

---

# Nuclear Reactor Simulation

---

Andrés Rodríguez Hernández,  
Armando Miguel Gómez-Torres and  
Edmundo del Valle-Gallegos

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.79723>

---

## Abstract

A summary is described about nuclear power reactors analyses and simulations in the last decades with emphasis in recent developments for full 3D reactor core simulations using highly advanced computing techniques. The development of the computer code AZKIND is presented as a practical exercise. AZKIND is based on multi-group time dependent neutron diffusion theory. A space discretization is applied using the nodal finite element method RTN-0; for time discretization the  $\theta$ -method is used. A high-performance computing (HPC) methodology was implemented to solve the linear algebraic system. The numerical solution of large matrix-vector systems for full 3D reactor cores is achieved with acceleration tools from the open-source PARALUTION library. This acceleration consists of threading thousands of arithmetic operations into GPUs. The acceleration is demonstrated for different nuclear fuel arrays giving extremely large matrices. To consider the thermal-hydraulic (TH) feedback, several strategies are nowadays implemented and under development. In AZKIND, a simplified coupling between the neutron kinetics (NK) model and TH model is implemented for reactor core simulations, for which the TH variables are used to update nuclear data (cross sections). Test cases have been documented in the literature and demonstrate the HPC capabilities in the field of nuclear reactors analysis.

**Keywords:** HPC, high-performance computing, NFEM, nodal finite element method, parallel computing, GPU, graphics processing unit, NK-TH, neutronic-TH coupling

---

## 1. Introduction

The mathematical models representing the nuclear reactor physics are based mainly on two theoretical areas: neutron transport theory and neutron diffusion theory, where it is necessary to remark that neutron diffusion theory is really a simplification of the neutron transport theory.

Numerical methods are used to solve the partial differential equations representing the nuclear reactor physics, and these methods are derived from discretization techniques. For numerical solutions in any scientific area, computational tools have been developed including software and hardware. In the past, the former computer processing was the sequential execution of computer commands, meaning to say that program tasks are carried out one after one. Modern computational tools have been developed for parallel processing, executing several tasks concurrently.

The computing branch dealing with the system architecture and appropriate software related to the simultaneous execution of computer instructions and applications is known as parallel computing science. Former developments in parallel computing were made in the late 1950s, following the construction of supercomputers throughout the 1960s and 1970s. Nowadays, clusters are the workhorse of scientific computing and are the dominant architecture in data centers.

Since the late 1950s, the performance of safety analyses was essential in the nuclear industry, in research reactors, but mainly safety analyses of nuclear power plants for commercial purposes. Scientific computing calculations were vital to these safety analyses, but with important limitations in computer/computing capabilities. At the beginning, the objective was to give a solution to partial differential equation models based on neutron diffusion or neutron transport with technology and methods available in those years. Numerical techniques were used first with finite differences and finite element approaches, and gradually up to now, with nodal finite element methods (NFEMs). Despite the numerical method employed, the computer code user faces the problem of solving extremely large algebraic systems challenging hardware/software capabilities. Generation of results for any reactor simulation in considerable short times is a desirable achievement for computer code users [1].

Recent developments of high-performance computer equipment and software have made the use of supercomputing in many scientific areas possible. The appropriate selection of parallel computing software, like newly developed linear algebra libraries, to be used in a specific project may result in a suitable platform to simulate nuclear reactor states with relatively prompt results.

Throughout the world, several research projects in the last decade have been developed with the main objective of making full tridimensional (3D) coupling simulations of nuclear reactor cores, leaving aside the obsolescence of the point kinetics theory. Most of the modern nuclear reactor simulators are based on neutron transport theory, or on neutron diffusion theory, to obtain detailed 3D results. As light water is used for cooling/moderating light water reactors (LWRs), a comprehensive analysis of the reactor core physics must include thermal-hydraulic phenomena, so that modern simulations perform reactor calculations with thermal-hydraulic feedback coupled with neutron kinetics calculations.

All the discussions included in this chapter are centered in a simulator for light water reactors. The computer code AZtlan KInetics in Neutron Diffusion (AZKIND) is part of the neutronic codes selected for their implementation in the AZTLAN Platform<sup>1</sup> project in which neutron transport and neutron diffusion codes are being developed in Mexico. A (TH) model has been implemented recently and coupled with the neutronic (NK) model, and both models are based on HPC implementations.

## 2. Reactor core calculation overview

Although there has been growing interest in the transport-based core neutronics analysis methods for a more accurate calculation with high-performance computers, it is yet impractical to apply them in the real core design activities because their performance is not so practical on ordinary desktop or server computing machines. For this reason, most of the neutronics codes for reactor core calculations are still subject to the two-step calculation procedure, which consists of (1) homogenized group neutron parameters generation and (2) neutron diffusion core calculation.

In the core calculation steps that are the main concern of this work, nodal codes based on the diffusion theory have been used to determine the neutron multiplication factor and the corresponding core neutron flux (or power) distribution. Practically, almost all nuclear reactor simulation codes employ the two-group approach involving only fast and thermal neutron energy groups for the applications to light water reactors (LWR). However, numerical calculations with the two-group structure are not appropriate in the analysis of cores loaded with mixed oxide fuels or analysis of fast breeder reactors, since the neutron spectrum is influenced more by the core environment, requiring much more energy groups than only two groups.

As settled in Ref. [2], even using a high-performance computer, a direct core calculation with several tens of thousands of fuel pins is difficult to perform in its heterogeneous geometry model form, using fine groups of a prepared reactor cross-section library. The Monte Carlo method can handle such a core calculation (see also the Serpent code), but it is not easy to obtain enough accuracy for a local calculation or small reactivity because of accompanying statistical errors, besides the large calculation times. Instead of using neutron transport computer codes, the nuclear design calculation is performed in two steps: (1) lattice calculation in a two-dimensional infinite arrangement of fuel rods or assemblies for the generation of homogenized lattices jointly with their corresponding homogenized cross-sections and (2) core calculation in a three-dimensional whole core, with a neutron diffusion code using the information of the previous step.

As shown in **Figure 1** [2], the lattice calculation prepares few-group homogenized cross sections which maintain the energy dependence (neutron spectrum) of nuclear reactions, and these reduce the core calculation cost in terms of time and memory. The final core design

---

<sup>1</sup>This work was performed under the auspices of the financial support from the National Strategic Project No. 212602 (AZTLAN Platform) as part of the Sectorial Fund for Energetic Sustainability CONACYT–SENER, Mexico.

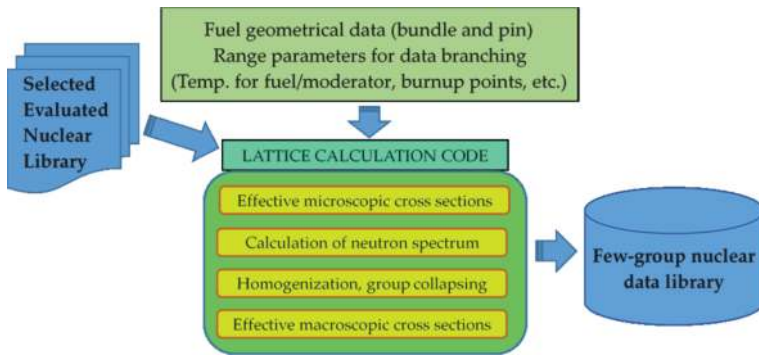


Figure 1. Typical lattice calculation process flow for light water reactors [2].

parameters are not concerned with continuous energy dependence, but spatial dependence, such as power distribution, is important to avoid high local neutron fluxes or high absorbing materials causing significant neutron flux gradients, mainly when safety analyses are performed upon the final proposed core designs.

In the core calculations with space-dependent data (cross sections and neutron flux), the effective cross sections are processed, with a little degradation in the accuracy as possible, by using the results from the multi-group lattice calculation. Lattice code calculation and codes are not discussed here.

There are two processes followed for lattice calculation. One is the homogenization to lessen the space-dependent information and the other is group-collapsing to reduce the energy-dependent information as shown in **Figure 2**. The fundamental idea of both methods is to preserve neutron reaction rate. The next step is to consider the conservation of reaction rate in the energy group  $G$  in the same manner as that in the homogenization.

The number of few groups depends on reactor type and computation code. Two or three groups are adopted for the NK- and TH-coupled core calculation of LWRs and much more groups (18, 33, etc.) are used for the core calculation of LMFRs (Liquid Metal Fast Reactors). Currently, revised methods exist for the improvement of cross-sections generation using computer codes dedicated to lattice calculation for few-groups approach, like in Ref. [3], where three topics are involved: (1) improved treatment of neutron-multiplying scattering reactions; (2) group constant generation in reflectors and other non-fissile regions, leading to the use of discontinuity factors in neutron diffusion codes; and (3) homogenization in leakage-corrected criticality spectrum, in which several leakage corrections are used to attain criticality, accounting for the non-physical infinite-lattice approximation. Another improvement was done in Monte Carlo codes [4], implementing reliable multi-group cross-sections calculations for collapsed flux spectrum. Ref. [4] focuses on calculating scattering cross sections, including the group-to-group scattering.

The following sections contain, as a matter of example, summarized explanations of the AZKIND nuclear reactor simulator in which the reactor physics is based on neutron diffusion theory.

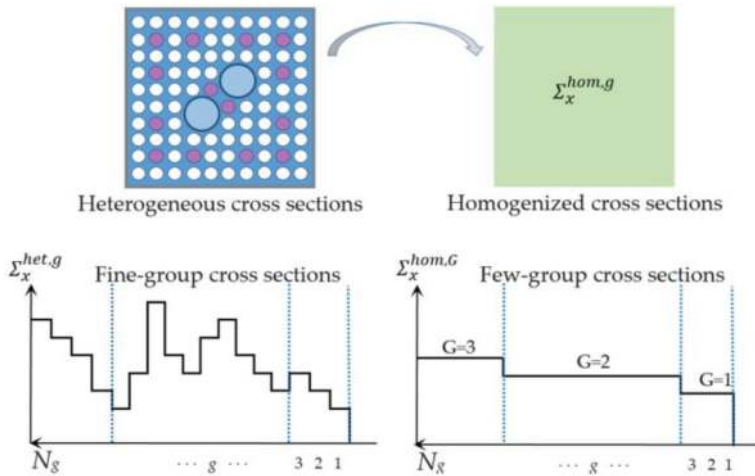


Figure 2. Homogenization and group collapsing of cross sections [2].

### 3. Neutron diffusion theory and nodal methods

#### 3.1. Multi-group time-dependent neutron diffusion equations

For  $G$  neutron energy groups and  $I_p$  delayed neutron precursor concentrations, the neutron diffusion kinetics equations are given by Eqs. (1) and (2) [5]. Although there has been a growing interest in the transport-based core neutronics analysis methods for more accurate calculation with high-performance computers, it is yet impractical to apply them in the real core design activities because their performance is not so practical on ordinary desktop or server computing machines. For this reason, most of the neutronics codes for reactor core calculations are still subject to the two-step calculation procedure, which consists of homogenized group neutron parameter generation and neutron diffusion core calculation

$$\frac{1}{v^g} \frac{\partial}{\partial t} \phi^g(\vec{r}, t) = \nabla \cdot D^g \nabla \phi^g(\vec{r}, t) - \Sigma_R^g(\vec{r}, t) \phi^g(\vec{r}, t) + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_S^{g' \rightarrow g}(\vec{r}, t) \phi^{g'}(\vec{r}, t) + (1 - \beta) \chi^g \sum_{g'=1}^G v^{g'}(\vec{r}, t) \Sigma_f^{g'}(\vec{r}, t) \phi^{g'}(\vec{r}, t) + \sum_{i=1}^{I_p} \chi_i^g \lambda_i C_i(\vec{r}, t); \quad g = 1, \dots, G; \quad (1)$$

$$\frac{\partial}{\partial t} C_i(\vec{r}, t) = \beta_i \sum_{g=1}^G v^g(\vec{r}, t) \Sigma_f^g(\vec{r}, t) \phi^g(\vec{r}, t) - \lambda_i C_i(\vec{r}, t); \quad i = 1, \dots, I_p; \quad \forall(\vec{r}, t) \in \Omega \times (0, T] \quad (2)$$

In addition to boundary conditions for neutron fluxes, initial conditions must be satisfied by neutron fluxes and neutron precursor functions. Parameters involved in the above equations are described in [5].

### 3.2. Spatial discretization

The spatial discretization of Eqs. (1) and (2) is strongly connected with the discretization of a nuclear reactor core of volume  $\Omega$ . Representing the neutron flux and the precursor concentrations in terms of base functions defined over  $\Omega$ , it is possible to write

$$\phi^g(\vec{r}, t) \equiv \sum_{k=1}^{N_f} u_k(\vec{r}) \phi_k^g(t); \quad g = 1, \dots, G; \quad \forall(\vec{r}, t) \in \Omega \times (0, T]; \quad (3)$$

$$C_i(\vec{r}, t) \equiv \sum_{m=1}^{N_p} v_m(\vec{r}) C_i^m(t); \quad i = 1, \dots, I_p; \quad \forall(\vec{r}, t) \in \Omega \times (0, T]; \quad (4)$$

where  $N_f$  and  $N_p$  are the number of unknowns to be determined for neutron flux and delayed neutron precursors, respectively. Substituting expressions (3) and (4) into (1) and (2), and applying the Galerkin process for spatial discretization, as described in [6], the resulting algebraic system of equations can be expressed in a matrix notation as follows:

$$\begin{aligned} \frac{1}{v^g} M_f \frac{d}{dt} \phi^g(t) = & -K^g \phi^g(t) - \sum_{g'=1}^G S^{g'-g} \phi^{g'}(t) \\ & + (1 - \beta) \chi^g \sum_{g'=1}^G F^{gg'}(t) \phi^{g'}(t) + \sum_{i=1}^{I_p} H^{gi}(t) C_i(t), \quad g = 1, \dots, G; \end{aligned} \quad (5)$$

$$M_p \frac{d}{dt} C_i(t) = \sum_{g'=1}^G P^{ig'}(t) \phi^{g'}(t) - \lambda_i M_p C_i(t), \quad i = 1, \dots, I_p; \quad \forall(\vec{r}, t) \in \Omega \times (0, T]; \quad (6)$$

where  $\phi^g(t) = [\phi_1^g(t), \dots, \phi_{N_f}^g(t)]^T$  and  $C_i(t) = [C_1^i(t), \dots, C_{N_p}^i(t)]^T$ . **Table 1** contains the expressions representing the calculation of each matrix coefficient.

| Matrix     | Type      | Dimension        | Elements  |
|------------|-----------|------------------|---|
| $M_f$      | Mass      | $N_f \times N_f$ | $m_{f,jk} = \int_{\Omega} u_j u_k d\vec{r}$                             |
| $M_p$      | Mass      | $N_p \times N_p$ | $m_{p,lm} = \int_{\Omega} v_l v_m d\vec{r}$                             |
| $K^g$      | Stiffness | $N_f \times N_f$ | $k_{jk}^g = \int_{\Omega} D^g \nabla u_j \cdot \nabla u_k d\vec{r}$     |
| $S^{g'-g}$ | Mass      | $N_f \times N_f$ | $s_{jk}^{g'-g} = \int_{\Omega} \sum_s^{g'-g} u_j u_k d\vec{r}$          |
| $F^{gg'}$  | Mass      | $N_f \times N_f$ | $f_{jk}^{gg'} = \chi^g \int_{\Omega} v^{g'} \sum_f^g u_j u_k d\vec{r}$  |
| $H^{gi}$   | Mass      | $N_f \times N_p$ | $h_{jm}^{gi} = \lambda_i \chi^g \int_{\Omega} v_j u_m d\vec{r}$         |
| $P^{ig'}$  | Mass      | $N_p \times N_f$ | $p_{lk}^{ig'} = \beta_l \int_{\Omega} v^{g'} \sum_f^g u_l v_k d\vec{r}$ |

**Table 1.** Matrix elements from the spatial discretization.

### 3.3. NFE method in spatial discretization

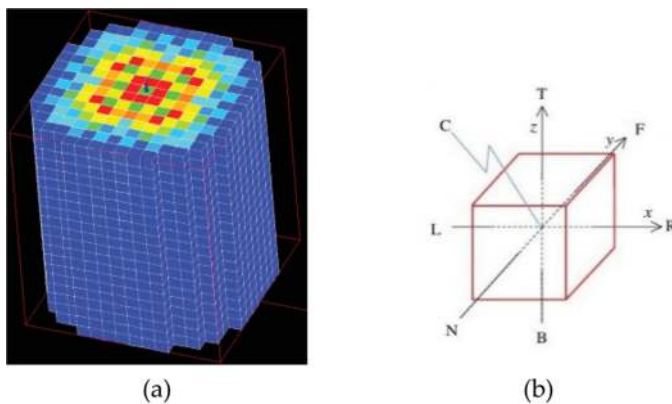
As fully explained in [6] and summarized in [1], a simple NFE element is characterized by the fact that for each node, the function unknowns to be determined are the (00) Legendre moment (average) of the unknown function over each face of the node and the (000) Legendre moment over the node volume. **Figure 3(a)** shows a physical domain  $\Omega$  graphically represented after generating an  $xyz$  mesh. **Figure 3(b)** shows a cuboid-type node with directions through the faces: (x) Right, Left; (y) Near, Far; (z) Top, Bottom; and C for the average of the function over the node volume. Taking into consideration the general form to build up nodal schemes [7], the moments of a function (at edges and body) over a node like the one shown in **Figure 3(b)** can be written for the NFE method RTN-0 (Raviart-Thomas-Nédélec).

In the NFE method RTN-0, the normalized zero-order Legendre polynomials defined over the unit cell  $\Omega_{ijk} = [-1,+1] \times [-1,+1] \times [-1,+1]$  and correlated to each physical cell  $\Omega_e = \Omega_{ijk} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}]$  are used to calculate the elements of the matrices in Eqs. (5) and (6).

The matrix elements are quantified introducing the following nodal basis functions [7]:

$$\begin{aligned} u_L^{00}(x, y, z) &= -\frac{1}{2}(P_{100} - P_{200}); u_R^{00}(x, y, z) = +\frac{1}{2}(P_{100} + P_{200}); \\ u_N^{00}(x, y, z) &= -\frac{1}{2}(P_{010} - P_{020}); u_F^{00}(x, y, z) = +\frac{1}{2}(P_{010} + P_{020}); \\ u_B^{00}(x, y, z) &= -\frac{1}{2}(P_{001} - P_{002}); u_T^{00}(x, y, z) = +\frac{1}{2}(P_{001} + P_{002}); \\ u_C^{000}(x, y, z) &= P_{000} - P_{200} - P_{020} - P_{002}; \end{aligned} \quad (7)$$

where  $P_{lpq}(x, y, z) = P_l(x)P_p(y)P_q(z)$ .



**Figure 3.** Discretization of reactor volume  $\Omega$  and a local node  $\Omega_e$ . (a) Domain  $\Omega$ . (b) Physical local node  $\Omega_e$ .

An extensive discussion on nodal diffusion methods can be found in Ref. [7] for space discretization using simplification approaches for calculating the moments over a node.

### 3.4. Discretization of the time variable

Once the spatial discretization is done, the  $\theta$ -method can be applied [6] for the discretization of the time variable appearing in the algebraic system given by (5) and (6). For the time integration over the interval  $(0, T]$ , this interval is divided in  $L$  time-steps  $[t_l, t_{l+1}]$ , and the following approach is assumed:

$$\int_{t_l}^{t_{l+1}} f(t)dt \cong h_l [\theta f_{l+1} + (1 - \theta)f_l] \quad (8)$$

where  $h_l = t_{l+1} - t_l$ ,  $f_l = f(t_l)$ ,  $f_{l+1} = f(t_{l+1})$ , and  $\theta$  is the time integration parameter.

For time integration, parameters  $\theta_f$  and  $\theta_p$  for neutron flux and delayed neutron precursors are considered with values in the interval  $[0, 1]$ , giving different time integration schemes [6].

Once the formulation to be used for time integration is established, the  $N_f G + N_p I$  system of equations that was spatially discretized, Eqs. (5) and (6) are discretized over the interval  $(0, T]$ . Integrating the referred equations over the time interval  $[t_l, t_{l+1}]$  using approximation (8), the following set of equations is generated:

$$A_{l+1} \Phi_{l+1} = Q_l; \quad l = 0, 1, 2, \dots; \quad (9)$$

$$\Phi_{l+1} = [\phi_{l+1}^1, \dots, \phi_{l+1}^G]^T; \quad Q_l = [S_{f,l}^1, \dots, S_{f,l}^G]^T;$$

For a known vector  $\Phi_l$  the algebraic system (9) is solved for the neutron fluxes  $\Phi_{l+1}$ . Therefore, the computing process requires an initial flux vector for the first time step, which is used in (9) to determine new neutron fluxes at the end of the time step, thus using these neutron fluxes to calculate a new delayed neutron precursor concentration vector. This process is sequentially performed for each time step over the total time interval  $(0, T]$ .

## 4. Reactor power distribution

Once the computer model to solve the reactor kinetics Eqs. (1) and (2) is able to provide the neutron flux profile, the next objective is to know the power distribution in the reactor configuration. It is necessary to be aware that the neutron flux is by itself the shape of the power distribution in multiplicative materials. The numerical methods presented in previous sections to solve Eq. (9) produce an algorithm capable to obtain the neutron flux profile for a reactor steady state. The calculated neutron flux has the following property over the domain  $\Omega$ :  $\|\phi\| = 1$ . To determine the real average neutron flux in the reactor core,  $\phi_c$ , it is necessary to specify the magnitude of the fluxes. For instance, a flux normalization factor  $\phi_{norm}$  can be introduced such that  $\phi_c = \phi_{norm} \phi \left[ \frac{\text{neutrons}}{\text{cm}^2 \cdot \text{seg}} \right]$ .



Theoretically, it would be best to determine the flux level which resulted in a critical reactor (eigenvalue  $\lambda_0 = 1$ ). This could be accomplished by coupling of the NK model with the TH model of the whole reactor. In practice, however, the scaling factor  $\phi_{norm}$  is determined such that the total generated thermal power corresponds to some user-specified value  $P_{th,tot}$ . Before showing how this is done, the relation between the fluxes and the generated thermal power is described. For a given discretization of the  $xy$ -plane with pieces of area  $\Delta a = \Delta x \cdot \Delta y$ , the thermal power  $P_{th,tot}$  can be expressed as follows:

$$P_{th,tot} = \sum_{\Delta a} \int_{z_b}^{z_t} q_f'''(z) da \cdot dz, \quad dV = da \cdot dz; \quad (10)$$

where  $q_f'''$  is the volumetric heat generation rate in the fuel in units of  $[W/cm^3]$ ,  $dV$  is a differential fuel volume, and the limits  $z_b$  and  $z_t$  refer to the coordinates of the bottom and top of the reactor core, respectively. For a given area  $\Delta a$ , the volumetric heat generation rate  $q_f'''(z)$  in an elevation  $z$  may be written in terms of the fluxes as

$$q_f'''(z) = \phi_{norm} E_{fiss} \sum_{g'=1}^G \Sigma_f^{g'}(z) \phi^{g'}(z); \quad (11)$$

where  $\phi_{norm}$  is a dimensionless factor,  $E_{fiss}$  is the energy released by a nuclear fission reaction in  $[MeV/fission]$ , and the sum over  $g'$  is the volumetric fission rate in  $[fissions/(cm^3 \cdot s)]$ . Thus, Eq. (10) is written as

$$P_{th,tot} = \phi_{norm} E_{fiss} \sum_{\Delta a} \int_{z_b}^{z_t} \sum_{g'=1}^G \Sigma_f^{g'}(z) \phi^{g'}(z) da \cdot dz. \quad (12)$$

In a more general way, for a reactor volume  $V$  composed by the union of sub-volumes  $V_e$  (see **Figure 3**), the total thermal power can be expressed as

$$P_{th,tot} = \phi_{norm} E_{fiss} \sum_{e=1}^{N_e} \sum_{g'=1}^G \kappa_{f,e}^{g'} \phi_e^{g'} V_e. \quad (13)$$

Therefore, using the reference total thermal power specified by the code user, the flux normalization factor can be written as

$$\phi_{norm} = P_{th,tot} \left[ \sum_{e=1}^{N_e} \sum_{g'=1}^G \kappa_{f,e}^{g'} \phi_e^{g'} V_e \right]^{-1}; \quad (14)$$

where the factors “kappa-fission” are  $\kappa_{f,e}^{g'} = E_{fiss} \Sigma_{f,e}^{g'}$ . With the flux normalization factor  $\phi_{norm}$  calculated as above, the actual thermal power distributions in the reactor core can be calculated using the current neutron flux in the reactor core  $\phi_e^e = \phi_{norm} \phi^e$ . Nevertheless, it is necessary to introduce the value of  $E_{fiss}$ . This value is used as an average energy released of  $\sim 200$  MeV (i.e.), based on the energies released by the fission of the  $U^{235}$  nuclei [8].

In summary, once the NK model is used to generate the neutron flux distribution in the reactor core, expression (12) can be used to calculate the thermal power being generated along all the nodes in a thermal-hydraulic channel of area  $\Delta a$  and height  $H$ . This thermal power can be the axial power profile needed by the TH model to produce the thermal-hydraulic state corresponding to the generated thermal power.

## 5. Neutronic and thermal-hydraulic coupling model (NK-TH)

The description contained in this section is based on a work published by Ceceñas in Ref. [9] about a TH model developed for boiling water reactors. The TH model was modified from a point kinetics approach with an extension of the NK model to 3D and implemented in the development of AZKIND.

The treatment of neutron kinetics in [9] has been improved by coupling a 3D solution of the neutron diffusion equations with an arrangement of TH channels in parallel. Each channel independently contemplates three regions: (1) one phase, (2) subcooled boiling, and (3) bulk boiling. The objective was to implement a detailed model of a nuclear reactor core, which is somehow perturbed to simulate NK-TH coupling. These perturbations are obtained when the power generated in a group of channels changes and thus affecting the TH state of each channel.

The original [9] TH model is based on a generic channel, which is adapted by transferring to it the operational data as flow area, generated power, axial power profile, and subcooling, among other parameters. Each channel is associated with a number of nuclear fuel assemblies and an axial power profile. Although the neutron model is a two-dimensional model for the radial power profile in each  $z$ -plane covering all the channels, information related to the axial power distribution is considered for each individual channel. In Ref. [9], it is assumed that this steady-state axial power profile is invariant over time, and it is used to weight the axial averages of macroscopic cross sections and void fractions. To perform the numerical implementation of the model, the arrangement of channels is obtained by grouping the total core assemblies into an appropriate number of thermal-hydraulic channels, which gives a definition of a set of channels per quadrant.

For the implementation in AZKIND of the TH model of Ceceñas, the grouping of fuel assemblies was maintained for generating a reduced number of TH channels; operational data are also used. The main difference is that the NK model recursively computes the axial power profile for each channel, and this thermal power is the updated source of power for TH model. Therefore, a "new" thermal-hydraulic condition is generated, and it is used by the NEMTAB model to update the nuclear data to generate new thermal power profiles with the NK model. The process is iterative, and it stops when the convergence is met. Convergence is achieved when updated conditions do not change in both NK and TH models.

The NK-TH coupling in AZKIND performs core calculations as described above to obtain a steady-state reactor core condition. For transient conditions in a time interval  $T$ , the NK-TH coupling process is the same for each time step  $\Delta T$  in  $T$ , that is, a different quasi-steady-state

condition for each successive  $\Delta T$ . Achieving converge for each  $\Delta T$  with respective reactor core conditions means to produce a time-dependent behavior of the reactor condition over the total time interval  $T$ .

The TH model comprises the solution of the mass, momentum, and energy conservation equations in the three regions contemplated by the channel: (1) one phase, (2) subcooled boiling, and (3) bulk boiling. The system receives heat through a non-uniform source whose profile is axially defined plane by plane. This axial use of the power profile allows the inclusion of a wide range of axial profiles, from relatively flat to profiles with their peak value at some axial point in each channel in the core.

In the following subsections, there are several expressions for which the corresponding parameters are defined in Refs. [10, 11].

### 5.1. Heat transfer in the fuel

The heat transfer and temperature distribution in the fuel and cladding can be calculated by a simple model where the heat diffusion equation is solved in one dimension (radial) for a fuel rod, since the conduction in axial direction is small compared to the radial one, it can be neglected. An energy balance per unit length yields

$$m_f c_{pf} \frac{dT_f}{dt} = q'(t) - \frac{1}{R'_g} [T_f(t) - T_c(t)] \quad (15)$$

$$m_c c_{pc} \frac{dT_c}{dt} = \frac{1}{R'_g} [T_f(t) - T_c(t)] - \frac{1}{R'_c} [T_c(t) - T_m(t)] \quad (16)$$

where  $R'_g$  and  $R'_c$  represent thermal resistances per unit length. The coefficient of heat transfer to the refrigerant fluid is calculated by the Dittus-Boelter or Chen correlation, depending on the type of flow, which can be in one or two phases. These equations are used for the radial averaging of the temperatures in the fuel rod.

### 5.2. Reactor coolant dynamics

The conservation equations of mass, energy, and momentum are applied in this case to a flow of water along a vertical channel, where the dynamics of the fluid heated by the wall of the fuel is modeled. Conservation equations can be expressed as [10]

$$\frac{\partial \rho_m}{\partial t} + \frac{\partial G_m}{\partial z} = 0 \quad (17)$$

$$\frac{\partial G_m}{\partial t} + \frac{\partial}{\partial z} \left( \frac{G_m^2}{\rho_m^+} \right) = - \frac{\partial p}{\partial z} - \frac{f G_m |G_m|}{2 D_e \rho_m} - \rho_m g \cos \theta \quad (18)$$

$$\rho_m \frac{\partial h_m}{\partial t} + G_m \frac{\partial h_m}{\partial z} = \frac{q'' P_h}{A_z} + \frac{\partial p}{\partial t} + \frac{G_m}{\rho_m} \left( \frac{\partial p}{\partial z} + \frac{f G_m |G_m|}{2 D_e \rho_m} \right) \quad (19)$$

In this work, the conservation equations are solved by the Integral Moment method [11], according to which it is assumed that the refrigerant is incompressible but thermally expandable, and the density is a function of enthalpy at a constant pressure

$$\frac{\partial \rho_m}{\partial t} = \frac{\partial \rho_m}{\partial h_m} \bigg|_p \frac{\partial h_m}{\partial t} + \frac{\partial \rho_m}{\partial p} \bigg|_{h_m} \frac{\partial p}{\partial t} = R_h \frac{\partial h_m}{\partial t} + R_p \frac{\partial p}{\partial t} \quad (20)$$

Neglecting terms related to pressure changes and wall friction forces, the energy equation is simplified as

$$\rho_m \frac{\partial h_m}{\partial t} + G_m \frac{\partial h_m}{\partial z} = \frac{q'' P_h}{A_z} \quad (21)$$

where the axial flow variation can be obtained by

$$\frac{\partial G_m}{\partial z} = -\frac{R_h}{\rho_m} \left( \frac{q'' P_h}{A_z} - G_m \frac{\partial h_m}{\partial z} \right) \quad (22)$$

This equation provides the flow variations with respect to an average value imposed as a boundary value or provided by the dynamics of the coolant recirculation system. Three regions are defined by which the coolant circulates as it ascends into the channel: a one-phase region, a subcooled boiling region, and a bulk boiling region. The first region begins at the bottom of the channel, where the coolant enters with known enthalpy and ends at the point of separation of the bubbles  $Z_{sc}$ . The bulk temperature at this point is obtained by the Saha and Zuber correlation. The subcooled boiling region ends when the bulk temperature reaches the saturation temperature, and its axial location is determined by an energy balance. The enthalpy distribution allows the calculation of the thermodynamic equilibrium quality, used to calculate the flow quality. The axial distribution of the void fractions is calculated by iteratively solving the equation for void fraction  $\alpha$  and the Bankoff correlation slip (S):

$$\alpha = \frac{x}{S \left( \frac{\rho_g}{\rho_f} \right) + x \left( 1 - S \left( \frac{\rho_g}{\rho_f} \right) \right)}, S = \frac{1 - \alpha}{k_s - \alpha + (1 - k_s) \alpha^r}, \quad (23)$$

where, in this case, the parameters  $k_s$  and  $r$  are functions of the system pressure:

$$k_s = 0.71 + 1.2865 \times 10^{-3} p, \text{ and, } r = 3.33 - 2.56021 \times 10^{-3} p + 9.306 \times 10^{-5} p^2.$$

The total pressure drop in the channel is made up of the contributions of each region. Every term in each region includes the contribution by acceleration, gravity, and friction. For the channel arrangement, the steady state is obtained by iterating the coolant flow rate of each channel to obtain the same pressure drop for all of them. This iteration consists of a correction to the flow defined by the deviation of the pressure drop of the channel with respect to the average of all the channels:

$$G_i^{k+1} = G_i^k + w G_i^k \left( \frac{\bar{P}^k - P_i^k}{P_i^k} \right) \quad (24)$$

where  $G_i$  is the flow rate for channel  $i$ , the index  $k$  represents the number of the iteration,  $w$  is an arbitrary weight to control the convergence, and  $P$  is the average pressure drop of all channels at iteration  $k$ , obtained as

$$\bar{P}^k = \frac{1}{N} \sum_{i=1}^N P_i^k \quad (25)$$

It is observed that even though the pressures are equaled, the value of the pressure drop in the core is not imposed as a boundary condition. Convergence is achieved when the following relationship is met:  $\sum_{i=1}^N |\bar{P}^k - P_i^k| < \varepsilon$ . By changing the flow rate of the channel for each iteration, the enthalpy and void fraction profiles are affected. It is necessary to recalculate the TH solution at each iteration for all channels, achieving convergence when every parameter involved in the thermal-hydraulic calculation remains unchanged.

### 5.3. Neutron kinetics: thermal-hydraulics (NK-TH) coupling model

Although reference [12] has important issues to be considered in the development of an NK-TH-coupled model, those issues are not repeated here, but taken into account. The most direct way of coupling NK module and TH module, as implemented in AZKIND, consists simply in that axially both NK mesh and TH mesh have the same partition, making possible to assign an NK node at position  $z$  to the TH node in the same position. This relationship is a one-to-one node correspondence.

As it can be seen in **Figure 4**, before initiating the NK-TH feedback process, the initial nuclear parameters and kinetics parameter (XS) are loaded from files constructed in NEMTAB format, previously generated by means of a lattice code. Then, following the reading of the nuclear reactor burn-up state and thermo-physics initial conditions, the XS parameters are obtained from the Nemtab multi-dimensional tables by means of interpolation calculations.

The process continues as follows. The corresponding neutron flux is calculated in the NK module with the *mgcs* numerical solver, and this power (initial neutron flux) is the heat source to be assigned to the TH model. The axial power profile can be that of each fuel assembly assigned to a unique TH channel or the power profile of a set of fuel assemblies assigned to a TH channel. The axial power profile is the heat source for each node in the  $z$ -direction. Once the axial power profiles have been constructed in the TH module, an initial thermal-hydraulic state of the reactor system is calculated. The thermal-hydraulic state is calculated for each node in the TH channels from the bottom to the top of the reactor core.

The important variables sent to the NK module are the fuel temperature ( $T_f$ ), moderator temperature ( $T_m$ ), and moderator density ( $Dens$ ). The XS parameters are updated using these 3D variables for interpolation in the NEMTAB tables. The next step is to calculate new 3D power profiles to be sent to the TH module. This cyclic NK-TH calculation continues and stops when the TH criterion and neutron-flux criterion are met. Stopping the cyclic calculation means that the reactor power and thermal-hydraulic conditions have reached a steady state.

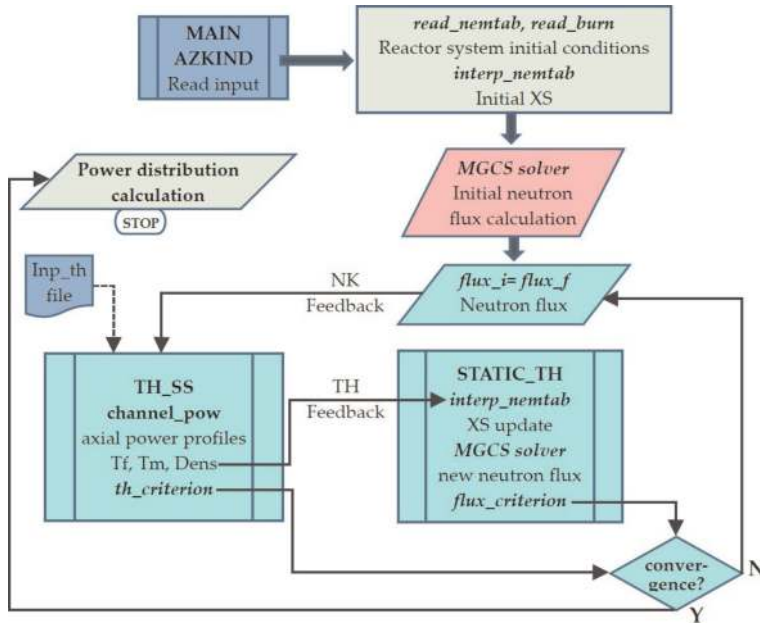


Figure 4. The NK-TH feedback process in AZKIND.

## 6. High-performance computing in AZKIND

### 6.1. PARALUTION linear algebra library

HPC was implemented in AZKIND with the support of the linear algebra solvers library PARALUTION [13]. This open-source library is optimized for parallel computing process using graphics processing units (GPUs). For the numerical solution of an algebraic system  $A \vec{v} = \vec{b}$  PARALUTION includes numerical solvers to obtain the solution vector  $\vec{v}$  for a known vector  $\vec{b}$  and a specific matrix  $A$  that can be a symmetric or a non-symmetric matrix being also a sparse or a dense matrix. The working matrices in AZKIND are sparse non-symmetric matrices, and the *bicgstab* solver [14] was used for reactor simulations. The matrix solvers in PARALUTION are optimized to use on the non-zero (*nnz*) elements in the working matrices, saving processing time and computer memory.

### 6.2. Parallel processing for neutronic model

To demonstrate the HPC implementation in AZKIND, as described in Ref. [1], very large matrices were constructed for fine spatial discretization of arrangements of nuclear fuel

assemblies of an LWR. Fine discretization means that each fuel assembly was subdivided in a mesh of size  $10 \times 10$ . As an example, an arrangement of  $6 \times 6$  fuel assemblies consists of a square with 36 fine-discretized fuel assemblies. The corresponding algebraic system for each fuel arrangement was solved with parallel processing performed by the *bicgstab* solver mentioned earlier. In **Tables 2** and **3**, the speedup of the different cases is shown [1] with a remarkable performance. Despite the speedup for small matrices that is comparable for the three computer architectures used, it is also important to notice that the speedup values listed in **Table 3** do not present a linear behavior, and the reason is because although more GPU processor cores are used with massive data transference to and from the GPU, a data traffic delay is present in the communication bus between the GPU and the CPU. For the analysis of the computing acceleration or “speedup,” a definition of speedup is used in [15], known as relative speedup or speedup ratio:  $S = T_1/T_n$ , where  $T_1$  is the computing time using a single processor (serial calculation) and  $T_n$  is the computing time using  $n$  processor cores. The “no memory” insert listed in **Table 2** is because for those large matrix dimensions, there is not enough memory to load the matrix and solvers.

**Figure 5** [1] shows the distribution of nuclear fuel assemblies in the core of a boiling water reactor. Excepting the blue-shaded zone, colors are for different types of fuel assemblies. In the plane  $xy$ , the mesh is  $24 \times 24$ , according to each fuel zone, and axially, there are 25 nodes. The matrix for this coarse mesh (1,274,304 *mmz*) is comparable to the matrix of the fine mesh created for the case of a unique assembly (case  $1 \times 1$  listed in **Table 2**).

As described in [1], a reactor power transient was simulated as the capability to remove neutrons was highly increased in the perturbed assembly shown in **Figure 5**. An increase as

| Assemblies array:         | $1 \times 1$ | $2 \times 2$ | $4 \times 4$ | $6 \times 6$ | $10 \times 10$ |
|---------------------------|--------------|--------------|--------------|--------------|----------------|
| Matrix dimension ( $m$ ): | 126,200      | 492,800      | 1,947,200    | 4,363,200    | 12,080,000     |
| <i>mmz</i> elements:      | 1,332,400    | 5,305,600    | 21,174,400   | 47,606,400   | 132,160,000    |
| Serial                    | 24           | 124          | 372          | 994          | 2471           |
| GTX 860 M                 | 2.1          | 7.9          | 31.3         | No memory    | No memory      |
| Tesla K20c                | 1.3          | 4.0          | 16.6         | 40.1         | No memory      |
| GTX TITAN X               | 1.0          | 2.6          | 10.4         | 26.7         | 95.4           |

**Table 2.** Parallel processing time (seconds) in different architectures [1].

| Assemblies  | $1 \times 1$ | $2 \times 2$ | $4 \times 4$ | $6 \times 6$ | $10 \times 10$ |
|-------------|--------------|--------------|--------------|--------------|----------------|
| GTX 860 M   | 11           | 16           | 12           | –            | –              |
| Tesla K20c  | 18           | 31           | 22           | 24           | –              |
| GTX TITAN X | 24           | 48           | 36           | 37           | 26             |

**Table 3.** Speedup comparison (S) [1].

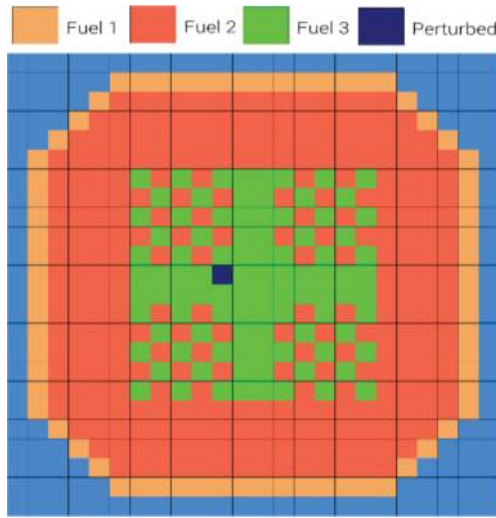


Figure 5. A map of fuel assemblies in an LWR [1].

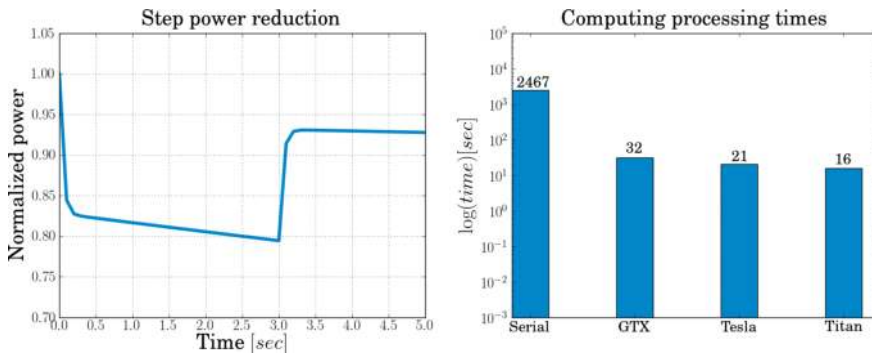


Figure 6. Simulation of a reactor power transient—serial and parallel processing.

step function in the neutrons removal capability during 3 s is implemented in the perturbed assembly, after that the perturbation finishes and the transient lasts for two more seconds, giving a reactor power reduction. The time step used in this simulation was 0.1 s. **Figure 6** shows the power behavior over time, departing from a normalized value of 1.0 and reducing the power reactor to almost 80% of its original value. This reactor power transient was simulated with the AZKIND code, running on the three different GPUs listed in **Tables 2** and **3**. The right side of **Figure 6** shows the time spent by AZKIND in a logarithmic scale, running in a sequential mode (Serial bar) and the times spent by each GPU card.



## 7. Simulation of a reactor core condition

A simple example was prepared to show the capability of the AZKIND code running with NK-TH coupling, and the thermal-hydraulic effect on power distribution is compared to the power distribution resulted from the NK model running standalone.

This example was prepared for a two energy group, that is, fast neutrons and thermal neutrons. In LWR, the nuclear fissions of the fuel atoms are mainly coming from the thermal neutrons present in the reactor core. The effect observed in **Figure 7** is that the TH feedback induces an increase in the thermal neutrons population and so increasing power. As the coolant/moderator enters the reactor core through the bottom part of the reactor and the core

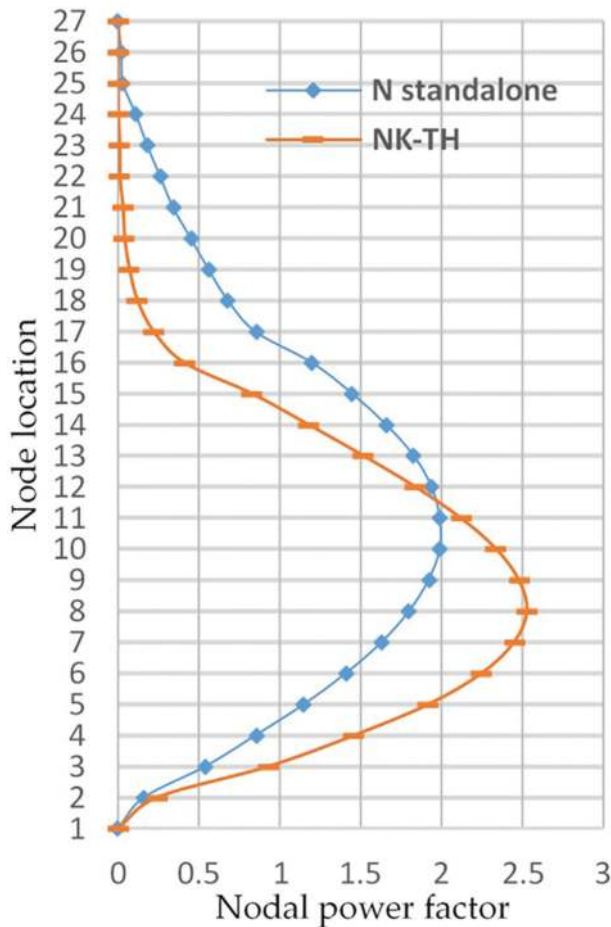


Figure 7. Axial power peaking profile location.

is beginning the production cycle, the core design allows more power generation in the first third of the core active fuel. Also, as it was expected, in the map of fuel assemblies of the reactor core, the location of the fuel assembly with the highest generation of thermal power remained unchanged with the insertion of TH feedback.

## 8. Some advances on nuclear reactor simulation

In the last two decades, there have been significant advances in the development of nuclear reactor codes for 3D simulation with coupling NK-TH, supported with new modeling techniques and modern computing capabilities in software and hardware. Some examples of these advances are listed subsequently:

1. DYN3D: Pin-based coupling of the simplified transport (SP3) version of DYN3D with the sub-channel code SUBCHANFLOW. See [16, 17]. The new coupled code system allows for a more realistic description of the core behavior under steady state and transient conditions. DYN3D has successfully been applied to analyze the behavior of one eighth of a PWR core during an REA transient by a pin-by-pin simulation consisting of a huge number of nodes. Some insights are pointed out on the convergence process with a detailed coupling solution modeling neighbor sub-channels and modeling adjacent assembly channels.
2. DYN3D: The code comprises various 3D neutron kinetics solvers, a thermal-hydraulics reactor core model, and a thermo-mechanical fuel rod model, see [18]. The following topics are delineated in the reference: the latest developments of models and methods, a status of verification and validation; code applications for selected safety analyses; multi-physics code couplings to thermal-hydraulic system codes, CFD, and sub-channel codes as well as to the fuel performance code TRANSURANUS.
3. TRACE/PARCS: See [19]. The study of the coupling capability of the TRACE and PARCS codes by analyzing the “Main Steam Line Break (MSLB) benchmark problem,” consisting of a double-ended MSLB accident assumed to occur in the Babcock and Wilcox Three Mile Island Unit 1. The model TRACE/PARCS generated data showing that these codes have the capability to predict expected phenomena typical of this transient and the related NK-TH feedback.
4. COBAYA3: See [20]. This reference describes a multi-physics system of codes including the 3D multi-group neutron diffusion codes, ANDES and COBAYA3-PBP, coupled with the sub-channel thermal-hydraulic codes COBRA-TF, COBRA-IIIc, and SUBCHANFLOW, for the simulation of LWR core transients. Implementation of the PARALUTION library to solve sparse systems of linear equations was done. It features several types of iterative solvers and preconditioners which can run on both multi-core CPUs and GPU devices without any modification from the interface point of view. By exploring this technology, namely the implementation of the PARALUTION library in COBAYA3, the code can decrease the solution time of the sparse linear systems by a factor of 5.15 on GPU and 2.56 on a multi-core CPU using standard hardware.

5. CNFR: See [21]. This reference summarizes three methods, implemented for multi-core CPU and GPU, to evaluate fuel burn-up in a pressurized light water nuclear reactor (PWR) using the solutions of a large system of coupled ordinary differential equations. The reactor physics simulation of a PWR with burn-up calculations spends long execution times, so that performance improvement using GPU can imply in a better core design and thus extended fuel life cycle. The results with parallel computing exhibit speed improvement exceeding 200 times over the sequential solver, within 1% accuracy.

## 9. Conclusions and remarks

The state of the art in the topic of nuclear reactor simulations shows significant advances in the development of computer codes. A wide range of applications focusing, besides on improving nuclear safety, on more efficient analyses to improve fuel cycles/depletion have been found in a recent study. A considerable “saving time” factor in obtaining nuclear reactor analyses has been observed.

One important part of a nuclear reactor simulator is the benchmarking process to demonstrate reliability and repeatability in the simulation of real cases, for which data from reactor operation or comprehensive data from experiments are well documented. In this sense, extensive documentation is necessary for theoretical basis, numerical techniques and tools, and validation of both codes and simulation models.

## Author details

Andrés Rodríguez Hernández<sup>1</sup>, Armando Miguel Gómez-Torres<sup>1\*</sup> and Edmundo del Valle-Gallegos<sup>2</sup>

\*Address all correspondence to: [armando.gomez@inin.gob.mx](mailto:armando.gomez@inin.gob.mx)

1 Instituto Nacional de Investigaciones Nucleares, Ocoyoacac, Edo. de México, México

2 Instituto Politécnico Nacional, Escuela Superior de Física y Matemáticas, Col. San Pedro Zacatenco, Cd. de México, México

## References

- [1] Rodríguez-Hernández A, Gomez-Torres A, Del Valle-Gallegos E. HPC implementation in the time-dependent neutron diffusion code AZKIND. *Annals of Nuclear Energy*. 2017;**99**: 174-182
- [2] Oka Y, editor. *Nuclear Reactor Design. (Series) An Advanced Course in Nuclear Engineering*. Japan: Springer; 2014

- [3] Fridman E, Leppänen J. Revised methods for few-group cross sections generation in the Serpent Monte Carlo code. PHYSOR 2012 – Advances in Reactor Physics. LaGrange Park, IL: American Nuclear Society; 2010
- [4] Reedmond EL. Multigroup cross section generation via Monte Carlo methods [PhD thesis]. Massachusetts Institute of Technology; 1998
- [5] Duderstadt JJ, Hamilton LJ. Nuclear Reactor Analysis. New York: John Wiley and Sons; 1976
- [6] Rodríguez-Hernández A. Solution of the nuclear reactor kinetics equations in 3D using the nodal method RTN-0 (in Spanish) [MSc thesis]. México: National Polytechnic Institute, ESFM; 2002
- [7] Grossman LM, Hennart JP. Nodal diffusion methods for space-time neutron kinetics. Progress in Nuclear Energy. 2007;**49**:181-216
- [8] Weismann J, editor. Elements of Nuclear Reactor Design. Elsevier Scientific Publishing Company; 1977
- [9] Ceceñas M, Campos R. Modelo acoplado de canales en paralelo y cinética neutrónica en dos dimensiones. In: International Joint Meeting Cancun 2004 LAS/ANS-SNM-SMSR. Cancun, Mexico; 2004
- [10] Todreas NE, Kazimi MS. Nuclear Systems I: Thermal Hydraulic Fundamentals. USA: Hemisphere Publishing; 1989
- [11] Todreas NE, Kazimi MS. Nuclear Systems II: Elements of Thermal Hydraulic Design. USA: Hemisphere Publishing; 1990
- [12] Ivanov K, Avramova M. Challenges in coupled thermal–hydraulics and neutronics simulations for LWR safety analysis. Annals of Nuclear Energy. 2007;**34**:501-513
- [13] Lukarski D. PARALUTION Project, Version 0.8.0. 2014. <http://www.paralution.com/>
- [14] Van der Vorst HA. Efficient and reliable iterative methods for linear systems. J. of Computational and Applied Mathematics. 2002;**149**:251-265
- [15] Nesmachnow S. Workshop Scientific computing on distributed memory systems. In: International Supercomputing Conference ISUM; Mexico; 2015
- [16] Gomez-Torres AM, Sanchez-Espinoza VH, Ivanov K, Macian-Juan R. DYNSUB: A high fidelity coupled code system for the evaluation of local safety parameters—Part I: Development, implementation and verification. Annals of Nuclear Energy. 2012;**48**:108-122
- [17] Gomez-Torres AM, Sanchez-Espinoza VH, Ivanov K, Macian-Juan R. DYNSUB: A high fidelity coupled code system for the evaluation of local safety parameters—Part II: Comparison of different temporal schemes. Annals of Nuclear Energy. 2012;**48**:123-129

- [18] Rohde U, Kliem S, Baier S, Bilodid Y, Duerigen S, Fridman E, Gommlich A, Grahn A, Holt L, Kozmenkov Y, Mittag S. The reactor dynamics code DYN3D e models, validation and applications. *Progress in Nuclear Energy*. 2016;**89**:170-190
- [19] Mascari F, Vella G, Casamassima V, Parozzi F. Analyses of TRACE-PARCS coupling capability. In: *International Conference on the Physics of Reactors. 20th International Conference—Nuclear Energy for New Europe; Slovenia; 2011*
- [20] Trost N, Jimenez J, Lukarski D, Sanchez V. Accelerating COBAYA3 on multi-core CPU and GPU systems using PARALUTION. *Annals of Nuclear Energy*. 2014;**82**:252-259
- [21] Heimlich A, Silva FC, Martinez AS. Parallel GPU implementation of PWR reactor burnup. *Annals of Nuclear Energy*. 2016;**91**:135-141

