Muñoz-Cobo, J.; Miró Herrero, R.; Wysocki, A.; Soler Domingo, A. (2019). 3D calculation of the lambda eigenvalues and eigenmodes of the two-group neutron diffusion equation by coarse-mesh nodal methods. Progress in Nuclear Energy. 110:393-409. https://doi.org/10.1016/j.pnucene.2018.10.008

3D Calculation of the Lambda Eigenvalues and Eigenmodes of the Two-Group Neutron Diffusion Equation by Coarse-Mesh Nodal Methods

J.L. Muñoz-Cobo[1,*], R. Miró[1], Aaron Wysocki[2], A. Soler[3]

1 Universitat Politècnica de València, Camino de Vera 14, Valencia (Spain)

2 ORNL Oak Ridge National Laboratory, Oak Ridge 37830 (TN) USA

3 NFQ, Advisory Solutions, Claudio Coello 78, Madrid, Spain

- Corresponding author email: jlcobos@iqn.upv.es

Abstract

This paper shows the development and roots of the lambda eigenvector and eigenmodes calculation by coarse-mesh finite difference nodal methods. In addition this paper shows an inter-comparison of the eigenvalues and power profiles obtained by different 3D nodal methods with two neutron groups, as the nodal collocation method (Verdú et al 1998, 1993, Hebert 1987) with different Legendre expansion orders, the modified coarse-mesh nodal method explained in this paper, and the method implemented in the PARCS code by Wysocki et al.(2014, 2015). In this paper we have developed a program NODAL-LAMBDA that uses a two-group modified coarse-mesh finite difference method with albedo boundary conditions. Some of the approximation performed originally by Borressen (1971) have been discarded and more exact expressions have been used. We compare for instance the eigenvalues and power profiles obtained with Borressen original approach of 1.5 group and with 2 groups. Also some improvements in the Albedo boundary conditions suggested by (Turney 1975, Chung et al 1981) have been incorporated to the code as an option. The goal is to obtain the eigenvalues and the sub-criticalities $(k_i - k_0)$ of the harmonic modes that can be excited during an instability event in a fast way and with an acceptable precision.

Keywords: Diffusion Equation Lambda Modes, Coarse Mesh Nodal Methods, Subcriticality of Out of Phase Modes

1. Introduction

The problem of the lambda eigenvectors and eigenvalues of the neutron diffusion equation plays an important role in the out-of-phase instabilities of boiling water reactors (BWR), and in the xenon spatial oscillations as was pointed out in the past by Doring et al (1993), Verdú et al (1994), Muñoz-Cobo et al (2001, 2002). The first step for solving this problem is the discretization in space of the two group neutron diffusion equation. On account of the problem boundary conditions, this equation is not self-adjoint and therefore the matrices that are obtained by discretization are not symmetric.

The important consequence of this fact is that we cannot apply the classical Lanczos methods to solve the matrix eigenvalue problem that merge after discretization in 3D space (Golub and Van Loan 1996).

Different methods have been used in the past to discretize the two-group neutron diffusion equations, Verdú et al (1993), and Hebert (1987) employed a nodal collocation method that performs a expansion of the neutron flux in Legendre polynomials, at the center of each nodal cell or node, in the three direction of space, followed by integration over the node volume. In this approach the unknowns are the coefficients of the expansion at each node, and obviously the number of unknowns increase with the total number of nodes and the expansion order. Other nodal methods exist as the coarse mesh nodal methods (Ott and Bezella 1989) or the modified coarse-mesh finite difference methods (Borresen 1971), in these methods the neutron diffusion equations are integrated over the volume of each node and the terms containing the divergence of the neutron currents reduce to a summation of the neutron currents through the boundary faces of each node, also the terms containing the products of the neutron flux times the macroscopic cross sections when integrated over each node and on account of the fact that the macroscopic cross sections have been previously homogenized at each node yield an expression that depends on the average neutron flux at each node. In this approach, the volume averaged fluxes (fast and thermal) at a given node n are expressed in terms of the neutron flux at the center of the given node n and the six neutron fluxes at the centers of the six surrounding nodes, plus two unknown parameters: one for the fast group and another one for the thermal group that must be determined from auxiliary calculations. The advantage is that these auxiliary parameters are always the same for a given type of reactor and were obtained by Borresen (1971). Recently Vidal Ferrandiz et al (2014) developed a h–p finite element method to compute the lambda eigenvalues and eigenvectors of the 3D PWR IAEA benchmark problem and the BIBLIS problem with good results for the eigenvalues and the eigenvectors when increasing the number of cells and the finite element degree p. Other modern nodal methods usually rely in the Nodal Expansion Method (NEM) (Singh et al., 2014), to avoid the problem of recalculating the coupling coefficients. Also is worthy to mention that Bernal et al (2014) have developed a method to obtain the eigenvalues and the eigenvectors of the two group neutron diffusion equation using the Finite Volume Method (FVM), and that can be easily applied to unstructured meshes.

In this paper we have developed a program NODAL-LAMBDA that uses a two-group modified coarse-mesh finite difference method with albedo boundary conditions. Some of the approximation performed originally by Borressen (1971) have been discarded and more exact expressions have been used. Also some improvements in the albedo boundary conditions suggested by (Turney 1975, Chung et al 1981) have been incorporated to the code as an option. The goal is to obtain the eigenvalues and the sub-criticalities $(k_i - k_0)$ of the harmonic modes that can be excited during an instability event in a fast way and with an acceptable precision. By computing the eigenvalue separation for a given reactor configuration it is possible to know in advance if a given mode can be excited or not for that configuration. This could happens if the subcriticality of the mode is about 1$ or less.

Once discretized the two group neutron diffusion equations for the eigenvalue problem, it must be solved obtaining the desired eigenvalues and eigenvectors or lambda modes. Normally for nuclear reactor problems one is interested in the biggest eigenvalues and its corresponding eigenvectors. The extension of the Lanczos method (Golub and Van Loan 1996, Tyrtyshnikov1997) by Arnoldi (1951) to compute eigenvalues and eigenvectors of large, sparse and non-symmetric Eigen-problems of the type $AX = \lambda X$ involves the generation of an orthogonal transformation Q_j that reduces the matrix A to the Hessenberg form i.e.$Q_j^T A Q_j = H_j$, where H_j is the Hessemberg reduction of A by the orthogonal transformation Q_j. The columns $[q_1, q_2, \dots, q_j]$ of Q_j are the Arnoldi vectors that are built through an orthogonalization process of the vector $Aq_j$to the preceding vectors $q_1, q_2, \dots, q_j$. The important issue is that the Arnoldi vectors $(q_1, q_2, \dots, q_i)$ generated in this way are a basis of the Krylov subspace $\mathcal{K}_i(q_1, A)$. In addition if this subspace is invariant under the action of A for some $i \geq j$, i.e. $\mathcal{K}_i(q_1, A) = \mathcal{K}_{i+1}(q_1, A)$, it follows that the spectrum of H_i is contained in the spectrum of A, i.e. $\sigma(H_i) \subset \sigma(A)$. The important consequence for nuclear reactor calculations is that the eigenvalues of the projective restrictions H_j can estimate with a sufficient degree of approximation, the eigenvalues of A, although the dimension of the projected sub-space $\mathcal{K}_j(q_1, A)$is much smaller than the invariant subspace$\mathcal{K}_i(q_1, A)$.

In nuclear reactor calculations we are always interested in the biggest eigenvalues and their corresponding eigenvectors, but the Arnoldi method does not always provide the desired eigenvalues and eigenvectors. Unwanted eigenvalues and eigenvectors are obtained mixed with the wanted ones. Saad(1980, 1984,1992), and Sorensen (1992) developed the so called Arnoldi Methods with Restarting; the most popular of these methods is the Arnoldi method with implicit restarting. What these authors discovered was that running the Arnoldi method with j steps and then restarting the process with a new initial vector $q_1^{(r)}$at the r-th iteration properly selected from the span of the Arnoldi vectors of the previous (r-1)–th iteration is crucial for obtaining a good estimation of the desired eigenvalues and eigenvectors (lambda modes). The NODAL-LAMBDA code uses the implicit restarting algorithm that uses QR iterations with shifts and is due to Sorensen (1992), and it is implemented in the ARPACK code (Arnoldi Package) due to Lelhoucq et al. (1998). Verdú et al. (1998) applied the implicit Arnoldi restarted method (IRAM) to a nuclear power reactor core to calculate the fundamental and subcritical modes, using a nodal collocation method with an expansion in Legendre polynomials inside each node. Also the IRAM method was used by Miró et al (2002). Wysocki et al. (2014, 2015) implemented the ARPACK IRAM solver into the PARCS (Purdue Advanced Reactor Core Simulator) code and performed coupled neutronic/thermal-hydraulic simulations of BWR limit cycles using PARCS coupled with TRACE (TRAC/RELAP Advanced Computational Engine).

The paper is organized as follows:

Section 2 is devoted to the fundamental concepts of the coarse-mesh nodal methods specially the family of coarse mesh nodal finite difference methods. Special attention is given to the roots of the Borressen method (1971) for two neutron groups although originally was developed for 1.5 groups. Sections 2.2 explain the deduction of the ALBEDO boundary conditions by means of the Laplace transform, and the implementation of the ALBEDO boundary conditions in the NODAL-LAMBDA code, and the modifications of the matrix elements of the production and leakage operators in the nodes with boundary conditions. Section 3 is devoted to the solution of the 3D eigenvalue matrix equations with two neutron groups and ALBEDO boundary conditions. These equations are solved by means of the Arnoldi with Implicit Restarting Method (IRAM) that is the method implemented in the NODAL-LAMBDA code. The fundamental root of the method is explained in this section. Because the Arnoldi method also provides unwanted eigenvalues, we study also in this section the algorithm used to filter out the unwanted eigenvalues from the Ritz spectrum or spectrum of the projected matrix on the Krylov subspace.

Section 4 is devoted to the validation of the NODAL-LAMBDA program comparing its results with the results of other programs based on different methods. At first we have compared the results of the NODAL-LAMBDA program with the results obtained with the LAMBDA program that uses a Nodal Collocation Methods (Miró et al 2002, Miró 2002) for the Ringhals stability Benchmark comparing the eigenvalues of the fundamental and the subcritical modes, and the power axial profiles.

Section 4 also display the results obtained with the NODAL-LAMBDA program for the IAEA Benchmark (Michelsen 1977) with the results obtained with the nodal collocation methods for order 2 and 3 of the expansion in Legendre Polynomials, and the results obtained with very fine mesh and extrapolation with VENTURE code but only for the fundamental mode in this last case. Also we show the results obtained with the subroutine implemented by Wysocki et al (2014) in the PARCS code to compute the lambda modes.

Also we have performed a modification of the NODAL-LAMBDA program for 1.5 groups and we also compare in section 4 the results obtained with 2 and 1.5 groups.

Finally Dong-Chung et al (1981) proposed some modifications in the Albedo Boundary conditions, we have performed a comparison of the eigenvalues and the power profile obtained using the recommended modification of Dong-Chung with the results of NODAL-LAMBDA and the collocation method.

2. The 3D Eigenvalue problem in the nodal method

In this section we first review the roots of coarse mesh nodal methods that can be used to compute the critical and subcritical lambda eigenvalues and eigenvectors of the neutron diffusion equation in 3D with two energy groups and 1.5 groups respectively. The goal is to know the method or methods that give the best results in computing the fundamental and the sub-critical lambda modes, what of them are most robust, and finally what methods are faster. For the case of 1.5 group we only display the results later in section 4.

2.1. The 3D nodal equations in coarse methods with 2 groups

To obtain the 3-D eigenvalues and eigenvectors of the neutron diffusion equation with two energy groups of neutrons one must solve the following equation:

$$L\phi(\vec{r}) = \lambda\, M\phi(\vec{r}) \tag{2.1}$$

Where L and M are the leakage and production operators respectively given by the expressions:

$$L = \begin{bmatrix} -\vec{\nabla}D_1.\vec{\nabla} + (\Sigma_{a1} + \Sigma_{r1}) & 0 \\ \Sigma_{r1} & -\vec{\nabla}D_2.\vec{\nabla} + \Sigma_{a2} \end{bmatrix} \tag{2.2}$$

$$M = \begin{bmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \\ 0 & 0 \end{bmatrix} \tag{2.3}$$

where the symbol λ=1/k denotes the lambda eigenvalue, being $\phi = column(\phi_{1,}\ \phi_2)$ the corresponding eigenvector. D_g, g=1,2, are the difussion coefficients for the fast and thermal groups respectively. The equation (2.1) can be recasted in the form:

$$\vec{\nabla}.\vec{J}_1 + (\Sigma_{a1} + \Sigma_{r1})\phi_1 = \frac{1}{k}(\nu\Sigma_{f1}\phi_1 + \nu\Sigma_{f2}\phi_2) \tag{2.4}$$

$$\vec{\nabla}.\vec{J}_2 + \Sigma_{a2}\phi_2 - \Sigma_{r1}\phi_1 = 0 \tag{2.5}$$

Being $\vec{J}_g = -D_g\vec{\nabla}\,\phi_g$ the neutron current vector for the g-th group. Integrating equations (2.4) and (2.5) over the n-th node volume $V_n = h_x\ h_y\ h_z$, and denoting the common face areas of the surrounding nodes by S_{nm}, where the subindex m ranges over the six neighbours (i-1,j,k), (i+1,j,k), (i,j-1,k), (i,j+1,k), (i,j,k-1) and (i,j,k+1), to node n(i,j,k), it is obtained after division by V_n:

$$\frac{1}{V_n}\sum_{m=1}^{6}\int_{S_{nm}}\vec{J}_1.d\vec{S} + (\Sigma_{a1,n} + \Sigma_{r1,n})\bar{\phi}_1 = \frac{1}{k}(\nu\Sigma_{f1,n}\bar{\phi}_{1,n} + \nu\Sigma_{f2,n}\bar{\phi}_{2,n}) \tag{2.6}$$

$$\frac{1}{V_n}\sum_{m=1}^{6}\int_{S_{nm}}\vec{J}_2.d\vec{S} + \Sigma_{a2,n}\bar{\phi}_{2,n} - \Sigma_{r1,n}\,\bar{\phi}_{1,n} = 0 \tag{2.7}$$

Where $\bar{\phi} = \text{column}(\bar{\phi}_{1,n}\ \bar{\phi}_{2,n})$ denotes the average values of the fast and thermal neutron fluxes over the volume of the n-th node. The cross sections have been assumed constants over the volume of each node.

The next step is to define the leakage terms of the fast and thermal groups through the S_{nm} area face, conecting the nodes n and m, as follows:

$$L_{nm}^1 = \int_{S_{nm}}\vec{J}_1.d\vec{S}\ ,\qquad L_{nm}^2 = \int_{S_{nm}}\vec{J}_2.d\vec{S} \tag{2.8}$$

Thus it is obtained the set of equations:

$$\frac{1}{V_n}\sum_{m=1}^{6} L_{nm}^{1} + (\Sigma_{a1,n} + \Sigma_{r1,n})\bar{\phi}_{1,n} = \frac{1}{k}(\nu\Sigma_{f1,n}\bar{\phi}_{1,n} + \nu\Sigma_{f2,n}\bar{\phi}_{2,n}) \tag{2.9}$$

$$\frac{1}{V_n}\sum_{m=1}^{6} L_{nm}^{2} + \Sigma_{a2,n}\bar{\phi}_{2,n} - \Sigma_{r1,n}\ \bar{\phi}_{1,n} = 0 \tag{2.10}$$

At the internal nodes not located at the boundary, and where Albedo boundary conditions apply, we denote by n(i, j, k), the internal node n defined by the index set (i,j,k) that denotes the node position in the 3D array and by m(i-1,j,k), m(i+1,j,k), m(i,j-1,k), m(i,j+1,k), m(i,j,k-1), and m(i,j,k+1), the adjacent nodes to node n. The leakage term for the g-th group, with g=1,2, in the face connecting the nodes n(i,j,k) and m(i-1,j,k) and with surface area S_{nm} is given according to the Fick Law by:

$$L_{nm}^{g} = D_{g,n}\frac{\phi_{g,n(i,j,k)} - \phi_{g,m(i-1/2,j,k)}}{h_x/2}S_{nm} \tag{2.11}$$

From the continuity of the current at the interface (i-1/2, j, k) between the adjacent nodes n(i, j, k) and m(i-1, j, k), it is obtained that the flux at the interface is a weighted average of the fluxes in the centers of these neighboring nodes, without loss of generality we assume that the nodal lengths are the same in each direction:

$$\phi_{g,(i-1/2,j,k)} = \frac{D_{g,m}\phi_{g,m(i-1j,k)} + D_{g,n}\phi_{g,n(i,j,k)}}{D_{g,m(i-1,j,k)} + D_{g,n(i,j,k)}} \tag{2.12}$$

Direct substitution of eq. (2.12) into (2.11) yields in general that the leakage terms can be written in the form:

$$L_{nm}^{g} == \frac{D_{g,n}D_{g,m}}{D_{g,n} + D_{g,m}}\frac{\phi_{g,n} - \phi_{g,m}}{h_{nm}/2}S_{nm} \tag{2.13}$$

Substitution of eq. (2.13) for the fast and thermal groups into equations (2.9) and (2.10), yield the following set of equations for the internal nodes:

$$\frac{1}{V_n}\sum_{m=1}^{6}\frac{D_{1,n}D_{1,m}}{D_{1,n} + D_{1,m}}\frac{\phi_{1,n} - \phi_{1,m}}{h_{nm}/2}S_{nm} + (\Sigma_{a1,n} + \Sigma_{r1,n})\bar{\phi}_{1,n} = \frac{1}{k}(\nu\Sigma_{f1,n}\bar{\phi}_{1,n} + \nu\Sigma_{f2,n}\bar{\phi}_{2,n}) \tag{2.14}$$

$$\frac{1}{V_n}\sum_{m=1}^{6}\frac{D_{2,n}D_{2,m}}{D_{2,n} + D_{2,m}}\frac{\phi_{2,n} - \phi_{2,m}}{h_{nm}/2}S_{nm} + \Sigma_{a2,n}\bar{\phi}_{2,n} - \Sigma_{r1}\ \bar{\phi}_{1,n} = 0 \tag{2.15}$$

The main inconvenience of these equations is that the average fluxes at node n appear mixed with the fluxes at the node centers. Borresen (1971) developed an expression to relate the average flux in a node with the flux in the center of that node and the fluxes in the centers of the six neighboring nodes in the approximation of 1.5 groups; proceeding in this form he obtained a sparse system of equations for the fluxes in the nodes centers. An extension to two groups can be found in Wulff et al. (1984). Now we give a detailed explanation of the roots of the method used by many codes RAMONA, POLCA, PRESTO (Borresen 1971) and others, and that produces a family of coarse mesh methods.

The way to obtain the expression for the average flux is to perform a Taylor expansion with origin in the node center n(i,j,k) with coordinates (x_i, y_j, z_k) as follows:

$$\phi_n(x,y,z) = \phi(x_i,y_j,z_k) + \sum_{m=1}^{\infty}\frac{1}{m!}\left[\left((x-x_i)\frac{\partial}{\partial x}+(y-y_j)\frac{\partial}{\partial y}+(z-z_k)\frac{\partial}{\partial z}\right)^m \phi\Big|_{(x_i,y_j,z_k)}\right] \tag{2.16}$$

The following step is to retain in equation (2.16) only up to second order terms, and to calculate the average flux at node n by means of the expression:

$$\bar{\phi}_{g,n} = \frac{1}{V_n}\int_{V_n}\phi_{g,n}(x,y,z)\,d^3r \tag{2.17}$$

Then, performing the change of variables $x - x_i = \frac{h_x}{2}\xi_x$, $y - y_j = \frac{h_y}{2}\xi_y$, $z - z_k = \frac{h_z}{2}\xi_z$, in equation (2.17), the expression that gives the average flux at node n reduces to:

$$\bar{\phi}_{g,n} = \frac{1}{8}\int_{-1}^{1}d\xi_x\int_{-1}^{1}d\xi_y\int_{-1}^{1}d\xi_z\phi_{g,n}(\xi_x,\xi_y,\xi_z) \tag{2.18}$$

Using the Taylor expansion (2.16) up to second order in (2.18) yields:

$$\bar{\phi}_{g,n} = \phi_{g,n} + \frac{1}{8}\sum_{m=1}^{2}\frac{1}{m!}\int_{-1}^{1}d\xi_x\int_{-1}^{1}d\xi_y\int_{-1}^{1}d\xi_z\left[\xi_x\frac{\partial}{\partial\xi_x}+\xi_y\frac{\partial}{\partial\xi_y}+\xi_z\frac{\partial}{\partial\xi_z}\right]^m\phi \tag{2.19}$$

Where $\phi_{g,n} = \phi_g(x_i,y_j,z_k)$ denotes the g-group flux at the center of node n. Next we take into account the fact that all the terms in equation (2.19) containing first order derivatives in any variable vanish as can be easily checked. Therefore only remain the terms containing second order derivatives and equation (2.19) reduces to:

$$\bar{\phi}_{g,n} = \phi_{g,n} + \frac{1}{8}\sum_{u=1}^{3}\frac{1}{2}\left(\frac{\partial^2\phi_g}{\partial\xi_u^2}\right)_n\int_{-1}^{1}d\xi_u\int_{-1}^{1}d\xi_{u'}\int_{-1}^{1}d\xi_{u''}\xi_u^2 = \phi_{g,n} + \frac{1}{6}\sum_{u=1}^{3}\left(\frac{\partial^2\phi_g}{\partial\xi_u^2}\right)_n \tag{2.20}$$

Returning to the original coordinates and using the symbol x_u with u=1, 2, 3 to denote the three orthogonal axes x, y, z, it is obtained:

$$\bar{\phi}_{g,n} = \phi_{g,n} + \frac{1}{24}\sum_{u=1}^{3}h_u^2\left(\frac{\partial^2\phi_g}{\partial x_u^2}\right)_n \tag{2.21}$$

The next step is to obtain an expression for the second derivatives of the neutron flux, that appear in equation (2.21), and which are evaluated at the node center, in order to understand the origin of the expressions used by RAMONA and PRESTO codes, and that were developed originally by Borressen (1971). We assume in a first step that the mesh is cubic and then in a second step the result will be generalized to non-cubic meshes.

Assuming that the mesh is cubic is equivalent to say that h_x=h_y=h_z, in this case the Borressen factor R defined as $R = h_x^2 / h_z^2$ is equal to 1. Then the second derivatives of the flux at the node center evaluated in any direction x, y or z, denoted in general by u ca be expressed in a difference scheme in terms of the values of the flux at the node center and the faces by means of the expression:

$$\frac{\partial^2 \phi_{g,n}}{\partial x_u^2} = \frac{w_{g,0}\phi_{g,n_u+1/2} - 2w_{g,1}\phi_{g,n} + w_{g,2}\phi_{g,n_u-1/2}}{(h_u/2)^2} \tag{2.22}$$

Where $\phi_{g,n}$ denotes the neutron flux at the node center $n(i,j,k)$, while $\phi_{g,n_u+1/2}$ denotes the neutron flux at the face m located advancing $h_u/2$ length units in the u direction given by the vector $\vec{n}_u$. In the classical central finite difference scheme the weighting factors $w_{g,0}, w_{g,1}, w_{g,2}$ are equal 1. Substitution of expression (2.22) into equation (2.21), and considering the three weighting factors to be equal i.e. $w_{g,0} = w_{g,1} = w_{g,2}$ but different from unity for a cubic coarse mesh yields the following result for the neutron average flux in the n-th node in terms of the neutron flux at the center and the six neighboring faces:

$$\bar{\phi}_{g,n} = (1 - w_{g,0})\phi_{g,n} + \frac{w_{g,0}}{6}\sum_{m=1}^{6}\phi_{g,nm} \tag{2.23}$$

Denoting the coefficient at center of the n-th node for the g-th group by a_g, to be determined later, we can write:

$$1 - w_{g,0} = a_g \Rightarrow w_{g,0} = 1 - a_g \tag{2.24}$$

Proceeding in this way one arrives to the expression:

$$\bar{\phi}_{g,n} = a_g\,\phi_{g,n} + \frac{1-a_g}{6}\sum_{m=1}^{6}\phi_{g,nm} \tag{2.25}$$

We notice that adding-up all the coefficients of the neutron fluxes at the center and the six faces of equation (2.25) yields the unity. Now if in Borresen expressions (Wulff 1984, Borresen 1971), one sets R=1, it is obtained the expression (2.25) for the average flux in a given node n, when expressed in terms of the neutron flux at the node center $\phi_{g,n}$ and at the surrounding faces $\phi_{g,nm}$.

To obtain a general expression when the mesh is not cubic, due to the fact that the mesh width in the horizontal directions x and y is in general different that the mesh width in the vertical direction z, i.e. in general we have that $h_x = h_y \neq h_z$, that means that the Borresen factor $R = h_x^2/h_z^2$ is now different from unity. Then, we assume that the weighting factors in equation (2.22) depend in general of the direction u (x, y or z) being considered. In this case, we assume that the second derivative in the u direction can be written as follows:

$$\frac{\partial^2 \phi_{g,n}}{\partial x_u^2} = \frac{w_{g,u}\phi_{g,n_u+1/2} - 2w_{g,u}\phi_{g,n} + w_{g,u}\phi_{g,n_u-1/2}}{(h_u/2)^2} \tag{2.26}$$

As mentioned previously, the flux weigh coefficients $w_{g,u}$ depend on the direction being considered for a non-cubical mesh. It is logic to consider that in the directions where the faces are closer to the node center these weigh factors should be different than in the directions where the faces are far away from the node center. Therefore we use the notation:

$$\begin{aligned} w_{g,nm} &= w_{g,1} = w_{g,2} \; for\ faces\ nm\ orthogonal \quad to\ x\ or\ y \\ w_{g,nm} &= w_{g,3}, for\ faces\ nm\ orthogonal \quad to\ z \end{aligned} \tag{2.27}$$

Therefore inserting (2.26), into equation (2.21), on account of (2.27) yields:

$$\overline{\phi}_{g,n} = \left(1 - \frac{\sum\limits_{m=1}^{6} w_{g,nm}}{6} \right) \phi_{g,n} + \frac{1}{6} \sum_{m=1}^{6} w_{g,nm} \phi_{g,nm} \tag{2.28}$$

Next it is assumed that the relative weight factors on each of the six interfaces are proportional to R_{nm} i.e. $w_{g,nm} \propto R_{nm}$ where $R_{nm} = h_x^2 / h_z^2$, for the faces m(i,j,k-1/2) and m(i,j,k+1/2), and $R_{nm} = 1$, for the rest of faces m(i-1/2,j,k), m(i+1/2,j,k), m(i,j-1/2,k), and m(i,j+1/2,k).

For the flux coefficient at the center of the node we write instead of a_g as in equations (2.24) and (2.25):

$$\left(1 - \frac{\sum\limits_{m=1}^{6} w_{g,nm}}{6} \right) = a_g f_g \tag{2.29}$$

Where f_g is a normalization factor to be determined later.

The flux coefficients at the four faces that are orthogonal to the directions x and y, are assumed to be identical by symmetry and its sum is given, instead of $4(1-a_g)/6$ as in a cubic lattice, by:

$$\frac{\sum\limits_{m=1}^{4} w_{g,nm}}{6} = \frac{4(1-a_g)}{6} f_g \tag{2.30}$$

For the flux coefficients at the two faces orthogonal to the z- direction, they are assumed identical by symmetry, and we may write on account of the fact that these weight factors are proportional to $R_{nm} = h_x^2 / h_z^2 = R$ for these faces:

$$\frac{\sum\limits_{m=1}^{2} w_{0g,nm}}{6} = \frac{2R(1-a_g)}{6} f_g \tag{2.31}$$

Where f_g is a normalizing factor of the flux coefficients in equation (2.28) that is obtained from the condition that the summation of all the coefficients in the center and the faces is equal to 1. From this condition it is immediately deduced the following expression for f_g :

$$f_g = \frac{3}{3a_g + 2(1-a_g) + R(1-a_g)} = \frac{3}{3a_g + (R+2)(1-a_g)} \qquad (2.32)$$

The it follows fro eq (2.32) that if R=1 then $f_g = 1$.

Therefore equation (2.28) can be written as follows:

$$\bar{\phi}_{g,n} = a_g f_g \, \phi_{g,n} + \frac{(1-a_g)}{6} f_g \sum_{m=1}^{4} \phi_{g,nm} + \frac{(1-a_g)}{6} f_g \sum_{m=5}^{6} R \; \phi_{g,nm} \qquad (2.33)$$

We notice that equation (2.33) is identical to Borresen equation for the average nodal flux (Borresen 1971). We remind that Borresen expression is usually written (Wulff , Dong Chung et al 1981) in the form:

$$\bar{\phi}_{1,n} = B_{1,n}\phi_{1,n} + 2C_{1,n} \sum_{m=1}^{6} R_{nm} \, \phi_{1,nm} \qquad (2.34)$$

$$\bar{\phi}_{2,n} = B_{2,n}\phi_{1,n} + 2\, C_{2,n} \sum_{m=1}^{6} R_{nm} \, \phi_{2nm} \qquad (2.35)$$

Where the group dependent constants B1, C1, B2, C2 are given by the expressions:

$$B_g = \frac{3a_g}{3a_g + (1-a_g)(R_{zz}+2)}, \quad g = 1,2 \qquad (2.36)$$

$$C_g = \frac{1-a_g}{4} \frac{1}{3a_g + (1-a_g)(R_{zz}+2)}, \quad g = 1,2 \qquad (2.37)$$

The typical values of the constants $a_g$for g=1, 2 for typical BWR reactors are $a_1 = 0.3$ and $a_2 = 0.7$. In equations (2.34) and (2.35) $\phi_{g,nm}$ denotes the neutron flux for group g at the face nm of the node n.

2.2 Matrix Equations in Coarse-Mesh Nodal Classical Methods

To obtain the lambda eigenvalues and eigenvectors we need to express the eigenvalue equation in matrix form as in Verdú et al (1994), or Doring et al. (1993).We will perform this procedure in two steps writing first the matrix elements for the interior nodes and then on account of the boundary conditions we will write the matrix elements for the boundary nodes. Note that the expression for the matrix elements are not the same that in Borressen (1971), and RAMONA code (Wulff et. al 1984) because some approximations performed by these authors are not considere here:

2.2.1 Algebraic matrix equations and matrix elements for the internal nodes not located at the boundaries

First we express the fluxes at the faces $\phi_{g,nm}$in expressions (2.34) and (2.35) in terms of the fluxes at the neighbouring nodes to each face using expression (2.12). Proceeding in this way, it is obtained for the internal nodes, not touching the boundary, the following result for the fast and thermal average fluxes:

$$\bar{\phi}_{1,n} = (B_{1,n} + C_{1,n}\, r_{1,n})\, \phi_{1,n} + C_{1,n} \sum_{m=1}^{6} R_{nm} \frac{2D_{1m}}{D_{1,n} + D_{1,m}} \phi_{1,m} \tag{2.38}$$

$$\bar{\phi}_{2,n} = (B_{2,n} + C_{2,n}\, r_{2,n})\, \phi_{2,n} + C_{2,n} \sum_{m=1}^{6} R_{nm} \frac{2D_{2m}}{D_{2,n} + D_{2,m}} \phi_{2,m} \tag{2.39}$$

Where the following coefficients, using standard notation, have been defined:

$$r_{1,n} = \sum_{m=1}^{6} R_{nm} \frac{2D_{1,n}}{D_{1,n} + D_{1,m}} \tag{2.40}$$

$$r_{2,n} = \sum_{m=1}^{6} R_{nm} \frac{2D_{2,n}}{D_{2,n} + D_{2,m}} \tag{2.41}$$

Next we substitute the expressions (2.38) and (2.39) for the average fluxes of the fast and thermal fluxes in the nodal equations (2.14) and (2.15) , and then the resulting equation for the fast group at node n is multiplied by $h_x^2 / D_{1,n}$, and the resulting equation for the thermal group is multiplied by $h_x^2 / D_{2,n}$, the final result is a set of algebraic equations, that writen in matrix form can be expressed as follows:

$$L_{11}\phi_1 = \frac{1}{k}(M_{11}\phi_1 + M_{12}\phi_2)$$

$$\tag{2.42}$$

$$-L_{21}\phi_1 + L_{22}\phi_2 = 0$$

Where for the internal nodes the matrix elements of equations (2.42) are given on account of (2.14),(2.15), (2.38) and (2.39), as can be easily checked, by the following set of expressions:

i) Diagonal elements of the fast group leakage matrix L11 denoted by $\left(L_{11}^{Diag}\right)_{n,n}$:

$$(L_{11}^{Diag})_{n,n} = \left(P_{1,n} + \frac{h_x^2}{D_{1,n}} (\Sigma_{a1} + \Sigma_{r1})_n (B_{1,n} + C_{1,n} r_{1,n}) \right) \tag{2.43}$$

ii) Non diagonal elements of the matrix L11 denoted by $\left(L_{11}^{nD}\right)_{n,m}$:

$$(L_{11}^{nD})_{n,m} = \left(\frac{h_x^2}{D_{1,n}} (\Sigma_{a1} + \Sigma_{r1})_n C_{1,n} - 1 \right) \frac{2R_{nm} D_{1,m}}{D_{1,n} + D_{1,m}} \tag{2.44}$$

iii) Diagonal elements of the neutron production matrix M11 denoted by $\left(M_{11}^{Diag}\right)_{n,n}$:

$$(M_{11}^{Diag})_{n,n} = \left(\frac{h_x^2}{D_{1,n}} (\nu_1 \Sigma_{f1})_n (B_{1,n} + C_{1,n} r_{1,n}) \right) \tag{2.45}$$

iv) Non diagonal elements of neutron production matrix M11 denoted by $\left(M_{11}^{nD}\right)_{n,m}$:

$$(M_{11}^{nD})_{n,m}=\left(\frac{h_x^2}{D_{1,n}}(\nu_1\Sigma_{f1})_n C_{1,n}\right)\frac{2R_{nm}D_{1m}}{D_{1n}+D_{1m}} \tag{2.46}$$

v) Diagonal elements of the neutron production matrix M_{12} denoted by $\left(M_{12}^{Diag}\right)_{n,n}$:

$$(M_{12}^{Diag})_{n,n}=\left(\frac{h_x^2}{D_{1,n}}(\nu_2\Sigma_{f2})_n(B_{2,n}+C_{2,n}r_{2,n})\right) \tag{2.47}$$

vi) Non diagonal elements of the neutron production matrix M_{12} denoted by $\left(M_{12}^{nD}\right)_{n,m}$:

$$(M_{12}^{nD})_{n,m}=\left(\frac{h_x^2}{D_{1,n}}(\nu_2\Sigma_{f2})_n C_{2,n}\right)\frac{2R_{nm}D_{2m}}{D_{2n}+D_{2m}} \tag{2.48}$$

vii) Diagonal elements of the matrix L_{21} denoted by $\left(L_{21}^{Diag}\right)_{n,n}$:

$$(L_{21}^{Diag})_{n,n}=\left(\frac{h_x^2}{D_{2,n}}(\Sigma_{r1})_n(B_{1,n}+C_{1,n}r_{1,n})\right) \tag{2.49}$$

viii) Non diagonal elements of the matrix L_{21} denoted by $\left(L_{21}^{nD}\right)_{n,m}$:

$$(L_{21}^{nD})_{n,m}=\left(\frac{h_x^2}{D_{2,n}}(\Sigma_{r1})_n C_{1,n}\right)\frac{2R_{nm}D_{1m}}{D_{1n}+D_{1m}} \tag{2.50}$$

ix) Diagonal elements of the leakage matrix for the thermal group L_{22} denoted by $\left(L_{22}^{Diag}\right)_{n,n}$:

$$(L_{22}^{Diag})_{n,n}=\left(P_{2,n}+\frac{h_x^2}{D_{2,n}}(\Sigma_{a2})_n(B_{2,n}+C_{2,n}r_{2,n})\right) \tag{2.51}$$

x) Non diagonal elements of the thermal group leakage matrix L_{22} denoted by $\left(L_{22}^{nD}\right)_{n,m}$:

$$(L_{22}^{nD})_{n,m}=\left(\frac{h_x^2}{D_{2,n}}(\Sigma_{a2})_n C_{2,n}\text{ - }1\right)\frac{2R_{nm}D_{2m}}{D_{2n}+D_{2m}} \tag{2.52}$$

In the previous matrices appear the constants $P_{1,n}$ and $P_{2,n}$, that are constants defined by the following expressions:

$$P_{1,n}=\sum_{m=1}^{6}R_{nm}\frac{2D_{1,m}}{D_{1,m}+D_{1,n}} \tag{2.53}$$

$$P_{2,n}=\sum_{m=1}^{6}R_{nm}\frac{2D_{2,m}}{D_{2,m}+D_{2,n}} \tag{2.54}$$

The next step is to obtain the matrix elements for the nodes that are touching the reactor boundary. These expressions are not the same ones used by Ramona code (Wulff 1984) and Borresen (1971) because some aproximations has not bee considered here.

2.2.2. ALBEDO's boundary conditions in the NODAL-LAMBDA code.

First we will obtain the expressions of the Albedo matrix elements from the reflector properties. These expressions were first obtained by Kalambokas and Henry (1974), an alternative way to deduce the expression for the Albedos matrix elements is to use the Laplace transform method (Petersen et al 2010), we give here a outline of the method.

The albedo matrix $[\alpha]$ is defined as the matrix that relates the boundary current with the boundary flux:

$$[J_B]=\begin{bmatrix} J_{B1} \\ J_{B2} \end{bmatrix}=\begin{bmatrix} \alpha_{11} & 0 \\ \alpha_{21} & \alpha_{22} \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}_B \tag{2.55}$$

where $[J_B]$ is the neutron current vector at the boundary, while $[\phi_B]$ is the neutron flux vector at the boundary. The inverse of the Albedo matrix $[\alpha]^{-1}$ relates the currents at the boundary with the fluxes at the boundary and is usually denoted by:

$$A=\begin{bmatrix} \alpha_{11} & 0 \\ \alpha_{21} & \alpha_{22} \end{bmatrix}^{-1}=\begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix} \tag{2.56}$$

Let us assume that we have a 1D reactor and that at the left of the coordinate origin we have fuel denoted by (F), and at the right hand side we have the reflector denoted by (r). Let us assume that the reflector thickness is Δ then the 1D diffusion equations, in two groups in the reflector region, are given by:

$$-D_{1,r}\frac{d^2\phi_{1,r}}{dx^2}+(\Sigma_{a1}+\Sigma_{r12})_r\phi_{1,r}(x)=0 \tag{2.57}$$

$$-D_{2,r}\frac{d^2\phi_{2,r}}{dx^2}+(\Sigma_{a2})_r\phi_{2,r}(x)=\Sigma_{r12,r}\phi_{1,r}(x) \tag{2.58}$$

The neutron current in the reflector is given by the Fick's law:

$$\mathrm{J}_{g,r}=-\mathrm{D}_{g,r}\frac{\mathrm{d}\phi_{g,r}(\mathrm{x})}{\mathrm{dx}} \tag{2.59}$$

Performing the Laplace transform of equations (2.57) and (2.58), on account of (2.59) it is obtained:

$$\tilde{\phi}_1(p)=\frac{p\phi_{1,r}(0)-\frac{J_{1,r}(0)}{D_{1,r}}}{p^2-\frac{\Sigma_{t,1}}{D_{1,r}}}=\frac{p\phi_{1,r}(0)-\frac{J_{1,r}(0)}{D_{1,r}}}{p^2-\frac{1}{L_{1,r}^2}} \tag{2.60}$$

$$\tilde{\phi}_2(p)=\frac{\Sigma_{r12,r}\tilde{\phi}_1(p)}{-D_{2,r}p^2+\Sigma_{a2,r}}+\frac{p\,\phi_{2,r}(0)-\frac{J_{2,r}(0)}{D_{2,r}}}{p^2-\frac{1}{L_{2,r}^2}} \tag{2.61}$$

where $\tilde{\phi}_g(p)$ is the Laplace transform of $\phi_g(x)$, and $L_{g,r}=\sqrt{\frac{D_{g,r}}{\Sigma_{t,g,r}}}$ is the diffusion length for the g-th neutron group, notice that in the reflector $\Sigma_{t,1,r}=\Sigma_{a,1,r}+\Sigma_{r12,r}$ and $\Sigma_{t,2,r}=\Sigma_{a,2,r}$. Expanding equation (2.60) in simple fractions and then performing the inverse Laplace transform of the resulting equation yields after some simplifications the following expression for the fast neutron flux in the reflector:

$$\phi_{1,r}(x)=\phi_1(0)\cosh\left(\frac{x}{L_{1,r}}\right)-\frac{L_{1,r}}{D_{1,r}}J_{1,r}(0)\ \sinh\left(\frac{x}{L_{1,r}}\right) \tag{2.62}$$

On account of the boundary condition that the neutron flux vanish at the reflector boundary i.e. $\phi_g(\Delta)=0$, it is obtained from (2.62) the following expression for the albedo matrix element α_{11} :

$$\alpha_{11}=\frac{1}{a_{11}}=\frac{J_1(0)}{\phi_1(0)}=\frac{D_{1,r}}{L_{1,r}}\frac{1}{tgh\left(\frac{\Delta}{L_{1,r}}\right)} \tag{2.63}$$

The procedure to obtain the rest of the Albedo matrix elements is lengthy, and here we only outline the main steps. Fist from the equation (2.61), it is possible to express the Laplace transform $\tilde{\phi}_2(p)$ of the thermal flux as follows:

$$\tilde{\phi}_2(p)=\frac{\tilde{\phi}_1(p)}{q(p)}+\frac{A_2}{p-\frac{1}{L_{2,r}}}+\frac{B_2}{p+\frac{1}{L_{2,r}}} \tag{2.64}$$

where q(p), A_2, and B_2 are given by the expressions:

$$q(p)=-\frac{D_{2,r}}{\Sigma_{r12}}(p^2-\frac{1}{L_2^2}) \tag{2.65}$$

$$A_2=\frac{1}{2}(\phi_2(0)-\frac{L_{2,r}}{D_{2,r}}J_2(0)) \qquad B_2=\frac{1}{2}(\phi_2(0)+\frac{L_{2,r}}{D_{2,r}}J_2(0)) \tag{2.66}$$

Performing on equation (2.64) the inverse Laplace transforms operation, and after some simple manipulations it is obtained:

$$\phi_{2,r}(x)=L^{-1}\left\{\frac{\tilde{\phi}_1(p)}{q(p)}\right\}+\phi_2(0)\cosh(\frac{x}{L_{2,r}})-\frac{L_{2,r}}{D_{2,r}}J_2(0)\sinh(\frac{x}{L_{2,r}}) \tag{2.67}$$

The next step is to evaluate the inverse Laplace transform of 1/q(p), that yields:

$$L^{-1}\left\{\frac{1}{q(p)}\right\} = -\frac{\Sigma_{r12}}{D_{2,r}}\sinh(\frac{x}{L_{2,r}}) \tag{2.68}$$

Then we perform the inverse Laplace transform of the first term of equation (2.67), using the convolution theorem, this calculation gives:

$$L^{-1}\left\{\frac{\tilde{\phi}_1(p)}{q(p)}\right\} = -\frac{\Sigma_{r12}L_{2,r}}{D_{2,r}}\int_0^x \phi_1(x')\sinh(\frac{x-x'}{L_{2,r}})\,dx' \tag{2.69}$$

Finally it is obtained the following expression for the thermal neutron flux $\phi_{2,r}(x)$ in the reflector:

$$\phi_{2,r}(x) = -\frac{\Sigma_{r12}L_{2,r}}{D_{2,r}}\int_0^x \phi_1(x')\sinh(\frac{x-x'}{L_{2,r}})\,dx' + \phi_2(0)\cosh(\frac{x}{L_{2,r}}) - \frac{L_{2,r}}{D_{2,r}}J_2(0)\sinh(\frac{x}{L_{2,r}}) \tag{2.70}$$

Next we take into account expression (2.62) for $\phi_{1,r}(x)$ and the boundary conditions at the reflector i.e. $\phi_{2,r}(\Delta) = 0$:

$$\phi_{2,r}(\Delta) = 0 = -\frac{\Sigma_{r12}L_{2,r}}{D_{2,r}}\int_0^\Delta \phi_1(x')\sinh(\frac{\Delta-x'}{L_{2,r}})\,dx' + \phi_2(0)\cosh(\frac{\Delta}{L_{2,r}}) - \frac{L_{2,r}}{D_{2,r}}J_2(0)\sinh(\frac{\Delta}{L_{2,r}}) \tag{2.71}$$

Substituting eq. (2.62) into eq. (2.71) yields:

$$-\frac{\Sigma_{r12}L_{2,r}}{D_{2,r}}\int_0^\Delta\left\{\phi_1(0)\cosh\left(\frac{x'}{L_{1,r}}\right) - \frac{L_{1,r}}{D_{1,r}}J_{1,r}(0)\ \sinh\left(\frac{x'}{L_{1,r}}\right)\right\}\sinh(\frac{\Delta-x'}{L_{2,r}})\,dx' + \phi_2(0)\cosh(\frac{\Delta}{L_{2,r}}) - $$
$$-\frac{L_{2,r}}{D_{2,r}}J_2(0)\sinh(\frac{\Delta}{L_{2,r}}) = 0 \tag{2.72}$$

Next, operating in equation (2.72), on account of (2.56) and (2.63), yields:

$$\phi_2(0) = a_{22}J_2(0) + a_{21}J_1(0) \tag{2.73}$$

Where the coefficients a_{21} and a_{22} are given by the expressions:

$$a_{22} = \frac{L_{2,r}}{D_{2,r}}tgh\frac{\Delta}{L_{2,r}} \tag{2.74}$$

$$a_{21} = \left(\frac{\Sigma_{r12}}{\Sigma_{a2}D_1 - \Sigma_{t1}D_2}\right)_r \left(D_{1,r}a_{11} - D_{2,r}a_{22}\right) \tag{2.75}$$

Next we note that the matrix A that gives the flux vector from the current vector at the boundary is the inverse one of the albedo matrix $[\alpha]$. So we obtain that $a_{11} = 1/\alpha_{11}$.

Then, we use the relationship between the elements of the albedo matrix and the A matrix and it is obtained:

$$\alpha_{11} = \frac{1}{a_{11}}$$
$$\alpha_{22} = \frac{1}{a_{22}} \qquad (2.76)$$
$$\alpha_{21} = -\frac{a_{21}}{a_{11}a_{22}} = -\frac{\Sigma_{r12}}{\Sigma_{a2}D_{1,r} - \Sigma_{t1}D_{2,r}}\{D_{1,r}\alpha_{22} - D_{2,r}\alpha_{11}\}$$

The final expression for the Albedo coefficient α_{21} is:

$$\alpha_{21} = -\frac{\Sigma_{r12}D_{1,r}D_{2,r}}{\Sigma_{a2}D_{1,r} - \Sigma_{t1}D_{2,r}}\left\{\frac{1}{L_{2,r}tgh\frac{\Delta}{L_{2,r}}} - \frac{1}{L_{1,r}tgh\frac{\Delta}{L_{1,r}}}\right\} \qquad (2.77)$$

Therefore the program NODAL-LAMBDA V2 computes for each face belonging to the boundary the ALBEDO's coefficients $\alpha_{11}, \alpha_{12}, \alpha_{22}$. These ALBEDO's coefficients depend on the reflector properties and the reflector widths including the extrapolation length. Obviously the reflector properties change from one boundary node to another one because the reflector cross sections are different.

2.2.3 Matrix elements for the nodes located at the boundaries

In this section we obtain the expressions for the matrix elements when the node is at the boundary. We must notice that a node located at the boundary can have one, two or three faces touching the boundary. The reflector has been eliminated and substituted by ALBEDO boundary conditions in NODAL-LAMBDA as explained in the previous section.

If $[\phi_I]$denotes the neutron flux $column[\phi_{1I}, \phi_{2I}]$ at the center of a node in which one of its enclosing surfaces is at the boundary in the direction $\vec{n}_u$ from the node center, and we assume that at this surface we have an ALBEDO boundary condition and this boundary surface is at a distance $h_u/2$from the center of the node. Then we can apply the Fick law to calculate the current at this boundary surface denoted by the sub-index B. Therefore we can write:

$$[J_B] = -\{[D][\vec{n}_B.\vec{\nabla}\phi]_B\} \approx [D_I]\frac{[\phi_I]-[\phi_B]}{h_u/2} \qquad (2.78)$$

Where [D]is the matrix of diffusion coefficients, [J$_B$] denotes the current vector at the boundary surface and $[\phi_I]$ the neutron flux column vector at the center of the node:

$$[D] = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, [J_B] = \begin{bmatrix} J_{B,1} \\ J_{B,2} \end{bmatrix}, [\phi_I] = \begin{bmatrix} \phi_{I,1} \\ \phi_{I,2} \end{bmatrix} \qquad (2.79)$$

h_u is h_x , h_y or h_z if the boundary face is orthogonal to the x, y or z axes respectively. Also we notice that the distance from the center of node I to the boundary face in the $\vec{n}_u$ direction is $h_u/2$. Finally $\vec{n}_B$ is the unit vector of the boundary surface.

Because we have computed in the previous section the ALBEDO's matrix elements as we have already explained, then we can relate the currents with the fluxes at the boundary so we can write equation (2.78) as follows:

$$[J_B] = \{[\alpha][\phi]\}_B \approx [D_I]\frac{[\phi_I]-[\phi_B]}{h_u/2} \qquad (2.80)$$

To express $[\phi_B]$in terms of $[\phi_I]$, we solve the matrix equation (2.80) as follows, first we pre-multiply both terms of equation (2.80) by the inverse of the diffusion coefficient matrix multiplied by $h_u/2$ i.e.$\frac{h_u}{2}[D_I]^{-1}$, and we isolate the term containing $[\phi_I]$in the right hand side, this calculation yields:

$$[\phi_B]+\frac{h_u}{2}[D_I]^{-1}\{[\alpha][\phi]\}_B = [\phi_I] \qquad (2.81)$$

From equation (2.81) and after some simple algebraic manipulations it is obtained a relationship between the neutron flux at the boundary and the neutron flux at the center

$$[\phi_B] = \left([I]+\frac{h_u}{2}[D_I]^{-1}[\alpha]\right)^{-1}[\phi_I] \qquad (2.82)$$

Where the symbol [I] denotes the unit matrix. Next performing the set of operations necessary to compute the elements of the inverse matrix, these calculations yield the following result:

$$\left([I]+\frac{h_u}{2}[D_I]^{-1}[\alpha]\right)^{-1} = \begin{bmatrix} \dfrac{1}{(1+\frac{h_u}{2}\frac{\alpha_{11}}{D_1})} & 0 \\ -\dfrac{\frac{h_u}{2}\frac{\alpha_{21}}{D_2}}{(1+\frac{h_u}{2}\frac{\alpha_{11}}{D_1})(1+\frac{h_u}{2}\frac{\alpha_{22}}{D_2})} & \dfrac{1}{(1+\frac{h_u}{2}\frac{\alpha_{22}}{D_2})} \end{bmatrix} \qquad (2.83)$$

Therefore using (2.82) on account of (2.70), it is possible to express the fast and thermal neutron fluxes at the boundary surface in terms of the fast and thermal neutron fluxes at the node center adjacent to the boundary, so we write:

$$\phi_{B1} = \frac{1}{(1+\frac{h_u}{2}\frac{\alpha_{11}}{D_1})}\phi_{I1} \qquad (2.84)$$

$$\phi_{B2} = -\frac{\frac{h_u}{2}\frac{\alpha_{21}}{D_2}}{(1+\frac{h_u}{2}\frac{\alpha_{11}}{D_1})(1+\frac{h_u}{2}\frac{\alpha_{22}}{D_2})}\phi_{I1} + \frac{1}{(1+\frac{h_u}{2}\frac{\alpha_{22}}{D_2})}\phi_{I2} \qquad (2.85)$$

Equations (2.84) and (2.85) were obtained by Wulff et al (1984). The importance of these equations relies in that for the nodes touching the reflector, it is possible to express the average fluxes at the boundary faces in terms of the nodal fluxes at the centers of these nodes. Let us deduce now the expressions for the different matrix elements when one or more nodal faces touch the reflector. We will display only the expressions for the case that only one face touches the boundary, these expression can be generalized easily to te case that more that one face touches the boundary and are not displayed here for the sake of simplicity, but are programmed in the NODAL-LAMBDA code.

Now denoting by the subindex m=r the face of the node n that touch the reflector. Then, we study first the case that only one face touch the reflector, aulthough we can have up to 3 faces touching the reflector, then equations (2.38) and (2.39) for the fast and thermal average fluxes at the node n when only one face is in contact with the reflector, are modified on account of equations (2.84) and (2.85) as follows:

$$\bar{\phi}_{1,n} = \left(B_{1,n} + 2C_{1,n}\frac{R_{nr}}{[1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}}]} + 2C_{1,n}\sum_{m\neq r} R_{nm}\frac{D_{1n}}{D_{1n}+D_{1m}} \right)\phi_{1,n} + C_{1,n}\sum_{m\neq r} R_{nm}\frac{2D_{1,m}}{D_{1,n}+D_{1,m}}\phi_{1m}$$

(2.86)

$$\bar{\phi}_{2,n} = \left(B_{2,n} + 2C_{2,n}\frac{R_{nr}}{[1+0.5h_{nr}(\alpha_{22}/D_{2,n})]} + 2C_{2n}\sum_{m\neq r} R_{nm}\frac{D_{2n}}{D_{2n}+D_{2m}} \right)\phi_{2,n}$$

$$-2C_{2,n}\frac{R_{nr}\frac{h_{nr}}{2}\frac{\alpha_{21}}{D_{2,n}}}{[1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}}][1+\frac{h_{nr}}{2}\frac{\alpha_{22}}{D_{2,n}}]}\phi_{1,n} + C_{2,n}\sum_{m\neq r} R_{nm}\frac{2D_{2,m}}{D_{2,n}+D_{2,m}}\phi_{2m} \qquad (2.87)$$

In addition, the expressions for the leakage terms at the faces r of a node n in contact with the reflector are modified, because in these faces the ALBEDO boundary condition must be used. For instance, the modification in the matrix terms due to the neutron thermal current at the boundary yields for these terms

$\frac{h_x^2}{D_{2,n}}(J_{B,nr}^{(2)}\frac{A_{nr}}{V_n}) = \left(\frac{h_x}{D_{2,n}}J_{B,nr}^{(2)}\right)$, when the currrent is in the x or y direction.

$\frac{h_x^2}{D_{2,n}}(J_{B,nr}^{(2)}\frac{A_{nr}}{V_n}) = \left(\frac{h_x^2}{D_{2,n}h_z}J_{B,nr}^{(2)}\right)$, when the current is in the z direction.

Both terms can we writen in compact notation as follows:

$$\frac{h_x^2}{D_{2,n}}(J^{(2)}_{B,nr}\frac{A_{nr}}{V_n}) = \left(\frac{h_{nr}R_{nr}}{D_{2,n}}J^{(2)}_{B,nr}\right) \tag{2.88}$$

On account of the expression for the current in the boundaries in terms of the average fluxes in the faces using the ALBEDOS yields:

$$\frac{h_x^2}{D_{2,n}}(J^{(2)}_{B,nr}\frac{A_{nr}}{V_n}) = \frac{h_{nr}R_{nr}\alpha_{21}}{D_{2,n}(1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}})}\left(1-\frac{\frac{h_{nr}}{2}\alpha_{22}}{D_{2,n}(1+\frac{h_{nr}}{2}\frac{\alpha_{22}}{D_{2,n}})}\right)\phi_{1,n} + \frac{h_{nr}R_{nr}\alpha_{22}}{D_{2,n}(1+\frac{h_{nr}}{2}\frac{\alpha_{22}}{D_{2,n}})}\phi_{2,n}$$

(2.89)

Then the expresssion for the matrix elements when one face is in contact with the reflector are:

i) Diagonal elements of the matrix L11 denoted by L_{11}^{Diag}, when only one face touches the reflector:

$$(L_{11}^{Diag})_{n,n} = P_{1,n}(r) + \frac{h_x^2}{D_{1,n}}(\Sigma_{a1}+\Sigma_{r1})_n(B_{1,n}+C_{1,n}r_{1,n}(r)) \tag{2.90}$$

Where for the case of nodes that are in contact with the reflector (r), the coefficients P_{1n} and r_{1n}, denoted now by $P_{1n}(r)$ and $r_{1n}(r)$, are given by the following expressions:

$$P_{1,n}(r) = \frac{h_{nr}}{D_{1n}}\frac{\alpha_{11}R_{nr}}{(1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1n}})} + \sum_{m\neq r}R_{nm}\frac{2D_{1,m}}{D_{1,m}+D_{1,n}} \tag{2.91}$$

$$r_{1,n}(r) = \frac{2R_{nr}}{(1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}})} + \sum_{m\neq r}R_{nm}\frac{2D_{1,n}}{D_{1,n}+D_{1,m}} \tag{2.92}$$

The variable h_{nr} depends on the face r, if this face is orthogonal to the x or y axis then $h_{nr}=h_x$, when this face is orthogonal to the z-axis then $h_{nr}=h_z$.

ii) Non-diagonal elements of the matrix L11, denoted by $\left(L_{11}^{nD}\right)_{n,m}$, when one face r touches the reflector, only appear in this matrix the elements nm with $m \neq r$, this elements are given by the expression:

$$(L_{11}^{nD})_{n,m} = \left(\frac{h_x^2}{D_{1,n}}(\Sigma_{a1}+\Sigma_{r1})_n C_{1,n} - 1\right)\frac{2R_{nm}D_{1,m}}{D_{1,n}+D_{1,m}}, \quad with\; m \neq r \tag{2.93}$$

The equations (2.86) and (2.87) for the nodal average fluxes at node n when this node has one boundary touching the reflector, produce modifications in the elements of the production matrices M11 and M12 that after multiplication by $h_x^2 / D_{1,n}$ yield:

iii) Diagonal elements of the production matrix M11 denoted by $\left(M_{11}^{Diag}\right)_{n,n}$, when one face r of the node n touches the reflector:

$$(M_{11}^{Diag})_{n,n} =$$

$$\frac{h_x^2}{D_{1,n}}\left\{(\nu_1\Sigma_{f1})_n\left(B_{1,n}+C_{1,n}\left[\frac{2R_{nr}}{\left(1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}}\right)}+\sum_{m\neq r}\frac{2D_{1n}R_{nm}}{D_{1n}+D_{1m}}\right]\right)-\frac{2(\nu_2\Sigma_{f2})_n C_{2,n}\frac{h_{nr}}{2}\frac{\alpha_{21}}{D_{2,n}}R_{nr}}{\left(1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}}\right)\left(1+\frac{h_{nr}}{2}\frac{\alpha_{22}}{D_{2,n}}\right)}\right\}$$

(2.94)

iv) Non diagonal elements of the production matrix M11, denoted by $(M_{11}^{nD})_{n,m}$, these elements are the same ones that for the internal nodes, but only exist non- diagonal elements with $m \neq r$:

$$(M_{11}^{nD})_{n,m} = \left(\frac{h_x^2}{D_{1,n}}(\nu_1\Sigma_{f1})_n C_{1,n}\right)\frac{2R_{nm}D_{1m}}{D_{1n}+D_{1m}}, \text{ with } m \neq r \qquad (2.95)$$

v) Diagonal elements of the production matrix M12 denoted by $(M_{12}^{nD})_{n,n}$ when one face of the node n touches the reflector, that are given by the expression:

$$(M_{12}^{Diag})_{n,n} = \frac{h_x^2}{D_{1,n}}(\nu_2\Sigma_{f2})_n\left(B_{2,n}+2C_{2,n}\left[\frac{R_{nr}}{(1+\frac{h_{nm}}{2}\frac{\alpha_{22}}{D_{2,n}})}+\sum_{m\neq r}\frac{D_{2,n}R_{nm}}{D_{2,n}+D_{2,m}}\right]\right) \qquad (2.96)$$

vi) Non diagonal elements of the production matrix M12 denoted by , $(M_{12}^{nD})_{n,m}$, these elements are the same ones that for the internal nodes, but only exist the non-diagonal matrix elements with $m \neq r$:

$$(M_{12}^{nD})_{n,m} = \left(\frac{h_x^2}{D_{1,n}} (\nu_2 \Sigma_{f2})_n C_{2,n} \right) \frac{2R_{nm}D_{2,m}}{D_{2,n}+D_{2,m}} \quad with\ m \neq r \qquad (2.97)$$

vii) Diagonal elements of the matrix L21 denoted by $\left(L_{21}^{Diag}\right)_{n,n}$:

$$(L_{21}^{Diag})_{n,n} = \frac{h_x^2}{D_{2,n}}(\Sigma_{r1})_n \left(B_{1,n} + 2C_{1,n}\left[\frac{R_{nr}}{(1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}})} + \sum_{m\neq r} \frac{D_{1,n}R_{nm}}{D_{1n}+D_{1m}}\right]\right) + \left(L_{12}^{Diag.new}\right)_{n,n}$$

(2.98)

The term inside the braket containing α_{11}comes from the expression (2.86) of the average flux for the fast group when one face is in contact with the reflector. But there is a new matrix term $\left(L_{21}^{Diag.new}\right)_{n,n}$ given by the expression:

$$(L_{21}^{Diag,new})_{n,n} = -\frac{h_{nr}R_{nr}\alpha_{21}}{D_{2,n}\,(1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}})}\left[1-\frac{\frac{h_{nr}}{2}\frac{\alpha_{22}}{D_{2,n}}}{1+\frac{h_{nr}}{2}\frac{\alpha_{22}}{D_{2,n}}}\right] + \frac{h_x^2}{D_{2,n}}\Sigma_{a2,n}\frac{2C_{2,n}R_{nr}\frac{h_{nr}}{2}\frac{\alpha_{21}}{D_{2,n}}}{(1+\frac{h_{nr}}{2}\frac{\alpha_{11}}{D_{1,n}})(1+\frac{h_{nr}}{2}\frac{\alpha_{22}}{D_{2,n}})}$$

(2.99)

The term $\mathrm{L}_{21}^{\mathrm{Diag,new}}$ contains two contributions, the first one comes from the contribution to -L21 originating as consequence of the fast neutrons leaving the reactor and reentering as thermal neutrons on account of the ALBEDO α_{21}.This contribution comes from the first term in the right hand side of the thermal current at the boundary that contributes to the matrix $-L_{21}$. The second term of $\mathrm{L}_{21}^{\mathrm{Diag,new}}$comes from the third term of the expression of the average flux in the thermal group when one face touches the reflector, for this case there is a contribution to the thermal absorptions that depends also on the ALBEDO $\alpha_{21}$and the fast neutron flux in the center of the boundary node. These expressions are only valid when one single face touches the reflector. If more faces are in the frontier of the core with the reflector then one must add more similar terms. For the shake of simplicity we do not show these additional terms that however have been included in the NODAL-LAMBDA code.

viii) Non diagonal elements of the matrix L21, that are denoted by $\left(L_{21}^{nD}\right)_{n,m}$. For this case we have only non-diagonal elements of this matrix with $\mathrm{m} \neq \mathrm{r}$:

$$(L_{21}^{nD})_{n,m} = \frac{h_x^2}{D_{2,n}}(\Sigma_{r1})_n C_{1,n}\frac{2R_{nm}D_{1m}}{D_{1n}+D_{1m}} \quad with\ m \neq r \qquad (2.100)$$

ix) Diagonal elements of the leakage matrix L22, denoted by $\left(L_{22}^{Diag}\right)_{n,n}$, for the cases that one of the node faces has ALBEDO boundary conditions. For these cases the matrix elements are given by the following expression:

$$(L_{22}^{Diag})_{n,n} = P_{2,n}(r) + \frac{h_x^2}{D_{2,n}}(\Sigma_{a2})_n(B_{2,n} + C_{2,n}r_{2,n}(r)) \tag{2.101}$$

In this case when only one face has Albedo boundary conditions, i.e. this face is in contact with the reflector (r), the coefficient $\mathrm{P_{2,n}(r)}$ y $\mathrm{r_{2,n}(r)}$ are given by:

$$\mathrm{P_{2,n}(r) = \frac{h_{nr}\ R_{nr}\alpha_{22}}{D_{2,n}(1+\frac{h_{nm}}{2}\frac{\alpha_{22}}{D_{2,n}})} + \sum_{m\neq r} \frac{2D_{2m}R_{nm}}{D_{2m}+D_{2n}}} \tag{2.102}$$

$$\mathrm{r_{2,n}(r) = \frac{R_{nr}}{(1+\frac{h_{nr}}{2}\frac{\alpha_{22}}{D_{2,n}})} + \sum_{m\neq r} \frac{D_{2m}R_{nm}}{D_{2m}+D_{2n}}} \tag{2.103}$$

x) Non diagonal elements of the matrix *L22*, denoted by $\left(L_{22}^{nD}\right)_{n,m}$, these elements only exist when $\mathrm{m \neq r}$, and are given by:

$$(\, L_{22}^{nD}\,)_{nm} = \left(\frac{h_x^2}{D_{2,n}}(\, \Sigma_{a2}\,)_n C_{2,n} - 1 \right) \frac{2R_{nm}D_{2m}}{D_{2n}+D_{2m}},\ m \neq r \tag{2.104}$$

2.2.4 Improving the Albedo boundary conditions in coarse nodal methods

Several authors (Chung and Kim 1981) tried to improve the Albedo boundary condition for coarse mesh methods. They reasoned as follows based in previous developments by Turney (1975). For the case of one reflector homogenous and one dimension the fast neutron flux in the boundary is related with the neutron current by the approximated expression:

$$\phi_{B,1} = a_{11}J_{B,1} \approx a_{11}D_{I,1}\frac{\phi_{I,1}-\phi_{B,1}}{\left(\frac{h}{2}\right)} \tag{2.105}$$

Where $\phi_{I,1}$ denotes the fast flux at the node center and $\phi_{B,1}$ the fast flux at the boundary interface. This approximation according to Chung and Kim (1981) is very poor by two reasons, the first one is that we have a coarse meh approach and the second is that for most cases the gradient of the fast flux group is decreasing near the reflector interface. By the previous reasons Chung at al. proposed a coarse mesh correction factor following Turney (1975):

$$\phi_{B,1} = (a_{11}f_{CT})D_{I,1}\frac{\phi_{I,1}-\phi_{B,1}}{\left(\frac{h}{2}\right)} \tag{2.106}$$

These authors also substituted $a_{21}$by $a_{21}f_{CT}$. Therefore the elements of the Albedo matrix are modified as follows:

$$\alpha_{11} = \frac{1}{a_{11}f_{CT}} \tag{2.107}$$

$$\alpha_{21} = -\frac{a_{21}f_{ct}}{a_{11}f_{ct}a_{22}} = -\frac{a_{21}}{a_{11}a_{22}} \tag{2.108}$$

These authors recommended for this factorf_{CT} the value of 1.1. We have added to the LAMDA-NODAL code the possibility of using this coarse mesh factor.

3. Solving the eigenvalue 3D matrix equations with two groups and ALBEDO boundary conditions.

The matrix equations (2.42), to be solved, can be writen in the following form as shown by Doring et al. (1993) and Verdú et al. (1994)

$$A\phi_1 = L_{11}^{-1}(M_{11} + M_{12}L_{22}^{-1}L_{21})\phi_1 = k\phi_1 \tag{3.1}$$

The code NODAL-LAMBDA solves this matrix equation and computes its eigenvalues and eigenvectors using two well know packages ARPACK (Arnoldi Package), and LAPACK. We have modified several subroutines of the LAPACK package in order to store simultaneously in compress format CSR the non diagonal matrixes L_{11}, L_{21}, M_{11}, M_{12} that are found when solving this eigenvalue problem.

The ARPACK package is used to obtain the eigenvalues and eigenvectors by iterating in the Krylov subspace (Lehoucq 1997). The LAPACK library is used to solve the linear equation systems that appear when iterating in the Krylov subspace to solve the matrix eigenvalue equation (3.1).

We outline here the method that uses the NODAL-LAMBDA code to find the eigenvalues and eigenvectors of equation (3.1). The code uses a ARNOLDI method with implicit restart known as IRAM (Implicit Restart Arnoldi Method). This method is based on the improvement to the Arnoldi method performed by Sorensen (1992).

As it is well known the natural basis for the Krylov sub-space $K^j(A,x)$ in $R^n$is generated by sucesive aplications of the matrix A to an initial vector x, that gives $(x, Ax, A^2x,..., A^{j-1}x)$, the problem with this basis is that A^jxconverges to the direction of the eigenvector corresponding to the bigger eigenvalue in module. The consequence of this fact is that the Krylov basis is ill conditioned when j increases.

To try to solve this problem Arnoldi develop a procedure based on the Gram-Schmidt method of ortonormalization. In this procedure defining the initial normal vector $q_1 = x/\|x\|$, then the sucessive aplications of A^jq_1 for j=1,2,…are normalized by the Gram-Schmidt ortonormalization method. Instead of using this method that needs to orthonormalize A^jq_1 againts all previous orthonormal vectors $q_1, q_2, \ldots\ldots, q_j$ to obtain q_{j+1}. It is better, to obtain q_{j+1}, to orthonormalize Aq_j againts $q_1, q_2, \ldots\ldots, q_j$. Denoting by $r_j$the component of Aq_j that is orthogonal to $q_1, q_2, \ldots\ldots, q_j$, then we can write this component as follows:

$$r_j = Aq_j - \sum_{i=1}^{j}(q_i^{\mathrm{T}}A\,q_j)q_i \tag{3.2}$$

And make $q_{j+1} = r_j / \|r_j\|$.

Premultipliying equation (3.2) by the traspose q_{j+1}^{T} it is obtained:

$$q_{j+1}^{\mathrm{T}}\, r_j = q_{j+1}^{\mathrm{T}}\, Aq_j = \|r_j\| = h_{j+1,j} \tag{3.3}$$

Where we have defined the matrix element $h_{i,j} = q_i^{\mathrm{T}} A q_j$. Then on account of (3.2) and (3.3), Aq_j can be recasted as follows:

$$Aq_j = \sum_{i=1}^{j+1} h_{i,j}\, q_i \tag{3.4}$$

If happens that $h_{j+1,j}$=0, for a given j=J value, this means according to equation (3.3) that $\|r_J\|$=0, and from (3.2) it is obtained that $Aq_J = \sum_{i=1}^{J} (q_i^{\mathrm{T}} A\, q_J) q_i$, and we have found an invariant subspace with the orthonormal basis formed by the vectors $q_1, q_2, \ldots\ldots, q_J$.

Denoting by $e_j^T$the row vector $e_j^T = (0,0,0,\ldots\ldots,1)_{1\times j}$ with 1 row and j columns, and defining the matrix Q$_j$ formed by the first j elements of the Orthonormal Arnoldi basis :

$$Q_j = [q_1, q_2, \ldots, q_j] \tag{3.5}$$

Then the action of the matrix A on Q_j, on account of (3.4) yields:

$$AQ_j = [Aq_1, A\, q_2, \ldots, Aq_j] = \left[\sum_{i=1}^{2} q_i h_{i,1}, \sum_{i=1}^{3} q_i h_{i,2}, \ldots, \sum_{i=1}^{j+1} q_i h_{i,j}\right] \tag{3.6}$$

Defining now the upper Hessemberg matrix H_j, in the standard form:

$$H_j = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots\cdots & h_{1,j} \\ h_{2,1} & h_{2,2} & \cdots\cdots & h_{2,j} \\ 0 & h_{3,2} & \cdots & h_{3,j} \\ 0 & 0 & \ddots & \\ \vdots & \vdots & \cdots\ddots & \vdots \\ 0 & 0 & \cdots\ h_{j-1,j} & h_{j,j} \end{bmatrix} \tag{3.7}$$

We can express equation (3.6) in terms of the matrices Q$_j$ and H$_j$:

$$AQ_j = Q_j H_j + h_{j+1,j} q_{j+1} e_j^{\mathrm{T}} = Q_j H_j + r_j e_j^{\mathrm{T}} \tag{3.8}$$

This last equation is known as the Arnoldi-Lanczos relation (Arbenz et al 2012), and is the cornerstone of the Arnoldi method. If for j=J we have that $h_{J+1,J}$=0, then the spectrum of the Hessemberg matrix $H_J$is contained in the spectrum of A, i.e. $\sigma(H_J) \subset$

$\sigma(A)$, if for instance y_i denotes an eigenvector of the matrix, H_J with eigenvalue $\lambda_i$then we can write:

$$H_J y_i{=}\lambda_i y_i \tag{3.9}$$

On account of the Arnoldi-Lanczos relation for $j = J$ with $\|r_J\|$=0 we write:

$$AQ_J y_i = Q_J H_J y_i = \lambda_i Q_J y_i \tag{3.10}$$

Therefore if equation (3.9) is verified, then from equation (3.10) it follows that $\lambda_i \in \sigma(H_J) \subset \sigma(A)$, then λ_i belongs to the spectrum of H_J and therefore to the spectrum of A, being $Q_J y_i$the corresponding eigenvalue of A. These eigenvalues and eigenvectors $(\lambda_i, Q_J y_i)$ are known as the Ritz eigenvalues and eigenvectors of A.

The basic idea behind the Arnoldi method of the Krylov subspace, is to reduce the dimension of the search space, i.e. to stop the iterations at some value m before $\|r_J\|$=0. Therefore the Arnoldi iteration performed to obtain the basis $q_1, q_2, \ldots\ldots, q_J$ of the Krylov subspace is stopped after a number of steps $m \leq J$ (which should be bigger than the number of desired eigenvalues). The basic idea is to reduce the dimension of the search space without destroying the structure of the Krylov subspace in such a way that$\|r_m\|$ is small. Proceeding in this way, it is guaranteed that the eigenvalues of H_m are good approximations of the true eigenvalues of A, and the eigenvectors $Q_m y_i$ are also good approximations of the true eigenvectors of A (Sorensen 1992, Arbenz 2012, Tyrtyshnikov 1997, Saad 1992).

In the Arnoldi iteration the information obtained about the eigenvalues depends crucially on the election of the starting vector q_1. For this reason to restart the Arnoldi iteration the best way to do it is to choose carefully the initial vector q_1 in such a way as to eliminate the non-desired part of the spectrum. This goal is achieved by the Arnoldi implicit restart algorithm known as IRAM that applies $k < m$, Arnoldi implicit iterations with displacements, in order to assure convergence to the desired eigenvalues. The steps performed in this algorithm that is the one that uses the NODAL-LAMBDA code are:

i) First step

Starting from a unit vector $q_1 = x/\|x\|$ the IRAM algorithm executes first m step of the basic ARNOLDI algoritm i.e. each vector q_{j+1}, is obtained orthonormalizing Aq_j againts $q_1, q_2, \ldots\ldots, q_j$ as was explained previously. In this way after m step the following items are obtained: the Hessemberg matrix H_m given by (3.7), the matrix Q_m whose colums are the orthogonal vector of the Krylov subspace, and $r_m$that is the residual that still remains, given by equation (3.2), and that means that the subspace generated by the vectors $q_1, q_2, \ldots\ldots, q_m$, is not invariant under the action of A. When this residual is zero the subspace generated by $q_1, q_2, \ldots\ldots, q_m$ in invariant under the action of A. To start the second step we denote by: $H_m^{(1)} = H_m$, and $Q_m^{(1)} = Q_m$ to the matrices obtained in the first step.

ii) Second Step, for n=1, until convergence ii-a Compute the eigenvalues $\lambda_i, for\ \ i = 1,2,..m,$ ordered from the bigger to the smaller ones and the eigenvectors $y_i for\ , i = 1,2, \dots, m,$ of the Hessember matrix H$_m$ defined in (3.7).

ii-b Select the number of not desired eigenvalues, that is given by p=m-nev, being nev the number of eigenvalues to be computed. These p eigenvalues are the smaller ones of the set i.e the more subcritical ones, and are $\lambda_i, for\ \ i = m - p + 1, m - p + 2, \dots.., m.$

ii-c Perform p implicit steps of QR factorization denoted by VR with shift applied to H$_m$

```
    1-for i=1,2,….p
    2- $H_m^{(i)} - \mu_i$ I = $V_i\ R_i$ being the displacements $\mu_1 = \lambda_{m-p+1}, \dots., \mu_p = \lambda_p$
                                                                            (3.11)
    3- $H_m^{(i+1)} := V_i^T H_m V_i = R_i V_i + \mu_i$ I                          (3.12)
    4- $V^{(i+1)} := V^{(i)}\ V_i,\ Q_m^{(i+1)} := Q_m^{(i)} V_i$                  (3.13)
    4-End for
```

After execution of the p iterations it is obtained the matrix $V \in R^{m\times m}$ given by: $V = V^{(p+1)} = V_1\, V_2 \dots V_p$, and the new matrix $H_m^+ = V^T H_m$V. Also after the p steps we have a new orthogonal matrix $Q_m^+ = Q_m$V, being $Q_m^+ \in R^{n\times m}$. It is noticed that the matrix V is the product of p orthogonal matrices of the Hessemberg type and it has p off diagonals different from zero below the main diagonal (Arbenz 2012, Saad 1992). Therefore from the Arnoldi relation (3.8) for j=m, and postmultiplying this relation by the orthogonal matrix V it is obtained:

$$A(Q_m\mathrm{V}) = (Q_m V)(V^T\ H_m\mathrm{V}) + r_m e_m^{\mathrm{T}} V \tag{3.14}$$

On account of the matrix definitions:

$$Q_m^+ := Q_m\mathrm{V}\ , \text{and } H_m^+ := V^T H_m\mathrm{V} \tag{3.15}$$

Equation (3.14) can be expressed as follows:

$$\mathrm{A}Q_m^+ = Q_m^+\ H_m^+ + \mathrm{r_m e_m^{\mathit{T}} V} \tag{3.16}$$

On account of the structure of the matrix V, with p off diagonals below the main diagonal, it is obtained for the last term of (3.16):

$$\mathrm{e}_{\mathrm{m}}^{T}\mathrm{V} = (0,0, \dots 0, \mathrm{V_{m,k}}, \mathrm{V_{m,k+1}}, \dots,, \mathrm{V_{m,k+p}}) \tag{3.17}$$

This row vector has the last p+1 elements different from zero, and the first k-1 equal to zero. So simply the next step is to retain in equation (3.16) the first k colums and to discard the last p=m-k columns. These k columns of the first term of the right hand side of equation (3.16) yields the following matrix, where we have used for the left hand side the colon notation of Golub and Van Loan (1996):

$$Q_m^+ \, H_m^+(:,1{:}k) = [Q_m^+ \, h_1^+, Q_m^+ \, h_2^+, \ldots, Q_m^+ h_k^+ \,] \tag{3.18}$$

Where $h_1^+, h_2^+, \ldots, h_k^+$ are the first k columns of the Hessember m x m matrix H_m^+. The last column vector of (3.18) can be expressed in terms of the colum vector of the nxm matrixQ_m^+ and the components of h_k^+, this yields:

$$Q_m^+ \, h_k^+ = \sum_{i=1}^{k} h_{i,k}^+ \, q_i^+ + h_{k+1,k}^+ q_{k+1}^+ \tag{3.19}$$

In obtaining (3.19) we have considered the fact that $h_{i,k}^+ = 0$ for $i > k+1$. Therefore from equations (3.16), (3.17) and (3.19), it is finally obtained the following result for the first k columns of $\mathrm{A}Q_m^+$ expressed in the colon notation of Golub and Van Loan (1996):

$$\mathrm{A}Q_m^+(:,1{:}k) = Q_m^+(:,1{:}\mathrm{k}) \; H_m^+(1{:}k, 1{:}k) + (h_{k+1,k}^+ q_{k+1}^+ + \mathrm{V}_{\mathrm{m,k}} \mathrm{r}_{\mathrm{m}}) \mathrm{e}_{\mathrm{k}}^{\mathrm{T}} \tag{3.20}$$

We note that in this notation $H_m^+(1{:}k, 1{:}k) = H_k^+$, is the matrix formed by the first k rows and k columns of the matrix H_m^+.

Therefore the remaining vector is:

$$r_k = (h_{k+1,k}^+ q_{k+1}^+ + \mathrm{V}_{\mathrm{m,k}} \mathrm{r}_{\mathrm{m}}) \tag{3.21}$$

ii-d Then starting with equation (3.20) execute p aditional steps of the Anoldi algorithm, extending the Arnoldi factorization of length k=m-p to length m obtaining H_m^+ , Q_m^+, and r_m.

ii-e Continue until convergence is attained. If convergence is not attained go to the second step. The new starting vector denoted by $q_1^{(new)}$used as initial vector to restart the new iteration is the first column of Q_m^+, therefore we write:

$$q_1^{(new)} = Q_m^+(:, \; 1) \tag{3.22}$$

It is worthwile to discuss why to choose this initial vector to restart the iterations. Note that equation (3.22) can be expressed in the equivalent form (Golub and Van Loan 1996):

$$q_1^{(new)} = Q_m^+(:, \; 1) = Q_m V e_1 \tag{3.23}$$

Where $e_1 = column(1,0,0, \ldots..0)$ being m the dimension.of this vector. We notice that $Ve_1$is the first column of V that can be obtained as follows, consider the product of

$VR_pR_{p-1}\dots R_1$, then take into account the fact that $R_pR_{p-1}\dots R_1$is upper triangular we can write:

$$VR_pR_{p-1}\dots R_1 = V_1V_2\dots(V_pR_p)R_{p-1}\dots R_1 = V_1V_2..V_{p-1}(H_m^{(p)} - \mu_p I))R_{p-1}\dots R_1 \tag{3.24}$$

Then take into account in equation (3.24) that $H_m^{(p)} = V_{p-1}^T H_m^{(p-1)} V_{p-1}$and using the fact that $V_{p-1}$is an orthogonal matrix, it is deduced:

$$VR_pR_{p-1}\dots R_1 = V_1V_2..V_{p-1}(V_{p-1}^T H_m^{(p-1)} V_{p-1} - \mu_p V_{p-1}^T V_{p-1})R_{p-1}\dots R_1 \tag{3.25}$$

Using again the orthogonal character of V_{p-1}it is obtained on account of (3.11), the following result

$$\begin{aligned} VR_p\dots R_1 &= V_1V_2..V_{p-2}(H_m^{(p-1)} - \mu_p I)V_{p-1}R_{p-1}\dots R_1 = \\ &= V_1..V_{p-2}(H_m^{(p-1)} - \mu_p I)(H_m^{(p-1)} - \mu_{p-1} I)R_{p-2}\dots R_1 \end{aligned} \tag{3.26}$$

Repeating the same procedure p-2 times more one finally arives to the following result:

$$VR_p\dots R_1 = (H_m^{(1)} - \mu_p I)(H_m^{(1)} - \mu_{p-1} I)\cdots(H_m^{(1)} - \mu_1 I) \tag{3.27}$$

Now because the product $R_pR_{p-1}\dots R_1$is an upper triangular matrix, then on account of (3.27) it is obtained that:

$$Ve_1 = \alpha(H_m^{(1)} - \mu_p I)(H_m^{(1)} - \mu_{p-1} I)\cdots(H_m^{(1)} - \mu_1 I) \tag{3.28}$$

Being α an scalar, so finally on account of (3.8) and (3.28) one arrives after some algebra to:

$$q_1^{(new)} = \alpha\big(A - \mu_p I\big)\big(A - \mu_{p-1} I\big)\cdots(A - \mu_1 I)Q_m e_1 + e_m^T \mathcal{P}^{(p-1)}(H_m)e_1 \tag{3.29}$$

Being $\mathcal{P}^{(p-1)}(H_m)$ a matrix polynomial in H_m of order (p-1). Because the matrix obtained from this polynomial is a lower bandwidth p-1 one, then it is easily checked that the last term of (3.30) vanishes.

$$q_1^{(new)} = Q_m V e_1 = \alpha(A - \mu_p I)(A - \mu_{p-1} I)\cdots(A - \mu_1 I)Q_m e_1 \tag{3.30}$$

So according to equation (3.30) we can say that in the restarting Arnoldi method the sucessive initial guesses are sucesively filtered of the unwanted portion of the spectrum, proceeding in this way the convergence to the "wanted" portion of the spectrum is accelerated.

4. Validation of the NODAL-LAMBDA code and Results obtained for the Eigenvalues of the Harmonic modes

Several well known test cases have been used to validate the Nodal-Lambda code with 3D geometry and two groups, also comparison with the results of other codes have been performed. A version of the NODAL-Lambda code have been also developped for 1.5 groups. The results obtained when using this simplification are also shown and compared with the case of two group for the fundamental and the subcritical modes. The goal is to know in what cases we can use the 1.5 group approximation, and in what cases this approximation must be discarded because does not provide good values for the calculation of the subcritical modes.

4.1. Validation of the NODAL-LAMBDA code with test cases of Ringhals Nuclear Power Plant

The first calculation performed with the NODAL-LAMBDA two groups code was the Ringhals NPP test 14G (Lefvert, 1994). The cross section of this test were provided by IBERINCO that performed the calculations with the CASMO-SIMULATE codes; the value obtained with the SIMULATE code for the multiplication constant was 1.00263 and we call it the "reference value" although it is not necessarily the true value. The NODAL-LAMBDA (NL) code yields a value for this same case of 1.00225 with a difference of 38 pcm (1pcm=10^{-5}) as displayed in table 4.1. Also we performed calculations of this same case with the LAMDA (L) code that is based on an expansion in Legendre polynomials performed inside each node and the method of nodal collocation (Verdú et al. 1994, Hebert 1987). Two kind of calculations were performed, the first one using an expansion in Legendre polynomials of order 2 (see table 4.1), and the second one using a more exact approach of order 3 (see table 4.2). In the first case the difference found with the reference case was -35 pcm, and in the second case the difference was -53 pcm. When we compute the differences between NL and L codes it is found $k_{0,NL}$-$k_{0,L}$= -3 pcm for order 2, and of $k_{0,NL}$-$k_{0,L}$=14 pcm for order 3. So the differences among both codes are practically negligible in spite of the fact that the method of solving the neutron diffusion equation in both codes is different and also is different the way to implement the boundary conditions (we remind that the L code use zero flux boundary conditions while NL uses Albedos). Both codes yield similar values for the subcritical eigenvalues.

In table 4.1 and 4.2 we also display the first 3 subcritical eigenvalues computed with the NL and L codes when using order 2 and 3 of the Legendre expansion in the LAMBDA code. It is observed that the differences between the eigenvalues computed by both codes is small, and that the three first subcritical eigenvalues have sub-criticalities ranging between 1 and 2 dollars. Finally it is important the comparison of the axial power profiles. Figure 1 shows that the profiles practically match except at the first point from below. When using a higher polynomial expansion of order 3 (L3) that consumes much more time the first point computed with the Lambda code is closer to the first point of the profile computed with the NODAL-LAMBDA code as displayed at figure 2, and the points 2, 3, and 4 displaces slightly downward as observed in figure 2.

Table 4.1. Eigenvalues of Ringhals NPP at point (14G) with the codes NODAL-LAMBDA and LAMBDA using a Legendre 2 expansion in the LAMBDA code

k	Reference	Nodal-Lambda V-1 Mesh 32x32x27	Δk Nodal – Lambda	LAMBDA-2 Mesh 32x32x27	Δk Lambda
k_0	1.00263	1.00225	k_0-$k_{0,ref}$=-38 pcm	1.00228	k_0-$k_{0,ref}$ =-35 pcm
k_1	-	0.99519	k_1-k_0=-0.00706 (-1.22$)	0.99553	k_1-k_0=-0.00675 (-1.17$)
k_2	-	0.99412	k_2-k_0=-0.00813 (-1.407$)	0.99403	k_2-k_0=-0.00825 (-1.429$)
k_3	-	0.99116	k_2-k_0=0.01009 (-1.74$)	0.99153	k_3-k_0=0.01075 (-1.86$)

Table 4.2. Eigenvalues of Ringhals NPP at point (14G) with the codes NODAL-LAMBDA and LAMBDA using a Legendre 3 expansion in the LAMBDA code

k	Reference	Nodal-Lambda V-1 Mesh 32x32x27	Δk Nodal – Lambda	LAMBDA-3 Mesh 32x32x27	Δk Lambda
k_0	1.00263	1.00225	k_0-$k_{0,ref}$=-38 pcm	1.00211	k_0-$k_{0,ref}$ =-53 pcm
k_1	-	0.99519	k_1-k_0=-0.00706 (-1.22$)	0.99529	k_1-k_0=-0.006817 (-1.18$)
k_2	-	0.99412	k_2-k_0=-0.00813 (-1.407$)	0.99365	k_2-k_0=-0.008457 (-1.463$)
k_3	-	0.99116	k_2-k_0=0.01009 (-1.74$)	0.99153	k_3-k_0=-0.01057 (-1.831 $)

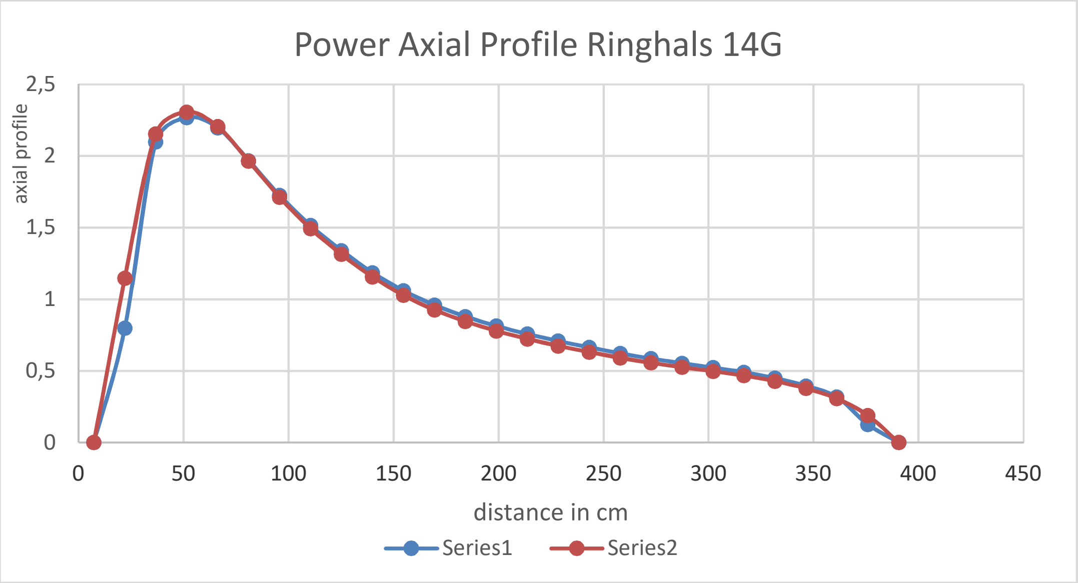

Figure 1 Power axial profile computed with the Nodal-Lambda code (Series 1) and the Lambda code (Series 2) using a legendre expansion of order 2 in the Lambda code.

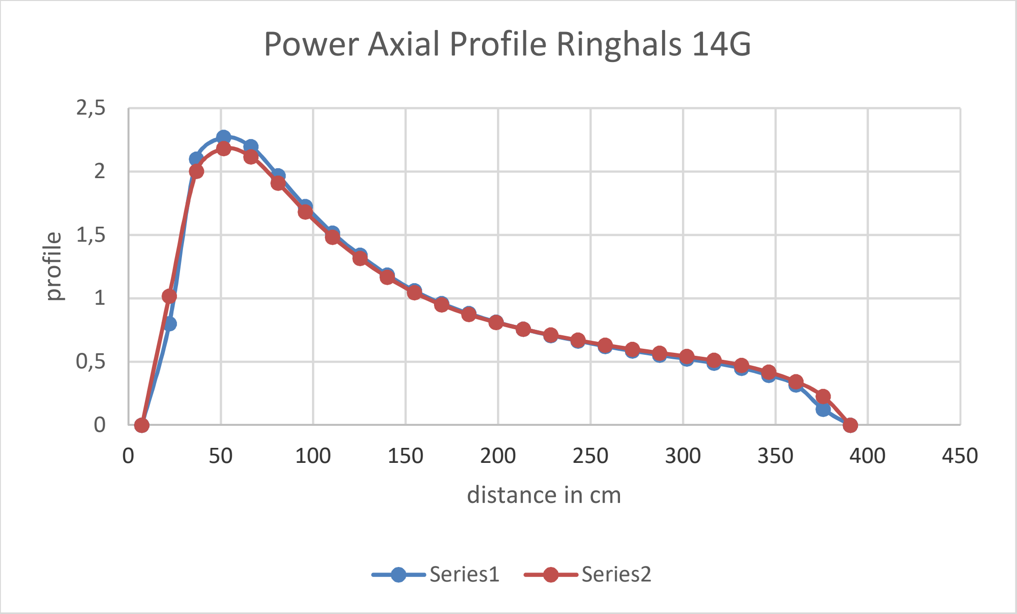

Figure 2 Power axial profiles computed with the Nodal-Lambda code (Series 1)and the Lambda code (Series 2) using a Legendre expansion of order 3 in the Lambda code.

Because both codes uses different types of boundary conditions we are going to discuss this issue. The code Nodal-Lambda uses albedo boundary conditions so it was decided to investigate the effect of the Albedos on the results. So we tried some improvements proposed for the Albedos in the past by Chung (1981) and Turney (1975) and that will be discussed in the next subsection.

4.2 Validation with the 3D PWR Benchmark of the IAEA

One of the selected bechmark was the 3D IAEA benchmark for a PWR reactor the reason is that for this benchmark the fundamental multiplication constat of the fundamental node k_0 is known with very high precision because a very fine mesh was used, and the multiplication constant value was obtained by Richardson extrapolation (Michelsen 1977). Also this Benchmark have been computed by many authors, so we have a very high degree of confidence about the results of this benchmark. The fundamental value of the multiplication constant for this benchmark is 1.02903. Because normally the codes does not provide the values of the subcritical eigenvalues, we have compared these values with the ones obtained by the LAMBDA code that uses an expansion in legendre polynomials (Verdú et al 1994). Because the methods used by both codes are completely different we think that if the results of both codes are closer we can have a big degree of confidence in the results of the NODAL-LAMBDA (NL) code. Obviously because the NL code uses a coarse mesh approximation the computing time in much lower in the NL code that in the LAMBDA code. First we performed a comparison with the results of the LAMBDA code using an expansion in Legendre polynomials of order two, and a Krylov dimension of 12, and the results are displayed at table 4.3. For the fundamental mode the diference with the reference value computed with NL is 59 pcm, and with the LAMBDA code using a legendre expansion of order 2 the difference is 35 pcm. However this difference for the fundamental eigenvalue disminishes for the LAMBDA code when using a Legendre expansion of order 3 to only 9 pcm as displayed in table 4.4. When comparing the subcriticalities i.e k_1-k_0 obtained by both methods for the first mode i.e. the NL code yields -0.01377, and the Lambda code with legendre expansion of order two (L2) yields -0.01248 so the difference between both methods is of 129 pcm. This difference become smaller for the subcriticality of the third eigenvalue i.e. $k_3 - k_0$, the value obtained with NL is -0.010405, while L2 yields -0.01400, therefore the difference between both codes is only 5 pcm. Both codes the NL and the Lambda code identify the two subcritical eigenvalues with the same subcriticality that must appear mathematically due to the quarter simmetry and degeneracy of the problem as displayed in table 4.3.. When increasing the order of the legendre polynomial expansion in the Lambda code from 2 to 3, the subcriticality k_1-k_0 changes from -0.01248 to -0.01210 as it is observed in the last column of tables 4.3 and 4.4.

An additional comparison was perfomed taking into account the Chung-Turney correction factor f_{CT} in the Albedo matrix. Table 4.5 displays the results of the NODAL-LAMDA code for the fundamental mode and the harmonics when the Albedo matrix is modifid with f_{CT}=1.1, that is the recommended value for this parameter. In this case the difference between the fundamental eigenvalue computed with the NODAL-LAMDA code and the reference value is of 49 pcm. The value obtained for the fundamental eigenvalue is closer to the reference one for this case. The subcriticality k_1-k_0 for the first and second eigenvalues changes -0.01377 to -0.01373. We observe that also for this case appears two eigenvalues with the same subcriticality.

Table 4.3. Lambda Eigenvalues of the 3D-PWR IAEA Benchmark obtained with NODAL-LAMBDA with a Krylov subspace of dimension 12, and with the code LAMDA using a Legendre expansion of order 2.

k	Reference	Nodal-Lambda Mesh 10cm x10cm x20cm	Δk Nodal - Lambda	LAMBDA Mesh 10x10x20 Legendre Order 2	Δk Lambda
k_0	1.02903	1.02844		1.02868	
k_1	-	1.01467	k_1-k_0=-0.01377	1.01620	k_1-k_0=-0.01248
k_2	-	1.01467	k_2-k_0=-0.01377	1.01620	k_2-k_0=-0.01248
k_3	-	1.01439	k_3-k_0=-0.01405	1.01449	k_3-k_0=-0.01405

Table 4.4. Lambda Eigenvalues of the 3D-PWR IAEA Benchmark obtained with NODAL-LAMBDA with a Krylov subspace of dimension 12, and with the code LAMDA using a Legendre expansion of order 3.

k	Reference	Nodal-Lambda Mesh 10cm x10cm x20cm	Δk Nodal - Lambda	LAMDA Mesh 10x10x20 Legendre Order 3	Δk Lambda
k_0	1.02903	1.02844		1.02894	
k_1	-	1.01467	k_1-k_0=-0.01377	1.01684	k_1-k_0=-0.01256
k_2	-	1.01467	k_2-k_0=-0.01377	1.01684	k_2-k_0=-0.01256

k_3	-	1.01439	k_3-k_0=-0.01405	1.01494	k_3-k_0=-0.01400

Table 4.5. Lambda Eigenvalues of the 3D-PWR IAEA Benchmark obtained with NODAL-LAMBDA with a Krylov subspace of dimension 12 and a Churney-Turney correction factor in the Albedo of 1.1, and with the code LAMDA using a Legendre expansion of order 2, and a Krylov dimension of 12.

k	Reference	Nodal-Lambda V-2 Mesh 10cm x10cm x20cm	Δk Nodal - Lambda	LAMBDA Mesh 10x10x20 Legendre Order 2	Δk Lambda
k_0	1.02903	1.02854		1.02869	
k_1	-	1.01481	k_1-k_0=-0.01373	1.01620	k_1-k_0=-0.01248
k_2	-	1.01481	k_2-k_0=-0.01373	1.01620	k_2-k_0=-0.01248
k_3	-	1.01464	k_3-k_0=-0.01390	1.01449	k_3-k_0=-0.01419

Table 4.6. Lambda Eigenvalues of the 3D-PWR IAEA Benchmark obtained with PARCS using the IRAM method with a Krylov subspace of dimension 12.

k	Reference	PARCS Mesh 10cm x10cm x20cm	Δk PARCS
k_0	1.02903	1.02909	
k_1	-	1.01723	k_1-k_0=-0.01186
k_2	-	1.01723	k_2-k_0=-0.01186
k_3	-	1.01539	k_3-k_0=-0.01370

Table 4.6 shows the four largest eigenvalues calculated by the PARCS code for the IAEA 3D PWR benchmark problem (Wysocki 2014, 2015). In addition, the 2D (axially-averaged) contours for these four eigenmodes, as calculated by PARCS, are shown in Figure 3. The fundamental mode shape associated with k_0 corresponds to the radial assembly power distribution. k_1 and k_2 correspond to the first two azimuthal modes, with -45 degree and +45 degree lines of symmetry, respectively, relative to the x-axis. Due to the quarter-core symmetry of the IAEA 3D PWR Benchmark Problem core, these eigenmodes are degenerate and have identical eigenvalues (i.e. k_1=k_2), which was also true of the NODAL-LAMBDA code results. PARCS calculated a subcriticality for modes 1 and 2 (i.e. k_1-k_0 or k_2-k_0) that was 187 pcm less than that calculated by NODAL-LAMBDA with the Churney-Turney correction factor. However, the subcriticality calculated by PARCS for mode 3 (i.e. k_3-k_0) was only 20 pcm less than that calculated by NODAL-LAMBDA with the Churney-Turney correction factor.

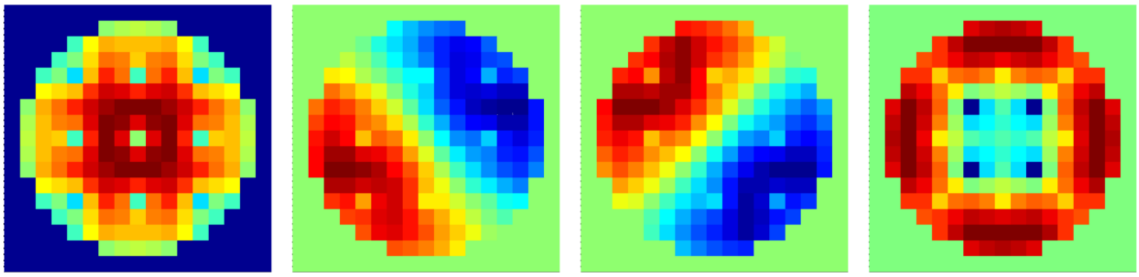

Figure 3 Normalized, axially-averaged 2D contour plots for the first four eigenmodes calculated by PARCS for the 3D-PWR IAEA Benchmark Problem. From left to right, the contours correspond to Mode 0 (fundamental), Mode 1, Mode 2, and Mode 3.

Concerning the axial power profiles of IAEA 3D PWR Benchmark Problem, figure 4 displays the axial power profiles obtained with PARCS and the Nodal-Lambda code. We see that both profiles are very close at all the points except a small difference in the lower node and a little bit smaller in the upper node. Also we performed the same calculation with the Nodal lambda code but now using the NODAL-LAMBDA code with the Chung-Turney correction in the Albedo and practically the power profile was the same one that with the PARCS code. Finally figure 5 displays the power axial profiles obtained with the Lambda code using a Legendre expansion of order two and three and the nodal collocation method we see that the results of this code practically match the results obtained with the NODAL-LAMBDA code for both cases.

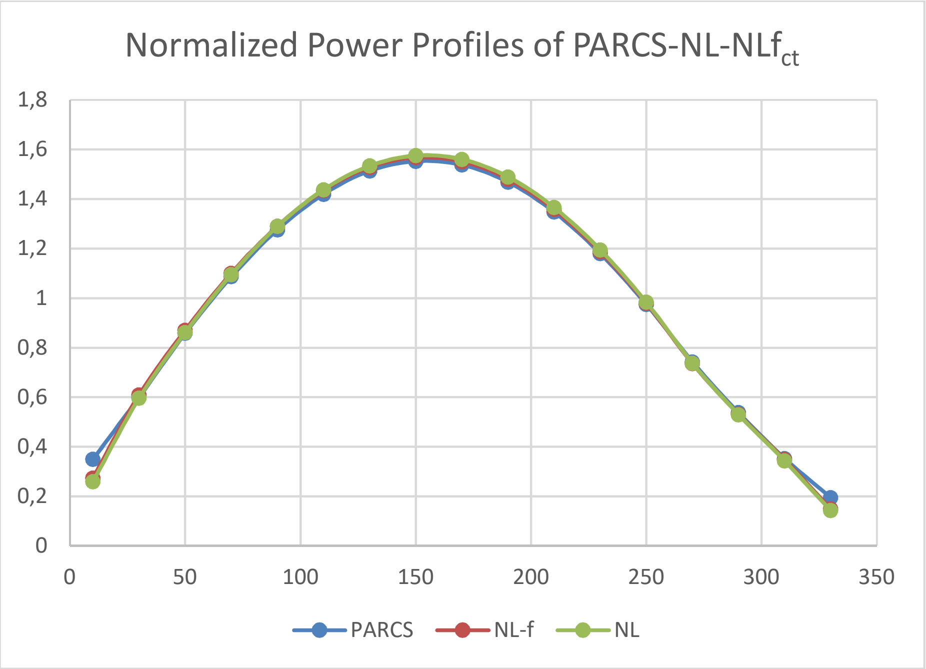

Figure 4 Power axial profiles vs the axial distance (cm) computed with PARCS, the Nodal-Lambda code (NL) and the Nodal-Lambda code (NL-f) using a Chung-Turney correction factor of 1.1, for the 3D PWR IAEA Benchmark.

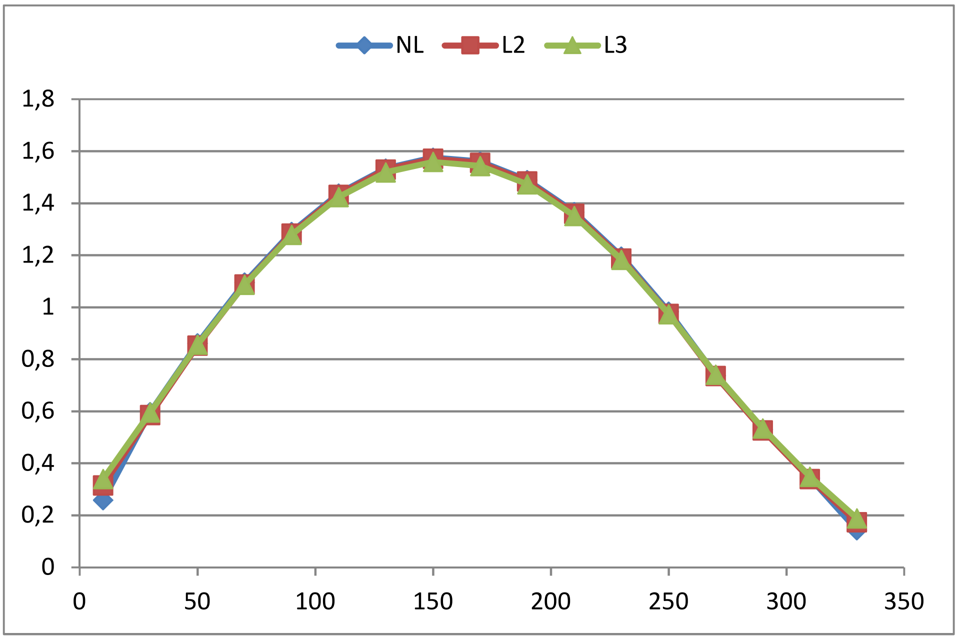

Figure 5 Power axial profiles vs the axial distance computed with the Nodal-Lambda code (NL) without Chung-Turney correction, and the Lambda code using a Legendre expansion of order 2 (L2) and order 3 (L3) for the IAEA 3D PWR Benchmark.

4.3 Validation with the Instability event of Cofrentes NPP 1991

In this case we have an asymmetric power distribution with a peak in the lower part of the core bigger than the usual one produced by the the fail of one of the preheaters as displayed in figure 6. In this case we obtained good results using a Chung-Turney factor of 1.2. The difference in the fundamental eigenvalue between the Nodal-Lambda code and the reference value computed with the SIMULATE code was 177 pcm, this difference increases to 515 pcm for the Lambda code with an expansion of order 3. Concerning the two orthogonal eigenvalues and eigenmodes with subcriticalities that are expected to be very close, the NODAL-LAMBDA and the LAMBDA code of order 3 computed both eigenvalues, k_1 and k_2, being k_1 and k_2 very close as displayed in the third and sixth column of table 4.7. The subcriticalties calculated by both codes for the first three hamonic modes i.e. k_1-k_0, k_2-k_0, k_3-k_0 are very close as it is displayed in columns 4 and 6 of table 4.7, only 25 pcm for the subcriticality of the first harmonic and 3 pcm for the subcriticality of the second harmonic.

In order to know the influence of the Chung-Turney factor on the lambda eigenvalues we display at table 4.8 the first 4 eigenvalues computed with the NODAL-LAMBDA code without Chung-Turney correction and with Chung-Turney factors equal to 1.1, 1.2, 1.3. The effect of increasing this correction factor in the fast group Albedo is to increase the effective multiplication factor as shown in table 4.8. The subcritical eigenvalues also increases with the incremets of the Chung-Turney correction factor. However the effect on the subcriticalities of the first and second harmonic modes is very small 1 pcm for the case with f_{CT}=1.1, and 3 pcm for the case with f_{CT}=1.2, as displayed in table 4.9. These subcriticalities are all them very similar to the ones obtained with the Lambda code using a Legendre expansion of order 3. Finally, Ffgure 5 displays the power profile obtained with the N-L code for f_{CT}=1.2, and 1.3, and the profile obtained with the Lambda code using an expansion of order 2.

Tabla 4.7. Lambda eigenvalues obtained with the Lambda code (Order 3) and with the NODAL-LAMBDA code for Cofrentes instability event 1991 (β=0.00556)

k	Reference SIMULATE 32 x 32 x 27	Nodal-Lambda f_{CT}=1.2 Mesh 32x32x27	Δk Nodal – Lambda	LAMBDA Mesh32x32x27 (Order 3)	Δk Lambda
k_0	1.00268	1.00091	$k_{0,simulate}$-$k_{0,nodal-L}$ =0.00177 (177 pcm)	0.99753	$k_{0,simulate}$-$k_{0,Lambda}$ =0.00515 (515 pcm)
k_1	-	0.99496	k_1-k_0=-0.00595 (-1.0701$)	0.99133	k_1-k_0=-0.00620 (-1.115$)
k_2	-	0.99458	k_2-k_0=-0.00633 (-1.133$)	0.99123	k_2-k_0=-0.00630 close in (-1.133$)
k_3		0.99252	k_3-k_0=-0.00839 (-1.508$)	0.98904	k_3-k_0=-0.00849 (-1.516$)

Table 4.8. Lambda eigenvalues obtained with the Lambda code (Order 3) and with the NODAL-LAMBDA code with f_{CT}=1, 1.1, 1.2, 1.3 for Cofrentes instability event 1991 (β=0.00556)

k	Nodal-Lambda Mesh 32x32x27	Nodal-Lambda f_{CT}=1.1 Mesh 32x32x27	Nodal-Lambda f_{CT}=1.2 Mesh 32x32x27	Nodal-Lambda f_{CT}=1.3 Mesh 32x32x27	LAMBDA Mesh32x32x27 (Order 3)
k_0	1.00042	1.00066	1.00091	1.00117	0.99753
k_1	0.99436	0.99467	0.99496	0.99526	0.99133
k_2	0.99412	0.99435	0.99458	0.99481	0.99123
k_3	0.99115	0.99183	0.99252	-	0.98904

Tabla 4.9. Lambda Subcriticalities with respect to the fundamental eigenvalue obtained with the Lambda code (Order 3) and with the NODAL-LAMBDA code with f_{CT}=1, 1.1, 1.2, 1.3 for Cofrentes instability event 1991 (β=0.00556)

Δk	Nodal-Lambda Mesh 32x32x27	Nodal-Lambda f_{CT}=1.1 Mesh 32x32x27	Nodal-Lambda f_{CT}=1.2 Mesh 32x32x27	Nodal-Lambda f_{CT}=1.3 Mesh 32x32x27	LAMBDA Mesh32x32x27 (Order 3)
k_1- k_0	-0.00606 (-1.089$)	-0.00599	-0.00595	-0.00591	-0.00620 (-1.115$)
k_2-k_0	-0.00630 (-1.133 $)	-0.00631	-0.00633	-0.00636	-0.00630 (-1.133$)
k_0-k_3	-0.00927	-0.00883	-0.00839	-	-0.00849

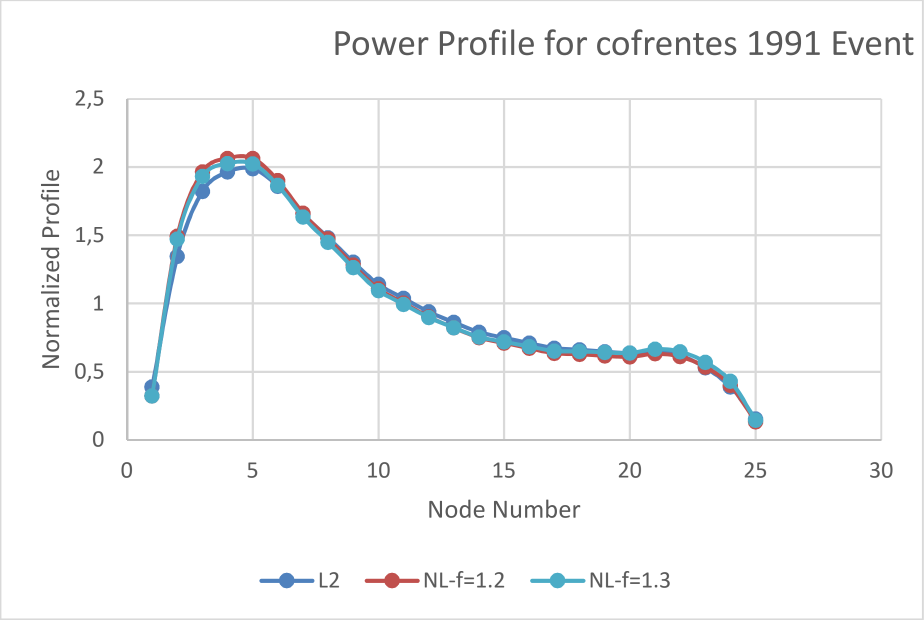

Figure 6 Power axial profiles vs the node number computed with the Nodal-Lambda code (NL) with Chung-Turney correction factor equal to 1.2 and 1.3, and the Lambda code using a Legendre expansion of order 2 (L2) for Cofrentes 1991 Instability Event.

4.4 Results obtained with the approximation of 1.5 Group

We developped a version of the NODAL Lambda code for 1.5 groups but the results for the subcriticaliy of the harmonic modes for the Ringhals tets case were different from the values obtained with the version of the NODAL-LAMBDA code for two neutron groups and the values obtained with the Lambda code for expansions of order 2 and 3. Table 4.10 displays a comparison of the results obtained with the Nodal-Lambda code for two groups and for 1.5 groups. So we conclude that the 1.5 group approximation is not a good method to obtain the eigenvalues of the reactor cores.

Tabla 4.10. Eigenvalues of Ringhals NPP at point (14G) with the codes NODAL-LAMBDA with two groups and using the approximation of 1.5 group with one special version of the NODAL-LAMBDA code. (β =0.00578)

k	Reference	Nodal-Lambda 2-groups Mesh 32x32x27	Δk Nodal–Lambda 2-groups	Nodal-Lambda 1.5 groups Mesh 32x32x27	Δk Nodal-Lambda, 1.5-groups
k_0	1.00263	1.00225	k_0-$k_{0,ref}$=-38 pcm	1.004220	k_0-$k_{0,ref}$ =-159 pcm
k_1	-	0.99519	k_1-k_0=-0.00706 (-1.22$)	0.998363	k_1-k_0=-0.00586 (-1.013 $)
k_2	-	0.99412	k_2-k_0=-0.00813	0.99775	k_2-k_0=-0.006473

			(-1.407$)		(-1.12 $)

5 Conclusions

The calculation of the subcriticality respect to the fundamental mode, for the first, second and higher order harmonic modes is important in nuclear reactor stability problems, specially for out of phase oscillations in BWR reactors and in xenon spatial oscillations. The goal of this paper was to develop a fast method to obtain the eigenvalues and eigenvectors for the eigenvalues closer to the fundamental one with a high degree of confidence, and the subcriticality of these harmonic modes i.e the difference $k_i - k_0$. The reason is that this difference appears in the modal equations (Muñoz-Cobo et al 2000, Hashimoto et al 1997) as a negative damping factor term and when more negative is this difference more damped is the respective amplitude of the corresponding harmonic mode. So that computing the subcriticality of these harmonic modes for different reactor situations give us a knowledge of the possibility to suffer rector instabilities of the out-of-phase type, specially if the subcriticality is close to 1$, and by some feedback mechanism that involves some asymetry in the power production in the core it is provided enough reactivity feedback in the harmonic mode equation to overcome the eigenvalue separation. The subcriticality of the i-th mode depends on the values of k_i and also on the value of k_0 so good precision is necessary in computing both eigenvalues.

The nodal collocation methods (Verdú et al 1994) provide a method to estimate the eigenvalues and eigenvectors of the 2-group diffussion equation, although it is needed at least an expansion of order 2 or 3 in Legendre polynomials in order to have good estimations of the eigenvalues of the subcritical modes. The consequence is that to have a good estimation of the fundamental and subcritical eigenvalues the number of polynomial coefficients that must be obtained at each node increases with the order expansion. So the size of the matrix problem to be solved becomes very large for high expansion orders, and for this reason this kind of method do not go beyond order 3.

For instance using the nodal collocation method with the serendipity approximation in 3D geometry yields that the vector dimension is$\frac{3(3+1)(3+2)N}{6} = 10N$, being N the number of nodes. Good estimations of the power profiles and the fundamental eigenvalue are obtained with this method using the order 3 polynomial expansion as displayed in figure 4. However the LAMBDA code with the expansion of order 3, need a dimension of the Krylov subspace bigger or equal than 9 and to increase the number of desired eigenvalues to find all the eigenvalues closer to 1 in particular the degenerate ones. In this case the LAMBDA code used the ARNOLDI package and the implemented method was the IRAM.

The kind of nodal methods known as coarse mesh finite difference methods that were started by Borressen (1971), using the 1.5 group approximation have less unkonwns than the Nodal collocation methods but does not give good estimation of the subcritical eigenvalues in the 1.5 group approximation, as we have proved in this paper developping an especific code that uses this approximation. The results are much better using the 2 Groups Borresen approximation (2G NODAL-LAMBDA code) with albedo boundary conditions. In this case the power profile was very similar to the one obtained with the LAMBDA code. The NODAL-LAMBDA (NL) code was able to find the two identical subcritical eigenvalues that must appear by symmentry in the IAEA 3D PWR stability benchmark, also the PARCS with the modifications performed by Wysocki (Wysocki et al. 2014), and the LAMBDA code found this couple of identical subcritical eigenvalues. The subcriticalities computed with the NL code, without Chung-Turney correction in the albedo for Cofrentes 1991 instability event with a 32x32x27 mesh, agree pretty well with the results of the Lambda code using the same mesh and an expansion of order 3 (L3). For instance the first harmonic mode subcriticality was -1.089 $ computed with NL and -1.115$ with L3. The subcriticality computed for the second harmonic with NL was -1.133 $ and exactly the same value -1.133 $ with the L3 code. In this case the vector dimension was N in the NODAL-LAMBDA code and $10 \times N$ in L3, so the computing time was larger in L3 than in NL. The subcriticalities computed by both codes, NL and L3, also agree pretty well for Ringhals stability test at point 14G as displayed in tables 4.1 and 4.2. The Chung-Turney correction factor in the Albedo does not change significatively the subcriticalities because this correction factor changes both eigenvalues k_0 and k_i in the same direction and the changes in the subcriticalities are very small only 4 or 5 pcm as it is displayed in table 4.9. However increasing this factor to 1.2 yields better results for the power profile and the fundamental eigenmode of Cofrentes instability event.

Recently Vidal-Ferrandiz et al (2014) developped a finite elements code h-p to calculate the lambda modes, for the most refinesh mesh with p=2 order these authors obtained for the eigenvalue of the fundamental mode of the IAEA 3D PWR bechmark problem a value of 1.29923 using a number of DoF(Degrees of Freedom) of 193466, the error for this case was 89 pcm. The NODAL-LAMBDA code yields 1.02844 with an error of -59 pcm with 19652 unknowns, and the LAMBDA code yields an error of 35 pcm for L2 and only 9 pcm with L3 although the number of unkowns was 10 times bigger for this last case. Obviously increasing the order of p and with a finest mesh the error can be reduced but the computational cost increases. The same thing happens in NODAL-LAMBDA code using a finer mesh the results can be improved.

Acknowledments

The authors of this paper are indepted to IBERDROLA generación nuclear for provide support to perform this work.

References

- M.Abramowitz, A.Stegun, “Handbook of Mathematical Functions”, Editorial Dover (1972).
- Arnoldi, W.E.”The principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem”, Quarterly of Applied Mathematics, 9, pp 17-29, (1951)
- Arbenz, P., Kressner, D.,” Lecture Notes on Solving Large Scale Eigenvalue Problems”, ETH Zurich, 2012.
- Bernal, A., Miró, R., Ginestar, D., and Verdú, G. (2014), Resolution of the Generalized Eigenvalue Problem in the Neutron Diffusion Equation Discretized by the Finite Volume Method, Hindawi Publishing Corporation, *Abstract and Applied Analysis* Volume 2014, Article ID 913043, 15 pages http://dx.doi.org/10.1155/2014/913043

- Committee of Computational Benchmark Problems, “Benchmark Problem Book”. Report Argonne National Laboratory ANL-7416 Supplement 2 (UC-32), 1977.
- Borresen, S., “A Simplified Coarse-Mesh, 3D Diffusion Scheme for Calculating the Gross Power Distribution in a BWR” Nucl. Sci and Eng, Vol 44, pags 37-43 (1971)
- Chung, B.D., Kim, C.H., Chung, C.H. “Borresen Model for Neutronic Benchmark Problems”, Journal of the Korean Nuclear Society, Vol 13, Number 2, 1981.
- Cho B.O., “A Comparison of Three Coarse –Mesh Nodal Methods for BWR Analysis.” MSc Thesis Oregon State University 1988.
- Doring M.G., Kalkkuhl, “Subspace Iteration for Nonsymmetric Eigenvalue Problems Applied to the Lambda Eigenvalue Problem”, Nucl.Sci. and Eng. Vol 115, pp 244-252 (1993).
- Golub, G.H., Van Loan, C., “Matrix Computations”, Third edition, John Hopkins University Press, Baltimore, ISBN 0-8018-5414-8, 1996.
- Hashimoto, K., Hotta, A., Takeda, T., “Neutronic Model for Modal Multi-Channel Analysis of Out-of-Phase Instability in Boiling Water Reactor Cores”, Annals of Nuclear Energy, Vol 24, No 2, pp 99-111, (1997).
- Hébert, A., “Development of the nodal collocation method for solving the neutron diffusion equation”. Annals of Nucl. Energy , Vol 14, N0 10, pp 527-541 (1987).
- Kalambokas, C., “The Replacement of Reflectors by Albedo Type boundary Conditions”, Ph.D. Thesis MIT (1974)
- Lefvert, T., “Ringhals 1 Stability Benchmark, Specifications.” NEA/NSC/DOC(94)15, 1994.
- Lehoucq, R.B., Sorensen, D.C., Yang, C., “ARPACK user’s Guide”, Society for Industrial and Applied Mathematics SIAM, Philadelphia, 1998.
- Michelsen, Committee of Computational Benchmark Problems, “Benchmark Problem Book”. Report Argonne National Laboratory ANL-7416 Supplement 2 (UC-32), 1977
- Miró, R., “Métodos Modales para el Estudio de Inestabilidades en Reactores Nucleares BWR”, Tesis Doctoral Universidad Politécnica de Valencia 2002.
- Miró, R., Ginestar, Verdú, G., Henig, D., “A nodal modal method for the neutron diffusion equation. Application to BWR instabilities analysis”, Annals of Nuclear Energy, Vol 29, issue 10 , pp 1171-1194 (2002).
- Muñoz-Cobo, J.L., Roselló, O., Miró, R., Ginestar, D., Escrivá, A., Verdú, G., “Coupling of density wave oscillations with high order modal kinetics: application to B.W.R. out of phase oscillations “, Annals of Nuclear Energy 27, (15), pp 1347-1371 (2001).
- Petersen, C.Z., Vilhena, M.T., Moreira, D., Barros, D.C., “The Laplace transform method for the Albedo boundary conditions in neutron diffusion eigenvalue problems”, Published in Integral Method in Science and Engineering, Editorial Birkhauser, DOI 10.1007/978-0-8176-4897-8, 2010.

- Saad, Y., “Numerical Methods for Large Eigenvalue Problems”, Manchester University Press, Series in Algorithms and Architectures for Advanced Scientific Computing (1992).
- Singh, T., Mazumdar, T., Pandey, P., 2014. “Nemsqr: a 3-d multi group diffusion theory code based on the nodal expansion method for square geometry”. Ann. Nucl. Energy 64 (0), 230–243.
- Sorensen, D.C., “Implicit Application of Polynomial Filters in a k-stepArnoldi-Method”. SIAM Journal of Matrix Analysis. Appl. 13, pp 357-385 (1992).
- Turney, W.B. “Albedo Ajusted Fast Neutron Diffusion Coefficient in Reactor Reflectors”. Nuclear Science and Engineering, Vol 57, 239 (1975).
- Tyrtyshnikov E.E. “ A Brief Introduction to Numerical Analysis”, lectures 19 and 20, Birkhouser 1997.
- Vidal, V., “Métodos Numéricos para la Obtención de los modos Lambda de un Reactor Nuclear. Técnicas de Aceleración y Paralelización”. Tesis Doctoral Universidad Politécnica de Valencia 1997.
- Vidal-Ferrandiz, A., Fayez, R., Ginestar, D., Verdú, G. “Solution of the Lambda modes problem of a nuclear power reactor using an h–p finite element method”, Annals of Nuclear Energy Vol 72, pp 338-349 (2014).
- Verdú, G., Ginestar, D., Vidal, V., Muñoz-Cobo, J.L.”3D Lambda Modes of the Neutron Diffusion Equation” Annals of Nuclear Energy 21, 405-421, 1994.
- Verdú, G., Miró, R., Ginestar, D., Vidal, V. “The implicit restarted Arnoldi method, an efficient alternative to solve the neutron diffusion equation” Volume 26, Issue 7, May 1999, Pages 579–593.
- Wulff, W, Cheng, H.S., Diamond, D.J., Khatib-Rahbar, M., “A Description and Assesment of RAMONA 3D MoD 0 Cycle 4; A computer Code with 3D Neutron Kinetics for BWR System Transients.” NUREG/CR-3664 1984.
- Wysocki, A., March-Leuba, J., Manera, A., Downar, T., “TRACE/PARCS analysis of out-of-phase power oscillations with a rotating line of symmetry.” Annals of Nuclear Energy 67, 59-69, 2014.
- Wysocki, A., March-Leuba, J., Manera, A., Downar, T., “A Physical Mechanism for Rotating Lines of Symmetry in BWR Out-of-Phase Limit Cycle Oscillations.” 16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-16), Chicago, USA, 2015.