: Comparison between *CORE SIM+* and experimental data for experiment No. 7 (rotating absorber); reliable detectors are 1, 2 and 4.

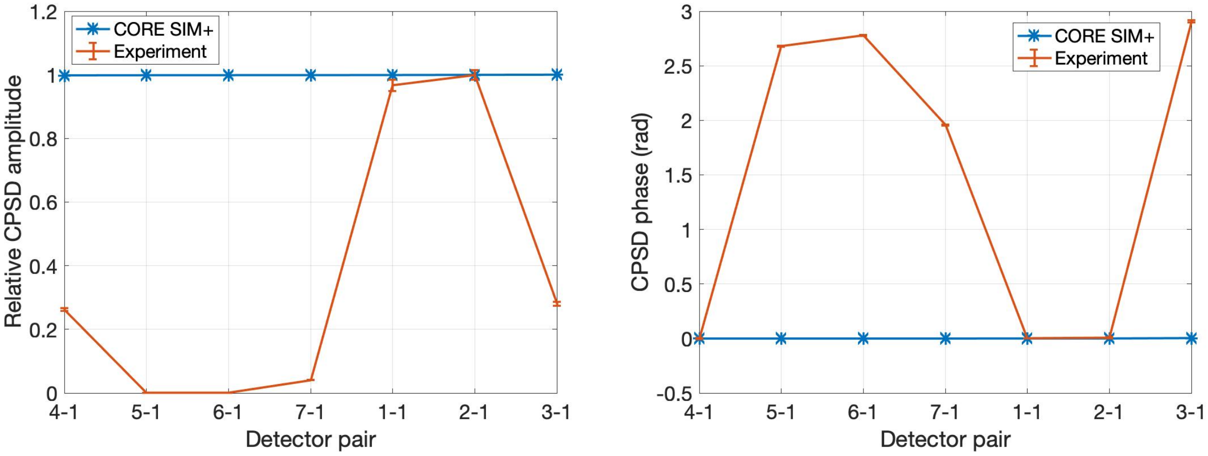


Figure 53: Comparison between *CORE SIM+* and experimental data for experiment No. 22 (vibrating absorber); reliable detectors are 1, 2, 3 and 4.

3.6 Simulation of neutron noise induced by vibrations in the core

The vibrations of core components are among the most relevant neutron noise sources in a nuclear reactor. Therefore, two hypothetical scenarios are analysed with *CORE SIM+*, in which a vibration of a fuel assembly and a pendular vibration of the core barrel are respectively prescribed in a PWR. For these simulations, the system configuration is a MOX/UO2 core of a four-loop Westinghouse PWR, see section 3.5.2 and [57].

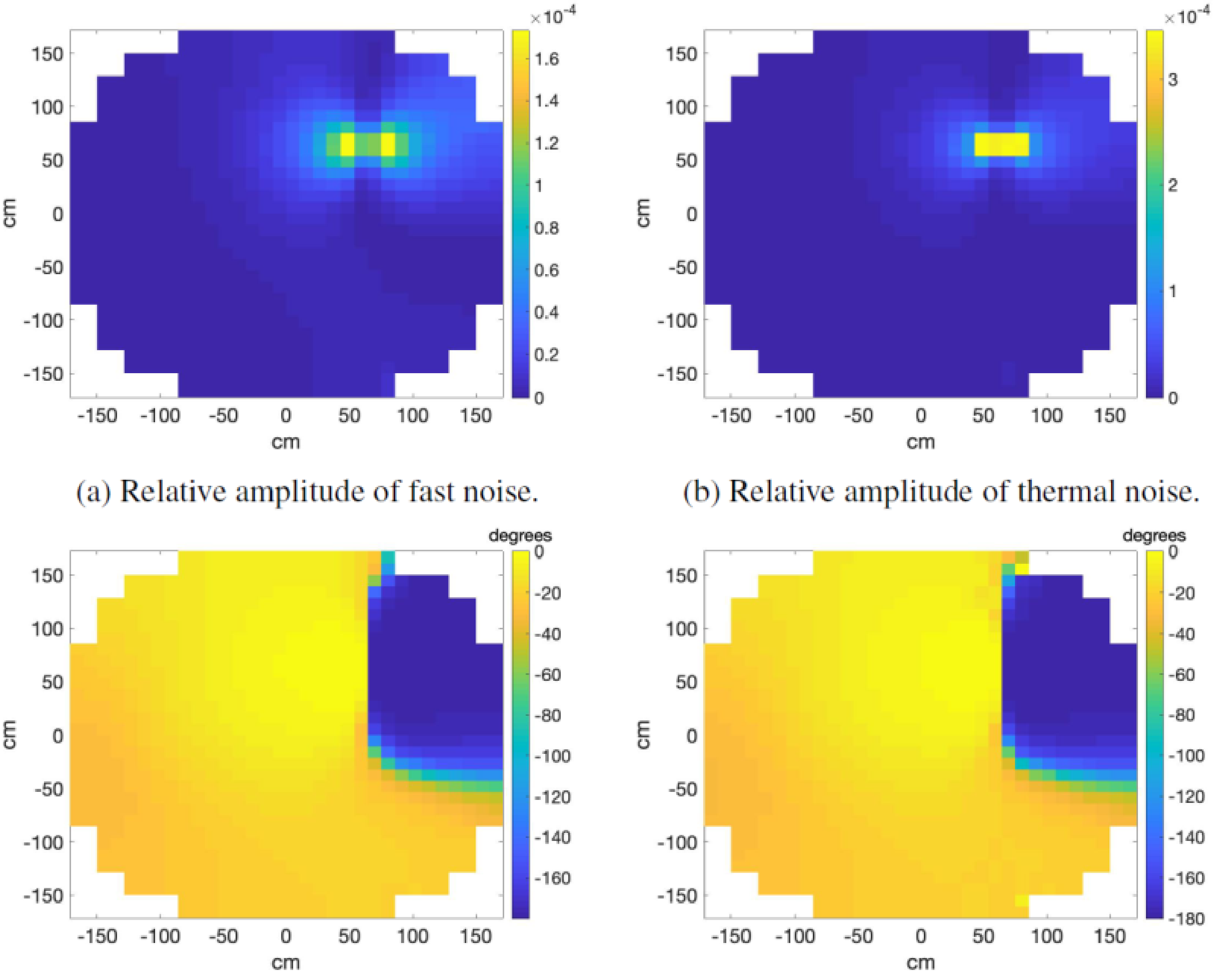
3.6.1 Fuel assembly vibration

Fuel assemblies in a nuclear reactor may vibrate according to different modes. A simply supported on both sides vibration of the second mode is considered hereafter. For this case, the perturbation is modelled with equation (53) in which the axial shape $h(z)$ is given by the following expression:

$$h(z) = A \times \sin(k_v z) \tag{58}$$

The parameter k_v is equal to $2\pi/H$, with H being the height of the reactor.

Figure 54 shows the radial distribution of the computed neutron noise at elevation $z = 234.32\ \mathrm{cm}$ from the bottom of the core, when oscillating the fuel assembly located at $x = y = 74.97\ \mathrm{cm}$, with a frequency of 7 Hz. As expected, the highest relative amplitude is found in the vicinity of the vibrating fuel assembly (Figure 54a-b). The solver also predicts an out-of-phase behaviour of the noise between the left and the right sides of the vibrating fuel assembly (Figure 54c-d). Such a feature is typical of the perturbation (e.g., [59]). Figure 55 gives the distribution of the thermal neutron noise calculated along the axial direction, at the centre of the core. The relative amplitude has a minimum at mid-elevation (Figure 55a), and the noise in the upper and lower parts of the fuel assembly is out of phase (Figure 55b). This behaviour is consistent with the prescribed axial shape of the noise source (Figure 55c-d).



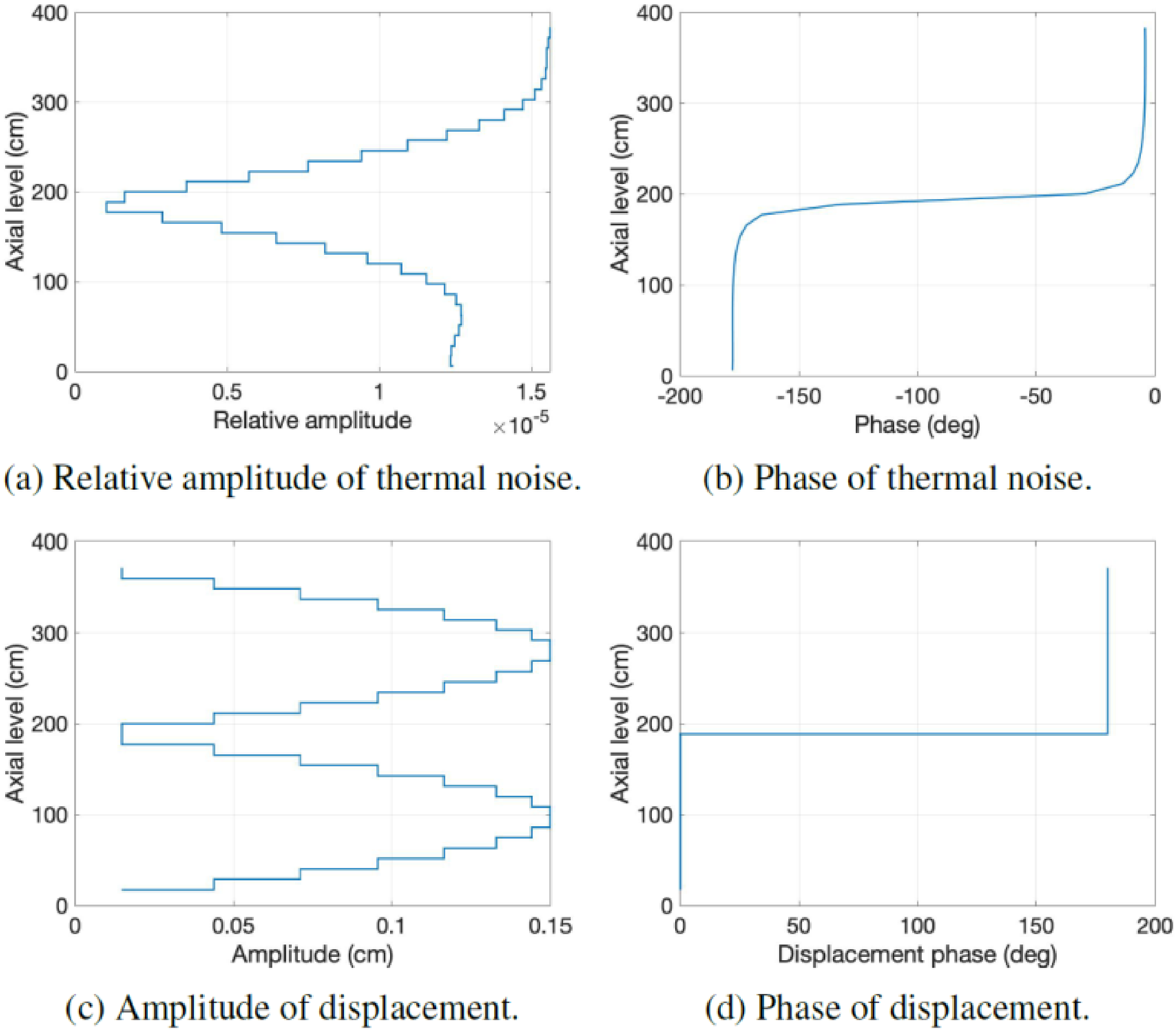
(a) Relative amplitude of fast noise.

(b) Relative amplitude of thermal noise.

(c) Phase of fast noise.

(d) Phase of thermal noise.

Figure 54: Radial distribution of the computed neutron noise induced by the vibration of a fuel assembly vibration, at axial elevation z= 234.32 cm.



(a) Relative amplitude of thermal noise.

(b) Phase of thermal noise.

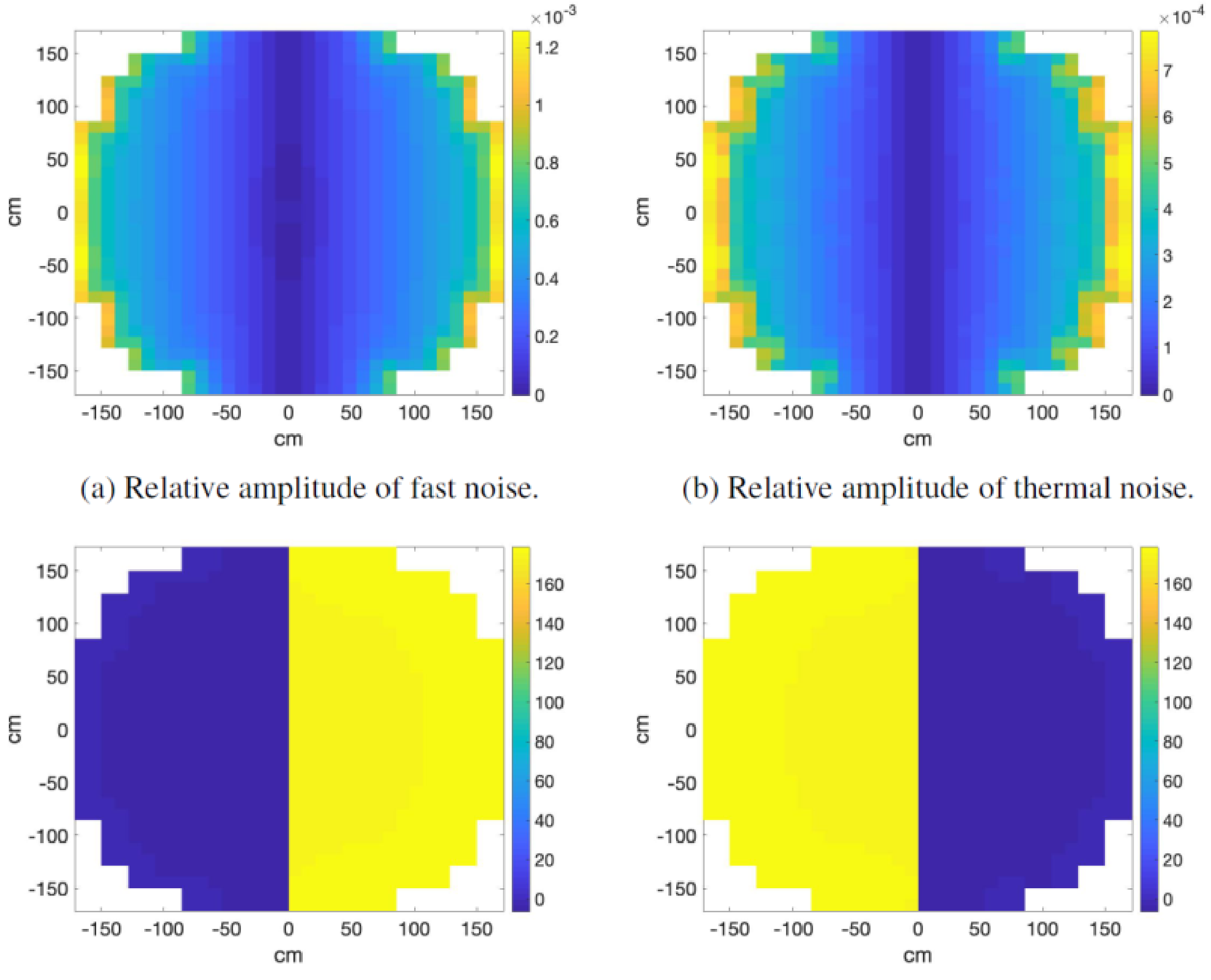
(c) Amplitude of displacement.

(d) Phase of displacement.

Figure 55: Axial thermal neutron noise calculated at the centre of the core (top) and prescribed axial perturbation associated with the vibrating fuel assembly (bottom).

3.6.2 Core barrel pendular vibration

As discussed in section 3.3.3, the pendular vibration of the core barrel can be modelled as a collective movement of the fuel assemblies. In this simulation, the frequency of the vibration is assumed to be 7 Hz. Figure 56 shows the computed neutron noise radial distribution at mid-elevation of the reactor core. As expected, the amplitude is higher around the interface between the fuel region and the reflector, i.e. at the location of the noise source. The noise amplitude is zero in the middle of the reactor, along the line perpendicular to the direction of vibration, because of the symmetry of the problem. The simulation also predicts well the out-of-phase behaviour between the two halves of the core, which is typical of the perturbation.



(a) Relative amplitude of fast noise.

(b) Relative amplitude of thermal noise.

(c) Phase of fast noise.

(d) Phase of thermal noise.

Figure 56: Computed neutron noise induced by core barrel pendular vibration, at mid-elevation of the core.

4 Modelling strategy at PSI

The Paul Scherrer Institute (PSI) aimed to utilize the commercially available advanced simulation tools as basis for the modeling of the various neutron noise sources in PWRs and study their impact on the neutron noise phenomenology. The Studsvik Scandpower (SSP) developed CMS platform codes, i.e. CASMO-5/SIMULATE-3/SIMULATE-3K [60–65], were selected as the codes of choice for numerical noise analysis, given PSI's prior extensive experience with such tools for the analysis of the Swiss nuclear reactors since the late 90s.

4.1 Introduction

The PSI methodology for modelling fuel assembly vibrations is based on the time-domain nodal code *SIMULATE-3K*. In this approach, fuel assembly vibration is assumed to be a stationary process. The noise source is modeled in terms of fluctuations of two group assembly-wise homogenized macroscopic cross sections. This modeling approach is based on the principle that a vibrating fuel assembly leads to time-dependent modification of the water-gap thickness surrounding the fuel assembly in the direction of motion, which is reflected in the perturbed cross sections. Nodal perturbed cross sections are obtained with *CASMO-5* via the so-called 'delta-gap model'. The nuclear data obtained with *CASMO-5*, including multi-group cross-sections, discontinuity factors and kinetics data is post-processed by the CMS-LINK5 code into a readable binary-formatted library. The processed nodal perturbed cross-sections are fed to the downstream codes *SIMULATE-3* and *SIMULATE-3K* for nodal full core calculations. *SIMULATE-3* calculates the static flux and stores the state points corresponding to the operating conditions of interest in 'restart files' for the transient code *SIMULATE-3K*. Lastly, the transient nodal code *SIMULATE-3K* uses the 'assembly vibration model' to imitate fuel assembly vibrations in a dynamic manner, with the help of a set of PSI supporting *MATLAB* Scripts for Input Deck preparation (PSI-SMSID), that has been developed for automatized generation of the time-dependent water gap widths between the vibrating fuel assemblies for a variety of vibrational patterns. The text files generated via the *MATLAB* scripts as support to *SIMULATE-3K* are key to modeling complex fuel assembly vibrations in a time-dependent manner, which are close to the real noise scenarios. A schematic representation of the PSI neutron noise methodology is presented in Figure 57. A detailed description of the modelling steps involving the various simulations codes, the delta-gap model, the assembly vibration model and the supplementary *MATLAB* procedures is given in the following sub-sections.

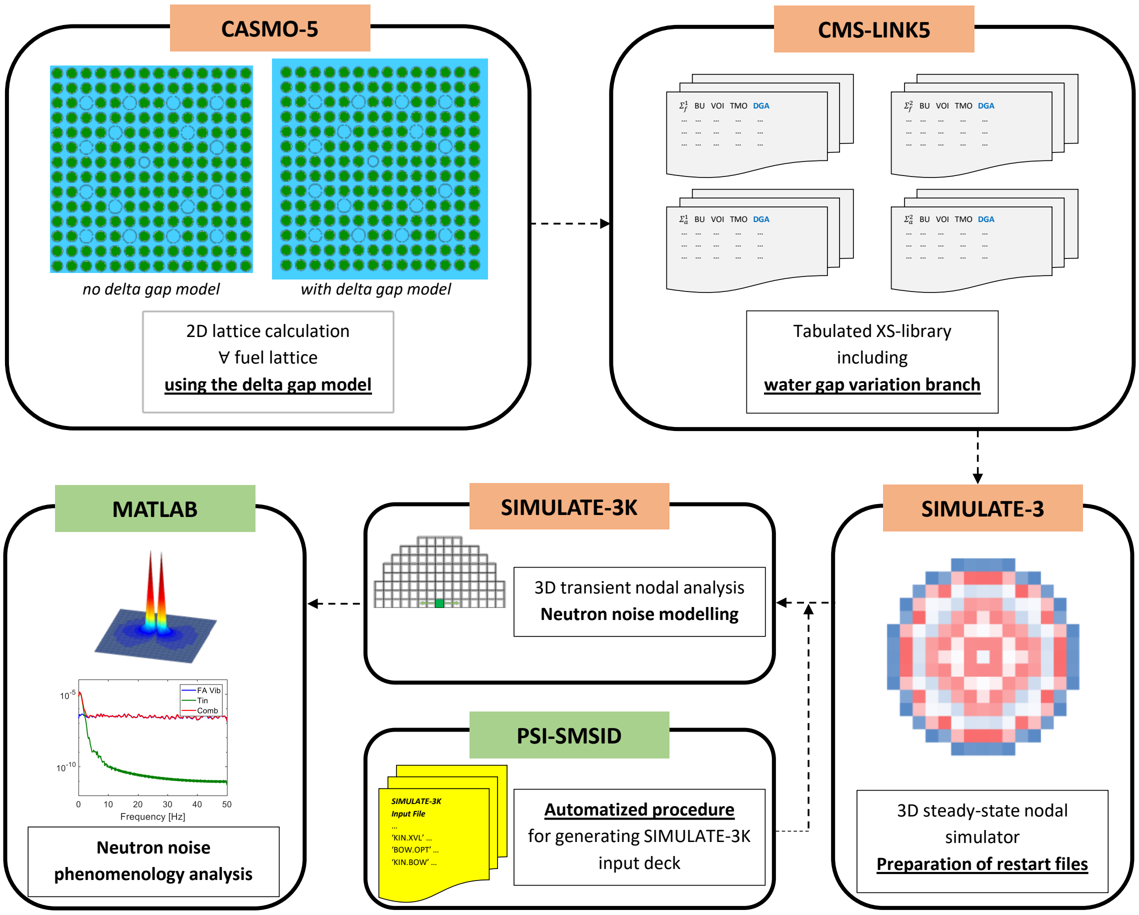


Figure 57: Schematic representation of the PSI neutron noise modelling methodology [66].

4.2 Simulation codes and models

4.2.1 CASMO-5

CASMO-5 is a multi-group two-dimensional lattice code for modelling light water reactor fuel, where the transport solution is based upon the Method of Characteristics. Typically, the code contains an automated case matrix capability for generating nuclear data suitable for downstream the 3D nodal codes. Using the microscopic cross-section data library ENDF/B-VII.1, it generates the two-group homogenized nuclear parameters for each 2D fuel segment whose geometrical and material composition are explicitly defined by the user at various state points (e.g. moderator temperature, boron concentration, control rod positions, etc.).

In the first step of modelling the neutron noise source, *CASMO-5* is employed to generate the two-group homogenized macroscopic cross sections taking into account the vibrating fuel assemblies via the varying water-gap widths using the 'delta-gap model'. To that effect, an additional delta gap branch calculation is performed on top of the default standard case matrix.

4.2.1.1 Delta gap model

The delta gap model assumes that the lateral vibration of a fuel assembly can be represented as modification of the water-gaps surrounding the moving fuel assembly, instead of modelling physically oscillating fuel pins, and by extension, fuel assemblies. The latter approach is referred to as the fuel displacement model. Although the fuel displacement model seems like a more obvious modelling choice, a detailed study of the validity of the delta gap model assumption was performed [67]. It is found that the nuclear data exhibit small discrepancies up to 0.29 % and 0.21 %, respectively, compared to the realistic approach in the fuel displacement model. Note that only the delta gap model is compatible with *SIMULATE-3K*. Therefore, the delta gap model was adopted for generation of perturbed cross-sections for modelling of vibrating fuel assemblies.

To elaborate the delta gap model, when a fuel assembly moves δ mm in the positive x-direction, the surrounding water-gap increases by δ mm in the negative x-direction and decreases by δ mm in the direction of fuel assembly motion. The perturbed cross-sections resulting from the changes in water gap thicknesses are used to represent the vibrating fuel assemblies. However, the execution of the incremented water gap in the lattice calculation is dependent on the lattice symmetry, as specified by the user. In principle, the delta-gap model is capable of introducing water-gap widths on any of the four faces of the assembly, i.e. North, South, East and West. However, only the nuclear data generated with the quarter and octant lattice symmetries are compatible with the downstream nodal codes. For a user-defined water width increase δ on one lattice side, the *CASMO-5* delta gap model automatically imposes the same water width increase on the symmetric side in case of quarter symmetry, and on all the four sides in case of octant symmetry. This means that the lattice calculation in quarter or octant symmetry automatically assumes larger water width increase than the user request. In order to compensate for the effect, the cross-sections are adjusted during post-processing of the nuclear data by CMS-LINK5.

It is to note that *CASMO-5* does not perform an added branch calculation for the generation of cross-sections corresponding to negative delta-gap widths for PWR fuel assemblies. The cross-sections corresponding to the negative delta-gap widths are obtained in the latter steps at nodal level by means of extrapolation.

4.2.2 CMS-LINK5

The next step of the methodology is to prepare the cross-section data library to be called for the 3D nodal *SIMULATE-3* and *SIMULATE-3K* calculations. CMS-LINK5 code collects all the nuclear data generated by the lattice and depletion calculations by *CASMO-5* for every fuel segment, and post-processes them into a binary formatted library for further use by the nodal solvers.

4.2.3 SIMULATE-3

SIMULATE-3 is a 3D full core solver with coupled neutronic and thermal-hydraulic capabilities for estimating the 3D nodal power at every fuel assembly of a PWR or BWR core. Every fuel assembly is equally discretized into Z axial nodes, and every node is further split into 2x2 planar sub-nodes for a spatially precise solution. The cross-sections from *CASMO-5* are homogenized within each such sub-node. The two-group 3D diffusion equation is solved for each sub-node using the two-group homogenized cross-section, generated by *CASMO-5*, interpolated at the local operating conditions. The spatial scalar flux is estimated using a fourth order polynomial with quadratic transverse leakage. The intra assembly exposure is represented by a quadratic polynomial in two directions. In addition, the thermal-hydraulics model of *SIMULATE-3* solves the total mixture mass, energy, and momentum equations for each fuel assembly/bundle, by estimating the void fraction using a drift flux model. A core-follow calculation, evaluating the reactor state over one operational cycle is performed, and *SIMULATE-3* stores the state points in ‘restart files’ for the transient nodal code *SIMULATE-3K*.

4.2.4 SIMULATE-3K

Lastly, *SIMULATE-3K*, the three-dimensional two-group transient nodal code, reads the operating conditions of the core at the analyzed core state via the respective restart file, and then initiates the transient full core calculation to calculate three-dimensional time-dependent two-group fluxes. The development of the improved modules of *SIMULATE-3K* and the set of supplementary *MATLAB* scripts, PSI-SMSID, to model fuel assembly vibrations is described in this section.

During the initial investigations, several limitations were identified in the *SIMULATE-3K* code by PSI for successful modelling of realistic noise scenario of vibrating fuel assemblies [67].

Some of the limitations include:

- lack of conservation of the gap width on either side of the moving fuel assembly,
- inability to impose realistic physical displacements of fuel assemblies,
- lack of flexibility to impose different vibrational characteristics and patterns to simultaneously vibrating fuel assemblies, and
- the inability to vibrate clusters of fuel assemblies with different fuel design identifiers.

Therefore, a close collaboration with the *SIMULATE-3*K code developers was established for performing necessary modifications and refinements that were aligned with the need for realistic modelling of fuel assemblies vibrations, allowing the user to dynamically change the water gap widths of any node of the core at any time step. The improved *SIMULATE-3K* version (beta version) employs the assembly vibration model to imitate fuel assembly vibrations in a time-dependent manner. The vibrations of fuel assemblies in x- or/and y-direction are simulated by dynamically modifying the water-gap widths between any two fuel assemblies.

4.2.4.1 Assembly vibration model

The implementation of the assembly vibration model is described hereafter. In this modelling scheme, vibration of a central fuel assembly affecting the water gap widths between the two neighboring fuel assemblies involves eight sub-nodes. To illustrate the model, a schematic diagram of the introduction of perturbed cross sections in the assembly vibration model in *SIMULATE-3K* is shown in Figure 58. A central fuel assembly, FA_i, is displaced to the left direction, and the vibration is represented by introducing pre-calculated perturbed *CASMO*-5 cross sections corresponding to the modified water-gap width of $\delta/2$ in these eight affected sub-nodes, as marked in the Figure 58.

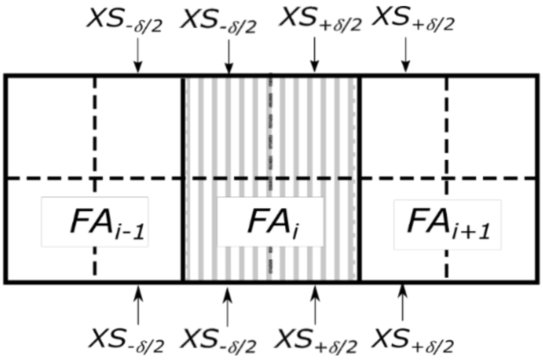


Figure 58 Modification of cross sections when central FA, FA_i (striped) moves to the left direction. The modified cross sections, $XS_{-\delta/2}$ and $XS_{+\delta/2}$ are introduced in the eight labelled sub-nodes.

Perturbed cross-sections, $XS_{+\delta/2}$, corresponding to an increased water-gap width of δ/2 are introduced in the four sub-nodes on the right, two each belonging to the oscillating assembly FA_i and the first neighbor, FA_{i+1}. Likewise, perturbed cross-sections, $XS_{-\delta/2}$, corresponding to a decreased water-gap width of δ/2 are introduced in the four sub-nodes on the left belonging to FA_i and FA_{i-1} at a certain time step. The introduction of the cross-sections in such a way is essential to successfully model the displacement of FA_i towards FA_{i-1}, while ensuring the fixed computational mesh in the core is conserved. The procedure is applied at every time step by appropriately modifying the delta gap widths between the involved fuel assemblies.

As mentioned earlier, the perturbed cross-sections corresponding to the negative delta-gap widths that are missing in the pre-calculated *CASMO-5* nuclear data, are obtained with *SIMULATE-3K* by extrapolating the values to delta gap width of δ. The maximum displacement amplitude of the fuel assemblies that can be modelled is limited by the distance between the neighboring fuel assemblies.

4.2.5 MATLAB support

A set of supplementary *MATLAB* scripts (PSI-SMSID) were developed by PSI to facilitate the preparation of *SIMULATE-3K* input for simulating flexible and realistic fuel assembly vibrations. The scripts allow calculation of the time-dependent water gap widths between the chosen vibrating fuel assemblies and their adjacent assemblies; and generate support files to be included in the *SIMULATE-3K* calculation. It allows the user to input dynamic water-gap widths to *SIMULATE-3K*, corresponding to different vibrational patterns in terms of choice of vibrating assemblies, and vibrational characteristics such as amplitude, phase and frequency, etc., and patterns such synchronized or unsynchronized vibrations.

The beta version of *SIMULATE-3K* includes three new input cards, i.e. *KIN.XVL*, *KIN.YVL*, and *KIN.ZVL*. The two first input cards, describe the dynamic modification of the delta gap sizes in the x- and y-directions, respectively. The user has the flexibility to impose any type of oscillation mode (e.g. random, stepwise, sinus patterns, etc.) and vibrational characteristics (i.e. amplitude and frequency) on both water gap width sides of the vibrating assembly by preserving its geometrical size. The third input card, KIN.ZVL enables the user to impose pre-defined functions representative of the vibration modes of the fuel assemblies. This is done by creating an input vector of the axial shape by assigning factored coefficients between zero and one to each axial node. In other words, the fuel assembly is modeled to vibrate in a certain axial pattern by displacing each of the axial nodes by a width, δ, that is calculated using the coefficients and the water-gap widths at every time-step. With the external support file generated with the in-house *MATLAB* script containing the time-wise delta-gaps, S3K performs transient full core calculations to obtain three-dimensional time-dependent two-group fluxes. This modelling scheme enables S3K to replicate time-dependent realistic movements of the fuel assemblies, and faithfully calculate the associated neutron noise in the core. It is important to note that the realistic modelling of vibrating fuel assemblies is not possible with the standalone *SIMULATE-3K* code. The addition of the *MATLAB* support scripts to the modelling approach enables simulation of realistic modes of vibrating fuel assemblies. The various steps involved in the preparation of the include files for the input deck of *SIMULATE-3K* are illustrated in Figure 59, and examples to demonstrate generation of time-dependent delta gap widths for two scenarios are shown in detail in Annex 9.2.

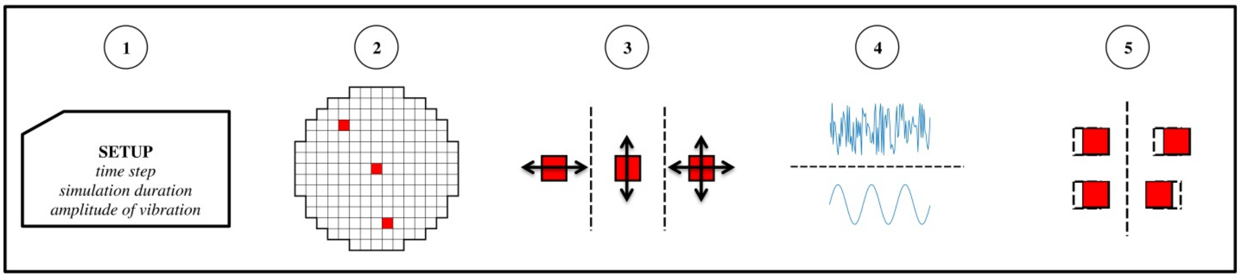


Figure 59: Schematic illustration of the steps of generation of delta-gap widths for simulating fuel assembly vibration using PSI methodology.

The generation of simulated data containing nodal neutron fluxes and detector signals is essential for further performing noise analysis. As the signals from the neutron detector locations in the core are limited, the nodal three-dimensional time-dependent neutron fluxes are also used to obtain the fast and thermal neutron noise amplitude and phase. The detection and localization of the vibrating fuel assemblies is based on the spectral analysis of the neutron detector signals. The PSI methodology employs signal processing techniques, which involves a standard time- to frequency-domain analysis based on the Fourier transform of the autocorrelation function. The procedure is implemented using *MATLAB* scripts. The detector signals are unfolded to reveal properties of the neutron noise source. The analysis includes radial and axial noise phenomenology based on the assessment of the auto power spectral densities (APSD) of the detector signals, and the phase and coherence between the azimuthal and axial neutron detectors. In the noise scenario of a fuel assembly vibration, a resonance peak is observed in the APSD at the excitation frequency, and an out-of-phase response between the signals belonging to the fuel assemblies surrounding the vibrating central fuel assembly [68]. The presence of a phase difference of 180 degrees between the signals is in principle a characteristic feature, which can be used for localizing the source of oscillation in a system, in this case, a fuel assembly vibration. A deviation from the out-of-phase behavior is typically seen when fuel assemblies are vibrating in peripheral locations, where large reactivity effects are present.

4.3 Simulation description and results

An example illustrating the implementation of the above-described methodology is shown here. A typical four-loop Westinghouse 15x15 mixed core PWR of the OECD/NEA transient benchmark is used for the study [57]. A radial layout of the core, along with an axially discretized single fuel assembly containing 32 nodes, are shown in Figure 60. The neutron detectors are located at few discrete locations in the reactor as shown in the figure. A set of eight radial in-core neutron detectors are modeled at six axial locations each, and a set of four ex-core neutron detectors are modeled at two axial locations each.

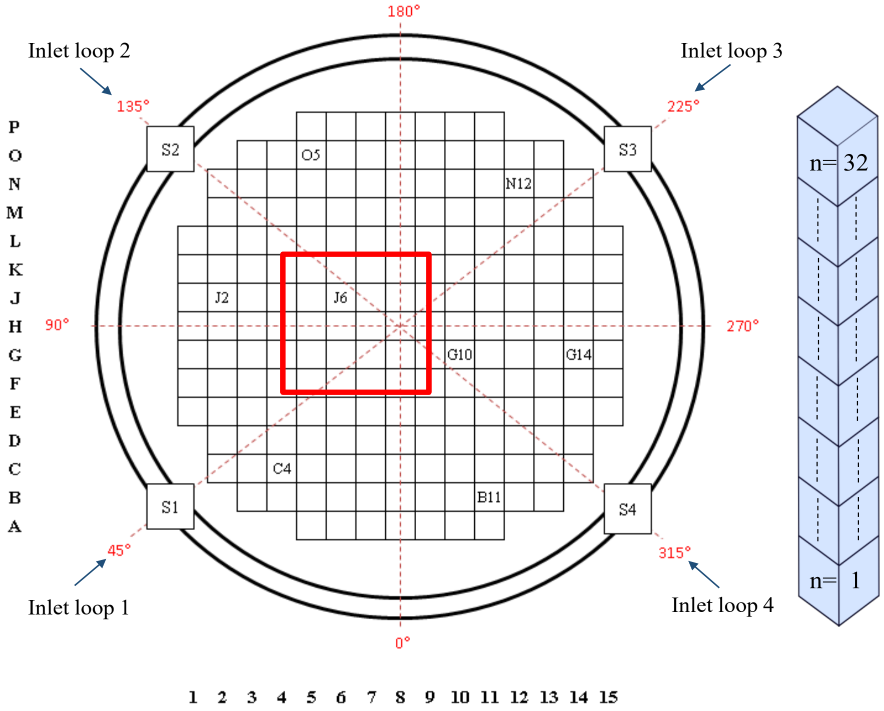


Figure 60 Left: Radial layout of the OECD-PWR core model. The locations of the in-core and ex-core neutron detectors, and the coolant loops are labelled. Right: Axial cross section of a fuel assembly discretized into 32 nodes.

Out of the various possible modes of fuel assembly vibrations, the most significant ones are the cantilevered mode at 0.6 - 2.0 Hz, where the fuel assembly is clamped-free at the top but fixed at the bottom; and the C-shaped and the S-shaped modes at 0.8 - 4.0 Hz and 5.0 - 10.0 Hz, respectively, where the fuel assembly is fixed at both the top and the bottom. The three vibrational modes are illustrated in Figure 61. Simulations are performed for noise scenarios based on the three modes of vibrations and their combination with thermal hydraulic perturbations of inlet coolant temperature and coolant flow, and the parameters of the noise scenarios are listed in Table 12.

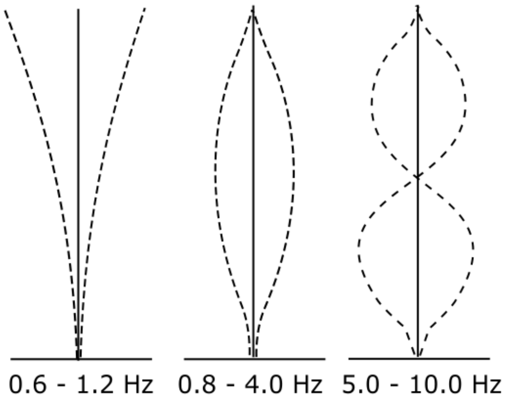


Figure 61: Vibrational modes of fuel assemblies and their range of vibrations.

Table 12 Conditions of the noise scenarios

Transient scenarios				**Vibrational frequency**	**Maximum radial displacement**	**TH Fluctuations**
1.	Fuel assembly vibration	**a.**	Cantilevered mode	1.2 Hz	1.0 mm	-
		b.	C-shaped mode	1.2 Hz	1.0 mm	-
		c.	S-shaped mode	5.0 Hz	1.0 mm	-
2.	Fuel assembly vibration in Cantilevered mode +			1.2 Hz	1.0 mm	-
	Coolant inlet temperature fluctuations +			-		286.67°C ± 1°C
	Coolant flow fluctuations			-		100%± 1%

The simulations are performed for a duration of 35s at a time step of 0.01s. A 5x5 central cluster of fuel assemblies, as marked in red in the Figure 60, are modelled as synchronized pure sinusoidal vibrations in three vibrational modes along the x-direction with a maximum displacement amplitude of 0.1 cm. In principle, the model is also capable of simulating several other possible noise scenarios such as synchronized and unsynchronized random vibrations or noisy sinusoidal vibrations of fuel assemblies, but they have been omitted from the analysis here. For the thermal-hydraulic noise source, fluctuations are introduced synchronously in all the four coolant loops of the reactor.

The variation of the neutron fluxes in the fast and thermal groups, obtained with SIMULATE-3K at every nodal point in the core, is used to derive the induced neutron noise in terms of the statistical quantity, the coefficient of variation (CV). CV, expressed as percentage, is defined as the ratio of standard deviation σ to the mean value $\bar{\varphi}$ of the neutron flux φ in the energy group G, obtained at any given node located at i, j, z in the core. It is expressed as,

$$CV_G = 100 \cdot \frac{\sigma_{\varphi_G^{i,j,z}}}{\bar{\varphi}_G^{i,j,z}}. \tag{59}$$

The radial and axial distributions of the induced fast and thermal neutron noise are shown in Figure 62, Figure 63 and Figure 64 for the vibration in cantilevered mode, C-shaped mode and S-shaped mode, respectively. The phases of the induced neutron noise for the three cases, i.e. 1a, 1b and 1c, are shown in Figure 65. The noise levels simulated by the in-core neutron detectors and the results from the spectral analysis are shown in Figure 66, respectively, only for the first scenario of cantilevered mode vibrations, i.e., 1a. The radial and axial noise distributions in case of the combination scenario, '2' are shown in Figure 67.

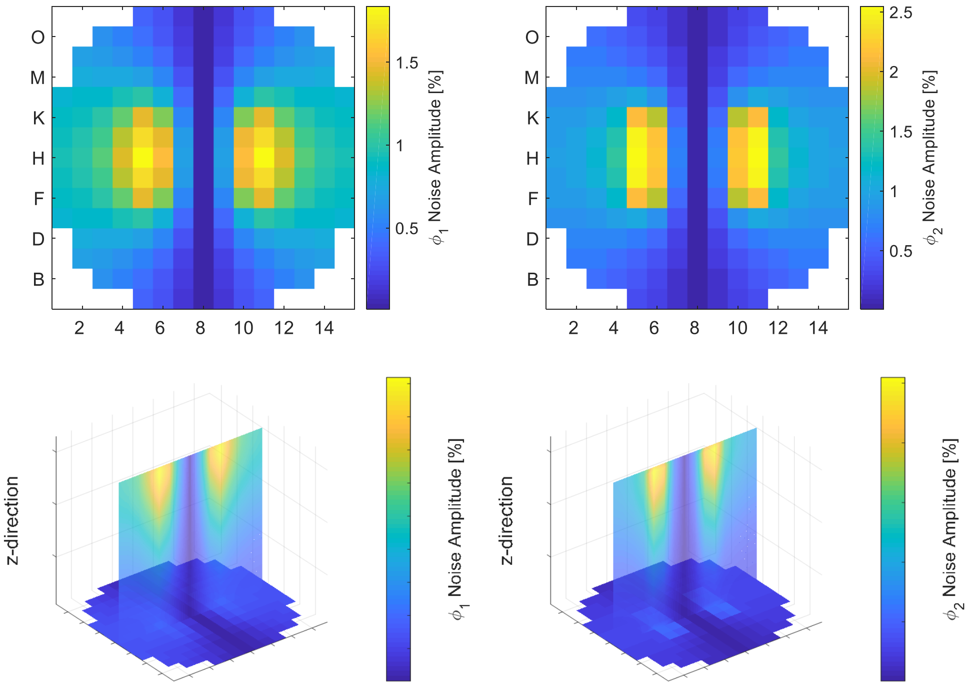


Figure 62: Radial (top) and axial (bottom) distributions of the fast (left) and thermal (right) induced neutron noise due to vibration of the 5x5 central fuel assembly cluster in x-direction in the cantilevered mode at 1.2 Hz. The radial distributions are obtained at the axial node where the noise level is the highest.

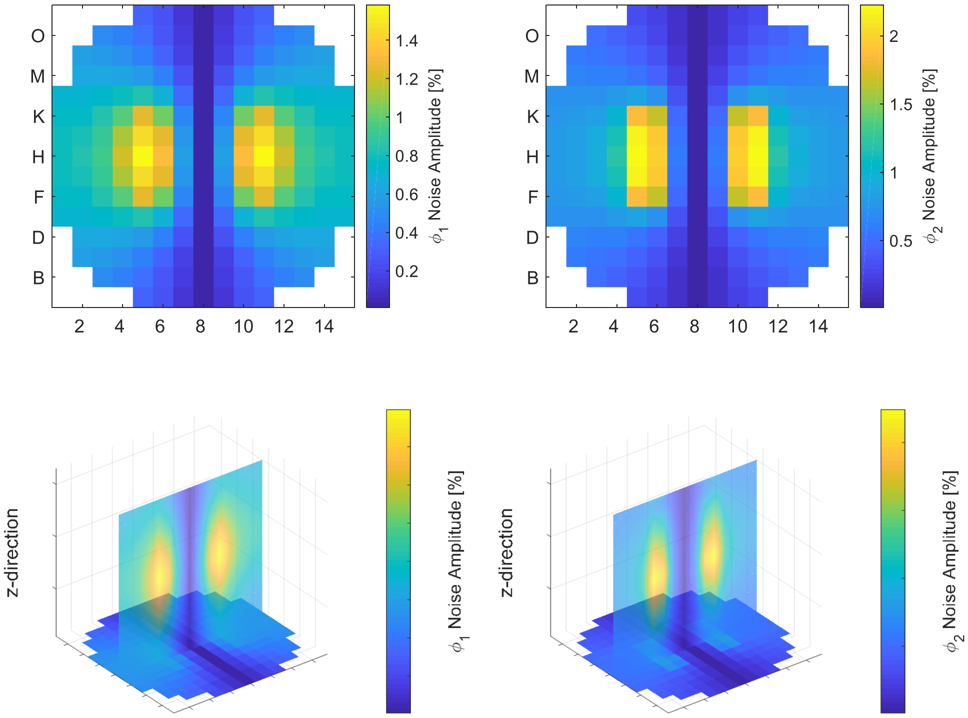


Figure 63: Radial (top) and axial (bottom) distributions of the fast (left) and thermal (right) induced neutron noise due to vibration of the 5x5 central fuel assembly cluster in x-direction in the C-shaped mode at 1.2 Hz. The radial distributions are obtained at the axial node where the noise level is the highest.

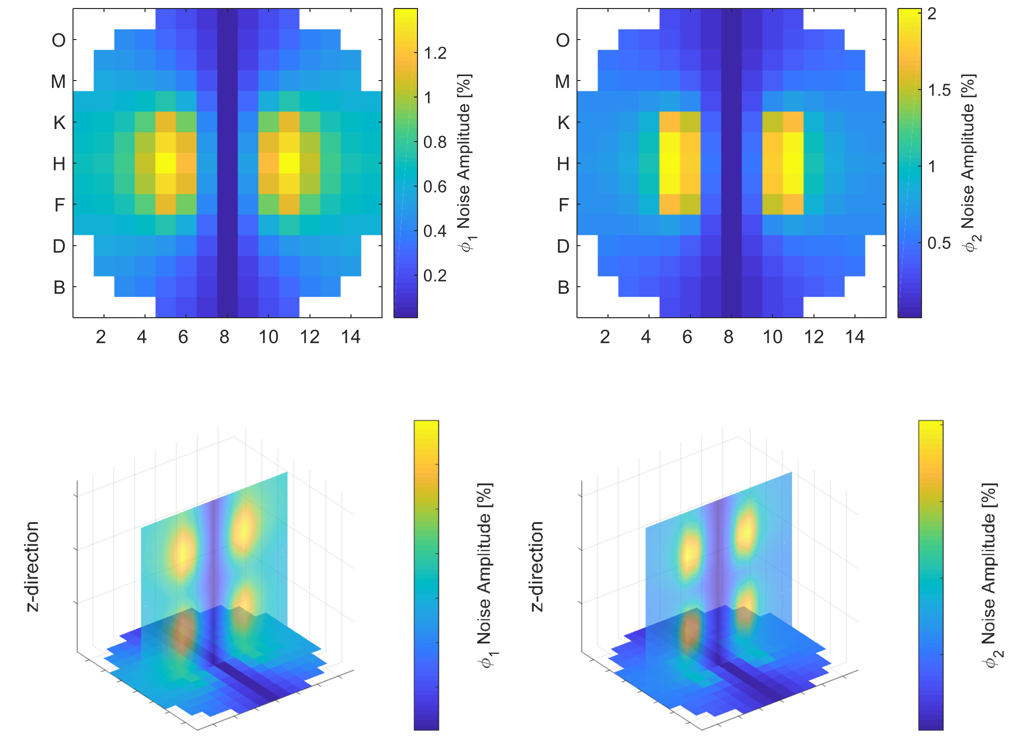


Figure 64: Radial (top) and axial (bottom) distributions of the fast (left) and thermal (right) induced neutron noise due to vibration of the 5x5 central fuel assembly cluster in x-direction in the S-shaped mode at 5 Hz. The radial distributions are obtained at the axial node where the noise level is the highest.

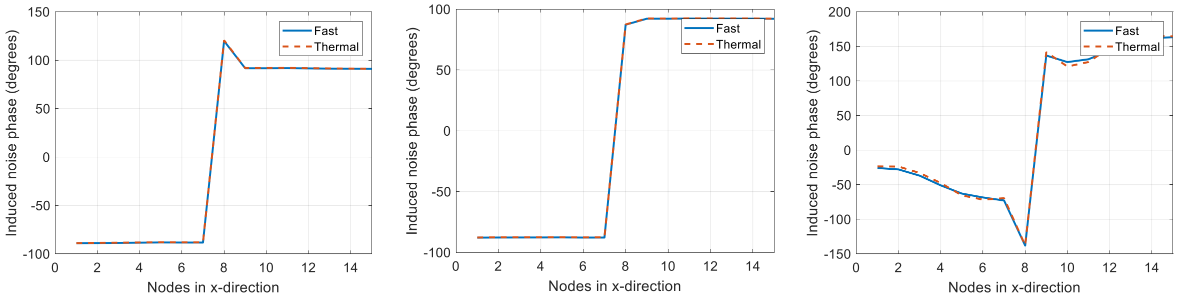


Figure 65: Induced neutron noise phase distribution along the nodes in the x-direction due to vibration of a 5x5 cluster of fuel assemblies in cantilevered mode (left), C-shaped (middle) and S-shaped (right) mode.

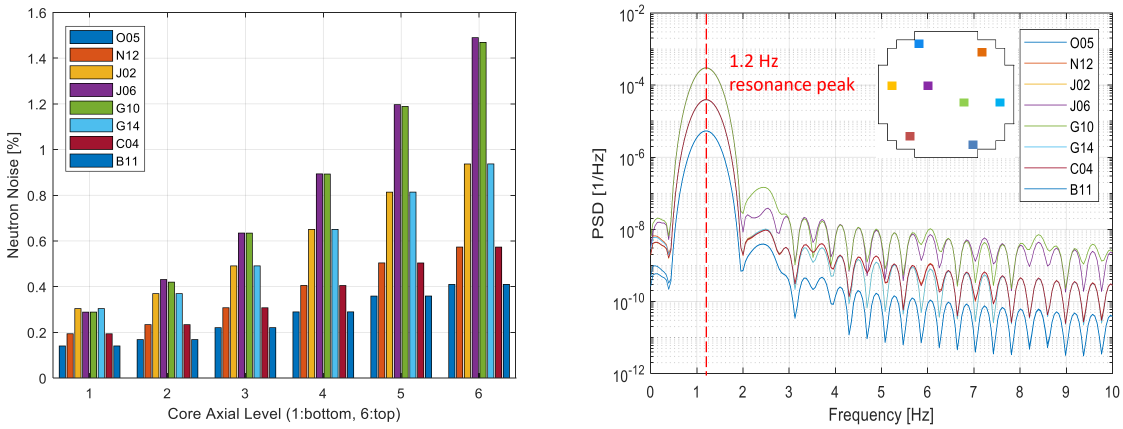


Figure 66 Left: Noise levels in in-core neutron detectors and Right: Power spectral densities obtained with in-core neutron detectors. The dashed red line represents the excitation frequency of the vibrating fuel assemblies, as described in transient scenario '1.a' of Table 12.

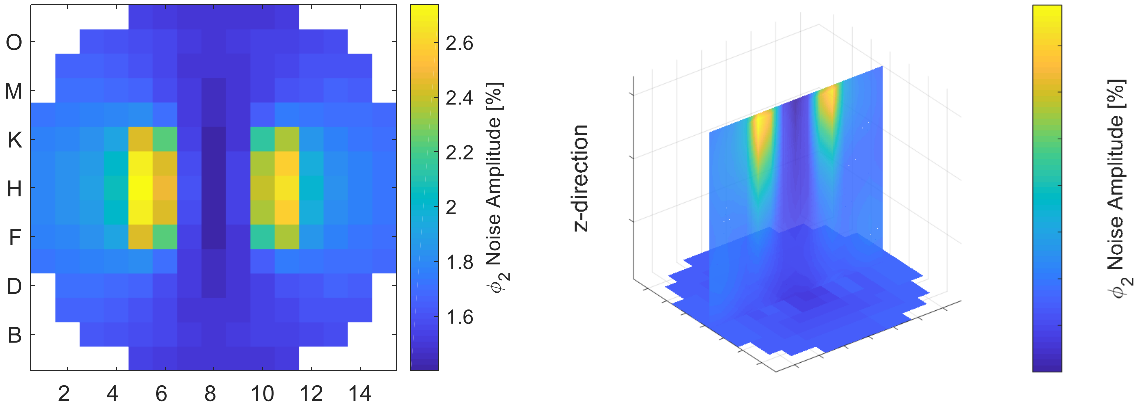


Figure 67 Radial (core-top) and axial induced thermal noise distribution due to a superimposed noise sources, as described in scenario '2' of Table 12.

The main observations are summarized here:

- The noise profiles in Figure 62, Figure 63 and Figure 64 follow the imposed noise sources and reflect the pre-defined axial shape representative of the imposed vibrational modes. In case of the cantilevered mode vibration, higher neutron noise amplitude is obtained at the core-top compared to the core-bottom as the mode is modelled such that the fuel assembly is free to vibrate at the top while supported at the bottom by the core-support plate, which introduces much larger water gap widths at the top. The noise levels observed in the in-core neutron detectors in Figure 66 (left) confirm the behavior. In case of the C-shaped mode, the core-lower and -upper support plates offer resistance to the vibrations of assemblies from the bottom and the top, and introduces wider water-gap widths in the middle, which is reflected in the induced noise. In case of the S-shaped mode, the time-dependent vibration of the fuel assembly is reflected in the noise distribution, as it appears to be split into two symmetric halves axially, representative of the S-shaped noise source. In general, the neutron noise amplitude is higher at the top-half compared to the bottom-half due to the more negative moderator temperature coefficient of reactivity at the top.

- For all the three scenarios of fuel assembly vibrations, the neutron noise distribution is symmetric around the oscillating fuel assemblies. In case of a vibrating cluster of assemblies, the relative movement of the central fuel assembly is zero compared to the rest of the moving cluster. This is understood from the way perturbed cross-sections are introduced at the level of *SIMULATE-3K*. Therefore, the induced noise at the center of the source is almost zero, while the highest noise amplitude is obtained in the adjacent fuel assemblies and the peripheral assemblies of the vibrating cluster, along the direction of motion, as these assemblies are impacted the most from the water-gap modification. The noise decreases significantly a few mean free paths away from the noise source. For a randomly vibrating fuel assembly cluster, the behavior is modified slightly, and the noise amplitude is distributed uniformly in all the directions radially.

- The neutron noise behavior is qualitatively consistent in both energy groups. However, the fast neutron noise is observed to be more diffused in the core due to their relatively large mean free paths. The thermal noise has higher amplitude because of the higher relative impact on the absorption cross sections due to change in the water gap widths.

- As expected, the phase response of the induced neutron noise due to vibration of central cluster is out-of-phase between the two halves of the symmetric core, as seen in Figure 65.

- Spectral analysis of the induced neutron noise in the frequency domain reveals that the excitation frequencies of the vibrating fuel assemblies are observable in the noise spectrum, as shown in Figure 66.

- In the combination scenario with superimposed noise sources of fuel assembly vibrations and thermal-hydraulic fluctuations, the effect of vibrations of fuel assemblies dominate over the latter at the core-top, and therefore, the axial noise distribution resembles the one from independent cantilevered mode vibration. It is found that the detectors closest to the coolant loops have highest noise levels, and as we move towards core-top, the detectors closest to the vibrating cluster witness the highest noise. Spectral analysis of the detectors signals (not included here) reveals that in addition to the effect of thermal-hydraulic fluctuations that are seen in the lower frequency ranges of < 2 Hz, the signature frequency due to vibration of central cluster of fuel assemblies is still observed, albeit shifted slightly towards the lower side of the spectrum. Such studies are detailed in a forthcoming work [68].

- The modelling approach is based on transient nodal codes such as *SIMULATE-3K*, which takes into account thermal-hydraulic feedback, and therefore, inherently handle any non-linear effects. The user is able to simulate different datasets based on various noise scenarios in a flexible manner. The results illustrate the maturity of the PSI methodology to simulate realistic neutron noise sources in heterogeneous systems that are possibly observed in real plant conditions, and to calculate and analyse the amplitude and phase of the induced neutron noise.

5 Modelling strategy at TUD

The work presented in this section describes the efforts to use the time-domain reactor code *DYN3D* to study the effect of mechanical vibrations of the reactor core components inside a 4-loop pre-Konvoi nuclear reactor, on the neutron noise measured by the in-core detectors. The methodology employs a full core modelling strategy, where the mechanical vibrations of all reactor core components are considered at the same time, rather than focussing on a single component. As this simulation was anticipated to be computationally expensive in addition to the simulation of the reactor core, a reduced order modelling technique was applied to a detailed mechanical model developed by GRS. The work aims at gaining a general understanding of the influence of the geometry effect induced by the vibration of the reactor core components on the neutron noise.

5.1 Introduction

DYN3D is a three-dimensional time-domain reactor code, developed by HZDR (Helmholtz-Zentrum Dresden-Rossendorf), to simulate thermal reactor cores [69]. It utilizes a three-dimensional nodal model to solve the two-group or multi-group diffusion equation for neutrons in both Cartesian and hexagonal-z geometry. The modelling of the thermal-hydraulics inside the core is based on a four-equation model representing the conservation of mass, momentum, energy of the mixture and the mass balance of the vapour (in the case of a boiling water reactor). These are solved numerically by the method of characteristics (for the energy conservation) and a so called MIRONOV scheme (for the solution of the coupled mass and momentum balance). The coupled neutronic and thermal-hydraulic models are solved via an internal iteration loop until a convergence of the feedback parameters is reached.

For the simulation of the mechanical vibrations of the reactor core components, a reduced order model was utilized which has been developed in Task 1.1 of the CORTEX project. A precise description of the reduced order model and the detailed model which is based upon, can be found in deliverable D1.2 [70]. The model represents all reactor core components of a pre-Konvoi 4-loop reactor, i.e. the fuel assemblies, the core barrel and the reactor pressure vessel by means of one-dimensional mechanical beams. For the fuel assemblies, a model with four types of assemblies with varying stiffness values was used. These account for the fact that the lateral stiffness of the fuel assemblies inside a reactor core changes under the influence of the neutron flux on each assembly during their time in the reactor core (a detailed description of these effects can be found in D1.2). To save computational time, the detailed model was reduced in dimension by the POD model order reduction technique [71,72]. The model used in this work does not utilize fluidic near field coupling and uses mechanical damping based on mass and stiffness damping (i.e. Rayleigh damping) according to the following equation:

$$D = \alpha M + \beta K,$$

D being the damping matrix, M the mass matrix and K the stiffness matrix of the mechanical system.

The mechanical model is used in an offline phase separate from the *DYN3D* calculation. This approach is valid if it is assumed that the fluidic near-field coupling of the reactor core components is small in comparison to their coupling via the common fixation inside the core. The effect of the coolant flow is modelled by an external stochastic force acting with Gaussian white noise on the core barrel and the reactor pressure vessel (RPV).

5.2 Description of coupling

Originally, *DYN3D 3.2* had no capabilities to model the fluid-induced vibrations of reactor core components. Therefore, it was necessary to use an enhanced version of *DYN3D* for which it is possible to couple mechanical vibrations of reactor core internals to the simulation of the neutron flux, the thermal-hydraulics, as well as the heat transfer. The method applied in this work builds upon the work described in [73]. After an initial steady state calculation with *DYN3D*, a mean fuel assembly distance variable d_{m} is calculated for each node and each time step in *DYN3D*. This is done via the calculation of the mean distance d_{m} of each fuel assembly node to its neighbours in the respective Cartesian coordinate directions at each time step (see Figure 68) according to:

$$d_{\mathrm{m}} = \frac{1}{4}\left(d_{\mathrm{l,x}} + d_{\mathrm{r,x}} + d_{\mathrm{t,y}} + d_{\mathrm{b,y}}\right).$$

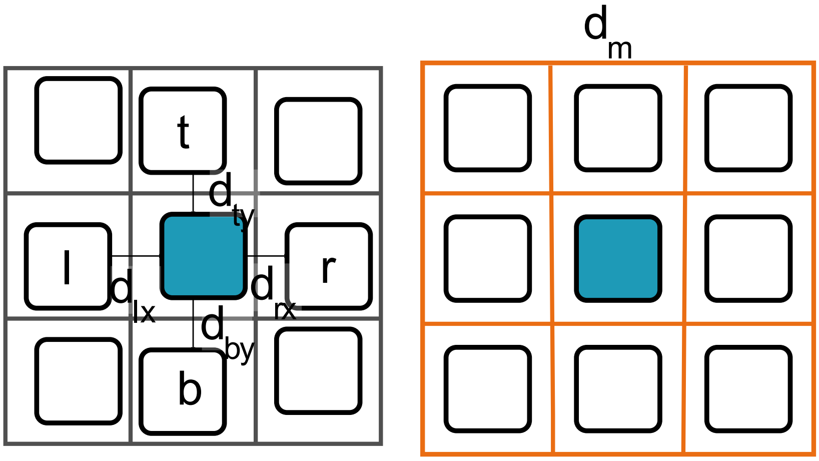


Figure 68: Scheme of a specific fuel assembly and its neighbours and the distances used to calculate the mean distance needed as cross-section parameter.

Then the corresponding cross sections for the specific value of the mean assembly distance are looked up in a custom cross-section library which was generated with the lattice code *CASMO5* under a variation of the fuel assembly pitch. These cross-sections are then used for the full core simulation in *DYN3D*.

In order to facilitate the investigation of different test scenarios, a Python framework was written to provide an interface of the different components of the simulation. This framework comprises the model order reduction, the time series generation, the conversion of the time series to the format used by *DYN3D* by calculating the average distance of each fuel assembly at each time step and node, launching *DYN3D* and executing the post processing. The distribution of the four different types of fuel assemblies can be customized according to the user's needs.

5.3 Results and Discussion

Different tests were simulated to assess the performance of the model. As an example, two test scenarios of a reactor core at the end of cycle at 100% power, are shown using two different damping values. The distribution of the fuel assemblies was performed according to their burnup as shown in Figure 69. The simulations were run with stiffness damping of 0.01 and 0.05 in this setup for 100 s. The postprocessing calculates the APSD as well as the coherence and phase at the in-core detector positions according to Figure 70. The results are depicted in Figure 71 and Figure 72 for the first test, and in Figure 73 and Figure 74 for the second test.

Both scenarios show a high coherence for low frequencies between the axial positions, with respect to the reference detector (channel 33). The APSDs undergo a typical decline for higher frequencies, but there are still peaks stemming from the mechanical eigenmodes for lower damping values. In both scenarios the axial phase is zero for low frequencies. There is also an out-of-phase behaviour for detectors in the core half opposite to the reference detector, which is more pronounced for lower damping.

Overall, the work shows qualitatively some right behaviour of in-core detector signals like out-of-phase behaviour of opposite detectors for low frequencies, whereas other characteristics especially in the high frequency-domain could not yet be modelled. It showed that it might be important to consider the effects of the different core loading schemes together with the stiffness of the fuel assemblies in order to model the aspects of the low frequency-domain of measurement signals and that additional effects have to be considered for the higher frequency aspects of the signals.

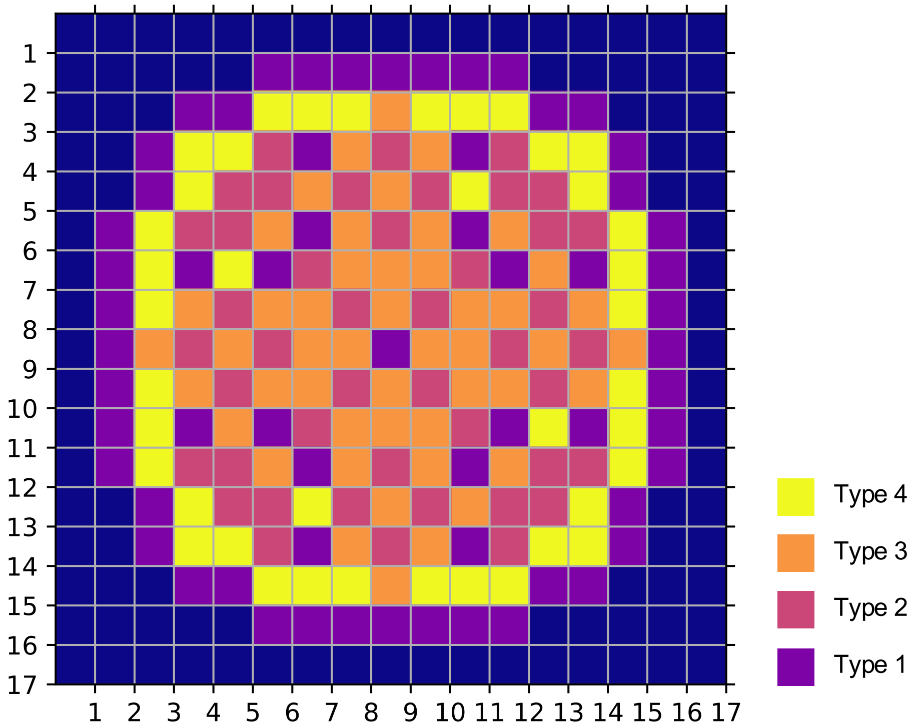


Figure 69: Distribution of the four types of fuel assemblies in the model. Higher number means lower stiffness.

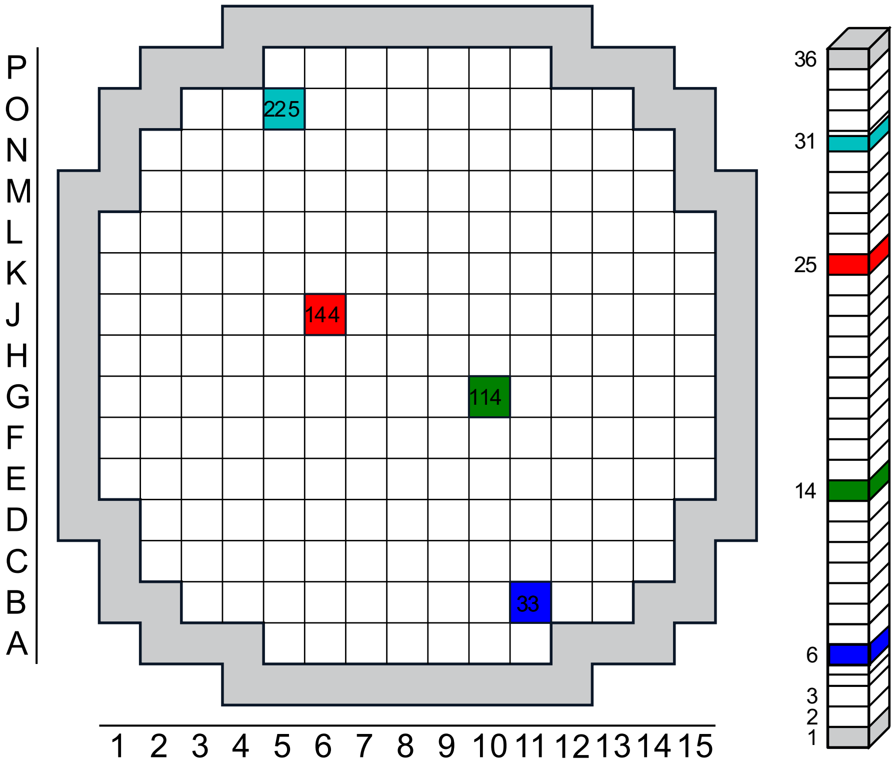


Figure 70: Detector positions in the *DYN3D* nodalization scheme. The numbers represent the *DYN3D* internal numbering of nodes for channels and axial layers.

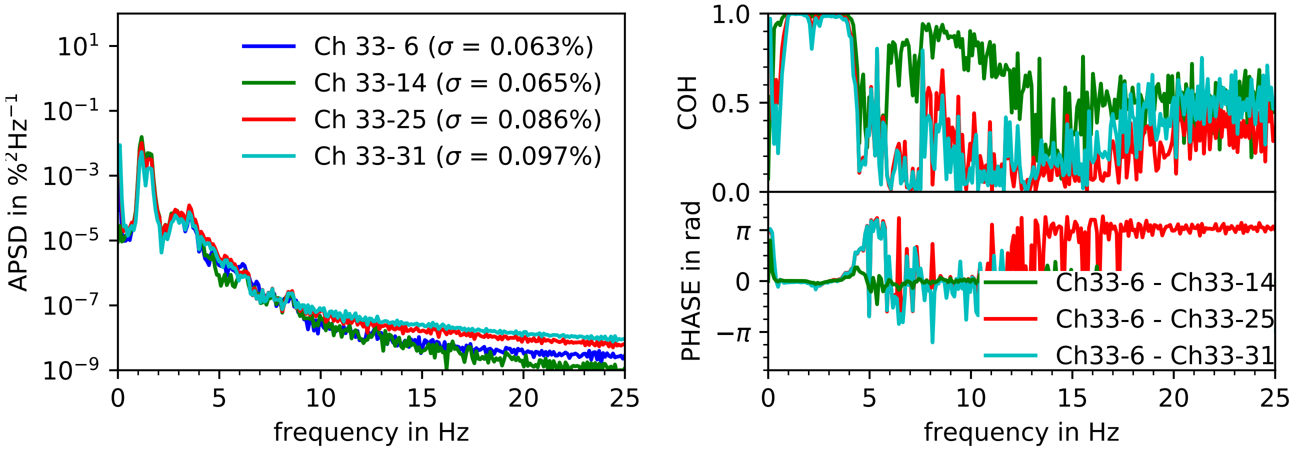


Figure 71: Axial dependency of APSD (left) and coherence and phase (right) of the detector signals at a value of 0.01 for stiffness damping.

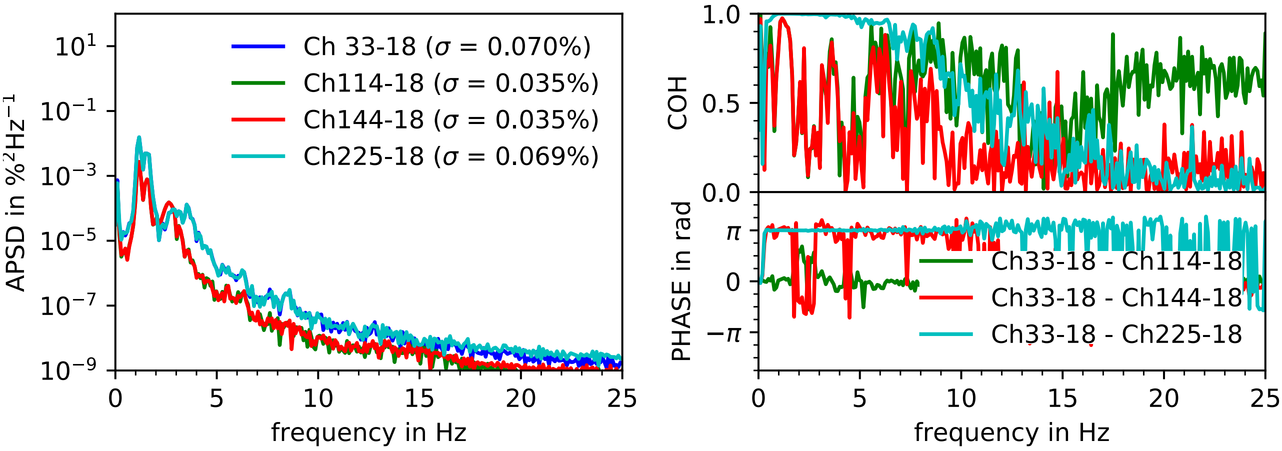


Figure 72: Radial dependency of APSD (left) and coherence and phase (right) of the detector signals at a value of 0.01 for stiffness damping.

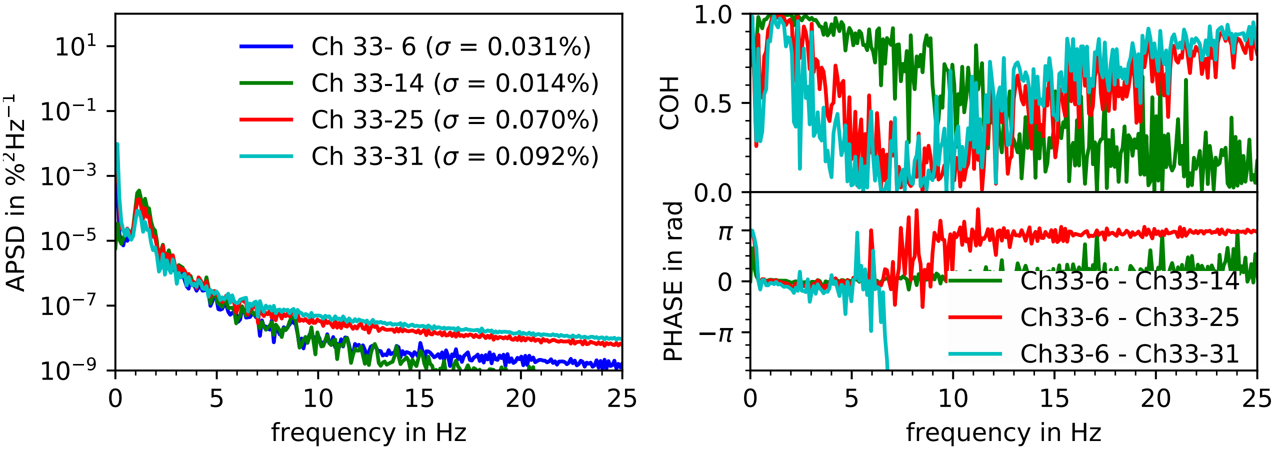


Figure 73: Axial dependency of APSD (left) and coherence and phase (right) of the detector signals at a value of 0.05 for stiffness damping.

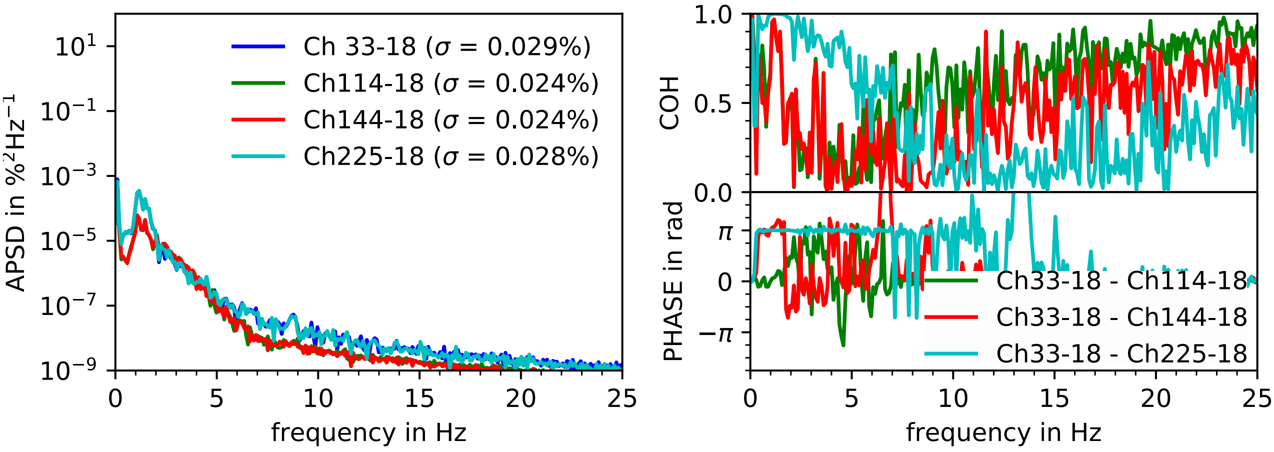


Figure 74: Radial dependency of APSD (left) and coherence and phase (right) of the detector signals at a value of 0.05 for stiffness damping.

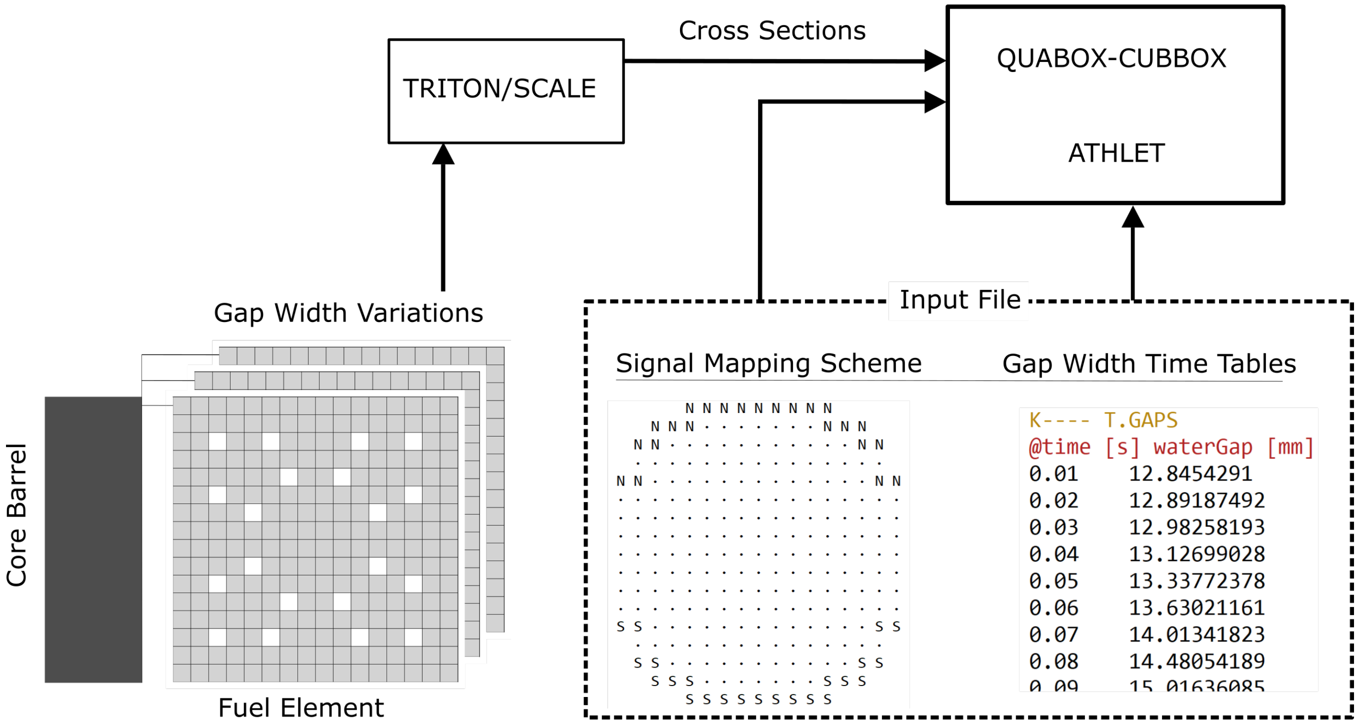
6 Modelling strategy at GRS

6.1 Introduction

GRS numerically investigated the effects of time-dependent size variations of the water gap between the outermost fuel element row and the core baffle as a very simple model of collective, coherent fuel element motions or core barrel vibrations. In addition to the reaction of the model core to different frequencies, the dependence on the burn-up state of the reactor was investigated. All calculations relied on the GRS core simulator KMACS [74] to control the coupled GRS codes ATHLET [75] and QUABOX-CUBBOX [76] for thermal-hydraulics and neutron flux calculations, respectively, as well as the TRITON [77] sequence from the SCALE package [78] by ORNL for the preparation of macroscopic nuclear cross-sections.

6.2 Model

Since neither QUABOX-CUBBOX nor ATHLET currently provide models for dynamic geometry, the water gap size dependence was encoded into the macroscopic radial reflector cross-sections by varying the respective geometry in the lattice calculations and by introducing the gap width as an additional dimension into the nuclear data tables already parametrized by boron concentration and moderator density. In order to control the spatial distribution and temporal evolution of the water gaps, a two-dimensional mapping was added to the ATHLET input which assigns ATHLET GCSM (General Control and Simulation Module) signals, governed by user-defined timing tables, to radial reflector assembly positions. The interface between ATHLET and QUABOX-CUBBOX had to be extended, so that the GCSM-encoded water gaps could be communicated to the flux solver to be then used as part of the reactor state in the interpolation of the nuclear cross-sections at runtime. A schema of the approach is depicted in Figure 75.



```
        N N N N N N N N N
    N N N . . . . . . . N N N
  N N . . . . . . . . . . . N N
  . . . . . . . . . . . . . . .
N N . . . . . . . . . . . . . N N
. . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . .
S S . . . . . . . . . . . . . S S
  . . . . . . . . . . . . . . .
  S S . . . . . . . . . . . S S
    S S S . . . . . . . S S S
        S S S S S S S S S
```

```
K---- T.GAPS
@time [s] waterGap [mm]
0.01     12.8454291
0.02     12.89187492
0.03     12.98258193
0.04     13.12699028
0.05     13.33772378
0.06     13.63021161
0.07     14.01341823
0.08     14.48054189
0.09     15.01636085
```

Figure 75: Schematic view of the modelling approach.

6.3 Application to the German four-loop pre KONVOI reactor

The *CASMO/SIMULATE* model of a German four-loop pre KONVOI reactor contributed by Preussen Elektra was manually converted to the input format of KMACS. Thus configured, the GRS core simulator calculated the time evolution of the reactor over one cycle. The resulting state at each burn-up point was saved and the conditions at Beginning Of Cycle (BOC), Middle Of Cycle (MOC) and End Of Cycle (EOC) served as starting points for transient calculations with ATHLET/QUABOX-CUBBOX.

6.3.1 Verification against the SIMULATE model and variation of cross-sections

Figure 76 shows a comparison of the boron let-down curve achieved using KMACS with the one computed by SIMULATE. The maximum deviation between the two curves is below 14 ppm and provides integral validation for the unperturbed KMACS model.

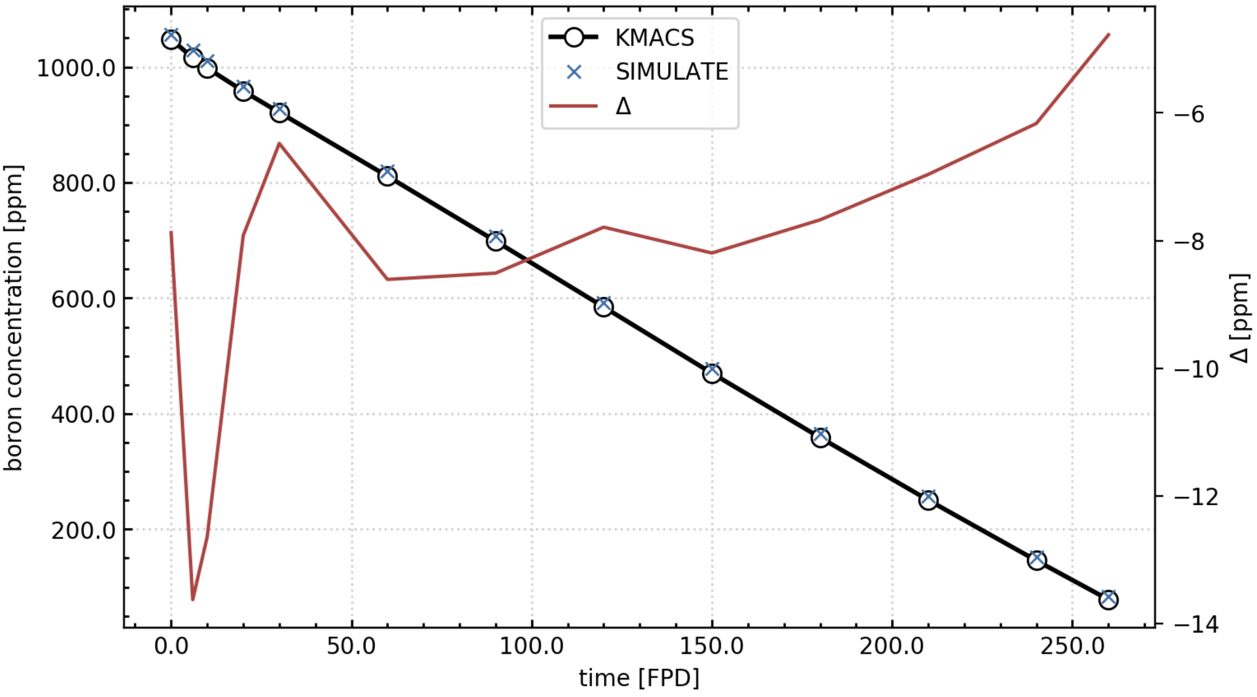


Figure 76: Verification of KMACS results against *CASMO/SIMULATE* using the boron let-down curve.

In the next step the nuclear data for the radial reflector were recalculated for several gap widths ranging from zero to the maximal achievable space of about 2.6 cm in the fuel element row containing the most elements. The resulting curves for the absorption (Σ_a) and the transport (Σ_t) cross-sections are shown in Figure 77 for nominal thermal-hydraulic conditions and unmodified geometry. Non-linear behaviour can be observed starting around 1 cm.

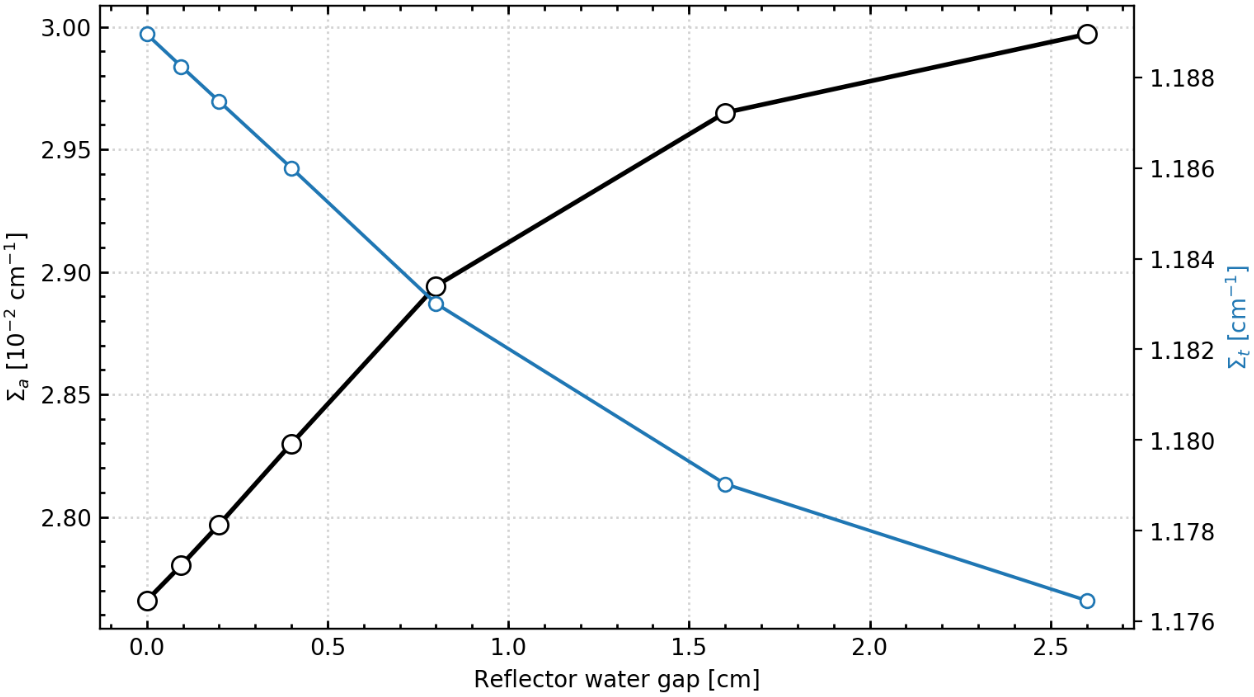


Figure 77: Water gap size dependence of the absorption and transport cross-sections in the radial reflector.

6.3.2 Transient calculations

Two ATHLET GCSM signals governing the gap widths were respectively assigned to the southern and northern halves of the reactor, according to the top view depicted in Figure 78. The two signals were varied in a sinusoidal manner and phase shifted against each other by 180 degrees. The amplitude depends on the maximal achievable space in a given column, i.e. 2.6 cm for the first and last rows. This unrealistic assumption was made in order to maximize the flux changes caused by the geometry variation. Unfortunately, the gap variations cannot be fully constrained to horizontal gaps due to model constraints: a reflector assembly whose cross sections are dynamically changed to reflect the time-dependence of the north/south distance to a neighbouring fuel assembly below or above may also have an east/west neighbour, so that in effect the water gaps in both directions are modified. The presented simulation results are taken from a horizontal plane in the middle of the reactor and a node located at the northern reflector.

The shape of the time evolution of the thermal flux in the measurement node is plotted against the gap signal in Figure 79 for an excitation frequency of 0.8 Hz. It can be observed that the general shape of the two functions appears to be the same, and that the flux lags the driving force.

As expected from experimental reactor measurements, Figure 80 exhibits the rise of the evoked flux variations during a burn-up cycle from the beginning (BOC) over the middle (MOC) to the end of cycle (EOC). Compared to BOC the amplitude increased at MOC by a factor of 1.4 and by 2.1 at EOC.

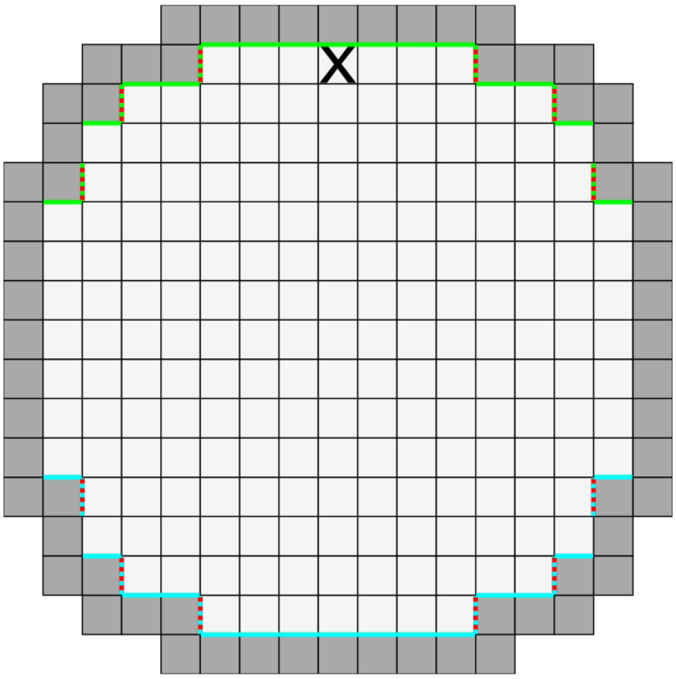


Figure 78: Division of horizontal fuel node-facing gaps at reflector nodes (grey) into a southern (turquoise) and northern (green) part governed by separate ATHLET signals. East-west gaps, whose variation is an unwanted side effect, are marked by red dots.

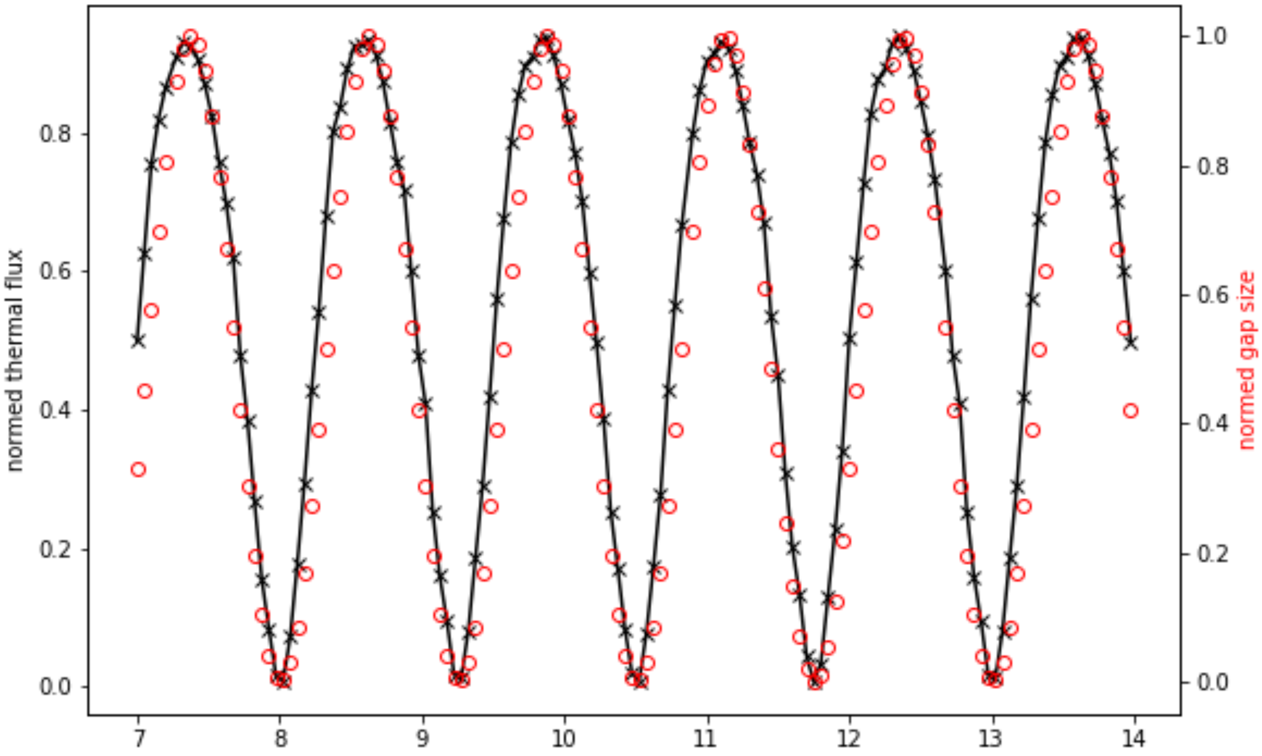


Figure 79: Comparison of the time evolution of the thermal flux compared to the gap signal.

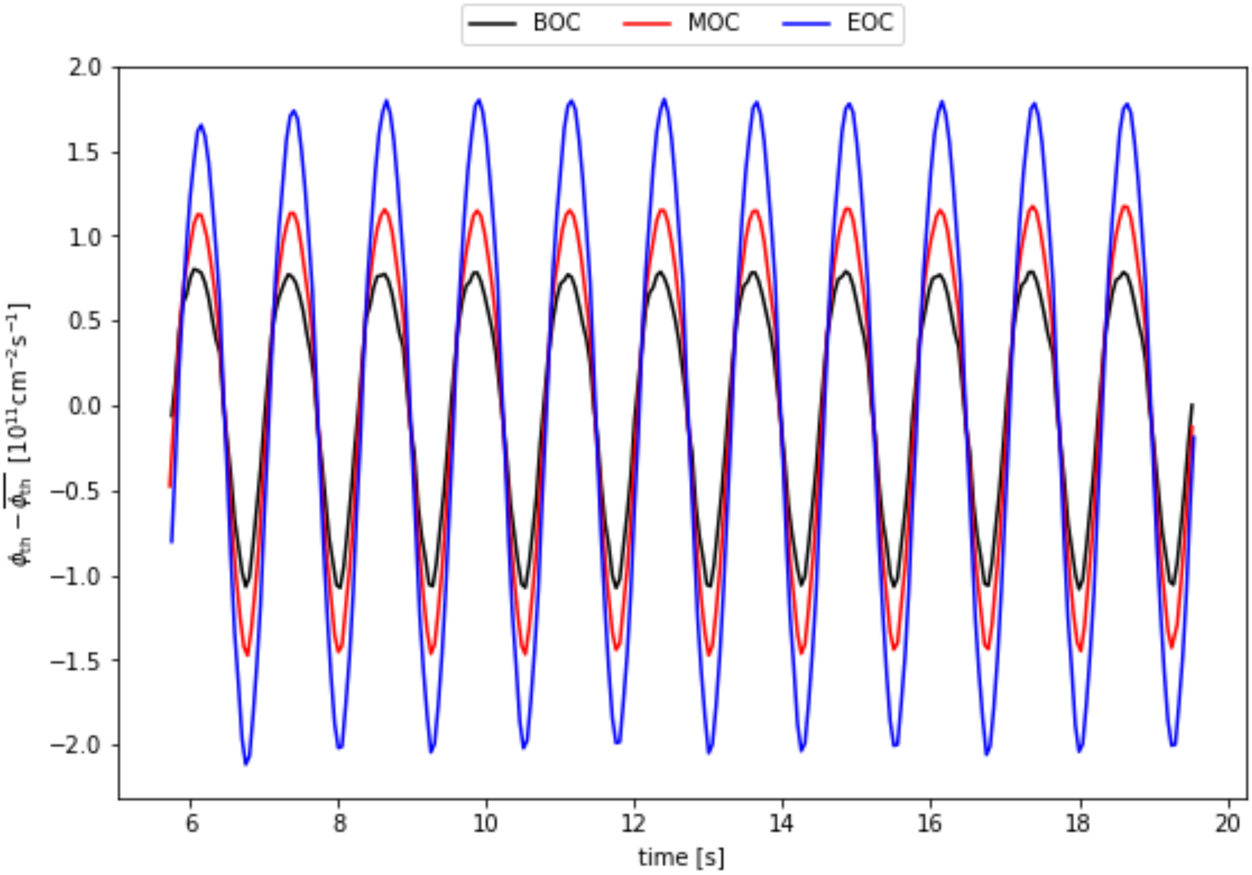


Figure 80: Increase of the evoked flux variations with cycle burn-up.

In order to analyse the frequency content of the evoked thermal flux signal, the Auto Power Spectral Density (APSD) was calculated for the abovementioned three cycle time points (Figure 81). A peak is clearly visible at the excitation frequency ν. As the burnup increases, so does the maximum of this peak. Aside from the excitation frequency, the data for BOC and MOC show the occurrence of a weakly developed second maximum at 2ν, which could be caused by non-linear effects. At MOC this second maximum decreased and it can barely be discerned at EOC. When the excitation frequency is increased to 8 Hz, not only does a higher harmonic frequency show up in the spectrum, but in addition, there is a peak at an intermediate frequency (Figure 82). The trend, that the first peak increasingly dominates the spectrum towards the end of the cycle can also be found at 8 Hz. Further studies are needed to rule out numerical as well as model artefacts and to thus corroborate the interpretation as higher order frequencies.

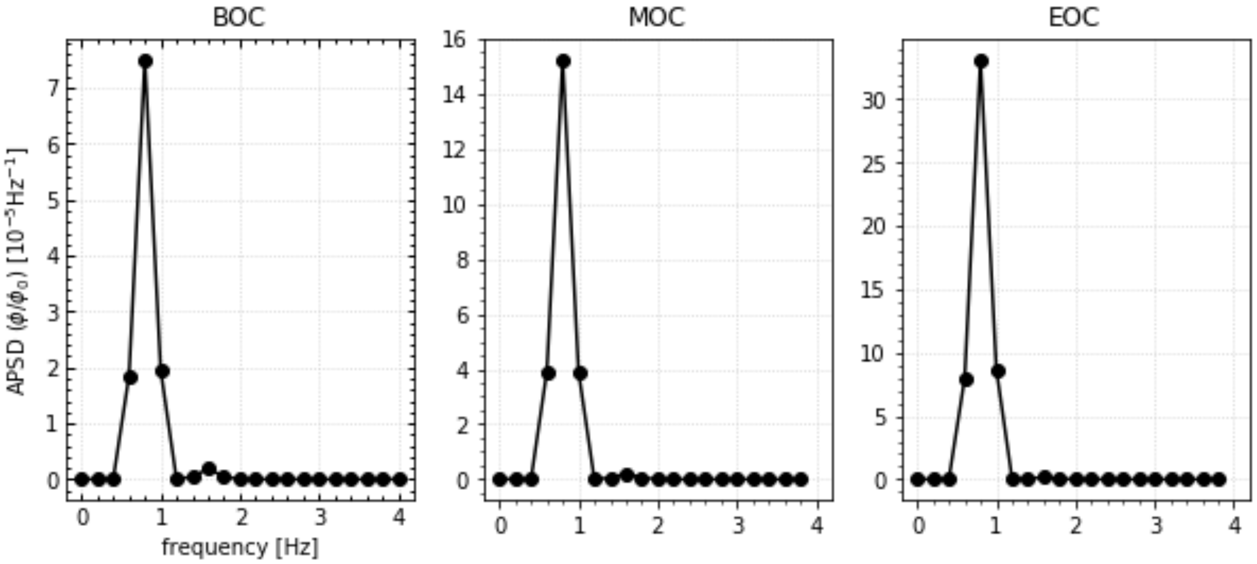


Figure 81: Power spectral density of the thermal flux normed by the static flux (0.8 Hz). The y-axes have different scales.

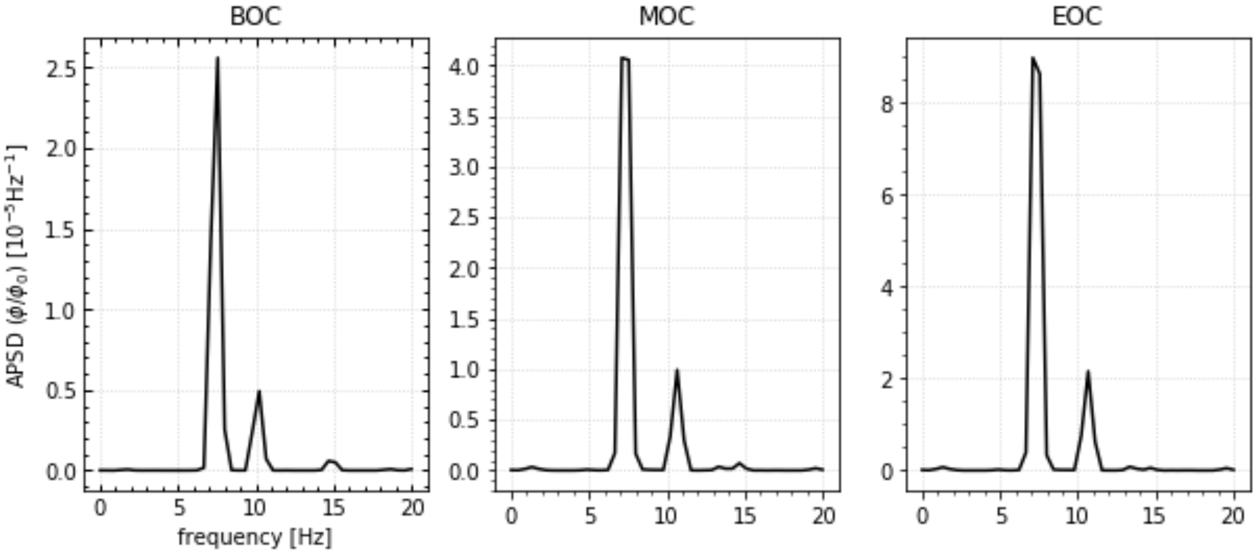


Figure 82: Power spectral density of the thermal flux normed by the static flux (8 Hz). The y-axes have different scales.

An overview of the spatial distribution of the induced fluctuations in the thermal flux, together with the difference between the maximum and minimum value in each node, is given in Figure 83. The signal amplitude amounts to 0.74% of the static flux (nominal gap widths) in the measurement node and reaches a maximum of 1.98% in the lateral nodes of the bottom and topmost rows in the considered horizontal reactor layer in which the effect is overestimated due to the unwanted east-west gap variation.

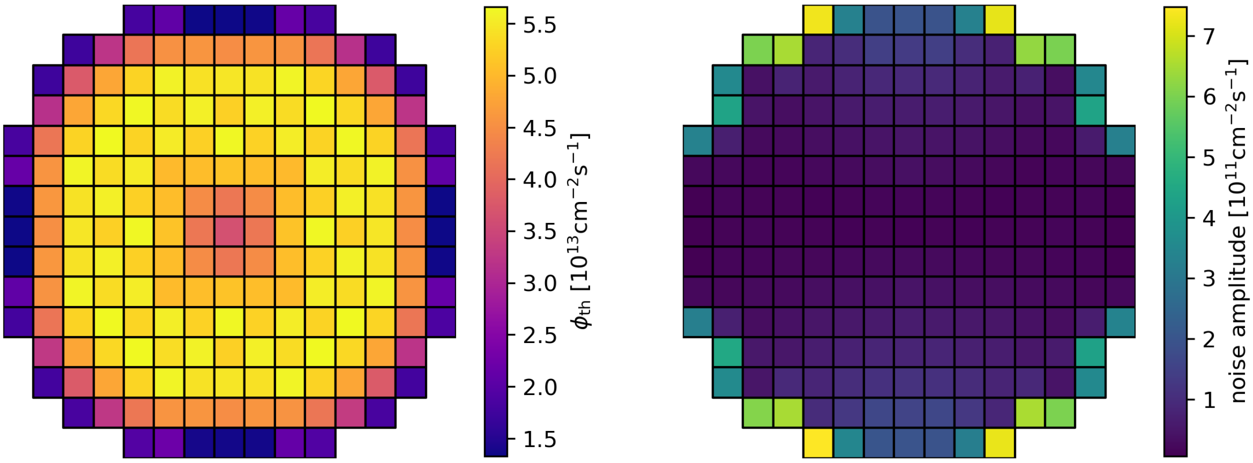


Figure 83: Spatial distribution of thermal flux (left) and difference between maximum and minimum signal induced (right) in each node of the middle horizontal layer.

6.4 Discussion

A model for time-dependent geometry has been implemented for the GRS code system *ATHLET*/*QUABOX-CUBBOX* employing a cross-section-based approach for encoding water gap width variations at the reflector. For this purpose, the reflector generation process had to be adapted, the water gap added as a parameter in the cross-section libraries and the coupling interface between *ATHLET* and *QUABOX/CUBBOX* extended in order to include the current gap size of each radial reflector node. The application of this newly developed model to the German four-loop pre KONVOI reactor, has shown that behaviour characteristic of neutron flux noise like the amplitude increase with cycle burn-up could be reproduced qualitatively. In addition, non-linear effects like frequency doubling have been detected in the computed power spectral densities. Their origin, either numerical, model-induced, or physical should be further investigated.

The GRS model as presented, suffers from several limitations, not all of which are easily addressed. It would certainly be possible to extend the ansatz to include an axial dependence of the gap widths, missing currently. In contrast the undesired collateral east-west excitations are more difficult to eliminate in this framework, because *QUABOX-CUBBOX* does not support anisotropic assembly discontinuity factors. This is also the reason why no attempt was made to extend the model to the analysis of vibrating fuel assemblies away from the reflector. Moreover, *QUABOX-CUBBOX* only allows global and not spatially resolved values for the fraction of delayed neutrons (β) and decay constants (λ_i). Above all, the employed neutron flux solver suffers from the numeric inadequacies inherent to all diffusion-based codes faced with large flux gradients, occurring in particular at the reactor boundary, which is targeted in this work. In view of these shortcomings, the usefulness of the approach may be restricted to the study of qualitative behaviour.

Nevertheless, the GRS approach has its advantages since it allows to examine the effects of cycle burn-up and to combine the analysis with thermal-hydraulic feedback models and noise sources. Moreover, results from advanced mechanical models can be incorporated. Further analyses will have to chart the applicability and explanatory power of the numerical model.

7 Conclusions

In this document, different types of methodologies for the study of the influence of the mechanical vibrations of fuel assemblies on the neutron flux in the reactor core, have been discussed. The methodologies employ the diffusion approximation to simulate the neutron noise in the time or frequency domain. The diffusion-based approach is expected to be less accurate in the vicinity of the vibrating fuel assemblies, but correct when considering distances larger than a few diffusion lengths away from the perturbation.

All methodologies provide consistent results and can reproduce typical features of the neutron noise induced by mechanical vibrations of core components. The neutron noise related to the fast energy groups has a more diffusive behaviour in the reactor core because of the larger mean free paths of fast neutrons. On the other hand, the thermal neutron noise is more localized, and it has a higher amplitude because of the higher impact of the perturbation on the absorption cross sections. In addition, the expected out-of-phase behaviour of the neutron noise is predicted in opposite sides of the reactor with respect to the location of vibration.

Spectral analysis of the neutron noise in the frequency domain shows that the main frequency of the neutron noise corresponds to the fundamental frequency of the mechanical vibrations. Higher harmonics may play a role, although the contribution estimated with the models described in this report, is small.

UPV has developed the code FEMFFUSION/FEMFUSSION-FD and neutron noise models for the core simulator PARCS. The efforts have been centred in showing the consistency between the time-domain and the frequency-domain calculations. Numerical results identify two different effects in the neutron field caused by the fuel assembly vibration. First, a global slow variation of the power is observed because of a change in the criticality of the system. This effect is small and will be compensated by the thermal-hydraulic coupling because, in an operating nuclear reactor, the temperature coefficient of reactivity is negative. Second, an oscillation of the neutron flux with the same frequency as the assembly mechanical vibration is demonstrated. The corresponding neutron noise is highly spatially dependent. For this second effect, a very close agreement between the time-domain and the frequency-domain approach is found.

Chalmers University of Technology has developed the frequency-domain code CORESIM+. The code can be used to study various neutron noise scenarios in realistic three-dimensional reactor configurations. Its numerical scheme allows to build more accurate models of neutron noise sources and to perform reactor neutron noise calculations in an optimized manner, at a relatively cheap computational cost. CORE SIM+ has been verified and tested on different numerical and experimental neutron noise problems. Examples of vibrations of fuel assemblies and core barrel in a PWR, were analysed.

PSI has developed a neutron noise methodology using the commercial simulation tools CASMO5/SIMULATE-3/SIMULATE-3K. The methodology allows to simulate in the time domain the neutron noise induced by different neutron noise sources in a nuclear reactor. In particular, the modelling of fuel assembly vibrations is based on the perturbation of the macroscopic cross-sections that is obtained from the variation of the water gap of the vibrating fuel assemblies. Thermal-hydraulic feedback effects are also included in the simulations. Using this methodology, the characteristics of neutron noise due to different vibration modes of fuel assemblies, with and without additional fluctuations of coolant temperature and flow, have been investigated in a PWR.

TUD has developed a methodology that combines a reduced order model for mechanical vibrations of reactor core components and a model for full-core neutron noise calculations with the time-domain reactor code DYN3D. The methodology has been applied to study the neutron noise of a 4-loop pre-Konvoi PWR when considering the vibration of all the core components at the same time. The results showed that the core loading scheme and the stiffness of the fuel assemblies may play an important role in the neutron noise.

GRS has developed models for the simulation of neutron noise induced by core barrel vibrations, using the coupling between the neutronic code QUABOX-CUBBOX and the thermal-hydraulic code ATHLET, via the reactor code KMACS. The strategy is based on modelling the time-dependent variation of the water gap between the outermost row of fuel elements and the reactor core baffle. The dependence of the calculated reactor neutron noise on burnup and frequency has been investigated for the case of a German 4-loop pre-Konvoi PWR. These simulations have predicted an increase of the neutron noise amplitude with burnup, similarly to what has been observed in plant measurements. In addition, frequencies different from the fundamental one may contribute to the neutron noise and the burnup may affect such contributions.

8 References

[1] G.G. Davidson, T.M. Evans, J.J. Jarrell, S.P. Hamilton, T.M. Pandya, R.N. Slaybaugh, Massively Parallel , Three-Dimensional Transport Solutions for the k-Eigenvalue Problem, Nucl. Sci. Eng. 177 (2014) 111–125. https://doi.org/10.13182/NSE12-101.

[2] I. Pázsit, Investigation of the space-dependent noise induced by a vibrating absorber, Atomkernenergie. 30 (1977) 29–35.

[3] I. Pázsit, Control-rod models and vibration induced noise, Ann. Nucl. Energy. 15 (1988) 333–346. https://doi.org/10.1016/0306-4549(88)90081-3.

[4] C. Demazière, P. Vinai, M. Hursin, S. Kollias, J. Herb, Noise-Based Core Monitoring and Diagnostics - Overview of the project, in: Adv. React. Phys., Mumbai, India, 2017: pp. 1–4.

[5] M. Seidl, K. Kosowski, U. Schüler, L. Belblidia, Review of the historic neutron noise behavior in German KWU built PWRs, Prog. Nucl. Energy. 85 (2015) 668–675. https://doi.org/10.1016/J.PNUCENE.2015.08.016.

[6] D.N. Fry, J. March-Leuba, F.J. Sweeney, Use of neutron noise for diagnosis of in-vessel anomalies in light-water reactors, (1984).

[7] D.N. Fry, Experience in Reactor Malfunction Diagnosis Using On-Line Noise Analysis, Nucl. Technol. 10 (1971) 273–282. https://doi.org/10.13182/NT71-A30959.

[8] V. Lamirand, M. Hursin, A. Rais, S. Hübner, C. Lange, J. Pohlus, U. Paquee, C. Pohl, O. Pakari, A. Laureau, CORTEX D2.1: Experimental report of the 1st campaign at AKR-2 and CROCUS, 2018. https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bfaf5ce8&appId=PPGMS.

[9] D. Chionis, A. Dokhane, L. Belblidia, M. Pecchia, G. Girardin, H. Ferroukhi, A. Pautz, SIMULATE-3K Analyses of Neutron Noise Response to Fuel Assembly Vibrations and Thermal-Hydraulics Parameters Fluctuations, in: M&C 2017 - Int. Conf. Math. Comput. Methods Appl. to Nucl. Sci. Eng., Jeju, Korea, 2017.

[10] M. Viebach, N. Bernt, C. Lange, D. Hennig, A. Hurtado, On the influence of dynamical fuel assembly deflections on the neutron noise level, Prog. Nucl. Energy. 104 (2018) 32–46. https://doi.org/10.1016/J.PNUCENE.2017.08.010.

[11] N. Olmo-Juan, C. Demazière, T. Barrachina, R. Miró, G. Verdú, PARCS vs CORE SIM neutron noise simulations, Prog. Nucl. Energy. 115 (2019) 169–180. https://doi.org/10.1016/j.pnucene.2019.03.041.

[12] A. Vidal-Ferràndiz, A. Carreño, D. Ginestar, C. Demazière, G. Verdú, A time and frequency domain analysis of the effect of vibrating fuel assemblies on the neutron noise, Ann. Nucl. Energy. 137 (2020) 107076. https://doi.org/10.1016/j.anucene.2019.107076.

[13] J. Park, J.H. Lee, T.-R. Kim, J.-B. Park, S.K. Lee, I.-S. Koo, Identification of reactor internals' vibration modes of a Korean standard PWR using structural modeling and neutron noise analysis, Prog. Nucl. Energy. 43 (2003) 177–186. https://doi.org/10.1016/S0149-1970(03)00021-0.

[14] T. Yamamoto, Monte Carlo method with complex weights for neutron leakage-corrected calculations and anisotropic diffusion coefficient generations, Ann. Nucl. Energy. 50 (2012) 141–149. https://doi.org/10.1016/j.anucene.2012.06.025.

[15] W.M. Stacey, Nuclear Reactor Physics, 2nd ed., Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2007. https://doi.org/10.1002/9783527611041.

[16] A. Vidal-Ferràndiz, R. Fayez, D. Ginestar, G. Verdú, Solution of the Lambda modes problem of a nuclear power reactor using an h-p finite element method, Ann. Nucl. Energy. 72 (2014) 338–349. https://doi.org/10.1016/j.anucene.2014.05.026.

[17] W. Bangerth, R. Hartmann, G. Kanschat, deal.II—A general-purpose object-oriented finite element library, ACM Trans. Math. Softw. 33 (2007) 24-es. https://doi.org/10.1145/1268776.1268779.

[18] W. Bangerth, O. Kayser-Herold, Data Structures and Requirements for hp Finite Element

Software, ACM Trans. Math. Softw. 36 (2009) 4/1-4/31.

[19] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, S. Zampini, H. Zhang, PETSc Users Manual, 2016. http://www.mcs.anl.gov/petsc.

[20] V. Hernández, J.E. Román, V. Vidal, SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems, ACM Trans. Math. Softw. 31 (2005) 351–362.

[21] G. Verdú, D. Ginestar, V. Vidal, J.L. Muñoz-Cobo, A consistent multidimensional nodal method for transient calculations, Ann. Nucl. Energy. 22 (1995) 395–410. https://doi.org/10.1016/0306-4549(94)00067-O.

[22] D. Ginestar, G. Verdú, V. Vidal, R. Bru, J. Marín, J.L. Muñoz-Cobo, High order backward discretization of the neutron diffusion equation, Ann. Nucl. Energy. 25 (1998) 47–64. https://doi.org/10.1016/S0306-4549(97)00046-7.

[23] R. Bru, D. Ginestar, J. Marin, G. Verdú, J. Mas, T. Manteuffel, Iterative schemes for the neutron diffusion equation, Comput. Math. with Appl. 44 (2002) 1307–1323.

[24] S. González-Pintor, D. Ginestar, G. Verdú, Preconditioning the solution of the time-dependent neutron diffusion equation by recycling Krylov subspaces, Int. J. Comput. Math. 91 (2014) 42–52.

[25] R. Miró, Métodos Modales para el Estudio de Inestabilidades en Reactores Nucleares BWR, Universitat Politècnica de València, 2002.

[26] A. Carreño, A. Vidal-Ferràndiz, D. Ginestar, G. Verdú, Modal methods for the neutron diffusion equation using different spatial modes, Prog. Nucl. Energy. 115 (2019) 181–193. https://doi.org/10.1016/j.pnucene.2019.03.040.

[27] S. Dulla, E.H. Mund, P. Ravetto, The quasi-static method revisited, Prog. Nucl. Energy. 50 (2008) 908–920.

[28] A. Vidal-Ferràndiz, A. Carreño, D. Ginestar, G. Verdú, FEMFFUSION, an open finite element code for nuclear reactor modelling, (2018). https://bitbucket.org/Zonni/femffusion/src.

[29] G. Verdú, D. Ginestar, V. Vidal, J.L. Muñoz-Cobo, 3D Lambda-modes of the neutron-diffusion equation, Ann. Nucl. Energy. 21 (1994) 405–421.

[30] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, 6th edition, Butterworth-Heinemann, 2005.

[31] S. González-Pintor, D. Ginestar, G. Verdú, High Order Finite Element Method for the Lambda modes problem on hexagonal geometry, Ann. Nucl. Energy. 36 (2009) 1450–1462. https://doi.org/10.1016/j.anucene.2009.07.003.

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

[33] A. Carreño, A. Vidal-Ferràndiz, D. Ginestar, G. Verdú, Multilevel method to compute the lambda modes of the neutron diffusion equation, Appl. Math. Nonlinear Sci. 2 (2017) 225–236. https://doi.org/10.21042/AMNS.2017.1.00019.

[34] A. Carreño, A. Vidal-Ferràndiz, D. Ginestar, G. Verdú, Block hybrid multilevel method to compute the dominant λ-modes of the neutron diffusion equation, Ann. Nucl. Energy. 121 (2018) 513–524. https://doi.org/10.1016/j.anucene.2018.08.010.

[35] C. Demazière, CORE SIM: A multi-purpose neutronic tool for research and education, Ann. Nucl. Energy. 38 (2011) 2698-Block Krylov–Schur method for large symmetric. https://doi.org/10.1016/j.anucene.2011.06.010.

[36] Purdue University, Purdue Advanced Reactor Core Simulator, (2014). https://engineering.purdue.edu/PARCS.

[37] C. Demazière, G. Andhill, Identification and localization of absorbers of variable strength in nuclear reactors, Ann. Nucl. Energy. 32 (2005) 812–842. https://doi.org/10.1016/J.ANUCENE.2004.12.011.

[38] C. Demazière, Analysis methods for the determination of possible unseated fuel assemblies

in BWRs, Int. J. Nucl. Energy Sci. Technol. 2 (2006) 167. https://doi.org/10.1504/IJNEST.2006.010713.

[39] C. Demazière, I. Pázsit, Numerical tools applied to power reactor noise analysis, Prog. Nucl. Energy. 51 (2009) 67–81. https://doi.org/10.1016/J.PNUCENE.2008.01.010.

[40] V. Larsson, C. Demazière, Comparative study of 2-group P1 and diffusion theories for the calculation of the neutron noise in 1D 2-region systems, Ann. Nucl. Energy. 36 (2009) 1574–1587. https://doi.org/10.1016/j.anucene.2009.07.009.

[41] I. Pázsit, C. Demazière, Noise Techniques in Nuclear Systems, Handb. Nucl. Eng. (2010) 1629–1737. https://doi.org/10.1007/978-0-387-98149-9_14.

[42] A. Rouchon, Analyse et développement d ' outils numériques déterministes et stochastiques résolvant les équations du bruit neutronique et applications aux réacteurs thermiques et rapides, Université Paris-Saclay, 2016. https://tel.archives-ouvertes.fr/tel-01381245.

[43] Z. Akcasu, General Solution of the Reactor Kinetic Equations without Feedback, Nucl. Sci. Eng. 3 (1958) 456–467. https://doi.org/10.13182/NSE58-A25482.

[44] P. Ravetto, Reactivity oscillations in a point reactor, Ann. Nucl. Energy. 24 (1997) 303–314. https://doi.org/10.1016/S0306-4549(96)00066-7.

[45] A. Hébert, Application of the Hermite Method for Finite Element Reactor Calculations, Nucl. Sci. Eng. 91 (1985) 34–58. https://doi.org/10.13182/NSE85-A17127.

[46] A. Gammicchia, S. Santandrea, I. Zmijarevic, R. Sanchez, Z. Stankovski, S. Dulla, P. Mosca, A MOC-based neutron kinetics model for noise analysis, Ann. Nucl. Energy. 137 (2020) 107070. https://doi.org/10.1016/j.anucene.2019.107070.

[47] C. Demazière, V. Dykin, K. Jareteg, Development of a point-kinetic verification scheme for nuclear reactor applications, J. Comput. Phys. 339 (2017) 396–411. https://doi.org/10.1016/j.jcp.2017.03.020.

[48] H. Malmir, N. Vosoughi, Propagation noise calculations in VVER-type reactor core, Prog. Nucl. Energy. 78 (2015) 10–18. https://doi.org/10.1016/J.PNUCENE.2014.07.010.

[49] G. Alexandridis, C. Bläsius, C. Demazière, C. Destouches, A. Dokhane, A. Durrant, V. Fiser, G. Girardin, J. Herb, G. Ioannou, R. Jacqmin, A. Knospe, S. Kollias, C. Lange, G. Leontidis, S. Lipcsei, C. Montalvo, A. Mylonakis, L. Pantera, Y. Perin, C. Pohl, M. Seidl, A. Stafylopatis, P. Stulik, L.A. Torres, G. Verdú, V. Verma, A. Vidal-Ferràndiz, M. Viebach, P. Vinai, CORTEX D4.4: Results of the application and demonstration calculations, 2020. http://cortex-h2020.eu/wp-content/uploads/2020/08/CORTEX-D4.4-Results-of-the-application-and-demonstration-calculations.pdf.

[50] École Polytechnique Fédérale de Lausanne, CROCUS Reactor: A Fission Reactor on Campus, (2020). https://www.epfl.ch/labs/lrs/facilities/crocus-reactor/.

[51] A.G. Mylonakis, P. Vinai, C. Demazière, Numerical solution of two-energy-group neutron noise diffusion problems with fine spatial meshes, Ann. Nucl. Energy. 140 (2020) 107093. https://doi.org/10.1016/j.anucene.2019.107093.

[52] D.R. Ferguson, K.L. Derstine, Optimized Iteration Strategies and Data Management Considerations for Fast Reactor Finite Difference Diffusion Theory Codes, Nucl. Sci. Eng. 64 (1977) 593–604. https://doi.org/10.13182/NSE77-5.

[53] D.F. Gill, Y. Azmy, Newton's Method for Solving k-Eigenvalue Problems in Neutron Diffusion Theory, Nucl. Sci. Eng. 167 (2011) 141–153.

[54] C. Demazière, A. Dokhane, CORTEX D3.1: Description of scenarios for the simulated data, 2019. http://cortex-h2020.eu/wp-content/uploads/2019/07/CORTEX_D3_1_Description_of_scenarios_for_the_simulated_data_V1.pdf.

[55] A. Mylonakis, C. Demazière, P. Vinai, V. Lamirand, A. Rais, O. Pakari, P. Frajtag, D. Godat, M. Hursin, G. Perret, A. Laureau, C. Fiorina, A. Pautz, CORE SIM+ simulations of COLIBRI fuel rods oscillation experiments and comparison with measurements, in: Int. Conf. Phys. React. - PHYSOR 2020 Transit. to a Scalable Nucl. Futur., Cambridge, UK, 2020.

[56] G.I. Bell, S. Glasstone, Nuclear reactor theory, Van Nostrand Reinhold Company, New

York, USA, 1970.

[57] T. Kozlowski, T.J. Downar, PWR MOX/UO2 Core Transient Benchmark, 2006. https://www.oecd-nea.org/science/wprs/MOX-UOX-transients/benchmark_documents/final_report/mox-uo2-bench.pdf.

[58] V. Lamirand, P. Frajtag, D. Godat, O. Pakari, A. Laureau, A. Rais, M. Hursin, G. Hursin, C. Fiorina, A. Pautz, The COLIBRI experimental program in the CROCUS reactor: characterization of the fuel rods oscillator, in: EPJ Web Conf., 2020: p. 225:04020.

[59] V. Verma, C. Demazière, P. Vinai, G. Ricciardi, R. Jacqmin, Assessment of the neutron noise induced by stationary fuel assembly vibrations in a light water reactor, in: Int. Conf. Math. Comput. Methods Appl. to Nucl. Sci. Eng. - M&C 2019, 2019.

[60] J. Rhodes, K. Smith, D.J. Lee, CASMO-5 development and applications, in: PHYSOR 2006 Adv. Nucl. Anal. Simul., American Nuclear Society, Vancuber, BC, Canada, n.d. https://doi.org/978-0-89448-697-5.

[61] R. Ferrer, CASMO-5 methodology manual - SSP Technical Report SSP-08/405, Rev. 4, 2015.

[62] T. Bahadir, CMS-LINK user 's manual - SSP Technical Report SOA-97/04, Rev. 2, 1999.

[63] J. Cronin, K. Smith, D. Ver Planck, SIMULATE-3 methodology - SSP Technical Report SOA-95/18, Rev 0., 1995.

[64] G. Grandi, SIMULATE-3K models & methodology - SSP Technical Report SSP-98/13, Rev. 7, 2011.

[65] H. Ferroukhi, K. Hofer, J. Hollard, A. Vasiliev, M.A. Zimmermann, Core modelling and analysis of the Swiss nuclear power plants for qualified R&D applications, in: PHYSOR'08 Int. Conf. Phys. React. 'Nuclear Power A Sustain. Resour., Interlaken (Switzerland), 2008.

[66] D. Chionis, Development of Advanced Methodologies for Monitoring and Modelling of Neutron Noise in Modern LWR Cores, École Politechnique Federale de Lausanne (EPFL), 2020.

[67] D. Chionis, A. Dokhane, L. Belblidia, H. Ferroukhi, G. Girardin, A. Pautz, Development and verification of a methodology for neutron noise response to fuel assembly vibrations, Ann. Nucl. Energy. 147 (2020) 107669. https://doi.org/10.1016/j.anucene.2020.107669.

[68] V. Verma, D. Chionis, A. Dokhane, H. Ferroukhi, Studies of reactor noise response to vibrations of reactor internals and thermal-hydraulic fluctuations in PWRs, Submitt. to Ann. Nucl. Energy. (2020).

[69] U. Rohde, S. Kliem, U. Grundmann, S. Baier, Y. Bilodid, S. Duerigen, E. Fridman, A. Gommlich, A. Grahn, L. Holt, Y. Kozmenkov, S. Mittag, The reactor dynamics code DYN3D – models, validation and applications, Prog. Nucl. Energy. 89 (2016) 170–190. https://doi.org/10.1016/j.pnucene.2016.02.013.

[70] C. Blaesius, J. Herb, J. Sievers, A. Knospe, C. Lange, CORTEX D1.2: Modelling of FSIs for reactor vessel internals, 2020. http://cortex-h2020.eu/wp-content/uploads/2020/04/CORTEX_D1_2_Modelling_of_FSIs_for_reactor_vessel_internals_V2.pdf.

[71] A. Steindl, H. Troger, Methods for dimension reduction and their application in nonlinear dynamics, Int. J. Solids Struct. 38 (2001) 2131–2147. https://doi.org/10.1016/S0020-7683(00)00157-8.

[72] V. Lenaerts, G. Kerschen, J.C. Golinval, Proper orthogonal decomposition for model updating of non-linear mechanical systems, Mech. Syst. Signal Process. 15 (2001) 31–43. https://doi.org/10.1006/mssp.2000.1350.

[73] M. Viebach, C. Lange, M. Seidl, M. Bilodid, A. Hurtado, Neutron Noise Patterns from Coupled Fuel Assembly Vibrations, in: Int. Conf. Phys. React. - PHYSOR 2020 Transit. to a Scalable Nucl. Futur., Cambridge, United Kingdom, 2020.

[74] J. Bousquet, M. Zilly, M. Küntzel, KMACS User Manual, Version 1.0, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, 2018.

[75] Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, ATHLET 3.2 - Program

Updates Since Release ATHLET 3.1A, 2019.

[76] S. Langenbuch, W. Maurer, W. Werner, Coarse-Mesh Flux-Expansion Method for the Analysis of Space-Time Effects in Large Light Water Reactor Cores, Nucl. Sci. Eng. 63 (1977) 437–456. https://doi.org/10.13182/NSE77-A27061.

[77] B.J. Ade, SCALE / TRITON Primer : A Primer for Light Water Reactor Lattice Physics Calculations (NUREG/CR-7041, ORNL/TM-2011/21), Oak Ridge, TN, USA, 2012.

[78] W.A. Wieselquist, R.A. Lefebvre, M. A. Jessee, SCALE Code System, ORNL/TM-2005/39, Oak Ridge, TN, 2020. https://www.ornl.gov/file/scale-62-manual/display.

9 Annexes

9.1 Codes

Table 13: Python script to generate custom cross-sections for vibrating FA.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
This script creates a geometry file and a custom cross section file for PARCS
@author: Antoni Vidal @ UPV
@email anvifer2@upv.es
"""
import numpy as np
from utils_vibration import print_cell

exp = 12
xsec_file = '2D_CROCUS/2D_CROCUS_uniform_'+ str(exp) + '.xs'
geom_file = '2D_CROCUS/2D_CROCUS_uniform_'+ str(exp) + '.gm'
inpt_file = '2D_CROCUS/2D_CROCUS_uniform_'+ str(exp) + '.inp'


if (exp == 12):
    freq = 0.1; # Hz
    amplitude = 0.2; #cm

omega = 2 * np.pi * freq;
keff = 1.0306557975758472
t_end = 3.0 / freq
delta_t = t_end / 300.0

n_cells = 1569

n_cells_x = 44
n_cells_y = 44
cell_pitch_x = 44* [2.917]
cell_pitch_y = 44* [2.917]

refs_x = 1; # x - Refinements per cell before PARCS, 1 means no refinements
neut_x = 4; # x - Internal PARCS, cell per each defined cell
refs_y = 1; # y - Refinements per cell before PARCS, 1 means no refinements
neut_y = 4; # y- Internal PARCS, cell per each defined cell

# Vibration Position
vib_pos_0 = [12*2.917, 15*2.917,  19*2.917, 25*2.917]
vib_pos_0 = np.array(vib_pos_0)
vib_mat = 2

# (0 = reflective, 2 = vaccum)
# !ibc_west ibc_east ibc_north ibc_south ibc_bottom ibc_top
bound_coond = '2 2 2 2 0 0'

# Radial Configuartion
# 0 - No material, 2 - Fuel
rad_conf     = [] # Radial configuration

assert(np.count_nonzero(rad_conf) == n_cells)
assert(len(rad_conf) == n_cells_y)
for i in range(n_cells_y):
    assert(len(rad_conf[i]) == n_cells_x)

materials = []      # Materials configuration
assert(len(materials) == n_cells)



#   [sigma_tr1    sigma_a1      nu_sigma_f1   k_sigma_f1   sigma_12
#    sigma_tr2    sigma_a2    nu_sigma_f2    k_sigma_f2]
xs = [
   [2.64361E-01,  7.02429E-03, 4.67255E-03, 4.67255E-03, 2.72040E-02,
    1.38380E+00, 6.91678E-02, 9.71695E-02, 9.71695E-02],
# 1002 Outer Fuel
   [2.84424E-01,  8.76391E-03, 6.57177E-03, 6.57177E-03, 3.05630E-02,
    1.59262E+00, 6.54373E-02, 7.51222E-02, 7.51222E-02],
# 1003 Control Rods
   [2.22204E-01,  1.72495E-03, 8.58287E-04, 8.58287E-04, 3.38760E-02,
    1.43289E+00, 1.94197E-02, 8.85938E-03, 8.85938E-03],
# 1004 Reflector
   [2.78811E-01,  5.49966E-04, 9.65025E-07, 9.65025E-07, 5.99460E-02,
    2.20208E+00, 1.89578E-02, 3.42205E-06, 3.42205E-06],
# 1005 Top inner fuel
   [8.62859E-02,  4.02325E-03, 3.46517E-03, 3.46517E-03, 1.30000E-04,
    1.44127E-01, 4.65613E-02, 7.67696E-02, 7.67696E-02],
# 1006 Top outer fuel
   [8.57050E-02,  5.38474E-03, 4.79915E-03, 4.79915E-03, 2.79930E-05,
    1.40871E-01, 4.80653E-02, 6.74231E-02, 6.74231E-02],
# 1007 Top ctrl rods
   [2.65900E-02,  1.58468E-03, 6.22365E-04, 6.22365E-04, 2.25970E-05,
    3.78245E-02, 1.39969E-02, 8.89872E-03, 8.89872E-03]]

"============================================================================="

def make_critical(xs, keff):
    """ """
    for mat in range(len(xs)):
        for x in range(len(xs[mat])):
            if (x == 2 or x == 3 or x == 7 or x == 8 ):
                xs[mat][x] /= keff
    return xs

xs = make_critical(xs, keff)


assert(len(cell_pitch_x) == n_cells_x )
assert(len(cell_pitch_y) == n_cells_y )

cell_pitch_x_orig = cell_pitch_x.copy()
cell_pitch_y_orig = cell_pitch_y.copy()
#n_cells, materials, cell_pitch_x, cell_pitch_y = refine_2D(refs_x,
#                                                           refs_y,
#                                                           n_cells,
#                                                           materials,
#                                                           cell_pitch_x,
#                                                           cell_pitch_y)
```

```
"============================================================================="
# Some useful functions
def make_geometry_file():
    """ Make Geomtry File"""
    out=open(geom_file, 'w')

    print(' geo_dim ', n_cells_x, n_cells_y, 1, file=out)
    print(' rad_conf', file=out)
    print(' ', end='', file=out)
    for i in range(n_cells_y):
        for j in range(n_cells_x):
            print(rad_conf[i][j], end=' ', file=out)
        print('\n ', end='', file=out)


    # x Grid
    print('grid_x ', end='', file=out)
    for i, pitch in enumerate(cell_pitch_x_orig):
        print(str(refs_x)+'*'+str(pitch/refs_x), end=' ', file=out)
    print('', file=out)

    print(' neutmesh_x ', end='', file=out)
    for i in range(n_cells_x):
        print(neut_x, end=' ', file=out)
    print('', file=out)

    # Grid y
    print(' grid_y ', end='', file=out)
    for i, pitch in enumerate(cell_pitch_y_orig):
        print(str(refs_y)+'*'+str(pitch/refs_y), end=' ', file=out)
    print('', file=out)

    print(' neutmesh_y ', end='', file=out)
    for i in range(n_cells_y):
        print(neut_y, end=' ', file=out)
    print('', file=out)

    print(' boun_cond  ' + bound_coond,  file=out)
    print(' grid_z     10000',  file=out)
    print(' Planar_Reg 1',  file=out)
    print(' ', end='', file=out)
    assem = 0
    for i in range(n_cells_y):
        for j in range(n_cells_x):
            if (rad_conf[i][j] > 0):
                assem += 1
                print(assem, end=' ', file=out)
        print('\n ', end='', file=out)
    print('PR_Assign 1', file=out)

    out.close()


"=========================================================================="
def make_xs_file():
    """ Make the XS file """
    tol = 1e-12;
#    n_mats = len(xs)
```

```
    out=open(xsec_file, 'w')

    n_steps = int(t_end/ delta_t) + 1

    minp = [0.0, 0.0]
    maxp = [0.0, 0.0]
    vib_pos = vib_pos_0.copy()

    print('TIME_STEPS', n_steps, file=out)
    print('N_COMP', n_cells, file=out)
    for step in range(n_steps):
        time = delta_t * step;

        vib_pos[0] = vib_pos_0[0] + amplitude*np.sin(time*omega)
        vib_pos[1] = vib_pos_0[1] + amplitude*np.sin(time*omega)

        print('TIME', time, file=out)
        cell = -1
        for celly in range(n_cells_y):
            for cellx in range(n_cells_x):
                # There is no cell
                if rad_conf[celly][cellx] == 0:
                    continue;
                cell += 1

                minp[0] = cpos_min_x[cellx]
                maxp[0] = cpos_max_x[cellx]
                minp[1] = cpos_min_y[celly]
                maxp[1] = cpos_max_y[celly]

                print_cell(cell, vib_pos, minp, maxp,
                           vib_mat, tol, xs, materials, out)


    out.close()




make_geometry_file()
# Init and end of the cell
cpos_min_x = [ sum(cell_pitch_x[0:cell]) for cell in range(n_cells_x)]
cpos_max_x = [ sum(cell_pitch_x[0:cell+1]) for cell in range(n_cells_x)]
cpos_min_y = [ sum(cell_pitch_y[0:cell]) for cell in range(n_cells_y)]
cpos_max_y = [ sum(cell_pitch_y[0:cell+1]) for cell in range(n_cells_y)]

make_xs_file()
```

Table 14: Code to read all custom cross-sections into PARCS at file InpProcM.f90.

```
INTEGER(sik) :: nsteps, st
CHARACTER(len=100) :: xsec_filename
REAL(sdk) :: time
! --------------------------------------------------------------
 ! Read Custom XS
 CASE('FICHERO')

    READ(oneline,*) cardname,xsec_filename
    write(*,*)' LEYENDO XS DEL FICHERO... ', xsec_filename

    OPEN(89,file=xsec_filename)
    READ(89,*) , cardname, nsteps
    READ(89,*) , cardname, ncomp

    !CALL get_open_unit(89)
    !IF(popt(17))THEN
    !    WRITE(*,*)'N_STEPS', nsteps
    !    WRITE(*,*)'N_COMP ', ncomp
    !ENDIF
    DO st=1,nsteps
       READ(89,*) , cardname, time
       !IF(popt(17))THEN
       !   WRITE(*,'(A, F8.6)')'READ CROSS SECTIONS AT TIME = ',time
       !ENDIF

       timevec(st) = time
       DO icomp=1,ncomp
          READ(89,*) mat, (fdum(i),i=1,9)
          timesigtr(st,1,icomp)=fdum(1)
          timesiga(st,1,icomp)=fdum(2)
          timesignf(st,1,icomp)=fdum(3)
          timesigkf(st,1,icomp)=fdum(4)
          timesig12(st,icomp)=fdum(5)
          timesigtr(st,2,icomp)=fdum(6)
          timesiga(st,2,icomp)=fdum(7)
          timesignf(st,2,icomp)=fdum(8)
          timesigkf(st,2,icomp)=fdum(9)

       ENDDO
    ENDDO
    ! Declare initial XS
    DO icomp=1,ncomp
            sigtr(1,icomp)=timesigtr(1,1,icomp)
            siga(1,icomp)=timesiga(1,1,icomp)
            signf(1,icomp)=timesignf(1,1,icomp)
            sigkf(1,icomp)=timesigkf(1,1,icomp)
            sig12(icomp)=timesig12(1,icomp)
            sigs(1,2,icomp)=sig12(icomp)
            sigtr(2,icomp)=timesigtr(1,2,icomp)
            siga(2,icomp)=timesiga(1,2,icomp)
            signf(2,icomp)=timesignf(1,2,icomp)
            sigkf(2,icomp)=timesigkf(1,2,icomp)
            dif(1,icomp)=1/(sigtr(1,icomp)*3.0)
            dif(2,icomp)=1/(sigtr(2,icomp)*3.0)
```

```
        ENDDO
    CLOSE(89)
    ! ------------------------------------------------------------

    reigv=1/eigv
    WRITE(*,'(A)') '  Make Custom Cross Sections Critical'
        DO st=1,nsteps
           DO icomp=1,ncomp
                  timesignf(st,1,icomp)= timesignf(st,1,icomp) * reigv
                  timesigkf(st,1,icomp)= timesigkf(st,1,icomp) * reigv
                  timesignf(st,2,icomp)= timesignf(st,2,icomp) * reigv
                  timesigkf(st,2,icomp)= timesigkf(st,2,icomp) * reigv
           ENDDO
        ENDDO
```

Table 15: Code to update the cross-section each time step in PARCS at file XSecFdbkM.f90.

```
DO st=1,nsteps-1
  IF ((time .GT. timevec(st)) .AND. (time .LE. timevec(st+1))) THEN
    frac = (time - timevec(st)) / (timevec(st+1) - timevec(st))
    WRITE(*,’(A, F10.5)’)’Update XS at t= ’, time
    DO k=1,nz
      DO l=1,nxy
        icomp=xsid(l,k)
        xstr(1,l,k) = (1.0-frac) * timesigtr(st,1,icomp) + frac * timesigtr(st+1,1,icomp)
        xstr(2,l,k) = (1.0-frac) * timesigtr(st,2,icomp) + frac * timesigtr(st+1,2,icomp)

        xsd(1,l,k) = 1/(3.0 * xstr(1,l,k))
        xsd(2,l,k) = 1/(3.0 * xstr(2,l,k))
        xsa(1,l,k) = (1.0-frac) * timesiga(st,1,icomp) + frac * timesiga(st+1,1,icomp)
        xsa(2,l,k) = (1.0-frac) * timesiga(st,2,icomp) + frac * timesiga(st+1,2,icomp)
        xsnf(1,l,k)= ((1.0-frac) * timesignf(st,1,icomp) + frac * timesignf(st+1,1,icomp))
        xsnf(2,l,k)= ((1.0-frac) * timesignf(st,2,icomp) + frac * timesignf(st+1,2,icomp))

        xskf(1,l,k)= ((1.0-frac) * timesigkf(st,1,icomp)+ frac * timesigkf(st+1,1,icomp))
        xskf(2,l,k)= ((1.0-frac) * timesigkf(st,2,icomp)+ frac * timesigkf(st+1,2,icomp))

        xss(l,k) = (1.0-frac) * timesig12(st,icomp) + frac * timesig12(st+1,icomp)
        xst(1,l,k)= xsa(1,l,k) + xss(l,k)
        xst(2,l,k)= xsa(2,l,k)

      ENDDO
    ENDDO
  ENDIF
ENDDO
```

Table 16: Python script to postprocess time-domain results.

```
# -*- coding: utf-8 -*-
"""
Postprocess time-domain results from PARCS and FEMFFUSION
for the COLIBRI experiments.
@author: Antoni Vidal, anvifer2@upv.es
@institution: Universitat Politecnica de Valencia (UPV)
"""
```

```
#from utils import  parseFileSameLine,  parseFile
import pandas as pd
from utils import   parse_vtk_file, parse_vtk_grid, parse_file
import numpy as np
import matplotlib.pyplot as plt
import scipy.io as sio
from utils import get_CROCUS_detector_position
import matplotlib
import sys
plt.close('all')



#%%############################################################################

grid = 'local'  # 'local' or 'uniform
exp = 12
if (exp == 12):
    looking_freq = 0.1; # Hz

problem = '2D_CROCUS_exp_' + str(exp) + '_' + grid
folder = "../2D_CROCUS/exp"+ str(exp) + "_" + grid + "/"
tol = 1e-8


# FEMFFUSION DATA
file_out = folder + 'crocus_' + grid + '_0.out'
static_file = folder + 'crocus_'+ grid +'_0.out.vtk'
steps_fem = range(0, 300)
files =[]
for i, step in enumerate(steps_fem):
    files.append(file_out+ str(step) +  '.vtk')

# PARCS DATA
folder_parcs = "../2D_CROCUS/PARCS/"
parcs_outfile = folder_parcs + '2D_CROCUS_' + grid + '_'  +str(exp) + '.out'
delta_t_parcs = (1 / looking_freq) *  0.01
#%%############################################################################
color = ['#1f77b4', '#d62728', '#2ca02c', '#ff7f0e', '#ff9896',
         '#c5b0d5', '#8c564b', '#e377c2', '#f7b6d2', '#7f7f7f',
         '#c7c7c7', '#bcbd22', '#dbdb8d', '#17becf', '#9edae5']

def find_nearest(array, value):
    """Find the nearest value """
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return array[idx]

det_pos = get_CROCUS_detector_position()
n_detectors = len(det_pos)

#%%############################################################################
print('PARSE FEM .OUT FILES')

time_fem = parse_file(file_out, begin='Time vector', n_max_lines=1)
if (len(time_fem) == (len(steps_fem) + 1)):
    time_fem = time_fem[:-1]
x_fem = parse_vtk_grid(files[0])[0]
```

```
y_fem = parse_vtk_grid(files[0])[1]
n_nodes_fem = len(x_fem)
n_steps_fem = len(steps_fem)

det_pos_fem = dict()
for det_name in det_pos.keys():
    det_pos_fem[det_name] = [find_nearest(x_fem, det_pos[det_name][0]),
                             find_nearest(y_fem, det_pos[det_name][1])]

# Static Fluxes
static_flux_fs = parse_vtk_file(static_file, 'phi_g1_eig_1')
static_flux_th = parse_vtk_file(static_file, 'phi_g2_eig_1')
assert(len(static_flux_fs) == len(static_flux_th))
norm = max(static_flux_fs)
static_flux_fs = static_flux_fs / norm
static_flux_th = static_flux_th / norm

noise_fs = np.zeros([n_nodes_fem, n_steps_fem])
noise_th = np.zeros([n_nodes_fem, n_steps_fem])

for t, f in enumerate(files):
    ffs = parse_vtk_file(f, 'noise_g1')
    fth = parse_vtk_file(f, 'noise_g2')
    for node in range(n_nodes_fem):
        noise_fs[node][t] = ffs[node] / norm
        noise_th[node][t] = fth[node] / norm

for t, f in enumerate(files):
    if (t==0.0):
        continue
    for node in range(n_nodes_fem):
        noise_fs[node][t] -=  noise_fs[node][1]
        noise_th[node][t] -=  noise_th[node][1]

# GET RELATIVE NOISE AT DETECTOR
noise_fem_dect_fs = []
noise_fem_dect_th = []

for detector_name in det_pos_fem.keys():
    for node in range(n_nodes_fem):
        if (abs(x_fem[node] - det_pos_fem[detector_name][0]) < tol):
            if (abs(y_fem[node] - det_pos_fem[detector_name][1]) < tol):
                print("   FEMFFUSION", detector_name, "detected at node", node)
                noise_fem_dect_fs.append(noise_fs[node] / static_flux_fs[node])
                noise_fem_dect_th.append(noise_th[node] / static_flux_th[node])
                break


detector_data_fem = noise_fem_dect_th.copy()

# PLOT DETECTOR DATA
fig0 = plt.figure()
ax0 = fig0.add_subplot(1, 1, 1)
for i in range(len(det_pos_fem.keys())):
    ax0.plot(time_fem, detector_data_fem[i], c=color[i],
             label='Detector ' + str(i+1))

ax0.grid(True)
```

```
ax0.set_ylabel("Detector Output")
ax0.set_xlabel("Time (s)")
ax0.legend()
ax0.set_title( "Detector Data FEMFFUSSION")
fig0.savefig(problem + "_detector_fem.pdf", format='pdf', bbox_inches='tight')


#%%#############################################################################
# DETECTORS SIGNAL FFT FEM

delta_t_fem = time_fem[1]
# Compute Fast Fourier Transform
freq_fem = np.fft.rfftfreq(n_steps_fem, d=delta_t_fem)
detector_data_fem_fft_all = np.fft.rfft(detector_data_fem) * 2.0 / n_steps_fem

# We cut at looking_freq Hz

cut_freq = int (looking_freq * n_steps_fem * delta_t_fem)
assert(abs(freq_fem[cut_freq] - looking_freq) < 1e-4)

detector_data_fem_fft = np.zeros([n_detectors], dtype='cfloat')
for dtc in range(n_detectors):
    detector_data_fem_fft[dtc] = detector_data_fem_fft_all[dtc][cut_freq]

# RELATIVE CPSD WITH RESPECT TO DETECTOR 5
cpsd_fem_no_order= detector_data_fem_fft[4].conjugate() * detector_data_fem_fft

det_name = ['5_8', '5_6', '5_7', '5_10', '5_3', '5_9', '5_4', '5_5']
order = np.array([8, 6, 7, 10, 3, 9, 4, 5]) - 1
cpsd_fem = np.zeros(len(order), dtype='cfloat')
for i, o in enumerate(order):
    cpsd_fem[i] = cpsd_fem_no_order[o]

# Normalize to detector 5_6
cpsd_fem /= abs(cpsd_fem[1])

variables =  {
        'time': time_fem,
        'detector_data': detector_data_fem,
        'det_name': np.array(det_name),
        'cpsd': cpsd_fem
        }
sio.savemat(problem +'_FEMFFUSION.mat', variables)


#%%#############################################################################
print('PARSE PARCS .OUT FILES')
from utils import parse_fluxes_parcs_2D, parse_parcs_geometry

fl_parcs_fs, fl_parcs_th = parse_fluxes_parcs_2D(parcs_outfile)
x_parcs, y_parcs = parse_parcs_geometry(parcs_outfile)

# Remove last time step
fl_parcs_fs = fl_parcs_fs[0:-1]
fl_parcs_th = fl_parcs_th[0:-1]

n_steps_parcs = len(fl_parcs_th)
n_nodes_y = len(fl_parcs_th[0])
```

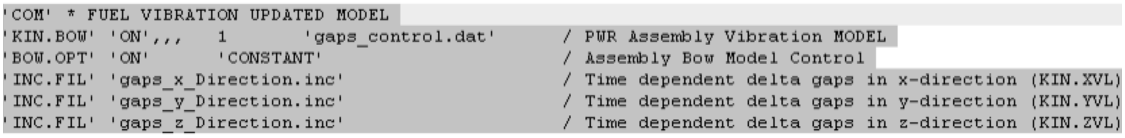
```
n_nodes_x = len(fl_parcs_th[0][0])

time_parcs = delta_t_parcs *  np.array(range(n_steps_parcs))

# Get Normalized static flux
static_flux_parcs_fs =  fl_parcs_fs[0].copy()
static_flux_parcs_th =  fl_parcs_th[0].copy()

#norm = np.max(static_flux_parcs_fs)

# Compute time and space dependent neutron noise
for st in range(n_steps_parcs) :
    for ny in range(n_nodes_y):
        for nx in range(n_nodes_x):
            fl_parcs_fs[st][ny][nx] -= static_flux_parcs_fs[ny][nx]
            fl_parcs_th[st][ny][nx] -= static_flux_parcs_th[ny][nx]

# Also, set the noise in time step 1 olso to 0
for ny in range(n_nodes_y):
    for nx in range(n_nodes_x):
        noise_fs_st1 = fl_parcs_fs[1][ny][nx]
        noise_th_st1 = fl_parcs_th[1][ny][nx]
        for st in range(1, n_steps_parcs) :
            fl_parcs_fs[st][ny][nx] -= noise_fs_st1
            fl_parcs_th[st][ny][nx] -= noise_th_st1

# Set a new detector position to the nearest mesh point
det_pos_fem = dict()
for det_name in det_pos.keys():
    det_pos_fem[det_name] = [find_nearest(x_parcs, det_pos[det_name][0]),
                             find_nearest(y_parcs, det_pos[det_name][1])]

# Get noise at detectors
noise_parcs_dect_fs = []
noise_parcs_dect_th = []
det  = 0
for detector_name in det_pos_fem.keys():
    for ny in range(n_nodes_y):
        if (abs(y_parcs[ny] - det_pos_fem[detector_name][1]) < tol):
            for nx in range(n_nodes_x):
                if (abs(x_parcs[nx] - det_pos_fem[detector_name][0]) < tol):
                    print("PARCS", detector_name,
                          "detected at (" + str(nx) + ", " + str(ny) + ")")
                    noise_parcs_dect_fs.append(np.zeros(n_steps_parcs))
                    noise_parcs_dect_th.append(np.zeros(n_steps_parcs))

                    for st in range(n_steps_parcs) : # Relative nosie
                        noise_parcs_dect_fs[det][st] = fl_parcs_fs[st][ny][nx]
                            / static_flux_parcs_fs[ny][nx]
                        noise_parcs_dect_th[det][st] = fl_parcs_th[st][ny][nx]
                            / static_flux_parcs_th[ny][nx]
    det += 1

# The detector read the thermal flux
detector_data_parcs = noise_parcs_dect_th.copy()

# Plot the detector data
fig0 = plt.figure()
```

```
ax0 = fig0.add_subplot(1, 1, 1)
for i in range(len(det_pos_fem.keys())):
    ax0.plot(time_parcs, detector_data_parcs[i], c=color[i],
             label='Detector ' + str(i+1))
ax0.grid(True)
ax0.set_ylabel("Detector Output")
ax0.set_xlabel("Time (s)")
ax0.set_title( "Detector Data PARCS")
ax0.legend()
fig0.savefig(problem + "_detector_parcs.pdf", format='pdf')

#%%###############################################################################
# DETECTORS SIGNAL FFT PARCS

freq = np.fft.rfftfreq(n_steps_parcs, d=delta_t_parcs)
detector_data_parcs_fft_all = np.fft.rfft(detector_data_parcs)
 * 2.0 / n_steps_parcs

# We cut at looking_freq Hz
cut_freq = int (looking_freq * n_steps_parcs * delta_t_parcs)
assert(abs(freq[cut_freq] - looking_freq) < 1e-4)

detector_data_parcs_fft = np.zeros([n_detectors], dtype='cfloat')
for dtc in range(n_detectors):
    detector_data_parcs_fft[dtc] = detector_data_parcs_fft_all[dtc][cut_freq]

# RELATIVE CPSD WITH RESPECT TO DETECTOR 5
cpsd_parcs_no_order = detector_data_parcs_fft[4].conjugate()
                     * detector_data_parcs_fft

det_name = ['5_8', '5_6', '5_7', '5_10', '5_3', '5_9', '5_4', '5_5']
order = np.array([8, 6, 7, 10, 3, 9, 4, 5]) - 1
cpsd_parcs = np.zeros(len(order), dtype='cfloat')
for i, o in enumerate(order):
    cpsd_parcs[i] = cpsd_parcs_no_order[o]

# Normalize to detector 5_6
cpsd_parcs /= abs(cpsd_parcs[1])

variables =  {
        'time': time_fem,
        'detector_data': detector_data_parcs,
        'det_name': np.array(det_name),
        'cpsd': cpsd_parcs
        }
sio.savemat(problem +'_PARCS.mat', variables)
```

9.2 Delta gap widths preparation for SIMULATE-3K using MATLAB

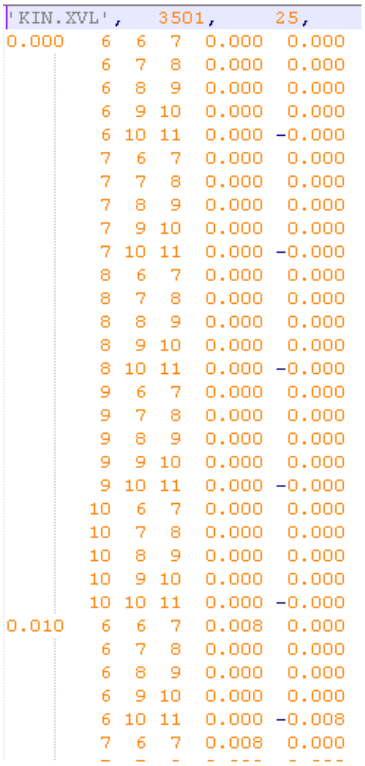
For simulating vibrations of fuel assemblies using SIMULATE-3K, the assembly vibration model is introduced in the input deck in a way represented in Figure 84. The aim of the MATLAB support scripts is to generate the include files titled “gaps_x_Direction.inc”, “gaps_y_Direction.inc”, and “gaps_z_Direction.inc”, containing the time-dependent delta-gap widths in x-, y- and z-direction, respectively, depending on the chosen scenario.

An example of the “gaps_x_Direction.inc” for a 5x5 central fuel assembly cluster vibrating sinusoidally at 1.2 Hz in x-direction in a synchronized manner is shown in Figure 85.



```
'COM' * FUEL VIBRATION UPDATED MODEL
'KIN.BOW' 'ON',,,   1       'gaps_control.dat'      / PWR Assembly Vibration MODEL
'BOW.OPT' 'ON'       'CONSTANT'                     / Assembly Bow Model Control
'INC.FIL' 'gaps_x_Direction.inc'                    / Time dependent delta gaps in x-direction (KIN.XVL)
'INC.FIL' 'gaps_y_Direction.inc'                    / Time dependent delta gaps in y-direction (KIN.YVL)
'INC.FIL' 'gaps_z_Direction.inc'                    / Time dependent delta gaps in z-direction (KIN.ZVL)
```

Figure 84 Assembly vibration model

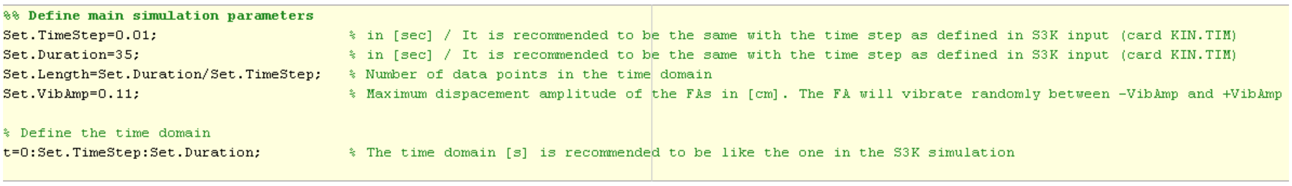


```
'KIN.XVL',    3501,      25,
0.000     6  6  7  0.000  0.000
          6  7  8  0.000  0.000
          6  8  9  0.000  0.000
          6  9 10  0.000  0.000
          6 10 11  0.000 -0.000
          7  6  7  0.000  0.000
          7  7  8  0.000  0.000
          7  8  9  0.000  0.000
          7  9 10  0.000  0.000
          7 10 11  0.000 -0.000
          8  6  7  0.000  0.000
          8  7  8  0.000  0.000
          8  8  9  0.000  0.000
          8  9 10  0.000  0.000
          8 10 11  0.000 -0.000
          9  6  7  0.000  0.000
          9  7  8  0.000  0.000
          9  8  9  0.000  0.000
          9  9 10  0.000  0.000
          9 10 11  0.000 -0.000
         10  6  7  0.000  0.000
         10  7  8  0.000  0.000
         10  8  9  0.000  0.000
         10  9 10  0.000  0.000
         10 10 11  0.000 -0.000
0.010     6  6  7  0.008  0.000
          6  7  8  0.000  0.000
          6  8  9  0.000  0.000
          6  9 10  0.000  0.000
          6 10 11  0.000 -0.008
          7  6  7  0.008  0.000
```

Figure 85 Example of the KIN.XVL input card in SIMULATE-3K for describing the random vibration of a single bundle in the x-direction

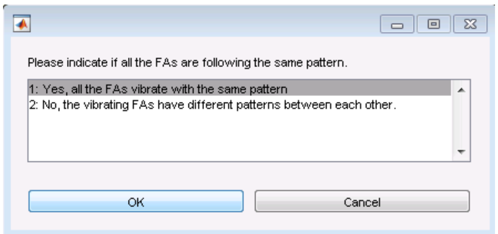
The various steps to generate the include file in the Figure 85 are presented here along with screenshots of the graphical user interface of the MATLAB support scripts. Based on Figure 59, the steps are:

Step 1: Define the simulation parameters such as simulation duration, time step of the transient simulation, and maximum displacement amplitude of the fuel assemblies.



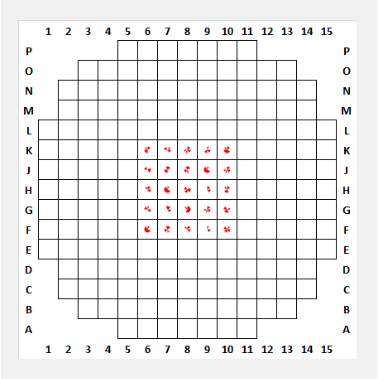
```
%% Define main simulation parameters
Set.TimeStep=0.01;                    % in [sec] / It is recommended to be the same with the time step as defined in S3K input (card KIN.TIM)
Set.Duration=35;                      % in [sec] / It is recommended to be the same with the time step as defined in S3K input (card KIN.TIM)
Set.Length=Set.Duration/Set.TimeStep; % Number of data points in the time domain
Set.VibAmp=0.11;                      % Maximum dispacement amplitude of the FAs in [cm]. The FA will vibrate randomly between -VibAmp and +VibAmp

% Define the time domain
t=0:Set.TimeStep:Set.Duration;        % The time domain [s] is recommended to be like the one in the S3K simulation
```

Step 2: Choose if all the vibrating assemblies follow the same pattern or not. Enter the total number of vibrating fuel assemblies and select their locations.

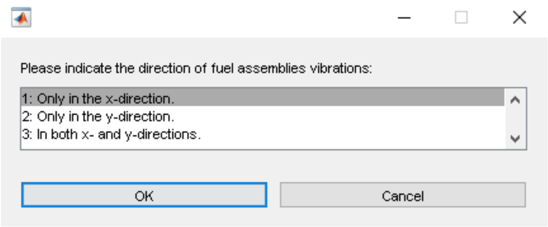


Please indicate if all the FAs are following the same pattern.

1: Yes, all the FAs vibrate with the same pattern
2: No, the vibrating FAs have different patterns between each other.



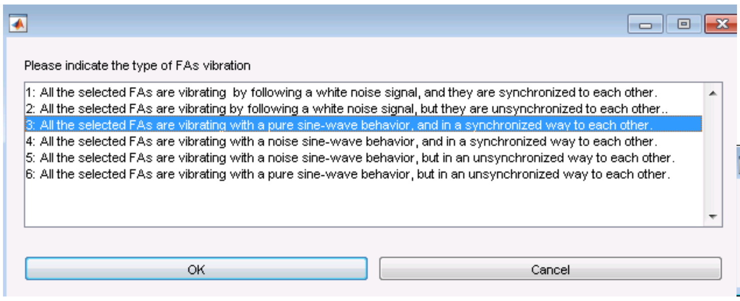
Step 3: Enter the direction of vibration of fuel assemblies.



Please indicate the direction of fuel assemblies vibrations:

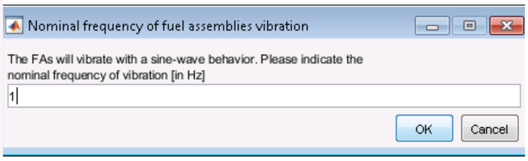
1: Only in the x-direction.
2: Only in the y-direction.
3: In both x- and y-directions.

Step 4-5: Choose the pattern of vibration, such as random or sinusoidal and synchronized or unsynchronized.



Please indicate the type of FAs vibration

1: All the selected FAs are vibrating by following a white noise signal, and they are synchronized to each other.
2: All the selected FAs are vibrating by following a white noise signal, but they are unsynchronized to each other..
3: All the selected FAs are vibrating with a pure sine-wave behavior, and in a synchronized way to each other.
4: All the selected FAs are vibrating with a noise sine-wave behavior, and in a synchronized way to each other.
5: All the selected FAs are vibrating with a noise sine-wave behavior, but in an unsynchronized way to each other.
6: All the selected FAs are vibrating with a pure sine-wave behavior, but in an unsynchronized way to each other.



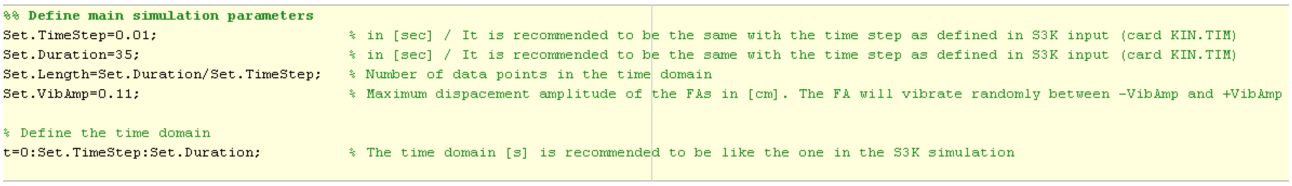
Nominal frequency of fuel assemblies vibration

The FAs will vibrate with a sine-wave behavior. Please indicate the nominal frequency of vibration [in Hz]

1

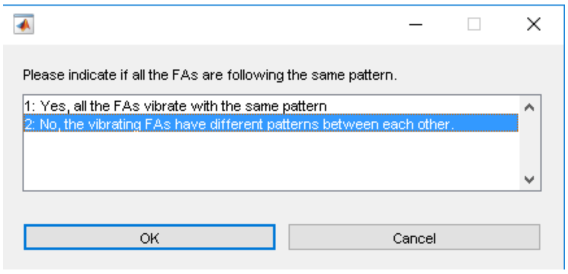
Additionally, there is a possibility of simulating complex scenarios where separate clusters of fuel assemblies and simultaneously vibrating with different vibrational patterns. In that case, the illustrative steps are shown here:

Step 1: Define the simulation parameters such as simulation duration, time step of the transient simulation, and maximum displacement amplitude of the fuel assemblies.



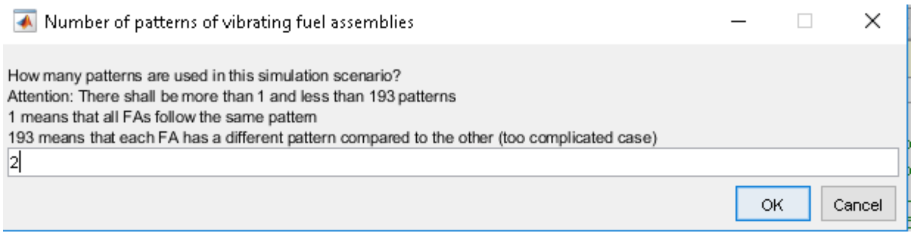
```
%% Define main simulation parameters
Set.TimeStep=0.01;                        % in [sec] / It is recommended to be the same with the time step as defined in S3K input (card KIN.TIM)
Set.Duration=35;                          % in [sec] / It is recommended to be the same with the time step as defined in S3K input (card KIN.TIM)
Set.Length=Set.Duration/Set.TimeStep;     % Number of data points in the time domain
Set.VibAmp=0.11;                          % Maximum dispacement amplitude of the FAs in [cm]. The FA will vibrate randomly between -VibAmp and +VibAmp

% Define the time domain
t=0:Set.TimeStep:Set.Duration;            % The time domain [s] is recommended to be like the one in the S3K simulation
```

Step 2: Choose if all the vibrating assemblies follow the same pattern or not. Enter the total number of patterns of vibrating fuel assemblies, total number of vibrating assemblies in each pattern and select their locations.



Please indicate if all the FAs are following the same pattern.

1: Yes, all the FAs vibrate with the same pattern
2: No, the vibrating FAs have different patterns between each other.



Number of patterns of vibrating fuel assemblies

How many patterns are used in this simulation scenario?
Attention: There shall be more than 1 and less than 193 patterns
1 means that all FAs follow the same pattern
193 means that each FA has a different pattern compared to the other (too complicated case)

2

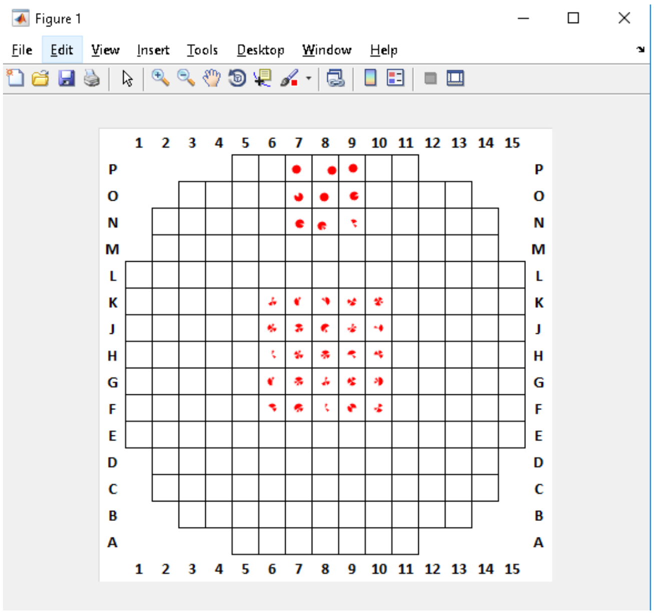
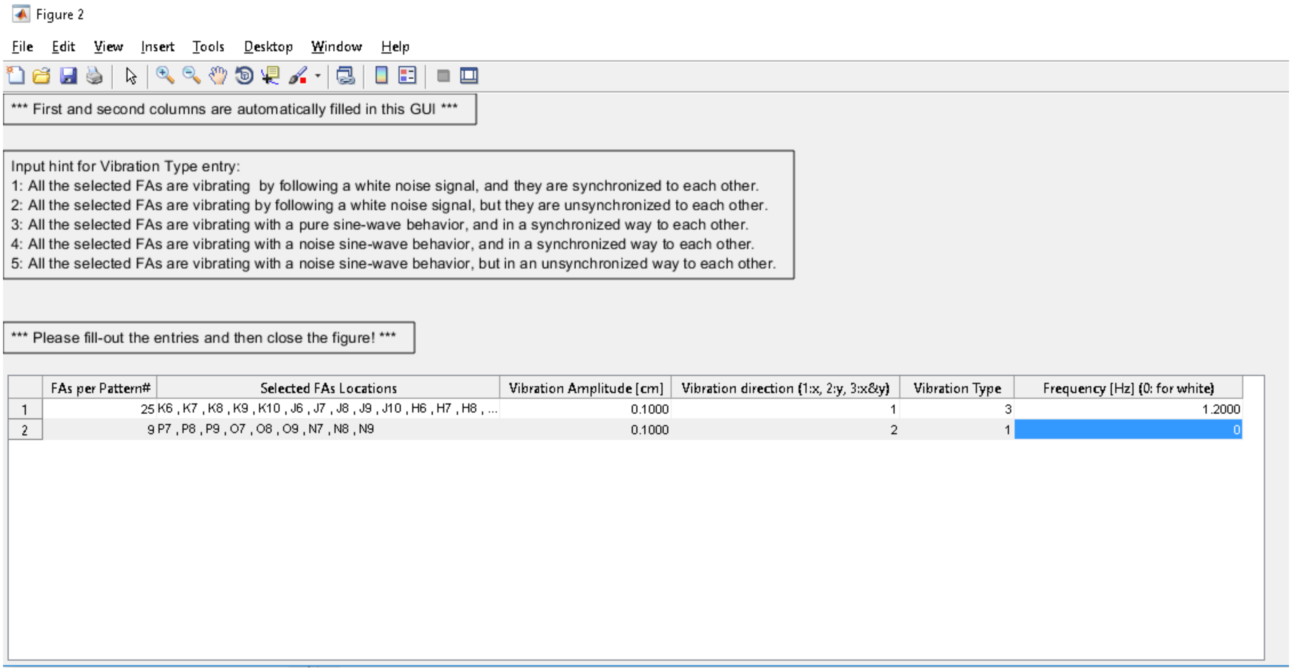


Figure 1
File Edit View Insert Tools Desktop Window Help
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
P O N M L K J H G F E D C B A
P O N M L K J H G F E D C B A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Step 3: Enter the vibrational characteristics, i.e. direction of vibration, vibrational amplitude, vibration type, and frequency for the two chosen patterns.



*** First and second columns are automatically filled in this GUI ***

Input hint for Vibration Type entry:
1: All the selected FAs are vibrating by following a white noise signal, and they are synchronized to each other.
2: All the selected FAs are vibrating by following a white noise signal, but they are unsynchronized to each other.
3: All the selected FAs are vibrating with a pure sine-wave behavior, and in a synchronized way to each other.
4: All the selected FAs are vibrating with a noise sine-wave behavior, and in a synchronized way to each other.
5: All the selected FAs are vibrating with a noise sine-wave behavior, but in an unsynchronized way to each other.

*** Please fill-out the entries and then close the figure! ***

	FAs per Pattern#	Selected FAs Locations	Vibration Amplitude [cm]	Vibration direction (1:x, 2:y, 3:x&y)	Vibration Type	Frequency [Hz] (0: for white)
1	25	K6 , K7 , K8 , K9 , K10 , J6 , J7 , J8 , J9 , J10 , H6 , H7 , H8 , ...	0.1000	1	3	1.2000
2	9	P7 , P8 , P9 , O7 , O8 , O9 , N7 , N8 , N9	0.1000	2	1	0