MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

AN ALGORITHM FOR MULTI–GROUP TWO–DIMENSIONAL NEUTRON DIFFUSION KINETICS IN NUCLEAR REACTOR CORES

por

Marcelo Schramm

Tese para obtenção do Título de
Doutor em Engenharia

Porto Alegre, Abril de 2016

AN ALGORITHM FOR MULTI–GROUP TWO–DIMENSIONAL NEUTRON DIFFUSION KINETICS IN NUCLEAR REACTOR CORES

por

Marcelo Schramm

Mestre em Engenharia

Tese submetida ao Corpo Docente do Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Doutor em Engenharia

Área de Concentração: Fenômenos de Transporte

Orientador: Prof. Dr. Marco Tullio Menna Barreto de Vilhena

Co-Orientador: Prof. Dr. Antonio Carlos Marques Alvim

Aprovada por:

Prof. Dr. Fernando Carvalho da Silva (UFRJ/RJ)

Prof. Dr. Ricardo Barros de Carvalho (UERJ/RJ)

Prof. Dr. Glênio Gonçalves (UFPEL/RS)

Prof. Dr. Volnei Borges (PROMEC–UFRGS/RS)

Prof. Dr. Luiz Alberto Oliveira Rocha

Coordenador do PROMEC

Porto Alegre, 29 de Abril de 2016

AGRADECIMENTOS

O autor é grato

à CAPES, pelo suporte financeiro;

ao PROMEC, pela oportunidade;

aos professores Marco Vilhena, Antonio Alvim, Cynthia Segatto, Bardo Bodmann e Claudio Petersen pelas orientações na pesquisa ao longo dos anos;

e especialmente aos colegas do GENUC e externos.

RESUMO

O objetivo desta tese é introduzir uma nova metodologia para a cinética bidimensional multigrupo de difusão de nêutrons em reatores nucleares. A metodologia apresentada usa uma aproximação polinomial em um domínio homogêneo retangular com condições de contornos não homogêneas. Como ela consiste em uma série de Taylor truncada, sua estimativa de erro varia de acordo com o tamanho do retângulo. Os coeficientes são obtidos principalmente pelas suas relações com o termo independente, que é determinado pela equação diferencial. Estas relações são obtidas apenas pelas condições de contorno, e é demonstrado serem linearmente independentes. Um esquema numérico é feito para assegurar uma rápida convergência. Estes procedimentos feitos para um retângulo homogêneo são feitos para construir soluções para problemas de autovalor e dependentes do tempo de geometria ortogonal global com parâmetros seccionalmente constantes pelo método iterativo SOR. O autovalor dominante e sua autofunção são obtidos pelo método da potência no problema de autovalor. A solução para casos dependentes do tempo usam o método de Euler modificado na variável tempo. Quatro casos–teste clássicos são considerados para ilustração.

Palavras–chave: difusão de nêutrons, métodos numéricos, controle de erro.

ABSTRACT

The objective of this thesis is to introduce a new methodology for two–dimensional multi–group neutron diffusion kinetics in a reactor core. The presented methodology uses a polynomial approximation in a rectangular homogeneous domain with non–homogeneous boundary conditions. As it consists on a truncated Taylor series, its error estimates varies with the size of the rectangle. The coefficients are obtained mainly by their relations with the independent term, which is determined by the differential equation. These relations are obtained by the boundary conditions only, and these relations are proven linear independent. A numerical scheme is made to assure faster convergence. The procedures done for one homogeneous rectangle are used to construct the solution of global orthogonal geometry with step–wise constant parameters steady state and time dependent problems by the iterative SOR algorithm. The dominant eigenvalue and its eigenfunction are obtained by the power method in the eigenvalue problem. The solution for the time dependent cases uses the modified Euler method in the time variable. Four classic test cases are considered for illustration.

Keywords: neutron diffusion, numerical methods, error control.

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADS	Accelerator–driven system
UFRGS	Universidade Federal do Rio Grande do Sul
UERJ	Universidade Estadual do Rio de Janeiro
UFRJ	Universidade Federal do Rio de Janeiro
UFPel	Universidade Federal de Pelotas
PROMEC	Programa de Pós–Graduação em Engenharia Mecânica
RS	Rio Grande do Sul
RJ	Rio de Janeiro
SOR	Successive over–relaxation
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
PEM	Polynomial expansion methodology

LIST OF SYMBOLS

1. Arabic Characters

0	(subscript) indicates initial condition
A	local boundary condition coefficient of $\hat{\mathbf{n}} \cdot \nabla\phi$ $[cm]$
$\mathfrak{A}$	global boundary condition coefficient of $\hat{\mathbf{n}} \cdot \nabla\phi$ $[cm]$
B	local boundary condition coefficient of ϕ
$\mathfrak{B}$	global boundary condition coefficient of ϕ
B^2	z axis geometric buckling $[cm^{-2}]$
C	precursors concentration $[cm^{-3}]$
$\mathfrak{C}$	global boundary condition independent term $[cm^{-2}s^{-1}]$
D	Diffusion coefficient $[cm]$
$\mathfrak{D}$	The two-dimensional domain
E	(subscript) indicates East orientation
f	local boundary condition coefficient for polynomial independent term $[cm^{-2}s^{-1}, cm^{-3}s^{-1}, cm^{-4}s^{-1}]$
g	(subscript) indicates energy group
G	total number of neutron fluxes energy groups
$[j]$	(superscript) indicates global given region
k	dominant eigenvalue
m	(subscript) indicates integer power of x
n	(subscript) indicates integer power of y
$\hat{\mathbf{n}}$	unit vector normal to a surface
N	(subscript) indicates North orientation
(old)	(superscript) indicates older value of an unknown
p	(subscript) indicates precursors group
P	total number of delayed neutrons precursors concentrations groups
$\mathfrak{P}$	number of neutrons generated by fission
$\mathbf{r}$	position vector: $x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}}$ $[cm]$
$[r]$	(superscript) indicates segment
R	total number of segments (regions)

LIST OF SYMBOLS

1. Arabic Characters (continuation)

S	(subscript) indicates South orientation
$\mathfrak{S}$	external source term $[cm^{-3}s^{-1}]$
t	time $[s]$
T	time limit $[s]$
v	neutron speed $[cm/s]$
V	the domain (may be three– or two–dimensional)
W	(subscript) indicates West orientation
x	rectangular coordinates variable $[cm]$
y	rectangular coordinates variable $[cm]$

LIST OF SYMBOLS

2. Greek Characters

β	delayed neutrons fraction
δ_{ab}	Kronecker delta
Δx	half the horizontal size of the rectangle domain $[cm]$
Δy	half the vertical size of the rectangle domain $[cm]$
Δt	half the time step $[s]$
ϵ	methodology error $[cm^{-3}s^{-1}]$
ε	relative difference between two consecutive iterations
λ	decay constant $[s^{-1}]$
ν	average number of neutrons emitted by fission
Σ_a	absorption cross section $[cm^{-1}]$
Σ_c	capture cross section $[cm^{-1}]$
Σ_f	fission cross section $[cm^{-1}]$
Σ_s	scattering cross section $[cm^{-1}]$
Σ_t	total cross section $[cm^{-1}]$
ϕ	neutron scalar flux $[cm^{-2}s^{-1}]$
φ	coefficient of PEM in ϕ $[cm^{-2}s^{-1}, \ldots, cm^{-6}s^{-1}]$
χ	integrated (total) fission spectrum
χ_F	integrated prompt fission spectrum
χ_D	integrated delayed fission spectrum

1 INTRODUCTION

A solution to the neutron space diffusion kinetics model is quite difficult to obtain, specially for irregular (non–rectangular) geometry and heterogeneous medium. This model typically contain a stiff set of partial differential equations that consists in an initial and boundary values problem. Due to this complexity it is usual to separate the time and space problems with different approaches. Even separating the time and space issues, analytical solutions are quite rare, even for simpler cases. Anyway, the community is considerably interested in numerical methods to solve heterogeneous diffusion problems. As numerical methods are usually subject to errors, there is always a concern about the methodology precision. Also, many numerical methods ignore the interface conditions: the continuity of the neutron density current and scalar flux. This thesis presents a new approach that uses a polynomial approximation as basis to the neutron fluxes, where the interface and boundary conditions contribute considerably more than usual codes to thermal nuclear reactor cores, with proper error estimates.

Almost from the beginning of the applied neutron physics, numerical methods have been used much more often than purely analytical methods to simulate neutron population in nuclear reactor cores. This apparent tradition is almost mandatory as the geometries and physical parameters usually are combined in a complex way to solve their models analytically. Also, the neutron transport model, that more precisely describes the behavior of the neutron flux through the space and time, is not simply solved, even with numerical methods and powerful computers. That is why the neutron diffusion space kinetics approximation is used instead, under some restrictions: to obtain approximate enough solutions with lesser computer efforts. However, even the space kinetics is complicated to solve analytically with most actual geometry and physical parameters configurations. It naturally leads to numerical algorithms, that naturally have some approximations. The textbooks contain further information about nuclear reactor physics [Duderstadt and Hamilton, 1976; Reuss, 2008].

As example, Alcouffe and Albrecht, 1970 show a generalization of the finite differences method applied to neutron space kinetics problems. Also, Mohsen Ayyoubzadehh

et al., 2012 contains a modified finite elements methodology, with direct determination of the coefficients in an iterative method. A nodal methodology is used in diffusion problems in Barros et al., 2003. Orellana and Barros, 2002, use an analytical method for mono–energetic neutrons in one–dimensional rectangular geometry for neutron diffusion models. Picca and Furfaro, 2012 show a new reverse problem for neutron space kinetics with application to accelerator–driven systems (ADS). Ganapol, 1992 uses an analytical approach to implement one speed one–dimensional neutron diffusion in heterogeneous medium. A Monte Carlo approach, in Camargo et al., 2013, is used to describe neutron quantities with continuous dependence of energy. An analytical benchmark for ADS was published in Dulla et al., 2007. A modified nodal method was used to neutron space kinetics in Grossman and Hennart, 2007. Guessous and Akhmouch, 2002 show a higher order nodal method for the neutron diffusion equations. Guyot et al., 2015 use a coupled three–dimensional neutron space kinetics code with thermal–hydraulics phenomena to find a coherent neutron point kinetics model.

A list of most usual numerical methods can be found in Han et al., 2009. Using finite elements method, a two–dimensional algorithm for neutron diffusion is described in Hosseini and Vosoughi, 2013. Quintero-Leyva, 2010 shows a polynomial approximation for neutron diffusion calculations. Lima et al. published new methodologies with pseudo–harmonic functions, a new method was developed and it is presented in [Lima et al., 2009, 2004]. Galerkin–type elements were used together with the boundary element–response matrix method to solve the neutron diffusion equations [Maiani and Montagnini, 2004]. In Moghaddam et al., 2015, a two–dimensional neutron space kinetics model with a fractional order Laplacian was solved using a modified finite volumes method. Aboanber and Hamada, 2008 and Aboanber and Nahla, 2006 contain some numerical approaches to solve the two– and three–dimensional neutron space kinetics equations, using Taylor series and Padé approximations. Nahla et al., 2012 contains some numerical approaches to solve the neutron diffusion equations. To accelerator–driven systems, a multi–point method is used to the kinetics problems in Ravetto et al., 2004. A modified (element–free) Galerkin basis is used to two–dimensional neutron diffusion equation problems in Rokrok et al., 2012. Sutton and Aviles, 1996 present some diffusion theory methods used to obtain solutions to neutron diffusion problems. Using a fractional derivatives approach, a new neutron diffusion equation

is proposed in Vyawahare and Nataraj, 2013. Yasinsky and Henry, 1965 show some numerical experiments results for time dependent neutron space kinetics. Argonne Code Center, 1977 is a numerical benchmark book that shows results for the eigenvalue problem in several geometries and nuclear parameters combinations. Vosoughi et al., 2004 show a modified differences equations to calculate neutron population in nuclear reactors. It is noteworthy that, despite these papers were published through decades, most of them emphasize numerical methods. As cited before, analytical approaches to neutron diffusion are not usual, specially when dealing with two– or three–dimensional heterogeneous problems. Also, further reading show that these methods take very few concern about the interface conditions, using mostly or only the differential equation in their methodologies. Anyway, these papers show that the community is concerned about computational methods for neutron physics in nuclear reactor, in a general way.

Further, there is some concern on computational algorithms in the time variable. The neutron point kinetics model treats the behavior of the neutron through time only, and it is proper for algorithm testing, as it is known that not any methodology works with this stiff problem. E. g., Mitchell, 1977 used Taylor series to obtain solutions for the neutron point kinetics equations. Nahla and Zayed propose a method to solve the nonlinear point kinetics equations and later used a Taylor series approach to the same purpose [Nahla and Zayed, 2010; Nahla, 2011]. The article Dulla et al., 2008 contain a quasi–static method applied to the neutron transport equation, together with some analytical results. There are many other not cited published papers in years about the neutron point kinetics, but that is not the focus. The proposed methods, together with the neutron space kinetics methods, deal with time in a very particular and delicate way, as the time problem is stiff.

Ceolin et al. use a polynomial approximation to solve both the steady state and time dependent one dimensional slab geometry neutron space kinetics problems [Ceolin et al., 2014, 2015]. In this document, the author presents the generalization of this methodology to two–dimensional neutron diffusion model of nuclear reactor cores. The objective is a new methodology and its analysis of applicability in nuclear reactor cores for both the steady state (eigenvalue) and time dependent problems. A low order polynomial is proposed to solve the neutron space kinetics equations, together with the neutron diffusion classic eigenvalue problem, for multiple regions, and multiple energy groups coupled with the precursors

concentrations for time dependent cases. The space domain is segmented, and each unknown is expanded into a second order polynomial for both orthogonal variables. As the unknowns are expanded in low order polynomials, the differential equation is used to determine a relation between the powers coefficients, apparently independent of segment sizes. The boundary and interface conditions complete the system that determine the polynomial coefficients. As the polynomials coefficients are related one another algebraically at the same segment, some numerical simplifications are done to reduce computation time. This methodology contains some analytical characteristic, as the complete Taylor series would converge to the analytical solution locally. Thus, the error may be controlled either by increasing the truncation order or restricting its validity. The last option leads to segment the domain even further, so a low order polynomial is used to represent a curve in a smaller and smaller space. It is understandable that, if one considers the segment size going to zero, the solution also converges to an analytical solution if the function is smooth enough (and it is). As usual in many numerical methodologies, a local error analysis is presented, however this methodology uses the differential equations themselves as reference to determine the methodology error. In other words, the whole model itself is used as an error determination, as the analytical solution is not known for the whole domain, but any solution that solves exactly the presented model is analytical, as the neutron diffusion model has been proven to have an unique solution. This new methodology consists in solving the space problem only. The time problem uses the modified method of Euler, as described in Burden and Faires, 2008, that was proven stable and proper to use in stiff problems despite requiring a very short step to obtain a precise result. The proposed methodology is used in some test cases, for both for the eigenvalue problem and the time dependent case.

This document is composed by 4 chapters. The present chapter which introduces the document. Chapter 2 describes the methodology used locally and their link to form the global solutions for the whole reactor in one time step. Then, the eigenvalue and time dependent problems for the neutron diffusion model are presented and properly fit in the methodology application. Chapter 3 shows some test cases for several geometries and time dependent nuclear parameters. The last chapter 4 concludes this document with some commentaries and future perspectives.

2 FORMAL AND COMPUTATIONAL IMPLEMENTATION

In this chapter we are going to see the novelty of this thesis: a precise methodology for the neutron diffusion model in thermal nuclear reactors. The two–dimensional (x, y) time dependent neutron space kinetics and the neutron eigenvalue problem are considered to the implementation. In the time dependent case, the analytical continuity is used to ensure the precision in every time step. In the eigenvalue problem situation, the power method is used to find the dominant eigenvalue and its respective eigenfunction. The domain is segmented, and a polynomial expansion in x, y and t, when applicable, is used for all unknowns and functions for each segment. After, the application of initial conditions (for certain time step) reduces the effective number of unknowns, and a system for all segments (or regions) is made, linking the solutions of a segment to the others by the interface and boundary conditions. Some algebraic work also reduces even more the amount of coefficients to be determined. This set of algebraic system is solved by an iterative method. Last, a not so mathematically rigorous local error estimate is presented, mathematically speaking. Further explanations about the power method can be found in text books [Duderstadt and Hamilton, 1976; Reuss, 2008]. Some other numerical methods used to solve the same type of problem can be found in cited articles [Han et al., 2009; Barros et al., 2003; Maiani and Montagnini, 2004; Rokrok et al., 2012].

2.1 Local solution, a polynomial expansion methodology (PEM)

In this section the author presents a methodology to solve a linear boundary value problem. The methodology presented here is appropriated to rectangular homogeneous domain with constant source term and constant coefficients and second order polynomial

independent term on the boundary conditions, as

$$\left(\frac{\partial^2\phi_g}{\partial x^2}+\frac{\partial^2\phi_g}{\partial y^2}\right)+H_g\phi_g+\sum_{g'=1}^{G}X_{gg'}\phi_{g'}=Q_g\ , \tag{2.1a}$$

$$A_{gN}\frac{\partial\phi_g}{\partial y}(x,\Delta y)+B_{gN}\phi_g(x,\Delta y)=\sum_{m=0}^{2}f_{gNm}x^m\ , \tag{2.1b}$$

$$A_{gS}\frac{\partial\phi_g}{\partial y}(x,-\Delta y)+B_{gS}\phi_g(x,-\Delta y)=\sum_{m=0}^{2}f_{gSm}x^m\ , \tag{2.1c}$$

$$A_{gE}\frac{\partial\phi_g}{\partial x}(\Delta x,y)+B_{gE}\phi_g(\Delta x,y)=\sum_{n=0}^{2}f_{gEn}y^n\ , \tag{2.1d}$$

$$A_{gW}\frac{\partial\phi_g}{\partial x}(-\Delta x,y)+B_{gW}\phi_g(-\Delta x,y)=\sum_{n=0}^{2}f_{gWn}y^n\ , \tag{2.1e}$$

for $(x,y)\in V$ with H_g, $X_{gg'}$, Q_g, $A_{g\iota}$, $B_{g\iota}$, $u_{g\iota}$, $v_{g\iota}$ and $w_{g\iota}$ constants and $\phi_g=\phi_g(x,y)$ is the g–th unknown, for $\iota=N,S,E,W$ meaning the orientation North, South, East and West, respectively. Consider V the rectangle with x in $[-\Delta x,\Delta x]$ and y in $[-\Delta y,\Delta y]$, and $g=1,2\ldots G$ unless specified. Also, $H_g\neq 0$, $Q_g\neq 0$ and $|A_{g\iota}|+|B_{g\iota}|>0$. This document presents a methodology to find a local solution to this case in particular, however it is shown later that some cases may be reduced to multiple boundary value problems like (2.1), whose solutions are linked somehow. To this end, at this time we focus on solving (2.1).

To solve (2.1), the author developed PEM. It consists in expanding the unknown in a second order polynomial, as

$$\phi_g=\sum_{m=0}^{2}\sum_{n=0}^{2}\varphi_{gmn}x^m y^n\ , \tag{2.2}$$

generating $9G$ constant unknowns, φ_{gmn}. To obtain these unknowns, (2.1) is recast using the expansion (2.2). First, the boundary conditions,

$$\left(B_N\Delta y^2+2A_N\Delta y\right)\varphi_{m2}+\left(A_N+B_N\Delta y\right)\varphi_{m1}+B_N\varphi_{m0}=f_{Nm}, \tag{2.3a}$$

$$\left(B_S\Delta y^2-2A_S\Delta y\right)\varphi_{m2}+\left(A_S-B_S\Delta y\right)\varphi_{m1}+B_S\varphi_{m0}=f_{Sm}\ , \tag{2.3b}$$

$$\left(B_E\Delta x^2+2A_E\Delta x\right)\varphi_{2n}+\left(A_E+B_E\Delta x\right)\varphi_{1n}+B_E\varphi_{0n}=f_{En}\ , \tag{2.3c}$$

$$\left(B_W\Delta x^2-2A_W\Delta x\right)\varphi_{2n}+\left(A_W-B_W\Delta x\right)\varphi_{1n}+B_W\varphi_{0n}=f_{Wn}\ , \tag{2.3d}$$

for $m, n = 0, 1, 2$, and omitting the subscript g in φ–s, A–s, B–s and f–s. All m and n from now on will be considered as $m, n = 0, 1, 2$ unless specified. This set is an arrange of 12 equations made by the substitution of (2.2) into the boundary conditions. After the substitution, the coefficients of each power of x and y are equaled. Since the approximation is a second order one, there are three x or y coefficients to be equaled (corresponding to x^0, x^1 and x^2, for example). We can make some algebraic work to find expressions that relate the unknowns with φ_{m0} and φ_{0n} only:

$$\varphi_{m2} = \mathcal{A}\varphi_{m0} + \mathcal{B}_m \ , \tag{2.4a}$$

$$\varphi_{m1} = \mathcal{C}\varphi_{m0} + \mathcal{D}_m \ , \tag{2.4b}$$

$$\varphi_{2n} = \mathcal{E}\varphi_{0n} + \mathcal{F}_n \ , \tag{2.4c}$$

$$\varphi_{1n} = \mathcal{G}\varphi_{0n} + \mathcal{H}_n \ , \tag{2.4d}$$

with

$$\mathcal{A} = \frac{B_S\left(A_N + B_N\Delta y\right) - B_N\left(A_S - B_S\Delta y\right)}{\left(B_N\Delta y^2 + 2A_N\Delta y\right)\left(A_S - B_S\Delta y\right) - \left(B_S\Delta y^2 - 2A_S\Delta y\right)\left(A_N + B_N\Delta y\right)} \ , \tag{2.5a}$$

$$\mathcal{B}_m = \frac{f_{Nm}\left(A_S - B_S\Delta y\right) - f_{Sm}\left(A_N + B_N\Delta y\right)}{\left(B_N\Delta y^2 + 2A_N\Delta y\right)\left(A_S - B_S\Delta y\right) - \left(B_S\Delta y^2 - 2A_S\Delta y\right)\left(A_N + B_N\Delta y\right)} \ , \tag{2.5b}$$

$$\mathcal{C} = \frac{B_N\left(B_S\Delta y^2 - 2A_S\Delta y\right) - B_S\left(B_N\Delta y^2 + 2A_N\Delta y\right)}{\left(B_N\Delta y^2 + 2A_N\Delta y\right)\left(A_S - B_S\Delta y\right) - \left(B_S\Delta y^2 - 2A_S\Delta y\right)\left(A_N + B_N\Delta y\right)} \ , \tag{2.5c}$$

$$\mathcal{D}_m = \frac{f_{Sm}\left(B_N\Delta y^2 + 2A_N\Delta y\right) - f_{Nm}\left(B_S\Delta y^2 - 2A_S\Delta y\right)}{\left(B_N\Delta y^2 + 2A_N\Delta y\right)\left(A_S - B_S\Delta y\right) - \left(B_S\Delta y^2 - 2A_S\Delta y\right)\left(A_N + B_N\Delta y\right)} \ , \tag{2.5d}$$

$$\mathcal{E} = \frac{B_W\left(A_E + B_E\Delta x\right) - B_E\left(A_W - B_W\Delta x\right)}{\left(B_E\Delta x^2 + 2A_E\Delta x\right)\left(A_W - B_W\Delta x\right) - \left(B_W\Delta x^2 - 2A_W\Delta x\right)\left(A_E + B_E\Delta x\right)} \ , \tag{2.5e}$$

$$\mathcal{F}_n = \frac{f_{En}\left(A_W - B_W\Delta x\right) - f_{Wn}\left(A_E + B_E\Delta x\right)}{\left(B_E\Delta x^2 + 2A_E\Delta x\right)\left(A_W - B_W\Delta x\right) - \left(B_W\Delta x^2 - 2A_W\Delta x\right)\left(A_E + B_E\Delta x\right)} \ , \tag{2.5f}$$

$$\mathcal{G} = \frac{B_E\left(B_W\Delta x^2 - 2A_W\Delta x\right) - B_W\left(B_E\Delta x^2 + 2A_E\Delta x\right)}{\left(B_E\Delta x^2 + 2A_E\Delta x\right)\left(A_W - B_W\Delta x\right) - \left(B_W\Delta x^2 - 2A_W\Delta x\right)\left(A_E + B_E\Delta x\right)} \ , \tag{2.5g}$$

$$\mathcal{H}_n = \frac{f_{Wn}\left(B_E\Delta x^2 + 2A_E\Delta x\right) - f_{En}\left(B_W\Delta x^2 - 2A_W\Delta x\right)}{\left(B_E\Delta x^2 + 2A_E\Delta x\right)\left(A_W - B_W\Delta x\right) - \left(B_W\Delta x^2 - 2A_W\Delta x\right)\left(A_E + B_E\Delta x\right)} \ . \tag{2.5h}$$

This set of equations contains 12 equations and 9 unknowns, but four of these relations are proven linear dependent. A demonstration of this linear dependency is shown on the next

topic. The differential equation (2.1a) with (2.2) becomes

$$2\sum_{\kappa=0}^{2}\left[\varphi_{g2\kappa}y^{\kappa}+\varphi_{g\kappa 2}x^{\kappa}\right]+H_g\sum_{m=0}^{2}\sum_{n=0}^{2}\varphi_{gmn}x^m y^n+\sum_{g'=1}^{G}X_{gg'}\sum_{m=0}^{2}\sum_{n=0}^{2}\varphi_{g'mn}x^m y^n=Q_g\ . \quad (2.6)$$

Clearly, this expression determine index equations, grouping the coefficients of the same $x^m y^n$, however, the set of φ_{gmn} would not satisfy the boundary conditions. That is why the author chose to set only the coefficients of x^0y^0 as true,

$$2\left(\varphi_{g20}+\varphi_{g02}\right)+H_g\varphi_{g00}+\sum_{g'=1}^{G}X_{gg'}\varphi_{g'00}=Q_g\ , \quad (2.7)$$

and the remaining terms are the error of this methodology, as the boundaries are as precise as a second order polynomial approximation may be. Using (2.4), we get a linear algebraic system depending only on φ_{g00}. Some algebraic work lead to the linear system

$$\left(2\mathcal{A}_g+2\mathcal{E}_g+H_g\right)\varphi_{g00}+\sum_{g'=1}^{G}X_{gg'}\varphi_{g'00}=Q_g-2\mathcal{B}_{g0}-2\mathcal{F}_{g0}\ . \quad (2.8)$$

This is a linear algebraic system, and can be solved by any standard methodology, even Gauss' elimination (specially when G is not a large number). After solving (2.8), we use (2.4) to get the other φ_{gmn}.

2.2 Error estimates

The methodology described to solve (2.1) solves exactly the boundaries, but commits an error in the differential equation, making this methodology's output to differ from the analytical solution. As usual in local error analysis, the $(0,0)$–centered Taylor series expansion of the unknown ϕ_g writes

$$\phi_g=\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\Phi_{gmn}x^m y^n\ , \quad (2.9a)$$

where

$$\Phi_{gmn} = \frac{1}{m!n!}\frac{\partial^{m+n}\phi_g}{\partial x^m \partial y^n}(0,0) \tag{2.9b}$$

and the differential equation (2.1) writes

$$\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\left[(m+1)(m+2)\,\Phi_{g(m+2)n} + (n+1)(n+2)\,\Phi_{gm(n+2)} + H_g\Phi_{gmn} + \sum_{g'=1}^{G} X_{gg'}\Phi_{g'mn}\right] x^m y^n = Q_g\ . \tag{2.10}$$

This equations generates the index equations from the coefficients of $x^m y^n$

$$(m+1)(m+2)\,\Phi_{g(m+2)n} + (n+1)(n+2)\,\Phi_{gm(n+2)} + H_g\Phi_{gmn} + \sum_{g'=1}^{G} X_{gg'}\Phi_{g'mn} = \delta_{(m+n)0}Q_g\ . \tag{2.11}$$

This equation is valid for $m, n = 0, 1, 2, \ldots$, and δ_{ab} is the Kronecker delta. Comparing (2.11) to (2.7), the only index equation satisfied is the one for $m = n = 0$, which states a relation between φ_{g20}, φ_{g02} and φ_{g00}. It means that, despite these second order derivatives are relating one another correctly with the polynomial approximations, all other terms not necessarily are. In other words, the present methodology coincides with the complete Taylor series at only one degree of freedom. Besides that, the infinite Taylor series and the remaining index equations are substituted by a second order polynomial and the relations (2.4), generated by the boundary conditions. It means that taking φ_{00} as the starting point to build an accurate solution, its index equation is considered in the present methodology, but others are not, so the solution would be, rigorously, a zeroth order truncation approximation, as it considers only one degree of freedom required to construct the complete Taylor series of ϕ_g.

As mentioned before, the present methodology solves exactly a boundary whose independent function is a second order polynomial, so the error lies on the differential equation. If the differential equation is considered as reference instead of the solution itself, the methodology error is given by the complete substitution of the polynomial approximation

into the differential equation. Using the solution obtained by the present methodology, (2.6) is actually a false statement. Actually, a more accurate expression would be its substitution squared, generating a square methodology error expression,

$$\epsilon_g = \left[2\sum_{\kappa=0}^{2} (\varphi_{g2\kappa} y^{\kappa} + \varphi_{g\kappa 2} x^{\kappa}) + H_g \sum_{m=0}^{2}\sum_{n=0}^{2} \varphi_{gmn} x^m y^n + \right. \\ \left. \sum_{g'=1}^{G} X_{gg'} \sum_{m=0}^{2}\sum_{n=0}^{2} \varphi_{g'mn} x^m y^n - Q_g \right]^2 , \tag{2.12}$$

where $\epsilon_g = \epsilon_g(x, y)$ is the so–called g–th methodology error. Note, some terms are going to cancel themselves as expected, according to (2.8), and ϵ is simply the remaining terms. As ϵ_g is a polynomial, it is expected that it is limited at its rectangular domain for x in $[-\Delta x, \Delta x]$ and y in $[-\Delta y, \Delta y]$. The absolute extremes of ϵ_g are the maximum error estimates expected for this methodology. The "candidates" to absolute extremes of ϵ_g may be obtained by analyzing the critical points of it:

- where its gradient is null;
- at the boundary where its derivative of the orthogonal direction of that boundary is null; and
- at the corners.

Clearly, to determine exactly where the maximum error will occur, it is necessary to solve (2.8) first. This is actually a low point of the present procedure, however once the solution is found the actual error may be calculated anywhere in the rectangular domain. Also, it is noteworthy that we may imagine a rectangle with area bigger than zero that do not contain the point (x_0, y_0) where the error is extreme. It means that we can segment the domain in order to improve accuracy, as it is known that two subsequent polynomials approximate better a smooth function than one. To do that, it will be necessary to link the segmented solutions somehow showed later. There are actually G methodology errors, and they are linked one another, however it is not assured similarity in their behavior. It means that simply fixing the error for one ϵ_g does not imply that the other ϵ_g will also reduce. Finally, in this section it was shown that, for a solution ϕ_g approximated by a determined second

order polynomial, the error of the methodology is given by (2.12), and as it is a continuous function in all domain, its extreme values may be used to refine the rectangular domain into smaller rectangular ones, as it is shown later in this document.

2.3 Linear dependency

As mentioned before, the set of relations (2.4) contain 12 relations of 9 unknowns. Here, it is shown that 4 of these relations are linear combinations of others, so they lead to the same value. We are going to hide the subscript g to make a cleaner notation at this part. This procedure is made similarly for each g, hence produces similar conclusions for each ϕ_g.

For example, the unknown φ_{21} might be written using (2.4b) and (2.4c). If there is linear dependency, these expressions should result the same value for φ_{21} no matter which way we choose, so

$$\mathcal{C}\varphi_{20} + \mathcal{D}_2 = \mathcal{E}\varphi_{01} + \mathcal{F}_1 \ , \tag{2.13}$$

and using (2.4b) and (2.4c) again and simplifying the relation as much as possible, we get

$$\mathcal{C}\mathcal{F}_0 + \mathcal{D}_2 = \mathcal{E}\mathcal{D}_0 + \mathcal{F}_1 \ . \tag{2.14}$$

Substituting these compact constants into their expanded forms using (2.5),

$$\begin{aligned}
&\Big[B_N\left(B_S\Delta y^2 - 2A_S\Delta y\right) - B_S\left(B_N\Delta y^2 + 2A_N\Delta y\right)\Big]\Big[f_{E0}\left(A_W - B_W\Delta x\right) - \\
&\quad f_{W0}\left(A_E + B_E\Delta x\right)\Big] + \Big[\left(B_S\Delta y^2 - 2A_S\Delta y\right)\left(A_N + B_N\Delta y\right) - \\
&\quad \left(B_N\Delta y^2 + 2A_N\Delta y\right)\left(A_S - B_S\Delta y\right)\Big]\Big[f_{E1}\left(A_W - B_W\Delta x\right) - f_{W1}\left(A_E + B_E\Delta x\right)\Big] = \\
&\quad \Big[\left(B_W\Delta x^2 - 2A_W\Delta x\right)\left(A_E + B_E\Delta x\right) - \left(B_E\Delta x^2 + 2A_E\Delta x\right)\left(A_W - B_W\Delta x\right)\Big]\cdot \\
&\quad \Big[f_{S2}\left(B_N\Delta y^2 + 2A_N\Delta y\right) - f_{N2}\left(B_S\Delta y^2 - 2A_S\Delta y\right)\Big] + \Big[B_W\left(A_E + B_E\Delta x\right) - \\
&\quad B_E\left(A_W - B_W\Delta x\right)\Big]\Big[f_{S0}\left(B_N\Delta y^2 + 2A_N\Delta y\right) - f_{N0}\left(B_S\Delta y^2 - 2A_S\Delta y\right)\Big] \ .
\end{aligned} \tag{2.15}$$

It means that unless (2.15) is true, the methodology has an inconsistency: two different expressions for the same unknown. Fortunately, this is not the case. If we find

(2.15) by some other (doubtless truthful) way, we confirm the linear dependency. Beginning with the boundary conditions in their general form, for the East and South boundaries, both evaluated at the same position $(\Delta x, -\Delta y)$

$$A_E \frac{\partial \phi}{\partial x}(\Delta x, -\Delta y) + B_E \phi(\Delta x, -\Delta y) = \sum_{n=0}^{2} f_{En} (-\Delta y)^n \ , \tag{2.16a}$$

$$A_S \frac{\partial \phi}{\partial y}(\Delta x, -\Delta y) + B_S \phi(\Delta x, -\Delta y) = \sum_{m=0}^{2} f_{Sm} (\Delta x)^m \ , \tag{2.16b}$$

we operate these expressions like $\{B_S(2.16\text{a}) - B_E(2.16\text{b})\}$ and get

$$B_S A_E \frac{\partial \phi}{\partial x}(\Delta x, -\Delta y) - B_E A_S \frac{\partial \phi}{\partial y}(\Delta x, -\Delta y) = \sum_{\kappa=0}^{2} \left[B_S f_{E\kappa} (-\Delta y)^\kappa - B_E f_{S\kappa} (\Delta x)^\kappa \right] \ . \tag{2.17}$$

Also, we might make $\left\{ A_E \frac{\partial}{\partial x}(2.16\text{b}) - A_S \frac{\partial}{\partial y}(2.16\text{a}) \right\}$ and find

$$A_E B_S \frac{\partial \phi}{\partial x}(\Delta x, -\Delta y) - A_S B_E \frac{\partial \phi}{\partial y}(\Delta x, -\Delta y) = \sum_{\kappa=0}^{1} \left[(\kappa+1) \left(A_E f_{S(\kappa+1)} (\Delta x)^\kappa - A_S f_{E(\kappa+1)} (-\Delta y)^\kappa \right) \right] \ . \tag{2.18}$$

As the left hand side of (2.17) and (2.18) are the same, we combine them to find a relation with f_{En} and f_{Sm} only (no φ_{mn}),

$$\left(B_S \Delta y^2 - 2 A_S \Delta y \right) f_{E2} + \left(A_S - B_S \Delta y \right) f_{E1} + B_S f_{E0} = \left(B_E \Delta x^2 + 2 A_E \Delta x \right) f_{S2} + \left(A_E + B_E \Delta x \right) f_{S1} + B_E f_{S0} \ . \tag{2.19a}$$

The same procedure is done with the other corners N–E, S–W and N–W,

$$\left(B_N\Delta y^2+2A_N\Delta y\right)f_{E2}+\left(A_N+B_N\Delta y\right)f_{E1}+B_Nf_{E0}= \\ \left(B_E\Delta x^2+2A_E\Delta x\right)f_{N2}+\left(A_E+B_E\Delta x\right)f_{N1}+B_Ef_{N0}\ , \tag{2.19b}$$

$$\left(B_S\Delta y^2-2A_S\Delta y\right)f_{W2}+\left(A_S-B_S\Delta y\right)f_{W1}+B_Sf_{W0}= \\ \left(B_W\Delta x^2-2A_W\Delta x\right)f_{S2}+\left(A_W-B_W\Delta x\right)f_{S1}+B_Wf_{S0}\ , \tag{2.19c}$$

$$\left(B_N\Delta y^2+2A_N\Delta y\right)f_{W2}+\left(A_N+B_N\Delta y\right)f_{W1}+B_Nf_{W0}= \\ \left(B_W\Delta x^2-2A_W\Delta x\right)f_{N2}+\left(A_W-B_W\Delta x\right)f_{N1}+B_Wf_{N0}\ . \tag{2.19d}$$

It is important that these equations are not in any way generated by the pseudo–index equations created from the expansion of ϕ into a polynomial. They are made with the independent terms second order polynomial expression and linear operations, so for all purposes they are exact. As we found these φ_{mn} independent expressions and they are verified true, all we got to do is some algebra to find our desired confirmation of linear dependency. Recalling the example of the φ_{21} unknown, to find (2.15) we can manipulate (2.19) to eliminate the f–s that do not appear on it. E.g. to eliminate f_{W2}, we make the linear operation $\{(B_N\Delta y^2+2A_N\Delta y^2)\,(2.19\text{c})-\ (B_S\Delta y^2-2A_S\Delta y)\,(2.19\text{d})\}$ and get

$$\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)\left(A_S-B_S\Delta y\right)-\left(B_S\Delta y^2-2A_S\Delta y\right)\left(A_N+B_N\Delta y\right)\Big]f_{W1}+ \\ \Big[\left(B_N\Delta y^2+2A_N\Delta y\right)B_S-\left(B_S\Delta y^2-2A_S\Delta y\right)B_N\Big]f_{W0}=\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)f_{S2}- \\ \left(B_S\Delta y^2-2A_S\Delta y\right)f_{N2}\Big]\left(B_W\Delta x^2-2A_W\Delta x\right)+\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)f_{S1}- \\ \left(B_S\Delta y^2-2A_S\Delta y\right)f_{N1}\Big]\left(A_W-B_W\Delta x\right)+\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)f_{S0}- \\ \left(B_S\Delta y^2-2A_S\Delta y\right)f_{N0}\Big]B_W\ . \tag{2.20}$$

Similarly, we operate $\{(B_N\Delta y^2+2A_N\Delta y^2)\,(2.19\text{a})-(B_S\Delta y^2-2A_S\Delta y)\,(2.19\text{b})\}$ to elimi-

nate f_{E2},

$$\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)\left(A_S-B_S\Delta y\right)-\left(B_S\Delta y^2-2A_S\Delta y\right)\left(A_N+B_N\Delta y\right)\Big]f_{E1}+$$
$$\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)B_S-\left(B_S\Delta y^2-2A_S\Delta y\right)B_N\Big]f_{E0}=\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)f_{S2}-$$
$$\left(B_S\Delta y^2-2A_S\Delta y\right)f_{N2}\Big]\left(B_E\Delta x^2+2A_E\Delta x\right)+\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)f_{S1}-$$
$$\left(B_S\Delta y^2-2A_S\Delta y\right)f_{N1}\Big]\left(A_E+B_E\Delta x\right)+\Big[\left(B_N\Delta y^2+2A_N\Delta y\right)f_{S0}-$$
$$\left(B_S\Delta y^2-2A_S\Delta y\right)f_{N0}\Big]B_E\ . \quad (2.21)$$

Finally, making $\{(A_E+B_E\Delta x)\,(2.20)-(A_W-B_W\Delta x)\,(2.21)\}$ in order to find an expression without f_{N1} and f_{S1}, we find (2.15). It means that (2.15) is true. This result means that, although there are two ways to express φ_{21} in terms of φ_{00}, both lead to the same result, so we can simply choose any of (2.4b) and (2.4c) to express this unknown. The same procedure might be done for φ_{22}, φ_{12} and φ_{11}, providing similar results, and there are the four extra equations that have been proven linear dependent. The present procedure was made for generic third species boundary conditions in all directions, but it can be done for any combination of any type of boundary conditions, generating the same output: there are exactly four redundant equations in (2.4), as there are two different ways to calculate φ_{11}, φ_{12}, φ_{21} and φ_{22} and both lead to the same result. In other words, we can safely use any of the (2.4) relations to express the unknowns in terms of φ_{00}. The set of these four last equations must contain at least once the variables φ_{11}, φ_{12}, φ_{21} and φ_{22}.

As a final remark, it is known that the flux and current continuities are assured by the physical model itself, however it is seen in the next section an iterative solution for sectionally homogeneous domain. This iterative scheme begins with estimates, so (2.16) is not necessarily true. Although in the iterative solution (2.16) will be assured, the algorithm stops with a typical stop criterion that may be reached before the actual convergence, meaning that the evaluation of φ_{mn} $(m,n=1,2)$ by one mean only can lead to an error. To avoid this situation, these classic expressions (that in the convergence will be satisfied) are substituted by their average expressions. In short words, doing this "forces" the unknowns to their actual value in the iterative method. Then, the complete set of φ_{00} dependent expressions

of φ_{mn} is

$$\varphi_{01} = \mathcal{C}\varphi_{00} + \mathcal{D}_0 \ , \tag{2.22a}$$

$$\varphi_{02} = \mathcal{A}\varphi_{00} + \mathcal{B}_0 \ , \tag{2.22b}$$

$$\varphi_{10} = \mathcal{G}\varphi_{00} + \mathcal{H}_0 \ , \tag{2.22c}$$

$$\varphi_{11} = \frac{\mathcal{C}\varphi_{10} + \mathcal{D}_1 + \mathcal{G}\varphi_{01} + \mathcal{H}_1}{2} \ , \tag{2.22d}$$

$$\varphi_{12} = \frac{\mathcal{A}\varphi_{10} + \mathcal{B}_1 + \mathcal{G}\varphi_{02} + \mathcal{H}_2}{2} \ , \tag{2.22e}$$

$$\varphi_{20} = \mathcal{E}\varphi_{00} + \mathcal{F}_0 \ , \tag{2.22f}$$

$$\varphi_{21} = \frac{\mathcal{C}\varphi_{20} + \mathcal{D}_2 + \mathcal{E}\varphi_{01} + \mathcal{F}_1}{2} \ , \tag{2.22g}$$

$$\varphi_{22} = \frac{\mathcal{A}\varphi_{20} + \mathcal{B}_2 + \mathcal{E}\varphi_{02} + \mathcal{F}_2}{2} \ . \tag{2.22h}$$

This numerical trick was proven efficient, increasing precision in the test cases. This set of equations replaces (2.16) for numerical purposes. Needless to say, the procedure made in this part of the thesis is valid for any ϕ_g.

Now we are able to write an algorithm with error estimate that solves (2.1). In short terms, we describe the **{local}** algorithm as

1. Identify Δx, Δy, ϕ_g, H_g, $X_{gg'}$, Q_g, $A_{g\iota}$, $B_{g\iota}$ and $f_{g\iota\kappa}$ in (2.1) for $g, g' = 1, 2, \ldots, G$, $\iota = N, S, E, W$ and $\kappa = 0, 1, 2$;

2. Calculate $\mathcal{A}_g$, $\mathcal{B}_{g\kappa}$, $\mathcal{C}_g$, $\mathcal{D}_{g\kappa}$, $\mathcal{E}_g$, $\mathcal{F}_{g\kappa}$, $\mathcal{G}_g$, and $\mathcal{H}_{g\kappa}$ using (2.5) for $g = 1, 2, \ldots, G$, and $\kappa = 0, 1, 2$;

3. Calculate φ_{g00} with (2.8) for $g = 1, 2, \ldots, G$;

4. Calculate the remaining φ_{gmn} with (2.22) for $g = 1, 2, \ldots, G$ and $m, n = 0, 1, 2$;

5. Final solution ϕ_g and error ϵ_g are given by (2.2) and (2.12) for $g = 1, 2, \ldots, G$;

2.4 Iterative global solution

So far this thesis presented a methodology to solve boundary value problems with an homogeneous rectangular domain with heterogeneous differential equation, however the point of this document is an algorithm that solves actual geometries for nuclear reactors.

Figure 2.1a represents an example geometry to be solved by the presented methodology, where the hatches indicates same H_g and $X_{gg'}$. To this end, the complete geometry is segmented into rectangular homogeneous domains indicated by the hatched and thick lines (interfaces) in the interior of the domain in figure 2.1a. The local solutions are linked one another by the interface conditions, usual in diffusive models and generally described as

$$\lim_{a\to 0}\left[\phi_g\left(x+a,y\right)-\phi_g\left(x-a,y\right)\right]=0\ , \tag{2.23a}$$

$$\lim_{a\to 0}\left[K_g\left(x+a,y\right)\frac{\partial\phi_g}{\partial x}\left(x+a,y\right)-K_g\left(x-a,y\right)\frac{\partial\phi_g}{\partial x}\left(x-a,y\right)\right]=0\ , \tag{2.23b}$$

at the vertical interfaces, and

$$\lim_{a\to 0}\left[\phi_g\left(x,y+a\right)-\phi_g\left(x,y-a\right)\right]=0\ , \tag{2.23c}$$

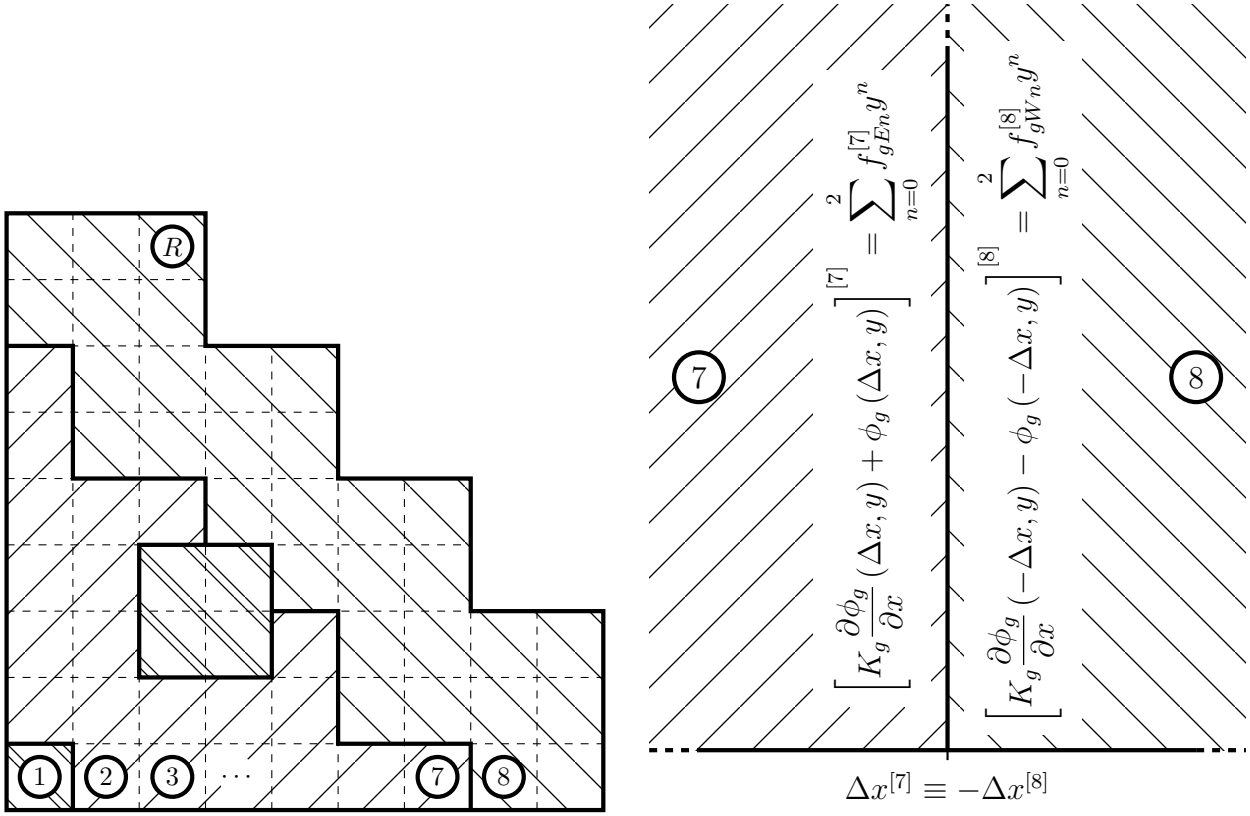
$$\lim_{a\to 0}\left[K_g\left(x,y+a\right)\frac{\partial\phi_g}{\partial y}\left(x,y+a\right)-K_g\left(x,y-a\right)\frac{\partial\phi_g}{\partial y}\left(x,y-a\right)\right]=0\ , \tag{2.23d}$$

at the horizontal interfaces. Consider $K_g=K_g\left(x,y\right)$ a piecewise constant function according to the hatches in figure 2.1a. The solution of each segment respects a differential equation and boundary conditions that fits (2.1) locally. As we intend to solve all regions iteratively, we are going to arrange the equations in such a way that the matrix of coefficients of φ_{gmn} is block–diagonally dominant. To this end, an algebraic work is made with the interface conditions. As each interface represents two degrees of freedom in the global linear system and we are going to use an iterative solution, each segment will consider that interface as a fictitious boundary in its solution. As example, we are going to consider the interface between the element 7 and 8 in figure 2.1a. We operate the interface conditions as $\{(2.23a)\pm(2.23b)\}$ to get

$$\left[K_g\frac{\partial\phi_g}{\partial x}\left(\Delta x,y\right)-\phi_g\left(\Delta x,y\right)\right]^{[7]}=\left[K_g\frac{\partial\phi_g}{\partial x}\left(-\Delta x,y\right)-\phi_g\left(-\Delta x,y\right)\right]^{[8]}\ , \tag{2.24a}$$

$$\left[K_g\frac{\partial\phi_g}{\partial x}\left(\Delta x,y\right)+\phi_g\left(\Delta x,y\right)\right]^{[7]}=\left[K_g\frac{\partial\phi_g}{\partial x}\left(-\Delta x,y\right)+\phi_g\left(-\Delta x,y\right)\right]^{[8]}\ , \tag{2.24b}$$

where the superscript $[r]$ represents the proper segment. Here, it is important to notice that each region has its own K_g, ϕ_g and local variables x and y, and its proper orientations (e.g.



(a) Segmented domain (b) Interface conditions between regions 7 and 8.

Figure 2.1: Two–dimensional domain segmentation.

the interface between the regions 7 and 8 is located at East of 7, but at West of 8).

These equations are used in the iterative system, but not at the same time. In order to assure convergence, (2.24a) is used as a fictitious boundary condition for region 8, and (2.24b) for region 7. This way, we cast these fictitious boundary conditions as

$$K_g \frac{\partial \phi_g}{\partial x}(\Delta x, y) + \phi_g(\Delta x, y) = \sum_{n=0}^{2} f_{gEn} y^n \tag{2.25a}$$

for region 7, and

$$K_g \frac{\partial \phi_g}{\partial x}(-\Delta x, y) - \phi_g(-\Delta x, y) = \sum_{n=0}^{2} f_{gWn} y^n \tag{2.25b}$$

for region 8, where the boundary independent terms are determined by the neighbor regions. Figure 2.1b demonstrates the interface conditions between regions 7 and 8 of our example. In

this example, the boundary independent terms are directly determined by the evaluation of the neighbor region's solution. This happens because the mesh consists of regular rectangle subdivisions. When this does not happen, the interface conditions will always generate a function that can be approximated by a second order polynomial, resulting in (2.25) anyway (remember we are dealing with rectangular meshing). This procedure and operations are made for all interfaces, creating a relation that we will treat as fictitious boundary conditions in our iterative algorithm. If the border of an element is an actual boundary, the actual boundary condition is used. This way, all segments contain one boundary condition at all orientations. The operations for these fictitious boundary conditions are

$$\{(2.23\text{d}) + (2.23\text{c})\} \,, \tag{2.26a}$$

$$\{(2.23\text{d}) - (2.23\text{c})\} \,, \tag{2.26b}$$

$$\{(2.23\text{b}) + (2.23\text{a})\} \,, \tag{2.26c}$$

$$\{(2.23\text{b}) - (2.23\text{a})\} \,, \tag{2.26d}$$

for North, South, East and West orientations, respectively.

Now we know how to build the local boundary conditions, the iterative algorithm to solve all regions is presented. The author chose to use the successive over–relaxation method (SOR), which consists in solving iteratively all regions, one at a time, updating their local solution until a stop criterion is satisfied. Further details about this methodology are found in Patankar, 1980, Strong, 2005 and Feingold and Varga, 1962. This iterative method to solve linear systems converges if the coefficients matrix is diagonally dominant. As the present algorithm solves many unknowns at a time (solve all unknowns in one region at the same time, then solve the next region and so on), this sufficient criterion is not simply applied. In an *ad hoc* attempt to make the global system pseudo diagonal dominant, the interface conditions are algebraically operated in a way that, together with PEM (2.2), SOR converges. Actually the linear operation made in the interface conditions that generated (2.24) has been proven sufficient to assure convergence after several tests by the author.

In short terms, the author applied the **{global}** iterative algorithm to find the solution for the whole domain described as

1. Segment the domain in R rectangular homogeneous regions (know how to find each

segment's neighbors).

2. Set ε_{max}.

3. Make initial guess to $\varphi^{[r]}_{gmn}$ and calculate $\varepsilon \leftarrow \varepsilon_{max} + 1$.

4. Store the old solution: $\varphi^{[r](\text{old})}_{gmn} \leftarrow \varphi^{[r]}_{gmn}$ for $r = 1, 2, \ldots, R$, $g = 1, 2, \ldots, G$ and $m, n = 0, 1, 2$.

5. For $r = 1, 2, \ldots, R$,

   (a) For $\iota = N, S, E, W$, if there is a boundary at ι, use the boundary condition, else use (2.26) to determine the proper fictitious boundary condition with the proper updated neighbors $\varphi^{[r]}_{gmn}$.

   (b) At this point, H_g and $X_{gg'}$ are constants. If Q_g is not constant, properly approximate it. If the boundary independent terms are not in second order polynomial form, approximate them*.

   (c) Use **{local}** to update $\varphi^{[r]}_{gmn}$.

6. Calculate the relative discrepancy: $\varepsilon \leftarrow \max\left(\left|\dfrac{\varphi^{[r]}_{gmn} - \varphi^{[r](\text{old})}_{gmn}}{\varphi^{[r]}_{gmn} + \varphi^{[r](\text{old})}_{gmn}}\right|\right)$ for $r = 1, 2, \ldots, R$, $g = 1, 2, \ldots, G$ and $m, n = 0, 1, 2$.

7. If $\varepsilon > \varepsilon_{max}$, repeat from step 4.

Needless to say, the (old) superscript stands for the older values of each unknown, while an unknown without such superscript represents the updated values. The (old) superscript is used with this meaning at the whole thesis. In the next sections, the actual diffusion model is presented in its eigenvalue problem and time dependent forms, which formalizes the application of this thesis.

2.5 Neutron diffusion: eigenvalue problem

One kind of problem that the present methodology is able to solve is the eigenvalue problem in neutron multi–group diffusion model, given by a diffusion differential equation,

*In all applications of this thesis, the author set Q_g and $f_{g\iota\kappa}$ as their zeroth and second order truncated Taylor series, respectively.

boundary condition, flux continuity condition and current density continuity condition:

$$-\nabla\cdot\left(D_g\nabla\phi_g\right)+\Sigma_{t_g}\phi_g-\sum_{g'=1}^{G}\Sigma_{s_{g'g}}\phi_{g'}=\frac{1}{k}\chi_g\sum_{g'=1}^{G}\nu_{g'}\Sigma_{f_{g'}}\phi_{g'}\ , \tag{2.27a}$$

valid for $\mathbf{r}\in V$, the boundary conditions,

$$\mathfrak{A}_g\hat{\mathbf{n}}\cdot\nabla\phi_g+\mathfrak{B}_g\phi_g=0\ , \tag{2.27b}$$

valid for $\mathbf{r}\in\partial V$, and the interface conditions: the flux continuity condition and the current density continuity condition,

$$\lim_{a\to 0}\left[\phi_g\left(\mathbf{r}+a\hat{\mathbf{x}}\right)-\phi_g\left(\mathbf{r}-a\hat{\mathbf{x}}\right)\right]=0 \tag{2.27c}$$

$$\lim_{a\to 0}\left[D_g\left(\mathbf{r}+a\hat{\mathbf{x}}\right)\nabla\phi_g\left(\mathbf{r}+a\hat{\mathbf{x}}\right)-D_g\left(\mathbf{r}-a\hat{\mathbf{x}}\right)\nabla\phi_g\left(\mathbf{r}-a\hat{\mathbf{x}}\right)\right]=\mathbf{0}\ , \tag{2.27d}$$

both valid for $\mathbf{r}\in(V-\partial V)$ and $g=1,2,\ldots,G$ unless specified, where G is the total number of energy groups. Here, $D_g=D_g\left(\mathbf{r}\right)$, $\Sigma_{t_g}=\Sigma_{t_g}\left(\mathbf{r}\right)$, $\Sigma_{s_{g'g}}=\Sigma_{s_{g'g}}\left(\mathbf{r}\right)$, ν_g, $\Sigma_{f_g}=\Sigma_{f_g}\left(\mathbf{r}\right)$ and χ_g are the nuclear parameters g–th group diffusion coefficient, g–th group total cross section, scattering cross section from g'–th to g–th group, average number of g–th group neutrons emitted by fission, g–th group fission cross section and g–th group integrated fission spectrum. Also, $\mathfrak{A}_g=\mathfrak{A}_g\left(\mathbf{r}\right)$ and $\mathfrak{B}_g=\mathfrak{B}_g\left(\mathbf{r}\right)$ are the g–th group boundary parameters, $\hat{\mathbf{n}}$ is the unit vector normal to the boundary's surface and $\hat{\mathbf{x}}$ is a unit vector (usually, not mandatorily, described as normal to an interface surface). k and $\phi_g=\phi_g\left(\mathbf{r}\right)$ are the unknowns dominant eigenvalue and its respective eigenfunctions, called effective multiplication factor and g–th group neutron scalar flux. $\mathbf{r}$ is the position vector and V is its domain. The boundary parameters always respect $|\mathfrak{A}_g|+|\mathfrak{B}_g|>0$ in their domain.

An usual configuration for this kind of problem is shown in figure 2.1a. It is noteworthy that most geometries involves orthogonal boundaries and step–wise constant nuclear and boundary parameters, like the one displayed in figure 2.1a. To solve this eigenvalue problem, we use the power method, which consists in iteratively updating the dominant eigenvalue and its respective eigenfunction until a stop criterion is satisfied. Further details about this method is found in Duderstadt and Hamilton, 1976. The iterations of power method consist

in solving

$$-\nabla\cdot\left(D_g\nabla\phi_g\right)+\Sigma_{t_g}\phi_g-\sum_{g'=1}^{G}\Sigma_{s_{g'g}}\phi_{g'}=s_g\ , \tag{2.28a}$$

$$\mathfrak{A}_g\hat{\mathbf{n}}\cdot\nabla\phi_g+\mathfrak{B}_g\phi_g=0\ , \tag{2.28b}$$

$$\lim_{a\to 0}\left[\phi_g\left(\mathbf{r}+a\hat{\mathbf{x}}\right)-\phi_g\left(\mathbf{r}-a\hat{\mathbf{x}}\right)\right]=0 \tag{2.28c}$$

$$\lim_{a\to 0}\left[D_g\left(\mathbf{r}+a\hat{\mathbf{x}}\right)\nabla\phi_g\left(\mathbf{r}+a\hat{\mathbf{x}}\right)-D_g\left(\mathbf{r}-a\hat{\mathbf{x}}\right)\nabla\phi_g\left(\mathbf{r}-a\hat{\mathbf{x}}\right)\right]=\mathbf{0}\ , \tag{2.28d}$$

where s_g is a fictitious source term,

$$s_g=\frac{1}{k^{(\mathrm{old})}}\chi_g\sum_{g'=1}^{G}\nu_{g'}\Sigma_{f_{g'}}\phi_{g'}^{(\mathrm{old})}\ , \tag{2.29}$$

updating both ϕ_g and k each iteration. (2.28) is used to update ϕ_g, and

$$k=k^{(\mathrm{old})}\frac{\iiint_V\sum_{g=1}^{G}\nu_g\Sigma_{f_g}\left(\mathbf{r}\right)\phi_g\left(\mathbf{r}\right)\mathrm{d}^3\mathbf{r}}{\iiint_V\sum_{g=1}^{G}\nu_g\Sigma_{f_g}\left(\mathbf{r}\right)\phi_g^{(\mathrm{old})}\left(\mathbf{r}\right)\mathrm{d}^3\mathbf{r}} \tag{2.30}$$

updates k.

The author uses this **{power method}** algorithm for the determination of these unknowns for a two–dimensional orthogonal geometry domain, with step–wise constant nuclear and boundary parameters:

1. Set ε_{max}.
2. First guess for k and ϕ_g, and $\varepsilon=\varepsilon_{max}+1$.
3. Store the old solution: $\phi_g^{(\mathrm{old})}\leftarrow\phi_g$ and $k^{(\mathrm{old})}\leftarrow k$.
4. Calculate s_g with (2.29), using $\phi_g^{(\mathrm{old})}$ and $k^{(\mathrm{old})}$.
5. Use **{global}** to solve (2.28) updating ϕ_g.
6. Update k with (2.30).

7. Calculate ε with $\varepsilon \leftarrow \max\left(\left|\frac{k-k^{(\text{old})}}{k+k^{(\text{old})}}\right|, \left|\frac{\phi_g-\phi_g^{(\text{old})}}{\phi_g+\phi_g^{(\text{old})}}\right|\right)$.

8. If $\varepsilon > \varepsilon_{max}$, repeat from step 3.

2.6 Neutron space kinetics: time dependent problem

Another kind of problem that can be solved by the present methodology is the time dependent problem of neutron diffusion in nuclear reactor cores. This problem consists in the time dependent multi–group neutron diffusion model

$$\frac{1}{v_g}\frac{\partial \phi_g}{\partial t} - \nabla\cdot(D_g\nabla\phi_g) + \Sigma_{t_g}\phi_g - \sum_{g'=1}^{G}\Sigma_{s_{g'g}}\phi_{g'} = \chi_{F_g}(1-\beta)\sum_{g'=1}^{G}\nu_{g'}\Sigma_{f_{g'}}\phi_{g'} + \chi_{D_g}\sum_{p=1}^{P}\lambda_p C_p + \mathfrak{S}_g \quad (2.31a)$$

for $g = 1, 2, \ldots G$ and

$$\frac{\partial C_p}{\partial t} + \lambda_p C_p = \beta_p \sum_{g=1}^{G}\nu\Sigma_{f_g}\phi_g \quad (2.31b)$$

for $p = 1, 2, \ldots, P$ unless specified. These equations are valid for any $\mathbf{r} \in V$ and $t \in [t_0, \infty)$. The boundary conditions are

$$\mathfrak{A}_g\hat{\mathbf{n}}\cdot\nabla\phi_g + \mathfrak{B}_g\phi_g = \mathfrak{C}_g \ , \quad (2.31c)$$

valid for $\mathbf{r} \in \partial V$ and $t \in [t_0, \infty)$, and the interface conditions are

$$\lim_{a\to 0}\left[\phi_g(\mathbf{r}+a\hat{\mathbf{x}}) - \phi_g(\mathbf{r}-a\hat{\mathbf{x}})\right] = 0 \quad (2.31d)$$

$$\lim_{a\to 0}\left[D_g(\mathbf{r}+a\hat{\mathbf{x}})\nabla\phi_g(\mathbf{r}+a\hat{\mathbf{x}}) - D_g(\mathbf{r}-a\hat{\mathbf{x}})\nabla\phi_g(\mathbf{r}-a\hat{\mathbf{x}})\right] = \mathbf{0} \ , \quad (2.31e)$$

both valid for $\mathbf{r} \in (V - \partial V)$ and $t \in [t_0, \infty)$. The initial conditions are expressed as

$$\phi_g = \phi_{g0} \ , \tag{2.31f}$$

$$C_p = C_{p0} \ , \tag{2.31g}$$

both valid for $\mathbf{r} \in V$ and $t = t_0$. Here, P is the total number of delayed neutron precursors groups and V stands for the spacial domain (x, y and z in rectangular geometry). In addition to the previously declared parameters ($D_g = D_g(\mathbf{r}, t)$, $\Sigma_{t_g} = \Sigma_{t_g}(\mathbf{r}, t)$, $\Sigma_{s_{g'g}} = \Sigma_{s_{g'g}}(\mathbf{r}, t)$, ν_g, $\Sigma_{f_g} = \Sigma_{f_g}(\mathbf{r}, t)$, $\mathfrak{A}_g = \mathfrak{A}_g(\mathbf{r}, t)$, $\mathfrak{B}_g = \mathfrak{B}_g(\mathbf{r}, t)$), $\mathfrak{S}_g = \mathfrak{S}_g(\mathbf{r}, t)$, v_g, β, β_p and λ_p are the g–th group external source term, average g–th group neutron speed, delayed neutron fraction, p–th group delayed neutron fraction, p–th group decay constant, respectively. χ_{F_g} and χ_{D_g} are the g–th group integrated prompt and delayed fission spectra, respectively. $\mathfrak{C}_g = \mathfrak{C}_g(\mathbf{r}, t)$ is the independent term of the boundary conditions. $\phi_{g0} = \phi_{g0}(\mathbf{r})$ and $C_{p0} = C_{p0}(\mathbf{r})$ are the known fluxes and concentrations at the time $t = t_0$. $\phi_g = \phi_g(\mathbf{r}, t)$ and $C_p = C_p(\mathbf{r}, t)$ are the unknowns g–th group neutron scalar flux and p–th group delayed neutrons precursors concentration, and t is the variable time. This boundary and initial value problem is commonly called the multi–group neutron space kinetics model.

PEM is used to solved the neutron space kinetics model for rectangular geometry, however it is used a numerical trick first. To treat the time dependent problem it is used the one step average Euler method as in Burden and Faires, 2008, also called step–wise constant approximation, which consists in approximating the time dependent data and the unknowns as constants in a regular $2\Delta t$ time step, and it requires a very small Δt to obtain precision. In other words, the continuous time is displayed in a discrete form, so instead of $t \in [t_0, \infty)$, we have $t_q \in \{t_0, t_1, \dots\}$ for $q = 0, 1, 2, \dots$. For a time dependent function $f = f(t)$, it is written as its discrete form f_q

$$f(t_q) = f_q \approx \frac{f(t_q + \Delta t) + f(t_q - \Delta t)}{2} \tag{2.32}$$

for $q = 1, 2, \dots$. As a concluding statement, the known time dependent functions like the nuclear parameters are fully determined, and the unknowns are going to be determined by solving the problem. This procedure also uses the so–called analytical continuation, which

consists in using the present step's solution to determine the initial condition for the next time step in an initial value problem. Time derivatives are treated as its central finite difference form

$$\left[\frac{\mathrm{d}f}{\mathrm{d}t}\right]_q \approx \frac{f\left(t_q + \Delta t\right) - f\left(t_q - \Delta t\right)}{2\Delta t} \ . \tag{2.33}$$

Combining (2.32) with (2.33), we got the three main relations for the unknowns:

$$\left[\frac{\mathrm{d}f}{\mathrm{d}t}\right]_q \approx \frac{f_q - f\left(t_q - \Delta t\right)}{\Delta t} \ , \tag{2.34a}$$

$$f\left(t_q + \Delta t\right) = 2f_q - f\left(t_q - \Delta t\right) \ , \tag{2.34b}$$

$$f\left(t_q + \Delta t\right) = f\left(t_{q+1} - \Delta t\right) \ . \tag{2.34c}$$

In the sense of applying this approximation to the unknowns, we set the time $t_q - \Delta t$ always as the time instant for the initial condition, so $f\left(t_q - \Delta t\right)$ is always known. In this sense, we can avoid using the q subscript and write (2.34) again with $f \equiv f_q$ and $f_0 \equiv f\left(t_q - \Delta t\right)$, bearing in mind that f_0 is updated every time step

$$\left[\frac{\mathrm{d}f}{\mathrm{d}t}\right]_q \approx \frac{f - f_0}{\Delta t} \ , \tag{2.35a}$$

$$f_0 \leftarrow 2f - f_0 \ , \tag{2.35b}$$

Before using the constant approximation on (2.31), the precursors deserve a special focus, as their differential equation has derivatives in t only. Using (2.35) on (2.31b) results in

$$C_p = \frac{\Delta t}{1 + \lambda_p \Delta t}\left[\beta_p \sum_{g=1}^{G} \nu_g \Sigma_{f_g} \phi_g + \frac{C_{p0}}{\Delta t}\right] \tag{2.36}$$

Substituting each term on (2.31) as its proper approximation and using (2.36), we get a

pseudo boundary value problem in ϕ_g to solve for each time step:

$$\frac{1}{v_g}\frac{\phi_g - \phi_{g0}}{\Delta t} - \nabla\cdot(D_g\nabla\phi_g) + \Sigma_{t_g}\phi_g - \sum_{g'=1}^{G}\Sigma_{s_{g'g}}\phi_{g'} = \chi_{F_g}(1-\beta)\sum_{g'=1}^{G}\nu_{g'}\Sigma_{f_{g'}}\phi_{g'} + \chi_{D_g}\left[\sum_{p=1}^{P}\frac{\lambda_p\beta_p\Delta t}{1+\lambda_p\Delta t}\right]\sum_{g'=1}^{G}\nu_{g'}\Sigma_{f_{g'}}\phi_{g'} + \chi_{D_g}\sum_{p=1}^{P}\frac{\lambda_p C_{p0}}{1+\lambda_p\Delta t} + \mathfrak{S}_g \ , \quad (2.37a)$$

valid for any $\mathbf{r}\in V$,

$$\mathfrak{A}_g\hat{\mathbf{n}}\cdot\nabla\phi_g + \mathfrak{B}_g\phi_g = \mathfrak{C}_g \ , \quad (2.37b)$$

valid for $\mathbf{r}\in\partial V$, and

$$\lim_{a\to 0}\left[\phi_g(\mathbf{r}+a\hat{\mathbf{x}}) - \phi_g(\mathbf{r}-a\hat{\mathbf{x}})\right] = 0 \ , \quad (2.37c)$$

$$\lim_{a\to 0}\left[D_g(\mathbf{r}+a\hat{\mathbf{x}})\nabla\phi_g(\mathbf{r}+a\hat{\mathbf{x}}) - D_g(\mathbf{r}-a\hat{\mathbf{x}})\nabla\phi_g(\mathbf{r}-a\hat{\mathbf{x}})\right] = \mathbf{0} \ , \quad (2.37d)$$

both valid for $\mathbf{r}\in(V-\partial V)$. Note, (2.37) considers the time dependent parameters and unknowns as their constant approximation at a generic time t_q, but they depend only on the variable position, as mentioned before.

As it is known from the literature, the error for this kind of approximation is $O(4\Delta t^2)$, so it might work well for smooth nuclear parameters in the time variable. Fortunately, in practice the nuclear parameters vary very slowly with time, so this approximation does not result in unacceptable error. It is noteworthy that if the **{local}** algorithm were adapted to t dependent cases, with first order polynomial expansion in t like

$$\phi_g = \sum_{\ell=0}^{1}\sum_{m=0}^{2}\sum_{n=0}^{2}\varphi_{g\ell mn}t^\ell x^m y^n \ , \quad (2.38)$$

the set of linear equations to be solved each step would be similar to (2.37), so they would produce the same results with the same error estimate. Using the initial condition the results for the first steps are determined by (2.37). Then, the author chose to slightly modify the constant approximation method in a way that fits the proposed methodology. Instead of

calculating the initial conditions by (2.35b), the initial conditions for the next step are obtained by

$$\phi_{g0}^{[r]} = \frac{1}{2\Delta x \Delta y} \int_{-\Delta x}^{\Delta x} \int_{-\Delta y}^{\Delta y} \phi_g^{[r]}(x,y)\,\mathrm{d}y\mathrm{d}x - \phi_{g0}^{[r](\mathrm{old})} \ , \tag{2.39a}$$

$$C_{p0}^{[r]} = \frac{1}{2\Delta x \Delta y} \int_{-\Delta x}^{\Delta x} \int_{-\Delta y}^{\Delta y} C_p^{[r]}(x,y)\,\mathrm{d}y\mathrm{d}x - C_{p0}^{[r](\mathrm{old})} \ , \tag{2.39b}$$

and so on, according to (2.34). Obviously, x and y are local $[r]$ coordinates. This modification is made to be consistent with the methodology: the initial conditions are constants in $[r]$.

The author used the **{time dependent}** algorithm for the determination of these unknowns for a two–dimensional orthogonal geometry domain, with time dependent step–wise constant nuclear and boundary parameters:

1. Set the time step $2\Delta t$, initial time $t_0 = -\Delta t$, $q = 0$ and a maximum time limit T.
2. Make $q \leftarrow q + 1$.
3. Set $t_q = t_{q-1} + 2\Delta t$ and approximate all parameters as (2.32).
4. Use **{global}** to solve (2.37).
5. Calculate C_p with (2.36).
6. Update the initial conditions with (2.39).
7. If $t_q < T$, repeat from step 2.

3 TEST CASES

In this chapter some two–dimensional neutron diffusion test cases are shown, considering the eigenvalue and the time dependent problem. For all cases, $G = 2$ (group $g = 1$ is called fast group, $g = 2$ is called thermal group), $\Sigma_{s_{gg'}} = 0$ except $\Sigma_{s_{12}}$ and $\chi_{F_1} = \chi_{D_1} = \chi_1 = 1$, $\chi_{F_2} = \chi_{D_2} = \chi_2 = 0$. For both the power method and the iterative SOR algorithms, $\varepsilon_{max} = 10^{-8}$. In the time dependent cases, $\Delta t = 10^{-4}\ s$, and their initial conditions are the solutions of the eigenvalue steady state problems. Reducing Δt does not significantly reduce the error in any case, and in some cases reducing it to $\Delta t = 10^{-5}$ or less makes the program fail due to numerical over stack (lack of memory). All parameters with time dependence will be specified in each case, and no external source term is considered $\mathfrak{S}_g = 0$). The results are displayed as a geometric table for each given rectangular region, as displayed in figure 3.1.

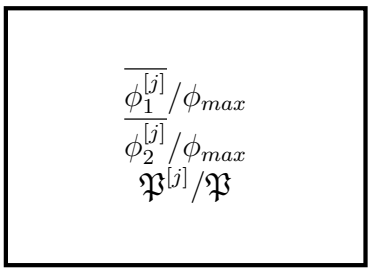


Figure 3.1: Results display in a given rectangle.

In this figure, $j = 1, 2, \ldots, J$, where J is the total number of given regions. $\overline{\phi_g^{[j]}}$ is the j–th region average neutron scalar flux (in time t, when applicable), given by

$$\overline{\phi_g^{[j]}} = \frac{1}{A^{[j]}} \iint\limits_{\mathfrak{D}^{[j]}} \phi_g^{[j]}\,(x, y, t)\,\mathrm{d}y\mathrm{d}x\ , \tag{3.1}$$

where $\mathfrak{D}^{[j]}$ is the j–th local domain and $A^{[j]}$ is its area, $\mathfrak{P}^{[j]}$ is the number of neutrons

generated by fission in the j–th region, given by

$$\mathfrak{P}^{[j]} = \int_{-\Delta x}^{\Delta x} \int_{-\Delta y}^{\Delta y} \sum_{g=1}^{G} \Sigma_{f_g}^{[j]}(x,y,t)\, \phi_g^{[j]}(x,y,t)\, \mathrm{d}y\mathrm{d}x \ , \tag{3.2}$$

where $\mathfrak{E}$ is the average energy liberated by neutron by fission. ϕ_{max} and $\mathfrak{P}$ are the maximum neutron scalar flux in t and total number of neutrons generated by fission, given by

$$\phi_{max} = \max_{(x,y)\in V} (|\phi_1|, |\phi_2|) \tag{3.3}$$

and

$$\mathfrak{P} = \sum_{j=1}^{J} \mathfrak{P}^{[j]} \ . \tag{3.4}$$

In (3.3) and (3.4), x and y are the global variables, and V is the entire geometry domain. For the eigenvalue steady state cases, hence the initial conditions for the time dependent cases, the fluxes and power are normalized so $\phi_{max} = 1$:

$$\phi_g \leftarrow \phi_g / \phi_{max} \ . \tag{3.5}$$

Also, the results are displayed in percentages, and ϕ_{max} and $\mathfrak{P}$ are always written in the text for each case, together with the values of Δx, Δy and t, when applicable. It is important to note that this form of displaying results considers the rectangle regions given by the geometry itself, so they are not necessarily equivalent to the rectangular mesh. All results were obtained in an Intel Core i5–4500U CPU @ 1.60 GHz, 4.0 GB RAM in a Windows 8.1 64–bit operational system, using Code::Blocks v13.12 IDE to compile and run C programs made by the author with the present methodologies and algorithms.

As a final comment before the test cases' results, the nuclear parameters undeclared so far are the capture and absorption cross sections $\Sigma_{c_g} = \Sigma_{c_g}(x,y,t)$, $\Sigma_{a_g} = \Sigma_{a_g}(x,y,t)$.

Their relations with other parameters are

$$\Sigma_{a_g} = \Sigma_{c_g} + \Sigma_{f_g} \ , \tag{3.6a}$$

$$\Sigma_{t_g} = \Sigma_{a_g} + \sum_{g'=1}^{G} \Sigma_{s_{gg'}} \ . \tag{3.6b}$$

Also, some configurations will use the geometric buckling B^2 in their model. It is an approximation of the behaviour of the second derivative in the z axis and, in this configuration, will be a constant number for all regions and groups. That way, a new Σ_{t_g} is written as a modification to fit the described methodology:

$$\Sigma_{t_g} \leftarrow \Sigma_{t_g} + B^2 D_g \ . \tag{3.7}$$

As β is the total delayed neutron fraction,

$$\beta = \sum_{p=1}^{P} \beta_p \ . \tag{3.8}$$

All parameters are described with its usual dimensions: D, Δx and Δy in cm; t, T and Δt in s, Σ in cm^{-1}, B^2 in cm^{-2}, λ in s^{-1}, v in cm/s, ϕ in $cm^{-2}s^{-1}$, $\mathfrak{P}$ in $cm^{-1}s^{-1}$ and the rest is dimensionless. In all configurations there are figures with their geometry, where $\hat{\mathbf{n}}$ are the outgoing unit vector at the boundary. In these figures, dashed lines mean given regions, thick lines mean interface from one material to another, circled numbers indicate a material and similar hatches mean similar material. Tables with its nuclear parameters are displayed in their following.

3.1 Configuration 1

This geometry in particular is usual in papers about two–dimensional neutron kinetics algorithms. Here, the mesh is homogeneous with $\Delta x = \Delta y = 5$. Reducing Δx and Δy does not make significant changes in the results.

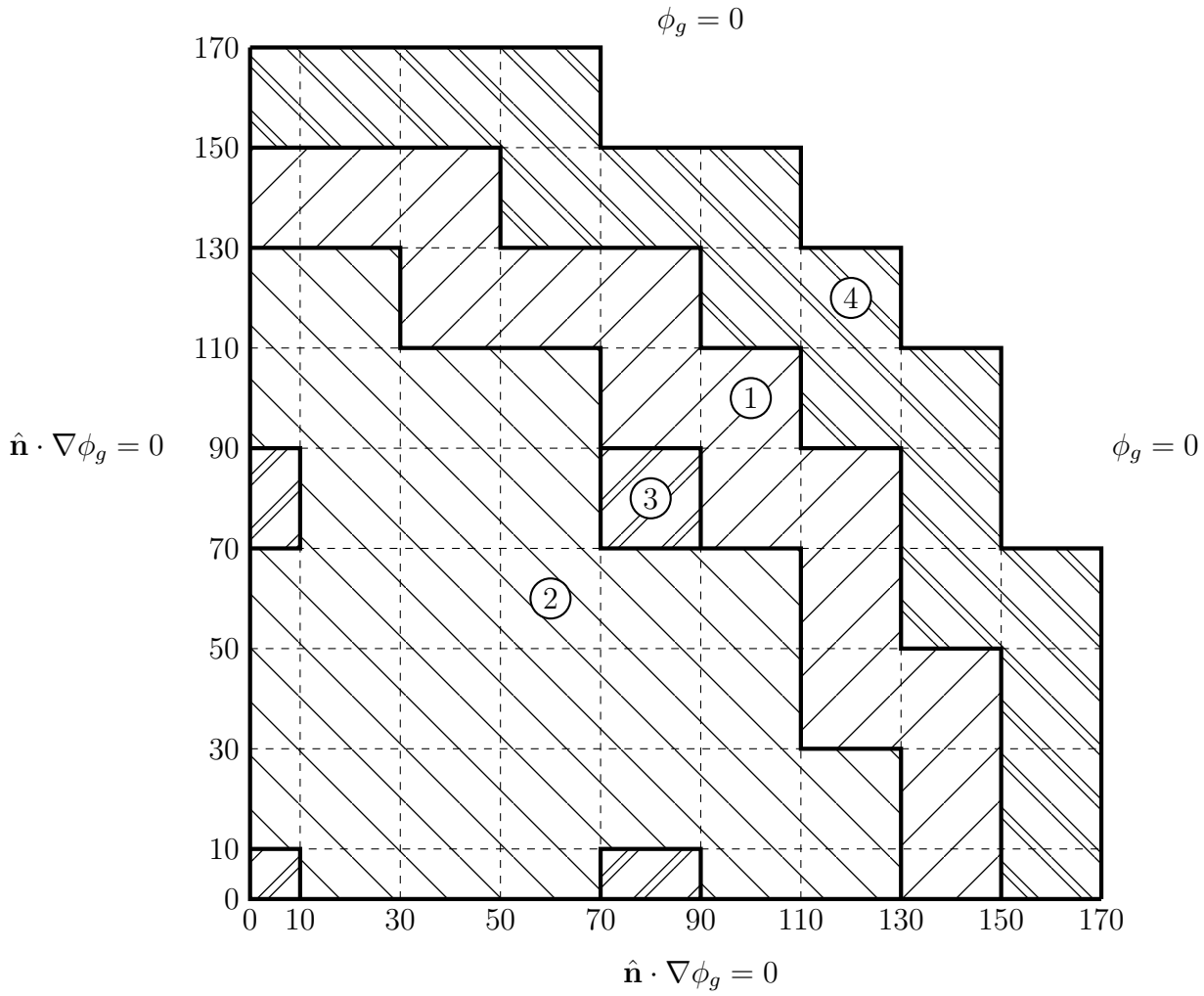
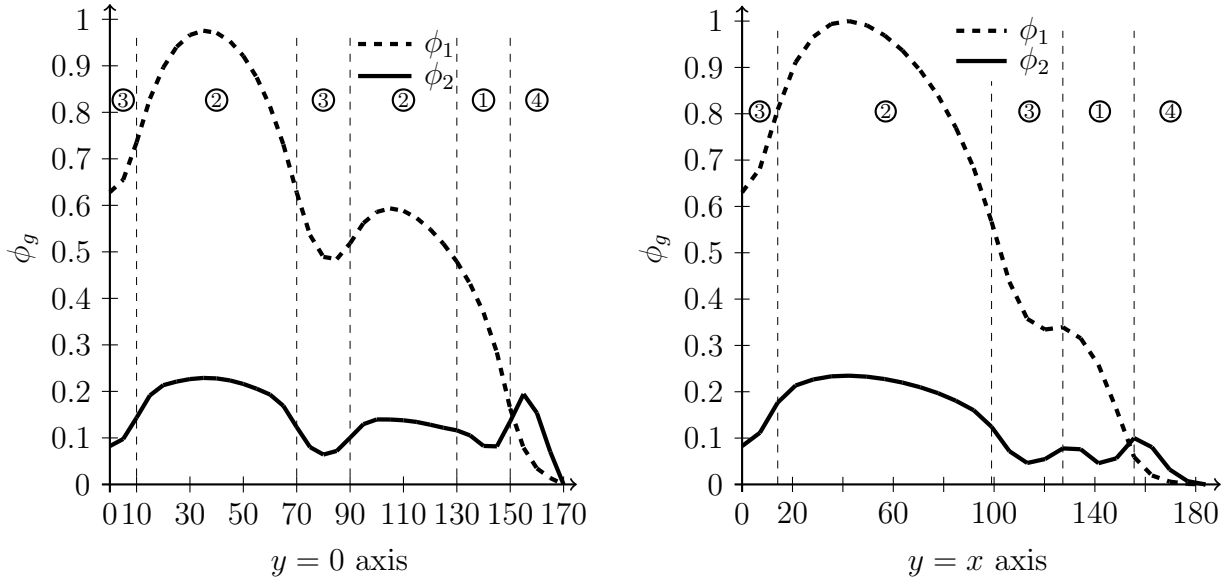


Figure 3.2: Geometry and boundary conditions for configurations 1,2 and 3.

The geometry used comes from Argonne Code Center [1977] with its representation of boundary conditions and regions shown in figure 3.2. In this case, no time dependent cases are considered due to the lack of kinetic parameters and because the system is super–prompt–critical, so only the results of the eigenvalue problem is shown.

Material	D_1	D_2	Σ_{a_1}	Σ_{a_2}	$\nu_2\Sigma_{f_2}$	$\Sigma_{s_{12}}$
1	1.5	0.4	0.01	0.08	0.135	0.02
2	1.5	0.4	0.01	0.085	0.135	0.02
3	1.5	0.4	0.01	0.13	0.135	0.02
4	2	0.3	0	0.01	0	0.04

Table 3.1: Nuclear parameters for configuration 1.



(a) Fluxes profiles in $y = 0$.

(b) Fluxes profiles in $y = x$.

Figure 3.3: Fluxes graphics in some axes for configuration 1.

Table 3.1 shows the nuclear parameters for the eigenvalue problem. In this configuration, all $\Sigma_{f_1} = 0$ and the geometric buckling is $B^2 = 8 \times 10^{-5}$ for all regions and all energy groups.

Figure 3.4 shows the results for the eigenvalue problem, with maximum eigenvalue $k = 1.02903$, computational time 7.6 s, $\phi_{max} = 1$ and $\mathfrak{P} = 346.217$ with 2169 unknowns. ϕ_1 has its maximum value 1 at $(x, y) = (30, 30)$, ϕ_2 has its maximum value 0.463150 at $(x, y) = (55, 135), (135, 55)$. Figure 3.3 shows the graphics for the neutron fluxes in the global axes $y = 0$ and $y = x$. The results does not vary from the ones in Argonne Code Center [1977] more than 1%. The next configuration shows the results to a similar case, but with kinetic parameters and time dependent cases.

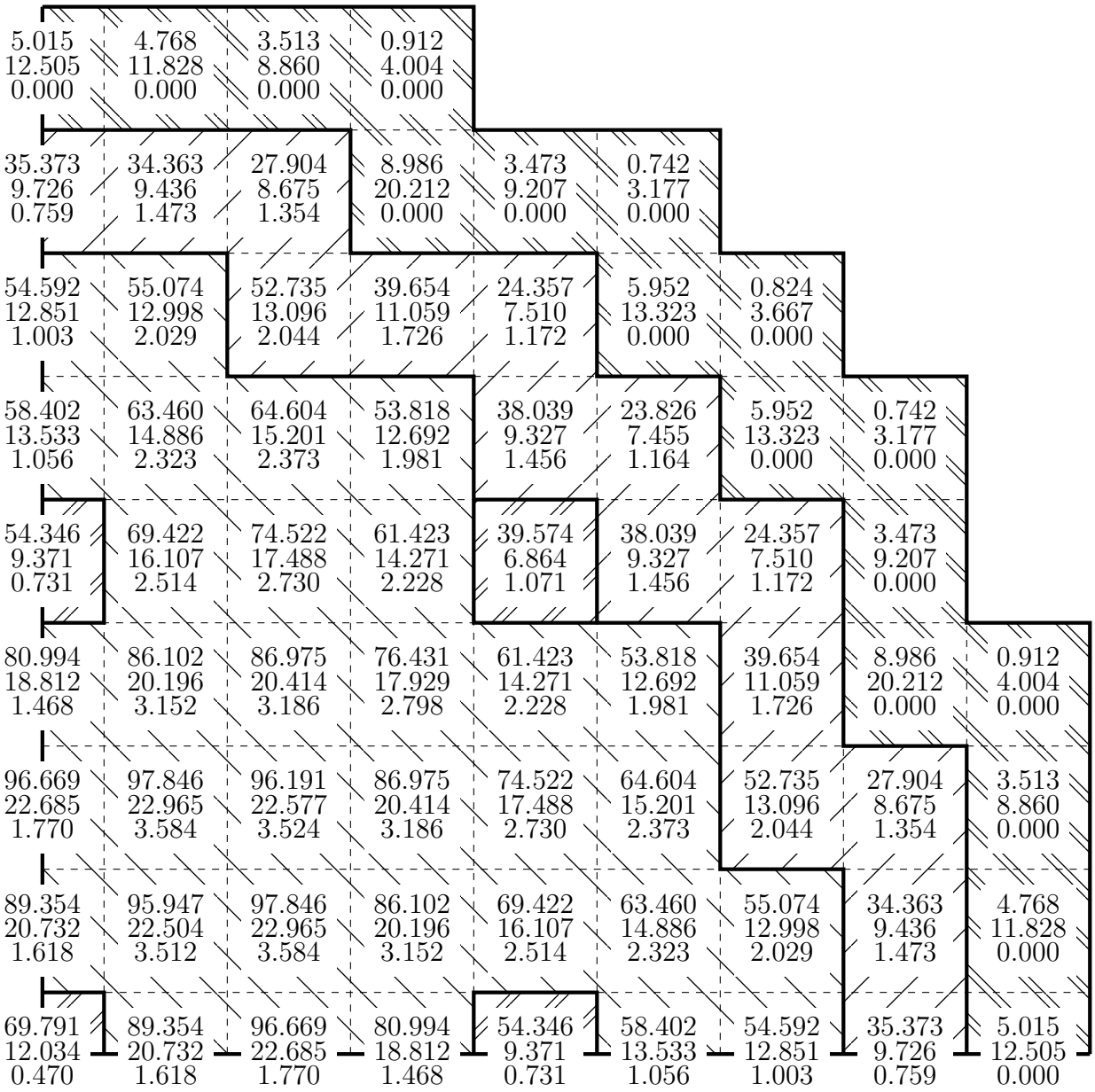


Figure 3.4: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for configuration 1, eigenvalue problem.

Material	D_1	D_2	Σ_{a_1}	Σ_{a_2}	$\nu_2\Sigma_{f_2}$	$\Sigma_{s_{12}}$	λ_p		$\beta_p \times 10^3$	
1	1.5	0.4	0.01	0.0825	0.135	0.02	0.0127	0.0317	0.285	1.5975
2	1.5	0.4	0.01	0.0875	0.135	0.02	0.115	0.311	1.41	3.0525
3	1.5	0.4	$*$	$*$	0.135	0.02	1.4	3.87	0.96	0.195
4	2	0.3	0	0.01	0	0.04	$v_1 = 1 \times 10^7$		$v_2 = 3 \times 10^5$	

$*$: parameters substituted by (3.9).

Table 3.2: Parameters for configuration 2.

3.2 Configuration 2

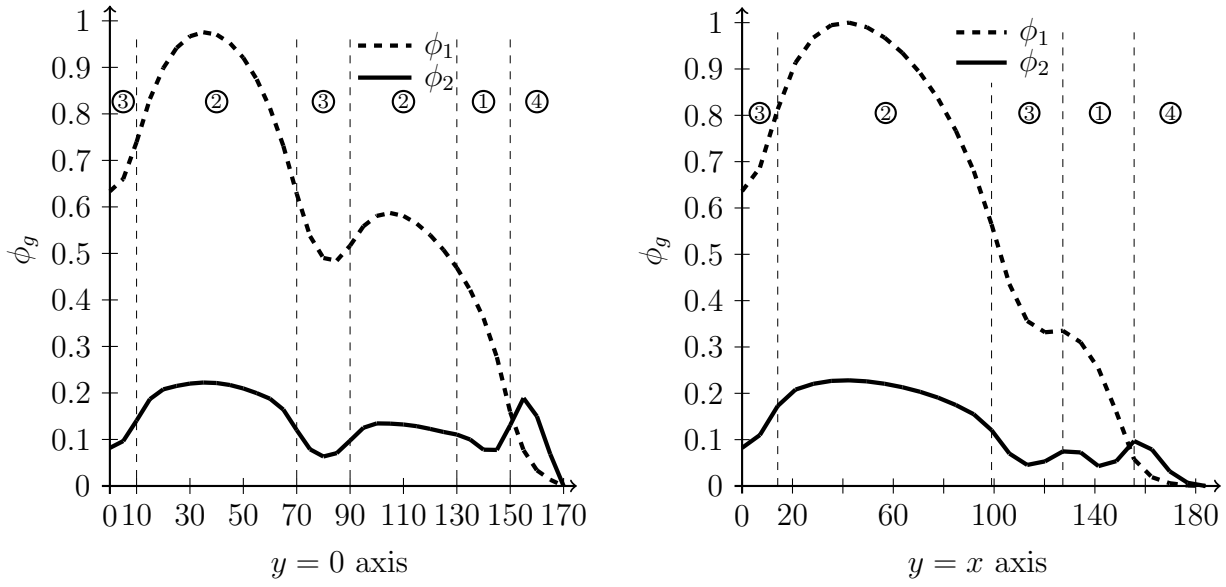
This configuration has the same geometry as configuration 1 given by figure 2.1a, however with different nuclear parameters. The geometry and nuclear parameters came from Argonne Code Center [1977] with a slight modification, and the kinetic parameters v_g, λ_p and β_p came from Ceolin et al. [2015], as Argonne Code Center [1977] does not contain any kinetic parameter. The methodology used homogeneous mesh with $\Delta x = \Delta y = 5$, and reducing their value does not change significantly any results. The eigenvalue problem with parameters equivalent to $t = 0$ resulted in maximum eigenvalue $k = 0.999741$ and the code solved 2169 unknowns each time step. The computational time for eigenvalue and time dependent problems were 9.8 s and 1645.2 s, respectively. Table 3.2 shows the parameters used in this case. Table 3.3 shows the values and positions of maximum ϕ_g and $\mathfrak{P}$ for $t = 0, 1, 3, 5, 7, 8$. Figure 3.7 shows the results for this problem, and figure 3.5 shows the graphics for the neutron fluxes in the global axes $y = 0$ and $y = x$. Figures 3.8 to 3.12 show the results for $t = 1, 3, 5, 7, 8$ respectively. Figure 3.6 shows the behavior of $\mathfrak{P}$ in time. The minimum $\mathfrak{P}$ happened at about $t = 3.12$ with value $\mathfrak{P} = 237.987$. The maximum $\mathfrak{P}$ happened at about $t = 5.12$ with value $\mathfrak{P} = 643.906$. In this configuration, all $\Sigma_{f_1} = 0$, the geometric buckling is $B^2 = 8 \times 10^{-5}$ for all regions and all energy groups, and Σ_{a_g} in

material 3 are time dependent functions, given by

$$\Sigma_{a_1} = \begin{cases} 0.01\,(1+0.1t)\ , & 0 \le t < 1 \\ 0.01\,(0.9+0.1\,(t-1))\ , & 1 \le t < 3 \\ 0.01\,(1.1-0.1\,(t-3))\ , & 3 \le t < 5 \\ 0.01\,(0.9+0.1\,(t-5))\ , & 5 \le t < 7 \\ 0.01\,(1.1-0.1\,(t-7))\ , & 7 \le t \le 8 \end{cases} \tag{3.9a}$$

$$\Sigma_{a_2} = \begin{cases} 0.1325\,(1+0.1t)\ , & 0 \le t < 1 \\ 0.1325\,(0.9+0.1\,(t-1))\ , & 1 \le t < 3 \\ 0.1325\,(1.1-0.1\,(t-3))\ , & 3 \le t < 5 \\ 0.1325\,(0.9+0.1\,(t-5))\ , & 5 \le t < 7 \\ 0.1325\,(1.1-0.1\,(t-7))\ , & 7 \le t \le 8 \end{cases} \tag{3.9b}$$

Note, in table 3.2 they are marked with the symbol $*$.



(a) Fluxes profiles in $y = 0$.

(b) Fluxes profiles in $y = x$.

Figure 3.5: Fluxes graphics in some axes for configuration 2, eigenvalue problem.

	ϕ_1		ϕ_2		
t	value	position	value	position	$\mathfrak{P}$
0	1	$(30, 30)$	0.294203	$(55, 135), (135, 55)$	333.580
1	1.89058	$(25, 30), (30, 25)$	0.500120	$(60, 135), (135, 60)$	651.536
3	0.707430	$(30, 30)$	0.192343	$(55, 135), (135, 55)$	237.167
5	1.88953	$(25, 30), (30, 25)$	0.500378	$(60, 135), (135, 60)$	651.512
7	0.710987	$(30, 30)$	0.193610	$(55, 135), (135, 55)$	238.539
8	0.904799	$(30, 30)$	0.243703	$(60, 135), (135, 60)$	307.256

Table 3.3: Maximum values and positions (x, y) of fluxes and $\mathfrak{P}$ for several t, configuration 2.

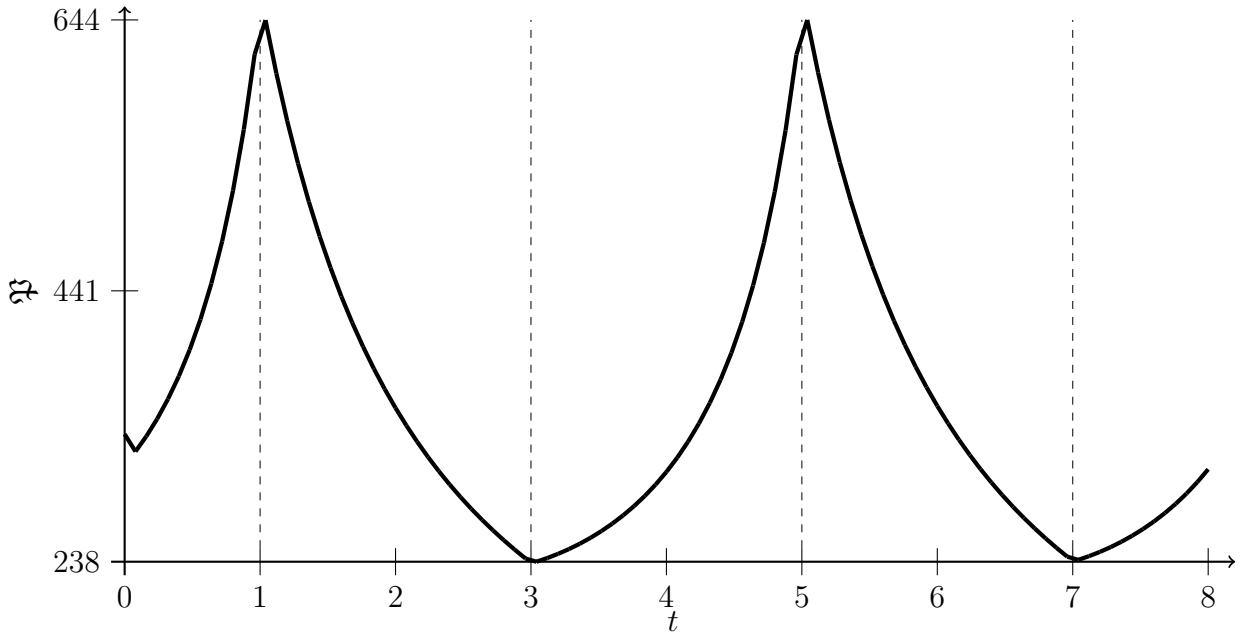


Figure 3.6: $\mathfrak{P}$ in time for configuration 2.

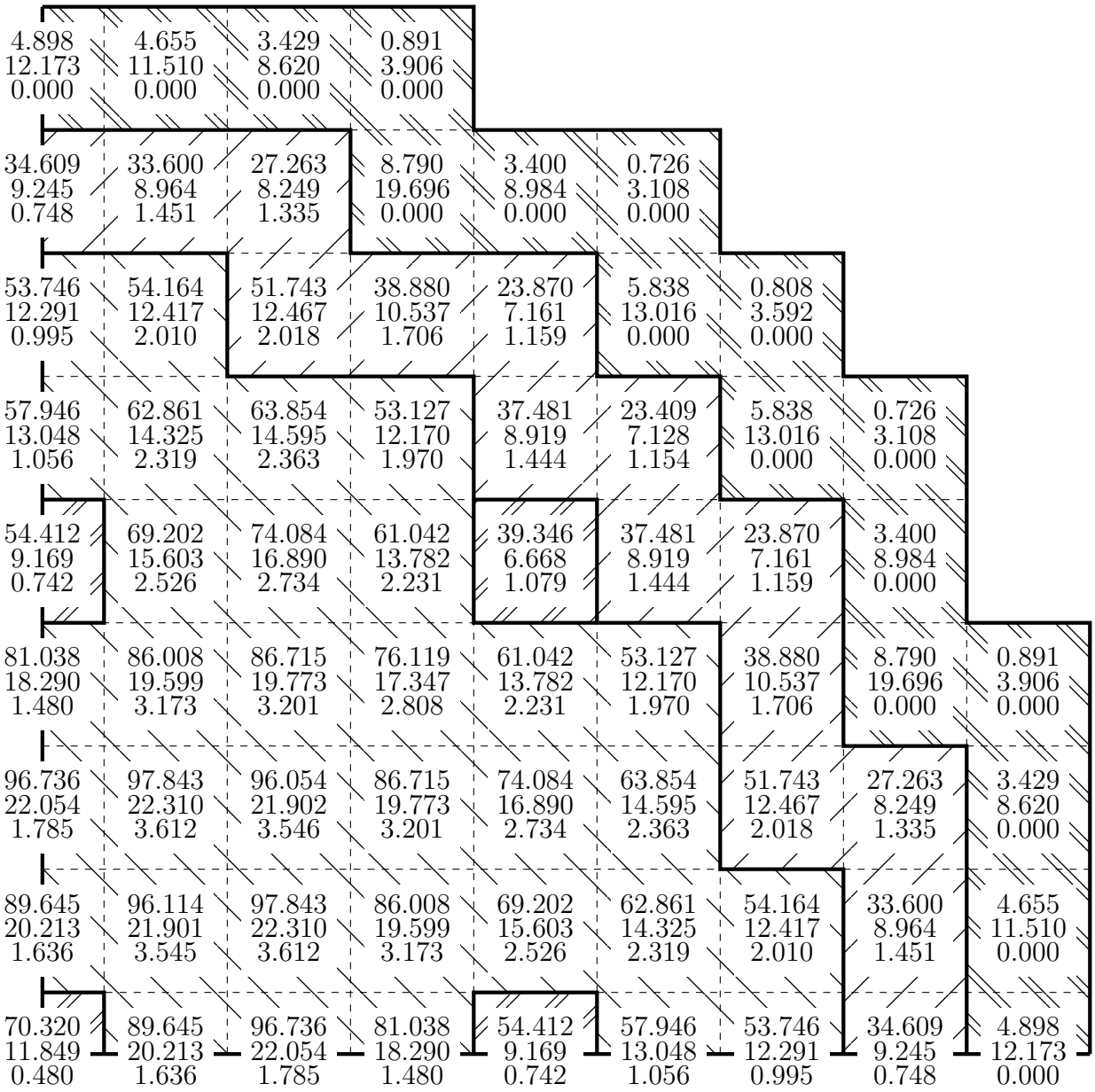


Figure 3.7: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for configuration 2, eigenvalue problem.

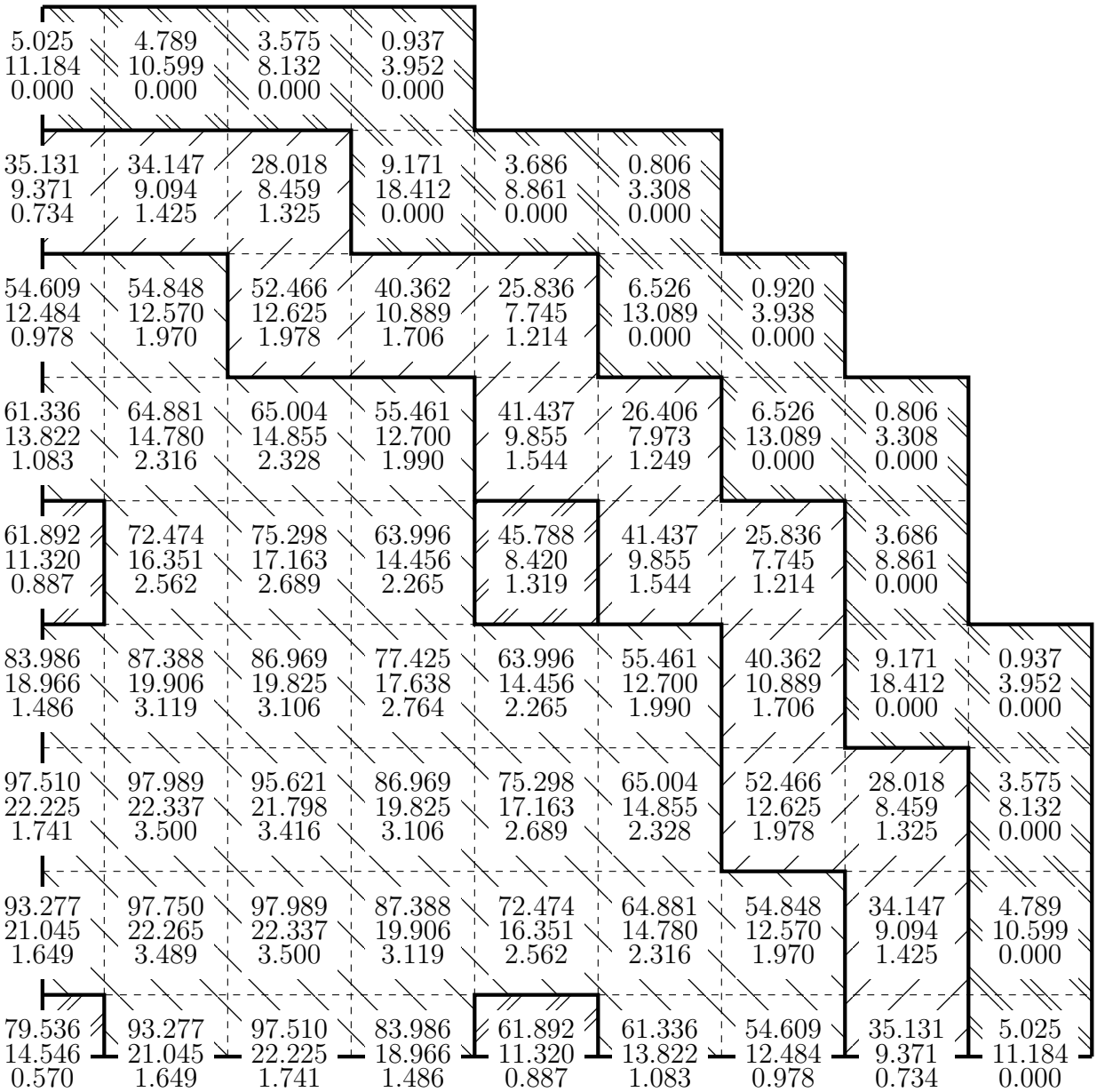


Figure 3.8: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for configuration 2, $t = 1$.

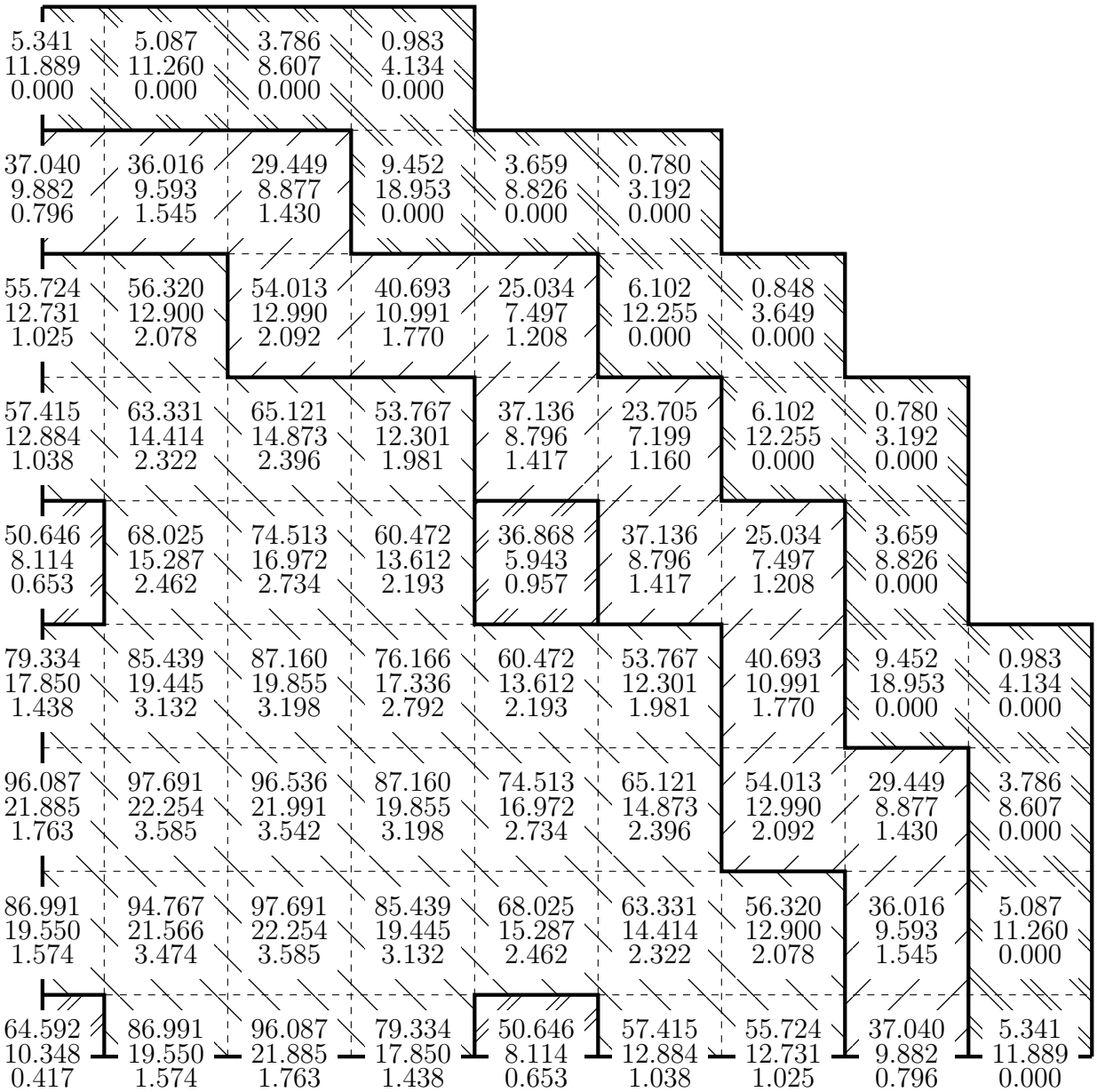


Figure 3.9: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for configuration 2, $t = 3$.

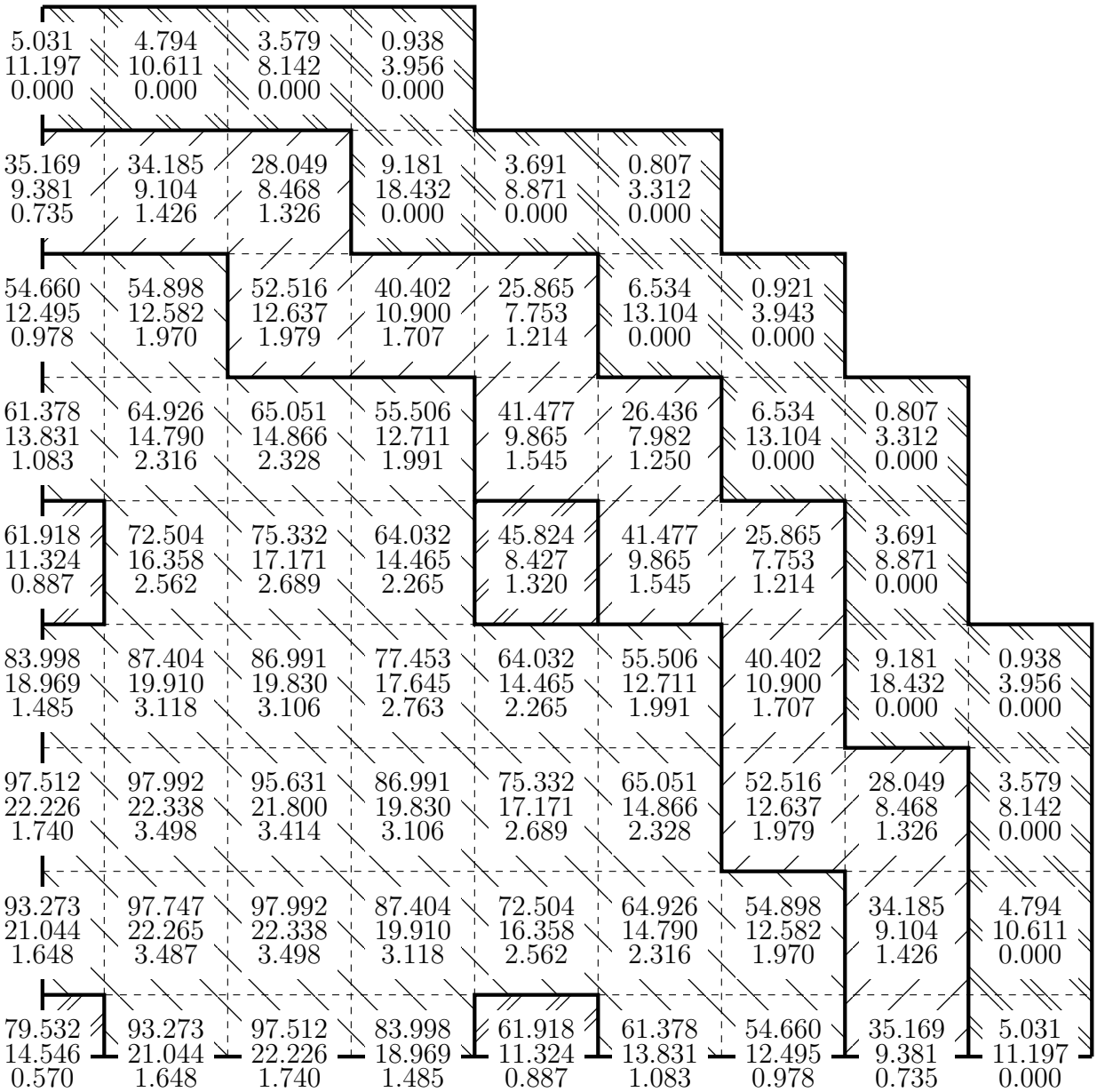


Figure 3.10: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for configuration 2, $t = 5$.

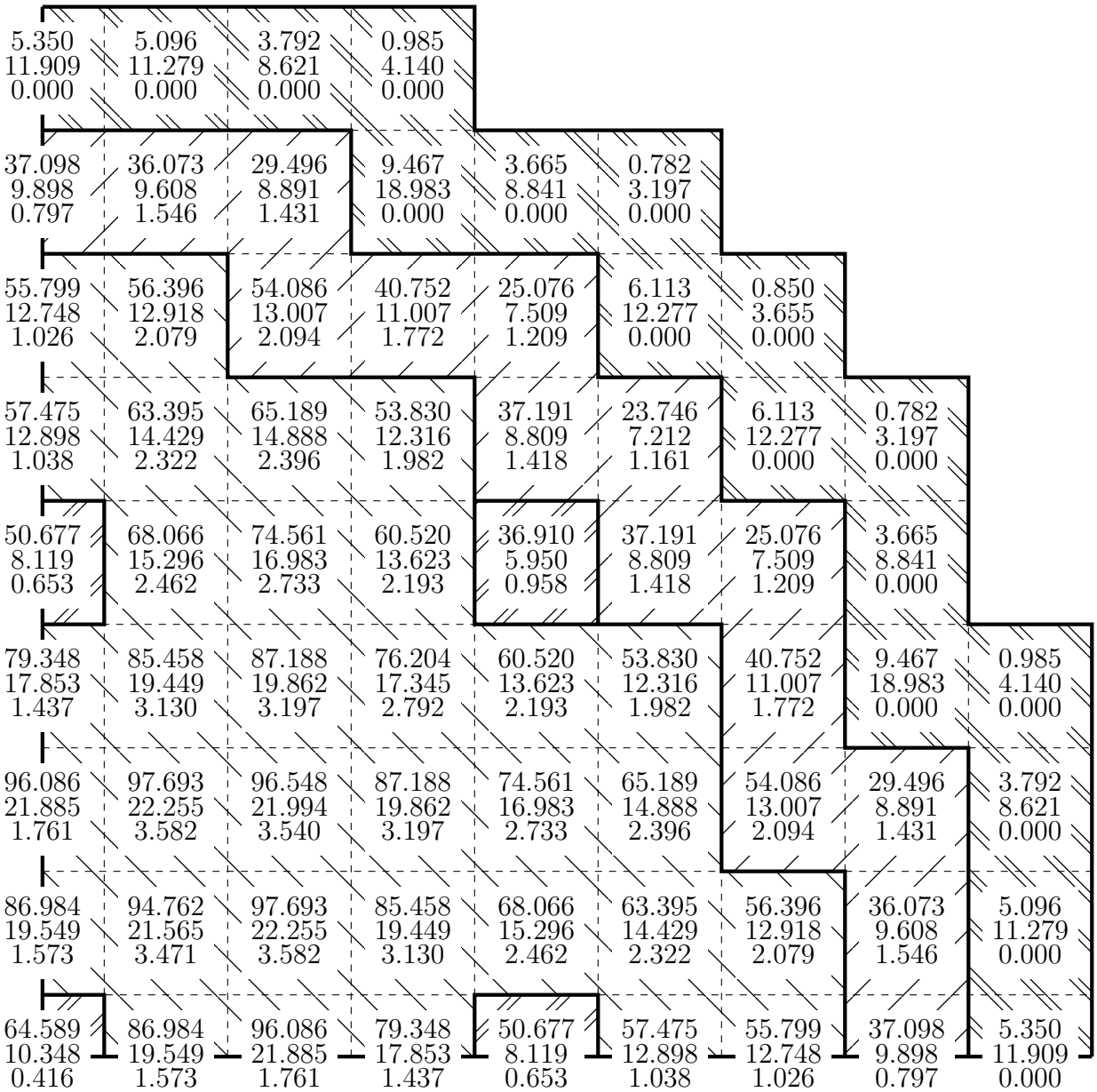


Figure 3.11: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for configuration 2, $t = 7$.

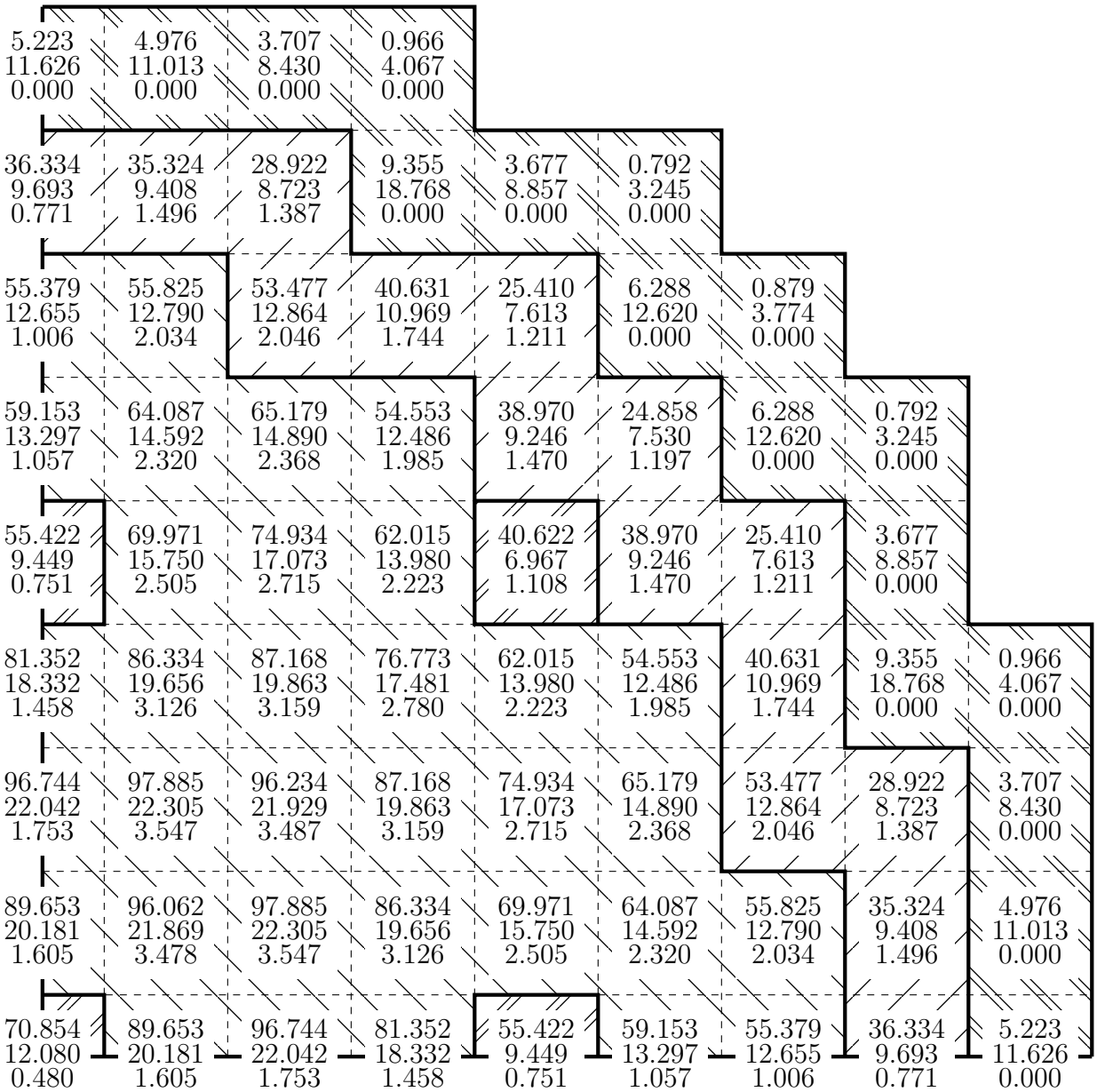


Figure 3.12: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for configuration 2, $t = 8$.

3.3 Configuration 3

This configuration is taken from Aboanber and Nahla [2006], without the scale changes the authors did in that paper. The methodology used homogeneous mesh with $\Delta x = \Delta y = 4$, and reducing their value does not change significantly any results. The eigenvalue problem with parameters equivalent to $t = 0$ resulted in maximum eigenvalue $k = 0.998565$ and the code solved 900 unknowns each time step. The computational time for eigenvalue and time dependent problems were 1.2 s and 101.0 s, respectively.

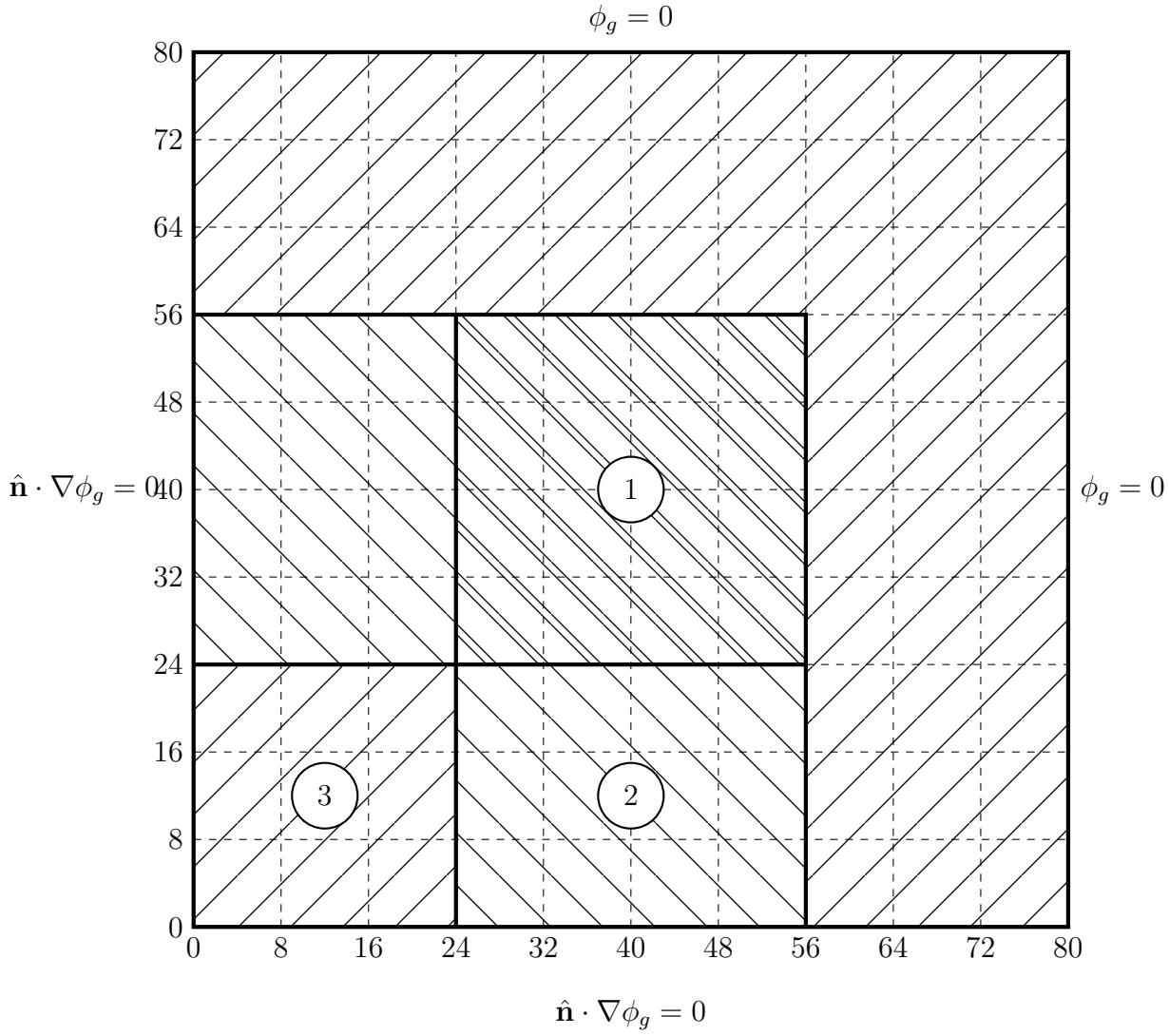
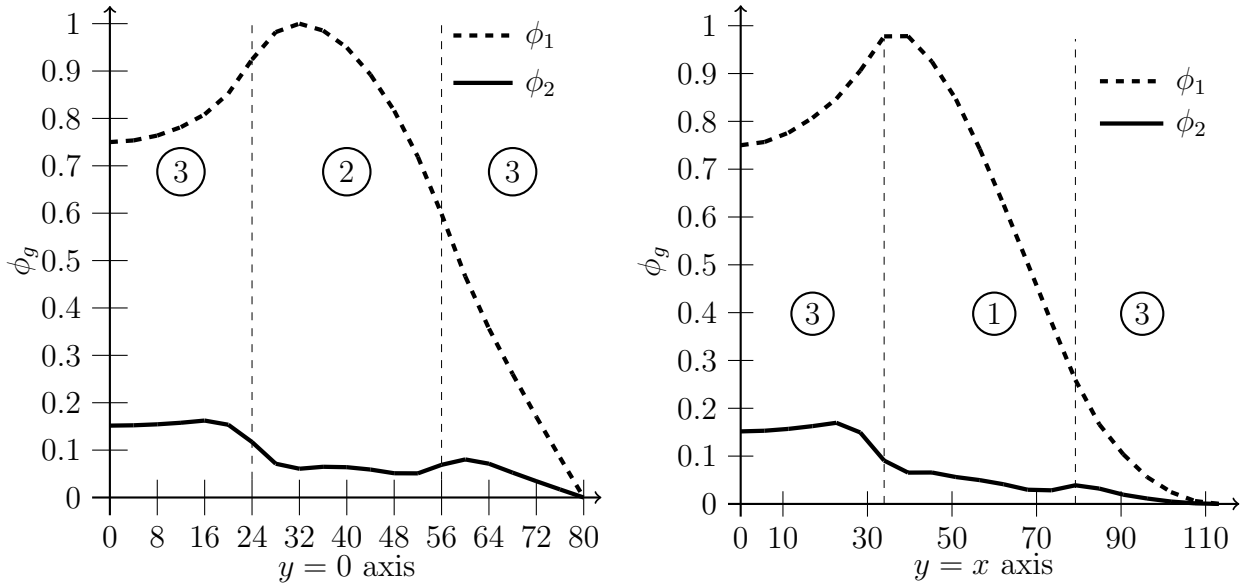


Figure 3.13: Geometry and boundary conditions for configuration 3.

Material	D_1	D_2	Σ_{c_1}	Σ_{c_2}	Σ_{f_1}	Σ_{f_2}	$\Sigma_{s_{12}}$	ν_g	λ	β
1	1.4	0.4	0.065	$*$	0.0035	0.1	0.01	2.1877	0.08	0.0075
2	1.4	0.4	0.065	0.05	0.0035	0.1	0.01	$v_1 = 1 \times 10^7$		
3	1.3	0.5	0.065	0.02	0.0015	0.03	0.01	$v_2 = 2 \times 10^5$		

$*$: parameter substituted by (**??**).

Table 3.4: Parameters for configuration 3.



(a) Fluxes profiles in $y = 0$.

(b) Fluxes profiles in $y = x$.

Figure 3.14: Fluxes graphics in some axes for configuration 3, eigenvalue problem.

Table 3.4 shows the parameters used in this case. Table 3.5 shows the values and positions of maximum ϕ_g and $\mathfrak{P}$ for $t = 0, 0.2, 0.4, 0.6, 0.8, 1$. Figure 3.16 shows the results for the eigenvalue problem ($t = 0$), and figure 3.14 shows the graphics for the neutron fluxes in the global axes $y = 0$ and $y = x$. Figures 3.17 to 3.21 show the results for $t = 0.2, 0.4, 0.6, 0.8, 1$ respectively. Figure 3.15 shows the behavior of the power level in time. The minimum $\mathfrak{P}$ happened at about $t = 0.01$ with value $\mathfrak{P} = 59.3792$. The maximum $\mathfrak{P}$ happened at $t = 1$ with value $\mathfrak{P} = 110.553$. Σ_{c_2} in material 1 is a time dependent function, given by

Note, in table 3.4 it is marked with the symbol $*$.

	ϕ_1		ϕ_2		
t	value	position	value	position	$\mathfrak{P}$
0	1	$(32,0),(0,32)$	0.169643	$(16,16)$	64.5200
0.2	1.57297	$(32,16),(16,32)$	0.258811	$(16,16)$	102.780
0.4	1.64054	$(32,16),(16,32)$	0.270066	$(16,16)$	107.214
0.6	1.65740	$(32,16),(16,32)$	0.272839	$(16,16)$	108.316
0.8	1.67443	$(32,16),(16,32)$	0.275641	$(16,16)$	109.429
1	1.69161	$(32,16),(16,32)$	0.278469	$(16,16)$	110.553

Table 3.5: Maximum values and positions (x, y) of fluxes and $\mathfrak{P}$ for several t, configuration 3.

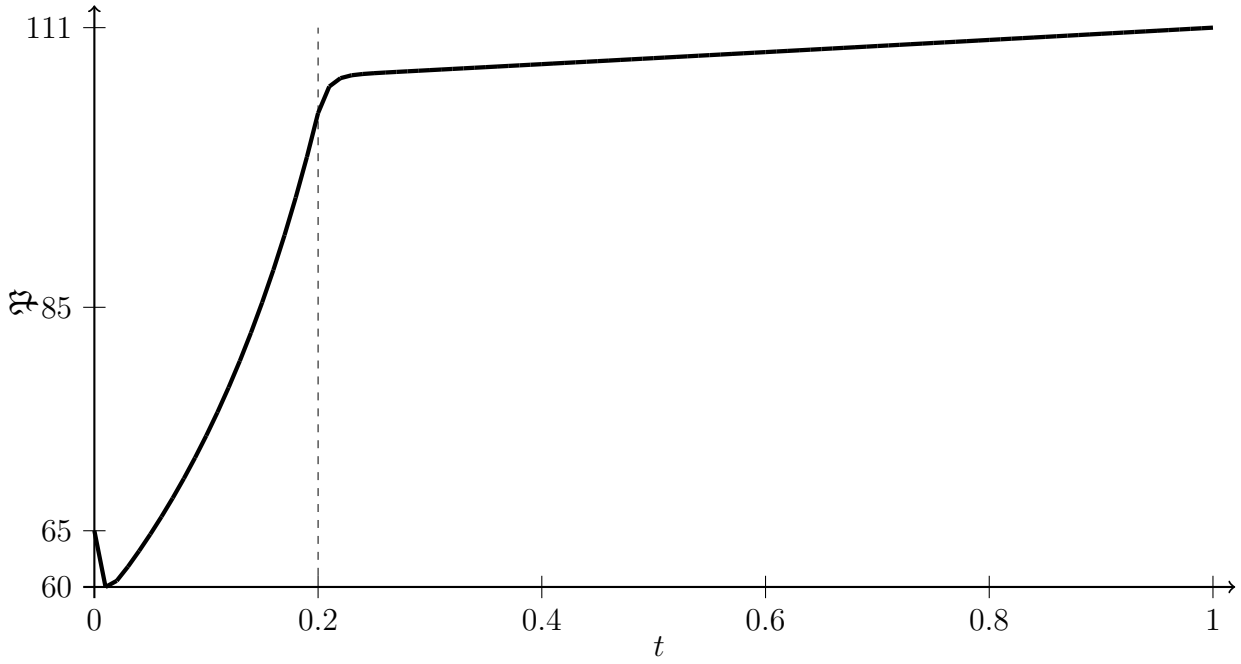


Figure 3.15: $\mathfrak{P}$ in time for configuration 3.

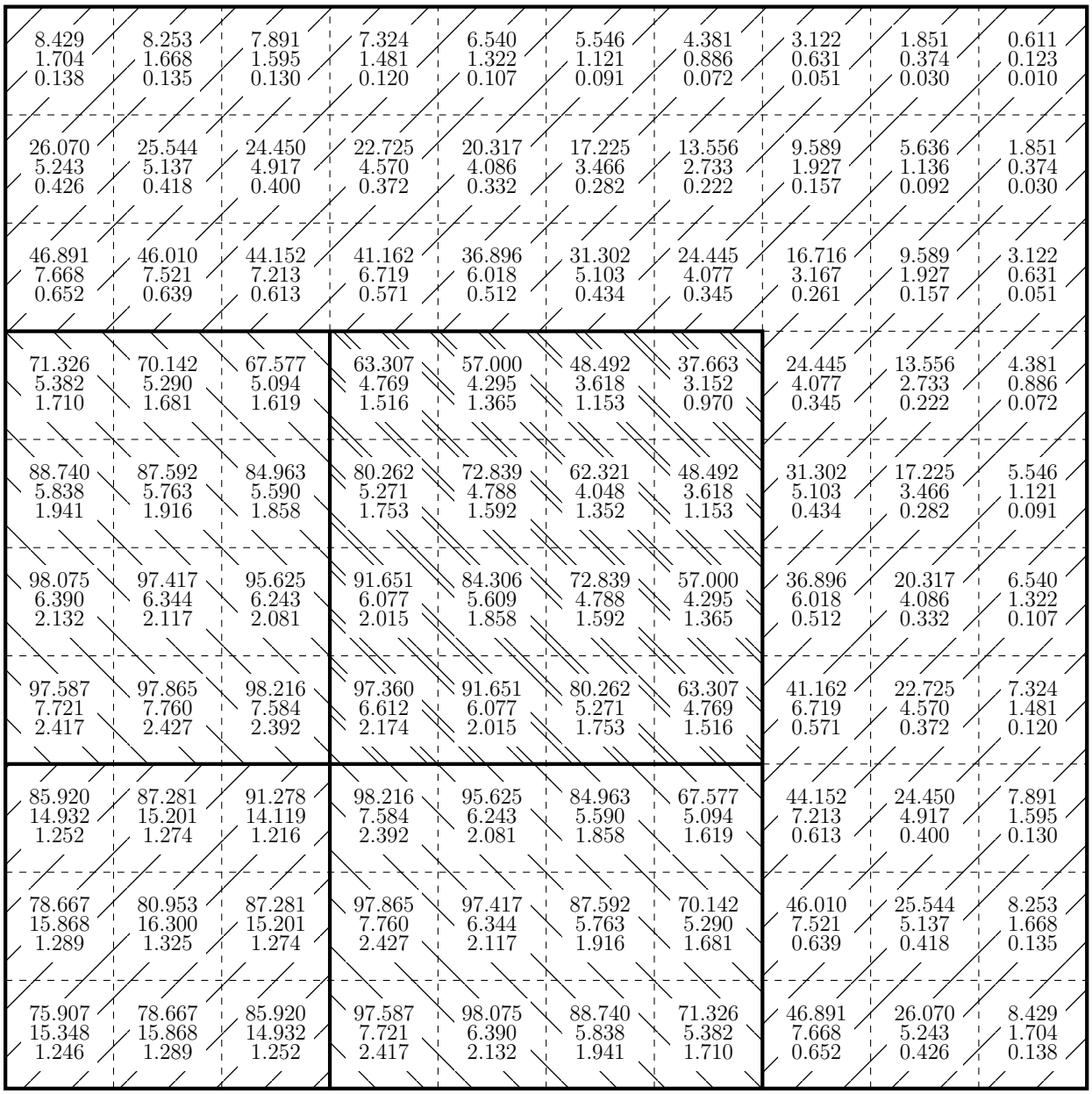


Figure 3.16: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 3, eigenvalue problem.

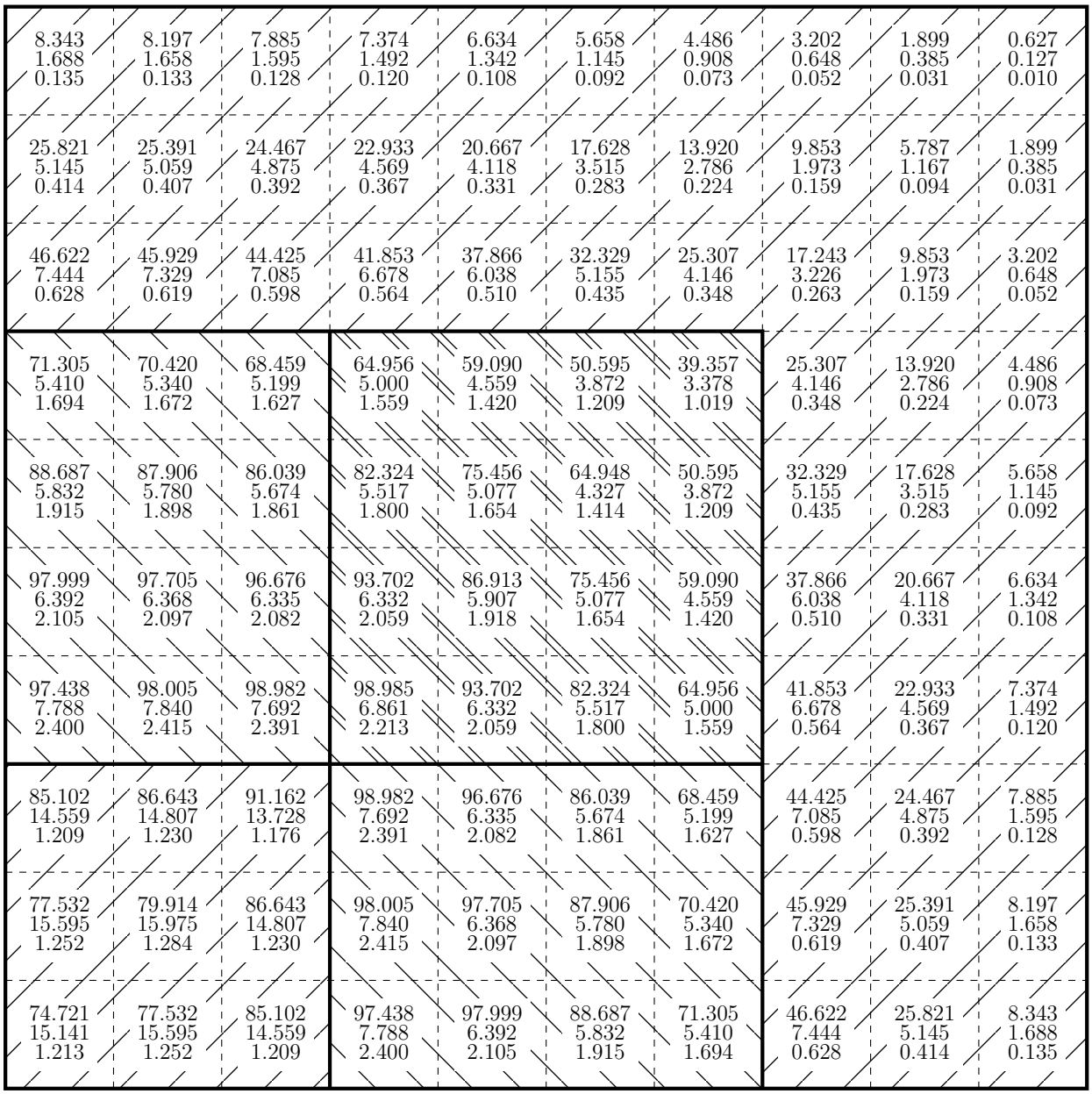


Figure 3.17: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 3, $t = 0.2$.

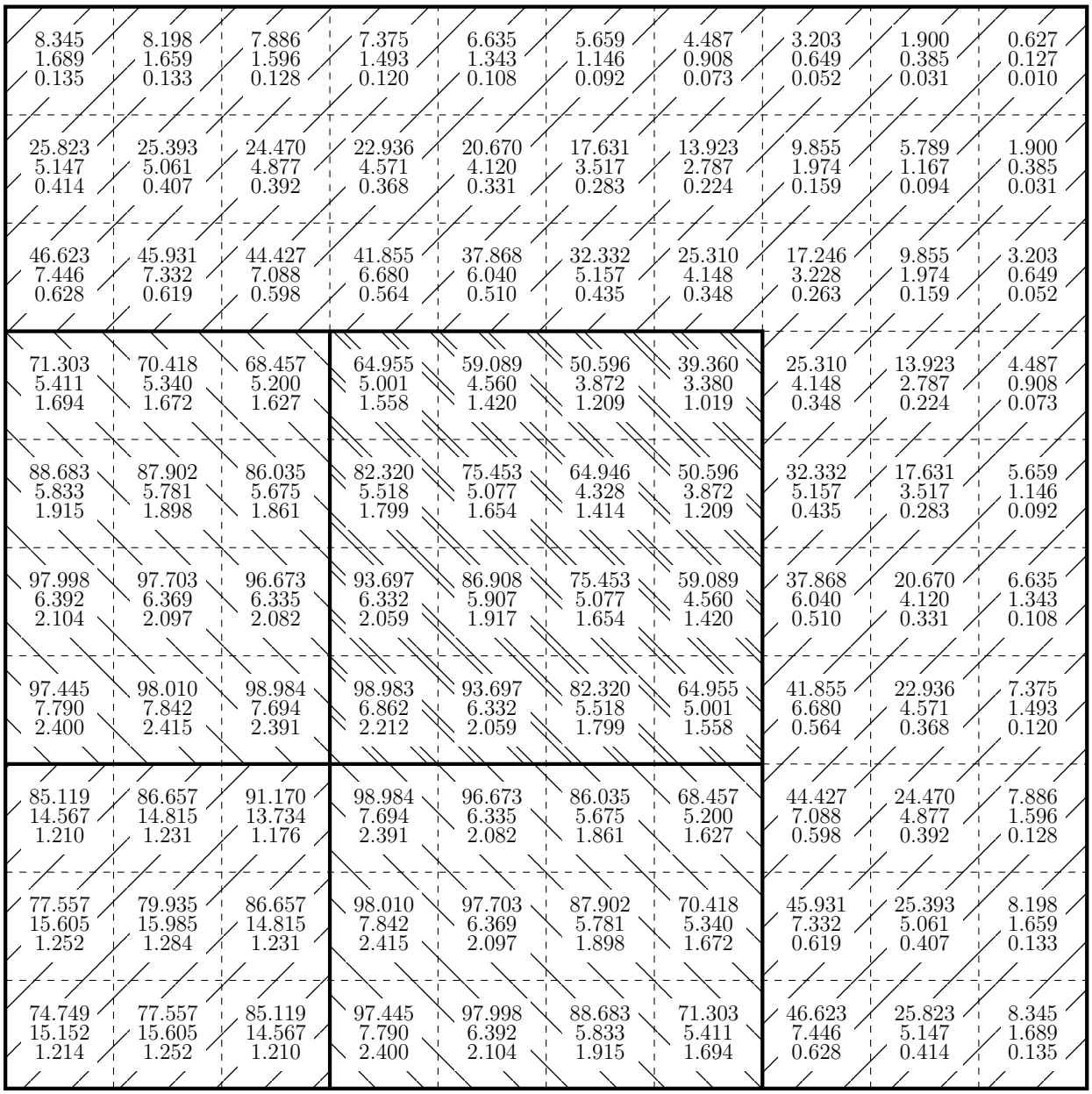


Figure 3.18: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 3, $t = 0.4$.

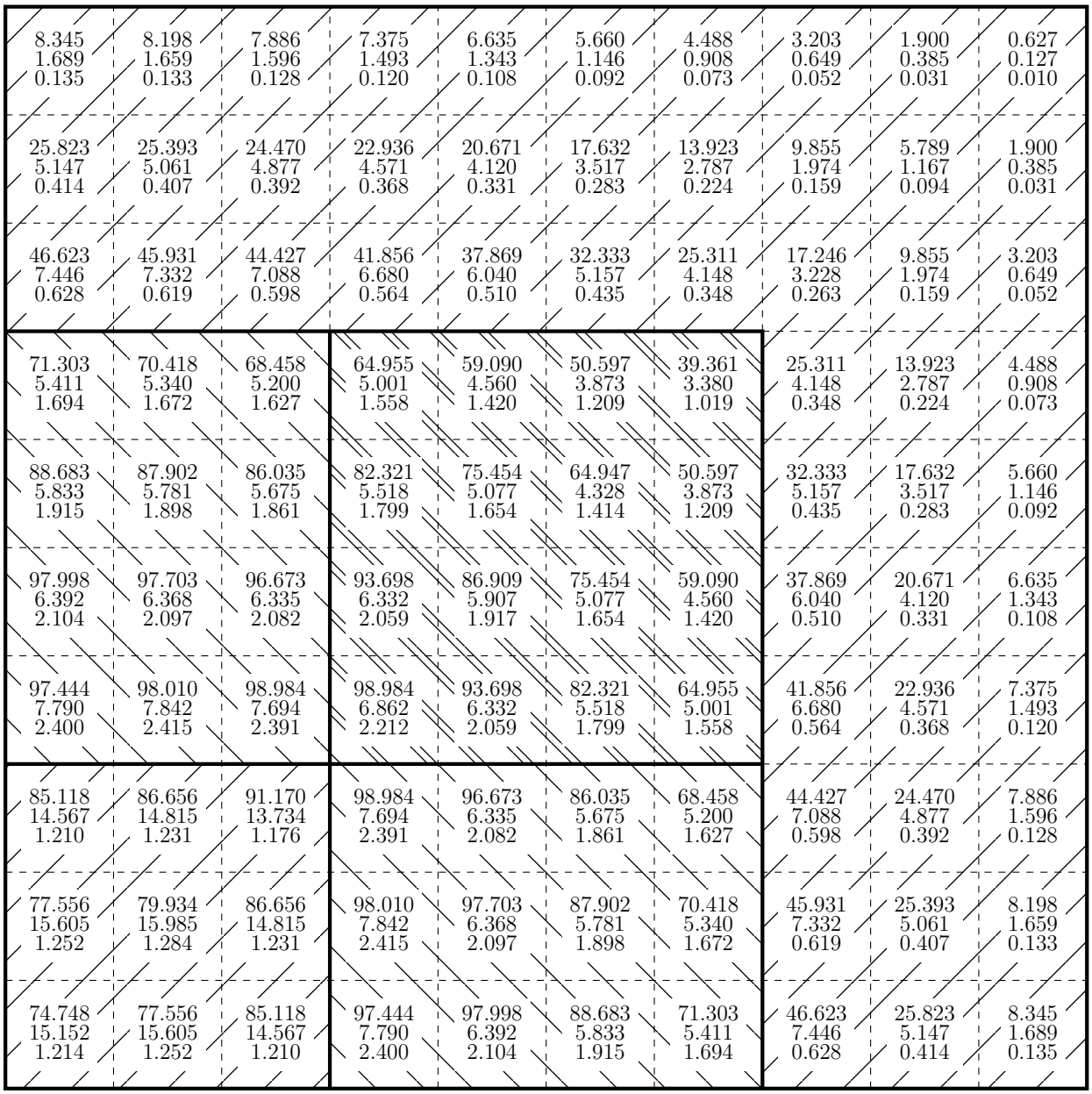


Figure 3.19: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 3, $t = 0.6$.

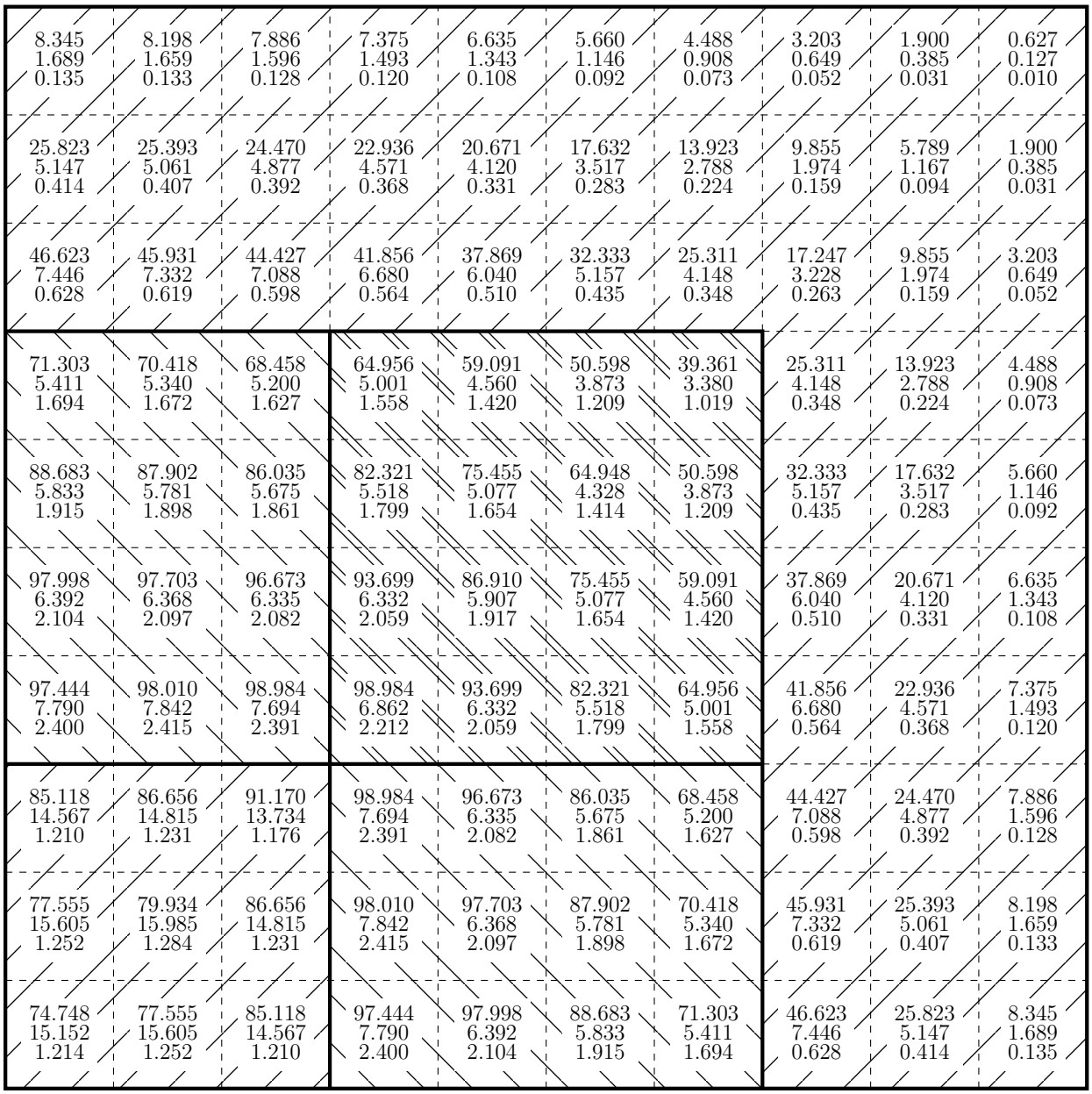


Figure 3.20: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 3, $t = 0.8$.

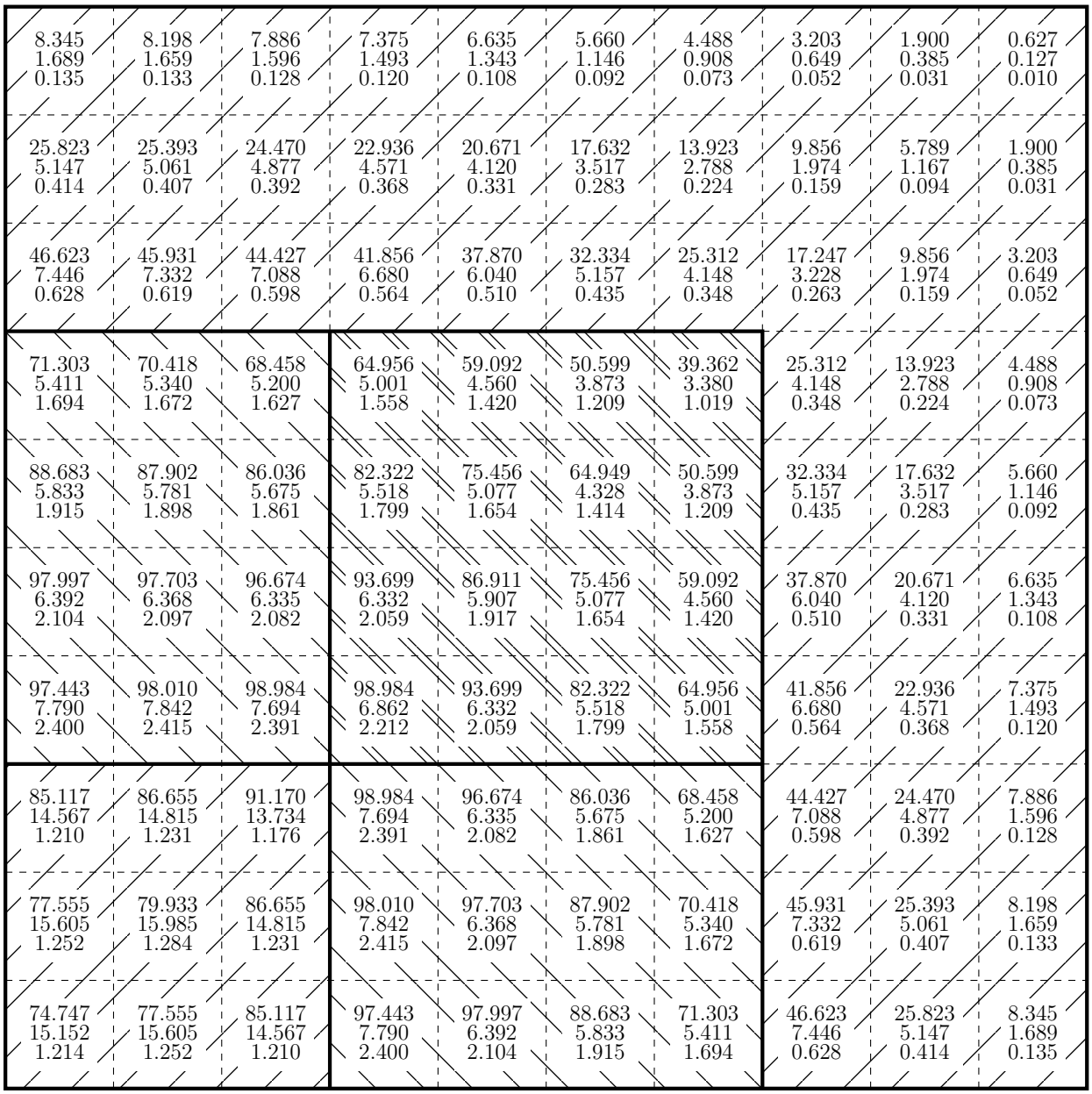


Figure 3.21: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 3, $t = 1$.

3.4 Configuration 4

This configuration is a known benchmark for two–dimensional time dependent neutron space kinetics codes, however the author did not find the proper reference to this case and limited himself to show the obtained results without any comparing. This case is modeled until $T = 60$ and it had to be divided in 6 parts of 10 s each because the computer did not had enough memory to stock the transients vector. The methodology used homogeneous mesh with $\Delta x = \Delta y = 5$, and reducing their value does not change significantly any results.

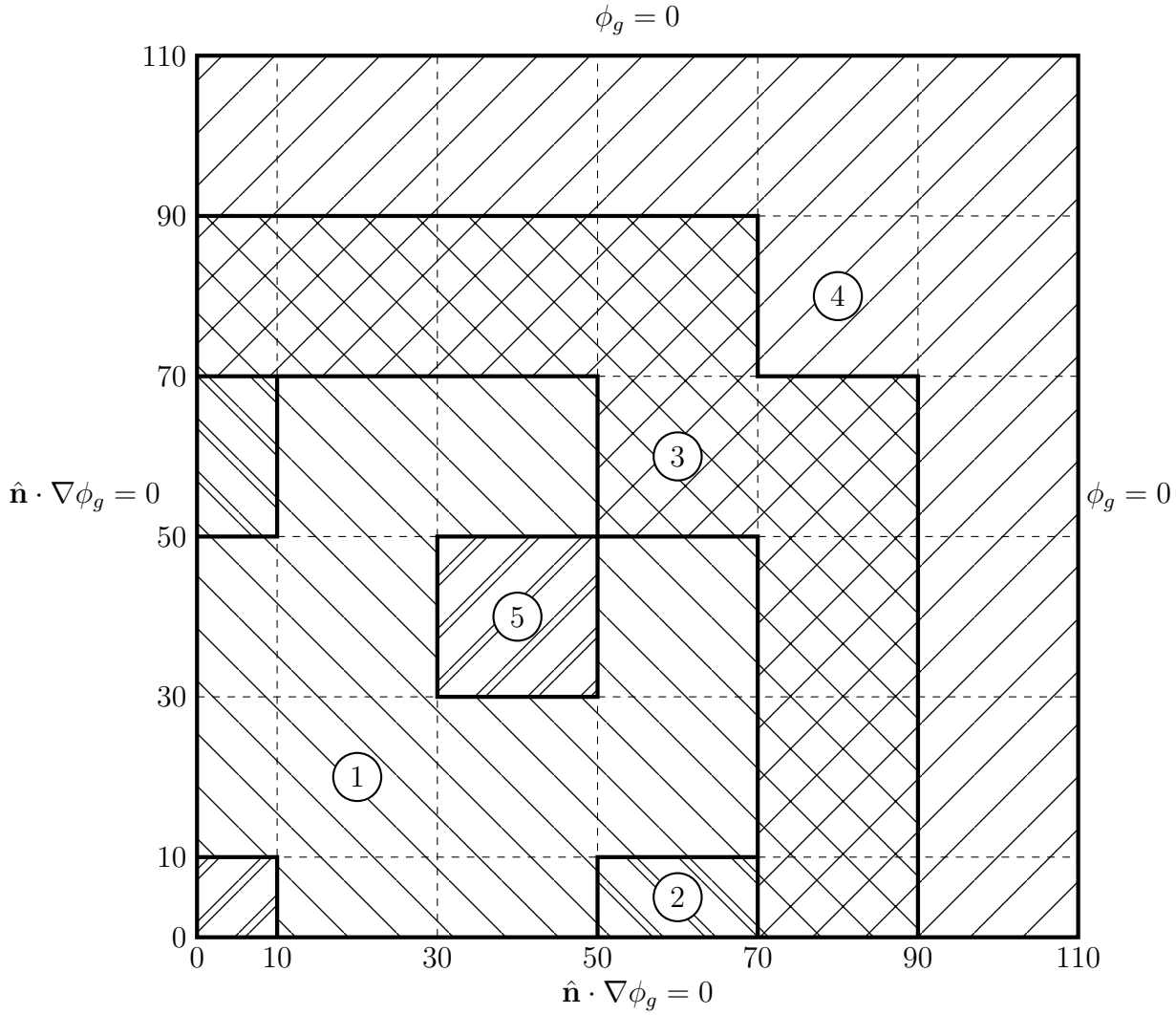
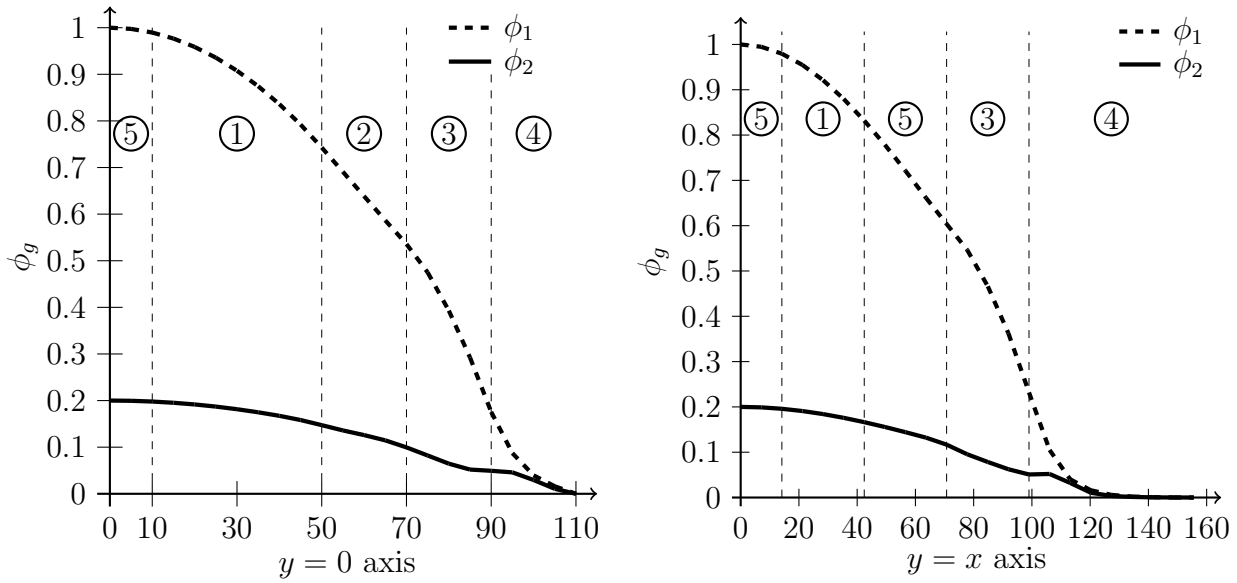


Figure 3.22: Geometry and boundary conditions for configuration 4.

	Material				
Parameter	1	2	3	4	5
D_1	1.423913	1.423913	1.425611	1.634227	1.423913
D_2	0.356306	0.356306	0.350574	0.264002	0.356306
Σ_{c_1}	0.0078109836	$*$	0.0079913164	0.002660573	$*$
Σ_{c_2}	0.04256905	$*$	0.04413618	0.04936351	$*$
Σ_{f_1}	0.0025910764	0.0025910764	0.0030013136	0	0.0025910764
Σ_{f_2}	0.04509312	0.04509312	0.05512016	0	0.04509312
$\Sigma_{s_{12}}$	0.0175555	0.0175555	0.01717763	0.02759693	0.0175555
$\nu_g = 2.4564332$, $v_1 = 1.25 \times 10^7$, $v_2 = 2.5 \times 10^5$, $\lambda = 0.078408534$, $\beta = 0.0064995$					

$*$: parameters substituted by (3.10) and (3.11).

Table 3.6: Parameters for configuration 4.



(a) Fluxes profiles in $y = 0$.

(b) Fluxes profiles in $y = x$.

Figure 3.23: Fluxes graphics in some axes for configuration 4, eigenvalue problem.

The eigenvalue problem with parameters equivalent to $t = 0$ resulted in maximum eigenvalue $k = 0.999042$ and the code solved 1089 unknowns each time step. The computational time for eigenvalue problem was 2.7 s.

Table 3.6 shows the parameters used in this case. Table 3.7 shows the values and positions of maximum ϕ_g and $\mathfrak{P}$ for $t = 0, 7.5, 10, 20, \frac{25}{1.1}, 30, 40, 47.5, 50, 60$ and the 10 s evolution computational times for $t = 10, 20, 30, 40, 50, 60$. Figure 3.25 shows the results for the eigenvalue problem ($t = 0$), and figure 3.23 shows the graphics for the neutron fluxes in the global axes $y = 0$ and $y = x$. Figures 3.26 to 3.34 show the results for

$t = 0, 7.5, 10, 20, \frac{25}{1.1}, 30, 40, 47.5, 50, 60$ respectively. Figure 3.24 shows the behavior of the power level in time. The minimum $\mathfrak{P}$ happened at $t = 60$ with value $\mathfrak{P} = 28.0782$. The maximum $\mathfrak{P}$ happened at $t = 0$ with value $\mathfrak{P} = 127.990$.

Σ_{a_g} in material 2 are time dependent functions, given by

$$\Sigma_{a_1} = \begin{cases} 1.062210 \times 10^{-2} - 2.2 \times 10^{-4} \dfrac{1.1t}{25} , & 0 \leq t < \dfrac{25}{1.1} \\ 1.040210 \times 10^{-2} , & \dfrac{25}{1.1} \leq t \leq 60 \end{cases} , \tag{3.10a}$$

$$\Sigma_{a_2} = \begin{cases} 8.91822 \times 10^{-2} - 1.52 \times 10^{-3} \dfrac{1.1t}{25} , & 0 \leq t < \dfrac{25}{1.1} \\ 8.76622 \times 10^{-2} , & \dfrac{25}{1.1} \leq t \leq 60 \end{cases} . \tag{3.10b}$$

Σ_{a_g} in material 5 are time dependent functions, given by

$$\Sigma_{a_1} = \begin{cases} 1.040206 \times 10^{-2} , & 0 \leq t < 7.5 \\ 1.040206 \times 10^{-2} + 3.85 \times 10^{-4} \dfrac{t - 7.5}{40} , & 7.5 \leq t < 47.5 \\ 1.078706 \times 10^{-2} , & 47.5 \leq t \leq 60 \end{cases} , \tag{3.11a}$$

$$\Sigma_{a_2} = \begin{cases} 8.76217 \times 10^{-2} , & 0 \leq t < 7.5 \\ 8.76217 \times 10^{-2} + 2.66 \times 10^{-3} \dfrac{t - 7.5}{40} , & 7.5 \leq t < 47.5 \\ 9.032217 \times 10^{-2} , & 47.5 \leq t \leq 60 \end{cases} . \tag{3.11b}$$

Note, in table 3.6 in materials 2 and 5 their Σ_{c_g} are marked with the symbol $*$, as their Σ_{a_g} are determined by these expressions instead.

	ϕ_1		ϕ_2			
t	value	position	value	position	$\mathfrak{P}$	c.time
0	1	$(0,0)$	0.200038	$(0,0)$	127.990	
7.5	0.875662	$(0,0)$	0.175145	$(0,0)$	113.677	
10.0	0.847555	$(0,0)$	0.169194	$(0,0)$	110.541	613.5
20.0	0.730632	$(0,0)$	0.144726	$(0,0)$	97.0949	814.3
25/1.1	0.698505	$(0,0)$	0.138349	$(0,0)$	93.2013	
30.0	0.535145	$(0,0)$	0.106159	$(15,0),(0,15)$	71.8141	777.6
40.0	0.385653	$(10,0),(0,10)$	0.0768145	$(15,0),(0,15)$	52.2755	959.2
47.5	0.294537	$(15,0),(0,15)$	0.0587293	$(15,0),(0,15)$	40.1463	
50.0	0.273473	$(15,0),(0,15)$	0.0545262	$(15,0),(0,15)$	37.2833	911.9
60.0	0.205895	$(15,0),(0,15)$	0.0410523	$(15,0),(0,15)$	28.0782	855.2

Table 3.7: Maximum values and positions (x, y) of fluxes and $\mathfrak{P}$ for several t, configuration 4.

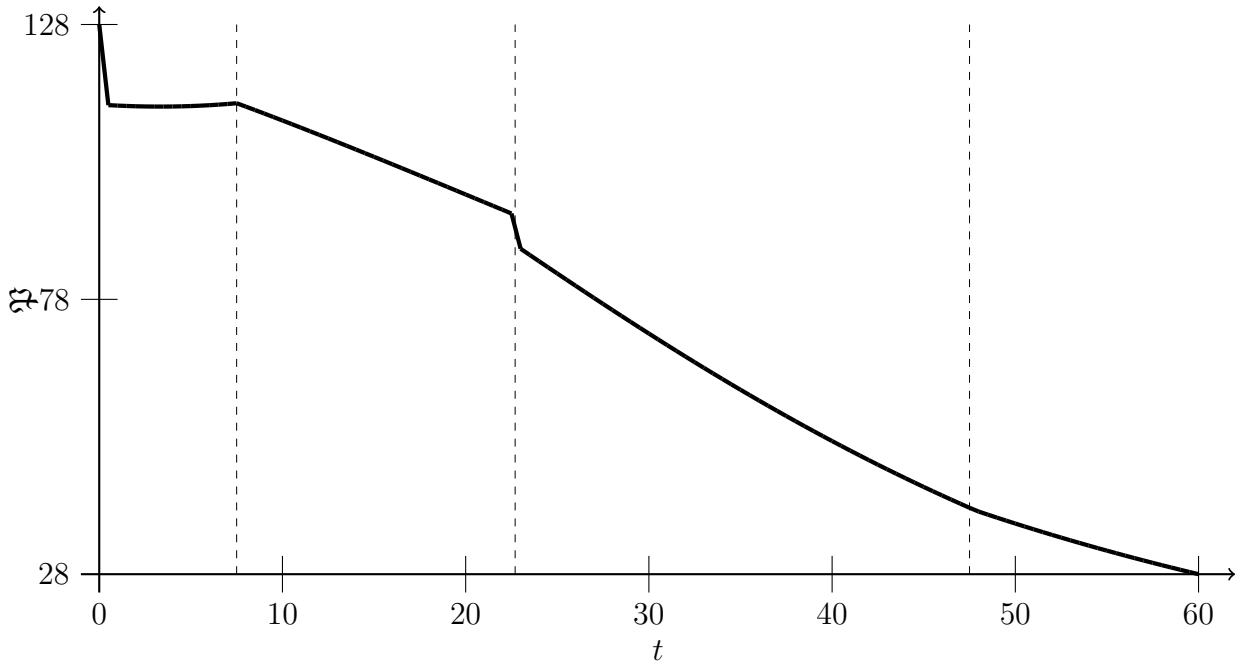


Figure 3.24: $\mathfrak{P}$ in time for configuration 4.

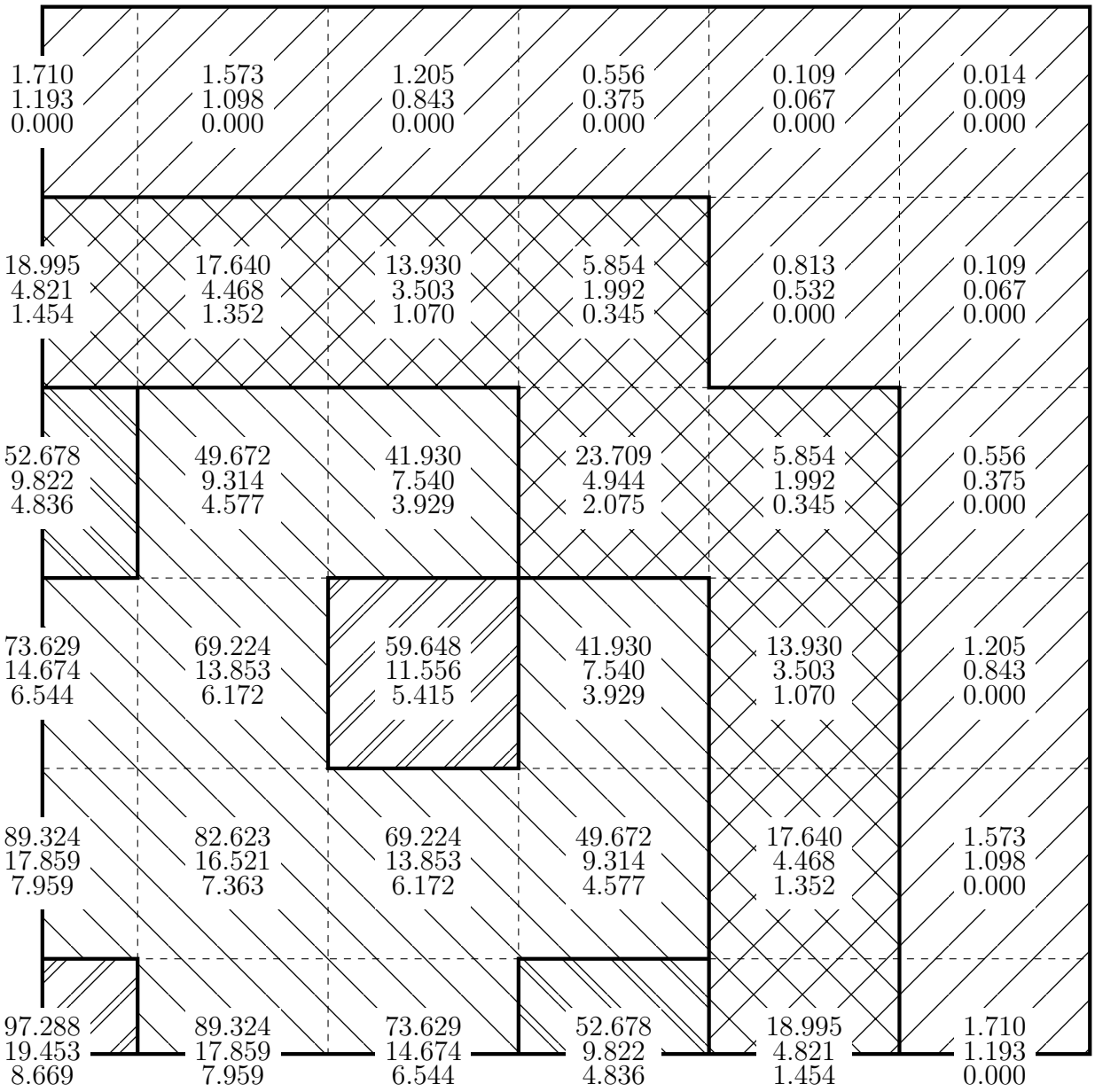


Figure 3.25: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 0$.

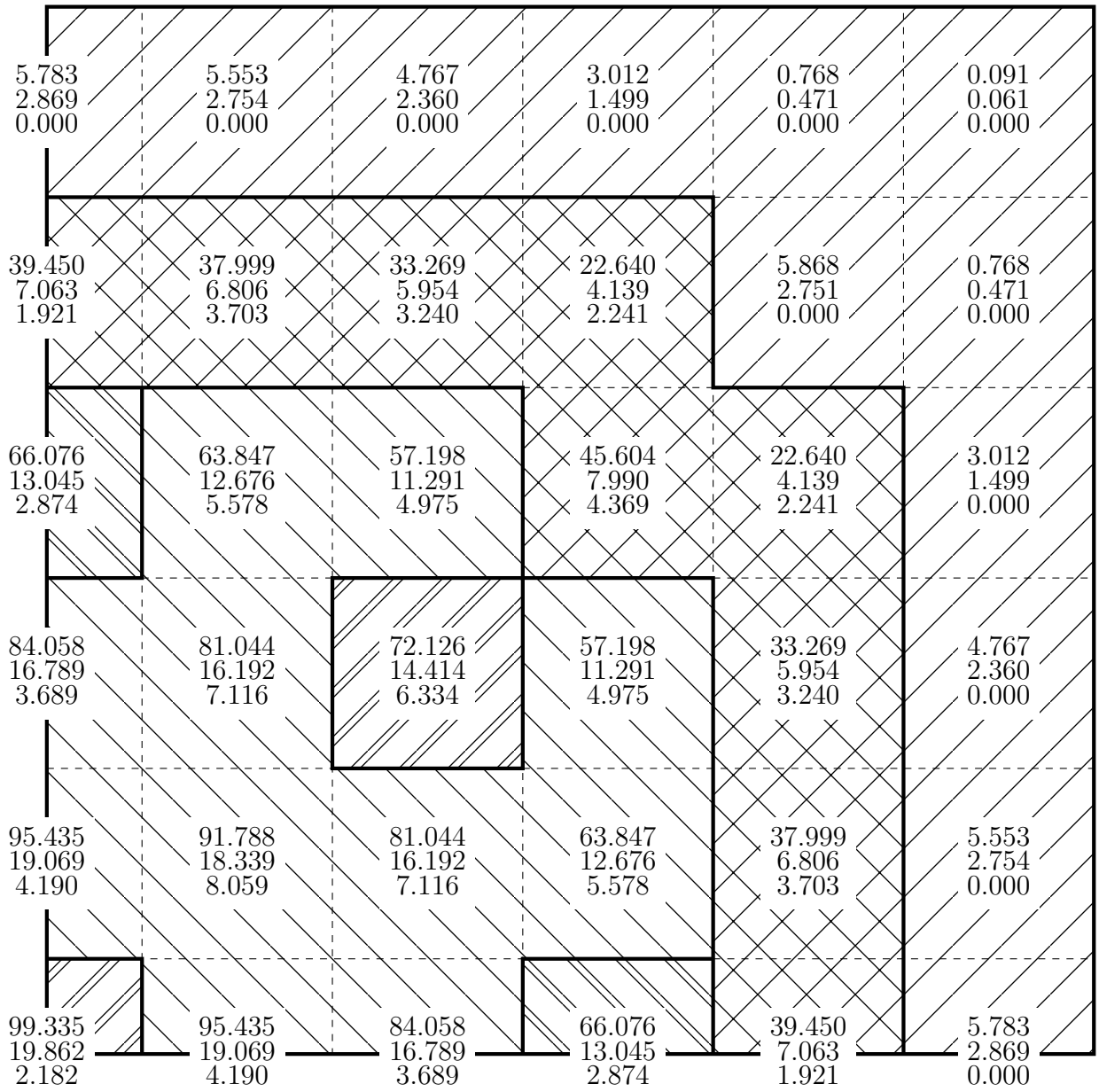


Figure 3.26: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 7.5$.

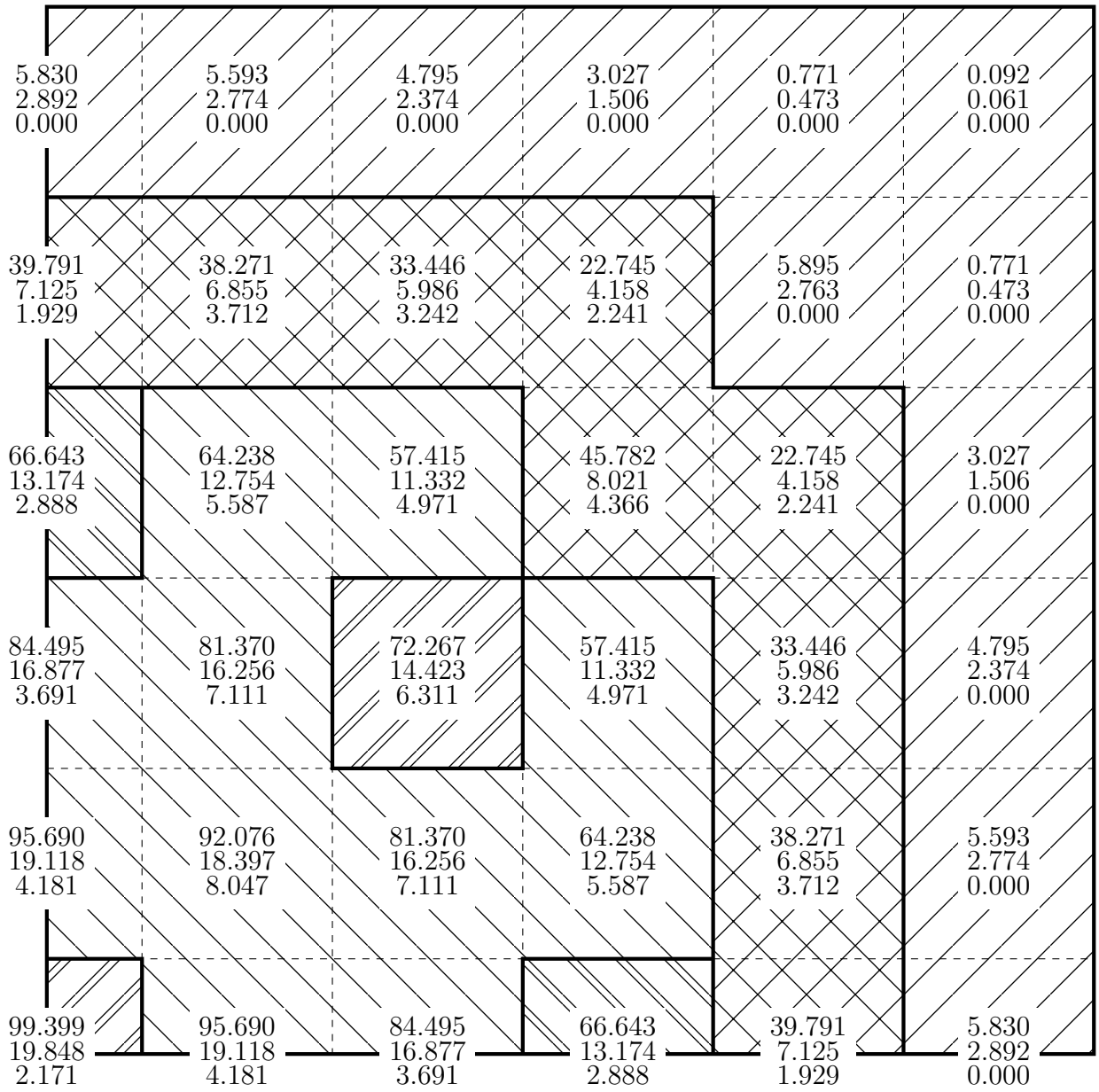


Figure 3.27: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 10$.

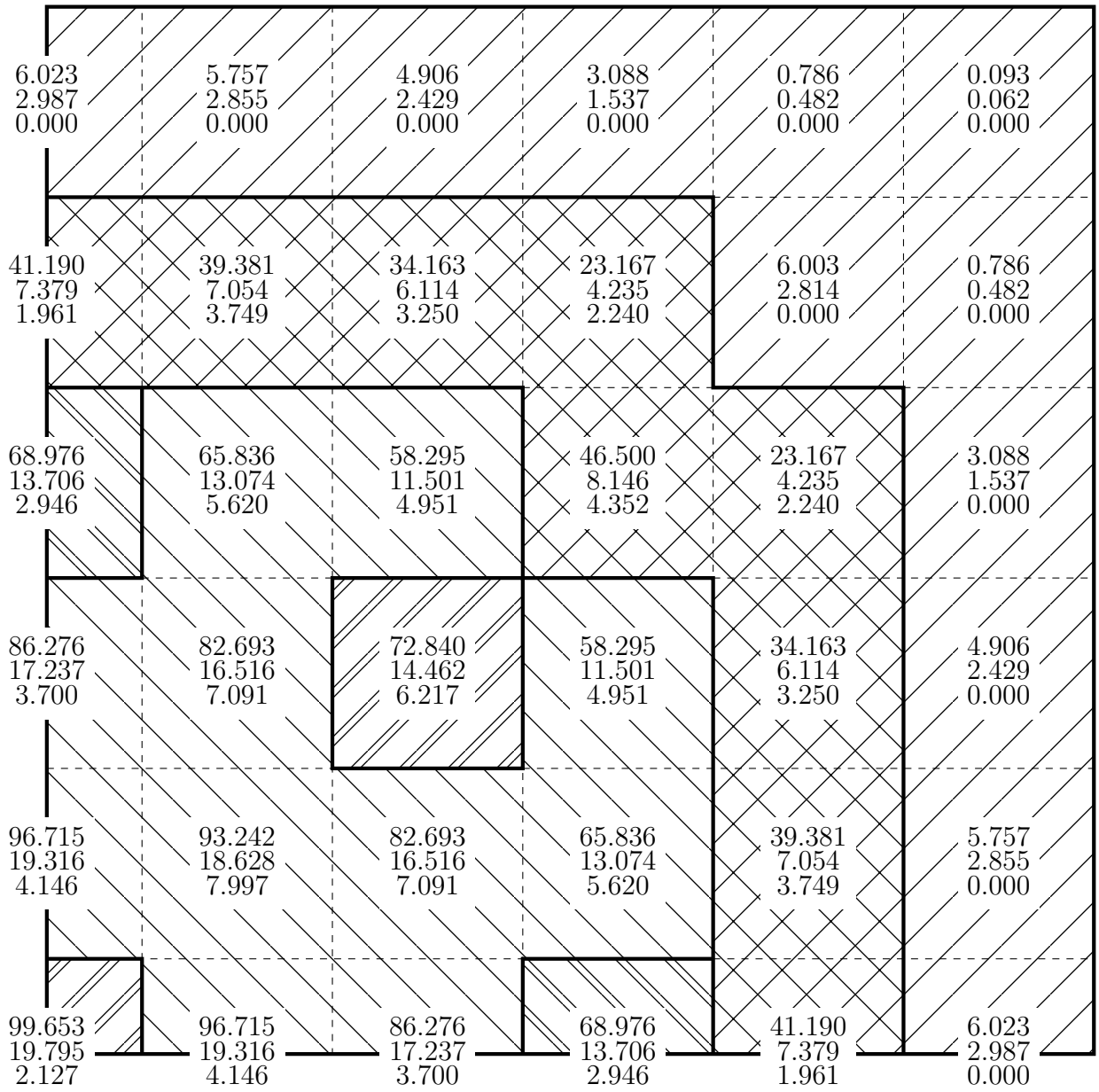


Figure 3.28: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 20$.

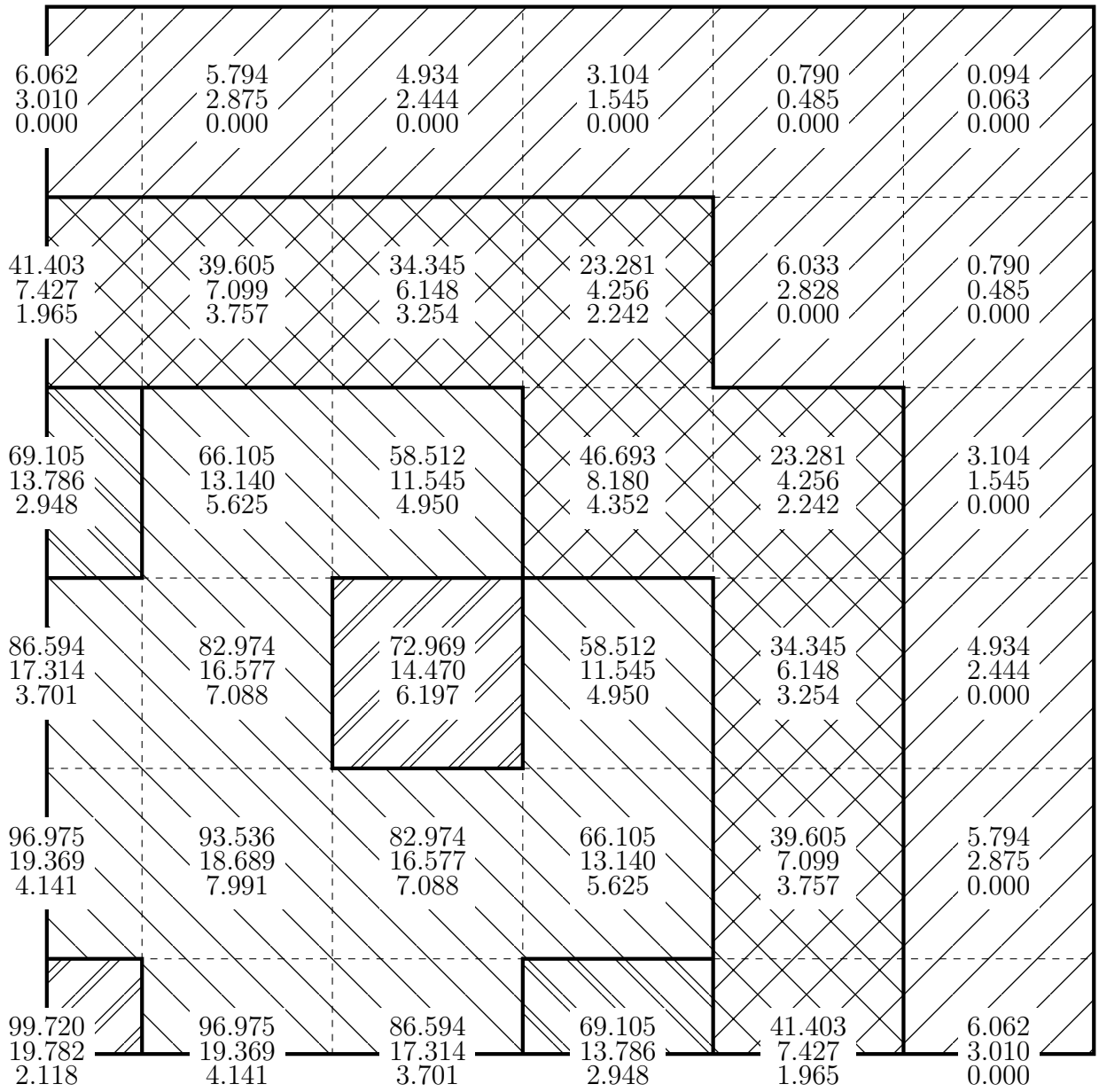


Figure 3.29: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = \frac{25}{1.1}$.

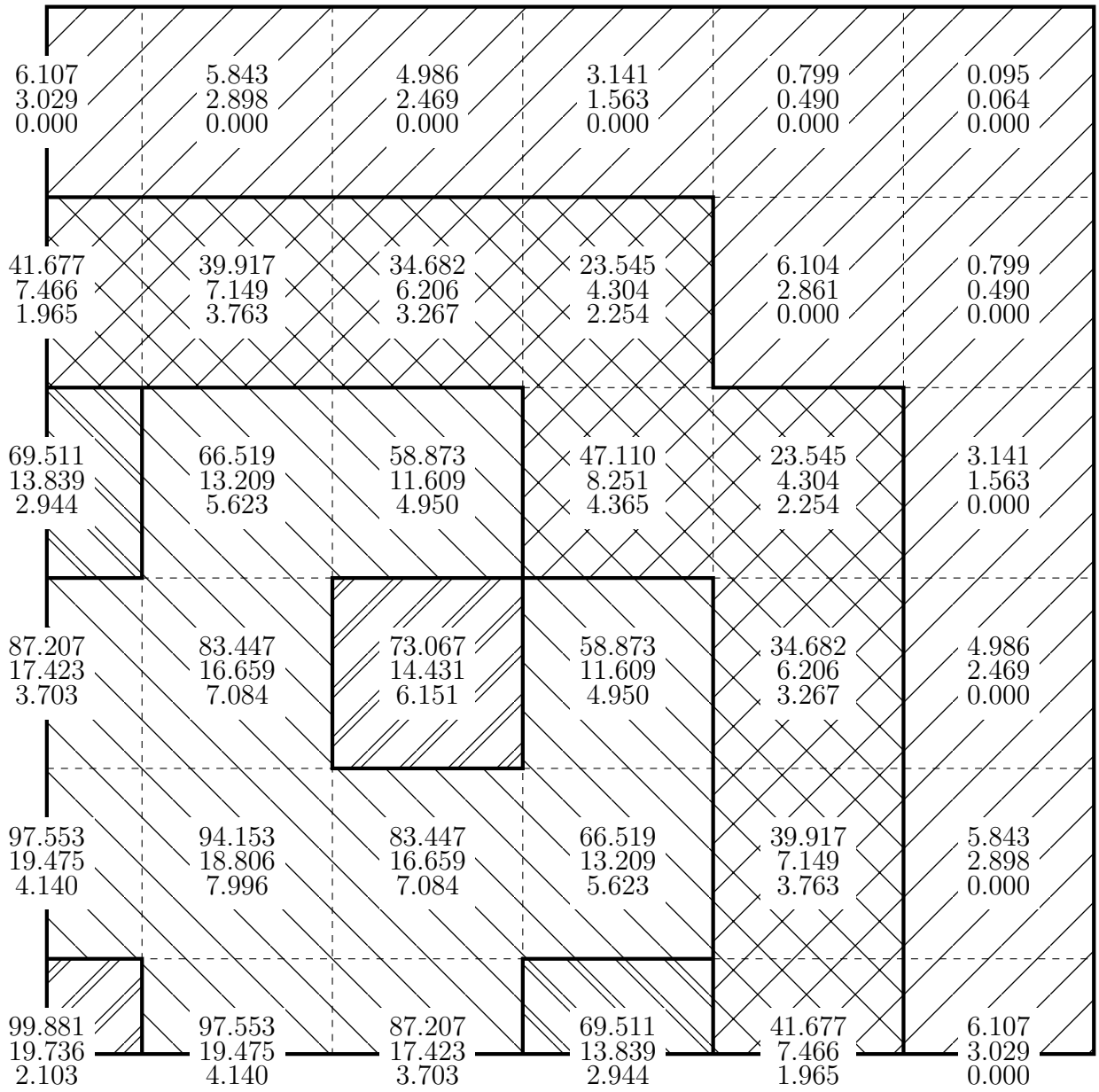


Figure 3.30: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 30$.

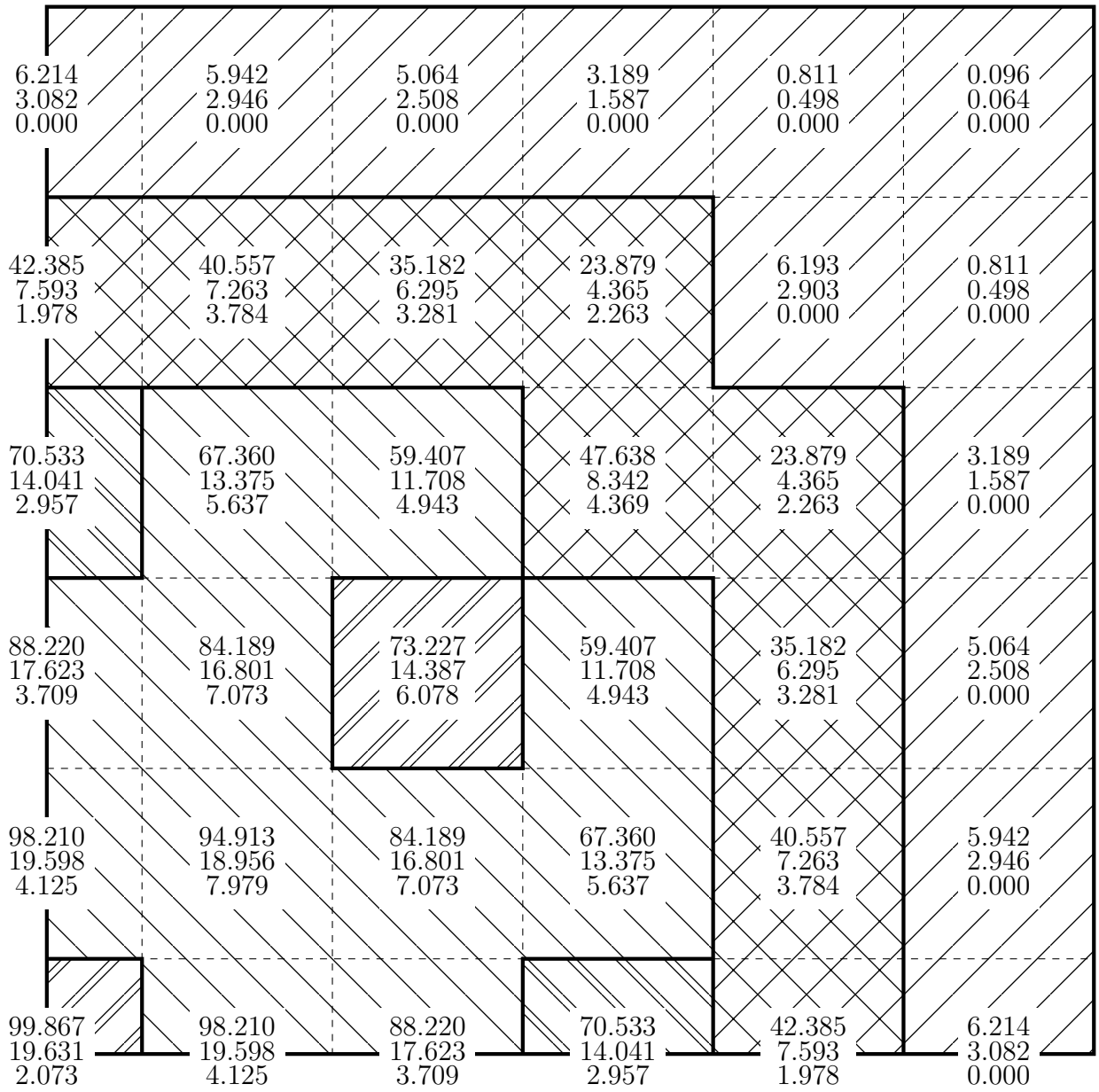


Figure 3.31: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 40$.

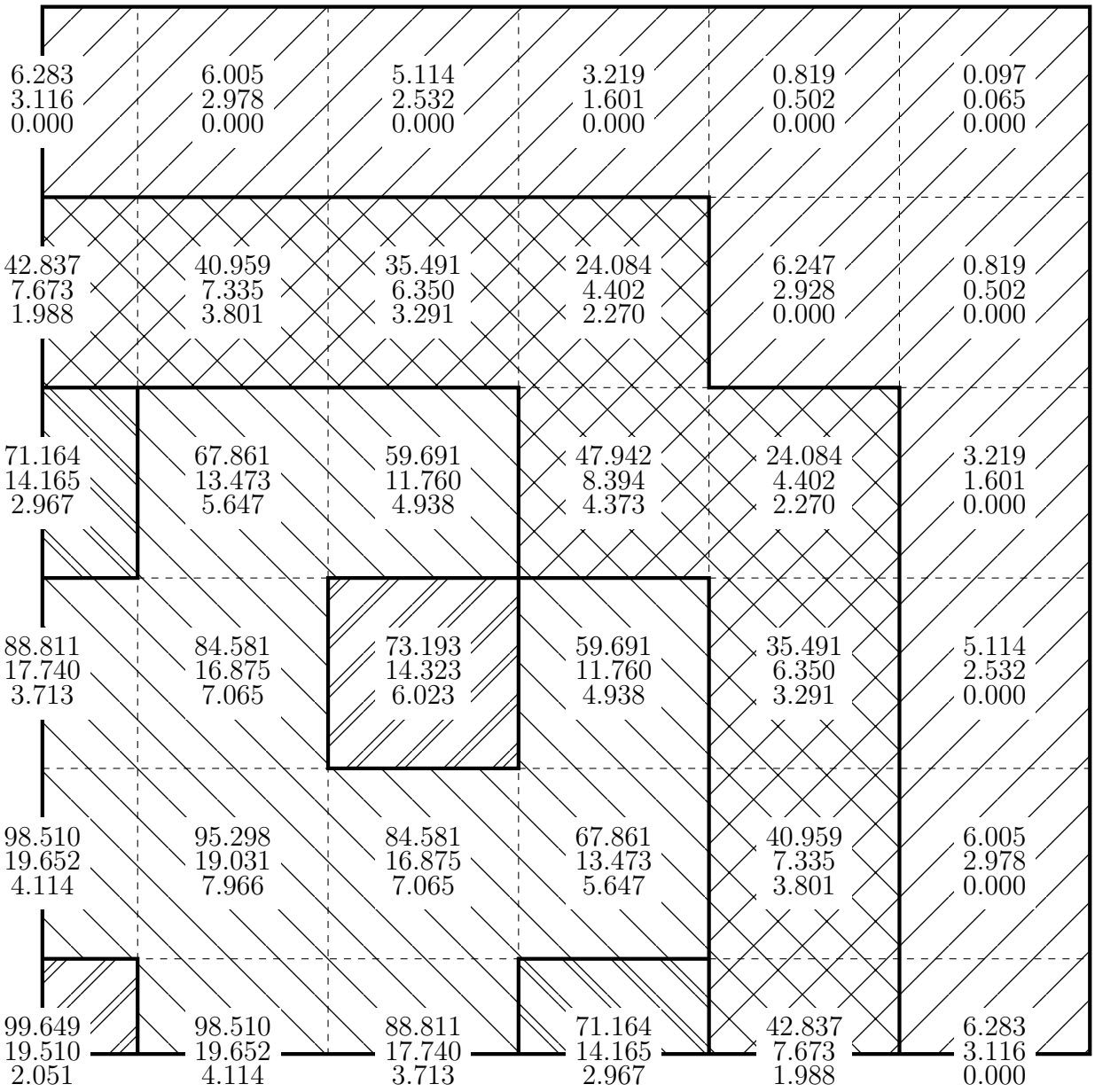


Figure 3.32: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 47.5$.

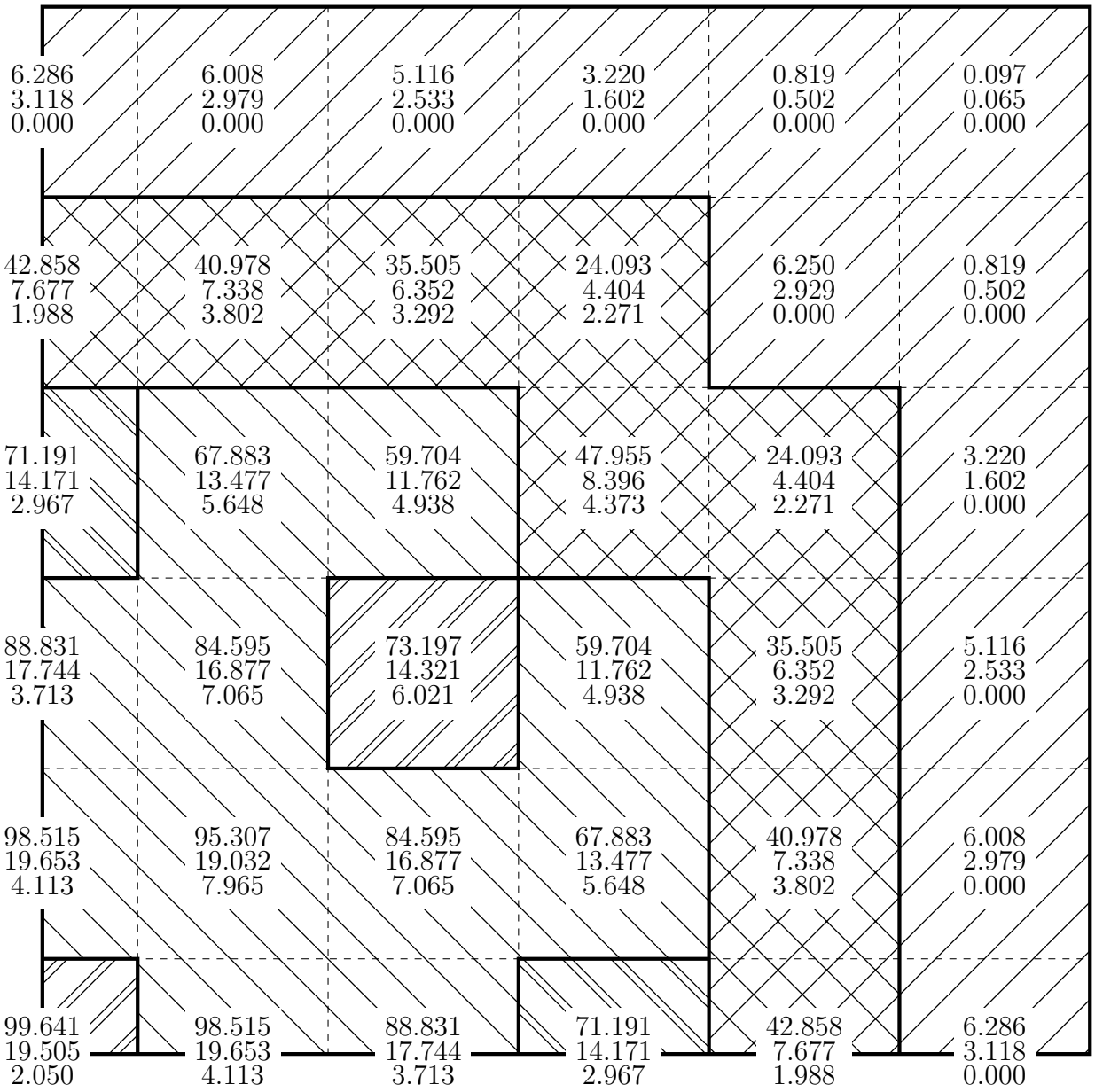


Figure 3.33: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 50$.

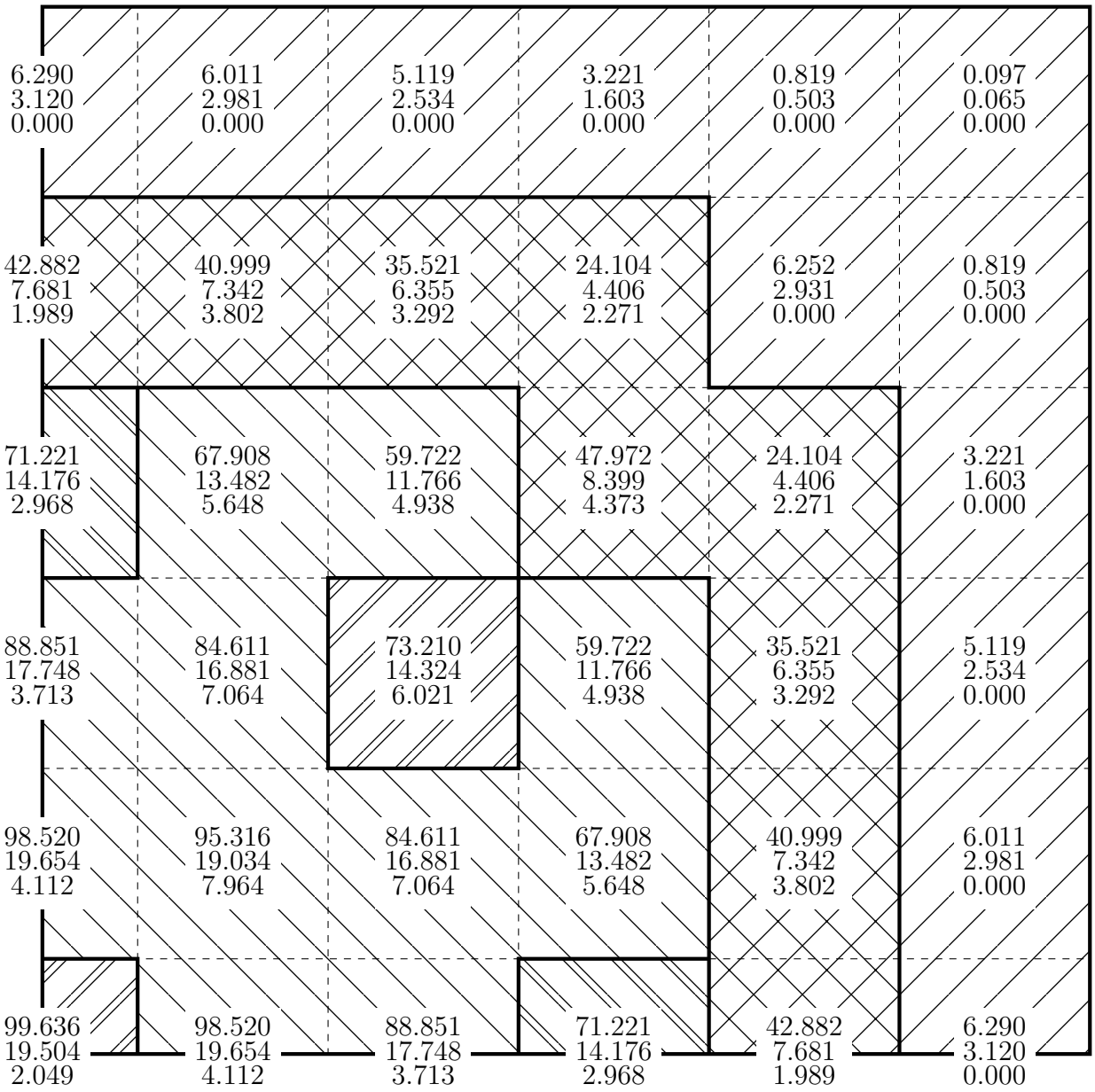


Figure 3.34: Local average normalized neutron scalar fluxes and $\mathfrak{P}^{[j]}$ for confituration 4, $t = 60$.

4 CONCLUSION

In the present thesis, a new methodology to solve neutron diffusion problems is presented. PEM keeps the simplicity of a numerical method like finite differences but it also keeps an analytical character at the complete series solution, which provides, locally, the analytical solution. The methodology error is more applicable to computer codes, as it does not require the implementation of further terms on the polynomial. Also, it is noteworthy that the methodology error is determined by the differential equation alone, bearing in mind that the boundary conditions are locally satisfied. In other words, this methodology error is as applicable as the proper approximations required in the boundary conditions and the differential equation independent terms. The author found that using a higher order polynomial could cause the code to easily reach an under or overflow due to the difference of scales between the coefficients. Despite this methodology is not particular to this type of problem, it is adapted to the nuclear reactor particularities, like the presence of many energy groups, precursors groups and rectangular geometry. The iterative scheme that links the local solutions does have a proper convergence indicative given by the referenced papers at chapter 2, section 2. A proof of linear independence helps to assure the uniqueness of the solution and the numerical operations to accelerate the convergence avoids any ambiguities that could happen by the application of PEM. Also, in the time variable, the known modified Euler method is used with a slight modification. The author is aware that there was no outputs comparing whatsoever in the time dependent cases at the results chapter, however this methodology has already been proven to be stable. It is known from the cited literature that it has proper global error estimates. Also from the literature the expected behavior of Δt is affirmed: as it goes to infinity, the discrete time problem returns the steady state problem, and as it goes to zero the solution of each time step goes to the analytical solution. As the problem is linear, an applicable algorithm for $0 < \Delta t < \infty$ is expected. The author chose this methodology for the time variable because of its stability and because the application of PEM in the time variable (using a linear basis instead of second order polynomial) provides the same algorithm as the Euler modified method. The methodology has been successfully used to actual nuclear reactor benchmark problems used in the literature, however without

any comparison of results except the configuration 1, which is the most usual for matter of comparison in recent papers. As expected, the obtained results do not differ significantly from the reference.

This document describes the dependence of powers of $x^m y^n$ coefficients, which explicits how much the terms φ_{00} contribute on the determination of other φ–s. These contributions are determined by the local (fictitious) boundary conditions only, so they are independent of the differential equation. In fact, these relations may be the topic for a future investigation of this methodology: as some coefficients represent the Laplacian of the neutron scalar flux, how exactly the boundary conditions influence the final solution itself by interfering with the independent term by the diffusion equation? Also, the methodology error uses the diffusion equation to its determination. In the present calculations, the solution was obtained first, and then the error was determined, as seen on the previous pages. However, a question rises from this fact: could the error estimate be used as an input to obtain a more precise solution? The answer to this question is being developed. Another issue to be considered in a future time is the parameters dependence on the unknowns themselves. This kind of problem is usual in reactor dynamics, that considers the thermo–physical changes on the parameters, making the diffusion problem turn into a non–linear one, which is closer to real problems.

BIBLIOGRAPHIC REFERENCES

Aboanber, A. E. and Hamada, Y. M. Generalized Runge–Kutta method for two– and three–dimensional space–time diffusion equations with a variable time step, **Annals of Nuclear Energy**, vol. 35, pp. 1024–1040, 2008.

Aboanber, A. E. and Nahla, A. A. Solution of two–dimensional space–time multi-group reactor kinetics equations by generalized Padé and cut–product approximations, **Annals of Nuclear Energy**, vol. 33, pp. 209–222, 2006.

Alcouffe, E. R. and Albrecht, R. W. A Generalization of the Finite Difference Approximation Method with an Application to Space–Time Nuclear Reactor Kinetics, **Nuclear Science and Engineering**, vol. 39, pp. 1–13, 1970.

Argonne Code Center. **Benchmark Problem Book, Report ANL-7416 (Suppl. 2)**. Argonne National Laboratorry, USA, 1977.

Barros, R. C., Alves Filho, H., Orellana, E. T. V., Silva, F. C., Couto, N., Dominguez, D. S., and Hernández, C. R. G. The Application of Spectral Nodal Methods to Discrete Ordinates and Diffusion Problems in Cartesian Geometry for Neutron Multiplying Systems, **Progress in Nuclear Energy**, vol. 42, pp. 385–426, 2003.

Boyarinov, V. F., Kondrushin, A. E., and Fomichenko, P. A. Two–Dimensional Equations of the Surface Harmonics Method for Solving Problems of Spatial Neutron Kinetics in Square–Lattice Reactors, **Physics of Atomic Nuclei**, vol. 77, pp. 1572–1582, 2014.

Burden, R. L. and Faires, J. D. **Numerical Analysis 8E**. Cengage Learning, USA, 2008.

Camargo, D. Q., Bodmann, B. E. J., Vilhena, M. T., Leite, S. Q. B., and Alvim, A. C. M. A stochastic model for neutrons simulation considering the spectrum and nuclear properties with continuous dependence of energy, **Progress in Nuclear Energy**, vol. 69, pp. 59–63, 2013.

Ceolin, C., Schramm, M., Bodmann, B. E. J., Vilhena, M. T., and Leite, Q. B. On an analytical evaluation of the flux and dominant eigenvalue problem for the steady state multi–group multi–layer neutron diffusion equation, **Kerntechnik**, vol. 79 (2014) 5, pp. 430–434, 2014.

Ceolin, C., Schramm, M., Vilhena, M. T., and Bodmann, B. E. J. On the Neutron multi–group kinetic diffusion equation in a heterogeneous slab: An exact solution on a finite set of discrete points, **Annals of Nuclear Energy**, vol. 76, pp. 271–282, 2015.

Chiba, G. Application of the hierarchical domain decomposition boundary element method to the simplified P_3 equation, **Annals of Nuclear Energy**, vol. 38, pp. 1033–1038, 2011.

Duderstadt, J. J. and Hamilton, L. J. **Nuclear Reactor Analysis**. John Wiley & Sons, USA, 1976.

Dulla, S., Mund, E., and Ravetto, P. The Quasi–Static Method Revisited, **Progress in Nuclear Energy**, vol. 50, pp. 908–920, 2008.

Dulla, S., Ravetto, P., Picca, P., and Tomatis, D. Analytical benchmarks for the kinetics of accelerator–driven systems, **Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications**, vol. 1, 2007.

Feingold, D. G. and Varga, R. S. Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem, **Pacific Journal of Mathematics**, vol. 12(4), pp. 1241–1250, 1962.

Ganapol, B. D. **The Analytical Benchmark Library for Nuclear Engineering: PC Version**. Idaho National Engineering Laboratory, Idaho Falls, 1992.

Grossman, L. M. and Hennart, J. P. Nodal diffusion methods for space–time neutron kinetics, **Progress in Nuclear Energy**, vol. 49, pp. 181–216, 2007.

Guessous, N. and Akhmouch, M. Higher order analytical nodal methods in response–matrix formulation for the multigroup neutron diffusion equations, **Annals of Nuclear Energy**, vol. 29, pp. 1765–1778, 2002.

Guyot, M., Gubernatis, P., and Le Tellier, R. Space–time effects in the initiating phase of sodium fast reactors and their evaluation using a three–dimensional neutron kinetics model, **Annals of Nuclear Energy**, vol. 85, pp. 115–126, 2015.

Hamieh, S. D. and Saidinezhad, M. Analytical solution of the point reactor kinetics equations with temperature feedback, **Annals of Nuclear Energy**, vol. 42, pp. 148–152, 2012.

Han, S., Dulla, S., and Ravetto, P. Computational Methods for Multidimensional Neutron Difussion Problems, **Science and Technology of Nuclear Installations**, vol. 1, pp. 1–11, ID 973605, 2009.

Hosseini, S. A. and Vosoughi, N. Development of two–dimensional, multigroup neutron diffusion computer code based on GFEM with unstructured triangle elements, **Annals of Nuclear Energy**, vol. 51, pp. 213–226, 2013.

Kaplan, S., Henry, A. F., Margolis, S. G., and Taylor, J. J. Space–time reactor dynamics, **Proceedings of the Geneva Conference on Peaceful Uses of Atomic Energy**, vol. 4, pp. 41–50, 1964.

Lima, Z. R., Silva, F. C., and Alvim, A. C. M. Solution of the fixed source neutron diffusion equation by using the pseudo–harmonics method, **Annals of Nuclear Energy**, vol. 31, pp. 1649–1666, 2004.

Lima, Z. R., Silva, F. C., and Alvim, A. C. M. A modal multidimensional kinetics method using pseudo–harmonics, **Annals of Nuclear Energy**, vol. 36, pp. 752–759, 2009.

Maiani, M. and Montagnini, B. A Galerkin approach to the boundary element–response matrix method for the multigroup neutron diffusion equations, **Annals of Nuclear Energy**, vol. 31, pp. 1447–1475, 2004.

Mitchell, B. Taylor series method for the solution of the point kinetics equations, **Annals of Nuclear Energy**, vol. 4, pp. 169–176, 1977.

Moghaddam, N. M., Afarideh, H., and Espinosa-Paredes, G. Development of a 2D–Multigroup Code (NFDE–2D) based on the neutron spatial–fractional diffusion equation, **Applied Mathematical Modelling**, vol. 39, pp. 3637–3652, 2015.

Mohsen Ayyoubzadehh, S., Vosoughi, N., and Ayyoubzadeh, S. M. On an improved Direct Discrete Method and its application in two dimensional multi–group neutron diffusion equation, **Annals of Nuclear Energy**, vol. 44, pp. 1–7, 2012.

Mund, E. H. Spectral element solutions for the P_N neutron transport equations, **Computer & Fluids**, vol. 43, pp. 102–106, 2011.

Nahla, A. A. Taylor's series method for solving the nonlinear point kinetics equations, **Nuclear Engineering and Design**, vol. 241, pp. 1592–1595, 2011.

Nahla, A. A., Al-Malki, F. A., and Rokaya, M. Numerical Techniques for the Neutron Diffusion Equations in the Nuclear Reactors, **Adv. Studies Theor. Physics**, vol. 6, pp. 649–664, 2012.

Nahla, A. A. and Zayed, E. M. E. Solution of the nonlinear point nuclear reactor kinetics equations, **Progress in Nuclear Energy**, vol. 52, pp. 743–746, 2010.

Orellana, E. T. V. and Barros, R. C. Analytical Coarse–Mesh Method for Monoenergetic Slab–Geometry Reactor Kinetics using Neutron Diffusion Model, **Journal of Neutron Research**, vol. 10, pp. 1–18, 2002.

Ott, K. O. and Madell, J. T. Quasistatic treatment of spatial phenomenon in reactor dynamics, **Nuclear Science and Engineering**, vol. 26, 1966.

Patankar, S. V. **Numerical Heat Transfer and Fluid Flow**. Hemisphere Publishing Corporation, USA, 1980.

Picca, P. and Furfaro, R. Neutron inverse kinetics via Gaussian Processes, **Annals of Nuclear Energy**, vol. 47, pp. 146–154, 2012.

Quintero-Leyva, B. The multi–group integro–differential equations of the neutron diffusion kinetics. Solutions with the progressive polynomial approximation in multi–slab geometry, **Annals of Nuclear Energy**, vol. 37, pp. 766–770, 2010.

Ravetto, P., Rostagno, M. M., Bianchini, G., Carta, M., and D'Angelo, A. Application of the multipoint method to the kinetics of accelerator–driven systems, **Nuclear Science and Engineering**, vol. 148, pp. 79–88, 2004.

Reuss, P. **Neutron Physics**. EDP Sciences, France, 2008.

Rokrok, B., Minuchehr, H., and Zolfaghari, A. Element–free Galerkin modeling of neutron diffusion equation in X–Y geometry, **Annals of Nuclear Energy**, vol. 43, pp. 39–48, 2012.

Strong, D. M., **Iterative Methods for Solving Ax=b – Convergence Analysis of Iterative Methods**, `http://www.maa.org/press/periodicals/loci/joma/iterative-methods-for-solving-iaxi-ibi-convergence-analysis-of-iterative-methods`, 2005. Accessed in: 2016–02–15.

Sutton, T. M. and Aviles, B. N. Diffusion theory methods for spatial kinetics calculations, **Progress in Nuclear Energy**, vol. 30, pp. 119–182, 1996.

Talamo, A. Numerical solution of the time dependent neutron transport equation by the method of the characteristics, **Journal of Computational Physics**, vol. 240, pp. 248–267, 2013.

Tashakor, S., Jahanfarnia, G., and Hashemi-Tilehnoee, M. Numerical solution of the point reactor kinetics equations with fuel burn–up and temperature feedback., **Annals of Nuclear Energy**, vol. 37, pp. 265–269, 2010.

Vosoughi, N., Salehi, A. A., and Shahriari, M. Discrete formulation for two–dimensional multigroup neutron diffusion equations, **Annals of Nuclear Energy**, vol. 31, pp. 231–253, 2004.

Vyawahare, V. A. and Nataraj, P. S. V. Fractional–order modeling of neutron transport in a nuclear reactor, **Applied Mathematical Modelling**, vol. 37, pp. 9747–9767, 2013.

Wang, K., Rineiski, A., Maschek, W., Wu, H., and Cao, L. Neutron transport model based on the transmission probability method, **Energy Conservation and Management**, vol. 72, pp. 33–38, 2013.

Yasinsky, J. B. and Henry, A. F. Some numerical experiments concerning space–time reactor kinetics behavior, **Nucl. Sci. Eng.**, vol. 22, pp. 171–181, 1965.

Zhang, D. and Rahnema, F. An efficient hybrid stochastic/deterministic coarse mesh neutron transport method, **Annals of Nuclear Energy**, vol. 41, pp. 1–11, 2012.