Nuclear Engineering and Technology 53 (2021) 399–413

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Adaptive time-step control for modal methods to integrate the neutron diffusion equation

A. Carreño [a], A. Vidal-Ferràndiz [a], D. Ginestar [b, *], G. Verdu [a]

[a] Instituto Universitario de Seguridad Industrial, Radiofísica y Medioambiental, Universitat Politècnica de València, Camino de Vera, s/n, 46022, Valencia, Spain
[b] Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera, s/n, 46022, Valencia, Spain

ARTICLE INFO

Article history:
Received 24 March 2020
Received in revised form
10 June 2020
Accepted 6 July 2020
Available online 17 August 2020

Keywords:
Adaptive time-step
Modal method
Finite element method
Time dependent Neutron diffusion equation

ABSTRACT

The solution of the time-dependent neutron diffusion equation can be approximated using quasi-static methods that factorise the neutronic flux as the product of a time dependent function times a shape function that depends both on space and time. A generalization of this technique is the updated modal method. This strategy assumes that the neutron flux can be decomposed into a sum of amplitudes multiplied by some shape functions. These functions, known as modes, come from the solution of the eigenvalue problems associated with the static neutron diffusion equation that are being updated along the transient. In previous works, the time step used to update the modes is set to a fixed value and this implies the need of using small time-steps to obtain accurate results and, consequently, a high computational cost. In this work, we propose the use of an adaptive control time-step that reduces automatically the time-step when the algorithm detects large errors and increases this value when it is not necessary to use small steps. Several strategies to compute the modes updating time step are proposed and their performance is tested for different transients in benchmark reactors with rectangular and hexagonal geometry.

1. Introduction

The solution of time-dependent neutron diffusion equations approximates the neutron distribution that describes reactor kinetics. This equation with the energy multigroup approximation depends on the position and time. Different spatial discretizations can be considered that lead to a time-dependent ordinary differential system of equations (ODE). In this work, a high order continuous Galerkin finite element method is used to make the spatial discretization of the equations yielding to a time-dependent system of ODE. This method can be applied to different reactor geometries, including rectangular and hexagonal geometries [1]. Some characteristics of the dynamics of a nuclear reactor, such as the presence of both prompt and delayed neutrons, usually produce a so-called stiff problem in time. It means that obtaining the solution of this system efficiently will depend heavily on the methodology used as well as the time steps for integration.

Several approaches have been proposed to solve the time-dependent neutron diffusion equation. One possibility is to use an approximation of the differential operator based on the implicit Euler method [2]. Other type of method is the point kinetic approximation, that eliminates the spatial dependence of the solution obtaining a differential equation for an amplitude function. However, in some transients, mainly when local effects are relevant, it is not possible to represent the evolution of the neutron flux with the amplitude function. One way to improve this approximation is the quasi-static method. This methodology factorizes the solution as a product of an amplitude function (what has a fast change over time) times a shape function (that changes slowly in time). The shape function is updated using a differential scheme a certain interval of time [3–5]. The recomputation of this spatial shape is usually the most expensive part. Another strategy to improve the point kinetics approximation is the adiabatic method that obtains the shape functions by static computations that are updated at specific time points during the transients [6–8]. Finally, we have the modal method [[8,9]]. It assumes that the solution can be described by the sum of several amplitude functions multiplied by some shape functions. These shape functions can be computed

* Corresponding author.
E-mail addresses: amcarsan@iqn.upv.es (A. Carreño), anvifer2@upv.es (A. Vidal-Ferràndiz), dginesta@mat.upv.es (D. Ginestar), gverdu@iqn.upv.es (G. Verdu).

https://doi.org/10.1016/j.net.2020.07.004

by solving eigenvalue problems for different spatial modes [10]. This last approach has a great interest when the neutron power distribution cannot be well approximated by using only one shape function, mainly, when local perturbations are applied during the transient [10]. In this work, the eigenfunctions are updated to avoid to use a high number of modes.

Classically, the implementation of all these methods uses small fix time-steps over which there are little changes in the neutron population, to ensure the stability of the solution and this has a high computational cost. One way to avoid to use small time-steps is using high order schemes such as the backward differential methods proposed in Ref. [11]. However, obtaining the approximations at each time-step is also computationally expensive. Other approach very common in the methods to approximate the solution of large systems of differential equations is the implementation of adaptive control-step where the time-step to be used in the discretization is computed step by step. Theoretically, this methodology picks up an optimal time-step adaptively based on the current state of the transient. In practice, this time-step is selected from the solutions computed in the previous steps to satisfy some approximated error tolerance [12–14]. This kind of methodology improves computational efficiency because time is not wasted solving the system to a level of accuracy beyond relevance, but also it allows to reduce the step size if a numerical instability is detected. Moreover, adaptive schemes remove the necessity for the user to select appropriate time steps before the simulation has been run.

Recent works have incorporated an adaptive time-step for the Backward differential method for different orders [15–18]. The quasi-static methodology is also implemented by selecting an appropriate step size from a given tolerance [19]. Step size controllers require an error estimation. The time step selection for these works are based on some approximations of the local truncation error caused by the time discretization, since both works use a finite difference discretization.

The aim of this paper is to study updated modal methods and to propose an efficient strategy to integrate the time-dependent neutron diffusion equation based on an adaptive control for the time-step used to update the modes along the transient. In this way, different error estimations must be defined. Several error approximations can be proposed to select the time-step along the transient. First, an error estimator based on the change in the neutron power is considered. Another error estimator is defined from the residual error of the associated static problem corresponding to the modes computation. Finally, the last error estimator considered is based on the change on the cross-sections along the transient.

Moreover, the memory resources and the CPU time to assemble the matrices at each time-step are in general very large. Thus, the implementation of the modal method will be compatible with a matrix-free methodology [20,21].

The structure of the rest of the paper is as follows. Section 2 presents the time dependent neutron diffusion equation and the λ-modes problem in the approximation of two energy groups. This Section also briefly describes the spatial discretization used for the differential equations and the definition of the adjoint problems associated for the λ-modes problems. Section 3 describes the matrix-free strategy. Section 4 presents the backward differential method. Section 5 includes the development of the updated modal kinetics equations associated with the different spatial modes. Section 6 presents the theory related to the adaptive time-step control for the update modal method. Section 8 collects numerical results to test the performance of the adaptive time-step control for modal equations in several different benchmark problems. Finally, Section 9 synthesizes the main conclusions of this paper.

2. Time dependent neutron diffusion equation

In the following, we develop the methodology for the two energy groups neutron diffusion approximation. However, this strategy can be likewise applied for more energy groups and other angular approximations of the neutron transport such as the S_N or the P_N equations. Nowadays, the integration of these problems with differential or quasi-static methods can be very slow [22], and the modal methods can be an interesting alternative to study some type of transients with higher order approximations of the neutron transport.

For a given transient in a nuclear reactor, the neutronic flux inside the core can be described by means of the time dependent neutron diffusion equation with two energy groups, without up-scattering and K groups of delayed neutron precursors [23]. These equations are expressed as

$$\begin{aligned}
&\mathscr{V}^{-1}\frac{\partial\Phi}{\partial t}+(\mathscr{L}+\mathscr{S})\Phi=(1-\beta)\chi^{p}\mathscr{F}\Phi+\sum_{k=1}^{K}\lambda_k^d\mathscr{C}_k\chi^{d,k},\\
&\frac{dC_k}{dt}(\vec{r},t)=\beta_k\mathscr{F}_1\Phi-\lambda_k^dC_k,\qquad k=1,\ldots,K,
\end{aligned}\tag{1}$$

where,

$$\mathscr{L}=\begin{pmatrix}-\vec{\nabla}\cdot(D_1\vec{\nabla})+\Sigma_{a_1}+\Sigma_{12} & 0\\ 0 & -\vec{\nabla}\cdot(D_2\vec{\nabla})+\Sigma_{a_2}\end{pmatrix},$$

$$\mathscr{S}=\begin{pmatrix}0 & 0\\ -\Sigma_{12} & 0\end{pmatrix},\quad \mathscr{F}=\begin{pmatrix}\nu_1\Sigma_{f_1} & \nu_2\Sigma_{f_2}\\ 0 & 0\end{pmatrix},\quad \mathscr{F}_1=\begin{pmatrix}\nu\Sigma_{f_1} & \nu\Sigma_{f_2}\end{pmatrix},$$

$$\mathscr{V}^{-1}=\begin{pmatrix}\nu_1^{-1} & 0\\ 0 & \nu_2^{-1}\end{pmatrix},\quad \chi^p=\chi^{d,k}=\begin{pmatrix}1\\ 0\end{pmatrix},\quad \Phi=\begin{pmatrix}\Phi_1\\ \Phi_2\end{pmatrix}.$$

The operator $\mathscr{L}$ is known as leakage operator, the operator $\mathscr{S}$, the scattering operator and $\mathscr{F}$, the fission operator. The vectors Φ_1 and Φ_2 represent the fast and thermal fluxes of neutrons, respectively. The term $\mathscr{C}_k$ is the concentration of delayed neutron precursors of group k. The fission spectrum is denoted by χ. The average number of neutrons produced in each fission and energy group g is ν_g. The rest of the coefficients, where the subindex represent the energy group g ($g=1,2$), are: the diffusion coefficients, D_g; the absorption cross sections, Σ_{ag} and the fission cross sections, Σ_{fg}. The scattering cross section from group 1 to group 2 is Σ_{12}. The neutron velocities are ν_1, ν_2. The spectrum of the prompt neutrons χ^p and the delayed neutrons $\chi^{d,k}$, in this special case of two energy groups, are considered equal. All of these coefficients are, in general, position and time dependent functions.

Given a configuration of the reactor, the criticality can be forced of several forms transforming Equation (1) into several time-independent eigenvalue problems. If the fission operator is divided by a positive number, λ_m, the λ-modes equation is obtained as

$$(\mathscr{L}+\mathscr{S})\varphi_m=\frac{1}{\lambda_m}\mathscr{F}\varphi_m.\tag{2}$$

A Galerkin high order finite element method has been used for the spatial discretization of equation (1) making use of Lagrange polynomials. More details can be found in Ref. [1,24]. The finite element method has been implemented by using the open source finite elements library `Deal.ii` [25]. In this way, equations (1) can be approximated by solving the time dependent system of ordinary differential equations,

$$V^{-1}\frac{d\tilde{\Phi}}{dt}+(L+S)\tilde{\Phi}=(1-\beta)F\tilde{\Phi}+\sum_{k=1}^{K}\lambda_k^d XC_k,$$
$$\frac{dC_k}{dt}=\beta_k F_1\tilde{\Phi}-\lambda_k^d C_k,\qquad k=1,\ldots,K, \tag{3}$$

where L, S, F, F_1 are the matrices obtained from the discretization of operators $\mathscr{L}$, $\mathscr{S}$, $\mathscr{F}$, $\mathscr{F}_1$, respectively. Vectors $\tilde{\Phi}$ and C_k are the corresponding coefficients of Φ and $\mathscr{C}_k$ in terms of the Lagrange polynomials. The matrices X and $[V^{-1}]$ are, then, defined as

$$X=\begin{pmatrix} I \\ 0 \end{pmatrix},\qquad V^{-1}=\begin{pmatrix} Pv_1^{-1} & 0 \\ 0 & Pv_2^{-1} \end{pmatrix}, \tag{4}$$

where I is the identity matrix and P is the mass matrix related to the spatial discretization, which is different from the identity matrix because the Lagrange basis of polynomials is not orthonormal.

For λ-modes problem (Equations (2)), the algebraic problem associated after the discretization has the following structure

$$(L+S)\tilde{\varphi}_m=\frac{1}{\lambda_m}F\tilde{\varphi}_m, \tag{5}$$

where $\tilde{\varphi}$ is the algebraic vectors of weights associated with the function vectors φ. In the following, to simplify the notation, the algebraic vectors are denoted by Φ(for the time-dependent flux) and φ (for the eigenfunctions) by removing the tildes from the original notation.

Associated to the λ-modes problem one can introduce the adjoint problem

$$(L+S)^T\varphi_l^\dagger=\frac{1}{\lambda_l}F^T\varphi_l^\dagger, \tag{6}$$

where L^T, S^T and F^T are the transpose matrices of L, S and F, respectively [8]. They are equal to the matrices obtained from the discretization of the adjoint operators, $\mathscr{L}^\dagger$, $\mathscr{S}^\dagger$ and $\mathscr{F}^\dagger$. The adjoint modes, solutions of (6), $\varphi_l^\dagger$, $l=1,\ldots,q$ satisfy the biorthogonality condition

$$\langle\varphi_l^\dagger,F\varphi_m\rangle=\langle\varphi_m^\dagger,F\varphi_m\rangle\delta_{l,m},\quad l,m=1,\ldots,q, \tag{7}$$

where $\langle\ ,\rangle$ is the scalar product for vectors and $\delta_{l,m}$ is the Kronecker's delta.

3. Matrix-free implementation

The schemes proposed to solve the semi-discrete time dependent neutron diffusion equations (3) require the updating of the different matrices at each time-step. In this work, a matrix-free strategy is used to avoid the full assembly of the matrices. In this way, the CPU time and the CPU memory needed to allocate the matrices are removed. To apply this technique, matrix-vector products are computed 'on the fly' in a cell-based interface.

For instance, one can consider that the finite element Galerkin approximation leads to the matrix A and takes a vector u as input. A module has been built such that it computes the integrals associated with the different matrix elements and the output vector v is computed. This operation can be expressed as a sum of N_c cell-based operations,

$$v=Au=\sum_{c=1}^{N_c}P_c^TA^cP_cu, \tag{8}$$

where P_c denotes the matrix that defines the location of cell-related degrees of freedom in the global vector and A^c denotes the submatrix of A on cell c. This sum is optimized through a *sum-factorization* (see more details in Ref. [21]). This strategy minimizes the memory used by the matrix elements and it can improve matrix-vector multiplication computational times.

The main difficult of this strategy is to obtain efficient algebraic solvers that only use matrix-vector multiplications. For this reason, two types of implementations are used in this work depending on the integration scheme used. For the Backward differential method, a semi matrix-free implementation is used, where only the diagonal blocks of the matrix T^n (described in Section 4) are allocated in memory. For the modal method, any matrix obtained from the discretization of the operator is assembled.

4. Backward differential method

The time-dependent system of ordinary differential equations (3) is, in general, stiff, thus the use of implicit methods is necessary. Particularly, the backward differential method, that is a classical implicit method, is used [11,24,26].

This method starts with the steady-state flux, i.e., the approximated $\Phi(0)=\varphi$ obtained from the solution of the problem (5), where the system is made critical by dividing the matrix F by $k_{\text{eff}}=\lambda_1$. An approximated solution in the $(n+1)$-th step, Φ^{n+1}, with a time-step of $\Delta t_n=t_{n+1}-t_n$, is obtained by solving the following system of linear equations

$$T^{n+1}\Phi^{n+1}=R^n\Phi^n+\sum_{k=1}^{K}\lambda_k^d e^{-\lambda_k\Delta t_n}XC_k^n\,, \tag{9}$$

where the matrices are defined as

$$T^{n+1}=R^n+L^{n+1}+S^{n+1}-\hat{a}F^{n+1},\qquad R^n=\frac{1}{h_n}V^{-1}\,,$$

and the coefficient $\hat{a}$ is computed as

$$\hat{a}=1-\beta+\sum_{k=1}^{K}\beta_k\left(1-e^{-\lambda_k^d h_n}\right).$$

The neutron precursors equation is discretized in time by using the explicit scheme

$$C_k^{n+1}=C_k^ne^{-\lambda_k h_n}+\frac{\beta_k}{\lambda_k}\left(1-e^{-\lambda_k h_n}\right)\left(F_{11}^{n+1}F_{12}^{n+1}\right)\Phi^{n+1}\,,k=1,\ldots,K. \tag{10}$$

This system of equations is large and sparse and has to be solved for each new time-step, t_{n+1}. In this work, the GMRES method has been used to solve the linear systems preconditioned with the block Gauss-Seidel method (see more details in Ref. [20]). The inverses in the Gauss-Seidel method are approximated solving linear systems with the conjugate gradient method with the ILU preconditioner and a residual error of 10^{-6}. This strategy permits to avoid the assembly of the full matrices at each time-step.

5. Updated modal method

The time-dependent neutron diffusion equation (3) can also be

solved by using a modal method [6–9]. For that, it is supposed that $\Phi(\overrightarrow{r},t)$ can be expanded as,

$$\Phi(\overrightarrow{r},t)=\sum_{m=1}^{q} n_m(t)\varphi_m(\overrightarrow{r}), \tag{11}$$

where $\varphi_m(\overrightarrow{r})$ is the unitary eigenvector associated with the m-th dominant eigenvalue of the λ-modes problem (5) and q is the number of dominant modes considered. The amplitude coefficients $n_m(t)$ are only time dependent. To simplify the notation, we will write n_m and φ_m instead of $n_m(t)$and $\varphi_m(\overrightarrow{r})$.

After that, we choose the matrices L, S and F, that we denote by L_0, S_0 and F_0, as the matrices related to the configuration of the reactor core at $t=0$. We start with this reactor in critical state by dividing the matrix related to the fission terms (F_0) by $k_{eff}=\lambda_1$. In this way, we express the matrices L, S and F as

$$L=L_0+\delta L,\qquad S=S_0+\delta S,\qquad F=F_0+\delta F. \tag{12}$$

Now, if one substitutes the expansion (11) into the time dependent neutron diffusion equation (Equation (3)), the λ-modes problem is taken into account and it is multiplied on the left by the adjoint modes $\varphi_l^{\dagger}$ ($l=1,\ldots,q$), one can obtain the coefficients $n_l$by solving the following system of $q(K+1)$equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{N}=\mathbf{T}\mathbf{N}, \tag{13}$$

where

$$\mathbf{N}^{\lambda}=\left(n_1\cdots n_q\quad c_{11}\cdots c_{q1}\quad\cdots\quad c_{1K}\cdots c_{qK}\right)T, \tag{14}$$

$$\mathbf{T}=\begin{pmatrix}\Lambda^{-1}\left((1-\beta)I-[\lambda]^{-1}-\Delta L-\Delta S+(1-\beta)\Delta F\right)&\Lambda^{-1}\lambda_1^d\cdots\Lambda^{-1}\lambda_K^d\\ \beta_1(I+\Delta F)&-\lambda_1^dI\ \cdots\ 0\\ \vdots&\vdots\ \ddots\ \vdots\\ \beta_K(I+\Delta F)&0\ \cdots\ -\lambda_K^dI\end{pmatrix}, \tag{15}$$

and

$$\begin{aligned}&\Lambda_{lm}=\langle\varphi_l^{\dagger},V^{-1}\varphi_m\rangle,\quad \Delta L_{lm}=\langle\varphi_l^{\dagger},\delta L\varphi_m\rangle,\\ &\Delta F_{lm}=\langle\varphi_l^{\dagger},\delta F\varphi_m\rangle,\qquad c_{lk}^{\lambda}=\langle\varphi_l^{\dagger},XC_k\rangle,\\ &\Delta S_{lm}=\langle\varphi_l^{\dagger},\delta S\varphi_m\rangle.\end{aligned} \tag{16}$$

The block $[\lambda]$ denotes the diagonal matrix whose elements are the dominant λ-modes. The initial conditions in the steady state are

$$n_1(0)=1,\qquad n_m(0)=0,\ m=2,\ldots,q$$

$$c_{1k}(0)=\frac{\beta_k}{\lambda_k^d}\langle\varphi_1^{\dagger},F_0\varphi_1\rangle,\quad c_{mk}(0)=0,\ m=2,\ldots,q,\quad k=1,\ldots,K,$$

with φ_1 and $\varphi_1^{\dagger}$, the corresponding eigenvector and its adjoint associated with the dominant eigenvalue λ_1.

5.1. Updating the modal method

In realistic transient computations, the neutronic flux shape in the reactor over time can suffer large spatial variations with respect to the initial flux. Thus, in order to obtain good approximations by using the modal method, high number of modes would be necessary in the modal expansion [10], which implies a large computational cost in the calculation. A solution for this challenge was proposed in Ref. [9], where a small number of modes are computed but they are updated from time to time, generalizing the quasi-static method by using the modes as several shape functions.

Moreover, the modal methodology with one eigenvector in its expansion is known as the adiabatic method [8]. It is well-known that this method suffers problems in its convergence to the solution if the eigenfunction is not well suited to represent the evolution of the system under analysis [27]. Using for the representation a higher number of modes (eigenvectors) logically improves the accuracy of the solution.

In this method, the time domain is divided into several intervals $[t_i,t_i+\Delta t_i]=[t_i,t_{i+1}]$. In each interval $[t_i,t_{i+1}]$, the neutron diffusion equation can be integrated by using the λ-modes associated with the problem

$$\left(L^i+S^i\right)\varphi_m^i=\frac{1}{\lambda_m^i}F^i\varphi_m^i, \tag{17}$$

where L^i, S^i and F^i are the matrices associated with the reactor configuration at time t_i, and the differential equations that are needed to be solved are of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{N}^{i,\lambda}=\mathbf{T}^{i,\lambda}\mathbf{N}^{i,\lambda}, \tag{18}$$

where

$$\mathbf{N}^i=\left(n_1^i\cdots n_q^i\quad c_{11}^i\cdots c_{q1}^i\quad\cdots\quad c_{1K}^i\cdots c_{qK}^i\right)T, \tag{19}$$

$$\mathbf{T}^i=\begin{pmatrix}\Lambda_i^{-1}\left((1-\beta)I-[\lambda_i]^{-1}-\Delta L^i-\Delta S^i+(1-\beta)\Delta F^i\right)&\Lambda_i^{-1}\lambda_1^d\cdots\Lambda_i^{-1}\lambda_K^d\\ \beta_1\left(I+\Delta F^i\right)&-\lambda_1^dI\ \cdots\ 0\\ \vdots&\vdots\ \ddots\ \vdots\\ \beta_K\left(I+\Delta F^i\right)&0\ \cdots\ -\lambda_K^dI\end{pmatrix}, \tag{20}$$

and

$$\begin{aligned}&\Lambda_{lm}^i=\langle\varphi_l^{\dagger,i},V^{-1}\varphi_m\rangle,\quad \Delta L_{lm}^i=\langle\varphi_l^{\dagger,i},\delta L^i\varphi_m^i\rangle,\\ &\Delta F_{lm}^i=\langle\varphi_l^{\dagger,i},\delta F^i\varphi_m^i\rangle,\qquad c_{lk}^{\lambda}=\langle\varphi_l^{\dagger,i},XC_k\rangle,\\ &\Delta S_{lm}^i=\langle\varphi_l^{\dagger,i},\delta S^i\varphi_m^i\rangle.\end{aligned} \tag{21}$$

where the operators δL^i, δS^i and δF^ialso change into the interval $[t_i,t_{i+1}]$. Observe that this system is the same as the one in the modal method without modes updating, but matrices L^i, S^i, F^i and δL^i, $\delta S^i$and δF^i have to be updated. Nevertheless, the initial conditions (at time t_i) must be reformulated to ensure the continuity of the solution. These initial conditions will depend on the solution in the previous interval $[t_{i-1},t_i]$, the eigenvectors associated with direct modes (φ_m^i) and the eigenvectors associated with adjoint modes ($\varphi_l^{\dagger,i}$). With the solution obtained in the interval $[t_{i-1},t_i]$, we compute the solution in the interval $[t_i,t_{i+1}]$.

In this way, the initial conditions for n_m^i at time t_i must be defined to solve the problem (20) in the interval $[t_i,t_{i+1}]$. For that

purpose, we reconstruct the vector

$$\Phi(t_i) = \sum_{m=1}^{q} n_m^{i-1}(t_i)\varphi_m^{i-1}, \tag{22}$$

from the variables $n_m^{i-1}(t_i)$, φ_m^{i-1} obtaining from the integrating in the interval $[t_{i-1}, t_i]$. As the function $\Phi(t)$must be continuous on all its domain, one could use the expansion

$$\Phi(t_i) = \sum_{m=1}^{q} n_m^{i}(t_i)\varphi_m^{i},$$

and obtain the value of $n_m^i(t_i)$ as

$$n_m^{i,\lambda}(t_i) = \frac{\langle \varphi_m^{\dagger,i}, F^i\Phi(t_i)\rangle}{\langle \varphi_m^{\dagger,i}, F^i\varphi_m^i\rangle},$$

where $\Phi(t_i)$ is computed from Equation (22).

In order to compute the initial conditions related to the concentration of the precursor k at time t_i, $c_{l,k}^i(t_i) = \langle \varphi_l^{\dagger,i}, XC_k\rangle(t_i)$, we use the known $c_{m,k}^{i-1}(t_i)$ computed in the previous integration on $[t_{i-1}, t_i]$. We assume that

$$\varphi_l^{\dagger,i} = \sum_{m=1}^{q} a_{lm}\varphi_m^{\dagger,i-1}.$$

Using the adjoint modes and the biorthogonality relation (7) in Equation (5.1) it is obtained that,

$$a_{lm} = \frac{\langle \varphi_l^{\dagger,i}, F^{i-1}\varphi_m^{i-1}\rangle}{\langle \varphi_m^{\dagger,i-1}, F^{i-1}\varphi_m^{i-1}\rangle}. \tag{23}$$

Thus, the concentration of precursors at time t_i can be computed as

$$c_{l,k}^{i,\lambda}(t_i) = \langle \varphi_l^{\dagger,i}, XC_k\rangle(t_i) = \sum_{m=1}^{q} a_{lm}\langle \varphi_m^{\dagger,i-1}, XC_k\rangle(t_i) = \sum_{m=1}^{q} a_{lm}c_{m,k}^{i-1,\lambda}(t_i). \tag{24}$$

6. Adaptive time-step control for the modes update

In previous works, the updating of the modes was made with a fix time-step that is selected by the user. This updating with a fix time-step implies several limitations. First, one needs to select a time-step previously that leads to obtain results with unpredictable errors. If, a small time-step is used to ensure good approximations, this small time-step may be not necessary in some stages of the transient. Moreover, the computational cost also increases, since most of the time in the modal computation is spent in the solutions of the modal problems. On the other hand, if we use large time-steps the computational cost is smaller but the obtained results could be inaccurate. For these reasons, there is a big interest to implement an adaptive control of the time-step for the modal methods. For doing that, there are two fundamental issues that must be studied: The error estimation due to the finite modal expansion assumption, and a suitable constraint to select the time-step based on the error estimation. Several approaches are considered in the following subsections.

6.1. Estimates of the local error

After applying the modal methodology, the obtained error essentially comes from the assumption that the neutronic flux can be described as a finite linear combination of the spatial modes, since they do not form a complete basis of the function space [8]. It is reasonable to assume that larger variations in the flux will imply larger errors in the modal method.

The first error estimate proposed is based on the difference between the different sets of eigenfunctions. This approach will be called the *modal difference error*. One can compute the modes in the next time to predict how the total flux will change. According to the difference between the previous modes and the new modes, an error can be defined as,

$$\varepsilon_{md}^i = \max_{m=1,\dots,q} \frac{\left| |\varphi_{i-1,m}^{\delta} - \varphi_{i,m}^{\delta}| \right|_1}{\left| |\varphi_{i-1,m}^{\delta}| \right|_1} k_{md},$$

where q is the total number of modes and k_{md} a constant to adjust the accuracy of the approximation and its value will depend on the transient analyzed. This error estimation requires the computation of eigenvalue problems and this can be computationally very expensive.

The second proposed approach is based on the residual error that appears when the actual modes are substituted on the problem corresponding to the next time. Larger residual errors are obtained when the eigenfunctions change more spatially. For that, we define the *modal residual error* as

$$\varepsilon_{mr} = \max_{m=1,\dots,q} \frac{\left| |A^{\delta,i}\varphi_{i-1,m}^{\delta} - \delta_m^{i-1}B^{\delta,i}\varphi_{i-1,m}^{\delta}| \right|_1}{\left| |\varphi_{i-1,m}^{\delta}| \right|_1} k_{mr},$$

where k_{mr} is a constant for this type of error and q the total number of modes.

Finally, we assume that the flux along the time will change depending on the variation in the cross-sections. For this reason, we define the *cross-section perturbation error* as

$$\varepsilon_{xs} = \sum_c \frac{\left\| \mathrm{XS}^{i-1}(c) - \mathrm{XS}^{i}(c)_1 \right\|}{\left\| \mathrm{XS}^{i-1}(c)_1 \right\|} k_{xs},$$

where k_{xs} is an accuracy constant for this error, the value of c denotes the cell c of the reactor and XS, one cross-section type that depends on the perturbation applied to the transient. This estimation is the cheapest estimation and it is used in other neutronic codes, such as PARCS [28]. Note that the same *cross-section perturbation error* is obtained when a cross-section is increased or decreased with the same value, but the response in the relative power is not the same. The type of cross-section to be considered will depend on the transient analyzed.

6.2. Control algorithm for modal update time steps

Once the error estimation is selected, it is necessary to define the control algorithm to compute the time-step from the error estimation. For that, we have studied two strategies.

The first one depends on the error in the previous step in a fixed way. It is called as *banded* time-step control. If the error is greater than some value max_{le}, the time-step is divided by 2. On the other hand, if the error is lower than other value min_{le}, the next time-step is multiplied by 2. For errors between min_{le} and max_{le}, the time-

step remains constant. This can be written as,

$$\Delta t_i = \begin{cases} \Delta t_{i-1}*2, & \varepsilon < min_{le}, \\ \Delta t_{i-1}, & min_{le} < \varepsilon < max_{le}, \\ \Delta t_{i-1} \,/\, 2, & max_{le} < \varepsilon, \end{cases} \tag{25}$$

where ε is one of the error estimations presented in the previous Section. The values of min_{le} and max_{le} are dependent on the transient analyzed. In the numerical results presented below, the value of min_{le} has been fixed to 1.0 and the value of $max_{le} = 2.0$.

The second option to select the time-step is based on the control algorithms defined for other differential methods implemented for stiff problems [13]. It is called as *dynamic* time-step control. In particular, we obtain the step Δt_i as

$$\Delta t_i = \Delta t_{i-1} \min\left\{2.0, \max\left\{0.5, \sqrt{1.0/\varepsilon}\right\}\right\}, \tag{26}$$

where ε is some error defined in Section 6.1.

Finally, to avoid using very high or very small time-steps, we have implemented a minimum time-step and maximum time-step. These values are computed as $\Delta t_{\min} = \Delta t_0/2$, $\Delta t_{\max} = \Delta t_0 * 50$, where $\Delta t_0$is the initial Δtgiven by the user.

7. Rod cusping correction

Some transient calculations in reactor cores are based on dynamic changes in the reactor configuration due to the movement of control rods, which are usual maneuvers in the reactor operation. When a control rod is partially inserted into a cell, a new material is needed to be defined for this cell or moving meshes are used, where the cell is divided into two subcells: one rodded cell and one unrodded cell [24].

The classical method to define the materials of partially inserted cells, $\overline{\Sigma}$, is based on the *volume weighting average* of the cross sections of the rodded (Σ_R) and unrodded (Σ_{NR}) part with the volume fraction inserted as

$$\overline{\Sigma} = f_{\text{ins}}\Sigma_R + (1 - f_{\text{ins}})\Sigma_{NR}, \tag{27}$$

where f_{ins} is the volume fraction of insertion of the rod in the cell. However, for some transients, this type of approximation presents what is known as the *rod cusping* effect [29]. This is a non-physical behaviour of different magnitudes such as the neutron power along the transient.

Different strategies can be applied to reduce the rod cusping problem. One possibility is using a high number of planes in the z-axis by increasing also the problem size. Another possibility is to use a simple correction model by utilizing the surrounding averaged cell values [30]. This strategy is known as *flux weighting method*. The cross-sections of the partially inserted cell, $\overline{\Sigma}$, are given by

$$\overline{\Sigma} = \frac{(1 - f_{\text{ins}})\Sigma_{NR}\varphi_{NR} + f_{\text{ins}}\Sigma_R\varphi_R}{(1 - f_{\text{ins}})\varphi_{NR} + f_{\text{ins}}\varphi_R}, \tag{28}$$

where the average flux in the unrodded and rodded fractions of the cells are approximated by

$$\varphi_{NR} = \frac{h_{c-1}\varphi_{c-1} + (1 - f_{\text{ins}})h_c\varphi_c}{h_{c-1} + (1 - f_{\text{ins}})h_c}, \tag{29}$$

$$\varphi_R = \frac{h_{c+1}\varphi_{c+1} + f_{\text{ins}}h_c\varphi_c}{h_{c+1} + f_{\text{ins}}h_c}, \tag{30}$$

where φ_c is the average neutron flux in the axial cell c and h_c is the length of the cell in the z-direction. More strategies to avoid the rod cusping could be applied [31,32].

8. Numerical results

In this Section, we analyze the performance of the updated modal methodology and the adaptive time-step control for two three-dimensional benchmarks with rectangular and hexagonal geometries. First, two different transients are studied for the Langenbuch reactor [33]: a control rods movement and an out-of-phase periodic perturbation. Also, we present some numerical results for an asymmetric control rod ejection accident in a VVER-440 core [34].

In the finite element method, the spatial discretizations is made by using Lagrange polynomials of degree 3. It has been shown in other works that this degree is enough to obtain accurate results for usual reactor calculations [1].

The solution of the eigenvalue problems (Equations (5) and (6)) has been computed with a hybrid method using a residual error for the generalized eigenvalue problem of 10^{-7} (see more details in Ref. [35]). For the direct modes computation, the initial guess used is the solution of the λ-modes problem with Lagrange polynomials degree equal to 1 in the FEM, which is previously computed. For the adjoint modes computation, the solution of the direct modes is used to initialize the solver.

In the following, the relative errors for the neutron power are described.

The neutron power distribution, P, is defined as

$$P(\vec{r}, t) = \Sigma_{f1}\Phi_1(\vec{r}, t) + \Sigma_{f2}\Phi_2(\vec{r}, t),$$

where $\Phi_1$and $\Phi_2$are the neutron fluxes related to the fast and thermal energy group, respectively.

The Local Error (LE) in % at time t is given by

$$LE(t) = \frac{\left\| |P(t) - P^{\text{ref}}(t)| \right\|_1}{\left\| P^{\text{ref}}(t)_1 \right\|} \cdot 100,$$

where $P^{\text{ref}}(t)$is the reference power at time t, and $||\ ||_1$ is the norm-1 of a vector, that is, its highest component in absolute value.

The Mean Power Error (MPE) in % in the interval $[t_0, t_N]$ is defined by

$$MPE = \frac{1}{(t_N - t_0)} \sum_{n=1}^{N} LE(t_n)(t_n - t_{n-1}).$$

The code has been implemented in C++ based on the data structures provided by the library Deal.II [25] and PETSc [36]. The computer used for the computations has an Intel® Core™ i7-4790 3.60 GHz with 32 Gb of RAM running on Ubuntu GNU/Linux 16.04 LTS.

8.1. Langenbuch-CRM transient

The Langenbuch reactor was defined in Ref. [33]. It is a small LWR core with 77 fuel assemblies and two types of fuel. Fig. 1(a) shows the geometry of the reactor model. Table 1 shows the macroscopic cross-sections that define the transient.

Table 2 displays the neutron precursors data. The neutron velocities for the fast and thermal groups are, respectively, $v_1 = 1.25 \cdot 10^7$ cm/s and $v_2 = 2.5 \cdot 10^5$cm/s. The value of ν is assumed constant and equal to 2.5 for every material and energy group. The boundary conditions are set to zero flux. The calculations for the reactor have been performed with a whole core model, using 1170 cells.

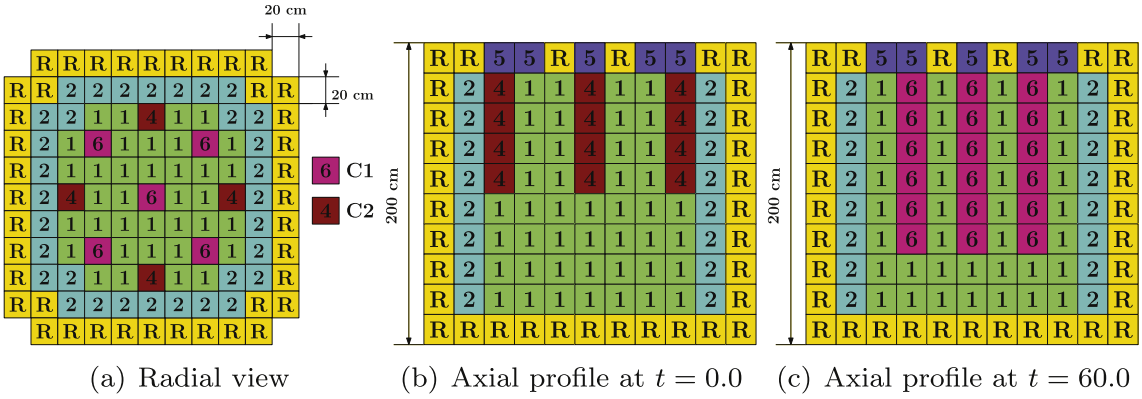


Fig. 1. Distribution of the materials along the Langenbuch-CRM transient.

Table 1
Cross sections data of the Langenbuch reactor.

Material	Group	$\mathbf{D_g}$ (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{s12} (cm^{-1})
1 − Fuel	1	1.423913	0.01040206	0.00647769	0.01755550
	2	0.356306	0.08766217	0.11273280	
2 − Fuel	1	1.425611	0.01099263	0.00750328	0.13780040
	2	0.350574	0.09925634	0.01717768	
R −Reflector	1	1.634227	0.00266057	0.00000000	0.02759693
	2	0.264002	0.049363510	0.00000000	
4,6 − Absorbent	1	1.423913	0.01095206	0.00647769	0.11273228
	2	0.356306	0.09146217	0.01755550	
5 − Reflector + Absorbent	1	1.634227	0.00321050	0.00000000	0.02759693
	2	0.264002	0.05316351	0.00000000	

Langenbuch-CRM corresponds to an operational transient of the Langenbuch reactor. Materials 4 and 6 represent control rods. It has been initiated by the withdrawal of a bank of four partially inserted control rods (C1 in Fig. 1(a)) at a rate of 3 cm/s over $0 < t < 26.7$ s. A second bank of control rods (C2 in Fig. 1(a)) is inserted at the same rate over $7.5 < t < 47.5$ s. The transient is followed during 60 s. In Fig. 1 the axial profiles at initial state, $t = 0.0$ s and at $t = 60$s are represented.

The numerical solution of this problem has been obtained by many authors but no reference solution is provided. Fig. 2 shows the relative power evolution along the transient obtained with the PARCS code [28] and the Backward differential method (BKM) described in Section 4. In the code PARCS Equation (3) is solved using a nodal strategy with polynomials of degree 2 and two refinements over the original mesh. For the BKM, the time-step has been set to $\Delta t = 0.01$ s and for the PARCS code, the maximum time step has been set to $\Delta t = 0.01$ s. Moreover, different methodologies to compute the materials in the cells when the control rod is partially inserted are compared. For the BKM, the *volume weighting method* (VWM) and the *flux weighting method* (FWM) are used. In the PARCS solutions, the volume weighting method (VWM) and the *DECUSP* method (that is a *flux weighting method* type) are used [28]. Fig. 2 does not show significative differences between the two codes when the volume weighting method is used. The mean

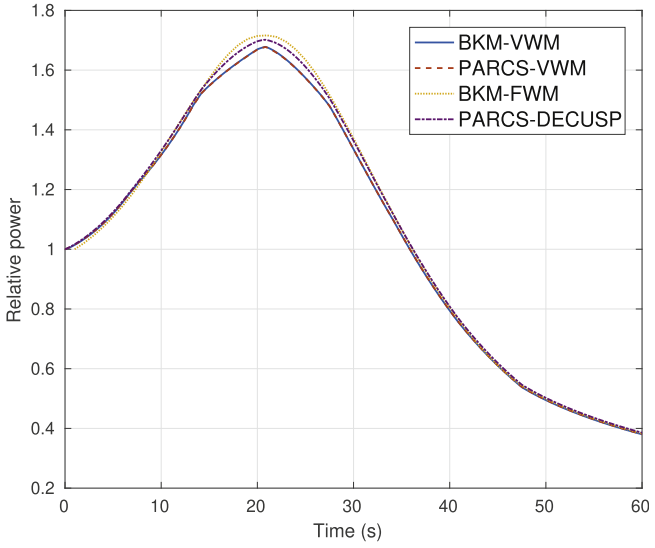


Fig. 2. Relative power of the Langenbuch-CRM transient computed with the BKM and PARCS code.

power error between the two solutions is $MPE = 4{\cdot}10^{-3}$. However, in the two solutions obtained with FWM and *DECUSP* methods a small difference between BKM and PARCS solutions is appreciated. In this case, the error between these two solutions is $MPE = 1.8{\cdot}10^{-2}$. For these last approximations, the rod cusping effect is reduced. The reference for this problem has been taken as the solution computed with the BKM and the *flux weighting method*. Fig. 3 represents the neutronic power distribution in the middle plane of the reactor at several relevant times, showing the radial symmetry of the power.

First, we study the efficiency of the matrix-free implementation proposed in this work. For that purpose, we compare the same algorithms with the same settings to solve the Langenbuch-CRM transient where only the matrix allocation type and, consequently, the preconditioner used to solve the linear systems are changed. Table 3 displays the mean of the iterations needed for the

Table 2
Neutron precursors data of the Langenbuch reactor.

Group (k)	1	2	3	4	5	6
β_k	0.000247	0.0013845	0.001222	0.0026455	0.000832	0.000169
$\lambda_k^d(s^{-1})$	0.0127	0.0317	0.115	0.311	1.4	3.87

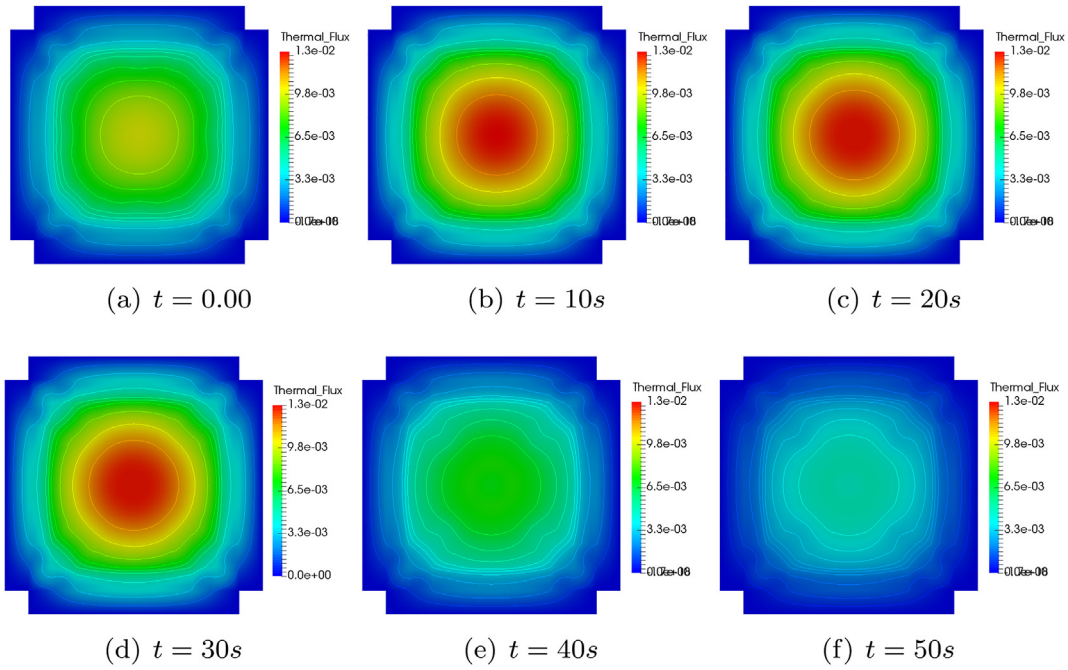


Fig. 3. Evolution power at the middle plane of the Langenbuch-CRM transient.

Table 3
Matrix-free backward differential method performance for the Langenbuch-CRM transient.

Matrix Alloc.	Mean GMRES its.	Max. memory	CPU Time
CSR	15.7	1422 MB	151 min
Matrix-free	16.6	478 MB	55 min

GMRES to solve the linear systems (Mean GMRES its.), the maximum computational memory required (Max. memory) and the computational time (CPU Time) to solve the transient by using the backward differential method (BKM). The results show that the matrix-free implementation decreases the computational memory and the CPU time considerably, even though the number of iterations needed for the convergence of the linear systems is greater than when the matrices are stored in the CSR format [37]. To analyze the matrix-free performance in the updated modal method (where $\Delta t = 5.0$s), Table 4 shows the maximum computational memory and CPU needed to apply this methodology with several settings and number of eigenvalues in the expansion, q. All cases show a high reduction in computational resources, especially in the CPU time, if the matrix-free implementation is used.

To test the performance of the updated modal method for the Langenbuch-CRM transient, data have been collected when different number of modes (q) and different fixed time-steps (Δt) are used. Table 5 displays the mean power error (MPE) and the CPU time needed to obtain each approximation. This Table shows that using only $q = 1$ λ mode is enough to approximate the solution of this transient with the modal method, because increasing the number of modes reduces the MPE slightly, but increases enormously the CPU time. Note that for this transient, the evolution of the power can be described with an adiabatic approximation. However, this characteristic of the modal methods is heavily dependent on the type of transient analyzed.

Fig. 4 represents the power evolution computed with the BKM and the updated modal method with one eigenvalue and with several fix time-steps. The errors in the neutron power between the

Table 4
Matrix-free updated modal method performance for the Langenbuch-CRM transient.

Matrix Alloc.	q	Updating	Max. memory	CPU Time
CSR	1	No	1244 Mb	176 min
Matrix-free	1	No	345 Mb	2 min
CSR	1	Yes	1253 Mb	113 min
Matrix-free	1	Yes	444 Mb	3 min
CSR	3	No	1126 Mb	319 min
Matrix-free	3	No	350 Mb	7 min
CSR	3	Yes	1394 Mb	225 min
Matrix-free	3	Yes	456 Mb	16 min

Table 5
Updated modal method performance for Langenbuch-CRM transient.

N. eigs. (q)	Δt	MPE (%)	CPU Time
1	1.0 s	0.278	11 min
1	2.0 s	0.281	6 min
1	5.0 s	0.630	3 min
1	10.0 s	1.880	2 min
3	1.0 s	0.260	31 min
3	2.0 s	0.261	19 min
3	5.0 s	0.509	16 min
3	10.0 s	1.650	4 min

BKM and the updated modal method are mainly produced at the maximum value of power. These differences are reduced as the time-step for updating the modes decreases. This fact can be also observed in the evolution of the local errors (LE) shown in Fig. 5. The local errors in the power increase along the time until $t \approx 20$ s then, they decrease and from $t \approx 30$ s they begin to increase again. It happens because of the definition of the MPE, since from $t = 30$ s the curve of the modal approximation is above the BKM solution It is also observed that the errors decrease just at times when the dominant λ-mode is updated.

To analyze the adaptive time-step control for the updated modal methodology only one mode is used in the modal expansion. The

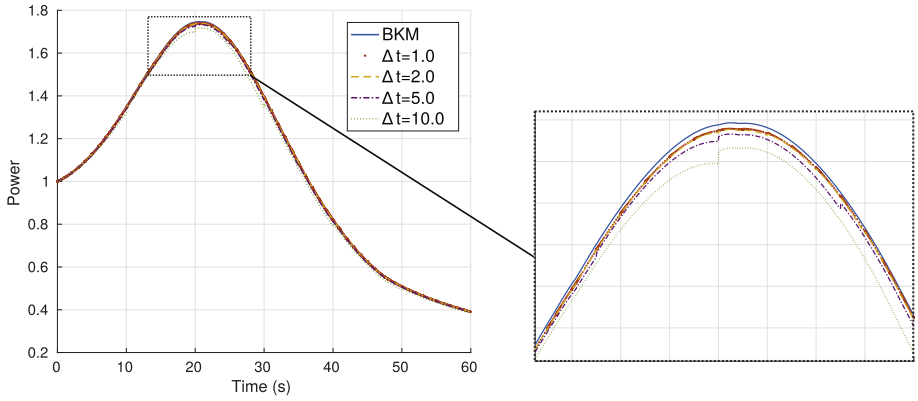


Fig. 4. Evolution of the relative power computed with the updated modal method with fix time-steps for the Langenbuch-CRM transient.

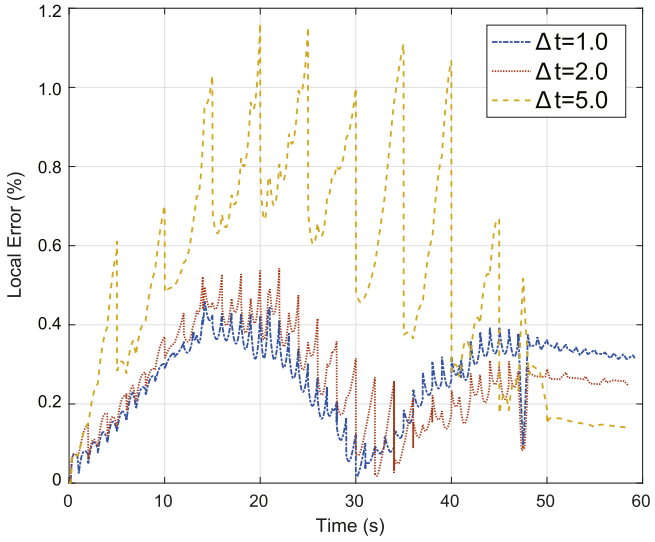


Fig. 5. Evolution of local error (%) in the Langenbuch-CRM transient in the updated modal method with one mode.

Table 6
Errors and CPU time obtained with the adaptive time-step modal method for the Langenbuch-CRM transient.

Type of Error	Banded Control		Dynamic Control	
K	**MPE (%)**	**CPU Time**	**MPE (%)**	**CPU Time**
ε_{md}				
0.5	3.050	2.5 min	1.181	5.3 min
1.0	1.830	4.0 min	0.396	8.6 min
2.0	0.464	12.5 min	0.887	8.4 min
ε_{mr}				
200	0.615	3.6 min	0.407	4.0 min
300	0.432	4.0 min	0.310	5.5 min
500	0.312	6.0 min	0.276	8.2 min
ε_{xs}				
0.5	2.651	7.2 min	1.793	9.2 min
1.0	1.572	9.4 min	1.041	11.0 min
2.0	0.965	12.0 min	0.503	16.0 min

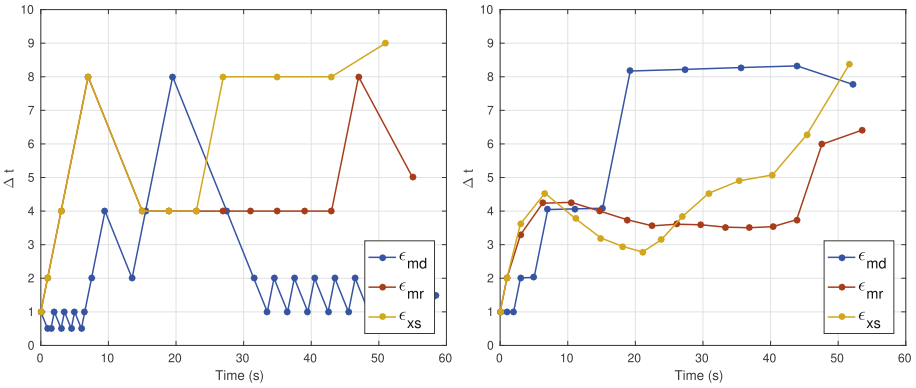
initial time-step for all strategies has been set to $\Delta t_0 = 1.0$s. For the *cross-section perturbation error*, the absorption cross-section has been used to quantify the error because the movement of control rods in this benchmark only produces changes in this cross-section. Table 6 shows the Mean Power Errors (MPE) and CPU times obtained by using the different error estimations, time-step controls and accuracy coefficients (k_{md}, k_{mr}, k_{xs}). First, one can observe that using higher values for the accuracy coefficients gives more accurate results because the time-steps are reduced, but it also needs more time for the computation. Higher values for the *modal difference error* are needed for adapting the time-steps. In the comparison of the different error estimators, one can deduce that the *modal difference error* (ε_{md}) is not very efficient. Between the other error estimations, the *modal difference error* (ε_{mr}) gives better results using less CPU time.

Table 6 does not show relevant differences between the dynamic control and the banded control. The dynamic control selects smaller time-steps and the approximations obtained are, consequently, better, but also the CPU time is increased. To better analyze both time-step controls, Fig. 6 shows the time-steps and the local errors obtained with the different types of error estimators. The accuracy coefficients are selected as $k_{md} = 2.0$, $k_{mr} = 200$ and $k_{xs} = 2.0$, because similar MPE are obtained with these settings (Table 6). First, if the time-steps are analyzed (in the first row of Fig. 6), it is deduced that the time-steps obtained with the *modal residual error* depend on the type of error control. From $t = 20$ s, the banded error reduces the time-step while the dynamic error increases it. The time-steps computed with *the modal difference error*, with banded and dynamic control error, follows a behaviour similar to the local error. Finally, the *cross-section perturbation error*, for both types of control error, reduces the time-steps only when the perturbation in the cross-section is higher. The evolution of the time-steps is reflected in the evolution in the local errors (second row of Fig. 6).

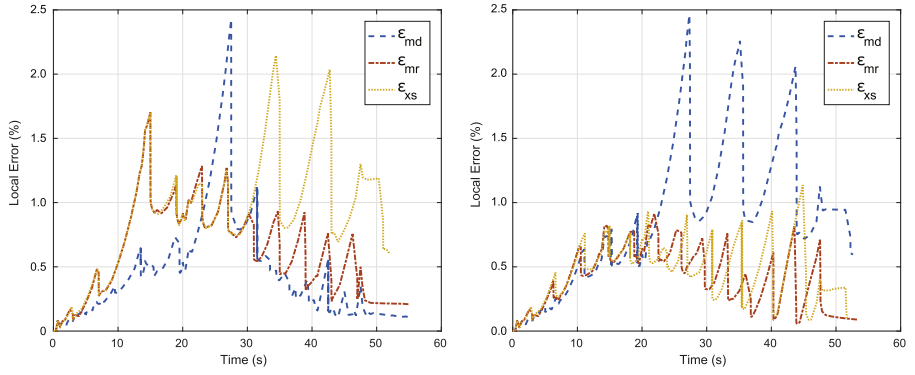
If we compare updated modal method using an adaptive time-step for the modes updating with the updated modal method using a fix time-step, it can be observed that for similar mean power errors (MPE), more distributed local errors are obtained. It can be appreciated, for instance, to obtain a MPE$\approx$0.6% first, with the adaptive updated modal method with ε_{mr}, dynamic control and $k_{mr} = 200$ (Fig. 6) and second, with the updated modal method with fixed $\Delta t = 5.0$ s (Fig. 5). All modal strategies improve considerably the CPU time required by the BKM method to approximate the evolution of the power.

8.2. Langenbuch-OPP transient

The Langenbuch-OPP transient has been defined by perturbing the fission cross sections of material 1 in the Langenbuch reactor represented in Fig. 7 with a striped pattern. In this case, we have defined two types of local sinusoidal perturbations that are out of phase between them. They are expressed as,

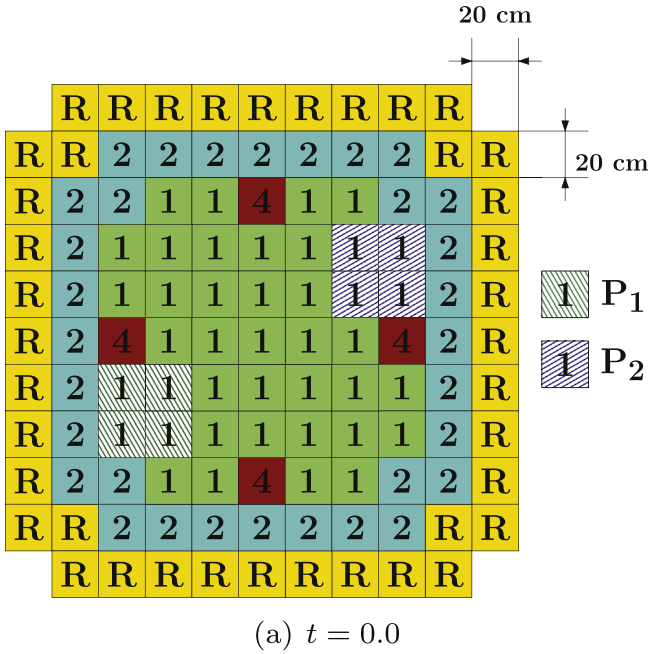


(a) Banded Δt (b) Dynamic Δt



(c) Banded Local Error (d) Dynamic Local Error

Fig. 6. Evolution of the time-step (Δt) and local error (%) in the adaptive time-step control for the Langenbuch-CRM transient.



(a) $t = 0.0$

Fig. 7. Location of the perturbation areas for the Langenbuch-OPP transient.

$$\Sigma_{f,g}(t) = \Sigma_{f_0,g}(t) + \delta\Sigma_{f,g}(t) \qquad g = 1,2. \tag{31}$$

The perturbation 1, represented as P1 in Fig. 7, is given by

$$\delta\Sigma_{f,g}(t) = 5 \cdot 10^{-4}\sin(2\pi ft) \qquad g = 1,2, \tag{32}$$

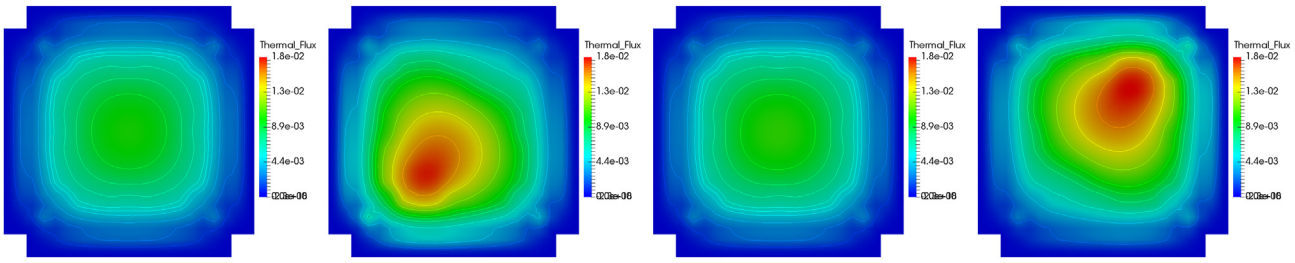
and the perturbation 2, denoted by P2, is given by

$$\delta\Sigma_{f,g}(t) = 5 \cdot 10^{-4}\sin(2\pi ft + \pi) \qquad g = 1,2, \tag{33}$$

where $f = 1.0$ Hz. This transient duration is 2 s.

Before starting with the modal method, we have computed the power evolution using the Backward Difference method (BKM), which is taken as the reference solution. The time-step for the BKM has been set to 0.001 s. Fig. 8 represents the radial average power distribution at four relevant times ($t = 0.00$ s, $t = 0.25$ s, $t = 0.50$ s and $t = 0.75$ s). At the beginning, the maximum power goes from the center to the perturbation 1 zone then, it comes back to the center and then, it goes to the perturbation 2 zone. This behavior is repeated along the transient.

To study the performance of the modal method, the mean power errors (MPE) between the backward differential method (BKM) with $\Delta t = 0.001$ s and some settings of the updated modal method with fixed time-steps are computed. The number of modes (q) used in the expansion has been $q = 3$ and $q = 5$. First, Table 7 shows that small time-steps for the modes updating gives more accurate approximations. In contrast, the CPU times are also higher. For $q = 3$, good approximations are obtained with updating time-steps smaller or equal to 0.2 s. Table 7 also shows that increasing the number of modes is not an efficient strategy because improves very little the obtained errors with a high CPU time cost. In this case, the increasing of two extra modes in the expansion does not improve considerably the spatial representation in the evolution of the solution. This characteristic depends on the transient analyzed and



(a) $t = 0.00$ (b) $t = 0.25$ (c) $t = 0.50$ (d) $t = 0.75$

Fig. 8. Evolution power of the out-of-phase perturbation in the Langenbuch-OPP transient.

Table 7
Data of the updated modal method with fixed time-steps for the Langenbuch-OPP transient.

N. eigs. (q)	Δt	MPE (%)	CPU Time
3	0.40 s	1.793	4 min
3	0.20 s	0.988	7 min
3	0.10 s	0.558	12 min
3	0.05 s	0.187	30 min
5	0.40 s	0.953	7 min
5	0.20 s	0.375	15 min
5	0.10 s	0.167	24 min
5	0.05 s	0.077	42 min

usually, the next eigenfunctions do not provide more relevant information.

Fig. 9 represents the evolution of the global power computed with the BKM and the updated modal method with several fixed time-steps (Δt) and 3 modes. It is observed that large errors between the BKM and the updated modal method are produced when the perturbations reach their maximums. However, these differences are reduced for small time-steps. Fig. 9(b) displays the local error (LE) along the time for several time-steps. It shows non-uniform errors along the time and high errors near to the extremes of the power (maximums and minimums) and before to the modal updating. Note that, in this type of transient, the errors are larger than the ones obtained for the Langenbuch-CRM transient and the time-steps used are quite smaller. Thus, the time-step needed to update the modes is very dependent on the behaviour of the transient.

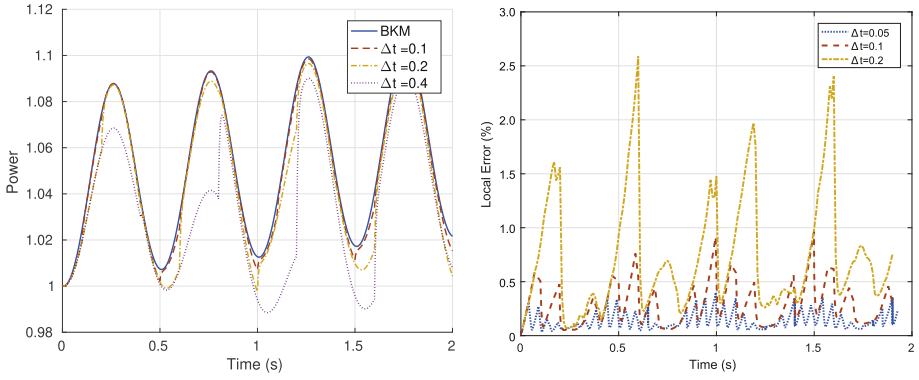
Now, the updated modal method with an adaptive time step is used for the Langenbuch-OPP transient. The number of modes for the modal method has been set to $q = 3$. The initial time-step for all strategies has been set to $\Delta t_0 = 0.05$s. The cross-section used for the *cross-section perturbation error* has been the fission cross-section.

Table 8
Mean Power Errors (MPE) and CPU time obtained with the adaptive time-step modal method for the Langenbuch-OPP transient.

Type of Error	Banded Control		Dynamic Control	
k	**MPE (%)**	**CPU Time**	**MPE (%)**	**CPU Time**
ε_{md}				
1.0	3.901	3 min	3.748	5 min
2.0	2.902	22 min	1.681	14 min
5.0	1.641	32 min	1.519	20 min
ε_{mr}				
50	6.383	3 min	2.117	3 min
100	1.011	7 min	1.162	8 min
200	1.531	7 min	0.713	18 min
ε_{xs}				
0.5	1.338	4 min	0.809	7 min
1.0	0.647	11 min	0.656	12 min
2.0	0.617	13 min	0.607	13 min

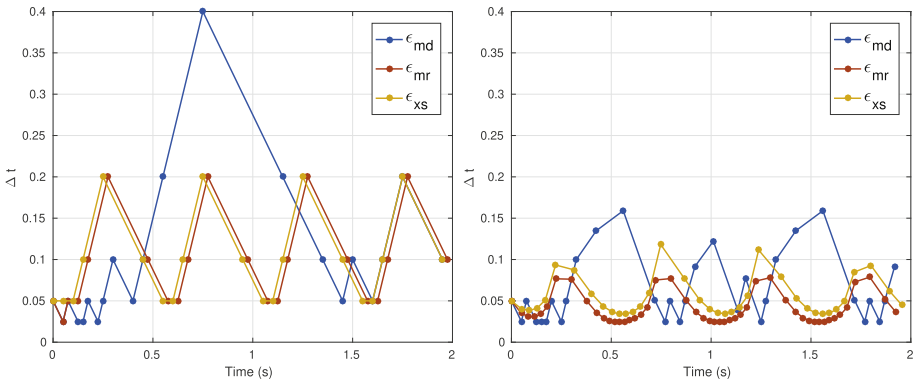
Table 8 shows the Mean Power Errors and CPU times obtained by setting the different error estimations, control errors and accuracy coefficients. As in the previous transient, the *modal difference error* (ε_{md}) is not very efficient because it needs to compute the modes to estimate the error and this is very expensive. Regarding the other error estimations, the *cross-section perturbation error* (ε_{mr}) gives lower errors, but using more time than the *modal residual error*. If the type of error control is compared, in general, the dynamic control gives better approximations than the banded control, but also using more CPU time.

Fig. 10 shows the time-steps obtained with $k_{md} = 2.0$, $k_{mr} = 200$ and $k_{xs} = 0.5$ with the dynamic and the banded control. A similar



(a) Relative power (b) Local Error

Fig. 9. Relative power and local error (%) obtained with the updated modal method with 3 modes for the Langenbuch-OPP transient.



(a) Banded Error Control (b) Dynamic Error Control

Fig. 10. Evolution of the time-step (Δt) with the adaptive time-step control for the Langenbuch-OPP transient.

behaviour in the computation of the time-steps for the *modal residual error* and the *cross-section perturbation error* can be observed, but for the *modal residual error*, the time-steps computed are slightly smaller. In contrast with the previous transient, here the *cross-section perturbation error* is adapted according to the local error.

Fig. 11 displays the local errors for the different accuracy coefficients in the *modal residual error* with dynamic control error. It is showed that higher values in the coefficient decrease the local errors. However, these errors are not as distributed along the transient as the local errors obtained with higher values in the coefficients.

If we compare the local error for the updated modal method with fixed $\Delta t = 0.1$ s and for the adaptive modal method with the *cross-section perturbation residual error*, *dynamic* control time-step and $k_{xs} = 0.5$, both strategies obtain similar mean power errors (MPE $\approx$1%) but with the adaptive control time-step the results are obtained in less time.

Finally, the modal strategies are compared with the BKM. The CPU time to approximate the solution with the BKM and $\Delta t = 0.001$ s is 195 min. All configurations for the modal expansions are more competitive than the BKM in terms of CPU time.

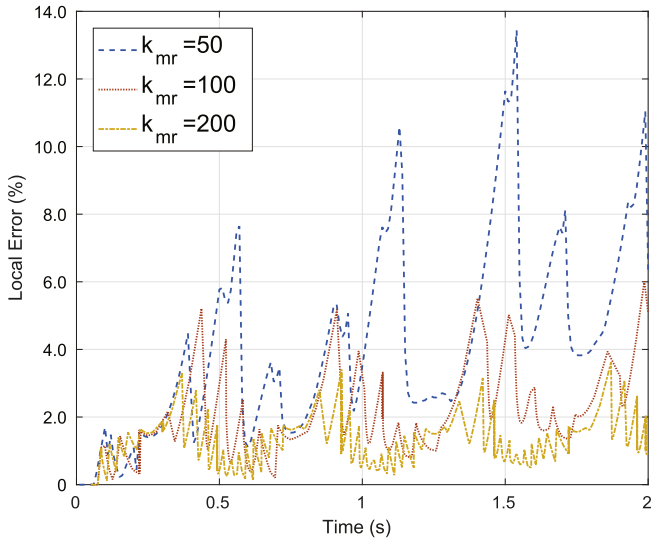


Fig. 11. Comparison of the evolution of the local error (%) with the fixed updated modal method and the adaptive updated modal method for the Langenbuch-OPP transient.

8.3. The AER-DYN-001 problem

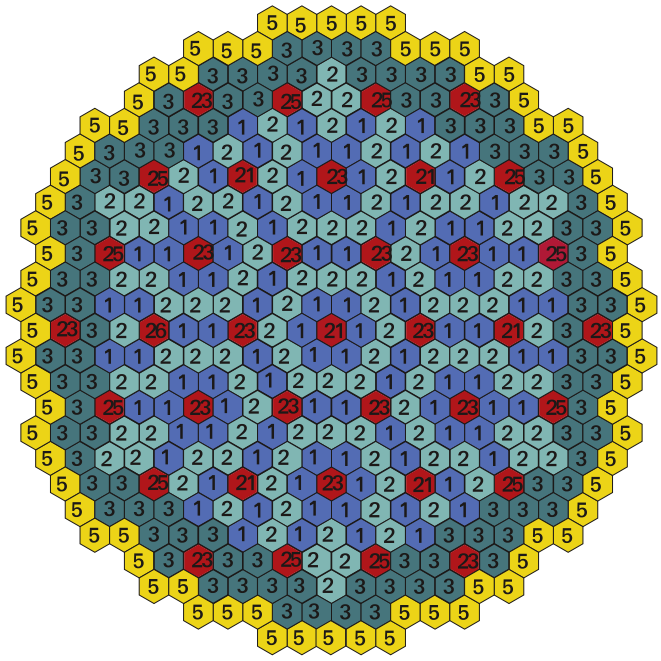
The AER-DYN-001 problem was introduced in Refs. [34] and corresponds to an asymmetric control rod ejection accident without any feedback in a VVER440 reactor. The core has 250 cm height, with two reflector layers of 25 cm each added, one to the top and the other one to the bottom of the core. The assembly pitch is 14.7 cm. The core is a VVER-440 core type, with 25 fuel elements across the diameter. The disposition of materials together with the initial position of the control rods are shown in Fig. 12. Albedo boundary condition are applied on the outer edge of the reflector nodes. The extrapolation length is 2.13*D_g in both groups, where D_g is the diffusion coefficient given for the material 5. Table 9 collects the cross-section data. The values of ν are $\nu_1 = 2.55$, $\nu_2 = 2.43$and they are constant for all materials. The neutron precursors data are the same as the ones for the Langenbuch reactor (Table 2). The neutron velocities are $\nu_1 = 1.27 \cdot 10^7$cm/s and $\nu_2 = 2.5 \cdot 10^5$ cm/s. For the spatial discretization, the deal.II library cannot handle hexagonal cells, thus each hexagon is subdivided into 3 quadrilaterals to obtain a total number of 15156 cells.

The transient is defined as follows. The control rod denoted by number 26 is ejected in the first 0.08 s with velocity 25 m/s. Then, scram is initiated inserting the safety rods 23 and 25 at $t = 1.0$s with velocity 0.25 m/s, so that the bottom position is reached at $t = 11.0$ s. The drop of control rod group 21 is also started at 1.0 s with the same velocity.

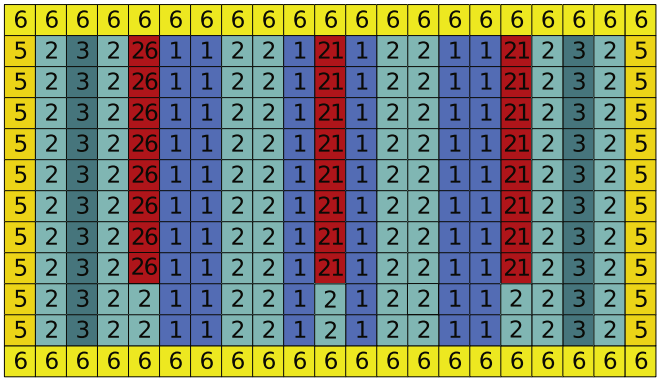
Before analyzing the modal method, we study the strategy to avoid the rod cusping, which is very large in this transient. To integrate the time-dependent equation the backward differential method (BKM) with $\Delta t = 0.01$ is used. Fig. 13 displays the power obtained with different strategies of computing the cross-sections for cells where the control rod is partially inserted: the volume weighting method (VWH) with 12 planes in the z-axis, the flux weighting method (FWM) with 12 planes and the flux weighting method with 24 planes.

First, it is observed that the ejection of the control assembly *26* increases the reactor power till 1.0 s. After this time the rest of the control rods begin to fall into the core causing a reduction in the power beyond 1.5 s. The rod cusping effect begins to show from 1.0 s. The graphic exhibits that the flux weighting method reduces the rod cusping problem by using only 24 planes. In the following, this configuration is used to obtain the approximations with the BKM and the modal method.

The neutronic power evolution for this transient has been computed using several codes and it has not only one reference solution. Fig. 14 compares the solution obtained with the BKM



(a) Plant



(b) Profile

Fig. 12. VVER-440 geometry and materials distribution.

Table 9
Cross section definition for the VVER 440.

Mat.	Group	$\mathbf{D_g}$ (cm)	Σ_{ag}(cm^{-1})	$\nu_g\Sigma_{fg}$(cm^{-1})	Σ_{12}(cm^{-1})
1 – Fuel	1	1.3466e+0	8.3620e-3	4.4339e-3	1.6893e-2
	2	3.7169e-1	6.4277e-2	7.3503e-2	
2 – Fuel	1	1.3377e+0	8.7970e-3	5.5150e-3	1.5912e-2
	2	3.6918e-1	7.9361e-2	1.0545e-1	
3 – Fuel	1	1.3322e+0	9.4700e-3	7.0120e-3	1.4888e-2
	2	3.6502e-1	1.0010e-1	1.4908e-1	
4 – Absorber	1	1.1953e+0	1.3372e-2	0.0000e+0	2.2264e-2
	2	1.9313e-1	1.3498e-1	0.0000e+0	
5 – Radial Refl.	1	1.4485e+0	9.2200e-2	0.0000e+0	3.2262e-2
	2	2.5176e-1	3.2839e-2	0.0000e+0	
6 – Axial Refl.	1	1.3413e+0	2.1530e-3	0.0000e+0	2.7148e-2
	2	2.4871e-1	6.4655e-2	0.0000e+0	

(24p-FWM), the solutions available in the benchmark problem computed with the adiabatic method, the DYN3D code and KIKO3D code [27], the solution computed with the TRIKIN code [38] and the

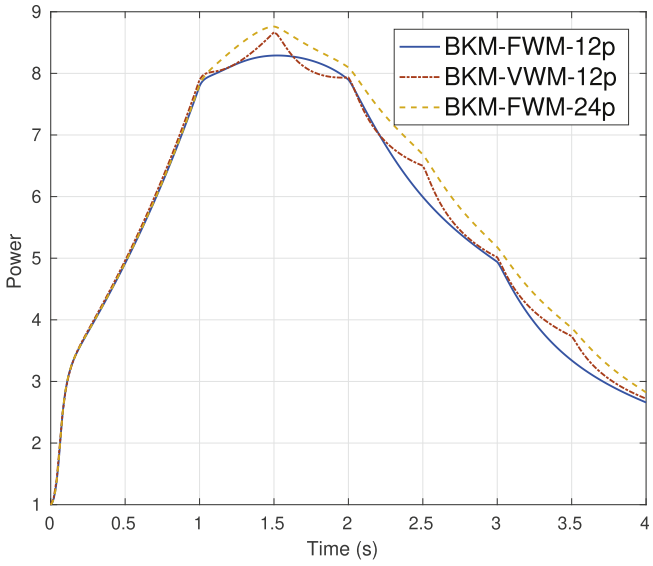


Fig. 13. Power evolution computed with different methodologies for the rod cusping effect in the AER-DYN-001 transient.

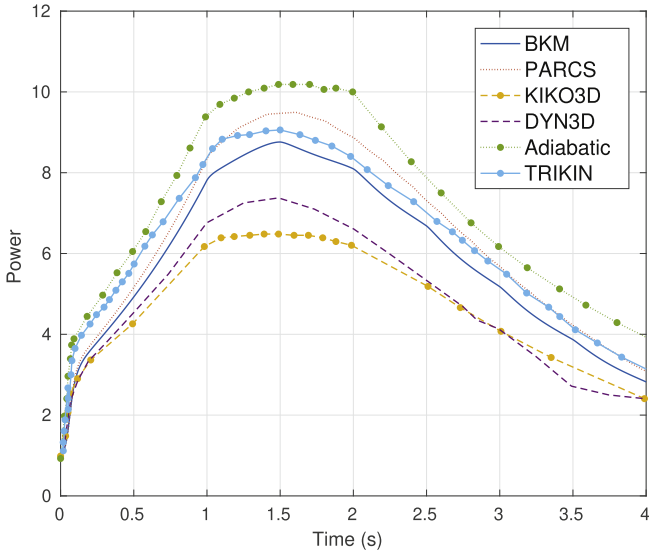


Fig. 14. Comparison of the power along the AER-DYN-001 transient with different codes.

solution obtained with the PARCS code using 60 axial planes and the flux weighting method for homogenizing the cross sections. The BKM results are very similar to the PARCS ones until 1.0 s and then, the power obtained with BKM is more similar to the TRIKIN power distribution.

First, the updated modal method with fixed time-steps is analyzed. Table 10 displays the mean power error (MPE), taking the solution computed with BKM as a reference, and the CPU time obtained with the updated modal method using 2 modes. This choice is because the solution with one eigenvalue gives non-accurate approximations.

Table 10 shows that very small time-steps are needed in the updated modal method to approximate accurately the drop out of the bar at the beginning of the transient. This fact can be also appreciated in Fig. 15. However, beyond $t = 1$s, using small time-steps implies high local errors (Fig. 15(b)). Thus, different time-

Table 10
Comparison of the Backward Differential Method (BKM), the updated modal method and the adaptive updated modal method for the AER-DYN-001 transient.

Method	Δt	MPE (%)	CPU Time
BKM	0.01 s	–	89 h
Updated modal method	0.01 s	5.90e-02	140 h
Updated modal method	0.05 s	4.60e-02	38 h
Updated modal method	0.10 s	4.86e-02	23 h
Adaptive updated modal method	–	3.59e-02	17 h

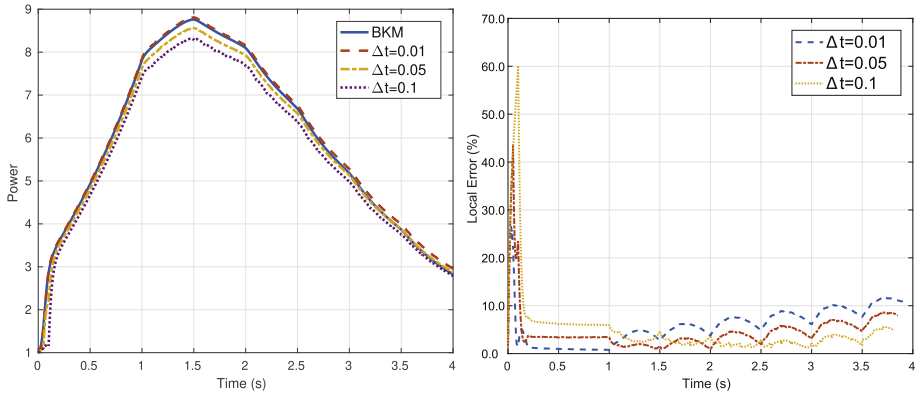
steps are needed to be used along the transient. Thus, the solution with the adaptive updated modal method with 2 eigenvalues, modal residual error, dynamic control time-step and $k_{mr} = 100$is computed. The obtained mean power error (MPE) and the CPU time are included in Table 10. With this method, a smaller mean power error than with the rest of the solutions computed with the fixed updated modal method is obtained. Furthermore, the CPU time is reduced considerably.

Fig. 16 shows the relative power, the local error and the time-step (Δt) obtained with the updated modal method with adaptive time step. The local error has been compared with the local error obtained with the updated modal method using a fix time step $\Delta t = 0.05$ s. It observed that the adaptive modal method reduces the local error in the first times, but also reduces the local error beyond $t = 1$ s.

9. Conclusions

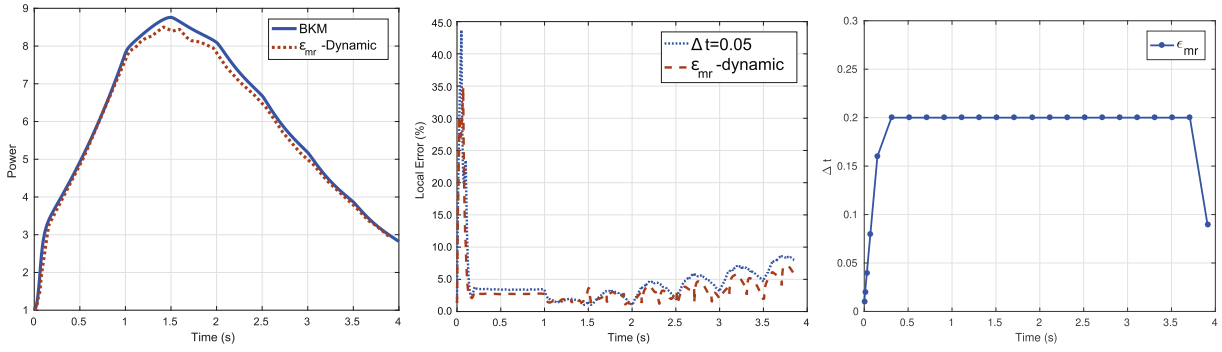
In this work, an updated modal method using an adaptive time step for the modes updating is proposed to solve the time-dependent neutron diffusion equation. For the spatial discretization of the differential equations, we have used a high order finite element method. The performance of this methodology has been analyzed and compared by solving several three-dimensional transients with both rectangular and hexagonal geometries. To validate the implemented code, the obtained results have been compared with the ones obtained with the backward differential method (BKM) applied directly to the semi-discrete neutron diffusion equation.

Several time-step controls have been proposed depending on different errors associated with the modal expansion of the neutron flux and the cross sections variation along the transient. The most efficient results are obtained with an error estimation based on the difference between the eigenfunctions (*modal residual error*). However, the *cross-section perturbation error* is also a good strategy to control the time step update. On the other hand, the *modal difference error* is computationally very expensive. With respect to the type of control, the dynamic control error is more adapted to the local errors, but there are not relevant differences between the dynamic and the banded time step control. Moreover, different coefficients are defined to adjust the accuracy obtain in the different approaches. These values are not highly dependent on the transient analyzed. For the *modal residual error*, the authors



(a) Relative power (b) Local Error (LE)

Fig. 15. Relative power and local error (%) obtained with the updated modal method with 2 modes for the AER-DYN-001 transient.



(a) Relative power (b) Local Error (c) Δt

Fig. 16. Relative power, local error (%) and time-step (Δt) obtained with the updated modal method and the adaptive updated modal method with 2 modes for the AER-DYN-001 transient.

recommend to use $k_{mr} \approx 100$. For the *modal difference error* a value of $k_{md} \approx 2.0$ gives good results. Lastly, for the *cross-section perturbation error* values of $k_{xs} \approx 2$ are a good option. However, we would like to remark that these values are obtained based on the numerical results studied in this work. Therefore, for other types of transients, using alternative values for k may be a more efficient option. The numerical results show that the updated modal method with an adaptive time step control decreases the errors in the obtained solutions with the same or smaller CPU times than the updated modal method with fixed time-step, and also obtains more distributed local errors over the simulation.

When the different methodologies to integrate the neutron diffusion equation used in this work are compared, the modal method is faster than the BKM in the transients analyzed. For this reason, future works are devoted to implement modal methods for other type of angular approximations whose integration with other type of strategies is computationally very expensive.

Declaration of competing interest

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

Acknowledgements

This work has been partially supported by Spanish Ministerio de Economía y Competitividad under projects ENE2017-89029-P and MTM2017-85669-P and financed with the help of a Primeros Proyectos de Investigación (PAID-06-18) from Vicerrectorado de Investigación, Innovación y Transferencia of the Universitat Politècnica de València.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.net.2020.07.004.

References

[1] A. Vidal-Ferrandiz, R. Fayez, D. Ginestar, G. Verdú, Solution of the lambda modes problem of a nuclear power reactor using an *h-p* finite element method, Ann. Nucl. Energy 72 (2014) 338–349.
[2] W. Stacey, Space-time Nuclear Reactor Kinetics, vol. 5, Academic Press, 1969.
[3] K. Ott, D. Meneley, Accuracy of the quasistatic treatment of spatial reactor kinetics, Nucl. Sci. Eng. 36 (3) (1969) 402–411.
[4] S. Dulla, P. Ravetto, M. Rostagno, Neutron kinetics of fluid–fuel systems by the quasi-static method, Ann. Nucl. Energy 31 (15) (2004) 1709–1733.
[5] S. Dulla, E. Mund, P. Ravetto, The quasi-static method revisited, Prog. Nucl. Energy 50 (8) (2008) 908–920.
[6] A. Henry, N. Curlee, Verification of a method for treating neutron space-time problems, Nucl. Sci. Eng. 4 (6) (1958) 727–744.
[7] J. McFadden, Solution of the Space-Time Dependent Neutron Kinetics Equations for a Reflected Slab Reactor, Ph.D. thesis, Iowa State University, 1968.
[8] A. Henry, Nuclear-reactor Analysis, vol. 4, MIT press Cambridge, Massachusetts, 1975.
[9] R. Miró, D. Ginestar, G. Verdú, D. Hennig, A nodal modal method for the neutron diffusion equation. Application to BWR instabilities analysis, Ann. Nucl. Energy 29 (10) (2002) 1171–1194.
[10] A. Carreño, A. Vidal-Ferràndiz, D. Ginestar, G. Verdú, Modal methods for the neutron diffusion equation using different spatial modes, Prog. Nucl. Energy 115 (2019) 181–193.
[11] D. Ginestar, G. Verdú, V. Vidal, R. Bru, J. Marín, J. Munoz-Cobo, High order backward discretization of the neutron diffusion equation, Ann. Nucl. Energy 25 (1–3) (1998) 47–64.
[12] K. Gustafsson, M. Lundh, G. Söderlind, Api stepsize control for the numerical solution of ordinary differential equations, BIT Numerical Mathematics 28 (2) (1988) 270–287.
[13] G. Wanner, E. Hairer, Solving Ordinary Differential Equations II, Springer Berlin Heidelberg, 1996.
[14] G. Söderlind, Automatic control and adaptive time-stepping, Numer. Algorithm. 31 (1–4) (2002) 281–310.
[15] C. Shim, Y. Jung, J. Yoon, H. Joo, Application of backward differentiation formula to spatial reactor kinetics calculation with adaptive time step control, Nucl. Eng.Technol 43 (6) (2011) 531–546.
[16] A. Avvakumov, V. Strizhov, P. Vabishchevich, A. Vasilev, Automatic time step selection for numerical solution of neutron diffusion problems, in: International Conference on Finite Difference Methods, Springer, 2018, pp. 145–152.
[17] J. Boffie, J. Pounders, An adaptive time step control scheme for the transient diffusion equation, Ann. Nucl. Energy 116 (2018) 280–289.
[18] Y. Cai, X. Peng, Q. Li, K. Wang, X. Qin, R. Guo, The numerical solution of space-dependent neutron kinetics equations in hexagonal-z geometry using backward differentiation formula with adaptive step size, Ann. Nucl. Energy 128 (2019) 203–208.
[19] D. Caron, S. Dulla, P. Ravetto, Adaptive time step selection in the quasi-static methods of nuclear reactor dynamics, Ann. Nucl. Energy 105 (2017) 266–281.
[20] A. Vidal-Ferràndiz, A. Carreño, D. Ginestar, G. Verdú, A block arnoldi method for the spn equations, Int. J. Comput. Math. (2019) 1–17.
[21] M. Kronbichler, K. Kormann, A generic interface for parallel cell-based finite element operator application, Comput. Fluid 63 (2012) 135–147.
[22] A. Seubert, K. Velkov, TORT-TD transient simulations of the C5G7-TD benchmark, in: M & C 2017, KNS, 2017, pp. 1–7.
[23] W. Stacey, Nuclear Reactor Physics, John Wiley & Sons, 2007.
[24] A. Vidal-Ferràndiz, R. Fayez, D. Ginestar, G. Verdú, Moving meshes to solve the time-dependent neutron diffusion equation in hexagonal geometry, J. Comput. Appl. Math. 291 (2016) 197–208.
[25] W. Bangerth, R. Hartmann, K. G, Deal.II – a general purpose object oriented finite element library, ACM Trans. Math Software 33 (4) (2007) 24/1–24/27.
[26] G. Verdú, D. Ginestar, V. Vidal, J. Muñoz-Cobo, A consistent multidimensional nodal method for transient calculations, Ann. Nucl. Energy 22 (6) (1995) 395–410.
[27] A. Keresztri, M. Telbisz, Dynamic Benchmark 1, 2009 [Dyn001.doc], http://aerbench.kfki.hu/aerbench/.
[28] T. Downar, D. Lee, Y. Xu, T. Kozlowski, J. Staudenmier, Parcs V2. 6 Us Nrc Core Neutronics Simulator Theory Manual, School of Nuclear Engineering Purdue University.
[29] H.S. Joo, Resolution of the Control Rod Cusping Problem for Nodal Methods, Ph.D. thesis, Department of Nuclear Engineering, MIT, Cambridge, MA, USA, 1984.
[30] J. Gehin, A Quasi-Static Polynomial Nodal Method for Nuclear Reactor Analysis, Tech. Rep., Oak Ridge Inst. for Science and Education, TN (United States); Massachusetts, 1992.
[31] A. Yamamoto, A simple and efficient control rod cusping model for three-dimensional pin-by-pin core calculations, Nucl. Technol. 145 (1) (2004) 11–17.
[32] A. Dall'Osso, Reducing rod cusping effect in nodal expansion method calculations, in: Proceedings of the International Conference on the New Frontiers of Nuclear Technology: Reactor Physics, Safety and High-Performance Computing, PHYSOR, 2002, pp. 1–13.
[33] S. Langenbuch, W. Maurer, W. Werner, Coarse-mesh flux-expansion method for the analysis of space-time effects in large light water reactor cores, Nucl. Sci. Eng. 63 (4) (1977) 437–456.
[34] A. Keresztúri, M. Telbisz, A three dimensional hexagonal kinetic benchmark problem, in: 2nd AER Symposium, Paks Hungary, 1992, pp. 1–20.
[35] A. Carreño, A. Vidal-Ferràndiz, D. Ginestar, G. Verdú, Block hybrid multilevel method to compute the dominant λ-modes of the neutron diffusion equation, Ann. Nucl. Energy 121 (2018) 513–524.
[36] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, et al., Petsc Users Manual: Revision 3.10, Tech. Rep., Argonne National Lab.(ANL), Argonne, IL (United States), 2018.
[37] Y. Saad, Iterative Methods for Sparse Linear Systems, vol. 82, siam, 2003.
[38] K. Obaidurrahman, J. Doshi, R. Jain, V. Jagannathan, Development and validation of coupled dynamics code 'trikin' for vver reactors, Nucl. Eng.Technol 42 (3) (2010) 259–270.