



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Approximation of the Neutron Diffusion Equation on Hexagonal Geometries

Author: Sebastián González Pintor
segonpin@isirymp.upv.es

Directors: Damián Ginestar Peiró
dginesta@mat.upv.es

Gumersindo Verdú Martín
gverdu@iqn.upv.es

Summary

The neutron diffusion equation describes the neutron population in a nuclear reactor core. This work deals with this model for nuclear reactors with hexagonal geometries. First, the stationary neutron diffusion equation is studied. This is a differential eigenvalue problem, called Lambda modes problem. To solve the Lambda modes problem, different methods have been compared in one-dimensional geometries, resulting as the best one the Spectral Element Method. The operators are then discretized using this scheme in two- and three-dimensional geometries, and the resulting algebraic eigenvalue problem is solved with the implicit restarted Arnoldi method.

Once the solution for the steady state neutron distribution is obtained, it is used as initial condition for the time integration neutron diffusion equation. Initially, a one step backward Euler method is used to discretize this equation in time. The transients to test the behaviour of the method are based on moving the control rods of the reactor, simulating an accident where a control rod is ejected and a scram is initialized to control the power evolution. An unphysical behaviour appears when a node is partially rodded, the rod cusping effect, which is corrected by weighting the cross sections with the flux of the previous time step. Good results are obtained for the power evolution and the rod cusping effect is corrected for different transients. To solve the algebraic systems arising in the backward method, a Krylov method is used, and different preconditioning strategies are tested. The first one consists of using the block structure obtained by the energy groups to solve the system by blocks, and different acceleration techniques for the block iterative scheme and a preconditioner using this block structure are proposed. Also, a spectral preconditioner, which makes use of the information in a Krylov subspace to preconditionate the next system is studied. Second and fourth order exponential methods are also proposed to integrate the time dependent neutron diffusion equation, where the exponential of the system matrix has to be multiplied by a vector. These schemes allow us to work without building explicitly the system matrix, and different methods are compared to calculate the product of the system matrix by a vector, such as a Krylov method, a Chebyshev approximation method, and a Leja Points method.

Some situations arise in which a set of modes of a nuclear reactor core have to be updated, as in perturbative calculations or in the use of modal methods. Updating this set of modes can be very expensive when using the Arnoldi method, and for this reason several methods based on a Newton Iteration are proposed, such as the Modified Block Newton method, a One Sided Block Newton method and a Two Sided Block Newton method. As an alternative strategy, a reduced order model to update a set of modes based on the Proper Generalized Decomposition is also proposed. This method obtains an approximated solution as a sum of separable functions over the whole domain, reducing the multidimensional problem to a set of one-dimensional problems.

Resumen

La ecuación de la difusión neutrónica describe la población de neutrones dentro de un reactor nuclear. Este trabajo está enfocado a reactores nucleares con geometría hexagonal. En primer lugar se estudia la ecuación de la difusión neutrónica estacionaria. Este es un problema diferencial de valores propios, llamado problema de los modos Lambda. Para resolver el problema de los modos Lambda se han comparado diferentes métodos en geometrías unidimensionales, resultando como el mejor, el método de elementos espectrales. Usando este método discretizamos los operadores en geometrías bidimensionales y tridimensionales, resolviendo el problema algebraico de valores propios resultante con el método de Arnoldi.

La distribución de neutrones en estado estacionario se utiliza como condición inicial para la integración de la ecuación de la difusión neutrónica dependiente del tiempo. Se utiliza un método de Euler implícito para integrar en el tiempo. Cuando una barra de control está parcialmente insertada en un nodo aparece un comportamiento no físico de la solución, el efecto “rod cusping”, que se corrige mediante la ponderación de las secciones eficaces con el flujo del paso de tiempo anterior. Para la resolución de los sistemas algebraicos que surgen en el método implícito, se utiliza un método de Krylov, y se evalúan diferentes estrategias de preconditionamiento. La primera consiste en el uso de la estructura a bloques definida por los grupos de energía para resolver el sistema. Además se proponen diferentes técnicas de aceleración para el esquema iterativo de bloques y un preconditionador utilizando esta estructura. Además se estudia un preconditionador espectral, que hace uso de la información del subespacio de Krylov obtenida cuando se resuelve un sistema para preconditionar el siguiente sistema. También se proponen métodos exponenciales de segundo y cuarto orden para integrar la ecuación de difusión neutrónica dependiente del tiempo, donde la exponencial de la matriz del sistema tiene que ser multiplicada por un vector. Estos esquemas nos permiten trabajar sin construir explícitamente la matriz del sistema, y se comparan diferentes métodos para aproximar el producto de la exponencial matricial por un vector, tal como un método de Krylov, un método de Chebyshev y un método basado en los puntos de Leja.

Surgen algunas situaciones en las que un conjunto de modos tiene que ser actualizado, como en los cálculos perturbados o en el uso de métodos modales. La actualización de estos de modos puede ser muy costosa cuando se utiliza el método de Arnoldi y, por esta razón, se proponen varios métodos basados en una iteración de Newton, tales como el método de Newton a bloques modificado, y dos alternativas basadas en iterar con uno o dos subespacios. Como estrategia alternativa se propone un modelo de orden reducido para actualizar un conjunto de modos basados en la “Proper Generalized Decomposition”. Este método obtiene una solución aproximada como una suma de funciones separables sobre todo el dominio, reduciendo el problema multidimensional a un conjunto de problemas unidimensionales.

Resum

L'equació de la difusió neutrònica descriu la població de neutrons d'un reactor nuclear. Aquest treball tracta amb aquest model per a reactors amb geometria hexagonal. En primer lloc, s'estudia la part estacionària de les equacions. Aquest és un problema diferencial de valors propis, anomenat problema dels modes Lambda. Per resoldre el problema dels modes Lambda, els diferents mètodes han estat comparats en geometries unidimensionals, resultant el millor el mètode d'elements espectrals. Els operadors són discretitzats fent servir aquest esquema en geometries bidimensionals i tridimensionals, i el problema de valors propis algebraics obtingut es resol amb el mètode d'Arnoldi amb reinici implícit.

Una vegada que s'obté la distribució de neutrons estacionària, aquesta s'utilitza com a condició inicial per l'integració de l'equació de difusió neutrònica en el temps. Inicialment, s'utilitza un mètode d'Euler implícit d'un pas per integrar en el temps. Els transitoris per comprovar el comportament del mètode es basen en moure les barres de control del reactor, simulant un accident on s'expulsa una barra de control i una parada d'emergència es inicialitzada per controlar l'evolució de potència. Un comportament no físic apareix quan un node té les barres de control parcialment insertades, l'efecte del "rod cusping", que es corregeix mitjançant la ponderació de les seccions eficaces amb el flux neutrònic del pas de temps anterior. Per a obtenir la solució dels sistemes algebraics que sorgeixen en el mètode Euler implícit, un mètode de Krylov s'utilitza per resoldre els sistemes resultants, i s'avaluen diferents estratègies de preconditionament. La primera consisteix en l'ús de l'estructura de blocs deguda als grups d'energia, i es proposen diferents tècniques d'acceleració per al esquema iteratiu per blocs i un preconditionador utilitzant aquesta estructura de blocs. A més, s'estudia un preconditionador espectral, que fa ús de la informació obtinguda del subespai de Krylov quan es resol un sistema per preconditionar el sistema següent. També es proposen mètodes exponencials de segon i quart ordre per integrar l'equació de difusió de neutrons depenent del temps, on l'exponencial de la matriu del sistema ha de ser multiplicada per un vector. Aquests esquemes ens permeten treballar sense construir explícitament la matriu del sistema, i es comparen diferents mètodes per calcular el producte de la matriu del sistema per un vector.

Sorgeixen algunes situacions en les qual un conjunt de modes ha de ser actualitzat, com en els càlculs de perturbació o en l'ús de mètodes modals. L'actualització d'aquesta sèrie de maneres pot ser molt costosa quan s'utilitza el mètode d'Arnoldi, i per això es proposen diversos mètodes basats en l'iteració de Newton, com ara el mètode modificat de Newton per blocs, i altres dues variants amb un o amb dos subespais. Com estratègia alternativa, es proposa un model d'ordre reduït per actualitzar un conjunt de modes basats en la "Proper Generalized Decomposition". Aquest mètode obté una solució aproximada com una suma de funcions separables sobre tot el domini, reduïnt el problema multi-dimensional a un conjunt de problemes uni-dimensionals.

Contents

General Index	VII
List of Figures	XI
List of Tables	XVII
1 Introduction	1
1.1 Neutron diffusion equation	2
1.2 Lambda Modes equation	13
1.3 VVER Reactors	14
1.4 Spectral Element Methods (SEM)	19
2 Spectral Element Method for the Neutron Diffusion Equation	23
2.1 One-dimensional geometries	24
2.1.1 Spectral Element Methods	26
2.1.2 Numerical results	35
2.2 Two- and three-dimensional geometries	41
2.2.1 Variational Formulation	43
2.2.2 Reference element	46
2.2.3 Polynomial basis	48
2.2.4 Building the matrices	52
2.2.5 Algebraic problem	55
2.2.6 Numerical Results	56

3	Time Dependent Neutron Diffusion Equation	81
3.1	Backward Method	82
3.1.1	Time discretization	83
3.1.2	Rod cusping	85
3.1.3	Numerical results	87
3.2	Block Preconditioning	94
3.2.1	Solvers for the linear systems	96
3.2.2	Numerical Results	99
3.3	Spectral Preconditioning	104
3.3.1	Spectral Preconditioner	105
3.3.2	Numerical Results	106
3.4	Exponential Method	111
3.4.1	Exponential operator and its Magnus Expansion	112
3.4.2	Implementation of the product matrix exponential - vector	120
3.4.3	Numerical Results	125
4	Updating Eigenvalue Methods	127
4.1	Modified Block Newton Method	128
4.1.1	Spatial discretization	129
4.1.2	Modified Block Newton Method	133
4.1.3	Numerical Results	135
4.2	Alternative Newton Methods	146
4.2.1	Modified Block Newton Method	147
4.2.2	One Sided Block Newton Method	148
4.2.3	Two Sided Block Newton Method	148
4.2.4	Numerical Results	149
4.3	Proper Generalized Decomposition for eigenvalue computations	153
4.3.1	Eigenvalue problem	157
4.3.2	Numerical Results	161
5	Conclusions	165
A	Special Functions	171
A.1	Legendre polynomials	172
A.2	Associated Legendre functions	173

A.3 Spherical Harmonics.	175
A.4 Relations	176
B Evaluation of the polynomials and the integrals	181
B.1 Evaluation of the modified Dubiner's polynomials	181
B.2 Evaluation of the integrals by means of quadrature rules	182
Bibliography	185

List of Figures

1.1	Core mesh for both VVER and PWR reactors.	15
1.2	Layout of the VVER 440 Core.	18
1.3	Layout of the VVER 1000 Core.	19
2.1	Discretization of a 1D reactor.	26
2.2	Hat functions.	34
2.3	Bubble functions.	34
2.4	Boundary functions.	35
2.5	Error for the first eigenvector of the homogeneous reactor.	37
2.6	Error for the second eigenvector of the homogeneous reactor.	38
2.7	Spatial distribution of materials for the 1D BWR reactor.	39
2.8	Mean square error for the power distribution of the 1D BWR reactor.	40
2.9	Spatial power distribution associated with the first two modes.	41
2.10	Core mesh for both VVER and PWR reactors.	42
2.11	Two-dimensional change of variables from a arbitrary triangle to the Reference Domain.	46
2.12	Three-dimensional change of variables from a arbitrary prism to the Reference domain.	47
2.13	Modified Dubiner's polynomials for $K = 4$	48
2.14	Numbering the Modified Dubiner's Polynomials.	50

2.15	Continuity conditions for vertex and edge polynomials for $K = 2$. . .	51
2.16	Vertex polynomials associated with an internal vertex and their coefficients. The corresponding global vertex coefficient is shown inside a circle.	53
2.17	Edge polynomials of third order (left) and fourth order (right) whose coefficients are fixed to a global edge coefficient (inside a circle) and their corresponding coefficients.	54
2.18	Interior polynomials of different order whose coefficients are fixed to a global interior coefficient (inside a circle) and their corresponding coefficients.	54
2.19	Vertex (left) and edge (right) polynomials and their corresponding coefficients, which are fixed to a global boundary coefficient (inside a circle).	55
2.20	Geometry of the IAEA without reflector problem.	57
2.21	Power distribution for the IAEA problem without reflector.	59
2.22	Symmetry pattern for the four dominant modes of the IAEA problem without reflector.	60
2.23	IAEA problem with reflector geometry.	61
2.24	Power distribution for the IAEA problem with reflector.	61
2.25	Symmetry pattern for the four dominant modes of the IAEA problem with reflector.	62
2.26	Geometry of the VVER-1000 reactor.	63
2.27	Power distribution for the VVER-1000 reactor.	64
2.28	Symmetries of the four dominant modes for the VVER-1000 problem.	65
2.29	Geometry of the reactor VVER-440.	66
2.30	Power distribution for the VVER-440 problem.	67
2.31	Symmetry patterns of the four dominant modes for the VVER-440 problem.	68
2.32	Geometry of the HWR reactor	69
2.33	Power distribution for the HWR problem.	71
2.34	Symmetry patterns of the four dominant modes for the HWR problem.	72

2.35	Geometry of the 3D VVER-440 problem with reflector	73
2.36	Modes for the VVER-440 problem	75
2.37	Numbers identifying the position of the cells of the VVER 440 reactor for the local power distribution	75
2.38	Geometry of the 3D VVER-1000 problem	77
2.39	Modes for the VVER-1000 problem	78
2.40	Numbers identifying the position of the cells of the VVER 1000 reactor for the local power distribution	79
3.1	Rod cusping effect.	85
3.2	Partially Rodded Node.	86
3.3	Geometry of VVER 440 reactor core.	87
3.4	Results at $t = 0$ s.	89
3.5	Average power evolution.	90
3.6	Results at $t = 6$ s.	90
3.7	Small Reactor	91
3.8	Evolution of the k_{eff} (LEFT); Detailed view of the first 0.75 s (RIGHT).	93
3.9	Absolute error respect to the method (LEFT); Detailed view (RIGHT).	93
3.10	Evolution of the power along the transient (LEFT); Detailed view (RIGHT).	94
3.11	VVER 440 2D.	100
3.12	Power evolution for the VVER 440 2D transient.	101
3.13	Geometry of VVER 440 reactor core.	102
3.14	Small reactor.	107
3.15	Power along the transient.	107
3.16	Influence of the rank of the subspace, k , for a fixed $L_{\text{max}} = 25$. (spatial discretization with polynomial degree 1, $N = 1898$, $NNZ = 65860$).	108

3.17	Influence of the number of terms, L_{max} , for a fixed $k = 3$. (spatial discretization with polynomial degree 1, $N = 1898$, $NNZ = 65860$).	109
3.18	Influence of the rank of the subspace, k , for a fixed $L_{max} = 25$. (spatial discretization with polynomial degree 2, $N = 12950$, $NNZ = 1064284$).	109
3.19	Influence of the number of terms, L_{max} , for a fixed $k = 3$. (spatial discretization with polynomial degree 2, $N = 12950$, $NNZ = 1064284$).	110
3.20	Small reactor.	125
4.1	Geometry of the VVER-1000 reactor.	137
4.2	Shapes of the power distributions associated with the first two dominant modes of the VVER-1000 problem for the initial configuration (left) and the perturbed one (right).	138
4.3	VVER 440 2D.	139
4.4	Shapes of the normalised power distributions associated with the first three dominant modes of the initial configuration (left) and perturbed configuration (right) of reactor VVER-440.	141
4.5	Geometry of the 3D IAEA without reflector problem.	142
4.6	Shapes of the normalised power distributions of the first two dominant Lambda modes of the initial and the perturbed configuration of 3D IAEA reactor.	144
4.7	Shapes of the normalised power distributions of the first two dominant Lambda modes of the initial and the perturbed configuration of BWR reactor.	146
4.8	Radial layout for the fundamental (left) and for the first subcritical mode (right) for the initial configuration (without Boron) and the perturbed configuration (with Boron).	147
4.9	Convergence for the first 3 eigenvalues from 0 to 1500 ppm of boron.	151
4.10	Convergence for the first 3 eigenvalues from 750 to 1500 ppm of boron.	152
4.11	Geometry of reactor Biblis 2D (a) and Relative error evolution for the eigenvalue λ (b)	162
4.12	Reference solution and the initial approximation ($k = 1$) for the power distribution.	163

4.13 Error and corrections for the power distribution for $k = 1$ and $k = 2$ 163

A.1 Magnitudes for the Addition theorem for the Legendre polynomials. 174

B.1 Change of variables mapping the reference domain into the rectangle
 $[-1, 1] \times [-1, 1]$ 183

List of Tables

2.1	Results for λ_1 and λ_2 for the homogeneous eigenvalue problem. . .	37
2.2	Nuclear cross sections for the 1D BWR reactor.	39
2.3	1D BWR results for $\lambda_1(k_{\text{eff}})$ and λ_2	40
2.4	Cross sections of the IAEA problem.	58
2.5	k_{eff} results for the IAEA reactor without reflector.	58
2.6	First 3 subcritical eigenvalues for the IAEA problem without reflector.	59
2.7	k_{eff} results for the IAEA reactor with reflector.	59
2.8	First 3 subcritical eigenvalues for the IAEA problem with reflector.	61
2.9	Cross sections of the VVER-1000 problem	63
2.10	k_{eff} results for the VVER-1000 problem.	63
2.11	First 3 subcritical eigenvalues for the VVER-1000 reactor.	64
2.12	Cross sections of the VVER-440 problem.	66
2.13	k_{eff} results for the VVER-440 problem.	67
2.14	First 3 subcritical eigenvalues for the VVER-440 reactor.	67
2.15	Cross sections of the HWR reactor.	70
2.16	k_{eff} results for the HWR reactor.	70
2.17	First 3 subcritical eigenvalues for the HWR reactor.	70
2.18	Cross sections of the three-dimensional VVER-440 problem.	73
2.19	Results for the k_{eff} for the 3D VVER 440 problem	74

2.20	3 first subcritical eigenvalues for the 3D VVER 440 problem	74
2.21	Local power for the 3D VVER 440 problem	76
2.22	Cross sections of the 3D VVER-1000 problem	77
2.23	Results for the k_{eff} for the 3D VVER-1000 problem	77
2.24	3 first subcritical eigenvalues for the 3D VVER-1000 problem . . .	78
2.25	Local power for the 3D VVER 1000 problem	79
3.1	Cross sections of the three-dimensional VVER-440 transient. . . .	88
3.2	Neutron precursors parameters for the VVER 440 transient. . . .	88
3.3	Steady state k_{eff} for different degrees on xy polynomials.	89
3.4	Cross sections (in cm^{-1}) for the 3-D small reactor.	91
3.5	Neutron precursors parameters for the reactor.	92
3.6	Cross sections for the 2-D VVER 440 reactor.	100
3.7	Neutron precursors parameters for the reactor VVER 440.	101
3.8	Number of iterations and CPU time for matrices obtained with $K_{xy} = 5$	102
3.9	Number of iterations and CPU time for matrices obtained with $K_{xy} = 1$ and $K_z = 1$	103
3.10	Number of iterations and CPU time for matrices obtained with $K_{xy} = 2$ and $K_z = 2$	103
3.11	Iterations count for the second order scheme with different expo- nential approximations	125
3.12	Iterations count for the fourth order scheme with different exponen- tial approximations	126
4.1	Cross sections of the VVER-1000 problem.	137
4.2	Eigenvalues of the initial and perturbed configurations of VVER- 1000 reactor.	137
4.3	Evolution of the residual in the problem VVER-1000 for different steps of the Newton method and different number of modes.	139

4.4	Cross sections for the 2-D VVER 440 reactor.	140
4.5	Eigenvalues for the initial and perturbed configurations of reactor VVER 440.	140
4.6	Residual evolution for different steps of the Newton method for different number of modes of reactor VVER-440.	141
4.7	Cross sections of 3D IAEA problem.	142
4.8	First three dominant eigenvalues of 3D IAEA reactor.	143
4.9	Residual for different steps of Newton's method to update the modes of 3D IAEA reactor.	143
4.10	Eigenvalues associated with the two configurations of the BWR reactor.	145
4.11	Residual evolution for different steps of the Newton method to update the modes of the BWR reactor.	145
4.12	Eigenvalues for three boron concentrations.	150
4.13	Iterations to converge and final relative residual from 0 to 1500 ppm.	151
4.14	Iterations to converge and final relative residual from 750 to 1500 ppm.	152
4.15	Cross sections of 2-dimensional Biblis problem.	161

Chapter 1

Introduction

Contents

1.1 Neutron diffusion equation	2
1.2 Lambda Modes equation	13
1.3 VVER Reactors.	14
1.4 Spectral Element Methods (SEM).	19

In nuclear reactor core physics two main objects to study can be distinguished. On one hand, the localisation and dynamics (or behaviour) of neutrons and on the other hand, the criticality of the reactor, i.e., whether it is possible to maintain the chain reaction in its interior. Among the many different methods that exist to answer these questions, the most commonly used ones are based on the neutron diffusion theory, which is an approximation to the neutron transport theory.

The rigorous treatment of this problem is completely analogous to that used in classic studies of gaseous diffusion [1]. The method of study consists of taking a control volume at some point of the reactor, and deriving expressions that account for the different ways of entry and exit from this control volume of neutrons, having a velocity vector given by introducing effective cross sections avoiding to consider specific interactions of neutrons within the control volume. The balance

between proportion of neutrons which enter and the proportion of neutrons which exit the control volume, gives rise to what is known as the Boltzmann transport equation. To simplify this equation, we introduce hypothesis such as that all the neutrons can be grouped into energy ranges (groups) and that the distribution of the neutron velocity vectors is independent of the direction. Under these conditions, one can obtain the neutron diffusion equation as an approximation of the transport equation.

There are mainly two types of calculations associated with the neutron diffusion. A first type of static calculations involving the determination of the Lambda modes associated with a given configuration of the reactor in a given time. This is a generalized eigenvalue problem associated with a differential operator with given boundary conditions. The determination of the fundamental mode allows us to describe the behaviour of the reactor in steady state. Another type of calculations are those made for the determination of a transient from a perturbation made on a stationary configuration of the reactor, using for that the time dependent neutron diffusion equation.

Several algorithms to integrate the time dependent neutron diffusion equation have been developed, by means of modal methods [2, 3] or implicit methods for the time integration scheme. Moreover some paralelization approaches have been proposed for these algorithms [4], making these methods capables to calculate real reactors with reasonable timings. The algebraic part for the resolution of the eigenvalue problems has been also solved in several ways [5, 6].

The present memory is focused on developing methods to deal with these two problems on hexagonal geometries. The developed methods are mainly aimed to be used with a spatial discretization or spatial mesh consisting on large elements. Numerical results are reported about “benchmark” problems found at the literature.

1.1 Neutron diffusion equation

The behaviour of a nuclear reactor is modelled by means of the neutron distribution at the reactor as a function depending on the position, the velocity, the energy and the time. Thus, one of the main problems for the nuclear reactor theory is to predict this distribution. It can be done solving the neutron transport equation over the reactor domain. But, due to the complexity of this equation, the neutron diffusion equation is widely used as an approximation [7]. We will proceed to show the process followed to deduce the neutron diffusion equation as an approximation for the transport equation.

Within the transport theory the neutron is considered as a classical particle, in the sense that it can be fully determined by means of knowing its position and its velocity. The interaction between neutrons and atomic nuclei is dealt from a

macroscopic point of view, avoiding the details of the interaction process inside the core. Moreover, the cross sections associated to the probability of a determined type of reaction is defined and it is supposed that this interaction is produced instantly.

The balance equation in a differential control volume is obtained taking into account that the rate of change for the density respect to the time inside the volume $dVdEd\vec{\Omega}$ of the phase space is given by the difference between the neutron rate flow into and out of that control volume. To describe the neutron population the magnitude denoted as *neutron angular density*, $N(\vec{r}, E, \vec{\Omega}, t)$, is used, defined as the probable (or expected) number of neutrons at position \vec{r} with direction $\vec{\Omega}$ and energy E at time t , per unit volume per unit solid angle per unit energy. Moreover, the *neutron angular flux* is defined as

$$\Psi(\vec{r}, E, \vec{\Omega}, t) \equiv vN(\vec{r}, E, \vec{\Omega}, t),$$

where v is the neutron speed, and the balance equation inside the control volume which describes the *neutron transport equation* is expressed as follows [7]

$$\begin{aligned} \frac{1}{v} \frac{\partial \Psi}{\partial t}(\vec{r}, E, \vec{\Omega}, t) &= -\vec{\Omega} \cdot \vec{\nabla} \Psi(\vec{r}, E, \vec{\Omega}, t) \\ &- \Sigma_T(\vec{r}, E, t) \Psi(\vec{r}, E, \vec{\Omega}, t) + Q(\vec{r}, E, \vec{\Omega}, t) \\ &+ (1 - \beta) \frac{\chi_p(E)}{4\pi} \int_0^\infty dE' \nu \Sigma_f(\vec{r}, E', t) \int_{\vec{\Omega}'} d\vec{\Omega}' \Psi(\vec{r}, E', \vec{\Omega}', t) \\ &+ \int_0^\infty dE' \int_{\vec{\Omega}'} d\vec{\Omega}' \Sigma_s(\vec{r}; E', \vec{\Omega}' \rightarrow E, \vec{\Omega}; t) \Psi(\vec{r}, E', \vec{\Omega}', t) \\ &+ \sum_{k=1}^K \lambda_k \frac{\chi_k(E)}{4\pi} C_k(\vec{r}, t). \end{aligned} \quad (1.1)$$

The first term on the right hand of the equation, $\vec{\Omega} \cdot \vec{\nabla} \Psi(\vec{r}, E, \vec{\Omega}, t)$, takes into account the neutron advection flowing out the control volume, where $\vec{\Omega}$ is the unit vector denoting the direction. The second term, $\Sigma_T(\vec{r}, E, t) \Psi(\vec{r}, E, \vec{\Omega}, t)$, describes the rate at which neutrons flow out the control volume by means of scattering and absorption processes. $Q(\vec{r}, E, \vec{\Omega}, t)$ denotes a possible external source of neutrons. The fourth term indicates the neutrons introduced into the volume element by fission processes, assuming isotropic fission distribution. The fifth term describes the neutrons introduced into the volume element by scattering. The *delayed neutrons* appearing in the volume from the precursor's decay is taken into account by means of the last term.

Σ_T and Σ_f denote the *total cross section* and the *fission cross section* respectively. χ_p indicates the spectrum of the neutron produced by fission and χ_k is the spectrum of neutrons produced by the precursors decay. The probability for a neutron to be scattered from a volume $dVdE'd\vec{\Omega}'$ to other $dVdEd\vec{\Omega}$, is represented by $\Sigma_s(\vec{r}; E', \vec{\Omega}' \rightarrow E, \vec{\Omega}; t)$. The fission neutron rate scattered due to the

transformation of a precursor of type k is β_k , where $\beta = \sum_{k=1}^K \beta_k$ and K is the number of neutron precursors considered. The decay rate for which a precursor of type k decays is $\lambda_k C_k$.

The concentration of delayed neutron precursors satisfies the following balance equation

$$\frac{\partial C_k}{\partial t}(\vec{r}, t) = \beta_k \int_0^\infty dE \int_{\vec{\Omega}} d\vec{\Omega} \nu \Sigma_f(\vec{r}, E, t) \Psi(\vec{r}, E, \vec{\Omega}, t) - \lambda_k C_k(\vec{r}, t), \quad (1.2)$$

where ν is the average number of neutrons arising in one fission, and $k = 1, \dots, K$.

It is assumed that the angular dependency for the neutron scattering is mainly due to the angle between the direction of the incident neutron, $\vec{\Omega}'$, and the direction of the emerging neutron, $\vec{\Omega}$. It is defined

$$\begin{aligned} \vec{\Omega} &= (\Omega_1, \Omega_2, \Omega_3) = (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta), \\ \vec{\Omega}' &= (\Omega'_1, \Omega'_2, \Omega'_3) = (\sin \theta' \cos \varphi', \sin \theta' \sin \varphi', \cos \theta'), \\ \mu^* &= \vec{\Omega}' \cdot \vec{\Omega} = \cos \theta^*, \quad \mu = \cos \theta, \quad \mu' = \cos \theta', \end{aligned}$$

where θ and θ' are the angles of directions $\vec{\Omega}$ and $\vec{\Omega}'$, respectively, with the z axis of the chosen reference system (see Figure A.1 at the Appendix).

The spherical harmonics method to approximate the neutron transport equation consists of expanding the angular neutron flux, $\Psi(\vec{r}, E, \vec{\Omega}, t)$, and the external neutron source, $Q(\vec{r}, E, \vec{\Omega}, t)$, as follows

$$\Psi(\vec{r}, E, \vec{\Omega}, t) = \sum_{l=0}^{\infty} \sum_{m=-l}^{+l} \Phi_l^m(\vec{r}, E, t) Y_l^m(\vec{\Omega}), \quad (1.3)$$

$$Q(\vec{r}, E, \vec{\Omega}, t) = \sum_{l=0}^{\infty} \sum_{m=-l}^{+l} Q_l^m(\vec{r}, E, t) Y_l^m(\vec{\Omega}), \quad (1.4)$$

where $Y_l^m(\vec{\Omega})$ are the (normalized) spherical harmonics defined at (A.9). It is assumed that scattering only depends on the relative angle, $\mu^* = \vec{\Omega} \cdot \vec{\Omega}'$, and that the scattering cross section can be expanded as the following Legendre polynomials series:

$$\Sigma_s(\vec{r}; E', \vec{\Omega}' \rightarrow E, \vec{\Omega}; t) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) P_l(\mu^*),$$

where P_l are the Legendre polynomials. Making use of the addition theorem for the Legendre polynomials (A.8), it can be written

$$\begin{aligned}
 & \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) P_l(\mu^*) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) \left(P_l(\mu) P_l(\mu') \right. \\
 & + 2 \sum_{m=1}^l \frac{(l-m)!}{(l+m)!} P_l^m(\mu) P_l^m(\mu') \cos m(\varphi - \varphi') \Big) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) \\
 & \times \left(P_l(\mu) P_l(\mu') + 2 \sum_{m=1}^l \frac{(l-m)!}{(l+m)!} P_l^m(\mu) P_l^m(\mu') \left(\frac{e^{im\varphi} e^{-im\varphi'} + e^{-im\varphi} e^{im\varphi'}}{2} \right) \right) \\
 & = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) \left(\frac{4\pi}{2l+1} Y_l^0(\vec{\Omega}) Y_l^0(\vec{\Omega}') \right. \\
 & + \sum_{m=1}^l \frac{(l-m)!}{(l+m)!} P_l^m(\mu) P_l^m(\mu') e^{im\varphi} e^{-im\varphi'} \\
 & + \left. \sum_{m=1}^l \frac{(l-m)!}{(l+m)!} \frac{(l+m)!}{(l-m)!} \frac{(l+m)!}{(l-m)!} P_l^{-m}(\mu) P_l^{-m}(\mu') e^{-im\varphi} e^{im\varphi'} \right) \\
 & = \sum_{l=0}^{\infty} \sum_{m=-l}^{+l} \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) Y_l^m(\vec{\Omega}) Y_l^{m*}(\vec{\Omega}') ,
 \end{aligned}$$

finally obtaining the equality

$$\Sigma_s(\vec{r}; E', \vec{\Omega}' \rightarrow E, \vec{\Omega}; t) = \sum_{l=0}^{\infty} \sum_{m=-l}^{+l} \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) Y_l^m(\vec{\Omega}) Y_l^{m*}(\vec{\Omega}') . \quad (1.5)$$

It will be assumed the external neutron source to be isotropic, i.e., $S_l^m = 0$ but for S_0^0 . Due to this assumption equation (1.4) becomes

$$Q(\vec{r}, E, \vec{\Omega}, t) = Q_0^0(\vec{r}, E, t) Y_0^0(\vec{\Omega}) . \quad (1.6)$$

For the P_L approximation, previous series are truncated for a given value $l = L$, taking into account that when $L \rightarrow \infty$ we recover the exact solution.

Now the identities (1.3), (1.5) and (1.6) are substituted in equation (1.1), obtaining the following equation

$$\begin{aligned}
 & \sum_{l=0}^L \sum_{m=-l}^{+l} Y_l^m(\vec{\Omega}) \left(\frac{1}{v} \frac{\partial \Phi_l^m}{\partial t}(\vec{r}, E, t) + \vec{\Omega} \cdot \vec{\nabla} \Phi_l^m(\vec{r}, E, t) + \Sigma_T(\vec{r}, E, t) \Phi_l^m(\vec{r}, E, t) \right. \\
 & \left. - \int_0^\infty \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) \sum_{\alpha=0}^\infty \sum_{\gamma=-\alpha}^{+\alpha} \int_{\Omega'} Y_\alpha^\gamma(\vec{\Omega}') Y_l^{m*}(\vec{\Omega}') d\Omega' \Phi_\alpha^\gamma(\vec{r}, E', t) dE' \right) \\
 & - (1 - \beta) \frac{\chi_p(E)}{4\pi} \int_0^\infty \nu \Sigma_f(\vec{r}, E', t) \sum_{l=0}^L \sum_{m=-l}^{+l} \Phi_l^m(\vec{r}, E', t) \int_{\Omega'} Y_l^m(\vec{\Omega}') d\Omega' dE' \\
 & - Q_0^0(\vec{r}, E, t) Y_0^0(\vec{\Omega}) - \sum_{k=1}^K \lambda_k \frac{\chi_k(E)}{4\pi} \mathcal{C}_k(\vec{r}, t) = 0. \quad (1.7)
 \end{aligned}$$

Then integrals with respect to $\vec{\Omega}'$ are solved using the orthogonality properties of the spherical harmonics (A.11), obtaining

$$\begin{aligned}
 & \sum_{\alpha=0}^\infty \sum_{\gamma=-\alpha}^{+\alpha} \left(\int_{\Omega'} Y_\alpha^\gamma(\vec{\Omega}') Y_l^{m*}(\vec{\Omega}') d\Omega' \right) \Phi_\alpha^\gamma(\vec{r}, E', t) \\
 & = \sum_{\alpha=0}^\infty \sum_{\gamma=-\alpha}^{+\alpha} \delta_\gamma^m \delta_\alpha^l \Phi_\alpha^\gamma(\vec{r}, E', t) = \Phi_l^m(\vec{r}, E', t), \quad (1.8)
 \end{aligned}$$

$$\sum_{l=0}^\infty \sum_{m=-l}^{+l} \int_{\Omega'} Y_l^m(\vec{\Omega}') d\Omega' \Phi_l^m(\vec{r}, E', t) = \frac{\Phi_0^0(\vec{r}, E', t)}{H_0^0}, \quad (1.9)$$

where H_l^m is defined at equation (A.10), and substituting equation (1.8) and (1.9) in the equation (1.7), it is obtained

$$\begin{aligned}
 & \sum_{l=0}^L \sum_{m=-l}^{+l} Y_l^m(\vec{\Omega}) \left(\frac{1}{v} \frac{\partial \Phi_l^m}{\partial t}(\vec{r}, E, t) + \vec{\Omega} \cdot \vec{\nabla} \Phi_l^m(\vec{r}, E, t) \right. \\
 & \left. + \Sigma_T(\vec{r}, E, t) \Phi_l^m(\vec{r}, E, t) - \int_0^\infty dE' \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) \Phi_l^m(\vec{r}, E', t) \right) \\
 & - (1 - \beta) \frac{\chi_p(E)}{4\pi} \int_0^\infty dE' \nu \Sigma_f(\vec{r}, E', t) \frac{\Phi_0^0(\vec{r}, E', t)}{H_0^0} \\
 & - Q_0^0(\vec{r}, E, t) Y_0^0(\vec{\Omega}) - \sum_{k=1}^K \lambda_k \frac{\chi_k(E)}{4\pi} \mathcal{C}_k(\vec{r}, t) = 0. \quad (1.10)
 \end{aligned}$$

Now the objective is to decouple the equation (1.10) into a system of partial differential equations, weighted by a set of spherical harmonics. First, the operator $\vec{\Omega} \cdot \vec{\nabla}$

is rewritten by means of the expression for $Y_l^m(\vec{\Omega})(\vec{\Omega} \cdot \vec{\nabla})$ of the identity (A.21) of the Appendix A.4. Thus, equations will be decoupled substituting at equation (1.10), multiplying by $Y_\alpha^{\gamma*}(\vec{\Omega})$ and integrating for all direction, making use of the orthogonality property for the spherical harmonics (A.11), it is obtained

$$\begin{aligned}
 & \frac{1}{v} \frac{\partial \Phi_\alpha^\gamma}{\partial t}(\vec{r}, E, t) \\
 & + \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) \left(A_1^{\alpha-1, \gamma-1} \Phi_{\alpha-1}^{\gamma-1}(\vec{r}, E, t) - A_2^{\alpha+1, \gamma-1} \Phi_{\alpha+1}^{\gamma-1}(\vec{r}, E, t) \right) \\
 & + \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) \left(-A_3^{\alpha-1, \gamma+1} \Phi_{\alpha-1}^{\gamma+1}(\vec{r}, E, t) + A_4^{\alpha+1, \gamma+1} \Phi_{\alpha+1}^{\gamma+1}(\vec{r}, E, t) \right) \\
 & + \frac{\partial}{\partial z} \left(A_5^{\alpha+1, \gamma} \Phi_{\alpha+1}^\gamma(\vec{r}, E, t) + A_6^{\alpha-1, \gamma} \Phi_{\alpha-1}^\gamma(\vec{r}, E, t) \right) \\
 & + \Sigma_T(\vec{r}, E, t) \Phi_\alpha^\gamma(\vec{r}, E, t) - \int_0^\infty dE' \Sigma_{sl}(\vec{r}; E' \rightarrow E; t) \Phi_\alpha^\gamma(\vec{r}, E', t) \\
 & - (1 - \beta) \frac{\chi_p(E)}{\sqrt{4\pi}} \int_0^\infty dE' \nu \Sigma_f(\vec{r}, E', t) \frac{\Phi_0^0(\vec{r}, E', t)}{H_0^0} H_0^0 \delta_0^\alpha \delta_0^\gamma \\
 & - \frac{Q_0^0(\vec{r}, E, t)}{H_0^0} H_0^0 \delta_0^\alpha \delta_0^\gamma - \sum_{k=1}^K \lambda_k \frac{\chi_k(E)}{4\pi} C_k(\vec{r}, t) H_0^0 \delta_0^\alpha \delta_0^\gamma = 0, \\
 & \alpha = 0, 1, \dots, L, \quad \gamma = -\alpha, \dots, \alpha.
 \end{aligned}$$

where constants $A_n^{l,m}$, $n = 1, \dots, 6$ are defined at the Appendix A.4 (equations (A.22)), and the terms with invalid values for α and γ will be assumed to be 0.

To obtain the P_1 approximation, $L = 1$ is taken in the previous system (1.11), and in the series expansion for the angular flux the remaining terms are Φ_0^0 , Φ_1^{-1} , Φ_1^0 and Φ_1^1 , and all the coefficients of higher order are considered equal to zero. The following notation is introduced

$$\begin{aligned}
 \frac{Q_0^0(\vec{r}, E, t)}{H_0^0} &= \tilde{Q}(\vec{r}, E, t), \\
 \frac{\Phi_0^0(\vec{r}, E, t)}{H_0^0} &= \Phi(\vec{r}, E, t),
 \end{aligned} \tag{1.11}$$

where \tilde{Q} is a mean neutron source, and Φ is the scalar neutron flux. Imposing the following conditions

$$\frac{\partial \Phi_1^{-1}}{\partial t}(\vec{r}, E, t) = \frac{\partial \Phi_1^0}{\partial t}(\vec{r}, E, t) = \frac{\partial \Phi_1^1}{\partial t}(\vec{r}, E, t) = 0,$$

it is obtained the following system of equations:

$$\begin{aligned}
 -\frac{1}{v} \frac{\partial \Phi}{\partial t}(\vec{r}, E, t) &= -\sqrt{\frac{1}{6}} \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) \frac{\Phi_1^{-1}(\vec{r}, E, t)}{H_0^0} \\
 &+ \sqrt{\frac{1}{6}} \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) \frac{\Phi_1^1(\vec{r}, E, t)}{H_0^0} \\
 &+ \sqrt{\frac{1}{3}} \left(\frac{\partial}{\partial z} \right) \frac{\Phi_1^0(\vec{r}, E, t)}{H_0^0} + \Sigma_T(\vec{r}, E, t) \Phi(\vec{r}, E, t) \\
 &- \int_0^\infty dE' \Sigma_{s0}(\vec{r}; E' \rightarrow E; t) \Phi(\vec{r}, E', t) \\
 &- (1 - \beta) \frac{\chi_p(E)}{\sqrt{4\pi}} \int_0^\infty dE' \nu \Sigma_f(\vec{r}, E', t) \Phi(\vec{r}, E', t) \\
 &- \tilde{Q}(\vec{r}, E, t) - \sum_{k=1}^K \lambda_k \frac{\chi_k(E)}{4\pi} \mathcal{C}_k(\vec{r}, t), \tag{1.12}
 \end{aligned}$$

$$\begin{aligned}
 \sqrt{\frac{1}{6}} \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) \Phi_0^0(\vec{r}, E, t) &= \Sigma_T(\vec{r}, E, t) \Phi_1^{-1}(\vec{r}, E, t) \\
 &- \int_0^\infty dE' \Sigma_{s1}(\vec{r}; E' \rightarrow E; t) \Phi_1^{-1}(\vec{r}, E', t), \tag{1.13}
 \end{aligned}$$

$$\begin{aligned}
 -\sqrt{\frac{1}{3}} \left(\frac{\partial}{\partial z} \right) \Phi_0^0(\vec{r}, E, t) &= \Sigma_T(\vec{r}, E, t) \Phi_1^0(\vec{r}, E, t) \\
 &- \int_0^\infty dE' \Sigma_{s1}(\vec{r}; E' \rightarrow E; t) \Phi_1^0(\vec{r}, E', t), \tag{1.14}
 \end{aligned}$$

$$\begin{aligned}
 -\sqrt{\frac{1}{6}} \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) \Phi_0^0(\vec{r}, E, t) &= \Sigma_T(\vec{r}, E, t) \Phi_1^1(\vec{r}, E, t) \\
 &- \int_0^\infty dE' \Sigma_{s1}(\vec{r}; E' \rightarrow E; t) \Phi_1^1(\vec{r}, E', t). \tag{1.15}
 \end{aligned}$$

Moreover, it is supposed that the inelastic neutron scattering is isotropic, which implies that Σ_{s1} only describes the elastic scattering. It is considered that the anisotropic elastic scattering is done without change in the neutron's energy, and then it can be written

$$\Sigma_{s1}(\vec{r}, E' \rightarrow E) J(\vec{r}, E') dE' = \Sigma_{s1}(\vec{r}, E \rightarrow E') J(\vec{r}, E) dE'$$

where the right-hand side can also be rewritten as

$$\Sigma_{s1}(\vec{r}, E) J(\vec{r}, E)$$

with

$$\Sigma_{s1}(\vec{r}, E) = \int_0^\infty \Sigma_{s1}(\vec{r}, E \rightarrow E') dE'$$

The transport cross section is defined as

$$\Sigma_{tr}(\vec{r}, E, t) = \Sigma_T(\vec{r}, E, t) - \bar{\Sigma}_{s1}(\vec{r}, E, t) ,$$

and the diffusion coefficient

$$D(\vec{r}, E, t) = \frac{1}{3\Sigma_{tr}(\vec{r}, E, t)} .$$

Now it will be used the newly defined terms to write equations (1.13), (1.14) and (1.15) as follows

$$\Phi_1^{-1}(\vec{r}, E, t) = \sqrt{\frac{3}{2}} D(\vec{r}, E, t) \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) \Phi_0^0(\vec{r}, E, t) , \quad (1.16)$$

$$\Phi_1^0(\vec{r}, E, t) = -\sqrt{3} D(\vec{r}, E, t) \left(\frac{\partial}{\partial z} \right) \Phi_0^0(\vec{r}, E, t) , \quad (1.17)$$

$$\Phi_1^1(\vec{r}, E, t) = -\sqrt{\frac{3}{2}} D(\vec{r}, E, t) \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) \Phi_0^0(\vec{r}, E, t) . \quad (1.18)$$

Substituting equations (1.16), (1.17) and (1.18) in the equation (1.12) to obtain a formulation depending only on Φ_0^0 , and then using the definition (1.11) of Φ , it is obtained

$$\begin{aligned} -\frac{1}{v} \frac{\partial \Phi}{\partial t}(\vec{r}, E, t) &= -\frac{\partial}{\partial x} D(\vec{r}, E, t) \frac{\partial}{\partial x} \Phi(\vec{r}, E, t) \\ &- \frac{\partial}{\partial y} D(\vec{r}, E, t) \frac{\partial}{\partial y} \Phi(\vec{r}, E, t) \\ &- \frac{\partial}{\partial z} D(\vec{r}, E, t) \frac{\partial}{\partial z} \Phi(\vec{r}, E, t) + \Sigma_T(\vec{r}, E, t) \Phi(\vec{r}, E, t) \\ &- \int_0^\infty dE' \Sigma_{s0}(\vec{r}; E' \rightarrow E; t) \Phi(\vec{r}, E', t) \\ &- (1 - \beta) \frac{\chi_p(E)}{\sqrt{4\pi}} \int_0^\infty dE' \nu \Sigma_f(\vec{r}, E', t) \Phi(\vec{r}, E', t) \\ &- \tilde{Q}(\vec{r}, E, t) - \sum_{k=1}^K \lambda_k \frac{\chi_k(E)}{4\pi} C_k(\vec{r}, t), \end{aligned} \quad (1.19)$$

i.e.,

$$\begin{aligned}
 -\frac{1}{v} \frac{\partial \Phi}{\partial t}(\vec{r}, E, t) &= -\vec{\nabla} \cdot \left(D(\vec{r}, E, t) \vec{\nabla} \Phi(\vec{r}, E, t) \right) + \Sigma_T(\vec{r}, E, t) \Phi(\vec{r}, E, t) \\
 &- \int_0^\infty dE' \Sigma_{s0}(\vec{r}; E' \rightarrow E; t) \Phi(\vec{r}, E', t) \\
 &- (1 - \beta) \frac{\chi_p(E)}{\sqrt{4\pi}} \int_0^\infty dE' \nu \Sigma_f(\vec{r}, E', t) \Phi(\vec{r}, E', t) \\
 &- \tilde{Q}(\vec{r}, E, t) - \sum_{k=1}^K \lambda_k \frac{\chi_k(E)}{4\pi} \mathcal{C}_k(\vec{r}, t), \tag{1.20}
 \end{aligned}$$

Moreover, taking into account that the neutron current, \vec{J} , is defined as

$$\vec{J}(\vec{r}, E, t) \equiv \int_{\vec{\Omega}} d\vec{\Omega} \vec{\Omega} \Phi(\vec{r}, E, \vec{\Omega}, t),$$

from where

$$\begin{aligned}
 \vec{J}(\vec{r}, E, t) &= \frac{\Phi_1^1(\vec{r}, E, t) - \Phi_1^{-1}(\vec{r}, E, t)}{2H_1^1} \vec{i} \\
 &- \frac{i(\Phi_1^1(\vec{r}, E, t) + \Phi_1^{-1}(\vec{r}, E, t))}{2H_1^1} \vec{j} + \frac{\Phi_1^0(\vec{r}, E, t)}{H_1^0} \vec{k}. \tag{1.21}
 \end{aligned}$$

Substituting equations (1.16), (1.17) and (1.18) in equation (1.21) it is obtained

$$\vec{J}(\vec{r}, E, t) = -D(\vec{r}, E, t) \vec{\nabla} \Phi(\vec{r}, E, t),$$

which is known as the Fick's first law.

The cross sections, generally, are functions depending on the neutrons energy, and to simplify the neutron diffusion equation it is used a *multigroup approximation*. Such an approximation consists of obtaining an equation for the neutrons whose energy belongs to the interval $[E_g, E_{g+1}]$, $g = 1, \dots, G - 1$, where G is the number of groups to be considered.

The magnitudes associated with group g are defined as follows,

$$\begin{aligned}
 \Phi_g(\vec{r}, t) &= \int_{E_g}^{E_{g+1}} dE \Phi(\vec{r}, E, t) , \\
 \frac{1}{v_g} &= \int_{E_g}^{E_{g+1}} dE \frac{1}{v} \frac{\Phi(\vec{r}, E, t)}{\Phi_g(\vec{r}, t)} , \\
 \Sigma_{Tg}(\vec{r}, t) &= \int_{E_g}^{E_{g+1}} dE \Sigma_T(\vec{r}, E, t) \frac{\Phi(\vec{r}, E, t)}{\Phi_g(\vec{r}, t)} , \\
 \nu \Sigma_{fg}(\vec{r}, t) &= \int_{E_g}^{E_{g+1}} dE \nu \Sigma_f(\vec{r}, E, t) \frac{\Phi(\vec{r}, E, t)}{\Phi_g(\vec{r}, t)} , \\
 \tilde{Q}_g(\vec{r}, t) &= \int_{E_g}^{E_{g+1}} dE \tilde{Q}(\vec{r}, E, t) , \\
 \chi_{pg} &= \int_{E_g}^{E_{g+1}} dE \chi_p(E) , \\
 \chi_{kg} &= \int_{E_g}^{E_{g+1}} dE \chi_k(E) , \\
 \Sigma_{sg'g}(\vec{r}, t) &= \int_{E_{g'}}^{E_{g'+1}} dE' \int_{E_g}^{E_{g+1}} dE \Sigma_{s0}(\vec{r}, E' \rightarrow E, t) \frac{\Phi(\vec{r}, E', t)}{\Phi_{g'}(\vec{r}, t)} , \quad (1.22)
 \end{aligned}$$

and for each spatial direction, j , the diffusion coefficient for the group g is defined as the function $D_g(\vec{r}, t)$ satisfying the following equation,

$$D_g(\vec{r}, t) = \int_{E_g}^{E_{g+1}} dE D(\vec{r}, E, t) \frac{\partial_j \Phi(\vec{r}, E, t)}{\partial_j \Phi_g(\vec{r}, t)} .$$

The total cross section is written as the sum of an absorption term and scattering terms of the form

$$\Sigma_{Tg}(\vec{r}, t) = \Sigma_{ag}(\vec{r}, t) + \sum_{g'=1}^G \Sigma_{sg'g}(\vec{r}, t) ,$$

and the scattering cross section for the group g is introduced as

$$\Sigma_{sg}(\vec{r}, t) = \sum_{g' \neq g}^G \Sigma_{sg'g}(\vec{r}, t) .$$

Integrating the equation (1.20) from E_g to E_{g+1} , and making use of the previous definitions, the diffusion equation for the group g is obtained as

$$\begin{aligned} \frac{1}{v_g} \frac{\partial \Phi_g(\vec{r}, t)}{\partial t} &= \vec{\nabla} \cdot (D_g(\vec{r}, t) \vec{\nabla} \Phi_g(\vec{r}, t)) - (\Sigma_{ag}(\vec{r}, t) + \Sigma_{sg}(\vec{r}, t)) \Phi_g(\vec{r}, t) \\ &+ \sum_{g' \neq g}^G \Sigma_{sg'g}(\vec{r}, t) \Phi_{g'}(\vec{r}, t) + (1 - \beta) \chi_{pg} \sum_{g'=1}^G \nu \Sigma_{fg'}(\vec{r}, t) \Phi_{g'}(\vec{r}, t) \\ &+ \sum_{k=1}^K \lambda_k \chi_{kg} \mathcal{C}_k(\vec{r}, t) + \tilde{Q}(\vec{r}, t) . \end{aligned} \quad (1.23)$$

With these approximations, the equation (1.2) for the concentration of neutron precursors is written in the form

$$\frac{\partial \mathcal{C}_k(\vec{r}, t)}{\partial t} = \beta_k \sum_{g=1}^G \nu \Sigma_{fg}(\vec{r}, t) \Phi_g(\vec{r}, t) - \lambda_k \mathcal{C}_k(\vec{r}, t) . \quad (1.24)$$

It is worth to note that when approximating equations (1.1) and (1.2) by means of equations (1.23) and (1.24), the space where the fields are defined has been reduced from the space characterized by the variables $(\vec{r}, E, \vec{\Omega}, t)$ to the coordinate space (\vec{r}, t) , with the consequent simplification of the problem.

From now on, the two energy groups neutron diffusion equation is considered, i.e., the energy spectrum is divided into a fast group E_1 , corresponding to the neutrons whose energy is higher to certain value, and a thermal group E_2 , corresponding to the neutrons whose energy is smaller than the previous quantity. Moreover, it is assumed that there are not scattering processes from the thermal to the fast group, i.e., $\Sigma_{21}(\vec{r}, t) = 0$, and there is not neutron production in the thermal group, i.e., $\chi_{p2} = 0$ and $\chi_{k2} = 0$. Finally, it is supposed that there is no neutron external source, and $\chi_{p1} = 1$ and $\chi_{k1} = 1$. Making use of these considerations, a system of partial differential equations for the fast and the thermal group is obtained with the following form

$$[v^{-1}] \dot{\Phi} + \mathcal{L} \Phi = (1 - \beta) \mathcal{M} \Phi + \sum_{k=1}^K \lambda_k \mathcal{C}_k \chi , \quad (1.25)$$

where

$$\mathcal{L} = \begin{bmatrix} -\vec{\nabla} \cdot (D_1 \vec{\nabla}) + \Sigma_{a1} + \Sigma_{12} & 0 \\ -\Sigma_{12} & -\vec{\nabla} \cdot (D_2 \vec{\nabla}) + \Sigma_{a2} \end{bmatrix} , \quad [v^{-1}] = \begin{bmatrix} \frac{1}{v_1} & 0 \\ 0 & \frac{1}{v_2} \end{bmatrix} ,$$

and

$$\mathcal{M} = \begin{bmatrix} \nu \Sigma_{f1} & \nu \Sigma_{f2} \\ 0 & 0 \end{bmatrix} , \quad \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} , \quad \chi = \begin{bmatrix} 1 \\ 0 \end{bmatrix} .$$

Equation (1.24) can be written as the following ordinary differential equation

$$\dot{C}_k = \beta_k [\nu \Sigma_{f1} - \nu \Sigma_{f2}] \Phi - \lambda_k C_k . \quad (1.26)$$

Boundary conditions

Vacuum boundary conditions are considered. Thus, the incoming neutron flux is considered zero at the external boundary of the considered domain. That is

$$\Psi(\vec{r}, E, \vec{\Omega}, t) = 0 \quad \vec{r} \in \Gamma \quad \text{such that} \quad \vec{\Omega} \cdot \vec{n} \leq 0 ,$$

where \vec{n} is the normal outward unit vector at the boundary of the problem, and Γ is the boundary of the reactor domain. This condition can be approximated by means of the Marshak boundary conditions [8]

$$\int_{\vec{\Omega} \cdot \vec{n} \leq 0} d\vec{\Omega} Y_l^{m*}(\vec{\Omega}) \Psi(\vec{r}, E, \vec{\Omega}, t) = 0, \quad l = 1, 3, \dots, L \text{ (odd)} . \quad (1.27)$$

Imposing these boundary conditions for the neutron diffusion approximation we have

$$\frac{1}{2} \Phi(\vec{r}, E, t) = \vec{n} \cdot \vec{J}(\vec{r}, E, t) , \vec{r} \in \Gamma . \quad (1.28)$$

A more general form of these boundary conditions are the albedo boundary conditions

$$\frac{1 - \beta}{2(1 + \beta)} \Phi(\vec{r}, E, t) = \vec{n} \cdot D(\vec{r}, E, t) \vec{\nabla} \Phi(\vec{r}, E, t) , \vec{r} \in \Gamma . \quad (1.29)$$

Other boundary conditions can be considered, such as zero scalar flux

$$\Phi(\vec{r}, E, t) = 0 , \vec{r} \in \Gamma . \quad (1.30)$$

and reflecting boundary conditions, that is

$$\vec{n} \cdot \vec{J}(\vec{r}, E, t) = 0 , \vec{r} \in \Gamma , \quad (1.31)$$

which are particular cases of the albedo boundary conditions.

1.2 Lambda Modes equation

A reactor is said to be critical when the material configuration inside the reactor is set in a way that the neutron production is equal to the neutron loss. Under these conditions the reactor is in steady state. To study the steady state of a given reactor, its criticality can be forced artificially [9], dividing the fission cross

sections by a number λ . In such a way, it is expected that there exists a number λ satisfying the equations

$$\mathcal{L}\Phi = \frac{1}{\lambda}(1 - \beta)\mathcal{M}\Phi + \sum_{k=1}^K \lambda_k \mathcal{C}_k \chi \quad , \quad (1.32)$$

$$0 = \frac{1}{\lambda} \beta_k [\nu \Sigma_{f1} - \nu \Sigma_{f2}] \Phi - \lambda_k \mathcal{C}_k \quad . \quad (1.33)$$

Substituting the equation (1.33) in equation (1.32), and taking into account that $\sum_{k=1}^K \beta_k = \beta$, it is obtained

$$\mathcal{L}\Phi = \frac{1}{\lambda} \mathcal{M}\Phi \quad , \quad (1.34)$$

which is known as the Lambda modes equation for the reactor, and is a generalized *differential eigenvalue problem* associated with the operators \mathcal{L} and \mathcal{M} .

The eigenvalues λ associated with the equation (1.34) are interpreted as cross sections factors. Then, they must be necessarily real numbers, and thus the eigenfunctions Φ will also be real functions.

It is usually supposed that the Lambda Modes form a complete set of functions for the development of modal methods for the integration of the time dependent neutron diffusion equation [3]. Thus, the problem of determining the Lambda modes of a given reactor can be considered as a previous problem for the study of the characteristics of the time dependent neutron diffusion equation.

1.3 VVER Reactors

The Lambda modes problem for reactor cores with geometries based on quadrilateral prisms, has been widely studied [10, 11, 4, 5, 6]. The developed methods are used to analyze reactors of type PWR and BWR, which are the most used occidental reactors. To analyze VVER reactors it is necessary to modify the methodologies applied when discretizing the equations, due to the geometry composed by hexagonal prisms (See Figure 1.1). Due to this fact it is interesting to study different methods for these reactors. VVER is the acronym of the Russian name *Voda Voda Energo Reactor* . VVER reactors are the Russian version of the PWR, and they have been developed along three different generations [12].

The first generation (VVER-440 Model 230) was developed in the 60's. Their principal Strengths are:

- Six primary coolant loops, each with a horizontal steam generator, which together provide a large volume of coolant.

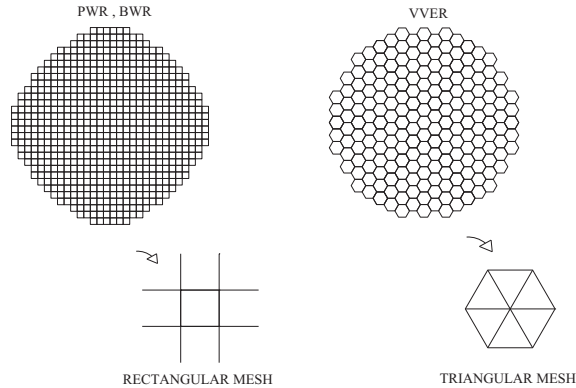


Figure 1.1: Core mesh for both VVER and PWR reactors.

- Isolation valves that allow plant operators to take one or more of the six coolant loops out of service for repair while continuing to operate the plant.
- Ability to sustain a simultaneous loss of coolant and off-site power, due to coolant pumps and two internal power generators that “coast down” after a shutdown.
- Plant worker radiation levels reportedly lower than many Western plants, due to selection of materials, high-capacity primary coolant purification system, and water chemistry control.
- Ability to produce significant amounts of power despite design and instrumentation and control deficiencies.

Meanwhile the principal deficiencies are:

- Accident Localization System (which serves as a reactor confinement) designed to handle only one four-inch pipe rupture. If larger coolant pipes rupture occurs, this system vents directly to the atmosphere through nine large vent valves. Western nuclear plants have containments designed for rupture of the largest pipes. In addition, the confinement has very small volume, very poor leak-tightness and poor hydrogen mitigation.
- No emergency core cooling systems or auxiliary feedwater systems similar to those required in Western nuclear plants.
- Major concern about embrittlement (gradual weakening) of the reactor pressure vessel surrounding nuclear fuel, due to lack of internal stainless-steel cladding and use of low-alloy steel with high levels of impurities.
- Plant instrumentation and controls, safety systems, fire protection systems, and protection for control room operators are below Western standards.

- Quality of materials, construction, operating procedures and personnel training are below Western standards.

The second generation (VVER-440 Model 213) was designed between 1950 and 1980. Their principal strengths are:

- Upgraded Accident Localization System vastly improved over the earlier VVER-440 Model V230 design, comparable to several Western plants, and using a vapour-suppression confinement structure called a “bubbler-condenser” tower.
- Addition of emergency core cooling and auxiliary feedwater systems.
- Reactor pressure vessel with stainless-steel internal lining to alleviate much concern about the vessel embrittlement associated with the earlier VVER-440 Model V230 design.
- Improved coolant pump, and continued use of six coolant loops (providing multiple paths for cooling the reactor) and horizontal steam generators (for better heat transfer) with large coolant volume.
- Standardization of plant components, providing extensive operating experience for many parts and making possible incremental improvements and backfits of components.

Meanwhile the principal deficiencies are:

- Plant instrumentation and controls—for example, reactor protection systems and diagnostics—behind Western standards. Significant variations exist among countries with VVER-440 Model V213 plants.
- Separation of plant safety systems (to help assure that an event in one system will not interfere with the operation of others), fire protection, and protection for control room operators improved over Model V230 plants, but generally below Western standards.
- Poor leak-tightness of confinement.
- Unknown quality of plant equipment and construction, due to lack of documentation on design, manufacturing and construction, and reported instances of poor-quality materials being re-worked at plant sites.
- Major variations in operating and emergency procedures, operator training, and operational safety (for example, use of control-room simulators) among plants. These aspects of plant operations depend primarily on the organization or country operating Model V213 plants rather than on the plant supplier. Some countries have added safety features to their Model V213 plants.

These two kind of reactors has six loops and produce 444 Mw of power.

The third generation (VVER-1000) was developed between 1975 and 1985. These reactors have four loops and produce 1000 Mw of power. They have a new design, to incorporate the new safety standards, where the principals strengths of these reactors are:

- Steel-lined, pre-stressed, large-volume concrete containment structure, similar in function to Western nuclear plants.
- “Evolutionary” design incorporating safety improvements over VVER-440 Model V213 plants. The Soviet approach to standardization was based on continued use of components that had performed well in earlier plants.
- Use of four coolant loops and horizontal steam generators-both considered improvements by Soviet designers.
- Redesigned fuel assemblies that allow better flow of coolant, and improved control rods.
- Plant worker radiation levels reportedly lower than in many Western plants, apparently due to selection of materials, high-capacity system for purifying primary coolant, and water chemistry control.

The most important deficiencies of this type of reactors are:

- Substandard plant instrumentation and controls. Wiring of emergency electrical system and reactor protection system does not meet Western standards for separation-control and safety functions are interconnected in ways that may allow failure of a control system to prevent operation of a safety system.
- Fire protection systems that do not appear to differ substantially from earlier VVER models, which do not meet Western standards.
- Quality control, design and construction significantly deficient by U.S. standards.
- Protection measures for control-room operators essentially unchanged from earlier VVER-440 Model V213 design, which does not meet U.S. standards. Unlike all U.S. nuclear plants, and most in Western countries, VVER-1000s have no on-site “technical support center” to serve as a command post for stabilizing the plant in an emergency. Technical support centers were incorporated in U.S. and many Western nuclear plants following the accident at Three Mile Island Unit 2 in 1979.
- Operating and emergency procedures that fall far short of Western standards and vary greatly among operators of VVER-1000 plants.

- Higher power densities and the smaller volume of primary and secondary systems result in a somewhat less forgiving and stable reactor.

Nowadays, new designs are being studied, such as the new VVER-1200 [13] (which suppose a substantial improvement of the VVER-1000), VVER-1500 and VVER-1600 [14], which are not operative jet.

The main difference between the VVER and the western PWR is the fuel assemblies design and the core geometry. The VVER have fuel assemblies with the shape of hexagonal prisms, instead of the fuel assemblies with the shape of paralepipedal prisms for the western PWR (See Figure 1.1).

The VVER-440 reactor core has 312 fuel assemblies and 37 control rods. 30 of these fuel rods are always outside, (as it is shown in the Figure 1.2). The different colors indicate different materials composing the fuel.

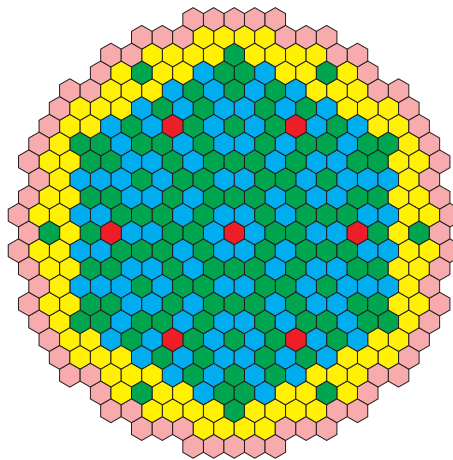


Figure 1.2: Layout of the VVER 440 Core.

The materials configuration for the VVER-1000 reactor core is different (as it is shown in Figure 1.3), where this reactor core has 167 fuel assemblies. The different colors indicate different materials composing the fuel.

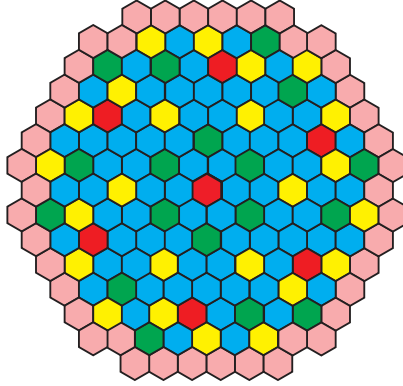


Figure 1.3: Layout of the VVER 1000 Core.

1.4 Spectral Element Methods (SEM)

Spectral Element Methods can be seen as an extremal development for the class of discretization schemes generally known as the Methods of Weighted Residuals (MWR) [15]. When a differential problem has to be solved, it can be written as

$$\begin{aligned} \mathcal{A}[\varphi(x)] &= f(x) & \text{in } & \Omega , \\ \mathcal{B}_i[\varphi(x)] &= f_i(x) & \text{on } & \partial\Omega_i \forall i , \end{aligned}$$

where \mathcal{A} is a differential operator defined over \mathcal{U} , and \mathcal{B}_i operators defining the boundary conditions over appropriate boundary functions spaces $\partial\Omega_i$, in such a way that $\cup_i \partial\Omega_i = \partial\Omega$.

The key point for the MWR are the trial functions (also known as approximating functions), $\{u\}_i \in \mathcal{U}_k$, and the test functions (also known as the weighted functions), $\{w\}_i \in \mathcal{W}_k$. Trial functions are used as a basis functions to expand the solution in a truncated basis

$$\varphi(x) \approx u(x) = \sum_{i=0}^k \varphi_i u_i(x) . \quad (1.35)$$

Test functions are used to ensure that the differential equation is satisfied as good as possible for the truncated expansion series. This is achieved by minimizing the residual, \mathcal{R} , defined as the error in the differential equation produced when using the truncated expansion instead of the exact solution

$$\mathcal{R}[u] = f(x) - \mathcal{A}[u(x)] . \quad (1.36)$$

with respect to a concrete norm. It is equivalent to force the residual to be orthogonal with respect to the space of test functions \mathcal{W}

$$\langle \mathcal{R}[u_i], w_j \rangle = 0, \quad \forall u_i(x) \in \mathcal{U}_k, \quad \forall w_j \in \mathcal{W}_k, \quad (1.37)$$

where the scalar product $\langle \cdot, \cdot \rangle$ is defined as

$$\langle u, v \rangle = \int_{\Omega} u(x)v(x)d\Omega. \quad (1.38)$$

At the beginning, these methods were applied to simple domains, where the solution has general character, i.e., the value of the solution at each point of the domain was determined by the value of the solution in the rest of the domain. The election of the trial functions is one of the characteristics making these methods different from linear finite element methods or finite differences methods. Trial functions for the Spectral Methods are infinitely smooth (typically they are tensor products of eigenfunctions of Sturm-Liouville problems). In the case of linear finite element methods, the domain, Ω , is divided into smaller subdomains Ω_i , and different trial functions are specified over each subdomain. Then trial functions have local character, and they are necessary when dealing with complex geometries.

Different choices of test functions lead to different spectral methods, as the Galerkin method, Collocation method, or Tau method. The Galerkin method is obtained when the trial functions space is the same as the test functions space, i.e., $\mathcal{U} = \mathcal{W}$, when the trial functions as the test functions are infinitely smooth functions fulfilling the boundary conditions. The Collocation method is obtained when translated Dirac delta functions $\delta_{x_0}(x)$ are used as test functions, positioned at certain collocation points. This approximations force the residual to be exactly zero at these collocation points. The Tau methods are similar to the Galerkin methods in the way they apply, but Tau methods are different in the sense that test functions do not need to satisfy the boundary conditions, and a complementary set of equations are needed to apply the boundary conditions.

The Collocation method is the most simple method to use, and it seems that it has been used first by Slater [16] and Kantorovic [17] with specific applications, Frazer, Jones and Skan [18] developed the methods as a general method to solve ordinary differential equations. They used a large amount of trial functions and an arbitrary distribution of collocation points. The work of Lanczos [19] established for the first time that a proper choice of test functions and the placement of the collocation points is crucial to the accuracy of the solution obtained.

The first applications of the Spectral Collocation methods to the partial differential equations was made for periodic problems by Kreiss and Olinger [20] (who call it Fourier method) and Orszag [21] (who call it pseudospectral method). This approximation is very attractive due to the facility it can be applied to variable coefficients, and even to nonlinear problems.

Galerkin type approximation is, however, the most esthetic of the Weighted Residual Methods. Due to the fact that the trial and test functions are the same, the physical problem can be discretized in terms of a variational principle. Finite element methods typically use this approximation. Moreover, the first serious ap-

plications of the Spectral Methods were Galerkin methods. However, Galerkin methods become practical to calculate big nonlinear problems, after Orszag [22] and Eliassen, Machenhauer and Rasmussen [23] development of transformation methods to evaluate the convolution sums coming from quadratic coefficients.

The first unified theory about Spectral Methods is on the monograph [24]. Since this moment, the theory has been extended to be applied to a wide variety of problems. Stability and convergence analysis for the Spectral methods are based on different types of approximations. Thinking on Spectral methods as Weighted Residual Methods (or a variational methods) has shown to be useful in the theoretical research, and it allowed functional analysis methods to analyze the convergence and stability of the Spectral methods.

In the half seventies, Gottlieb and Orszag [24] summarized the state of the art for the theory and applications of Spectral methods, and posteriori developments are reviewed at the symposium edited by Voigt, Gottlieb and Hussaini [25]. These references are focused on applications to fluid dynamics. Then, after the big development of theory and applications of the Spectral methods in simple domains, Spectral methods evolve to deal with problems over more general geometries. The basic idea is to split the whole domain of the problem into several subdomains. At this point Spectral methods are called Spectral Element methods, and they are identified as high order finite element methods, improving the accuracy by means of increasing the number of trial functions in each subdomain. The use of subdomains makes possible the implementation of Spectral methods on parallel computers.

Partition techniques have been used for a while in finite differences and finite element methods. At the end of the seventies they were studied also for Spectral methods. Orszag [26] describes a technique to connect the boundaries. Morchoisne [27] developed a method based on the overlapping of multiple domains. Patera [28] used a variational formulation, for which he used the term of *Spectral Element Method*.

A crucial aspect of any domain decomposition method is the way the solutions of adjacent domains are connected. Some methods take a classical point of view (point-wise) for the equations. If the equation is of order d , then on the inner boundary of adjacent domains the solution and its derivatives up to order $d - 1$ must be continuous. For second order problems, this is usually forced requiring continuity for the solution and for the derivatives in the normal direction at the inner boundaries. This condition for the derivative can be substituted by any other direction (but the tangent one). This continuity conditions are imposed by forcing them in a suitable set of points. Then they are exactly satisfied by any approximation.

Alternatively, the differential equation can be written in the weak form by means of a variational formulation. Moreover, there is a dual form for the weak formulation,

obtained by means of an “integration by parts” process, where the continuity conditions for the derivatives are imposed implicitly.

When imposing the weak formulation over the subdomains, it is necessary to take into account the type of elements which are the subdomains. In a two-dimensional domain, for example, these elements used to be quadrilateral or triangles, depending on the geometry, and the type of polynomials to deal with different elements is different [29].

The decomposition into quadrilaterals has the advantage that these elements are built by the cartesian product of unidimensional spaces, and then the solutions over these spaces can be written as a tensor product of unidimensional functions, usually polynomials of Lebesgue, Lagrange or Jacobi [30]. And whether a Collocation method is used to fulfil the differential equation requirements, the collocation points are a cartesian product of unidimensional collocation points. The most common choice use the Legendre-Gauss-Lobatto points.

For triangular elements, the approximation of the solution cannot be decomposed in a tensorial product of unidimensional functions, but different orthogonal polynomials over triangular domains are known, as the orthogonal polynomials of Dubiner [31]. A different kind of polynomials can also be used, maybe losing the orthogonality, but obtaining another property, as the way the boundary and continuity conditions can be imposed, as it is done by the modified Dubiner polynomials [31, 32]. When a Collocation method is used to discretize the differential equation, a typical set of collocation points are the Fekete points [33], which are considered to be the equivalent over a triangle to the Legendre-Gauss-Lobatto points, due to the fact that they coincide over a unidimensional domain.

Chapter 2

Spectral Element Method for the Neutron Diffusion Equation

Contents

2.1 One-dimensional geometries	24
2.1.1 Spectral Element Methods.	26
2.1.2 Numerical results	35
2.2 Two- and three-dimensional geometries.	41
2.2.1 Variational Formulation	43
2.2.2 Reference element	46
2.2.3 Polynomial basis.	48
2.2.4 Building the matrices.	52
2.2.5 Algebraic problem.	55
2.2.6 Numerical Results.	56

At this chapter, the method to discretize the spatial part of the neutron diffusion equation is explained. First, different methods will be introduced in one-dimensional geometries. These methods are formulated and compared when solving different test problems, an analytical slab and a heterogeneous problem. The results are compared with an analytical solution for the slab problem, and the solutions for the heterogeneous two groups problem. Once the best method among the different options to discretize the problem in one dimension has been determined, i.e. the Spectral Element Method, it is formulated for two- and three-dimensional geometries, and results are reported for the first eigenvalue (the k -effective) and the first eigenfunction, compared with the results reported in [34]. Results for the dominant Lambda modes are also reported for completeness.

2.1 One-dimensional geometries

The balance of neutrons inside a nuclear power reactor is described by the Boltzmann transport equation [8]. We will consider the eigenvalue problem known as the *Lambda modes problem* [9], which for reactors with 1D geometry of length L_r , in the approximation of one group of energy, takes the form

$$\begin{aligned} & \mu \frac{\partial \Psi(x, \mu)}{\partial x} + \Sigma_t(x) \Psi(x, \mu) \\ &= \int_{-1}^{+1} \Sigma_s(x, \mu_0) \Psi(x, \mu') d\mu' + \frac{1}{\lambda} \frac{\nu \Sigma_f(x)}{2} \int_{-1}^{+1} \Psi(x, \mu') d\mu' , \end{aligned} \quad (2.1)$$

with the boundary conditions

$$\Psi(0, \mu) = 0 , \quad 0 < \mu \leq 1 , \quad \Psi(L_r, \mu) = 0 , \quad -1 \leq \mu < 0 , \quad (2.2)$$

where $x \in [0, L_r]$, θ is the angle between the direction of the incident neutron velocity and the x axis, $\mu = \cos(\theta)$, θ_0 is the angle between the incident neutrons and the scattered neutrons, $\mu_0 = \cos(\theta_0)$. $\Sigma_t(x)$ is the total cross section, $\Sigma_s(x, \mu_0)$ is the scattering cross section, $\Sigma_f(x)$ is the fission cross section and ν is the average number of neutrons produced in each fission. λ is the eigenvalue of the problem and $\Psi(x, \mu)$ its corresponding eigenfunction.

Realistic transport problems are dealt with several groups of energy, but the generalization of this formulation to more groups of energy is simple.

The dominant eigenvalue of problem (2.1), $\lambda = k_{\text{eff}}$, is the k -effective of the system and measures its criticality. The corresponding eigenvector is the directional flux distribution of a stationary configuration of the system obtained dividing Σ_f by k_{eff} .

P_L approximations to the neutron transport equation (2.1) assume that the angular dependence of both the neutron flux distribution and the scattering cross section can be expanded in terms of $L + 1$ Legendre polynomials,

$$\begin{aligned} \Psi(x, \mu) &= \sum_{n=0}^L \left(\frac{2n+1}{2} \right) \Phi_n(x) P_n(\mu) , \\ \Sigma_s(x, \mu_0) &= \sum_{n=0}^L \left(\frac{2n+1}{2} \right) \Sigma_{sn}(x) P_n(\mu_0) . \end{aligned} \quad (2.3)$$

Inserting these expansions into equation (2.1) and with the aid of the orthogonality relations for the Legendre polynomials and the addition theorem for the associated Legendre functions we obtain the standard P_L approximation in one-dimensional

geometries [8, 35]

$$\begin{aligned} \frac{d\Phi_1(x)}{dx} + \Sigma_a \Phi_0(x) &= \frac{1}{\lambda} \nu \Sigma_f \Phi_0(x) , \\ (n+1) \frac{d\Phi_{n+1}(x)}{dx} + n \frac{d\Phi_{n-1}(x)}{dx} + (2n+1) (\Sigma_t - \Sigma_{sn}) \Phi_n(x) &= 0 , \\ 1 \leq n \leq L , \end{aligned} \quad (2.4)$$

where $\Sigma_a = \Sigma_t - \Sigma_{s0}$ is the absorption cross section, and

$$\Phi_n(x) = \int_{-1}^1 d\mu P_n(\mu) \Psi(x, \mu) , \quad \Sigma_{sn}(x) = \int_{-1}^1 d\mu_0 P_n(\mu_0) \Sigma_s(x, \mu_0) . \quad (2.5)$$

The P_L equations constitute a set of $L+1$ equations with $L+2$ unknowns. This problem is usually solved ignoring the term $\frac{d\Phi_{L+1}}{dx}$ in the $n=L$ equation.

Typical approximations for the boundary conditions (2.2) are Marshak's conditions [8]

$$\int_0^1 P_n(\mu) \Psi(0, \mu) d\mu = 0 , \quad \int_{-1}^0 P_n(\mu) \Psi(L, \mu) d\mu = 0 , \quad (2.6)$$

with n odd, $n = 1, 3, \dots, L$ (or $L-1$).

Considering $L=1$ in equations (2.3) and the notation $\Phi_0 = \Phi$ and $\Phi_1 = J$, we obtain the P_1 equations

$$\begin{aligned} \frac{dJ}{dx} + \Sigma_a \Phi &= \frac{1}{\lambda} \nu \Sigma_f \Phi , \\ \frac{d\Phi}{dx} + 3(\Sigma_t - \Sigma_{s1}) J &= 0 , \end{aligned}$$

which are equivalent to

$$J = -D \frac{d\Phi}{dx} , \quad (2.7)$$

$$- \frac{d}{dx} \left(D \frac{d\Phi}{dx} \right) + \Sigma_a \Phi = \frac{1}{\lambda} \nu \Sigma_f \Phi , \quad (2.8)$$

where $D = \frac{1}{3(\Sigma_t - \Sigma_{s1})}$ is the diffusion coefficient. The first equation (2.7), is the Fick's law and the second equation, (2.8), is the diffusive form of the P_1 equation, also known as the neutron diffusion equation.

From the vacuum Marshak's boundary conditions, we obtain the equations

$$\begin{aligned} \Phi(0) + 2J(0) &= 0 , \\ \Phi(L_r) - 2J(L_r) &= 0 , \end{aligned}$$

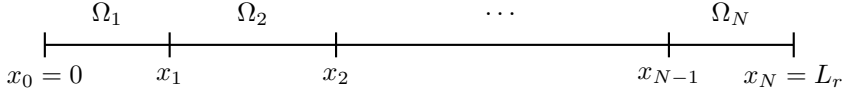


Figure 2.1: Discretization of a 1D reactor.

which, using the Fick's law, can be rewritten as

$$\begin{aligned}\Phi(0) - 2D(0) \frac{d\Phi}{dx}(0) &= 0, \\ \Phi(L_r) + 2D(L_r) \frac{d\Phi}{dx}(L_r) &= 0.\end{aligned}$$

More general albedo boundary conditions take the form

$$\alpha^- \Phi(0) + \beta^- D(0) \frac{d\Phi}{dx}(0) = 0, \quad (2.9)$$

$$\alpha^+ \Phi(L_r) + \beta^+ D(L_r) \frac{d\Phi}{dx}(L_r) = 0, \quad (2.10)$$

where α^\pm, β^\pm are arbitrary constants. Using boundary conditions (2.9) and (2.10) we include zero-flux, zero-current, zero-incoming current and fixed albedo conditions, for different values of α^\pm and β^\pm .

To discretize equation (2.8) for a nuclear power reactor we have to take into account that a mesh is naturally defined by the different compositions considered in the cell codes used to obtain the homogenized nuclear cross sections for the reactor. In this way, it is interesting to develop numerical methods that use this mesh and that to increase their accuracy it is not necessary to refine the mesh. These kind of methods are known as p -type methods [36]. As a particular kind of these methods, we will consider Spectral Element Methods based on the expansion of the neutron flux in terms of a continuous basis of polynomials for a one dimensional nuclear reactor. These methods are developed to study their behaviour and to find the best strategy to be generalized to multidimensional geometries using general triangular or tetrahedral cells.

2.1.1 Spectral Element Methods

To discretize the neutron diffusion equation (2.8), the first step is to divide the domain defining the reactor, Ω , into a set of elements $\Omega_e, 1 \leq e \leq N$, defined by the different materials considered in the reactor (see Figure 2.1).

To develop a Spectral Element Method, each element $\Omega_e = [x_{e-1}, x_e]$ of the reactor is transformed into the reference element $\Omega_{REF} = [0, 1]$ by means of the change of variables

$$u = \frac{x - x_{e-1}}{\Delta x_e}, \quad 1 \leq e \leq N, \quad (2.11)$$

where $\Delta x_e = x_e - x_{e-1}$ and it is assumed that the solution for the neutron flux over each element Ω_e , can be expanded as

$$\Phi_e(u) = \sum_{i=0}^K \Phi_{e,i} p_i(u), \quad 1 \leq e \leq N, \quad (2.12)$$

or

$$\Phi_e(x) = \sum_{i=0}^K \Phi_{e,i} p_{e,i}(x), \quad 1 \leq e \leq N,$$

where

$$p_{e,i}(x) = p_i\left(\frac{x - x_{e-1}}{\Delta x_e}\right).$$

being $\{p_i(u)\}_{i=0}^K$ a basis of polynomials.

Different methods will be obtained if different basis of polynomials are considered. As the neutron flux is expected to be a continuous function, we will consider a basis of polynomials with the characteristic that any function expanded in terms of these polynomials is continuous by construction in the interior of the domain defining the reactor.

2.1.1.1 Polynomial basis

The particular basis of polynomials that will be used for the reference element, Ω_{REF} , is the following one,

$$\begin{aligned} p_0(u) &= 1 - u, \\ p_1(u) &= u, \\ p_i(u) &= (1 - u)u P_{i-2}^{1,1}(2u - 1)2^{\frac{3}{2}}, \quad 2 \leq i \leq K, \end{aligned}$$

where $P_j^{1,1}(x)$ is the Jacobi polynomial [30] of degree j defined in $[-1, 1]$. These polynomials will be evaluated by means of the following recurrence relation:

$$\begin{aligned} P_0^{1,1}(x) &= 1, \\ P_1^{1,1}(x) &= 2x, \\ &\vdots \\ P_{i+1}^{1,1}(x) &= \frac{1}{a_{1i}} \left(a_{3i}x P_i^{1,1}(x) - a_{4i}P_{i-1}^{1,1}(x) \right), \quad 1 \leq i, \end{aligned}$$

where

$$\begin{aligned} a_{1i} &= 2(i+1)(i+3)(2i+2), \\ a_{3i} &= (2i+2)(2i+1)(2i), \\ a_{4i} &= 4(i+1)^2(i+2). \end{aligned}$$

We will assume that these polynomials vanish out of the reference element, $\Omega_{REF} = [0, 1]$.

From the definition of the polynomials it is easy to see that the only polynomial different from 0 at the boundary $u = 0$ from Ω_{REF} is $p_0(u) = 1 - u$, and at the other boundary, $u = 1$, is the polynomial $p_1(u) = u$. We impose continuity conditions for the neutron flux at the interior boundaries of the different elements of the reactor discretization

$$\Phi_e(x_e) = \Phi_{e+1}(x_e), \quad e = 1 \dots, N-1,$$

that is,

$$\sum_{i=0}^K \Phi_{e,i} p_i(1) = \sum_{i=0}^K \Phi_{e+1,i} p_i(0).$$

This condition is fulfilled if the coefficients of the flux expansions satisfy

$$\Phi_{e,1} = \Phi_{e+1,0}, \quad 1 \leq e \leq N-1. \quad (2.13)$$

The unknowns to be determined by the different pseudospectral methods are the coefficients $\Phi_{e,i}$, of expansions (2.12), which will named *local* unknowns of the method. As the relations (2.13) hold, the *local* unknowns are not independent. Eliminating the dependent unknowns we obtain the *global* unknowns.

The relation between the *local* and the *global* unknowns is illustrated by equation (2.14)

$$\begin{array}{c} \text{Local unknowns} \\ \left[\begin{array}{c} \Phi_{e-1,0} \\ \Phi_{e-1,1} \\ \Phi_{e,0} \\ \Phi_{e,1} \\ \Phi_{e+1,0} \\ \Phi_{e+1,1} \end{array} \right] \left[\begin{array}{c} 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 4 \end{array} \right] \Rightarrow \begin{array}{c} \text{Global unknowns} \\ \left[\begin{array}{c} \Phi_i \\ \Phi_{i+1} \\ \Phi_{i+2} \\ \Phi_{i+3} \end{array} \right] \left[\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \right] \end{array} \quad (2.14) \end{array}$$

where three contiguous elements (Ω_{e-1} , Ω_e and Ω_{e+1}), and a linear basis of polynomials for these elements $p_i(u)$, $i = \{0, 1\}$, are considered. The coefficients of the polynomials of higher order are not affected by the continuity relations (2.13), since they vanish at the borders of the reference element. The relation between the *local* and the *global* unknowns is established by means of an ordering vector.

Using this basis of polynomials for the expansions (2.12), different methods to discretize the neutron diffusion equation have been developed for a 1D reactor.

2.1.1.2 Continuous Current Method (CCM)

Any method to discretize the neutron diffusion equation must assure the continuity of the neutron flux and current. The continuity of the neutron flux is assured by the relation (2.13) satisfying the coefficients of the continuous basis of polynomials presented above.

The Continuous Current Method (CCM) presented here is characterized by the fact that the continuity of the current is strongly forced. To impose the continuity of the neutron current in the different elements Ω_e , of the reactor domain, we impose the conditions

$$-D_e \frac{d\Phi_e}{dx}(x_e) = -D_{e+1} \frac{d\Phi_{e+1}}{dx}(x_e), \quad 1 \leq e \leq N-1.$$

In terms of the variable u , these relations can be written as

$$-\frac{D_e}{\Delta x_e} \sum_{i=0}^K \Phi_{e,i} \frac{dp_i}{du}(1) = -\frac{D_{e+1}}{\Delta x_{e+1}} \sum_{i=0}^K \Phi_{e+1,i} \frac{dp_i}{du}(0), \quad 1 \leq e \leq N-1. \quad (2.15)$$

At the reactor boundaries we will impose the boundary conditions (2.9) and (2.10) that, in terms of the variable u , can be written as

$$\begin{aligned} \alpha^- \Phi_{1,0} + \beta^- \frac{D_1}{\Delta x_1} \sum_{i=0}^K \Phi_{1,i} \frac{dp_i}{du}(0) &= 0, \\ \alpha^+ \Phi_{N,1} + \beta^+ \frac{D_N}{\Delta x_N} \sum_{i=0}^K \Phi_{N,i} \frac{dp_i}{du}(1) &= 0. \end{aligned} \quad (2.16)$$

Finally, to approximate equation (2.8) over each element Ω_e of the mesh, we consider moment-like equations of the form

$$\begin{aligned} \int_{\Omega_e} \left(-\frac{d}{dx} \left(D_e \frac{d\Phi_e}{dx}(x) \right) + \Sigma_{a,e} \Phi_e(x) \right) p_{e,i}(x) dx &= \\ = \frac{1}{\lambda} \int_{\Omega_e} (\nu \Sigma_{f,e} \Phi_e(x)) p_{e,i}(x) dx, \quad 1 \leq e \leq N, \quad 0 \leq i \leq K-2. \end{aligned} \quad (2.17)$$

In terms of the variable u of the reference domain Ω_{REF} and the local unknowns, equations (2.17) can be rewritten as

$$\begin{aligned} -\frac{D_e}{(\Delta x_e)^2} \sum_{j=0}^K \Phi_{e,j} \int_0^1 \frac{d^2 p_j}{du^2}(u) p_i(u) du + \Sigma_{a,e} \sum_{j=0}^K \Phi_{e,j} \int_0^1 p_j(u) p_i(u) du \\ = \frac{1}{\lambda} \nu \Sigma_{f,e} \sum_{j=0}^K \Phi_{e,j} \int_0^1 p_j(u) p_i(u) du, \quad 1 \leq e \leq N, \quad 0 \leq i \leq K-2, \end{aligned} \quad (2.18)$$

where D_e , $\Sigma_{a,e}$ and $\nu\Sigma_{f,e}$ are the macroscopic cross sections, considered constant over each element Ω_e . We obtain in this way $K - 1$ constraints for the N elements of the reactor. These relations are completed with the $N - 1$ continuity constraints for the neutron flux (2.13), the continuity conditions for the neutron current (2.15), and the 2 boundary conditions (2.16).

To optimize the method, these equations are considered in terms of the *global* unknowns obtaining a reduction of $N - 1$ in the size of the resulting algebraic eigenvalue problem. On the other hand, the integrals appearing in equations (2.18) involve polynomial functions, thus, they can be computed exactly using a Gaussian quadrature rule. Particularly, the Gauss-Legendre rule [37] has been used.

2.1.1.3 Point-wise Collocation Method (PCM)

In this method, the continuity of neutron flux and current together with the boundary conditions (2.9) and (2.10) are imposed in the same way as it is done in the Continuous Current Method (CPM).

To complete the set of relations needed to determine the algebraic problem, we impose that the neutronic flux for each element Ω_e (2.12), must satisfy the neutron diffusion equation (2.8) on a set of collocation points over the reactor domain. The choice of these collocation points will determine the accuracy of the approximation and the conditioning of the resulting matrices defining the algebraic eigenvalue problem, which approximates the initial problem (2.8).

The set $\{\xi_{e,i}\}_{i=0}^K$ of $K + 1$ collocation points that we have chosen to use over each element in which we have discretize the reactor domain, will be the Gauss-Legendre quadrature points in $[-1, 1]$, $\{\xi_i\}_{i=0}^K$, applied to each element Ω_e by means of the change of variables

$$\xi_{e,i} = \frac{1}{2}(x_{e+1} + x_e) + \frac{1}{2}\Delta x_e \xi_i .$$

Using these points, we obtain the relations

$$-D_e \frac{d^2 \Phi_e}{dx^2}(\xi_{e,i}) + \Sigma_{a,e} \Phi_e(\xi_{e,i}) = \frac{1}{\lambda} \nu \Sigma_{f,e} \Phi_e(\xi_{e,i}) ,$$

$$1 \leq e \leq N , 0 \leq i \leq K - 2 ,$$

that, in terms of the variable u of the reference element Ω_{REF} , can be rewritten as

$$-\frac{D_e}{(\Delta x_e)^2} \sum_{j=0}^K \Phi_{j,e} \frac{d^2 p_j}{du^2}(\xi_i) + \Sigma_{a,e} \sum_{j=0}^K \Phi_{j,e} p_j(\xi_i) =$$

$$= \frac{1}{\lambda} \nu \Sigma_{f,e} \sum_{j=0}^K \Phi_{j,e} p_j(\xi_i) , \quad 1 \leq e \leq N , 0 \leq i \leq K - 2 .$$

These relations together with the continuity conditions for the neutronic current (2.15) and the boundary conditions (2.16), all them considered in terms of the *global* unknowns, define the algebraic approximation of the Lambda modes problem obtained with the PCM method.

2.1.1.4 Spectral Element Method (SEM)

Together with the spectral methods exposed above, we will consider a Spectral Element Method, based on a weak formulation of the neutron diffusion equation and that assumes that the neutron flux in each node of the discretization of the reactor can be expanded in terms of the continuous polynomial basis used in the other two methods. This Spectral Element Method is also identified with the *p*-type Finite Element Method.

The neutron diffusion equation (2.8), can be rewritten as

$$\frac{d}{dx} \left(-D \frac{d\Phi}{dx}(x) \right) + \Sigma_a \Phi(x) = S(x) , \quad (2.19)$$

where the source term, $S(x)$, is defined as

$$S(x) = \frac{1}{\lambda} \nu \Sigma_f \Phi(x) ,$$

and the boundary conditions (2.9) and (2.10) can be written formally as

$$D \frac{d\Phi}{dx}(x) + \frac{\alpha(x)}{\beta(x)} \Phi(x) = 0 ,$$

where

$$\alpha(0) = \alpha^- , \quad \alpha(L_r) = \alpha^+ , \quad \beta(0) = \beta^- , \quad \beta(L_r) = \beta^+ .$$

First, we will show that a stationary point of a functional defined in a Sobolev space is a solution of the neutron diffusion equation [38]. Let us consider the following functional

$$\begin{aligned} \mathcal{F}(\Phi) &= \int_{\Omega} D \left(\frac{d\Phi}{dx}(x) \right)^2 dx + \int_{\Omega} \Sigma_a \Phi(x)^2 dx \\ &\quad - \int_{\Omega} 2S(x)\Phi(x)dx + \frac{\alpha(L_r)}{\beta(L_r)} \Phi^2(L_r) - \frac{\alpha(0)}{\beta(0)} \Phi^2(0) , \end{aligned} \quad (2.20)$$

where $\Phi(x) \in \mathcal{H}_0(\Omega)$, a Sobolev space defined as

$$\mathcal{H}_0(\Omega) = \{h : h \in \mathcal{C}^0(\Omega) \quad \text{and} \quad \frac{dh}{dx} \in L^2(\Omega)\} .$$

A stationary point of the functional (2.20) is the function Φ satisfying the relation

$$\delta_{\delta\Phi}\mathcal{F} = \lim_{\epsilon \rightarrow 0} \left\{ \frac{d}{d\epsilon} \mathcal{F}(\Phi + \epsilon\delta\Phi) \right\} = 0, \quad (2.21)$$

$\forall \delta\Phi \in \mathcal{H}_0(\Omega)$.

Calculating the limit (2.21), we obtain

$$\begin{aligned} \delta_{\delta\Phi}\mathcal{F}(\Phi) &= \int_{\Omega} D \left(\frac{d\Phi}{dx}(x) \right) \left(\frac{d\delta\Phi}{dx}(x) \right) dx + \int_{\Omega} \Sigma_a \Phi(x) \delta\Phi(x) dx \\ &- \int_{\Omega} S(x) \delta\Phi(x) dx - \frac{\alpha(L_r)}{\beta(L_r)} \Phi(L_r) \delta\Phi(L_r) + \frac{\alpha(0)}{\beta(0)} \Phi(0) \delta\Phi(0), \end{aligned}$$

$\forall \delta\Phi(x) \in \mathcal{H}_0(\Omega)$.

Taking into account that the domain defining the reactor, $\Omega = [0, L_r]$ is divided into elements Ω_e , we have

$$\begin{aligned} \delta_{\delta\Phi}\mathcal{F}(\Phi) &= \sum_{e=1}^N \int_{\Omega_e} D_e \frac{d\Phi}{dx}(x) \frac{d\delta\Phi}{dx}(x) dx + \sum_{e=1}^N \int_{\Omega_e} \Sigma_{a,e} \Phi(x) \delta\Phi(x) dx \\ &- \sum_{e=1}^N \int_{\Omega_e} S_e(x) \delta\Phi(x) dx - \frac{\alpha(L_r)}{\beta(L_r)} \Phi(L_r) \delta\Phi(L_r) + \frac{\alpha(0)}{\beta(0)} \Phi(0) \delta\Phi(0). \end{aligned} \quad (2.22)$$

Integrating by parts,

$$\begin{aligned} \int_{\Omega_e} D_e \frac{d\Phi}{dx}(x) \frac{d\delta\Phi}{dx}(x) dx &= \left[D_e \delta\Phi(x) \frac{d\Phi}{dx}(x) \right]_{x_{e-1}}^{x_e} \\ &- \int_{\Omega_e} \delta\Phi(x) \frac{d}{dx} \left(D_e \frac{d\Phi}{dx}(x) \right) dx. \end{aligned} \quad (2.23)$$

Substituting equation (2.23) into equation (2.22), we have

$$\begin{aligned} \delta_{\delta\Phi}\mathcal{F}(\Phi) &= \sum_{e=1}^N \int_{\Omega_e} \left(-\frac{d}{dx} \left(D_e \frac{d\Phi}{dx}(x) \right) + \Sigma_{a,e} \Phi(x) - S_e(x) \right) \delta\Phi(x) dx \\ &+ \sum_{e=1}^{N-1} \left(D_e \frac{d\Phi}{dx}(x_e) - D_{e+1} \frac{d\Phi}{dx}(x_e) \right) \delta\Phi(x_e) \\ &+ \left(D_N \frac{d\Phi}{dx}(L_r) - \frac{\alpha(L_r)}{\beta(L_r)} \Phi(L_r) \right) \delta\Phi(L_r) \\ &- \left(D_1 \frac{d\Phi}{dx}(0) + \frac{\alpha(0)}{\beta(0)} \Phi(0) \right) \delta\Phi(0) = 0. \end{aligned}$$

As $\delta\Phi(x)$ is an arbitrary variation, $\Phi(x)$ is a stationary point of functional (2.20) if, and only if, the following conditions are satisfied:

1. The function $\Phi(x)$ holds the relation

$$\frac{d}{dx} \left(-D \frac{d\Phi}{dx}(x) \right) + \Sigma_a \Phi(x) = S(x) , \quad (2.24)$$

which is the neutron diffusion equation (2.19).

2. The derivatives are continuous across any inner connection of elements Ω_e ,

$$-D_e \frac{d\Phi}{dx}(x_e) = -D_{e+1} \frac{d\Phi}{dx}(x_e) \quad 1 \leq e \leq N , \quad (2.25)$$

that is, the neutron current is continuous.

3. The function $\Phi(x)$ should satisfy the boundary conditions,

$$\begin{aligned} D_1 \frac{d\Phi}{dx}(0) - \frac{\alpha(0)}{\beta(0)} \Phi(0) &= 0 , \\ D_N \frac{d\Phi}{dx}(L_r) - \frac{\alpha(L_r)}{\beta(L_r)} \Phi(L_r) &= 0 , \end{aligned}$$

which are the boundary conditions (2.9) and (2.10).

Using this result, the Spectral Element Method proposed is based on obtaining an approximation for the stationary point of functional (2.20). It is assumed that the neutron flux in each element can be expressed in terms of the *global* unknowns, and the stationary point of the functional is obtained making zero the derivatives with respect to these, Φ_i .

We have to distinguish different cases. When we derivate with respect to the common coefficient of the polynomials of the basis $p_{e,1}(x)$ and $p_{e+1,0}(x)$, that take the value 1 at $x = x_e$, which is the connection point between both elements Ω_e and Ω_{e+1} , (these functions are called hat functions, see Figure 2.2), we obtain the equations,

$$\begin{aligned} &\int_{\Omega_e} \left(D_e \frac{d\Phi}{dx}(x) \frac{dp_{e,1}}{dx}(x) + \left(\Sigma_{a,e} - \frac{1}{\lambda} \nu \Sigma_{f,e} \right) \Phi(x) p_{e,1}(x) \right) dx \\ &+ \int_{\Omega_{e+1}} \left(D_{e+1} \frac{d\Phi}{dx}(x) \frac{dp_{e+1,0}}{dx}(x) + \left(\Sigma_{a,e+1}(x) - \frac{1}{\lambda} \nu \Sigma_{f,e+1} \right) \Phi(x) p_{e+1,0}(x) \right) dx = 0 , \end{aligned} \quad (2.26)$$

for $1 \leq e \leq N - 1$.

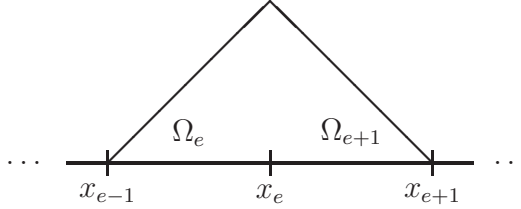


Figure 2.2: Hat functions.

If we derivate with respect to the coefficient corresponding to the polynomials $p_{e,j}(x)$, $2 \leq j \leq K$, which are called bubble functions, and take the value 0 in x_{e-1} and x_e (see Figure 2.3), we obtain the equations,

$$\int_{\Omega_e} D_e \frac{d\Phi}{dx}(x) \frac{dp_{e,j}}{dx}(x) dx + \int_{\Omega_e} \Sigma_{a,e} \Phi(x) p_{e,j}(x) dx - \frac{1}{\lambda} \int_{\Omega_e} \nu \Sigma_{f,e} \Phi(x) p_{e,j}(x) dx = 0, \quad (2.27)$$

for $1 \leq e \leq N$.

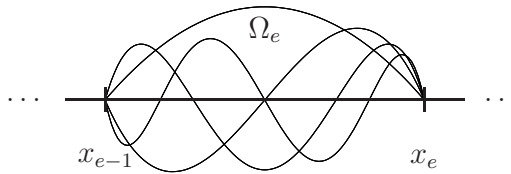


Figure 2.3: Bubble functions.

Finally, to determine the problem, we derivate with respect to the coefficient of the polynomial $p_{N,1}(x)$ taking the value 1 at the point $x = x_N$ and with respect to the coefficient of the polynomial $p_{1,0}(x)$, that takes the value 1 at the point $x = x_0$. These functions are called boundary functions and are shown in Figure 2.4. In

this case we obtain the equations,

$$\begin{aligned}
 & \int_{\Omega_N} D_N \frac{d\Phi}{dx}(x) \frac{dp_{N,1}}{dx}(x) dx + \int_{\Omega_N} \Sigma_{a,N} \Phi(x) p_{N,1}(x) dx \\
 & - \frac{1}{\lambda} \int_{\Omega_N} \nu \Sigma_{f,N} \Phi(x) p_{N,1}(x) dx + \frac{\alpha(L_r)}{\beta(L_r)} \Phi(L_r) = 0, \\
 & \int_{\Omega_1} D_1 \frac{d\Phi}{dx}(x) \frac{dp_{1,0}}{dx}(x) dx + \int_{\Omega_1} \Sigma_{a,1} \Phi(x) p_{1,0}(x) dx \\
 & - \frac{1}{\lambda} \int_{\Omega_1} \nu \Sigma_{f,1} \Phi(x) p_{1,0}(x) dx + \frac{\alpha(L_0)}{\beta(L_0)} \Phi(L_0) = 0. \tag{2.28}
 \end{aligned}$$



Figure 2.4: Boundary functions.

The integrals appearing in equations (2.26), (2.27) and (2.28), are computed exactly using the Gauss-Legendre quadrature rule, as in the case of the CCM method.

2.1.2 Numerical results

The spectral methods for the neutron diffusion equation presented above, have been implemented into a computer code written in FORTRAN 77, which solves the resultant algebraic eigenvalue problem for an arbitrary approximation degree, K , in the polynomial expansion of the neutron flux. The algebraic eigenvalue problem has been solved to compute the dominant modes of the reactor using the Implicit Restarted Arnoldi Method [5].

In this section, first, we consider a homogeneous one energy group eigenvalue problem, which has analytical solution. Second, we study a more realistic 1D typical BWR reactor problem. For this second problem, we compare the results obtained using the three spectral methods exposed above, with the results obtained by the code KTRAC [39], widely used for reactor calculations.

2.1.2.1 Homogeneous eigenvalue problem

We consider a homogeneous slab of length 2 cm [35], in the approximation of one group of energy and vacuum boundary conditions, that is, $\alpha^- = 1$, $\alpha^+ = 1$, $\beta^- = 2$ and $\beta^+ = -2$. The nuclear cross sections for this problem are: $D = \frac{1}{3}$, $\Sigma_a = 0.1$, and $\nu\Sigma_f = 0.25$.

First, we present the analytical solution for this problem [35]. From equation (2.8) and the boundary conditions (2.10), defining the parameter

$$\tau = \frac{1}{D} \left(\frac{1}{\lambda} \nu\Sigma_f - \Sigma_a \right) ,$$

we obtain the following differential eigenvalue problem,

$$\begin{aligned} \frac{d^2\Phi}{dx^2}(x) + \tau\Phi(x) &= 0 , \quad x \in \Omega , \\ \Phi(x_0) - 2D\frac{d\Phi}{dx}(x_0) &= 0 , \\ \Phi(x_N) + 2D\frac{d\Phi}{dx}(x_N) &= 0 . \end{aligned} \quad (2.29)$$

There exist non trivial solutions for the eigenvalue problem if, and only if, $\tau > 0$. Thus, the eigenvectors of problem (2.29) are

$$\Phi(x) = c \left(\sin \left(t \frac{x}{x_N} \right) + \frac{2Dt}{x_N} \cos \left(t \frac{x}{x_N} \right) \right) ,$$

where $t = x_N\sqrt{\tau} > 0$ is a positive solution of the nonlinear equation

$$f(t) = \left(1 - \frac{D^2 4}{x_N^2} t^2 \right) \sin(t) + \frac{4D}{x_N} t \cos(t) = 0 ,$$

and the constant c is fixed once a normalization for the eigenvectors is chosen.

Once we obtain t , the zeros of $f(t)$, the eigenvalues of (2.29) are

$$\lambda = \frac{x_N \nu \Sigma_f}{x_N \Sigma_a + \frac{Dt^2}{x_N}} . \quad (2.30)$$

The first 2 dominant eigenvalues for this problem obtained from the analytical solution (2.30) are $\lambda_1 = 0.587489$ and $\lambda_2 = 0.149135$. In Table 2.1, we show the results obtained for these first 2 dominant eigenvalues using the spectral methods CCM, PCM and SEM for different values of K in the neutron flux expansion. We observe that for this problem all methods present a good convergence rate for the eigenvalues.

Table 2.1: Results for λ_1 and λ_2 for the homogeneous eigenvalue problem.

K	CCM		PCM		SEM	
	$\lambda_1(k_{\text{eff}})$	λ_2	$\lambda_1(k_{\text{eff}})$	λ_2	$\lambda_1(k_{\text{eff}})$	λ_2
4	0.587484	0.141509	0.587725	0.141509	0.587489	0.148478
5	0.587484	0.149100	0.587484	0.149677	0.587489	0.149134
6	0.587489	0.149100	0.587489	0.149100	0.587489	0.149134
7	0.587489	0.149135	0.587489	0.149136	0.587489	0.149135
8	0.587489	0.149135	0.587489	0.149135	0.587489	0.149135

To study the convergence of the computed eigenvectors, we fix the normalization constant imposing that the eigenvectors satisfy the normalization condition

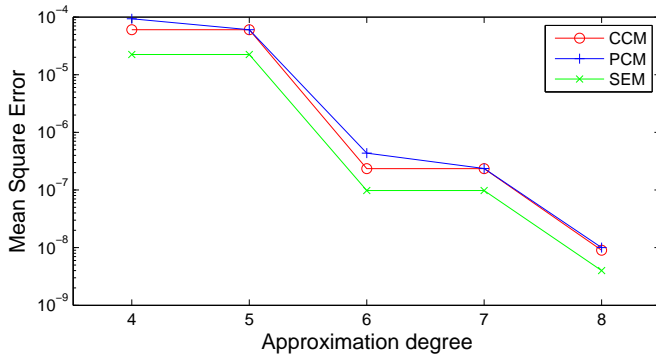
$$\int_{\Omega} |\Phi(x)| dx = 1 .$$

We define the mean square error

$$E(\Phi) = \left(\frac{1}{L_r} \int_{\Omega} (\Phi_{\text{ref}}(x) - \Phi(x))^2 dx \right)^{\frac{1}{2}} ,$$

where $\Phi_{\text{ref}}(x)$ is the eigenvector taken as reference and $\Phi(x)$ is the computed eigenvector.

In Figures 2.5 and 2.6 we show, in a logarithmic scale, the mean square error for different flux approximation degrees, K , for the computed eigenvectors corresponding to the eigenvalues λ_1 and λ_2 , respectively.

**Figure 2.5:** Error for the first eigenvector of the homogeneous reactor.

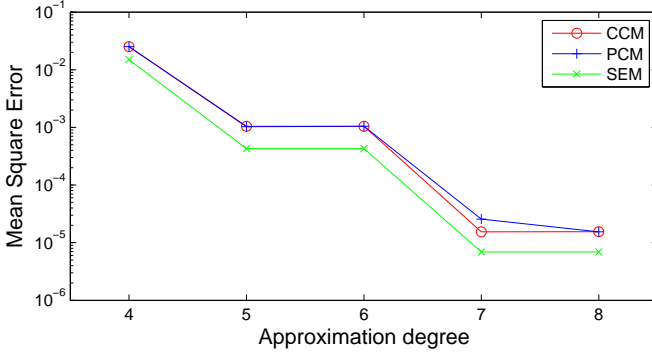


Figure 2.6: Error for the second eigenvector of the homogeneous reactor.

The behaviour of the mean square error is similar for the two eigenvectors, and although the three methods show similar results, the SEM is more accurate than the other two methods for the calculation of the first 2 eigenvectors.

2.1.2.2 1D typical BWR reactor

Now, we consider a more realistic problem, that consists of computing the dominant Lambda modes of a typical 1D reactor. This problem is formulated in the approximation of two groups of energy assuming that the neutrons are born in the fast group, and that there is no upscattering from the thermal to the fast group. In this way, the problem we deal with is to find the dominant eigenvalues and their corresponding eigenfunctions of the problem

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

where

$$\mathcal{L} = \begin{bmatrix} -\frac{d}{dx}(D_1 \frac{d}{dx}) + \Sigma_{a1} + \Sigma_{12} & 0 \\ -\Sigma_{12} & -\frac{d}{dx}(D_2 \frac{d}{dx}) + \Sigma_{a2} \end{bmatrix},$$

and

$$\mathcal{M} = \begin{bmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \\ 0 & 0 \end{bmatrix}, \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}.$$

We use zero-flux boundary conditions, i.e., the values $\beta^- = 0$ and $\beta^+ = 0$, that implies $\Phi_{1,2}(x_0) = \Phi_{1,2}(x_N) = 0$. Following the same methodology as the one presented above for the mono-energetic approximation, we have developed the spectral methods CCM, PCM and SEM for this problem.

The reactor is divided into 25 fuel assemblies and 2 reflector nodes, as it is shown in Figure 2.7. The nuclear cross sections for this problem have been obtained by

means of a consistent generation process [40] from the 3D cross sections of the reactor core, obtaining the cross sections shown in Table 2.2.

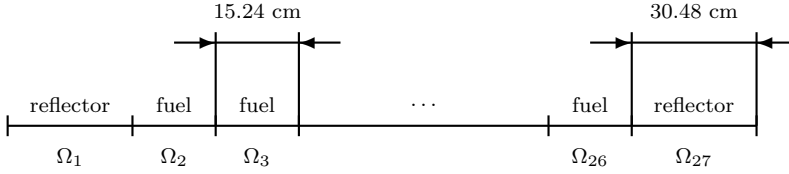


Figure 2.7: Spatial distribution of materials for the 1D BWR reactor.

Table 2.2: Nuclear cross sections for the 1D BWR reactor.

Mat.	$D_1(cm)$	$D_2(cm)$	$\Sigma_{a1}(cm^{-1})$	$\Sigma_{a2}(cm^{-1})$	$\Sigma_{12}(cm^{-1})$	$\nu\Sigma_{f1}(cm^{-1})$	$\nu\Sigma_{f2}(cm^{-1})$
Ω_1	1.045600	0.315390	0.006759	0.042792	0.026483	0.000000	0.000000
Ω_2	1.400000	0.344100	0.006123	0.034803	0.020791	0.002735	0.034955
Ω_3	1.424300	0.343700	0.008097	0.055085	0.018634	0.004232	0.073338
Ω_4	1.440100	0.349150	0.008074	0.053924	0.018297	0.003974	0.068992
Ω_5	1.463400	0.357540	0.008164	0.053309	0.017755	0.003855	0.067221
Ω_6	1.498000	0.369890	0.008157	0.053255	0.016920	0.003816	0.067254
Ω_7	1.540300	0.385070	0.008126	0.053497	0.015890	0.003814	0.068310
Ω_8	1.585000	0.401320	0.008084	0.053781	0.014817	0.003822	0.069648
Ω_9	1.620800	0.414780	0.008032	0.054029	0.014014	0.003826	0.070696
Ω_{10}	1.654100	0.427590	0.007985	0.054287	0.013313	0.003832	0.071755
Ω_{11}	1.692600	0.442740	0.007928	0.054566	0.012557	0.003839	0.072991
Ω_{12}	1.728100	0.457050	0.007867	0.054783	0.011887	0.003842	0.074049
Ω_{13}	1.744200	0.464010	0.007843	0.054977	0.011580	0.003846	0.074696
Ω_{14}	1.767600	0.473770	0.007796	0.055161	0.011145	0.003852	0.075503
Ω_{15}	1.816300	0.493390	0.007717	0.055270	0.010296	0.003854	0.076852
Ω_{16}	1.811800	0.492410	0.007721	0.055381	0.010358	0.003855	0.076873
Ω_{17}	1.823900	0.497610	0.007709	0.055496	0.010151	0.003858	0.077387
Ω_{18}	1.867100	0.515640	0.007630	0.055528	0.009468	0.003853	0.078503
Ω_{19}	1.870800	0.518060	0.007624	0.055557	0.009409	0.003850	0.078634
Ω_{20}	1.864900	0.516380	0.007651	0.055622	0.009500	0.003860	0.078783
Ω_{21}	1.890200	0.527490	0.007582	0.055656	0.009126	0.003876	0.079692
Ω_{22}	1.928500	0.544280	0.007458	0.055537	0.008572	0.003897	0.080804
Ω_{23}	1.896500	0.632430	0.007522	0.055653	0.009036	0.003972	0.080954
Ω_{24}	1.900800	0.535970	0.007310	0.055492	0.008983	0.004066	0.081710
Ω_{25}	1.961700	0.564160	0.007563	0.055242	0.008123	0.004142	0.081839
Ω_{26}	1.886100	0.577310	0.004319	0.030169	0.009894	0.002442	0.039040
Ω_{27}	2.214700	0.549020	0.012728	0.008695	0.011400	0.000000	0.000000

Table 2.3 shows the results for the first 2 dominant eigenvalues λ_1 and λ_2 obtained with the different spectral methods using different approximation degrees, K , in the flux expansion. In this case, we use as a reference the value for the dominant eigenvalue $\lambda_1 = k_{\text{eff}} = 1.005234$ and the neutron power distribution computed with the code KTRAC. We observe that all the spectral methods present a good convergence rate for the eigenvalues.

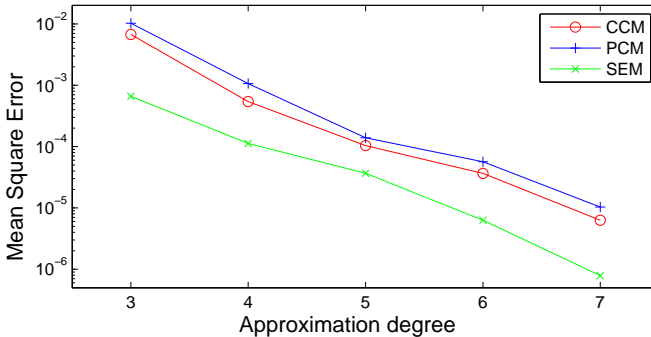
Table 2.3: 1D BWR results for $\lambda_1(k_{\text{eff}})$ and λ_2 .

K	CCM		PCM		SEM	
	$\lambda_1(k_{\text{eff}})$	λ_2	$\lambda_1(k_{\text{eff}})$	λ_2	$\lambda_1(k_{\text{eff}})$	λ_2
3	1.005257	0.994445	1.005275	0.994515	1.005236	0.994348
4	1.005232	0.994348	1.005233	0.994352	1.005235	0.994351
5	1.005234	0.994346	1.005233	0.994345	1.005235	0.994349
6	1.005234	0.994348	1.005234	0.994348	1.005234	0.994349
7	1.005234	0.994349	1.005234	0.994349	1.005234	0.994349

To compare the obtained eigenvectors corresponding the dominant eigenvalue, λ_1 , we compute the neutron power distribution

$$P = \frac{L_r (\nu \Sigma_{f1} \Phi_1 + \nu \Sigma_{f2} \Phi_2)}{\int_{\Omega} (\nu \Sigma_{f1} \Phi_1 + \nu \Sigma_{f2} \Phi_2) dx}.$$

In Figure 2.8, we show the mean square error for the neutron power distribution, $E(P)$, in terms of the approximation degree K in the flux expansion.

**Figure 2.8:** Mean square error for the power distribution of the 1D BWR reactor.

For this problem the SEM is again the most accurate method for the calculation of the power distribution of this reactor. Now, the mean square error of the SEM approximation is one order of magnitude smaller than the errors obtained with the others two methods for the same approximation degree.

Finally, in Figure 2.9 we show the spatial power distribution associated with the eigenvectors corresponding with the eigenvalues λ_1 and λ_2 .

Summarizing, we have that the calculations for the dominant Lambda modes of the core of a nuclear power reactor in multidimensional geometries requires a

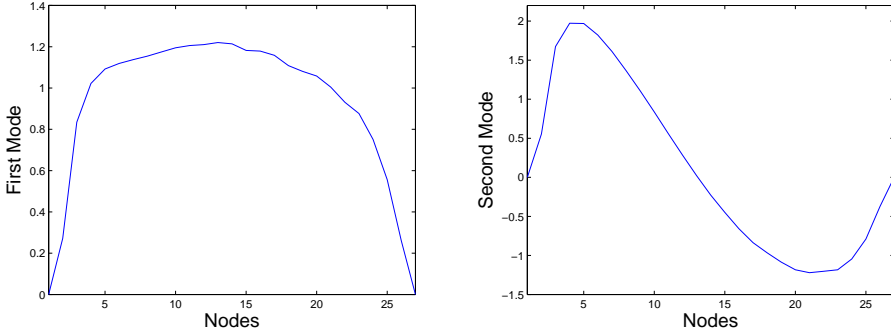


Figure 2.9: Spatial power distribution associated with the first two modes.

large amount of memory and calculation time to obtain reasonable results. This makes necessary to study different methodologies to discretize the neutron diffusion equation and to compute the dominant modes of a reactor in order to find the most efficient strategy to solve this problem. For this reason, we have compared three discretization methods for the Lambda modes problem for reactors in 1D geometry, with the aim to generalize the method with better behaviour to study reactors in a geometry 3D using an arbitrary mesh.

All the methods considered are based on the expansion of the neutron flux in terms of a continuous polynomial basis. Different methods are obtained considering different ways to approximate the neutron diffusion equation and the continuity conditions for the neutronic current. To test the performance of the methods, we have considered two benchmark problems, a homogeneous slab with vacuum boundary conditions and a typical 1D BWR reactor. SEM has shown the best results for the calculations of both the eigenvalues and the eigenvectors, being the difference higher in the 1D BWR reactor problem. Moreover, with this method we obtain symmetric diagonal dominant matrices. This is an important fact to solve large systems using iterative methods.

2.2 Two- and three-dimensional geometries

As it has been explained in Chapter 1, for a given configuration of a nuclear reactor core it is always possible to force its criticality dividing the neutron production rate due to fission by a positive number, λ , obtaining a neutron balance equation of the form

$$\mathcal{L}\Phi = \frac{1}{\lambda}\mathcal{M}\Phi, \quad (2.31)$$

where \mathcal{L} is the neutron loss operator and \mathcal{M} is the neutron production operator. Equation (2.31) is known as the Lambda modes equation.

The Lambda modes problem has been studied for reactors with rectangular geometry as the PWR and BWR [10, 11, 2, 6]. To discretize the problem for reactors with a hexagonal geometry, as for example the VVER reactors, a different strategy is necessary because the design of these reactors define a natural mesh with hexagonal cells instead of rectangular cells, as it is shown in Figure 2.10.

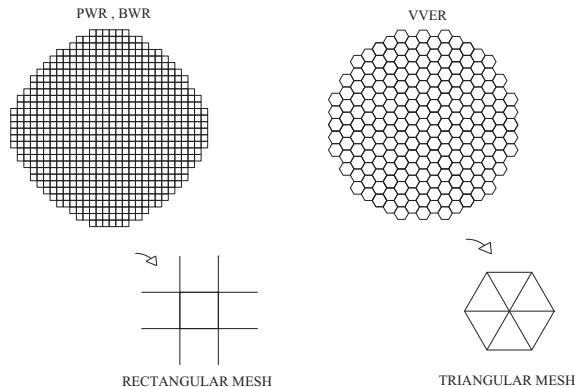


Figure 2.10: Core mesh for both VVER and PWR reactors.

Different methods have been proposed to solve the neutron diffusion equation on hexagonal geometry as, for example, the Fourier transform method [41], the conformal mapping method [34], the polynomial expansion nodal method [42], etc. All these methods compute the k_{eff} and the stationary neutron flux in the reactor core by means of a system of non linear equations, which is solved iteratively. To obtain a set of dominant modes it is necessary to approximate the initial differential eigenvalue problem (2.31), by a generalized algebraic eigenvalue problem. This can be done, for example, using a finite element method [43] or finite difference method [44]. On the other hand, for nuclear reactors the spatial mesh is naturally defined by the different materials defining the core, and for this reason, it is interesting to use a method that uses a fixed mesh and increases its accuracy without changing this mesh.

A Spectral Element Method is formulated for two- and three-dimensional geometries to approximate the solution of the neutron diffusion equation. This method is based on approximating the solution of the problem as a truncated expansion in terms of a suitable basis of polynomials. The accuracy obtained in the solution is controlled by means of the number of polynomials considered on the expansion and it is not necessary to refine the mesh to increase the accuracy of the method. A first attempt to develop one of these methods was presented in [45], but the resulting method is too expensive from the computational point of view because a large amount of unknowns are necessary to obtain reasonable accuracy in the solution. Here we follow a high order primal finite element method based on the division of each hexagon of the mesh into six equilateral triangles. This method uses as a

suitable basis of polynomials the modified Dubiner's polynomials [31, 46, 36], and increases its accuracy increasing the number of polynomials used in the solution expansions, leaving the initial mesh unaltered.

The first step to discretize a multidimensional reactor with hexagonal geometry, for dimensions $d = 2, 3$, is to divide each hexagonal prism or hexagonal element into six triangular prisms or triangular elements. The triangular prisms or triangular elements of the discretization will be denoted by Ω_e and will be referred as elements, independently of the dimension for the problem, $d = 2, 3$. To show the development of the Spectral Element Method, the monoenergetic approximation of the Lambda Modes problem has been considered,

$$-\vec{\nabla} D(\vec{r}) \vec{\nabla} \Phi(\vec{r}) + \Sigma_a(\vec{r}) \Phi(\vec{r}) = \frac{1}{\lambda} \nu \Sigma_f(\vec{r}) \Phi(\vec{r}) . \quad (2.32)$$

The generalization of the method to more energy groups is straightforward.

2.2.1 Variational Formulation

Similarly as we have done in the one-dimensional case, without lose of generality, the Lambda Modes equation (2.32) can be written as

$$-\vec{\nabla} \left(D(\vec{r}) \cdot \vec{\nabla} \Phi(\vec{r}) \right) + \Sigma_a(\vec{r}) \Phi(\vec{r}) = S(\vec{r}) , \quad (2.33)$$

where $\Phi(\vec{r})$ is the neutron flux, and the source term, $S(\vec{r})$, is defined as follows,

$$S(\vec{r}) = \frac{1}{\lambda} \nu \Sigma_f(\vec{r}) \Phi(\vec{r}) . \quad (2.34)$$

The domain defined by the reactor core will be called $\Omega = \bigcup_e \Omega_e$, and to solve the problem defined by equation (2.33), boundary conditions on the reactor boundary, $\partial\Omega$, are necessary. The surfaces defining $\partial\Omega$ are divided into two components, $\partial\Omega_0$ and $\partial\Omega_\beta$. $\partial\Omega_0$ is the part of the surface where zero flux boundary condition are imposed, and $\partial\Omega_\beta$ is the part of the surface where albedo-type boundary condition are required. Formally, albedo-type boundary condition can be written as [8]

$$\vec{n}(\vec{r}) \cdot D(\vec{r}) \vec{\nabla} \Phi(\vec{r}) + \frac{1}{2} \frac{1 - \beta}{1 + \beta} \Phi(\vec{r}) = 0 , \quad \vec{r} \in \partial\Omega_\beta , \quad (2.35)$$

where $\vec{n}(\vec{r})$ is the normal outward vector to the surface defining the boundary.

There exist different variational formulations for which the neutron diffusion equation is a stationary point of a suitable functional on a Sobolev space. The functional used here is [38]

$$\begin{aligned} \mathcal{F}(\Phi) &= \frac{1}{2} \iint_{\Omega} D(\vec{r}) \vec{\nabla} \Phi(\vec{r}) \cdot \vec{\nabla} \Phi(\vec{r}) d\vec{r} + \frac{1}{2} \iint_{\Omega} \Sigma_a(\vec{r}) \Phi^2(\vec{r}) d\vec{r} \\ &- \iint_{\Omega} S(\vec{r}) \Phi(\vec{r}) d\vec{r} + \int_{\partial\Omega_\beta} \frac{1}{4} \frac{1 - \beta}{1 + \beta} \Phi^2(\vec{r}) dS . \end{aligned} \quad (2.36)$$

where $\Phi(x, y)$ is defined in the Sobolev space

$$\mathcal{H}_{\partial\Omega_0}(\Omega) = \left\{ h : h \in L^2(\Omega), h(\vec{r}) = 0 \forall (\vec{r}) \in \partial\Omega_0 \text{ and } \vec{\nabla}h \in [L^2(\Omega)]^2 \right\} .$$

That is, the space of square-integrable functions over the domain Ω , which are equal to zero at $\partial\Omega_0$, and whose gradients have square-integrable components over the domain Ω . A stationary point of this functional can be defined by the relation [38]

$$\delta_{\delta\Phi}\mathcal{F} = \lim_{\epsilon \rightarrow 0} \left\{ \frac{d}{d\epsilon} \mathcal{F}(\Phi + \epsilon\delta\Phi) \right\} = 0 , \quad (2.37)$$

for all variations $\delta\Phi \in \mathcal{H}_{\partial\Omega_0}(\Omega)$. Thus, we obtain

$$\begin{aligned} \delta_{\delta\Phi}\mathcal{F}(\Phi) &= \iint_{\Omega} D(\vec{r}) \vec{\nabla}\Phi(\vec{r}) \cdot \vec{\nabla}\delta\Phi(\vec{r}) d\vec{r} + \iint_{\Omega} \Sigma_a(\vec{r}) \Phi(\vec{r}) \delta\Phi(\vec{r}) d\vec{r} \\ &\quad - \iint_{\Omega} S(\vec{r}) \delta\Phi(\vec{r}) d\vec{r} + \int_{\partial\Omega_\beta} \frac{1}{2} \frac{1-\beta}{1+\beta} \Phi(\vec{r}) \delta\Phi(\vec{r}) dS = 0 . \end{aligned} \quad (2.38)$$

The reactor domain Ω is split into subdomains Ω_e ($e = 1, \dots, N_t$), where it is assumed that the nuclear cross sections remain constant, being denoted by D_e and $\Sigma_{a,e}$. We also define $\partial\Omega_e$ as the polygonal that surrounds each subdomain and $\vec{n}_e(\vec{r})$ is the unitary vector perpendicular to $\partial\Omega_e$ at \vec{r} over the outward direction to Ω_e .

Now, we apply the vectorial identity $\vec{\nabla} \cdot (u\vec{\nabla}v) = (\vec{\nabla}u) \cdot (\vec{\nabla}v) + u(\vec{\nabla} \cdot \vec{\nabla}v)$ to the first term of the right hand side of equation (2.38), over a generic element Ω_e of the discretization, obtaining

$$\begin{aligned} \iint_{\Omega_e} D_e \vec{\nabla}\Phi(\vec{r}) \cdot \vec{\nabla}\delta\Phi(\vec{r}) d\vec{r} &= \iint_{\Omega_e} D_e \vec{\nabla} \cdot (\delta\Phi(\vec{r}) \vec{\nabla}\Phi(\vec{r})) d\vec{r} \\ &\quad - \iint_{\Omega_e} \delta\Phi(\vec{r}) \vec{\nabla} \cdot (D_e \vec{\nabla}\Phi(\vec{r})) d\vec{r} . \end{aligned}$$

Finally, we apply the Gauss divergence theorem over the element $\partial\Omega_e$, obtaining

$$\begin{aligned} \iint_{\Omega_e} D_e \vec{\nabla}\Phi(\vec{r}) \cdot \vec{\nabla}\delta\Phi(\vec{r}) d\vec{r} &= \int_{\partial\Omega_e} D_e \delta\Phi(\vec{r}) \vec{\nabla}\Phi(\vec{r}) \cdot \vec{n}_e(\vec{r}) dS \\ &\quad - \iint_{\Omega_e} \delta\Phi(\vec{r}) \vec{\nabla} \cdot (D_e \vec{\nabla}\Phi(\vec{r})) d\vec{r} . \end{aligned} \quad (2.39)$$

We split each polygonal $\partial\Omega_e$ into three components,

$$\partial\Omega_e = \partial U_e + \partial V_e + \partial W_e ,$$

where ∂U_e is the part of $\partial\Omega_e$ belonging to $\partial\Omega_\beta$, ∂V_e is the part of $\partial\Omega_e$ belonging to $\partial\Omega_0$, and ∂W_e is the other part not belonging to $\partial\Omega_\beta$ nor to $\partial\Omega_0$. That is, ∂U_e is the part of $\partial\Omega_e$ where albedo boundary conditions are imposed, ∂V_e is the part of $\partial\Omega_e$ where there is a zero flux boundary condition, and ∂W_e is composed of internal edges of the mesh.

Substituting equation (2.39) in equation (2.38), we obtain

$$\begin{aligned} \delta_\delta \mathcal{F}(\Phi) &= \sum_{e=1}^{N_t} \iint_{\Omega_e} \left(-\vec{\nabla} D_e \cdot \vec{\nabla} \Phi(\vec{r}) + \Sigma_{a,e} \Phi(\vec{r}) - S(\vec{r}) \right) \delta\Phi(\vec{r}) d\vec{r} \\ &+ \sum_{e=1}^{N_t} \int_{\partial U_e} \left(D_e \vec{\nabla} \Phi(\vec{r}) \cdot \vec{n}_e(\vec{r}) + \frac{1}{2} \frac{1-\beta}{1+\beta} \Phi(\vec{r}) \right) \delta\Phi(\vec{r}) dS \\ &+ \sum_{e=1}^{N_t} \int_{\partial W_e} \left(D_e \vec{\nabla} \Phi(x, y) \cdot \vec{n}_e(\vec{r}) \right) \delta\Phi(\vec{r}) dS = 0, \end{aligned} \quad (2.40)$$

$$\forall \delta\Phi(\vec{r}) \in \mathcal{H}_{\partial\Omega_0}(\Omega).$$

Equation (2.40) is the stationary condition for the functional $\mathcal{F}(\Phi)$.

This condition is distinguished from the natural conditions, as zero flux boundary conditions over $\partial\Omega_0$, which are imposed on the possible functions of space $\mathcal{H}_{\partial\Omega_0}(\Omega)$. Thus, $\Phi(\vec{r})$ is a stationary point of functional (2.36) with respect to an arbitrary variation $\delta\Phi(\vec{r})$ if, and only if, the following Euler conditions are fulfilled [38]:

1. The neutron flux satisfies the condition

$$-\vec{\nabla} \left(D(\vec{r}) \cdot \vec{\nabla} \Phi(\vec{r}) \right) + \Sigma_a(\vec{r}) \Phi(\vec{r}) = S(\vec{r}), \quad (2.41)$$

which is equation (2.33).

2. The neutron flux must satisfy the equation

$$D(\vec{r}) \vec{n}(\vec{r}) \cdot \vec{\nabla} \Phi(\vec{r}) + \frac{1}{2} \frac{1-\beta}{1+\beta} \Phi(\vec{r}) = 0, \quad (2.42)$$

if $\vec{r} \in \partial\Omega_\beta$. These are the albedo-type boundary conditions (2.35).

3. The neutron current normal to the interior surfaces among the sub domains has to be continuous

$$D_e \vec{\nabla} \Phi(\vec{r}) \cdot \vec{n}_e(\vec{r}) = -D_{e'} \vec{\nabla} \Phi(\vec{r}) \cdot \vec{n}_{e'}(\vec{r}). \quad (2.43)$$

2.2.2 Reference element

To develop the method, we use a change of variables mapping each one of the elements of the mesh, Ω_e , into a *Reference Domain*, Ω_{REF} . For two-dimensional problems, this Reference Domain is also known as the *Right Triangle* (see Figure 2.11), defined as

$$\Omega_{REF} = \{(x, y) / x \geq 0, y \geq 0, 0 \leq x + y \leq 1\}, \quad (2.44)$$

where we have used the conventions:

$$\begin{aligned} \text{vertex 1} &= (0, 0), & \text{edge 1} &= \{(t, 0) : t \in [0, 1]\}, \\ \text{vertex 2} &= (1, 0), & \text{edge 2} &= \{(1 - t, t) : t \in [0, 1]\}, \\ \text{vertex 3} &= (0, 1), & \text{edge 3} &= \{(0, t) : t \in [0, 1]\}. \end{aligned}$$

This change of variables, relating physical coordinates, (x, y) , with the coordinates

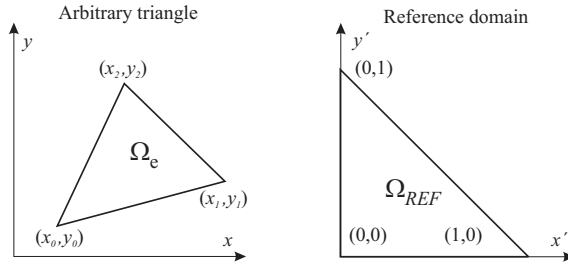


Figure 2.11: Two-dimensional change of variables from a arbitrary triangle to the Reference Domain.

of the Reference Domain, (x', y') , is given by

$$\begin{aligned} x &= x_0 + (x_1 - x_0)x' + (x_2 - x_0)y', \\ y &= y_0 + (y_1 - y_0)x' + (y_2 - y_0)y', \end{aligned} \quad (2.45)$$

and

$$\begin{aligned} x' &= a_1x + b_1y + c_1, \\ y' &= a_2x + b_2y + c_2, \end{aligned} \quad (2.46)$$

where a_i, b_i, c_i are the following constants depending on the coordinates of the vertices of each triangle,

$$\begin{aligned}
 a_1 &= \frac{-(y_0 - y_2)}{-x_1y_0 + x_2y_0 + x_0y_1 - x_2y_1 - x_0y_2 + x_1y_2}, \\
 a_2 &= \frac{y_0 - y_1}{-x_1y_0 + x_2y_0 + x_0y_1 - x_2y_1 - x_0y_2 + x_1y_2}, \\
 b_1 &= \frac{x_0 - x_2}{-x_1y_0 + x_2y_0 + x_0y_1 - x_2y_1 - x_0y_2 + x_1y_2}, \\
 b_2 &= \frac{-(x_0 - x_1)}{-x_1y_0 + x_2y_0 + x_0y_1 - x_2y_1 - x_0y_2 + x_1y_2}, \\
 c_1 &= \frac{x_2y_0 - x_0y_2}{-x_1y_0 + x_2y_0 + x_0y_1 - x_2y_1 - x_0y_2 + x_1y_2}, \\
 c_2 &= \frac{-(x_1y_0 - x_0y_1)}{-x_1y_0 + x_2y_0 + x_0y_1 - x_2y_1 - x_0y_2 + x_1y_2}.
 \end{aligned} \tag{2.47}$$

And for three-dimensional geometries, the *Reference Domain* is defined as the Cartesian product, $\Omega_{REF} = \Omega_{xy} \times \Omega_z$, where Ω_{xy} is the Reference Domain for two-dimensional geometries ($\Omega_{xy} = \{(x, y) / 0 \leq x, y \leq 1; x + y \leq 1\}$) and Ω_z is the Reference Domain for one-dimensional geometries, $\Omega_z = [0, 1]$ (see Figure 2.12).

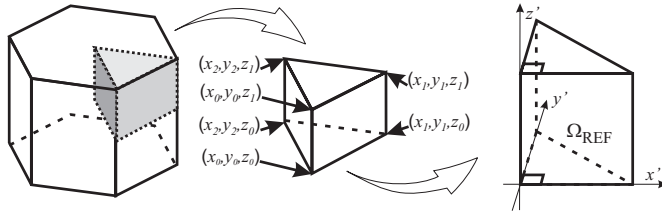


Figure 2.12: Three-dimensional change of variables from an arbitrary prism to the Reference domain.

We will denote by Φ_e the restriction of the neutron flux, Φ , to the element Ω_e of the mesh, and it is assumed that the solution of problem (2.32) for two- or three-dimensional geometries can be approximated by a finite expansion of the form [47],

$$\Phi_e(\vec{r}') = \sum_{i,j=0}^{i+j \leq K_{xy}} \phi_{e,ij} g_{ij}(x', y'), \quad \text{two-dimensions} \tag{2.48a}$$

$$\Phi_e(\vec{r}') = \sum_{i,j=0}^{i+j \leq K_{xy}} \sum_{k=0}^{K_z} \phi_{e,ijk} g_{ij}(x', y') h_k(z'), \quad \text{three-dimensions} \tag{2.48b}$$

where g_{ij} are elements of a polynomial basis over Ω_{xy} up to order K_{xy} , and h_k are elements of a polynomial basis over Ω_z up to order K_z .

The aim of the discretization method is to obtain a generalized algebraic eigenvalue problem which approximates the differential eigenvalue problem (2.32), where components of the eigenvectors are related with the coefficients $\phi_{e,ijk}$ ($\phi_{e,ij}$ for two-dimensional geometries) for the different elements of the mesh.

2.2.3 Polynomial basis

Any physical solution of the problem satisfies that the neutron flux has to be a continuous function over the domain defining the reactor core Ω . To obtain continuous solutions, the modified Dubiner's polynomial basis [31, 36] will be used, and some of the coefficients $\phi_{e,ijk}$ of the expansion (2.48) will be fixed to get that any function expanded in terms of these polynomials is continuous by construction [32, 46].

The modified Dubiner polynomials are defined on the Reference Domain (see Figure 2.11). Let g_{ij} be the polynomials on the XY plane, different kind of polynomials are distinguished: vertex polynomials, edge polynomials and interior polynomials (see Figure 2.13). The definition of these polynomials is the following

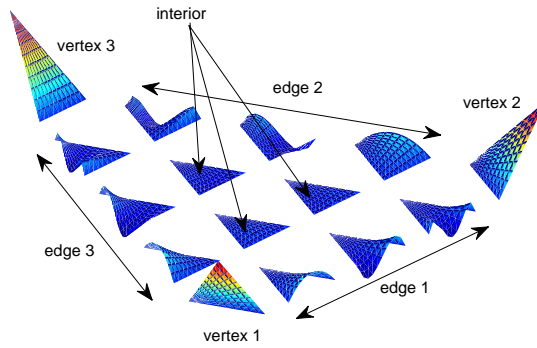


Figure 2.13: Modified Dubiner's polynomials for $K = 4$.

one [36]:

- Vertex polynomials

$$g_{11}(x', y') = 1 - x' - y' , \quad (\text{vertex 1}),$$

$$g_{21}(x', y') = x' , \quad (\text{vertex 2}),$$

$$g_{31}(x', y') = y' , \quad (\text{vertex 3}),$$

- Edge polynomials ($2 \leq j \leq K_{xy}$)

$$g_{1j}(x', y') = (1 - x' - y') x' Q_{j-2}^{1,1}(x', y') 2^{\frac{3}{2}}, \quad (\text{edge 1}),$$

$$g_{2j}(x', y') = x' y' P_{j-2}^{1,1}(2y' - 1) 2^{\frac{3}{2}}, \quad (\text{edge 2}),$$

$$g_{3j}(x', y') = y' (1 - x' - y') P_{j-2}^{1,1}(1 - 2y') 2^{\frac{3}{2}}, \quad (\text{edge 3}),$$

- Interior polynomials ($4 \leq i, 1 \leq j, i + j \leq K_{xy} + 2$)

$$g_{ij}(x', y') = (1 - x' - y') x' y' Q_{i-4}^{1,1}(x', y') P_{j-1}^{2i-5,1}(2y' - 1) 2^j, \quad (\text{interior}).$$

where

$$Q_i^{1,1}(x', y') = P_i^{1,1} \left(\frac{2x'}{1 - y'} - 1 \right) (1 - y')^i,$$

and $P_i^{\alpha,\beta}(z)$ are the Jacobi polynomials, orthogonal on $[-1, 1]$ [30]. The evaluation of these polynomials at a given point is carried out using the recurrence relations presented in the Appendix B.1.

Similarly, $h_k(z')$ are the polynomials in the Z axis, defined as follows:

$$h_0(z') = 1 - z', \quad (\text{vertex})$$

$$h_1(z') = z', \quad (\text{vertex})$$

$$h_i(z') = (1 - z') z' P_{i-2}^{1,1}(2z' - 1) 2^{\frac{3}{2}}, \quad (\text{interior})$$

where $2 \leq i \leq K_z$ and $P_j^{1,1}(x)$ is the Jacobi polynomial of order j .

The product of each type of polynomial on the XY plane by the different type of polynomials on the Z axis defines different kind of global polynomials in the space (vertex, edge, face and interior polynomials). The coefficients of the vertex, edge and face polynomials associated with different elements of the mesh, Ω_e which share a common vertex, edge or face, are selected to be the same coefficient, in such a way that the continuity of the solution is assured over the whole domain [36].

One-dimensional polynomials have been already explained at the first part of this chapter, when checking the different Spectral Element Methods to use with more general geometries. Thus, to build the three-dimensional methods it remains to explain how the two-dimensional Dubiner's polynomials work. First the notation is simplified, where polynomials $g_{mn}(x', y')$ are denoted with only one subindex i , $g_i(x', y')$, with $i = 1, \dots, M_K$, where

$$M_K = (K + 1)(K + 2)/2,$$

the total number of modified Dubiner's polynomials until the degree K used in the expansions. This order is obtained numbering for a given element, e , first the polynomial of vertex 1, after this, the polynomials of the edge 1 in ascent

order with the polynomial degree, after the second vertex polynomial followed of the polynomials of the second edge with the same criterion, the third vertex polynomial and the polynomials of the third edge with the same order as before, and then the interior polynomials (see in Figure 2.14).

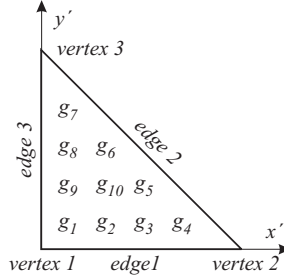


Figure 2.14: Numbering the Modified Dubiner's Polynomials.

These polynomials are defined over each element of the spatial discretization, Ω_e , by means of the change of variables from the physical element to the Reference Domain, and we rename the coefficients being coherent with the notation used for the polynomials. In this way, $g_{e,i}(x', y')$, the polynomial i over the element Ω_e , is associated to the coefficient $\phi_{e,i}$. These coefficients will be the *local unknowns* of the problem.

Following this notation, we write the neutron flux over each element defining the reactor domain in the physical coordinates (x, y) as

$$\Phi_e(x, y) = \sum_{i=1}^{M_K} \phi_{e,i} g_{e,i}(x, y), \quad (2.49)$$

and in the coordinates (x', y') of the reference element as

$$\Phi_e(x', y') = \sum_{i=1}^{M_K} \phi_{e,i} g_i(x', y'),$$

where the coordinates (x', y') are related with the coordinates (x, y) by means of the change of variables (2.45).

Now, we establish relations among coefficients of some polynomials corresponding to adjacent triangles, to assure the continuity of the neutron flux. Eliminating the redundant coefficients of expansions (2.48), we obtain a reduction in the number of unknowns necessary to describe the problem. This new set of coefficients will be called the *global unknowns*.

To explain this process, we use an example shown in Figure 2.15. In this Figure, a typical configuration for two adjacent interior triangles, Ω_{e_1} and Ω_{e_2} , using a degree $K = 2$ in the polynomial expansions is shown.

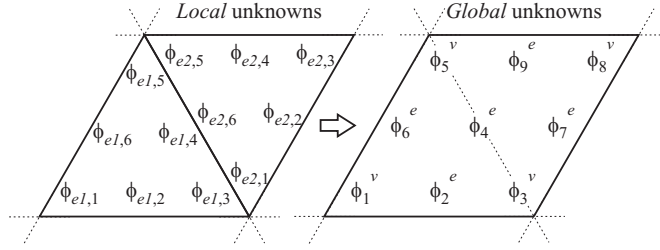


Figure 2.15: Continuity conditions for vertex and edge polynomials for $K = 2$.

Given a vertex of a triangle, only the corresponding vertex polynomial takes a value different from zero on this vertex. Thus, the continuity of the neutron flux at this vertex can be assured fixing the coefficient of the corresponding vertex polynomial on a first triangle, and making the coefficients associated to the same vertex polynomial of the others triangles, that have this vertex in common, equal to the coefficient of the first triangle. For this reason, for each internal vertex of the mesh, we have only one *global* unknown to be determined.

In the same way, for a given edge of a triangle, the only polynomials that take a value different from zero on this edge are the corresponding edge polynomials. The continuity for the neutron flux at the inner edges of the mesh is assured fixing the coefficients of the edge polynomials of the first triangle and making the corresponding coefficients of the edge polynomials of the adjacent triangle equal to the fixed edge coefficients for the first triangle.

At the external vertices and edges belonging to $\partial\Omega_0$, the coefficients of the corresponding polynomials are set equal to zero. In this way, we must determine

$$N_u = N_{iv} + N_{ie}(K - 1) + N_t \frac{(K - 1)(K - 2)}{2},$$

global unknowns, where N_{iv} is the number of internal vertex of the mesh, N_{ie} is the number of internal edges and N_t is the number of triangles of the spatial discretization.

The reduction of unknowns from the local unknowns is carried out with the aid of an order array relating the local unknowns with the global unknowns after the reduction process. An example for $K = 2$ with two adjacent triangles is shown in

equation (2.50).

$$\begin{array}{c}
 \textit{Local unknowns} \\
 \left[\begin{array}{c} \phi_{e_{1,1}} \\ \phi_{e_{1,2}} \\ \phi_{e_{1,3}} \\ \phi_{e_{1,4}} \\ \phi_{e_{1,5}} \\ \phi_{e_{1,6}} \\ \hline \phi_{e_{2,1}} \\ \phi_{e_{2,2}} \\ \phi_{e_{2,3}} \\ \phi_{e_{2,4}} \\ \phi_{e_{2,5}} \\ \phi_{e_{2,6}} \end{array} \right] \left[\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ \hline 3 \\ 7 \\ 8 \\ 9 \\ 5 \\ 4 \end{array} \right] \Rightarrow \begin{array}{c}
 \textit{Global unknowns} \\
 \left[\begin{array}{c} \phi_1^v \\ \phi_2^e \\ \phi_3^v \\ \phi_4^e \\ \phi_5^v \\ \phi_6^e \\ \phi_7^e \\ \phi_8^v \\ \phi_9^e \end{array} \right] \left[\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \right]
 \end{array} \quad (2.50)
 \end{array}$$

where ϕ_j^v refer to the global vertex coefficient associated with an interior vertex and ϕ_j^e refer to global edge coefficients corresponding to an interior edge. The coefficients corresponding to interior polynomials are unaltered after the reduction process.

In the following, we will denote by ϕ_j^i the global interior coefficients corresponding to an interior polynomial, and by ϕ_j^b the global boundary coefficient associated with a boundary polynomial, i.e., the polynomials which are different to zero at the boundary Ω_β of the reactor core.

2.2.4 Building the matrices

The equations defining the algebraic eigenvalue problem that approximates the Lambda Modes Problem are obtained using expansions (2.49) for the neutron flux in the functional (2.36), and making equal to zero the derivatives with respect to the global coefficients. To explain how it works, different polynomials corresponding to the different kind of functions defined at the two-dimensional plane are used. It has to be distinguished among derivatives with respect to the coefficients associated to an interior vertex, to an interior edge, or to a vertex and an edge belonging to $\partial\Omega_\beta$.

When we consider a coefficient associated with an internal vertex, ϕ_j^v , as we have already mentioned, the only polynomials that will be different from zero at this vertex are the vertex polynomials corresponding to this vertex (see Figure 2.16), for each triangle containing this vertex. Thus, when we derive respect to ϕ_j^v , which is equal to the local coefficients $\phi_{e_{1,i_1}}$, $\phi_{e_{2,i_2}}$, $\phi_{e_{3,i_3}}$, $\phi_{e_{4,i_4}}$, $\phi_{e_{5,i_5}}$ and $\phi_{e_{6,i_6}}$

we obtain an equation of the form

$$\begin{aligned} & \sum_{(e,i) \in \mathcal{I}_j^v} \left(D_e \iint_{\Omega_e} \vec{\nabla} g_{e,i} \cdot \vec{\nabla} \Phi_e \, dx \, dy + \Sigma_{a,e} \iint_{\Omega_e} g_{e,i} \Phi_e \, dx \, dy \right) \\ &= \sum_{(e,i) \in \mathcal{I}_j^v} \left(\frac{1}{\lambda} \nu \Sigma_{f,e} \iint_{\Omega_e} g_{e,i} \Phi_e \, dx \, dy \right), \end{aligned} \quad (2.51)$$

where $\mathcal{I}_j^v = \{(e_1, i_1), (e_2, i_2), (e_3, i_3), (e_4, i_4), (e_5, i_5), (e_6, i_6)\}$, the set of pairs of indices that contain the six vertex polynomials showed in Figure 2.16.

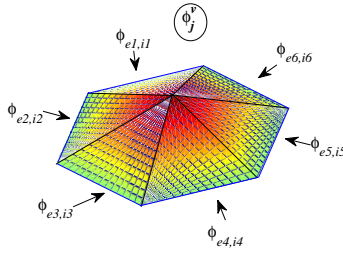


Figure 2.16: Vertex polynomials associated with an internal vertex and their coefficients. The corresponding global vertex coefficient is shown inside a circle.

When we derive with respect to an internal edge coefficient, ϕ_j^e , shared by the elements Ω_{e_1} and Ω_{e_2} , there are two edge polynomials that have their coefficients fixed to this global edge coefficient (see Figure 2.17), and we obtain

$$\begin{aligned} & \sum_{(e,i) \in \mathcal{I}_j^e} \left(D_e \iint_{\Omega_e} \vec{\nabla} g_{e,i} \cdot \vec{\nabla} \Phi_e \, dx \, dy + \Sigma_{a,e} \iint_{\Omega_e} g_{e,i} \Phi_e \, dx \, dy \right) \\ &= \sum_{(e,i) \in \mathcal{I}_j^e} \left(\frac{1}{\lambda} \nu \Sigma_{f,e} \iint_{\Omega_e} g_{e,i} \Phi_e \, dx \, dy \right), \end{aligned} \quad (2.52)$$

where $\mathcal{I}_j^e = \{(e_1, i_1), (e_2, i_2)\}$ is the set of pairs of indices that contain the two edge polynomials showed in Figure 2.17.

The coefficient associated to an interior polynomial is itself a global interior coefficient (see Figure 2.18) and, thus, when we take the derivatives with respect to a global interior coefficient we obtain

$$D_e \iint_{\Omega_e} \vec{\nabla} g_{e,i} \cdot \vec{\nabla} \Phi_e \, dx \, dy + \Sigma_{a,e} \iint_{\Omega_e} g_{e,i} \Phi_e \, dx \, dy = \frac{1}{\lambda} \nu \Sigma_{f,e} \iint_{\Omega_e} g_{e,i} \Phi_e \, dx \, dy, \quad (2.53)$$

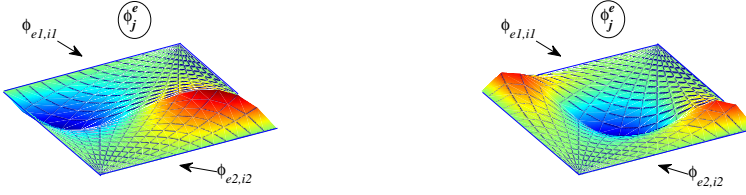


Figure 2.17: Edge polynomials of third order (left) and fourth order (right) whose coefficients are fixed to a global edge coefficient (inside a circle) and their corresponding coefficients.



Figure 2.18: Interior polynomials of different order whose coefficients are fixed to a global interior coefficient (inside a circle) and their corresponding coefficients.

obtaining $N_t(K-1)(K-2)/2$ constrains from interior polynomials.

If we deal with zero flux boundary conditions, the global boundary coefficients associated with polynomials that do not take the value zero over Ω_0 are set to zero. On the other hand, if we deal with albedo type boundary conditions, we take the derivatives with respect to the boundary global coefficients, associated with their respective vertex or edge polynomials (see Figure 2.19), obtaining equations of the form

$$\begin{aligned}
 & \sum_{(e,i) \in \mathcal{I}_j^b} \left(D_e \iint_{\Omega_e} \vec{\nabla} g_{e,i} \cdot \vec{\nabla} \Phi_e \, dx \, dy + \Sigma_{a,e} \iint_{\Omega_e} g_{e,i} \Phi_e \, dx \, dy \right) \\
 = & \sum_{(e,i) \in \mathcal{I}_j^b} \left(\frac{1}{\lambda} \nu \Sigma_{f,e} \iint_{\Omega_e} g_{e,i} \Phi_e \, dx \, dy + \int_{\partial\Omega_\beta} \frac{1}{2} \frac{1-\beta}{1+\beta} g_{e,i} \Phi_e \, dl \right), \quad (2.54)
 \end{aligned}$$

where $\mathcal{I}_j^b = \{(e_1, i_1), (e_2, i_2), (e_3, i_3), (e_4, i_4)\}$ in the case of a global boundary coefficient such as the one shown on the left of Figure 2.19, or $\mathcal{I}_j^b = \{(e, i)\}$ in the case of a global boundary coefficient such as the one showed on the right of Figure 2.19.

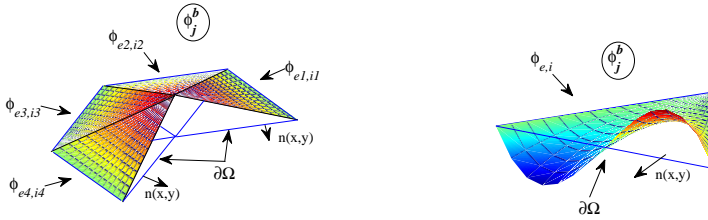


Figure 2.19: Vertex (left) and edge (right) polynomials and their corresponding coefficients, which are fixed to a global boundary coefficient (inside a circle).

To obtain algebraic expressions involving the global coefficients from the equations presented above, we must calculate the integrals present in these equations. The process used to calculate these integrals is presented in the Appendix B.2.

2.2.5 Algebraic problem

With the discretization method exposed above, we approximate the two energy groups approximation of the neutron diffusion equation (2.31) by a generalized algebraic eigenvalue problem with the following block structure

$$\begin{bmatrix} L_{11} & 0 \\ -L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} M_{11} & M_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

where ϕ_1 is a vector with the global unknowns corresponding to the neutron flux for the fast group, and ϕ_2 is a vector with the global unknowns of the thermal flux.

To solve this problem, it is reduced to the ordinary eigenvalue problem

$$L_{11}^{-1} (M_{11} + M_{12} L_{22}^{-1} L_{21}) \phi_1 = \lambda \phi_1 ,$$

which is solved for the dominant eigenvalues and their corresponding eigenvectors using the Implicit Restarted Arnoldi method [5]. For this purpose it is necessary to solve linear systems associated with blocks L_{11} and L_{22} . We will solve these systems by means of an iterative scheme as the preconditioned GMRES method [48].

Particularly, a reordering of matrices L_{11} and L_{22} to reduce the fill-in, together with an incomplete LU factorization of these matrices is used for the preconditioning [48, 49].

2.2.6 Numerical Results

To check the behaviour of the Spectral Element Method, the dominant Lambda Modes for the different two- and three-dimensional benchmark problems reported in [34] have been calculated. Particularly, five two-dimensional problems have been studied, the IAEA with and without reflector, the VVER-1000, the VVER-440 and the HWR. Also, the three-dimensional problems VVER-440 and VVER-1000 have been studied. We study all these problems to because different behaviour of the numerical methods can be shown at different benchmark problems, and then an exhaustive test has been done. The references for the k_{eff} and for the power distribution, calculated by means of the finite difference code DIF3D, have been obtained from [34].

We take into account different kinds of boundary conditions to solve the problems, as zero flux boundary conditions, i.e.,

$$\Phi(\vec{r}) = 0 \quad \forall(\vec{r}) \in \partial\Omega , \quad (2.55)$$

and albedo boundary conditions of the form

$$D(\vec{r})\vec{n}(\vec{r}) \cdot \vec{\nabla}\Phi(\vec{r}) + \frac{1}{2} \frac{1-\beta}{1+\beta} \Phi(\vec{r}) = 0 \quad \forall(\vec{r}) \in \partial\Omega , \quad (2.56)$$

where β is the parameter that determines the albedo. In particular $\beta = 0.0$ simulates vacuum boundary conditions.

We take as a reference the results of the k_{eff} and the power distribution computed using the finite difference code DIF3D, where for the HWR problem a mesh of 384 triangles per hexagon was used, and for the rest of the problems a mesh of 864 triangles per hexagon was used.

The neutron power distribution in a reactor core is computed using the expression

$$P(\vec{r}) = \frac{A(\Omega) (\nu\Sigma_{f1}(\vec{r})\Phi_1(\vec{r}) + \nu\Sigma_{f2}(\vec{r})\Phi_2(\vec{r}))}{\iint_{\Omega} (\nu\Sigma_{f1}(\vec{r})\Phi_1(\vec{r}) + \nu\Sigma_{f2}(\vec{r})\Phi_2(\vec{r})) d\vec{r}} ,$$

where $A(\Omega)$ is defined as the area (or volume) of the region defining the core, Ω , $\Phi_1(\vec{r})$ is the neutron flux for the fast group and $\Phi_2(\vec{r})$ is neutron flux for the thermal group. From the neutron power distribution, we define the average neutron power for each element of the spatial discretization as

$$P_e = \frac{1}{A(\Omega_e)} \iint_{\Omega_e} P(\vec{r}) d\vec{r} ,$$

where $A(\Omega_e)$ is the area (or volume) for the element Ω_e , i.e., $A(\Omega_e) = \iint_{\Omega_e} d\vec{r}$. All these integrals reference double and triple integrals, respectively, for two-dimensional and three-dimensional geometries.

The errors showed for the power distribution over the hexagonal nodes, P_i , with respect to the reference solution, P_i^{ref} , have been calculated in the following form,

$$\begin{aligned} \epsilon_{\max}(\%) &= \max_i \left| \frac{P_i - P_i^{\text{ref}}}{P_i^{\text{ref}}} \right| \times 100, \\ \bar{\epsilon}(\%) &= \frac{1}{N_e} \sum_{i=1}^{N_e} \left| \frac{P_i - P_i^{\text{ref}}}{P_i^{\text{ref}}} \right| \times 100, \quad N_e = \text{Number of elements} \quad (2.57) \end{aligned}$$

For all the problems we will show tables with the different values of the k_{eff} obtained with different values of the degree K_{XY} and K_Z used in the neutron flux expansions. Also the difference between the obtained result and the reference value are presented. This difference is denoted as Δk_{eff} in *pcm* (percent-milli, i.e., 10^{-5}).

We also present results of the subcritical eigenvalues and the symmetry patterns of the power distribution associated with the subcritical modes computed with the spectral element method, for each one of the reactor problems.

2.2.6.1 Two-dimensional IAEA problem without reflector

This problem is a modification of the PWR *benchmark* problem IAEA for rectangular elements. The core has 13 fuel elements across its diameter, as it is shown in Figure 2.20.

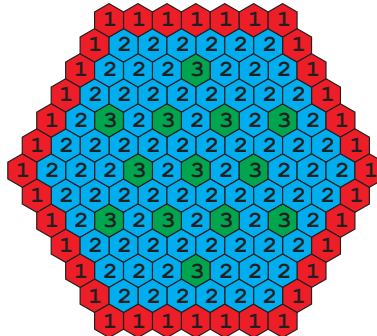


Figure 2.20: Geometry of the IAEA without reflector problem.

This problem has 13 control rods inserted, and has a 1/12 reflective symmetry but, as the subcritical modes do not maintain this symmetry, the computations

are performed for all the problems considering the whole reactor. The assembly pitch is 20.0 cm. The reflector is not included in the core, and we will consider vacuum boundary conditions, that is, albedo boundary conditions with $\beta = 0$. The nuclear cross sections for this problem are shown in Table 2.4.

Table 2.4: Cross sections of the IAEA problem.

Material	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.5	0.01	0.02	0.0
	2	0.4	0.08		0.135
2	1	1.5	0.01	0.02	0.0
	2	0.4	0.085		0.135
3	1	1.5	0.01	0.02	0.0
	2	0.4	0.13		0.135
4	1	1.5	0.0	0.04	0.0
	2	0.4	0.01		0.0

Table 2.5 shows the different results obtained for the k_{eff} of this reactor core, together with the dimension of the blocks L_{11} and L_{22} , N, and the nonzero elements of these matrices, NNZ.

Table 2.5: k_{eff} results for the IAEA reactor without reflector.

K_{xy}	$\lambda_1(k_{\text{eff}})$	$\Delta k_{\text{eff}}(pcm)$	$\epsilon_{\text{max}}(\%)$	$\bar{\epsilon}(\%)$	N	NNZ
1	0.9733475	472.9429	9.05	2.44	343	2251
2	0.9775987	47.8308	0.87	0.37	1447	15691
3	0.9780083	6.8649	0.30	0.08	3313	50091
4	0.9780645	1.2446	0.29	0.04	5941	110739
5	0.9780735	0.3492	0.29	0.03	9331	194489
6	0.9780754	0.1604	0.29	0.03	13483	297581
Ref.	0.9780770					

Figure 2.21 shows the neutronic power distribution for each hexagon computed with the spectral element method (SEM) using a degree $K_{xy} = 6$ in the expansions, together with the reference result and the percentage of the relative error on each hexagon.

In Table 2.6, the results obtained for the first three subcritical eigenvalues computed with the SEM method using different values of the degree K_{xy} are presented.

A symmetry pattern for the power distribution associated with the four dominant modes of this reactor is shown in Figure 2.22.

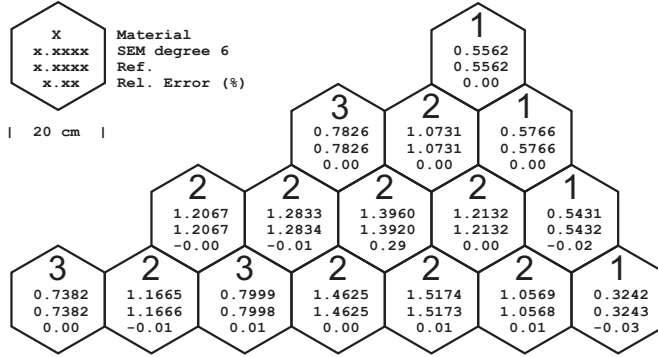


Figure 2.21: Power distribution for the IAEA problem without reflector.

Table 2.6: First 3 subcritical eigenvalues for the IAEA problem without reflector.

K_{xy}	λ_2	λ_3	λ_4
1	0.9587564	0.9587564	0.9338572
2	0.9627567	0.9627567	0.9380710
3	0.9631196	0.9631196	0.9383851
4	0.9631683	0.9384269	0.9196477
5	0.9631771	0.9631771	0.9384355
6	0.9631791	0.9631791	0.9384376

2.2.6.2 Two-dimensional IAEA problem with reflector

This problem is the same as the previous one, except that in this problem an additional layer of reflector surrounding the core is included, as shown in Figure 2.23. Also, vacuum boundary conditions are considered and the nuclear cross sections presented in Table 2.4.

Table 2.7 shows the different results obtained for the k_{eff} of this reactor core.

Table 2.7: k_{eff} results for the IAEA reactor with reflector.

K_{xy}	$\lambda_1(k_{\text{eff}})$	$\Delta k_{\text{eff}}(\text{pcm})$	$\epsilon_{\text{max}}(\%)$	$\bar{\epsilon}(\%)$	N	NNZ
1	1.0104126	-490.5520	20.95	7.70	463	3067
2	1.0062264	-71.9450	2.76	1.09	1939	21199
3	1.0055754	-6.8360	0.31	0.12	4429	67423
4	1.0055166	-0.9680	0.04	0.02	7933	148755
5	1.0055102	-0.3270	0.02	0.01	12451	260721
6	1.0055096	-0.2590	0.01	0.00	17983	398409
Ref.	1.0055070					

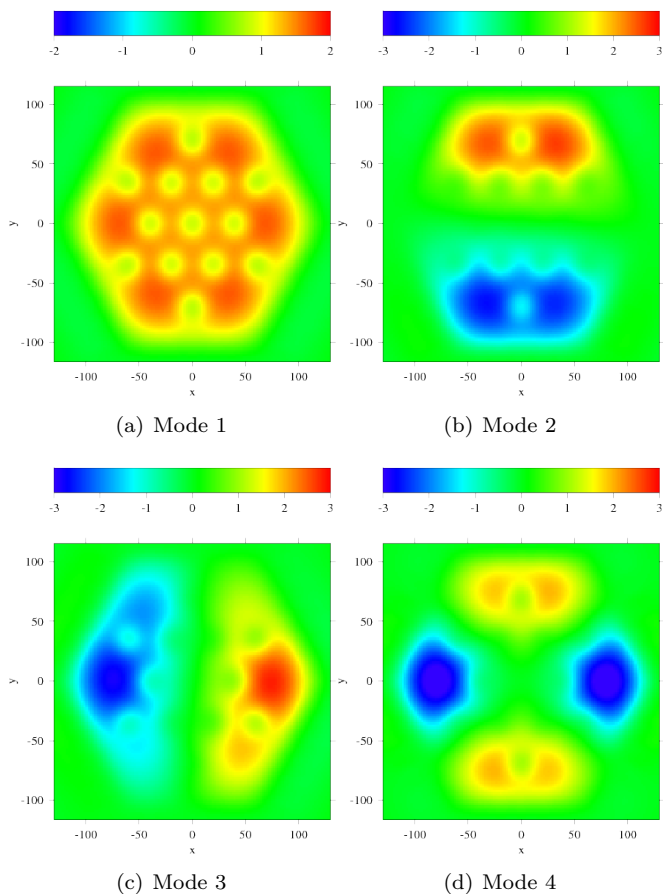


Figure 2.22: Symmetry pattern for the four dominant modes of the IAEA problem without reflector.

Figure 2.24 shows the neutronic power distribution for each hexagon computed with the SEM method using a degree $K_{xy} = 6$ in the flux expansions, together with the reference result and the percentage of the relative error on each hexagon.

In Table 2.8, the results obtained for the first three subcritical eigenvalues computed with the SEM method using different values of the degree K_{xy} , are presented.

A symmetry pattern for the four dominant modes of this problem is shown in Figure 2.25.

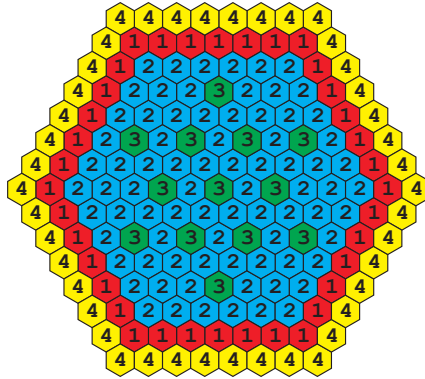


Figure 2.23: IAEA problem with reflector geometry.

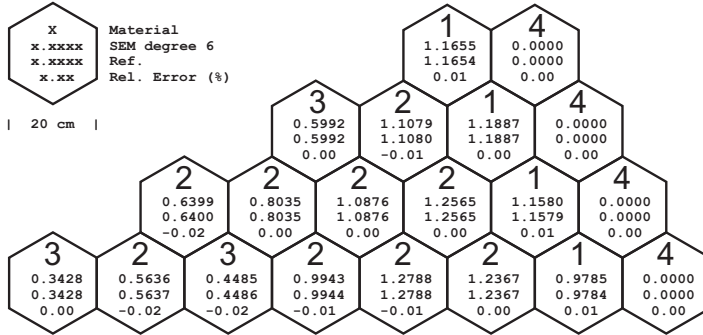


Figure 2.24: Power distribution for the IAEA problem with reflector.

Table 2.8: First 3 subcritical eigenvalues for the IAEA problem with reflector.

K_{xy}	λ_2	λ_3	λ_4
1	1.0024229	1.0024229	0.9836710
2	0.9973698	0.9973698	0.9778537
3	0.9965746	0.9965746	0.9768964
4	0.9964991	0.9964991	0.9768022
5	0.9964908	0.9964908	0.9767917
6	0.9964899	0.9964899	0.9767904

2.2.6.3 Two-dimensional VVER-1000 problem

This core is a VVER-1000 type with 15 fuel elements across the diameter with 25 control rods inserted. The geometry of this problem is shown in Figure 2.26.

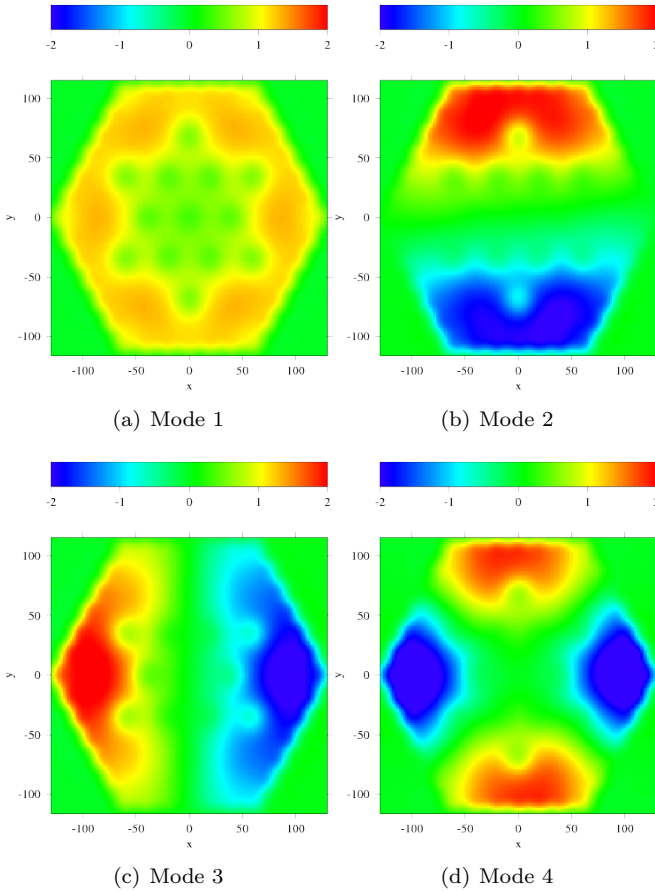


Figure 2.25: Symmetry pattern for the four dominant modes of the IAEA problem with reflector.

The core has an assembly pitch of 23.60 cm. The reflector is not explicitly modeled, and vacuum boundary conditions are assumed. The nuclear cross sections of this problem are shown in Table 2.9.

Table 2.10 shows the different results obtained for the k_{eff} of this reactor core.

Figure 2.27 shows the neutronic power distribution for each hexagon computed with the SEM method using a degree $K_{xy} = 6$ in the flux expansions, together with the reference result and the percentage of the relative error on each hexagon.

In Table 2.11, the results obtained for the first three subcritical eigenvalues computed with the SEM method using different values of the degree K_{xy} are presented.

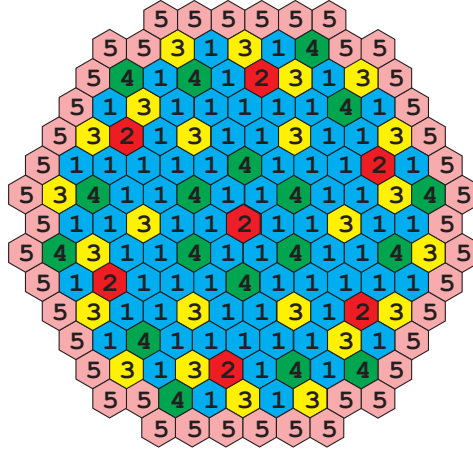

Figure 2.26: Geometry of the VVER-1000 reactor.

Table 2.9: Cross sections of the VVER-1000 problem

Fuel	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.38320	0.0083859	0.0164977	0.00481619
	2	0.386277	0.0673049		0.0846154
2	1	1.38299	0.0115490	0.0147315	0.00466953
	2	0.389403	0.0810328		0.0852264
3	1	1.39522	0.0089441	0.0156219	0.00604889
	2	0.386225	0.0844801		0.1194280
4	1	1.39446	0.0119932	0.0140185	0.00591507
	2	0.387723	0.0989670		0.1204970
5	1	1.39506	0.0091160	0.0154981	0.00640256
	2	0.384492	0.0893878		0.1292810

Table 2.10: k_{eff} results for the VVER-1000 problem.

K_{xy}	$\lambda_1(k_{\text{eff}})$	$\Delta k_{\text{eff}}(pcm)$	$\epsilon_{\text{max}}(\%)$	$\bar{\epsilon}(\%)$	N	NNZ
1	1.0048301	165.4840	10.97	4.42	445	2941
2	1.0063968	8.8270	1.27	0.51	1867	20371
3	1.0064499	3.5050	0.28	0.11	4267	66353
4	1.0064518	3.3210	0.13	0.07	7645	150987
5	1.0064534	3.1620	0.10	0.06	12001	262207
6	1.0064540	3.1040	0.09	0.06	17335	398449
Ref.	1.0064850					

A symmetry pattern for the four dominant modes of this reactor core is shown in Figure 2.28.

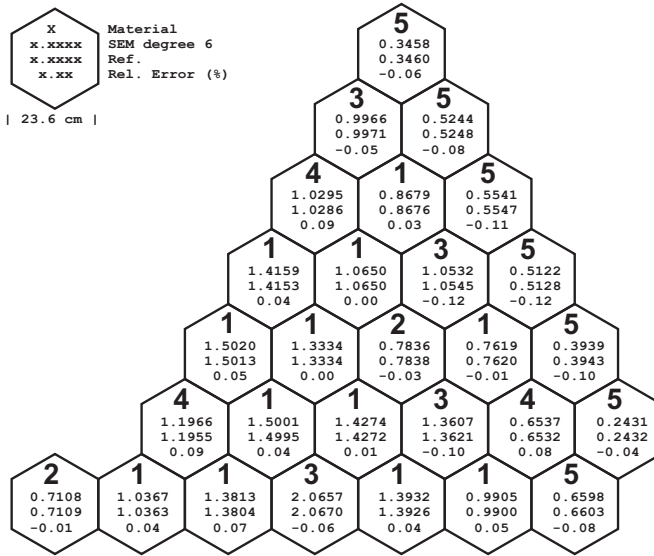


Figure 2.27: Power distribution for the VVER-1000 reactor.

Table 2.11: First 3 subcritical eigenvalues for the VVER-1000 reactor.

K_{xy}	λ_2	λ_3	λ_4
1	0.9927145	0.9927145	0.9705486
2	0.9947271	0.9947271	0.9736192
3	0.9948069	0.9948069	0.9737548
4	0.9948118	0.9948118	0.9737671
5	0.9948145	0.9948145	0.9737718
6	0.9948153	0.9948153	0.9737733

2.2.6.4 Two-dimensional VVER-440 problem

For this problem, the core is a VVER-440 core type, with 25 fuel elements across the diameter, as is shown in Figure 2.29.

The core has 7 control rods inserted and a layer of reflector at the boundary of the core. The assembly pitch is 14.7 cm. Vacuum boundary conditions are considered at the external boundary of the reflector. Nuclear cross sections for this geometry are shown in Table 2.12.

Table 2.13 shows the different results obtained for the k_{eff} of this reactor core.

Figure 2.30 shows the neutronic power distribution for each hexagon computed with the spectral element method (SEM) using a degree $K_{xy} = 6$ in the expansions,

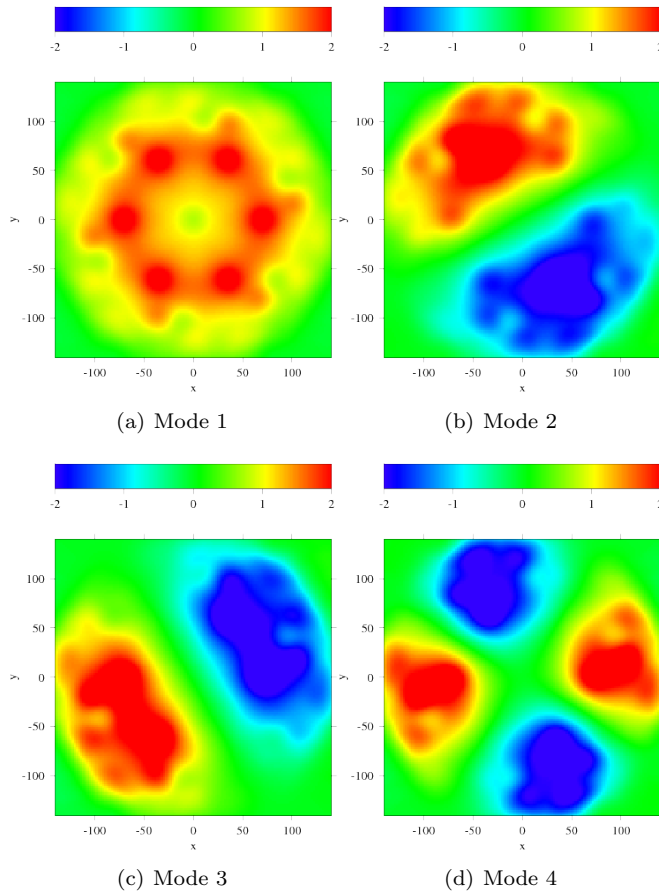


Figure 2.28: Symmetries of the four dominant modes for the VVER-1000 problem.

together with the reference result and the percentage of the relative error on each hexagon.

In Table 2.14 the results obtained for the first three subcritical eigenvalues, computed with the SEM method using different values of the degree K_{xy} are presented, where we have just consider the decimal precision obtained from the reference result [34]. This table shows that for $K_{xy} = 5$ we have a very accurate solution.

A symmetry pattern for the power distribution associated with the four dominant modes is shown in Figure 2.31.

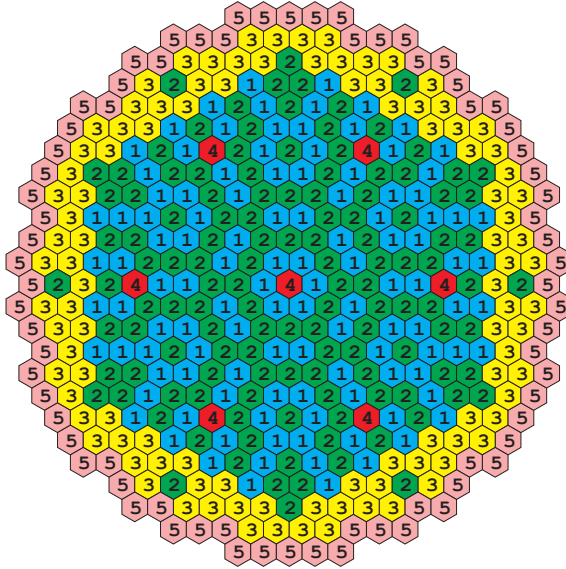


Figure 2.29: Geometry of the reactor VVER-440.

Table 2.12: Cross sections of the VVER-440 problem.

Fuel	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.3466	0.008362	0.016893	0.0044488
	2	0.37169	0.064277		0.073753
2	1	1.3377	0.008797	0.015912	0.0055337
	2	0.36918	0.079361		0.10581
3	1	1.3322	0.009462	0.014888	0.0070391
	2	0.36502	0.1001		0.14964
4	1	1.1953	0.013372	0.022264	0.0
	2	0.19313	0.13498		0.0
5	1	1.4485	0.000922	0.032262	0.0
	2	0.25176	0.032839		0.0

2.2.6.5 Two-dimensional HWR problem

This core is a very large HWR core of 35 assemblies across the core diameter, as it is shown in Figure 2.32.

The fuel assemblies are surrounded by a tritium-generating target zone outside of which is the reflector zone. There are many rodded assemblies and some vacancy assemblies. The assembly pitch is 17.78 cm. The boundary conditions considered for this core are zero flux at the outside boundary of the reflector. Table 2.15 shows the nuclear cross sections data for the HWR core.

Table 2.13: k_{eff} results for the VVER-440 problem.

K_{xy}	$\lambda_1(k_{\text{eff}})$	$\Delta k_{\text{eff}}(\text{pcm})$	$\epsilon_{\text{max}}(\%)$	$\bar{\epsilon}(\%)$	N	NNZ
1	1.01131	-161	11.02	3.86	1189	8029
2	1.01011	-41	1.43	0.60	4903	54535
3	1.00974	-4	0.21	0.06	11143	174069
4	1.00971	-1	0.22	0.08	19909	388475
5	1.00970	≈ 0	0.22	0.08	31201	673611
6	1.00970	≈ 0	0.22	0.08	45019	1022241
Ref.	1.00970					

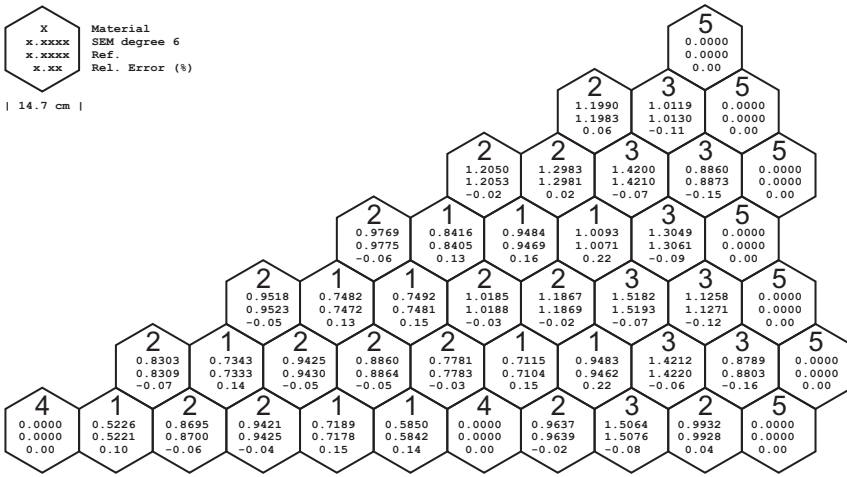


Figure 2.30: Power distribution for the VVER-440 problem.

Table 2.14: First 3 subcritical eigenvalues for the VVER-440 reactor.

K_{xy}	λ_2	λ_3	λ_4
1	1.0047100	1.0047100	0.9911925
2	1.0031413	1.0031413	0.9894401
3	1.0027047	1.0027047	0.9889541
4	1.0026585	1.0026585	0.9889027
5	1.0026534	1.0026534	0.9888969
6	1.0026526	1.0026526	0.9888960

Table 2.16 shows the different results obtained for the k_{eff} of this reactor core, and we can see that the results obtained for $K_{xy} = 3$ or $K_{xy} = 4$ are very accurate.

Figure 2.33 shows the neutronic power distribution for each hexagon computed with the Spectral Element Method (SEM) using a degree $K_{xy} = 5$ in the expan-

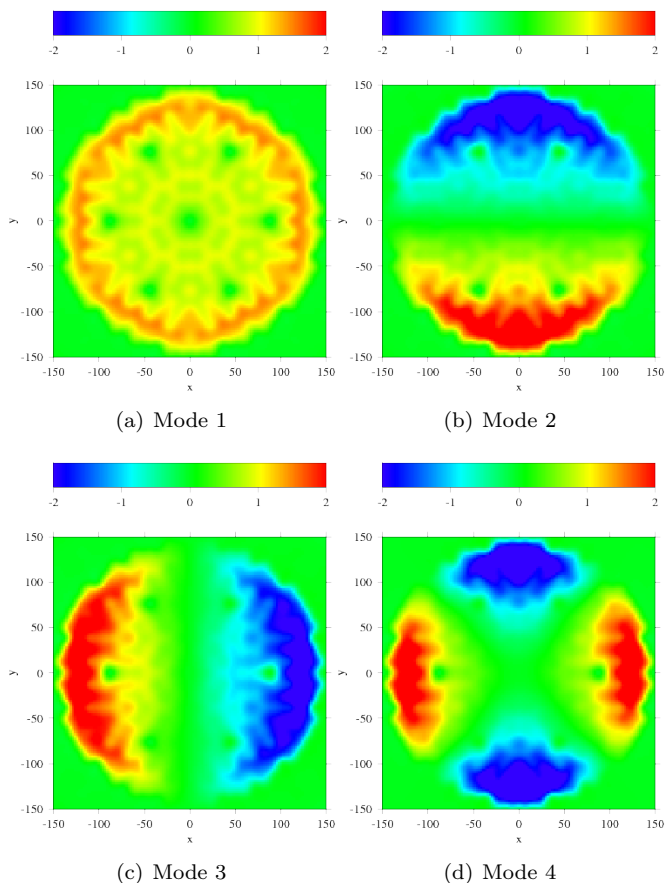


Figure 2.31: Symmetry patterns of the four dominant modes for the VVER-440 problem.

sions, together with the reference result and the percentage of relative error on each hexagon.

In Table 2.17 the results obtained for the first three subcritical eigenvalues, computed with the SEM method using different values of the degree K_{xy} , are presented.

A symmetry pattern for the power distribution associated with the four dominant modes is shown in Figure 2.34.

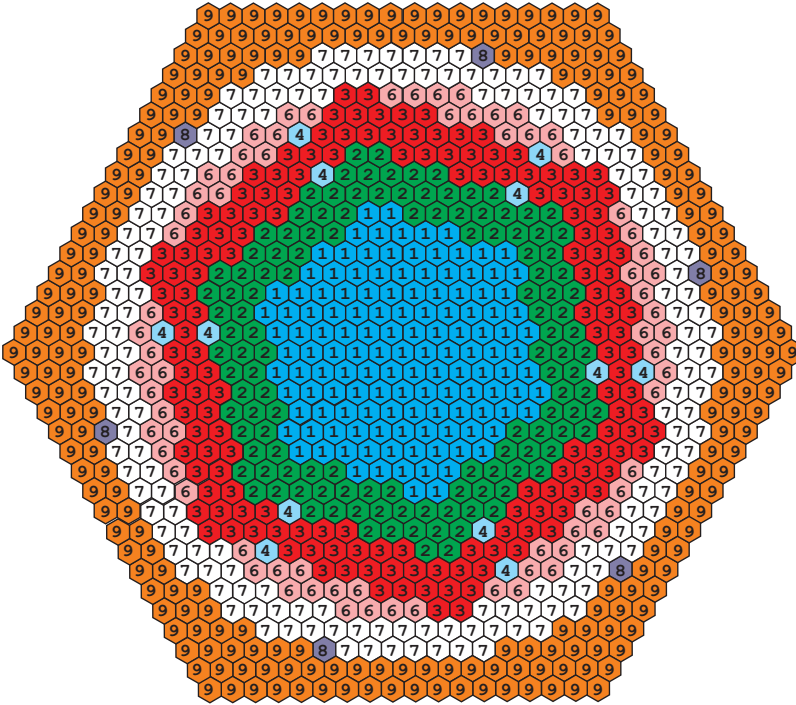


Figure 2.32: Geometry of the HWR reactor

2.2.6.6 Three-dimensional VVER 440 problem

The 3D VVER-440 problem with reflector is a 3D reactor of 250.0 cm height, with two reflector layers of 25.0 cm each added, one to the top and the other one to the bottom of the core. The core is a VVER-440 type with 25 fuel elements along the diameter, with 7 control rods half-way inserted from the top, and another reflector layer added to the radial boundary of the core, as it is shown in Figure 2.35. This core has a radial symmetry by reflection of $1/12$ and the pitch is 14.7 cm. Nevertheless, the calculations have been performed taking into account the whole core since the subcritical modes do not maintain the radial symmetry. Vacuum boundary conditions have been considered ($\beta = 0$ in equation (2.56)). The nuclear cross sections for the geometry are reported in Table 2.18, where the cross sections of the rodded nodes in the unrodded part is composed of fuel type 2.

Table 2.19 shows the different results obtained for the k_{eff} of the 3D VVER 440 problem with reflector with different values of K_{xy} and K_z used in the expansions of the Spectral Element Method (SEM). The reference result reported in [34] and the dimension of the eigenvalue problem which has to be solved, N , as well as the non-zero entries of the matrices, NNZ have been also included.

Table 2.15: Cross sections of the HWR reactor.

Fuel	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.38250058	0.0029412350	0.00816457	0.00226216
	2	0.89752185	0.022306487		0.0230623
2	1	1.38255219	0.0029508050	0.00822378	0.00222750
	2	0.89749043	0.022387609		0.0226849
3	1	1.37441741	0.0025322079	0.00808816	0.00214281
	2	0.88836771	0.016946527		0.0204887
4	1	1.31197955	0.00037645300	0.0123115	0.0
	2	0.87991376	0.00052900925		0.0
6	1	1.38138909	0.0027974400	0.00776568	0.00239469
	2	0.90367052	0.021902980		0.0266211
7	1	1.30599110	0.00063382099	0.0110975	0.0
	2	0.83725587	0.0043330365		0.0
8	1	1.29192957	0.00035711600	0.0115582	0.0
	2	0.81934103	0.00030056488		0.0
9	1	1.06509884	0.0021482210	0.0261980	0.0
	2	0.32282849	0.033348874		0.0

Table 2.16: k_{eff} results for the HWR reactor.

K_{xy}	$\lambda_1(k_{\text{eff}})$	$\Delta k_{\text{eff}}(pcm)$	$\epsilon_{\text{max}}(\%)$	$\bar{\epsilon}(\%)$	N	NNZ
1	0.9919827	-1.7688	2.85	0.26	2653	18157
2	0.9919870	-2.2040	0.13	0.02	10819	121819
3	0.9919620	0.2996	0.02	0.01	24499	382313
4	0.9919610	0.4020	0.02	0.01	43693	848783
5	0.9919610	0.3994	0.02	0.01	68401	1475979
Ref.	0.991965					

Table 2.17: First 3 subcritical eigenvalues for the HWR reactor.

K_{xy}	λ_2	λ_3	λ_4
1	0.9835954	0.9835954	0.9641395
2	0.9836219	0.9836219	0.9642709
3	0.9835938	0.9835938	0.9642393
4	0.9835926	0.9835926	0.9642380
5	0.9835926	0.9835926	0.9642380

Table 2.20 shows the results obtained for the first three subcritical eigenvalues calculated with the SEM with different values of the polynomial expansion orders K_{xy} and K_z .

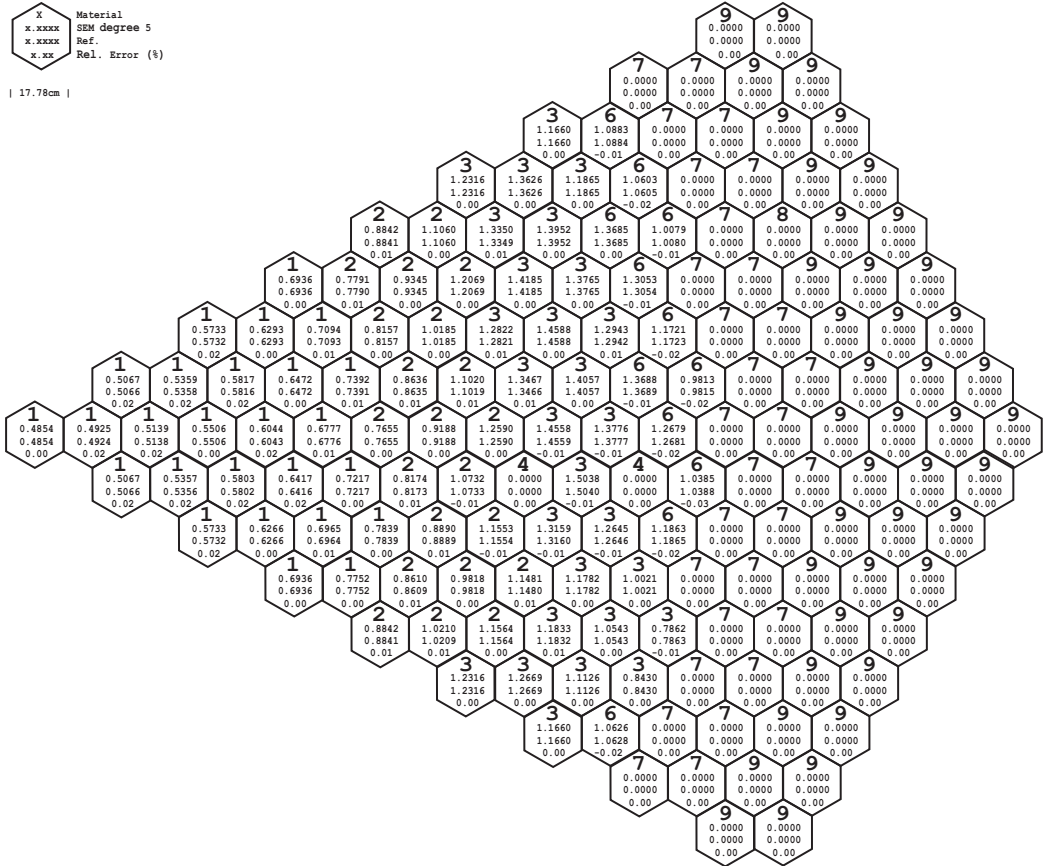


Figure 2.33: Power distribution for the HWR problem.

Figure 2.36 shows a schematic view of the 3D fundamental mode and the first three subcritical power modes calculated with SEM, showing suitable slices of the power along the axial axis Z , and along the radial plane XY , to show the different behaviour of each mode.

Local results for the normalized power distribution for the best approximation, i.e., the one obtained with $K_{XY} = 3$ and $K_Z = 4$ is summarized in Table 2.21

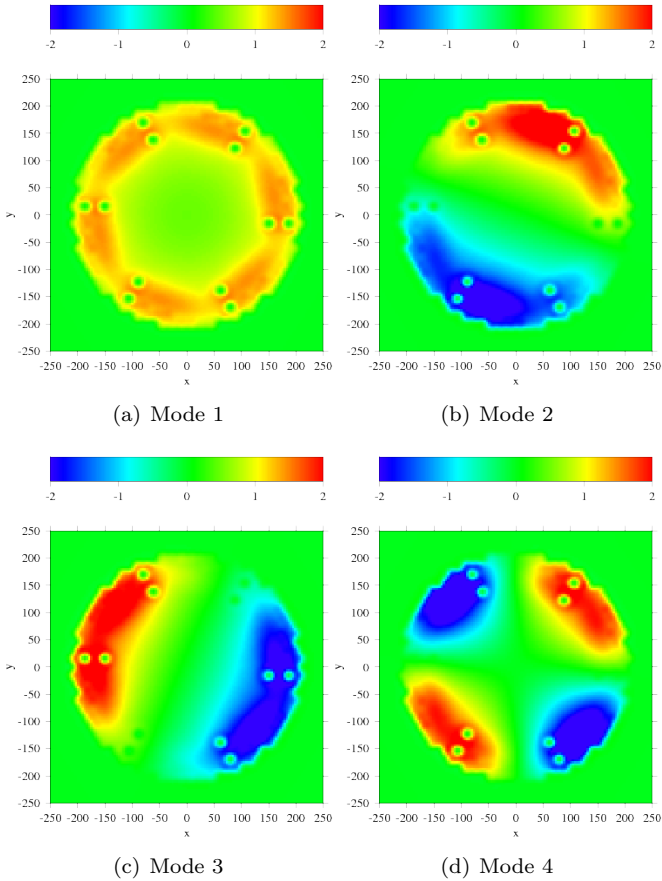


Figure 2.34: Symmetry patterns of the four dominant modes for the HWR problem.

2.2.6.7 Three-dimensional VVER 1000 problem

This 3D-dimensional problem is 200.0 cm height. The reactor core is VVER-1000 type, with 15 fuel elements along its diameter, with 25 control rods inserted, of which 19 are fully inserted and the 6 closer to the center are half-way inserted from the top, as it is shown in Figure 2.38. This reactor core has 1/6 cyclic geometry, the pitch assembly is 23.60 cm. Both radial and axial albedo boundary conditions have been considered with values $\beta = 0.6$ and $\beta = 0.5384615$, respectively, in the equation (2.56)). The nuclear cross sections for the chosen geometry are reported in Table 2.22, where in the mixed assemblies noted as 4/3 the assemblies are rodded with material 4 from the top until the half of the reactor, and the unrodded assembly is composed of material 3.

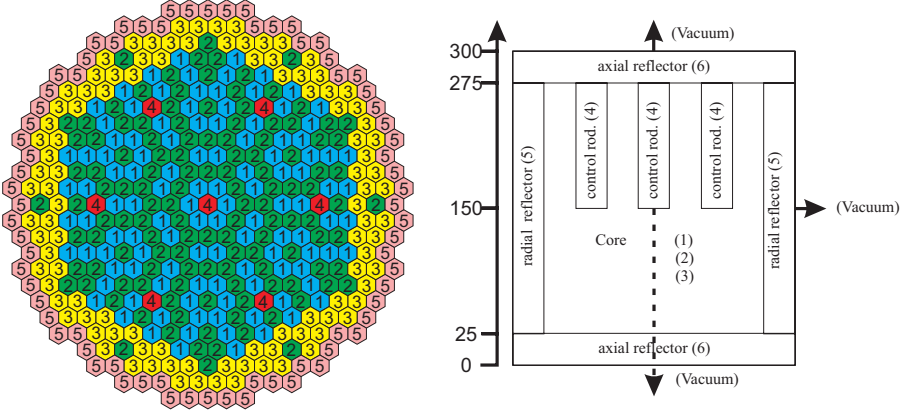


Figure 2.35: Geometry of the 3D VVER-440 problem with reflector

Table 2.18: Cross sections of the three-dimensional VVER-440 problem.

Fuel	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.3466	0.008362	0.016893	0.0044488
	2	0.37169	0.064277		0.073753
2	1	1.3377	0.008797	0.015912	0.0055337
	2	0.36918	0.079361		0.10581
3	1	1.3322	0.009462	0.014888	0.0070391
	2	0.36502	0.1001		0.14964
4	1	1.1953	0.013372	0.022264	0.0
	2	0.19313	0.13498		0.0
5	1	1.4485	0.000922	0.032262	0.0
	2	0.25176	0.032839		0.0
6	1	1.3413	0.002153	0.027148	0.0
	2	0.24871	0.064655		0.0

Table 2.23 shows the different results obtained for the k_{eff} for this problem calculated with the SEM method for different values of the expansion order K_{xy} and K_z . The reference result reported in [34] and the eigenvalue problem dimension, N , as well as the number of non-zero entries for these matrices, NNZ are also included.

Table 2.24 shows the results obtained for the first three subcritical eigenvalues calculated with the SEM method for different values of the expansion order K_{xy} and K_z .

As it has been done for the previous problem, in Figure 2.39 schematic views of the 3D fundamental mode and the first three subcritical modes calculated with SEM are shown.

Table 2.19: Results for the k_{eff} for the 3D VVER 440 problem

K_{xy}	K_z	$\lambda_1(k_{\text{eff}})$	$\Delta k_{\text{eff}}(pcm)$	$\epsilon_{\text{max}}(\%)$	$\bar{\epsilon}(\%)$	N	NNZ
1	1	1.01271	138.79	12.92	2.96	17407	335479
2	1	1.01117	-14.90	9.73	1.22	67639	2171863
3	1	1.01085	-47.45	8.83	1.06	150709	6722345
4	1	1.01081	-50.98	8.73	1.05	266617	14540297
1	2	1.01322	190.03	13.49	2.81	33475	879499
2	2	1.01168	36.34	3.80	0.63	130075	5693803
3	2	1.01136	3.98	2.69	0.36	289825	17623445
4	2	1.01132	0.49	2.65	0.36	512725	38119157
1	3	1.01323	191.20	12.61	2.75	49543	1423519
2	3	1.01170	37.71	3.00	0.53	192511	9215743
3	3	1.01137	5.40	1.70	0.23	428941	28524545
4	3	1.01134	1.91	1.66	0.24	758833	61698017
1	4	1.01322	189.65	12.56	2.72	65611	1749931
2	4	1.01168	36.23	2.19	0.47	254947	11328907
3	4	1.01136	3.94	0.69	0.11	568057	35065205
Ref.		1.01132	-	-	-	-	-

Table 2.20: 3 first subcritical eigenvalues for the 3D VVER 440 problem

K_{xy}	K_z	λ_2	λ_3	λ_4
1	1	1.00461	1.00461	0.98950
2	1	1.00277	1.00277	0.98756
3	1	1.00237	1.00237	0.98710
4	1	1.00232	1.00232	0.98705
1	2	1.00507	1.00507	0.98991
2	2	1.00323	1.00323	0.98796
3	2	1.00283	1.00283	0.98750
4	2	1.00278	1.00278	0.98746
1	3	1.00508	1.00508	0.98991
2	3	1.00324	1.00324	0.98797
3	3	1.00284	1.00284	0.98751
4	3	1.00279	1.00279	0.98746
1	4	1.00506	1.00506	0.98989
2	4	1.00322	1.00322	0.98795
3	4	1.00282	1.00282	0.98749

Local results for the normalized power distribution for the best approximation, i.e., the one obtained with $K_{XY} = 3$ and $K_Z = 3$ is summarized in Table 2.25

Summarizing, a high order Spectral Element Method (SEM) has been implemented to approximate the solution of Lambda Modes problem of a nuclear reactor with

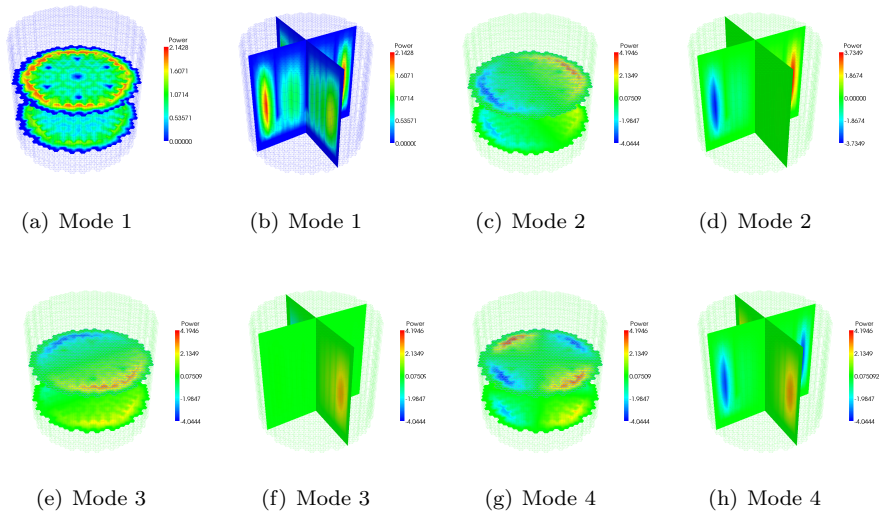


Figure 2.36: Modes for the VVER-440 problem

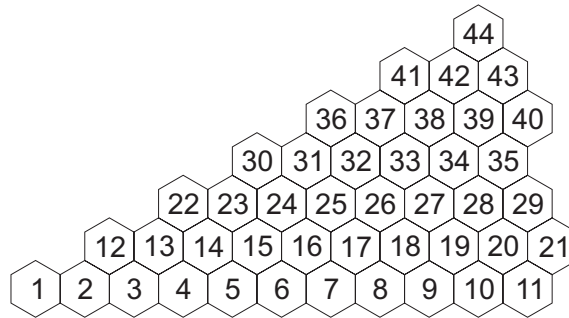


Figure 2.37: Numbers identifying the position of the cells of the VVER 440 reactor for the local power distribution

hexagonal geometry. This method is based on splitting each one of the hexagonal prisms defined by the geometry into six equilateral triangular prisms, obtaining a mesh which is kept fixed. The SEM method is based on the expansion of the neutron flux in terms of a polynomials basis constructed using the modified Dubiner's Polynomials. The accuracy of the method is improved increasing the order in the polynomial expansions and it is not necessary to refine the mesh.

To test the performance of the method, an extensive analysis of 2D benchmark reactor cores has been carried out. In all the studied reactors the SEM method has provided good results for the k_{eff} and the neutron power distribution and

Table 2.21: Local power for the 3D VVER 440 problem

Pos.	Axial zones									
	2	3	4	5	6	7	8	9	10	11
1	0.5210	1.0748	1.4548	1.5840	1.3407	0	0	0	0	0
2	0.4204	0.8685	1.1761	1.2837	1.1413	0.7926	0.5662	0.3978	0.2485	0.1101
3	0.5508	1.1366	1.5402	1.6884	1.5610	1.2494	0.9336	0.6589	0.4120	0.1825
4	0.5639	1.1641	1.5785	1.7337	1.6187	1.3249	1.0012	0.7085	0.4436	0.1966
5	0.4502	0.9312	1.2633	1.3857	1.2775	1.0149	0.7562	0.5348	0.3354	0.1487
6	0.4629	0.9580	1.3005	1.4227	1.2586	0.8561	0.6076	0.4292	0.2698	0.1200
7	0.6151	1.2722	1.7293	1.8928	1.6072	0	0	0	0	0
8	0.6493	1.3443	1.8309	2.0164	1.8154	1.2882	0.9440	0.6784	0.4305	0.1925
9	0.7844	1.6226	2.2135	2.4542	2.3092	1.8946	1.4573	1.0544	0.6704	0.2999
10	0.4763	0.9873	1.3482	1.5007	1.4357	1.2181	0.9539	0.6931	0.4411	0.1970
11	0	0	0	0	0	0	0	0	0	0
12	0.5458	1.1263	1.5259	1.6710	1.5343	1.2058	0.8940	0.6302	0.3940	0.1745
13	0.4403	0.9101	1.2339	1.3551	1.2653	1.0355	0.7826	0.5537	0.3465	0.1533
14	0.5613	1.1590	1.5722	1.7274	1.6115	1.3159	0.9938	0.7041	0.4413	0.1958
15	0.5686	1.1748	1.5948	1.7492	1.6013	1.2464	0.9221	0.6531	0.4103	0.1824
16	0.5910	1.2218	1.6604	1.8197	1.6160	1.1075	0.7924	0.5628	0.3550	0.1584
17	0.4934	1.0226	1.3920	1.5313	1.3734	0.9649	0.7027	0.5038	0.3193	0.1425
18	0.5118	1.0617	1.4480	1.6032	1.4971	1.2055	0.9199	0.6647	0.4224	0.1885
19	0.6918	1.4311	1.9540	2.1738	2.0738	1.7498	1.3664	0.9923	0.6316	0.2826
20	0.4109	0.8511	1.1628	1.2967	1.2476	1.0686	0.8422	0.6132	0.3905	0.1746
21	0	0	0	0	0	0	0	0	0	0
22	0.5553	1.1466	1.5555	1.7105	1.6031	1.3215	1.0035	0.7117	0.4461	0.1978
23	0.4345	0.8989	1.2205	1.3426	1.2543	1.0259	0.7769	0.5521	0.3469	0.1539
24	0.4413	0.9136	1.2419	1.3668	1.2665	1.0133	0.7619	0.5433	0.3426	0.1524
25	0.5860	1.2125	1.6510	1.8219	1.6876	1.3392	1.0102	0.7253	0.4594	0.2050
26	0.6176	1.2793	1.7452	1.9354	1.8224	1.4968	1.1524	0.8340	0.5301	0.2368
27	0.7273	1.5048	2.0555	2.2897	2.1914	1.8581	1.4568	1.0600	0.6750	0.3021
28	0.5184	1.0735	1.4674	1.6386	1.5808	1.3594	1.0751	0.7844	0.4999	0.2237
29	0	0	0	0	0	0	0	0	0	0
30	0.5421	1.1204	1.5234	1.6804	1.5776	1.3003	0.9922	0.7092	0.4473	0.1991
31	0.4392	0.9100	1.2397	1.3728	1.2979	1.0805	0.8333	0.6003	0.3803	0.1694
32	0.4625	0.9594	1.3096	1.4570	1.3912	1.1758	0.9189	0.6671	0.4242	0.1892
33	0.4641	0.9634	1.3171	1.4711	1.4190	1.2190	0.9642	0.7039	0.4487	0.2003
34	0.5825	1.2064	1.6506	1.8479	1.7935	1.5558	1.2392	0.9073	0.5790	0.2593
35	0	0	0	0	0	0	0	0	0	0
36	0.5728	1.1864	1.6201	1.8053	1.7345	1.4823	1.1661	0.8480	0.5394	0.2409
37	0.5804	1.2035	1.6465	1.8431	1.7891	1.5526	1.2367	0.9051	0.5775	0.2581
38	0.6155	1.2745	1.7454	1.9590	1.9124	1.6731	1.3415	0.9853	0.6296	0.2821
39	0.3786	0.7847	1.0751	1.2081	1.1823	1.0381	0.8348	0.6141	0.3926	0.1758
40	0	0	0	0	0	0	0	0	0	0
41	0.5128	1.0638	1.4574	1.6373	1.6020	1.4061	1.1303	0.8312	0.5313	0.2377
42	0.4251	0.8813	1.2081	1.3596	1.3348	1.1772	0.9500	0.7002	0.4480	0.2007
43	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0

also has succeeded computing the subcritical modes of each reactor. Also, two 3D benchmarks have been studied, the VVER-1000 and the VVER-440. The obtained results for the k_{eff} and the neutronic power distribution have been compared with the reference solution obtained with the DIF3D code. The SEM method have provided good results for the k_{eff} and the power distributions, calculating also successfully a set of subcritical modes for these reactors.

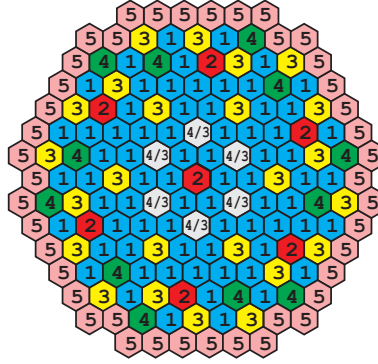

Figure 2.38: Geometry of the 3D VVER-1000 problem

Table 2.22: Cross sections of the 3D VVER-1000 problem

Fuel	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.38320	0.0083859	0.0164977	0.00481619
	2	0.386277	0.0673049		0.0846154
2	1	1.38299	0.0115490	0.0147315	0.00466953
	2	0.389403	0.0810328		0.0852264
3	1	1.39522	0.0089441	0.0156219	0.00604889
	2	0.386225	0.0844801		0.1194280
4	1	1.39446	0.0119932	0.0140185	0.00591507
	2	0.387723	0.0989670		0.1204970
5	1	1.39506	0.0091160	0.0154981	0.00640256
	2	0.384492	0.0893878		0.1292810

Table 2.23: Results for the k_{eff} for the 3D VVER-1000 problem

K_{xy}	K_z	$\lambda_1(k_{\text{eff}})$	$\Delta k_{\text{eff}}(pcm)$	$\epsilon_{\text{max}}(\%)$	$\bar{\epsilon}(\%)$	N	NNZ
1	1	1.009884	-147	12.70	3.69	4815	88975
2	1	1.010812	-54	4.30	1.15	18423	571375
3	1	1.010806	-54	3.39	1.19	40833	1764325
1	2	1.010310	-104	11.28	4.07	9095	231335
2	2	1.011229	-12	1.78	0.40	34799	1485575
3	2	1.011222	-13	0.87	0.10	77129	4587245
1	3	1.010341	-101	11.36	4.09	13375	373695
2	3	1.011261	-9	1.57	0.43	51175	2399775
3	3	1.011254	-10	0.66	0.10	113425	7410165
Ref.		1.011350	-	-	-	-	-

Table 2.24: 3 first subcritical eigenvalues for the 3D VVER-1000 problem

K_{xy}	K_z	λ_2	λ_3	λ_4
1	1	0.995022	0.995022	0.976204
2	1	0.996765	0.996765	0.978993
3	1	0.996822	0.996822	0.979109
1	2	0.995250	0.995250	0.977463
2	2	0.996989	0.996989	0.979293
3	2	0.997045	0.997045	0.979338
1	3	0.995271	0.995271	0.977514
2	3	0.997009	0.997009	0.979342
3	3	0.997065	0.997065	0.979387

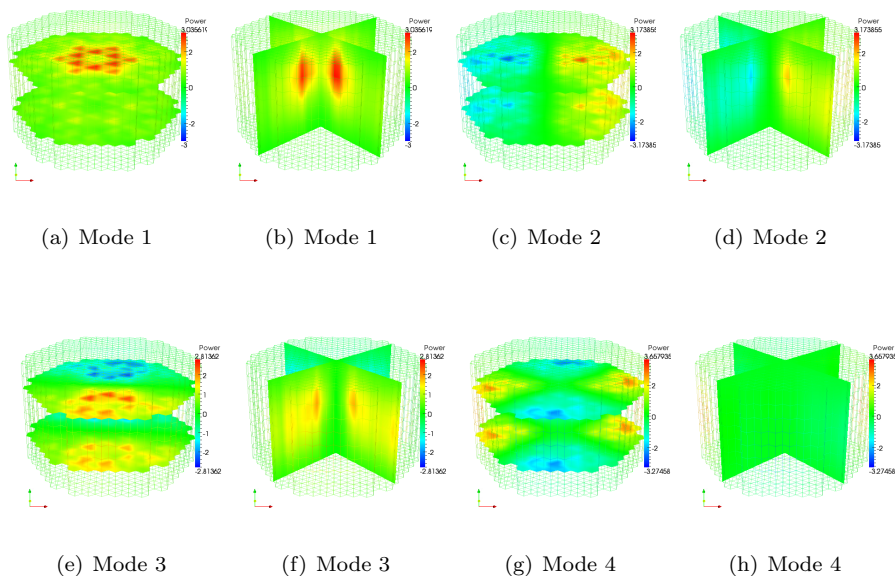


Figure 2.39: Modes for the VVER-1000 problem

Table 2.25: Local power for the 3D VVER 1000 problem

Pos.	Axial zones								Bott.	Top
	1	2	3	4	5	6	7	8		
1	0.1780	0.4119	0.6941	1.0827	1.5074	1.6393	1.3331	0.6715	0.5917	1.2878
2	0.2576	0.5928	0.9888	1.5585	2.3015	2.5239	2.0467	1.0277	0.8494	1.9749
3	0.3406	0.7714	1.2395	1.8333	2.5438	2.7091	2.1717	1.0856	1.0462	2.1275
4	0.5155	1.1483	1.7752	2.3888	2.8504	2.8476	2.2368	1.1118	1.4569	2.2617
5	0.3656	0.8038	1.2048	1.5347	1.7138	1.6383	1.2601	0.6202	0.9772	1.3081
6	0.3003	0.6512	0.9525	1.1705	1.2564	1.1640	0.8785	0.4290	0.7686	0.9320
7	0.2753	0.5913	0.8548	1.0340	1.0909	0.9959	0.7444	0.3628	0.6888	0.7985
8	0.2969	0.6739	1.0959	1.7203	3.0768	3.3861	2.7268	1.3675	0.9467	2.6393
9	0.3713	0.8329	1.3057	1.8245	2.3311	2.3978	1.9004	0.9459	1.0836	1.8938
10	0.3641	0.8055	1.2222	1.5897	1.8197	1.7702	1.3738	0.6785	0.9954	1.4105
11	0.3809	0.8292	1.2245	1.5252	1.6615	1.5579	1.1848	0.5814	0.9899	1.2464
12	0.2237	0.4815	0.6996	0.8520	0.9054	0.8318	0.6244	0.3053	0.5642	0.6667
13	0.1392	0.2983	0.4304	0.5193	0.5463	0.4975	0.3713	0.1810	0.3468	0.3990
14	0.3720	0.8346	1.3082	1.8276	2.3345	2.4010	1.9029	0.9471	1.0856	1.8964
15	0.3393	0.7529	1.1487	1.5113	1.7567	1.7255	1.3444	0.6646	0.9381	1.3728
16	0.2193	0.4783	0.7091	0.8893	0.9766	0.9215	0.7032	0.3456	0.5740	0.7367
17	0.2518	0.5444	0.7923	0.9670	1.0303	0.9485	0.7129	0.3475	0.6389	0.7598
18	0.1894	0.4066	0.5876	0.7103	0.7488	0.6831	0.5103	0.2488	0.4735	0.5478
19	0.3629	0.8028	1.2183	1.5847	1.8144	1.7653	1.3700	0.6765	0.9922	1.4065
20	0.3005	0.6564	0.9722	1.2174	1.3347	1.2578	0.9591	0.4701	0.7866	1.0054
21	0.3549	0.7661	1.1141	1.3588	1.4466	1.3308	0.9997	0.4877	0.8985	1.0662
22	0.2405	0.5168	0.7469	0.9031	0.9524	0.8691	0.6493	0.3163	0.6018	0.6968
23	0.2985	0.6482	0.9551	1.1865	1.2892	1.2061	0.9159	0.4498	0.7721	0.9652
24	0.2971	0.6419	0.9321	1.1344	1.2051	1.1066	0.8302	0.4041	0.7514	0.8865
25	0.2635	0.5659	0.8174	0.9874	1.0402	0.9484	0.7081	0.3448	0.6585	0.7604
26	0.3584	0.7712	1.1168	1.3540	1.4321	1.3103	0.9806	0.4779	0.9001	1.0502
27	0.2576	0.5528	0.7972	0.9610	1.0103	0.9195	0.6856	0.3337	0.6421	0.7373
28	0.1962	0.4205	0.6058	0.7294	0.7658	0.6961	0.5187	0.2525	0.4880	0.5583

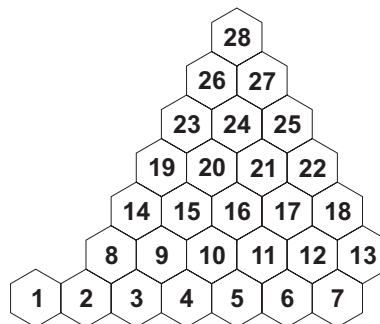


Figure 2.40: Numbers identifying the position of the cells of the VVER 1000 reactor for the local power distribution

Chapter 3

Time Dependent Neutron Diffusion Equation

Contents

3.1 Backward Method	82
3.1.1 Time discretization	83
3.1.2 Rod cusping	85
3.1.3 Numerical results	87
3.2 Block Preconditioning	94
3.2.1 Solvers for the linear systems	96
3.2.2 Numerical Results	99
3.3 Spectral Preconditioning	104
3.3.1 Spectral Preconditioner	105
3.3.2 Numerical Results	106
3.4 Exponential Method	111
3.4.1 Exponential operator and its Magnus Expansion	112
3.4.2 Implementation of the product matrix exponential - vector. . .	120
3.4.3 Numerical Results	125

An implicit difference method for the time discretization of the time dependent neutron diffusion equation is presented and the performance of the code is tested solving the first exercise of the AER transient benchmark. The obtained results are compared with the reference results of the benchmark and with the results provided by PARCS code.

In nodal methods the nuclear cross sections are defined constant on each node. This causes a problem in transients consisting of moving control rods. The easiest way to interpolate the cross sections of a partially rodded node is by means of the portion of the rod inserted on the node. This method is called volume weighted method. With the cross sections calculated by means of the volume weighted

method, an unphysical behaviour of the k_{eff} appears on the calculation, and this is called rod cusping effect [50]. This effect is unphysical because it does not correspond to the physical behavior of the k -effective for a transient defined by moving the control rods with constant velocity. The correction of this effect by means of a flux weighted method is studied here. The methods are tested in a small reactor where the rod cusping is a dominant effect.

3.1 Backward Method

Computational codes for transient analysis of nuclear reactors are very important to understand the consequences of different events on nuclear safety. In this way, it is interesting to develop efficient methods to simulate nuclear reactors. In particular, we will consider here VVER-type reactors which use the neutron diffusion equation discretized on hexagonal meshes.

We consider the time dependent neutron diffusion equation in the approximation of two energy groups. This model is of the form [8]

$$\begin{aligned} [v^{-1}] \frac{\partial \Phi}{\partial t} + \mathcal{L}\Phi &= (1 - \beta)\mathcal{M}\Phi + \sum_{k=1}^K \lambda_k \chi \mathcal{C}_k, \\ \frac{\partial \mathcal{C}_k}{\partial t} &= \beta_k [\nu \Sigma_{f1} \nu \Sigma_{f2}] \Phi - \lambda_k \mathcal{C}_k, \quad k = 1, \dots, K, \end{aligned} \quad (3.1)$$

where, K is the number of delayed neutron precursors groups considered,

$$\mathcal{L} = \begin{bmatrix} -\vec{\nabla} \cdot (D_1 \vec{\nabla}) + \Sigma_{a1} + \Sigma_{12} & 0 \\ -\Sigma_{12} & -\vec{\nabla} \cdot (D_2 \vec{\nabla}) + \Sigma_{a2} \end{bmatrix}, \quad [v^{-1}] = \begin{bmatrix} \frac{1}{v_1} & 0 \\ 0 & \frac{1}{v_2} \end{bmatrix},$$

and

$$\mathcal{M} = \begin{bmatrix} \nu \Sigma_{f1} & \nu \Sigma_{f2} \\ 0 & 0 \end{bmatrix}, \quad \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}, \quad \chi = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The diffusion constants and cross-sections, D_g , Σ_{12} , Σ_{ag} , $\nu \Sigma_{fg}$, $g = 1, 2$, appearing in the equations depend on the reactor materials, that is, they are position and time dependent functions. Associated with this problem, there is the Lambda modes problem

$$\mathcal{L}\Phi_i = \frac{1}{k_i} \mathcal{M}\Phi_i. \quad (3.2)$$

To solve both problems (3.1) and (3.2), a spatial discretization of the equations has to be selected. Different neutronic codes have been developed during the last few years that can deal with reactors with hexagonal geometry such as [51] BIPR8, DYN3D, KIKO-3D, PARCS, etc. Most of these codes make use of balance equations for the neutron population in a node of the discretization and the inter-node

coupling equations are typically obtained by integration of the neutron diffusion equation over directions transverse to each coordinate axis, ending with a system of nonlinear equations, which is solved iteratively. In this work, a Spectral Element Method based on a triangular mesh, as explained at Chapter 2, has been used [52]. With this method the initial problem is reduced to a system of ordinary differential equations.

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

After the spatial discretization by means of the Spectral Element Method, the semidiscrete two energy groups time dependent neutron diffusion equation is of the form

$$\begin{aligned} [v^{-1}] \dot{\psi} + L\psi &= (1 - \beta)M\psi + \sum_{k=1}^K \lambda_k X C_k, \\ X \dot{C}_k &= \beta_k M\psi - \lambda_k X C_k, \end{aligned} \quad (3.3)$$

where matrices L , M and X have the following block structure

$$[v^{-1}] = \begin{bmatrix} v_1^{-1}P & 0 \\ 0 & v_2^{-1}P \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & 0 \\ -L_{21} & L_{22} \end{bmatrix},$$

and

$$M = \begin{bmatrix} M_{11} & M_{12} \\ 0 & 0 \end{bmatrix}, \quad X = \begin{bmatrix} P \\ 0 \end{bmatrix},$$

and the matrix P is the mass matrix of the spatial discretization, that appears due to the fact that the polynomial basis used in the spatial discretization is not orthogonal [36].

3.1.1 Time discretization

For the time discretization of the semidiscrete time dependent neutron diffusion equation (3.3) a one-step implicit finite differences method is considered. This method consists of integrating the above ordinary differential equations (3.3) over a series of time intervals $[t_n, t_{n+1}]$. To integrate these equations from t_n to t_{n+1} , it is supposed that the term $M\psi$ varies linearly between these instants. Hence, approximating

$$X \dot{C}_k = -\lambda_k X C_k + \beta_k M^n \psi^n + \frac{\beta_k}{\Delta t} (t - t_n) (M^{n+1} \psi^{n+1} - M^n \psi^n), \quad (3.4)$$

where $\Delta t = t_{n+1} - t_n$, and $M^n \psi^n$ is $M\psi$ evaluated in the time t_n .

Integrating (3.4), the solution XC_k in t_{n+1} can be expressed as

$$XC_k^{n+1} = XC_k^n e^{-\lambda_k \Delta t} + \beta_k (a_k M^n \psi^n + b_k M^{n+1} \psi^{n+1}) , \quad (3.5)$$

where the coefficients a_k and b_k are given by [53]

$$a_k = \frac{(1 + \lambda_k \Delta t)(1 - e^{-\lambda_k \Delta t})}{\lambda_k^2 \Delta t} - \frac{1}{\lambda_k} , \quad b_k = \frac{\lambda_k \Delta t - 1 + e^{-\lambda_k \Delta t}}{\lambda_k^2 \Delta t} .$$

Discretizing equation (3.1) by means of a one-step implicit method [53], it is obtained

$$[v^{-1}] \frac{1}{\Delta t} (\psi^{n+1} - \psi^n) + L^{n+1} \psi^{n+1} = (1 - \beta) M^{n+1} \psi^{n+1} + \sum_{k=1}^K \lambda_k XC_k^{n+1} . \quad (3.6)$$

Taking into account equation (3.5), equation (3.6) is rewritten as the system of linear equations

$$[T^{n+1}] \psi^{n+1} = [R^n] \psi^n + \sum_{k=1}^K \lambda_k e^{-\lambda_k \Delta t} X [C_k^n] , \quad (3.7)$$

where the matrices are defined as

$$\begin{aligned} [T^{n+1}] &= \frac{1}{\Delta t} [v^{-1}] + L^{n+1} - \hat{b} M^{n+1} , \\ [R^n] &= \frac{1}{\Delta t} [v^{-1}] + \hat{a} M^n , \end{aligned}$$

and the coefficients \hat{a} and \hat{b} are

$$\hat{a} = \sum_{k=1}^K \lambda_k \beta_k a_k , \quad \hat{b} = 1 - \beta + \sum_{k=1}^K \lambda_k \beta_k b_k .$$

This system of equations is sparse and of large dimension and it has to be solved for each time step. Initially, the preconditioned BICGSTAB method has been chosen to solve these systems [48]. The preconditioner used has been the incomplete LU preconditioner $\text{ilu}(t)$, where the tolerance t is set to a value that provides a reasonable fill-in of the matrix [48]. Also, a particular reordering of the rows of the matrix is used to minimize this fill-in.

3.1.2 Rod cusping

In nodal methods the nuclear cross sections are defined constants on each node. This causes a problem in transients consisting on moving control rods. When a control rod is partially inserted in a node, this node is divided in two: the upper part of the node, where the cross sections are modified due to the effect of the control rod, and the lower part of the node, which has the cross sections without modifications. Then the cross sections of the whole node is calculated by means of an interpolation procedure taking into account the position of the control rod tip.

The easiest way to interpolate the cross sections of a partially rodded node is by means of the portion of the rod inserted in the node [50]. This method is called volume weighted method. With the cross sections calculated by means of the volume weighted method, a unphysical behaviour of the k_{eff} appears on the calculation, and this is called rod cusping effect (see Figure 3.1). This unphysical behaviour is also translated to the neutronic flux distribution.

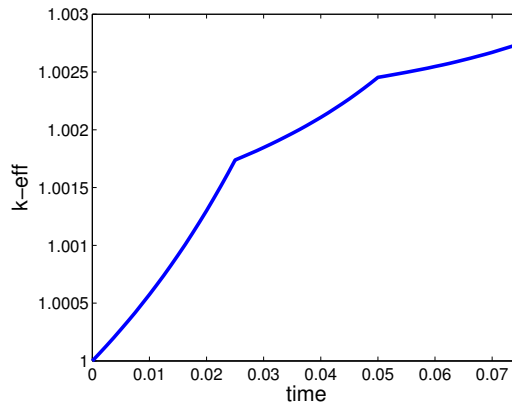


Figure 3.1: Rod cusping effect.

To avoid the rod cusping effect on nodal methods, different strategies have been developed [50]. As, for example, a flux weighting method [54], moving meshes method [55], an equivalence method [54], and a three nodes homogenization method [56]. These methods have to solve a small one-dimensional eigenvalue problem for each one of the partially rodded nodes. Then, different schemes are applied to obtain the new cross sections of the partially rodded node from the old cross sections of the two parts of the node (the rodded and the unrodded) and the heterogeneous flux for the small isolated problem with suitable boundary conditions. These solutions can be improved by means of the use of assembly discontinuity factors for the interface of the node with the ones on a neighbourhood. Here, a weighted method for the cross sections interpolation of the partially rod-

ded node is proposed. This method is well adapted for a high order finite element method based on a primal variational formulation.

When a node is partially rodded (See Figure 3.2), a homogenized cross section is necessary to apply nodal methods. In this case, the heterogeneous cross section

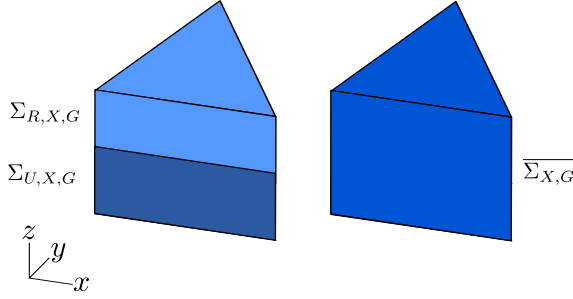


Figure 3.2: Partially Rodded Node.

$\Sigma_{X,G}^{n+1}$ for the partially rodded node for group G at time t_{n+1} is defined by

$$\Sigma_{X,G}^{n+1}(z) = \begin{cases} \Sigma_{R,X,G}^{n+1} & \text{if } 0 \leq z \leq f_{ins} \\ \Sigma_{U,X,G}^{n+1} & \text{if } f_{ins} < z \leq 1 \end{cases}, \quad (3.8)$$

where $\Sigma_{R,X,G}^{n+1}$ and $\Sigma_{U,X,G}^{n+1}$ are, respectively, the cross sections for the rodded and unrodded part of the node at time t_{n+1} , and f_{ins} is the fraction of the rod inserted in the node.

The flux for a group G in a node Ω_e at time t_n is given by

$$\Phi_G^n(\vec{r}) = \sum_{i,j} \sum_k \phi_{ijk,G}^n g_{ij}(x,y) h_k(z). \quad (3.9)$$

Due to the fact that the control rod is inserted in the axial direction, the flux we need can be collapsed on the others directions, obtaining the collapsed flux

$$\hat{\Phi}_G^n(z) = \iint_{\Omega_e} \Phi_G^n(\vec{r}) dx dy = \sum_k \hat{\phi}_{k,G}^n h_k(z) \quad (3.10)$$

where

$$\hat{\phi}_{k,G}^n = \iint_{\Omega_e} \sum_{i,j} \phi_{ijk,G}^n g_{ij}(x,y) dx dy. \quad (3.11)$$

Thus, our new interpolated cross sections at time t_{n+1} are defined by

$$\overline{\Sigma_{X,G}^{n+1}} = \frac{\int_0^1 \hat{\Phi}_G^n(z) \Sigma_{X,G}^{n+1}(z) dz}{\int_0^1 \hat{\Phi}_G^n(z) dz} = \frac{\int_0^{f_{ins}} \hat{\Phi}_G^n(z) \Sigma_{R,X,G}^{n+1} dz + \int_{f_{ins}}^1 \hat{\Phi}_G^n(z) \Sigma_{U,X,G}^{n+1} dz}{\int_0^1 \hat{\Phi}_G^n(z) dz}. \quad (3.12)$$

When calculating the steady state, the spatial mesh can be forced to coincide with the control rods peaks, thus the only problem to deal with for this calculation is when these rods are moved.

3.1.3 Numerical results

To test the performance of the method to integrate the time dependent neutron diffusion equation exposed above the 3-D transient benchmark AER-DYN-001 proposed in [57] has been studied. The problem corresponds to an asymmetric control rod ejection accident in a VVER-440 core.

A plane of this reactor showing the disposition of materials together with the initial position of control rods is shown in Figure 3.3. The hexagonal lattice pitch is 14.7 cm.

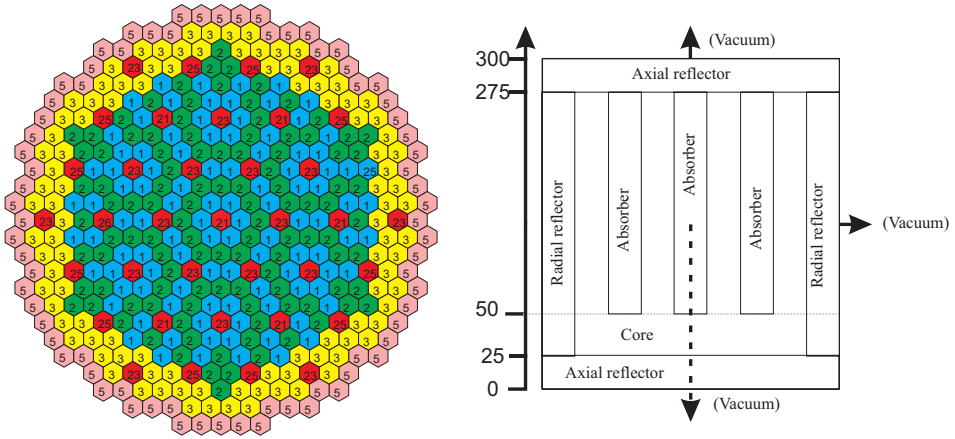


Figure 3.3: Geometry of VVER 440 reactor core.

Type numbers 1,2,3 represent fuel assemblies of different enrichment (1.6 %, 2.4 %, 3.6 %). Type numbers “21, 23, 25, 26” represent the positions where either absorber assemblies (type “4”) or their fuel followers (“1,2,3”) has to be placed into, due to the axial position of the given absorber assembly. Type “26” is used for the ejected rod. Type number “5” represents reflector cells. The albedo boundary

condition is applied on the outer edge of the reflector nodes, the extrapolation length is $2.13 * D_g$ in both groups, where D_g is the diffusion coefficient given for type "5".

Figure 3.3 also shows the axial arrangement of the problem. For the sake of comparable output, the minimum number of axial nodes has to be 12. The 1st node for the bottom reflector, 2nd-11th nodes for the active length of the core (250 cm), the 12th node for the top reflector. In this case, the height of the nodes is 25 cm. The initial position of control rod groups 21 and 26 is at 50 cm above the bottom reflector. (Follower in 2nd-3rd axial nodes, absorber rod in 4th-11th. axial nodes). Control assembly groups type "23 and 25" are out of the core at the beginning of the process. The cross sections for this reactor are summarized in Table 3.1, and the kinetic parameters for the six groups of neutron precursors and for the velocities are shown in table 3.2.

Table 3.1: Cross sections of the three-dimensional VVER-440 transient.

Fuel	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.3466	0.008362	0.016893	0.0044339
	2	0.37169	0.064277		0.073503
2	1	1.3377	0.008797	0.015912	0.0055150
	2	0.36918	0.079361		0.105450
3	1	1.3322	0.009462	0.014888	0.0070120
	2	0.36502	0.1001		0.149080
4	1	1.1953	0.013372	0.022264	0.0
	2	0.19313	0.13498		0.0
5	1	1.4485	0.000922	0.032262	0.0
	2	0.25176	0.032839		0.0
6	1	1.3413	0.002153	0.027148	0.0
	2	0.24871	0.064655		0.0

Table 3.2: Neutron precursors parameters for the VVER 440 transient.

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
β_i	0.000247	0.0013845	0.001222	0.0026455	0.000832	0.000169
$\lambda_i (s^{-1})$	0.0127	0.0317	0.115	0.311	1.4	3.87
	$v_1 = 1.25 \times 10^7$		$v_2 = 2.5 \times 10^5$		$\beta = 0.0065$	

The transient is analyzed at zero power and the feedback effects can be neglected. The control rod denoted by number 26 is ejected at 0.08 s. The worth of the ejected rod is marginally below the prompt critical value. The delayed neutron precursors are moving together with the fuel of the absorber followers. Scram is initiated at 1 sec by dropping safety rods 23 and 25 at a constant velocity. These rods take 11 seconds to reach the bottom of the core. The drop of the control rod

group 21 is also started at 1 sec with the same velocity. The transient is followed up to 6 s.

To check the necessary accuracy for the spatial discretization the fundamental mode together with the k_{eff} for the initial state of the reactor core is computed. 48 axial planes are considered and the order of the expansions in the z -axis have been fixed to $K_z = 2$. In Table 3.3 the values of the k_{eff} obtained for different values of the order expansions in x and y directions are shown. Also the value obtained with the codes PARCS and DYN3D are included.

Table 3.3: Steady state k_{eff} for different degrees on xy polynomials.

$K_x = K_y = 1$	$K_x = K_y = 2$	$K_x = K_y = 3$	PARCS	DYN3D
1.000894	0.999663	0.999347	0.999376	1.000000

In Figure 3.4(a) the normalized radial power profile for the initial state of the reactor computed with the code based on the Spectral Element Method (called HEXMODKIN) is presented setting $K_x = K_y = 3$ and $K_z = 2$ and 48 axial planes. Also, the results obtained with PARCS and DYN3D are presented. In Figure 3.4(b) the relative error in each node of the radial profile of the HEXMODKIN and PARCS codes is presented, using the solution obtained with DYN3D code as the reference. It is observed that the maximum error obtained is less than a 4%.

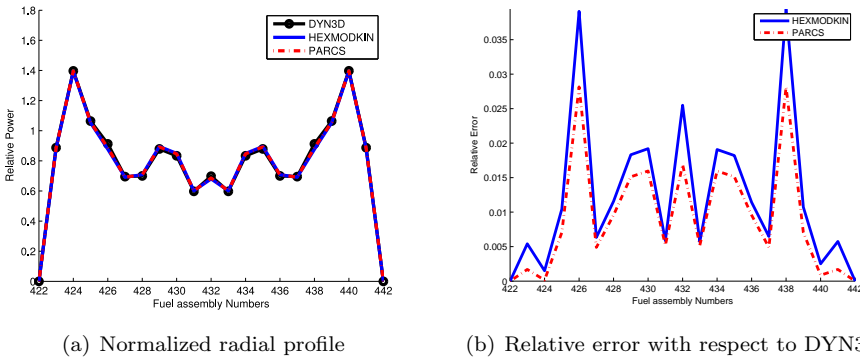


Figure 3.4: Results at $t = 0$ s.

The obtained average power evolution along the transient is compared with the reference result provided by DYN3D code and the result provided by PARCS in Figure 3.5.

It is noticed that there are divergences in the peak power prediction. This can be due because [58] the transient analysed is a reactivity initiated transient involv-

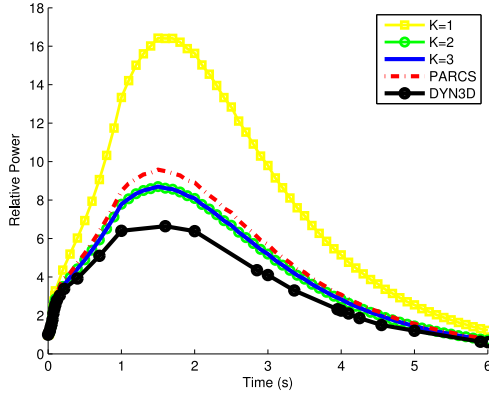
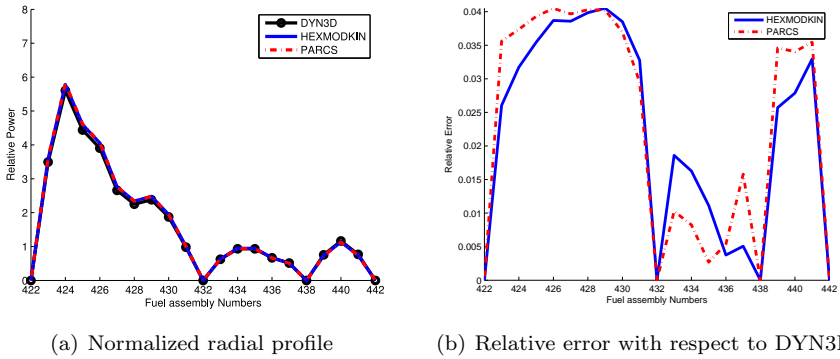


Figure 3.5: Average power evolution.

ing a rod worth that is near the prompt critical value and a small deviation in the dynamic rod worth may cause significant deviations on the power prediction. Nevertheless, the agreement in the normalized power radial profile at second 6 is good with both the DYN3D and the PARCS code, as it is shown in Figure 3.6.



(a) Normalized radial profile

(b) Relative error with respect to DYN3D

Figure 3.6: Results at $t = 6s$.

3.1.3.1 Numerical results for the rod cusping correction

To test the performance of the rod cusping correction method in 3-D geometries, a transient for a small reactor where the rod cusping is a dominant effect has been studied. Fig 3.7 shows horizontal layout map of the core, for which the hexagonal lattice pitch is 23.6 cm.

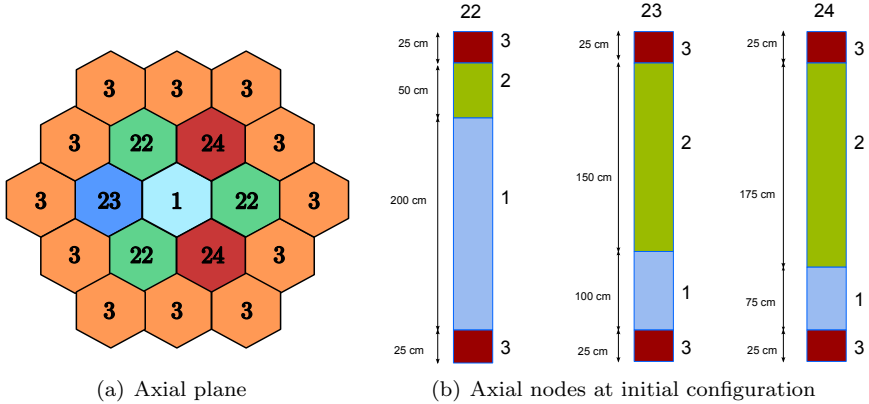


Figure 3.7: Small Reactor

The material distribution is described by “type” numbers and 2-group data belonging to the different material types are given in Table 3.4. Time dependent data are given in Table 3.5.

Table 3.4: Cross sections (in cm^{-1}) for the 3-D small reactor.

Fuel	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.346557	0.008312	0.016976	0.004413
	2	0.370075	0.064282		0.072784
2	1	1.337728	0.008745	0.016000	0.005491
	2	0.367411	0.079145		0.104256
3	1	1.332264	0.009411	0.014974	0.006990
	2	0.363171	0.099536		0.147261

Type number 1 represents the fuel assemblies, type number 2 represents the absorber assemblies, and type number 3 represents the reflector. Type numbers 22 and 24 represent the positions where, either absorber assemblies (type 2) or their fuel followers (type 1), has to be placed into, due to the axial position of the given absorber assembly. Type 23 is used for the ejected rod. One layer of axial reflector at the top, another one at the bottom of the reactor, and a radial layer of reflec-

Table 3.5: Neutron precursors parameters for the reactor.

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
β_i	0.000247	0.0013845	0.001222	0.0026455	0.000832	0.000169
λ_i (s^{-1})	0.0127	0.0317	0.115	0.311	1.4	3.87
	$v_1 = 1.25 \times 10^7$		$v_2 = 2.5 \times 10^5$		$\beta = 0.0065$	

tor are used. The albedo boundary condition is applied on the outer edge of the reflector nodes, with the albedo boundary condition imposed by the extrapolation length $2 * D_g$.

The height of the core is 300 cm, and 12 axial planes are considered, each one of 25 cm. The first one and the last one are reflector layers defined by material of type 3. The initial position of control rod group 22 (see Fig 3.7) is at 200 cm above the bottom reflector (Follower in 2nd-9th axial nodes, absorber rod in 10th-11th axial nodes), the initial position of control rod group 23 is at 100 cm above the bottom reflector, and the initial position of control rod group 24 is at 75 cm above the bottom reflector.

In contrast to reality, the top and bottom reflectors do not contain control rod absorbers or followers in this benchmark.

The transient is defined as follows:

- At time $t = 0.0s$, starting from the initial configuration, the height of the absorber (material 2) at position 23 becomes smaller at constant velocity until it is completely removed at time $t = 0.15s$, remaining only the fuel (material 1) in the unrodded cells, simulating a rod ejection accident.
- From time $t = 0.15s$ until time $t = 1.0s$ nothing happens.
- When the security system acts, a scram is produced inserting absorbers at positions 22 from time $t = 1.0s$ until the bottom of the reactor is reached at time $t = 9.0s$.

Evolution of the normalized mean power, starting from a critical situation of the reactor, is monitored until time $t = 7.0s$.

The flux weighted method for the rod cusping effect proposed here will be compared with the classical volume weighted method. As a reference value, a calculation of the k_{eff} where the heterogeneous cross sections of the partially rodded node is used, and it will be called real method.

The evolution of the k_{eff} along the transient by three methods is shown at Figure 3.8. The first of these methods, named “Vol” at the legend of the Figure is the volume weighted method, the one named “Flux” is the flux weighted method,

and the curve for the k_{eff} named “Real” is obtained solving a stationary problem where the mesh fits exactly with the peaks of the control rod to obtain a reference solution without the rod cusping effect. Good results for the flux weighted methods compared with the classical volume weighted method are observed.

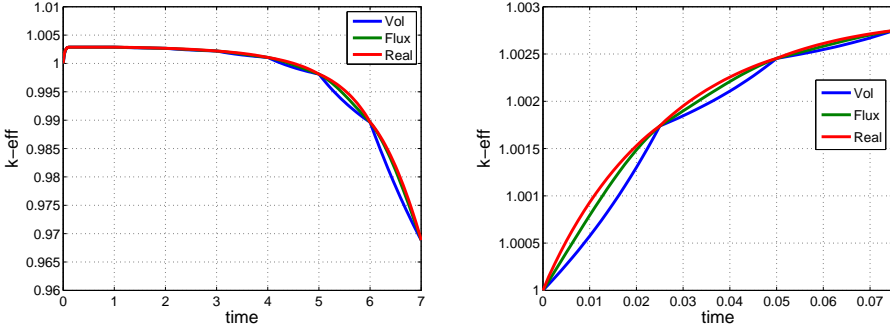


Figure 3.8: Evolution of the k_{eff} (LEFT); Detailed view of the first 0.75 s (RIGHT).

The absolute error in the k_{eff} calculation for the volume weighted method and for the flux weighted method is shown in Figure 3.9.

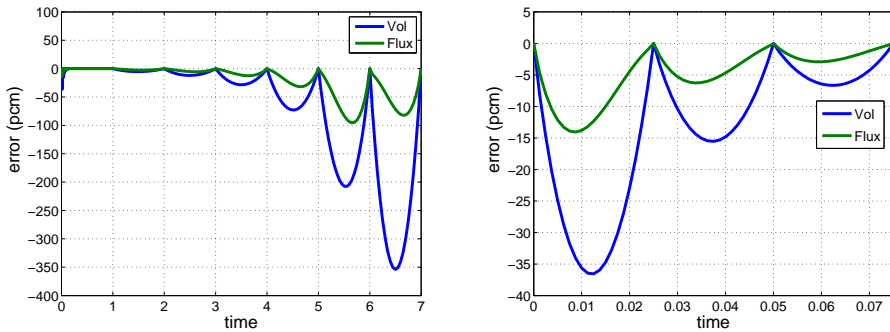


Figure 3.9: Absolute error respect to the method (LEFT); Detailed view (RIGHT).

The effect of the rod cusping has been evaluated in the k_{eff} calculations, but it can also be observed in the evolution of the normalized power (see Figure 3.10). The reference solution is obtained by homogenizing the nuclear cross sections of the partial rodded nodes along the transient with the stationary solution for each time, and then the reference power evolution is obtained using these homogenized cross sections for each time step.

Summarizing, for the time discretization a one-step implicit method has been used. This method needs to solve a system of linear equations for each time

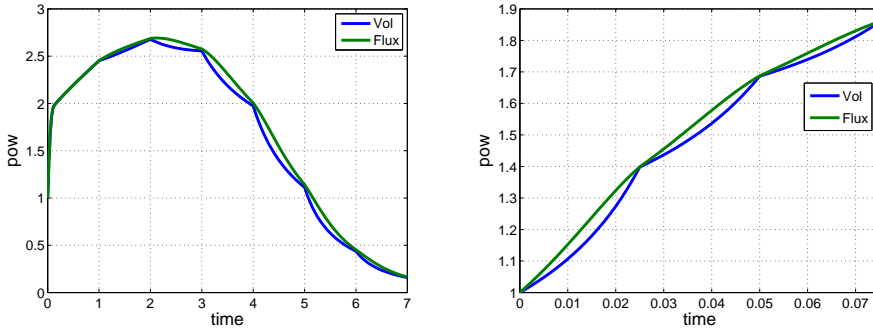


Figure 3.10: Evolution of the power along the transient (LEFT); Detailed view (RIGHT).

step, and the size of this system depends on the spatial accuracy needed in the calculations. The performance of the method has been tested solving the 3-D transient benchmark AER-DYN-001, showing that the proposed method provides good results when it is compared with other codes such as PARCS and DYN3D.

When nodal methods are used, the cross sections have to be homogenized over each node. In a transient based on moving control rods, when the homogenization method used is the volume weighting method, the rod cusping effect arise in the k_{eff} calculations, leading to large errors in the approximated flux near to the tip of the control rod along the transient. To solve this problem, a homogenization method based on weighting the cross sections with the flux is presented here. This method has been tested for a 3-D transient of a small reactor, where the rod cusping effect is a dominant fact. Good results are obtained, either at the moment of a ejection control rod, or where a scram is occurring.

3.2 Block Preconditioning

The solution of the time dependent neutron diffusion equation using a backward method needs to solve a large system of linear equations for each time step and this process is the most expensive one from the computational point of view. To speed-up the solution of the linear system a preconditioning technique is needed.

Different variational techniques are proposed to accelerate two classical block iterative methods for the solution of the linear systems arising from the time discretization of the multigroup time dependent neutron diffusion equation (3.1) when a backward method is used to integrate in time. Also, a block preconditioner is proposed. The performance of these methods is tested solving linear systems of

two transient problems associated with a 2D VVER 440 reactor and with a 3D VVER 440 reactor.

Since the energy groups structure defines a natural partition of the matrix of the system into different blocks that have good properties, in particular they are symmetric and positive definite matrices, two different methods to solve the linear systems are studied that use this block structure, and different variational acceleration techniques are used to improve the convergence behaviour of the methods.

As shown in section 3.1 of this chapter, the discretization of the neutron diffusion equation (3.1) by means of a one-step implicit method is written as

$$[T^{n+1}] \psi^{n+1} = [R^n] \psi^n + \sum_{k=1}^K \lambda_k e^{-\lambda_k \Delta t} X [C_k^n] . \quad (3.13)$$

Taking into account the structure of matrices L and M , we reexpress (3.13) as the system of linear equations

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \psi^{n+1} = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \psi^n + \sum_{k=1}^K \lambda_k e^{-\lambda_k \Delta t} \begin{bmatrix} C_k^n \\ 0 \end{bmatrix} , \quad (3.14)$$

where

$$\begin{aligned} T_{11} &= \frac{1}{\Delta t} v_1^{-1} + L_{11}^{n+1} - \hat{b} M_{11}^{n+1}, & T_{12} &= -\hat{b} M_{12}^{n+1}, \\ T_{21} &= -L_{21}^{n+1}, & T_{22} &= \frac{1}{\Delta t} v_2^{-1} + L_{22}^{n+1}, \\ R_{11} &= \frac{1}{\Delta t} v_1^{-1} + \hat{a} M_{11}^n, & R_{12} &= \hat{a} M_{12}^n, \\ & & R_{22} &= \frac{1}{\Delta t} v_2^{-1}, \end{aligned}$$

where \hat{a} and \hat{b} have been previously defined by

$$\hat{a} = \sum_{k=1}^K \lambda_k \beta_k a_k, \quad \hat{b} = 1 - \beta + \sum_{k=1}^K \lambda_k \beta_k b_k. \quad (3.15)$$

This is a large system of linear equations that has to be solved for each time step, thus an efficient method has to be used in order to perform the computations in a reasonable time.

3.2.1 Solvers for the linear systems

To use the Euler backward discretization method exposed above, it is necessary to solve systems of linear equations large and sparse with the following block structure

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} . \quad (3.16)$$

Due to the large size of system (3.16), an iterative method is recommended for its resolution. The finite element method used for the spatial discretization of the neutron diffusion equation leads to blocks T_{11} , T_{12} , T_{21} , and T_{22} , with special properties, in particular, the blocks T_{11} , T_{12} , T_{21} , and T_{22} are symmetric and diagonal dominant matrices. The whole matrix of the system of linear equations, T , has none of these properties. In this way, we propose the use of block iterative methods that make use of the properties of the blocks to solve the whole system [53].

3.2.1.1 Block iterative algorithms

Two classical block iterative methods are considered, the block Jacobi method and the block Gauss-Seidel method.

Each iteration of the block Jacobi method has to solve the problem

$$\begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} \psi_1^{i+1} \\ \psi_2^{i+1} \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} - \begin{bmatrix} 0 & T_{12} \\ T_{21} & 0 \end{bmatrix} \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} , \quad (3.17)$$

which is reduced to solve the systems

$$\begin{aligned} T_{11}\psi_1^{i+1} &= E_1 - T_{12}\psi_2^i , \\ T_{22}\psi_2^{i+1} &= E_2 - T_{21}\psi_1^i , \end{aligned}$$

where ψ_1^0 and ψ_2^0 are a suitable initial guess. The iteration is stopped when the following stopping criterion is satisfied

$$\left\| \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} - \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} \psi_1^{i+1} \\ \psi_2^{i+1} \end{bmatrix} \right\| < \text{tol} \left\| \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \right\| , \quad (3.18)$$

where tol is a constant, which is set equal to 10^{-5} .

Each iteration of the block Gauss-Seidel method has to solve the problem

$$\begin{bmatrix} T_{11} & 0 \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} \psi_1^{i+1} \\ \psi_2^{i+1} \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} - \begin{bmatrix} 0 & T_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} , \quad (3.19)$$

which is reduced to solve the systems

$$\begin{aligned} T_{11}\psi_1^{i+1} &= E_1 - T_{12}\psi_2^i , \\ T_{22}\psi_2^{i+1} &= E_2 - T_{21}\psi_1^{i+1} , \end{aligned}$$

where ψ_1^0 and ψ_2^0 are also a suitable initial guess. The stopping criterion is the same as the one used for the previous method.

Using these processes, only linear systems associated with matrices T_{11} and T_{22} have to be solved. These matrices are sparse, symmetric and diagonal dominant, therefore, these linear systems can effectively be solved by means of a standard iterative method as the preconditioned Conjugate Gradient method [59].

Usually, to solve the system (3.16) using either the method (3.17) or (3.19), it is necessary to perform many external iterations of the process, then it will be convenient to accelerate its convergence. To accomplish with it, we have alternated these external iterations with iterations of a variational acceleration technique.

3.2.1.2 Variational acceleration techniques

For a generic system of linear equations

$$T\psi = E ,$$

a possible variational technique relies on iterations of the form

$$\psi^{i+1} = \psi^i + \sum_{l=1}^k \alpha_l d^{i-l+1} , \quad (3.20)$$

where the vectors d^j are

$$d^j = \psi^j - \psi^{j-1} ,$$

and the acceleration factors, α_l , are computed imposing that the residual of the new solution has to be minimum. Equation (3.20) is equivalent to equation for the residual

$$r^{i+1} = r^i - \sum_{l=1}^k \alpha_l T d^{i-l+1} . \quad (3.21)$$

Thus, the quantity to be minimised is

$$\varepsilon^2 = (r^{i+1})^T (r^{i+1}) = \left(r^i - \sum_{l=1}^k \alpha_l T d^{i-l+1} \right)^T \left(r^i - \sum_{l=1}^k \alpha_l T d^{i-l+1} \right) .$$

Computing

$$\frac{\partial \varepsilon^2}{\partial \alpha_l} = 0 , \quad l = 1, \dots, k ,$$

leads to the reduced system of normal equations

$$M\alpha = b , \quad (3.22)$$

where matrices elements are

$$M_{lm} = (Td^{i-l+1})^T (Td^{i-m+1}); \quad b_l = (Td^{i-l+1})^T r^i .$$

This variational technique will be called technique *var1*.

Different variational techniques will be used. The variational technique called *var2* is based on iterations of the form

$$\psi^{i+1} = \psi^i + \sum_{l=1}^k \alpha_l d^{i-l+1} + \sum_{l=1}^k \beta_l r^{i-l+1} , \quad (3.23)$$

where vectors d_i^l are direction vectors corresponding to the different energy groups.

Taking into account the variational acceleration techniques *var1* and *var2*, other two techniques are proposed. These techniques take into account the group structure of the problem, and then use one parameter for the fast group and other parameter for the thermal group, due to the fact that they scale in a different way.

The variational technique called *var3*, is based on iterations of the form

$$\begin{bmatrix} \psi_1^{i+1} \\ \psi_2^{i+1} \end{bmatrix} = \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} + \sum_{l=1}^k \alpha_{1,l} \begin{bmatrix} d_1^{i-l+1} \\ 0 \end{bmatrix} + \sum_{l=1}^k \alpha_{2,l} \begin{bmatrix} 0 \\ d_2^{i-l+1} \end{bmatrix} . \quad (3.24)$$

The variational technique *var4*, is based on iterations of the form

$$\begin{aligned} \begin{bmatrix} \psi_1^{i+1} \\ \psi_2^{i+1} \end{bmatrix} &= \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} + \sum_{l=1}^k \alpha_{1,l} \begin{bmatrix} d_1^{i-l+1} \\ 0 \end{bmatrix} + \sum_{l=1}^k \alpha_{2,l} \begin{bmatrix} 0 \\ d_2^{i-l+1} \end{bmatrix} \\ &+ \sum_{l=1}^k \beta_{1,l} \begin{bmatrix} r_1^{i-l+1} \\ 0 \end{bmatrix} + \sum_{l=1}^k \beta_{2,l} \begin{bmatrix} 0 \\ r_2^{i-l+1} \end{bmatrix} . \end{aligned} \quad (3.25)$$

For all these variational techniques the parameters α_l and β_l are derived from a normal equations similar to equations (3.22).

3.2.1.3 Preconditioning the whole system

It is well known that the inverse of a matrix M with the following block structure

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} ,$$

with regular blocks A and D , is the matrix

$$M^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} . \quad (3.26)$$

Thus, to find a cheap preconditioner to our global system (3.16), we consider the matrix

$$\tilde{T} = \begin{bmatrix} \tilde{T}_{11} & \tilde{T}_{12} \\ \tilde{T}_{21} & \tilde{T}_{22} \end{bmatrix}, \quad (3.27)$$

where the new blocks, \tilde{T}_{ij} are defined as the diagonal blocks defined by the diagonal entries of block matrix T_{ij} .

Using the inversion formula (3.26), it is easy to find the inverse of \tilde{T} , and this inverse will be used as a right preconditioner for our global system (3.16) as follows

$$\begin{bmatrix} \tilde{T}_{11} & \tilde{T}_{12} \\ \tilde{T}_{21} & \tilde{T}_{22} \end{bmatrix}^{-1} \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \begin{bmatrix} \tilde{T}_{11} & \tilde{T}_{12} \\ \tilde{T}_{21} & \tilde{T}_{22} \end{bmatrix}^{-1} \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}. \quad (3.28)$$

Note that the block matrices arising from this preconditioned system are non symmetric and, due to this fact, the systems associated with the new blocks must be solved with, for example, the preconditioned BICGSTAB method [60].

3.2.2 Numerical Results

The performance of the methods presented above is tested in a transient for a two-dimensional VVER 440 reactor core proposed in [61], and for the 3-D transient benchmark AER-DYN-001 proposed in [57, 62].

3.2.2.1 Two-dimensional VVER 440 problem

For the two-dimensional problem, the nuclear cross section given in the 3D benchmark have been collapsed into a single plane. The description of the geometry for this problem, together with the data for the cross sections and the neutron precursors parameters are presented in [61]. Materials of the two-dimensional reactor have been defined as it is shown in Figure 3.11.

The cross sections of the materials and the neutron precursors parameters for the transient are presented in Tables 3.6 and 3.7.

A transient that simulates the movement of two control rods has been defined by means of the time evolution of the absorption cross section Σ_{a2} for the material 8 as follows,

$$\Sigma_{a2}(t) = \begin{cases} 0.118870 \cdot (1 - t) + 0.016917 \cdot t & \text{if } 0 \leq t \leq 1, \\ 0.118870 \cdot (t - 1) + 0.016917 \cdot (2 - t) & \text{if } 1 \leq t \leq 2, \\ 0.118870 & \text{if } 2 \leq t \leq 3. \end{cases}$$

This function simulates that two control rods are extracted during the first second then, the same rods are inserted during the next second and the transient is

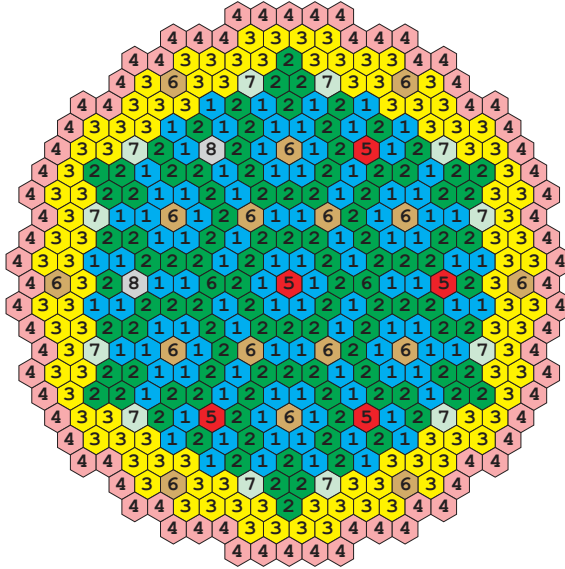


Figure 3.11: VVER 440 2D.

Fuel	Group	$D_g(cm)$	$\Sigma_{ag}(cm^{-1})$	$\Sigma_{g \rightarrow g+1}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$
1	1	1.346557	0.008312	0.016976	0.004413
	2	0.370075	0.064282		0.072784
2	1	1.337728	0.008745	0.016000	0.005491
	2	0.367411	0.079145		0.104256
3	1	1.332264	0.009411	0.014974	0.006990
	2	0.363171	0.099536		0.147261
4	1	1.447520	0.000933	0.032215	0.000000
	2	0.251741	0.033037		0.000000
5	1	1.231711	0.012120	0.020782	0.001345
	2	0.240027	0.118846		0.027352
6	1	1.337727	0.008747	0.015996	0.005492
	2	0.367479	0.079153		0.104316
7	1	1.346561	0.008317	0.016968	0.004416
	2	0.370177	0.064282		0.072846
8	1	1.231640	0.012123	0.020785	0.001342
	2	0.239942	0.118870		0.027299

Table 3.6: Cross sections for the 2-D VVER 440 reactor.

followed one more second, and the mean power evolution is shown at Figure 3.12.

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
β_i	0.000247	0.0013845	0.001222	0.0026455	0.000832	0.000169
$\lambda_i (s^{-1})$	0.0127	0.0317	0.115	0.311	1.4	3.87
	$v_1 = 1.25 \times 10^7$		$v_2 = 2.5 \times 10^5$		$\beta = 0.0065$	

Table 3.7: Neutron precursors parameters for the reactor VVER 440.

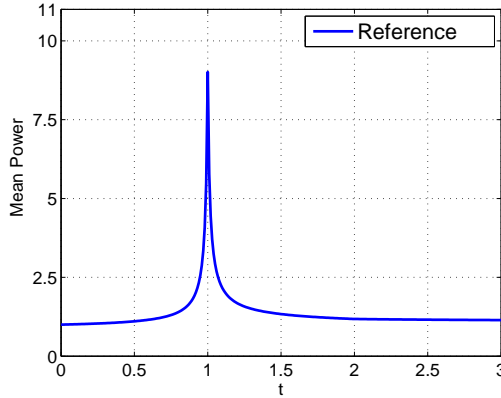


Figure 3.12: Power evolution for the VVER 440 2D transient.

Calculations have been carried out for a system of equations corresponding to a given time step of the transient. The spatial discretization of the equations has been done using polynomials of degree $K_{xy} = 5$. The block matrices corresponding to this polynomial expansion have size $N = 31951$, and the nonzero elements of the blocks T_{11} and T_{22} are 659799 for each one.

To test the numerical methods proposed above, a theoretical solution, ψ^* has been set equal to a vectors of ones, computing the right hand vector, E , as the product of the matrix T by this theoretical solution.

A combination of five iterations of the block iterative algorithm combined with one iteration of a variational acceleration technique, setting $K_{xy} = 5$ in the variational definitions (3.20)-(3.25), has shown to be the optimal. Thus, numerical experiments use this combination. The different block iterative methods combined with the different variational acceleration techniques proposed for both preconditioned and non-preconditioned problems have been tested.

Table 3.8, shows results for the solution of the linear systems corresponding to matrices obtained with the spatial discretization of degree $K_{xy} = 5$. In this Table, the number of global iterations needed to solve the system together with the CPU time needed to solve the system are shown. The different methods have been programing in FORTRAN 90 and have been run in a personal computer with two

processors Intel Core 2, of 1.86 GHz and with 1 GB of RAM. The best variational

	Without preconditioning		With preconditioning	
	Jacobi	Gauss-Seidel	Jacobi	Gauss-Seidel
<i>var1</i>	198 (119.75 s)	48 (33.71 s)	82 (80.56 s)	18 (21.65 s)
<i>var2</i>	176 (107.35 s)	54 (33.79 s)	84 (80.09 s)	18 (22.51 s)
<i>var3</i>	84 (52.01 s)	54 (39.00 s)	52 (48.86 s)	18 (22.40 s)
<i>var4</i>	168 (108.45 s)	60 (42.40 s)	30 (32.82 s)	18 (22.42 s)

Table 3.8: Number of iterations and CPU time for matrices obtained with $K_{xy} = 5$.

acceleration technique results in terms of the CPU-time for each block method, with or without preconditioned is presented in boldface.

At this Table we can see that the preconditioned methods work better than the non-preconditioned ones, and block Gauss-Seidel method is faster than the Jacobi method. Also, it can be seen that the number of iterations in the block Gauss-Seidel with preconditioning is not dependent on the variational technique used.

3.2.2.2 Three-dimensional VVER 440 problem

The three-dimensional benchmark VVER 440 (See Figure 3.13) used in Section 3.1.3 is used to check the convergence behaviour of the methods on three-dimensional matrices, which are larger than the ones arising in two-dimensional problems.

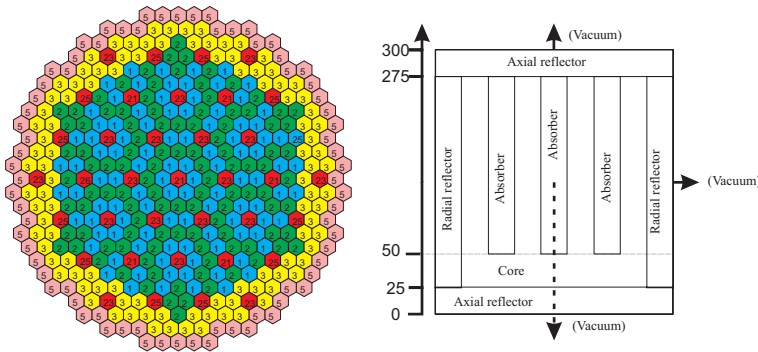


Figure 3.13: Geometry of VVER 440 reactor core.

The matrices for steady state configuration of the reactor are used. Also a theoretical solution vector of all entries equal to one is used, and the right hand term is determined by multiplying the matrix by the solution vector, as it was done in the previous example.

The spatial discretization of the equations has been done using polynomials of degree $K_{xy} = 1$ and $K_z = 1$. The block matrices corresponding to this polynomial expansion have size $N = 17407$, and the nonzero elements of the blocks T_{11} and T_{22} are $NNZ = 335479$ for each one. Table 3.9 shows results for the solution of this linear systems. In this Table, the number of global iterations needed to solve the system, and the CPU time needed to solve the system are reported.

	Without preconditioning		With preconditioning	
	Jacobi	Gauss-Seidel	Jacobi	Gauss-Seidel
<i>var1</i>	155 (12.80 s)	60 (5.24 s)	134 (13.03 s)	24 (3.05 s)
<i>var2</i>	168 (16.04 s)	66 (6.78 s)	110 (12.28 s)	24 (3.29 s)
<i>var3</i>	114 (8.92 s)	66 (6.04 s)	45 (4.59 s)	24 (3.10 s)
<i>var4</i>	78 (10.88 s)	72 (9.83 s)	30 (5.34 s)	24 (4.58 s)

Table 3.9: Number of iterations and CPU time for matrices obtained with $K_{xy} = 1$ and $K_z = 1$.

At Table 3.9, the results are similar to the ones in Table 3.8, i.e., the preconditioned methods work better than the non-preconditioned ones, and block Gauss-Seidel method is faster than the Jacobi method. Also, it can be seen that the number of iterations in the block Gauss-Seidel with preconditioning is not dependent on the variational technique used.

Another configuration for the spatial discretization of the equations has been done using polynomials of degree $K_{xy} = 2$ and $K_z = 2$. The block matrices corresponding to this polynomial expansion have size $N = 130075$, and the nonzero elements of the blocks T_{11} and T_{22} are $NNZ = 5693803$ for each one. Table 3.10 shows results for the solution of this linear system. The number of global iterations needed to solve the system and the CPU time needed to solve the system are reported. The best variational acceleration technique results in terms of the

	Without preconditioning		With preconditioning	
	Jacobi	Gauss-Seidel	Jacobi	Gauss-Seidel
<i>var1</i>	207 (299.97 s)	60 (91.27 s)	113 (238.56 s)	32 (83.81 s)
<i>var2</i>	299 (487.80 s)	60 (108.46 s)	72 (173.99 s)	30 (79.85 s)
<i>var3</i>	131 (177.06 s)	66 (106.07 s)	51 (108.77 s)	41 (90.46 s)
<i>var4</i>	126 (269.80 s)	66 (151.72 s)	42 (134.50 s)	24 (85.13 s)

Table 3.10: Number of iterations and CPU time for matrices obtained with $K_{xy} = 2$ and $K_z = 2$.

CPU-time for each block method, with or without preconditioned is also presented in boldface in the tables for the three-dimensional problem. At Table 3.10, where the matrices are quite larger than in the other problems, the fastest method is the Gauss-Seidel method, with and without preconditioning, over the Jacobi method.

Also, the preconditioned Gauss-Seidel method is the fastest one, but in this case, the number of iteration is dependent on the variational technique used, being the best acceleration technique for this method the technique var1.

Since the ordinary differential equations arising from the spatial discretization of the neutron diffusion equation in hexagonal geometry are stiff, implicit methods are used for the time discretization of these equations. This implies that for each time step, a large and sparse system of linear equations has to be solved. The groups of energy considered in these equations define a block structure in the matrix associated to these systems. In this way, we have studied the performance of different variational acceleration techniques applied to classical block iterative methods for the solutions of these systems of equations such as the block Jacobi and the block Gauss-Seidel methods. Also we have proposed a cheap preconditioner for the whole system, and the behaviour of the variational acceleration techniques is tested for the preconditioned systems. Two problems, a 2D transient and a 3D transient benchmark, are used to test the behaviour of the variational techniques. For these cases, we have observed that the most efficient acceleration technique depends on the method, and these results are also sensitive to the effect of the preconditioner.

As a future work, this research can be extended analysing the behaviour of the methods during a whole transient, where for each time step the system is a perturbed configuration of the previous one, and the solution at the previous time step is a good initial guess for the new system. Also, parallel codes can take advantage of the block structure of the matrices, and the behaviour of the acceleration techniques on these situations might be studied.

3.3 Spectral Preconditioning

The algebraic system (3.29),

$$[T^{n+1}] \psi^{n+1} = [R^n] \psi^n + \sum_{k=1}^K \lambda_k e^{-\lambda_k \Delta t} X [C_k^n] , \quad (3.29)$$

obtained from the discretization by a backward method of the time dependent neutron diffusion equation (3.1), can also be solved with a standard Krylov solver without using the block structure of the matrices given by the multigroup approximation of the energy. To solve this problem in an efficient way, a good preconditioner for the different sparse matrices associated with the systems of linear equations obtained after the discretization of the problem has to be developed. A spectral preconditioner is studied for a sequence of systems of linear equations based on modifying the eigenvalues distribution of the coefficient matrices, and different strategies to update the preconditioner are compared for the matrices appearing in a typical transient associated with a VVER-type nuclear reactor.

Krylov methods suffer from slow convergence unless a preconditioner is used. The most popular preconditioners for linear systems are based on incomplete factorizations of the coefficient matrix. Incomplete LU preconditioners are robust, but for very large matrices they are very expensive from the point of view of the memory needed for their storage. In this way, an alternative methodology is proposed here that can be implemented using only *matrix-vector* products and has not storage problems.

It is known that the rate of convergence of Krylov methods depends on the distribution of the eigenvalues of the system matrix together with its eigenvectors. In this way, the Krylov methods present slow convergence if the eigenvalues of the matrix are close to zero. Spectral preconditioners use this idea to improve the convergence implementing transformations on the linear system, in such a way that the matrix obtained after the spectral transformation has its eigenvalues shifted away from zero. The spectral transformations can be implemented using matrix-vector products and can be applied without the necessity of building explicitly the system matrix [63].

3.3.1 Spectral Preconditioner

Let us consider a linear system as

$$Ax = b,$$

and an initial preconditioner M . It is proposed the use of a preconditioner based on a low rank correction that shifts the smallest eigenvalues of the matrix.

Let V be the rank- k matrix associated to a right invariant subspace for the preconditioned matrix AM , such as

$$AMV = VJ_k$$

where the eigenvalues of J_k are $\{\lambda_1, \dots, \lambda_k\}$, the following proposition holds [63]:

Proposition 1 *Let W be such that $A_c = WAMV$ is non singular. Defining*

$$M^{(2)} = M + MVA_c^{-1}W,$$

the eigenvalues of $AM^{(2)}$ are

$$\begin{cases} \eta_i^{(2)} = \lambda_i + 1 & \text{if } i \leq k \\ \eta_i^{(2)} = \lambda_i & \text{if } i > k. \end{cases}$$

This preconditioner can be applied in a recursive way to a sequence of linear systems,

$$A^{(n)}x^{(n)} = b^{(n)}, \quad n = 1, 2, \dots \quad (3.30)$$

where the sequence of matrices and right hand terms correspond to the ones in equation (3.29)

$$\begin{aligned} A^{(n+1)} &= [T^{n+1}] , \\ b^{(n+1)} &= [R^n] \psi^n + \sum_{k=1}^K \lambda_k e^{-\lambda_k \Delta t} X [C_k^n] , \\ x^{(n+1)} &= \psi^{n+1} . \end{aligned}$$

Then for the $(l + 1)$ -th linear system the preconditioner used is of the form

$$M^{(l+1)} = M + \sum_{j=1}^l M^{(j)} V^{(j)} \left(W^{(j)} A M^{(j)} V^{(j)} \right)^{-1} W^{(j)} \quad (3.31)$$

For the preconditioner implementation we have used $W^{(j)} = V^{(j)H}$.

In some transients, the spectral properties of the system matrix change slowly with time, then it is assumed than the invariant subspace associated to the lower part of the spectrum for the matrix of system n can be used to precondition the matrix system at step $n + 1$.

The problem of approximating the invariant subspace of each matrix $A^{(n)}$ is solved by means of the use of a Krylov solver such as the GMRES-DR(m, k) [64] for the linear systems, which itself is based on approximating an invariant subspace for the matrix system as a strategy to restart the GMRES method.

To control the memory required by the preconditioner (3.31), a limited number of low rank matrices $V^{(j)}$ defining the invariant subspace can be stored, and because the matrix $A^{(n)}$ is changing in time, the strategy is to define the preconditioner as

$$M^{(l+1)} = M + \sum_{j \in \mathcal{I}} M^{(j)} V^{(j)} \left(W^{(j)} A M^{(j)} V^{(j)} \right)^{-1} W^{(j)} \quad (3.32)$$

where \mathcal{I} is a set of L_{max} indices j such that the invariant subspaces at these indices are approximately associated with the lowest part of the spectrum of matrix $A^{(n+1)}$. Matrix $W^{(j)} A M^{(j)} V^{(j)}$ is $k \times k$ size, so the inverse can be calculated in a very fast way for low values of k ($k = 1, 2, 3, \dots$).

3.3.2 Numerical Results

To test the preconditioner (3.32) for the neutron diffusion equation, we consider the transient in the small reactor of type VVER (see Figure 3.14 described at Section 3.1.3.1).

Evolution of the normalized mean power, starting from a critical situation of the reactor, is monitored until time $t = 7.0$ s. Figure 3.15 shows the evolution of the

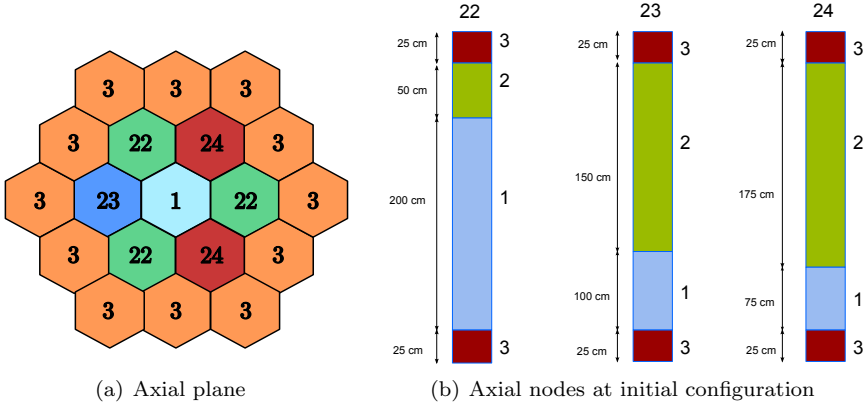


Figure 3.14: Small reactor.

power during the transient computed with the Spectral Element Method of order 2 (HEXFEM when referenced here) and the solution obtained with PARCS code [65]. The preconditioner is tested with a sequence of 100 linear systems corresponding

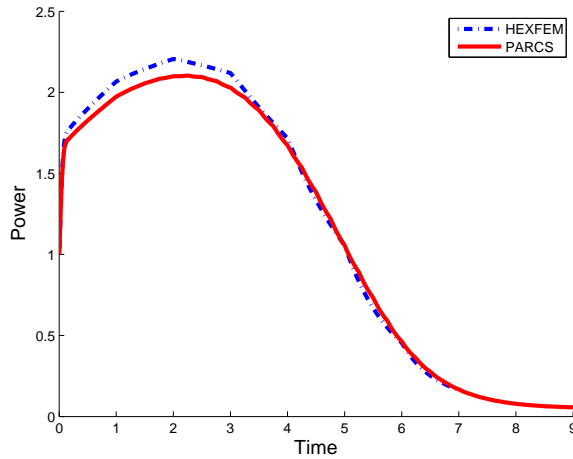


Figure 3.15: Power along the transient.

to the fast part of the transient (from $t = 0.0s$ to $t = 0.125s$), where the time step used is $\Delta t = 0.00125s$.

Two different strategies are considered to test the preconditioner. The first one varies the number of terms considered at sum (3.32). We will use $L_{max} = 5, 10,$

15, 20 and 25. The second strategy consists of changing the rank of the subspace considered in each term of this sum. In this way we will use $k = 1, 2,$ and 3 . Initial preconditioner M is chosen in all cases as the identity matrix, to simulate the situations where the system matrix A is not explicitly known, only its product by a vector.

First an algebraic problem obtained with linear finite elements for the spatial discretization, is considered. The system matrices are $N \times N$, where $N = 1898$, and the number of non zero elements of the matrices is $NNZ = 65860$. The number of iterations needed by the Krylov method GMRES to solve the 100 first time steps of the transient, when the different spectral preconditioner strategies are used are shown in Figures 3.16–3.17.

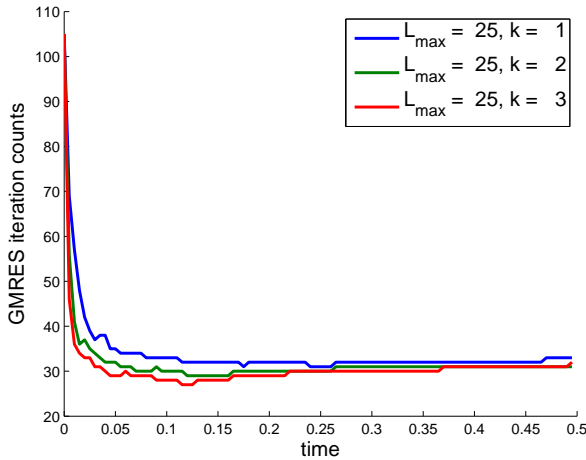


Figure 3.16: Influence of the rank of the subspace, k , for a fixed $L_{max} = 25$. (spatial discretization with polynomial degree 1, $N = 1898$, $NNZ = 65860$).

We observe that the Krylov solver needs about 110 iterations for the first system and the preconditioner success on the reduction of the iteration number very quickly. When the number of iterations is reduced about 35 the preconditioner stops improving the number of iterations required to solve the linear systems.

At Figures 3.18 - 3.19, the results corresponding to an algebraic problem obtained from a spatial discretization of the reactor using second order finite elements are shown. The system matrices are of dimension $N = 12950$, and the number of non zero elements of the matrices is $NNZ = 1064284$.

We observe that starting with 380 iterations for the first system, the preconditioner success on the reduction of the iteration number very quickly, and when the number

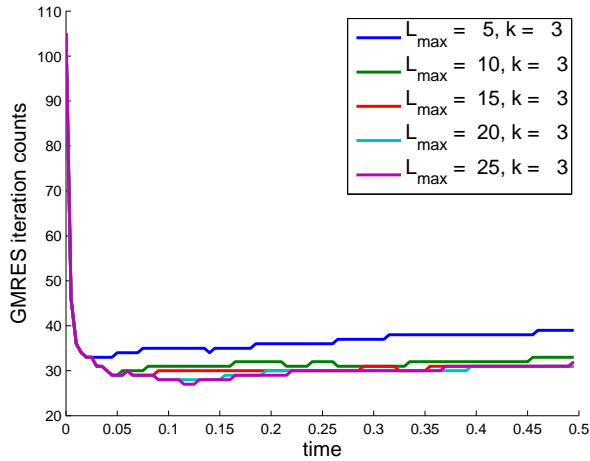


Figure 3.17: Influence of the number of terms, L_{\max} , for a fixed $k = 3$. (spatial discretization with polynomial degree 1, $N = 1898$, $NNZ = 65860$).

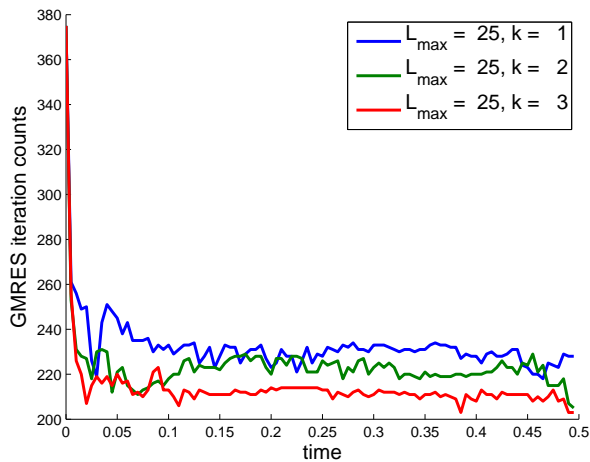


Figure 3.18: Influence of the rank of the subspace, k , for a fixed $L_{\max} = 25$. (spatial discretization with polynomial degree 2, $N = 12950$, $NNZ = 1064284$).

of iterations is reduced around 220 – 240 the preconditioner stops improving the number of iterations required to solve the linear systems.

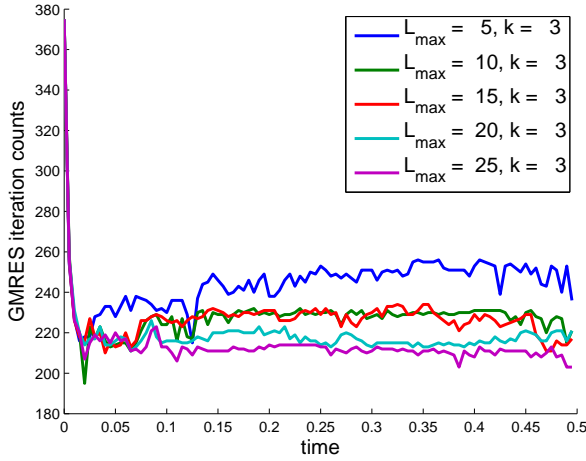


Figure 3.19: Influence of the number of terms, L_{max} , for a fixed $k = 3$. (spatial discretization with polynomial degree 2, $N = 12950$, $NNZ = 1064284$).

Summarizing we have that spatial discretization of time dependent neutron diffusion equation combined with an implicit finite difference method implies that very large systems of equations have to be solved at each time step.

Spectral preconditioners, which are based on a low rank update of a previous preconditioner, improve spectral properties of the system matrix when a linear system is solved with a Krylov method.

To study the feasibility of using this kind of preconditioners for the sequence of linear systems arising in a given transient, a transient associated to a control rod movement in a small reactor has been considered. In a first approach, different strategies for the implementation of the preconditioner have been considered and it has been observed that the preconditioner is very efficient for the reduction of iterations needed to solve the linear systems for the first 10-15 systems and the reduction is stagnated for the rest of systems. It remains for further studies the use of this kind of preconditioners for larger problems such as the ones associated with commercial reactors, and the combination of the preconditioner with different minimal residual Krylov methods.

3.4 Exponential Method

We take into account that since the ordinary differential equations resulting of the discretization of diffusion equations are, in general, stiff, implicit methods are used. The use of time integrators based on the exponential matrix to integrate the semidiscrete system of ODE's arising after the spatial discretization of the time dependent neutron diffusion equation is studied. This is done because some studies suggest than Krylov methods for the approximation of the matrix exponential converge faster than the methods for the solution of the linear systems arising on implicit methods [66, 67]. These methods seems to work better when a good preconditioner for the system matrix is not available for the implicit methods, which is the case when it is desirable not to load the system matrix due to memory requirements, and only the matrix vector product by this matrix is available. Moreover, other studies suggest that polynomial interpolation for the exponential matrix, such as the Real Leja points Method [68], converge as fast as Krylov methods without the memory requirements to save all the vectors defining the Krylov subspace. A Chebyshev approximation for the exponential of a matrix [69, 70] is included for completeness, and also higher order exponential integrators based on the Magnus expansion with a commutator free formulation [71, 72, 73] are tested against a classical second order exponential integrator.

The time dependent neutron diffusion equation (3.1) can also be written with the compact form $y' = Ay$ as follows

$$\begin{bmatrix} \Phi'_1 \\ \Phi'_2 \\ C'_1 \\ \vdots \\ C'_k \end{bmatrix} = \begin{bmatrix} \mathcal{T}_{11} & \mathcal{T}_{12} & \lambda_1 v_1 & \dots & \lambda_k v_1 \\ \mathcal{T}_{21} & \mathcal{T}_{22} & 0 & \dots & 0 \\ \beta_1 \mathcal{M}_{11} & \beta_1 \mathcal{M}_{12} & \lambda_1 & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ \beta_k \mathcal{M}_{11} & \beta_k \mathcal{M}_{12} & 0 & \dots & \lambda_k \end{bmatrix} \cdot \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ C_1 \\ \vdots \\ C_k \end{bmatrix}. \quad (3.33)$$

where \mathcal{T}_{ij} are the blocks associated with the energy groups of the operator \mathcal{T} defined as

$$\mathcal{T} = [v] ((1 - \beta)\mathcal{M} - \mathcal{L}). \quad (3.34)$$

After the spatial discretization by means of the Spectral Element Method, exposed in chapter 2, the semidiscrete two energy groups time dependent neutron diffusion equation is written, as a homogeneous system of linear equations $y' = Ay$, as follows,

$$\begin{bmatrix} P\psi'_1 \\ P\psi'_2 \\ PC'_1 \\ \vdots \\ PC'_k \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & \lambda_1 v_1 P & \dots & \lambda_k v_1 P \\ T_{21} & T_{22} & 0 & \dots & 0 \\ \beta_1 M_{11} & \beta_1 M_{12} & \lambda_1 P & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ \beta_k M_{11} & \beta_k M_{12} & 0 & \dots & \lambda_k P \end{bmatrix} \cdot \begin{bmatrix} \psi_1 \\ \psi_2 \\ C_1 \\ \vdots \\ C_k \end{bmatrix}. \quad (3.35)$$

where T_{ij} are the discrete matrix blocks corresponding to \mathcal{T}_{ij} , and P is the mass matrix.

Then, moving the mass matrix, P , from the matrix to neutron precursors it is obtained PC_k at both vectors. Since it is not necessary to know the value of C_k , we avoid to solve exactly some systems with P . System (3.35) can be rewritten as

$$\begin{bmatrix} P\psi'_1 \\ P\psi'_2 \\ PC'_1 \\ \vdots \\ PC'_k \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & \lambda_1 v_1 & \dots & \lambda_k v_1 \\ T_{21} & T_{22} & 0 & \dots & 0 \\ \beta_1 M_{11} & \beta_1 M_{12} & \lambda_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_k M_{11} & \beta_k M_{12} & 0 & \dots & \lambda_k \end{bmatrix} \cdot \begin{bmatrix} \psi_1 \\ \psi_2 \\ PC_1 \\ \vdots \\ PC_k \end{bmatrix}. \quad (3.36)$$

The vector ψ_i has to be known and, for this reason, it is necessary to invert matrix P . This process can be a very expensive task and, for this reason an approximation \hat{P} for the mass matrix P is considered by means of a mass lumping technique (adding all the elements of each row to the diagonal one and considering only the principal diagonal of the matrix), which is equivalent to calculate the integrals involving polynomials up to order s approximately with a quadrature rule up to order $s - 1$ (see [36]). The new mass lumped matrix \hat{P} is a diagonal matrix very easy to invert, obtaining the modified system

$$\begin{bmatrix} \psi'_1 \\ \psi'_2 \\ PC'_1 \\ \vdots \\ PC'_k \end{bmatrix} = \begin{bmatrix} \hat{P}^{-1}T_{11} & \hat{P}^{-1}T_{12} & \lambda_1 v_1 \hat{P}^{-1} & \dots & \lambda_k v_1 \hat{P}^{-1} \\ \hat{P}^{-1}T_{21} & \hat{P}^{-1}T_{22} & 0 & \dots & 0 \\ \beta_1 M_{11} & \beta_1 M_{12} & \lambda_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_k M_{11} & \beta_k M_{12} & 0 & \dots & \lambda_k \end{bmatrix} \cdot \begin{bmatrix} \psi_1 \\ \psi_2 \\ PC_1 \\ \vdots \\ PC_k \end{bmatrix}. \quad (3.37)$$

3.4.1 Exponential operator and its Magnus Expansion

Let us consider the initial value problem associated with a homogeneous linear ordinary differential equation

$$Y'(t) = A(t)Y(t), \quad Y(t_0) = Y_0. \quad (3.38)$$

If $A(t)$ is a scalar, the solution of this problem is

$$Y(t) = \exp\left(\int_{t_0}^t A(s)ds\right) Y_0. \quad (3.39)$$

When $A(t)$ is a matrix the solution (3.39) is not true any more, except if the matrix operator evaluated at different times, $t_1, t_2 \in [t_0, t]$, commutes, $A(t_1)A(t_2) = A(t_2)A(t_1)$, that is $[A(t_1), A(t_2)] = 0$, (which is the case for a constant matrix A).

Now, looking for the solution of equation (3.38) for the general case, let us use $U(t, t_0)$, defined through

$$Y(t) = U(t, t_0)Y_0, \quad (3.40)$$

some times referred as the time evolution operator. $U(t, t_0)$ satisfies the following differential equation and initial condition

$$U'(t, t_0) = A(t)U(t, t_0), \quad U(t_0, t_0) = I, \quad (3.41)$$

where I stands for the identity matrix. The proposal by Magnus [74] as a solution to (3.41) is a matrix exponential

$$U(t, t_0) = \exp(\Omega(t, t_0)), \quad \Omega(t_0, t_0) = 0, \quad (3.42)$$

and a series expansion for the matrix in the exponent

$$\Omega(t, t_0) = \sum_{k=1}^{\infty} \Omega_k(t, t_0), \quad (3.43)$$

which is known as the *Magnus expansion*.

3.4.1.1 Differential equation for $\Omega(t, t_0)$

If $U(t, t_0)$ obeys equation (3.41), a differential equation for $\Omega(t, t_0)$ will be obtained. As we are not allowed to use the familiar differentiation rules, we have to follow a different route to proceed. Here two important results are fulfilled. The first one is the intuitively clear “group property” of the time-evolution operator:

$$U(t_2, t_0) = U(t_2, t_1)U(t_1, t_0). \quad (3.44)$$

The second one is the related to a famous classical result in matrix algebra known as the Baker-Campbell-Hausdorff (BCH) formula [75] for the product of two exponentials. It states that for any two, in general noncommuting, operators X and Y , one has

$$\exp(X) \exp(Y) = \exp \left(X + Y + \frac{1}{2}[X, Y] + \frac{1}{12}([X, [X, Y]] + [Y, [Y, X]]) + \dots \right) \quad (3.45)$$

The exponent in this equation is an infinite series whose terms are nested commutators of increasing order.

The simplicity in (3.45) is lost because higher order terms become quickly very much involved. The explicit form of the series is not known, although there exist algorithms to compute it to a finite order.

Furthermore, a compact formula exists that gives the piece of the BCH series to all orders in one operator, say Y , and to first order in X , namely [73]

$$\exp(X)\exp(Y) = \exp\left(X + Y + \sum_{k=1}^{\infty} (-1)^k \frac{B_k}{k!} \overbrace{[Y, [\dots [Y, X]] \dots]}^{\text{k-times}} + \mathcal{O}(X^2)\right), \quad (3.46)$$

where multiple nested commutators are explicitly indicated. Here B_k are Bernoulli numbers [30].

Now, to derive the equation satisfied by $\Omega(t, t_0)$, we consider a short time interval δt after time t ; at the end, we will get $\delta t \rightarrow 0$. We use the group property (3.44) with $t_2 = t + \delta t$, $t_1 = t$ to write $U(t + \delta t, t_0) = U(t + \delta t, t)U(t, t_0)$. In exponential form it reads

$$\exp(\Omega(t + \delta t, t_0)) = \exp(\Omega(t + \delta t, t)) \exp(\Omega(t, t_0)). \quad (3.47)$$

The matrix A is assumed to take the constant value $A(t)$ during the interval $(t, t + \delta t)$. The equation (3.38) can then be solved in the exponential form $\exp(\Omega(t + \delta t, t)) \simeq \exp(A(t)\delta t)$ and so

$$\exp(\Omega(t + \delta t, t_0)) \simeq \exp(A(t)\delta t) \exp(\Omega(t, t_0)). \quad (3.48)$$

Applying (3.46) to (3.48) and keeping just the first order in δt , we get

$$\begin{aligned} \Omega(t + \delta t, t_0) &= \Omega(t, t_0) \\ &+ \delta t \left\{ A(t) + \sum_{k=1}^{\infty} (-1)^k \frac{B_k}{k!} \overbrace{[\Omega(t, t_0), [\dots [\Omega(t, t_0), A(t)]] \dots]}^{\text{k-times}} \right\} + \mathcal{O}(\delta t^2). \end{aligned} \quad (3.49)$$

In the limit $\delta t \rightarrow 0$, this yields the exact result

$$\frac{\partial}{\partial t} \Omega(t, t_0) = A(t) + \sum_{k=1}^{\infty} (-1)^k \frac{B_k}{k!} \overbrace{[\Omega(t, t_0), [\dots [\Omega(t, t_0), A(t)]] \dots]}^{\text{k-times}}, \quad \Omega(t_0, t_0) = 0, \quad (3.50)$$

which is a highly nonlinear differential equation for Ω , where the firsts terms of this equations are

$$\Omega'(t, t_0) = A(t) - \frac{1}{2} [\Omega(t, t_0), A(t)] + \frac{1}{12} [\Omega(t, t_0), [\Omega(t, t_0), A(t)]] + \dots, \quad (3.51)$$

By defining

$$\Omega^{[0]}(t, t_0) = 0, \quad \Omega^{[1]}(t, t_0) = \int_0^t A(t_1) dt_1, \quad (3.52)$$

and applying Picard fixed point iteration, one gets

$$\Omega^{[n]}(t, t_0) = \int_0^t \left(A(t_1) - \frac{1}{2}[\Omega^{[n-1]}(t_1, t_0), A(t_1)] + \frac{1}{12}[\Omega^{[n-1]}(t_1, t_0), [\Omega^{[n-1]}(t_1, t_0), A(t_1)]] + \dots \right) dt_1, \quad (3.53)$$

and $\lim_{n \rightarrow \infty} \Omega^{[n]}(t, t_0) = \Omega(t, t_0)$ in a suitably small neighbourhood of the origin.

3.4.1.2 First terms in Magnus expansion

Suppose now that $A(t)$ is of first order in some parameter ε and try a solution in the form of a series

$$\Omega(t, t_0) = \sum_{n=1}^{\infty} \Omega_n(t, t_0), \quad (3.54)$$

where $\Omega_n(t, t_0)$ is supposed to be of order ε^n . Equivalently, it can be replaced $A(t)$ by $\varepsilon A(t)$ in (3.38) and determine the successive terms of

$$\Omega(t, t_0) = \sum_{n=1}^{\infty} \varepsilon^n \Omega_n(t, t_0). \quad (3.55)$$

This can be done explicitly, at least for the first terms, by substituting the series (3.55) at equation (3.51) and equating powers of ε . Obviously, the Magnus series (3.54) is recovered by taking $\varepsilon = 1$. Thus, using the notation $A(t_i) \equiv A_i$, the first two orders read

1. $\Omega'_1(t, t_0) = A(t)$, so that

$$\Omega_1(t, t_0) = \int_0^t dt_1 A_1. \quad (3.56)$$

2. $\Omega'_2(t, t_0) = -\frac{1}{2}[\Omega_1(t, t_0), A(t)]$. Thus

$$\Omega_2(t, t_0) = \frac{1}{2} \int_0^t dt_1 \int_0^{t_1} dt_2 [A_1, A_2]. \quad (3.57)$$

3.4.1.3 Magnus integrators for linear systems

If time is discretized with an equally spaced mesh, $t_{n+1} = t_n + h$, then $A(t)$ can be expanded around the midpoint, $t_{n+1/2} = t_n + h/2$, of the subinterval $[t_n, t_{n+1}]$,

$$A(t) = \sum_{j=0}^{\infty} a_j (t - t_{n+1/2})^j, \quad \text{where} \quad a_j = \frac{1}{j!} \frac{d^j A}{dt^j}(t_{n+1/2}) \quad (3.58)$$

and insert the series (3.58) into the different terms (3.56) and (3.57) defining the Magnus expansion. In this way, one gets explicitly, the expression of $\Omega_k(t_{n+1}, t_n)$, for example, up to order h^4 as

$$\begin{aligned}\Omega_1(t_{n+1}, t_n) &= ha_0 + h^3 \frac{1}{12} a_2 + \mathcal{O}(h^5), \\ \Omega_2(t_{n+1}, t_n) &= h^3 \frac{-1}{12} [a_0, a_1] + \mathcal{O}(h^5),\end{aligned}\quad (3.59)$$

where $\Omega_3(t_{n+1}, t_n) = \mathcal{O}(h^5)$, $\Omega_4(t_{n+1}, t_n) = \mathcal{O}(h^5)$, $\Omega_5(t_{n+1}, t_n) = \mathcal{O}(h^7)$.

Let us denote $\alpha_i \equiv h^i a_{i-1}$. Then $[\alpha_{i_1}, \alpha_{i_2}]$ is an element of order $h^{i_1+i_2}$. It is possible to build methods of order $p \equiv 2s$ by considering only terms involving $\alpha_1, \dots, \alpha_s$ in Ω [72]. These terms can be approximated by linear combinations of the matrix $A(t)$ evaluated at different points. In particular, up to order two or up to order four, we have to approximate

$$\begin{aligned}\Omega(t_{n+1}, t_n) &= \alpha_1 + \mathcal{O}(h^3) \\ &= \alpha_1 + \frac{1}{12} (\alpha_3 - [\alpha_1, \alpha_2]) + \mathcal{O}(h^5).\end{aligned}\quad (3.60)$$

We introduce the averaged generalised momentum matrices

$$A^{(i)}(h) \equiv \frac{1}{h^i} \int_{t_n}^{t_n+h} (t - t_{n+1/2})^i A(t) dt = \frac{1}{h^i} \int_{-h/2}^{h/2} t^i A(t + t_{n+1/2}) dt \quad (3.61)$$

for $i = 0, \dots, s-1$. If their exact evaluation is not possible, or is computationally expensive, a numerical quadrature rule is used as, for example, the Gauss-Legendre quadratures of order p ,

$$A^{(0)}(h) = \int_{t_n}^{t_n+h} A(t) dt = h \sum_{j=1}^k b_j A_j + \mathcal{O}(h^{p+1}), \quad (3.62)$$

with $A_i \equiv A(t_n + c_i h)$, where b_j are the Gauss-Legendre weights and c_j the corresponding collocation points at the interval $[0, 1]$. Then it is possible to approximate all the integrals $A^{(i)}$ (up to the required order) by using only the evaluations A_i at the nodes c_i of the quadrature rule required to compute $A^{(0)}$. Specifically,

$$A^{(i)} = h \sum_{j=1}^k b_j \left(c_j - \frac{1}{2} \right)^i A_j, \quad i = 0, \dots, s-1, \quad (3.63)$$

or equivalently, $A^{(i)} = h \sum_{j=1}^k \left(Q_G^{(x,k)} \right)_{ij} A_j$ with $\left(Q_G^{(x,k)} \right)_{ij} = b_j \left(c_j - \frac{1}{2} \right)^i$. In particular, if second and fourth order Gauss-Legendre quadrature rules are considered, then for $s = k = 1$ we have

$$b_1 = 1, \quad c_1 = \frac{1}{2}, \quad (3.64)$$

and

$$Q_G^{(1,1)} = 1. \tag{3.65}$$

For fourth order, where $s = k = 2$,

$$b_1 = b_2 = \frac{1}{2}, \quad c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, \quad c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}, \tag{3.66}$$

so that

$$Q_G^{(2,2)} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{\sqrt{3}}{12} & \frac{\sqrt{3}}{12} \end{pmatrix}. \tag{3.67}$$

Furthermore, using expression (3.58) in terms of α_i up to order $s - 1$

$$\begin{aligned} A(t) &= \sum_{j=0}^{s-1} a_j (t - t_{n+1/2})^j = \sum_{j=0}^{s-1} \frac{1}{h^{j+1}} \alpha_{j+1} (t - t_{n+1/2})^j \\ &= \sum_{j=1}^s \frac{1}{h^j} \alpha_j (t - t_{n+1/2})^{j-1}, \end{aligned} \tag{3.68}$$

at equation (3.61) it is obtained

$$\begin{aligned} A^{(i)}(h) &\equiv \frac{1}{h^i} \int_{-h/2}^{h/2} t^i A(t + t_{n+1/2}) dt \\ &= \frac{1}{h^i} \int_{-h/2}^{h/2} t^i \left(\sum_{j=1}^{p+1} \frac{1}{h^j} \alpha_j t^{j-1} \right) dt \\ &= \sum_{j=1}^s \left(\frac{1}{h^{i+j}} \int_{-h/2}^{h/2} \alpha_j t^{i+j-1} dt \right) \\ &= \sum_{j=1}^s \alpha_j \left(\int_{-h/2}^{h/2} \frac{t^{i+j-1}}{h^{i+j}} dt \right) \\ &= \sum_{j=1}^s \alpha_j \frac{1 - (-1)^{i+j}}{(i+j)2^{i+j}}, \end{aligned} \tag{3.69}$$

or in an equivalent compact form defining a matrix $T^{(s)}$ as follows

$$A^{(i)} = \sum_{j=1}^s \left(T^{(s)} \right)_{ij} \alpha_j \equiv \sum_{j=1}^s \frac{1 - (-1)^{i+j}}{(i+j)2^{i+j}} \alpha_j, \quad 0 \leq i \leq s - 1. \tag{3.70}$$

If this relation is inverted (to order two $s = 1$) one has

$$R^{(1)} = \left(T^{(1)} \right)^{-1} = (1), \tag{3.71}$$

and to order four $s = 2$ it is obtained

$$R^{(2)} = \left(T^{(2)}\right)^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 12 \end{pmatrix}, \quad (3.72)$$

so that the corresponding expression of α_i in terms of $A^{(i)}$ or A_j is given by

$$\alpha_i = \sum_{j=1}^s \left(R^{(s)}\right)_{ij} A^{(j-1)} = \sum_{j=1}^s \left(R^{(s)} Q_G^{(s,k)}\right)_{ij} A_j \quad (3.73)$$

3.4.1.4 Commutator free Magnus expansion

We have seen that the second order method $e^{\Omega^{[2]}}$

$$y_{n+1} = \exp(\Omega^{[2]}(h))y_n + \mathcal{O}(h^3) = \exp(\alpha_1)y_n + \mathcal{O}(h^3), \quad (3.74)$$

does not contain commutators, so the expression for the method using the Gauss-Legendre quadrature rule and equation (3.73) to express α_i in terms of matrix evaluations is obtained as

$$y_{n+1} = \exp(hA(t_{n+1/2}))y_n + \mathcal{O}(h^3). \quad (3.75)$$

The fourth order method $e^{\Omega^{[4]}}$ containing commutators corresponds to

$$y_{n+1} = \exp(\Omega^{[4]}(h))y_n + \mathcal{O}(h^5) = \exp\left(\alpha_1 - \frac{1}{12}[\alpha_1, \alpha_2]\right)y_n + \mathcal{O}(h^5). \quad (3.76)$$

Alternatively, approximations up to the same order can be obtained by a product of exponentials of linear combinations of the α_i which avoid the presence of commutators [71]. For example, we can consider

$$\Psi^4 \equiv \prod_{k=1}^2 \exp(x_{k,1}\alpha_1 + x_{k,2}\alpha_2) = \exp(\Omega^{[4]}(h)) + \mathcal{O}(h^5), \quad (3.77)$$

where the coefficients $x_{k,1}$, $x_{k,2}$ have to be determined. To work with the Lie algebra generated by the $A^{(i)}$ s (or the A_i s) is equivalent to work with the Lie algebra generated by the α_i . However, in the last case the problem simplifies considerably since the number of terms of the Lie algebra is reduced. To get fourth-order integrators it suffices to consider α_1 and α_2 [71]. Then, the problem reduces to solve the equation

$$\Psi^4 \equiv \prod_{k=1}^2 \exp(x_{k,1}\alpha_1 + x_{k,2}\alpha_2) = \exp\left(\alpha_1 - \frac{1}{12}[\alpha_1, \alpha_2]\right) + \mathcal{O}(h^5). \quad (3.78)$$

We can use the Baker-Campbell-Hausdorff (BCH) formula (3.45)

$$\exp(X) \exp(Y) = \exp \left(X + Y + \frac{1}{2}[X, Y] + \frac{1}{12}([X, [X, Y]] + [Y, [Y, X]]) + \dots \right) \quad (3.79)$$

Notice that the solution is time symmetric and only odd terms appear in the equation. Then those terms at even orders up to the order of the method have to be cancelled with a proper choice of the parameters $x_{k,j}$. An integrator, ψ_h , is time symmetric if $\psi_{-h}^{-1} = \psi_h$. Then the following symmetry for the coefficients

$$x_{m+1-k,j} = (-1)^{j+1} x_{k,j}, \quad j = 1, 2 \quad (3.80)$$

makes the schemes time symmetric and all even order terms are cancelled.

At this point we have to look for the coefficients x_1 and x_2 fulfilling the equation

$$\Psi^{[4]} \equiv \exp(x_1\alpha_1 + x_2\alpha_2) \exp(x_1\alpha_1 - x_2\alpha_2) = \exp\left(\alpha_1 - \frac{1}{12}[\alpha_1, \alpha_2]\right) + \mathcal{O}(h^5), \quad (3.81)$$

and, after some calculations

$$\begin{aligned} \Psi^{[4]} &\equiv \exp(x_1\alpha_1 + x_2\alpha_2) \exp(x_1\alpha_1 - x_2\alpha_2) \\ &= \exp((x_1\alpha_1 + x_2\alpha_2) + (x_1\alpha_1 - x_2\alpha_2) + [(x_1\alpha_1 + x_2\alpha_2), (x_1\alpha_1 - x_2\alpha_2)]) + \mathcal{O}(h^5) \\ &= \exp(2x_1\alpha_1 - x_1x_2[\alpha_1, \alpha_2]) + \mathcal{O}(h^5) \\ &= \exp\left(\alpha_1 - \frac{1}{12}[\alpha_1, \alpha_2]\right) + \mathcal{O}(h^5) \end{aligned} \quad (3.82)$$

the following restrictions for the coefficients are obtained

$$\left. \begin{array}{l} 2x_1 = 1 \\ -x_1x_2 = -\frac{1}{12} \end{array} \right\} \Rightarrow \left. \begin{array}{l} x_1 = \frac{1}{2} \\ x_2 = \frac{1}{6} \end{array} \right\} . \quad (3.83)$$

With these coefficients, the solution at time t_{n+1} can be achieved from the solution at time t_n by means of

$$y(t_{n+1}) = \exp\left(\frac{1}{2}\alpha_1 + \frac{1}{6}\alpha_2\right) \exp\left(\frac{1}{2}\alpha_1 - \frac{1}{6}\alpha_2\right) y(t_n), \quad (3.84)$$

or, in terms of $A_1 = A(t_n + (1/2 - \sqrt{3}/6)h)$ and $A_2 = A(t_n + (1/2 + \sqrt{3}/6)h)$, using (3.73)

$$y(t_{n+1}) = \exp(a_1hA_1 + a_2hA_2) \exp(a_2hA_1 + a_1hA_2) y(t_n), \quad (3.85)$$

with $a_1 = \frac{3-2\sqrt{3}}{12}$, $a_2 = \frac{3+2\sqrt{3}}{12}$.

3.4.2 Implementation of the product matrix exponential - vector

Different ways to approximate the calculation for the matrix exponential for large and sparse matrix systems are compared. One of them consists on the use of a Krylov subspace to project our operator and to exponentiate the projected matrix obtained.

The other two ways to approximate the exponential matrix are based on a polynomial approximation of the exponential function in a interval containing the eigenvalues of the corresponding matrix, and then use the matrix polynomial to approximate the matrix exponential. It can be done either using a Newton interpolation of the exponential functions at a sequence of Leja points on the real focal interval of a family of confocal ellipses in the complex plane, or using Chebyshev polynomials approximating the exponential on the interval containing the relevant eigenvalues of the matrix. We briefly describe these methods.

3.4.2.1 Krylov methods

For $A \in \mathbb{R}^{n \times n}$ and $v \in \mathbb{R}^n$, the Arnoldi process [48] yields, after k steps, vectors $v_1, \dots, v_{k+1} \in \mathbb{R}^n$ that are orthonormal and span the Krylov subspace

$$\mathcal{K}_k(v, Av, \dots, A^{k-1}v). \quad (3.86)$$

With the basis vectors v_i , the Arnoldi process delivers an upper-Hessenberg matrix $\bar{H}_k \in \mathbb{R}^{(k+1) \times k}$ such that the following relation holds

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T, \quad (3.87)$$

where $V_k \in \mathbb{R}^{n \times k}$ has columns v_1, \dots, v_k , $H_k \in \mathbb{R}^{k \times k}$ is the matrix \bar{H}_k without the last row $(0, \dots, 0, h_{k+1,k})$, and $e_k = (0, \dots, 0, 1)^T \in \mathbb{R}^k$. The first basis vector v_1 is the normalized vector v : $v_1 = v/||v||$.

Based on $V_k^T AV_k = H_k$, the matrix exponential can be approximated by [76, 67]

$$e^A v \approx \beta V_k e^{H_k} e_1, \quad (3.88)$$

or by its corrected scheme

$$e^A v \approx \beta V_{k+1} e^{\bar{H}_k} e_1 \quad (3.89)$$

where

$$\bar{H}_k = \begin{pmatrix} H_k & 0 \\ h_{k+1,k} e_k^T & 0 \end{pmatrix} \quad (3.90)$$

3.4.2.2 Chebyshev polynomials

A well known method to approximate $e^{hA}v$ is based on the Chebyshev polynomial expansion (see for instance [69, 68]).

Here, to interpolate a function $f(x)$ over the interval $[-1, 1]$ the truncated formula

$$f(x) \approx \sum_{i=1}^k c_i T_i(x) + \frac{c_0}{2} I, \quad (3.91)$$

is used, where $T_i(x)$ are the Chebyshev polynomials of the first kind defined at $[-1, 1]$, which can be computed by the Chebyshev recursion

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x), \quad i = 1, 2, \dots, \quad (3.92)$$

and the coefficients c_i can be computed, for a large M , as

$$c_i = \frac{2}{M} \sum_{j=1}^M f(\cos(\theta_j)) \cos(i\theta_j), \quad k = 0, 1, \dots, m, \quad \theta_j = \frac{\pi(j - \frac{1}{2})}{M}, \quad (3.93)$$

which means interpolating $f(x)$ at the Chebyshev polynomial roots, where this Chebyshev approximation have been previously used at [77, 70].

Assuming matrix hA has the eigenvalues in $[ha, hb]$, to interpolate the function $\exp(x)$ over this interval is equivalent to interpolate over the interval $[-1, 1]$ the function $f(x)$ defined as follows

$$f(x) := \exp(x\gamma + c), \quad (3.94)$$

where $\gamma = h(b - a)/2$ and $c = h(a + b)/2$.

Finally, defining $B = (hA - c)/\gamma$ (with the eigenvalues of B in $[-1, 1]$) we approximate the action of the exponential matrix, $e^{hA} = e^{B\gamma+c}$ ($e^{tB\gamma-c}$ with $t = 1$), over a vector as

$$e^{tB\gamma+c}v \approx \left[\sum_{i=1}^k c_i T_i(tB) + \frac{c_0}{2} I \right] v, \quad (3.95)$$

where the coefficients c_i are approximated by means of

$$c_i = \frac{2}{M} \sum_{j=1}^M \exp(\cos(\theta_j)\gamma + c) \cos(i\theta_j), \quad k = 0, 1, \dots, m, \quad \theta_j = \frac{\pi(j - \frac{1}{2})}{M}. \quad (3.96)$$

As a stopping criterion for the truncated series we approximate the residual as in [70]. Thus, there will be used the well known relations for the Chebyshev

polynomials of first kind, $T_k(x)$, and second kind, $U_k(x)$,

$$\begin{aligned} T'_i(x) &= iU_{i-1}(x), \\ xT_i(x) &= \frac{1}{2}(T_{i+1}(x) + T_{i-1}(x)), \\ xU_i(x) &= \frac{1}{2}(U_{i+1}(x) + U_{i-1}(x)), \\ T_i(x) &= \frac{1}{2}(U_i(x) - U_{i-2}(x)), \end{aligned} \quad (3.97)$$

where the Chebyshev polynomials of second kind are defined by means of the recurrence formula

$$U_0(x) = 1, \quad U_1(x) = 2x, \quad U_{i+1}(x) = 2xU_i(x) - U_{i-1}(x), \quad i = 1, 2, \dots, \quad (3.98)$$

where it is defined $U_{-1}(x) = 0$ and $U_{-2}(x) = -1$. In terms of the Chebyshev polynomials of second kind, $U_i(x)$, the formula (3.95) for the solution is rewritten as

$$\begin{aligned} y_k(t) &= \left[\sum_{i=1}^k c_i T_i(tB) + \frac{c_0}{2} I \right] v \\ &= y_{k-1}(t) + \frac{c_k}{2} (U_k(tB)v - U_{k-2}(tB)v), \quad k = 1, 2, \dots \end{aligned} \quad (3.99)$$

The recurrence to calculate $U_k(tB)v$ is written as

$$U_{-2}(tB)v = -v, \quad U_{-1}(tB)v = 0, \quad U_0(tB)v = v, \quad (3.100)$$

$$U_1(tB)v = 2tBv = \frac{2th}{\gamma}Av - \frac{2tc}{\gamma}v, \quad (3.101)$$

$$U_{i+1}(tB)v = 2tBU_i(tB)v - U_{i-1}(tB)v \quad (3.102)$$

$$= \frac{2th}{\gamma}AU_i(tB)v - \frac{2tc}{\gamma}U_i(tB)v - U_{i-1}(tB)v, \quad i = 1, 2, \dots \quad (3.103)$$

Because $\exp(hA)v = \exp(B\gamma + c)v$ is the exact solution $y(t) = \exp(tB\gamma + c)v$ at $t = 1$ of the initial value problem

$$y'(t) = \gamma B y(t), \quad y(0) = v, \quad (3.104)$$

the residual $r_k(t) := \gamma B y_k(t) - y'_k(t)$ can be approximated substituting the expansion (3.95) at equation (3.104) and using the relations (3.97), obtaining

$$\begin{aligned} y'_k(t) &= \left[\sum_{i=1}^k \frac{c_i}{t} (tB) T'_i(tB) \right] v \\ &= \left[\sum_{i=1}^k \frac{ic_i}{2t} (U_i(tB) + U_{i-2}(tB)) \right] v, \\ &= y'_{k-1}(t) + \frac{kc_k}{2t} (U_k(tB)v + U_{k-2}(tB)v), \quad k = 1, 2, \dots, \end{aligned} \quad (3.105)$$

and

$$\begin{aligned}
\gamma B y_k(t) &= \gamma \left[\sum_{i=1}^k \frac{c_i}{t} (tB) T_i(tB) + \frac{c_0}{2} B \right] v \\
&= \gamma \left[\sum_{i=1}^k \frac{c_i}{2t} (T_{i+1}(tB) + T_{i-1}(tB)) + \frac{c_0}{2} B \right] v \\
&= \gamma \left[\sum_{i=1}^k \frac{c_i}{4t} (U_{i+1}(tB) - U_{i-3}(tB)) + \frac{c_0}{2} B \right] v \\
&= \gamma B y_{k-1}(t) + \frac{\gamma c_k}{4t} (U_{k+1}(tB)v - U_{k-3}(tB)v), \quad k = 1, 2, \dots,
\end{aligned} \tag{3.106}$$

thus, the formula for the residual is

$$\begin{aligned}
r_k(t) &= \gamma B y_k(t) - y'_k(t) \\
&= r_{k-1}(t) + \frac{\gamma c_k}{4t} (U_{k+1}(tB)v - U_{k-3}(tB)v) \\
&\quad - \frac{k c_k}{2t} (U_k(tB)v + U_{k-2}(tB)v), \quad k = 1, 2, \dots
\end{aligned} \tag{3.107}$$

In terms of the Chebyshev polynomials of second kind, $U_i(x)$, and using the same polynomials, the formula for the solution is

$$\begin{aligned}
y_k(t) &= \left[\sum_{i=1}^k c_i T_i(tB) + \frac{c_0}{2} I \right] v \\
&= y_{k-1}(t) + \frac{c_k}{2} (U_k(tB)v - U_{k-2}(tB)v), \quad k = 1, 2, \dots
\end{aligned} \tag{3.108}$$

The algorithm to calculate the exponential matrix by means of a Chebyshev expansion polynomials of second kind with the residual estimation for the error described before is scheduled at Algorithm 1.

3.4.2.3 Real Leja points method

The ReLPM (Real Leja Points Method) introduced in [78] has shown very attractive computational features. It rests on Newton interpolation of the exponential functions at a sequence of Leja points on the real focal interval of a family of confocal ellipses in the complex plane. The use of Leja points is suggested by the fact that they guarantee superlinear convergence of the corresponding matrix polynomials to the matrix exponential functions. A key step in the approximation procedure, as in Chebyshev polynomial approximation, is given by estimating cheaply a real focal interval, say $[a, b]$, such that the minimal ellipse which contains the spectrum of the matrix is not too “large” (numerical experience has

Algorithm 1 Chebyshev expansion algorithm for the matrix exponential**Require:** A, v, h, ε, t **Ensure:** $y = \exp(thA)$;

```

1:  $u_{-2} := -v, \quad u_{-1} := 0, \quad u_0 := v$ 
2:  $u_1 := (2 * t * h / \gamma) * A * v - (2 * t * c / \gamma) * v$ 
3: compute  $c_0$ .
4:  $y := (c_0 / 2) * u_0, \quad r := (\gamma * c_0 / 4 / t) * u_1$ 
5: for  $k = 1, \dots, N_{\max}$  do
6:    $u_2 := (2 * t * h / \gamma) * A * u_1 - (2 * t * c / \gamma) * u_1 - u_0$ 
7:   compute  $c_k$ .
8:    $y := y + (c_k / 2) * (u_1 - u_{-1})$ 
9:    $r := r + (\gamma * c_k / 4 / t) * (u_2 - u_{-2}) - (k * c_k / 2 / t) * (u_1 + u_{-1})$ 
10:   $u_{-2} := u_{-1}$ 
11:   $u_{-1} := u_0$ 
12:   $u_0 := u_1$ 
13:   $u_1 := u_2$ 
14:  if  $\|r\| < \varepsilon$  then
15:    break
16:  end if
17: end for

```

shown that good results can be obtained at a very low cost, simply by intersecting the Gershgorin's circles).

Thus, the method is based on interpolating $\varphi(x) := \exp(x-1)/x$ at Leja points of the interval $[a, b] = [c - 2\gamma, c + 2\gamma]$. Observe that once $\varphi(\tau A)v$ is computed, then $\exp(\tau A)v = \tau A\varphi(\tau A)v + v$. In practice, it is numerically convenient to interpolate the function $\varphi(\tau(c + \gamma\xi))$ at Leja points $\{\xi_s\}$ of a reference interval $[-2, 2]$. Then, given the corresponding divided differences $\{d_i\}$ for such a function which can be calculated as in [79], the matrix Newton polynomial of degree k is

$$p_k(A) = \sum_{i=0}^k d_i \Omega_i \approx \varphi(hA), \quad \Omega_i = \prod_{s=0}^i -1((A - cI)/\gamma - \xi_s I). \quad (3.109)$$

In general, it is not feasible to interpolate with the original time step Δt , which has to be split. This happens, for example, when the expected degree for convergence is too large. The ReLPM code subdivides dynamically Δt into smaller substeps $h = h_k$, and recovers the required vector $\varphi(\delta t A)v$ according to the time marching scheme

$$y_{k+1} = y_k + h_k \varphi(h_k A)(A y_k + v), \quad k = 1, \dots, k^*; y_0 = 0; \quad (3.110)$$

where $\sum_{k=1}^{k^*} h_k = \Delta t$.

3.4.3 Numerical Results

To test the exponential integrators we consider the transient for the small reactor of type VVER (see Figure 3.20 described at Section 3.1.3.1). The method is tested

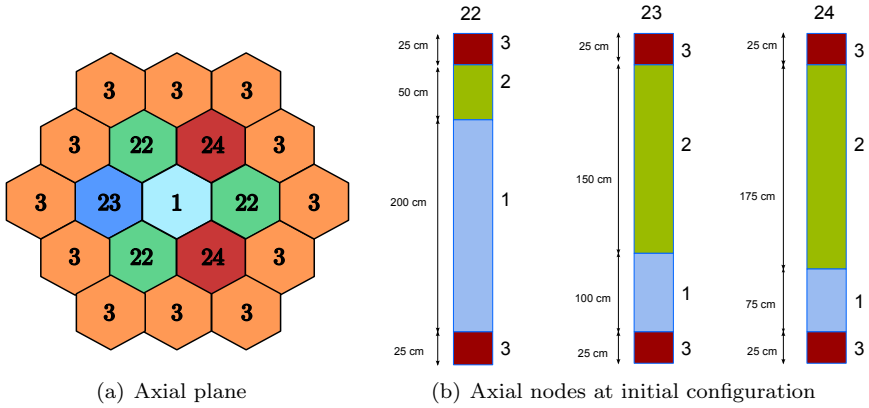


Figure 3.20: Small reactor.

with the first part of the transient (from $t = 0.0s$ to $t = 0.15s$), where two different time steps are used.

Tables 3.11 and 3.12 show the iterations count needed to integrate the system until a stopping criterion is satisfied. The Krylov method uses different sizes of the Krylov subspace $m = 10, 20, 30, 40, 50, 60$. Depending on the method, the value for the stopping criterion is chosen in a different way to ensure fair comparisons, imposing this value as the largest one that ensures convergence respect to a reference solution.

Table 3.11: Iterations count for the second order scheme with different exponential approximations

Method	$\Delta t = 6.25E-3$	$\Delta t = 1.5625E-3$
Krylov $m = 10$	12274	12366
Krylov $m = 20$	6144	6516
Krylov $m = 30$	4374	6366
Krylov $m = 40$	4184	7776
Krylov $m = 50$	3974	9696
Krylov $m = 60$	3974	9696
Chebyshev	10372	19148
Real Leja Points	6168	6611

We observe that the Real Leja Points approximation offers good number of iterations when compared with Krylov methods, and that both them are much faster than Chebyshev method for the chosen problem. The Real Leja Points method has the advantage that it is not necessary to save more than six vectors for the recurrence, having less memory requirements. Also, no extra operations are needed as, for example, the computation of the exponential of the small Hessenberg matrix used by the Krylov approximation.

Table 3.12: Iterations count for the fourth order scheme with different exponential approximations

Method	$\Delta t = 1.25\text{E-}2$	$\Delta t = 3.125\text{E-}3$
Krylov $m = 10$	11944	12826
Krylov $m = 20$	6064	7256
Krylov $m = 30$	4524	6966
Krylov $m = 40$	4304	7776
Krylov $m = 50$	4224	9696
Krylov $m = 60$	4284	11376
Chebyshev	10372	19148
Real Leja Points	6797	7151

It can also be observed that the fourth order method does not improve the results from the second order method, obtaining similar iteration counts with the double of the time step. Nevertheless, it is expected that for slower transients the results of fourth order methods can take advantage to second order methods using smaller time steps.

Chapter 4

Updating Eigenvalue Methods

Contents

4.1 Modified Block Newton Method	128
4.1.1 Spatial discretization	129
4.1.2 Modified Block Newton Method	133
4.1.3 Numerical Results	135
4.2 Alternative Newton Methods	146
4.2.1 Modified Block Newton Method	147
4.2.2 One Sided Block Newton Method	148
4.2.3 Two Sided Block Newton Method	148
4.2.4 Numerical Results	149
4.3 Proper Generalized Decomposition for eigenvalue computations	153
4.3.1 Eigenvalue problem	157
4.3.2 Numerical Results	161

Starting from a steady state configuration of a nuclear power reactor some situations arise in which the reactor configuration is perturbed. The Lambda modes are eigenfunctions associated with a given configuration of the reactor, which have successfully been used to describe unstable events in BWRs with a modal method [3], which uses a set of dominant modes to expand the neutron flux. For the transient calculations using the modal method with a moderate number of modes, these modes must be updated each time step to maintain the accuracy of the solution. The updating modes process is also interesting to study perturbed configurations of a reactor.

To compute several eigenvalues and its corresponding eigenfunctions for a nuclear reactor is quite expensive from the computational point of view. The Arnoldi

method, based on approximating the invariant subspaces of the original problem projected onto a Krylov subspace is an efficient method to compute the dominant Lambda modes associated with a given configuration of the reactor, but if the Lambda modes have to be computed for different perturbed configurations of the reactor more efficient methods can be used.

In this chapter, different methods for updating the Lambda modes for a nuclear reactor core are proposed. The first one is based on a Newton strategy, which increases the convergence order respect to Arnoldi method, but it depends on the initial guess to obtain a good performance. Finally, a method based on a Proper Generalized Decomposition is proposed, where a reduced order model of the initial problem is solved and improved iteratively. This method can also be used to update a given invariant subspace by means of low rank modifications of an initial guess.

4.1 Modified Block Newton Method

In the BWR and PWR reactors the fuel elements are square prisms and in the VVER reactors the fuel elements are hexagonal prisms. For this reason, it is interesting to develop efficient methods for both rectangular and hexagonal meshes. As it has been already mentioned, under general assumptions, the neutronic population inside a nuclear power reactor can be modeled by the time dependent neutron diffusion equation in the approximation of two energy groups. This model is of the form [8],

$$\begin{aligned} [v^{-1}] \frac{\partial \Phi}{\partial t} + \mathcal{L}\Phi &= (1 - \beta)\mathcal{M}\Phi + \sum_{k=1}^K \lambda_k \chi \mathcal{C}_k, \\ \frac{\partial \mathcal{C}_k}{\partial t} &= \beta_k [\nu \Sigma_{f1} \quad \nu \Sigma_{f2}] \Phi - \lambda_k \mathcal{C}_k, \quad k = 1, \dots, K, \end{aligned} \quad (4.1)$$

where, K is the number of delayed neutron precursor groups considered,

$$\mathcal{L} = \begin{bmatrix} -\vec{\nabla} \cdot (D_1 \vec{\nabla}) + \Sigma_{a1} + \Sigma_{12} & 0 \\ -\Sigma_{12} & -\vec{\nabla} \cdot (D_2 \vec{\nabla}) + \Sigma_{a2} \end{bmatrix}, \quad [v^{-1}] = \begin{bmatrix} v_1^{-1} & 0 \\ 0 & v_2^{-1} \end{bmatrix},$$

and

$$\mathcal{M} = \begin{bmatrix} \nu \Sigma_{f1} & \nu \Sigma_{f2} \\ 0 & 0 \end{bmatrix}, \quad \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}, \quad \chi = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The diffusion constants and cross-sections, D_g , Σ_{12} , Σ_{ag} , $\nu \Sigma_{fg}$, $g = 1, 2$, appearing in these equations depending on the reactor materials, that is, they are position and time dependent functions, as explained at the introduction. Associated with this problem, there is the Lambda Modes problem,

$$\mathcal{L}\Phi_i = \frac{1}{k_i} \mathcal{M}\Phi_i. \quad (4.2)$$

To solve both problems (4.1) and (4.2), a spatial discretization of the equations has to be selected. Once this discretization has been selected, the semidiscrete version of the time dependent neutron diffusion equation is solved. Since the ordinary differential equations resulting from the discretization of diffusion equations are, in general, stiff, implicit methods are necessary. With the aim of reducing the computational cost of implicit methods, Modal Methods have been used [3], based on expanding the neutron flux in terms of the dominant Lambda modes of the reactor core.

Starting from a steady state configuration of a nuclear power reactor some situations arise in which the reactor configuration is perturbed. To compute several eigenvalues and their corresponding eigenfunctions for the new configuration of the reactor is quite expensive from the computational point of view. Krylov subspace methods are efficient methods to compute the dominant Lambda modes associated with a given configuration of the reactor, but when the Lambda modes have to be computed from previous perturbed configurations of the reactor other kind of methods that use the information provided by the computed modes in previous steps can be more convenient. In this line, we have studied a Modified Block Newton Method (MBNM) to speed-up the calculations in the updating modes process.

4.1.1 Spatial discretization

The spatial mesh used to discretize a reactor core is naturally defined by the different compositions of the materials present in the core. Thus, we use a coarse mesh adapted to the fuel bundles composing the core and different strategies are used for the spatial discretization of the equations depending on the core geometry.

4.1.1.1 PWR and BWR reactors

In PWR and BWR reactors the fuel assemblies are rectangular prisms. Thus, a rectangular mesh is suitable to describe these prisms. Each one of these prisms is divided into different nodes and, each node, e , is mapped onto a reference domain by means of a change of variables from the real space (x, y, z) to the variables of the reference domain (x', y', z') . Then the neutron flux into each node e is expanded in terms of the orthonormal Legendre polynomials as follows [11],

$$\Phi_e(x', y', z') = \sum_{i=0}^K \sum_{j=0}^K \sum_{k=0}^K \psi_{e,ijk} P_i(x') P_j(y') P_k(z'). \quad (4.3)$$

Continuity over the inner interfaces of the elements for the neutron flux, continuity for the neutron current over the normal directions of these interfaces and, boundary conditions over the external boundary of the reactor are ensured by means of imposing them explicitly, and then, some algebraic manipulations are used to

introduce these conditions into weighted equations for each face of the nodes obtaining a set of algebraic equations that constitute the approximation known as the Nodal Collocation Method for the Lambda modes problem [11].

4.1.1.2 VVER Reactors

The fuel assemblies in VVER reactors are hexagonal prisms. Thus, different strategies from the ones used for rectangular reactors have to be considered. The strategy used at the previous chapter is used to discretize the spatial part of the equations, which is based on expand the neutronic flux in terms of the modified Dubiner's polynomials [36].

4.1.1.3 Algebraic Problem

After performing the spatial discretization, the problem (4.1), it can be approximated by the following semidiscrete system of equations

$$\begin{aligned} [v^{-1}] \dot{\psi} + L\psi &= (1 - \beta)M\psi + \sum_{k=1}^K \lambda_k X C_k, \\ X \dot{C}_k &= \beta_k M\psi - \lambda_k X C_k, \end{aligned} \quad (4.4)$$

where matrices L , M and X have the following block structure

$$L = \begin{bmatrix} L_{11} & 0 \\ -L_{21} & L_{22} \end{bmatrix}, \quad M = \begin{bmatrix} M_{11} & M_{12} \\ 0 & 0 \end{bmatrix},$$

and

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

where P is the mass matrix for the spatial discretization [36], as explained at Chapter 3. For the Lambda modes problem (4.2) we have

$$L\psi_l = \frac{1}{k_l} M\psi_l. \quad (4.5)$$

As the matrix L is nonsymmetric, we also consider the adjoint problem

$$L^\dagger \psi_l^\dagger = \frac{1}{k_l} M^\dagger \psi_l^\dagger, \quad (4.6)$$

since the eigenvectors solutions of problems (4.5) and (4.6) satisfy the following biorthogonality relationship

$$\langle \psi_m^\dagger, M\psi_n \rangle = \langle \psi_m^\dagger, M\psi_m \rangle \delta_{n,m} = N_m \delta_{n,m}. \quad (4.7)$$

4.1.1.4 Modal method

Different methods have been proposed to solve equations (4.4). One possibility that makes use of the dominant Lambda modes associated with a given configuration of the reactor core is to use a modal method [3]. This method assumes that $\psi(t)$ can be expressed approximately as

$$\psi(t) = \sum_{l=1}^{M_d} n_l(t) \psi_l \quad , \quad (4.8)$$

where ψ_l , $l = 1, \dots, M_d$ are the dominant Lambda modes of a given configuration of the core. A small amount of the dominant Lambda modes and their corresponding adjoint modes can be efficiently computed using, for example, the Implicit Restarted Arnoldi method [5]. Multiplying equations (4.4) by ψ_m^\dagger , writing

$$L = L_0 + \delta L \quad , \quad M = M_0 + \delta M \quad ,$$

and making use of expansion (4.8), we obtain the equations

$$\begin{aligned} & \sum_{l=1}^{M_d} \langle \psi_m^\dagger, [v^{-1}] \psi_l \rangle \frac{d}{dt} n_l + \sum_{l=1}^{M_d} \frac{1}{k_l} \langle \psi_m^\dagger, M_0 \psi_l \rangle n_l \\ & + \sum_{l=1}^{M_d} \langle \psi_m^\dagger, \delta L \psi_l \rangle n_l = (1 - \beta) \sum_{l=1}^{M_d} \langle \psi_m^\dagger, M_0 \psi_l \rangle n_l \\ & + (1 - \beta) \sum_{l=1}^{M_d} \langle \psi_m^\dagger, \delta M \psi_l \rangle n_l + \sum_{k=1}^K \lambda_k \langle \psi_m^\dagger, X C_k \rangle \quad , \\ & \frac{d}{dt} \langle \psi_m^\dagger, X C_k \rangle = \beta_k \sum_{l=1}^{M_d} \langle \psi_m^\dagger, M_0 \psi_l \rangle n_l \\ & + \beta_k \sum_{l=1}^{M_d} \langle \psi_m^\dagger, \delta M \psi_l \rangle n_l - \lambda_k \langle \psi_m^\dagger, X C_k \rangle \quad . \end{aligned} \quad (4.9)$$

Using the biorthogonality relationship (4.7), introducing the notation,

$$\begin{aligned} \Lambda_{ml} &= \langle \psi_m^\dagger, [v^{-1}] \psi_l \rangle \quad , & A_{ml}^L &= \langle \psi_m^\dagger, \delta L \psi_l \rangle \quad , \\ A_{ml}^M &= \langle \psi_m^\dagger, \delta M \psi_l \rangle \quad , & C_{mk} &= \langle \psi_m^\dagger, X C_k \rangle \quad , \end{aligned}$$

and the mode m reactivity, defined as $\rho_m = (k_m - 1)/k_m$, equations (4.9) can be expressed as the following matrix equations (see reference [3] for full details)

$$\begin{aligned} \frac{d[n]}{dt} &= [\Lambda]^{-1}([\rho - \beta I][N][n] + (1 - \beta)[A^M][n] \\ &\quad - [\Lambda]^{-1}[A^L][n] + \sum_{k=1}^K \lambda_k [C_k]) , \\ \frac{d[C_k]}{dt} &= \beta_k [N][n] + \beta_k [A^M][n] - \lambda_k [C_k] , \quad k = 1, \dots, K . \end{aligned} \quad (4.10)$$

Because of the stiffness of the differential equations (4.10), to solve this system we have used a high order implicit method [80].

As initial conditions for the time integration of a transient, we start from a critical configuration of the core. To obtain this critical configuration we solve the Lambda modes problem (4.5) for a given initial configuration, searching for the fundamental mode. Dividing the fission cross-sections of the initial configuration by the fundamental eigenvalue, k_1 , we obtain

$$L_0 \psi_1 = M^{Crit} \psi_1 , \quad (4.11)$$

where M^{Crit} is a matrix whose components are the components of M divided by k_1 . Equation (4.11) together with equation

$$0 = \beta_k M^{Crit} \psi_1 - \lambda_k X C_k , \quad (4.12)$$

constitute the set of equations defining the steady state associated with the critical configuration.

For realistic transients, the nuclear cross-sections are time dependent functions and to obtain good accuracy using the modal method a large amount of modes are necessary. This is prohibitive from the computational point of view. Thus, instead of this, we use a small number of modes together with an updating modes strategy that is performed at each certain *updating time step* [3]. In this way, to update the modes it is necessary to develop an efficient strategy that uses the modes computed in the previous steps as starting initial guess to speed-up the computation. A method of this kind is the Block Newton Method, presented in next section. Also, a modes updating methodology is of interest when perturbed configurations of the reactor core are studied.

4.1.2 Modified Block Newton Method

With the spatial discretization methods exposed above, the Lambda modes equation with two groups of energy (4.2) can be approximated by an algebraic generalised eigenvalue problem with the following block structure

$$\begin{bmatrix} L_{11} & 0 \\ -L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} M_{11} & M_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}, \quad (4.13)$$

where ψ_1 is a vector with the unknowns corresponding to the neutron flux for the fast group, and ψ_2 is a vector with the unknowns corresponding to the thermal flux. To solve this problem, it is reduced to the ordinary eigenvalue problem

$$A\psi_1 = \lambda\psi_1, \quad (4.14)$$

where the matrix A in equation (4.14) is defined as

$$A = L_{11}^{-1} (M_{11} + M_{12}L_{22}^{-1}L_{21}). \quad (4.15)$$

As it has already mentioned, to update the modes along a transient or for a perturbed configuration of the core it is interesting to take into account that the space spanned by the new modes should be close to the space spanned by the previous ones. For this reason, a Modified Block Newton Method [81], which has a high order of local convergence, is applied to update the modes using as initial guess the modes obtained in a previous step. In the following, we expose the main ideas of this method.

Given a partial eigenvalue problem of the form

$$AV = V\Lambda, \quad (4.16)$$

where $V \in \mathbb{R}^{n \times q}$ is the matrix of eigenvectors and $\Lambda \in \mathbb{R}^{q \times q}$ is a diagonal matrix whose elements are the dominant eigenvalues. It is assumed that the eigenvectors can be factorized as

$$V = ZS, \quad (4.17)$$

where $Z^T Z = I_q$. Problem (4.16) can be rewritten as

$$AV = V\Lambda \Rightarrow AZS = ZS\Lambda \Rightarrow AZ = ZS\Lambda S^{-1} \Rightarrow AZ = ZK, \quad (4.18)$$

where matrix K is not necessarily a diagonal matrix. This system is undetermined [81]. To determine the problem we introduce the biorthogonality condition $W^T Z = I_q$, where W is a fixed matrix of rank q . Then the Newton method is used to solve the problem

$$F_W(Z, K) := \begin{bmatrix} AZ - ZK \\ W^T Z - I_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (4.19)$$

From Newton's method the new iterated solution arises

$$Z^{(k+1)} = Z^{(k)} - \Delta Z^{(k)}, \quad K^{(k+1)} = K^{(k)} - \Delta K^{(k)}, \quad (4.20)$$

where $\Delta Z^{(k)}$ and $\Delta K^{(k)}$ are solutions of the system

$$\begin{aligned} A\Delta Z^{(k)} - \Delta Z^{(k)}K^{(k)} - Z^{(k)}\Delta K^{(k)} &= AZ^{(k)} - Z^{(k)}K^{(k)}, \\ W^T\Delta Z^{(k)} &= W^T Z^{(k)} - I_q. \end{aligned} \quad (4.21)$$

The system (4.21) is coupled because of the off-diagonal elements of $K^{(k)}$.

To avoid this difficulty the Modified Block Newton Method (see Algorithm (4)) applies two previous steps to decouple system (4.21). The first step consists of an orthogonalisation to the $Z^{(k)}$ matrix (see Algorithm (2)).

Algorithm 2 Modified Gram-Schmidt Orthogonalisation (Orth)

Require: A set of p independent vectors v_1, \dots, v_p

- 1: **for** $i = 1$ to k **do**
 - 2: **for** $j = 1$ to $i - 1$ **do**
 - 3: $v_i := v_i - \langle v_j, v_i \rangle v_j$
 - 4: **end for**
 - 5: $v_i := v_i / \|v_i\|_2$
 - 6: **end for**
-

Once $Z^{(k)}$ is an orthonormal matrix, i.e., $Z^{(k)T}Z^{(k)} = I_q$, as a second step, a Rayleigh-Ritz procedure is applied, which consists of obtaining the eigenvectors $U^{(k)}$ and their corresponding eigenvalues $R^{(k)}$, of $B^{(k)} = Z^{(k)T}AZ^{(k)}$, in such a way that

$$B^{(k)}U^{(k)} = U^{(k)}R^{(k)}. \quad (4.22)$$

Then, taking into account the definition of $B^{(k)}$ and making use of the orthogonality of matrix $Z^{(k)}$ on equation (4.22), the following equation is obtained

$$Z^{(k)T}AZ^{(k)}U^{(k)} = Z^{(k)T}Z^{(k)}U^{(k)}R^{(k)}. \quad (4.23)$$

Defining $\tilde{Z}^{(k)} := Z^{(k)}U^{(k)}$ and $\Lambda^{(k)} := R^{(k)}$, from equation (4.23) we have that $\Lambda^{(k)}$ is a diagonal matrix whose elements, $\lambda_i^{(k)}$ are called the Ritz values and $\tilde{Z}^{(k)}$ are the approximated Ritz eigenvectors, that satisfy the equation

$$Z^{(k)T} \left(A\tilde{Z}^{(k)} - \tilde{Z}^{(k)}\Lambda^{(k)} \right) = 0. \quad (4.24)$$

This Rayleigh-Ritz procedure is implemented in Algorithm (3). At each iteration, the matrix W is chosen as the previous approximation for the invariant subspace,

that is $W = Z^{(k)}$, and system (4.21) is decoupled into the q linear systems

$$\begin{bmatrix} A - I\lambda_i^{(k)} & Z^{(k)} \\ Z^{(k)\top} & 0 \end{bmatrix} \begin{bmatrix} \Delta z_i^{(k)} \\ -\Delta\lambda_i^{(k)} \end{bmatrix} = \begin{bmatrix} AZ^{(k)} - Z^{(k)}\Lambda^{(k)} \\ 0 \end{bmatrix} \mathbf{e}^i, \quad i = 1, \dots, q. \quad (4.25)$$

where \mathbf{e}^i denotes the i -th coordinate vector, and $\Delta z_i^{(k)}$ is the i -th column of $\Delta Z^{(k)}$.

Algorithm 3 Rayleigh-Ritz procedure (rr)

Require: Initial approximation Z for the desired invariant subspace of A

- 1: $B := Z^T A Z$
 - 2: Calculate $[U, R]$ such that $BU = UR$ for the small matrix B
 - 3: $V := ZU$ {Approximated Ritz eigenvectors}
 - 4: $\Lambda := R$ {Ritz Values}
-

The Modified Block Newton Method can be summarised as in Algorithm (4).

Algorithm 4 Modified Block Newton Method (MBNM)

Require: Initial approximation $\tilde{V}^{(0)}$ of the eigenmodes

- 1: $[Z^{(0)}] := \text{orth}(\tilde{V}^{(0)})$
 - 2: $[V^{(0)}, \Lambda^{(0)}] := \text{rr}(Z^{(0)}, A)$
 - 3: **while** $V^{(k)}$ does not satisfy a termination criterion **do**
 - 4: $\Delta V^{(k)} = [\Delta v_1^{(k)}, \dots, \Delta v_p^{(k)}]$ {Correction determined with the Newton iteration of equation (4.25)}
 - 5: $\tilde{V}^{(k)} := V^{(k)} - \Delta V^{(k)}$
 - 6: $Z^k := \text{orth}(\tilde{V}^{(k)})$
 - 7: $[V^{(k)}, \Lambda^{(k)}] := \text{rr}(Z^{(k)}, A)$
 - 8: $k := k + 1$
 - 9: **end while**
-

4.1.3 Numerical Results

The behaviour of the Block Newton method presented above for updating the dominant Lambda modes of a nuclear reactor core is tested with four different problems, two of them with hexagonal geometry for the fuel assemblies, and the other two with rectangular geometry. The computational effort of these methods are not compared with the classical methods to calculate the Lambda modes without update, due to the different implementations. Thus, the number of iterations is shown because it gives some insight of the behavior.

Starting from a steady state configuration of a nuclear power reactor some situations arise in which the reactor configuration is perturbed. Different events justify

the study of these perturbations: poisoning by Xenon, insertion of control rods, subcooling of the entry, study of Doppler effect of a nuclear core by means of a uniform perturbation of the core, etc. To test the behaviour of the Modified Block Newton method different perturbations have been defined for different reactors. First, a bidimensional VVER 1000 reactor [34] will be perturbed increasing the second group absorption cross section Σ_{a2} of seven control rods equally distributed over the reactor. This is a symmetric perturbation, where the shapes of the modes after the perturbation is close to the one of the initial state, although the corresponding eigenvalues are quite different.

Similarly, the second problem is a perturbation for a VVER 440 reactor core which is based on a collapsed three dimensional benchmark for a transient in the reactor [57]. In this transient the peak of the power distribution performs a large increase in a non-symmetric form and the dominant eigenmodes change their shapes along the transient.

The first problem with rectangular geometry considered is a perturbed configuration of the well known three dimensional IAEA reactor [82], which is a PWR reactor. The perturbation is performed by means of the extraction of a set of control rods and the insertion of another set symmetrically.

The last example considered corresponds to the computation of the dominant eigenmodes of two configurations of a three dimensional BWR reactor in a boron injection transient. This problem shows a strong perturbation for the shape of the dominant modes to be updated, and it is proposed to check the behaviour of Newton method for updating the modes in strong perturbations.

4.1.3.1 VVER-1000 problem

The VVER-1000 problem [34] with the geometry shown in Figure 4.1 is considered. The core has a 1/6 cyclic symmetry, and the assembly pitch is of 23.60 cm. The reflector is not explicitly modeled, but it is assumed to be represented by means of albedos at the boundary of the core.

This original configuration will be perturbed increasing the second group absorption macroscopic cross section Σ_{a2} (see Table 4.1) of material 2 from 0.0810328 to 0.2010328, simulating a perturbation due to the insertion of control rod banks at these positions.

The results for the first seven dominant eigenvalues of the initial configuration, together with the eigenvalues corresponding to the perturbed configuration are shown in Table 4.2.

The shapes for the normalised power distributions associated with the fundamental mode and the first subcritical harmonic mode are shown in Figure 4.2.

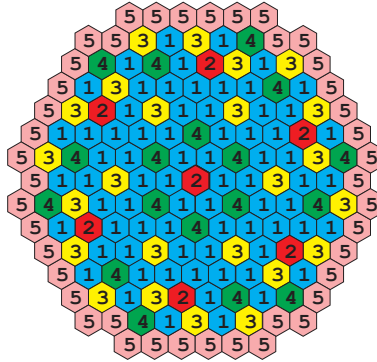


Figure 4.1: Geometry of the VVER-1000 reactor.

Table 4.1: Cross sections of the VVER-1000 problem.

Mat.	g	$D_g(cm)$	$\Sigma_{a,g}(cm^{-1})$	$\Sigma_{g,g+1}(cm^{-1})$	$\nu\Sigma_{f,g}(cm^{-1})$
1	1	1.38320	0.0083859	0.0164977	0.00481619
	2	0.386277	0.0673049		0.0846154
2	1	1.38299	0.0115490	0.0147315	0.00466953
	2	0.389403	0.0810328		0.0852264
3	1	1.39522	0.0089441	0.0156219	0.00604889
	2	0.386225	0.0844801		0.1194280
4	1	1.39446	0.0119932	0.0140185	0.00591507
	2	0.387723	0.0989670		0.1204970
5	1	1.39506	0.0091160	0.0154981	0.00640256
	2	0.384492	0.0893878		0.1292810

Table 4.2: Eigenvalues of the initial and perturbed configurations of VVER-1000 reactor.

l	λ_l	
	Initial conf.	Perturbed conf.
1	1.006451	0.995906
2	0.994809	0.984379
3	0.994809	0.984379
4	0.973759	0.964955
5	0.973759	0.964955
6	0.955171	0.953445
7	0.948344	0.941234

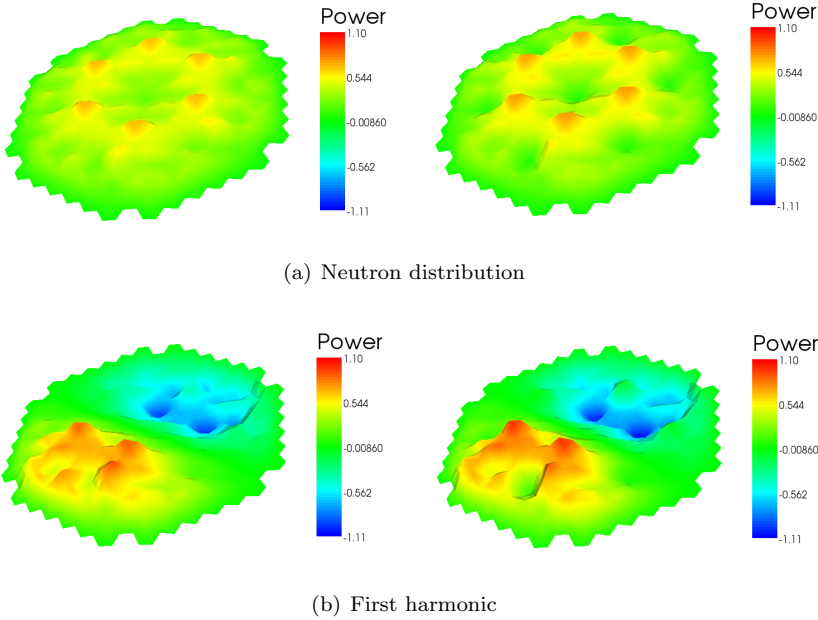


Figure 4.2: Shapes of the power distributions associated with the first two dominant modes of the VVER-1000 problem for the initial configuration (left) and the perturbed one (right).

The stopping criterion used in the Modified Block Newton method is based on the residual of the calculated eigenvalues and eigenvectors at step k ,

$$Res_k := \|AV^k - V^k\Lambda^k\|_2 \leq eps . \quad (4.26)$$

The tolerance has been set to $eps = 10^{-6}$. To show the performance of the Newton method along the different steps, in Table 4.3, we present the value of the residual Res_k for different steps of the Newton Method using eigenvector spaces with different size q (q is the number of modes). We observe that for this problem three steps of the Modified Block Newton method are enough to achieve the convergence of 7 modes of the perturbed configuration.

4.1.3.2 VVER-440 problem

This problem is based on the 3-dimensional transient benchmark AER-DYN-001 proposed in [57]. The nuclear cross-section given in the 3D benchmark have been collapsed in a single plane. Materials of the bidimensional reactor have been defined as is shown in Figure 4.3 and the cross sections of the materials are shown in Table 4.4.

Table 4.3: Evolution of the residual in the problem VVER-1000 for different steps of the Newton method and different number of modes.

q	Res_k			
	$k = 0$	$k = 1$	$k = 2$	$k = 3$
1	0.043835	0.000628	0.000006	0.000000
2	0.044356	0.000797	0.000005	0.000000
3	0.044396	0.000851	0.000006	0.000000
4	0.044505	0.001341	0.000008	0.000000
5	0.044489	0.001376	0.000008	0.000000
6	0.044494	0.001379	0.000007	0.000000
7	0.047095	0.001305	0.000008	0.000000

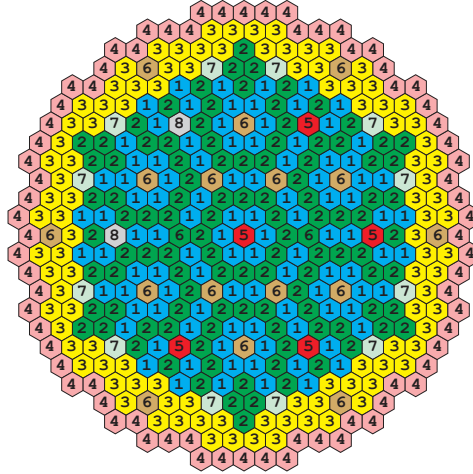


Figure 4.3: VVER 440 2D.

A perturbed configuration that simulates the movement of two control rods has been defined by means of changing the absorption cross section Σ_{a2} for the material 8 from 0.11887 to 0.016917.

The first 7 dominant Lambda modes of the initial configuration of the reactor have been updated to obtain a set of dominant modes of the perturbed configuration. The eigenvalues of these modes are shown in Table 4.5.

The shapes for the normalised power distributions associated with the fundamental mode and the first two subcritical modes are shown in Figure 4.4.

The residual of the calculated eigenvalues and eigenvectors at step k of the Modified Block Newton Method using different numbers of modes, q , is presented at

Table 4.4: Cross sections for the 2-D VVER 440 reactor.

Mat.	g	$D_g(cm)$	$\Sigma_{a,g}(cm^{-1})$	$\Sigma_{g,g+1}(cm^{-1})$	$\nu\Sigma_{f,g}(cm^{-1})$
1	1	1.346557	0.008312	0.016976	0.004413
	2	0.370075	0.064282		0.072784
2	1	1.337728	0.008745	0.016000	0.005491
	2	0.367411	0.079145		0.104256
3	1	1.332264	0.009411	0.014974	0.006990
	2	0.363171	0.099536		0.147261
4	1	1.447520	0.000933	0.032215	0.000000
	2	0.251741	0.033037		0.000000
5	1	1.231711	0.012120	0.020782	0.001345
	2	0.240027	0.118846		0.027352
6	1	1.337727	0.008747	0.015996	0.005492
	2	0.367479	0.079153		0.104316
7	1	1.346561	0.008317	0.016968	0.004416
	2	0.370177	0.064282		0.072846
8	1	1.231640	0.012123	0.020785	0.001342
	2	0.239942	0.118870		0.027299

Table 4.5: Eigenvalues for the initial and perturbed configurations of reactor VVER 440.

l	λ_l	
	Initial conf.	Perturbed conf.
1	1.005186	1.011159
2	0.998421	1.001079
3	0.998420	1.000791
4	0.984674	0.988257
5	0.984675	0.985502
6	0.967776	0.967903
7	0.964815	0.965150

Table 4.6. We observe that also in this case the modes can be updated with three steps of the Newton method.

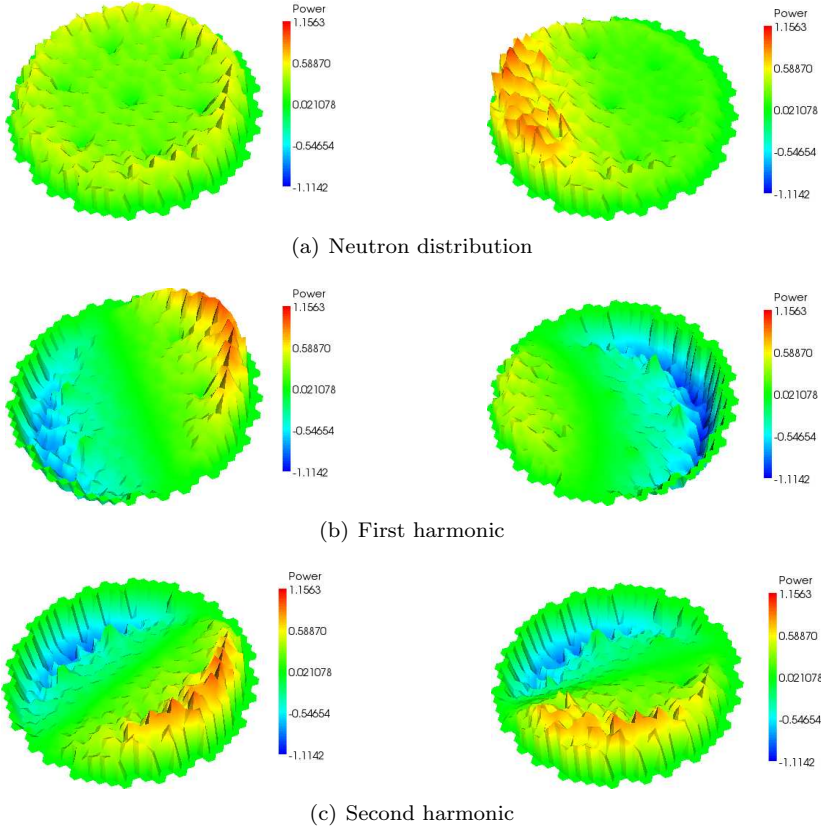


Figure 4.4: Shapes of the normalised power distributions associated with the first three dominant modes of the initial configuration (left) and perturbed configuration (right) of reactor VVER-440.

Table 4.6: Residual evolution for different steps of the Newton method for different number of modes of reactor VVER-440.

q	Res_k			
	$k = 0$	$k = 1$	$k = 2$	$k = 3$
1	0.015864	0.002868	0.000292	0.000000
2	0.015866	0.002868	0.000292	0.000000
3	0.023867	0.000599	0.000000	0.000000
4	0.025380	0.000398	0.000000	0.000000
5	0.025380	0.000546	0.000000	0.000000
6	0.025380	0.000546	0.000000	0.000000
7	0.025630	0.000546	0.000000	0.000000

4.1.3.3 IAEA 3D problem

The IAEA 3-D PWR problem [82] is a standard benchmark problem to measure the performance of neutronic calculation methods. The core is composed by 177 fuel assemblies including 9 fully rodded fuel assemblies and 4 partially rodded fuel assemblies, as it is shown in Figure 4.5. Radial reflector is modelled by means of 64 assemblies surrounding the core. The fuel assembly pitch is 20cm and the active height of a fuel assembly is 340cm. The thickness of axial reflector is 20cm. Nuclear cross sections for this problem are shown in Table 4.7.

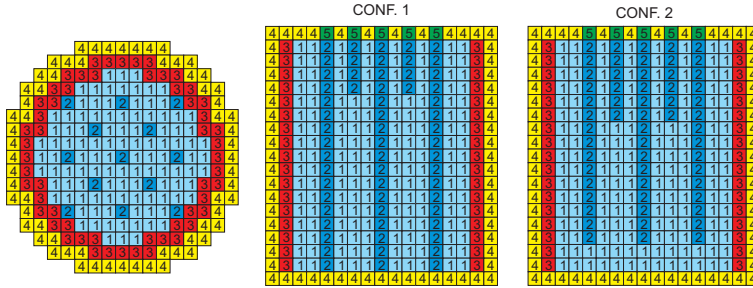


Figure 4.5: Geometry of the 3D IAEA without reflector problem.

Table 4.7: Cross sections of 3D IAEA problem.

Mat.	g	$D_g(cm)$	$\Sigma_{a,g}(cm^{-1})$	$\Sigma_{g,g+1}(cm^{-1})$	$\nu\Sigma_{f,g}(cm^{-1})$
1	1	1.500	0.010	0.020	0.000
	2	0.400	0.085		0.135
2	1	1.500	0.010	0.020	0.000
	2	0.400	0.130		0.135
3	1	1.500	0.010	0.020	0.000
	2	0.400	0.080		0.135
4	1	2.000	0.000	0.040	0.000
	2	0.300	0.010		0.000
5	1	2.000	0.000	0.040	0.000
	2	0.300	0.055		0.000

This reactor, and the following ones with rectangular geometry for the fuel assemblies, are spatially discretized using the Nodal Collocation Method [11]. As it is shown in Figure 4.5, two configurations are considered for this problem, CONF.1 is the initial configuration and CONF.2 is the perturbed one. In the perturbed configuration the control rods of fully inserted assemblies have been extracted 40cm, and the control rods of partially rodded assemblies are inserted 40cm. Table 4.8, shows the three dominant eigenvalues obtained for both configurations.

Table 4.8: First three dominant eigenvalues of 3D IAEA reactor.

l	λ_l	
	Initial conf.	Perturbed conf.
1	1.028857	1.027907
2	1.016490	1.015933
3	1.016490	1.015933

The axial and radial profiles of the normalised power distributions associated with the fundamental mode and the first harmonic mode are shown in Figure 4.6.

The evolution of the residual of the calculated eigenvalues and eigenvectors at step k of the Modified Block Newton Method using different number of modes q , is presented at Table 4.9. We can see that only 3 steps of the Newton's method are enough to reach the convergence, as in the previous problems.

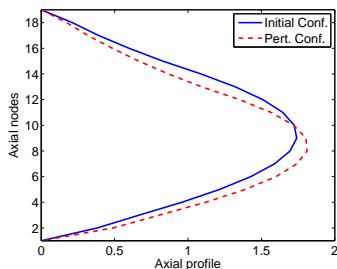
Table 4.9: Residual for different steps of Newton's method to update the modes of 3D IAEA reactor.

q	Res_k			
	$k = 0$	$k = 1$	$k = 2$	$k = 3$
1	0.012160	0.000090	0.000001	0.000000
2	0.012160	0.000119	0.000001	0.000000
3	0.019517	0.000246	0.000002	0.000000

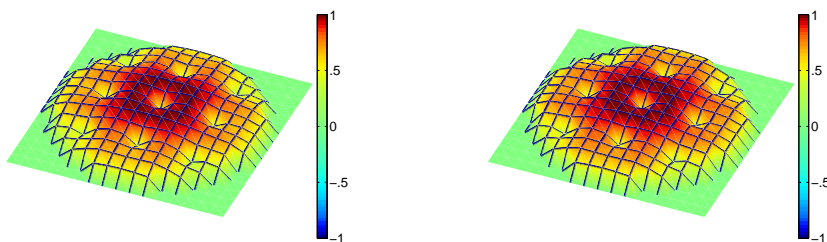
4.1.3.4 Boron Injection in a BWR

An interesting transient is found in the BWR reactors when the SLCS (standby liquid control system) injects water with a high proportion of boron dissolved [83]. This system is expected to work when the temperature of suppression pool is high enough after the closure of the MSIV (Main Steam Isolation Valve) and with a ATWS (Anticipated Transient Without Scram). In this situation, the reactor can increase its power and an instability event can occur. The Lambda Modes are eigenfunctions, which have been successfully used to describe the instable events in BWRs [3].

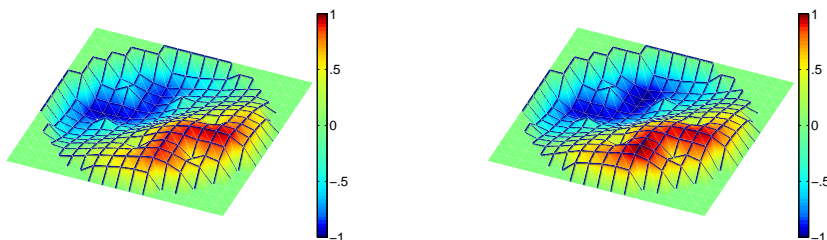
An initial configuration of a typical BWR reactor is considered and a perturbed configuration is proposed due to the injection of water with a proportion of boron dissolved of 1500 ppm. This consists of a strong perturbation of the reactor core. The first three dominant eigenvalues associated with the reactor before the poisoning are shown in Table 4.10, together with the three dominant eigenvalues associated with the reactor perturbed by boron.



(a) Axial profile for the fundamental mode of the two configurations of the 3D IAEA reactor.



(b) Radial profile of the fundamental mode for the initial configuration (left) and the perturbed configuration (right) of the 3D IAEA reactor.



(c) Radial profile for the first subcritical mode of the initial configuration (left) and the perturbed configuration (right) of the 3D IAEA reactor.

Figure 4.6: Shapes of the normalised power distributions of the first two dominant Lambda modes of the initial and the perturbed configuration of 3D IAEA reactor.

The profiles of the normalised power distributions associated with the fundamental mode and the first subcritical mode are shown in Figure 4.7.

Table 4.10: Eigenvalues associated with the two configurations of the BWR reactor.

l	λ_l	
	Without Boron	With Boron
1	1.000487	0.937197
2	0.989536	0.919430
3	0.989396	0.919146

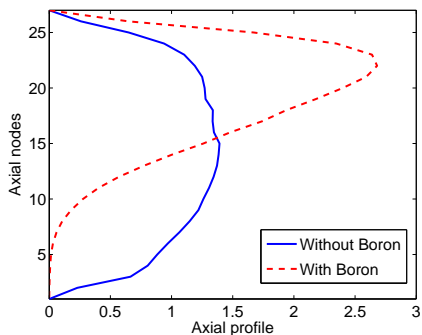
The evolution of the residual of the calculated eigenvalues and eigenvectors at different steps of the Block Newton method for different number of modes q , is presented in Table 4.11. We can see that even with a strong perturbation, as the one induced by Boron in the BWR reactor, a small number of steps of the Newton method are enough to update the modes.

Table 4.11: Residual evolution for different steps of the Newton method to update the modes of the BWR reactor.

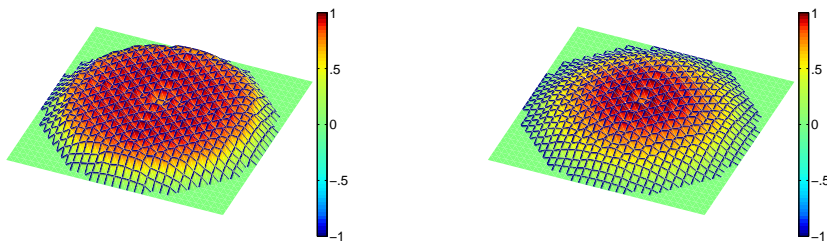
q	Res_k					
	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
1	0.037841	0.017591	0.004369	0.000045	0.000000	0.000000
2	0.046302	0.008186	0.000207	0.000006	0.000003	0.000000
3	0.045572	0.005513	0.000300	0.000014	0.000000	0.000000

Summarizing, the Lambda Modes of a reactor for a given configuration of the core are used to integrate the time dependent neutron diffusion equation, using modal methods. To compute a large amount of Lambda modes is a prohibitive task from the computational point of view and to obtain accurate results with a small amount of modes a modes updating strategy is needed. Also to update the modes of a reactor core can be useful in perturbative calculations. Because the calculation of the Lambda modes is an expensive task, efficient methods to update the Lambda modes using the information provided by previous calculations are necessary to improve the utility of modal and perturbative methods.

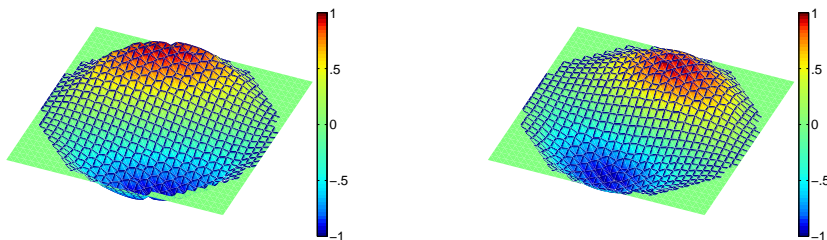
Once an initial set of Lambda modes has been obtained, to update these modes we have used a Modified Block Newton Method, which has a local cubic convergence [81]. By studying different benchmark problems, we have shown that this method is robust to deal with problems that have degenerate or clustered eigenvalues, and it has fast convergence in such a way that the number of steps needed to update the modes is not very dependent on the kind of perturbation considered and the number of modes to be updated. For these reasons it seems to be interesting to take advantage of the convergence properties of this kind of updating methods when perturbed configurations of a reactor core have to be studied.



(a) Axial profiles of the fundamental mode for the initial and the perturbed configuration.



(b) Radial profile of the fundamental mode for the initial configuration (left) and the perturbed configuration (right).



(c) Radial profile of the first subcritical mode for the initial configuration (left) and the perturbed configuration (right).

Figure 4.7: Shapes of the normalised power distributions of the first two dominant Lambda modes of the initial and the perturbed configuration of BWR reactor.

4.2 Alternative Newton Methods

The updating modes problem can be viewed as a subspace tracking problem [84]. Thus, we have to compute a set of dominant eigenvalues and their corresponding

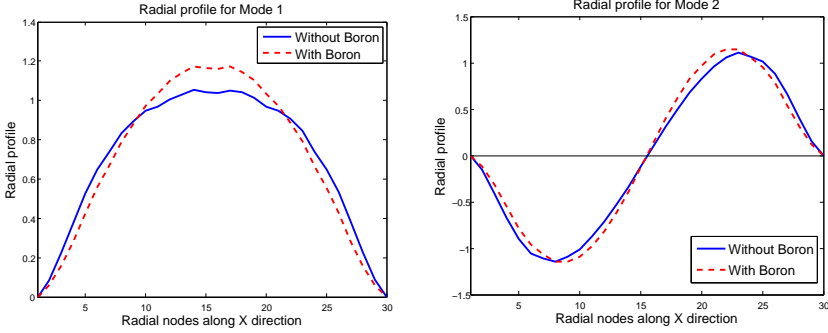


Figure 4.8: Radial layout for the fundamental (left) and for the first subcritical mode (right) for the initial configuration (without Boron) and the perturbed configuration (with Boron).

eigenvectors of the problem

$$Av_i^1 = k_i^1 v_i^1, \quad i = 1, \dots, p \quad (4.27)$$

using as initial guess the eigenvectors $\{v_1^0, \dots, v_p^0\}$ of a matrix which is ‘close’ to matrix A .

The Lambda modes problem is usually solved for the dominant eigenvalues and their corresponding eigenvectors using, for example, the Implicit Restarted Arnoldi method [85, 5]. To update the modes along a transient it is interesting to take into account that the space spanning the new modes should be close to the space spanning the previous ones. We will consider different ‘Block Newton-like’ methods, [81], for the subspace tracking problem and we will compare their performance for updating the Lambda modes problem. These methods are a Modified Block Newton Method (MBNM), a One Sided Block Newton Method (OSBNM), and a Two Sided Block Newton Method (TSBNM).

4.2.1 Modified Block Newton Method

The Modified Block Newton Method has been exposed in the section 4.1 and can be summarized as in Algorithm 5, where the function `orth` uses the modified Gram-Schmidt algorithm to orthogonalize the columns of the initial eigenvectors (see algorithm 2), and the `rr` is the Rayleigh–Ritz procedure (see algorithm 3).

Algorithm 5 Modified Block Newton Method (MBNM)**Require:** Initial approximation $V^{(0)}$ of the eigenmodes

- 1: $[Z^{(0)}] := \text{orth}(V^{(0)})$
- 2: $[\tilde{Z}^{(0)}, \Lambda^{(0)}] := \text{rr}(Z^{(0)}, A)$
- 3: **while** $\tilde{Z}^{(k)}$ does not satisfy a termination criterion **do**
- 4: $\Delta\tilde{Z}^{(k)} = [\Delta\tilde{z}_1^{(k)}, \dots, \Delta\tilde{z}_p^{(k)}]$ {Correction determined with the Newton iteration of equation (4.25)}
- 5: $V^{(k)} := \tilde{Z}^{(k)} - \Delta\tilde{Z}^{(k)}$
- 6: $Z^k := \text{orth}(V^{(k)})$
- 7: $[\tilde{Z}^{(k)}, \Lambda^{(k)}] := \text{rr}(Z^{(k)}, A)$
- 8: $k := k + 1$
- 9: **end while**

4.2.2 One Sided Block Newton Method

Once the initial approximated eigenvectors, Z , of problem (4.18) are an orthonormal set, we are interested in getting a correction, ΔZ , to these eigenvectors satisfying

$$A(Z + \Delta Z) = (Z + \Delta Z)(K + \Delta K) . \quad (4.28)$$

Developing the previous expression and removing the second order terms, we get an expression of the form

$$AZ + A\Delta Z = ZK + \Delta ZK + Z\Delta K . \quad (4.29)$$

Reordering the terms, and naming $R = AZ - ZK$, we get

$$A\Delta Z - \Delta ZK - Z\Delta K = -R . \quad (4.30)$$

If we use an extra condition to the correction term ΔZ , which is $Z^T \Delta Z = 0$, it is easy to show that $(I - ZZ^T)R = R$, $(I - ZZ^T)\Delta Z = \Delta Z$, and $(I - ZZ^T)Z\Delta K = 0$. Using the projector $(I - ZZ^T)$, the following projected equation is obtained

$$(I - ZZ^T)A(I - ZZ^T)\Delta Z - \Delta ZK = -R, \quad (4.31)$$

which is a Sylvester equation [86]. This equation can be solved by using the Schur method. The One Sided Block Newton method can be structured as shown in Algorithm 6.

4.2.3 Two Sided Block Newton Method

Given a partial eigenvalue problem, and its transpose problem,

$$AU = U\Lambda , \quad A^T V = V\Lambda , \quad (4.32)$$

Algorithm 6 One Sided Block Newton Method (OSBNM)**Require:** Initial approximation $V^{(0)}$ of the eigenmodes

- 1: $[Z^{(0)}] := \text{orth}(V^{(0)})$
- 2: $[K] := Z^{(0)T}AZ^{(0)}$
- 3: **while** $Z^{(k)}$ does not satisfy a termination criterion **do**
- 4: $\Delta Z^{(k)}$ {Correction determined with the Sylvester equation (4.31)}
- 5: $V^{(k)} := Z^{(k)} - \Delta Z^{(k)}$
- 6: $Z^{(k)} := \text{orth}(V^{(k)})$
- 7: $K := Z^{(k)T}AZ^{(k)}$
- 8: $k := k + 1$
- 9: **end while**
- 10: $[V^{(k)}, \Lambda^{(k)}] := \text{rr}(Z^{(k)}, A)$

it is assumed that we can biorthogonalize the sets of vectors U and V in such a way that $V^T U = I_p$.

Looking for the corrections ΔU and ΔV , by using the same procedure as the one used in the previous method, we arrive at a set of equations for the direct and for the transposed problem as follows

$$\begin{aligned} A\Delta U - \Delta U K - U\Delta K &= -R_u, \\ A^T\Delta V - \Delta V K^T - V\Delta K^T &= -R_v, \end{aligned} \quad (4.33)$$

where $R_u = AU - UK$, and $R_v = A^T V - V K^T$.

Multiplying the first equation by the oblique projector $(I - UV^T)$ and the second equation by $(I - VU^T)$, and using the properties $V^T \Delta U = 0$ and $\Delta V^T U = 0$, the following Sylvester equations are obtained

$$\begin{aligned} (I - UV^T)A(I - UV^T)\Delta U - \Delta U K &= -R_u, \\ (I - VU^T)A'(I - VU^T)\Delta V - \Delta V K^T &= -R_v. \end{aligned} \quad (4.34)$$

Algorithm 7 summarizes the Two sided block Newton Method (TSBNM), where the function **BMGS** uses the Biorthogonalized Modified Gram-Schmidt algorithm, and **rr** is the Rayleigh–Ritz procedure.

4.2.4 Numerical Results

Starting from a steady state configuration of a nuclear power reactor some situations arise in which the reactor configuration is perturbed. Different events justify the study of these perturbations: poisoning by Xenon, insertion of control rods, subcooling of the entry, study of the Doppler effect of a nuclear core by means of a uniform perturbation of the core, etc. An interesting transient of this kind is

Algorithm 7 Two Sided Block Newton Method (TSBNM)

Require: Initial approximation \tilde{U}^0 of the direct eigenvectors, and initial approximation for the adjoints \tilde{V}^0

- 1: $[U^0, V^0] := \text{BMGS}(\tilde{U}^0, \tilde{V}^0)$
- 2: $[K] := V^{(0)T} A U^{(0)}$
- 3: **while** $U^{(k)}$, and $V^{(k)}$ does not satisfy a termination criterion **do**
- 4: $\Delta U^{(k)}; \Delta V^{(k)}$ {Corrections determined with the Sylvester equation (4.34)}
- 5: $\tilde{U}^{(k)} := U^{(k)} + \Delta U^{(k)}$ and $\tilde{V}^{(k)} := V^{(k)} + \Delta V^{(k)}$
- 6: $[U^{(k)}, V^{(k)}] := \text{BMGS}(\tilde{U}^{(k)}, \tilde{V}^{(k)})$
- 7: $K := V^{(k)T} A U^{(k)}$
- 8: $k := k + 1$
- 9: **end while**
- 10: $[\tilde{U}^k, \Lambda^k] := \text{rr}(U^k, A)$, and $[\tilde{V}^k, \Lambda^k] := \text{rr}(V^k, A^T)$

found in the BWR reactors when the SLCS (standby liquid control system) injects water with a high proportion of boron dissolved [83].

We consider a transient associated with a boron injection from the SLCS system. During the transient the boron injected is mixed with the coolant increasing the concentration with time. To study the configuration of the reactor core along the transient we consider three different configurations associated with three different concentrations of boron in the coolant, that is: a configuration with 0 ppm of boron, a configuration with 750 ppm of boron, and a last configuration with 1500 ppm of boron. The modes of the first configuration are computed by means of the Arnoldi method, and the updating methods are used to compute the perturbed configuration.

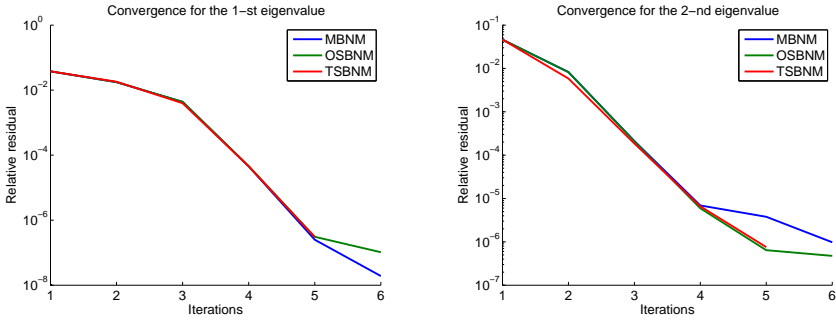
The first 3 eigenvalues for the three configurations considered are shown in Table 4.12.

Table 4.12: Eigenvalues for three boron concentrations.

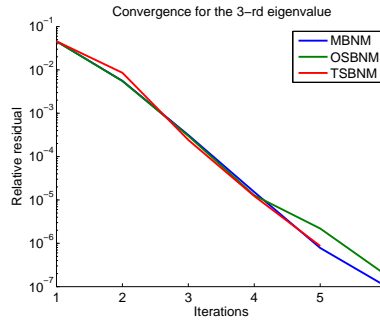
	0 ppm	750 ppm	1500 ppm
k_1	1.000487	0.964937	0.937197
k_2	0.989396	0.948851	0.919146
k_3	0.989536	0.948736	0.919430

Figure 4.9 shows the convergence behaviour with the iterations used by the three methods.

Figure 4.10 shows the convergence behaviour with the iterations used by the three methods.



(a) Convergence for the first eigenvalue. (b) Convergence for the second eigenvalue.



(c) Convergence for the third eigenvalue.

Figure 4.9: Convergence for the first 3 eigenvalues from 0 to 1500 ppm of boron.

Results of updating these eigenvalues using the different Block Newton type methods are summarized in Tables 4.13 and 4.14, where we can see the good convergence properties of the three methods. Also, the dependence of the convergence on the initial guess can be observed, because of the local behaviour of the methods.

Table 4.13: Iterations to converge and final relative residual from 0 to 1500 ppm.

	k_1	k_2	k_3
MBNM	5 (2.4903 E-7)	6 (9.8444 E-8)	5 (6.8177 E-7)
OSBNM	5 (3.0750 E-7)	5 (6.4349 E-7)	6 (1.8426 E-7)
TSBNM	5 (3.0136 E-7)	5 (7.5582 E-7)	5 (8.6700 E-7)

The updating of the Lambda modes process is interesting for several applications as the development of modal methods and perturbative computations. Different methods for this process based on the block Newton method have been proposed and their performance has been tested using a Boron injection transient.

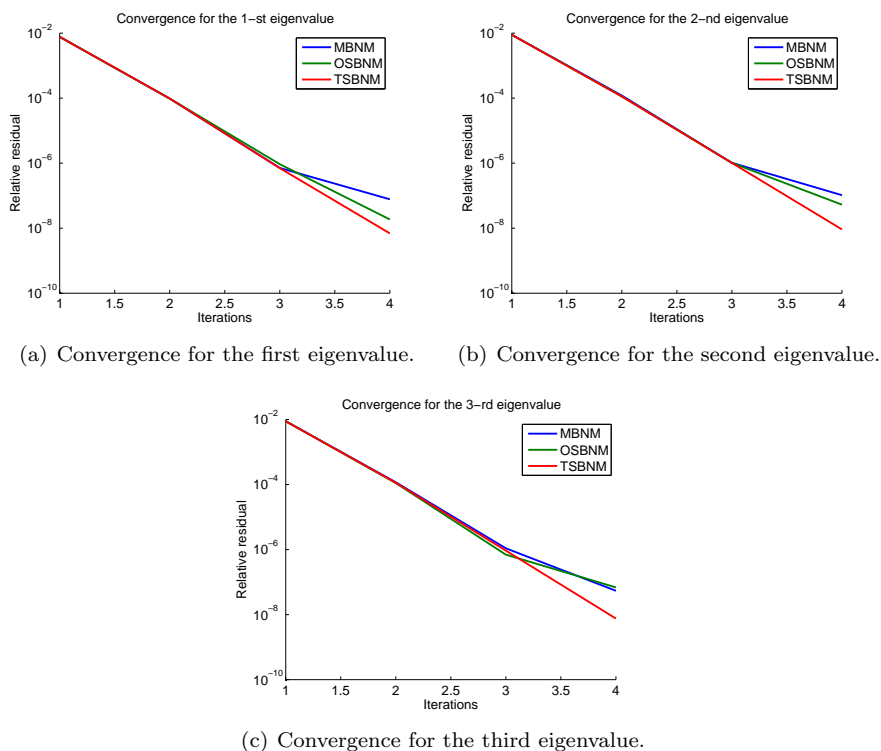


Figure 4.10: Convergence for the first 3 eigenvalues from 750 to 1500 ppm of boron.

Table 4.14: Iterations to converge and final relative residual from 750 to 1500 ppm.

	k_1	k_2	k_3
MBNM	3 (7.1018 E-7)	4 (1.0315 E-7)	4 (5.4099 E-8)
OSBNM	3 (9.2920 E-7)	4 (5.2896 E-8)	3 (6.9954 E-7)
TSBNM	3 (7.0191 E-7)	4 (9.1274 E-9)	3 (9.3166 E-7)

The One Sided and Two Sided Block Newton methods have shown the best performance regarding the number of iterations. Since the cost per iteration is cheaper in the One Sided Block Newton method, this is the best method for this application.

4.3 Proper Generalized Decomposition for eigenvalue computations

The Proper Generalized Decomposition (PGD) method is a recently developed strategy [87, 88] that uses the fact that some models encountered in science and engineering are defined in multidimensional spaces. When mesh-based discretization techniques are applied, these models exhibit what is called the curse of dimensionality, i.e., the number of unknowns increases exponentially with the spatial dimension of the geometry, leading to a very large amount of unknowns for high dimensional problems.

The PGD method is based on the use of a separated representation for the solution of a d -dimensional problem over a separable domain $\Omega = \Omega_1 \times \dots \times \Omega_d$, of the form

$$\Phi(x_1, \dots, x_d) = \sum_{i=1}^{\infty} \alpha_i \prod_{j=1}^d F_j^i(x_j) , \quad (4.35)$$

where functions $F_j^i(x)$ are normalized as follows

$$\|F_j^i\|_j^2 = \langle F_j^i, F_j^i \rangle_j = 1 , \quad \text{where} \quad \langle u, v \rangle_j = \int_{\Omega_j} u(x_j)v(x_j) dx_j . \quad (4.36)$$

The value of $\Phi(x_1, \dots, x_d)$ is approximated by using a finite number of terms in the sum (4.35).

The PGD technique computes the different functions involved in Equation (4.35) by means of an alternating directions linearization strategy. This technique is summarized here to establish the notation and methodology necessary to develop our strategy to solve differential eigenvalue problems. For the sake of clarity and without loss of generality, the discussion is restricted to a 2-dimensional Poisson's equation of the form

$$-\nabla \cdot D\nabla\Phi(x, y) = f(x, y) , \quad (4.37)$$

over a domain Ω which is separable into a Cartesian product of 1-dimensional spaces $\Omega = \Omega_x \times \Omega_y$. The weak formulation of equation (4.37) is

$$\iint_{\Omega} -\nabla \cdot D\nabla\Phi(x, y) \varphi(x, y) dx dy = \iint_{\Omega} f(x, y)\varphi(x, y) dx dy , \quad (4.38)$$

where $\Phi(x, y)$ belongs to a given trial space and $\varphi(x, y)$ to a test space of functions.

Applying Green's identity to equation (4.38) and assuming homogeneous boundary conditions, we obtain the formulation

$$\iint_{\Omega} D\nabla\Phi(x, y) \cdot \nabla\varphi(x, y) dx dy = \iint_{\Omega} f(x, y)\varphi(x, y) dx dy . \quad (4.39)$$

Let us assume now that the solution is separable, i.e.,

$$\Phi(x, y) = G_1(x)G_2(y) , \quad (4.40)$$

where $G_1(x) \in W_1(\Omega_x)$, being $W_1(\Omega_x)$ a finite element space for a suitable discretization of domain Ω_x , and $G_2(y) \in W_2(\Omega_y)$ defined in the same way; and the trial functions are of the form [87]

$$\varphi(x, y) = G_1^*(x)G_2(y) + G_1(x)G_2^*(y) , \quad (4.41)$$

being G_i the unknown functions defining $\Phi(x, y)$, and G_i^* are arbitrary functions. Substituting (4.40) and (4.41) into equation (4.39) we obtain

$$\begin{aligned} & \iint_{\Omega} \left(D \frac{\partial G_1}{\partial x}(x) \frac{\partial G_1^*}{\partial x}(x) (G_2(y))^2 + D G_1(x) G_1^*(x) \left(\frac{\partial G_2}{\partial y}(y) \right)^2 \right) dx dy + \\ & \iint_{\Omega} \left(D \left(\frac{\partial G_1}{\partial x}(x) \right)^2 G_2(y) G_2^*(y) + D (G_1(x))^2 \frac{\partial G_2}{\partial y}(y) \frac{\partial G_2^*}{\partial y}(y) \right) dx dy = \\ & \iint_{\Omega} f(x, y) G_1^*(x) G_2(y) dx dy + \iint_{\Omega} f(x, y) G_1(x) G_2^*(y) dx dy . \end{aligned} \quad (4.42)$$

Due to the arbitrariness of G_1^* and G_2^* we can solve equation (4.42) by solving recursively the following two equations [88]

$$\begin{aligned} & \iint_{\Omega} D \frac{\partial G_1}{\partial x}(x) \frac{\partial G_1^*}{\partial x}(x) (G_2(y))^2 + D G_1(x) G_1^*(x) \left(\frac{\partial G_2}{\partial y}(y) \right)^2 dx dy = \\ & \iint_{\Omega} f(x, y) G_1^*(x) G_2(y) dx dy , \end{aligned} \quad (4.43)$$

and

$$\begin{aligned} & \iint_{\Omega} D \left(\frac{\partial G_1}{\partial x}(x) \right)^2 G_2(y) G_2^*(y) + D (G_1(x))^2 \frac{\partial G_2}{\partial y}(y) \frac{\partial G_2^*}{\partial y}(y) dx dy = \\ & \iint_{\Omega} f(x, y) G_1(x) G_2^*(y) dx dy , \end{aligned} \quad (4.44)$$

i.e., solving equation (4.43) for an arbitrarily fixed $G_2(y)$, and using the solution $G_1(x)$ to solve equation (4.44), and then proceeds recursively until the convergence for $G_1(x)G_2(y)$ is achieved. This is equivalent to solve the following two 1-dimensional problems

$$\int_{\Omega_x} \frac{\partial G_1}{\partial x}(x) \frac{\partial G_1^*}{\partial x}(x) D_x dx + \int_{\Omega_x} G_1(x) G_1^*(x) \hat{D}_x dx = \int_{\Omega_x} G_1^*(x) f_x(x) dx , \quad (4.45)$$

and

$$\int_{\Omega_y} G_2(y) G_2^*(y) \hat{D}_y dy + \int_{\Omega_y} \frac{\partial G_2}{\partial y}(y) \frac{\partial G_2^*}{\partial y}(y) D_y dy = \int_{\Omega_y} G_2^*(y) f_y(y) dx , \quad (4.46)$$

defining the new coefficients

$$\begin{aligned} D_x &= \int_{\Omega_y} D(G_2(y))^2 dy, & D_y &= \int_{\Omega_x} D(G_1(x))^2 dx, \\ \hat{D}_x &= \int_{\Omega_y} D\left(\frac{\partial G_2}{\partial y}(y)\right)^2 dy, & \hat{D}_y &= \int_{\Omega_x} D\left(\frac{\partial G_1}{\partial x}(x)\right)^2 dx, \\ f_x(x) &= \int_{\Omega_y} f(x,y)G_2(y) dy, & f_y(y) &= \int_{\Omega_x} f(x,y)G_1(x) dx. \end{aligned}$$

From here, the PGD method consists of an iterative procedure that can be outlined in the following three steps:

Step 1: Projecting the solution. Assuming we are at the k -th step of the iterative procedure, we have

$$\Phi(x, y) = \sum_{i=1}^k \alpha_i F_1^i(x) F_2^i(y). \quad (4.47)$$

where $F_1^i(x)$ and $F_2^i(y)$ are normalized as in (4.36).

For the first step, $k = 1$, functions F_j^1 , ($j = 1, 2$) are obtained normalizing functions G_j calculated using Equations (4.45) and (4.46).

Assuming that functions F_j^i are known, we obtain

$$\Phi^k(x, y) \in V^k := \left\{ \sum_{i=1}^k \alpha_i F_1^i(x) F_2^i(y) : \alpha_i \in \mathbb{R}, i = 1, \dots, k \right\}, \quad (4.48)$$

where the coefficients α_i are those that minimize the residual of Galerkin formulation

$$\iint_{\Omega} D \nabla \Phi^k(x, y) \cdot \nabla \Phi^{k*}(x, y) dx dy = \iint_{\Omega} f(x, y) \Phi^{k*}(x, y) dx dy, \quad (4.49)$$

where the test functions are

$$\Phi^{k*}(x, y) = \sum_{i=1}^k \alpha_i^* F_1^i(x) F_2^i(y) \in V^k. \quad (4.50)$$

Introducing (4.47) and (4.50) into equation (4.49) we obtain

$$\begin{aligned} & \iint_{\Omega} D \nabla \left(\sum_{i=1}^k \alpha_i F_1^i(x) F_2^i(y) \right) \cdot \nabla \left(\sum_{i=1}^k \alpha_i^* F_1^i(x) F_2^i(y) \right) dx dy \\ &= \iint_{\Omega} f(x, y) \left(\sum_{i=1}^k \alpha_i^* F_1^i(x) F_2^i(y) \right) dx dy. \end{aligned} \quad (4.51)$$

All the integrals in (4.51) can be evaluated, and making use of the arbitrariness of α_i^* , we obtain the system of linear equations

$$L\alpha = f , \quad (4.52)$$

where the elements of matrix L are defined as

$$L_{i,j} = \iint_{\Omega} D\nabla (F_1^i(x)F_2^i(y)) \cdot \nabla (F_1^j(x)F_2^j(y)) \, dx dy , \quad (4.53)$$

and the elements of f are

$$f_i = \iint_{\Omega} f(x,y)F_1^i(x)F_2^i(y) \, dx dy . \quad (4.54)$$

Step 2: Checking convergence. Let \mathcal{R}_1 be defined as

$$\mathcal{R}_1(u,v) = \iint_{\Omega} D\nabla u \cdot \nabla v \, dx dy - \iint_{\Omega} f(x,y)v \, dx dy \quad (4.55)$$

From the solution Φ^k at iteration k , given by Equation (4.47), we compute the relative residual Re_1 associated with Equation (4.39):

$$Re_1 = \frac{\mathcal{R}_1(\Phi, \Phi)}{\|\Phi\|} , \quad \|\Phi\|^2 = \iint_{\Omega} \Phi^2 \, dx dy , \quad (4.56)$$

and stop if $Re_1 < \varepsilon$, where ε is the desired tolerance for the residual.

Step 3: Enriching the approximation basis. Third step consists of enriching the approximation basis by obtaining a new term $\delta\Phi(x,y) = G_1(x)G_2(y)$ to correct the solution Φ^k . The correction equation we have to solve is

$$-\nabla \cdot D\nabla (\Phi^k(x,y) + \delta\Phi(x,y)) = f(x,y) , \quad (4.57)$$

that can be rewritten as

$$-\nabla \cdot D\nabla \delta\Phi(x,y) = f^{k+1}(x,y) , \quad (4.58)$$

where f^{k+1} is a known function defined as

$$f^{k+1}(x,y) = f(x,y) + \nabla \cdot D\nabla \Phi^k(x,y) . \quad (4.59)$$

Now it will be assumed that the test functions are of the form

$$\delta\varphi(x,y) = G_1^*(x)G_2(y) + G_1(x)G_2^*(y) , \quad (4.60)$$

and then, the weak form of equation (4.58) is written as

$$\iint_{\Omega} D\nabla \delta\Phi(x,y) \cdot \nabla \varphi(x,y) \, dx \, dy = \iint_{\Omega} f^{k+1}(x,y)\varphi(x,y) \, dx \, dy . \quad (4.61)$$

This equation is solved iteratively, in the same way as it is done in (4.39), by solving 1-dimensional problems similar to (4.45) and (4.46).

Functions F_j^{k+1} for the sum (4.47) are obtained normalizing G_j . The algorithm follows going again to *step 1* until the convergence criterion in *step 2* is fulfilled.

4.3.1 Eigenvalue problem

One example of a differential eigenvalue problem useful in nuclear reactor physics is the problem known as the Lambda modes problem [89, 8]. The one energy group approximation of the Lambda modes problem will be used to develop the method. The one energy group neutron diffusion equation is of the form

$$\mathcal{L}\Phi = \frac{1}{\lambda}\mathcal{M}\Phi, \quad (4.62)$$

where the neutron loss operator and the neutron production operator are defined, respectively, as

$$\mathcal{L} = -\vec{\nabla} \cdot (D_1 \vec{\nabla}) + \Sigma_{a1}, \quad \text{and} \quad \mathcal{M} = \nu \Sigma_{f1}.$$

This problem is assumed to be defined in a 2-dimensional separable domain $\Omega = \Omega_x \times \Omega_y$.

The algorithm proposed can be divided into an initialization step, a Rayleigh-Ritz procedure, another step for checking the convergence and a fourth step for enriching the approximation.

4.3.1.1 Initialization

The PGD method to compute the dominant eigenvalue of a differential eigenvalue problem and its corresponding eigenvector, is based on expressing the eigenvector Φ associated with the dominant eigenvalue λ in the k -th step as

$$\Phi^k(x, y) = \sum_{i=1}^k \alpha_i F_1^i(x) F_2^i(y). \quad (4.63)$$

For a first step, $k = 1$, the solution is completely separable, i.e., $\Phi^1(x, y) = \Phi_1(x)\Phi_2(y)$, and integrating in the y direction we obtain the 1-dimensional eigenvalue problem

$$\hat{\mathcal{L}}\Phi_1 = \frac{1}{\lambda}\hat{\mathcal{M}}\Phi_1, \quad (4.64)$$

where

$$\hat{\mathcal{L}} = -\vec{\nabla} \cdot (D_x \vec{\nabla}) + \hat{D}_x + \Sigma_{xa} \quad \text{and} \quad \hat{\mathcal{M}} = \nu \Sigma_{xf}.$$

and

$$\begin{aligned} D_x &= \int_{\Omega_y} D (\Phi_2(y))^2 dy, & \hat{D}_x &= \int_{\Omega_y} D \left(\frac{\partial \Phi_2}{\partial y}(y) \right)^2 dy, \\ \Sigma_{xa} &= \int_{\Omega_y} \Sigma_a (\Phi_2(y))^2 dy, & \Sigma_{xf} &= \int_{\Omega_y} \Sigma_f (\Phi_2(y))^2 dy, \end{aligned}$$

that can be accurately solved in a fast way using 1-dimensional solver as in [90].

Once $\Phi_1(x)$ has been obtained for a certain initial approximation $\Phi_2(y)$, $\Phi(x, y)$ is updated with the new function and it starts an iterative process, this time to update $\Phi_2(x)$ using an equation similar to (4.64) but integrating in the x -direction. This iteration follows until a convergence criterion is satisfied.

4.3.1.2 Rayleigh-Ritz Procedure

Let us assume we have a solution as (4.63), where the approximation functions F_j^i are known. A Rayleigh-Ritz procedure is used to project the eigenproblem onto a small subspace. To do this, first we have the actual solution in the form

$$\Phi^{k+1}(x, y) = \gamma_1 \Phi^k(x, y) + \gamma_2 \delta \Phi(x, y) , \quad (4.65)$$

with the normalizations

$$\begin{aligned} \int_{\Omega_x} (\Phi_1^i(x))^2 dx = 1 , \quad \int_{\Omega_y} (\Phi_2^i(y))^2 dy = 1 , \quad i = 1, \dots, k \\ \iint_{\Omega} \Phi^k \mathcal{L} \Phi^k dx dy = 1 , \quad \iint_{\Omega} \delta \Phi \mathcal{L} \delta \Phi dx dy = 1 , \end{aligned}$$

and the orthogonality condition

$$\iint_{\Omega} \Phi^k \mathcal{L} \delta \Phi dx dy = 0 , \quad (4.66)$$

where γ_j are unknown coefficients. We need to approximate the value of γ_j to obtain an approximation for the eigenvector associated with the dominant eigenvalue. In this way, we project our problem by means of

$$\lambda \iint_{\Omega} \Phi^{*k+1}(x, y) \mathcal{L} \Phi^{k+1}(x, y) dx dy = \iint_{\Omega} \Phi^{*k+1}(x, y) \mathcal{M} \Phi^{k+1}(x, y) dx dy . \quad (4.67)$$

where

$$\Phi^{*k+1}(x, y) = \gamma_1^* \Phi^k(x, y) + \gamma_2^* \delta \Phi(x, y) , \quad (4.68)$$

with γ_j^* arbitrary constants. Since the previous equality should be fulfilled for every γ_j^* , Equation (4.67) can be rewritten as the reduced generalized algebraic eigenvalue problem

$$\lambda L \vec{\gamma} = M \vec{\gamma} , \quad (4.69)$$

where

$$\begin{aligned} L &= \begin{pmatrix} \iint_{\Omega} \Phi^k \mathcal{L} \Phi^k dx dy & \iint_{\Omega} \Phi^k \mathcal{L} \delta \Phi dx dy \\ \iint_{\Omega} \delta \Phi \mathcal{L} \Phi^k dx dy & \iint_{\Omega} \delta \Phi \mathcal{L} \delta \Phi dx dy \end{pmatrix} , \\ M &= \begin{pmatrix} \iint_{\Omega} \Phi^k \mathcal{M} \Phi^k dx dy & \iint_{\Omega} \Phi^k \mathcal{M} \delta \Phi dx dy \\ \iint_{\Omega} \delta \Phi \mathcal{M} \Phi^k dx dy & \iint_{\Omega} \delta \Phi \mathcal{M} \delta \Phi dx dy \end{pmatrix} , \\ \vec{\gamma} &= (\gamma_j) , \quad j = 1, 2 . \end{aligned}$$

The dominant eigenvalue λ of problem (4.69) is our approximated dominant eigenvalue, and the eigenvector is given by (4.65).

4.3.1.3 Checking convergence

Let us define a residual operator, \mathcal{R}_2 , in a similar way as in step 2 of PGD method

$$\mathcal{R}_2(u, v) = \iint_{\Omega} \lambda u \mathcal{L}v \, dx dy - \iint_{\Omega} u \mathcal{M}v \, dx dy . \quad (4.70)$$

From the solution Φ^k at iteration k , given by Equation (4.63), we compute the residual Re_2 related to Equation (4.62):

$$Re_2 = \frac{\mathcal{R}_2(\Phi^k, \Phi^k)}{\|\Phi^k\|} , \quad (4.71)$$

and stop if $Re_2 < \varepsilon$, where ε is the desired tolerance for the residual of our approximated eigenpair.

4.3.1.4 Enriching the approximation

Now we want to improve our approximation by means of adding a correction term [91]. It is assumed that the new iteration is written as

$$\Phi^{k+1}(x, y) = \Phi^k(x, y) + \delta\Phi(x, y) . \quad (4.72)$$

where the correction term will have the form

$$\delta\Phi(x, y) = \sum_{i=1}^{n_{k+1}} \alpha_{k+1,i} \Phi_x^{k+1,i}(x) \Phi_y^{k+1,i}(y) . \quad (4.73)$$

We are interested in solving approximately the following equation,

$$\mathcal{L}(\Phi + \delta\Phi)(\lambda + \delta\lambda) = \mathcal{M}(\Phi + \delta\Phi) , \quad (4.74)$$

that can be approximated as

$$\lambda \mathcal{L}\Phi + \lambda \mathcal{L}\delta\Phi + \delta\lambda \mathcal{L}\Phi = \mathcal{M}\Phi + \mathcal{M}\delta\Phi , \quad (4.75)$$

where second order term $\delta\lambda \mathcal{L}\delta\Phi$ has been eliminated. This is, essentially, an iteration of the Newton method to compute eigenpairs for an eigenvalue problem [91]. Reordering the elements of equation (4.75) we obtain the correction equation,

$$(\lambda \mathcal{L} - \mathcal{M})\delta\Phi + \delta\lambda \mathcal{L}\Phi = -(\lambda \mathcal{L} - \mathcal{M})\Phi . \quad (4.76)$$

To solve this correction equation, a PGD based technique will be used, obtaining a correction $\delta\Phi$ composed of a sum of separable functions, as noted by equation (4.73). As in the PGD method, to solve this source problem, the initial approximation for the $\delta\Phi$ with n_{k+1} will be of the form

$$\delta\Phi(x, y) = G_1(x)G_2(y) . \quad (4.77)$$

Then the system will be weighted against a test function φ of the form

$$\varphi(x, y) = G_1(x)G_2^*(y) + G_1^*(x)G_2(y) , \quad (4.78)$$

obtaining

$$\mathcal{R}_2(\delta\Phi, \varphi) + \delta\lambda \iint \varphi(x, y)\mathcal{L}\Phi(x, y) \, dx dy = -\mathcal{R}_2(\Phi, \varphi) . \quad (4.79)$$

To solve this equation, and due to the arbitrariness of functions G_1^* and G_2^* , we obtain both equations to solve as in (4.43) and (4.44). Dealing with the first of these equations,

$$\begin{aligned} \mathcal{R}(G_1(x)G_2(y), G_1^*(x)G_2(y)) + \delta\lambda \iint_{\Omega} (\mathcal{L}\Phi(x, y)) G_1^*(x)G_2(y) \, dx \, dy \\ = -\mathcal{R}(\Phi(x, y), G_1^*(x)G_2(y)) , \end{aligned} \quad (4.80)$$

we start an iterative procedure similar to the one performed in *Step 3* of PGD method, assuming $G_2(y)$ is known. Integrating with respect to the y direction, we obtain a 1-dimensional system for the variable x , which can be discretized to obtain an algebraic linear system

$$R_{2x}\delta\Psi_x + \delta\lambda L_x\Psi_x = -R_{2x}\Psi_x , \quad (4.81)$$

where Ψ_x is a vector containing the coefficients of the finite element expansion for function Φ_x , and $\delta\Psi_x$ is a vector containing the coefficients of a finite element expansion for the correction G_1 . This system requires an extra condition to be solved because it has one more unknown than equations. We use the orthogonality condition $\delta\Psi_x^T L\Psi_x = 0$ to obtain the system

$$\begin{bmatrix} R_{2x} & L_x\Psi_x \\ (L_x\Psi_x)^T & 0 \end{bmatrix} \begin{bmatrix} \delta\Psi_x \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} -R_{2x}\Psi_x \\ 0 \end{bmatrix} . \quad (4.82)$$

Once we have the update for the correction $\delta\Psi_x$, we go back to the equation (4.76) and the system will be weighted with a similar correction term, but this time the part which depends on y will be free.

After the first term for the sum (4.73) is obtained, as in the PGD method explained at the introduction, another term is added to the sum until a convergence criterion

is satisfied for the residual of the correction term $\delta\Phi$. Then, for the correction $k+1$, it is repeated n_{k+1} times until we get a correction of the form

$$\delta\Phi(x, y) = \sum_{i=1}^{n_{k+1}} \alpha_{k+1,i} \Phi_x^{k+1,i}(x) \Phi_y^{k+1,i}(y) . \quad (4.83)$$

where $\Phi_x^{k+1,i}$ and $\Phi_y^{k+1,i}$ are the normalized G_1 and G_2 at each step, and α_i are obtained solving a reduced system similar to (4.52).

With this correction, the new function $\Phi(x, y)$ is as follows

$$\Phi(x, y) = \gamma_1 \Phi^k(x, y) + \gamma_2 \delta\Phi(x, y) = \gamma_1 \sum_{j=1}^k \left(\sum_{i=1}^{n_j} \alpha_{j,i} \Phi_x^{j,i}(x) \Phi_y^{j,i}(y) \right) + \gamma_2 \sum_{i=1}^{n_{k+1}} \alpha_{k+1,i} \Phi_x^{k+1,i}(x) \Phi_y^{k+1,i}(y) , \quad (4.84)$$

where γ_i should be calculated as in *step 1*. Also note that coefficients γ_i are used to update the value of all the $\alpha_{j,i}$ calculated before, i.e., after calculate the correction $k+1$ we get that $\alpha_{j,i} = \gamma_1 \alpha_{j,i}$, $j = 1, \dots, k$ and $\alpha_{k+1,i} = \gamma_2 \alpha_{k+1,i}$. For this reason the previous γ_i calculated are not in this formula.

4.3.2 Numerical Results

The PGD method to compute the dominant eigenvalue and its corresponding eigenfunction for a differential eigenvalue problem, presented above, has been tested with a 2-dimensional reactor core, called Biblis 2D reactor [92], in one group of energy.

The geometry of the core has been extended to be a Cartesian product of 1-dimensional spaces. This geometry together with the fuel assembly distribution is shown in Figure 4.11(a). The nuclear cross sections of the different materials composing the core are shown at Table 4.15.

Table 4.15: Cross sections of 2-dimensional Biblis problem.

Mat.	$D(cm)$	$\Sigma_a(cm^{-1})$	$\nu\Sigma_f(cm^{-1})$
1	1.4360	0.0095042	0.0058708
2	1.4366	0.0096785	0.0061908
3	1.3200	0.0026562	0.0000000
4	1.4389	0.0103630	0.0074527
5	1.4381	0.0100030	0.0061908
6	1.4385	0.0101320	0.0064285
7	1.4389	0.0101650	0.0061908
8	1.4393	0.0102940	0.0064285

The reference solution for this problem has been computed with a nodal collocation method [11] using $K = 5$ polynomials in the expansions. The reference value for the dominant eigenvalue is $\lambda_1 = 0.618502$.

The 1-dimensional systems arising as in equation (4.82) to get the correction function for x , and the analogous systems to obtain the corrections for y , are solved by means of an iterative method as the Preconditioned Conjugate Gradient method, due to the symmetry of the systems.

The convergence of the eigenvalue as a function of the steps k of the PGD method is shown in Figure 4.11(b). We observe that for this problem a small number of steps of the proposed method are enough to converge the eigenvalue.

The spatial power distribution associated with the corresponding eigenfunction is shown in Figure 4.12(a). The initial approximation obtained for the first step (as a product of two one-dimensional functions), $k = 1$, is shown in Figure 4.12(b).

In Figures 4.13(a)–4.13(d) we show the error associated and the corrections introduced by PGD method for steps $k = 2$ and $k = 3$, respectively. The convergence of this problem is also good for the eigenfunction using a small number of steps of the PGD method.

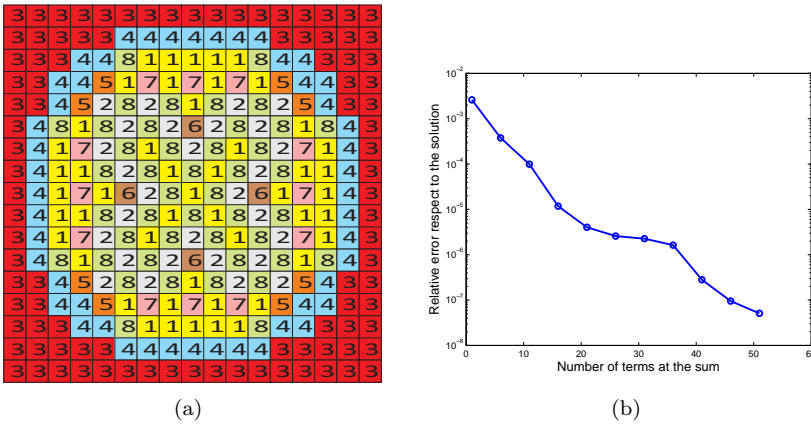


Figure 4.11: Geometry of reactor Biblis 2D (a) and Relative error evolution for the eigenvalue λ (b)

We have proposed a PGD method based technique to approximate the dominant eigenvalue and its corresponding eigenfunction of a differential eigenvalue problem defined on a separable domain has been proposed. This method is based on the representation of the eigenfunction as a sum of separable functions and the use of the Newton method to compute the eigenvalues of an eigenvalue problem. This allows to compute the multidimensional eigenvalue problem solving linear systems associated with 1-dimensional problems, which can be solved in a fast way.

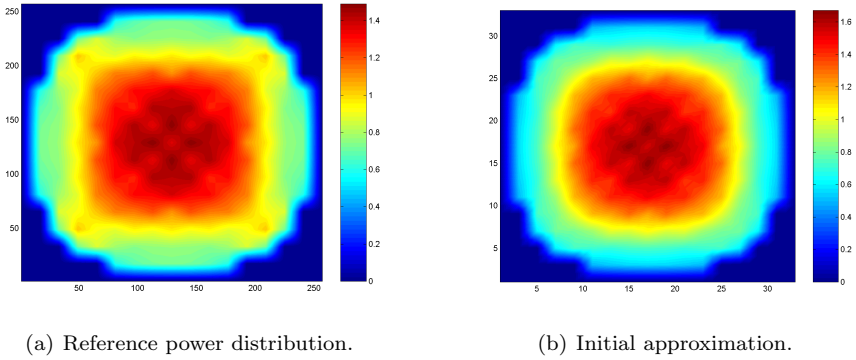


Figure 4.12: Reference solution and the initial approximation ($k = 1$) for the power distribution.

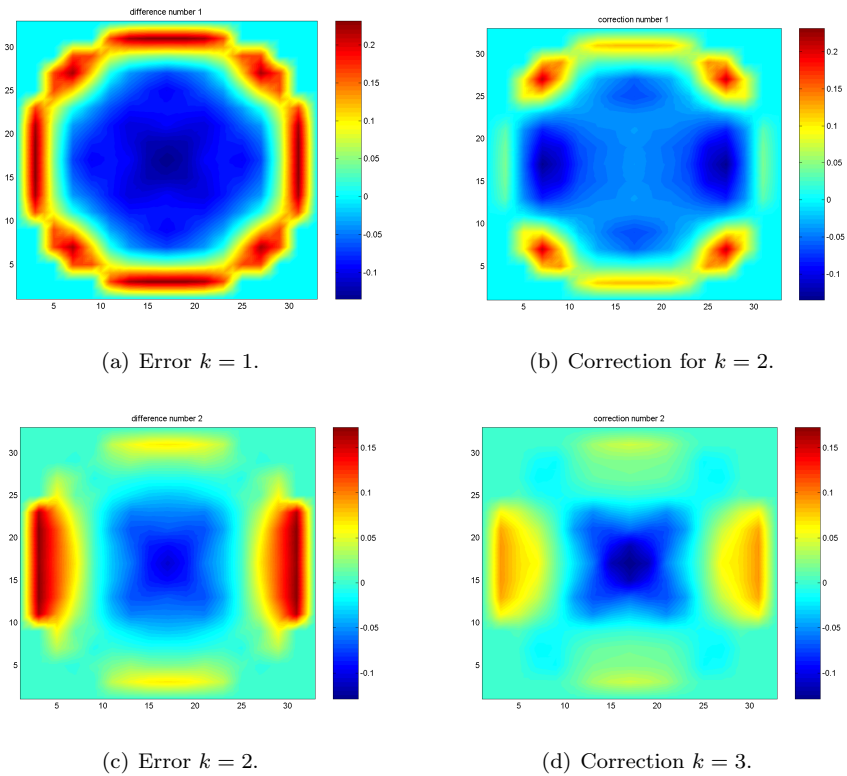


Figure 4.13: Error and corrections for the power distribution for $k = 1$ and $k = 2$

The proposed method is applied to obtain the dominant eigenvalue and its corresponding eigenfunction of a nuclear reactor core. Particularly, the Biblis 2D benchmark problem in the approximation of one group of energy has been studied, obtaining a fast convergence both for the eigenvalue and the eigenfunction. It remains for a future work to extend the formalism to be able to compute the problem in the energy multi-group approximation and to be able to study reactors in 3D geometries.

Chapter 5

Conclusions

Different numerical methods to deal with the neutron diffusion equation have been studied for reactors with hexagonal geometries. Calculations for the dominant Lambda modes of the core of a nuclear power reactor in multidimensional geometries require a large amount of memory and computing time to obtain reasonable results. This makes necessary to study different methodologies to discretize the neutron diffusion equation and to compute the dominant modes of a reactor in order to find the most efficient strategy to solve this problem.

First of all, a comparison of methods to solve the Lambda modes problem in one-dimensional geometries have been studied. All the methods considered are based on the expansion of the neutron flux in terms of a continuous polynomial basis. Different methods are obtained considering different ways to approximate the neutron diffusion equation and the continuity conditions for the neutronic current. To test the performance of the methods, we have considered two benchmark problems, a homogeneous slab with vacuum boundary conditions and a typical 1D Boiling Water Reactor (BWR). These methods are a Continuous Current Method (CCM), a Pointwise Collocation Method (PCM), and a Spectral Element Method (SEM). These methods are compared, for the first eigenvalue (the k-effective) and for the first eigenfunction, when solving different test problems, an analytical slab and a heterogeneous problem. The results are compared with an analytical solution for the slab problem, and the solutions for the heterogeneous two groups problem. All the methods show an exponential convergence rate with respect to the polynomial expansion degree.

The Spectral Element Method has shown the best results for the calculations of both the eigenvalues and the eigenvectors, being the difference higher in the 1D BWR reactor problem. Moreover, with this method we obtain symmetric diagonal

dominant matrices. This is an important fact to solve large systems using iterative methods.

Once the best method among the different options to discretize the problem in one dimension has been determined, i.e. the Spectral Element Method (SEM), it is formulated for two- and three-dimensional geometries. It has been implemented to approximate the solution of Lambda Modes problem of a nuclear reactor with hexagonal geometry. This method is based on splitting each one of the hexagonal prisms defined by the geometry into six equilateral triangular prisms, obtaining a mesh which is kept fixed. The SEM method is based on the expansion of the neutron flux in terms of a polynomials basis constructed using the modified Dubiner's Polynomials. The accuracy of the method is improved increasing the order in the polynomial expansions and it is not necessary to refine the mesh.

To test the performance of the method, an extensive analysis of 2D benchmark reactor cores has been carried out. In all the studied reactors the SEM method has provided good results for the k_{eff} and the neutron power distribution and also has succeeded computing the subcritical modes of each reactor. Also, two 3D benchmarks have been studied, the VVER-1000 and the VVER-440. The obtained results for the k_{eff} and the neutronic power distribution have been compared with the reference solution obtained with the DIF3D code. The SEM method has provided good results for the k_{eff} and the power distributions, calculating also successfully a set of subcritical modes for these reactors. The results are reported for the k_{eff} and for the power distribution, compared with the results reported in [34]. Results for a set of dominant Lambda modes are also reported for completeness.

Once the Lambda modes problem is solved, the dominant eigenvalue and its associated eigenfunction defines the neutron distribution in the steady state. This solution is used as an initial condition for any transient calculation. Then to integrate the time dependent neutron diffusion equation, an implicit difference method for the time discretization of the time dependent neutron diffusion equation is presented. This method needs to solve a system of linear equations for each time step, and the size of this system depends on the spatial accuracy needed in the calculations. The performance of the method has been tested solving the 3-D transient benchmark AER-DYN-001, showing that the proposed method provides good results when it is compared with other codes such as PARCS and DYN3D.

When nodal methods are used, the nuclear cross sections have to be homogenized over each node. This causes a problem in transients consisting on moving control rods. The easiest way to interpolate the cross sections of a partially rodded node is by means of the portion of the rod inserted on the node. This method is called volume weighted method. With the cross sections calculated by means of the volume weighted method, a unphysical behaviour of the k_{eff} appears on the calculation, leading to large errors in the approximated flux near to the tip of the control rod along the transient. This is called rod cusping effect. The correction of this effect by means of a flux weighted method is studied here. This method

has been tested for a three-dimensional transient of a small reactor, where the rod cusping effect is a dominant effect. Good results are obtained, either at the moment of a ejection control rod, or where a scram is occurring.

The use of implicit methods for the time discretization of the time dependent neutron diffusion equation implies that a large and sparse system of linear equations has to be solved for each time step. The groups of energy considered in these equations define a block structure in the matrix associated to these systems. In this way, we have studied the performance of different variational acceleration techniques applied to classical block iterative methods for the solutions of these systems of equations such as the block Jacobi and the block Gauss-Seidel methods. Also we have proposed a cheap preconditioner for the system, and the behaviour of the variational acceleration techniques is tested for the preconditioned systems. Two problems, a 2D transient and a 3D transient benchmark, are used to test the behaviour of the variational techniques. For these cases, we have observed that the most efficient acceleration technique depends on the method, and these results are also sensible to the effect of the preconditioner. As a future work, this research can be extended analysing the behaviour of the methods during a whole transient, where for each time the system is a perturbed configuration of the previous one, and the solution at the previous time step is a good initial guess for the new system. Also, parallel codes can take advantage of the block structure of the matrices, and the behaviour of the acceleration techniques on these situations might be studied.

Spatial discretization of time dependent neutron diffusion equation combined with an implicit finite difference method implies that very large systems of equations have to be solved at each time step, but there are some situations where the matrix can not be explicitly built, due to the memory requirements. Moreover, to preconditionate these systems either with classical incomplete factorizations (as the incomplete LU, or the incomplete Cholesky), or with the block preconditioning explained before are more restrictive for the memory requirements than to load the full system matrix.

Spectral preconditioners, which are based on a low rank update of a previous preconditioner improving spectral properties of the matrix have been studied here, for the use combined with a Krylov solver. These preconditioners can be applied using only a matrix-vector product procedure, with the advantage that the matrix does not need to be built explicitly, and the storage memory requirement are very low when compared with incomplete factorizations preconditioners. To study the feasibility of using this kind of preconditioner for the sequence of linear systems arising in a given transient, a transient associated to a control rod movement in a small reactor has been considered. To use the spectral update for the sequence of preconditioners it is assumed that the spectral properties of the matrix change slowly along the transient. In a first approach, different strategies for the implementation of the preconditioner have been considered and it has been observed that the preconditioner is very efficient for the reduction of iterations needed to

solve the linear systems for the first 10-15 systems and the reduction is stagnated for the rest of systems. It remains for further studies the use of this kind of preconditioners for larger problems such as the ones associated with commercial reactors, and the combination of the preconditioner with different minimal residual Krylov methods.

Also, an exponential method has been used to integrate the time dependent neutron diffusion equation. Some studies indicate that approximating the exponential matrix to integrate in time with a Krylov method, if a good preconditioner is not available, can converge faster than approximating with a Krylov method the system arising with an implicit method [67, 66], while keeping its stability conditions. Moreover, polynomials approximation as the Leja Points Method have shown good convergence properties when compared with Krylov approximation for the exponential matrix. Here a second and a fourth order exponential integrators are used to integrate the time dependent neutron diffusion equation, and different methods to approximate the matrix exponential are compared.

Starting from a steady state configuration of a nuclear power reactor some situations arise in which the reactor configuration is perturbed. The Lambda modes are eigenfunctions associated with a given configuration of the reactor, which have successfully been used to describe unstable events in BWRs with a modal method [3], which uses a set of dominant modes to expand the neutron flux. For the transient calculations using the modal method with a moderate number of modes, these modes must be updated each time step to maintain the accuracy of the solution. The updating modes process is also interesting to study perturbed configurations of a reactor. To compute several eigenvalues and its corresponding eigenfunctions for a nuclear reactor is quite expensive from the computational point of view. The Arnoldi method, based on approximating the invariant subspaces of the original problem projected onto a Krylov subspace is an efficient method to compute the dominant Lambda modes associated with a given configuration of the reactor, but if the Lambda modes have to be computed for different perturbed configurations of the reactor more efficient methods can be used.

Once an initial set of Lambda modes has been obtained, to update these modes we have used a Modified Block Newton Method, which has a local cubic convergence [81]. By studying different benchmark problems, we have shown that this method is robust to deal with problems that have degenerate or clustered eigenvalues, and it has fast convergence in such a way that the number of steps needed to update the modes is not very dependent on the kind of perturbation considered and the number of modes to be updated. For these reasons it seems to be interesting to take advantage of the convergence properties of this kind of updating methods when perturbed configurations of a reactor core have to be studied.

Once it has been shown that block Newton methods are a good alternative to update the Lambda modes of a nuclear reactor, different methods based on this strategy are studied, and their performance has been tested using a Boron injection

transient. The One Sided and Two Sided Block Newton methods have shown the best performance regarding the number of iterations. Since the cost per iteration is cheaper in the One Sided Block Newton method, this is the best method for this application.

Finally, a method based on a Proper Generalized Decomposition has been proposed to approximate the dominant eigenvalue and its corresponding eigenfunction of a differential eigenvalue problem defined on a separable domain. This method is based on the representation of the eigenfunction as a sum of separable functions and the use of the Newton method to compute the eigenvalues of an eigenvalue problem. This allows to compute the multidimensional eigenvalue problem solving linear systems associated with one-dimensional problems, which can be solved in a fast way. The proposed method is applied to obtain the dominant eigenvalue and the corresponding eigenfunction of a nuclear reactor core. Particularly, the Biblis 2D benchmark problem in the approximation of one group of energy has been studied, obtaining a fast convergence both for the eigenvalue and the eigenfunction. It remains for a future work to extend the formalism to be able to compute the problem in the energy multi-group approximation and to be able to study reactors in 3D geometries.

Appendix A

Special Functions

Contents

A.1 Legendre polynomials	172
A.2 Associated Legendre functions	173
A.3 Spherical Harmonics	175
A.4 Relations	176

Let the laplace equation in spherical coordinates be

$$\Delta y \equiv \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial y}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial y}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 y}{\partial \phi^2} = 0 ,$$

applying the separation of variables technique, with $y(r, \theta, \varphi) = R(r)Y(\theta, \varphi)$, where $Y(\theta, \varphi) = f(\theta)g(\varphi)$, the angular part of the equation becomes

$$\frac{\sin \theta}{f} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial f}{\partial \theta} \right) - l(l+1) \sin^2 \theta = -\frac{1}{g} \frac{\partial^2 g}{\partial \varphi^2} .$$

From this equation it is concluded that both parts of the equation must be equal to a constant, m^2 :

$$\frac{d^2 g}{d\varphi^2} = -m^2 g , \tag{A.1}$$

$$\frac{1}{\sin \theta} \frac{d}{d\theta} \left(\sin \theta \frac{df}{d\theta} \right) - \frac{m^2}{\sin^2 \theta} f - l(l+1)f = 0 . \tag{A.2}$$

Equation (A.1) is easily solved

$$g(\varphi) = e^{im\varphi} ,$$

and the, if the angular part of the solution for the Laplace equation in spherical coordinates, $Y(\varphi, \theta)$, has a unique value associated when φ takes the value 2π , then m must be equal to zero, or a positive or negative integer: $m = 0, \pm 1, \pm 2, \dots$. Furthermore, the equation (A.2) can be rewritten in the form of Legendre differential equation making the change $\cos \theta = x$,

$$\frac{d}{dx} (1-x^2) \frac{d}{dx} f(x) - \left(l(l+1) + \frac{m^2}{1-x^2} \right) f(x) = 0. \quad (\text{A.3})$$

It is worth to note that the differential equation (A.3) remains the same after changing x by $-x$. Thus, the solutions must be chosen as even or odd functions for x . This equation has nontrivial solution in $[-1, 1]$ if it is satisfied that $-l \leq m \leq l$.

A.1 Legendre polynomials

Before starting with equation (A.3) the solution for the ordinary Legendre equation is commented for $m^2 = 0$;

$$\frac{d}{dx} \left((1-x^2) \frac{d}{dx} f(x) \right) + l(l+1)f(x) = 0. \quad (\text{A.4})$$

The solutions of this equation are the called Legendre polynomials [93], defined by means of the following recurrence

$$(2l+1)xP_l(x) = (l+1)P_{l+1}(x) + lP_{l-1}(x), \quad l = 1, 2, \dots, \\ P_0(x) = 1, \quad P_1(x) = x.$$

Working with this recurrence it is possible to obtain a more compact representation for the Legendre polynomials, which is known as the *Rodrigues' Formula* [30]

$$P_l(x) = \frac{1}{2^l l!} \frac{d^l}{dx^l} ((x^2-1)^l).$$

The differential operator of equation (A.4) is selfadjoint. It is known that the solutions, $P_l(x)$, are orthonormal in $[-1, 1]$, i.e.,

$$\int_{-1}^1 P_l(x)P_m(x) dx = \frac{2}{2l+1} \delta_l^m,$$

where δ_l^m is the Kronecker Delta function, which takes the value 1 if $l = m$, and takes the value 0 if $l \neq m$.

A.2 Associated Legendre functions

The regular solutions for the equation (A.3), denoted as $P_l^m(x)$, are

$$P_l^m(x) = (1 - x^2)^{m/2} \frac{d^m}{dx^m} P_l(x) ,$$

and they are called associated Legendre functions, for the indices $m = 0, 1, \dots, l$, and the relation with the negative indices is given by

$$P_l^{-m}(x) = (-1)^m \frac{(l - m)!}{(l + m)!} P_l^m(x) .$$

Making use of the Rodrigues formula it is obtained

$$P_l^m(x) = \frac{(1 - x^2)^{m/2}}{2^l l!} \frac{d^{l+m}}{dx^{l+m}} ((x^2 - 1)^l) .$$

By means of the change of variables $x = \cos \theta$, the trigonometric representation of the associated Legendre functions is

$$P_l^m(\cos \theta) = \sin^m \theta \frac{d^m}{d(\cos \theta)^m} P_l(\cos \theta) .$$

The associated Legendre functions for the same index, m , are orthogonal in $[-1, 1]$ by means of the relation

$$\int_{-1}^1 P_p^m(x) P_q^m(x) dx = \frac{2}{2q + 1} \frac{(q + m)!}{(q - m)!} \delta_p^q ,$$

or in polar coordinates,

$$\int_0^\pi P_p^m(\cos \theta) P_q^m(\cos \theta) \sin \theta d\theta = \frac{2}{2q + 1} \frac{(q + m)!}{(q - m)!} \delta_p^q .$$

It is also possible to find a orthogonality relationship for the associated Legendre functions with the same subindex but different upperindex as follows,

$$\int_{-1}^1 \frac{P_l^m(x) P_l^n(x)}{1 - x^2} dx = \frac{2}{2q + 1} \frac{(l + m)!}{m(l - m)!} \delta_m^n ,$$

or in polar coordinates,

$$\int_0^\pi P_p^m(\cos \theta) P_q^m(\cos \theta) \sin \theta d\theta = \frac{2}{2q + 1} \frac{(q + m)!}{(q - m)!} \delta_p^q .$$

The associated Legendre functions satisfy different recurrence relations. Due to the existence of two indices instead of one, there is a wide variety of recurrence relations [93], as follows

$$(2l + 1)xP_l^m(x) = (l + m)P_{l-1}^m(x) + (l - m + 1)P_{l+1}^m(x) , \quad (\text{A.5})$$

$$\frac{(2l + 1)}{\sin \theta} P_l^m(\cos \theta) = P_{l+1}^{m+1}(\cos \theta) - P_{l-1}^{m+1}(\cos \theta) , \quad (\text{A.6})$$

$$\begin{aligned} \frac{(2l + 1)}{\sin \theta} P_l^m(\cos \theta) &= (l + m)(l + m - 1)P_{l+1}^{m+1}(\cos \theta) \\ &- (l - m + 1)(l - m + 2)P_{l+1}^{m+1}(\cos \theta) , \end{aligned} \quad (\text{A.7})$$

Based on Figure A.1, the Legendre polynomial of $\mu^* = \cos \theta^*$ (the cosine of the angle between $\vec{\Omega}$ and $\vec{\Omega}^*$), can be rewritten in terms of the Legendre polynomials for $\mu = \cos \theta$ and $\mu' = \cos \theta'$ due to the *Addition Theorem*

$$P_l(\mu^*) = P_l(\mu)P_l(\mu') + 2 \sum_{m=1}^l \frac{(l - m)!}{(l + m)!} P_l^m(\mu)P_l^m(\mu') \cos m(\varphi - \varphi') , \quad (\text{A.8})$$

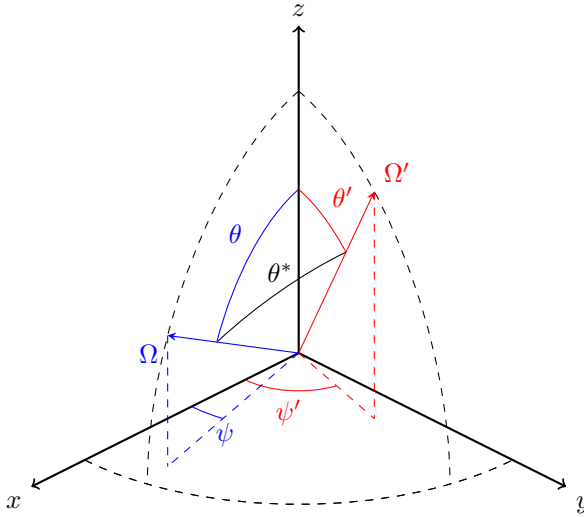


Figure A.1: Magnitudes for the Addition theorem for the Legendre polynomials.

A.3 Spherical Harmonics

The azimuthal equation (A.1) has the solution

$$g_m(\varphi) = e^{im\varphi} ,$$

with the orthogonality condition

$$\int_0^{2\pi} e^{-im_1\varphi} e^{im_2\varphi} d\varphi = 2\pi\delta_{m_1}^{m_2} .$$

Note that this is the inner product $g_{m_1}^*(\varphi)g_{m_2}(\varphi)$, where * indicates the complex conjugate function. This basis can be made orthonormal by means of the definition

$$\mathcal{G}_m(\varphi) = \frac{1}{\sqrt{2\pi}} e^{im\varphi} .$$

Then, removing the azimuthal dependence (φ), the polar dependence (θ) has the associated Legendre equation (A.2), which is satisfied by the associated Legendre functions. These functions are orthogonal, and an orthonormal formulation is obtained by means of

$$\mathcal{P}_l^m(\cos\theta) = \sqrt{\frac{2l+1}{2} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) , \quad -l \leq m \leq l .$$

The function $\mathcal{G}_m(\varphi)$ is orthonormal with respect to the azimuthal angle φ , while the function $\mathcal{P}_l^m(\cos\theta)$ is orthonormal with respect to the polar angle θ . Then, the product of both functions define the spherical harmonics

$$Y_l^m(\theta, \varphi) \equiv H_l^m P_l^m(\cos\theta) e^{im\varphi} , \quad (\text{A.9})$$

where

$$H_l^m = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} , \quad (\text{A.10})$$

when functions depending of both angles (and with two indices) are obtained, which are orthonormals over the surface defining the sphere. The orthogonality relationship for these functions is described by

$$\int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} Y_{l_1}^{m_1*}(\theta, \varphi) Y_{l_2}^{m_2}(\theta, \varphi) \sin\theta d\theta d\varphi = \delta_{l_1}^{l_2} \delta_{m_1}^{m_2} . \quad (\text{A.11})$$

A basic set of these functions is written as follows

$$Y_0^0(\theta, \varphi) = \sqrt{\frac{1}{4\pi}} , \quad (\text{A.12a})$$

$$Y_1^{-1}(\theta, \varphi) = -\sqrt{\frac{3}{8\pi}} \sin \theta (\cos \varphi - i \sin \varphi) , \quad (\text{A.12b})$$

$$Y_1^0(\theta, \varphi) = \sqrt{\frac{3}{4\pi}} \cos \theta , \quad (\text{A.12c})$$

$$Y_1^1(\theta, \varphi) = \sqrt{\frac{3}{8\pi}} \sin \theta (\cos \varphi + i \sin \varphi) . \quad (\text{A.12d})$$

A.4 Relations

Taking into account Figure A.1, and using the values obtained at the equations (A.12), the directions represented by the unit vector $\vec{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$ defined by means of the spherical harmonics as follows

$$\Omega_x \equiv \sin \theta \cos \varphi = \frac{1}{2H_1^1} (Y_1^1 - Y_1^{-1}) , \quad (\text{A.13a})$$

$$\Omega_y \equiv \sin \theta \sin \varphi = \frac{-i}{2H_1^1} (Y_1^1 + Y_1^{-1}) , \quad (\text{A.13b})$$

$$\Omega_z \equiv \cos \theta = \frac{1}{H_1^0} Y_1^0 . \quad (\text{A.13c})$$

Using the definition of the terms (A.13), and reordering the equation it is obtained that

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} &= \frac{1}{2H_1^1} (Y_1^1 - Y_1^{-1}) \frac{\partial}{\partial x} + \frac{-i}{2H_1^1} (Y_1^1 + Y_1^{-1}) \frac{\partial}{\partial y} + \frac{1}{H_1^0} Y_1^0 \frac{\partial}{\partial z} \\ &= \frac{Y_1^1(\vec{\Omega})}{2H_1^1} \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) - \frac{Y_1^{-1}(\vec{\Omega})}{2H_1^1} \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) + \frac{Y_1^0(\vec{\Omega})}{H_1^0} \frac{\partial}{\partial z} . \end{aligned} \quad (\text{A.14})$$

Now it is necessary an alternative expression of $Y_l^m(\vec{\Omega})(\vec{\Omega} \cdot \vec{\nabla})$ as a linear combination of the spherical harmonics. First of all, we deal with the term $Y_l^m(\vec{\Omega}) \frac{Y_1^1(\vec{\Omega})}{2H_1^1}$,

using the property (A.6) for the associated Legendre functions, it is obtained that

$$\begin{aligned}
\frac{Y_l^m(\vec{\Omega})Y_1^1(\vec{\Omega})}{2H_1^1} &= \frac{1}{2H_1^1} (H_l^m P_l^m(\mu)e^{i\varphi m}) (H_1^1 \sin\theta e^{i\varphi}) \\
&= \frac{H_l^m}{2} \sin\theta P_l^m(\mu)e^{i\varphi(m+1)} \\
&= \frac{H_l^m}{2} \frac{P_{l+1}^{m+1}(\mu) - P_{l-1}^{m+1}(\mu)}{2l+1} e^{i\varphi(m+1)} \\
&= \frac{H_l^m}{2(2l+1)H_{l+1}^{m+1}} Y_{l+1}^{m+1}(\vec{\Omega}) - \frac{H_l^m}{2(2l+1)H_{l-1}^{m+1}} Y_{l-1}^{m+1}(\vec{\Omega}) \\
&= \frac{1}{2(2l+1)} \left(\frac{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}}{\frac{2l+3}{4\pi} \frac{(l-m)!}{(l+m+2)!}} \right)^{\frac{1}{2}} Y_{l+1}^{m+1}(\vec{\Omega}) \\
&\quad - \frac{1}{2(2l+1)} \left(\frac{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}}{\frac{2l-1}{4\pi} \frac{(l-m-2)!}{(l+m)!}} \right)^{\frac{1}{2}} Y_{l-1}^{m+1}(\vec{\Omega}) \\
&= \frac{1}{2} \left(\frac{(l+m+2)(l+m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}} Y_{l+1}^{m+1}(\vec{\Omega}) \\
&\quad - \frac{1}{2} \left(\frac{(l-m)(l-m-1)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}} Y_{l-1}^{m+1}(\vec{\Omega}) \\
&= A_1^{l,m} Y_{l+1}^{m+1}(\vec{\Omega}) - A_2^{l,m} Y_{l-1}^{m+1}(\vec{\Omega}), \tag{A.15}
\end{aligned}$$

where the following notation has been used

$$\begin{aligned}
A_1^{l,m} &= \frac{1}{2} \left(\frac{(l+m+2)(l+m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}}, \\
A_2^{l,m} &= \frac{1}{2} \left(\frac{(l-m)(l-m-1)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}}. \tag{A.16}
\end{aligned}$$

Now, to deal with the term $Y_l^m(\vec{\Omega}) \frac{Y_1^{-1}(\vec{\Omega})}{2H_1^1}$, using the property (A.5) of the associated Legendre functions, it is obtained

$$\begin{aligned}
 \frac{Y_l^m(\vec{\Omega})Y_1^{-1}(\vec{\Omega})}{2H_1^1} &= \frac{1}{2H_1^1} (H_l^m P_l^m(\mu) e^{i\varphi m}) (H_1^1(-\sin\theta) e^{-i\varphi}) \\
 &= -\frac{H_l^m}{2} \sin\theta P_l^m(\mu) e^{i\varphi(m-1)} \\
 &= -H_l^m \frac{-(l-m+2)(l-m+1)P_{l+1}^{m-1}(\mu)}{2(2l+1)} e^{i\varphi(m-1)} \\
 &\quad -H_l^m \frac{(l+m-1)(l+m)P_{l-1}^{m-1}(\mu)}{2(2l+1)} e^{i\varphi(m-1)} \\
 &= \frac{H_l^m(l-m+2)(l-m+1)}{2(2l+1)H_{l+1}^{m-1}} Y_{l+1}^{m+1}(\vec{\Omega}) \\
 &\quad - \frac{H_l^m(l+m)(l+m-1)}{2(2l+1)H_{l-1}^{m-1}} Y_{l-1}^{m+1}(\vec{\Omega}) \\
 &= \frac{(l-m+2)(l-m+1)}{2(2l+1)} \left(\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!} \right)^{\frac{1}{2}} Y_{l+1}^{m-1}(\vec{\Omega}) \\
 &\quad - \frac{(l+m)(l+m-1)}{2(2l+1)} \left(\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!} \right)^{\frac{1}{2}} Y_{l-1}^{m-1}(\vec{\Omega}) \\
 &= \frac{1}{2} \left(\frac{(l-m+2)(l-m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}} Y_{l+1}^{m-1}(\vec{\Omega}) \\
 &\quad - \frac{1}{2} \left(\frac{(l+m)(l+m-1)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}} Y_{l-1}^{m-1}(\vec{\Omega}) \\
 &= A_3^{l,m} Y_{l+1}^{m-1}(\vec{\Omega}) - A_4^{l,m} Y_{l-1}^{m-1}(\vec{\Omega}), \tag{A.17}
 \end{aligned}$$

where the following notation has been used

$$\begin{aligned}
 A_3^{l,m} &= \frac{1}{2} \left(\frac{(l-m+2)(l-m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}}, \\
 A_4^{l,m} &= \frac{1}{2} \left(\frac{(l+m)(l+m-1)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}}. \tag{A.18}
 \end{aligned}$$

Finally, to deal with the term $Y_l^m(\vec{\Omega}) \frac{Y_1^0(\vec{\Omega})}{H_1^0}$, using the property (A.7) for the associated Legendre functions, it is obtained

$$\begin{aligned}
\frac{Y_1^0(\vec{\Omega})Y_l^m(\vec{\Omega})}{H_1^0} &= \cos \theta (H_l^m P_l^m(\mu) e^{im\varphi}) \\
&= H_l^m e^{im\varphi} \frac{(l+m)P_{l-1}^m(\mu) + (l-m+1)P_{l+1}^m(\mu)}{2l+1} \\
&= \frac{H_l^m}{H_{l-1}^m} \frac{l+m}{2l+1} Y_{l-1}^m(\vec{\Omega}) + \frac{H_l^m}{H_{l+1}^m} \frac{l-m}{2l+1} Y_{l+1}^m(\vec{\Omega}) \\
&= \frac{l+m}{2l+1} \left(\frac{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}}{\frac{2l-1}{4\pi} \frac{(l-m-1)!}{(l+m-1)!}} \right)^{\frac{1}{2}} Y_{l-1}^m(\vec{\Omega}) \\
&\quad + \frac{l-m+1}{2l+1} \left(\frac{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}}{\frac{2l+3}{4\pi} \frac{(l-m+1)!}{(l+m+1)!}} \right)^{\frac{1}{2}} Y_{l+1}^m(\vec{\Omega}) \\
&= \left(\frac{(l+m)(l-m)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}} Y_{l-1}^m(\vec{\Omega}) \\
&\quad + \left(\frac{(l+m+1)(l-m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}} Y_{l+1}^m(\vec{\Omega}) \\
&= A_5^{l,m} Y_{l-1}^m(\vec{\Omega}) + A_6^{l,m} Y_{l+1}^m(\vec{\Omega}), \tag{A.19}
\end{aligned}$$

where the following notation has been used

$$\begin{aligned}
A_5^{l,m} &= \left(\frac{(l+m)(l-m)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}}, \\
A_6^{l,m} &= \left(\frac{(l+m+1)(l-m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}}. \tag{A.20}
\end{aligned}$$

Thus, multiplying the equation (A.14) by $Y_l^m(\vec{\Omega})$, and substituting by means of the equalities (A.15), (A.16) and (A.18), the following identity is obtained

$$\begin{aligned}
Y_l^m(\vec{\Omega})(\vec{\Omega} \cdot \vec{\nabla}) &= \left(A_1^{l,m} Y_{l+1}^{m+1}(\vec{\Omega}) - A_2^{l,m} Y_{l-1}^{m+1}(\vec{\Omega}) \right) \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) \\
&\quad + \left(-A_3^{l,m} Y_{l+1}^{m-1}(\vec{\Omega}) + A_4^{l,m} Y_{l-1}^{m-1}(\vec{\Omega}) \right) \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) \\
&\quad + \left(A_5^{l,m} Y_{l-1}^m(\vec{\Omega}) + A_6^{l,m} Y_{l+1}^m(\vec{\Omega}) \right) \frac{\partial}{\partial z}, \tag{A.21}
\end{aligned}$$

where using the information of the identities (A.16), (A.18) and (A.20), it is obtained that

$$\begin{aligned}
 A_1^{l,m} &= \frac{1}{2} \left(\frac{(l+m+2)(l+m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}}, \\
 A_2^{l,m} &= \frac{1}{2} \left(\frac{(l-m)(l-m-1)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}}, \\
 A_3^{l,m} &= \frac{1}{2} \left(\frac{(l-m+2)(l-m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}}, \\
 A_4^{l,m} &= \frac{1}{2} \left(\frac{(l+m)(l+m-1)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}}, \\
 A_5^{l,m} &= \left(\frac{(l+m)(l-m)}{(2l+1)(2l-1)} \right)^{\frac{1}{2}}, \\
 A_6^{l,m} &= \left(\frac{(l+m+1)(l-m+1)}{(2l+3)(2l+1)} \right)^{\frac{1}{2}}. \tag{A.22}
 \end{aligned}$$

Appendix B

Evaluation of the polynomials and the integrals

Contents

B.1 Evaluation of the modified Dubiner's polynomials . . .	181
B.2 Evaluation of the integrals by means of quadrature rules	182

B.1 Evaluation of the modified Dubiner's polynomials

To evaluate the modified Dubiner's polynomials we have used the following recurrence relations:

$$\begin{aligned} Q_0^{1,1}(x', y') &= 1, & Q_1^{1,1}(x', y') &= 2x' + y' - 1, \\ Q_{i+1}^{1,1}(x', y') &= \frac{1}{a_{1i}} \left(a_{3i}(2x' + y' - 1)Q_i^{1,1}(x', y') \right. \\ &\quad \left. - a_{4i}(1 - y')^2 Q_{i-1}^{1,1}(x', y') \right), \quad i \geq 2, \end{aligned} \tag{B.1}$$

where

$$\begin{aligned} a_{1i} &= 4(i+1)^2(i+3), \\ a_{3i} &= 4(i+1)(i+2)(2i+3), \\ a_{4i} &= 4(i+1)^2(i+2), \end{aligned}$$

and

$$P_0^{\alpha,\beta}(x') = 1, \quad P_1^{\alpha,\beta}(x') = \frac{1}{2}(\alpha - \beta + (\alpha + \beta + 2)x'),$$

$$P_{j+1}^{\alpha,\beta}(x') = \frac{1}{b_{1j}} \left((b_{2j} + b_{3j}x') P_j^{\alpha,\beta}(x') - b_{4j} P_{j-1}^{\alpha,\beta}(x') \right), \quad (\text{B.2})$$

where

$$\begin{aligned} b_{1j} &= 2(j+1)(j+\alpha+\beta+1)(2j+\alpha+\beta), \\ b_{2j} &= (2j+\alpha+\beta+1)(\alpha^2 - \beta^2), \\ b_{3j} &= (2j+\alpha+\beta)(2j+\alpha+\beta+1)(2j+\alpha+\beta+2), \\ b_{4j} &= 2(j+\alpha)(j+\beta)(2j+\alpha+\beta+2). \end{aligned}$$

The derivatives of the polynomials can be evaluated by means of the recurrence relations obtained deriving the relations (B.1) and (B.2).

B.2 Evaluation of the integrals by means of quadrature rules

The first step to calculate the integrals appearing in equations (2.51), (2.52), (2.53) and (2.54) is to map each physical element, Ω_e , into the reference element, Ω_{REF} , by means of the change of variables (2.46).

It is easy to see that the Jacobian and the differential area of this change of variables for each element are

$$|J| = \left| \frac{\partial(x, y)}{\partial(x', y')} \right| = -x_1 y_0 + x_2 y_0 + x_0 y_1 - x_2 y_1 - x_0 y_2 + x_1 y_2 = C,$$

$$dx \, dy = C \, dx' \, dy',$$

where C is a constant depending on the vertex coordinates of each element.

We have to calculate three kinds of integrals, and each one has a particular form. Thus, we distinguish three cases.

case 1) To evaluate an integral of the form

$$\iint_{\Omega_e} \vec{\nabla} h_1(x, y) \cdot \vec{\nabla} h_2(x, y) \, dx \, dy,$$

by means of the change of variables (2.45) we get

$$\begin{aligned} \iint_{\Omega_e} \vec{\nabla} h_1(x, y) \cdot \vec{\nabla} h_2(x, y) \, dx \, dy &= \iint_{\Omega_{REF}} \left((a_1^2 + a_2^2) \frac{\partial h_1}{\partial x'} \frac{\partial h_2}{\partial x'} \right. \\ &+ (a_1 a_2 + b_1 b_2) \left(\frac{\partial h_1}{\partial x'} \frac{\partial h_2}{\partial y'} + \frac{\partial h_2}{\partial x'} \frac{\partial h_1}{\partial y'} \right) + (b_1^2 + b_2^2) \frac{\partial h_1}{\partial y'} \frac{\partial h_2}{\partial y'} \Big) C \, dx' \, dy', \end{aligned}$$

where each one of the terms inside the integral are polynomials.

case 2) To evaluate an integral of the form

$$\iint_{\Omega_e} h_1(x, y)h_2(x, y) dx dy ,$$

by means of the same change of variables used above, we obtain the following integral

$$\iint_{\Omega_{REF}} h_1(x', y')h_2(x', y') C dx' dy' .$$

case3) Finally, if we deal with a line integral of the form

$$\int_{\Gamma} h_1(x(l), y(l))h_2(x(l), y(l)) dl ,$$

where the parameterization of the polygonal is defined on the domain $[-1, 1]$, then we have that the product inside the integral is a product of two polynomials. For this reason we can express this integral as the integral of a polynomial on the domain $[-1, 1]$.

For **case 1)** and **case 2)** we obtain integrals of the form

$$I = \iint_{\Omega_{REF}} f(x', y') dx' dy' , \tag{B.3}$$

where $f(x', y')$ is a two dimensional polynomial over the reference domain Ω_{REF} .

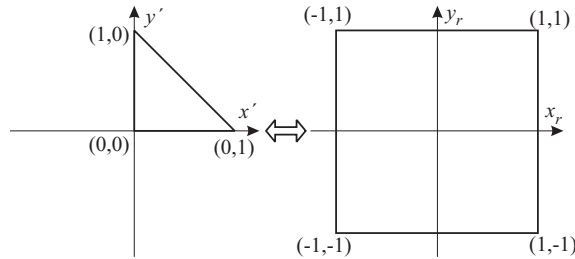


Figure B.1: Change of variables mapping the reference domain into the rectangle $[-1, 1] \times [-1, 1]$.

To evaluate these integrals, we make use of the change of variables (B.4).

$$\begin{aligned} x' &= \frac{1}{4}(1+x_r)(1-y_r) \\ y' &= \frac{1}{2}(y_r+1) \end{aligned} \iff \begin{aligned} x_r &= \frac{2x'}{1-y'} - 1 \\ y_r &= 2y' - 1 \end{aligned} , \tag{B.4}$$

As it is shown in Figure B.1, this change of variables transforms the reference domain, Ω_{REF} , into the rectangle $[-1, 1] \times [-1, 1]$. The Jacobian and the differential area for this new change of variables are given by

$$|J| = \left| \frac{\partial(x'y')}{\partial(x_r, y_r)} \right| = \frac{1}{8}(1 - y_r) , \quad dx' dy' = \frac{1}{8}(1 - y_r) dx_r dy_r .$$

Thus, we obtain

$$\begin{aligned} I &= \iint_{\Omega_{REF}} f(x', y') dx' dy' \\ &= \int_{-1}^1 \int_{-1}^1 f\left(\frac{(1+x_r)(1-y_r)}{4}, \frac{(y_r+1)}{2}\right) \left(\frac{1-y_r}{8}\right) dx_r dy_r , \end{aligned} \quad (\text{B.5})$$

which can be computed calculating two one-dimensional integrals over $[-1, 1]$. As the integrands are polynomials, these integrals can be computed exactly making use, for example, of the Gauss-Legendre quadrature rules [32, 94]. From equations (B.5), we can write

$$I = \sum_{i=1}^{s_y} \sum_{j=1}^{s_x} \left(\frac{1-y_{ri}}{8}\right) w_i w_j f\left(\frac{(1+x_{rj})(1-y_{ri})}{4}, \frac{(y_{ri}+1)}{2}\right) \quad (\text{B.6})$$

where x_{rj} , y_{ri} are Gaussian quadrature points for the directions x_r and y_r , and the corresponding weights are w_j and w_i . We can calculate this formula by means of

$$I = \sum_{i=1}^{s_y} \sum_{j=1}^{s_x} c_{ij} f(x'_{ij}, y'_{ij})$$

where, c_{ij} , x'_{ij} and y'_{ij} can be obtained from the relations

$$c_{ij} = \frac{(1-y_{rj})}{8} w_i w_j , \quad x'_{ij} = \frac{(1+x_{rj})(1-y_{ri})}{4} , \quad y'_{ij} = \frac{(y_{rj}+1)}{2} ,$$

and the value of s_x and s_y depend on the degree of the polynomials constituting the integrands.

Bibliography

- [1] S. Glasstone and A. Sesonske. *Ingeniería de Reactores Nucleares*. Ed. Reverté, 1975.
- [2] G. Verdú, D. Ginestar, V. Vidal, and J. L. Muñoz-Cobo. A consistent multidimensional nodal method for transient calculations. *Ann. Nucl. Energy*, 22:395–410, 1995.
- [3] R. Miró, D. Ginestar, G. Verdú, and D. Hennig. A nodal modal method for the neutron diffusion equation. application to BWR instabilities analysis. *Annals of Nuclear Energy*, 29(10):1171–1194, 2002.
- [4] V. Vidal, G. Verdú, D. Ginestar, and J. L. Muñoz-Cobo. Eigenvalues calculation algorithms for λ -modes determination. parallelization approach. *Ann. Nucl. Energy*, 24:387–410, 1997.
- [5] G. Verdú, R. Miró, D. Ginestar, and V. Vidal. The implicit restarted Arnoldi method, an efficient alternative to solve the neutron diffusion equation. *Ann. Nucl. Energy*, 26(7):579–593, 1999.
- [6] G. Verdú, D. Ginestar, R. Miró, and V. Vidal. Using the Jacobi–Davidson method to obtain the dominant Lambda modes of a nuclear power reactor. *Ann. Nucl. Energy*, 32:1274–1296, 2005.
- [7] Weston M. Stacey. *Space-Time Nuclear Reactor Kinetics*. Academic Press, 1969.
- [8] Weston M. Stacey. *Nuclear Reactor Physics*. John Wiley & Sons Inc, 2001.
- [9] Allan F. Henry. *Nuclear-Reactor Analysis*. The MIT Press, fourth edition, 1986.
- [10] A. Hébert. Development of the nodal collocation method for solving the neutron diffusion equation. *Annals of Nuclear Energy*, 14:527–541, 1987.
- [11] G. Verdú, D. Ginestar, V. Vidal, and J.L. Muñoz-Cobo. 3D λ -modes of the neutron-diffusion equation. *Annals of Nuclear Energy*, 21(7):405–421, 1994.

- [12] Nuclear Energy Institute. Source book: Soviet-designed nuclear power plants in Russia, Ukraine, Lithuania, Armenia, the Czech Republic, the Slovak Republic, Hungary and Bulgaria. Technical report, Nuclear Energy Institute, 1997.
- [13] Yu. G. Dragunov, S. B. Ryzhov, V. P. Denisov, and V. A. Mokhov. Prospects for development of VVER-type pressurized light-water reactor installations. *Thermal Engineering*, 54:343–347, 2007.
- [14] I. N. Vasil’chenko, S. N. Kobelev, V. V. V’yalitsyn, and D. V. Mal’chevskii. Succession in choosing structural solutions for a VVER-1500 core. *Atomic Energy*, 99:412–416, 2005.
- [15] B. A. Finlayson and L. E. Scriven. The method of weighted residuals – a review. *Applied Mechanics Reviews*, 19:735–748, 1966.
- [16] J. C. Slater. Electronic energy bands in metal. *Phys. Rev.*, 45:794–801, 1934.
- [17] L. V. Kantorovic. On a new method of approximate solution of partial differential equations. *Dokl. Akad. Nauk SSSR*, 4:532–536, 1934.
- [18] R. A. Frazer, W. P. Jones, and S. W. Skan. *Approximation to Functions and to the Solution of Differential Equations*. R & M 1799, Aeronautical Research Council, London, 1937.
- [19] C. Lanczos. Trigonometric interpolation of empirical and analytical functions. *J. Math. Phys.*, 17:123–199, 1938.
- [20] H.O. Kreiss and J. Olinger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24:199–215, 1972.
- [21] S. A. Orszag. Comparison of pseudospectral and spectral approximations. *Stud. Appl. Math.*, 51:253–259, 1972.
- [22] S. A. Orszag. Numerical methods for the simulation of turbulence. *Phys. Fluids*, 12(Suppl. II):250–257, 1969.
- [23] E. Eliassen, B. Machenhauer, and E. Rasmussen. On a numerical method for integration of the hydrodynamical equations with a spectral representation of the horizontal fields. Technical report, Institut for Teoretisk Meteorologi, Univ. Copenhagen, 1970.
- [24] D. Gottlieb and S. A. Orszag. Numerical analysis of spectral methods: Theory and applications. In *SIAM-CBMS*, Philadelphia, 1977.
- [25] R. G. Voigt, D. Gottlieb, and M. Y. Hussaini. Spectral methods for partial differential equations. In *SIAM-CBMS*, Philadelphia, 1984.
- [26] S. A. Orszag. Spectral methods for problems in complex geometries. *J. Comput. Phys.*, 37:70–92, 1980.

-
- [27] Y. Morchoisne. *Spectral Methods for Partial Differential Equations*, chapter Inhomogeneous Flow Calculations by Spectral Methods: Mono-Domain and Multi-Domain Techniques, pages 181–208. SIAM-CBMS, 1984.
- [28] A. T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.*, 54:468–488, 1984.
- [29] R. Pasquetti and F. Rapetti. Spectral element methods on triangles and quadrilaterals: comparisons and applications. *Journal of Computational Physics*, 198:349–362, 2004.
- [30] M. A. Abramowitz and I. A. Stegun. *Handbook of mathematical functions: With formulas, graphs and mathematical tables*. Dover, New York, 1972.
- [31] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6:345–390, 1991.
- [32] S. J. Sherwin and G. E. Karniadakis. A triangular spectral element method: applications to the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 123:189–229, 1995.
- [33] M. A. Taylor, B. A. Wingate, and B. E. Vincent. An algorithm for computing Fekete points in the triangle. *SIAM Journal on Numerical Analysis*, 38:1707–1720, 2000.
- [34] Y.A. Chao and Y.A. Shatilla. Conformal mapping and hexagonal nodal methods-II: Implementation in the ANC-H Code. *Nuclear Science and Engineering*, 121:210–225, 1995.
- [35] M. Capilla, C. F. Talavera, D. Ginestar, and G. Verdú. A nodal collocation method for the calculation of the Lambda modes of the PL equations. *Ann. Nucl. Energy*, 32:1825–1853, 2005.
- [36] G. EM. Karniadakis and Spencer J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2005.
- [37] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 77, The Art of Scientific Computing*. Cambridge University Press, Cambridge, second edition edition, 1992.
- [38] A. Hébert. Application of a dual variational formulation to finite element reactor calculations. *Ann. Nucl. Energy*, 20:823–845, 1993.
- [39] D. W. Niggs. *One Dimensional Nodal neutronics Routines for TRAC-BD1 Thermal-Hydraulics Program*. EG&G, 1984.
- [40] J. L. Muñoz-Cobo, G. Verdú, C. Pereira, A. Escrivá, J. Ródenas, F. Castillo, and J. Serra. Consistent generation and functionalization of one-dimensional cross-sections for TRAC-BF1. *Nucl. Technol.*, 107:125–137, 1994.

- [41] K. Sing and V. Kumar. Solution of the multigroup diffusion equation in hex-z geometry by finite Fourier transform. *Annals of Nuclear Energy*, 20:153–161, 1993.
- [42] J. Y. Cho and C. H. Kim. Higher order polynomial expansion nodal method for hexagonal core neutronics analysis. *Annals of Nuclear Energy*, 25:1021–1031, 1998.
- [43] Alain Hébert. A Raviart-Thomas-Schneider solution of the diffusion equation in hexagonal geometry. *Annals of Nuclear Energy*, 35(3):363–376, 2008.
- [44] K. L. Derstine. Dif3d: A code to solve one-, two-, and three-dimensional finite-difference diffusion theory problems. Technical report, Argonne National Laboratory, Argonne, IL, 1984.
- [45] T. Barrachina, D. Ginestar, and G. Verdú. Lambda modes of the neutron diffusion equation in hexagonal geometry. In *Proceedings of PHYSOR-2006*, 2006.
- [46] S. J. Sherwin and G. E. Karniadakis. A new triangular and tetrahedral basis for high-order (hp) finite element methods. *International Journal for Numerical Methods in Engineering*, 38:3775–3802, 1995.
- [47] W. Heinrichs and B. I. Loch. Spectral schemes on triangular elements. *J. Comp. Physics*, 173:279–301, 2001.
- [48] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.
- [49] Michele Benzi. Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, 182(2):418 – 477, 2002.
- [50] F. Bennewitz, H. Finnemann, and M.R. Wagner. Higher-order corrections in nodal reactor computations. *Trans. Am. Nucl. Soc.*, 22:205, 1975.
- [51] Nuclear Energy Agency. Advanced thermal-hydraulic and neutronic codes: Current and future applications, summary and conclusion. In *CSNI Workshop NEA/CSNI/R(2001)9*, Barcelona, Spain, 2001.
- [52] S. González-Pintor, D. Ginestar, and G. Verdú. High order finite element method for the lambda modes problem on hexagonal geometry. *Annals of Nuclear Energy*, 36(9):1450–1462, 2009.
- [53] D. Ginestar, G. Verdú, V. Vidal, R. Bru, J. Marín, and J. L. Muñoz-Cobo. High order backward discretization of the neutron diffusion equation. *Annals of Nuclear Energy*, 25(1-3):47–64, 1998.
- [54] A. Dall’Osso. Reducing rod cusping effect in nodal expansion method calculations. In *Proceedings of PHYSOR-2002*, 2002.

-
- [55] D. Gilbert, J.E. Roman, Wm. J. Garland, and W.F.S. Poehlman. Simulating control rod and fuel assembly motion using moving meshes. *Annals of Nuclear Energy*, 35(2):291 – 303, 2008.
- [56] Juan-Andrés Lozano, Nuria García-Herranz, Carol Ahnert, and José-María Aragonés. The analytic nodal diffusion solver andes in multigroups for 3d rectangular geometry: Development and performance analysis. *Annals of Nuclear Energy*, 35(12):2365 – 2374, 2008.
- [57] A. Keresztúri and M. Telbisz. A three dimensional hexagonal kinetic benchmark problem. In *Proceedings of 2nd AER Symposium*, 1992.
- [58] K. Obaidurrahman, J.D. Doshi, R.P. Jain, and V. Jagannathan. Development and validation of coupled dynamics code 'TRIKIN' for VVER reactors. *Nuclear Engineering and Technology*, 42(3):259–270, 2010.
- [59] Magnus R Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems 1. *Journal Of Research Of The National Bureau Of Standards*, 49(6):409–436, 1952.
- [60] H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.
- [61] S. González-Pintor, D. Ginestar, and G. Verdú. Time integration of the neutron diffusion equation on hexagonal geometries. *Mathematical and Computer Modelling*, 52(7-8):1203–1210, 2010.
- [62] A. Keresztúri, Gy. Hegyi, Cs. Marázcy, I. Panka, M. Telbisz, I. Trosztel, and Cs. Hegedüs. Development and validation of the three-dimensional dynamic code-kiko3d. *Annals of Nuclear Energy*, 30(1):93–120, 2003.
- [63] L. Giraud, S. Gratton, and E. Martin. Incremental spectral preconditioners for sequences of linear systems. *Applied Numerical Mathematics*, 57(11–12):1164 – 1180, 2007.
- [64] Ronald B. Morgan. GMRES with deflated restarting. *SIAM Journal on Scientific Computing*, 24(1):20–37, 2002.
- [65] T. Downar, V. Seker Y. Xu, and A. Ward. Parcs: Purdue advance reactor core simulator. In *Proceedings of ANS Reactor Physics Topical Meeting PHYSOR 6*, 2006.
- [66] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.
- [67] Marlis Hochbruck, Christian Lubich, and Hubert Selhofer. Exponential integrators for large systems of differential equations. *SIAM Journal on Scientific Computing*, 19(5):1552–1574, 1998.

- [68] L. Bergamaschi, M. Caliari, and M. Vianello. Efficient approximation of the exponential operator for discrete 2d advection-diffusion problems. *Numerical Linear Algebra with Applications*, 10(3):271–289, 2003.
- [69] L. Bergamaschi and M. Vianello. Efficient computation of the exponential operator for large, sparse, symmetric matrices. *Numerical Linear Algebra with Applications*, 7(1):27–45, 2000.
- [70] M. A. Botchev. Residual, restarting and Richardson iteration for the matrix exponential, revised. *ArXiv e-prints*, December 2011.
- [71] S. Blanes and P.C. Moan. Fourth- and sixth-order commutator-free magnus integrators for linear and non-linear dynamical systems. *Applied Numerical Mathematics*, 56(12):1519–1537, 2006.
- [72] S. Blanes, F. Casas, J.A. Oteo, and J. Ros. The magnus expansion and some of its applications. *Physics Reports*, 470(5–6):151 – 238, 2009.
- [73] S. Blanes, F. Casas, J. A. Oteo, and J. Ros. A pedagogical approach to the Magnus expansion. *European Journal of Physics*, 31:907–918, July 2010.
- [74] Wilhelm Magnus. On the exponential solution of differential equations for a linear operator. *Communications on Pure and Applied Mathematics*, 7(4):649–673, 1954.
- [75] R. M. Wilcox. Exponential Operators and Parameter Differentiation in Quantum Physics. *Journal of Mathematical Physics*, 8:962–982, April 1967.
- [76] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.
- [77] Michele Benzi and Nader Razouk. Decay bounds and $\mathcal{O}(n)$ algorithms for approximating functions of sparse matrices. *Electronic Transactions on Numerical Analysis*, 28:16–39, 2007.
- [78] M. Caliari, M. Vianello, and L. Bergamaschi. Interpolating discrete advection-diffusion propagators at leja sequences. *Journal of Computational and Applied Mathematics*, 172(1):79–99, 2004.
- [79] M. Caliari. Accurate evaluation of divided differences for polynomial interpolation of exponential propagators. *Computing (Vienna/New York)*, 80(2):189–201, 2007.
- [80] A. C. Hindmarsh. ODEPACK, a systematized collection of ode solvers. In R.S. et al. Stepleman, editor, *Scientific Computing*, pages 55–64, Amsterdam, Netherlands; New York, U.S.A., 1983. North-Holland.

-
- [81] Ralf Lösche, Hubert Schwetlick, and Gisela Timmermann. A modified block newton iteration for approximating an invariant subspace of a symmetric matrix. *Linear Algebra and its Applications*, 275-276:381–400, 1998.
- [82] Argonne National Laboratory. *Benchmark Problem Book, ANL-7416, Suppl. 2*. Mathematics and Computers, 1977.
- [83] Hernan Tinoco, Przemyslaw Buchwald, and Wiktor Frid. Numerical simulation of boron injection in a BWR. *Nuclear Engineering and Design*, 240(2):221–234, 2010.
- [84] G.W. Stewart. An updating algorithm for subspace tracking. *Signal Processing, IEEE Transactions on*, 40(6):1535–1541, jun 1992.
- [85] D. C. Sorensen. Implicit application of polynomial filters in a k -step arnoldi method. *SIAM Journal on Matrix Analysis and Applications*, 13(1):357–385, 1992.
- [86] V. Simoncini. On the numerical solution of $ax - xb = c$. *BIT Numerical Mathematics*, 36:814–830, 1996. 10.1007/BF01733793.
- [87] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics*, 139(3):153–176, 2006.
- [88] David González, Amine Ammar, Francisco Chinesta, and Elías Cueto. Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering*, 81(5):637–659, 2010.
- [89] A. F. Henry. *Nuclear reactor analysis*. The MIT Press, 1975.
- [90] S. González-Pintor, D. Ginestar, and G. Verdú. Continuous pseudospectral methods for the neutron diffusion equation in 1d geometries. *Mathematical and Computer Modelling*, 50(5-6):783–793, 2009.
- [91] S. González-Pintor, D. Ginestar, and G. Verdú. Updating the lambda modes of a nuclear power reactor. *Mathematical and Computer Modelling*, 54(7–8):1796 – 1801, 2011.
- [92] A. Hébert. Application of the hermite method for finite element reactor calculations. *Nuclear Science and Engineering*, 91:34–58, 1985.
- [93] G. B. Arfken and H. J. Weber. *Mathematical Methods for Physicists*. Elsevier Academic Press, sixth edition, 2005.
- [94] H. T. Rathod, K. V. Nagaraja, and B. Venkatesusu. Symetric Gauss Legendre quadrature formulas for composite numerical integration over a triangular surface. *Applied Mathematics and Computation*, 188:865–876, 2007.

