CTH-NT-300 August 2014

DESCRIPTION OF THE MODELS AND ALGORITHMS USED IN THE COUPLED CORE SIM NEUTRONIC AND THERMO-HYDRAULIC TOOL

VICTOR DYKIN AND CHRISTOPHE DEMAZIÈRE

Division of Nuclear Engineering
Department of Applied Physics
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden 2014

ABSTRACT
The development of an innovative coupled neutronic/thermo-hydraulic tool is reported hereafter. The novelty of the tool resides in its versatility, since many different systems can be investigated and different kinds of calculations can be performed. More precisely, both critical systems and subcritical systems with an external neutron source can be studied, static and dynamic cases in the frequency domain (i.e. for stationary fluctuations) can be considered. For each situation, the three dimensional distributions of static neutron fluxes, all thermo-hydraulic parameters, their respective first-order noise are estimated, as well as the effective multiplication factor of the system. The main advantages of the tool, which is entirely *MATLAB* based, lie with the robustness of the implemented numerical algorithms, its high portability between different computer platforms and operative systems, and finally its ease of use since no input deck writing is required. The present version of the tool, which is based on two-group diffusion theory, is mostly suited to investigate thermal systems, both Pressurized and Boiling Water Reactors (PWR and BWR, respectively). This report describes the neutronic and thermo-hydraulic models, their coupling and numerical algorithms implemented in the tool, whereas the demonstration of the tool is reported in a companion report [1]. The tool, for which a complete user's manual exists [2], is freely available on direct request to the authors of the present report.

Contents

Chapter 1

INTRODUCTION

The deterministic modelling of Light Water Reactors (LWRs) requires special techniques due to the multi-physics and multi-scale aspects of such systems. The fact that water is used both as a neutron moderator and a coolant makes the modelling of such system particularly challenging.

The multi-physics aspects come primarily from the macroscopic cross-sections being dependent on the temperature of the coolant/moderator and of the fuel. As a result, the neutron density field (i.e. the spatial and temporal distribution of the neutron density throughout the core) can only be determined if the density field of the coolant and the temperature field of the fuel (i.e. the spatial and temporal distribution of the enthalpy of the coolant/temperature of the fuel throughout the core) are known. Likewise, the density field of the coolant/temperature field of the fuel can only be determined if the neutron density field is known, since the latter is directly related to the heat production in the nuclear fuel pins. Consequently, the determination of all fields needs to be carried out simultaneously if one wants to determine the behavior of LWRs. The determination of the neutron field is usually referred to as reactor physics or neutronic calculations, whereas the determination of the density/velocity/enthalpy fields of the coolant/temperature field of the fuel is usually referred to as thermal-hydraulic calculations. The strong coupling between the neutron physics and thermal-hydraulics is a unique feature of LWRs, which makes the calculation of their behavior difficult.

The other characteristic feature of LWRs is their multi-scale aspects, i.e. phenomena occurring at different scales. The multi-scale character of LWRs is explained by the fact that nuclear reactors are strongly heterogeneous systems, and by the fact that phenomena involving different characteristic lengths play role in the system.

Using general purpose multi-physics tools for modelling LWRs usually does not allow properly capturing the multi-scale aspects of nuclear reactors. The common deterministic modelling strategies all rely on separate modelling tools for resolving the different fields (the neutron density, the temperature of the fuel, and the density/velocity/enthalpy of the coolant) and possibly the different scales. The interdependence between the different fields/scales is usually accounted for by codes being run sequentially and/or by software coupling. Although procedures have been developed for making deterministic modelling possible at a high level of details and sophistication, such procedures require the use of complex codes and the development of complicated input decks. The lack of transparency and the complexity of these procedures makes it difficult to check the validity of the obtained results and to get insight into the physical phenomena occurring in

a given scenario. Complementing such state-of-the-art modelling techniques by simpler computational tools that can still catch the main physical phenomena and at the same time provide some physical insight is the purpose of the work reported in this report.

The present tool is a coupled neutronic/thermo-hydraulic tool consisting of two separate modules: neutronic and thermo-hydraulic, respectively. The first one is based on two-group diffusion equations and uses sets of macroscopic cross-sections as input parameters for calculating the three-dimensional spatial distributions of the neutron fluxes and thus of the generated power. The second one includes mass, momentum and energy balance equations and uses the reactor power as an input parameter to calculate the spatial distributions of all thermo-hydraulic parameters. The coupling between the two modules is performed by updating the macroscopic cross-sections. The corresponding dynamic calculations are performed in the frequency domain by using first order perturbation theory. In the present tool, the user can freely define configurations representative of any actual core and most importantly can perturb the system by directly defining perturbations in the thermo-hydraulic quantities (core inlet velocity, core inlet temperature or core exit pressure) as well as in the macroscopic cross-sections. Some of the examples of the perturbations which can be modelled with this tool are density wave oscillations, core barrel vibrations and fuel assembly vibrations, among others.

In addition to the flexibility in the definition of the problems to be investigated by the tool, another main novelty is its multi-purpose character. Namely, the tool can consider both critical systems and subcritical systems with an external neutron source, static cases and dynamic cases in the frequency domain (i.e. for stationary fluctuations). For each situation, the three-dimensional spatial distributions of static neutron fluxes, all thermo-hydraulic parameters, their respective first-order noise are estimated, as well as the effective multiplication factor of the system.

The spatial discretisation of the governing equations is based on finite differences. The coding was implemented in *MATLAB*, which makes the pre- and post-processing of data easy, as well as the code highly portable between different operative systems and computer platforms. Two types of neutronic equations are solved within the tool: homogeneous or eigenvalue equations, and non-homogeneous or source equations. For the former, the explicitly-restarted Arnoldi method was implemented. In case of convergence problem, the user has also the possibility of choosing the power iteration method, which was implemented using Wielandt's shift technique. The initial guess of the eigenvalues required for the application of Wielandt's shift technique is provided by an Arnoldi run without restart. For the latter, Gaussian elimination is used. In all cases, the matrix operations are based on a LU factorization of the matrices with full pivoting, in order to preserve the sparsity of the matrices. The thermo-hydraulic model is based on three equation homogeneous equilibrium model (HEM) including mass, momentum and energy (enthalpy) mixture conservation equations which are solved in an iterative manner. In order to improve the accuracy of HEM in void fraction and mixture density calculations, the tool is complemented with a simple slip ratio model. The heat transfer is modelled via heat balance equation written for a homogenized fuel assembly. The coupling between the neutronic and thermo-hydraulic modules is performed through the corresponding cross-section model (linear or tabulated). Although the accuracy of this tool cannot be compared to commercial core simulators, the tool offers several advantages such as: its ease of use, the robustness of the algorithms, and the fact that non-conventional systems can be easily investigated. Another main strength of the tool is that no input deck writing

is required.

The structure of the report is as follows. First, the neutronic models on which the tool relies are presented, both for the static and dynamic cases. Thereafter, the numerical algorithms used for the spatial discretisation of the implemented equations, as well as for solving the different types of equations, are touched upon. Next, the thermo-hydraulic models implemented in the tool, both for the static and dynamic cases, are described followed by the description of the numerical algorithms utilized for the spatial discretisation and solving of the corresponding equations. Further, the static and dynamic heat-transfer models are introduced together with the description of the numerical algorithms utilized for the spatial discretisation and solving of the corresponding equations. Finally, the calculation methodology used for solving a coupled system in both static and dynamic cases is discussed.

Chapter 2

NEUTRONIC MODELS

2.1 Introduction

In this section, the neutronic models implemented in the tool are presented. The neutronic part of the tool has the ability to calculate the solution to static problems with or without any external neutron source, as well as the solution to dynamic problems in linear theory and in the frequency domain. The equations thus solved in these different cases are presented.

Further, the numerical algorithms developed and utilized in the neutronic module of the tool are touched upon. First, the numerical scheme used for spatially discretising the neutronic equations is described. Thereafter, the numerical algorithms allowing solving the different types of equations (i.e. non-homogeneous equations and eigenvalue equations, respectively) are reported in detail.

2.2 Derivation of the static neutronic equations

The neutronic tool is based on diffusion theory with two energy groups and one group of delayed neutrons. In this formalism, the time- and space-dependent fast neutron flux, thermal neutron flux, and precursor density, can be expressed, respectively, as:

$$\frac{1}{v_1}\frac{\partial}{\partial t}\phi_1(\boldsymbol{r},t) = \nabla\cdot\left[D_{1,0}(\boldsymbol{r})\nabla\phi_1(\boldsymbol{r},t)\right] + \left[(1-\beta)\nu\Sigma_{f,1}(\boldsymbol{r},t) - \Sigma_{a,1}(\boldsymbol{r},t) - \Sigma_r(\boldsymbol{r},t)\right]\phi_1(\boldsymbol{r},t) +$$

$$(1-\beta)\nu\Sigma_{f,2}(\boldsymbol{r},t)\phi_2(\boldsymbol{r},t) + \lambda C(\boldsymbol{r},t) + S_1(\boldsymbol{r},t), \tag{2.1}$$

$$\frac{1}{v_2}\frac{\partial}{\partial t}\phi_2(\boldsymbol{r},t) = \nabla\cdot\left[D_{2,0}(\boldsymbol{r})\nabla\phi_2(\boldsymbol{r},t)\right] + \Sigma_r(\boldsymbol{r},t)\phi_1(\boldsymbol{r},t) - \Sigma_{a,2}(\boldsymbol{r},t)\phi_2(\boldsymbol{r},t) + S_2(\boldsymbol{r},t), \tag{2.2}$$

$$\frac{\partial}{\partial t}C(\boldsymbol{r},t) = \beta\nu\Sigma_{f,1}(\boldsymbol{r},t)\phi_1(\boldsymbol{r},t) + \beta\nu\Sigma_{f,2}(\boldsymbol{r},t)\phi_2(\boldsymbol{r},t) - \lambda C(\boldsymbol{r},t) \tag{2.3}$$

and where the macroscopic removal cross-section is defined as:

$$\Sigma_r(\boldsymbol{r},t) = \Sigma_{s0,1\to2}(\boldsymbol{r},t) - \frac{\Sigma_{s0,2\to1}(\boldsymbol{r},t)\phi_2(\boldsymbol{r},t)}{\phi_1(\boldsymbol{r},t)}. \tag{2.4}$$

In the previous equations, all the symbols have their usual meaning. The equations were obtained by assuming that both the prompt and delayed neutrons only contribute to the

fast energy group. $S_1(\boldsymbol{r},t)$ and $S_2(\boldsymbol{r},t)$ represent possible external neutron sources in the fast and thermal groups, respectively, and as such, the tool has the ability to model both critical systems, for which $S_1(\boldsymbol{r},t) = 0, \forall\,(\boldsymbol{r},t)$ and $S_2(\boldsymbol{r},t) = 0, \forall(\boldsymbol{r},t)$, or subcritical systems with external sources. All the macroscopic cross-sections and possible external neutron sources might be time-dependent. It was earlier demonstrated in [3] that allowing the diffusion coefficients to be time-dependent lead to dynamical results essentially identical to keeping such diffusion coefficients time-independent. Since the computational burden introduced by letting the diffusion coefficients vary with time increases drastically, the diffusion coefficients are kept time-independent in the tool reported hereafter.

If the system contains an external neutron source (case of a subcritical system driven by an external neutron source), Eqs. (2.1) - (2.3) written in steady-state conditions reduce to the following matrix equation:

$$[\nabla \cdot \overline{\overline{D}}(\boldsymbol{r})\nabla + \overline{\overline{\Sigma}}_{sta}(\boldsymbol{r}) - \overline{\overline{F}}(\boldsymbol{r})] \begin{bmatrix} \phi_{1,0}(\boldsymbol{r}) \\ \phi_{2,0}(\boldsymbol{r}) \end{bmatrix} = - \begin{bmatrix} S_{1,0}(\boldsymbol{r}) \\ S_{2,0}(\boldsymbol{r}) \end{bmatrix} \tag{2.5}$$

with

$$\overline{\overline{D}}(\boldsymbol{r}) = \begin{bmatrix} D_{1,0}(\boldsymbol{r}) & 0 \\ 0 & D_{2,0}(\boldsymbol{r}) \end{bmatrix}, \tag{2.6}$$

$$\overline{\overline{\Sigma}}_{sta}(\boldsymbol{r}) = \begin{bmatrix} -\Sigma_{a,1,0}(\boldsymbol{r}) - \Sigma_{r,0}(\boldsymbol{r}) & 0 \\ \Sigma_{r,0}(\boldsymbol{r}) & -\Sigma_{a,2,0}(\boldsymbol{r}) \end{bmatrix}, \tag{2.7}$$

$$\overline{\overline{F}}(\boldsymbol{r}) = \begin{bmatrix} -\nu\Sigma_{f,1,0}(\boldsymbol{r}) & -\nu\Sigma_{f,2,0}(\boldsymbol{r}) \\ 0 & 0 \end{bmatrix} \tag{2.8}$$

and where the subscript "0" refers to the static values of the different variables. From a mathematical viewpoint, the static problem of a subcritical source-driven system as given by Eq. (2.5) is represented by a non-homogeneous equation.

If the system does not contain any external neutron source, a steady-state solution to Eqs. (2.1) - (2.3) only exists if the system is critical. Eqs. (2.1) - (2.3) written in steady-state conditions then reduce to the following matrix equation:

$$[\nabla \cdot \overline{\overline{D}}(\boldsymbol{r})\nabla + \overline{\overline{\Sigma}}_{sta}(\boldsymbol{r})] \times \begin{bmatrix} \phi_{1,0}(\boldsymbol{r}) \\ \phi_{2,0}(\boldsymbol{r}) \end{bmatrix} = \overline{\overline{F}}(\boldsymbol{r}) \times \begin{bmatrix} \phi_{1,0}(\boldsymbol{r}) \\ \phi_{2,0}(\boldsymbol{r}) \end{bmatrix}. \tag{2.9}$$

If the system is not critical, a steady-state solution can still be obtained by re-normalizing the fission source terms by a factor k_m, and thus Eqs. (2.1) - (2.3) reduce to:

$$[\nabla \cdot \overline{\overline{D}}(\boldsymbol{r})\nabla + \overline{\overline{\Sigma}}_{sta}(\boldsymbol{r})] \times \begin{bmatrix} \phi_{1,m}(\boldsymbol{r}) \\ \phi_{2,m}(\boldsymbol{r}) \end{bmatrix} = \frac{1}{k_m}\overline{\overline{F}}(\boldsymbol{r}) \times \begin{bmatrix} \phi_{1,m}(\boldsymbol{r}) \\ \phi_{2,m}(\boldsymbol{r}) \end{bmatrix}. \tag{2.10}$$

From a mathematical viewpoint, the static problem without source as given by Eq. (2.10) corresponds to an eigenvalue problem, where both the eigenfunctions $\phi_{1,m}(\boldsymbol{r})$ and $\phi_{2,m}(\boldsymbol{r})$ and the corresponding eigenvalue $\frac{1}{k_m}$ have to be determined. There is an infinite number of solutions, i.e. an infinite number of pairs of solutions $[\phi_{1,m}(\boldsymbol{r})\ \phi_{2,m}(\boldsymbol{r})]$ and k_m, where the index "m" represents the mode number. The eigenfunctions having the same sign throughout the entire system correspond to the static fluxes denoted as $[\phi_{1,0}(\boldsymbol{r})\ \phi_{2,0}(\boldsymbol{r})]$ (fundamental mode) and the associated factor k_0 is then the effective multiplication factor of the system, i.e.

$$k_0 = k_{eff}. \tag{2.11}$$

All other eigenfunctions change sign throughout the system, and their associated factor k_m is strictly smaller than k_{eff}. It is customary to order the eigenmodes in increasing order of their eigenvalue $\frac{1}{k_m}$ (thus in decreasing order of the factor k_m):

$$k_{eff} = k_0 > k_1 > k_2 > ... > k_m. \tag{2.12}$$

It has to be noted that Eq. (2.9) is a sub-case of Eq. (2.10) obtained with $k_0 = k_{eff} = 1$. Depending on whether the system is subcritical with an external neutron source, or critical without neutron source, one notices that two types of equations need to be solved:

- an non-homogeneous type of equation for the subcritical system;
- an eigenvalue type equation for the critical system.

It has to be noted here that the eigenmodes can also be calculated for the subcritical system with an external neutron source. Nevertheless, the static flux is given by the fundamental mode only in the case of a critical system, since the static flux in the case of a subcritical system with an external neutron source is given by solving a non-homogeneous equation as represented by Eq. (2.5).

2.3 Derivation of the dynamic neutronic equations

In case of non-steady-state conditions, the time-dependent terms, generically expressed as $X(\boldsymbol{r},t)$, can be split into a mean value $X_0(\boldsymbol{r})$ (corresponding to the steady-state configuration of the system) and a fluctuating part $\delta X(\boldsymbol{r},t)$ around the mean value as:

$$X(\boldsymbol{r},t) = X_0(\boldsymbol{r}) + \delta X(\boldsymbol{r},t). \tag{2.13}$$

If the system is subcritical and driven by an external neutron source, using Eq. (2.13) for all time-dependent terms in Eqs. (2.1) - (2.3), removing the static equations [i.e. Eq. (2.5)], performing a temporal Fourier-transform, and neglecting second-order terms (linear theory), the following matrix equation is obtained:

$$[\nabla \cdot \overline{\overline{D}}(\boldsymbol{r})\nabla + \overline{\overline{\Sigma}}^{sub}_{dyn}(\boldsymbol{r},\omega)] \times \begin{bmatrix} \delta\phi_1(\boldsymbol{r},\omega) \\ \delta\phi_2(\boldsymbol{r},\omega) \end{bmatrix} = -\begin{bmatrix} \delta S_1(\boldsymbol{r},\omega) \\ \delta S_2(\boldsymbol{r},\omega) \end{bmatrix} + \overline{\phi}_r(\boldsymbol{r})\delta\Sigma_r(\boldsymbol{r},\omega) +$$

$$\overline{\overline{\phi}}_a(\boldsymbol{r}) \begin{bmatrix} \delta\Sigma_{a,1}(\boldsymbol{r},\omega) \\ \delta\Sigma_{a,2}(\boldsymbol{r},\omega) \end{bmatrix} + \overline{\overline{\phi}}^{sub}_f(\boldsymbol{r},\omega) \begin{bmatrix} \delta\nu\Sigma_{f,1}(\boldsymbol{r},\omega) \\ \delta\nu\Sigma_{f,2}(\boldsymbol{r},\omega) \end{bmatrix} \tag{2.14}$$

with

$$\overline{\overline{\Sigma}}^{sub}_{dyn}(\boldsymbol{r},\omega) = \begin{bmatrix} -\Sigma^{sub}_1(\boldsymbol{r},\omega) & \nu\Sigma_{f,2,0}(\boldsymbol{r})\left(1-\frac{i\omega\beta}{i\omega+\lambda}\right) \\ \Sigma_{r,0}(\boldsymbol{r}) & -\left(\Sigma_{a,2,0}(\boldsymbol{r})+\frac{i\omega}{v_2}\right) \end{bmatrix}, \tag{2.15}$$

$$\overline{\phi}_r(\boldsymbol{r}) = \begin{bmatrix} \phi_{1,0}(\boldsymbol{r}) \\ -\phi_{1,0}(\boldsymbol{r}) \end{bmatrix}, \tag{2.16}$$

$$\overline{\overline{\phi}}_a(\boldsymbol{r}) = \begin{bmatrix} \phi_{1,0}(\boldsymbol{r}) & 0 \\ 0 & \phi_{2,0}(\boldsymbol{r}) \end{bmatrix}, \tag{2.17}$$

$$\overline{\overline{\phi}}^{sub}_f(\boldsymbol{r},\omega) = \begin{bmatrix} -\phi_{1,0}(\boldsymbol{r})\left(1-\frac{i\omega\beta}{i\omega+\lambda}\right) & -\phi_{2,0}(\boldsymbol{r})\left(1-\frac{i\omega\beta}{i\omega+\lambda}\right) \\ 0 & 0 \end{bmatrix} \tag{2.18}$$

and with

$$\Sigma_1^{sub}(r,\omega) = \Sigma_{a,1,0}(\boldsymbol{r}) + \frac{i\omega}{v_1} + \Sigma_{r,0}(\boldsymbol{r}) - \nu\Sigma_{f,1,0}(\boldsymbol{r}) \times \left(1 - \frac{i\omega\beta}{i\omega+\lambda}\right), \tag{2.19}$$

and "i" standing for imaginary unit. The right-hand side of Eq. (2.14) gives the neutron noise source, resulting from either the fluctuations of the external neutron source around its mean value, or from the fluctuations of the macroscopic cross-sections (removal, absorption, and fission) around their mean value. Although the effect of the fluctuations of the macroscopic fission cross-sections and of the macroscopic absorption cross-sections are given by two separate terms in Eq. (2.14), any fluctuation in the macroscopic fission cross-section has also an impact on the macroscopic absorption cross-section (since fission is a special type of absorption).

If the neutron noise is induced by perturbations of the macroscopic cross-sections and if there is no external neutron source, then splitting the time-dependent parameters into mean values and fluctuations according to Eq. (2.13), removing the static equations [i.e. Eq. (2.10) taken with $n = 0$], performing a temporal Fourier-transform, and neglecting second-order terms (linear theory), the following matrix equation is obtained:

$$[\nabla \cdot \overline{\overline{D}}(\boldsymbol{r})\nabla + \overline{\overline{\Sigma}}_{dyn}^{crit}(\boldsymbol{r},\omega)] \times \begin{bmatrix} \delta\phi_1(\boldsymbol{r},\omega) \\ \delta\phi_2(\boldsymbol{r},\omega) \end{bmatrix} = \overline{\phi}_r(\boldsymbol{r})\delta\Sigma_r(\boldsymbol{r},\omega) + \overline{\overline{\phi}}_a(\boldsymbol{r}) \begin{bmatrix} \delta\Sigma_{a,1}(\boldsymbol{r},\omega) \\ \delta\Sigma_{a,2}(\boldsymbol{r},\omega) \end{bmatrix} + \overline{\overline{\phi}}_f^{crit}(\boldsymbol{r},\omega) \begin{bmatrix} \delta\nu\Sigma_{f,1}(\boldsymbol{r},\omega) \\ \delta\nu\Sigma_{f,2}(\boldsymbol{r},\omega) \end{bmatrix}. \tag{2.20}$$

When deriving this equation for the neutron noise, the system is assumed to be critical without source, since the system is supposed to be stationary. This means that Eq. (2.10) is assumed to be verified with $k_0 = k_{eff} = 1$ for $m = 0$. In reality, it is very unlikely that the eigenvalue of the first eigenmode is exactly equal to unity. Furthermore, Eq. (2.20) has to be spatially discretized (see Section 2.4.1), and such a spatial discretization might also lead to a discretized system deviating from criticality, even if the non-discretized system was exactly critical. One way to cope with this difficulty is to re-normalize the macroscopic fission cross-sections with k_{eff}, i.e. to replace in all equations $\upsilon\Sigma_{f,g}(\boldsymbol{r},t)$ by $\nu\Sigma_{f,g}(\boldsymbol{r},t)/k_{eff}$. This re-normalization guarantees that the discretized system is stationary. Therefore, the matrix $\overline{\overline{\Sigma}}_{dyn}^{crit}$ is defined as:

$$\overline{\overline{\Sigma}}_{dyn}^{crit}(\boldsymbol{r},\omega) = \begin{bmatrix} -\Sigma_1^{crit}(\boldsymbol{r},\omega) & \frac{\nu\Sigma_{f,2,0}(\boldsymbol{r})}{k_{eff}}\left(1 - \frac{i\omega\beta}{i\omega+\lambda}\right) \\ \Sigma_{r,0}(\boldsymbol{r}) & -\left(\Sigma_{a,2,0}(\boldsymbol{r}) + \frac{i\omega}{v_2}\right) \end{bmatrix} \tag{2.21}$$

with

$$\Sigma_1^{crit}(\boldsymbol{r},\omega) = \Sigma_{a,1,0}(\boldsymbol{r}) + \frac{i\omega}{v_1} + \Sigma_{r,0}(\boldsymbol{r}) - \frac{\nu\Sigma_{f,1,0}(\boldsymbol{r})}{k_{eff}}\left(1 - \frac{i\omega\beta}{i\omega+\lambda}\right) \tag{2.22}$$

and the matrix $\overline{\overline{\phi}}_f^{crit}$ is given as:

$$\overline{\overline{\phi}}_f^{crit}(\boldsymbol{r},\omega) = \begin{bmatrix} -\frac{\phi_{1,0}(\boldsymbol{r})}{k_{eff}}\left(1 - \frac{i\omega\beta}{i\omega+\lambda}\right) & -\frac{\phi_{2,0}(\boldsymbol{r})}{k_{eff}}\left(1 - \frac{i\omega\beta}{i\omega+\lambda}\right) \\ 0 & 0 \end{bmatrix}. \tag{2.23}$$

The expressions for $\overline{\phi}_r(\boldsymbol{r})$ and $\overline{\overline{\phi}}_a(\boldsymbol{r})$ are identical to the ones given by Eqs. (2.16) and (2.17), respectively. The right-hand side of Eq. (2.20) gives the neutron noise source, resulting from the fluctuations of the macroscopic cross-sections (removal, absorption, and

fission) around their mean value. Although the effect of the fluctuations of the macroscopic fission cross-sections and of the macroscopic absorption cross-sections are given by two separate terms in Eq. (2.20), any fluctuation in the macroscopic fission cross-section has also an impact on the macroscopic absorption cross-section (since fission is a special type of absorption).

2.4 Description of the numerical algorithms

2.4.1 Algorithm used for the spatial discretisation

In the developed tool, any three-dimensional system is assumed to be made of adjacent volumes or nodes. In a cartesian coordinate system, a given node n can be represented by a triplet of indexes (I, J, K), where the indexes I, J, and K refer to the $x-$, $y-$, and $z-$directions, respectively. The equations derived in Section 2.2 and Section 2.3 will be spatially-averaged on each of these nodes. For the sake of simplicity, the possible dependence on frequency of the different terms appearing in these equations is dropped. The different notations and conventions used throughout this section are highlighted in Figs. 2.1 and 2.2.

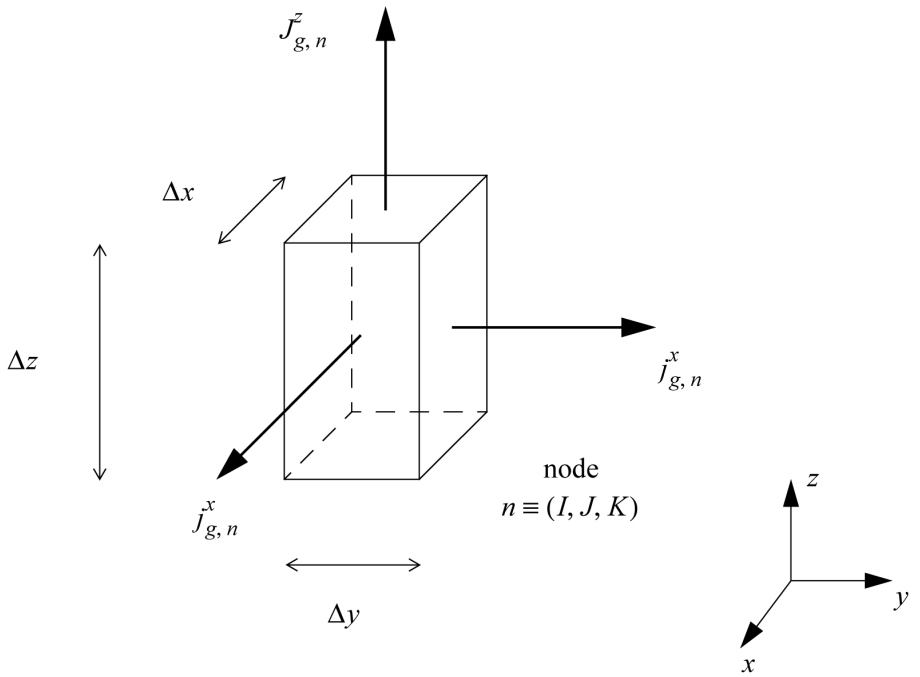

Figure 2.1: Principles and conventions used for the spatial discretisation of a neutronic node n.

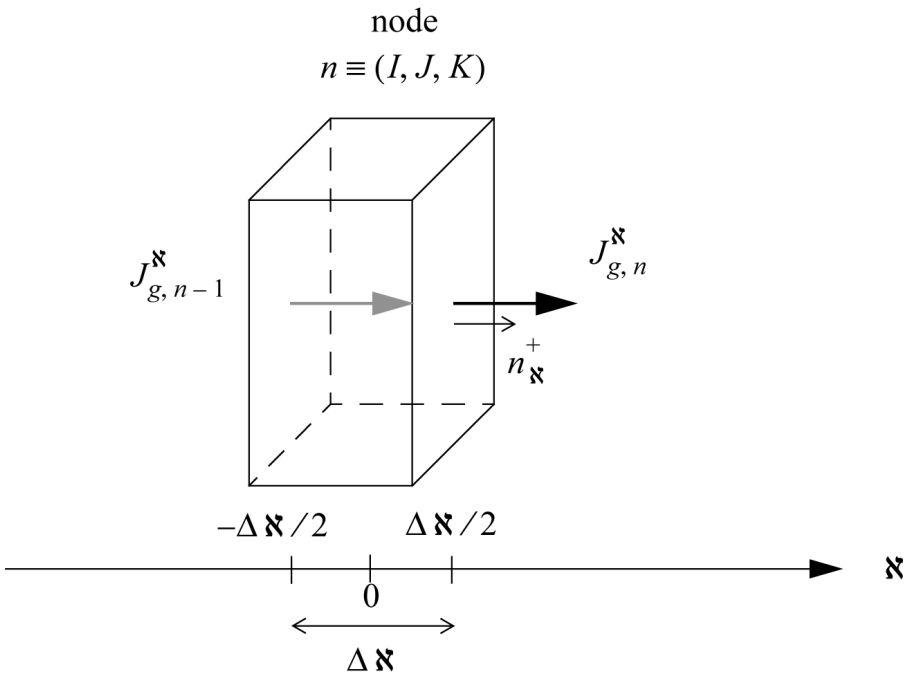

Figure 2.2: Generic notations relative to a neutronic node n used in the spatial discretisation along a direction א.

With Σ_g having the generic meaning of a macroscopic cross-section (either a static cross-section or its fluctuations), φ_g having the generic meaning of the scalar neutron flux (static flux or eigenmode or neutron noise), and s_g having the generic meaning of a neutron source (neutron source or its fluctuations), the following node-averaged quantities are defined:

$$\varphi_{g,n} = \frac{1}{V_n} \int_{V_n} \varphi_g(\boldsymbol{r}) d\boldsymbol{r}, \tag{2.24}$$

$$s_{g,n} = \frac{1}{V_n} \int_{V_n} s_g(\boldsymbol{r}) d\boldsymbol{r}, \tag{2.25}$$

$$\Sigma_{g,n} = \frac{\frac{1}{V_n} \int_{V_n} \Sigma_g(\boldsymbol{r}) \varphi_g(\boldsymbol{r}) d\boldsymbol{r}}{\varphi_{g,n}}, \tag{2.26}$$

where V_n represents the volume of the node n. This way of defining the node-averaged data, which preserves the actual reaction rates per node, is consistent with the group constants provided by any static core calculator.

When integrating each of the terms in the equations presented in Section 2.2 and Section 2.3 on a given node n, quantities of the form

$$\frac{1}{V_n} \int_{V_n} \Sigma_g(\boldsymbol{r}) \varphi_g(\boldsymbol{r}) d\boldsymbol{r} = \Sigma_{g,n} \varphi_{g,n} \tag{2.27}$$

and

$$\frac{1}{V_n} \int_{V_n} s_g(\boldsymbol{r}) d\boldsymbol{r} = s_{g,n}, \tag{2.28}$$

then appear, as well as quantities of the form

$$\frac{1}{V_n}\int_{V_n}\overline{\nabla}\cdot[D_{g,0}(\boldsymbol{r})\nabla\varphi_g(\boldsymbol{r})]d\boldsymbol{r}. \tag{2.29}$$

These last quantities cannot be directly expressed with the node-averaged quantities defined in Eqs. (2.27) and (2.29) due to the spatial operator $\overline{\nabla}$. Introducing the neutron current density vector $\boldsymbol{J}_g$ using Fick's law as:

$$\boldsymbol{J}_g(\boldsymbol{r}) = -D_{g,0}(\boldsymbol{r})\overline{\nabla}\varphi_g(\boldsymbol{r}), \tag{2.30}$$

where, as before, $\boldsymbol{J}_g$ has a generic meaning (neutron current corresponding to the static flux or eigenmode or neutron noise), one obtains, using Gauss' divergence theorem:

$$\frac{1}{V_n}\int_{V_n}\overline{\nabla}\cdot[D_{g,0}(\boldsymbol{r})\overline{\nabla}\varphi_g(\boldsymbol{r})]d\boldsymbol{r} = -\sum_{\aleph=x,y,z}\frac{J^{\aleph}_{g,n}-J^{\aleph}_{g,n-1}}{\Delta\aleph}. \tag{2.31}$$

In this equation, $\aleph$ represents the direction x, y, or z, $\Delta\aleph$ is the node width in the $\aleph$-direction. In the following, the subscripts "+1" and "-1" represent the nodes adjacent to the node n along the $\aleph$-direction for increasing, decreasing, respectively, $\aleph$ values (see Fig. 2.2). Furthermore, the quantities appearing on the right-hand side of Eq. (2.31) are defined as:

$$J^{\aleph}_{g,n} = \frac{1}{\Delta\Im\cdot\Delta\wp}\int_{-\Delta\Im/2}^{\Delta\Im/2}\int_{-\Delta\wp/2}^{\Delta\wp/2}\boldsymbol{J}_g(\boldsymbol{r}_{\aleph})\cdot\boldsymbol{n}^{+}_{\aleph}d\Im d\wp \tag{2.32}$$

and correspond to the surface-averaged net neutron current relative to the node n in the $\aleph$-direction, with $\boldsymbol{n}^{+}_{\aleph}$ being the outward normal relative to node n, as represented in Fig. 2.2. In the expression given by Eq. (2.32), $\boldsymbol{r}_{\aleph}$ thus represents the position of any point belonging to the surface normal to $\boldsymbol{n}^{+}_{\aleph}$ and defined as:

$$\begin{cases} \aleph = \Delta\aleph/2 \\ \Im\in[-\Delta\Im/2,\Delta\Im/2] \\ \wp\in[-\Delta\wp/2,\Delta\wp/2] \end{cases} \tag{2.33}$$

In order to get a solvable system of equations, a relationship between the surface-averaged neutron net current $J^{\aleph}_{g,n}$ and the node-averaged scalar neutron flux $\varphi_{g,n}$ needs to be derived. Using Fick's law and assuming that the scalar neutron flux in the middle of the nodes is equal to the node-averaged scalar neutron flux (box-scheme), the surface-averaged net neutron currents are approximated by the following formula [4]:

$$J^{\aleph}_{g,n} = -D_{g,0,n}\frac{\varphi^{b}_{g}-\varphi_{g,n}}{\Delta\aleph/2} \tag{2.34}$$

for a node n, where φ^{b}_{g} represents the scalar neutron flux at the node boundary located in $\Delta\aleph/2$ for node n. Such a net neutron current can also be evaluated from the volume-averaged scalar neutron flux in the adjacent node $n+1$ as:

$$J^{\aleph}_{g,n} = -D_{g,0,n+1}\frac{\varphi_{g,n+1}-\varphi^{b}_{g}}{\Delta\aleph/2}, \tag{2.35}$$

where φ^{b}_{g} is the scalar neutron flux at the node boundary located in $-\Delta\aleph/2$ for node $n+1$. In the derivation of Eqs. (2.34) and (2.35), the continuity of the neutron scalar

flux and of the net neutron current at the boundary between the nodes n and $n+1$ is preserved. Combining these two equations allows determining the scalar neutron flux at the interface as:

$$\varphi_g^b = \frac{D_{g,0,n}\varphi_{g,n} + D_{g,0,n+1}\varphi_{g,n+1}}{D_{g,0,n} + D_{g,0,n+1}}. \tag{2.36}$$

Using this expression in Eq. (2.34) leads to:

$$J_{g,n}^{\aleph} = -\frac{D_{g,0,n}D_{g,0,n+1}}{D_{g,0,n} + D_{g,0,n+1}} \times \frac{\varphi_{g,n+1} - \varphi_{g,n}}{\Delta\aleph/2}. \tag{2.37}$$

Utilizing this result in Eq. (2.31) finally gives:

$$\frac{1}{V_n}\int_{V_n} \overline{\nabla} \cdot [D_{g,0}(\boldsymbol{r})\overline{\nabla}\varphi_g(\boldsymbol{r})]d\boldsymbol{r} = -\sum_{\aleph=x,y,z} (a_{g,n}^{\aleph}\varphi_{g,n} + b_{g,n}^{\aleph}\varphi_{g,n+1} + c_{g,n}^{\aleph}\varphi_{g,n-1}) \tag{2.38}$$

with

$$a_{g,n}^{\aleph} = \frac{2D_{g,0,n-1}D_{g,0,n}}{(\Delta\aleph)^2(D_{g,0,n-1} + D_{g,0,n})} + \frac{2D_{g,0,n}D_{g,0,n+1}}{(\Delta\aleph)^2(D_{g,0,n} + D_{g,0,n+1})}, \tag{2.39}$$

$$b_{g,n}^{\aleph} = -\frac{2D_{g,0,n}D_{g,0,n+1}}{(\Delta\aleph)^2(D_{g,0,n} + D_{g,0,n+1})}, \tag{2.40}$$

$$c_{g,n}^{\aleph} = -\frac{2D_{g,0,n-1}D_{g,0,n}}{(\Delta\aleph)^2(D_{g,0,n-1} + D_{g,0,n})}. \tag{2.41}$$

The expressions of the above coefficients are only valid when the nodes $n-1$, n, and $n+1$ exist. At the boundaries of the system, boundary conditions need to be defined. In the developed computational tool, Marshak boundary conditions are used. In the case of multigroup diffusion theory, such boundary conditions read as:

$$\boldsymbol{J}_g(\boldsymbol{r}_{\aleph}) \cdot \boldsymbol{n}_{\aleph} = \frac{1}{2}\varphi_g^b, \tag{2.42}$$

where $\boldsymbol{r}_{\aleph}$ represents a spatial point on the boundary with $\boldsymbol{n}_{\aleph}$ being the outward normal to the boundary and φ_g^b represents the scalar neutron flux at the boundary. Assuming again that the scalar neutron flux in the middle of the nodes is equal to the node-averaged scalar neutron flux (box-scheme) and using Fick's law, the neutron current can be approximated and one then obtains:

- when the node $n-1$ does not exist:

$$\frac{1}{2}\varphi_g^b = D_{g,0,n}\frac{\varphi_{g,n} - \varphi_g^b}{\Delta\aleph/2}, \tag{2.43}$$

- when the node $n+1$ does not exist:

$$\frac{1}{2}\varphi_g^b = -D_{g,0,n}\frac{\varphi_g^b - \varphi_{g,n}}{\Delta\aleph/2}. \tag{2.44}$$

When the node $n-1$ does not exist, one finds, using Eq. (2.43), that:

$$\varphi_g^b = \frac{\varphi_{g,n}}{1 + \frac{\Delta\aleph}{4D_{g,0,n}}} \tag{2.45}$$

from which one deduces, using Fick's law, that:

$$J^{\aleph}_{g,n-1} = -D_{g,0,n}\frac{\varphi_{g,n} - \varphi^b_g}{\Delta\aleph/2} = -\frac{1/2}{1+\frac{\Delta\aleph}{4D_{g,0,n}}}\varphi_{g,n}. \tag{2.46}$$

The coupling coefficients appearing in Eq. (2.38) are thus expressed as:

$$a^{\aleph}_{g,n} = \frac{2D_{g,0,n}D_{g,0,n+1}}{(\Delta\aleph)^2(D_{g,0,n}+D_{g,0,n+1})} + \frac{1/2}{\Delta\aleph + \frac{(\Delta\aleph)^2}{4D_{g,0,n}}}, \tag{2.47}$$

$$b^{\aleph}_{g,n} = -\frac{2D_{g,0,n}D_{g,0,n+1}}{(\Delta\aleph)^2(D_{g,0,n}+D_{g,0,n+1})}, \tag{2.48}$$

$$c^{\aleph}_{g,n} = 0. \tag{2.49}$$

Likewise, when the node $n+1$ does not exist, one finds, using Eq. (2.49), that:

$$\varphi^b_g = \frac{\varphi_{g,n}}{1+\frac{\Delta\aleph}{4D_{g,0,n}}} \tag{2.50}$$

from which one deduces, using Fick's law, that:

$$J^{\aleph}_{g,n} = -D_{g,0,n}\frac{\varphi^b_g - \varphi_{g,n}}{\Delta\aleph/2} = \frac{1/2}{1+\frac{\Delta\aleph}{4D_{g,0,n}}}\varphi_{g,n}. \tag{2.51}$$

The coupling coefficients appearing in Eq. (2.38) are thus expressed as:

$$a^{\aleph}_{g,n} = \frac{2D_{g,0,n-1}D_{g,0,n}}{(\Delta\aleph)^2(D_{g,0,n-1}+D_{g,0,n})} + \frac{1/2}{\Delta\aleph + \frac{(\Delta\aleph)^2}{4D_{g,0,n}}}, \tag{2.52}$$

$$b^{\aleph}_{g,n} = 0, \tag{2.53}$$

$$c^{\aleph}_{g,n} = -\frac{2D_{g,0,n-1}D_{g,0,n}}{(\Delta\aleph)^2(D_{g,0,n-1}+D_{g,0,n})}. \tag{2.54}$$

2.4.2 Algorithm used for solving non-homogeneous equations

After applying the spatial discretisation presented in Section 2.4.1, the non-homogeneous equations [i.e. Eqs. (2.5), (2.14), (2.20)] reduce to the following generic form:

$$\overline{\overline{M}} \times \begin{bmatrix} \overline{\varphi}_1 \\ \overline{\varphi}_2 \end{bmatrix} = \begin{bmatrix} \overline{s}_1 \\ \overline{s}_2 \end{bmatrix}, \tag{2.55}$$

where $\overline{\varphi}_1$ and $\overline{\varphi}_2$ are column vectors representing the spatially-discretized generic scalar neutron flux (static flux or neutron noise) in the fast and thermal groups, respectively, and $\overline{s}_1$ and $\overline{s}_2$ are column vectors representing the spatially-discretized generic neutron source (neutron source or its fluctuations) or the spatially-discretized fluctuation of the macroscopic cross-sections, in the fast and thermal groups, respectively. If the system was discretized using N nodes, then $\overline{\varphi}_1$, $\overline{\varphi}_2$, $\overline{s}_1$, and $\overline{s}_2$ are column vectors of size N, and $\overline{\overline{M}}$ is a $2N \times 2N$ -matrix.

The solution to Eq. (2.55) is readily obtained as:

$$\begin{bmatrix} \overline{\varphi}_1 \\ \overline{\varphi}_2 \end{bmatrix} = \overline{\overline{M}}^{-1} \times \begin{bmatrix} \overline{s}_1 \\ \overline{s}_2 \end{bmatrix}. \tag{2.56}$$

Due to the very large number of nodes used in reactor calculations, the direct determination of the inverse of $\overline{\overline{M}}$ is usually impossible. Instead, the matrix $\overline{\overline{M}}$, which is sparse, is first factorized into a unit lower triangular matrix $\overline{\overline{L}}$ and an upper triangular matrix $\overline{\overline{U}}$ such that:

$$\overline{\overline{L}} \times \overline{\overline{U}} = \overline{\overline{P}} \times \overline{\overline{M}} \times \overline{\overline{Q}}, \tag{2.57}$$

where $\overline{\overline{P}}$ is a row permutation matrix and $\overline{\overline{Q}}$ is a column reordering matrix. The matrices $\overline{\overline{P}}$ and $\overline{\overline{Q}}$ are unitary matrices, i.e. they fulfill the following relationships:

$$\overline{\overline{P}} \times \overline{\overline{P}}^T = \overline{\overline{I}} = \overline{\overline{P}}^T \times \overline{\overline{P}} \tag{2.58}$$

and

$$\overline{\overline{Q}} \times \overline{\overline{Q}}^T = \overline{\overline{I}} = \overline{\overline{Q}}^T \times \overline{\overline{Q}}. \tag{2.59}$$

The row permutation and column reordering matrices are determined so that the lower triangular matrix $\overline{\overline{L}}$ and the upper triangular matrix $\overline{\overline{U}}$ are also sparse, in order to lower the requirements in data storage. Without row permutation and column reordering, the matrices $\overline{\overline{L}}$ and $\overline{\overline{U}}$ would be full. The matrix factorization as given by Eq. (2.58) is directly performed in *MATLAB* via the built-in *UMFPACK* package [5].

Eq. (2.55), which is rewritten as:

$$\overline{\overline{M}} \times \overline{\varphi} = \overline{s}, \tag{2.60}$$

can be simply solved by noticing from Eq. (2.57) that:

$$\overline{\overline{L}} \times \overline{\overline{U}} \times \overline{\overline{Q}}^{-1} = \overline{\overline{P}} \times \overline{\overline{M}}. \tag{2.61}$$

Multiplying Eq. (2.60) by $\overline{\overline{P}}$ and using Eq. (2.61), one finds that:

$$\overline{\overline{P}} \times \overline{\overline{M}} \times \overline{\varphi} = \overline{\overline{L}} \times \overline{\overline{U}} \times \overline{\overline{Q}}^{-1} \times \overline{\varphi} = \overline{\overline{P}} \times \overline{s}. \tag{2.62}$$

Since $\overline{\overline{L}}$ is a lower triangular matrix, $\overline{\overline{U}} \times \overline{\overline{Q}}^{-1} \times \overline{\varphi}$ can be simply obtained by forward substitution, which can be formally written as:

$$\overline{\overline{U}} \times \overline{\overline{Q}}^{-1} \times \overline{\varphi} = \overline{\overline{L}} \setminus (\overline{\overline{P}} \times \overline{s}). \tag{2.63}$$

Likewise, since $\overline{\overline{U}}$ is an upper triangular matrix, $\overline{\overline{Q}}^{-1} \times \overline{\varphi}$ can be simply obtained by backward substitution, which can be formally written as:

$$\overline{\overline{Q}}^{-1} \times \overline{\varphi} = \overline{\overline{U}} \setminus [\overline{\overline{L}}(\overline{\overline{P}} \times \overline{s})] \tag{2.64}$$

from which one finally obtains:

$$\overline{\varphi} = \overline{\overline{Q}} \times \{\overline{\overline{U}} \setminus [\overline{\overline{L}} \setminus (\overline{\overline{P}} \times \overline{s})]\}. \tag{2.65}$$

2.4.3 Algorithm used for solving eigenvalue equations

After applying the spatial discretisation presented in Section 2.4.1, the eigenvalue equations [i.e. Eq. (2.10)] reduce to the following generic form:

$$\overline{\overline{M}} \left[\begin{array}{c} \overline{\phi}_{1,m} \\ \overline{\phi}_{2,m} \end{array} \right] = \frac{1}{k_m} \overline{\overline{F}} \times \left[\begin{array}{c} \overline{\phi}_{1,m} \\ \overline{\phi}_{2,m} \end{array} \right], \tag{2.66}$$

where $\overline{\phi}_{1,m}$ and $\overline{\phi}_{2,m}$ are column vectors representing the spatially-discretized eigenfunction corresponding to mode m in the fast and thermal groups, respectively. If the system was discretized using N nodes, then $\overline{\phi}_{1,m}$ and $\overline{\phi}_{2,m}$ are column vectors of size N, and $\overline{\overline{M}}$ and $\overline{\overline{F}}$ are $2N \times 2N$-matrices.

Iterative techniques are required to solve Eq. (2.66), which can be rewritten as:

$$\overline{\overline{A}} \times x_m = k_m x_m \tag{2.67}$$

with

$$\overline{\overline{A}} = \overline{\overline{M}}^{-1} \times \overline{\overline{F}} \tag{2.68}$$

and

$$x_m = \left[\begin{array}{c} \overline{\phi}_{1,m} \\ \overline{\phi}_{2,m} \end{array} \right]. \tag{2.69}$$

As earlier explained in Section 2.4.2, the calculation of the inverse of $\overline{\overline{M}}$ is avoided by first performing a LU factorization of $\overline{\overline{M}}$ will full pivoting as:

$$\overline{\overline{L}} \times \overline{\overline{U}} = \overline{\overline{P}} \times \overline{\overline{M}} \times \overline{\overline{Q}} \tag{2.70}$$

leading for Eq. (2.67) to:

$$\overline{\overline{Q}} \times \{\overline{\overline{U}} \setminus [\overline{\overline{L}} \setminus (\overline{\overline{P}} \times \overline{\overline{F}} \times x_m)]\} = k_m x_m. \tag{2.71}$$

Numerous techniques exist for solving the eigenvalue equation given by Eq. (2.67). They are all based on the power iteration method, which is first recalled hereafter [4].

The power iteration method consists in determining a new and better estimate of the solution (x_m, k_m) from a previous estimate. If p represents the iteration number, the power iteration method simply reads as:

$$x_m^{(p)} = \frac{1}{k_m^{(p-1)}} \overline{\overline{A}} \times x_m^{(p-1)}. \tag{2.72}$$

The power iteration method can then be seen as an operator $\overline{\overline{A}}$ acting on the vector $x_m^{(p-1)}$. When the iterative scheme converges, Eq. (2.72) becomes:

$$\overline{x}_m^{(\infty)} = \frac{1}{k_m^{(\infty)}} \overline{\overline{A}} \times \overline{x}_m^{(\infty)} \tag{2.73}$$

from which one deduces that:

$$k_m^{(\infty)} = \frac{\overline{x}_m^{(\infty),T} \times [\overline{\overline{A}} \times \overline{x}_m^{(\infty)}]}{\overline{x}_m^{(\infty),T} \times \overline{x}_m^{(\infty)}}. \tag{2.74}$$

Based on this expression for the converged eigenvalue, the new iterate of k_m can be estimated, once the new iterate of the vector $\overline{x}_m$ has been determined, as:

$$k_m^{(p)} = \frac{\overline{x}_m^{(p-1),T} \times [\overline{\overline{A}} \times \overline{x}_m^{(p-1)}]}{\overline{x}_m^{(p-1),T} \times \overline{x}_m^{(p-1)}} = k_m^{(p-1)} \frac{\overline{x}_m^{(p-1),T} \times \overline{x}_m^{(p)}}{\overline{x}_m^{(p-1),T} \times \overline{x}_m^{(p-1)}}, \tag{2.75}$$

where the last equality was obtained using Eq. (2.72). The iterative scheme given by Eqs. (2.72) and (2.75) completely defines the power iteration method. In the following, it will be demonstrated that the power iteration method converges to the eigenvector of the iterative matrix $\overline{\overline{A}}$ having the largest eigenvalue. When the power iteration method is applied to any initial start vector $\overline{x}_m^{(0)}$ with a given value for $k_m^{(0)}$, the p iterate can be written, using Eq. (2.72), as:

$$\overline{x}_m^{(p)} = \frac{\overline{\overline{A}}^p \times \overline{x}_m^{(0)}}{k_m^{(p-1)} k_m^{(p-2)} ... k_m^{(0)}}. \tag{2.76}$$

The initial vector can be expanded on the eigenvectors $\overline{x}_l$ of the matrix $\overline{\overline{A}}$ as:

$$\overline{x}_m^{(0)} = \sum_l \alpha_l \overline{x}_l \tag{2.77}$$

with

$$\alpha_l = \overline{x}_l^T \times \overline{x}_m^{(0)}. \tag{2.78}$$

Using Eqs. (2.77) and (2.67) into Eq. (2.76) leads to:

$$\overline{x}_m^{(p)} = \frac{\sum_l \alpha_l k_l^p \overline{x}_l}{k_m^{(p-1)} k_m^{(p-2)} ... k_m^{(0)}} = \frac{\alpha_0 k_0^p}{k_m^{(p-1)} k_m^{(p-2)} ... k_m^{(0)}} \times [\overline{x}_0 + \sum_{l \geq 1} \frac{\alpha_l}{\alpha_0} \left(\frac{k_l}{k_0} \right)^p \overline{x}_l]. \tag{2.79}$$

According to Eq. (2.12), one has:

$$\frac{k_l}{k_0} < 1, \text{ for } l \geq 1, \tag{2.80}$$

which results, for Eq. (2.79), in:

$$\lim_{p \to \infty} \overline{x}_m^{(p)} = \frac{\alpha_0 k_0^p}{k_m^{(p-1)} k_m^{(p-2)} ... k_m^{(0)}} \overline{x}_0. \tag{2.81}$$

Using Eq. (2.78) into Eq. (2.75) also leads to:

$$k_m^{(p)} = k_m^{(p-1)} \times \frac{\overline{x}_m^{(p-1),T} \times \frac{\alpha_0 k_0^p}{k_m^{(p-1)} k_m^{(p-2)} ... k_m^{(0)}} \times \left[\overline{x}_0 + \sum_{l \geq 1} \frac{\alpha_l}{\alpha_0} \left(\frac{k_l}{k_0} \right)^p \overline{x}_l \right]}{\overline{x}_m^{(p-1),T} \times \frac{\alpha_0 k_0^{p-1}}{k_m^{(p-2)} ... k_m^{(0)}} \times \left[\overline{x}_0 + \sum_{l \geq 1} \frac{\alpha_l}{\alpha_0} \left(\frac{k_l}{k_0} \right)^{p-1} \overline{x}_l \right]} \tag{2.82}$$

or

$$k_m^{(p)} = k_0 \times \frac{\overline{x}_m^{(p-1),T} \times \left[\overline{x}_0 + \sum_{l \geq 1} \frac{\alpha_l}{\alpha_0} \left(\frac{k_l}{k_0} \right)^p \overline{x}_l \right]}{\overline{x}_m^{(p-1),T} \times \left[\overline{x}_0 + \sum_{l \geq 1} \frac{\alpha_l}{\alpha_0} \left(\frac{k_l}{k_0} \right)^{p-1} \overline{x}_l \right]}. \tag{2.83}$$

Because of Eq. (2.80), one thus finds that:

$$\lim_{p\to\infty} k_m^{(p)} = k_0. \tag{2.84}$$

Eqs. (2.81) and (2.84) therefore mean that the power iteration method, as implemented in Eqs. (2.72) and (2.75), leads to the solution corresponding to the fundamental mode, i.e. (x_0, k_0). The convergence of the power iteration method is directly related to the ratio between the higher eigenvalues and the first eigenvalue, i.e. $\frac{k_l}{k_0}$. In the case of nuclear reactor calculations, the eigenvalues are usually clustered eigenvalues, i.e. quite close to each other. This decreases the convergence rate of the power iteration method. In the developed computational tool, other techniques have been implemented in order to be able to determine any eigenmode m and to speed up the convergence rate of the iterations. Namely, the explicitly-restarted Arnoldi method and the power iteration method with Wielandt's shift have been used.

Explicitly-restarted Arnoldi method

Some of the most efficient techniques to solve eigenvalue problems are based on Krylov subspace methods. The explicitly-restarted Arnoldi method belongs to this class of techniques [6]. The Arnoldi method is based on the fact that useful information is lost during the application of the classical power iteration method. Namely, only the latest estimates of the eigenvector and eigenvalue are used for subsequently calculating a new estimate of the eigenvector and eigenvalue. In the Arnoldi method instead, a Krylov subspace containing an estimate of the eigenvectors of the matrix $\overline{\overline{A}}$ obtained during the application of the power iteration method during t iterations is first constructed, i.e. the following space is constructed:

$$\Re_t\left(\overline{\overline{A}}, \overline{v}\right) = \text{span}\{\overline{v}, \overline{\overline{A}} \times \overline{v}, \overline{\overline{A}}^2 \times \overline{v}, ..., \overline{\overline{A}}^{t-1} \times \overline{v}\} \tag{2.85}$$

with

$$t \ll \text{dimension of } \overline{\overline{A}}. \tag{2.86}$$

Thereafter, an orthogonal basis of this subspace is estimated. Finally, the eigenvectors/ eigenvalues of this orthogonal basis are determined. It will be demonstrated thereafter that the eigenvectors of the matrix representing the projection of the original matrix on the Krylov subspace can be used for determining the eigenvectors of the original matrix, and therefore the eigenvectors of $\overline{\overline{A}}$ can be estimated from the eigenvectors of the matrix representing the projection of the original matrix on the Krylov subspace. The main advantage of this procedure is the fact that the projection matrix is an Hessenberg matrix of size $t \times t$, i.e. much smaller than the size of the original matrix. Consequently, the determination of the t eigenvectors and corresponding eigenvalues is relatively easy.

The iterative scheme of the explicitly-restarted Arnoldi method can be sketched as follows [7,8]:

- First, an orthogonal basis of $\Re_t(\overline{\overline{A}}, \overline{v})$ using the Gram-Schmidt orthogonalization process is constructed as detailed in Algorithm 1. If a breakdown of the algorithm occurs at step j (when the norm of $\overline{w}_{:,j}$ is equal to zero), then the projection on the subspace $\Re_j$ is exact. The matrix $\overline{\overline{V}}_t = (v_{:,1}, v_{:,2}, ..., v_{:,t})$ represents an orthogonal basis of $\Re_t(\overline{\overline{A}}, \overline{v})$. In addition, this first step also results in the construction of a Hessenberg matrix of dimension $t+1 \times t$:

$$\overline{\overline{H}} = (h_{k,j})_{1\leq k\leq t+1, 1\leq j\leq t} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & ... & h_{1,t} \\ h_{2,1} & h_{2,2} & h_{1,1} & ... & h_{2,t} \\ 0 & h_{3,2} & h_{3,3} & ... & h_{3,t} \\ ... & 0 & ... & ... & ... \\ 0 & ... & 0 & h_{t,t-1} & h_{t,t} \\ 0 & ... & ... & 0 & h_{t+1,t} \end{bmatrix}. \tag{2.87}$$

Defining the reduced Hessenberg matrix $\overline{\overline{H}}_r$ as the original Hessenberg matrix $\overline{\overline{H}}$ deprived from its last row, i.e.

$$\overline{\overline{H}}_r = (h_{k,j})_{1\leq k\leq t, 1\leq j\leq t} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & ... & h_{1,t} \\ h_{2,1} & h_{2,2} & h_{1,1} & ... & h_{2,t} \\ 0 & h_{3,2} & h_{3,3} & ... & h_{3,t} \\ ... & 0 & ... & ... & ... \\ 0 & ... & 0 & h_{t,t-1} & h_{t,t} \end{bmatrix}, \tag{2.88}$$

one finds by the application of Algorithm 1 that [7]:

$$\overline{\overline{A}} \times v_{:,j} = \sum_{k=1}^{j+1} h_{k,j} v_{:,k} \text{ for } j = 1, 2, ..., t \tag{2.89}$$

resulting in:

$$\overline{\overline{A}} \times \overline{\overline{V}}_t = \overline{\overline{V}}_{t+1} \times \overline{\overline{H}} = \overline{\overline{V}}_t \times \overline{\overline{H}}_r + \overline{w}_{:,t} \bar{e}_t^T, \tag{2.90}$$

where $\bar{e}_t$ represents the $t-$th column of the $t \times t$ identity matrix. Making use of the orthonormality of $\{v_{:,1}, v_{:,2}, ..., v_{:,t}\}$, one finally obtains that:

$$\overline{\overline{V}}_t^T \times \overline{\overline{A}} \times \overline{\overline{V}}_t = \overline{\overline{H}}_r. \tag{2.91}$$

```
choice of an initial vector q̄ and of the subspace dimension t
normalization of the vector q̄, i.e. v̄:,1 = q̄/‖q̄‖2
for j=1 to t
    w̄:,j = Ā × v̄:,j
    for k=1 to j
        orthogonal projection of w̄:,j on the previous v̄:,k, i.e.
            hk,j = w̄:,j × v̄:,k
        deprivation of w̄:,j of its component on v̄:,k, i.e.
            w̄:,j − hk,j v̄:,k → w̄:,j
    end
    hj+1,j = ‖w̄:,j‖2
    If hj+1,j = 0 then stop
    v̄:,j+1 = w̄:,j/‖w̄:,j‖2
end
```

Algorithm 1 Gram-Schmidt orthogonalization procedure of the matrix $\overline{\overline{A}}$ in the Arnoldi method.

- Thereafter, the pairs of eigenvectors $x_{:,j}$ and eigenvalues λ_j of the reduced Hessenberg matrix $\overline{\overline{H}}_r$ (for $1 \leq j \leq t$) are determined, resulting in:

$$\overline{\overline{H}}_r \times \overline{\overline{X}} = \overline{\overline{X}} \times \overline{\overline{\lambda}} \tag{2.92}$$

with

$$\overline{\overline{X}} = (x_{:,1}, x_{:,2}, ..., x_{:,t}) \tag{2.93}$$

and

$$\overline{\overline{\lambda}} = \begin{bmatrix} \lambda_1 & 0 & ... & ... & 0 \\ 0 & \lambda_2 & 0 & ... & ... \\ ... & 0 & ... & 0 & ... \\ ... & ... & 0 & \lambda_{t-1} & 0 \\ 0 & ... & ... & 0 & \lambda_t \end{bmatrix}. \tag{2.94}$$

Since the reduced Hessenberg matrix is chosen to be of small size, the determination of its eigenvectors and eigenvalues is relatively easy. Such a determination is directly carried out in *MATLAB* via the built-in *LAPACK* package [9].

- Combining Eqs. (2.91) and (2.92), one notices that:

$$\overline{\overline{V}}_t^T \times \overline{\overline{A}} \times \overline{\overline{V}}_t \times \overline{\overline{X}} = \overline{\overline{H}}_r \times \overline{\overline{X}} = \overline{\overline{X}} \times \overline{\overline{\lambda}}. \tag{2.95}$$

Making use again of the orthonormality of $\{v_{:,1}, v_{:,2}, ..., v_{:,t}\}$, one finally obtains:

$$\overline{\overline{V}}_t \times \overline{\overline{V}}_t^T \times \overline{\overline{A}} \times \overline{\overline{V}}_t \times \overline{\overline{X}} = \overline{\overline{A}} \times \overline{\overline{V}}_t \times \overline{\overline{X}} = \overline{\overline{V}}_t \times \overline{\overline{X}} \times \overline{\overline{\lambda}}. \tag{2.96}$$

Eq. (2.96) demonstrates that the eigenvectors of $\overline{\overline{A}}$ are given by the columns of $\overline{\overline{V}}_t \times \overline{\overline{X}}$ and the corresponding eigenvalues are then $\overline{\overline{\lambda}}$.

- Since the eigenvalues of the reduced Hessenberg matrix $\overline{\overline{H}}_r$ might be bad approximations of the eigenvalues of the original matrix $\overline{\overline{A}}$, especially if the subspace dimension t is kept small, the algorithm is (explicitly) restarted with a linear combination of the eigenvectors of $\overline{\overline{A}}$ until some convergence criteria on the eigenvectors are fulfilled.

The explicitly-restarted Arnoldi method is an extremely efficient method for calculating the eigenfunctions/eigenvalues in a minimum computational time, since several eigenmodes can be estimated simultaneously. Nevertheless, it cannot be proved that the eigenvalues of $\overline{\overline{H}}_r$ will converge to the extreme eigenvalues of $\overline{\overline{A}}$ when $\overline{\overline{A}}$ is non-symmetric (even if such a convergence is usually observed) [10]. In order to circumvent possible convergence problem, a power iteration method using Wielandt's shift technique was also implemented in the numerical tool, and is explained in the following.

Power iteration method with Wielandt's shift technique

The basic idea in Wielandt's shift technique is to modify the original problem as given by Eq. (2.66) into the following one [4]:

$$\left(\overline{\overline{M}} - \frac{1}{k_{est}}\overline{\overline{F}}\right) \times \begin{bmatrix} \overline{\phi}_{1,m} \\ \overline{\phi}_{2,m} \end{bmatrix} = \left(\frac{1}{k_m} - \frac{1}{k_{est}}\right) \overline{\overline{F}} \times \begin{bmatrix} \overline{\phi}_{1,m} \\ \overline{\phi}_{2,m} \end{bmatrix}, \tag{2.97}$$

where k_{est} is a known (input) parameter. Eq. (2.97) can be rewritten as:

$$\overline{\overline{A}}_W \times x_m = \alpha_m x_m \tag{2.98}$$

with

$$\overline{\overline{A}}_W = \left(\overline{\overline{M}} - \frac{1}{k_{est}}\overline{\overline{F}}\right)^{-1} \times \overline{\overline{F}}, \tag{2.99}$$

$$\frac{1}{\alpha_m} = \frac{1}{k_m} - \frac{1}{k_{est}} \tag{2.100}$$

and

$$x_m = \begin{bmatrix} \overline{\phi}_{1,m} \\ \overline{\phi}_{2,m} \end{bmatrix}. \tag{2.101}$$

As earlier explained in Section 2.4.2, the calculation of the inverse of $\overline{\overline{M}} - \frac{1}{k_{est}}\overline{\overline{F}}$ is avoided by first performing a LU factorization of $\overline{\overline{M}} - \frac{1}{k_{est}}\overline{\overline{F}}$ will full pivoting as:

$$\overline{\overline{L}} \times \overline{\overline{U}} = \overline{\overline{P}} \times \left(\overline{\overline{M}} - \frac{1}{k_{est}}\overline{\overline{F}}\right) \times \overline{\overline{Q}} \tag{2.102}$$

leading for Eq. (2.98) to:

$$\overline{\overline{Q}} \times \left\{\overline{\overline{U}} \setminus \left[\overline{\overline{L}} \setminus \left(\overline{\overline{P}} \times \overline{\overline{F}} \times x_m\right)\right]\right\} = \alpha_m x_m. \tag{2.103}$$

The power iteration method, which was earlier presented, can be applied to the modified equation (2.98), thus leading to the following iterative scheme:

$$x_m^{(p)} = \frac{1}{\alpha_m^{(p-1)}}\overline{\overline{A}}_W \times x_m^{(p-1)} \tag{2.104}$$

and

$$\alpha_m^{(p)} = \frac{\overline{x}_m^{(p-1),T} \times [\overline{\overline{A}}_W \times \overline{x}_m^{(p-1)}]}{\overline{x}_m^{(p-1),T} \times \overline{x}_m^{(p-1)}} = \alpha_m^{(p-1)} \frac{\overline{x}_m^{(p-1),T} \times \overline{x}_m^{(p)}}{\overline{x}_m^{(p-1),T} \times \overline{x}_m^{(p-1)}}. \tag{2.105}$$

Using Eq. (2.100), one also obtains:

$$k_m^{(p)} = \left[\frac{1}{k_{est}} + \left(\frac{1}{k_m^{(p-1)}} - \frac{1}{k_{est}}\right) \frac{\overline{x}_m^{(p-1),T} \times \overline{x}_m^{(p-1)}}{\overline{x}_m^{(p-1),T} \times \overline{x}_m^{(p)}}\right]^{-1}. \tag{2.106}$$

The iterative scheme given by Eqs. (2.104) and (2.106) completely defines the power iteration method using Wielandt's shift technique. In the following, it will be demonstrated that this iterative scheme converges to the eigenvector of the matrix $\overline{\overline{A}}$ having the eigenvalue closest to k_{est}. When the iteration method is applied to any initial start vector $\overline{x}_m^{(0)}$ with a given value for $\alpha_m^{(0)}$, the p iterate can be written, using Eq. (2.104), as:

$$\overline{x}_m^{(p)} = \frac{\overline{\overline{A}}_W^p \times \overline{x}_m^{(0)}}{\alpha_m^{(p-1)}\alpha_m^{(p-2)}...\alpha_m^{(0)}}. \tag{2.107}$$

The initial vector can be expanded on the eigenvectors $\overline{x}_l$ of the matrix $\overline{\overline{A}}$ as:

$$\overline{x}_m^{(0)} = \sum_l \beta_l \overline{x}_l \tag{2.108}$$

with

$$\beta_l = \overline{x}_l^T \times \overline{x}_m^{(0)}. \tag{2.109}$$

It can also be noticed that the eigenvectors $\overline{x}_l$ of the matrix $\overline{\overline{A}}$ are also eigenvectors of the matrix $\overline{\overline{A}}_W$, i.e. one has both:

$$\overline{\overline{A}} \times \overline{x}_l = k_l x_l \tag{2.110}$$

and

$$\overline{\overline{A}}_W \times \overline{x}_l = \alpha_l x_l. \tag{2.111}$$

Using Eqs. (2.108) and (2.111) into Eq. (2.107) leads to:

$$\overline{x}_m^{(p)} = \frac{\sum_l \beta_l \alpha_l^p \overline{x}_l}{\alpha_m^{(p-1)} \alpha_m^{(p-2)} ... \alpha_m^{(0)}} = \frac{\beta_q \alpha_q^p}{\alpha_m^{(p-1)} \alpha_m^{(p-2)} ... \alpha_m^{(0)}} \times \left[\overline{x}_q + \sum_{l \neq q} \frac{\beta_l}{\beta_q} \left(\frac{\alpha_l}{\alpha_q} \right)^p \overline{x}_l \right] \tag{2.112}$$

with q being a positive integer. According to Eq. (2.100), one has:

$$\frac{\alpha_l}{\alpha_q} = \frac{\frac{k_{est}}{k_q} - 1}{\frac{k_{est}}{k_l} - 1}. \tag{2.113}$$

It can then be noticed from Eqs. (2.112) and (2.113) that the iterative algorithm will converge to the mode q fulfilling the following condition:

$$\left| \frac{k_{est}}{k_q} - 1 \right| < \left| \frac{k_{est}}{k_l} - 1 \right|. \tag{2.114}$$

The algorithm will thus converge to the mode q for which k_q is the closest to k_{est} as:

$$\lim_{p \to \infty} \overline{x}_m^{(p)} = \frac{\beta_q \alpha_q^p}{\alpha_m^{(p-1)} \alpha_m^{(p-2)} ... \alpha_m^{(0)}} \overline{x}_q. \tag{2.115}$$

Using Eq. (2.111) into Eq. (2.105) also leads to:

$$\alpha_m^{(p)} = \alpha_m^{(p-1)} \times \frac{\overline{x}_m^{(p-1),T} \times \frac{\beta_q \alpha_q^p}{\alpha_m^{(p-1)} \alpha_m^{(p-2)} ... \alpha_m^{(0)}} \times \left[\overline{x}_q + \sum_{l \neq q} \frac{\beta_l}{\beta_q} \left(\frac{\alpha_l}{\alpha_q} \right)^p \overline{x}_l \right]}{\overline{x}_m^{(p-1),T} \times \frac{\beta_q \alpha_q^{p-1}}{\alpha_m^{(p-2)} ... \alpha_m^{(0)}} \times \left[\overline{x}_q + \sum_{l \neq q} \frac{\beta_l}{\beta_q} \left(\frac{\alpha_l}{\alpha_q} \right)^{p-1} \overline{x}_l \right]} \tag{2.116}$$

or

$$\alpha_m^{(p)} = \alpha_q \times \frac{\overline{x}_m^{(p-1),T} \times \left[\overline{x}_q + \sum_{l \neq q} \frac{\beta_l}{\beta_q} \left(\frac{\alpha_l}{\alpha_q} \right)^p \overline{x}_l \right]}{\overline{x}_m^{(p-1),T} \times \left[\overline{x}_q + \sum_{l \neq q} \frac{\beta_l}{\beta_q} \left(\frac{\alpha_l}{\alpha_q} \right)^{p-1} \overline{x}_l \right]}. \tag{2.117}$$

Because of Eq. (2.114), one thus finds that:

$$\lim_{p \to \infty} \alpha_m^{(p)} = \alpha_q, \tag{2.118}$$

which is also equivalent to:

$$\lim_{p \to \infty} k_m^{(p)} = k_q. \tag{2.119}$$

Eqs. (2.115) and (2.119) therefore mean that the power iteration method with Wielandt's shift, as implemented in Eqs. (2.104) and (2.105), leads to the solution corresponding to the mode having its eigenvalue closest to k_{est}. The convergence of this method is directly related to how close to one of the existing eigenvalues k_{est} actually is. In the developed computational tool, a first guess of the different eigenvalues is obtained by applying the Arnoldi method outlined above without performing any restart. Thereafter, each of these estimated eigenvalues is used as the parameter k_{est} in the power iteration method with Wielandt's shift.

Chapter 3

THERMO-HYDRAULIC MODELS

3.1 Introduction

In this section, the thermo-hydraulic models implemented in the tool are discussed. The thermo-hydraulic part of the tool has the ability to calculate the solution to static problems, as well as the solution to dynamic problems in linear theory and in the frequency domain. The equations thus solved in these different cases are presented below.

First, the static thermo-hydraulic module of the tool and the respective equations it is based on are described. The corresponding numerical algorithms developed and utilized in the module are then given. Next, the numerical scheme used for spatially discretising the static thermo-hydraulic equations is described. Thereafter, the numerical algorithms allowing solving the static equations are summarized.

Further, the dynamic thermo-hydraulic module of the tool together with the respective equations are presented. Following the same structure as in the static case, the corresponding numerical algorithms developed and utilized in the module (including both the spatial discretization scheme and the numerical algorithms used for solving the dynamic equations) are touched upon.

The present thermo-hydraulic model is based on a set of three equations: mass balance, momentum balance and energy balance equations. For the sake of simplicity, the mixture model for all thermo-hydraulic quantities was utilized.

3.2 Derivation of the static thermo-hydraulic equations

The most general time-independent thermo-hydraulic model can be represented by the local conservation equations of mass, momentum and energy as a function of space, which are given as follows:

$$\nabla \cdot [\rho_m(\boldsymbol{r})\overline{v}_m(\boldsymbol{r})] = 0, \tag{3.1}$$

$$\nabla \cdot [\rho_m(\boldsymbol{r})\overline{v}_m(\boldsymbol{r}) \otimes \overline{v}_m(\boldsymbol{r})] = \nabla \cdot \overline{\overline{\tau}}(\boldsymbol{r}) - \nabla P(\boldsymbol{r}) + \overline{g}\rho_m(\boldsymbol{r}), \tag{3.2}$$

$$\nabla \cdot [\rho_m(\boldsymbol{r})\overline{v}_m(\boldsymbol{r})h_m(\boldsymbol{r})] = -\nabla \cdot \overline{q}''(\boldsymbol{r}), \tag{3.3}$$

where $\rho_m(\boldsymbol{r})$, $\overline{v}_m(\boldsymbol{r})$, $h_m(\boldsymbol{r})$ and $P(\boldsymbol{r})$ are the coolant/moderator density, velocity, enthalpy and pressure, respectively, $\overline{\overline{\tau}}(\boldsymbol{r})$ is the stress tensor, $\overline{q}''(\boldsymbol{r})$is the heat flux and $\overline{g}$ stands for gravitational constant. The other notations are standard. To simplify the calculations, the local energy conservation equation was replaced by the corresponding local enthalpy

conservation equation, where the enthalpy change due to variations in pressure, stresses and volumetric heat production was assumed to be negligible and, therefore, the corresponding terms in the equation were left out.

Since the full microscopic description of the two-phase flow is generally not possible (i.e. Eqs. (3.1)-(3.3) cannot be solved in space in an exact manner), another technique, referred to as a "macroscopic description" representing the spatial averaging of all quantities over the relevant node volume, is usually used.

3.2.1 Algorithm used for the spatial discretisation

All flow properties are first discretized and then averaged in space on relevant volumes. The different notations and conventions used in the discretization and averaging procedure are highlighted in Fig. 3.1. In the figure, the (I, J, K) indexes denote the spatial position of a thermo-hydraulic (moderator/coolant) node.

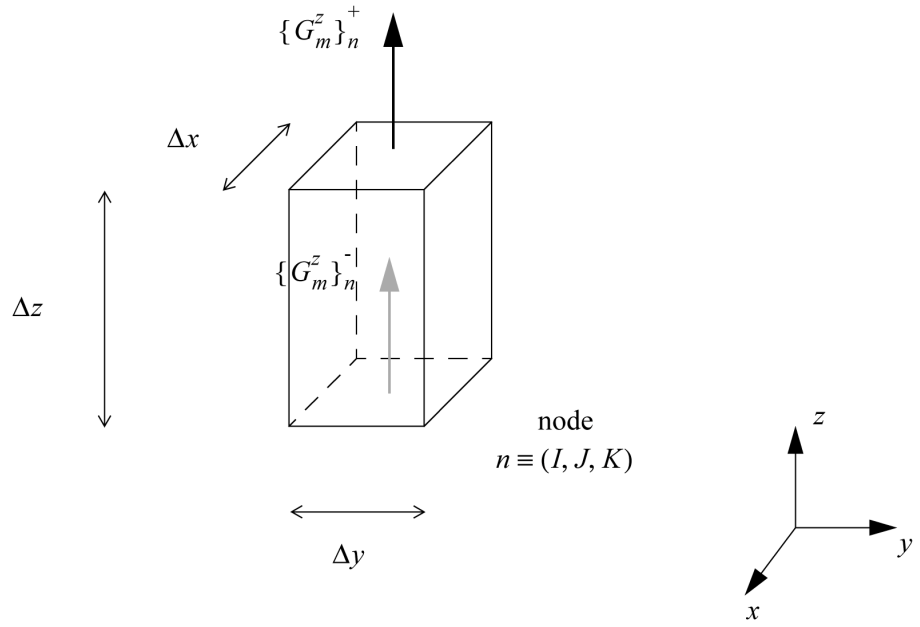

Figure 3.1: Principles and conventions used for the spatial discretisation of a thermo-hydraulic node n.

Such a spatial homogenization of Eqs. (3.1) - (3.3) can be written as

$$\int_{V_m} \nabla \cdot [\rho_m(\boldsymbol{r})\overline{v}_m(\boldsymbol{r})]\, dV = 0, \tag{3.4}$$

$$\int_{V_m} \nabla \cdot [\rho_m(\boldsymbol{r})\overline{v}_m(\boldsymbol{r}) \otimes \overline{v}_m(\boldsymbol{r})]\, dV = \int_{V_m} \nabla \cdot \overline{\overline{\tau}}(\boldsymbol{r}) dV - \int_{V_m} \nabla P(\boldsymbol{r}) dV + \int_{V_m} \overline{g}\rho_m(\boldsymbol{r}) dV, \tag{3.5}$$

$$\int_{V_m} \nabla \cdot [\rho_m(\boldsymbol{r})\overline{v}_m(\boldsymbol{r}) h_m(\boldsymbol{r})] = -\int_{V_m} \nabla \cdot \overline{q}''(\boldsymbol{r}) dV \tag{3.6}$$

by assuming one-dimensional upward vertical flow and neglecting the cross-flow between the bundles. One then defines the following notations for the volume- and area-averaged node-wise quantities:

$$\langle X_m \rangle = \frac{1}{V_m} \int_{V_m} X_m(\boldsymbol{r}) dV, \tag{3.7}$$

$$\{\rho_m\}_0 = \frac{1}{A_m} \int_{A_m} \rho_m(\boldsymbol{r}) dA, \tag{3.8}$$

$$\{G_m\}_0 = \frac{1}{A_m} \int_{A_m} G_m(\boldsymbol{r}) dA, \tag{3.9}$$

$$\{P_z\}_0 = \frac{1}{A_m} \int_{A_m} P_z(\boldsymbol{r}) dA, \tag{3.10}$$

$$\{q_r''\}_{m,0}^S = \frac{1}{S_m} \int_{S_m} q_r''(\boldsymbol{r}) dS, \tag{3.11}$$

$$\{v_{z,m}\}_0^{\pm} = \frac{\{\rho_m v_{z,m}\}_0^{\pm}}{\{\rho_m\}_0^{\pm}}, \tag{3.12}$$

$$\left[\{v_{z,m}\}_0^{\pm}\right]^2 = \frac{\{\rho_m v_{z,m}^2\}_0^{\pm}}{\{\rho_m\}_0^{\pm}}, \tag{3.13}$$

$$\{h_m\}_0^{\pm} = \frac{\{\rho_m v_{z,m} h_m\}_0^{\pm}}{\{\rho_m v_{z,m}\}_0^{\pm}}, \tag{3.14}$$

where X_m designates any thermo-hydraulic quantity, q_r'' is the radial heat flux, index "S" stands for the averaging over the lateral surface, index "m" in the last equation denotes that the averaging is performed over moderator region, A_m, V_m and S_m denote the cross-sectional area, volume and lateral surface area of a moderator node, respectively. For practical terms it was assumed that Eq. (3.13) can be computed just as the square of Eq. (3.12). The, by neglecting the axial heat flux and taking Gauss' divergence theorem into account, Eqs. (3.4) - (3.6) can thus be simplified as:

$$A_m \left(\{\rho_m\}_0^+ \{v_{z,m}\}_0^+ - \{\rho_m\}_0^- \{v_{z,m}\}_0^-\right) = 0, \tag{3.15}$$

$$\begin{aligned} A_m \left(\{v_{z,m}\}_0^+ - \{v_{z,m}\}_0^-\right) \{G_m\}_0 = &-\frac{\langle F_M \rangle_0 V_m}{2 D_e \rho_l} \langle G_m \rangle_0^2 \\ &- A_m \left[\{P_z\}_0^+ - \{P_z\}_0^-\right] - g V_m \langle \rho_m \rangle_0 , \end{aligned} \tag{3.16}$$

$$A_m \left(\{\rho_m\}_0^+ \{v_m\}_0^+ \{h_m\}_0^+ - \{\rho_m\}_0^- \{v_m\}_0^- \{h_m\}_0^-\right) = -S_m \{q_r''\}_{m,0}^S. \tag{3.17}$$

The "0" subscript is an indicator that the variable in question belongs to the static calculation. The superscripts "+" and "−" indicate the upper and the lower node interface, $\langle F_M \rangle_0$ is the pressure friction multiplier (factor) defined separately for single and two-phase regions (see next section), D_e is the hydraulic diameter and $\langle G_m \rangle_0$ corresponds to the total mass flux as liquid. In the present study, the size of a thermo-hydraulic node is chosen to be comparable with the size of the corresponding neutronic node. The total number of thermo-hydraulic nodes is limited to the number of nodes used in commercial codes for LWRs, i.e. $M \times N \times P$, where M specifies the total number of nodes in $x-$direction, N in $y-$direction and P in $z-$direction. The latter is justified by the fact that

the input static node-wise parameters are extracted from commercial codes. In the above pressure balance equation (momentum equation) (3.16), the following simplified model for the volume-averaged friction pressure drop in $z-$direction was implemented:

$$\langle \Delta P_z \rangle_0 = \langle F_M \rangle_0 \frac{\langle G_m \rangle_0^2 \Delta z}{2 D_e \rho_l}, \tag{3.18}$$

where $\langle F_M \rangle_0$ stands for the node-wise two-phase friction multiplier or friction coefficient.

The equations for the conservation of mass, momentum and energy were considered for the mixture of the coolant/moderator vapor and liquid at all locations in the core, leading to the so-called Homogeneous Equilibrium Model (HEM). In order to provide a more realistic computation of the mixture coolant/moderator density, the present model was complemented with a simple slip ratio model.

3.2.2 Algorithm used for solving static thermo-hydraulic equations

The steady-state three-dimensional spatial distributions of themo-hydraulic parameters can be computed by solving the system of coupled discretized equations (3.15)-(3.17). These equations are solved with respect to the coolant/moderator enthalpy, velocity and pressure. Throughout this section it is assumed that all the required input data are known and obtained from a commercial core simulator.

- *coolant/moderator enthalpy*
  Assuming that the nodal power has already been computed (for example, using Eq. (2.66) and given input cross-sections), the spatial distribution of the surface-averaged enthalpy can be estimated from the corresponding enthalpy equation (Eq. (3.17)) which is solved with respect to the surface-averaged enthalpy in the upper node as:

$$\{h_m\}_0^+ = \{h_m\}_0^- - \frac{S_m \{q_r''\}_{m,0}^S}{A_m \{\rho_m\}_0^- \{v_m\}_0^-}. \tag{3.19}$$

  Then, assuming that

$$S_m \{q_r''\}_{m,0}^S = -S_f \{q_r''\}_{f,0}^S, \tag{3.20}$$

  where indexes "m" and "f" denote the averaging over the moderator or fuel region, correspondingly, S_f designates the lateral surface area of the fuel, $\{q_r''\}_{f,0}^S$ and $\{q_r''\}_{m,0}^S$ stand for the radial heat flux in the fuel and in the moderator, respectively, taking into account the static heat balance equation (see Eq. (4.6)):

$$S_f \{q_r''\}_{f,0}^S = V_f \left\langle q''' \right\rangle_{f,0} \tag{3.21}$$

  and the mass conservation equation (3.15)

$$\{\rho_m\}_0^+ \{v_m\}_0^+ = \{\rho_m\}_0^- \{v_m\}_0^- = \langle G_m \rangle_0 , \tag{3.22}$$

  Eq. (3.19) can be rewritten as:

$$\{h_m\}_0^+ = \{h_m\}_0^- + \frac{V_f \left\langle q''' \right\rangle_{f,0}}{A_m \langle G_m \rangle_0}. \tag{3.23}$$

In the last equation, $\{h_m\}_0^-$ in the first node is assumed to be known and equal to the core inlet enthalpy, i.e.

$$\{h_m\}_0^- = \{h_m\}_0^{inlet} \tag{3.24}$$

and the mass flux $\langle G_m \rangle_0$ is also known from input data. Solving Eq. (3.23) sequentially from the first node (core inlet) to the final node (core outlet), the three-dimensional distribution of the surface-averaged coolant/moderator enthalpy can be estimated.

- *coolant/moderator density*
  Next, the spatial distribution of the coolant/moderator density is computed. By assuming saturation conditions at all nodes, first the spatial distribution of the surface-averaged quality is estimated by using the following definition:

$$\{x\}_0 = \frac{\{h_m\}_0 - \{h\}_l}{\{h\}_v - \{h\}_l}, \tag{3.25}$$

  where $\{h\}_l$ and $\{h\}_v$ stand for the saturated liquid and vapor enthalpy, respectively. Then, the spatial distribution of the surface-averaged void fraction can be calculated as:

$$\{\alpha\}_0 = \frac{1}{1 + \frac{1-\{x\}_0}{\{x\}_0} \frac{\{\rho\}_v}{\{\rho\}_l} s}, \tag{3.26}$$

  where $\{\rho\}_l$ and $\{\rho\}_v$ designate the saturated liquid and vapor density, respectively. The slip ratio s is assumed to be given and taken from a commercial core simulator and is used to avoid the overestimation of the surface-averaged void fraction by using solely the Homogeneous Equilibrium Model (HEM). Thus, the coolant/moderator density can be estimated as a function of the void fraction, i.e.:

$$\{\rho_m\}_0 = \{\alpha\}_0 \{\rho\}_v + (1 - \{\alpha\}_0) \{\rho\}_l. \tag{3.27}$$

  It should be pointed out that Eq. (3.27) is applicable only for two-phase region. In case of single phase region, the coolant/moderator density can be estimated from water tables and the coolant/moderator enthalpy and pressure (the latter one is not known in advance, therefore in the first iteration it is assumed to be equal to core exit pressure).

- *coolant/moderator velocity*
  Once the spatial distribution of the coolant/moderator density is obtained, the spatial distribution of the coolant/moderator velocity can be found by solving Eq. (3.15) as

$$\{v_{z,m}\}_0^+ = \frac{\langle G_m \rangle_0}{\{\rho_m\}_0^+}. \tag{3.28}$$

  In the above, again the mass conservation equation (3.22) was used. From Eq. (3.28), the three- dimensional distribution of the surface-averaged coolant/moderator velocity can be calculated.

- coolant/moderator temperature
  Next, the spatial distribution of the coolant/moderator temperature can be calculated by using water tables and the coolant/moderator enthalpy and pressure. Since the pressure distribution is not yet known, it is assumed to be constant through the core and equal to core exit pressure taken from input data.

- *coolant/moderator pressure*
  Further, the spatial distribution of the surface-averaged pressure can be estimated by taking into account the fact that the coolant/moderator velocity and density have been computed in advance and thus are known quantities. As can be seen from Eq. (3.16) in order to evaluate the coolant/moderator pressure, the friction factor should be estimated a priori. In the present methodology, the friction coefficient (friction multiplier) $\langle F_M \rangle_0$ is defined as:

$$\langle F_M \rangle_0 = \langle f_{lo} \rangle_0 \langle \varphi_{lo} \rangle_0^2 , \tag{3.29}$$

  where the index lo indicates the two-phase flow considered as liquid, $\langle f_{lo} \rangle_0$ is the node-wise single-phase Fanning friction factor and φ_{lo} is the node-wise two-phase multiplier which becomes equal to unity in the single phase region. The single-phase friction factor is based on the McAdams correlation [11], which is given by:

$$\langle f_{lo} \rangle_0 = \frac{0.184}{\langle Re \rangle_0^{0.2}}, \quad \text{for} \quad 3 \cdot 10^4 \leq Re \leq 2 \cdot 10^6 \tag{3.30}$$

  with $Re = \frac{\langle G_m \rangle_0}{D_e \mu_l}$ standing for node-wise Reynolds number and the two-phase multiplier φ_{lo} is based on the Chisholm correlation [12,13]:

$$\langle \varphi_{lo} \rangle_0 = 1 + \left(\langle Y \rangle_0^2 - 1 \right) \times \left(\langle B \rangle_0 \times \left[\langle x \rangle_0 \right]^{\frac{2-l}{2}} + \left[1 - \langle x \rangle_0 \right]^{\frac{2-l}{2}} + \left[\langle x \rangle_0 \right]^{2-l} \right), \tag{3.31}$$

  where $l = 0.2$ and

$$\langle Y \rangle_0^2 = \frac{\rho_l}{\rho_v} \cdot \left(\frac{\mu_v}{\mu_l} \right)^{\frac{1}{4}}, \tag{3.32}$$

  μ_v and μ_l stand for the saturated liquid and vapor dynamic viscosity, respectively and the coefficient $\langle B \rangle_0$ is defined as:

  – for $\langle Y \rangle_0^2 \leq 9.5$:

$$\langle B \rangle_0 = \begin{cases} \langle B \rangle_0 = 4.8, & \text{if} \quad \langle G_m \rangle_0 \leq 500, \\ \langle B \rangle_0 = \frac{2400}{\langle G_m \rangle_0}, & \text{if} \quad 500 \leq \langle G_m \rangle_0 \leq 1900, \\ \langle B \rangle_0 = \frac{55}{\sqrt{\langle G_m \rangle_0}}, & \text{if} \quad \langle G_m \rangle_0 \geq 1900. \end{cases} \tag{3.33}$$

  – for $9.5 \leq \langle Y \rangle_0^2 \leq 28$:

$$\langle B \rangle_0 = \begin{cases} \frac{520}{\langle Y \rangle_0 \sqrt{\langle G_m \rangle_0}}, & \text{if} \quad \langle G_m \rangle_0 \leq 600, \\ \langle B \rangle_0 = \frac{21}{\langle G_m \rangle_0}, & \text{if} \quad \langle G_m \rangle_0 \geq 600. \end{cases} \tag{3.34}$$

  – for $\langle Y \rangle_0^2 \geq 28$:

$$\langle B \rangle_0 = \frac{15000}{\langle Y \rangle_0^2 \sqrt{\langle G_m \rangle_0}}. \tag{3.35}$$

  In the above, the mass flux $\langle G_m \rangle_0$ is taken in $kg \cdot m^{-2} \cdot s^{-1}$.

The correlation (3.31) is only valid for $\langle G_m \rangle_0 \geq 100\ kg \cdot m^{-2} \cdot s^{-1}$ and $\frac{\mu_l}{\mu_v} \geq 1000$.

Once the friction coefficient is known, the coolant/moderator pressure can be evaluated by solving Eq. (3.16) as:

$$\{P_z\}_0^- = \{P_z\}_0^+ + \left(\{v_{z,m}\}_0^+ - \{v_{z,m}\}_0^-\right)\{G_m\}_0 + \frac{\langle F_M\rangle_0\,\Delta z}{2D_e\rho_l}\langle G_m\rangle_0^2 - \frac{gV_m}{2A_m}\langle\rho_m\rangle_0\,. \tag{3.36}$$

Here, again the mass conservation equation (3.22) was applied. It should be noted that since only the pressure at the core outlet is usually known, the calculations of the surface-averaged coolant/moderator pressure have to be performed in the reverse order, i.e. starting from the last node (core exit) and continuing towards the first node (core inlet).

In the last series of equations, in order to close the system, it was decided to imply the following approximate relation between the surface-averaged (node interface) quantities and volume-averaged (node-averaged) quantities was assumed:

$$\langle X_m\rangle = \frac{\{X_m\}^+ + \{X_m\}^-}{2} \tag{3.37}$$

with X_m again designating any static thermo-hydraulic quantity.

Thus, the equations derived above provide the coupling mechanism between the themo-hydraulic quantities estimated at the interfaces of two pairs of consecutive nodes. If one of these two values is given, another one can be easily estimated from Eqs. (3.23), (3.25)-(3.27), (3.28) and (3.36).

The thorough analysis of the derived thermo-hydraulic equations (3.23), (3.25)-(3.27), (3.28) and (3.36) shows that they represent a strongly coupled system of equations and can not be resolved independently. It should be pointed out that such a strong coupling between different quantities is mainly due to the presence of the pressure balance equation. In the case of the absence of the pressure equation (or constant pressure), the remaining equations can be decoupled and solved independently from each other. Due to the presence of such a coupling in the current thermo-hydraulic model, the above equations can be solved only in an iterative manner. In the current model, the following iterative scheme for solving the steady-state thermo-hydraulic equations was developed:

1. First, the spatial distribution of the surface-averaged enthalpy is computed via Eq. (3.23).

2. Then, the spatial distributions of the surface-averaged coolant/moderator density and velocity are estimated from Eqs. (3.25)-(3.27), (3.28), assuming that at the first iteration all the core nodes have the same constant pressure equal to the core outlet pressure; in the consecutive iterations, the pressure calculated at the previous iteration is used.

3. Further, the spatial distribution of the surface-averaged coolant/moderator pressure is recomputed via Eq. (3.36) by using the thermo-hydraulic parameters calculated in the previous step.

4. The last two steps are repeated until the required convergence in the corresponding thermo-hydraulic quantities is achieved.

5. Finally, the spatial distribution of the surface-averaged coolant/moderator temperature is computed from water tables and the spatial distributions of the surface-averaged pressure and enthalpy obtained from earlier steps.

The main steps of the static thermo-hydraulic iterative scheme are summarized in Fig. 3.2.

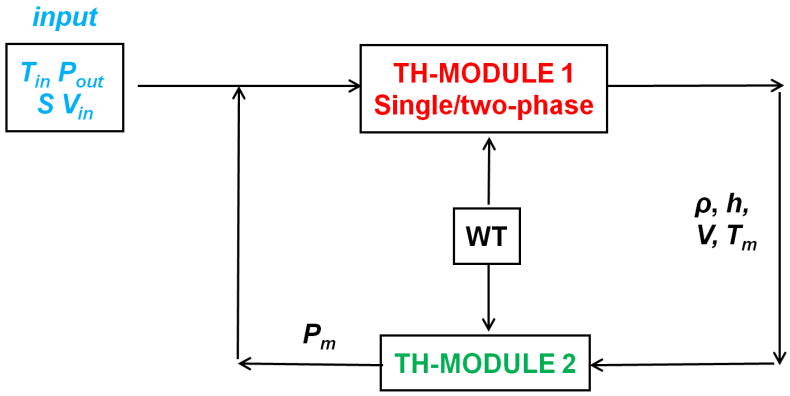

Figure 3.2: Iterative scheme for static thermo-hydraulic calculations (pressure calculations, "**WT**" stands for **W**ater **T**ables).

In Fig. 3.2, the iterative scheme used for the internal static thermo-hydraulic calculations (inner iterations), is shown. This loop is meant to solve the thermo-hydraulic problem, assuming that the neutron fluxes and cross-sections (i.e. power density) are given. As can be seen from Fig. 3.2, this scheme contains two separate thermo-hydraulic modules. The first one, designated as **TH-MODULE-1** (where "**TH**" stands for **T**hermo-**H**ydraulic) is based on Eqs. (3.23), (3.25)-(3.27), (3.28) and calculates the spatial distributions of the surface-averaged coolant/moderator density, velocity, enthalpy and temperature for a fixed (or given) pressure distribution. As an input for this module, the spatial distributions of the neutron fluxes and fission cross-sections (i.e. of the power density), the spatial distribution of the surface-averaged pressure and water tables are necessary. The second module, called **TH-MODULE-2**, is built on the basis of Eq. (3.36) and estimates the spatial distribution of the pressure throughout the core for fixed distributions of the other TH-parameters. As an input, one needs to provide water tables and the spatial distributions of required TH-quantities.

The convergence criteria in the present iterative scheme was imposed on the coolant/moderator density, velocity and pressure by requiring that the difference in any of these quantities between two consecutive iteration should be below certain values.

All nodal (i.e. volume-averaged) thermo-hydraulic quantities can be obtained from the surface-averaged parameters via Eq. (3.37).

3.3 Derivation of the dynamic thermo-hydraulic equations

In order to derive the driving equations for the thermo-hydraulic fluctuations (noise), the similar approach as described in Section 2.3 is applied, i.e. starting with the local time-

/space-dependent mass, momentum and enthalpy conservation equations given as:

$$\frac{\partial}{\partial t}\rho_m(\boldsymbol{r},t)+\nabla\cdot[\rho_m(\boldsymbol{r},t)\overline{v}_m(\boldsymbol{r},t)]=0, \tag{3.38}$$

$$\frac{\partial}{\partial t}[\rho_m(\boldsymbol{r},t)\overline{v}_m(\boldsymbol{r},t)]+\nabla\cdot[\rho_m(\boldsymbol{r},t)\overline{v}_m(\boldsymbol{r},t)\otimes\overline{v}_m(\boldsymbol{r},t)]=\nabla\cdot\overline{\overline{\tau}}(\boldsymbol{r},t)-\nabla P(\boldsymbol{r},t)+\overline{g}\rho_m(\boldsymbol{r},t), \tag{3.39}$$

$$\frac{\partial}{\partial t}[\rho_m(\boldsymbol{r},t)h_m(\boldsymbol{r},t)]+\nabla\cdot[\rho_m(\boldsymbol{r},t)\overline{v}_m(\boldsymbol{r},t)h_m(\boldsymbol{r},t)]=-\nabla\cdot\overline{q}''(\boldsymbol{r},t). \tag{3.40}$$

Here, we utilize the same set of approximations and notations as introduced in the static thermo-hydraulic equations. As has already been mentioned earlier, the full microscopic description of the two-phase flow is generally not possible (i.e. Eqs. (3.38)-(3.40) cannot be solved in both time and space in an exact manner). Therefore, the "macroscopic description" of the two-phase flow quantities will be used.

3.3.1 Algorithm used for the spatial discretisation

All flow properties are first discretized and then averaged in space on the relevant space volumes. The spatial homogenization of Eqs. (3.38)-(3.40) is defined as:

$$\frac{\partial}{\partial t}\int_{V_m}\rho_m(\boldsymbol{r},t)dV+\int_{V_m}\nabla\cdot[\rho_m(\boldsymbol{r},t)\overline{v}_m(\boldsymbol{r},t)]dV=0, \tag{3.41}$$

$$\begin{aligned}\frac{\partial}{\partial t}\int_{V_m}[\rho_m(\boldsymbol{r},t)\overline{v}_m(\boldsymbol{r},t)]dV+\int_{V_m}\nabla\cdot[\rho_m(\boldsymbol{r},t)\overline{v}_m(\boldsymbol{r},t)\otimes\overline{v}_m(\boldsymbol{r},t)]dV\\ =\int_{V_m}\nabla\cdot\overline{\overline{\tau}}(\boldsymbol{r},t)dV-\int_{V_m}\nabla P(\boldsymbol{r},t)dV+\int_{V_m}\overline{g}\rho_m(\boldsymbol{r},t)dV,\end{aligned} \tag{3.42}$$

$$\frac{\partial}{\partial t}\int_{V_m}[\rho_m(\boldsymbol{r},t)h_m(\boldsymbol{r},t)]dV+\int_{V_m}\nabla\cdot[\rho_m(\boldsymbol{r},t)\overline{v}_m(\boldsymbol{r},t)h_m(\boldsymbol{r},t)]=-\int_{V_m}\nabla\cdot\overline{q}''(\boldsymbol{r},t)dV. \tag{3.43}$$

Then, introducing the following notations for the volume-and area-averaged node-wise time-dependent quantities as compared to the static case (see Eqs. (3.7)-(3.14)):

$$\langle X_m\rangle(t)=\frac{1}{V_m}\int_{V_m}X_m(\boldsymbol{r},t)dV, \tag{3.44}$$

$$\{\rho_m\}(t)=\frac{1}{A_m}\int_{A_m}\rho_m(\boldsymbol{r},t)dA, \tag{3.45}$$

$$\{P_z\}(t)=\frac{1}{A_m}\int_{A_m}P_z(\boldsymbol{r},t)dA, \tag{3.46}$$

$$\left\{q_r''\right\}_m^S(t)=\frac{1}{S_m}\int_{S_m}q_r''(\boldsymbol{r},t)dS, \tag{3.47}$$

$$\langle v_{z,m}\rangle(t)=\frac{\langle\rho_m v_{z,m}\rangle(t)}{\langle\rho_m\rangle(t)}, \tag{3.48}$$

$$\langle h_m\rangle(t)=\frac{\langle\rho_m h_m\rangle(t)}{\langle\rho_m\rangle(t)}, \tag{3.49}$$

$$\{h_m\}^{\pm}(t)=\frac{\{\rho_m h_m\}^{\pm}(t)}{\{\rho_m\}^{\pm}(t)}, \tag{3.50}$$

$$\{\hat{h}_m\}^{\pm}(t) = \frac{\{\rho_m v_{z,m} h_m\}^{\pm}(t)}{\{\rho_m v_{z,m}\}^{\pm}(t)} \tag{3.51}$$

assuming one-dimensional upward vertical flow and taking into account Gauss' divergence theorem, Eqs. (3.41)-(3.43) can be simplified as:

$$V_m \frac{\partial}{\partial t} \langle \rho_m \rangle (t) + A_m \left(\{\rho_m\}^{+}(t) \{v_{z,m}\}^{+}(t) - \{\rho_m\}^{-}(t) \{v_{z,m}\}^{-}(t) \right) = 0, \tag{3.52}$$

$$V_m \frac{\partial}{\partial t} \langle \rho_m \rangle (t) \langle v_{z,m} \rangle (t) + A_m \left(\{\rho_m\}^{+}(t) \left[\{v_{z,m}\}^{+}\right]^2 (t) - \{\rho_m\}^{-}(t) \left[\{v_{z,m}\}^{-}\right]^2 (t) \right) =$$
$$- \frac{\langle F_M \rangle_0 V_m}{2 D_e \rho_l} \langle \rho_m \rangle^2 (t) \langle v_m \rangle^2 (t) - A_m \left[\{P_m\}^{+}(t) - \{P_m\}^{-}(t) \right] - g V_m \langle \rho_m \rangle (t), \tag{3.53}$$

$$V_m \frac{\partial}{\partial t} \langle \rho_m \rangle (t) \langle h_m \rangle (t) + A_m \left(\{\rho_m\}^{+}(t) \{v_{z,m}\}^{+}(t) \{h_m\}^{+}(t) - \right.$$
$$\left. \{\rho_m\}^{-}(t) \{v_{z,m}\}^{-}(t) \{h_m\}^{-}(t) \right) = -S_m \{q_r''\}_m^S(t). \tag{3.54}$$

In the above calculations, it is assumed that $\{\hat{h}_m\}^{\pm} \approx \{h_m\}^{\pm}$. It should be underlined that the term on the right-hand side of equation (3.54) requires a special treatment, namely it should be expressed through the known quantities. For this purpose, it is assumed that all radial heat produced in the fuel region of an assembly (or node) is directly and instantaneously transferred into the coolant region, i.e. the following approximation can be written:

$$S_m \{q_r''\}_m^S(t) = -S_f \{q_r''\}_f^S(t). \tag{3.55}$$

Then, using Fourier's law of heat conduction written as:

$$\overline{q}''(\boldsymbol{r}, t) = -k_f(T_f) \nabla T_f(\boldsymbol{r}, t) \tag{3.56}$$

after spatial averaging and discretisation, the area-averaged heat flux in the fuel region can be approximated as:

$$\{q_r''\}_f(t) \approx \langle H \rangle_0 \left(\langle T_f \rangle (t) - \langle T_m \rangle (t) \right), \tag{3.57}$$

where $\langle H \rangle_0$ stands for the "artificial" node-wise static heat transfer coefficient ($\langle H \rangle_0$ differs from the real one since one does not use the wall temperature), $\langle T_f \rangle$ and $\langle T_m \rangle$ designate the volume-averaged fuel and coolant/moderator temperature. The method for evaluating the heat transfer coefficient $\langle H \rangle_0$ will be discussed later on in Section 4.2.1. Further, combining Eqs. (3.55) and (3.57) for the heat flux in the moderator region times the lateral surface area of a moderator node, one gets:

$$S_m \{q_r''\}_m(t) = S_f \langle H \rangle_0 \left(\langle T_f \rangle (t) - \langle T_m \rangle (t) \right). \tag{3.58}$$

To obtain the equations for the fluctuations, the same procedure (i.e. first order perturbation theory) as for the neutron noise equations, is applied, i.e. splitting all time-dependent quantities in Eqs. (3.52)-(3.54) into their mean values and fluctuating parts as:

$$\{X_m\}^{\pm}(t) = \{X_m\}_0^{\pm} + \{\delta X_m\}^{\pm}(t), \tag{3.59}$$

$$\langle X_m \rangle (t) = \langle X_m \rangle_0 + \langle \delta X_m \rangle (t), \tag{3.60}$$

neglecting the second order terms, subtracting the static equations (3.27), (3.28) and (3.36), performing a Fourier transform and assuming the following approximate relation between the area-averaged and the volume-averaged values:

$$\langle X_m \rangle (t) = \frac{\{X_m\}^+(t) + \{X_m\}^-(t)}{2} \tag{3.61}$$

with X_m designating any time-dependent thermo-hydraulic quantity, the noise equations for calculating the thermo-hydraulic fluctuations read as:

$$V_m i\omega \langle \delta\rho_m \rangle (\omega) + A_m \left(\{\delta\rho_m\}^+(\omega)\{v_{z,m}\}_0^+ - \{\delta\rho_m\}^-(\omega)\{v_{z,m}\}_0^- + \right.$$
$$\left. \{\rho_m\}_0^+ \{\delta v_{z,m}\}^+(\omega) - \{\rho_m\}_0^- \{\delta v_{z,m}\}^-(\omega) \right) = 0, \tag{3.62}$$

$$\frac{V_m i\omega}{2} \left(\langle \rho_m \rangle_0 \{\delta v_{z,m}\}^+(\omega) + \langle \rho_m \rangle_0 \{\delta v_{z,m}\}^-(\omega) + \{\delta\rho_m\}^+(\omega) \langle v_{z,m} \rangle_0 + \right.$$
$$\left. \{\delta\rho_m\}^-(\omega) \langle v_{z,m} \rangle_0 \right) + A_m \left(\{\delta\rho_m\}^+(\omega) \left[\{v_{z,m}\}_0^+\right]^2 - \{\delta\rho_m\}^-(\omega) \left[\{v_{z,m}\}_0^-\right]^2 \right.$$
$$\left. + 2\{\rho_m\}_0^+ \{v_{z,m}\}_0^+ \{\delta v_{z,m}\}^+(\omega) - 2\{\rho_m\}_0^- \{v_m\}_0^- \{\delta v_{z,m}\}^-(\omega) \right)$$
$$+ \frac{\langle F_M \rangle_0 \langle G_m \rangle_0 V_m}{2 D_e \rho_l} \left(\{\delta\rho_m\}^+(\omega) \langle v_{z,m} \rangle_0 + \{\delta\rho_m\}^-(\omega) \langle v_{z,m} \rangle_0 + \right.$$
$$\left. \langle \rho_m \rangle_0 \{\delta v_{z,m}\}^-(\omega) + \langle \rho_m \rangle_0 \{\delta v_{z,m}\}^+(\omega) \right) + A_m \left[\{\delta P_m\}^+(\omega) - \{\delta P_m\}^-(\omega) \right]$$
$$+ \frac{g V_m}{2} \left(\{\delta\rho_m\}^+(\omega) + \{\delta\rho_m\}^-(\omega) \right) = 0, \tag{3.63}$$

$$\frac{V_m i\omega}{2} \left(\langle h_m \rangle_0 \left[\{\delta\rho_m\}^+(\omega) + \{\delta\rho_m\}^-(\omega) \right] + \langle \rho_m \rangle_0 \left[\{\delta h_m\}^+(\omega) + \{\delta h_m\}^-(\omega) \right] \right)$$
$$+ A_m \left(\{h_m\}_0^+ \{\rho_m\}_0^+ \{\delta v_{z,m}\}^+(\omega) + \{h_m\}_0^+ \{\delta\rho_m\}^+(\omega) \{v_{z,m}\}_0^+ + \right.$$
$$\{\delta h_m\}^+(\omega) \{\rho_m\}_0^+ \{v_{z,m}\}_0^+ - \{h_m\}_0^- \{\rho_m\}_0^- \{\delta v_{z,m}\}^-(\omega) - \{h_m\}_0^- \{\delta\rho_m\}^-(\omega) \times$$
$$\left. \{v_{z,m}\}_0^- - \{\delta h_m\}^-(\omega) \{\rho_m\}_0^- \{v_{z,m}\}_0^- \right) = V_f \langle H_{eff} \rangle_0 \left(\langle \delta T_f \rangle (\omega) - \langle \delta T_m \rangle (\omega) \right), \tag{3.64}$$

where "i" stands for imaginary unit and $\langle H_{eff} \rangle_0$ is the effective node-wise heat transfer coefficient. $\langle H_{eff} \rangle_0$ will be defined later on in Section 4.2.1. Equations (3.62)-(3.64) are then used to calculate the space-frequency distributions of the fluctuations in coolant/moderator density, velocity and pressure. More details on such calculations will be given in the next Section 3.3.2.

3.3.2 Algorithm used for solving dynamic thermo-hydraulic equations

The detailed analysis of the derived equations (3.62)-(3.64) shows that this set of noise equations is not fully closed, i.e. the number of unknown quantities exceeds the number of available equations. Therefore, additional correlations between different thermo-hydraulic noise quantities are needed and will be specified in this Section. Such correlations will then help to simplify the corresponding set of noise equations as well as to reduce the number of unknown parameters to be directly solved for.

Thermo-hydraulic correlations

First, it is assumed that the enthalpy fluctuations can be represented as a linear combination of the fluctuations in the coolant/moderator pressure and density as:

$$\{\delta h_m\}^{\pm}(\omega) = \alpha^{\pm}\{\delta P_m\}^{\pm}(\omega) + \beta^{\pm}\{\delta \rho_m\}^{\pm}(\omega), \tag{3.65}$$

where the coefficient $\alpha^{\pm}$ and $\beta^{\pm}$ stand for the coolant/moderator enthalpy derivatives with respect to the coolant/moderator pressure and density and defined correspondingly as:

$$\alpha^{\pm} = \frac{\partial\{h_m\}^{\pm}}{\partial\{P_{z,m}\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const} = \frac{dh_l^{\pm}}{dP_m^{\pm}}\bigg|_{\{\hat{\rho}_m\}^{\pm}=const} + \rho_v^{\pm}s^{\pm}\left(\frac{dh_v^{\pm}}{dP_m^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const} - \right.$$

$$\left.\frac{dh_l^{\pm}}{dP_m^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const}\right)\left(\frac{\{\rho_m\}_0^{\pm} - \rho_l^{\pm}}{\rho_l^{\pm}\rho_v^{\pm}(1-s^{\pm}) + \{\rho_m\}_0^{\pm}(s^{\pm}\rho_v^{\pm} - \rho_l^{\pm})}\right) + (h_v^{\pm} - h_l^{\pm})\rho_v^{\pm}s^{\pm}\times$$

$$\left(-\frac{\frac{d\rho_l^{\pm}}{d\{P_m\}^{\pm}}\Big|_{\{\rho_m\}^{\pm}=const}}{\rho_l^{\pm}\rho_v^{\pm}(1-s^{\pm}) + \{\hat{\rho}_m\}_0^{\pm}(s^{\pm}\rho_v^{\pm} - \rho_l^{\pm})} - \frac{\left(\{\rho_m\}_0^{\pm} - \rho_l^{\pm}\right)}{(\rho_l^{\pm}\rho_v^{\pm}(1-s^{\pm}) + \{\rho_m\}_0^{\pm}(s^{\pm}\rho_v^{\pm} - \rho_v^{\pm}))^2}\times\right.$$

$$\left(\rho_v^{\pm}(1-s^{\pm})\,\frac{d\rho_l^{\pm}}{d\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const} + \rho_l^{\pm}(1-s^{\pm})\frac{d\rho_v^{\pm}}{d\{P_m\}^{\pm}}\,|_{\{\rho_m\}^{\pm}=const} + \right.$$

$$\left.\left.\{\rho_m\}_0^{\pm}\left(s^{\pm}\,\frac{d\rho_v^{\pm}}{d\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const} - \frac{d\rho_l^{\pm}}{d\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const}\right)\right)\right) + s^{\pm}\times$$

$$\frac{(h_v^{\pm} - h_l^{\pm})\left(\{\hat{\rho}_m\}_0^{\pm} - \rho_l^{\pm}\right)}{\rho_l^{\pm}\rho_v^{\pm}(1-s^{\pm}) + \{\hat{\rho}_m\}_0^{\pm}(s^{\pm}\rho_v^{\pm} - \rho_l^{\pm})} \times \frac{d\rho_v^{\pm}}{d\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const}, \tag{3.66}$$

$$\beta^{\pm} = \frac{\partial\{h_m\}^{\pm}}{\partial\{\rho_m\}^{\pm}}\bigg|_{\{P_m\}^{\pm}=const} = \frac{s^{\pm}\rho_v^{\pm}\rho_l^{\pm}(h_v^{\pm} - h_l^{\pm})(\rho_v^{\pm} - \rho_l^{\pm})}{\left(\rho_l^{\pm}\rho_v^{\pm}(1-s^{\pm}) + \{\rho_m\}_0^{\pm}(s^{\pm}\rho_v^{\pm} - \rho_l^{\pm})\right)}^2, \tag{3.67}$$

where $s^{\pm}$ is the interfacial slip ratio, $\rho_l^{\pm}$, $\rho_v^{\pm}$, $h_l^{\pm}$ and $h_v^{\pm}$ are saturated densities and enthalpies of the liquid and vapour phases taken at the interfaces of the nodes, respectively, and their corresponding derivatives calculated from water tables at steady-state conditions are defined as:

$$\frac{d\rho_l^{\pm}}{d\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const} = \frac{\delta\rho_l^{\pm}}{\delta\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const}, \tag{3.68}$$

$$\frac{d\rho_v^{\pm}}{d\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const} = \frac{\delta\rho_v^{\pm}}{\delta\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const}, \tag{3.69}$$

$$\frac{dh_l^{\pm}}{d\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const} = \frac{\delta h_l^{\pm}}{\delta\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const}, \tag{3.70}$$

$$\frac{dh_v^{\pm}}{d\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const} = \frac{\delta h_v^{\pm}}{\delta\{P_m\}^{\pm}}\bigg|_{\{\rho_m\}^{\pm}=const}. \tag{3.71}$$

In general, the explicit expressions for the respective derivatives $\alpha^{\pm}$ and $\beta^{\pm}$ can be obtained from the following generic equation (derived from Eqs. (3.25)-(3.27) written for dynamic quantities):

$$\{h_m\}^{\pm} = h_l^{\pm} + \frac{(h_v^{\pm} - h_l^{\pm})\left(\{\rho_m\}_0^{\pm} - \rho_l^{\pm}\right)\rho_v^{\pm} s^{\pm}}{\rho_l^{\pm}\rho_v^{\pm}(1-s^{\pm}) + \{\rho_m\}_0^{\pm}(s^{\pm}\rho_v^{\pm} - \rho_l^{\pm})} \tag{3.72}$$

by taking the partial derivatives with respect to coolant/moderator pressure or density. The complexity of such expressions is mainly due to the presence of two phases where the drift flux model or slip model needs to be applied. However, for the single phase region, Eqs. (3.66)-(3.67) can be simplified as:

$$\alpha^{\pm} = \left.\frac{\partial\{h_m\}^{\pm}}{\partial\{P_m\}^{\pm}}\right|_{\{\rho_m\}^{\pm}=const} \approx \left.\frac{\delta\{h_m\}^{\pm}}{\delta\{P_m\}^{\pm}}\right|_{\{\rho_m\}^{\pm}=const}, \tag{3.73}$$

$$\beta^{\pm} = \left.\frac{\partial\{h_m\}^{\pm}}{\partial\{P_m\}^{\pm}}\right|_{\{\rho_m\}^{\pm}=const} \approx \left.\frac{\delta\{h_m\}^{\pm}}{\delta\{P_m\}^{\pm}}\right|_{\{\rho_m\}^{\pm}=const}. \tag{3.74}$$

Further, a similar linear approximation is applied to model the fluctuations in the coolant/moderator temperature which, as in the case of the enthalpy, is assumed to be a function of the coolant/moderator pressure and density and thus can be expressed as:

$$\{\delta T_m\}^{\pm}(\omega) = \gamma^{\pm}\{\delta\rho_m\}^{\pm}(\omega) + \theta^{\pm}\{\delta P_m\}^{\pm}(\omega), \tag{3.75}$$

where the coefficients $\gamma^{\pm}$ and $\theta^{\pm}$ represent the respective derivatives of the coolant/moderator temperature with respect to the coolant/moderator pressure and density, given as:

$$\gamma^{\pm} = \left.\frac{\partial\{T_m\}^{\pm}}{\partial\{\rho_m\}^{\pm}}\right|_{\{P_m\}^{\pm}=const} = \left.\frac{\delta\{T_m\}^{\pm}}{\delta\{\rho_m\}^{\pm}}\right|_{\{P_m\}^{\pm}=const}, \tag{3.76}$$

$$\theta^{\pm} = \left.\frac{\partial\{T_m\}^{\pm}}{\partial\{P_m\}^{\pm}}\right|_{\{\rho_m\}^{\pm}=const} = \left.\frac{\delta\{T_m\}^{\pm}}{\delta\{P_m\}^{\pm}}\right|_{\{\rho_m\}^{\pm}=const}. \tag{3.77}$$

$\gamma^{\pm}$ and $\theta^{\pm}$ are again estimated from thermo-hydraulic tables at steady-state conditions. Since the coolant/moderator temperature becomes constant in the two phase region , i.e. it is equal to the saturation temperature, $\gamma^{\pm}$ and $\theta^{\pm}$ are calculated only for the single-phase region and are assumed to be zero otherwise. It should also be underlined that all linear coefficients $\alpha^{\pm}$, $\beta^{\pm}$, $\gamma^{\pm}$ and $\theta^{\pm}$ are calculated for each node separately and thus are unique. The estimations of these coefficients are performed with the help of water tables by introducing a small 1% (from the respective node-wise mean value) perturbation, either in the coolant/moderator pressure for the constant coolant/moderator density or in the coolant/moderator density for the constant coolant/moderator pressure and then evaluating the induced fluctuations in the coolant/moderator enthalpy or temperature. Here, it should be pointed out that the assumed linear dependence between the fluctuations in the coolant/moderator enthalpy/temperature and the ones in the coolant/moderator pressure/density is valid only for small stationary fluctuations and cannot be applied to non-stationary or transient conditions. In the present study, where first-order perturbation theory is utilized, such an assumption of linearity holds rather well. In addition, for stationary processes, the dependence between the coolant/moderator enthalpy and temperature on the pressure and density can be assumed quite smooth, thus justifying the assumption mentioned above.

Thereafter, the set of thermo-hydraulic noise equations derived in the previous section are solved to obtain explicit expressions for the noise in the coolant/moderator velocity, density and pressure. Thus, combining Eqs. (3.62)-(3.64) with Eqs. (3.65), (3.75) and (4.20) (the equation for the fluctuations in the fuel temperature which will be derived later on in Section 4.2.1), one gets:

- *noise in the coolant/moderator density*

$$\{\delta\rho_m\}^+(\omega) = -\frac{1}{\widetilde{e} + \widetilde{g}\beta^+ - \widetilde{k}\frac{\widetilde{a}}{\widetilde{c}} + \frac{H_{eff}V_f}{2}\frac{i\omega\tau_f}{1+i\omega\tau_f}\gamma^+}\left[\{\delta\rho_m\}^-(\omega)\times\right.$$
$$\left(\widetilde{f} + \widetilde{h}\beta^- - \widetilde{k}\frac{\widetilde{b}}{\widetilde{c}}\frac{\langle H_{eff}\rangle_0 V_f}{2}\frac{i\omega\tau_f}{1+i\omega\tau_f}\gamma^-\right) + \{\delta P_m\}^+(\omega)\left(\widetilde{g}\alpha^+ + \right.$$
$$\left.\frac{\langle H_{eff}\rangle_0 V_f}{2}\frac{i\omega\tau_f}{1+i\omega\tau_f}\theta^+\right) + \{\delta P_m\}^-(\omega)\left(\widetilde{h}\alpha^- + \frac{\langle H_{eff}\rangle_0 V_f}{2}\frac{i\omega\tau_f}{1+i\omega\tau_f}\theta^-\right) +$$
$$\left.\{\delta v_{z,m}\}^-(\omega)\left(\widetilde{l} - \widetilde{k}\frac{\widetilde{d}}{\widetilde{c}}\right) - \frac{V_f\langle\delta q'''\rangle_f(\omega)}{1+i\omega\tau_f}\right] \tag{3.78}$$

with the node-wise coupling coefficients $\widetilde{e}$, $\widetilde{f}$, $\widetilde{g}$, $\widetilde{h}$, $\widetilde{k}$ and $\widetilde{q}$ defined as:

$$\widetilde{e} = \frac{i\omega V_m}{2}\langle h_m\rangle_0 + A_m\{v_{z,m}\}_0^+\{h_m\}_0^+, \tag{3.79}$$

$$\widetilde{f} = \frac{i\omega V_m}{2}\langle h_m\rangle_0 - A_m\{v_{z,m}\}_0^-\{h_m\}_0^-, \tag{3.80}$$

$$\widetilde{g} = \frac{i\omega V_m}{2}\langle \rho_m\rangle_0 + A_m\{\rho_m\}_0^+\{v_{z,m}\}_0^+, \tag{3.81}$$

$$\widetilde{h} = \frac{i\omega V_m}{2}\langle \rho_m\rangle_0 - A_m\{\rho_m\}_0^-\{v_{z,m}\}_0^-, \tag{3.82}$$

$$\widetilde{k} = A_m\{\rho_m\}_0^+\{h_m\}_0^+, \tag{3.83}$$

$$\widetilde{q} = -A_m\{\rho_m\}_0^-\{h_m\}_0^-, \tag{3.84}$$

where τ_f stands for the node-wise fuel time constant and will be defined in Section 4.2.1).

Assuming the external fluctuations in the coolant/moderator density at the core inlet are zero, taking into account the information about the noise in the surface-averaged coolant/moderator pressure (the latter one is not known in advance and, therefore, it is assumed to be equal to the initial fluctuation in the core exit pressure throughout the entire core and assumed to be taken from input data) and solving Eq. (3.78) sequentially from the first node (core inlet) to the final node (core outlet), the three-dimensional distribution of the noise in the surface-averaged coolant/moderator density can be estimated.

- *noise in the coolant/moderator velocity*

$$\{\delta v_{z,m}\}^+(\omega) = -\frac{1}{\widetilde{c}}\left(\{\delta\rho_m\}^+(\omega)\widetilde{a} + \{\delta\rho_m\}^-(\omega)\widetilde{b} + \{\delta v_{z,m}\}^-(\omega)\widetilde{d}\right) \tag{3.85}$$

with the node-wise coupling coefficients $\widetilde{a}$, $\widetilde{b}$, $\widetilde{c}$ and $\widetilde{d}$ specified as:

$$\widetilde{a} = \frac{i\omega V_m}{2} + A_m\{v_{z,m}\}_0^+, \tag{3.86}$$

$$\widetilde{b} = \frac{i\omega V_m}{2} - A_m\{v_{z,m}\}_0^-, \tag{3.87}$$

$$\widetilde{c} = A_m\{\rho_m\}_0^+, \tag{3.88}$$

$$\widetilde{d} = -A_m\{\rho_m\}_0^-. \tag{3.89}$$

Taking into account the information about the initial perturbation in the surface-averaged coolant/moderator velocity at the core inlet as well as assuming that the fluctuations in the coolant/moderator density were calculated in advance, the noise in the coolant/moderator velocity can be calculated from Eq. (3.85).

- *noise in the coolant/moderator pressure*

$$\{\delta P_m\}^-(\omega) = \frac{1}{\widetilde{r}}\left(\{\delta\rho_m\}^+(\omega)\widetilde{m} + \{\delta\rho_m\}^-(\omega)\widetilde{n} + \{\delta v_{z,m}\}^+(\omega)\widetilde{o} + \right.$$
$$\left.\{\delta v_{z,m}\}^-(\omega)\widetilde{p} + \{\delta P_m\}^+(\omega)\widetilde{q}\right) \tag{3.90}$$

with the node-wise coupling coefficients $\widetilde{m}$, $\widetilde{n}$, $\widetilde{o}$, $\widetilde{p}$, $\widetilde{q}$ and $\widetilde{r}$ defined as:

$$\widetilde{m} = \frac{i\omega V_m}{2}\langle v_{z,m}\rangle_0 + A_m\left[\{v_{z,m}\}_0^+\right]^2 + \frac{\langle F_M\rangle_0 V_m}{2D_e\rho_l}\langle G_m\rangle_0\langle v_{z,m}\rangle_0 + \frac{gV_m}{2}, \tag{3.91}$$

$$\widetilde{n} = \frac{i\omega V_m}{2}\langle v_{z,m}\rangle_0 + A_m\left[\{v_{z,m}\}_0^-\right]^2 + \frac{\langle F_M\rangle_0 V_m}{2D_e\rho_l}\langle G_m\rangle_0\langle v_{z,m}\rangle_0 + \frac{gV_m}{2}, \tag{3.92}$$

$$\widetilde{o} = \frac{i\omega V_m}{2}\langle\rho_m\rangle_0 + 2A_m\{G_m\}_0^+ + \frac{\langle F_M\rangle_0 V_m}{2D_e\rho_l}\langle G_m\rangle_0\langle\rho_m\rangle_0, \tag{3.93}$$

$$\widetilde{p} = \frac{i\omega V_m}{2}\langle\rho_m\rangle_0 - 2A_m\{G_m\}_0^- + \frac{\langle F_M\rangle_0 V_m}{2D_e\rho_l}\langle G_m\rangle_0\langle\rho_m\rangle_0, \tag{3.94}$$

$$\widetilde{q} = A_m, \tag{3.95}$$

$$\widetilde{r} = -A_m. \tag{3.96}$$

Using the information about the noise distribution in both surface-averaged coolant/moderator density and velocity, obtained from the previous two steps and taking into account the initial perturbation in the core exit pressure given as input data, the noise distribution in the surface-averaged coolant/moderator pressure can be computed from Eq. (3.90). Again, similarly to the static calculations, the noise calculations are performed in the reverse order, i.e. from the last node (core outlet) to the first node (core inlet) since only the pressure noise at the core outlet is usually known.

- *noise in the coolant/moderator enthalpy*
The noise distribution in the surface-averaged coolant/moderator enthalpy is estimated from Eq. (3.65) assuming that the noise distributions in the coolant/moderator pressure and density were calculated from the previous steps.

- *noise in the coolant/moderator temperature (in the one-phase region)*
  The noise distribution in the surface-averaged coolant/moderator temperature is calculated from Eq. (3.75) taking into account the information about the noise distribution in the surface-averaged coolant/moderator pressure and density obtained from earlier steps.

The equations derived above provide the coupling between the fluctuations estimated at the interfaces of two pairs of consecutive nodes. If one of these two fluctuations is given, the other one can be easily estimated from Eqs. (3.65), (3.75), (3.78), (3.85) and (3.90). Thus, providing an initial perturbation at the core inlet (or outlet) for any quantity of interest, the corresponding noise in all other nodes and quantities can be consecutively calculated. However, a detailed analysis of Eqs. (3.65), (3.75), (3.78), (3.85) and (3.90) shows that these equations cannot be solved independently from each other due to a nonlinear coupling between the different equations caused by the inclusion of pressure noise calculations. Therefore, the corresponding solution can be estimated only in an iterative manner. For this purpose, similarly to the static case, the following iterative scheme for solving noise thermo-hydraulic equations was introduced:

1. First, the space-frequency distributions of the noise in the surface-averaged coolant/moderator density and velocity are estimated from Eqs. (3.78), (3.85), respectively, assuming that at the first iteration all the core nodes have the same (axially constant) perturbation in the surface-averaged pressure equal to the initial pressure perturbation at the core outlet; in the consecutive iterations, the noise distribution in the coolant/moderator pressure defined at the previous iteration is used.

2. Further, the space-frequency distribution of the noise in the surface-averaged coolant/moderator pressure is recomputed via Eq. (3.90) by using the noise distributions in other thermo-hydraulic parameters (i.e. noise in the coolant/moderator density and velocity) calculated in the previous step.

3. The last two steps are repeated until the required convergence in the noise in the corresponding surface-averaged thermo-hydraulic quantities is achieved.

4. Then, the noise in the surface-averaged coolant/moderator enthalpy and temperature is calculated via Eqs. (3.65) and (3.75), respectively.

The main steps of the dynamic thermo-hydraulic iterative scheme are summarized in Fig. 3.3.

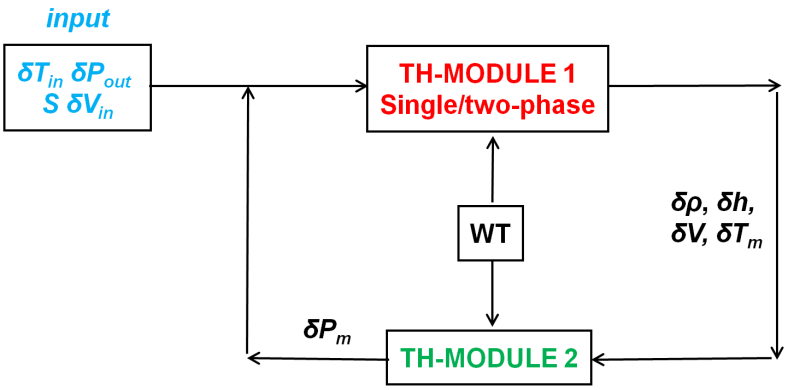

Figure 3.3: Internal thermo-hydraulic dynamic calculations ("**WT**" stands for **W**ater **T**ables).

In Fig. 3.3, the iterative scheme used for the internal thermo-hydraulic noise calculations (inner iterations), is shown. This loop is meant to solve the thermo-hydraulic problem, assuming that the neutron noise (and thus the noise in the power density) is given. As can be seen from Fig. 3.3, this scheme contains two separate thermo-hydraulic modules. The first one, designated as **TH-MODULE-1** is based on Eqs. (3.65), (3.75), (3.78), (3.85) and calculates the space-frequency distributions of the noise in the surface-averaged coolant/moderator density, velocity, enthalpy and temperature for a fixed (or given) pressure noise distribution (i.e. independently of the pressure noise). As an input for this module, the initial TH-perturbations (i.e. the corresponding boundary conditions for the noise in the coolant inlet velocity, temperature and core exit pressure), neutron noise, pressure noise and water tables are necessary. The second module, called **TH-MODULE-2**, is built on the basis of Eq. (3.90) and estimates the space-frequency distribution of the noise in the pressure for the fixed noise distribution in other TH-parameters (coolant/moderator velocity and density). As an input, one needs to provide water tables, boundary conditions for the pressure noise at the core exit and the noise distributions in other TH-quantities. The convergence criteria in the present iterative scheme was imposed on the noise in the coolant/moderator density, velocity and pressure by requiring that the difference in any of these noise quantities between two consecutive iteration should be below certain values.

Chapter 4

HEAT-TRANSFER MODELS

4.1 Introduction

In this section, the heat-transfer models implemented in the tool are presented. The tool has the ability to calculate the solution to static problems, as well as the solution to dynamic problems in linear theory and in the frequency domain. The equations thus solved in these different cases are presented here.

First, the static heat-transfer module of the tool and the respective equations it is based on are described. The corresponding numerical algorithms developed and utilized in the module are also discussed. Next, the numerical scheme used for spatially discretizing the static heat-transfer equations is described. Thereafter, the numerical algorithms allowing solving the static equations are summarized.

Further, the dynamic heat-transfer module of the tool together with the respective equations are presented. Following the same structure as in the static case, the corresponding numerical algorithms developed and utilized in the module (including both the spatial discretization scheme and the numerical algorithms used for solving the dynamic equations) are touched upon.

The present heat-transfer model is based on a generic heat conduction equation written for the fuel region.

4.2 Derivation of the static heat-transfer equations

The static heat-transfer equation in three dimensions reads as:

$$-\nabla \cdot \overline{q}''(\boldsymbol{r}) + q'''(\boldsymbol{r}) = 0, \tag{4.1}$$

where $\overline{q}''(\boldsymbol{r})$ stands for the heat flux and $q'''(\boldsymbol{r})$ designates the volumetric heat source.

4.2.1 Algorithm used for the spatial discretisation

To obtain the discretized heat transfer equations one first integrates both sides of Eq.(4.1) over the fuel volume of a node and obtains

$$-\int_{V_f} \nabla \cdot \overline{q}''(\boldsymbol{r}) dV + \int_{V_f} q'''(\boldsymbol{r}) dV = 0. \tag{4.2}$$

The different notations and conventions used in the discretization and averaging procedure are highlighted in Fig. 4.1. In the figure, the horizontal arrows outwards a fuel node designate the direction of the heat generation (heat flux) whereas the (I, J, K) indexes denote the spatial position of a fuel/moderator node. In the present model it is assumed that the fuel and coolant/moderator regions (nodes) can be separated and thus are modelled independently.

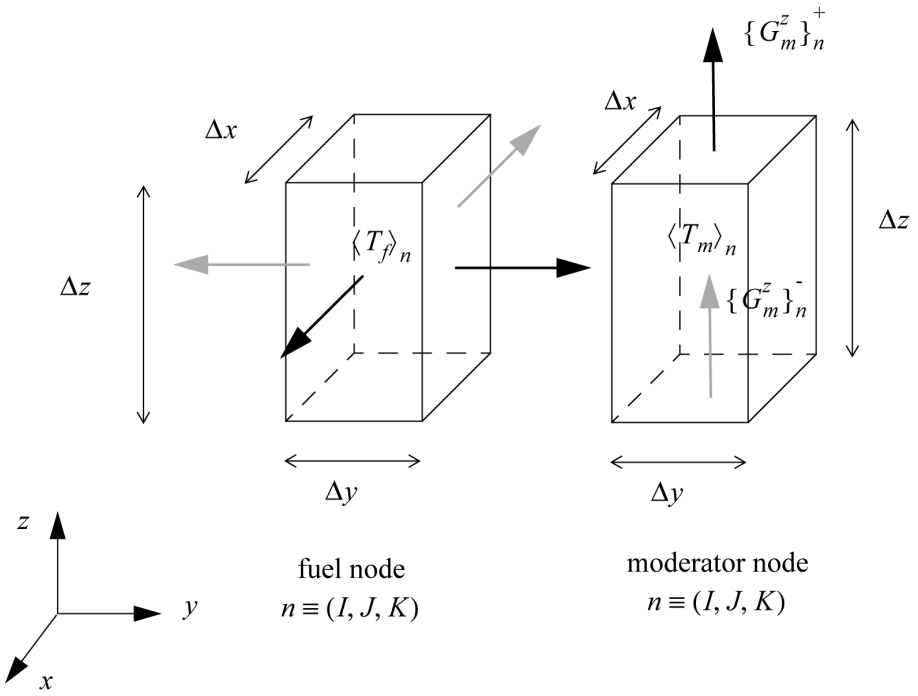

Figure 4.1: Principles and conventions used for the spatial discretisation of a fuel/moderator node n.

Introducing the following notations for the volume and area-averaged quantities in the fuel region:

$$\langle X_f \rangle = \frac{1}{V_f} \int_{V_f} X_f(\boldsymbol{r}) dV, \tag{4.3}$$

$$\{X_f\} = \frac{1}{A_f} \int_{A_f} X_f(\boldsymbol{r}) dA \tag{4.4}$$

and approximating the volume-averaged heat flux as:

$$\int_{V_f} \nabla \cdot \overline{q}''(\boldsymbol{r}) dV \approx \int_{S_f} q_r''(\boldsymbol{r}) dS = \frac{S_f \int_{S_f} q_r''(\boldsymbol{r}) dS}{S_f} = S_f \{q_r''\}_{f,0}^S, \tag{4.5}$$

Eq. (4.2) reads as:

$$-S_f \{q_r''\}_{f,0} + V_f \left\langle q''' \right\rangle_{f,0} = 0 \tag{4.6}$$

with A_f, V_f and S_f standing for the surface area, volume and cross-section area of a fuel node and index "S" denoting the averaging over the lateral surface area of a fuel node.

Using steady-state Fourier's law of heat conduction written as:

$$\overline{q}''(\boldsymbol{r}) = -k_f(T_f)\nabla T_f(\boldsymbol{r}), \tag{4.7}$$

the steady-state area-averaged heat flux in the fuel region can be approximated as:

$$\{q_r''\}_{f,0}^S \approx \langle H \rangle_0 \left(\langle T_f \rangle_0 - \langle T_m \rangle_0 \right), \tag{4.8}$$

where $\langle H \rangle_0$ stands for the artificial static heat transfer coefficient ($\langle H \rangle_0$ differs from the real one since one does not use the wall temperature). $\langle H \rangle_0$ varies from node to node and has to be computed from the steady-state output of a commercial core simulator (CS) as:

$$\langle H \rangle_0 = \frac{\{q_r''^{,CS}\}_{f,0}^S}{\left\langle T_f^{CS} \right\rangle_0 - \langle T_m^{CS} \rangle_0}, \tag{4.9}$$

where $\{q_r''^{,CS}\}_{f,0}^S$ stands for the surface-averaged heat flux from a commercial core simulator, $\left\langle T_f^{CS} \right\rangle_0$ and $\langle T_m^{CS} \rangle_0$ are the volume-averaged fuel and coolant/moderator temperature extracted again from a commercial core simulator. Combining Eq. (4.6) with Eq. (4.8), one obtains the following steady-state space-dependent equation describing the evolution of the fuel temperature:

$$-\langle H_{eff} \rangle_0 \left(\langle T_f \rangle_0 - \langle T_m \rangle_0 \right) + \left\langle q''' \right\rangle_{f,0} = 0, \tag{4.10}$$

where $\langle H_{eff} \rangle_0$ stands for the effective heat transfer coefficients defined as:

$$\langle H_{eff} \rangle_0 = H_0 \frac{S_f}{V_f} = \frac{S_f \{q_r''^{,CS}\}_{f,0}^S}{V_f \left(\left\langle T_f^{CS} \right\rangle_0 - \langle T_m^{CS} \rangle_0 \right)} = \frac{\left\langle q'''^{,CS} \right\rangle_{f,0}}{\left\langle T_f^{CS} \right\rangle_0 - \langle T_m^{CS} \rangle_0}. \tag{4.11}$$

4.2.2 Algorithm used for solving the static heat transfer equations

Once the spatial distribution of the coolant/moderator temperature is computed from Eqs. (3.23), (3.25)-(3.27), (3.28) and (3.36) combined with the water tables and the power density is calculated from the neutron fluxes (see Eq. (2.66)) and fission cross-section data, the spatial distribution of the steady-state fuel temperature can be estimated from Eq. (4.10) as:

$$\langle T_f \rangle_0 = \langle T_m \rangle_0 + \frac{\left\langle q''' \right\rangle_{f,0}}{\langle H_{eff} \rangle_0}. \tag{4.12}$$

In the above, it was decided to express the corresponding steady-state heat transfer equation in terms of volume-averaged (node-averaged) quantities.

4.3 Derivation of the dynamic heat-transfer equations

To derive the heat-transfer equations describing the space-frequency distribution of the noise in fuel temperature, one follows the same methodology as was implemented in the previous two Sections (2.3 and 3.3), namely starting with the local time- and space-dependent heat balance equation written as:

$$\rho_f(T_f) c_f(T_f) \frac{\partial}{\partial t} T_f(\boldsymbol{r}, t) = -\nabla \cdot \overline{q}''(\boldsymbol{r}, t) + q'''(\boldsymbol{r}, t). \tag{4.13}$$

4.3.1 Algorithm used for the spatial discretisation

Integrating both sides of Eq.(4.13) over the fuel volume of a node, one obtains

$$\rho_f(\langle T_f\rangle_0^{CS})c_f(\langle T_f\rangle_0^{CS})\frac{\partial}{\partial t}\int_{V_f} T_f(\boldsymbol{r},t)dV = -\int_{V_f}\nabla\cdot\overline{q}''(\boldsymbol{r},t)dV + \int_{V_f} q'''(\boldsymbol{r},t)dV, \quad (4.14)$$

where $\rho_f(\langle T_f\rangle_0^{CS})$ and $c_f(\langle T_f\rangle_0^{CS})$ denote the static node-wise fuel density and fuel specific heat taken at the steady-state node-averaged fuel temperature $\langle T_f\rangle_0^{CS}$ obtained from a commercial core simulator, respectively. Then, introducing the following notations for the volume and area-averaged quantities in the fuel region of a node:

$$\langle X_f\rangle(t) = \frac{1}{V_f}\int_{V_f} X_f(\boldsymbol{r},t)dV, \quad (4.15)$$

$$\{X_f\}(t) = \frac{1}{A_f}\int_{A_f} X_f(\boldsymbol{r},t)dA \quad (4.16)$$

and approximating the volume-averaged time-dependent heat flux as:

$$\int_{V_f}\nabla\cdot\overline{q}''(\boldsymbol{r},t)dV \approx \int_{S_f} q_r''(\boldsymbol{r},t)dS = \frac{S_f\int_{S_f} q_r''(\boldsymbol{r},t)dS}{S_f} = S_f\{q_r''\}_f^S(t), \quad (4.17)$$

Eq. (4.13) reads as:

$$\rho_f(\langle T_f\rangle_0^{CS})c_f(\langle T_f\rangle_0^{CS})V_f\frac{\partial}{\partial t}\langle T_f\rangle(t) = -S_f\{q_r''\}_f^S(t) + V_f\left\langle q'''\right\rangle_f(t) \quad (4.18)$$

with the index "f" indicating the averaging over the fuel region of a node. Combining Eq. (4.8) written for the time-dependent quantities with Eq. (4.18), one obtains the following time-space dependent equation describing the evolution of the fuel temperature:

$$\rho_f(\langle T_f\rangle_0^{CS})c_f(\langle T_f\rangle_0^{CS})\frac{\partial}{\partial t}\langle T_f\rangle(t) = -\langle H_{eff}\rangle_0(\langle T_f\rangle(t) - \langle T_m\rangle(t)) + \left\langle q'''\right\rangle_f(t). \quad (4.19)$$

Finally, after the application of first order perturbation theory (similarly to the neutronic and thermo-hydraulic noise equations) and a Fourier transform, the equation for the noise in the fuel temperature can be written as

$$\tau_f i\omega\langle\delta T_f\rangle(\omega) = -\langle\delta T_f\rangle(\omega) + \langle\delta T_m\rangle(\omega) + \frac{\left\langle\delta q'''\right\rangle_f(\omega)}{\langle H_{eff}\rangle_0}, \quad (4.20)$$

where τ_f is the node-wise fuel time constant defined as

$$\tau_f = \frac{\rho_f(\langle T_f\rangle_0^{CS})c_f(\langle T_f\rangle_0^{CS})}{\langle H_{eff}\rangle_0}. \quad (4.21)$$

4.3.2 Algorithm used for solving dynamic heat transfer equations

Once the space-frequency distribution of the noise in the coolant/moderator temperature is computed from Eqs. (3.65), (3.75), (3.78), (3.85), (3.90) and the noise in the power density is calculated from the neutron noise (see Eq. (2.55)) and fission cross-section data, the

space-frequency distribution of the fluctuations in the node-averaged fuel temperature can be estimated from Eq. (4.20) as:

$$\langle \delta T_f \rangle (\omega) = \frac{\frac{\langle \delta q''' \rangle_f (\omega)}{\langle H_{eff} \rangle_0} + \langle \delta T_m \rangle (\omega)}{1 + i\omega\tau_f}. \tag{4.22}$$

In the above equation, similarly to the static case, it was decided to express the corresponding noise equations in terms of volume-averaged (node-averaged) quantities.

Chapter 5

COUPLED CALCULATIONS

5.1 Introduction

This section aims at giving a detailed overview of the coupled calculation procedure used to evaluate both the static and dynamic neutronic and thermo-hydraulic quantities.

5.2 Static coupled calculations

A detailed analysis of the neutronic and thermo-hydraulic static equations (2.66), (3.23), (3.25)-(3.27), (3.28), (3.36) and (4.12) shows that these equations represent a strongly-coupled system where none of them can be solved separately from another one. Therefore, following the same approach as in the static thermo-hydraulic calculations, an iterative scheme for solving the coupled equations where a solution is first searched separately for thermo-hydraulic and neutronic models and then both models exchange their outputs until the convergence is achieved, was developed. To demonstrate the main principles of the iterative scheme as well as those of the newly-developed model, a block diagram is presented in Fig. 5.1. In the figure, the iterative scheme for coupled static calculations (outer iterations) is illustrated.

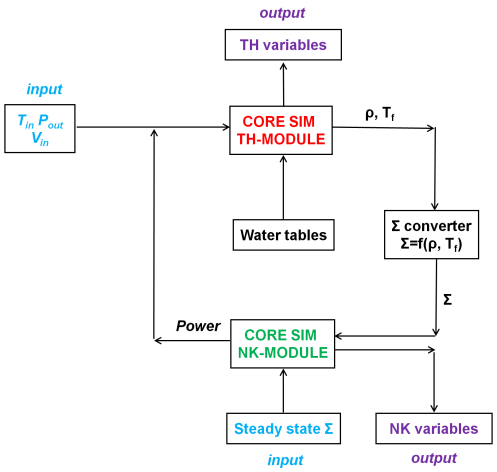

Figure 5.1: Iterative scheme for static coupled neutronic/thermo-hydraulic calculations.

As can be seen from Fig. 5.1, the scheme consists of two main modules. The first module, called **CORE SIM TH-MODULE** consists of a set of corresponding static thermo-hydraulic equations (3.23), (3.25)-(3.27), (3.28), (3.36) and (4.12) and is used to calculate the spatial distributions of the thermo-hydraulic parameters. As an input, it requires the spatial distributions of the neutron fluxes and fission cross-sections (i.e. the generated reactor core power) plus water tables and as an output, the static spatial distributions of the thermo-hydraulic parameters including the coolant/moderator enthalpy, density, velocity, pressure, temperature and fuel temperature are provided. The second block, named as **CORE SIM NK-MODULE** (where "**NK**" stands for **N**eutron-**K**inetic) is based on Eq. (2.66) and calculates the corresponding neutron densities (or neutron fluxes). As an input, it needs the steady-state two-group cross-sections and as an output it gives the static spatial distributions of the neutron densities and thus the spatial distribution of the power density. There are also two additional small modules called Σ**-CONVERTER** (where "Σ" stands for cross-section) and **INPUT** included into the iterative scheme. The Σ**-CONVERTER** block incorporates the cross-section model and converts the calculated thermo-hydraulic parameters (coolant/moderator density and fuel temperature) into the corresponding cross-sections. The **INPUT** block serves as a storage which accepts the necessary input data and the corresponding boundary conditions (such as core inlet coolant/moderator temperature and velocity, core exit pressure). Then, assuming that all input data are provided and the respective boundary conditions are specified, the calculation procedure of the respective neutronic and thermo-hydraulic quantities can be summarized as following:

- calculation of the neutron densities via **CORE SIM NK-MODULE** for a given set of two-group macroscopic cross-sections (obtained from a commercial core simulator); thereafter, the static spatial distribution of the power density can be computed from the calculated neutron fluxes and the corresponding fission cross-sections;

- calculation of the static spatial distributions of the coolant/moderator density, velocity, enthalpy, temperature, pressure and fuel temperature using **CORE SIM TH-MODULE** fed with the power density calculated at the previous step;

- conversion of the spatial distributions of the coolant/moderator density and fuel temperature into the spatial distribution of the cross-sections via Σ**-CONVERTER**;

- calculation of the neutron fluxes using **CORE SIM NK-MODULE** with a new set of cross-sections;

- the obtained neutron fluxes and cross-sections are thereafter used to calculate a new heating source (i.e. new power density) for the next cycle of thermo-hydraulic calculations;

- all previous four steps are repeated until the desired convergence is reached.

The convergence criteria for the coupled (outer) static iterations is imposed on some selected quantities (i.e. both neutronic (neutron densities) and thermo-hydraulic (coolant/moderator density, velocity, enthalpy, pressure and fuel temperature)) and by default set to specific values, i.e. the relative change in the corresponding quantities between two consecutive iterations should be less than those values.

5.3 Dynamic coupled calculations

The thorough analysis of the neutronic and thermo-hydaraulic noise equations (2.55), (3.65), (3.75), (3.78), (3.85), (3.90) and (4.22) derived in the corresponding sections demonstrates that this set of noise equations is not fully closed, i.e. the number of unknown quantities exceeds the number of available equations. Therefore, additional correlations between different neutronic and thermo-hydraulic parameters are needed and will be specified in this section. Such correlations will then help to simplify the corresponding set of equations as well as to reduce the number of unknown parameters to be directly solved for.

Neutronic/thermo-hydraulic correlations
In order to calculate the neutron noise, the corresponding neutron noise source should explicitly be given. Conventionally, in neutron noise theory, the respective noise source is often represented as a perturbation in the respective cross-section(s). In turn, the neutron cross-sections are usually functions of two thermo-hydraulic quantities, i.e. coolant/moderator density and fuel temperature. As a result of this dependence, any perturbation in thermo-hydraulic parameters will lead to the corresponding perturbation in cross-sections, creating the noise source for the fluctuations in neutron density. Thus, using a similar linear approach as in the previous case, a fluctuation in any of the cross-sections can be written as a linear combination between two fluctuations: one in the fuel temperature and another one in the coolant/moderator density. From the latter, one gets:

$$\delta\Sigma_X(\omega) \approx \sigma \left\langle \delta\rho_m \right\rangle (\omega) + \eta \left\langle \delta T_f \right\rangle (\omega), \tag{5.1}$$

where the index "X" denotes the cross-section type and the coefficients σ and η are the corresponding node-wise derivatives of the cross-sections with respect to the coolant/moderator density and fuel temperature, respectively, and defined as:

$$\sigma = \left.\frac{\partial\Sigma_X}{\partial \left\langle \rho_m \right\rangle}\right|_{\left\langle T_f \right\rangle = const} \approx \left.\frac{\delta\Sigma_X}{\delta \left\langle \rho_m \right\rangle}\right|_{\left\langle T_f \right\rangle = const}, \tag{5.2}$$

$$\eta = \left.\frac{\partial\Sigma_X}{\partial \left\langle T_f \right\rangle}\right|_{\left\langle \rho_m \right\rangle = const} \approx \left.\frac{\delta\Sigma_X}{\delta \left\langle T_f \right\rangle}\right|_{\left\langle \rho_m \right\rangle = const}. \tag{5.3}$$

The cross-section derivatives σ and η are calculated under steady-state conditions. For this purpose, three separate static simulations are performed using a static core simulator. The first simulation is done at steady-state conditions, the second one with a small homogeneous perturbation in the steady-state coolant/moderator density ($\delta \left\langle \rho_m \right\rangle = 0.01\ g \cdot cm^{-3}$ by default) for the fixed fuel temperature ($\delta \left\langle T_f \right\rangle = 0$ K) and the last one for a small homogeneous perturbation in the steady-state fuel temperature ($\delta \left\langle T_f \right\rangle = 40$ K by default) for the fixed coolant/moderator density ($\delta \left\langle \rho_m \right\rangle = 0\ g \cdot cm^{-3}$). The corresponding changes in the cross-section weighted with the respective perturbations in the coolant/moderator density or fuel temperature provide the required values of σ and η for each node. Again, the linear dependence of the cross-sections on thermo-hydraulic parameters, implemented in the above calculations, is justified by the fact that only small stationary fluctuations are considered in the present work.

The further analysis of the derived neutronic and thermo-hydaraulic noise equations indicates that these equations describe a strongly-coupled system and thus can not be

resolved independently. As a result, the same problem as in the static coupled calculations then arises when one tries to obtain the full solution for an entire coupled system. Therefore, an iterative scheme where the solution is first searched separately for thermo-hydraulic and neutronic models and then both models exchange their outputs until the convergence is achieved, was applied. To demonstrate the main principles of the iterative schemes as well as those of the newly-developed model, a block diagram is presented in Fig. 5.2. In the figure, the iterative scheme for coupled noise calculations (outer iterations) is illustrated.

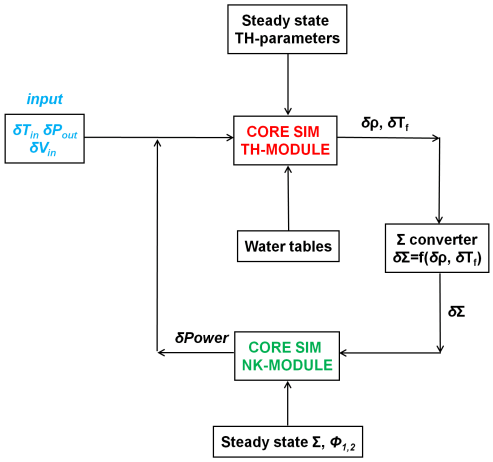

Figure 5.2: Coupled dynamic (noise) calculations.

As Fig. 5.2 shows, similarly to the static iterative scheme, the present dynamic scheme consists of two main modules. The first module, called **CORE SIM TH-MODULE** contains the set of corresponding thermo-hydraulic noise equations (3.65), (3.75), (3.78), (3.85), (3.90), (4.22) and is used to calculate the space-frequency distributions of the noise in the thermo-hydraulic quantities. As an input, it requires the static distributions of all thermo-hydraulic parameters, water tables, updated fission cross-sections and the neutron noise (i.e. the noise in the power density) calculated at the previous stage and as an output, the space-frequency distributions of the noise in the coolant/moderator density, velocity, enthalpy, pressure, temperature and fuel temperature are provided. The second block, named as **CORE SIM NK-MODULE** is based on Eq. (2.55) and calculates the space-frequency distribution of the noise in the neutron density. As an input, it needs the steady-state two-group cross-sections, static neutron densities (or fluxes) and the neutron noise source, i.e. some perturbation(s) in the cross-sections and as an output it gives the space-frequency distribution of the fluctuations in the neutron density and, thus, in the reactor power density. There are also two additional small modules called **Σ-CONVERTER** and **INPUT** included into the dynamic iterative scheme. **Σ-CONVERTER** block incorporates Eq. (5.1) and converts the thermo-hydraulic perturbations into the cross-section perturbations. The **INPUT** block serves as a storage which accepts initial TH-perturbation(s), necessary input data, cross-section data, the static spatial distributions of the neutron densities and all thermo-hydraulic quantities. In the present model, three types of thermo-hydraulic perturbations can be taken into consideration, i.e. a perturbation in the coolant inlet velocity (or flow), inlet coolant temperature and outlet pressure. In practice, only these quantities can really be measured since today's commercial

reactors are usually only equipped with inlet temperature sensors, inlet flow meters and outlet pressure sensors. Thus, this latter fact justifies such a specific selection of the quantities being perturbed and makes the choice of any other TH-parameters inappropriate. In addition, since only the core is actually modelled in this work, these quantities also corresponds to the necessary boundary conditions to solve the problem. Then, assuming that the initial perturbation is defined and the necessary input data are given, the calculation procedure of the noise in the respective quantities can be summarized as follows:

- calculation of the space-frequency distributions of the fluctuations in the coolant/moderator density, velocity, enthalpy, temperature, pressure and fuel temperature using **CORE SIM TH-MODULE** with zero noise in the power density;
- conversion of the calculated fluctuations in the coolant/moderator density and fuel temperature into cross-section fluctuations using Σ**-CONVERTER**;
- calculation of the space-frequency distribution of the neutron noise for a given neutron noise source using **CORE SIM NK-MODULE**;
- the obtained space-frequency distribution of the neutron noise is thereafter used as noise in the power density (i.e. as a new fluctuating heating source) for the next cycle of thermo-hydraulic noise calculations;
- all previous four steps are repeated until the desired convergence is reached.

The convergence criteria for the outer dynamic iterations was imposed on the noise in some selected quantities (both neutronic (neutron densities) and thermo-hydraulic (coolant/moderator density, velocity, enthalpy, pressure and fuel temperature)) and by default was set to specific values, i.e. the relative change in the noise between two consecutive iterations should be less than those values.

Chapter 6

CONCLUSIONS

In this report, the development of a unique multi-purpose coupled neutronic/thermo-hydraulic tool for research and education was reported. The tool can consider both critical systems and subcritical systems with an external neutron source, static cases and dynamic cases in the frequency domain (i.e. for stationary fluctuations). For each situation, the three dimensional spatial distributions of static neutron fluxes, all thermo-hydraulic quantities and their corresponding first-order neutron noise can be determined, as well as the effective multiplication factor of the system. The tool uses as input two-group macroscopic cross-sections, kinetic and thermo-hydraulic parameters as well as geometrical details from other static commercial core simulators.

The coding was implemented in *MATLAB*, which makes the pre- and post-processing of data easy, as well as the code highly portable between different operative systems and computer platforms. The code was developed while paying careful attention to data storage requirements and to the robustness of the algorithms. In particular, the numerical algorithms implemented in the tool take advantage of the sparsity of the matrices, and the *MATLAB* built-in linear algebra packages *LAPACK* and *UMFPACK* are extensively used. In addition, an explicitly-restarted Arnoldi method and a power iteration method using Wielandt's shift were implemented to solve eigenvalue problems.

The tool is freely available on request to the authors of the present report. The tool is distributed under a GNU General Public License (*http://www.gnu.org/licenses/gpl.html*).

In the present version of the computational tool, the calculation of both the neutron density and thermo-hydraulic fields is performed. The dynamic system can be directly defined via the fluctuations of the macroscopic cross-sections as well as via the fluctuations in the actual thermal-hydraulic boundary conditions. The latter ones are expressed in terms of core inlet coolant/moderator velocity, core inlet coolant/moderator temperature and core exit coolant/moderator pressure. The present report describes both neutronic and thermal-hydraulic modules as well as the coupling between them. In addition, the coupled neutronic/thermal-hydraulic tool also has the capability to determine the closed-loop reactor transfer function of any heterogeneous core.

Further improvements on the neutronic side have already been investigated, for instance via the development of a neutronic solver based on the Analytical Nodal Method for both static and dynamic calculations [3]. Some additional improvements of the thermo-hydraulic module such as the introduction of a Drift Flux Model or 6 equation model are planned to be undertaken in the future. Extension to fast systems is also being investigated, since rather large fluctuations have been observed in fast system-prototypes, for

which the computational tool has a large area of application.

ACKNOWLEDGEMENTS

The development of the reported computational tool would not have been possible without active discussions with and participations of some students from the Division of Nuclear Engineering, Department of Applied Physics, Chalmers University of Technology. The authors thus wish to express their appreciation to Dr. Augusto Hernández-Solís, Dr. Viktor Larsson, Dr. Carl Sunde, and MSc. Filippo Zinzani. Prof. Imre Pázsit is acknowledged for his long-term support during this project.

The continued financial support from the Swedish nuclear industry via different projects, for which the development of the tool was necessary, is also deeply acknowledged, and among others Ringhals AB (research contracts: 522351-003, 531970-003, 543672-002, 557700-003, 566379-003, and 578254-003), the Swedish Radiation Safety Authority SSM, formerly the Swedish Nuclear Power Inspectorate SKI (research contracts: 14.5-991060-99180, 14.5-000983-00156, 14.5-010892-01161, 14.5-011142-01261, SSM 2012-3299 and SSM 2013-903), and the Nordic Thermal-Hydraulic Network NORTHNET [research contracts: 4500131026 (Forsmark Kraftgrupp AB), 581422-025 (Ringhals AB), SKI 2007/1588/200705-015 (Swedish Radiation Safety Authority SSM formerly the Swedish Nuclear Power Inspectorate), and SE 08-018 (Westinghouse Electric Sweden AB)].

References

[1] Dykin V. and Demazière C., "Demonstration of the coupled CORE SIM neutronic/thermo-hydraulic tool", *CTH-NT-301 report*, Chalmers University of Technology (2014).

[2] Dykin V. and Demazière C., "User's manual of the coupled CORE SIM neutronic/thermo-hydraullic tool", *CTH-NT-302 report*, Chalmers University of Technology (2014).

[3] Larsson V. and Demazière C., "Comparative study of 2-group P1 and diffusion theories for the calculation of the neutron noise in 1D 2-region systems", *Annals of Nuclear Energy*, **36** (10), pp. 1574-1587 (2009).

[4] Nakamura S., "Computational methods in engineering and science with applications to fluid dynamics and nuclear systems", *John Wiley & Sons, Inc.*, New York, NY, USA (1977).

[5] Davis T. A., "*UMFPACK* version 4.6 user guide", Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA (2002).

[6] Arnoldi W.E.,"The principle of minimized iterations in the solution of the matrix eigenvalue problem", *Quarterly of Applied Mathematics*, **9** (1), pp. 17-29 (2009).

[7] Saad Y., "Iterative methods for sparse linear systems - Second edition", Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2003).

[8] Zinzani F., Demazière C., and Sunde C., "Calculation of the eigenfunctions of the two-group neutron diffusion equation and application to modal decomposition of BWR instabilities", *Annals of Nuclear Energy*, **35** (11), pp. 2109-2125 (2008).

[9] Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., and Sorensen D., "*LAPACK* user's guide", Third Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1999).

[10] Ericsson T., personal communication, Chalmers University of Technology, Gothenburg, Sweden (2008).

[11] Rust J. H., "Nuclear Power Plant Engineering", *Haralson Publishing Company*, Atlanta, USA (1979).

[12] Chisholm D., "Pressure gradients due to friction during the flow of evaporating two-phase mixtures in smooth tubes and channels", *Int. J. of Heat Mass and Transfer*, **16**, pp. 347-358 (1973).

[13] Shah R.K. and Sekulic D.P., "Fundamentals of Heat Exchanger Design", *Haralson Publishing Company*, Appendix C. John Wiley Sons Inc. (2003).