Vidal-Ferràndiz, A.; Ginestar Peiro, D.; Fayez-Moustafa Moawad, R.; Verdú Martín, GJ. (2016). Moving meshes to solve the time-dependent neutron diffusion equation in hexagonal geometry. Journal of Computational and Applied Mathematics. 291:197-208. doi:10.1016/j.cam.2015.03.040.

# Moving meshes to solve the time-dependent neutron diffusion equation in hexagonal geometry

A. Vidal-Ferràndiz[a], R. Fayez[a], D. Ginestar[b,*], G. Verdú[a]

[a] *Instituto de Seguridad Industrial: Radiofísica y Medioambiental, Universitat Politècnica de València, Camino de Vera s/n, 46022, València, Spain*

[b] *Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera s/n, 46022, València, Spain*

**Abstract**

To simulate the behaviour of a nuclear power reactor it is necessary to be able to integrate the time-dependent neutron diffusion equation inside the reactor core. Here the spatial discretization of this equation is done using a finite element method that permits *h-p* refinements for different geometries. This means that the accuracy of the solution can be improved refining the spatial mesh (*h*-refinement) and also increasing the degree of the polynomial expansions used in the finite element method (*p*-refinement). Transients involving the movement of the control rod banks have the problem known as the rod-cusping effect. Previous studies have usually approached the problem using a fixed mesh scheme defining averaged material properties. The present work proposes the use of a moving mesh scheme that uses spatial meshes that change with the movement of the control rods avoiding the necessity of using equivalent material cross sections for the partially inserted cells. The performance of the moving mesh scheme is tested studying one-dimensional and three-dimensional benchmark problems.

*Keywords:* Rod-cusping problem, Moving mesh scheme, Finite element method, neitron diffusion equation

## 1. Introduction

The neutron diffusion equation is an approximation of the neutron transport equation that states that the neutron current is proportional to the gradient of the neutron flux by means of a diffusion coefficient. This approximation is analogous to the Fick's law in species diffusion and to the Fourier's law in heat transfer. For a given transient, the balance of neutrons inside a nuclear reactor core can be modelled using the time dependent neutron diffusion equation in the two energy groups approximation assuming that fission neutrons are born in the fast group and there is no up-scattering [1]. This model is of the form of

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

$$\frac{\partial C_k}{\partial t} = \beta_k [\nu\Sigma_{f1}\ \nu\Sigma_{f2}]\Phi - \lambda_k C_k \quad , \quad k = 1, \ldots, K \quad , \tag{2}$$

*Corresponding author

*Email addresses:* anvifer2@upv.es (A. Vidal-Ferràndiz), rafamou@upv.es (R. Fayez), dginesta@mat.upv.es (D. Ginestar), gverdu@iqn.upv.es (G. Verdú)

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

$$\mathcal{L} = \begin{pmatrix} -\vec{\nabla}\cdot(D_1\vec{\nabla}) + \Sigma_{a1} + \Sigma_{12} & 0 \\ -\Sigma_{12} & -\vec{\nabla}\cdot(D_2\vec{\nabla}) + \Sigma_{a2} \end{pmatrix}, \quad [\mathrm{v}^{-1}] = \begin{pmatrix} \frac{1}{\mathrm{v}_1} & 0 \\ 0 & \frac{1}{\mathrm{v}_2} \end{pmatrix},$$
$$\mathcal{M} = \begin{pmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \\ 0 & 0 \end{pmatrix}, \quad \Phi = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}, \quad \chi = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

where $\phi_1$ and $\phi_2$ are the fast and thermal neutron fluxes, respectively. The diffusion constants and cross-sections, $D_g$, $\Sigma_{12}$, $\Sigma_{ag}$, $\nu\Sigma_{fg}$, $g = 1, 2$, appearing in the equations depend on the reactor materials, that is, they are position and time dependent functions. $\beta_k$ is the yield of delayed neutrons in the $k$-th precursors group and $\lambda_k$ is the corresponding decay constant. Both coefficient are related to the delayed neutron precursor decay.

For the spatial discretization of the neutron diffusion equation different methods have been proposed. Core-level codes traditionally use nodal methods. In these methods, the neutron diffusion equation is integrated over large homogenized regions, known as nodes, to obtain a balance with average surface currents and fluxes as unknowns. Modern nodal methods usually rely on the nodal expansion method (NEM) [2] and the analytical nodal method (ANM) [3, 4] to overcome the problem in the recalculation of coupling coefficients. Also, other methods as the nodal collocation method [5, 6] and the high order finite element method [7] have been satisfactorily used.

In this work, for the spatial discretization of the neutron diffusion equation a high order $hp$-finite element method for reactors with both rectangular and hexagonal geometry has been used [8]. The main characteristic of this method is that allows to increase the accuracy of the solution refining the spatial mesh ($h$-adaptivity) and also increasing the degree of the polynomial expansions used in the finite element method ($p$-adaptivity) allowing to obtain solutions to the problem with high accuracy in a reasonable amount of time. The $h$-$p$ finite element method used in this work has been implemented using the open source finite elements library Deal.II [9]. With the help of the library, the code proposed is dimension independent and can manage different cell sizes and different types of finite elements [10]. In order to solve the algebraic problems resulting from the discretizations, the numerical libraries PETSc [11], and SLEPc [12], have been used.

Different methods have been proposed for the time discretization of the time-dependent neutron diffusion equation [13]. Standard methods use backward difference formulas [14]. These methods, for each time step, need to solve a system of linear equations, which is large and sparse. Preconditioned iterative methods are used to solve these systems [15], [16]. Other kind of methods such as modal methods [17] or the quasi-static method [18] have been also used in nuclear engineering.

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 manoeuvres in the reactor operation. The simulation of these transients presents what is known as the *rod-cusping* problem. This problem is a non-physical behaviour of different magnitudes as the neutronic power and the $k$-effective of the reactor along the transient. This problem is caused by the use of fixed mesh schemes and averaged material properties for the partially rodded node, as Figure 1 represents. When a control rod is partially inserted in a node, this node is divided into two parts: the upper part of the node, where the cross sections are modified due to the effect of the control rod, and the lower part of the node, which has the cross sections without modifications and the cross sections of the whole node are calculated by means of an interpolation procedure taking into account the position of the control rod tip. To avoid the rod-cusping effect on nodal methods, different strategies have been developed [19] as, for example, a flux weighting method [20], [21], and an equivalence method [20]. These methods have to solve a small one-dimensional eigenvalue problem for each one of the partially rodded nodes. Then, different schemes are applied to obtain the new cross sections of the partially rodded node using the old cross sections of the two parts of the node (the rodded and the unrodded) and the heterogeneous flux for the small isolated problem with suitable boundary conditions. Usually, these solutions are improved using axial assembly discontinuity factors in the top and bottom of the partially rodded node. These discontinuity factors can be found by solving the two-group diffusion equations with flux-volume weighted cross sections and fixed current boundary conditions at the top and bottom of the node. Also some approaches have been discussed to estimate the flux distribution inside the partially rodded node [22]. Other strategy is based on interpolating the solution on refined

meshes near the moving control rod [23].

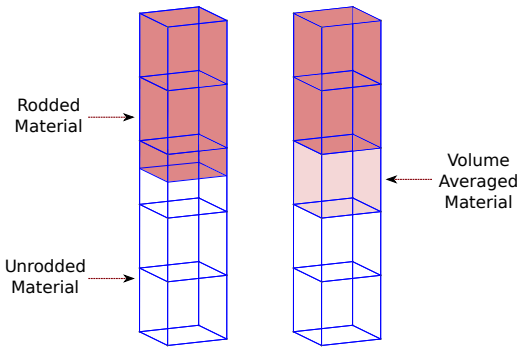



Figure 1: Rod-cusping problem in a fixed mesh scheme.

In this work, a moving mesh strategy is developed to reduce the rod-cusping problem. This method is based on the use of different spatial meshes for the different time steps following the movement of the control rod avoiding the necessity of the use of averaged material properties, as it is observed in Figure 2. To avoid the hanging nodes problem [24], the spatial mesh is moved in the same way for all the axial plane. The solutions obtained in each time step for the physical quantities are interpolated to a new spatial mesh in each time step.

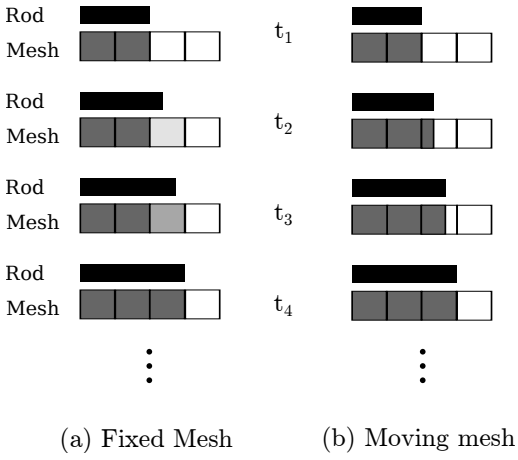



Figure 2: 1D representation of fixed and moving mesh schemes.

The rest of the paper is organized as follows, in Section 2, the spatial discretization used for the neutron diffusion equation is briefly presented. Then the time discretization of the problem is explained in Section 3, and the specific interpolation used for the moving mesh scheme is presented in subsection 3.1. To test the performance of the method, several benchmarks are analysed in Section 4, including the validation of the meothod for static conditions together with dynamic results. Finally, the main conclusions of the paper are summarized in Section 5.

## 2. Spatial Discretization

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

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

This problem is known as the *Lambda Modes problem* for a given configuration of the reactor core [6]. The fundamental eigenvalue (the largest one) is called the $k$-effective of the reactor core, and this eigenvalue and its corresponding eigenfunction describe the steady state neutron distribution in the core. In this way, the calculation of the stationary neutron flux distribution is the first step for any transient analysis. To solve both problems (1), (2) and (3), a spatial discretization of the equations has to be selected. In this work, a high order Galerkin finite element method [25] is used leading to an algebraic eigenvalue problem associated with the discretization of equation (3) with the following block structure,

$$\begin{pmatrix} L_{11} & 0 \\ -L_{21} & L_{22} \end{pmatrix}\tilde{\Phi} = \frac{1}{\lambda}\begin{pmatrix} M_{11} & M_{12} \\ 0 & 0 \end{pmatrix}\tilde{\Phi}\,, \tag{4}$$

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

$$(L_{11})_{ij} = \sum_{e=1}^{N_t}\left(D_1\int_{\Omega_e}\vec{\nabla}N_i\vec{\nabla}N_j\,\mathrm{d}V - D_1\int_{\Gamma_e}N_i\vec{\nabla}N_j\,\mathrm{d}\vec{S} + (\Sigma_{a1}+\Sigma_{12})\int_{\Omega_e}N_iN_j\,\mathrm{d}V\right)\,, \tag{5a}$$

$$(L_{21})_{ij} = \sum_{e=1}^{N_t}\Sigma_{12}\int_{\Omega_e}N_iN_j\,\mathrm{d}V\,, \tag{5b}$$

$$(L_{22})_{ij} = \sum_{e=1}^{N_t}\left(D_2\int_{\Omega_e}\vec{\nabla}N_i\vec{\nabla}N_j\,\mathrm{d}V - D_2\int_{\Gamma_e}N_i\vec{\nabla}N_j\,\mathrm{d}\vec{S} + \Sigma_{a2}\int_{\Omega_e}N_iN_j\,\mathrm{d}V\right)\,, \tag{5c}$$

$$(M_{11})_{ij} = \sum_{e=1}^{N_t}\nu\Sigma_{f1}\int_{\Omega_e}N_iN_j\,\mathrm{d}V\,, \tag{5d}$$

$$(M_{12})_{ij} = \sum_{e=1}^{N_t}\nu\Sigma_{f2}\int_{\Omega_e}N_iN_j\,\mathrm{d}V\,, \tag{5e}$$

where $N_i$ is the prescribed shape function for the $i$-th node. For simplicity, the shape functions used are part of Lagrange finite elements [25]. $\Omega_e$ ($e = 1, ..., N_t$) are the reactor subdomains (cells) in which the reactor domain is divided. In the same way, $\Gamma_e$ are the corresponding subdomain surfaces which are part of the reactor frontier. More details on the spatial discretization used can be found in [8].

To solve the algebraic eigenvalue problem (4) a Krylov-Schur method is used from the SLEPc library [12]. To accelerate the computation, the generalized eigenvalue problem is reduced to an ordinary eigenvalue problem of the form,

$$L_{11}^{-1}\left(M_{11} + M_{12}L_{22}^{-1}L_{21}\right)\tilde{\phi}_1 = \lambda\tilde{\phi}_1\,, \tag{6}$$

which is solved for the dominant eigenvalue ($k_{\text{eff}}$) and its corresponding eigenvector. In this way, for each matrix-vector product it is necessary to solve two linear systems associated with $L_{11}$ and $L_{22}$, to avoid the calculation of their inverse matrices. These systems are solved by means of an iterative scheme as the preconditioned Conjugate Gradient method [26]. Particularly, a Cuthill-McKee reordering is performed to reduce the bandwidth of the matrices, together with an incomplete Cholesky factorization is used for the preconditioning.

## 3. Time discretization

Once the spatial discretization has been selected, a discrete version of the time dependent neutron diffusion equation is solved. Since the system of ordinary differential equations resulting from the discretization of the neutron diffusion equations is stiff, implicit methods are necessary. Particularly, a first order backward method is used [14], needing this method to solve a large system of linear equations for each time step.

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

$$[\tilde{v}^{-1}]\frac{d\tilde{\Phi}}{dt} + L\tilde{\Phi} = (1-\beta)M\tilde{\Phi} + \sum_{k=1}^{K}\lambda_k X C_k \,, \tag{7}$$

$$P\frac{dC_k}{dt} = \beta_k\,(M_{11}M_{21})\,\tilde{\Phi} - \lambda_k P C_k \quad , \quad k = 1,\ldots,K \quad , \tag{8}$$

where $L$ and $M$ are the matrices obtained from the spatial discretization of operators $\mathcal{L}$ and $\mathcal{M}$, whose elements are given by equations (5). Matrix $X$ and $[\tilde{v}^{-1}]$ are defined as

$$X = \begin{pmatrix} P \\ 0 \end{pmatrix}, \qquad [\tilde{v}^{-1}] = \begin{pmatrix} P\,v_1^{-1} & 0 \\ 0 & P\,v_2^{-1} \end{pmatrix},$$

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

$$P_{ij} = \sum_{e=1}^{N_t}\int_{\Omega_e} N_i N_j \mathrm{d}V \,. \tag{9}$$

The time discretization of the precursors equations (8), is done using a one-step implicit finite differences scheme. To obtain this scheme, we make the change of function

$$PC_k = e^{-\lambda_k t}B_k \,, \tag{10}$$

obtaining

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

Integrating between $t_n$ and $t$,

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

Making use of the change (10), $C_k^{n+1}$ can be expressed as

$$PC_k^{n+1} = e^{-\lambda_k \Delta t_n}PC_k^n + e^{-\lambda_k t_{n+1}}\int_{t_n}^{t_{n+1}} e^{\lambda_k \tau}\,\beta_k\,(M_{11}M_{12})\,\tilde{\Phi}(\tau)\,d\tau \,. \tag{13}$$

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

$$PC_k^{n+1} = PC_k^n e^{-\lambda_k \Delta t} + \frac{\beta_k}{\lambda_k}\left(1 - e^{\lambda \Delta t}\right)\left(M_{11}^{n+1}M_{12}^{n+1}\right)\,\tilde{\Phi}^{n+1} \,. \tag{14}$$

In the same way, Euler's backward method is used in equation (7) obtaining,

$$[\tilde{v}^{-1}]\frac{1}{\Delta t}\left(\tilde{\Phi}^{n+1} - \tilde{\Phi}^n\right) + L^{n+1}\tilde{\Phi}^{n+1} = (1-\beta)M^{n+1}\tilde{\Phi}^{n+1} + \sum_{k=1}^{K}\lambda_k X C_k^{n+1} \,. \tag{15}$$

Taking into account equation (14), equation (15) is rewritten as the system of linear equations

$$T^{n+1}\tilde{\Phi}^{n+1} = R^n\tilde{\Phi}^n + \sum_{k=1}^{K} \lambda_k e^{-\lambda_k \Delta t} X C_k^n = E^n \,, \tag{16}$$

where the matrices are defined as,

$$T^{n+1} = \frac{1}{\Delta t}[\mathrm{v}^{-1}] + L^{n+1} - \hat{a}M^{n+1} \,,$$
$$R^n = \frac{1}{\Delta t}[\mathrm{v}^{-1}] = \frac{1}{\Delta t}\begin{pmatrix} P\,\mathrm{v}_1^{-1} & 0 \\ 0 & P\,\mathrm{v}_2^{-1} \end{pmatrix} \,,$$

and the coefficient $\hat{a}$ is

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

This system of equations is large and sparse and has to be solved for each time step. As it was done in the eigenvalue problem (6), the preconditioned GMRES [26] method has been chosen to solve these systems and the preconditioner used has been the incomplete LU preconditioner.

### *3.1. Mesh Interpolation*

Traditionally the time dependent neutron diffusion equation is solved using a spatial mesh that is fixed along all the transient. As it has been already mentioned, the simulation of transients where the control rod banks move suffer from the rod-cusping problem because averaged cross sections are used for the partially rodded nodes. In this work, we propose the use of a spatial mesh that changes each time step following the control rod in such a way that we do not have partially rodded nodes. This scheme requires the interpolation of the physical solutions of the equations, which are continuous functions, from the old mesh in step $n$ to the next mesh corresponding to step $n+1$. The mesh interpolation process consists of finding the solution in the new support point values corresponding to the new mesh by polynomial interpolation of the values of the solution in the old mesh. To maintain the accuracy of the solution this interpolation is done using a polynomial interpolation of the same degree as the degree used in the high order finite element method used for the spatial discretization.

To formalise the method we use the superscript notation to refer to the time step number and the subscript notation to the mesh number step. Then, $\Phi_m^n$ refers to the neutronic flux at time step $n$ defined in the mesh $m$. The interpolation process is implemented by means of a function $f$, and can be written as

$$\Phi_{m+1}^n = f(\Phi_m^n) \,, \tag{17}$$
$$C_{k,m+1}^n = f(C_{k,m}^n) \,. \tag{18}$$

Figure 3 shows an example of the neutron flux interpolation $\Phi^n(z)$ between two consecutive meshes $m$ and $m+1$. This interpolation is similar to the one used in $h$-refined finite elements codes to interpolate from the coarse mesh to the fine mesh and accelerate the convergence of solution in the fine mesh [9]. However in the moving mesh method, the support points of the mesh are moved and not only coarse cells are subdivided into finer cells.

In the moving mesh interpolation, only physical quantities, which are continuous, can be interpolated adequately. However, from equation (14) the obtained quantity is $PC_k^n$ and the physical magnitude needed for the interpolation is $C_k^n$. To avoid the computationally expensive task of inverting matrix $P$, a mass lumping technique is considered to obtain an approximation, $\hat{P}$, of matrix $P$ [25]. This procedure mainly consists of considering as matrix $\hat{P}$ a diagonal matrix whose elements are the result of adding all the elements of each row of matrix $P$. This is equivalent to calculate the integrals involving polynomials up to order $s$ approximately with a quadrature rule up to order $s-1$ in the finite element method [27].

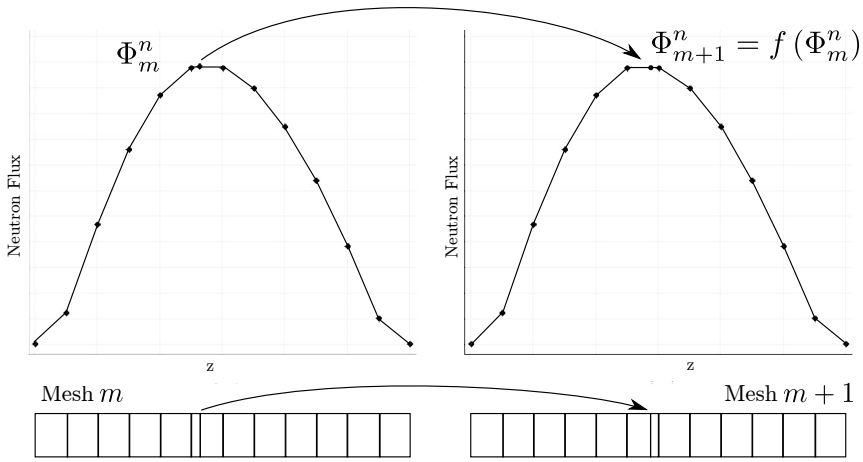



Figure 3: 1D Mesh interpolation example.

The inverse of the mass lumped matrix $\hat{P}$ is a diagonal matrix calculated as

$$\hat{P}_{ii}^{-1} = \frac{1}{\sum_{e=1}^{N_e}\left(\sum_{j=1}^{N_i}\int_{\Omega_e} N_i N_j \mathrm{d}V\right)} \,. \tag{19}$$

In the usual fixed mesh scheme, it is not necessary to know the value of $C_k^n$ because it is enough to obtain $PC_k^n$ for each time step.

The steps necessary for the implementation of the moving mesh procedure are summarized in the scheme shown in Figure 4. The computation starts with an eigenvalue computation to obtain the stationary configuration of the reactor core, which is used as initial condition. Then, the dynamic calculation starts. First, the neutron precursors concentration is solved in the initial mesh. Then the control rods and the mesh are moved and the neutronic flux and the precursors distribution are interpolated to the new mesh. Then, the system associated with the numerical scheme is solved obtaining the next flux distribution. This is clearly the most time consuming part of the computation. Finally, the stopping criterion is checked and if it is not fulfilled the dynamic computation is repeated for the next time step.

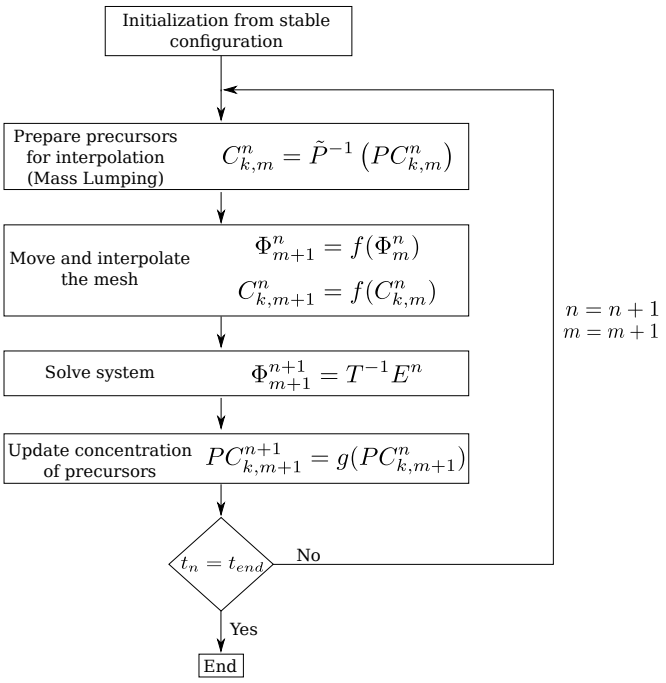



Figure 4: Scheme of the moving mesh time scheme.

## 4. Numerical Results

To study the performance of the moving mesh method proposed above, two benchmarks are studied. The first benchmark consists of a one dimensional problem where a control rod is ejected to a given velocity and then inserted back. The second benchmark is a small 3D hexagonal reactor where also a rod ejection accident is studied.

In the two benchmarks, first, the spatial discretization is tested solving the critical configuration of the reactor. Then, the time dependent problem is solved using the classical fixed mesh scheme and the proposed moving mesh scheme.

To compare the performance of the finite element method used for the spatial discretization, and to set the adequate spatial discretization parameters, different errors have been employed, which are shown in Table 1. In this Table, $Pw_i$ and $Pw_i^*$ are, respectively, the reference power and the computed power in the $i$-th cell [28] (cell averages), computed as

$$Pw_i = E_R\left(\Sigma_{f1}\phi_{i1} + \Sigma_{f2}\phi_{i2}\right) , \tag{20}$$

where $E_R$ is the recoverable energy per fission which is assumed to be constant, $V_i$ is the volume of the $i$-th cell and $V_t$ is the total volume of the reactor. $k_{\text{eff}}$ is the reference dominant eigenvalue of the reactor and $k^*_{\text{eff}}$ is the computed eigenvalue. Similar errors are defined for the neutron fluxes.

Table 1: Different errors used for the spatial discretization.

| | |
|---|---|
| Relative error | $\varepsilon_i = \lvert Pw_i - Pw_i^*\rvert/\lvert Pw_i\rvert$ |
| Mean relative error | $\bar{\varepsilon} = \frac{1}{V_t}\sum_i \varepsilon_i V_i$ |
| Maximum relative error | $\varepsilon_{\max} = \max_i \lvert\varepsilon_i\rvert$ |
| Eigenvalue error (pcm) | $\Delta k_{\text{eff}} = 10^5\ \lvert k_{\text{eff}} - k^*_{\text{eff}}\rvert$ |

### 4.1. Unidimensional problem

To validate the code a simple and small one-dimensional reactor is considered, which represents a simplified model for a rod-ejection accident. This reactor consists of 12 cells composed of different materials. The reactor geometry is defined in Figure 5 and the cross sections for the materials of each region are given in Table 2. Precursor parameters are given in Table 3. Zero-current boundary conditions are imposed at the boundaries of the system.

The transient consists of removing the control rod from time 0.0 $s$ to 4.0 $s$ with a constant velocity of 25 $cm/s$. Then the control rod is inserted again from 4.0 $s$ to 10 $s$ also with a constant velocity of 25 $cm/s$. All the transient calculations are made using cubic polynomials in the finite element method. Reference results for the neutronic flux and the $k_{\text{eff}}$ of the problem are computed with the neutronic code PARCS [29], using a fixed mesh with 120 cells where the rod-cusping problem should be eliminated.

The results obtained for the dominant eigenvalue $k_{\text{eff}}$ using different polynomial degrees for the finite element method (Degree of FE) are shown in Table 4. In this Table, also the number of degrees of freedom (DoF) are shown for the reduced eigenvalue problem (6) in order to have an idea of the size of the problem solved. Also the mean relative errors and maximum relative errors per cell for the neutronic flux and power are shown for the initial configuration for the reactor.

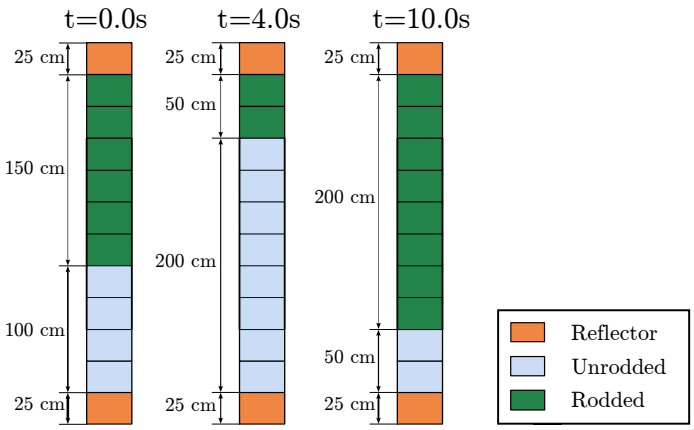



Figure 5: Geometry of the 1D reactor problem.

Table 2: Cross sections of the materials of the 1D reactor.

| **Fuel** | **Group** | $\mathbf{D_g}$ (cm) | $\boldsymbol{\Sigma_{ag}}$ (1/cm) | $\boldsymbol{\nu\Sigma_{fg}}$ (1/cm) | $\boldsymbol{\Sigma_{fg}}$ (1/cm) | $\boldsymbol{\Sigma_{12}}$ (1/cm) |
|---|---|---|---|---|---|---|
| Unrodded Fuel | 1 | 1.40343 | 1.17659e-2 | 5.62285e-3 | 2.20503e-3 | 1.60795e-2 |
| | 2 | 0.32886 | 1.07186e-1 | 1.45865e-1 | 5.90546e-2 | |
| Rodded Fuel | 1 | 1.40343 | 1.17659e-2 | 5.60285e-3 | 2.19720e-3 | 1.60795e-2 |
| | 2 | 0.32886 | 1.07186e-1 | 1.45403e-1 | 5.88676e-2 | |
| Reflector | 1 | 0.93344 | 2.81676e-3 | 0.00000e+0 | 0.00000e+0 | 1.08805e-2 |
| | 2 | 0.95793 | 8.87200e-2 | 0.00000e+0 | 0.00000e+0 | |

Figure 6 shows a detail of the evolution of the normalized mean power,

$$\bar{P}w(t) = \frac{\int_\Omega \left(\Sigma_{f1}\phi_1(t) + \Sigma_{f2}\phi_2(t)\right) \mathrm{d}V}{\int_\Omega \left(\Sigma_{f1}\phi_1(0) + \Sigma_{f2}\phi_2(0)\right) \mathrm{d}V} \ , \tag{21}$$

Table 3: Neutron precursors parameters for the reactor.

| | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 |
|---|---|---|---|---|---|---|
| $\beta_i$ | 0.000247 | 0.0013845 | 0.001222 | 0.026455 | 0.000832 | 0.000169 |
| $\lambda_i\ (1/s)$ | 0.0127 | 0.0317 | 0.115 | 0.311 | 1.4 | 3.87 |
| | $v_1 = 1.27 \times 10^7$ cm/s | | $v_2 = 2.5 \times 10^5$ cm/s | | $\beta = 0.0065$ | |

Table 4: Dominant eigenvalue and power distribution results for the 1D reactor.

| **Degree of FE** | **DoF** | **$k_{eff}$** | **$\Delta k_{eff}$** (pcm) | **Power** $\bar{\varepsilon}$ (%) | **Power** $\varepsilon_{max}$(%) | **Fast Flux** $\bar{\varepsilon}$ (%) | **Fast Flux** $\varepsilon_{max}$(%) | **Thermal Flux** $\bar{\varepsilon}$ (%) | **Thermal Flux** $\varepsilon_{max}$(%) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 13 | 0.978430 | 38.1 | 2.98 | 9.84 | 7.17 | 29.2 | 7.64 | 31.62 |
| 2 | 25 | 0.978757 | 5.4 | 0.49 | 1.54 | 1.02 | 3.85 | 0.62 | 1.55 |
| 3 | 37 | 0.978801 | 1.0 | 0.10 | 0.38 | 0.16 | 0.53 | 0.10 | 0.41 |
| PARCS | | 0.978811 | | | | | | | |

in the reactor during the transient computed using a classical fixed mesh scheme with a mesh of 12 nodes, the moving mesh scheme presented in this work and the reference values. As can be seen in this Figure, the fixed mesh computations present some unphysical jumps in the normalized mean power, mainly when the control rod is in the middle of a cell. However the rod-cusping problem is mitigated with the moving mesh scheme reducing the mean error in the power about three times, from 0.3% to 0.13%. Moreover, the relative errors for the reactor mean power for each one of the time steps obtained with the fixed mesh scheme and the moving mesh scheme are shown in Figure 7. Figure 8 shows the errors in the computation of the dominant eigenvalue ($\Delta k_{eff}$) solving an static problem for all the time steps of the transient. As it can be seen in these Figures, the errors for the $k_{eff}$ and the reactor power have very similar behaviours.

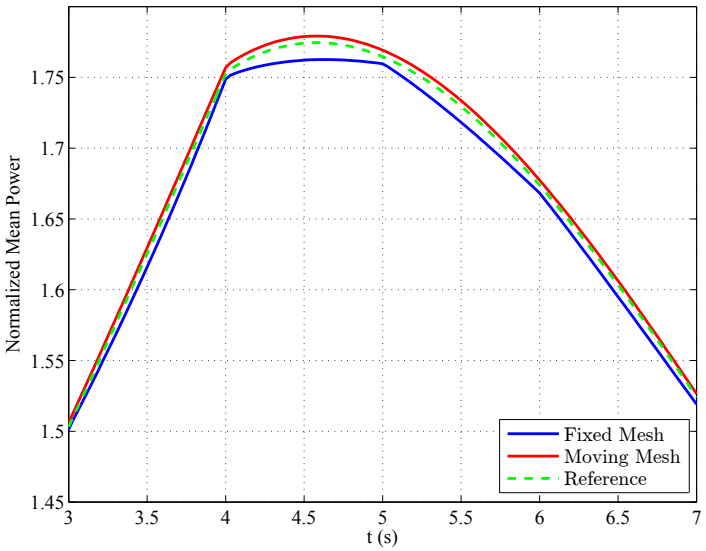



Figure 6: Normalized power evolution for the 1D reactor from 3s to 7s.

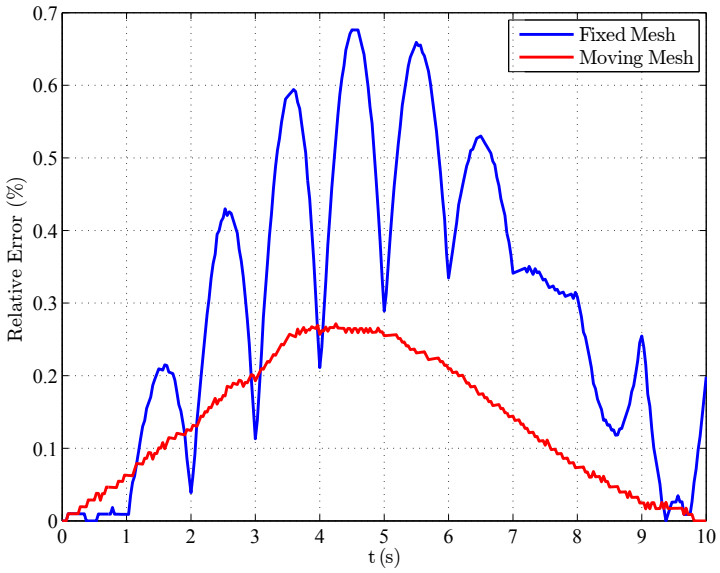



Figure 7: Comparative of errors over time in 1D reactor.

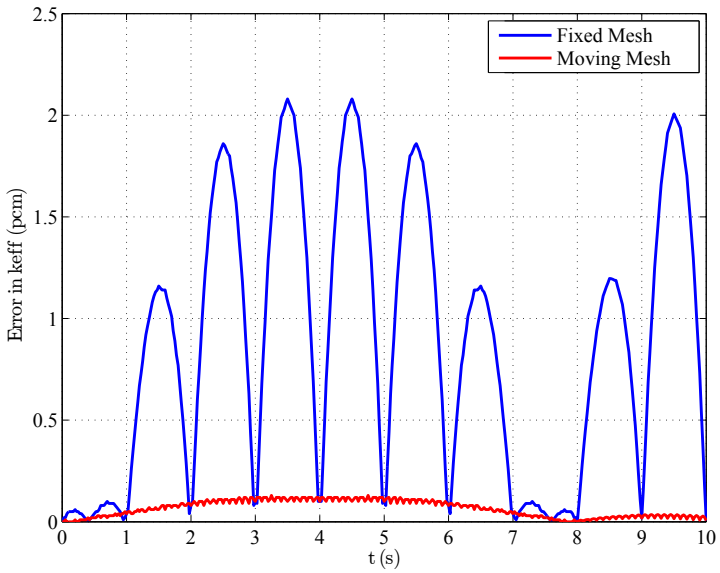



Figure 8: Errors in $k_{\text{eff}}$ during the transient.

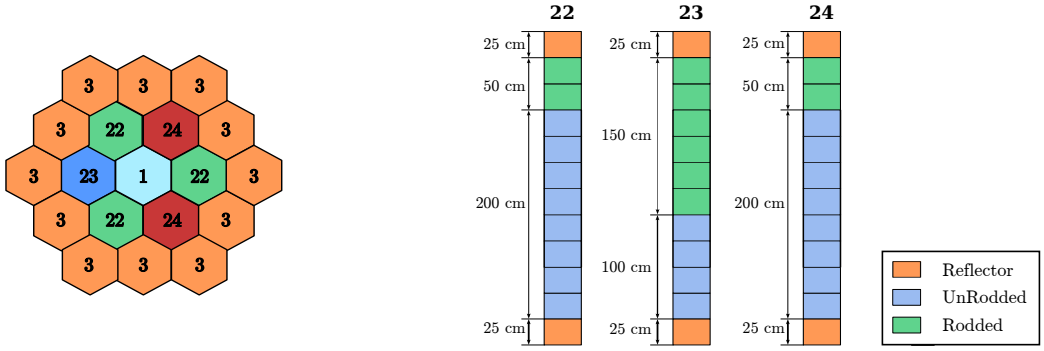
### *4.2. Three dimensional problem*

To test the performance of the method in 3D reactors, a small reactor that presents a large rod-cusping problem is studied [30]. Figure 9 shows the layout map of the core, for which the hexagonal lattice pitch is 23.6 cm. The material cross sections of the different materials composing the reactor are given in Table 5. The neutron precursors data used in this problem are given in Table 3.

Albedo boundary conditions are applied on the outer edge of the reflector cells. The extrapolation length is set to $2 \times D_g$. The height of the reactor is 300 cm and 12 axial planes are considered, each one of 25 cm.

As the Deal.II library cannot handle hexagonal finite elements, each hexagon is subdivided into 3 quadrilaterals with the help of the mesh generation code Gmsh [31], as it is shown in Figure 10. Thus, the used mesh for this reactor has a total 684 cells.

The transient simulates a rod ejection accident as follows. Starting from the initial configuration, see Figure 9, the rod 23 begins to be removed until it is completely removed at time $t = 0.15$s remaining only the unrodded fuel. From $t = 0.15$s until $t = 1.0s$ nothing happens. Then, the security system acts inserting the rods 22 at constant velocity of 25 cm/s until the bottom of the reactor is reached at $t = 9.0$s.




(a) Core definition.

(b) Rods definition.

Figure 9: Small 3D reactor geometry.

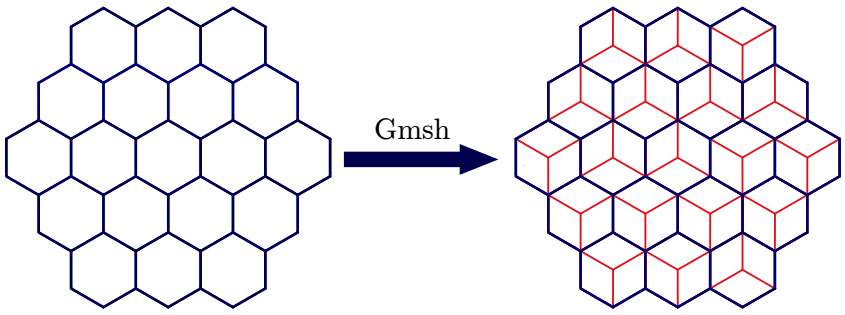



Figure 10: Hexadric cells transformed into quadrilaterals with Gmsh.

Table 5: Cross section definition for the small 3D reactor.

| **Fuel** | **Group** | $\mathbf{D_g}$ (cm) | $\mathbf{\Sigma_{ag}}$ (1/cm) | $\mathbf{\nu\Sigma_{fg}}$ (1/cm) | $\mathbf{\Sigma_{fg}}$ (1/cm) | $\mathbf{\Sigma_{12}}$ (1/cm) |
|---|---|---|---|---|---|---|
| Unrodded Fuel | 1 | 1.40343 | 1.17659e-2 | 5.62285e-3 | 2.20504e-3 | 1.60795e-2 |
| | 2 | 0.32886 | 1.07186e-1 | 1.45865e-1 | 6.00267e-2 | |
| Rodded Fuel | 1 | 1.36764 | 1.39118e-2 | 5.37719e-3 | 2.10870e-3 | 1.35108e-2 |
| | 2 | 0.25108 | 9.96214e-2 | 1.15403e-1 | 4.74909e-2 | |
| Reflector | 1 | 0.93344 | 2.81676e-3 | 0.00000e+0 | 0.00000e+0 | 1.08805e-2 |
| | 2 | 0.95793 | 8.87200e-2 | 0.00000e+0 | 0.00000e+0 | |

The results obtained for the dominant eigenvalue ($k_{\text{eff}}$) using different polynomial degrees for the finite element method (Degree of FE) are shown in Table 6. In this Table, the number of degrees of freedom (DoF) of the reduced eigenvalue problem are also shown together with the mean relative errors and maximum relative errors per cell in the neutronic flux and power for the initial configuration of the reactor.

Table 6: Critical eigenvalue and power distribution results for the small 3D reactor.

| **Degree of FE** | **DoF** | **$k_{\text{eff}}$** | **$\Delta k_{\text{eff}}$** (pcm) | **Power** | | **Fast Flux** | | **Thermal Flux** | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | $\bar{\varepsilon}$ (%) | $\varepsilon_{\max}$(%) | $\bar{\varepsilon}$ (%) | $\varepsilon_{\max}$(%) | $\bar{\varepsilon}$ (%) | $\varepsilon_{\max}$(%) |
| 1 | 949 | 0.801287 | 1539.0 | 6.38 | 17.70 | 15.49 | 38.13 | 16.34 | 36.45 |
| 2 | 6475 | 0.815211 | 146.4 | 0.97 | 0.84 | 1.98 | 3.40 | 1.14 | 3.26 |
| 3 | 20683 | 0.816024 | 65.3 | 0.15 | 0.35 | 0.17 | 0.66 | 0.23 | 0.81 |
| PARCS | | 0.816677 | | | | | | | |

Figure 11 shows the time evolution of the normalized mean power of the reactor in the first 0.15 seconds. In this Figure, the results obtained with the moving mesh scheme proposed in this work are compared with the results obtained with the classical fixed mesh scheme. The reactor is solved with a fixed mesh scheme using 120 axial planes where the rod-cusping problem is very small and the results of this computation are taken as a reference. All transient calculations are made using cubic polynomials in the finite element method.

The relative errors for the reactor mean power for each one of the time steps obtained with the fixed mesh scheme and the moving mesh scheme are shown in Figure 12. Thus, the moving mesh scheme reduces the mean error from 5.7% to 0.50%. As it can be seen in these Figures the moving mesh scheme produces better results than the fixed mesh scheme when a small number of axial planes are considered for the spatial discretization.

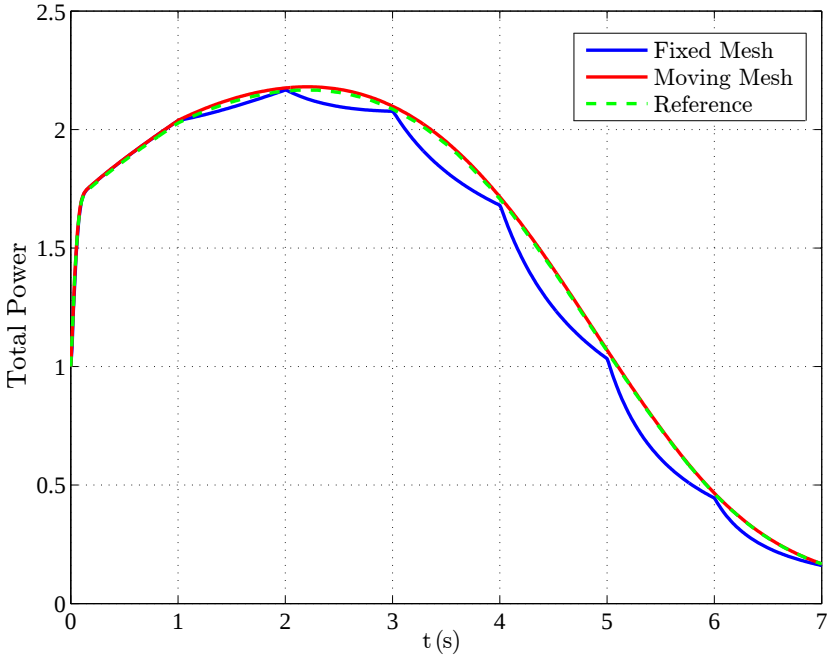



Figure 11: Normalized mean power evolution for the small 3D reactor.

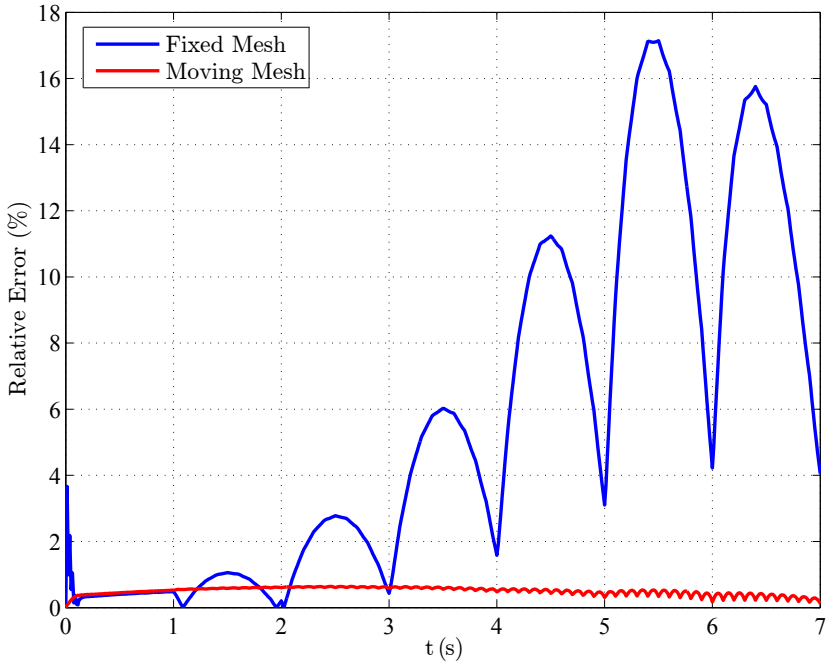



Figure 12: Normalized mean power error for the small 3D reactor.

## 5. Conclusions

Many transients in nuclear power reactors involve the movement of the control rod banks. For the simulation of this kind of transients with the classical methods, it is necessary to define equivalent material properties corresponding to partially inserted cells during the movement of the control rods. Volume averaged techniques are used to define this equivalent cross-sections, but this procedure leads to unphysical behaviour of some magnitudes during the simulation when a small number of axial planes are used in the spatial discretization and this problem is known as the rod cusping effect. To avoid it, a new method based on a high-order finite element method is proposed in this work. In this new method, the spatial mesh is moved together with the control rods in such a way that there is no partially inserted cells. The solutions of the physical magnitudes are transferred between different spatial meshes using a polynomial interpolation. To study the performance of the moving mesh scheme, two benchmark problems have been analysed. A one-dimensional reactor problem and a small three-dimensional reactor. The spatial discretization was validated solving the stationary state of the reactor. Cubic polynomials were used in the finite element method as a compromise between accuracy and problem size. Also, it is shown in both problems studied that the moving mesh method has a better performance that the traditional fixed mesh scheme when a small number of axial cells are used. The moving mesh scheme permits to use a coarser discretization and reduces the computational effort.

## Acknowledgements

The work has been partially supported by the spanish Ministerio de Economía y Competitividad under projects ENE 2014-59442-P and MTM2014-58159-P, the Generalitat Valenciana under the project PROMETEO II/2014/008 and the Universitat Politècnica de València under the project FPI-2013.

The work has also been partially supported by the Swedish Research Council (VR-Vetenskapsrådet) within a framework grant called DREAM4SAFER, research contract C0467701.

## References


[1] W. M. Stacey, Nuclear reactor physics, John Wiley & Sons, 2007.

[2] T. Singh, T. Mazumdar, P. Pandey, NEMSQR: A 3-D multi group diffusion theory code based on nodal expansion method for square geometry, Annals of Nuclear Energy 64 (2014) 230 – 243.

[3] A. Hébert, A simplified presentation of the multigroup analytic nodal method in 2-d cartesian geometry, Annals of Nuclear Energy 35 (11) (2008) 2142 – 2149.

[4] K. S. Smith, An analytic nodal method for solving the two-group, multidimensional, static and transient neutron diffusion equations, Ph.D. thesis, Massachusetts Institute of Technology (1979).

[5] A. Hébert, Development of the nodal collocation method for solving the neutron diffusion equation, Annals of Nuclear Energy 14 (10) (1987) 527 – 541.

[6] G. Verdú, D. Ginestar, V. Vidal, J. Muñoz-Cobo, 3D Lambda-modes of the neutron-diffusion equation, Annals of Nuclear Energy 21 (7) (1994) 405 – 421.

[7] S. González-Pintor, D. Ginestar, G. Verdú, High order finite element method for the Lambda modes problem on hexagonal geometry, Annals of Nuclear Energy 36 (9) (2009) 1450 – 1462.

[8] A. Vidal-Ferrandiz, R. Fayez, D. Ginestar, G. Verdú, Solution of the lambda modes problem of a nuclear power reactor using an hp finite element method, Annals of Nuclear Energy 72 (0) (2014) 338 – 349.

[9] W. Bangerth, R. Hartmann, G. Kanschat, deal.II – a general purpose object oriented finite element library, ACM Trans. Math. Softw. 33 (4) (2007) 24/1–24/27.

[10] W. Bangerth, O. Kayser-Herold, Data structures and requirements for hp finite element software, ACM Trans. Math. Softw. 36 (1) (2009) 4/1–4/31.

[11] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.

[12] V. Hernandez, J. E. Roman, V. Vidal, SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems, ACM Trans. Math. Software 31 (3) (2005) 351–362.

[13] G. Verdú, D. Ginestar, V. Vidal, J. Muñoz-Cobo, A consistent multidimensional nodal method for transient calculations, Annals of Nuclear Energy 22 (6) (1995) 395 – 410.

[14] D. Ginestar, G. Verdú, V. Vidal, R. Bru, J. Marín, J. Muñoz-Cobo, High order backward discretization of the neutron diffusion equation, Annals of Nuclear Energy 25 (13) (1998) 47 – 64.

[15] R. Bru, D. Ginestar, J. Marín, G. Verdú, J. Mas, T. Manteuffel, Iterative schemes for the neutron diffusion equation, Computers & Mathematics with Applications 44 (10) (2002) 1307–1323.

[16] S. González-Pintor, D. Ginestar, G. Verdú, Preconditioning the solution of the time-dependent neutron diffusion equation by recycling krylov subspaces, International Journal of Computer Mathematics 91 (1) (2014) 42 – 52.

[17] R. Miró, D. Ginestar, G. Verdú, D. Hennig, A nodal modal method for the neutron diffusion equation. application to BWR instabilities analysis, Annals of Nuclear Energy 29 (10) (2002) 1171 – 1194.

[18] S. Dulla, E. H. Mund, P. Ravetto, The quasi-static method revisited, Progress in Nuclear Energy 50 (8) (2008) 908 – 920.

[19] 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), Seoul, Korea, 2002.

[20] M. W. F. Bennewitz, H. Finnemann, Higher-order corrections in nodal reactor computations, Trans. Am. Nucl. Soc. 22 (1975) 205.

[21] S. González-Pintor, G. Verdú, D. Ginestar, Correction of the rod cusping effect for a high order finite element method, in: Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011). Rio de Janeiro, RJ, Brazil, May 8-12, 2011, on CD-ROM, Latin American Section (LAS) / American Nuclear Society (ANS), 2011.

[22] A. Yamamoto, A simple and efficient control rod cusping model for three-dimensional pin-by-pin core calculations, Nuclear Technology 145 (2004) 11–17.

[23] D. Gilbert, J. Roman, W. J. Garland, W. Poehlman, Simulating control rod and fuel assembly motion using moving meshes, Annals of Nuclear Energy 35 (2) (2008) 291 – 303.

[24] P. Šolín, J. Červený, I. Doležel, Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM, Mathematics and Computers in Simulation 77 (1) (2008) 117 – 132.

[25] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The finite element method: its basis and fundamentals, Butterworth-Heinemann, 2005.

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

[27] G. E. Karniadakis, S. J. Sherwin, Spectral/hp element methods for computational fluid dynamics, Numerical mathematics and scientific computation, Oxford University Press, Oxford, New York, Aukland, 2005.

[28] J. Lamarsh, Introduction to Nuclear Reactor Theory, American Nuclear Society, 2002.

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

[30] S. González-Pintor, D. Ginestar, Reducing rod cusping effect in nodal expansion method calculations, Joint International Conference Mathematics and Computation.

[31] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (11) (2009) 1309–1331.