

Universidade Federal Fluminense

PEDRO MARTINS MENEZES

Simulação Computacional do Transporte de
Nêutrons 2D pelo Método DD

VOLTA REDONDA

2020

PEDRO MARTINS MENEZES

Simulação Computacional do Transporte de Nêutrons 2D pelo Método DD

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

Orientador:

Wesley Luiz Da Silva Assis

Coorientadores:

Vanessa da Silva Garcia
Gustavo Benitez Alvarez

UNIVERSIDADE FEDERAL FLUMINENSE

VOLTA REDONDA

2020

Ficha catalográfica automática - SDC/BIF
Gerada com informações fornecidas pelo autor

M541s Menezes, Pedro Martins
Simulação Computacional do Transporte de Nêutrons 2D
pelo Método DD / Pedro Martins Menezes ; Wesley Luiz Da
Silva Assis, orientador ; Vanessa Da Silva Garcia,
coorientador. Niterói, 2020.
110 f.

Dissertação (mestrado)-Universidade Federal Fluminense,
Niterói, 2020.

DOI: <http://dx.doi.org/10.22409/PPGF.2020.m.09537478750>

1. Equação de Transporte de Nêutrons. 2. Método das
Ordenadas Discretas. 3. Método Multigrupo. 4. Harmônicos
Esféricos. 5. Produção intelectual. I. Da Silva Assis,
Wesley Luiz, orientador. II. Da Silva Garcia, Vanessa,
coorientador. III. Universidade Federal Fluminense. Instituto
de Física. IV. Título.

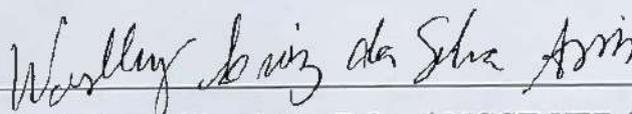
CDD -

Simulação Computacional do Transporte de Nêutrons 2D pelo Método DD

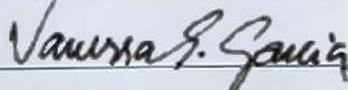
Pedro Martins Menezes

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

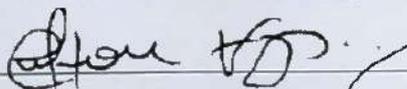
Aprovada por:



Prof. Wesley Luiz da Silva Assis, D.Sc. / MCCT-UFF (Orientador)

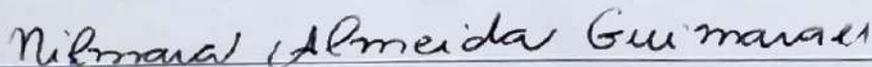


Prof^a. Vanessa da Silva Garcia, D.Sc. / MCCT-UFF (Coorientadora)

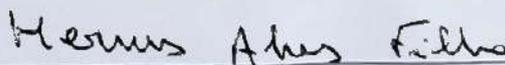


Prof. Welton Alves de Menezes, D.Sc. / MCCT-UFF

Prof. Tiago Araujo Neves D.Sc. / MCCT-UFF



Prof^a. Nilmara Almeida Guimarães, D.Sc. / IFRJ



Prof. Hermes Alves Filho, D.Sc. IPRJ-UERJ

Volta Redonda, 05 de Junho de 2020.

Agradecimentos

A Deus, pela saúde e pela ajuda que mais ninguém pode dar.

A minha família, em especial a minha mãe pelo apoio e incentivo.

Aos meus orientadores, pelo incentivo e paciência.

A todos os demais que contribuíram para a realização deste trabalho.

Resumo

O uso de simulação computacional para o transporte de partículas através da matéria possui muitas aplicações nas ciências físicas, como: física das partículas, física nuclear, medicina nuclear, entre outras. O desenvolvimento de códigos computacionais eficientes para resolver o problema de transporte é de grande importância para várias aplicações. Investiga-se sua aplicação no contexto da *Boron Neutron Capture Therapy* (BNCT). Apresenta-se o código computacional de transporte de nêutrons desenvolvido em C++ a partir do código de referência em F77 (Fortran 77) com o objetivo de obter os mesmos resultados de fluência no estado estacionário para as mesmas configurações de entrada e apresentando tempos menores de execução. Para tal, utilizam-se: método determinístico de ordenadas discretas S_N para a discretização angular combinada com o método P_N para o espalhamento anisotrópico; diferenças finitas DD (*Diamond Difference*) para a discretização espacial; e método multigrupo para a variável energética. A fundamentação teórica do glioma é apresentada, pontuando suas características e modelos crescimento. São abordados os fatores relevantes no cálculo da dose depositada no tecido e comentam-se os principais tipos de modelos de transporte de partículas usados nas simulações computacionais. O modelo de transporte adotado é descrito a partir da equação de Boltzmann linearizada onde são apresentados os passos de discretização da equação e os algoritmos computacionais usados do processo de transporte são detalhados. Os resultados obtidos para as simulações de transporte de nêutrons no meio material são apresentados para os diversos experimentos na forma de fluxos escalares obtidos no estado estacionário e as respectivas doses de energia liberadas como resultado da captura de nêutrons pelo Boro. São comparados os tempos de execução entre os códigos em C++ e F77 onde se verifica que a implementação em C++ executou de 4,5 a 11,5 vezes mais rápido que o código F77 usado como referência. As conclusões e propostas de melhorias para trabalhos futuros são aprestadas. O código de transporte desenvolvido em C++ foi capaz de obter os mesmos resultados do F77 em tempos menores devido a aceleração trazida pela adoção de uma tecnologia moderna (compilador *Intel Parallel Studio XE 2019*) com técnicas de vetorização. As soluções obtidas se comportaram como esperado, apresentando regiões de sumidouro de nêutrons nas regiões de alta concentração de Boro e maiores fluências nas demais regiões.

Abstract

The use of computer simulation for the transport of particles through the matter has many applications in the physical sciences, such as: particle physics, nuclear physics, nuclear medicine, among others. The development of efficient computational codes to solve the transport problem is of great importance for some applications. Its application is investigated in the context of BNCT (*Boron Neutron Capture Therapy*). The neutron transport computational code has been developed in C++ based on the reference code in F77 (Fortran 77) in order to obtain the same steady state results for the same input configuration and presenting shorter execution times. For this, the following are used: deterministic discrete-ordinates method S_N for angular discretization combined with P_N method for anisotropic scattering; finite differences DD (*Diamond Difference*) for spatial discretization; and multi-group method for the energy variable. The theoretical basis of the glioma is presented, addressing its characteristics and basic growth models. The relevant factors are addressed in the calculation of the dose deposited in the tissue and the main types of particle transport models used in computer simulations are commented. The transport model adopted is described from the linearized Boltzmann equation. The discretization steps are presented and the computational algorithms of the transport process are detailed. The results obtained for the simulations of neutron transport in the material medium for the various experiments shows the scalar flux at steady state and the energy doses released as a result of neutron capture by boron. Execution times are compared between C++ and F77 codes and it can be concluded that the C++ implementation ran 4.5 to 11.5 times faster than the F77 code used as a reference. The conclusions and proposals for improvement for future work are presented. The transport code developed in C++ was able to obtain the same results as the F77 in less time due to the acceleration brought by the adoption of modern technology (*Intel Parallel Studio XE 2019* compiler) with vectorization techniques. The solutions obtained behave as expected, with neutron sink regions in the regions with high concentration of Boron and greater fluencies in the other regions.

Palavras-chave

1. Equação de Transporte de Nêutrons
2. Método das Ordenadas Discretas
3. Método Multigrupo
4. Harmônicos Esféricos
5. Método Diamond Difference

Lista de Símbolos

ψ	:	Fluxo angular de nêutrons
\vec{r}	:	Vetor posição
$\hat{\Omega}$:	Vetor direção
E	:	Energia cinética do fluxo
ψ_s	:	Fluxo angular espalhado
ψ_{ext}	:	Fluxo angular externo
ψ_V	:	Fluxo angular nas faces verticais
ψ_H	:	Fluxo angular nas faces horizontais
ϕ	:	Fluxo escalar de nêutrons
ϕ_V	:	Fluxo escalar nas faces verticais
ϕ_H	:	Fluxo escalar nas faces horizontais
Σ_t	:	Seção de choque macroscópica total
Σ_s	:	Seção de choque macroscópica de espalhamento
Σ_a	:	Seção de choque macroscópica de absorção
ψ_0	:	Fonte fixa de fluxo angular
P_N	:	Método de harmônicos esféricos
S_N	:	Método de ordenadas discretas
P_n	:	Polinômio de Legendre
Δt	:	Intervalo de tempo
$\vec{\nabla}\psi$:	Gradiente do fluxo angular
$\hat{\Omega} \cdot \vec{\nabla}\psi$:	Derivada direcional do fluxo angular
\mathcal{D}	:	Domínio espacial
$\partial\mathcal{D}$:	Contorno do domínio espacial
$\partial\mathcal{D}_{feixe}$:	Porção do contorno do domínio espacial

G	:	Número de grupos de energia
g	:	Índice do grupo de energia
E_g	:	Grupo de energia
E_{min}	:	Energia cinética mínima do fluxo angular
E_{max}	:	Energia cinética máxima do fluxo angular
I	:	Número de colunas da malha espacial
i	:	Índice de i-ésima coluna
J	:	Número de linhas da malha espacial
j	:	Índice de j-ésima linha
\hat{n}	:	Vetor normal unitário orientado para fora do domínio
x_i	:	Coordenada x da i-ésima coluna da malha espacial
y_j	:	Coordenada y da j-ésima linha da malha espacial
M	:	Número de direções
m	:	Índice da m-ésima direção discreta
$\hat{\Omega}_m$:	Direção discreta do fluxo angular
μ_m	:	Componente horizontal de $\hat{\Omega}_m$
η_m	:	Componente vertical de $\hat{\Omega}_m$

Lista de Abreviações

SI	:	<i>Source Iteration</i>
DD	:	<i>Diamond Difference</i>
BNCT	:	<i>Boron Neutron Capture Therapy</i>
CPU	:	<i>Central Processing Unit</i>
GPU	:	<i>Graphics Processing Unit</i>
C++	:	Linguagem de programação C++
F77	:	Linguagem de programação Fortran77

Sumário

1	Introdução	15
1.1	Simulação Computacional em BNCT	16
1.2	Viabilidade e Justificativa do Trabalho	18
1.3	Objetivos	18
1.3.1	Objetivo Geral	18
1.3.2	Objetivos Específicos	18
1.4	Estrutura e Organização do Trabalho	19
2	Fundamentação Teórica	21
2.1	Glioma	21
2.2	Modelos de Crescimento	22
2.2.1	Modelos Contínuos	22
2.2.2	Modelos Discretos	23
2.2.3	Modelos Híbridos	23
2.3	Modelos de Dosimetria	23
2.4	Modelos de Transporte	26
2.4.1	Modelos Determinísticos	26
2.4.2	Modelos Estocásticos	27
2.4.3	Modelos Híbridos	27
2.4.4	Modelo de Transporte Adotado	27
2.4.5	Linguagem Computacional Adotada	28

3	Metodologia	29
3.1	Modelo de Transporte	29
3.1.1	Equação Analítica	30
3.1.2	Condições de Contorno	31
3.1.3	Equação Discreta Implícita	31
3.1.3.1	Discretização de $\hat{\Omega}$	32
3.1.3.2	Discretização de ψ_s	33
3.1.3.3	Discretização de E	34
3.1.3.4	Discretização de \vec{r} no Plano XY	35
3.1.4	Equação Discreta Explícita	39
3.1.4.1	Aproximação por <i>Diamond Difference</i>	39
3.2	Algoritmos	40
3.2.1	Transporte de Nêutrons	41
3.2.1.1	Inicialização dos Parâmetros de Quadratura S_N	41
3.2.1.2	Inicialização dos Parâmetros de Quadratura S_4	42
3.2.1.3	Inicialização dos Parâmetros de Quadratura S_6	43
3.2.1.4	Inicialização dos Parâmetros de Quadratura S_8	43
3.2.1.5	Inicialização dos Parâmetros de Quadratura S_{16}	44
3.2.1.6	Inicialização da Malha de Tipos de Materiais	46
3.2.1.7	Inicialização das Seções de Choque	46
3.2.1.8	Inicialização da Malha de Seções de Choque Total	47
3.2.1.9	Inicialização dos Fluxos da Fonte Externa	48
3.2.1.10	Inicialização da Lista de Seções de Choque de Espalhamento	49
3.2.1.11	Inicialização dos Coeficientes Internos de Espalhamento	50
3.2.1.12	Inicialização dos Coeficientes Externos de Espalhamento	51
3.2.1.13	Cálculo do Fluxo Angular Espalhado (ψ_s)	51

3.2.1.14	Cálculo do Fluxo Angular Transportado (ψ_s, ψ_V, ψ_H) . . .	53
3.2.1.15	Cálculo do Fluxo Escalar (ϕ)	54
3.2.1.16	Verificação do Critério de Parada	54
3.2.1.17	Atualização da Solução	55
3.2.1.18	Algoritmo <i>Dose2D</i>	56
3.3	Dados Experimentais	56
4	Resultados e Discussões	61
4.1	Simulações de Transporte	61
4.1.1	Experimentos comparativos entre os códigos F77 e C++	62
4.1.1.1	Experimento Numérico 1	63
4.1.1.2	Experimento Numérico 2	64
4.1.1.3	Experimento Numérico 3	65
4.1.1.4	Experimento Numérico 4	66
4.1.1.5	Experimento Numérico 5	67
4.1.1.6	Experimento Numérico 6	68
4.1.1.7	Experimento Numérico 7	69
4.1.1.8	Experimento Numérico 8	70
4.1.2	Experimentos com o código C++	71
4.1.2.1	Experimento Numérico 1	71
4.1.2.2	Experimento Numérico 2	72
4.1.2.3	Experimento Numérico 3	73
4.1.2.4	Experimento Numérico 4	74
4.1.2.5	Experimento Numérico 5	74
4.1.2.6	Experimento Numérico 6	75
5	Conclusões e Trabalhos Futuros	78

5.1	Conclusões	78
5.2	Trabalhos Futuros	79
Apêndice A – Fluxo Espalhado		80
A.1	Cálculo dos Momentos de Fluxo Espalhado	82
A.1.1	Expressão do Polinômio de Legendre	82
A.1.2	Momento de Fluxo de Grau 0 (ψ_{s0})	83
A.1.3	Momento de Fluxo de Grau 1 (ψ_{s1})	83
A.1.4	Momento de Fluxo de Grau 2 (ψ_{s2})	85
A.1.5	Momento de Fluxo de Grau 3 (ψ_{s3})	86
Apêndice B – Propriedades dos Harmônicos Esféricos 2D		89
B.0.1	Fluxo Nulo na Direção Perpendicular ao Plano	89
B.0.2	Direção Unitária de Espalhamento	89
Apêndice C – Harmônicos Esféricos		90
Apêndice D – Algoritmo Sweep		93
Apêndice E – Listagem de Códigos		98
E.1	saveas3.m	98
E.2	plotSweep.m	98
E.3	plotSweep1234.m	101
E.4	getOrdinates.m	103
E.5	testPlotSH2D.m	104
Referências		107

Capítulo 1

Introdução

O uso de simulação computacional para o transporte de partículas através da matéria possui muitas aplicações nas ciências físicas, como: física das partículas, física nuclear, medicina nuclear, entre outras.

Dentre os códigos mais utilizados para transporte de partículas, destacam-se: o EGS (*Electron Gamma Shower*); MCNP (*Monte Carlo N-Particle*) [21]; ITS (*Integrated Tiger Serie*) [10]; PENELOPE (*Penetration and Energy Loss of Positrons and Electrons*) [34] e o Geant4 (*Geometry And Tracking*) [13].

O Geant4 possui código aberto e se destaca por sua aplicabilidade em diversas áreas por suportar uma grande variedade de recursos como: simular partículas muito lentas e muito rápidas; lidar com geometrias complexas; trabalhar com fontes e geometrias em movimento; e processar campos elétricos e magnéticos. Sua arquitetura de software orientada a objetos possibilita o reuso e extensão de seus componentes. Podem-se citar aplicações na astrofísica, ciências espaciais, física médica e em proteção radiológica [2, 14]. No caso específico da medicina nuclear, tem-se o pacote GATE (*Geant4 Application for Tomography Emission*) que utiliza as bibliotecas do Geant4 e disponibiliza uma plataforma amigável para a simulação de Monte Carlo e geração de imagens de auxílio ao planejamento radioterápico e dosimetria.

Os códigos acima se baseiam em modelos estocásticos que aplicam o método estatístico de Monte Carlo e possuem elevado custo computacional, acarretando em tempos de execução na ordem de várias horas para a obtenção dos resultados.

O desenvolvimento de códigos computacionais eficientes e de baixo custo computacional para resolver o problema de transporte é de grande importância para várias aplicações. Neste trabalho, investiga-se sua aplicação no contexto da *Boron Neutron Capture The-*

rapy (BNCT), onde apresenta-se um código computacional de transporte de nêutrons desenvolvido em C++, a partir de um código de referência em F77 (Fortran 77) com o objetivo de obter os mesmos resultados de fluência no estado estacionário, para as mesmas configurações de entrada e apresentando tempos menores de execução.

Dentro desse contexto, o objetivo desta pesquisa foi desenvolver um código computacional com uma tecnologia atualizada de modo a simular o de transporte de nêutrons de modo a obter o fluxo escalar no estado estacionário, aplicando o método determinístico de ordenadas discretas S_N e usando as aproximações *DD* (*Diamond Difference*) e multigrupo.

1.1 Simulação Computacional em BNCT

A simulação computacional em BNCT é utilizada como ferramenta de auxílio ao planejamento do tratamento, através de uma série de etapas [5, 18, 28, 35]:

- Realização de tomografia computadorizada para capturar a imagem da região cerebral. Este processo gera a Imagem de Entrada (Figura 1.1 (a) e (b)) que permite identificar a olho nu as regiões sadias e doentes a partir das diferentes concentrações de massa presentes em cada região.
- Realização do processamento sobre a Imagem de Entrada para criar uma imagem segmentada em 3 estados. Este processo gera a Imagem de Tipo Material (Figura 1.2(a)) que contém: pixels do tipo 0 representando o vácuo; pixels do tipo 1 representando tecido sadio; e pixels do tipo 2 representado a região tumoral.
- Realização de tomografia computadorizada do tipo F-BPA-PET para detectar a distribuição de Boro na região cerebral. Este processo gera a Imagem de Concentração de Boro (Figura 1.2 (b)) para confirmar a posição central do tumor e área útil a ser usada como alvo da irradiação de nêutrons.
- Realização do cálculo de estimativa de concentração de Boro, a partir da Imagem de Concentração de Boro, de modo a auxiliar na definição da dose de tratamento.
- Simulação do transporte de nêutrons no meio material de modo a obter o fluxo no estado estacionário. Este processo considera as interações de captura e espalhamento do nêutron com os elementos que constituem o meio material simulado.
- Cálculo da dose de energia depositada nos tecidos sadio e doente.

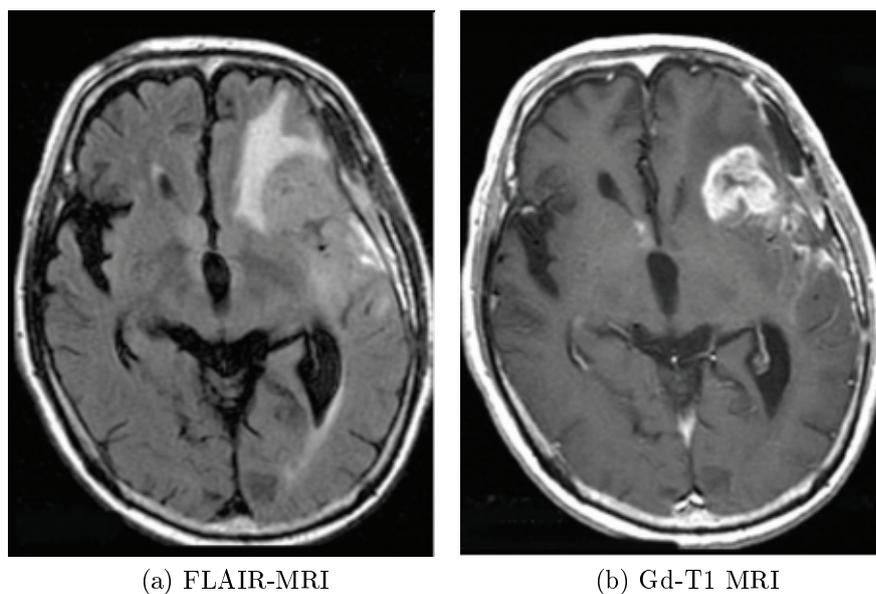


Figura 1.1: Imagens de ressonância magnética (MRI - *Magnetic Resonance Imaging*) e tomografia computadorizada (PET - *Positron Emission Tomography*). (a) FLAIR-MRI (*Fluid-Attenuated Inversion Recovery*) que detecta a região do edema. (b) Gd-T1 MRI (*Gadolinium Enhancement*) que detecta o núcleo do tumor [28].

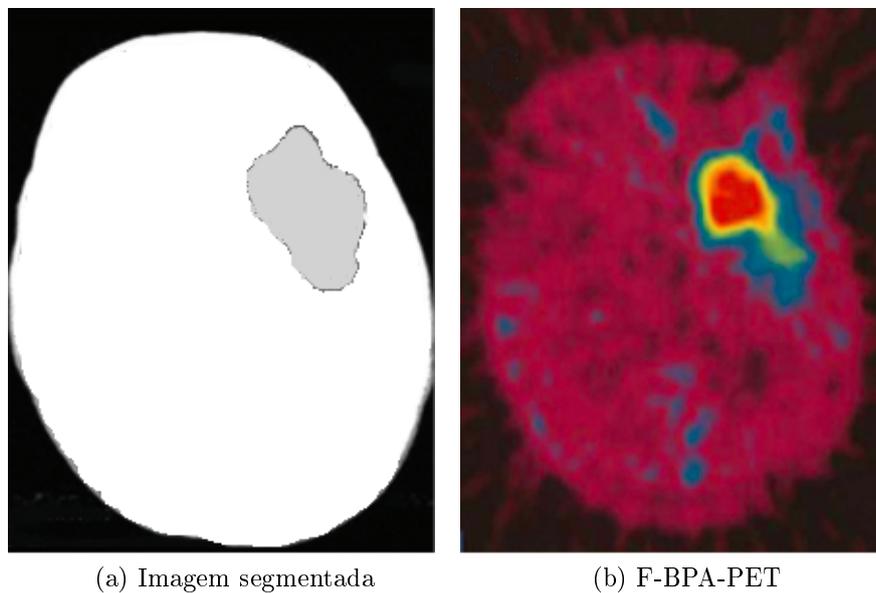


Figura 1.2: Imagens segmentada e de tomografia computadorizada (PET - *Positron Emission Tomography*). (a) Imagem segmentada destacando o vácuo, tecido sadio e tumor. (b) F-BPA-PET (*Fluoro Borophenylalanine*) que detecta a distribuição de Boro [28].

1.2 Viabilidade e Justificativa do Trabalho

A BNCT é uma terapia que se encontra em estágio de evolução e segundo evidências médicas, possui eficiência superior às terapias convencionais. Sua principal característica, que é a seletividade de células tumorais, progride a medida que novos compostos boro são desenvolvidos em conjunto com novos meios de entrega dessas substâncias até o interior das células doentes.

A maior seletividade promovida pela BNCT, leva a uma maior preservação das células saudáveis, minimizando os efeitos colaterais sobre o paciente. Isso permite uma maior flexibilidade na escolha das doses para a eliminação do tumor, seja com o uso de compostos contendo maior concentração de Boro, seja com o aumento da potência do feixe de nêutrons aplicado sobre a região.

A simulação computacional do tratamento é importante neste contexto, devido a sua capacidade de calcular os vários cenários de tratamento e auxiliar a obtenção do melhor conjunto de parâmetros para o planejamento do tratamento. A evolução do tumor pode ser analisada para diferentes esquemas de dose de radiação e fracionamento de sua aplicação, permitindo a escolha do tratamento que minimize o crescimento da massa tumoral.

A integridade dos pacientes é preservada, uma vez que a aplicação de um protocolo de tratamento genérico pode ser substituído por um que respeite seu caso clínico específico.

1.3 Objetivos

1.3.1 Objetivo Geral

Este trabalho tem como objetivo geral o desenvolvimento de código computacional para simular o comportamento do fluxo de nêutrons no tecido cerebral por meio da equação de transporte de Boltzmann. Para tal, utiliza-se o método determinístico de ordenadas discretas S_N para discretização angular, o método P_N para o espalhamento anisotrópico, diferenças finitas DD para a discretização espacial e multigrupo para a variável energética.

1.3.2 Objetivos Específicos

Este trabalho tem como objetivos específicos:

- Calcular o transporte de nêutrons no meio material, quantificando sua absorção e

espalhamento para se obter a fluência no estado estacionário.

- Calcular a dose de radiação liberada no tecido pela reação de captura do nêutron pelo Boro, em função da intensidade do fluxo de nêutrons e concentração de Boro.
- Utilizar a equação de transporte de Boltzmann para se obter o fluxo escalar de nêutrons no estado estacionário.
- Implementar um código determinístico de transporte aplicando o método das ordenadas discretas para a discretização da variável angular, o método DD (*Diamond Difference*) para a discretização da variável espacial e o método multigrupo para a discretização da variável energia.
- Utilizar a equação de cálculo da dose em função do fluxo escalar de nêutrons para calcular a energia depositada no tecido tumoral em cada ponto da malha espacial para cada grupo de energia.
- Implementar o código de cálculo da dose a partir do fluxo escalar de nêutrons de modo a obter o fluxo escalar da dose.

1.4 Estrutura e Organização do Trabalho

Este trabalho adota a seguinte metodologia para a obtenção dos resultados:

- Aplicação do modelo determinístico de transporte de nêutrons para quantificar o fluxo escalar de nêutrons no regime estacionário em tecido heterogêneo contendo regiões de baixa e alta concentração de Boro.

Este trabalho está organizado nos seguintes capítulos:

- **Capítulo 2 - Fundamentação Teórica**

Neste capítulo, são apresentadas as referências teóricas adotadas como base para a construção do trabalho.

- **Capítulo 3 - Metodologia**

Neste capítulo, são apresentadas as aplicações dos métodos adotados para o alcance dos objetivos, detalhando os modelos matemáticos e computacionais empregados.

- **Capítulo 4 - Resultados e Discussão**

Neste capítulo, são apresentados os resultados obtidos pelas simulações computacionais dos modelos de transporte de nêutrons, cálculo da dose e crescimento tumoral.

- **Capítulo 5 - Conclusões**

Neste capítulo, são apresentadas conclusões obtidas a partir dos resultados e as propostas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo apresenta-se a fundamentação teórica utilizada como base para a realização do trabalho proposto. Apresentam-se alguns modelos usados para descrever o crescimento de tumores, destacando-se as contribuições de cada um dos mesmos. Em seguida, são abordados os fatores de tratamento aplicados aos modelos de crescimentos.

2.1 Glioma

Segundo Murray [31], Gliomas são neoplasias de células gliais (células neurais capazes de se dividir) que geralmente ocorrem no hemisfério cerebral superior. São caracterizadas como tumores malignos devido sua propensão em se proliferar e invadir o tecido sadio. No entanto, este tumor não realiza metástase para outras partes do organismo, permanecendo localizado no cérebro e sendo capaz de se infiltrar nas regiões vizinhas ao núcleo central.

Para combater o crescimento deste tumor, é necessário conhecer em profundidade suas características e comportamentos para se definir um modelo teórico que o descreva com precisão suficiente para sua reprodução em laboratório, seja por modelos experimentais (tecidos artificiais) quanto por modelos computacionais (métodos numéricos).

Dentre as características e comportamentos observadas para o tumor, listam-se:

- **Capacidade de Proliferação:**

Células tumorais possuem uma taxa de proliferação maior que as células sadias, devido ao descontrole em seu sistema de contenção do crescimento. A taxa de proliferação é mais elevada nos estágios iniciais do crescimento, e desacelera a medida que o tumor aumenta sua densidade. Neste estágio, observa-se uma mudança de

comportamento celular, onde as células entram em estado necrótico no núcleo central (devido à ausência de nutrientes), seguido de células quiescentes (inertes) na região intermediária seguida por células migratórias, na fronteira do tumor [17, 31].

- **Capacidade de Ramificação:**

O crescimento se dá inicialmente de forma concentrada ao redor de um núcleo central e num segundo momento evolui de forma ramificada para outras regiões. Constatase portanto um comportamento proliferativo das células que evolui para um comportamento migratório, que origina o padrão de infiltração celular para além do núcleo central [17, 31].

- **Capacidade de Difusão:**

A difusão celular nas regiões do cérebro contendo matéria branca (menos densa) é superior à difusão nas regiões de matéria cinza (mais densa). Isso contribui para o espalhamento celular para o interior do cérebro, pois a matéria cinza concentrada na região externa do cérebro oferece maior resistência à difusão, fazendo com que a mesma se intensifique na matéria branca concentrada no interior, que atua como um corredor de espalhamento entre os hemisférios esquerdo e direito [31].

- **Capacidade de Regeneração:**

Mesmo após a ressecção cirúrgica no núcleo central, a presença de células distantes provoca a regeneração do tumor nas regiões de fronteira da área removida [31].

- **Capacidade de Adaptação:**

A distribuição celular no meio é heterogênea, contendo células mais responsivas ao tratamento e outras mais resistentes, devido a processos de mutação que geram novas gerações de células com maior capacidade de sobrevivência [31].

2.2 Modelos de Crescimento

Diversos modelos de descrição do crescimento de tumores foram desenvolvidos pelos pesquisadores, considerando abordagens contínuas, discretas e híbridas [17, 27].

2.2.1 Modelos Contínuos

Os modelos contínuos, modelam o crescimento de forma macroscópica, sem considerar a individualidade de cada célula. Os primeiros modelos desenvolvidos se baseiam em

equações diferenciais ordinárias (EDO's) que descrevem o crescimento tumoral ao longo do tempo. Pode-se citar o modelo de Gompertz, que fornece uma descrição quantitativa e qualitativa do crescimento através de uma curva sigmoide. Os modelos baseados em equações diferenciais parciais (EDP's) descrevem o crescimento ao longo do tempo e do espaço, sendo capazes de representar outras características celulares como proliferação (devido à reprodução) e mobilidade (devido à difusão pelo meio). A geometria do tumor ao longo do tempo pode ser melhor representada ao considerar os diferentes coeficientes de difusão e proliferação para cada ponto do meio material [8, 31].

2.2.2 Modelos Discretos

Os modelos discretos modelam o crescimento de forma microscópica, considerando relações de vizinhança a nível celular e outras propriedades intrínsecas [9]. Os primeiros trabalhos se devem aos modelos de Eden e de Willians & Bjerknæs. Um modelo estocástico é proposto por Smolle & Stettner. Modelos mais recentes empregam autômatos celulares em sua descrição e adotam a lei de crescimento de Gompertz de modo a reproduzir um crescimento exponencial para tempos curtos e saturado para tempos longos, em respeito às observações biológicas [17].

2.2.3 Modelos Híbridos

Os modelos híbridos combinam as abordagens dos modelos contínuos e discretos de modo a obter soluções mais representativas da realidade. Cita-se o modelo de Tanaka [37], utilizado para simular o crescimento do glioma e a invasão celular. A parte discreta do modelo representa o comportamento das células, tais como: migração; comunicação; heterogeneidade espacial das células; diferenciação fenotípica (células migratórias/estacionárias/necróticas); eventos epigenéticos; proliferação (reprodução controlada/descontrolada); e apoptose (morte celular). A partir da análise do comportamento celular, são estimados os pesos a serem aplicados na parte contínua do modelo, que simula o crescimento do tumor e permite avaliar a evolução do mesmo ao longo do tempo.

2.3 Modelos de Dosimetria

No contexto da BNCT, o cálculo da dose de radiação absorvida pelo tecido é fundamental para o planejamento do tratamento. Os modelos de dosimetria buscam identificar

o quanto de energia é depositado no tecido após a interação do nêutron com os elementos químicos encontrados tanto no tecido sadio quanto no tumoral.

As pesquisas na área buscam determinar quais os fatores relevantes no cálculo da dose emitida em cada ponto do volume considerado. Entre os principais fatores, citam-se:

- **Concentração dos Elementos Químicos Naturais do Tecido:**

Identificação dos tipos de elementos químicos presentes na região cerebral e sua distribuição percentual. Cada elemento possui uma capacidade de absorção de nêutrons distinta, dada por sua seção de choque de captura medida em barns (b) onde $1 \text{ b} = 10^{-24} \text{ cm}^2$. No tecido sadio, os principais elementos encontrados e suas seções são: Hidrogênio (0,333 b), Oxigênio (0,19 mb), Carbono (3,5 mb) e Nitrogênio (1,91 b). Conhecendo-se as concentrações desses elementos pode-se construir a seção de choque total do tecido equivalente pela soma ponderada das seções individuais e portanto, quantificar a taxa de absorção de nêutrons no meio [12, 32].

- **Concentração de Boro no Tecido:**

Além dos elementos naturalmente encontrados no cérebro, encontra-se na região tumoral uma maior concentração de Boro (^{10}B). Este elemento é o que possui o maior peso no cálculo da dose resultante, devido ao alto valor de seção de choque de captura (3838 b) de nêutrons térmicos (0,0253 eV), o que contribui para uma maior liberação de energia na reação ($^{10}\text{B}(n, \alpha)^7\text{Li}$). O valor da concentração depende do tipo de composto borado utilizado (ex: BPA, BSH), onde busca-se o composto de maior concentração no tecido tumoral e menor no tecido sadio. Estudos indicam que a concentração ideal de ^{10}B é de $60 \mu\text{g}$ por grama de tecido tumoral [12, 32].

- **Energia do Feixe de Nêutrons:**

A taxa de absorção de nêutrons pelo Boro depende da energia do fluxo que interage com o Boro. A faixa de energia térmica (0,0253 eV) é a que possui maior taxa de absorção e portanto é a energia ideal para maximizar a dose liberada na reação $^{10}\text{B}(n, \alpha)^7\text{Li}$. Sabendo que a energia dos nêutrons diminui a medida que o fluxo avança para o interior do cérebro, tem-se como opção a aplicação de um feixe epitérmico (0,0253 eV a 10 KeV) de modo a obter um fluxo que alcance a região tumoral com energias próximas da faixa térmica [12, 32].

- **Intensidade do Feixe de Nêutrons:**

A taxa de absorção de nêutrons também depende da quantidade de nêutrons que interagem com o ^{10}B por intervalo de tempo. Quanto maior a intensidade de fluxo, medido em $n\text{ cm}^{-2}\text{s}^{-1}$, tem-se uma correspondente maior fluência (intensidade integrada no tempo), dada em $n\text{ cm}^{-2}$. Estudos indicam que a fluência ideal de nêutrons térmicos é de $10^{13}n\text{ cm}^{-2}$ na região do tumor [12, 32].

- **Tempo de Permanência do Boro:**

O tempo de permanência do composto borado no organismo varia para cada indivíduo. Este parâmetro influencia na estimativa da intensidade do feixe de nêutrons utilizado. Tempos maiores de permanência possibilitam uma aplicação de uma menor intensidade de modo a minimizar os efeitos colaterais devidos à radiação de nêutrons nos demais tecidos [12, 32].

A literatura especializada considera que a dose total depositada no tecido corresponde à soma das doses individuais devido aos seguintes fatores de radiação:

- **Dose Devido aos Raios Gama (D_γ):**

Dose liberada pela reação de captura radioativa $^1\text{H}(n, \gamma)^2\text{H}$ que libera raios gama com energia 2,2 MeV, que alcança regiões mais profundas do tecido, percorrendo vários centímetros [7, 12, 29, 32].

- **Dose Devido aos Nêutrons Rápidos e Epitérmicos (D_n):**

Dose devido aos nêutrons rápidos (energia maior que 10 KeV) liberada pelas reações $H(n, n')H$, $O(n, n')O$ e $C(n, n')C$ de espalhamento elástico dos nêutrons. Dose devido aos nêutrons epitérmicos (0,0253 eV a 10 KeV) na reação $H(n, n')H$ [7, 12, 29, 32].

- **Dose Devido aos Prótons (D_p):**

Dose devido aos nêutrons térmicos (0,0253 eV) provocando a reação de produção de partículas $^{14}\text{N}(n, p)^{14}\text{C}$ que libera prótons de recuo com energia aproximada de 0,6 MeV capazes de percorrer $10\mu\text{m}$ no tecido [7, 12, 29, 32].

- **Dose Devido ao Boro (D_B):**

Dose liberada pela reação de captura do nêutrons térmicos pelo Boro $^{10}\text{B}(n, \alpha)^7\text{Li}$ que libera energia num diâmetro aproximado de $10\mu\text{m}$ capaz de destruir as células

cancerosas, que possuem diâmetro aproximado de $20 \mu\text{m}$. Corresponde a soma das doses depositadas pelo ${}^7\text{Li}$ (0,84 MeV) e ${}^4\text{He}$ (1,47 MeV) totalizando 2,31 MeV [7, 12, 29, 32].

Os demais elementos encontrados no tecido, como *Na*, *K*, *Ca*, *Cl*, *P* e *S* possuem baixas concentrações e menores seções de choque e portanto são desprezados no cálculo da dose total [20, 32]. Busca-se com o estudo da dosimetria obter a dose considerada adequada para o BNCT segundo os parâmetros de entrada: 1) Concentração de 10^9 átomos de Boro intracelular; 2) Ocorrência de 10 reações de captura de nêutrons por célula; e 3) Fluência de nêutrons térmicos de 10^{13} n/cm^2 [7, 32]. Neste trabalho é aplicado o cálculo da dose devido ao Boro, onde o processo é descrito pelo Algoritmo 18.

2.4 Modelos de Transporte

Os modelos de transporte de partículas podem ser didaticamente agrupados em modelos determinísticos, modelos estocásticos e modelos híbridos [3, 11, 25].

2.4.1 Modelos Determinísticos

Os modelos determinísticos se baseiam em leis físicas que descrevem o movimento preciso das partículas no espaço, a partir do conhecimento de suas posições e velocidades iniciais, obedecendo restrições ao movimento dadas por uma série de fatores, como: limitações espaciais impostas pelo domínio considerado; colisões com outras partículas; transferência de energia entre as partículas e o meio durante as colisões; etc. Tais modelos são eficientes quando as partículas obedecem às restrições da física clássica. Dentre os modelos determinísticos, destaca-se o método das ordenadas discretas S_N [3, 4, 12, 30].

O método das ordenadas discretas S_N consiste na discretização da variável angular do fluxo de nêutrons em M direções. O termo integral é aproximado por uma fórmula de quadratura numérica usando simetria de nível. O processo produz um sistema de M EDOs lineares para o fluxo angular, que pode ser resolvido analiticamente ou numericamente. O método S_N foi introduzido por G.C.WICK (1943) na solução de problemas de transporte e CHANDRASEKHAR (1950) em estudos de transferência radiativa.

Os modelos determinísticos buscam a obtenção da solução da equação de transporte de Boltzmann e o método S_N se destaca por sua baixa complexidade computacional comparada aos demais métodos, mas possui limitações como a ocorrência de fluxos negativos

durante os cálculos e a distribuição de fluxos em direções artificialmente dominantes (*Ray Effects*) [3, 4, 22, 23].

2.4.2 Modelos Estocásticos

Os modelos de transporte estocásticos, por outro lado, consideram a natureza caótica do movimento das partículas, e se baseiam em descrições probabilísticas das posições e direções das partículas para representar os eventos de colisão e espalhamento. É um modelo fisicamente mais adequado para representar o fenômeno de transporte uma vez que respeita o princípio de incerteza para o comportamento de partículas elementares. Simulações realizadas com tais modelos geram saídas diferentes para um mesmo estado inicial de movimento, em termos quantitativos. Em termos qualitativos, geram resultados semelhantes, uma vez que sua natureza estocástica não significa o mesmo que aleatória, o que faz com que as posições das partículas sigam um padrão de distribuição. Dentre os modelos estocásticos, destacam-se os modelos baseados em Monte Carlo [3, 12].

2.4.3 Modelos Híbridos

Os modelos de transporte híbridos buscam combinar a velocidade proporcionada pelos modelos determinísticos com a melhor aproximação física fornecida pelos métodos estocásticos. Tradicionalmente, para o método de ordenadas discretas, sabe-se que a mitigação dos *Ray Effects* pode ser obtida pelo aumento da ordem da quadratura angular e pelo refinamento da malha espacial. Desenvolvimentos recentes aplicam soluções híbridas para minimizar os *Ray Effects* aplicando rotações aleatórias no conjunto de quadraturas [6, 19]. Outras abordagens usam a integração *quasi* Monte Carlo [24] para selecionar as direções usadas na integração numérica do fluxo angular espalhado da equação de transporte.

2.4.4 Modelo de Transporte Adotado

Neste trabalho, adotou-se o modelo determinístico de transporte baseado na equação de Boltzmann lineariza, fazendo uso de quadraturas e polinômios de Legendre para aproximar o termo de espalhamento de partículas e de seções de choque pré-definidas para quantificar o percentual de nêutrons que são absorvidos e espalhados após os eventos de colisão com partículas mais pesadas (núcleos atômicos) durante o transporte através do meio material [11, 25, 36]. O código de referência desenvolvido em F77 [12] foi previamente validado a partir do código *Monte Carlo N-Particle* (MCNP) [21] e o código em

C++ deste trabalho foi desenvolvido para gerar os mesmo resultados a partir das mesmas entradas.

2.4.5 Linguagem Computacional Adotada

Adotou-se a linguagem de programação C++ como alternativa de implementação para a migração do código legado em F77 (Fortran 77), devido ao fato do C++ possuir semelhanças como uma linguagem nativa capaz de gerar programas de alta performance. O uso do C++ apresentou a vantagem de não se limitar ao uso de alocação estática de memória para o armazenamento das variáveis. A linguagem adotada possui maior flexibilidade, permitindo uma melhor modularização do código, facilitando o seu entendimento, manutenção e evolução. Para a obtenção de uma aceleração da execução do código de transporte, usou-se o compilador da Intel configurado para utilizar opções de otimização de performance. O fluxo do código original foi inteiramente revisado e simplificado para gerar as mesmas saídas a partir das mesmas entradas. Para tal, foram feitas simplificações em algumas funções para remover expressões desnecessárias ao modelo de transporte adotado. Os diversos *loops* aninhados foram revisados para iterar numa ordem mais propícia à forma vetorizada. O paralelismo por threads foi habilitado como uma opção de compilação (`/Qparallel`) e sua aplicação se deu de forma transparente para o programador.

Capítulo 3

Metodologia

3.1 Modelo de Transporte

O modelo de transporte adotado se baseia na equação (3.1) de balanço do número n de nêutrons num volume arbitrário V durante um intervalo de tempo Δt :

$$[n \text{ em } t + \Delta t] = [n \text{ em } t] + [\text{Acréscimo em } \Delta t] + [\text{Descréscimo em } \Delta t] \quad (3.1)$$

Este modelo considera que os nêutrons migram com distintas velocidades v , múltiplas direções $\hat{\Omega}$ e podem ocupar todas as posições \vec{r} do domínio espacial. Ao longo do texto adota-se a notação de fluxo $\psi = n \cdot v$ para descrever a quantidade de nêutrons n que se deslocam com velocidade v na direção $\hat{\Omega}$ e posição \vec{r} . Mais precisamente, estuda-se o fluxo angular de nêutrons $\psi = \psi(\vec{r}, E, \hat{\Omega})$ onde a energia cinética de repouso é dada por $E = \frac{1}{2}mv^2$ e m é a massa de repouso do nêutron. A fluência de nêutrons ψ possui unidades de $n \text{ cm}^{-2}\text{s}^{-1}$.

Quanto aos fatores de acréscimo de nêutrons, são considerados:

- **Ganho por Fonte Externa**, dada por um feixe colimado localizado próximo ao exterior da caixa craniana, que injeta um fluxo nas direções que apontam para o interior do domínio contém uma energia inicial conhecida.
- **Ganho por espalhamento**, ocorre quando $\psi(\vec{r}, E, \hat{\Omega})$ de menor energia recebe partículas de $\psi(\vec{r}, E', \hat{\Omega}')$ com maior energia, causando o aumento de $\psi(\vec{r}, E, \hat{\Omega})$. Isso ocorre durante eventos de colisão dos fluxos incidentes $\psi(\vec{r}, E', \hat{\Omega}')$ que desaceleram e originam fluxos emergentes $\psi(\vec{r}, E, \hat{\Omega})$ de menor energia.

Quanto aos fatores de decréscimo de nêutrons, são considerados:

- **Perda por absorção**, causada pela captura de nêutrons pelos núcleos dos átomos proveniente de colisões durante o transporte de nêutrons ao longo meio material.
- **Perda por espalhamento**, ocorre quando $\psi(\vec{r}, E, \hat{\Omega})$ de maior energia fornece partículas para $\psi(\vec{r}, E', \hat{\Omega}')$ de menor energia, causando a diminuição de $\psi(\vec{r}, E, \hat{\Omega})$. Isso ocorre durante eventos de colisão dos fluxos incidentes $\psi(\vec{r}, E, \hat{\Omega})$ que desaceleram e originam fluxos emergentes $\psi(\vec{r}, E', \hat{\Omega}')$ de menor energia.
- **Perda por fuga**, que ocorre quando o fluxo de nêutrons $\psi(\vec{r}, E, \hat{\Omega})$ em $t + \Delta t$ é maior que o fluxo conhecido $\psi(\vec{r}, E, \hat{\Omega})$ em t .

3.1.1 Equação Analítica

Neste trabalho o transporte de nêutrons é modelado pela Equação Linear de Transporte de Boltzmann no estado estacionário (3.2). Esta equação estabelece a relação de balanço do fluxo de nêutrons ψ no espaço de fase $(\vec{r}, E, \hat{\Omega})$ de modo a descrever a quantidade de nêutrons que se desloca na posição \vec{r} , com energia cinética E e fluindo na direção unitária $\hat{\Omega}$. No caso estacionário (onde $\partial\psi/\partial t = 0$) é expressa como:

$$\hat{\Omega} \cdot \vec{\nabla} \psi(\vec{r}, E, \hat{\Omega}) + \Sigma_t(\vec{r}, E) \psi(\vec{r}, E, \hat{\Omega}) = \psi_{ext}(\vec{r}, E, \hat{\Omega}) + \psi_s(\vec{r}, E, \hat{\Omega}) \quad (3.2)$$

onde o fluxo espalhado (3.3) é dado por:

$$\psi_s(\vec{r}, E, \hat{\Omega}) = \int_0^\infty \int_{4\pi} \Sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega}) \psi(\vec{r}, E', \hat{\Omega}') d\Omega' dE' \quad (3.3)$$

Os termos da equação são:

- $\hat{\Omega} \cdot \vec{\nabla} \psi(\vec{r}, E, \hat{\Omega})$ Termo de migração. Contabiliza a variação espacial de nêutrons $\vec{\nabla} \psi$ na direção do vetor $\hat{\Omega}$.
- $\Sigma_t(\vec{r}, E) \psi(\vec{r}, E, \hat{\Omega})$ Termo perdas por colisão. Contabiliza a perda de nêutrons devido à eventos de colisão com os núcleos atômicos do meio material.
- $\psi_{ext}(\vec{r}, E, \hat{\Omega})$ Termo fonte externa isotrópica. Contabiliza o ganho de nêutrons devido ao feixe, que neste trabalho está localizado no contorno do domínio.
- $\psi_s(\vec{r}, E, \hat{\Omega})$ Termo fonte de espalhamento. Contabiliza o ganho de nêutrons a partir de outros nêutrons, que após colisões, migram de direção $\hat{\Omega}' \rightarrow \hat{\Omega}$ e energia $E' \rightarrow E$.

As variáveis da equação são:

- $\psi(\vec{r}, E, \hat{\Omega})$ Fluxo angular de nêutrons na posição \vec{r} , energia E e direção $\hat{\Omega}$.
- $\psi_{ext}(\vec{r}, E, \hat{\Omega})$ Fluxo externo, representada pelo feixe fixo de nêutrons.
- $\psi_s(\vec{r}, E, \hat{\Omega})$ Fluxo espalhado em $(\vec{r}, E, \hat{\Omega})$ a partir de fluxos incidentes $\psi_s(\vec{r}, E', \hat{\Omega}')$.
- $\Sigma_t(\vec{r}, E)$ Seção de Choque Macroscópica Total. Quantifica a probabilidade de perda de nêutrons devido a absorção (captura e fissão) e espalhamento.
- $\Sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega})$ Seção de Choque Diferencial de Espalhamento. Quantifica a probabilidade de migração de energia e direção de nêutrons em eventos de colisão. Nêutrons incidentes, com energia E' e direção $\hat{\Omega}'$ migram como nêutrons emergentes, com energia E e direção $\hat{\Omega}$.

3.1.2 Condições de Contorno

O fluxo incidente de nêutrons posicionado no exterior do domínio \mathcal{D} (caixa craniana) é representado pela Condição de Contorno Prescrita, dada por (3.4):

$$\psi_{ext}(\vec{r}, E, \hat{\Omega}) = \psi_0 \text{ em } \vec{r} \in \partial\mathcal{D}_{feixe} \subset \partial\mathcal{D} \quad (3.4)$$

onde $\psi_0 > 0$ é o valor fixo da fonte de nêutrons usada no tratamento, $\partial\mathcal{D}$ corresponde à todas as posições do contorno do domínio e $\partial\mathcal{D}_{feixe}$, somente às posições onde se localiza o feixe de nêutrons.

A ausência de fluxo para dentro da caixa craniana, proveniente do meio externo é representada pela Condição de Contorno - Tipo Vácuo, representada por (3.5):

$$\psi(\vec{r}, E, \hat{\Omega}) = 0 \text{ para } \hat{\Omega} \cdot \hat{n} < 0 \text{ e } \vec{r} \in \partial\mathcal{D} - \partial\mathcal{D}_{feixe} \quad (3.5)$$

onde \hat{n} é o vetor normal na direção exterior ao domínio e $\hat{\Omega}$ é a direção do fluxo incidente e \vec{r} são as posições livres do contorno, onde não há o feixe incidente.

3.1.3 Equação Discreta Implícita

A forma discreta da Equação de Transporte de Nêutrons é obtida através da discretização das direções $\hat{\Omega}$ em M direções $\hat{\Omega}_m$, das energias E em G grupos de energia E_g e das posições \vec{r} em $I \times J$ coordenadas x_i e y_j . Nas demais seções são apresentados as etapas

do processo de transformação da EDP de Boltzmann num sistema linear de equações algébricas, passível de ser resolvido numericamente.

3.1.3.1 Discretização de $\hat{\Omega}$

As direções $\hat{\Omega}$ do fluxo angular são discretizadas em M direções prescritas, que no caso 2D possui valor $M = N(N+2)/2$ e a equação original se transforma em M equações, uma para cada direção. A Figura 3.1 ilustra as direções para o caso onde $N = 4$ e M possui 12 direções no total, 3 para cada um dos quatro quadrantes:

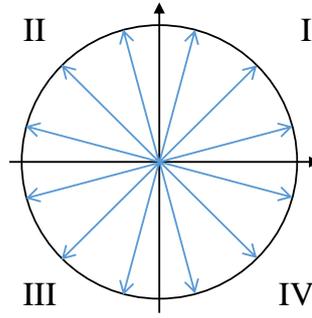


Figura 3.1: Discretização das direções

Expressa-se a direção $\hat{\Omega}_m$ como (3.6):

$$\hat{\Omega}_m = \hat{\Omega}(\mu_m, \eta_m) \quad (3.6)$$

O fluxo angular na direção $\hat{\Omega}_m$ é representado como:

$$\psi_m(\vec{r}, E) = \psi(\vec{r}, E, \hat{\Omega}_m) \quad (3.7)$$

O fluxo externo na direção $\hat{\Omega}_m$ é representado como:

$$\psi_{ext,m}(\vec{r}, E) = \psi_{ext}(\vec{r}, E, \hat{\Omega}_m) \quad (3.8)$$

O fluxo espalhado na direção $\hat{\Omega}_m$ é representado como:

$$\psi_{s,m}(\vec{r}, E) = \psi_s(\vec{r}, E, \hat{\Omega}_m) \quad (3.9)$$

A equação de transporte 3.2, discretizada em $\hat{\Omega}$ se transforma em M equações:

$$\hat{\Omega}_m \cdot \vec{\nabla} \psi_m(\vec{r}, E) + \Sigma_t(\vec{r}, E) \psi_m(\vec{r}, E) = \psi_{ext,m}(\vec{r}, E) + \psi_{s,m}(\vec{r}, E) \quad (3.10)$$

3.1.3.2 Discretização de ψ_s

O fluxo espalhado ψ_s dado pela equação 3.3 é discretizado usando expansão em polinômios de Legendre. A expressão resultante corresponde a uma aproximação do fluxo espalhado na direção Ω_m e energia E a partir do ganho de nêutrons proveniente de todos os outros fluxos provenientes das direções Ω_n e energias E' .

A partir da seção de choque de espalhamento conhecida Σ_s , constroem-se os L momentos de seção de choque de espalhamento Σ_{sl} com $0 \leq l < L$ combinada a uma quadratura que defina M direções discretas $\hat{\Omega}_m = (\mu_m, \eta_m)$ e pesos ω_m . Pode-se expressar o fluxo espalhado $\psi_{s,m}$ como a combinação linear dos momentos de fluxo espalhado $\psi_{sl,m}$.

Neste trabalho, adota-se $L = 3$ como grau de espalhamento anisotrópico onde adotou-se a quadratura de nível da referência [25]. O desenvolvimento das expressões do fluxo pode ser encontrado no apêndice A. O fluxo espalhado usando a aproximação de grau $L = 3$ é dada pela equação 3.11:

$$\psi_{s,m} = \psi_{s0,m} + \psi_{s1,m} + \psi_{s2,m} + \psi_{s3,m} \quad (3.11)$$

$$\psi_{s0,m} = \Sigma_{s0} \left(\frac{1}{4} \sum_{n=1}^M \omega_n \psi_n \right) \quad (3.12)$$

$$\psi_{s1,m} = \Sigma_{s1} \left(\begin{array}{l} \frac{3}{4} \mu_m \sum_{n=1}^M \mu_n \omega_n \psi_n \quad + \\ \frac{3}{4} \eta_m \sum_{n=1}^M \eta_n \omega_n \psi_n \end{array} \right) \quad (3.13)$$

$$\psi_{s2,m} = \Sigma_{s2} \left(\begin{array}{l} \left(\frac{5}{4} - \frac{15}{8} \mu_m^2 - \frac{15}{8} \eta_m^2 \right) \sum_{n=1}^M \omega_n \psi_n \quad + \\ \left(\frac{15}{4} \mu_m \eta_m \right) \sum_{n=1}^M \mu_n \eta_n \omega_n \psi_n \quad + \\ \left(\frac{15}{4} \mu_m^2 + \frac{15}{8} \eta_m^2 - \frac{15}{8} \right) \sum_{n=1}^M \mu_n^2 \omega_n \psi_n \quad + \\ \left(\frac{15}{4} \eta_m^2 + \frac{15}{8} \mu_m^2 - \frac{15}{8} \right) \sum_{n=1}^M \eta_n^2 \omega_n \psi_n \end{array} \right) \quad (3.14)$$

$$\psi_{s3,m} = \Sigma_{s3} \left(\begin{array}{l} \left(\frac{21}{2} \mu_m - \frac{105}{8} \mu_m^3 - \frac{105}{8} \mu_m \eta_m^2 \right) \sum_{n=1}^M \mu_n \omega_n \psi_n + \\ \left(\frac{21}{2} \eta_m - \frac{105}{8} \eta_m^3 - \frac{105}{8} \eta_m \mu_m^2 \right) \sum_{n=1}^M \eta_n \omega_n \psi_n + \\ \left(-\frac{105}{8} \eta_m + \frac{105}{8} \eta_m^3 + \frac{105}{4} \mu_m^2 \eta_m \right) \sum_{n=1}^M \mu_n^2 \eta_n \omega_n \psi_n + \\ \left(-\frac{105}{8} \mu_m + \frac{105}{8} \mu_m^3 + \frac{105}{4} \eta_m^2 \mu_m \right) \sum_{n=1}^M \eta_n^2 \mu_n \omega_n \psi_n + \\ \left(-\frac{105}{8} \mu_m + \frac{35}{2} \mu_m^3 + \frac{105}{8} \mu_m \eta_m^2 \right) \sum_{n=1}^M \mu_n^3 \omega_n \psi_n + \\ \left(-\frac{105}{8} \eta_m + \frac{35}{2} \eta_m^3 + \frac{105}{8} \eta_m \mu_m^2 \right) \sum_{n=1}^M \eta_n^3 \omega_n \psi_n \end{array} \right) \quad (3.15)$$

3.1.3.3 Discretização de E

A discretização da variável energética é realizada particionando o intervalo de energias $[E_{min}, E_{max}]$ em G grupos de energia com $E_0 = E_{max}$ e $E_G = E_{min}$ de tal forma que:

$$E_0 > E_1 > E_2 > \dots > E_G \quad (3.16)$$

O fluxo na direção $\hat{\Omega}_m$ e grupo E_g é expresso como a soma dos fluxos no intervalo de energias E_g e E_{g-1} :

$$\psi_{mg}(\vec{r}) = \int_{E_g}^{E_{g-1}} \psi_m(\vec{r}, E) dE \quad (3.17)$$

A seção de choque total para o grupo E_g é dada pela soma das seções de choque no intervalo de energias E_g e E_{g-1} ponderado pelo fluxo escalar ϕ neste intervalo:

$$\Sigma_{t,g}(\vec{r}) = \frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\vec{r}, E) \phi(\vec{r}, E) dE}{\int_{E_g}^{E_{g-1}} \phi(\vec{r}, E) dE} \quad (3.18)$$

onde o fluxo escalar corresponde a soma dos fluxos angulares:

$$\phi(\vec{r}, E) = \int_{4\pi} \psi(\vec{r}, E, \hat{\Omega}) d\hat{\Omega} \quad (3.19)$$

O fluxo externo na direção $\hat{\Omega}_m$ e grupo E_g é a soma dos fluxos externos nas energias

entre E_g e E_{g-1} :

$$\psi_{ext,mg}(\vec{r}) = \int_{E_g}^{E_{g-1}} \psi_{ext,m}(\vec{r}, E) dE \quad (3.20)$$

O fluxo espalhado na direção $\hat{\Omega}_m$ e grupo E_g é a soma dos fluxos espalhados nas energias entre E_g e E_{g-1} :

$$\psi_{s,mg}(\vec{r}) = \int_{E_g}^{E_{g-1}} \psi_{s,m}(\vec{r}, E) dE \quad (3.21)$$

A equação de transporte 3.2, discretizada em $\hat{\Omega}$ e E se transforma em $M \times G$ equações:

$$\hat{\Omega}_m \cdot \vec{\nabla} \psi_{mg}(\vec{r}) + \Sigma_{t,g}(\vec{r}) \psi_{mg}(\vec{r}) = \psi_{ext,mg}(\vec{r}) + \psi_{s,mg}(\vec{r}) \quad (3.22)$$

3.1.3.4 Discretização de \vec{r} no Plano XY

A discretização da variável espacial \vec{r} é realizada particionando-se o domínio XY em I intervalos na direção x e J intervalos na direção y , de modo que a equação de transporte 3.2 seja avaliada para cada célula (i, j) do plano. A Figura 3.2 ilustra o particionamento uniforme do plano nas direções X e Y.

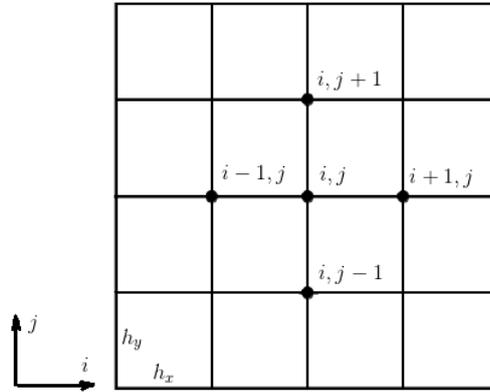


Figura 3.2: Discretização espacial

Expressando a equação de transporte 3.22 no plano tem-se a equação (3.23):

$$\mu_m \frac{\partial}{\partial x} \psi_{mg}(x, y) + \eta_m \frac{\partial}{\partial y} \psi_{mg}(x, y) + \Sigma_{t,g}(x, y) \psi_{mg}(x, y) = \psi_{ext,mg}(x, y) + \psi_{s,mg}(x, y) \quad (3.23)$$

Para obter a forma discretizada da equação, integra-se a 3.23 no volume de referência $V_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ para se obter os momentos de fluxo nas faces e centros de V_{ij} .

Expressando o fluxo nas faces verticais das células ψ_V como a média dos fluxos ao

longo das mesmas, tem-se:

$$\psi_{V_{mg}}^{ij} = \frac{1}{h_{yj}} \int_{y_j}^{y_{j+1}} \psi_{mg}(x_i, y) dy \quad (3.24)$$

Expressando o fluxo nas faces horizontais das células ψ_H como a média dos fluxos ao longo das mesmas, tem-se:

$$\psi_{H_{mg}}^{ij} = \frac{1}{h_{xi}} \int_{x_i}^{x_{i+1}} \psi_{mg}(x, y_j) dx \quad (3.25)$$

Expressando o fluxo no centro das células ψ como a média dos fluxos em seu interior, tem-se:

$$\psi_{mg}^{ij} = \frac{1}{h_{xi}h_{yj}} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \psi_{mg}(x, y) dy dx \quad (3.26)$$

Expressando a seção de choque no centro das células Σ_t como a média das seções em seu interior, tem-se:

$$\Sigma_{t,g}^{i,j} = \frac{1}{h_{xi}h_{yj}} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \Sigma_{t,g}(x, y) dy dx \quad (3.27)$$

Expressando o fluxo externo no centro das células ψ_{ext} como a média dos fluxos em seu interior, tem-se:

$$\psi_{ext,mg}^{ij} = \frac{1}{h_{xi}h_{yj}} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \psi_{ext,mg}(x, y) dy dx \quad (3.28)$$

Expressando o fluxo espalhado no centro das células ψ_s como a média dos fluxos em seu interior, tem-se:

$$\psi_{s,mg}^{ij} = \frac{1}{h_{xi}h_{yj}} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \psi_{s,mg}(x, y) dy dx \quad (3.29)$$

Ao integrar a equação 3.23 para uma célula (i, j) ilustrada na Figura 3.2 e aplicando as substituições, obtém-se 3.30:

$$\frac{\mu_m}{h_{xi}} (\psi_{V,mg}^{iOut,j} - \psi_{V,mg}^{iIn,j}) + \frac{\eta_m}{h_{yj}} (\psi_{H,mg}^{i,jOut} - \psi_{H,mg}^{i,jIn}) + \Sigma_{t,g}^{ij} \psi_{mg}^{ij} = \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij} \quad (3.30)$$

A equação discreta 3.30 corresponde à equação analítica 3.2 discretizada em $\hat{\Omega}$, E e \vec{r} . A equação 3.2 se transforma em $M \times G \times I \times J$ equações algébricas. Podem-se identificar três grupos de incógnitas: fluxos centrais ψ , ψ_{ext} e ψ_s ; fluxos nas faces verticais das células ψ_V ; e fluxos nas faces horizontais ψ_H . A Figura 3.3 ilustra as malhas para cada um dos

grupos de incógnitas, considerando $N = 4$ e $M = 12$:

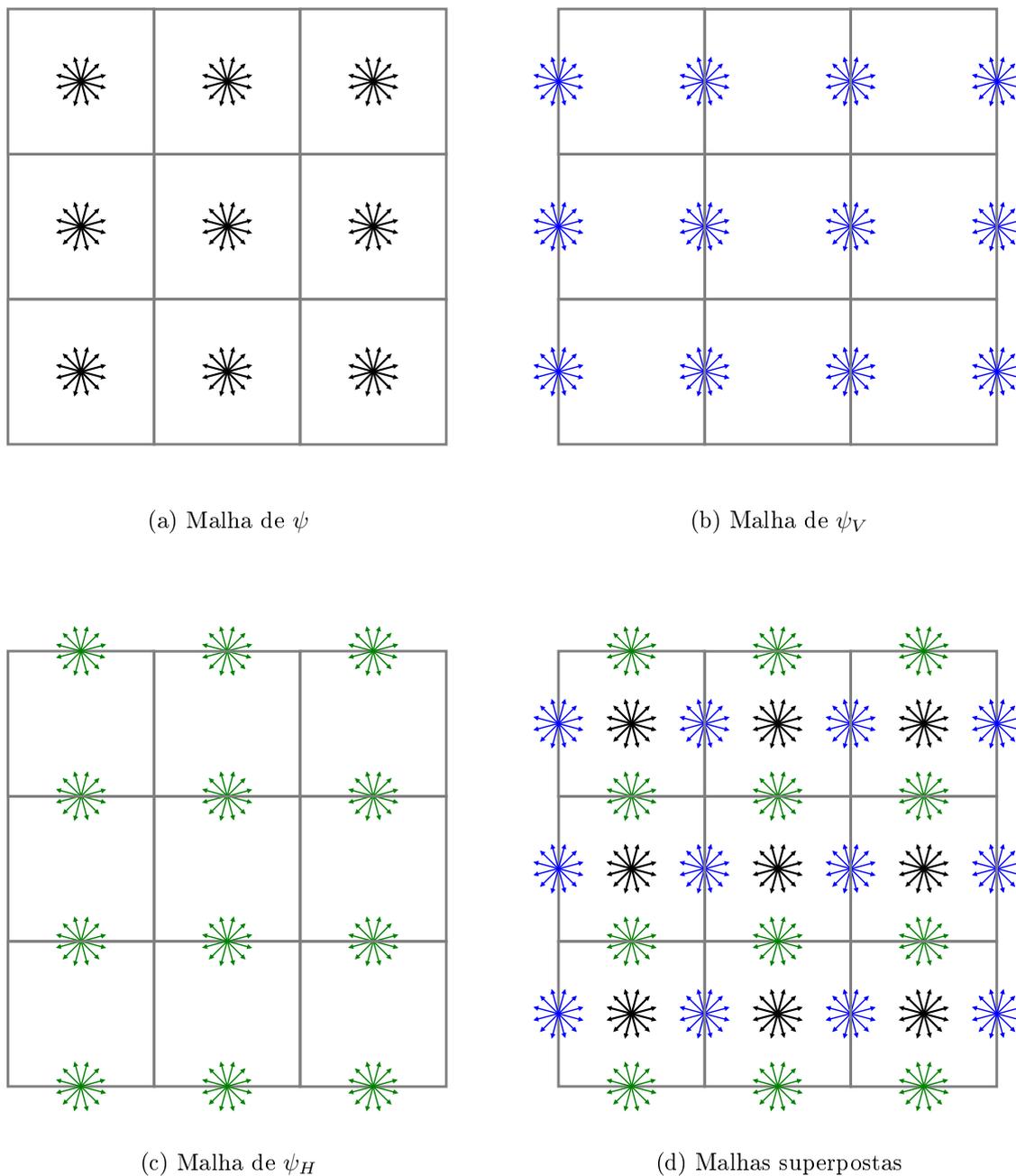


Figura 3.3: Malhas das discretização espacial

Os índices dos fluxos incidentes nas faces verticais (iIn) e horizontais (jIn) e dos fluxos emergentes ($iOut$) e ($jOut$) dependem dos sinais dos componentes de Ω_m de acordo com

[30]:

$$\begin{aligned}
 iIn &= i, & iOut &= i + 1, & \mu_m &> 0 \\
 jIn &= j, & jOut &= j + 1, & \eta_m &> 0 \\
 iIn &= i + 1, & iOut &= i, & \mu_m &< 0 \\
 jIn &= j + 1, & jOut &= j, & \eta_m &< 0
 \end{aligned} \tag{3.31}$$

A Figura 3.4 ilustra as direções de entrada e saída de cada quadrante.

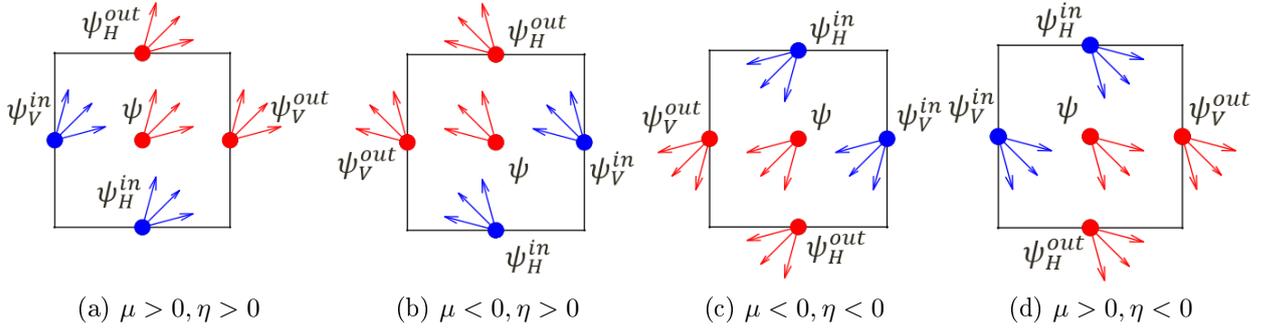


Figura 3.4: Direções de entrada e saída para cada quadrante

Portanto, a equação de transporte 3.30 é resolvida para cada quadrante.

A variável ψ é discretizada em $i = [1, \dots, I], j = [1, \dots, J], m = [1, \dots, M], g = [1, \dots, G]$ de modo a representar os $I \times J$ fluxos nos centros dos volumes (Figura 3.3(b)).

A variável ψ_V é discretizada em $i = [1, \dots, I + 1], j = [1, \dots, J], m = [1, \dots, M], g = [1, \dots, G]$ de modo a representar os I fluxos nas faces esquerdas dos volumes e o fluxo da face direita do último volume (Figura 3.3(c)).

A variável ψ_H é discretizada em $i = [1, \dots, I], j = [1, \dots, J + 1], m = [1, \dots, M], g = [1, \dots, G]$ de modo a representar os J fluxos nas faces inferiores dos volumes e o fluxo da face superior do último volume (Figura 3.3(d)).

A equação 3.30 para o primeiro quadrante adota $i = [1, \dots, I], j = [1, \dots, J], m = [1, \dots, M/4], g = [1, \dots, G]$, transformando-se na equação 3.32 (Figura 3.4(a)):

$$\frac{\mu_m}{h_{xi}}(\psi_{V,mg}^{i+1,j} - \psi_{V,mg}^{i,j}) + \frac{\eta_m}{h_{yj}}(\psi_{H,mg}^{i,j+1} - \psi_{H,mg}^{i,j}) + \sum_{t,g}^{ij} \psi_{mg}^{ij} = \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij} \tag{3.32}$$

A equação 3.30 para o segundo quadrante adota $i = [I, \dots, 1], j = [1, \dots, J], m = [M/4, \dots, 2M/4]$ e $g = [1, \dots, G]$, transformando-se na equação 3.33 (Figura 3.4(b)):

$$\frac{\mu_m}{h_{xi}}(\psi_{V,mg}^{i,j} - \psi_{V,mg}^{i+1,j}) + \frac{\eta_m}{h_{yj}}(\psi_{H,mg}^{i,j+1} - \psi_{H,mg}^{i,j}) + \sum_{t,g}^{ij} \psi_{mg}^{ij} = \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij} \tag{3.33}$$

A equação 3.30 para o terceiro quadrante adota $i = [I, \dots, 1]$, $j = [J, \dots, 1]$, $m = [2M/4, \dots, 3M/4]$ e $g = [1, \dots, G]$, transformando-se na equação 3.34 (Figura 3.4(c)):

$$\frac{\mu_m}{h_{xi}}(\psi_{V,mg}^{i,j} - \psi_{V,mg}^{i+1,j}) + \frac{\eta_m}{h_{yj}}(\psi_{H,mg}^{i,j} - \psi_{H,mg}^{i,j+1}) + \Sigma_{t,g}^{ij} \psi_{mg}^{ij} = \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij} \quad (3.34)$$

A equação 3.30 para o quarto quadrante adota $i = [1, \dots, I]$, $j = [J, \dots, 1]$, $m = [3M/4, \dots, M]$, $g = [1, \dots, G]$, transformando-se na equação 3.35 (Figura 3.4(d)):

$$\frac{\mu_m}{h_{xi}}(\psi_{V,mg}^{i+1,j} - \psi_{V,mg}^{i,j}) + \frac{\eta_m}{h_{yj}}(\psi_{H,mg}^{i,j} - \psi_{H,mg}^{i,j+1}) + \Sigma_{t,g}^{ij} \psi_{mg}^{ij} = \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij} \quad (3.35)$$

Nas próximas seções são descritos os passos para a resolução das quatro equações de varredura (3.32, 3.33, 3.34 e 3.35) através de um método iterativo apropriado para cada quadrante.

3.1.4 Equação Discreta Explícita

Ao considerar uma célula particular (m, g, i, j) e assumindo que os fluxos incidentes ψ_V^{in} e ψ_H^{in} na fronteira do domínio são conhecidos, tem-se:

- Uma equação conhecida: Equação 3.30.
- Três incógnitas a determinar: ψ , ψ_V^{out} e ψ_H^{out} .

Para completar o sistema de equações, faz-se necessário encontrar as duas outras equações. Neste trabalho adota-se o método DD (*Diamond Difference*) para completar o sistema e permitir a sua resolução por um método iterativo.

3.1.4.1 Aproximação por *Diamond Difference*

A aproximação por *Diamond Difference* prescreve duas equações auxiliares que aproximam os fluxos centrais das células como a média dos fluxos nas faces verticais e horizontais. Considerando os fluxos emergentes no primeiro quadrante 3.30 escrevem-se os fluxos centrais para cada célula (i, j) como:

$$\psi_{mg}^{ij} = \frac{1}{2}(\psi_{V,mg}^{ij} + \psi_{V,mg}^{i+1,j}) \quad (3.36)$$

$$\psi_{mg}^{ij} = \frac{1}{2}(\psi_{H,mg}^{ij} + \psi_{H,mg}^{i,j+1}) \quad (3.37)$$

A partir destas relações podem-se escrever 3 equações explícitas para as 3 incógnitas. As equações 3.38 e 3.39 expressam os fluxos emergentes nas faces vertical e horizontal:

$$\psi_{V_{mg}}^{i+1,j} = 2\psi_{mg}^{ij} - \psi_{V_{mg}}^{ij} \quad (3.38)$$

$$\psi_{H_{mg}}^{i,j+1} = 2\psi_{mg}^{ij} - \psi_{H_{mg}}^{ij} \quad (3.39)$$

A partir das 3 equações (3.30, 3.38 e 3.39) tem-se um sistema fechado. Substituindo 3.38 e 3.39 em 3.30 obtém-se a equação 3.40:

$$\begin{aligned} \frac{\mu_m}{h_{xi}}((2\psi_{mg}^{ij} - \psi_{V_{mg}}^{ij}) - \psi_{V_{mg}}^{ij}) + \\ \frac{\eta_m}{h_{yj}}((2\psi_{mg}^{ij} - \psi_{H_{mg}}^{ij}) - \psi_{H_{mg}}^{ij}) + \\ \sum_{t,g}^{ij} \psi_{mg}^{ij} = \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij} \end{aligned} \quad (3.40)$$

A equação 3.40 possui apenas uma incógnita ψ_{mg}^{ij} e simplificando tem-se 3.41:

$$\frac{\mu_m}{h_{xi}}(2\psi_{mg}^{ij} - 2\psi_{V_{mg}}^{ij}) + \frac{\eta_m}{h_{yj}}(2\psi_{mg}^{ij} - 2\psi_{H_{mg}}^{ij}) + \sum_{t,g}^{ij} \psi_{mg}^{ij} = \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij} \quad (3.41)$$

Usando os coeficientes $\epsilon_x = 2\mu_m/h_{xi}$ e $\epsilon_y = 2\eta_m/h_{yj}$ obtém-se 3.42:

$$\epsilon_x \psi_{mg}^{ij} - \epsilon_x \psi_{V_{mg}}^{ij} + \epsilon_y \psi_{mg}^{ij} - \epsilon_y \psi_{H_{mg}}^{ij} + \sum_{t,g}^{ij} \psi_{mg}^{ij} = \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij} \quad (3.42)$$

Resolvendo para ψ_{mg}^{ij} obtém-se a forma explícita da equação 3.30:

$$\psi_{mg}^{ij} = (\epsilon_x \psi_{V_{mg}}^{ij} + \epsilon_y \psi_{H_{mg}}^{ij} + \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij}) / (\epsilon_x + \epsilon_y + \sum_{t,g}^{ij}) \quad (3.43)$$

que expressa o fluxo central da célula (i, j) em termos dos fluxos incidentes nas faces ($\psi_{V_{mg}}^{ij}$ e $\psi_{H_{mg}}^{ij}$) e das fontes de espalhamento $\psi_{s,mg}^{ij}$ e externa $\psi_{ext,mg}^{ij}$. As equações 3.43, 3.38 e 3.39 são avaliadas na ordem de $i = [1..N_x]$ e $j = [1..N_y]$ de modo a se obter todos os fluxos centrais e emergentes em cada célula do quadrante 1.

3.2 Algoritmos

Nesta seção são apresentados os algoritmos que resolvem computacionalmente os processos abordados por este trabalho.

3.2.1 Transporte de Nêutrons

Nesta seção são apresentados os algoritmos que resolvem numericamente a equação de transporte de nêutrons. O fluxograma do algoritmo de transporte está representado na Figura 3.5 que ilustra os passos principais da simulação de transporte de nêutrons. Nas demais seções são apresentados os algoritmos de cada etapa do processo principal.

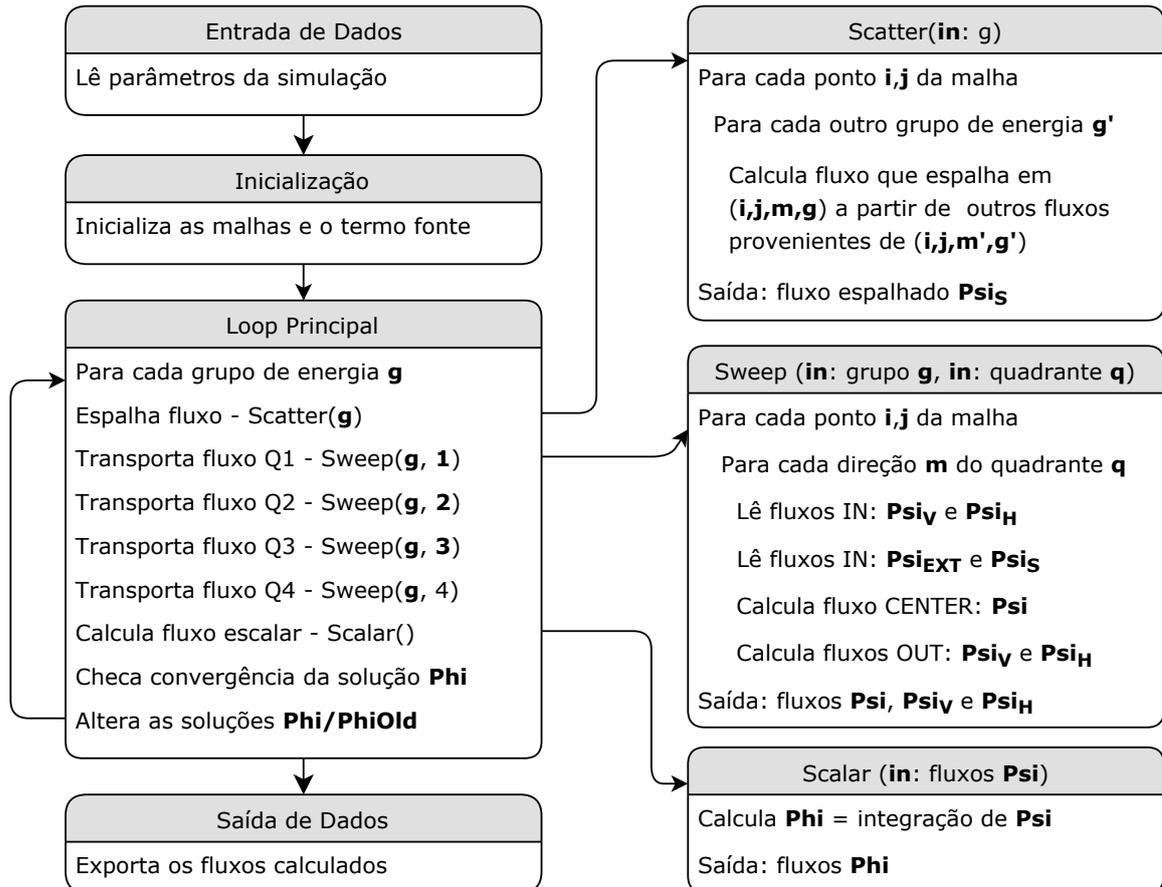


Figura 3.5: Fluxograma do algoritmo de transporte

3.2.1.1 Inicialização dos Parâmetros de Quadratura S_N

No Algoritmo 1, inicializam-se os vetores μ , η e ω de modo a armazenar as M componentes das direções e pesos pré-definidos pela quadratura do método S_N utilizada. A geração dos valores das ordenadas está contida na listagem E.4. O algoritmo recebe como entrada o valor de N (ordem da quadratura) proveniente do arquivo de entrada da simulação e prossegue com o fluxo correspondente, executando um dos quatro algoritmos de inicialização de ordenadas: Algoritmo 2 para $N=4$, Algoritmo 3 para $N=6$, Algoritmo 4

para $N=8$ ou Algoritmo 5 para $N=16$.

Algorithm 1: Init S_N Quadrature

In: $N \in \mathbb{N}$: Ordem de Quadratura (4, 6, 8 e 16)

Out: $\mu_m, \eta_m, \omega_m \in \mathbb{R}^M$: Ordenadas discretas e pesos ω_m da quadratura S_N

```

1 if  $N = 4$  then
2   └─ Init  $S_4$  Quadrature()
3 else if  $N = 6$  then
4   └─ Init  $S_6$  Quadrature()
5 else if  $N = 8$  then
6   └─ Init  $S_8$  Quadrature()
7 else if  $N = 16$  then
8   └─ Init  $S_{16}$  Quadrature()

```

3.2.1.2 Inicialização dos Parâmetros de Quadratura S_4

No Algoritmo 2, inicializam-se as 12 ordenadas discretas da quadratura angular S_4 [25]. A listagem E.4 contém o código MATLAB que gera as ordenadas através da chamada da função `getOrdinates` ($\mu_1 = 0.3500212$, $N = 4$) onde μ_1 é a ordenada de referência obtida da tabela 4.1 de [25]. Nas linhas de 3 a 5 inicializam-se as direções do primeiro quadrante e nas linhas 6, 7 e 8 as direções do segundo, terceiro e quarto quadrantes respectivamente.

Algorithm 2: Init S_4 Quadrature

Out: $\mu_m, \eta_m, \omega_m \in \mathbb{R}^M$: Ordenadas discretas e pesos ω_m da quadratura S_4

```

1  $N = 4$ 
2  $M = N(N + 2)/2$                                 ▷ 12 direções
3  $\mu_1 = 0.3500212$      $\eta_1 = 0.3500212$      $\omega_1 = 0.3333333$     ▷ Quadrante 1
4  $\mu_2 = 0.3500212$      $\eta_2 = 0.8688903$      $\omega_2 = 0.3333333$     ▷ Quadrante 1
5  $\mu_3 = 0.8688903$      $\eta_3 = 0.3500212$      $\omega_3 = 0.3333333$     ▷ Quadrante 1
6  $\mu_{4:6} = -\mu_{1:3}$      $\eta_{4:6} = \eta_{1:3}$                                 ▷ Quadrante 2
7  $\mu_{7:9} = -\mu_{1:3}$      $\eta_{7:9} = -\eta_{1:3}$                                 ▷ Quadrante 3
8  $\mu_{10:12} = \mu_{1:3}$      $\eta_{10:12} = -\eta_{1:3}$                                 ▷ Quadrante 4

```

3.2.1.3 Inicialização dos Parâmetros de Quadratura S_6

No Algoritmo 3, inicializam-se as 24 ordenadas discretas da quadratura angular S_6 [25]. A listagem E.4 contém o código MATLAB que gera as ordenadas através da chamada da função `getOrdinates` ($\mu_1 = 0.2666355$, $N = 6$) onde μ_1 é a ordenada de referência obtida da tabela 4.1 de [25]. Nas linhas de 3 a 8 inicializam-se as direções do primeiro quadrante e nas linhas 9, 10 e 11 as direções do segundo, terceiro e quarto quadrantes respectivamente.

Algorithm 3: Init S_6 Quadrature

Out: $\mu_m, \eta_m, \omega_m \in \mathbb{R}^M$: Ordenadas discretas e pesos ω_m da quadratura S_6

1	$N = 6$		
2	$M = N(N + 2)/2$		▷ 24 direções
3	$\mu_1 = 0.2666355$	$\eta_1 = 0.2666355$	$\omega_1 = 0.1761263$ ▷ Quadrante 1
4	$\mu_2 = 0.2666355$	$\eta_2 = 0.9261808$	$\omega_2 = 0.1761263$ ▷ Quadrante 1
5	$\mu_3 = 0.9261808$	$\eta_3 = 0.2666355$	$\omega_3 = 0.1761263$ ▷ Quadrante 1
6	$\mu_4 = 0.2666355$	$\eta_4 = 0.6815076$	$\omega_4 = 0.1572071$ ▷ Quadrante 1
7	$\mu_5 = 0.6815076$	$\eta_5 = 0.2666355$	$\omega_5 = 0.1572071$ ▷ Quadrante 1
8	$\mu_6 = 0.6815076$	$\eta_6 = 0.6815076$	$\omega_6 = 0.1572071$ ▷ Quadrante 1
9	$\mu_{7:12} = -\mu_{1:6}$	$\eta_{7:12} = \eta_{1:6}$	▷ Quadrante 2
10	$\mu_{13:18} = -\mu_{1:6}$	$\eta_{13:18} = -\eta_{1:6}$	▷ Quadrante 3
11	$\mu_{19:24} = \mu_{1:6}$	$\eta_{19:24} = -\eta_{1:6}$	▷ Quadrante 4

3.2.1.4 Inicialização dos Parâmetros de Quadratura S_8

No Algoritmo 4, inicializam-se as 40 ordenadas discretas da quadratura angular S_8 [25]. A listagem E.4 contém o código MATLAB que gera as ordenadas através da chamada da função `getOrdinates` ($\mu_1 = 0.2182179$, $N = 8$) onde μ_1 é a ordenada de referência obtida da tabela 4.1 de [25]. Nas linhas de 3 a 12 inicializam-se as direções do primeiro quadrante e nas linhas 13, 14 e 15 as direções do segundo, terceiro e quarto quadrantes

respectivamente.

Algorithm 4: Init S_8 Quadrature

Out: $\mu_m, \eta_m, \omega_m \in \mathbb{R}^M$: Ordenadas discretas e pesos ω_m da quadratura S_8

1	$N = 8$		
2	$M = N(N + 2)/2$		▷ 40 direções
3	$\mu_1 = 0.2182179$	$\eta_1 = 0.2182179$	$\omega_1 = 0.1209877$ ▷ Quadrante 1
4	$\mu_2 = 0.2182179$	$\eta_2 = 0.9511897$	$\omega_2 = 0.1209877$ ▷ Quadrante 1
5	$\mu_3 = 0.9511897$	$\eta_3 = 0.2182179$	$\omega_3 = 0.1209877$ ▷ Quadrante 1
6	$\mu_4 = 0.2182179$	$\eta_4 = 0.5773503$	$\omega_4 = 0.0907407$ ▷ Quadrante 1
7	$\mu_5 = 0.5773503$	$\eta_5 = 0.2182179$	$\omega_5 = 0.0907407$ ▷ Quadrante 1
8	$\mu_6 = 0.2182179$	$\eta_6 = 0.7867958$	$\omega_6 = 0.0907407$ ▷ Quadrante 1
9	$\mu_7 = 0.7867958$	$\eta_7 = 0.2182179$	$\omega_7 = 0.0907407$ ▷ Quadrante 1
10	$\mu_8 = 0.5773503$	$\eta_8 = 0.7867958$	$\omega_8 = 0.0907407$ ▷ Quadrante 1
11	$\mu_9 = 0.7867958$	$\eta_9 = 0.5773503$	$\omega_9 = 0.0907407$ ▷ Quadrante 1
12	$\mu_{10} = 0.5773503$	$\eta_{10} = 0.5773503$	$\omega_{10} = 0.0925926$ ▷ Quadrante 1
13	$\mu_{11:20} = -\mu_{1:10}$	$\eta_{11:20} = \eta_{1:10}$	▷ Quadrante 2
14	$\mu_{21:30} = -\mu_{1:10}$	$\eta_{21:30} = -\eta_{1:10}$	▷ Quadrante 3
15	$\mu_{31:40} = \mu_{1:10}$	$\eta_{31:40} = -\eta_{1:10}$	▷ Quadrante 4

3.2.1.5 Inicialização dos Parâmetros de Quadratura S_{16}

No Algoritmo 5, inicializam-se as 144 ordenadas discretas da quadratura angular S_{16} [25]. A listagem E.4 contém o código MATLAB que gera as ordenadas através da chamada da função `getOrdinates`($\mu_1 = 0.1389568$, $N = 16$) onde μ_1 é a ordenada de referência obtida da tabela 4.1 de [25]. Nas linhas de 3 a 38 inicializam-se as direções do primeiro quadrante e nas linhas 39, 40 e 41 as direções do segundo, terceiro e quarto quadrantes

respectivamente.

Algorithm 5: Init S_{16} Quadrature

Out: $\mu_m, \eta_m, \omega_m \in \mathbb{R}^M$: Ordenadas discretas e pesos ω_m da quadratura S_{16}

1	$N = 16$			
2	$M = N(N + 2)/2$			▷ 144 direções
3	$\mu_1 = 0.1389568$	$\eta_1 = 0.1389568$	$\omega_1 = 0.0489872$	▷ Quadrante 1
4	$\mu_2 = 0.1389568$	$\eta_2 = 0.3922892$	$\omega_2 = 0.0413296$	▷ Quadrante 1
5	$\mu_3 = 0.3922892$	$\eta_3 = 0.1389568$	$\omega_3 = 0.0413296$	▷ Quadrante 1
6	$\mu_4 = 0.1389568$	$\eta_4 = 0.5370966$	$\omega_4 = 0.0212326$	▷ Quadrante 1
7	$\mu_5 = 0.3922892$	$\eta_5 = 0.3922892$	$\omega_5 = 0.0360486$	▷ Quadrante 1
8	$\mu_6 = 0.5370966$	$\eta_6 = 0.1389568$	$\omega_6 = 0.0212326$	▷ Quadrante 1
9	$\mu_7 = 0.1389568$	$\eta_7 = 0.6504265$	$\omega_7 = 0.0256207$	▷ Quadrante 1
10	$\mu_8 = 0.3922892$	$\eta_8 = 0.5370966$	$\omega_8 = 0.0144589$	▷ Quadrante 1
11	$\mu_9 = 0.5370966$	$\eta_9 = 0.3922892$	$\omega_9 = 0.0144589$	▷ Quadrante 1
12	$\mu_{10} = 0.6504265$	$\eta_{10} = 0.1389568$	$\omega_{10} = 0.0256207$	▷ Quadrante 1
13	$\mu_{11} = 0.1389568$	$\eta_{11} = 0.7467506$	$\omega_{11} = 0.0256207$	▷ Quadrante 1
14	$\mu_{12} = 0.3922892$	$\eta_{12} = 0.6504265$	$\omega_{12} = 0.0344958$	▷ Quadrante 1
15	$\mu_{13} = 0.5370966$	$\eta_{13} = 0.5370966$	$\omega_{13} = 0.0085179$	▷ Quadrante 1
16	$\mu_{14} = 0.6504265$	$\eta_{14} = 0.3922892$	$\omega_{14} = 0.0344958$	▷ Quadrante 1
17	$\mu_{15} = 0.7467506$	$\eta_{15} = 0.1389568$	$\omega_{15} = 0.0256207$	▷ Quadrante 1
18	$\mu_{16} = 0.1389568$	$\eta_{16} = 0.8319966$	$\omega_{16} = 0.0212326$	▷ Quadrante 1
19	$\mu_{17} = 0.3922892$	$\eta_{17} = 0.7467506$	$\omega_{17} = 0.0144589$	▷ Quadrante 1
20	$\mu_{18} = 0.5370966$	$\eta_{18} = 0.6504265$	$\omega_{18} = 0.0085179$	▷ Quadrante 1
21	$\mu_{19} = 0.6504265$	$\eta_{19} = 0.5370966$	$\omega_{19} = 0.0085179$	▷ Quadrante 1
22	$\mu_{20} = 0.7467506$	$\eta_{20} = 0.3922892$	$\omega_{20} = 0.0144589$	▷ Quadrante 1
23	$\mu_{21} = 0.8319966$	$\eta_{21} = 0.1389568$	$\omega_{21} = 0.0212326$	▷ Quadrante 1
24	$\mu_{22} = 0.1389568$	$\eta_{22} = 0.9092855$	$\omega_{22} = 0.0413296$	▷ Quadrante 1
25	$\mu_{23} = 0.3922892$	$\eta_{23} = 0.8319966$	$\omega_{23} = 0.0360486$	▷ Quadrante 1
26	$\mu_{24} = 0.5370966$	$\eta_{24} = 0.7467506$	$\omega_{24} = 0.0144589$	▷ Quadrante 1
27	$\mu_{25} = 0.6504265$	$\eta_{25} = 0.6504265$	$\omega_{25} = 0.0344958$	▷ Quadrante 1

28	$\mu_{26} = 0.7467506$	$\eta_{26} = 0.5370966$	$\omega_{26} = 0.0144589$	▷ Quadrante 1
29	$\mu_{27} = 0.8319966$	$\eta_{27} = 0.3922892$	$\omega_{27} = 0.0360486$	▷ Quadrante 1
30	$\mu_{28} = 0.9092855$	$\eta_{28} = 0.1389568$	$\omega_{28} = 0.0413296$	▷ Quadrante 1
31	$\mu_{29} = 0.1389568$	$\eta_{29} = 0.9805009$	$\omega_{29} = 0.0489872$	▷ Quadrante 1
32	$\mu_{30} = 0.3922892$	$\eta_{30} = 0.9092855$	$\omega_{30} = 0.0413296$	▷ Quadrante 1
33	$\mu_{31} = 0.5370966$	$\eta_{31} = 0.8319966$	$\omega_{31} = 0.0212326$	▷ Quadrante 1
34	$\mu_{32} = 0.6504265$	$\eta_{32} = 0.7467506$	$\omega_{32} = 0.0256207$	▷ Quadrante 1
35	$\mu_{33} = 0.7467506$	$\eta_{33} = 0.6504265$	$\omega_{33} = 0.0256207$	▷ Quadrante 1
36	$\mu_{34} = 0.8319966$	$\eta_{34} = 0.5370966$	$\omega_{34} = 0.0212326$	▷ Quadrante 1
37	$\mu_{35} = 0.9092855$	$\eta_{35} = 0.3922892$	$\omega_{35} = 0.0413296$	▷ Quadrante 1
38	$\mu_{36} = 0.9805009$	$\eta_{36} = 0.1389568$	$\omega_{36} = 0.0489872$	▷ Quadrante 1
39	$\mu_{37:72} = -\mu_{1:36}$	$\eta_{37:72} = \eta_{1:36}$		▷ Quadrante 2
40	$\mu_{73:108} = -\mu_{1:36}$	$\eta_{73:108} = -\eta_{1:36}$		▷ Quadrante 3
41	$\mu_{109:144} = \mu_{1:36}$	$\eta_{109:144} = -\eta_{1:36}$		▷ Quadrante 4

3.2.1.6 Inicialização da Malha de Tipos de Materiais

No Algoritmo 6, inicializa-se a malha de tipos de materiais T^{ij} com os valores provenientes do arquivo de entrada da simulação, contendo os tipos 1 para sadio e 2 para tumor. A matriz T é usada para identificar qual o tipo de material que se encontra na posição (i, j) da malha espacial. Posteriormente, esta informação é utilizada para se inicializar a seção de choque total $\Sigma_{t,g}^{ij}$ usada no Algoritmo 8.

Algorithm 6: Init Materials

In: $ArqTipoMaterial \in \mathbb{R}^{I \times J}$: Arquivo de tipos de materiais da malha

Out: $T^{ij} \in \mathbb{R}^{I \times J}$: Tipo de material (1 ou 2) nos pontos da malha

1 $T(1 : I, 1 : J) = \text{LêArquivo}('ArqTipoMaterial')$ ▷ 1 (sadio) ou 2 (tumor)

3.2.1.7 Inicialização das Seções de Choque

No Algoritmo 7, inicializam-se as matrizes de seções de choque total Σ_t e espalhamento Σ_{s0} , Σ_{s1} , Σ_{s2} e Σ_{s3} a partir dos arquivos de entrada, obtidos da referência [12].

Na linha 1, lê-se a matriz $G \times 2$ de seção de choque total Σ_t , preenchendo a primeira coluna com G valores distintos de seção de choque para o tipo material 1 (Sadio) e a segunda coluna com G valores distintos de seção de choque para o tipo material 2 (Tumor).

Os valores são utilizados para inicializar a seção de choque total $\Sigma_{t,g}^{ij}$ usada no Algoritmo 8. Na linha 2, lê-se a matriz $G \times 2$ de seção de choque de absorção Σ_a , preenchendo a primeira coluna com G zeros para o tipo material 1 (Sadio) e a segunda coluna com G valores distintos de seção de choque para o tipo material 2 (Tumor). Os valores são utilizados no cálculo da dose pelo no Algoritmo 18. Nas linhas 3, 4, 5 e 6 são lidas as matrizes $G \times G$ de momento de seção de choque de ordem 0, 1, 2 e 3 respectivamente. Essas matrizes contém as distintas quantidades espalhadas em cada direção de modo a permitir a reconstrução do espalhamento anisotrópico observado em laboratório. As matrizes do tecido sadio são armazenadas nas posições $\Sigma_{s0}(:, :, 1)$, $\Sigma_{s1}(:, :, 1)$, $\Sigma_{s2}(:, :, 1)$ e $\Sigma_{s3}(:, :, 1)$ e ilustradas na Figura 3.10. As matrizes do tecido tumoral são armazenadas nas posições $\Sigma_{s0}(:, :, 2)$, $\Sigma_{s1}(:, :, 2)$, $\Sigma_{s2}(:, :, 2)$ e $\Sigma_{s3}(:, :, 2)$. e ilustradas nas Figura 3.11. Tais matrizes são usadas pelo Algoritmo 10 para inicializar as parcelas da expansão de polinômios de Legendre.

Algorithm 7: Init Cross Sections

In: $Arq\Sigma_t \in \mathbb{R}^{G \times 2}$: Arquivo da Seção de Choque Total
In: $Arq\Sigma_{s0}, Arq\Sigma_{s1}, Arq\Sigma_{s2}, Arq\Sigma_{s3} \in \mathbb{R}^{G \times G \times 2}$: Arquivos de Seções de Choque
Out: $\Sigma_t \in \mathbb{R}^{G \times 2}$: Seção de Choque Total para G energias e 2 materiais
Out: $\Sigma_a \in \mathbb{R}^{G \times 2}$: Seção de Choque de Absorção para G energias e 2 materiais
Out: $\Sigma_{s0}, \Sigma_{s1}, \Sigma_{s2}, \Sigma_{s3} \in \mathbb{R}^{G \times G \times 2}$: Seções de Choque de Espalhamento

▷ Lê a seção de choque total para cada grupo e tipo material

1 $\Sigma_t(1 : G, 1 : 2) = \text{LêArquivosCSV}(\text{"}\Sigma_t \text{ Sadio"}, \text{"}\Sigma_t \text{ Tumor"})$

▷ Lê a seção de choque de absorção para cada grupo e tipo material

2 $\Sigma_a(1 : G, 1 : 2) = \text{LêArquivosCSV}(\text{"}\Sigma_a \text{ Sadio"}, \text{"}\Sigma_a \text{ Tumor"})$

▷ Lê o momento de seção de choque de espalhamento de grau 0 (Isotrópico)

3 $\Sigma_{s0}(1 : G, 1 : G, 1 : 2) = \text{LêArquivosCSV}(\text{"}\Sigma_{s0} \text{ Sadio"}, \text{"}\Sigma_{s0} \text{ Tumor"})$

▷ Lê os momentos de seção de choque de espalhamento de graus 1,2,3

4 $\Sigma_{s1}(1 : G, 1 : G, 1 : 2) = \text{LêArquivosCSV}(\text{"}\Sigma_{s1} \text{ Sadio"}, \text{"}\Sigma_{s1} \text{ Tumor"})$

5 $\Sigma_{s2}(1 : G, 1 : G, 1 : 2) = \text{LêArquivosCSV}(\text{"}\Sigma_{s2} \text{ Sadio"}, \text{"}\Sigma_{s2} \text{ Tumor"})$

6 $\Sigma_{s3}(1 : G, 1 : G, 1 : 2) = \text{LêArquivosCSV}(\text{"}\Sigma_{s3} \text{ Sadio"}, \text{"}\Sigma_{s3} \text{ Tumor"})$

3.2.1.8 Inicialização da Malha de Seções de Choque Total

No Algoritmo 8, inicializa-se a malha de seções de choque $\Sigma_{t,g}^{ij}$ a partir da malha de tipos de materiais T^{ij} (definida pelo Algoritmo 6) e da seção de choque Σ_t (definida pelo Algoritmo 7) conhecida para cada tipo e grupo de energia. Nas linhas 1 e 2, realiza-se um

loop ao longo de todos os grupos e posições da malha espacial. Na linha 3 lê-se o tipo de material t encontrado na posição (i, j) da malha a partir da matriz T criada no Algoritmo 6. Na linha 4 lê-se a seção de choque $\Sigma_t(g, t)$ para o grupo g e tipo t para preencher a seção de choque $\Sigma_{t,g}^{ij}$ em cada ponto (i, j) e grupo g . Os valores de $\Sigma_{t,g}^{ij}$ são usados pelo algoritmo de varredura 14 no cálculo das quantidades espalhadas nas direções do plano cartesiano.

Algorithm 8: Init Total Cross Section

In: $T^{ij} \in \mathbb{R}^{I \times J}$: Tipo de material em cada ponto da malha
In: $\Sigma_t \in \mathbb{R}^{G \times 2}$: Seção de Choque Total para G energias e 2 materiais
Out: $\Sigma_{t,g}^{ij} \in \mathbb{R}^{G \times I \times J}$: Seção de Choque Total em cada ponto e grupo

```

1 for  $g = 1 : G$  do
2   for  $(i, j) \in [1 : I] \times [1 : J]$  do
3      $t = T^{ij}$  ▷ 1 (sadio) ou 2 (tumor)
4      $\Sigma_{t,g}^{ij} = \Sigma_t(g, t)$  ▷ Seção de Choque para o tipo  $t$  e grupo  $G$ 

```

3.2.1.9 Inicialização dos Fluxos da Fonte Externa

No Algoritmo 9, inicializam-se os fluxos externos ψ_{ext} no contorno do domínio com valores positivos somente para as direções que apontam para o interior do volume. A Figura 3.6 ilustra os pontos do contorno em azul $\partial\mathcal{D}$ e os pontos do feixe em vermelho \mathcal{D}_{feixe} . No exemplo ilustrado, o feixe se localiza na face esquerda do domínio, e os fluxos angulares apontam para a direita. Na linha 1, inicializa-se a variável ψ_{ext} com zeros para todas as posições (i, j) , direções m e grupo g . Na linha 2, realiza-se um *loop* ao longo de todos os grupos de energia. Na linha 3, visitam-se as posições (i, j) do contorno (região quadrada em azul). Na linha 4, verifica-se a posição pertença ao feixe incidente (pontos em vermelho) e em caso positivo, prossegue-se na linha 5 com a travessia pelas M direções de fluxo. Na linha 6, verifica-se o apontamento do fluxo angular para dentro do domínio através da expressão $(\hat{\Omega}_m \cdot \hat{d} > 0) = (\mu_m \cdot d_x + \eta_m \cdot d_y) > 0$ que para a Figura 3.6 define $\hat{d} = (1, 0)$ para excluir as direções contendo $\mu_m < 0$ e incluir as contendo $\mu_m > 0$. Finalmente na linha 7, define-se o valor não nulo da fonte externa ψ_{ext} .

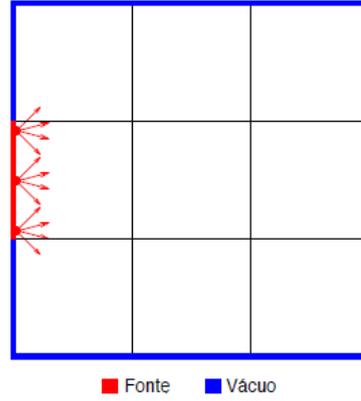


Figura 3.6: Fonte externa inicializada no contorno do domínio

Algorithm 9: Init External Source 2D

In: $\mathcal{D} \in \mathbb{R}^{I \times J}$: Domínio espacial

In: $\mathcal{D}_{feixe} : (i, j) \in \mathbb{R}^{I \times J}$: Feixe de nêutrons do contorno

In: $\hat{d} \in \mathbb{R}^2$: Direção fixa do Feixe de Nêutrons

In: ψ_0 : Intensidade do Feixe para cada grupo

Out: $\psi_{ext} \in \mathbb{R}^{M \times G \times I \times J}$: Fluxo angular externo

```

1  $\psi_{ext,mg}^{i,j} = 0$  para todo  $m, g, i, j$                                 ▷ inicializa com fluxos nulos
2 for  $g = 1 : G$  do                                                    ▷ para cada grupo de energia
3   for  $(i, j) \in \partial\mathcal{D}$  do                                          ▷ para cada posição do contorno
4     if  $(i, j) \in \mathcal{D}_{feixe}$  then                                  ▷ se o ponto é do feixe
5       for  $m = 1 : M$  do                                              ▷ para cada direção
6         if  $\hat{\Omega}_m \cdot \hat{d} > 0$  then                               ▷ que aponta para dentro do domínio
7            $\psi_{ext,mg}^{i,j} = \psi_0$                                 ▷ define fonte fixa

```

3.2.1.10 Inicialização da Lista de Seções de Choque de Espalhamento

Na expansão de polinômios de Legendre pelo método P_3 (equação 3.11) pode-se identificar a lista $\Sigma'_{s,p}$ das 13 parcelas dos momentos de seção de choque espalhado, com $0 \leq p \leq 12$. No Algoritmo 10, inicializa-se a lista de seções de choque $\Sigma'_{s,p}$ a partir dos momentos distintos Σ_{s0} , Σ_{s1} , Σ_{s2} e Σ_{s3} criados no Algoritmo 7 de modo a obter $\Sigma'_s = (\Sigma_{s0}, \Sigma_{s1}, \Sigma_{s1}, \Sigma_{s2}, \Sigma_{s2}, \Sigma_{s2}, \Sigma_{s2}, \Sigma_{s3}, \Sigma_{s3}, \Sigma_{s3}, \Sigma_{s3}, \Sigma_{s3})$. Os valores de Σ'_s são

usados no Algoritmo 13 para cálculo do fluxo angular espalhado ψ_s da equação 3.43.

Algorithm 10: Init Scatter 2D Cross Sections

In: $\Sigma_{s0}, \Sigma_{s1}, \Sigma_{s2}, \Sigma_{s3} \in \mathbb{R}^{G \times G}$: Momentos das seções de choque do método P_3

Out: $\Sigma'_{s,p} \in \mathbb{R}^{13 \times G \times G}$: Lista das Seções de Choque de Espalhamento

- 1 $\Sigma'_{s,0} = \Sigma_{s0}$ ▷ Ordem 0
 - 2 $\Sigma'_{s,1:2} = \Sigma_{s1}$ ▷ Ordem 1
 - 3 $\Sigma'_{s,3:6} = \Sigma_{s2}$ ▷ Ordem 2
 - 4 $\Sigma'_{s,7:12} = \Sigma_{s3}$ ▷ Ordem 3
-

3.2.1.11 Inicialização dos Coeficientes Internos de Espalhamento

Na expansão de polinômios de Legendre pelo método P_3 (equação 3.11) pode-se identificar a lista $C_{p,m}$ dos $13 \times M$ coeficientes internos aos somatórios. A obtenção dos coeficientes está detalhada no apêndice A pelas equações A.18, A.26, A.33 e A.40. O Algoritmo 11 pré-calcula os coeficientes internos aos somatórios a partir das ordenadas discretas μ_n, η_n e pesos ω_n da quadratura criados pelo Algoritmo 1. Os valores de $C_{p,m}$ são usados no Algoritmo 13 para cálculo do fluxo angular espalhado ψ_s da equação 3.43.

Algorithm 11: Init Scatter 2D Internal Coefficients

In: $\mu_n, \eta_n, \omega_n \in \mathbb{R}^M$: Ordenadas discretas e pesos da quadratura S_N

Out: $C_{p,n} \in \mathbb{R}^{13 \times M}$: Coeficientes internos dos momentos de fluxo

- 1 **for** $n = 1 : M$ **do**
 - 2 $C_{0,n} = \omega_n$ ▷ Ordem 0
 - 3 $C_{1,n} = \mu_n \omega_n$ ▷ Ordem 1
 - 4 $C_{2,n} = \eta_n \omega_n$ ▷ Ordem 1
 - 5 $C_{3,n} = \omega_n$ ▷ Ordem 2
 - 6 $C_{4,n} = \mu_n \eta_n \omega_n$ ▷ Ordem 2
 - 7 $C_{5,n} = \mu_n^2 \omega_n$ ▷ Ordem 2
 - 8 $C_{6,n} = \eta_n^2 \omega_n$ ▷ Ordem 2
 - 9 $C_{7,n} = \mu_n \omega_n$ ▷ Ordem 3
 - 10 $C_{8,n} = \eta_n \omega_n$ ▷ Ordem 3
 - 11 $C_{9,n} = \mu_n^2 \eta_n \omega_n$ ▷ Ordem 3
 - 12 $C_{10,n} = \mu_n \eta_n^2 \omega_n$ ▷ Ordem 3
 - 13 $C_{11,n} = \mu_n^3 \omega_n$ ▷ Ordem 3
 - 14 $C_{12,n} = \eta_n^3 \omega_n$ ▷ Ordem 3
-

3.2.1.12 Inicialização dos Coeficientes Externos de Espalhamento

Na expansão de polinômios de Legendre pelo método P_3 (equação 3.11) pode-se identificar a lista $K_{p,m}$ dos $13 \times M$ coeficientes externos aos somatórios. A obtenção dos coeficientes está detalhada no apêndice A pelas equações A.18, A.26, A.33 e A.40. O Algoritmo 11 pré-calcula os coeficientes internos aos somatórios a partir das ordenadas discretas μ_m e η_m da quadratura criados pelo Algoritmo 1. Os valores de $K_{p,m}$ são usados no Algoritmo 13 para cálculo do fluxo angular espalhado ψ_s da equação 3.43.

Algorithm 12: Init Scatter 2D External Coefficients

In: $\mu_m, \eta_m \in \mathbb{R}^M$: Ordenadas discretas e pesos da quadratura S_N

Out: $K_{p,m} \in \mathbb{R}^{13 \times M}$: Coeficientes externos dos momentos de fluxo

1 **for** $m = 1 : M$ **do**

2	$K_{0,m} = 0.25$	▷ Ordem 0
3	$K_{1,m} = 0.75 \mu_m$	▷ Ordem 1
4	$K_{2,m} = 0.75 \eta_m$	▷ Ordem 1
5	$K_{3,m} = 1.25 - 1.875\mu_m^2 - 1.875\eta_m^2$	▷ Ordem 2
6	$K_{4,m} = 3.75 \mu_m \eta_m$	▷ Ordem 2
7	$K_{5,m} = 3.75\mu_m^2 + 1.875\eta_m^2 - 1.875$	▷ Ordem 2
8	$K_{6,m} = 3.75\eta_m^2 + 1.875\mu_m^2 - 1.875$	▷ Ordem 2
9	$K_{7,m} = 10.5\mu_m - 13.125\mu_m^3 - 13.125\mu_m\eta_m^2$	▷ Ordem 3
10	$K_{8,m} = 10.5\eta_m - 13.125\eta_m^3 - 13.125\eta_m\mu_m^2$	▷ Ordem 3
11	$K_{9,m} = 13.125\eta_m^3 + 26.25\eta_m\mu_m^2 - 13.125\eta_m$	▷ Ordem 3
12	$K_{10,m} = 13.125\mu_m^3 + 26.25\mu_m\eta_m^2 - 13.125\mu_m$	▷ Ordem 3
13	$K_{11,m} = 17.5\mu_m^3 + 13.125\mu_m\eta_m^2 - 13.125\mu_m$	▷ Ordem 3
14	$K_{12,m} = 17.5\eta_m^3 + 13.125\eta_m\mu_m^2 - 13.125\eta_m$	▷ Ordem 3

3.2.1.13 Cálculo do Fluxo Angular Espalhado (ψ_s)

No Algoritmo 13, realiza-se o cálculo do fluxo angular espalhado $\psi_{s,mg}^{ij}$ da equação 3.43 para cada posição (i, j) , direção m e energia g . Recebe os seguintes parâmetros de entrada provenientes do *loop* principal: grupo de energia g dos fluxos a calcular; matriz de tipos de materiais para as posições (i, j) criada no Algoritmo 6; fluxos nas faces verticais ψ_V e horizontais ψ_H dados a partir da solução inicial ou calculados pela iteração anterior; coeficientes $C_{p,n}$ pré-calculados pelo Algoritmo 11; coeficientes $K_{p,m}$ pré-calculados pelo Algoritmo 12; e seções de choque $\Sigma'_{s,p}$ pré-calculadas pelo Algoritmo 10.

Na linha 1, realiza-se o *loop* sobre cada posição (i, j) do domínio. Na linha 2, lê-se em t o tipo de material da posição (i, j) para saber se é uma posição de tecido sadio ($t = 1$) ou tumoral ($t = 2$). Na linha 3, inicializa-se o vetor L das 13 parcelas da expansão de Legendre que aparecem na equação equação 3.11. Na linha 4, percorre-se sobre os demais grupos de energia g' contendo as partículas de maior velocidade que espalham para o grupo g de menor ou igual velocidade. Na linha 5, inicializa-se o vetor S dos 13 somatórios da equação 3.11. Na linha 6, itera-se por todas as n direções de fluxos incidentes das partículas de maior velocidade. Na linha 7, calcula-se o fluxo incidente central ψ'_n como a média dos fluxos nas faces (equação 3.36). Na linha 8, acumulam-se os termos do vetor S de somatórios da equação 3.11 usando do coeficientes internos $C_{p,n}$. Na linha 9, acumulam-se os termos do vetor L que multiplicam os somatórios da equação 3.11, usando a lista de seções de choque $\Sigma'_{s,p}$. Na linha 10, itera-se por todas as m direções de fluxos emergentes das partículas de menor velocidade. Finalmente, na linha 11 calcula-se o fluxo espalhado usando os coeficientes externos $K_{p,m}$ que multiplicam os somatórios pré-calculados em L_p , gerando os mesmos valores da equação 3.11.

Algorithm 13: Scatter 2D - Cálculo do Fluxo Angular Espalhado

In: $g \in \mathbb{R}$, $T^{i,j} \in \mathbb{R}^{I \times J}$, $\psi_V, \psi_H \in \mathbb{R}^{M \times G \times I \times J}$

In: $C_{p,n} \in \mathbb{R}^{13 \times M}$: Coeficientes internos dos momentos de fluxo

In: $K_{p,m} \in \mathbb{R}^{13 \times M}$: Coeficientes externos dos momentos de fluxo

In: $\Sigma'_{s,p} \in \mathbb{R}^{13 \times G \times G}$: Lista das Seções de Choque de Espalhamento

Out: $\psi_s \in \mathbb{R}^{M \times G \times I \times J}$: Fluxo angular espalhado em $\hat{\Omega}_{m=1:M}$ e $E_{g=g}$

```

1 for  $(i, j) \in [1 : I] \times [1 : J]$  do
2    $t = T(i, j)$  ▷ 1 = sadio e 2 = tumor
3    $L_{0:12} = 0$  ▷ declara as parcelas externas de momentos de fluxo
4   for  $g' = 1 : g$  do ▷ para cada grupo incidente  $E_{g'}$  a espalhar em  $E_g$ 
5      $S_{0:12} = 0$  ▷ declara as parcelas internas de momentos de fluxo
6     for  $n = 1 : M$  do ▷ para cada direção incidente  $\hat{\Omega}_n$  a espalhar em  $\hat{\Omega}_m$ 
7        $\psi'_n = 0.5(\psi_V(n, g', i, j) + \psi_V(n, g', i + 1, j))$  ▷ obtém o fluxo médio
8        $S_{0:12} += C_{0:12,n} \cdot \psi'_n$  ▷ acumula as parcelas internas
9        $L_{0:12} += \Sigma'_{s,0:12}(t, g', g) \cdot S_{0:12}$  ▷ acumula as parcelas externas
10    for  $m = 1 : M$  do ▷ para cada direção  $M$ , calcula o fluxo espalhado
11       $\psi_s(m, g, i, j) = \sum_{p=0}^{12} K_{p,m} L_p$  ▷ somando os momentos de fluxo

```

3.2.1.14 Cálculo do Fluxo Angular Transportado (ψ_s, ψ_V, ψ_H)

O Algoritmo 14 realiza o transporte do fluxo angular em cada posição, direção e energia. Recebe como entrada os fluxos incidentes nas faces, o fluxo espalhado, o fluxo externo e malha de seções de choque total. Para cada posição do domínio, calcula o fluxo nos centros das células e nas faces emergentes. A linha 12 avalia a equação 3.43 para calcular o valor de ψ no centro da célula de varredura. A linha 13 avalia a equação 3.36 para calcular o fluxo que sai pela face vertical da célula. A linha 14 avalia a equação 3.37 para calcular o fluxo que sai pela face horizontal da célula. O apêndice D contém as ilustrações das varreduras para cada um dos quadrantes.

Algorithm 14: Sweep 2D

In: g : grupo entre 1 e G , q : quadrante 0,1,2 ou 3

In: $\Sigma_{t,g}^{ij} \in \mathbb{R}^{G \times I \times J}$: Seção de choque total em cada ponto da malha

In: $\psi, \psi_V, \psi_H, \psi_{ext} \in \mathbb{R}^{M \times G \times I \times J}$: Fluxos angulares

Out: $\psi, \psi_V, \psi_H \in \mathbb{R}^{M \times G \times I \times J}$: Fluxos angulares das faces emergentes

```

1  $f_x = 0$  se ( $q=0$  ou  $q=3$ ) senão  $f_x = 1$            ▷ Frente de varredura horizontal
2  $f_y = 0$  se ( $q=0$  ou  $q=1$ ) senão  $f_y = 1$            ▷ Frente de varredura vertical
3  $x_{beg} = f_x I$     $x_{end} = (1 - f_x)I$     $x_{inc} = 1 - 2f_x$    ▷ Intervalos de varredura em x
4  $y_{beg} = f_y J$     $y_{end} = (1 - f_y)J$     $y_{inc} = 1 - 2f_y$    ▷ Intervalos de varredura em y
5 for  $(i, j) \in [x_{beg} : x_{inc} : x_{end}] \times [y_{beg} : y_{inc} : y_{end}]$  do           ▷ para cada célula
6      $m_{beg} = 1 + q(M/4)$     $m_{end} = (1 + q)(M/4)$            ▷ índices das direções
7     for  $m = m_{beg} : m_{end}$  do           ▷ para cada direção do quadrante  $q$ 
8         ▷ define as abreviações dos símbolos
9          $\epsilon_x = 2\mu_m/h_{xi}$             $\epsilon_y = 2\eta_m/h_{yj}$ 
10         $\psi_V^{in} \equiv \psi_V(m, g, i + f_x, j)$     $\psi_V^{out} \equiv \psi_V(m, g, i + 1 - f_x, j)$ 
11         $\psi_H^{in} \equiv \psi_H(m, g, i, j + f_y)$     $\psi_H^{out} \equiv \psi_H(m, g, i, j + 1 - f_y)$ 
12         $\psi_s \equiv \psi(m, g, i, j)$             $\psi_{ext} \equiv \psi_{ext}(m, g, i, j)$ 
13        ▷ calcula os fluxos no centro da célula e faces emergentes
14         $\psi(m, g, i, j) = (\epsilon_x \psi_V^{in} + \epsilon_y \psi_H^{in} + \psi_s + \psi_{ext}) / (\epsilon_x + \epsilon_y + \Sigma_{t,g}^{ij})$ 
15         $\psi_V^{out} = 2\psi(m, g, i, j)$ 
16         $\psi_H^{out} = 2\psi(m, g, i, j)$ 

```

3.2.1.15 Cálculo do Fluxo Escalar (ϕ)

O fluxo escalar é calculado a partir da soma dos fluxos angulares. Para cada grupo g de energia, varrem-se os pontos x_i, y_j da malha. O fluxo escalar ϕ é inicializado com 0 e efetua-se a soma dos fluxos angulares ψ para cada direção $\hat{\Omega}_m$, ponderados pelos respectivos pesos ω_m . Como os pesos somam 1 para cada quadrante, divide-se o fluxo resultante por 4 para que os pesos efetuem a mesma soma após a soma dos 4 quadrantes. O operador *max* é utilizado para garantir que o fluxo escalar resultante seja positivo.

Algorithm 15: Scalar 2D - Cálculo do Fluxo Escalar

In: $\psi \in \mathbb{R}^{M \times G \times I \times J}$: Fluxo angular, $\omega_m \in \mathbb{R}^M$: Pesos da quadratura S_N

Out: $\phi \in \mathbb{R}^{G \times I \times J}$: Fluxo escalar

```

1 for  $g = 1 : G$  do
2   for  $(i, j) \in [1 : I] \times [1 : J]$  do
3      $\phi(g, i, j) = 0$ 
4     for  $m = 1 : M$  do ▷ soma os fluxos angulares em cada direção
5        $\phi(g, i, j) += 0.25 \cdot \omega_m \cdot \psi(m, g, i, j)$ 
6      $\phi(g, i, j) = \max(0, \phi(g, i, j))$ 

```

3.2.1.16 Verificação do Critério de Parada

O critério de parada é dado pela verificação da convergência da solução aproximada com a solução da equação de transporte DD . Varre-se todo o domínio do fluxo escalar, para cada grupo g e posição x_i, y_j e verifica-se a diferença entre as soluções da iteração atual e anterior. Memoriza-se na variável *erro* o maior valor de diferença. Ao término da varredura verifica-se se o erro é menor que a uma tolerância pré-definida (10^{-6}). Em caso negativo retorna-se *false* para que o *loop* iterativo principal se reinicie. Em caso positivo,

retorna-se *true* indicando que a solução atingiu o estado estacionário.

Algorithm 16: Convergence 2D - Verificação do Critério de Parada

In: $\phi, \phi_{old} \in \mathbb{R}^{G \times I \times J}$: Fluxos escalares da iteração atual e anterior

In: *tolerancia* $\in \mathbb{R}$: Constante 10^{-6}

Out: *criterio* $\in \mathbf{bool}$: Critério de parada

```

1 criterio = false
2 erro = 0.0
3 for  $g = 1 : G$  do
4   for  $(i, j) \in [1 : I] \times [1 : J]$  do
5     diff =  $\text{abs}(\phi(g, i, j) - \phi_{old}(g, i, j))$ 
6     if  $\phi(g, i, j) \neq 0.0$  then
7       diff =  $(\text{diff} / \text{abs}(\phi(g, i, j))) \cdot 100$ 
8       erro =  $\max(\text{erro}, \text{diff})$ 
9 if  $\text{erro} < \text{tolerancia}$  then
10  criterio = true

```

3.2.1.17 Atualização da Solução

Caso a solução da equação de transporte não tenha atingido o estado estacionário, efetua-se a alternância entre as variáveis de fluxo escalar atual ϕ e anterior ϕ_{old} , de modo a preservar os valores da solução atual em ϕ_{old} e poder atualizar a próxima solução ϕ com novos valores, permitindo a comparação das soluções no Algoritmo 16. Na implementação computacional, as atribuições realizadas nas linhas 1, 2 e 3 são realizadas por ponteiros de modo a evitar cópias desnecessárias de $G \times I \times J$ componentes a cada atribuição.

Algorithm 17: Update2D - Atualiza a solução

In: $\phi, \phi_{old} \in \mathbb{R}^{G \times I \times J}$: Fluxos escalares da iteração atual e anterior

Out: $\phi, \phi_{old} \in \mathbb{R}^{G \times I \times J}$: Fluxos escalares da iteração atual e anterior

```

1 temp =  $\phi_{old}$ 
2  $\phi_{old}$  =  $\phi$ 
3  $\phi$  = temp

```

▷ [Alterna as soluções atual e anterior](#)

3.2.1.18 Algoritmo *Dose2D*

Após a solução atingir o estado estacionário, realiza-se o cálculo da dose de radiação devido ao Boro em cada ponto x_i, y_j da malha a partir do fluxo escalar ϕ e demais características do meio [26]. A dose calculada possui unidade Gy/h, onde Gy = J/Kg corresponde à energia (Joule) por unidade de massa (Kg).

Algorithm 18: Dose 2D - Cálculo da Dose de Energia Depositada no Tecido

In: $\phi \in \mathbb{R}^{G \times I \times J}$: Fluxo escalar em cada ponto da malha
In: $T^{ij} \in \mathbb{R}^{I \times J}$: Tipo de material em cada ponto da malha
In: $\Sigma_a \in \mathbb{R}^{G \times 2}$: Seção de Choque de Absorção para G energias e 2 materiais
In: $E_T \in \mathbb{R}$: Energia (J) total liberada pela reação dada por 2,31 MeV
In: $c \in \mathbb{R}$: Constante de conversão de MeV/gr.s em Gy/h dada por $5,76 \times 10^{-7}$
Out: $d \in \mathbb{R}^{G \times I \times J}$: Dose (Gy/h) devido ao Boro na posição (i, j) e grupo g

```

1 for  $g = 1 : G$  do
2   for  $(i, j) \in [1 : I] \times [1 : J]$  do
3      $t = T^{ij}$  ▷ 1 (sadio) ou 2 (tumor)
4      $d(g, i, j) = \phi(g, i, j) \cdot \Sigma_a(g, t) \cdot E_T \cdot c$ 

```

3.3 Dados Experimentais

Os dados experimentais utilizados nos experimentos foram obtidos da referência [12] e consistem em:

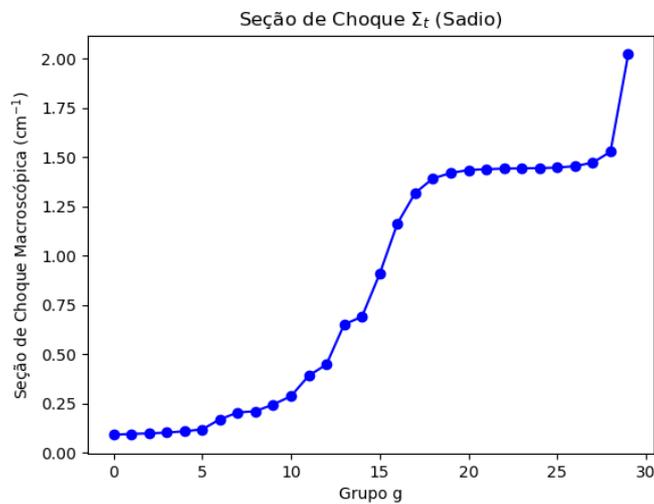
- Arquivos de seção de choque total Σ_t (Sadio e Tumor);
- Arquivo de seção de choque de absorção Σ_a (Tumor);
- Arquivos de momento 0 de seção de choque de espalhamento Σ_{s0} (Sadio e Tumor);
- Arquivos de momento 1 de seção de choque de espalhamento Σ_{s1} (Sadio e Tumor);
- Arquivos de momento 2 de seção de choque de espalhamento Σ_{s2} (Sadio e Tumor);
- Arquivos de momento 3 de seção de choque de espalhamento Σ_{s3} (Sadio e Tumor);

Os arquivos contém valores de seção de choque para $G=30$ grupos de energia, distribuídos numa faixa iniciada pelo grupo de maior energia 16 MeV e finalizada pelo grupo de menor energia $7,6 \times 10^{-8}$ MeV, como ilustrado na tabela 3.1 [12].

Tabela 3.1: Valores de energia para cada grupo

Grupo	Energia (MeV)	Grupo	Energia (MeV)	Grupo	Energia (MeV)
01	1.6000E+01	11	1.5455E+00	21	8.4450E-04
02	1.4250E+01	12	1.0880E+00	22	3.1050E-04
03	1.2750E+01	13	6.6150E-01	23	1.1420E-04
04	1.1000E+01	14	4.0150E-01	24	4.2000E-05
05	8.8950E+00	15	2.4350E-01	25	1.5460E-05
06	6.9300E+00	16	1.2580E-01	26	5.6900E-06
07	4.8750E+00	17	4.6200E-02	27	2.0950E-06
08	3.2725E+00	18	1.6960E-02	28	7.7200E-07
09	2.5485E+00	19	6.2350E-03	29	2.8300E-07
10	1.9850E+00	20	2.2925E-03	30	7.6070E-08

Os arquivos de seção de choque total Σ_t para o tecido sadio e tumoral estão ilustrados nas Figuras 3.7 e 3.8. Os valores de Σ_t (Sadio) e Σ_t (Tumor) são lidos pelo algoritmo 7 (linha 1) onde são interpretados como vetores de G componentes que preenchem a matriz Σ_t de dimensões $G \times 2$. Os valores são menores para os primeiros grupos de energia (região de altas energias) e maiores para os últimos grupos de energia (região de baixas energias) evidenciando uma maior facilidade dos nêutrons mais lentos de serem capturados.

Figura 3.7: Seção de Choque Σ_t (Sadio)

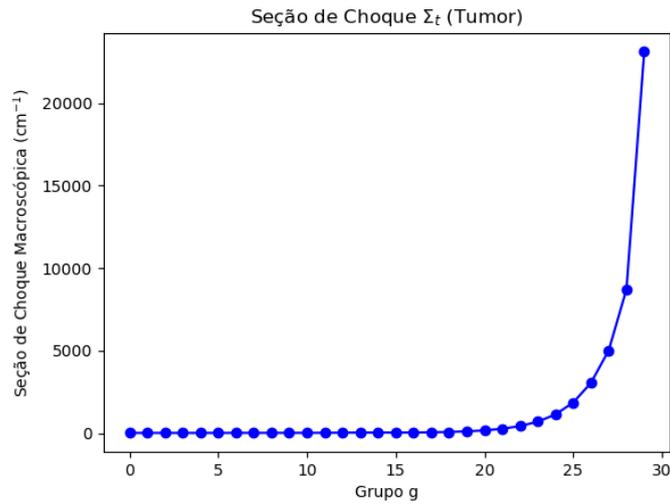


Figura 3.8: Seção de Choque Σ_t (Tumor)

O arquivo de seção de choque de absorção Σ_a é usado somente para o tecido tumoral, por este conter o elemento Boro em sua composição e está ilustrado na figura 3.9. Os valores de Σ_a (Tumor) são lidos pelo algoritmo 7 (linha 1) onde são interpretados como vetores de G componentes que preenchem a matriz Σ_t de dimensões $G \times 2$.

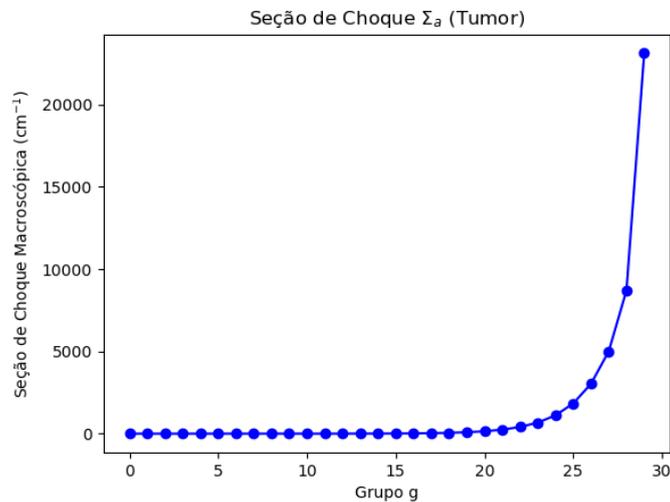
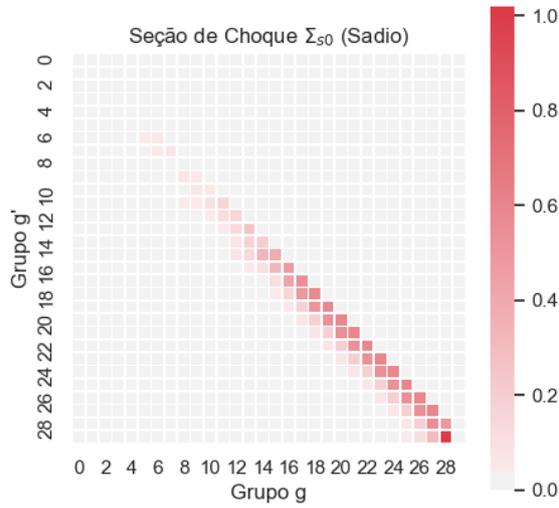


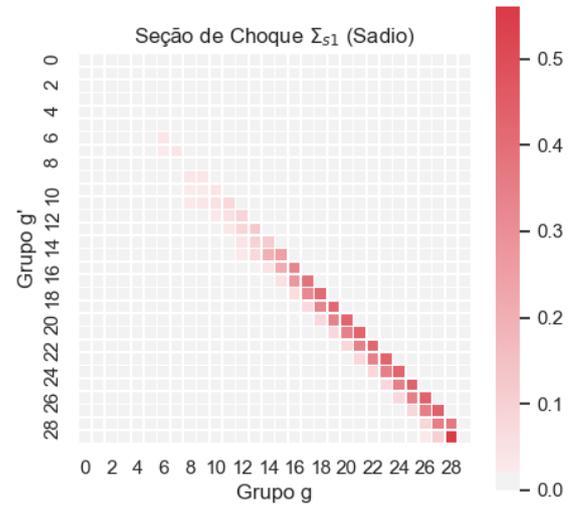
Figura 3.9: Seção de Choque Σ_a (Tumor)

Os arquivos de momento de seção de choque de espalhamento Σ_{s0} , Σ_{s1} , Σ_{s2} e Σ_{s3} estão ilustrados nas Figuras 3.10 e 3.11 que representam as componentes da matrizes $G \times G$ de momentos de seção de choque de espalhamento de graus 0 (Σ_{s0}), 1 (Σ_{s1}), 2 (Σ_{s2}) e 3 (Σ_{s3}). Estas matrizes são do tipo triangular inferior pois representam o processo de espalhamento do tipo *downscattering* onde só há espalhamento de partículas de mais alta energia para grupos de mais baixa energia. Os valores dessas matrizes são lidos pelo

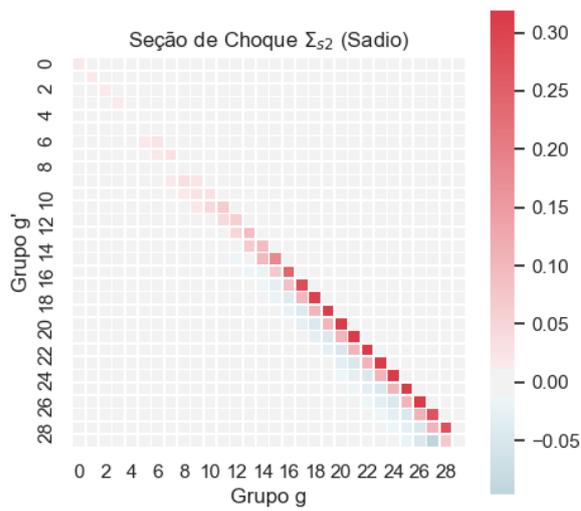
Algoritmo 7.



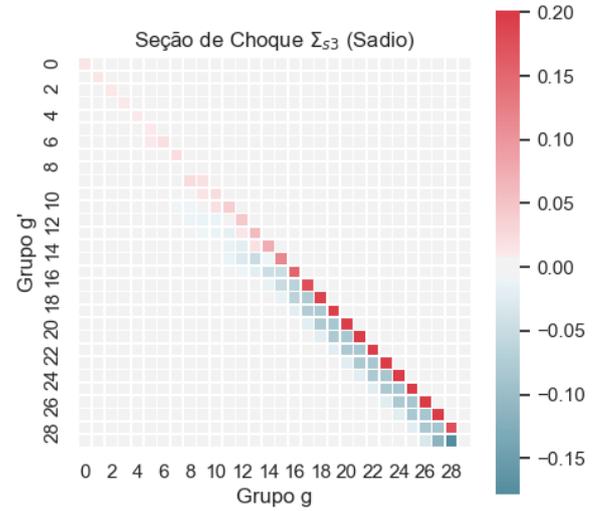
(a) Seção de Choque Σ_{s0}



(b) Seção de Choque Σ_{s1}



(c) Seção de Choque Σ_{s2}



(d) Seção de Choque Σ_{s3}

Figura 3.10: Seção de Choque de Espalhamento - Tecido Sadio

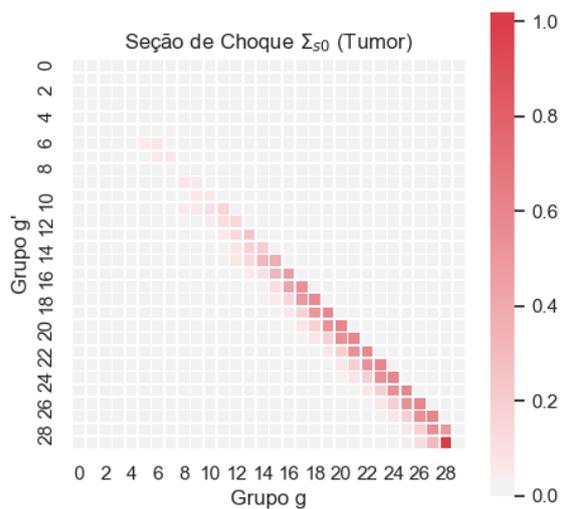
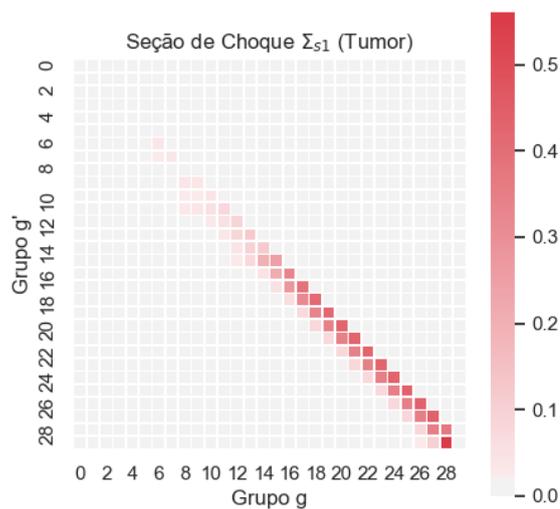
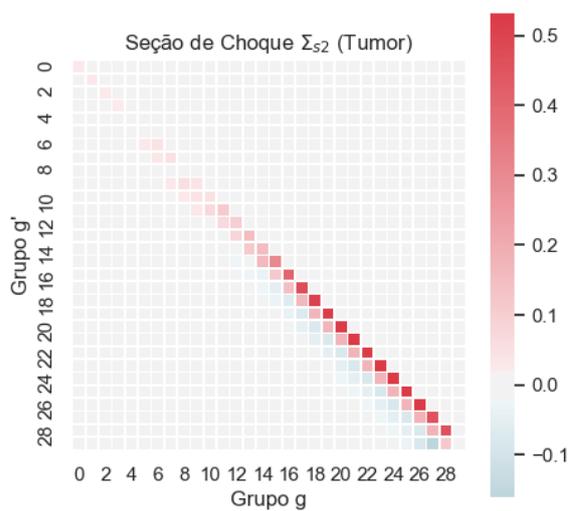
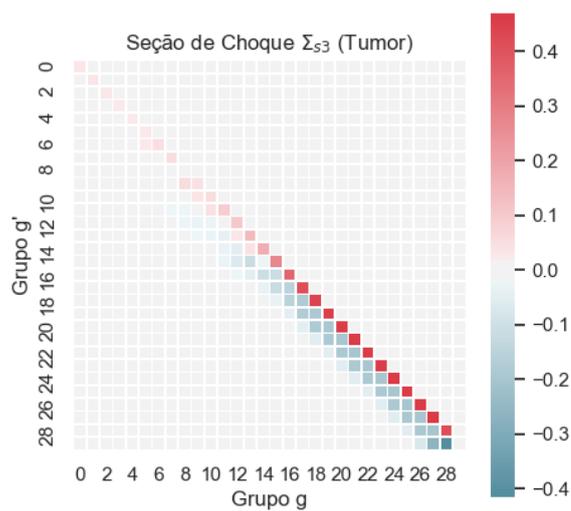
(a) Seção de Choque Σ_{s0} (b) Seção de Choque Σ_{s1} (c) Seção de Choque Σ_{s2} (d) Seção de Choque Σ_{s3}

Figura 3.11: Seção de Choque de Espalhamento - Tecido Tumoral

Capítulo 4

Resultados e Discussões

Neste capítulo são apresentados os resultados obtidos pelas simulações computacionais dos modelos descritos no capítulo de metodologia. Na seção 4.1 são apresentados os resultados referentes ao transporte de nêutrons no caso estacionário assim como a dose de energia liberada. As simulações foram realizadas em um desktop Intel(R) Core(R) i5-4460 @3,20GHz, 8GB RAM em sistema operacional Windows 7 64bits.

4.1 Simulações de Transporte

Nas demais seções, são apresentados os resultados dos fluxos escalares de nêutrons obtidos após a solução atingir o estado estacionário. Para todos os casos, considerou-se o domínio físico de $6\text{cm} \times 6\text{cm}$ e um domínio de 30 grupos de energia distribuídos entre $1,7 \times 10^7\text{eV}$ e $1,39 \times 10^{-4}\text{eV}$. O critério de parada utilizado foi dado pela satisfação de umas das duas condições: erro relativo entre duas soluções ser menor que $\epsilon = 10^{-6}$ para todos os elementos $\phi(g, i, j)$ como descrito pelo Algoritmo 16 ; ou número máximo de 50 iterações atingido.

Os experimentos são divididos em duas partes. A primeira, foca a comparação dos valores gerados entre os códigos em F77 e C++ para validar a igualdade dos resultados e permitir uma comparação dos tempos. A segunda, explora os diversos cenários de posicionamento de fonte, intensidade do feixe, resolução espacial e quantidade de direções de espalhamento, gerados pelo código C++.

O tecido sadio foi representado por seções de choque macroscópicas contendo os elementos C (Carbono), H (Hidrogênio), N (Nitrogênio) e O (Oxigênio). O valores de concentração de cada elemento foram obtidos de [12]. O tecido tumoral foi representado

por seções de choque contendo os elementos do tecido sadio acrescidos do elemento B (Boro) de modo a simular um material com maior capacidade de absorção de nêutrons. A geometria do tumor foi gerada pelo próprio código, a partir dos seguintes parâmetros fornecidos pelo arquivo de configuração: $(TumorX, TumorY)$ representando a posição do centro do tumor no domínio; $(TumorS_X, TumorS_Y)$ contendo a largura e altura do tumor; e $GeometriaTumor$ especificando o tipo de tumor como quadrado (0) ou circular (1). Durante a execução do Algoritmo 6 os pontos da malha externos à área do tumor são classificados como sadios (Tipo 1) e os internos como tumorais (Tipo 2).

A fonte fixa de nêutrons para todos os experimentos está localizada na lateral esquerda do domínio $6\text{cm} \times 6\text{cm}$, ocupando uma faixa de 2cm de altura (Figura 4.1).

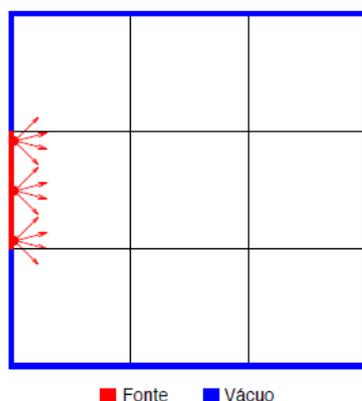


Figura 4.1: Fonte externa inicializada no contorno do domínio

Nas demais seções são apresentados os resultados obtidos para o grupo de energia na faixa térmica (grupo 28) por esta ser a faixa de energia com maior probabilidade de captura de nêutrons pelo Boro.

4.1.1 Experimentos comparativos entre os códigos F77 e C++

Nesta seção são apresentados os resultados de 8 experimentos idênticos realizados nas implementações do F77 e C++ de modo a comparar os resultados e os tempos de execução. Os dados comuns a todos os experimentos são: domínio físico quadrado de lado 6 cm ; tumor com geometria quadrada de lado 2 cm e posição $(x,y)=(3\text{ cm}, 3\text{ cm})$; feixe de nêutrons no canto esquerdo do domínio, na faixa de $(x,y)=(0\text{ cm}, 2\text{ cm})$ até $(x,y)=(0\text{ cm}, 4\text{ cm})$; e intensidade de feixe igual a 1. Os experimentos de 1 a 5 fixam $M=12$ direções e $G=30$ grupos, variando a resolução espacial $I \times J$ entre 120×120 , 180×180 , 240×240 , 300×300 e 360×360 . Os experimentos 6 e 7 fixam $I \times J$ em 120×120 células e $G=30$ grupos, variando a quantidade de direções entre $M=24$ (S_6) e $M=40$ (S_8). O experimento

8 usa $M=144$ (S_{16}) direções, $G=30$ grupos e resolução espacial $I \times J$ igual a 90×90 .

A tabela 4.1 sumariza os resultados das simulações. Os dados de resolução estão nas colunas: I e J (quantidade de colunas e linhas); M (quantidade de direções do espalhamento); e G (número de grupos de energia). Os dados de saída das simulações estão nas colunas: $Iter$ (número de iterações); $Erro$ (valor do resíduo); $F77(s)$ (tempo do F77 em segundos); $C++(s)$ (tempo do C++ em segundos); $F77/C++$ (razão entre o tempos do F77 e C++). Verifica-se pela última coluna da tabela ($F77/C++$) que a implementação em C++ executou de 4,5 a 11,5 vezes mais rápido que o código F77 usado como referência.

Tabela 4.1: Experimentos comparativos entre os códigos F77 e C++

Nº	I	J	M	G	Iter	Erro	F77(s)	C++(s)	F77/C++
1	120	120	12	30	37	10^{-6}	112	23	4,9
2	180	180	12	30	42	10^{-6}	265	58	4,6
3	240	240	12	30	37	10^{-6}	431	96	4,5
4	300	300	12	30	50	1.5×10^{-6}	927	202	4,6
5	360	360	12	30	50	1.5×10^{-6}	1309	292	4,5
6	120	120	24	30	38	10^{-6}	237	33	7,2
7	120	120	40	30	39	10^{-6}	400	50	8
8	90	90	144	30	32	10^{-6}	737	64	11,5

As demais seções ilustram os resultados obtidos por cada simulação na forma de quatro figuras contendo: (a) a matriz $I \times J$ do fluxo escalar para o grupo $g=28$ obtido pelo F77 (variável ϕ); (b) a matriz $I \times J$ do fluxo escalar para o grupo $g=28$ obtido pelo C++ (variável ϕ); (c) a matriz $I \times J$ de diferenças entre os fluxos (a) e (b); e (d) o resíduo da solução do C++ ao longo das iterações, calculado pelo Algoritmo 16.

4.1.1.1 Experimento Numérico 1

Os resultados obtidos para as resoluções $(I,J,M,G) = (120,120,12,30)$ são ilustrados na figura 4.2. A simulação termina após 37 iterações tanto no F77(a) quanto no C++(b). A matriz de diferenças (c) ilustra a igualdade das soluções e contém somente valores iguais a zero. O resíduo (d) decresce até atingir a tolerância $\epsilon = 10^{-6}$.

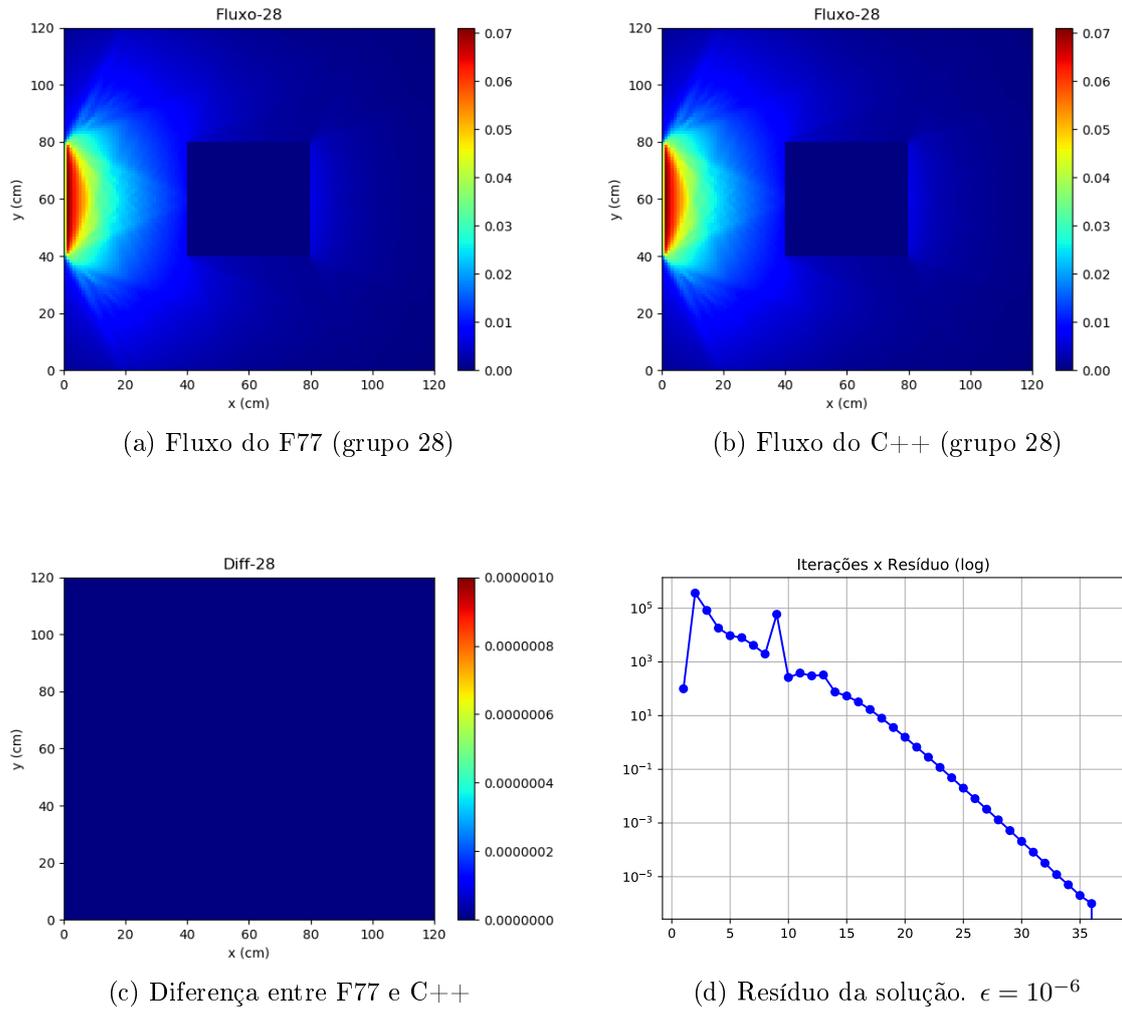


Figura 4.2: Experimento 1: Simulações comparadas para $(I,J,M,G) = (120,120,12,30)$

4.1.1.2 Experimento Numérico 2

Os resultados obtidos para as resoluções $(I,J,M,G) = (180,180,12,30)$ são ilustrados na figura 4.3. A simulação termina após 42 iterações tanto no F77(a) quanto no C++(b). A matriz de diferenças (c) ilustra a igualdade das soluções e contém somente valores iguais a zero. O resíduo (d) decresce até atingir a tolerância $\epsilon = 10^{-6}$.

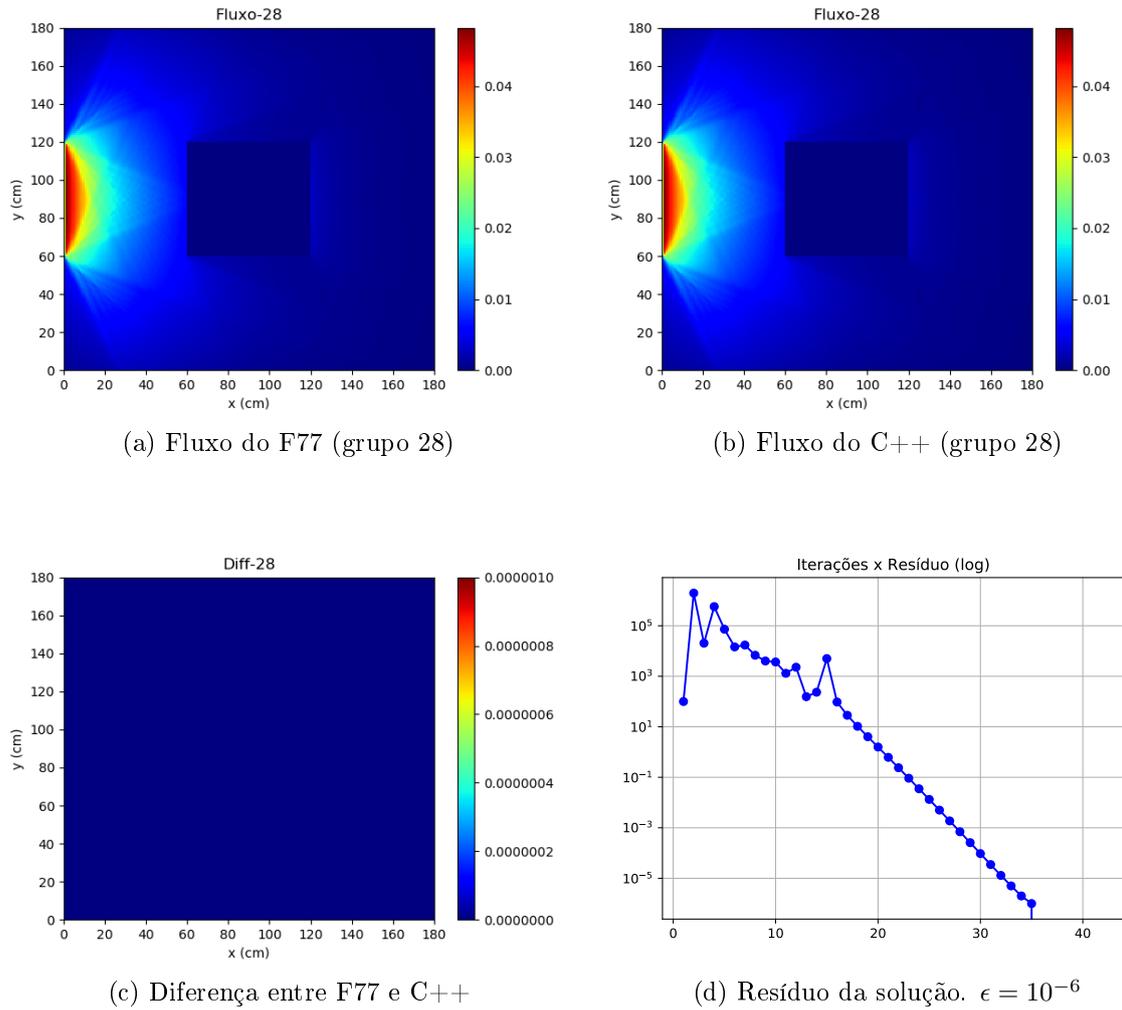


Figura 4.3: Experimento 2: Simulações comparadas para $(I,J,M,G) = (180,180,12,30)$

4.1.1.3 Experimento Numérico 3

Os resultados obtidos para as resoluções $(I,J,M,G) = (240,240,12,30)$ são ilustrados na figura 4.4. A simulação termina após 37 iterações tanto no F77(a) quanto no C++(b). A matriz de diferenças (c) ilustra a igualdade das soluções e contém somente valores iguais a zero. O resíduo (d) decresce até atingir a tolerância $\epsilon = 10^{-6}$.

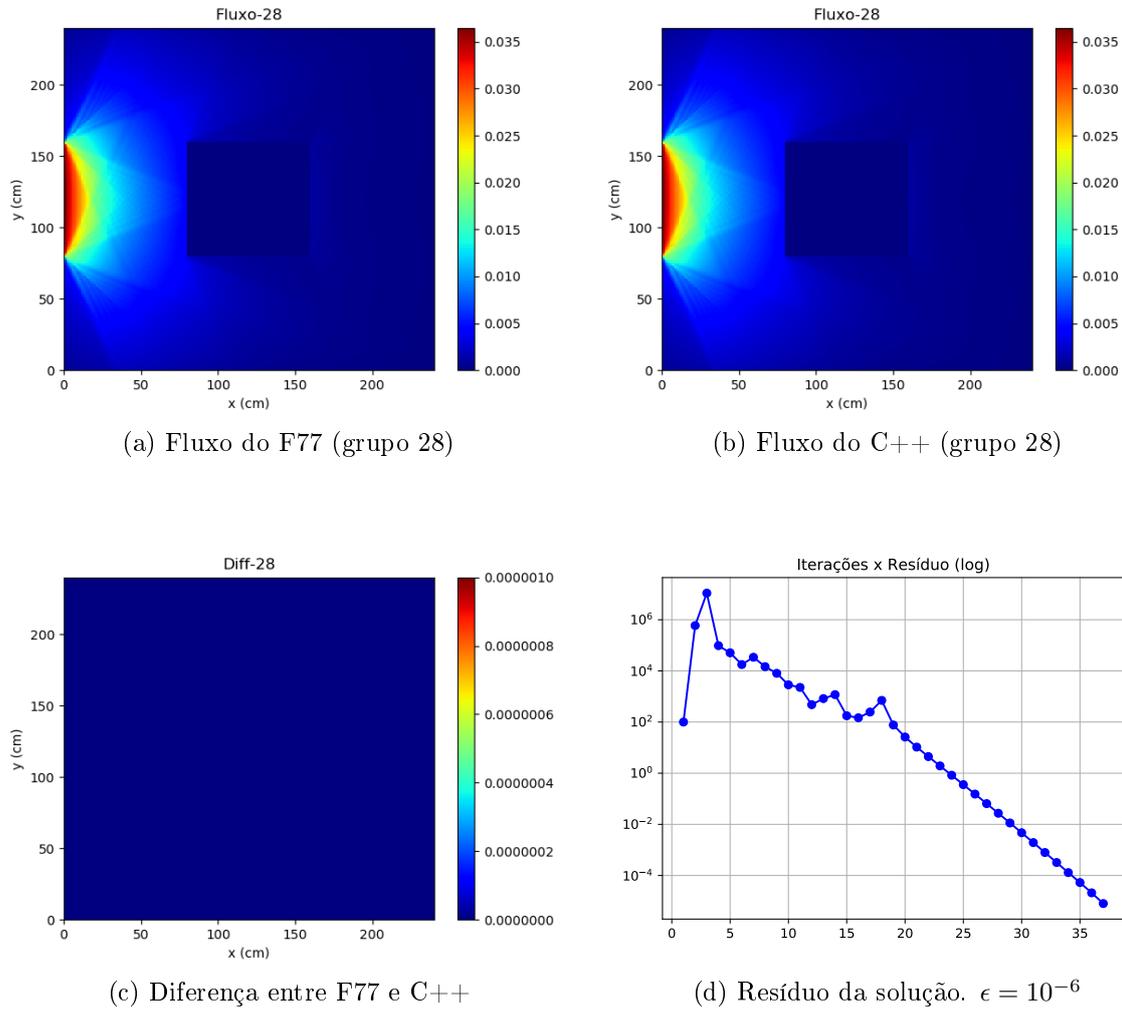


Figura 4.4: Experimento 3: Simulações comparadas para $(I, J, M, G) = (240, 240, 12, 30)$

4.1.1.4 Experimento Numérico 4

Os resultados obtidos para as resoluções $(I, J, M, G) = (300, 300, 12, 30)$ são ilustrados na figura 4.5. A simulação termina após 50 iterações tanto no F77(a) quanto no C++(b), devido ao critério de número máximo de iterações ser atingido antes do resíduo convergir para 10^{-6} . A matriz de diferenças (c) ilustra a igualdade das soluções e contém somente valores iguais a zero. O resíduo (d) decresce até se estabilizar na faixa $10^{-6} \leq \epsilon \leq 10^{-5}$

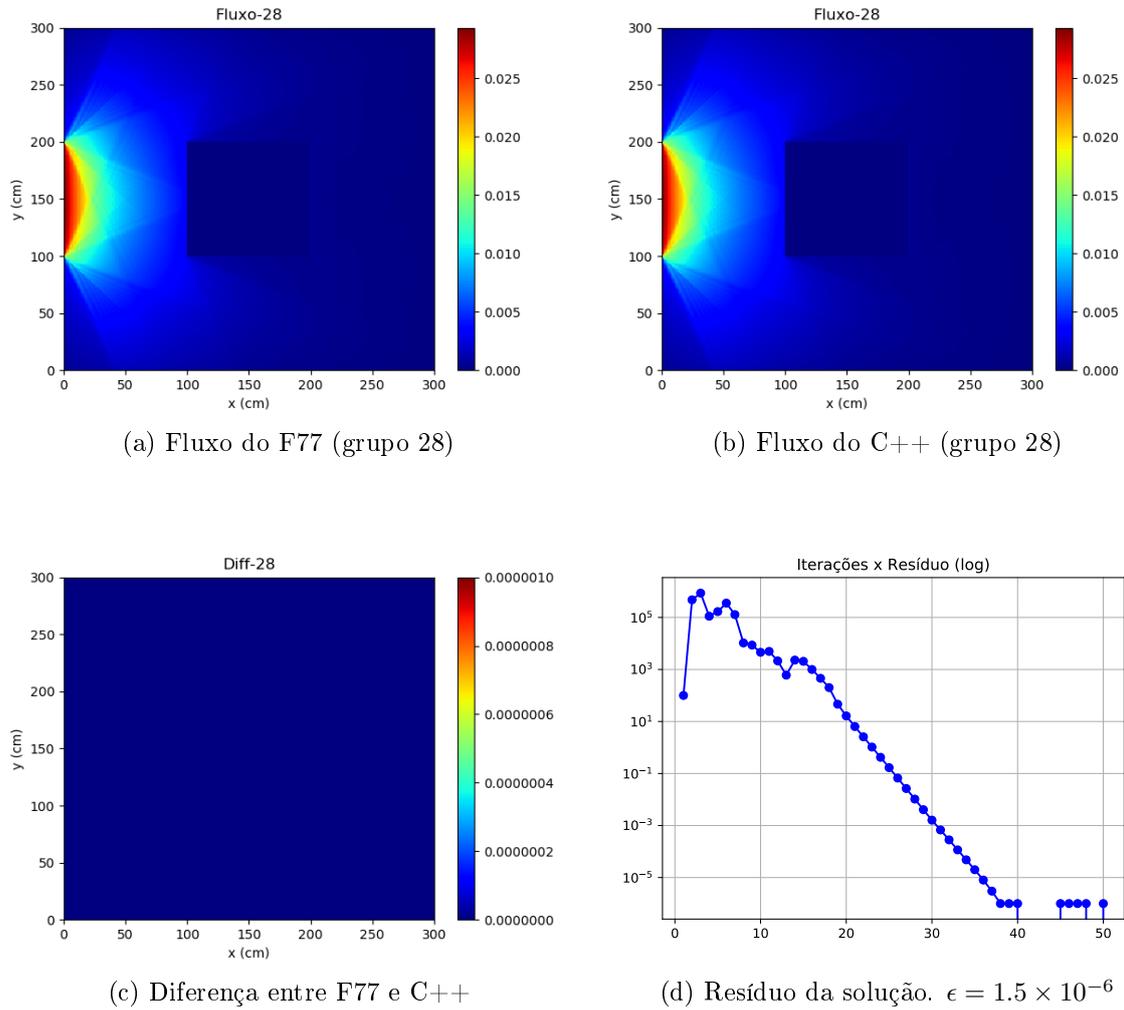


Figura 4.5: Experimento 4: Simulações comparadas para $(I,J,M,G) = (300,300,12,30)$

4.1.1.5 Experimento Numérico 5

Os resultados obtidos para as resoluções $(I,J,M,G) = (360,360,12,30)$ são ilustrados na figura 4.6. A simulação termina após 50 iterações tanto no F77(a) quanto no C++(b) devido ao critério de número máximo de iterações ser atingido antes do resíduo convergir para 10^{-6} . A matriz de diferenças (c) ilustra a igualdade das soluções e contém somente valores iguais a zero. O resíduo (d) decresce até se estabilizar na faixa $10^{-6} \leq \epsilon \leq 10^{-5}$

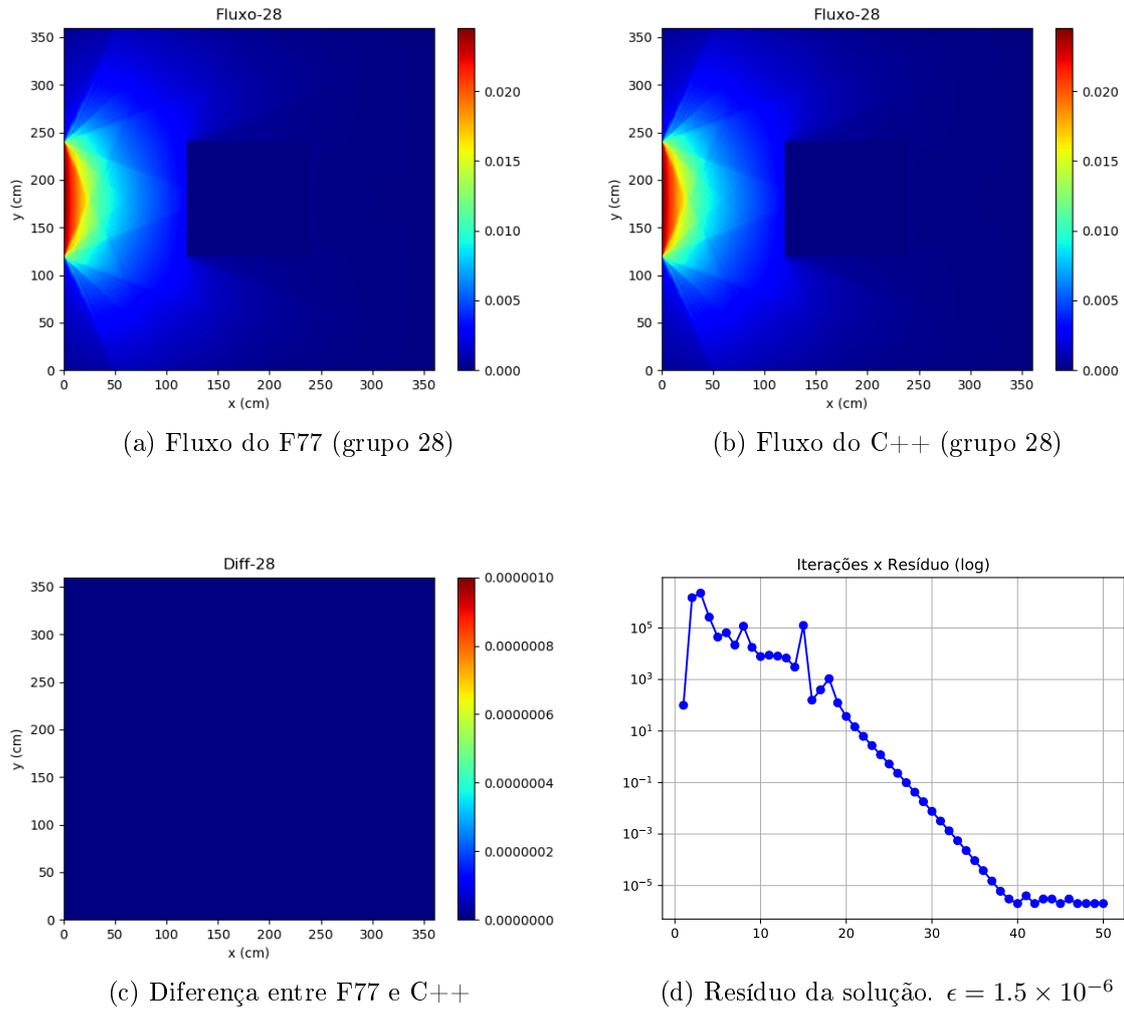


Figura 4.6: Experimento 5: Simulações comparadas para $(I, J, M, G) = (360, 360, 12, 30)$

4.1.1.6 Experimento Numérico 6

Os resultados obtidos para as resoluções $(I, J, M, G) = (120, 120, 24, 30)$ são ilustrados na figura 4.7. A simulação termina após 38 iterações tanto no F77(a) quanto no C++(b). A matriz de diferenças (c) ilustra a igualdade das soluções e contém somente valores iguais a zero. O resíduo (d) decresce até atingir a tolerância $\epsilon = 10^{-6}$.

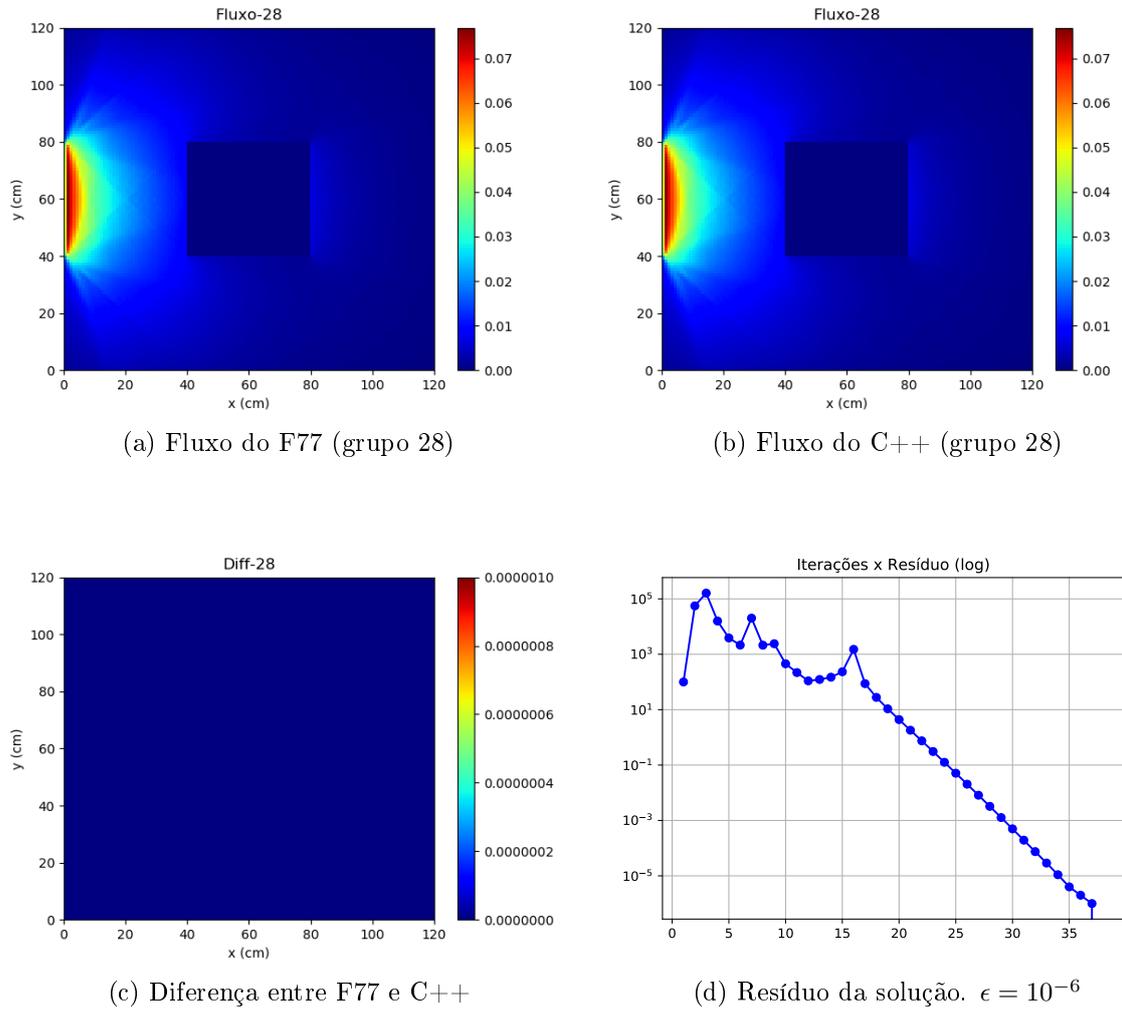


Figura 4.7: Experimento 6: Simulações comparadas para $(I, J, M, G) = (120, 120, 24, 30)$

4.1.1.7 Experimento Numérico 7

Os resultados obtidos para as resoluções $(I, J, M, G) = (120, 120, 40, 30)$ são ilustrados na figura 4.8. A simulação termina após 39 iterações tanto no F77(a) quanto no C++(b). A matriz de diferenças (c) ilustra a igualdade das soluções e contém somente valores iguais a zero. O resíduo (d) decresce até atingir a tolerância $\epsilon = 10^{-6}$.

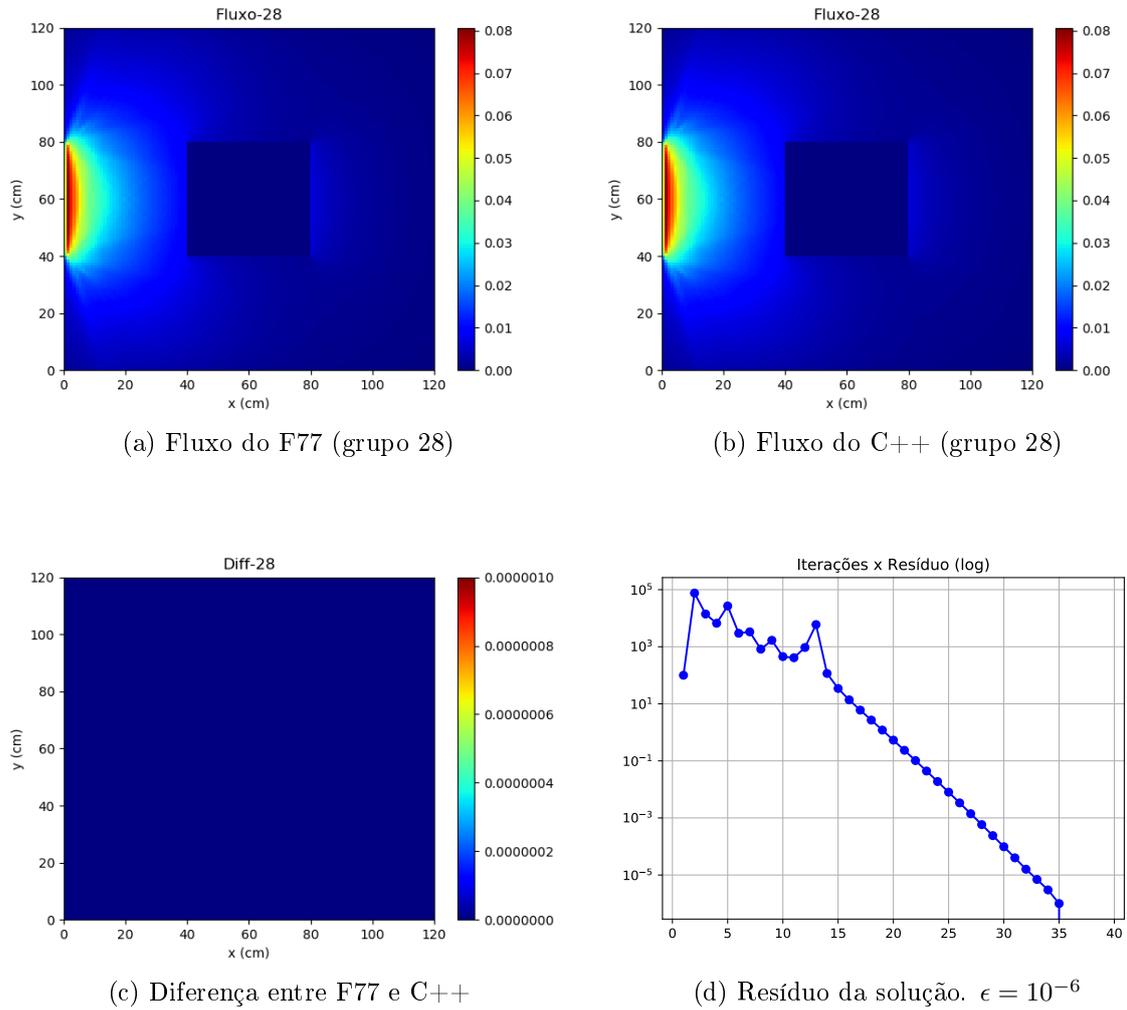


Figura 4.8: Experimento 7: Simulações comparadas para $(I, J, M, G) = (120, 120, 40, 30)$

4.1.1.8 Experimento Numérico 8

Os resultados obtidos para as resoluções $(I, J, M, G) = (90, 90, 144, 30)$ são ilustrados na figura 4.9. A simulação termina após 32 iterações tanto no F77(a) quanto no C++(b). A matriz de diferenças (c) ilustra a igualdade das soluções e contém somente valores iguais a zero. O resíduo (d) decresce até atingir a tolerância $\epsilon = 10^{-6}$.

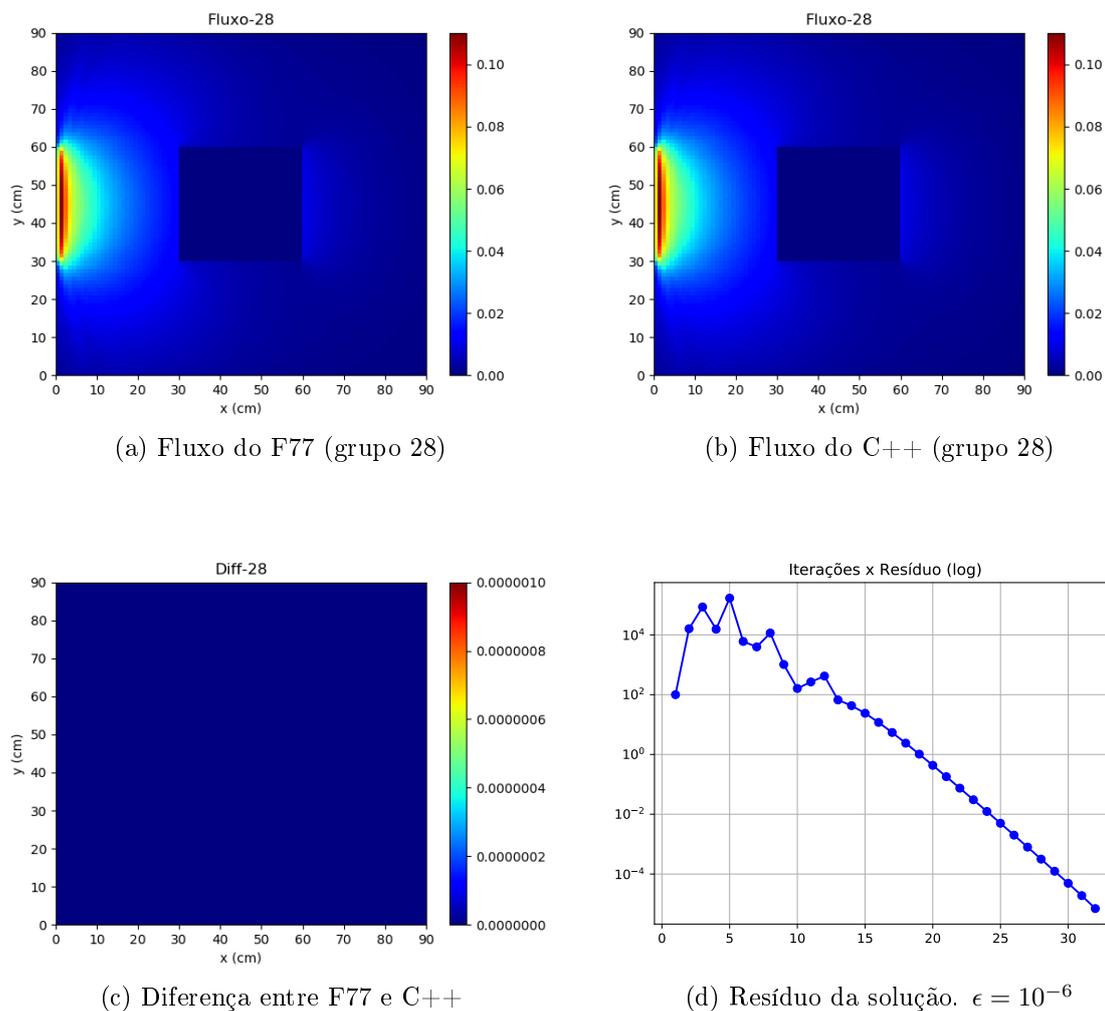


Figura 4.9: Experimento 8: Simulações comparadas para $(I,J,M,G) = (90,90,144,30)$

4.1.2 Experimentos com o código C++

Nesta seção são apresentados os resultados dos experimentos realizados na implementação do C++, explorando os diversos cenários de posicionamento de fonte, intensidade do feixe, resolução espacial e quantidade de direções de espalhamento.

4.1.2.1 Experimento Numérico 1

Neste experimento, o meio material é representado apenas por seções de choque do tecido sadio ocupando todo o domínio. O feixe da fonte fixa possui intensidade $240 N/(cm^2 s)$ e o espalhamento possui grau de anisotropia igual a 3. O fluxo escalar de nêutrons obtido no estado estacionário para o grupo de energia $g=28$ (faixa térmica) é apresentado na figura 4.10(a). Neste grupo todas as partículas possuem a mesma velo-

cidade e são mais facilmente capturadas ao longo do transporte. A região esquerda da imagem possui a maior concentração de partículas devido a proximidade com o feixe incidente. As partículas dissipam-se pelo meio a medida que avançam para o interior do domínio onde escapam para o vácuo após atravessarem o contorno do lado oposto. Como o meio não possui regiões contendo Boro, tem-se que a dose depositada é nula (figura 4.10(b)). A simulação termina após 35 iterações.

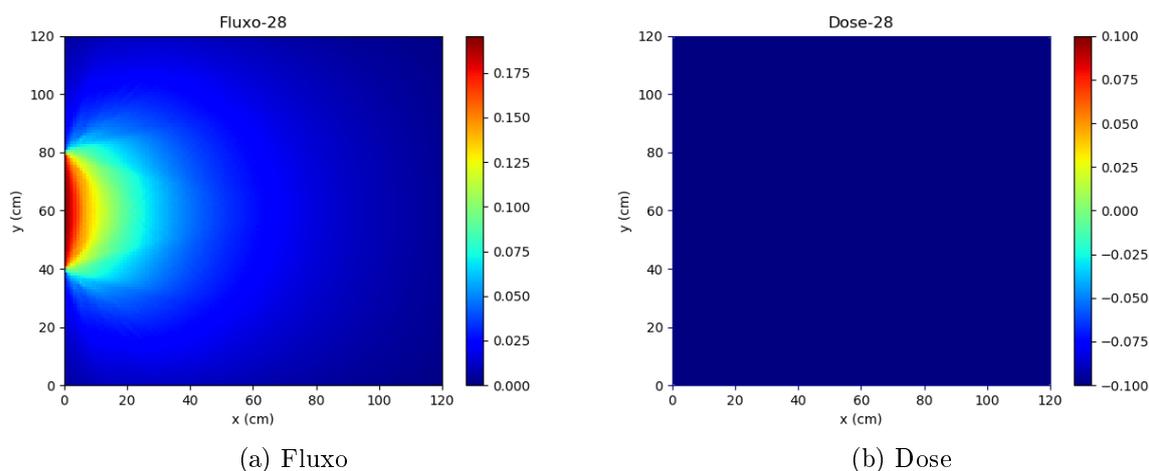


Figura 4.10: Resultado do experimento 1. (a) Fluxo escalar de nêutrons. (b) Dose de energia nula devido a ausência de Boro (Dose Total = 0 Gy).

4.1.2.2 Experimento Numérico 2

Neste experimento, o meio material heterogêneo é representado por seções de choque do tecido sadio ocupando a região externa do domínio e seções de choque contendo Boro na região quadrada de lado 2cm centralizada na posição $(x, y) = (3, 3)$ do domínio. O feixe da fonte fixa possui intensidade $240 N/(cm^2 s)$ e o espalhamento possui grau de anisotropia igual a 3. O fluxo escalar de nêutrons obtido no estado estacionário para o grupo de energia $g=28$ (faixa térmica) é apresentado na figura 4.11(a). Neste grupo todas as partículas possuem a mesma velocidade e são mais facilmente capturadas ao longo do transporte. A região esquerda da imagem possui a maior concentração de partículas devido a proximidade com o feixe incidente. As partículas dissipam-se pelo meio e são fortemente absorvidas na região quadrada contendo Boro e avançam com maior liberdade após atravessar a mesma. A figura 4.11(b) ilustra a dose de energia liberada na região quadrada contendo alta concentração de Boro e a dose é maior no lado esquerdo devido a proximidade com a fonte de nêutrons. A dose total é menor comparada aos outros

experimentos devido a maior distância à fonte. A simulação termina após 34 iterações.

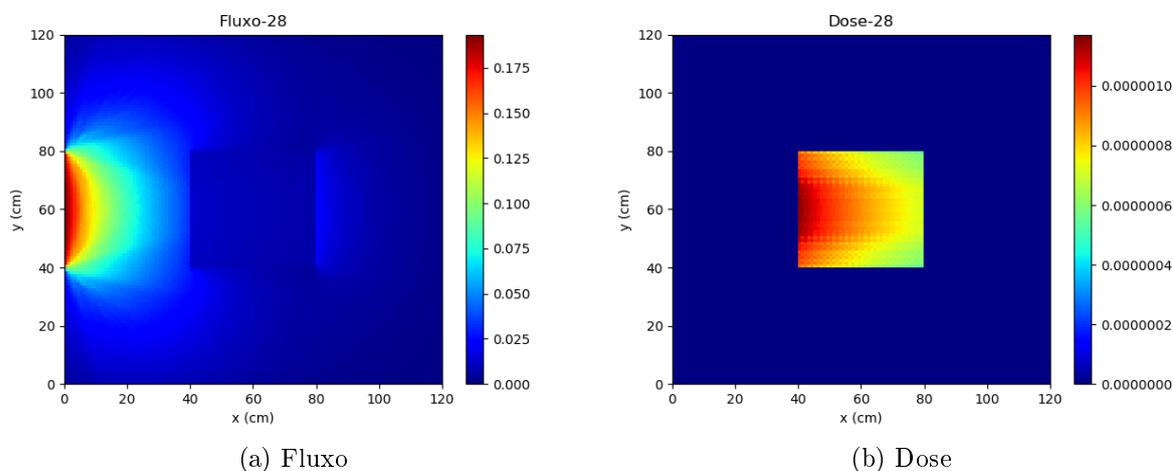


Figura 4.11: Resultado do experimento 2. (a) Fluxo escalar de nêutrons menor na região central devido à absorção. (b) Dose de energia concentrada na região quadrada (dose total = $1,343 \times 10^{-3}$ Gy).

4.1.2.3 Experimento Numérico 3

Neste experimento, o meio material heterogêneo é representado por seções de choque do tecido sadio ocupando a região externa do domínio e seções de choque contendo Boro na região circular de raio 1cm centralizada na posição $(x, y) = (3, 3)$. O feixe da fonte fixa possui intensidade $240 N/(cm^2 s)$ e o espalhamento possui grau de anisotropia igual a 3. O fluxo escalar de nêutrons obtido no estado estacionário para o grupo de energia $g=28$ (faixa térmica) é apresentado na figura 4.12(a). Neste grupo todas as partículas possuem a mesma velocidade e são mais facilmente capturadas ao longo do transporte. A região esquerda da imagem possui a maior concentração de partículas devido a proximidade com o feixe incidente. As partículas dissipam-se pelo meio e são fortemente absorvidas na região circular contendo Boro e avançam com maior liberdade após atravessar a mesma. A figura 4.12(b) ilustra a dose de energia liberada na região circular contendo alta concentração de Boro e a dose é maior no lado esquerdo devido a proximidade com a fonte de nêutrons. A dose total é menor compara aos outros experimentos devido a maior distância à fonte. A simulação termina após 36 iterações.

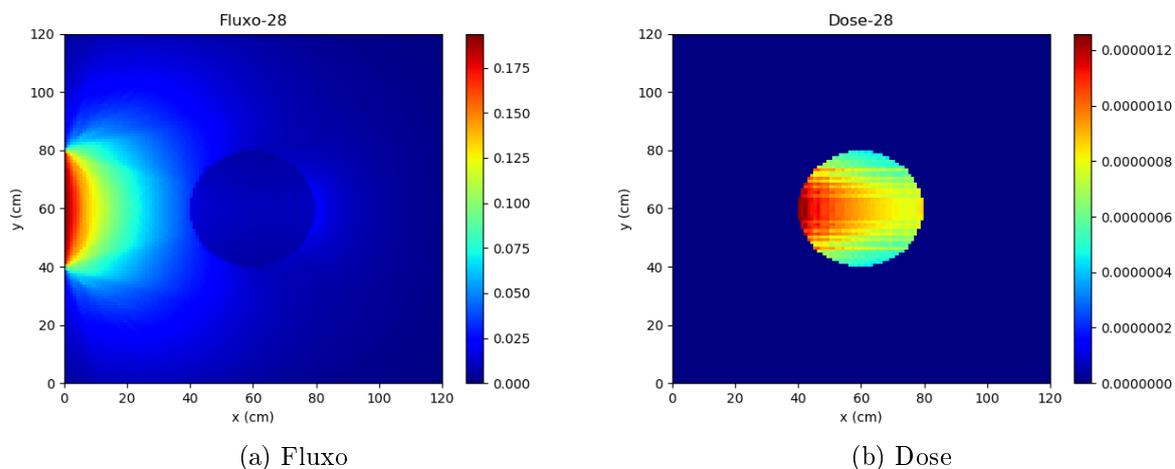


Figura 4.12: Resultado do experimento 3. (a) Fluxo escalar de nêutrons menor na região central devido à absorção. (b) Dose de energia concentrada na face esquerda da região circular (dose total = $9,972 \times 10^{-4}$ Gy).

4.1.2.4 Experimento Numérico 4

Neste experimento, o meio material heterogêneo é representado por seções de choque do tecido sadio ocupando a região externa do domínio e seções de choque contendo Boro na região circular de raio 1cm na posição $(x, y) = (2, 3)$ do domínio. O feixe da fonte fixa possui intensidade $240 N/(cm^2 s)$ e o espalhamento possui grau de anisotropia igual a 3. O fluxo escalar de nêutrons obtido no estado estacionário para o grupo de energia $g=28$ (faixa térmica) é apresentado na figura 4.13(a). A região esquerda da imagem possui a maior concentração de partículas devido a proximidade com o feixe incidente. As partículas dissipam-se pelo meio e são fortemente absorvidas na região circular contendo Boro. Percebem-se fachos de fluxo na saída do círculo que não aparecem na figura 4.12(a). A figura 4.13(b) ilustra a dose de energia liberada na região circular contendo alta concentração de Boro e a dose é maior no lado esquerdo devido a proximidade com a fonte de nêutrons. A dose total é maior compara aos outros experimentos devido a menor distância à fonte. A simulação termina após 39 iterações.

4.1.2.5 Experimento Numérico 5

Neste experimento, o meio material heterogêneo é representado por seções de choque do tecido sadio ocupando a região externa do domínio e seções de choque contendo Boro na região circular de raio 1cm na posição $(x, y) = (1, 3)$ do domínio. O feixe da fonte fixa possui intensidade $240 N/(cm^2 s)$ e o espalhamento possui grau de anisotropia igual

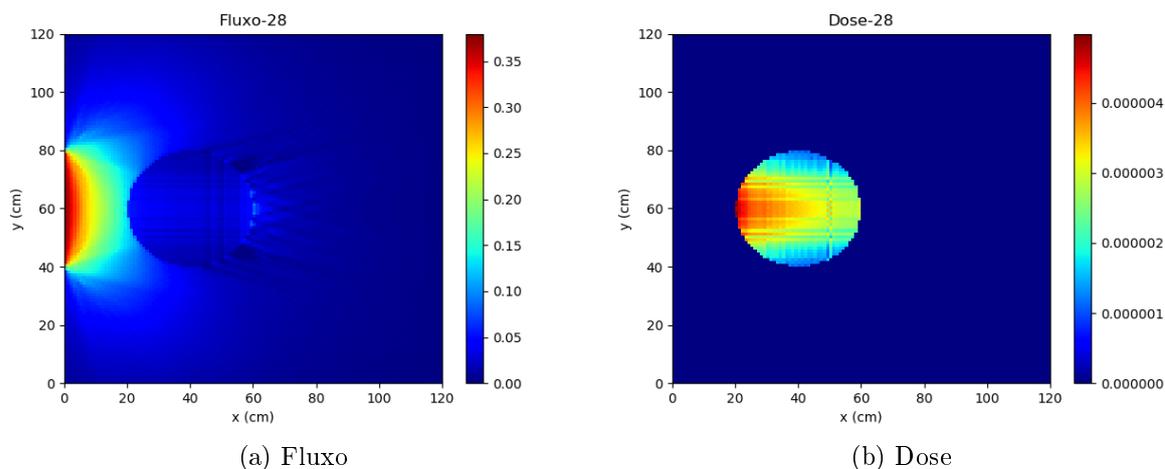


Figura 4.13: Resultado do experimento 4. (a) Fluxo escalar de nêutrons menor na região central devido à absorção. (b) Dose de energia concentrada na face esquerda da região circular (dose total = $3,409 \times 10^{-3}$ Gy).

a 3. O fluxo escalar de nêutrons obtido no estado estacionário para o grupo de energia $g=28$ (faixa térmica) é apresentado na figura 4.14(a). A região esquerda da imagem continua possuindo a maior concentração de partículas mesmo com a proximidade com a região borada. As partículas logo ao saírem da fonte são fortemente absorvidas pela região circular contendo Boro e o fluxo volta a aumentar no lado direito da mesma. Percebem-se fachos de fluxo nas diagonais de saída do círculo que não aparecem na figura 4.12(a). A figura 4.14(b) ilustra a dose de energia liberada na região circular contendo alta concentração de Boro e a dose é maior no lado esquerdo devido a proximidade com a fonte de nêutrons. A dose total é maior compara aos outros experimentos devido a menor distância à fonte. A simulação termina após 45 iterações.

4.1.2.6 Experimento Numérico 6

Neste experimento, o meio material heterogêneo é representado por seções de choque do tecido sadio ocupando a região externa do domínio e seções de choque contendo Boro na região circular de raio 1cm na posição $(x, y) = (3, 1)$ do domínio. O feixe da fonte fixa possui intensidade $240 N/(cm^2 s)$ e o espalhamento possui grau de anisotropia igual a 3. O fluxo escalar de nêutrons obtido no estado estacionário para o grupo de energia $g=28$ (faixa térmica) é apresentado na figura 4.15(a). A região esquerda da imagem possui a maior concentração de partículas devido a proximidade com o feixe incidente. As partículas dissipam-se pelo meio e são fortemente absorvidas na região circular contendo

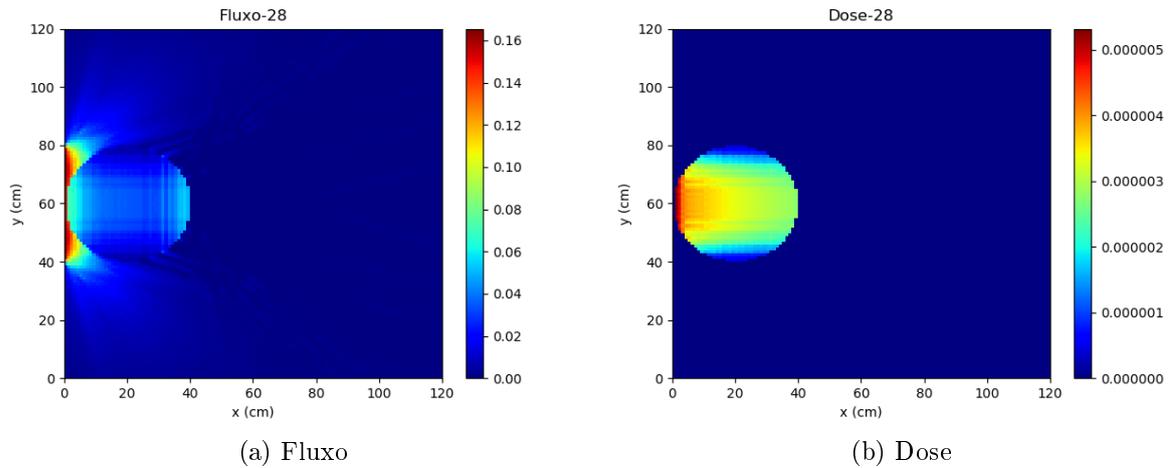


Figura 4.14: Resultado do experimento 5. (a) Fluxo escalar de nêutrons menor na região esquerda devido à absorção. (b) Dose de energia concentrada na face esquerda da região circular (dose total = $3,485 \times 10^{-3}$ Gy).

Boro e avançam com maior liberdade após atravessar a mesma. A figura 4.15(b) ilustra a dose de energia liberada na região circular contendo alta concentração de Boro e a dose é maior no canto superior esquerdo devido a proximidade com a fonte de nêutrons. A dose total é menor compara aos outros experimentos devido a maior distância à fonte. A simulação termina após 45 iterações.

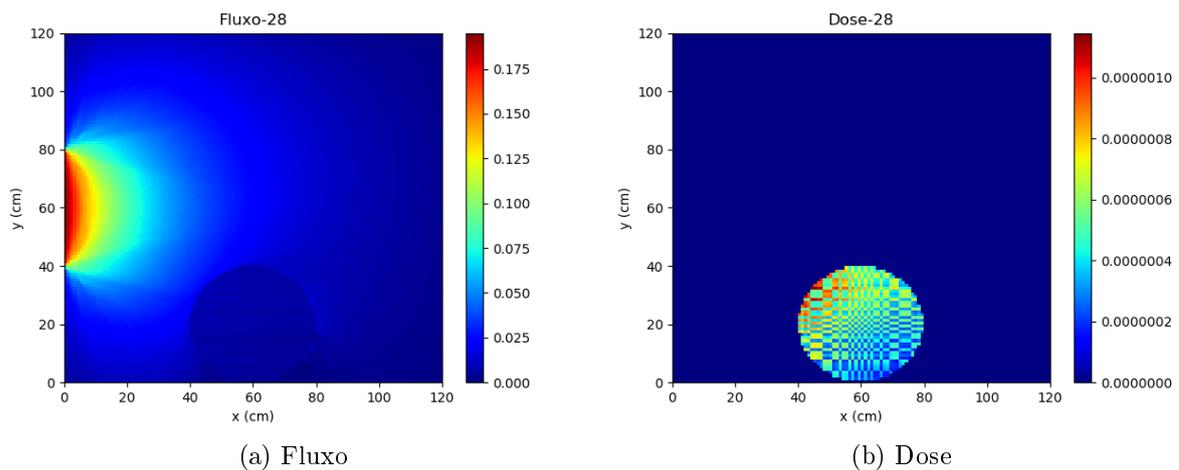


Figura 4.15: Resultado do experimento 6. (a) Fluxo escalar de nêutrons menor na região inferior devido à absorção. (b) Dose de energia concentrada na diagonal superior esquerda da região circular (dose total = $5,989 \times 10^{-4}$ Gy).

Os valores totais de dose depositada devido ao Boro para cada um dos experimentos estão listados na tabela 4.2. A coluna distância se refere a quantidade de centímetros

que separa o centro do tumor ao centro do feixe localizado em ($x = 0$ cm, $y = 3$ cm). A área dos tumores também estão listadas para permitir a distinção entre as formas quadrada e circular das regiões contendo Boro. Pode-se concluir que os maiores valores de dose ocorrem quando o tumor está mais próximo da fonte. Percebe-se também que o crescimento da dose não é linear para aproximações constantes do tumor ao longo do eixo x como pode ser visto nos experimentos 3, 4 e 5.

Tabela 4.2: Doses totais devido ao Boro para cada experimento.

Experimento	Dose (Gy)	Distância(cm)	Área(cm ²)
1	0	N/A	0
2	$1,343 \times 10^{-3}$	3	4
3	$9,972 \times 10^{-4}$	3	3,14
4	$3,409 \times 10^{-3}$	2	3,14
5	$3,485 \times 10^{-3}$	1	3,14
6	$5,989 \times 10^{-4}$	3,6	3,14

Capítulo 5

Conclusões e Trabalhos Futuros

Este trabalho objetivou o desenvolvimento de código computacional para simular o comportamento do fluxo de nêutrons no tecido cerebral, resolvendo numericamente a equação de transporte linear de Boltzmann. Aplicou-se o método determinístico de ordenadas discretas S_N para discretizar a variável angular $\hat{\Omega}$ e o método P_N para aproximar o espalhamento anisotrópico. Utilizou-se o método das diferenças finitas para discretizar a variável espacial \hat{r} e a aproximação DD para obter as equações auxiliares. Para a discretização da energia cinética E das partículas implementou-se o método multigrupo. O código de transporte desenvolvido foi capaz de obter a solução aproximada da equação de transporte e gerou resultados coerentes com o esperado, com regiões de sumidouro de nêutrons nas regiões contendo altas concentrações de Boro e maiores fluências nas demais regiões. A simulação gerou fluxos escalares positivos, devido ao tratamento aplicado na função *Scalar*. A dose correspondente ao fluxo comportou-se como esperado, sendo mais concentrada nas regiões contendo Boro.

5.1 Conclusões

- A simulação pelo método S_N apresentou convergência em um número de iterações pequeno comparado aos métodos estocásticos.
- O código de transporte em C++ simulou em menor tempo, comparado ao código de referência em Fortran 77.
- A vetorização dos *loops* das funções *Sweep* e *Scatter* acelerou o processamento.
- O código implementado possibilitou uma configuração mais flexível, facilitando a importação dos arquivos de seção de choque no formato CSV.

- O código passou a suportar a quadratura S16, que especifica 144 direções.
- A simulação gerou fluxos escalares positivos, devido a alteração da função *Scalar*.
- O fluxo escalar estacionário comportou-se como esperado, possuindo menor concentração nas regiões contento Boro.
- A dose correspondente ao fluxo comportou-se como esperado, apresentando maior concentração nas regiões contento Boro.

5.2 Trabalhos Futuros

Para trabalhos futuros, podem ser realizadas as seguintes melhorias:

- Extensão do modelo para o domínio 3D, adaptando o método de ordenadas discretas para contemplar a dimensão z na equação 3.43 de modo a obter a equação 5.1:

$$\psi_{mg}^{ij} = (\epsilon_x \psi_{Vmg}^{ij} + \epsilon_y \psi_{Hmg}^{ij} + \epsilon_z \psi_{Fmg}^{ij} + \psi_{ext,mg}^{ij} + \psi_{s,mg}^{ij}) / (\epsilon_x + \epsilon_y + \epsilon_z + \Sigma_{t,g}^{ij}) \quad (5.1)$$

- Utilização de malhas não estruturadas capazes de tratar geometrias irregulares e adoção de outro método numérico determinístico como o Método das Características [16].
- Utilização do Método das Características (MOC) como alternativa de método determinístico capaz de tratar malhas não estruturadas [16].
- Exploração de outras técnicas de paralelismo aplicadas em ambiente computacional heterogêneo contendo CPU's e GPU's de modo a melhor explorar os recursos computacionais de CPU's e GPU's.
- Utilização de outros métodos numéricos baseados em Monte Carlo para a realização de simulações comparativas com o método determinístico desenvolvido.
- Aplicação da adimensionalização da equação de transporte e adequar o algoritmo de modo a possibilitar o aumento das resoluções espacial e angular e produzir resultados quantitativamente semelhantes.

APÊNDICE A - Fluxo Espalhado

O fluxo espalhado da equação 3.2 corresponde ao ganho de fluxo na direção Ω e grupo de energia E a partir das perdas por espalhamento dos demais fluxos nas direção Ω' e energias E' . [1] O fluxo espalhado é dado por (3.3):

$$\psi_s(\vec{r}, E, \hat{\Omega}) = \int_0^\infty \int_{4\pi} \Sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega}) \psi(\vec{r}, E', \hat{\Omega}') d\Omega' dE' \quad (3.3)$$

A seção de choque diferencial de espalhamento Σ_s sofre duas simplificações para viabilizar seu desenvolvimento computacional:

- **Simplificação de Σ_s no parâmetro $E' \rightarrow E$:**

A probabilidade de espalhamento entre energias é discretizada para pares de grupos de energia, logo $\Sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega}) \approx \Sigma_s^{g'g}(\vec{r}, \hat{\Omega}' \cdot \hat{\Omega})$ que corresponde à probabilidade do fluxo do grupo g' espalhar para o grupo g , das partículas provenientes de $\hat{\Omega}'$ para $\hat{\Omega}$. O fluxo espalhado após a simplificação de $E' \rightarrow E$ para $g'g$ é:

$$\psi_{s,g}(\vec{r}, \hat{\Omega}) = \sum_{g'=0}^G \int_{4\pi} \Sigma_s^{g'g}(\vec{r}, \hat{\Omega}' \cdot \hat{\Omega}) \psi_{g'}(\vec{r}, \hat{\Omega}') d\Omega' \quad (A.1)$$

- **Simplificação de Σ_s no parâmetro $\hat{\Omega}' \cdot \hat{\Omega}$:**

A probabilidade de espalhamento entre direções é aproximada pela expansão da série de polinômios de Legendre:

$$\Sigma_s^{g'g}(\vec{r}, \mu) \approx \sum_{l=0}^L \frac{2l+1}{4\pi} \Sigma_{sl}^{g'g}(\vec{r}) P_l(\mu) \quad (A.2)$$

onde $\mu = \hat{\Omega}' \cdot \hat{\Omega} = \cos(\theta)$ é o cosseno do ângulo entre os vetores $\hat{\Omega}'$ e $\hat{\Omega}$ e $\Sigma_{sl}^{g'g}$ é o momento de seção de choque de grau l associado ao polinômio P_l . O grau L controla a qualidade da aproximação. Os valores de $\Sigma_{sl}^{g'g}$ são obtidos a partir de

valores experimentais conhecidos de $\Sigma_s^{g'g}$, através de:

$$\Sigma_{sl}^{g'g}(\vec{r}) = 2\pi \int_{-1}^1 \Sigma_s^{g'g}(\vec{r}, \mu) P_l(\mu) d\mu \quad (\text{A.3})$$

Combinando as equações A.2 e A.1 tem-se:

$$\psi_{s,g}(\vec{r}, \hat{\Omega}) = \sum_{g'=0}^G \int_{4\pi} \sum_{l=0}^L \frac{2l+1}{4\pi} \Sigma_{sl}^{g'g}(\vec{r}) P_l(\mu) \psi_{g'}(\vec{r}, \hat{\Omega}') d\Omega' \quad (\text{A.4})$$

O fluxo espalhado no grupo g devido a um dado grupo g' é dado por:

$$\psi_s^{g'g}(\vec{r}, \hat{\Omega}) = \int_{4\pi} \sum_{l=0}^L \frac{2l+1}{4\pi} \Sigma_{sl}^{g'g}(\vec{r}) P_l(\mu) \psi_{g'}(\vec{r}, \hat{\Omega}') d\Omega' \quad (\text{A.5})$$

Removendo do integrando os termos que não dependem de $\hat{\Omega}'$, tem-se:

$$\psi_s^{g'g}(\vec{r}, \hat{\Omega}) = \sum_{l=0}^L \frac{2l+1}{4\pi} \Sigma_{sl}^{g'g}(\vec{r}) \int_{4\pi} P_l(\mu) \psi_{g'}(\vec{r}, \hat{\Omega}') d\hat{\Omega}' \quad (\text{A.6})$$

O termo de integração esférica é discretizado via quadratura contendo M direções prescritas $\hat{\Omega}_n$ e M pesos associados ω_n (A.7):

$$\frac{1}{4\pi} \int_{4\pi} f(\hat{\Omega}) d\hat{\Omega} \approx \frac{1}{4} \sum_{n=1}^M \omega_n f(\hat{\Omega}_n) \quad (\text{A.7})$$

Aplicando a aproximação S_N (A.7) na expressão do fluxo espalhado (A.6) tem-se o fluxo espalhado discreto (A.8):

$$\psi_{s,m}^{g'g}(\vec{r}) = \sum_{l=0}^L \frac{2l+1}{4} \Sigma_{sl}^{g'g}(\vec{r}) \sum_{n=1}^M \omega_n P_l(\mu_n) \psi_{ng'}(\vec{r}) \quad (\text{A.8})$$

Nas demais seções são apresentados os cálculos dos fluxos espalhados para $L = 3$, de modo a obter $\psi_{s0,m}$, $\psi_{s1,m}$, $\psi_{s2,m}$ e $\psi_{s3,m}$.

A.1 Cálculo dos Momentos de Fluxo Espalhado

A.1.1 Expressão do Polinômio de Legendre

O polinômio de Legendre de grau $l \geq 0$ é aproximado pela expansão de produtos de harmônicos esféricos reais [15]:

$$P_l(\mu) = \frac{4\pi}{2l+1} \sum_{m=-l}^l Y_l^m(\hat{\Omega}) Y_l^m(\hat{\Omega}') \quad (\text{A.9})$$

onde

- $\hat{\Omega}$ é o vetor unitário de coordenadas cartesianas (η, ξ, μ) e esféricas $(\text{sen}\theta \cos\phi, \text{sen}\theta \text{sen}\phi, \cos\theta)$ com ângulo polar $0 < \theta < \pi$ e azimutal $0 < \phi < 2\pi$.
- $Y_l^m(\hat{\Omega})$ é o esférico harmônico real de grau l e ordem m na direção $\hat{\Omega}$.
- $Y_l^m(\hat{\Omega}')$ é o esférico harmônico real de grau l e ordem m na direção $\hat{\Omega}'$.

Os harmônicos esféricos reais $Y_l^m(\hat{\Omega})$ são (A.10):

$$Y_l^m(\hat{\Omega}) = \begin{cases} \sqrt{2} K_l^m P_l^m(\mu) \cos(m\phi), & m > 0 \\ \sqrt{2} K_l^m P_l^m(\mu) \text{sen}(m\phi), & m < 0 \\ K_l^0 P_l^0(\mu), & m = 0 \end{cases} \quad (\text{A.10})$$

onde

- K_l^m é a constante de normalização dada por A.11:

$$K_l^m = \left[\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!} \right]^{1/2} \quad (\text{A.11})$$

- P_l^m é o polinômio de Legendre associado de grau l e ordem m . Este polinômio é calculado segundo as 3 regras A.12, A.13, A.14:

$$P_l^m(\mu) = \frac{\mu(2l-1)}{l-m} P_{l-1}^m - \frac{(l+m-1)}{l-m} P_{l-2}^m \quad \text{para } l \geq 2 \quad (\text{A.12})$$

$$P_m^m(\mu) = -1^m (2m-1)!! (1-\mu^2)^{m/2} \quad \text{com } P_0^0 = 1 \quad (\text{A.13})$$

$$P_{m+1}^m(\mu) = \mu(2m+1) P_m^m \quad (\text{A.14})$$

Na equação A.13 o símbolo !! denota o fatorial dos termos ímpares entre 1 e $(2m-1)$, de modo a produzir a sequência $1 \times 3 \times 5 \times \dots \times (2m-1)$ onde assume-se $(-1)!! = 1$.

Nas demais seções são apresentadas os cálculos dos polinômios P_0 , P_1 , P_2 e P_3 . Para simplificar a notação, adotam-se $P_l = P_l(\mu)$, $P_l^m = P_l^m(\mu)$, $Y_l^m = Y_l^m(\hat{\Omega})$, $\psi_{sl} = \psi_{sl}^{g'g}(\vec{r}, \hat{\Omega})$.

A.1.2 Momento de Fluxo de Grau 0 (ψ_{s0})

Para o cálculo do momento de fluxo espalhado ψ_{s0} segue-se a sequência:

Cálculo do polinômio de Legendre P_0 :

$$\begin{aligned} P_0 &= \frac{4\pi}{2 \cdot 0 + 1} Y_0^0 Y_0^0 = 4\pi (K_0^0 P_0^0)^{1/2} (K_0^0 P_0^0)^{1/2} \\ &= 4\pi \left(\frac{(2 \cdot 0 + 1) 0!}{4\pi 0!} \right)^{1/2} \left(\frac{(2 \cdot 0 + 1) 0!}{4\pi 0!} \right)^{1/2} = 4\pi \left(\frac{1}{4\pi} \right)^{1/2} \left(\frac{1}{4\pi} \right)^{1/2} \\ &= 1 \end{aligned} \quad (\text{A.15})$$

Cálculo do momento de fluxo espalhado ψ_{s0} :

$$\psi_{s0} = \frac{1}{4\pi} \Sigma_{s0} \int_{4\pi} \psi' P_0 d\hat{\Omega}' = \frac{1}{4\pi} \Sigma_{s0} \int_{4\pi} \psi' d\hat{\Omega}' \quad (\text{A.16})$$

Cálculo do momento de fluxo espalhado discreto $\psi_{s0,m}$:

$$\psi_{s0,m} = \frac{1}{4} \Sigma_{s0} \left(\sum_{n=1}^M \omega_n \psi_n \right) = \psi_{s0,m} = \Sigma_{s0} \left(\frac{1}{4} \sum_{n=1}^M \omega_n \psi_n \right) \quad (\text{A.17})$$

Identificação dos coeficientes externo K e interno C de $\psi_{s0,m}$ dos algoritmos 12 e 11:

$$K_{0,m} = \frac{1}{4} = 0.25, \quad C_{0,n} = \omega_n \quad (\text{A.18})$$

A.1.3 Momento de Fluxo de Grau 1 (ψ_{s1})

Para o cálculo do momento de fluxo espalhado ψ_{s1} segue-se a sequência:

Cálculo dos polinômios de Legendre associados P_1^0 (A.14) e P_1^1 (A.13):

$$\begin{aligned} P_1^0 &= \mu(2 \cdot 0 + 1) = \mu \\ P_1^1 &= -1^1(2 \cdot 1 - 1)(1 - \mu^2)^{1/2} = -\text{sen}\theta \end{aligned} \quad (\text{A.19})$$

Cálculo dos harmônicos esféricos Y_1^0 , Y_1^1 e Y_1^{-1} :

$$\begin{aligned} Y_1^0 &= K_1^0 P_1^0 = \sqrt{3/(4\pi)} \mu \\ Y_1^1 &= \sqrt{2}\sqrt{3/(8\pi)} \cos \phi P_1^1 = -\sqrt{3/(4\pi)} \eta \\ Y_1^{-1} &= \sqrt{2}\sqrt{3/(8\pi)} \text{sen} \phi P_1^1 = -\sqrt{3/(4\pi)} \xi \end{aligned} \quad (\text{A.20})$$

Cálculo do produto entre harmônicos esféricos :

$$\begin{aligned} Y_1^0 Y_1'^0 &= \sqrt{3/(4\pi)} \mu \sqrt{3/(4\pi)} \mu' = 3/(4\pi) \mu \mu' \\ Y_1^1 Y_1'^1 &= \sqrt{3/(4\pi)} \eta \sqrt{3/(4\pi)} \eta' = 3/(4\pi) \eta \eta' \\ Y_1^{-1} Y_1'^{-1} &= \sqrt{3/(4\pi)} \xi \sqrt{3/(4\pi)} \xi' = 3/(4\pi) \xi \xi' \end{aligned} \quad (\text{A.21})$$

Cálculo do polinômio de Legendre P_1 :

$$\begin{aligned} P_1 &= \frac{4\pi}{3} (Y_1^0 Y_1'^0 + Y_1^1 Y_1'^1 + Y_1^{-1} Y_1'^{-1}) \\ &= \frac{4\pi}{3} \left(\frac{3}{4\pi} \mu \mu' + \frac{3}{4\pi} \eta \eta' + \frac{3}{4\pi} \xi \xi' \right) \end{aligned} \quad (\text{A.22})$$

Cálculo do momento de fluxo espalhado ψ_{s1} :

$$\begin{aligned} \psi_{s1} &= \frac{3}{4\pi} \sum_{s1} \int_{4\pi} \psi' P_1 d\Omega' = \frac{3}{4\pi} \sum_{s1} \langle \psi' P_1 \rangle \\ &= \frac{3}{4\pi} \sum_{s1} \langle \psi' \mu \mu' + \psi' \eta \eta' + \psi' \xi \xi' \rangle \end{aligned} \quad (\text{A.23})$$

Eliminação do termo ξ' pela aplicação da propriedade B.1:

$$\psi_{s1} = \frac{3}{4\pi} \sum_{s1} \langle \psi' \mu \mu' + \psi' \eta \eta' \rangle = \frac{3}{4\pi} \sum_{s1} (\mu \langle \mu' \psi' \rangle + \eta \langle \eta' \psi' \rangle) \quad (\text{A.24})$$

Cálculo do momento de fluxo espalhado discreto $\psi_{s1,m}$:

$$\begin{aligned} \psi_{s1,m} &= \frac{3}{4} \sum_{s1} \left(\mu_m \sum_{n=1}^M \mu_n \omega_n \psi_n + \eta_m \sum_{n=1}^M \eta_n \omega_n \psi_n \right) \\ &= \sum_{s1} \left(0.75 \mu_m \sum_{n=1}^M \mu_n \omega_n \psi_n + 0.75 \eta_m \sum_{n=1}^M \eta_n \omega_n \psi_n \right) \end{aligned} \quad (\text{A.25})$$

Identificação dos coeficientes externo K e interno C de $\psi_{s1,m}$ dos algoritmos 12 e 11:

$$\begin{aligned} K_{1,m} &= 0.75 \mu_m \quad , \quad C_{1,n} = \mu_n \omega_n \\ K_{2,m} &= 0.75 \eta_m \quad , \quad C_{2,n} = \eta_n \omega_n \end{aligned} \quad (\text{A.26})$$

A.1.4 Momento de Fluxo de Grau 2 (ψ_{s2})

Para o cálculo do momento de fluxo espalhado ψ_{s2} segue-se a sequência:

Cálculo dos polinômios de Legendre associados P_2^0 (A.12), P_2^1 (A.14) e P_2^2 (A.13):

$$\begin{aligned} P_2^0 &= \frac{\mu(2 \cdot 2 - 1)}{2 - 0} P_1^0 - \frac{2 + 0 - 1}{2 - 0} P_0^0 = \frac{3}{2} \mu \mu - \frac{1}{2} \cdot 1 = \frac{3}{2} \mu^2 - \frac{1}{2} \\ P_2^1 &= \mu(2 \cdot 1 + 1) P_1^1 = \mu \cdot 3(-\sin\theta) = -3\mu \sin\theta \\ P_2^2 &= -1^2(2 \cdot 2 - 1)!!(1 - \mu^2)^{2/2} = 3 \cdot 1(1 - \mu^2) = 3 \sin^2\theta \end{aligned} \quad (\text{A.27})$$

Cálculo dos harmônicos esféricos Y_2^0 , Y_2^1 , Y_2^{-1} , Y_2^2 e Y_2^{-2} :

$$\begin{aligned} Y_2^0 &= K_2^0 P_2^0 = \left(\frac{5}{4\pi}\right)^{1/2} \left(\frac{3}{2}\mu^2 - \frac{1}{2}\right) = \left(\frac{5}{16\pi}\right)^{1/2} (3\mu^2 - 1) \\ Y_2^1 &= \sqrt{2} K_2^1 \cos \phi P_2^1 = \sqrt{2} \sqrt{5/24\pi} \cos \phi (-3\mu \sin\theta) = -\sqrt{15/4\pi} \mu \eta \\ Y_2^{-1} &= \sqrt{2} K_2^1 \sin \phi P_2^1 = \sqrt{2} \sqrt{5/24\pi} \sin \phi (-3\mu \sin\theta) = -\sqrt{15/4\pi} \mu \xi \\ Y_2^2 &= \sqrt{2} K_2^2 \cos 2\phi P_2^2 = \sqrt{2} \sqrt{5/96\pi} \cos 2\phi \cdot 3 \sin^2\theta \\ &= \sqrt{15/16\pi} (\cos^2 \phi - \sin^2 \phi) \sin^2\theta = \sqrt{15/16\pi} (\eta^2 - \xi^2) \\ Y_2^{-2} &= \sqrt{2} K_2^2 \sin 2\phi P_2^2 = \sqrt{2} \sqrt{5/96\pi} (2 \sin \phi \cos \phi) 3 \sin^2\theta = \sqrt{15/4\pi} \eta \xi \end{aligned} \quad (\text{A.28})$$

Cálculo do produto entre harmônicos esféricos :

$$\begin{aligned} Y_2^0 Y_2'^0 &= 5/(16\pi)(3\mu^2 - 1)(3\mu'^2 - 1) \\ Y_2^1 Y_2'^1 &= 15/(4\pi)\mu\eta\mu'\eta' \\ Y_2^{-1} Y_2'^{-1} &= 15/(4\pi)\mu\xi\mu'\xi' \\ Y_2^2 Y_2'^2 &= 15/(16\pi)(\eta^2 - \xi^2)(\eta'^2 - \xi'^2) \\ &= 15/(16\pi)(2\eta^2 + \mu^2 - 1)(2\eta'^2 + \mu'^2 - 1) \\ Y_2^{-2} Y_2'^{-2} &= 15/(4\pi)\eta\xi\eta'\xi' \end{aligned} \quad (\text{A.29})$$

Cálculo do polinômio de Legendre P_2 :

$$\begin{aligned} P_2 &= \frac{4\pi}{5} (Y_2^0 Y_2'^0 + Y_2^1 Y_2'^1 + Y_2^{-1} Y_2'^{-1} + Y_2^2 Y_2'^2 + Y_2^{-2} Y_2'^{-2}) \\ &= \frac{4\pi}{5} \left(\left(\frac{5}{4\pi} - \frac{15}{8\pi}\mu^2 - \frac{15}{8\pi}\eta^2 \right) \frac{15}{4\pi} \mu\eta\mu'\eta' + \right. \\ &\quad \left. \left(\frac{15}{4\pi}\mu^2 + \frac{15}{8\pi}\eta^2 - \frac{15}{8\pi} \right) \mu'^2 + \left(\frac{15}{4\pi}\eta^2 + \frac{15}{8\pi}\mu^2 - \frac{15}{8\pi} \right) \eta'^2 \right) \\ &= \frac{4\pi}{5} (K_3 + K_4\mu'\eta' + K_5\mu'^2 + K_6\eta'^2) \end{aligned} \quad (\text{A.30})$$

Cálculo do momento de fluxo espalhado ψ_{s2} :

$$\begin{aligned}\psi_{s2} &= \frac{5}{4\pi} \Sigma_{s2} \int_{4\pi} \psi' P_2(\hat{\Omega} \cdot \hat{\Omega}') d\Omega' = \frac{5}{4\pi} \Sigma_{s2} \langle \psi' P_2 \rangle \\ &= \Sigma_{s2} \left(K_3 \langle \psi' \rangle + K_4 \langle \mu' \eta' \psi' \rangle + K_5 \langle \mu'^2 \psi' \rangle + K_6 \langle \eta'^2 \psi' \rangle \right)\end{aligned}\quad (\text{A.31})$$

Cálculo do momento de fluxo espalhado discreto $\psi_{s2,m}$:

$$\begin{aligned}\psi_{s2,m} &= \Sigma_{s2} \left(K_{3,m} \sum_{n=1}^M \omega_n \psi_n + K_{4,m} \sum_{n=1}^M \mu_n \eta_n \omega_n + \right. \\ &\quad \left. K_{5,m} \sum_{n=1}^M \mu_n^2 \omega_n \psi_n + K_{6,m} \sum_{n=1}^M \eta_n^2 \omega_n \right)\end{aligned}\quad (\text{A.32})$$

Identificação dos coeficientes externo K e interno C de $\psi_{s2,m}$ dos algoritmos 12 e 11:

$$\begin{aligned}K_{3,m} &= \left(\frac{5}{4} - \frac{15}{8} \mu_m^2 - \frac{15}{8} \eta_m^2 \right) \quad , \quad C_{3,n} = \omega_n \\ K_{4,m} &= \left(\frac{15}{4} \mu_m \eta_m \right) \quad , \quad C_{4,n} = \mu_n \eta_n \omega_n \\ K_{5,m} &= \left(\frac{15}{4} \mu_m^2 + \frac{15}{8} \eta_m^2 - \frac{15}{8} \right) \quad , \quad C_{5,n} = \mu_n^2 \omega_n \\ K_{6,m} &= \left(\frac{15}{4} \eta_m^2 + \frac{15}{8} \mu_m^2 - \frac{15}{8} \right) \quad , \quad C_{6,n} = \eta_n^2 \omega_n\end{aligned}\quad (\text{A.33})$$

A.1.5 Momento de Fluxo de Grau 3 (ψ_{s3})

Para o cálculo do momento de fluxo espalhado ψ_{s3} segue-se a sequência:

Cálculo dos polinômios de Legendre associados seguindo a ordem P_3^0 (A.12), P_3^1 (A.12), P_3^2 (A.13) e P_3^3 (A.14):

$$\begin{aligned}P_3^0 &= \frac{\mu(2.3-1)}{3-0} P_2^0 - \frac{(3+0-1)}{3-0} P_1^0 = \frac{5}{3\mu} \left(\frac{3}{2} \mu^2 - \frac{1}{2} \right) - \frac{3}{2} \mu = \frac{5}{2} \mu^3 - \frac{3}{2} \mu \\ P_3^1 &= \frac{\mu(2.3-1)}{3-1} P_2^1 - \frac{(3+1-1)}{3-1} P_1^1 \\ &= \frac{5}{2} \mu (-3\mu \text{sen}\theta) + \frac{3}{2} \text{sen}\theta = \left(-\frac{15}{2} \mu^2 + \frac{3}{2} \right) \text{sen}\theta \\ P_3^2 &= \mu(2.2+1) P_2^2 = 5\mu(3 \text{sen}^2\theta) = 15\mu \text{sen}^2\theta \\ P_3^3 &= -1^3(2.3-1)!!(1-\mu^2)^{3/2} = -(5.3.1)(\text{sen}^2\theta)^{3/2} = -15 \text{sen}^3\theta\end{aligned}\quad (\text{A.34})$$

Cálculo dos harmônicos esféricos Y_3^0 , Y_3^1 , Y_3^{-1} , Y_3^2 , Y_3^{-2} , Y_3^3 e Y_3^{-3} :

$$\begin{aligned}
Y_3^0 &= K_3^0 P_3^0 = \sqrt{\frac{7}{4\pi}} \left(\frac{5}{2}\mu^3 - \frac{3}{2}\mu \right) = \sqrt{\frac{7}{16\pi}} (5\mu^3 - 3\mu) \\
Y_3^1 &= \sqrt{2} K_3^1 \cos \phi P_3^1 = \sqrt{2} \sqrt{\frac{7}{48\pi}} \left(-\frac{15}{2}\mu^2 + \frac{3}{2} \right) \eta \\
&= \sqrt{\frac{7}{96\pi}} (-15\mu^2\eta + 3\eta) = \sqrt{\frac{21}{32\pi}} (-5\mu^2\eta + \eta) \\
Y_3^{-1} &= \sqrt{2} K_3^1 \sin \phi P_3^1 = \sqrt{2} \sqrt{\frac{7}{48\pi}} \left(-\frac{15}{2}\mu^2 + \frac{3}{2} \right) \xi \\
&= \sqrt{\frac{7}{96\pi}} (-15\mu^2\xi + 3\xi) = \sqrt{\frac{21}{32\pi}} (-5\mu^2\xi + \xi) \\
Y_3^2 &= \sqrt{2} K_3^2 \cos 2\phi P_3^2 = \sqrt{2} \sqrt{\frac{7}{480\pi}} (\cos^2\phi - \sin^2\phi) 15\mu \sin^2\theta \\
&= \sqrt{\frac{105}{16\pi}} \mu (\eta^2 - \xi^2) \\
Y_3^{-2} &= \sqrt{2} K_3^2 \sin 2\phi P_3^2 = \sqrt{2} \sqrt{\frac{7}{480\pi}} (2 \sin\phi \cos\phi) 15\mu \sin^2\theta \\
&= \sqrt{\frac{105}{4\pi}} \mu \eta \xi \\
Y_3^3 &= \sqrt{2} K_3^3 P_3^3 \cos 3\phi = -\sqrt{2} \sqrt{\frac{7}{2880\pi}} 15 \sin^3\theta (\cos^3\phi - 3 \sin^2\phi \cos\phi) \\
&= -\sqrt{\frac{35}{32\pi}} (\eta^3 - 3\eta\xi^2) = \sqrt{\frac{35}{32\pi}} (3\eta\xi^2 - \eta^3) \\
Y_3^{-3} &= \sqrt{2} K_3^3 P_3^3 \sin 3\phi = -\sqrt{2} \sqrt{\frac{7}{2880\pi}} 15 \sin^3\theta (3 \sin\phi \cos^2\phi - \sin^3\phi) \\
&= -\sqrt{\frac{35}{32\pi}} (-\xi^3 + 3\eta^2\xi) = \sqrt{\frac{35}{32\pi}} (\xi^3 - 3\eta^2\xi)
\end{aligned} \tag{A.35}$$

Cálculo do produto entre harmônicos esféricos :

$$\begin{aligned}
Y_3^0 Y_3'^0 &= 7/(16\pi)(5\mu^3 - 3\mu)(5\mu'^3 - 3\mu') \\
Y_3^1 Y_3'^1 &= 21/(32\pi)(-5\mu^2\eta + \eta)(-5\mu'^2\eta' + \eta') \\
Y_3^{-1} Y_3'^{-1} &= 21/(32\pi)(-5\mu^2\xi + \xi)(-5\mu'^2\xi' + \xi') \\
Y_3^2 Y_3'^2 &= 105/(16\pi)\mu(\eta^2 - \xi^2)\mu'(\eta'^2 - \xi'^2) \\
&= 105/(16\pi)(2\mu\eta^2 + \mu^3 - \mu)(2\mu'\eta'^2 + \mu'^3 - \mu') \\
Y_3^{-2} Y_3'^{-2} &= 105/(4\pi)\mu\eta\xi\mu'\eta'\xi' \\
Y_3^3 Y_3'^3 &= 35/(32\pi)(3\eta\xi^2 - \eta^3)(3\eta'\xi'^2 - \eta'^3) \\
&= 35/(32\pi)(3\eta - 3\mu^2\eta - 4\eta^3)(3\eta' - 3\mu'^2\eta' - 4\eta'^3) \\
Y_3^{-3} Y_3'^{-3} &= 35/(32\pi)(\xi^3 - 3\eta^2\xi)(\xi'^3 - 3\eta'^2\xi')
\end{aligned} \tag{A.36}$$

Cálculo do polinômio de Legendre P_3 :

$$\begin{aligned}
P_3 &= \frac{4\pi}{7} \left(Y_3^0 Y_3'^0 + Y_3^1 Y_3'^1 + Y_3^{-1} Y_3'^{-1} + Y_3^2 Y_3'^2 + Y_3^{-2} Y_3'^{-2} + \right. \\
&\quad \left. Y_3^3 Y_3'^3 + Y_3^{-3} Y_3'^{-3} \right) \\
&= \frac{4\pi}{7} \left(\left(\frac{21}{2}\mu - \frac{105}{8}\mu^3 - \frac{105}{8}\mu\eta^2 \right) \mu' + \left(\frac{21}{2}\eta - \frac{105}{8}\eta^3 - \frac{105}{8}\eta\mu^2 \right) \eta' + \right. \\
&\quad \left(-\frac{105}{8}\eta + \frac{105}{8}\eta^3 + \frac{105}{4}\mu^2\eta \right) \mu'^2\eta' + \left(-\frac{105}{8}\mu + \frac{105}{8}\mu^3 + \frac{105}{4}\eta^2\mu \right) \eta'^2\mu' \\
&\quad \left. \left(-\frac{105}{8}\mu + \frac{35}{2}\mu^3 + \frac{105}{8}\mu\eta^2 \right) \mu'^3 + \left(-\frac{105}{8}\eta + \frac{35}{2}\eta^3 + \frac{105}{8}\eta\mu^2 \right) \eta'^3 \right) \\
&= \frac{4\pi}{7} \left(K_7\mu' + K_8\eta' + K_9\mu'^2\eta' + K_{10}\eta'^2\mu' + K_{11}\mu'^3 + K_{12}\eta'^3 \right)
\end{aligned} \tag{A.37}$$

Cálculo do momento de fluxo espalhado ψ_{s3} :

$$\begin{aligned}
\psi_{s3} &= \frac{7}{4\pi} \Sigma_{s3} \int_{4\pi} \psi' P_3 d\Omega' = \frac{7}{4\pi} \Sigma_{s3} \langle \psi' P_3 \rangle \\
&= \Sigma_{s3} \left(K_7 \langle \mu' \psi' \rangle + K_8 \langle \eta' \psi' \rangle + K_9 \langle \mu'^2 \eta' \psi' \rangle + \right. \\
&\quad \left. K_{10} \langle \eta'^2 \mu' \psi' \rangle + K_{11} \langle \mu'^3 \psi' \rangle + K_{12} \langle \eta'^3 \psi' \rangle \right)
\end{aligned} \tag{A.38}$$

Cálculo do momento de fluxo espalhado discreto $\psi_{s3,m}$:

$$\begin{aligned}
\psi_{s3,m} &= \Sigma_{s3} \left(K_{7,m} \sum_{n=1}^M \mu_n \omega_n \psi_n + K_{8,m} \sum_{n=1}^M \eta_n \omega_n + K_{9,m} \sum_{n=1}^M \mu_n^2 \eta_n \omega_n + \right. \\
&\quad \left. K_{10,m} \sum_{n=1}^M \eta_n^2 \mu_n \omega_n \psi_n + K_{11,m} \sum_{n=1}^M \mu_n^3 \omega_n + K_{12,m} \sum_{n=1}^M \eta_n^3 \omega_n \right)
\end{aligned} \tag{A.39}$$

Identificação dos coeficientes externo K e interno C de $\psi_{s3,m}$ dos algoritmos 12 e 11:

$$\begin{aligned}
K_{7,m} &= \left(\frac{21}{2}\mu_m - \frac{105}{8}\mu_m^3 - \frac{105}{8}\mu_m\eta_m^2 \right) & , & \quad C_{7,n} = \mu_n \omega_n \\
K_{8,m} &= \left(\frac{21}{2}\eta_m - \frac{105}{8}\eta_m^3 - \frac{105}{8}\eta_m\mu_m^2 \right) & , & \quad C_{8,n} = \eta_n \omega_n \\
K_{9,m} &= \left(-\frac{105}{8}\eta_m + \frac{105}{8}\eta_m^3 + \frac{105}{4}\mu_m^2\eta_m \right) & , & \quad C_{9,n} = \mu_n^2 \eta_n \omega_n \\
K_{10,m} &= \left(-\frac{105}{8}\mu_m + \frac{105}{8}\mu_m^3 + \frac{105}{4}\eta_m^2\mu_m \right) & , & \quad C_{10,n} = \eta_n^2 \mu_n \omega_n \\
K_{11,m} &= \left(-\frac{105}{8}\mu_m + \frac{35}{2}\mu_m^3 + \frac{105}{8}\mu_m\eta_m^2 \right) & , & \quad C_{11,n} = \mu_n^3 \omega_n \\
K_{12,m} &= \left(-\frac{105}{8}\eta_m + \frac{35}{2}\eta_m^3 + \frac{105}{8}\eta_m\mu_m^2 \right) & , & \quad C_{12,n} = \eta_n^3 \omega_n
\end{aligned} \tag{A.40}$$

APÊNDICE B - Propriedades dos Harmônicos Esféricos 2D

B.0.1 Fluxo Nulo na Direção Perpendicular ao Plano

Segundo [1] para o caso bidimensional, onde o espalhamento ocorre somente nas direções vertical μ e horizontal η , tem-se:

$$\int_{4\pi} \psi(\hat{\Omega}) \mu^k \eta^m \xi^n d\hat{\Omega} = 0 \quad \text{para } n \text{ ímpar} \quad (\text{B.1})$$

B.0.2 Direção Unitária de Espalhamento

O vetor $\hat{\Omega} = (\mu, \eta, \xi)$ é unitário, logo:

$$\mu^2 + \eta^2 + \xi^2 = 1 \quad (\text{B.2})$$

A expressão de ξ^2 em termos de μ^2 e η^2 é dada por:

$$\xi^2 = 1 - \mu^2 - \eta^2 \quad (\text{B.3})$$

A expressão de $\eta^2 - \xi^2$ em termos de μ^2 e η^2 é dada por:

$$\eta^2 - \xi^2 = \mu^2 + 2\eta^2 - 1 \quad (\text{B.4})$$

APÊNDICE C - Harmônicos Esféricos

Nesta seção são apresentados os termos Y_l^m presentes na expansão do polinômio de Legendre (equação A.9). A Figura C.1 ilustra a forma dos harmônicos esféricos para os graus $0 \leq l \leq 3$ (linhas) e ordens $-l \leq m \leq l$ (colunas). Pontos em azul representam valores positivos e em verde, valores negativos. No trabalho desenvolvido, adotou-se geometria bidimensional, correspondendo ao plano $X \times Z$ da figura. Pode-se observar que os harmônicos esféricos de ordem $M < 0$ possuem projeção nula no plano $X \times Z$. Nas Figuras C.2 e C.3 são apresentados os harmônicos esféricos no plano bidimensional, geradas pelos códigos `testPlotSH2D.m` (E.5) e `plotSH2D.m` (??).

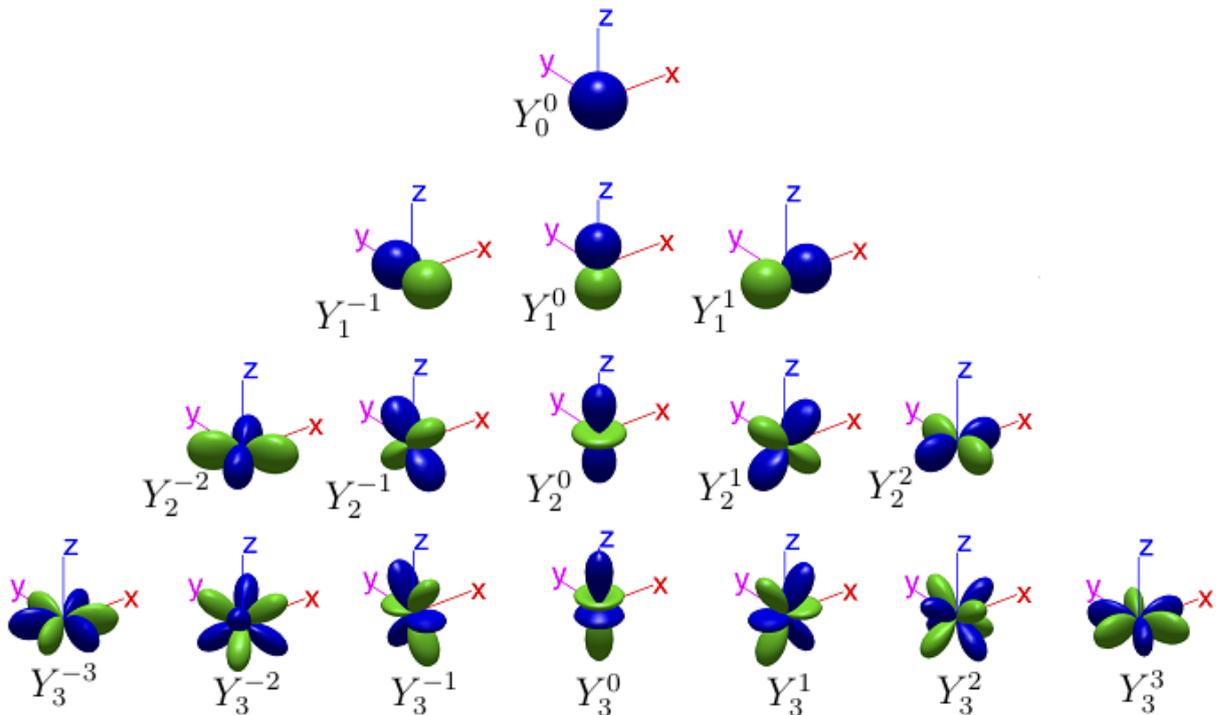


Figura C.1: Harmônicos esféricos reais para os graus $0 \leq l \leq 3$ e ordens $-l \leq m \leq l$ (adaptado de [33]).

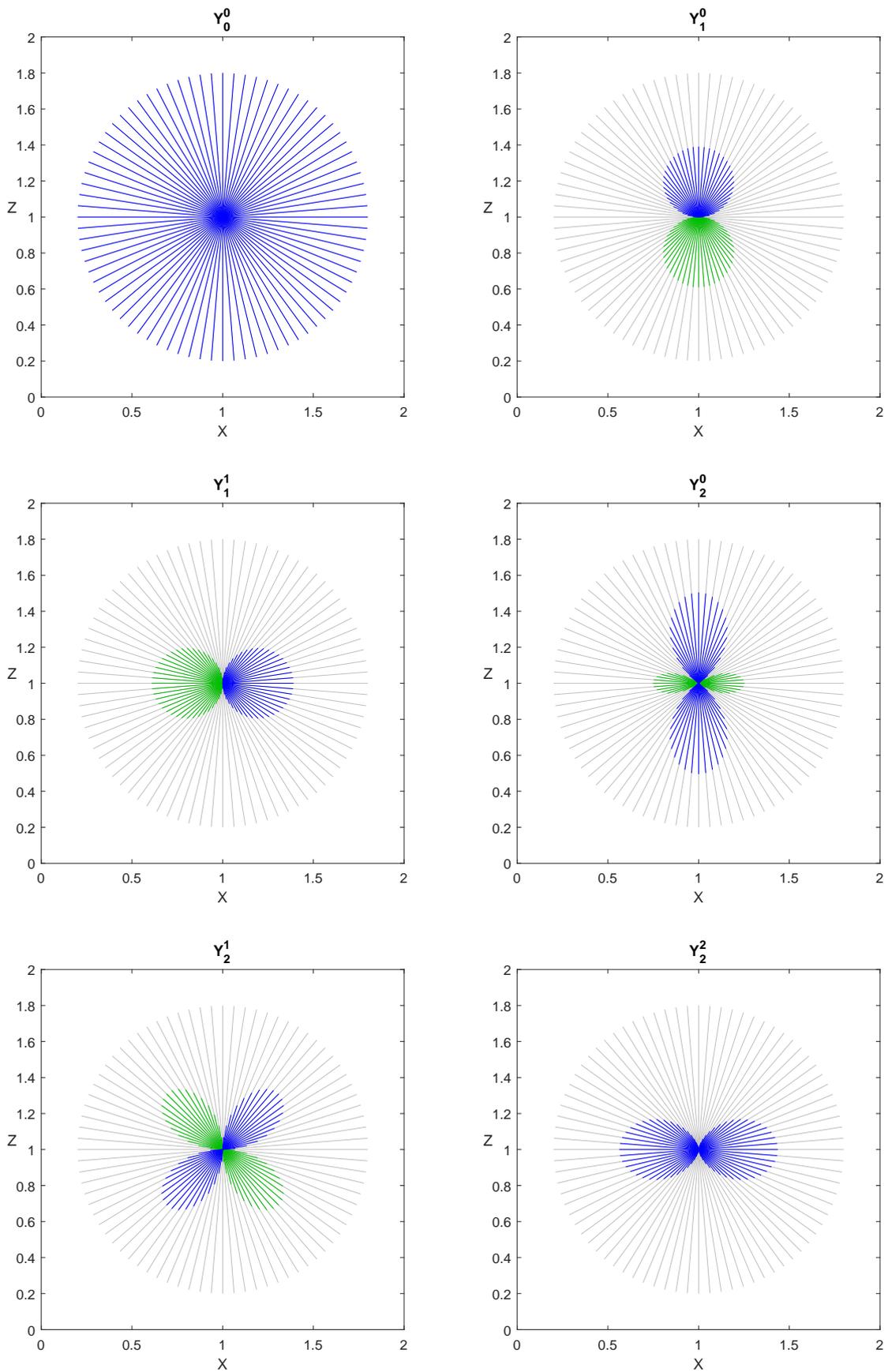


Figura C.2: Harmônicos esféricos reais: $Y_0^0, Y_1^0, Y_1^1, Y_2^0, Y_2^1, Y_2^2$ (??)

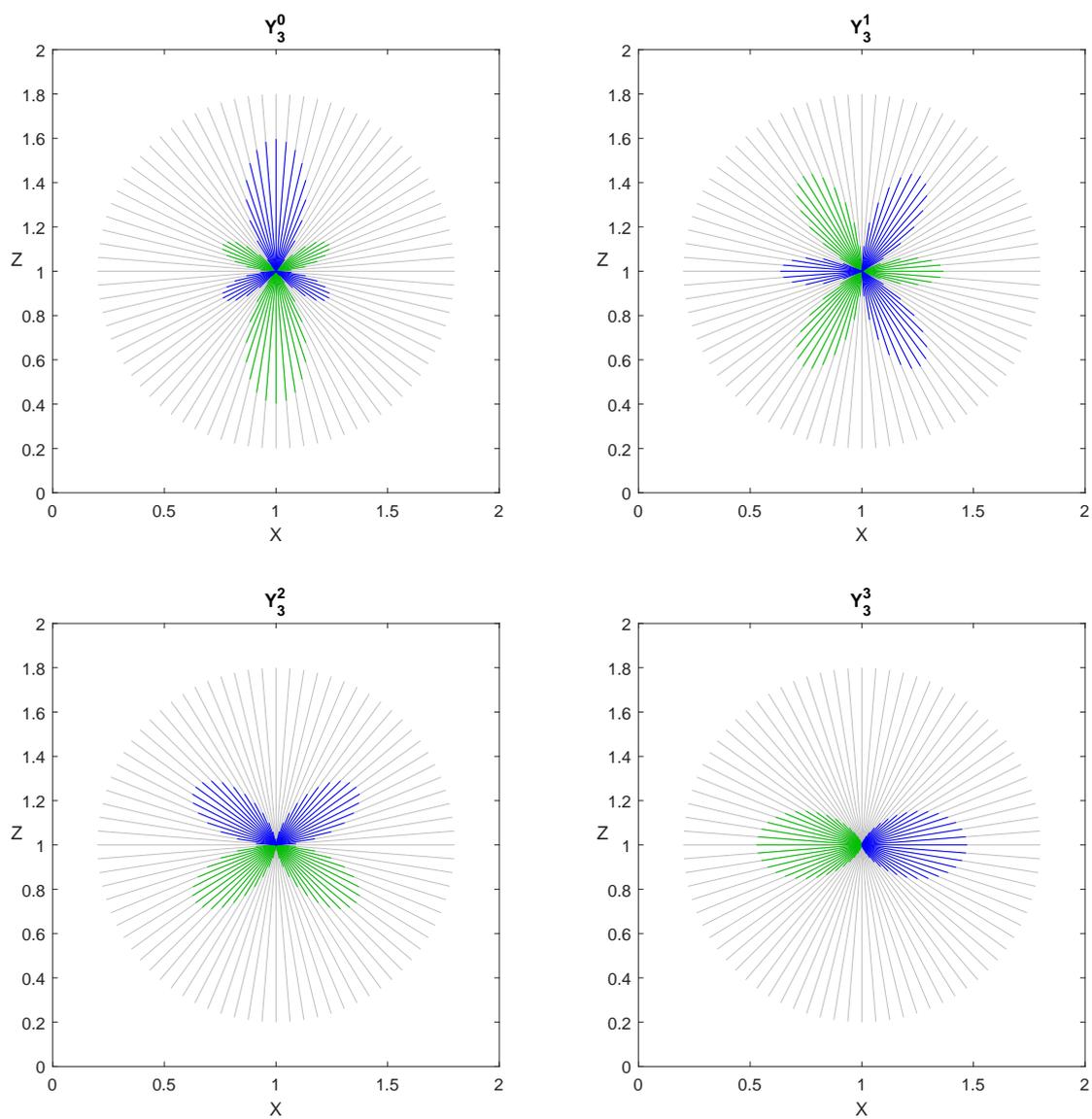


Figura C.3: Harmônicos esféricos reais: Y_3^0 , Y_3^1 , Y_3^2 , Y_3^3 (??)

APÊNDICE D - Algoritmo Sweep

A Figura D.1 ilustra o algoritmo *Sweep*(14) para uma malha hipotética de 3×3 calculando partículas transportadas nas direções do quadrante Q1 ($\mu > 0$ e $\eta > 0$). O transporte é calculado por frentes de varredura diagonais da célula inferior esquerda até a superior direita. As figuras ilustram os fluxos de saída em vermelho calculados em cada varredura a partir dos fluxos de entrada em azul.

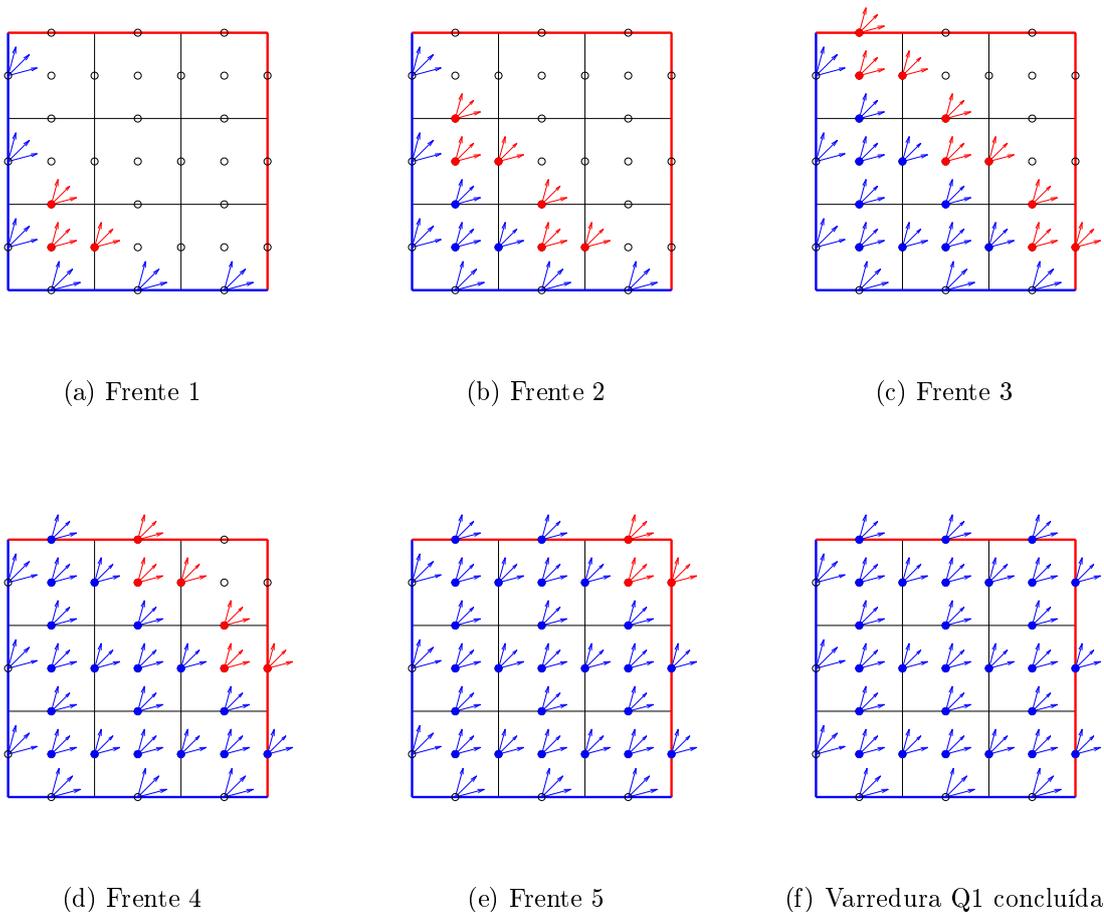


Figura D.1: Varredura do Quadrante 1

A Figura D.2 ilustra o algoritmo *Sweep*(14) para uma malha hipotética de 3×3 calculando partículas transportadas nas direções do quadrante Q2 ($\mu < 0$ e $\eta > 0$). O transporte é calculado por frentes de varredura diagonais da célula inferior direita até a superior esquerda. As figuras ilustram os fluxos de saída em vermelho calculados em cada varredura a partir dos fluxos de entrada em azul.

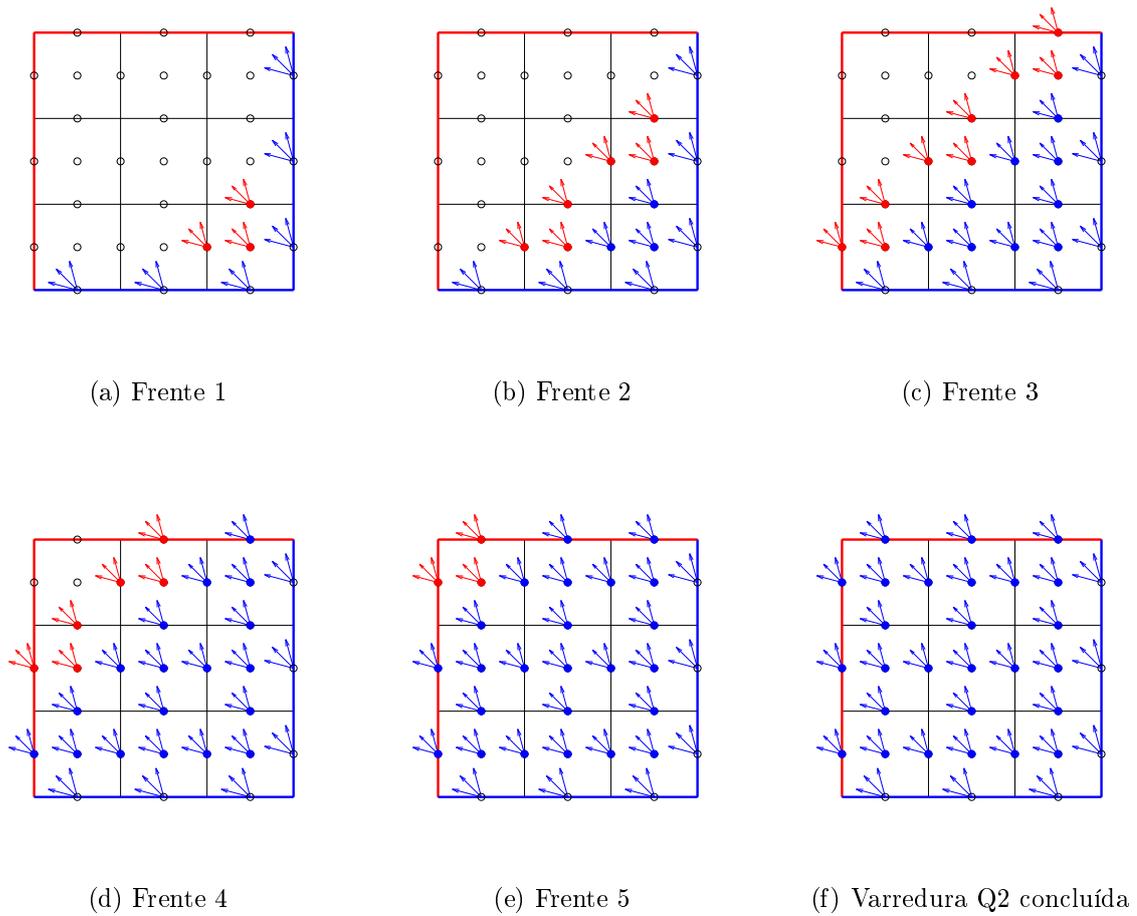
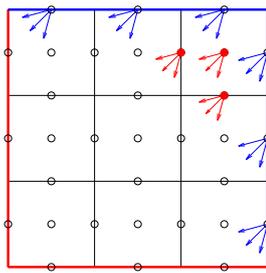
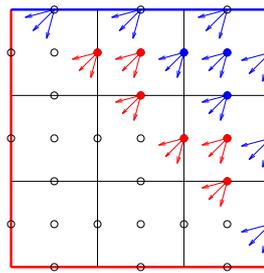


Figura D.2: Varredura do Quadrante 2

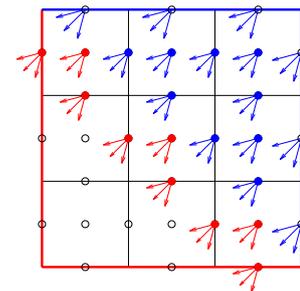
A Figura D.3 ilustra o algoritmo *Sweep*(14) para uma malha hipotética de 3×3 calculando partículas transportadas nas direções do quadrante Q3 ($\mu < 0$ e $\eta < 0$). O transporte é calculado por frentes de varredura diagonais da célula superior direita até a inferior esquerda. As figuras ilustram os fluxos de saída em vermelho calculados em cada varredura a partir dos fluxos de entrada em azul.



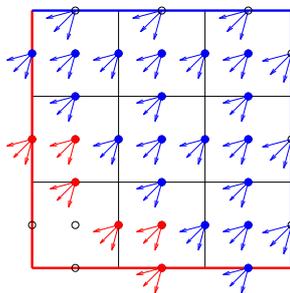
(a) Frente 1



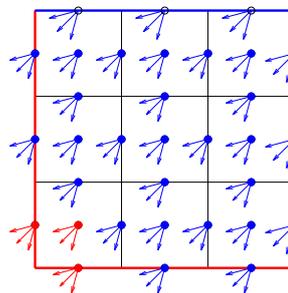
(b) Frente 2



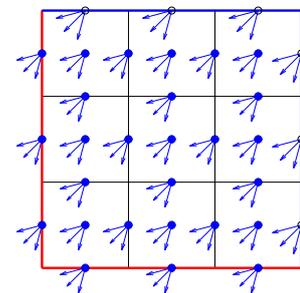
(c) Frente 3



(d) Frente 4



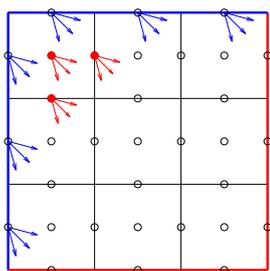
(e) Frente 5



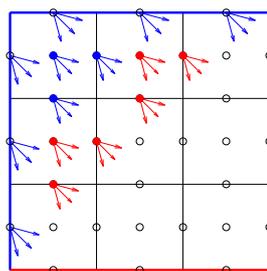
(f) Varredura Q3 concluída

Figura D.3: Varredura do Quadrante 3

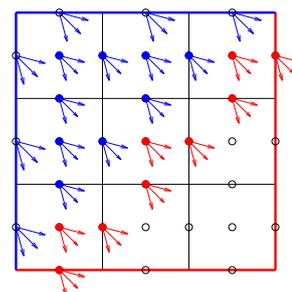
A Figura D.4 ilustra o algoritmo *Sweep*(14) para uma malha hipotética de 3×3 calculando partículas transportadas nas direções do quadrante Q4 ($\mu > 0$ e $\eta < 0$). O transporte é calculado por frentes de varredura diagonais da célula superior esquerda até a inferior direita. As figuras ilustram os fluxos de saída em vermelho calculados em cada varredura a partir dos fluxos de entrada em azul.



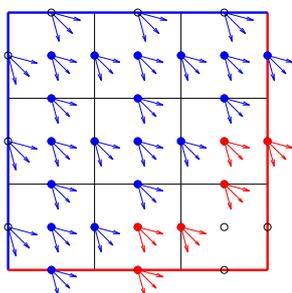
(a) Frente 1



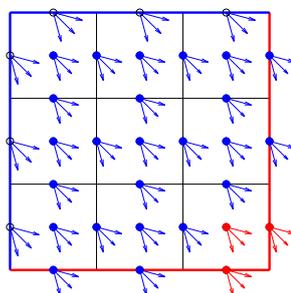
(b) Frente 2



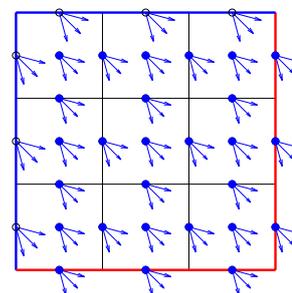
(c) Frente 3



(d) Frente 4



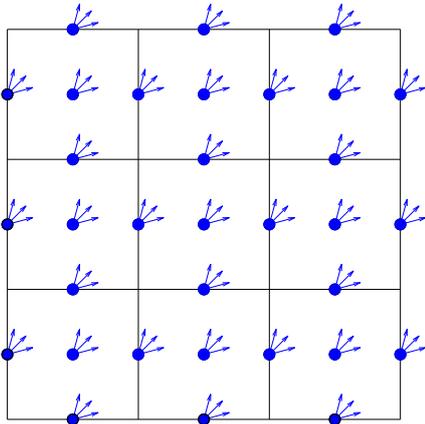
(e) Frente 5



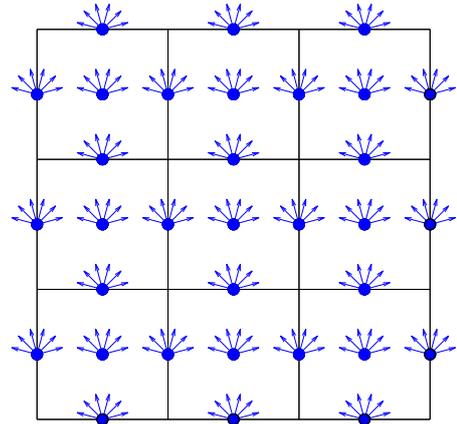
(f) Varredura Q4 concluída

Figura D.4: Varredura do Quadrante 4

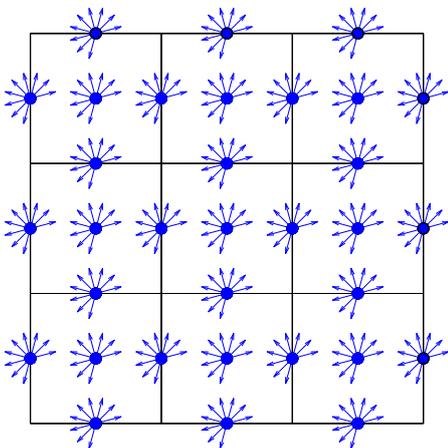
A Figura D.5 ilustra os fluxos resultantes do algoritmo *Sweep*(14) após a conclusão das varreduras Q1 (a), Q1 e Q2 (b), Q1, Q2 e Q3 (c) e Q1, Q2, Q3 e Q4 (d). Mais detalhes em E.3.



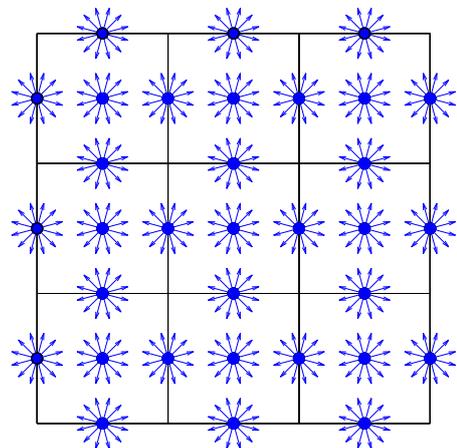
(a) Frente 1



(b) Frentes 1 e 2



(c) Frentes 1, 2 e 3



(d) Frentes 1, 2, 3 e 4

Figura D.5: Varreduras concluídas dos quatro quadrantes

APÊNDICE E – Listagem de Códigos

Nesta seção são apresentados alguns códigos em MATLAB usados na geração das ilustrações dos resultados obtidos nos experimentos.

E.1 saveas3.m

A função `saveas3` salva uma figura `fig` no arquivo `filename` segundo o formato `formattype` (pdf ou png), largura `width` (cm) e altura `height` (cm).

```

1 %Salva a figura no formato e dimensoes informados
2 function saveas3(fig, filename, formattype, width, height)
3     fig.Units = 'centimeters';
4     fig.PaperUnits = 'centimeters';
5     %define o tamanho da figura.
6     fig.PaperSize = [width height];
7     fig.PaperPosition = [0 0 width height];
8     %salva em arquivo
9     print(filename, sprintf('-d%s', formattype), '-r0');
10 end

```

Listing E.1: (saveas3.m) Salvamento de figura em arquivo

E.2 plotSweep.m

A função `plotSweep` exibe os fluxos angulares calculados pela frente de varredura q (0,1,2 ou 3) no diretório `dirPath` como ilustrado nas Figuras D.1, D.2, D.3 e D.4.

```

1 % Exibe a varredura no quadrante q=0/1/2/3. Salva a figura em arquivo
2 function plotSweep(q, dirPath)
3 L = 6;           %Largura total
4 w = L;          %Largura
5 h = L;          %Altura

```

```

6 N = 3;          %Resolucao NxN da grade
7 hx = w/N;      %Intervalo X
8 hy = h/N;      %Intervalo Y
9 NF = 2*N;      %Número de frentes de varredura
10 %define as frentes de varredura em X e Y -----
11 if (q == 0 || q == 3), fx=0; else; fx=1; end
12 if (q == 0 || q == 1), fy=0; else; fy=1; end
13 %define as posicoes iniciais e finais de varredura -----
14 sx = (-1)^fx; xbeg = fx*L + sx*hx/2;
15 sy = (-1)^fy; ybeg = fy*L + sy*hy/2;
16 %define as cores do contorno -----
17 if fx==0, cL='b'; cR='r'; else; cL='r'; cR='b'; end
18 if fy==0, cB='b'; cT='r'; else; cB='r'; cT='b'; end
19 % Para cada frente de verredura, uma figura diferente -----
20 for f = 0 : NF-1
21     % Figura -----
22     fig = figure(f+1); clf(f+1); set(fig,'WindowStyle','docked');
23     box off; view(2); axis equal;
24     set(gca,'XColor','none','YColor','none');
25     % define a area de visualizacao -----
26     dx = 0.3*hx; dy = 0.3*hy;
27     xmin=-dx; xmax=w+dx; ymin=-dy; ymax=h+dy;
28     axis([xmin, xmax, ymin, ymax]); hold on;
29     % Grade -----
30     [x,y] = meshgrid(0:hx:w, 0:hy:h);
31     plot(x,y,'k');
32     plot(y,x,'k');
33     % Contorno -----
34     plot([0,0],[0,h],'LineWidth',2,'Color',cL); %Left
35     plot([w,w],[0,h],'LineWidth',2,'Color',cR); %Right
36     plot([0,w],[0,0],'LineWidth',2,'Color',cB); %Bottom
37     plot([0,w],[h,h],'LineWidth',2,'Color',cT); %Top
38     % Fonte fixa -----
39     cxV = fx*w + zeros(1,N); cyV = hy/2 + hy*(0:N-1);
40     cyH = fy*h + zeros(1,N); cxH = hy/2 + hy*(0:N-1);
41     mu = cxV; eta = cyV;
42     for a = [15,45,75]
43         mu(:) = sx*cosd(a); eta(:) = sy*sind(a);
44         quiver(cxV,cyV,mu,eta, 0.3,'color','b','MaxHeadSize', 0.1);
45         quiver(cxH,cyH,mu,eta, 0.3,'color','b','MaxHeadSize', 0.1);
46     end
47     % Grade -----
48     cx = hx/2 + hx*(0:N-1); cy = hy/2 + hy*(0:N-1);

```

```

49 [x,y] = meshgrid(cx, cy); plot(x,y,'ko');
50 cx = 0 + hx*(0:N); cy = hy/2 + hy*(0:N-1);
51 [x,y] = meshgrid(cx, cy); plot(x,y,'ko');
52 cx = hx/2 + hx*(0:N-1); cy = 0 + hy*(0:N);
53 [x,y] = meshgrid(cx, cy); plot(x,y,'ko');
54 % Varredura da grade -----
55 for j = 0 : N-1 % para cada coluna
56     for i = 0 : N-1 % para cada linha
57         if (i+j > f); continue; end
58         if i+j==f, c='r'; else; c='b'; end
59         % plota o psiC (ponto) -----
60         cx = xbeg + sx*hx*j; cy = ybeg + sy*hy*i;
61         plot(cx, cy, 'o', 'color',c, 'MarkerFaceColor',c);
62         % plota o psiC (direcoes) -----
63         for a = [15,45,75]
64             mu = sx*cosd(a); eta = sy*sind(a);
65             quiver(cx,cy,mu,eta, 0.6,'color',c,'MaxHeadSize',0.8);
66         end
67         % plota o psiVout (ponto) -----
68         cx = xbeg + sx*hx/2 + sx*hx*j; cy = ybeg + sy*hy*i;
69         plot(cx, cy, 'o', 'color',c, 'MarkerFaceColor',c);
70         % plota o psiVout (direcoes) -----
71         for a = [15,45,75]
72             mu = sx*cosd(a); eta = sy*sind(a);
73             quiver(cx,cy,mu,eta, 0.6,'color',c,'MaxHeadSize',0.8);
74         end
75         % plota o psiHout (ponto) -----
76         cx = xbeg + sx*hx*j; cy = ybeg + sy*hy/2 + sy*hy*i;
77         plot(cx, cy, 'o', 'color',c, 'MarkerFaceColor',c);
78         % plota o psiHout (direcoes) -----
79         for a = [15,45,75]
80             mu = sx*cosd(a); eta = sy*sind(a);
81             quiver(cx,cy,mu,eta, 0.6,'color',c,'MaxHeadSize',0.8);
82         end
83     end
84 end
85 if exist('dirPath', 'var')
86     name = sprintf('%s/SweepQ%dF%d', dirPath, q+1, f+1);
87     saveas3(fig, name, 'pdf', 12, 12);
88 end
89 end

```

90 end

Listing E.2: (plotSweep.m) Exibição das frentes de varredura

E.3 plotSweep1234.m

A função plotSweep1234 exibe as varreduras concluídas para cada uma dos quatro quadrantes do plano (Figura D.5).

```

1 % Exibe a varredura no quadrante q e salva as figuras na pasta dirPath
2 function plotSweep1234(dirPath)
3 L = 6;          %Largura total
4 w = L;          %Largura
5 h = L;          %Altura
6 N = 3;          %Resolucao NxN da grade
7 hx = w/N;      %Intervalo X
8 hy = h/N;      %Intervalo Y
9 NF = 2*N;      %Número de frentes de varredura
10 for figNumber = 0:3
11 % cria a figura -----
12 fig = figure(10+figNumber); clf(10+figNumber);
13 set(fig,'WindowStyle','docked'); box off;
14 view(2); axis equal; set(gca,'XColor','none','YColor','none')
15 % define a area de visualizacao -----
16 dx = 0.3*hx; dy = 0.3*hy;
17 xmin=-dx; xmax=w+dx; ymin=-dy; ymax=h+dy;
18 axis([xmin, xmax, ymin, ymax]);
19 for q = 0 : figNumber
20 %define as frentes de varredura em X e Y -----
21 if (q == 0 || q == 3), fx=0; else; fx=1; end
22 if (q == 0 || q == 1), fy=0; else; fy=1; end
23 %define as posicoes iniciais e finais de varredura -----
24 sx = (-1)^fx; xbeg = fx*L + sx*hx/2;
25 sy = (-1)^fy; ybeg = fy*L + sy*hy/2;
26 a = [15,45,75]; mu = sx*cosd(a); eta = sy*sind(a);
27 % Para cada frente de verredura, uma figura diferente -----
28 f = NF-1;
29 % -----
30 hold on;
31 % Grade -----
32 [x,y] = meshgrid(0:hx:w, 0:hy:h); plot(x,y,'k'); plot(y,x,'k');
33 % Fonte fixa -----

```

```

34 x0 = xbeg*[1 1 1]; y0 = ybeg*[1 1 1];
35 for n = 0 : N-1
36     % fonte fixa face vertical
37     xv = x0 - sx*hx/2; yv = y0 + n*sy*hy;
38     plot(xv(1), yv(1), 'o', 'color','b', 'MarkerFaceColor','b');
39     quiver(xv, yv, mu, eta, 0.4, 'color','b', 'MaxHeadSize',0.8);
40     % fonte fixa face horizontal
41     xh = x0 + n*sx*hx; yh = y0 - sy*hy/2;
42     plot(xh(1), yh(1), 'o', 'color','b', 'MarkerFaceColor','b');
43     quiver(xh, yh, mu, eta, 0.4, 'color','b', 'MaxHeadSize',0.8);
44 end
45 % Grade -----
46 cx = hx/2 + hx*(0:N-1); cy = hy/2 + hy*(0:N-1);
47 [x,y] = meshgrid(cx, cy); plot(x,y,'ko');
48 cx = 0 + hx*(0:N); cy = hy/2 + hy*(0:N-1);
49 [x,y] = meshgrid(cx, cy); plot(x,y,'ko');
50 cx = hx/2 + hx*(0:N-1); cy = 0 + hy*(0:N);
51 [x,y] = meshgrid(cx, cy); plot(x,y,'ko');
52 % Varredura da grade -----
53 for j = 0 : N-1 % para cada coluna
54     for i = 0 : N-1 % para cada linha
55         if (i+j > f); continue; end
56         if i+j==f, c='r'; else; c='b'; end
57         cx = (xbeg + sx*hx*j)*[1 1 1];
58         cy = (ybeg + sy*hy*i)*[1 1 1];
59         % plota a origem e direcoes de psiC -----
60         plot(cx(1), cy(1), 'o','color',c,'MarkerFaceColor',c);
61         quiver(cx, cy, mu, eta, 0.4, 'color',c, 'MaxHeadSize',0.8);
62         % plota a origem e direcoes de psiVout -----
63         plot(cx(1)+sx*hx/2, cy(1), 'o','color',c,'MarkerFaceColor',c);
64         quiver(cx+sx*hx/2,cy,mu,eta, 0.4,'color',c,'MaxHeadSize',0.8);
65         % plota a origem e direcoes de psiHout -----
66         plot(cx(1),cy(1)+sy*hy/2, 'o','color',c,'MarkerFaceColor',c);
67         quiver(cx,cy+sy*hy/2,mu,eta, 0.4,'color',c,'MaxHeadSize',0.8);
68     end
69 end
70 end
71 if exist('dirPath', 'var')
72     name = sprintf('%s/SweepFinal%d', dirPath, q+1);
73     saveas3(fig, name, 'pdf', 12, 12);
74 end
75 end

```

76 `end`

Listing E.3: (plotSweep1234.m) Exibição das varredura concluídas

E.4 getOrdinates.m

Constrói a lista de ordenadas discretas (μ, η) da quadratura S_N de acordo com a tabela 4.1 da referência [25]. Gera os arquivos ordenadas4.csv, ordenadas6.csv, ordenadas8.csv e ordenadas16.csv. Os valores gerados são usados pelos algoritmos 2, 3, 4 e 5.

```

1 % Constroi as ordenadas discretas de S4, S6, S8 e S16
2 function controioOrdenadas()
3     %Quadratura S4
4     getOrdinates(0.3500212, 4);
5     %Quadratura S6
6     getOrdinates(0.2666355, 6);
7     %Quadratura S8
8     getOrdinates(0.2182179, 8);
9     %Quadratura S16
10    getOrdinates(0.1389568, 16);
11 end
12 % Retorna a componente de ordenada a partir da semente mu1
13 function muI = getOrdinate(i, mu1, N)
14     C = 2*(1 - 3*mu1^2)/(N - 2);
15     muI = sqrt(mu1^2 + (i - 1)*C);
16 end
17 % Constroi as ordenadas discretas de SN usando a semente mu1
18 function r = getOrdinates(mu1, N)
19     fprintf('\nQuadratura S%d. mu1(%.7f)\n', N, mu1);
20     M = N*(N+2)/2;
21     tabela = zeros(M, 3);
22     n = N/2;
23     %calcula as coordenadas simetricas -----
24     coords = zeros(1, n);
25     for i = 1:n
26         coords(i) = getOrdinate(i, mu1, N);
27     end
28     %-----
29     index = 1;
30     nivel = 1;
31     %exibe as componentes mu e eta dos vetores omega da quadratura
32     while nivel <= n

```

```

33     for i1 = 1:n
34         for i2 = 1:n
35             if (i1 + i2 == nivel+1)
36                 mu = coords(i1);
37                 eta = coords(i2);
38                 xi = sqrt(1 - mu^2 - eta^2);
39                 tabela(index,:) = [mu, eta, xi];
40                 index = index + 1;
41             end
42         end
43     end
44     nivel = nivel + 1;
45 end
46 %preenche os quadrantes Q2, Q3 e Q4
47 m = M/4;
48 tabela(1+1*m : 2*m, 1) = -tabela(1 : m, 1); %Q2 mu
49 tabela(1+1*m : 2*m, 2) =  tabela(1 : m, 2); %Q2 eta
50 tabela(1+1*m : 2*m, 3) =  tabela(1 : m, 3); %Q2 xi
51 tabela(1+2*m : 3*m, 1) = -tabela(1 : m, 1); %Q3 mu
52 tabela(1+2*m : 3*m, 2) = -tabela(1 : m, 2); %Q3 eta
53 tabela(1+2*m : 3*m, 3) =  tabela(1 : m, 3); %Q3 xi
54 tabela(1+3*m : 4*m, 1) =  tabela(1 : m, 1); %Q4 mu
55 tabela(1+3*m : 4*m, 2) = -tabela(1 : m, 2); %Q4 eta
56 tabela(1+3*m : 4*m, 3) =  tabela(1 : m, 3); %Q4 xi
57
58 csvwrite(sprintf('ordenadas_%2d.csv', N), tabela);
59 end

```

Listing E.4: (getOrdinates.m) Construção das ordenadas discretas

E.5 testPlotSH2D.m

Geração dos gráficos dos harmônicos esféricos de ordens: Y_{00} , Y_{10} , Y_{11} , Y_{20} , Y_{21} , Y_{22} , Y_{30} , Y_{31} , Y_{32} , Y_{33} . Chama a função plotSH2D com os parâmetros l (grau), m (ordem) e N (número de direções = 80).

```

1 % Geracao dos graficos dos harmonicos esfericos de ordens:
2 %   Y00, Y10, Y11, Y20, Y21, Y22, Y30, Y31, Y32, Y33
3 clear; clc;
4 % -----
5 h = plotSH2D( 0, 0, 80);
6 saveas3(h, 'Y00', 'pdf', 12, 12);

```

```

7 % -----
8 h = plotSH2D( 1, 0, 80);
9 saveas3(h, 'Y10', 'pdf', 12, 12);
10 h = plotSH2D( 1, 1, 80);
11 saveas3(h, 'Y11', 'pdf', 12, 12);
12 % -----
13 h = plotSH2D( 2, 0, 80);
14 saveas3(h, 'Y20', 'pdf', 12, 12);
15 h = plotSH2D( 2, 1, 80);
16 saveas3(h, 'Y21', 'pdf', 12, 12);
17 h = plotSH2D( 2, 2, 80);
18 saveas3(h, 'Y22', 'pdf', 12, 12);
19 % -----
20 h = plotSH2D( 3, 0, 80);
21 saveas3(h, 'Y30', 'pdf', 12, 12);
22 h = plotSH2D( 3, 1, 80);
23 saveas3(h, 'Y31', 'pdf', 12, 12);
24 h = plotSH2D( 3, 2, 80);
25 saveas3(h, 'Y32', 'pdf', 12, 12);
26 h = plotSH2D( 3, 3, 80);
27 saveas3(h, 'Y33', 'pdf', 12, 12);
28 % -----

```

Listing E.5: (testPlotSH2D.m) Geração dos harmônicos esféricos

```

1 % Plota o esférico harmônico 2D Ylm em N direções
2 % Gera o gráfico do harmônico esférico de grau l, ordem m, N direções
3 function [h] = plotSH2D(l, m, N)
4     Ylm = @(t) ones(1, length(t)); % Esférico Harmônico 2D
5     if l == 1 && m == 0
6         Ylm = @(t) sqrt(3/(4*pi)) .* cos(t);
7     elseif l == 1 && m == 1
8         Ylm = @(t) -sqrt(3/(4*pi)) .* sin(t);
9     elseif l == 2 && m == 0
10        Ylm = @(t) sqrt(5/(16*pi)) .* (3*cos(t).^2-1);
11    elseif l == 2 && m == 1
12        Ylm = @(t) -sqrt(15/(4*pi)) .* (cos(t).*sin(t));
13    elseif l == 2 && m == 2
14        Ylm = @(t) sqrt(15/(16*pi)) .* (2*sin(t).^2+cos(t).^2-1);
15    elseif l == 3 && m == 0
16        Ylm = @(t) sqrt(7/(16*pi)) .* (5*cos(t).^3-3*cos(t));
17    elseif l == 3 && m == 1
18        Ylm = @(t) sqrt(21/(32*pi)) .*
19            (-5*cos(t).^2.*sin(t)+sin(t));

```

```

20     elseif l == 3 && m == 2
21         Ylm = @(t) sqrt(105/(16*pi)) .*
22             cos(t).*(2*sin(t).^2+cos(t).^2-1);
23     elseif l == 3 && m == 3
24         Ylm = @(t) sqrt(35/(32*pi)) .*
25             (3*sin(t).*(1-cos(t).^2-sin(t).^2)-sin(t).^3);
26     end
27     if nargin < 3; N = 80; end
28     %define o numero da figura
29     if l > 0 && m >= 0, F = l*10+m; else, F = 1; end
30     t = (0:N-1) * (2*pi/N);      % angulos entre 0 e 2pi dos M vetores
31     ox(1:N) = 1; oy(1:N) = 1;    % origem dos M vetores
32     c = cos(t); s = sin(t);     % componentes x,y dos vetores
33     u = t-pi/2;                 % posicionando angulo 0 no norte
34     v = Ylm(u);                 % valores de Ylm (positivos e neg.)
35     V = abs(v);                 % valores de Ylm sempre positivos
36     %cria a figura, limpa a figura e delimita a area de visao
37     h = figure(F); clf(F); view(2); axis equal; axis([0, 2, 0, 2]);
38     set(h,'WindowStyle','docked');
39     box on;
40     xlabel('X');
41     ylabel('Z');
42     set(get(gca,'ylabel'),'rotation',0);
43     title(sprintf('Y_%d^%d', l, m));
44     %plota as direcoes em cinza
45     hold on;
46     quiver(ox, oy, c, s, 0.8, 'color', [0.75, 0.75, 0.75],
47         'MaxHeadSize', 0.0);
48     oxn = ox(v<0); oyn = oy(v<0); vn = V(v<0);
49     vxn = c(v<0); vyn = s(v<0);
50     %plota as intensidades negativas em verde
51     hold on;
52     quiver(oxn, oyn, vxn.*vn, vyn.*vn, 0.8, 'color', [0, 0.75, 0],
53         'MaxHeadSize', 0.0);
54     vp = V(v>=0); cp = c(v>=0); sp = s(v>=0); oxp = ox(v>=0); oyp = oy(v
55     %plota as intensidades positivas em azul
56     hold on;
57     quiver(oxp, oyp, cp.*vp, sp.*vp, 0.8, 'color', [0, 0, 1],
58         'MaxHeadSize', 0.0);
59 end

```

Listing E.6: (plotSH2D.m) Geração de harmônico esférico

Referências

- [1] ABREU, M. P. D. *Solução Numérica do Problema Linear do Transporte de Nêutrons Independente do Tempo na Formulação de Ordenadas Discretas*. PhD thesis, Universidade Federal do Rio de Janeiro, 1994.
- [2] ALLISON, J., ET AL. Geant4 developments and applications. *IEEE Transactions on Nuclear Science* 53 (2006), 270–278.
- [3] AZMY, Y., SARTORI, E. *Nuclear Computational Science*. Springer, 2009.
- [4] BARBARINO, A. *Numerical Methods for Neutron Transport Calculations of Nuclear Reactors*. PhD thesis, Politecnico di Torino, 2014.
- [5] BELLOMO, N., DE ANGELIS, E. *Selected Topics in Cancer Modeling: Genesis, Evolution, Immune Competition, and Therapy*. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser Boston, 2008.
- [6] CAMMINADY, T., FRANK, M., KÜPPER, K., KUSCH, J. Ray effect mitigation for the discrete ordinates method through quadrature rotation. *Journal of Computational Physics* 382 (Apr 2019), 105–123.
- [7] CAMPOS, T. Considerações sobre a terapia de captura de nêutrons pelo boro. *Revista Brasileira de Cancerologia* 46 (2000), 283–292.
- [8] CRUYWAGEN, G., E. WOODWARD, D., TRACQUI, P., T. BARTOO, G., D. MURRAY, J., C. ALVORD, E. The modelling of difusive tumours. *Journal of Biological Systems* 03 (11 2011).
- [9] DEISBOECK, T., STAMATAKOS, G. *Multiscale Cancer Modeling*. Chapman & Hall/-CRC Mathematical and Computational Biology. CRC Press, 2010.
- [10] DOE. The Integrated TIGER Series (ITS). World Wide Web, <https://www.osti.gov/servlets/purl/1315240>.
- [11] DUDERSTADT, J. J., MARTIN, W. R. *Transport Theory*. New York : Wiley, 1979.
- [12] GARCIA, V. D. S. *Estudo de Sensibilidade em BNCT usando Teoria da Perturbação Generalizada*. PhD thesis, Universidade Federal do Rio de Janeiro, 2009.
- [13] GEANT4. Geant4 A Simulation Toolkit. World Wide Web, <https://geant4.web.cern.ch/>.
- [14] GEANT4APP. Geant4 Applications. World Wide Web, <https://geant4.web.cern.ch/applications/>.

- [15] GREEN, R. Spherical harmonic lighting the gritty details. Em *GDC* (2003).
- [16] JAIN, L., KARTHIKEYAN, R., KANNAN, U. Diamond – a method of characteristics neutron transport code using unstructured meshing. *Annals of Nuclear Energy* 137 (2020), 107086.
- [17] JUNIOR, S. D. C. F. *Modelos de Reação Difusão para o Crescimento de Tumores*. PhD thesis, Universidade Federal de Minas Gerais, 2003.
- [18] KAWABATA, S., MIYATAKE, S., KAJIMOTO, Y., ET AL. The early successful treatment of glioblastoma patients with modified boron neutron capture therapy. *Journal of Neuro-Oncology* 65 (2003), 159–165.
- [19] KONZEN, P. H. D. A., GUIDI, L. F., RICHTER, T. Quasi-random discrete ordinates method for neutron transport problems. *Annals of Nuclear Energy* 133 (2019), 275–282.
- [20] LAMARSH, J. *Introduction to Nuclear Reactor Theory*. Addison-Wesley, 1978.
- [21] LANL. A General Monte Carlo N-Particle (MCNP) Transport Code. World Wide Web, <https://mcnp.lanl.gov/>.
- [22] LATHROP, K. D. Ray effects in discrete ordinates equations. *Nuclear Science and Engineering* 32, 3 (1968), 357–369.
- [23] LATHROP, K. D. Remedies for ray effects. *Nuclear Science and Engineering* 45, 3 (1971), 255–268.
- [24] LEOBACHER, G., PILLICHSHAMMER, F. *Introduction to Quasi-Monte Carlo Integration and Applications*. Springer, Springer Cham Heidelberg New York Dordrecht London, 2014.
- [25] LEWIS, E., MILLER, W. *Computational Methods of Neutron Transport*. American Nuclear Societ, Inc, La Grange Park, Illinois, USA, 1993.
- [26] MARASHI, M. K. Analysis of absorbed dose distribution in head phantom in boron neutron capture therapy. *Nuclear Instruments and Methods in Physics Research* 440 (2000), 446–452.
- [27] MARTINS, M. L., FERREIRA, S. C., VILELA, M. J. Multiscale models for the growth of avascular tumors. *Phys. Life* 4 (2007), 128–156.
- [28] MIYATAKE, S.-I., KAWABATA, S., HIRAMATSU, R., KUROIWA, T., SUZUKI, M., KONDO, N., ONO, K. Boron neutron capture therapy for malignant brain tumors. *Neurologia medico-chirurgica* 56, 7 (2016), 361–371.
- [29] MOGHADDASI, L., BEZAK, E. Development of an integrated monte carlo model for glioblastoma multiforme treated with boron neutron capture therapy. Em *Scientific Reports* (2017).
- [30] MOUSTAFA, S. *Massively Parallel Cartesian Discrete Ordinates Method for Neutron Transport Simulation*. PhD thesis, L’Université de Bordeaux, 2015.

-
- [31] MURRAY, J. D. *Mathematical Biology II: Spatial Models and Biomedical Applications*, vol. 18 em *Interdisciplinary Applied Mathematics*. Springer New York, 2003.
- [32] PEREIRA, L. D. O. Estudo da influência do tamanho de campo de um feixe de nêutrons epitérmicos na distribuição de dose em bnct. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, 2009.
- [33] POLITIS, A. Microphone array processing for parametric spatial audio techniques. World Wide Web, <https://www.mathworks.com/matlabcentral/fileexchange/43856-real-complex-spherical-harmonic-transform-gaunt-coefficients-and-rotations>. (2016).
- [34] SALVAT, F., FERNÁNDEZ-VERA, J., SEMPÁU, J. PENELOPE (Penetration and Energy Loss of Positrons and Electrons). World Wide Web, <http://pypenelope.sourceforge.net/>.
- [35] SHIBA, H., , ET AL. Boron neutron capture therapy combined with early successive bevacizumab treatments for recurrent malignant gliomas – a pilot study. *Neurologia medico-chirurgica* 58, 12 (2018), 487–494.
- [36] SILVA, J. J. D. Modelagem computacional aplicada ao tratamento de câncer via medicina nuclear. Dissertação de Mestrado, Universidade Federal Fluminense, 2014.
- [37] TANAKA, M. L., DEBINSKI, W., PURI, I. K. Hybrid mathematical model of glioma progression. *Cell Proliferation* 42, 5 (2009), 637–646.