Bruno Venâncio Barbosa

Desenvolvimento de uma interface gráfica para o tratamento de problemas de transporte de nêutrons

Rio Grande, Rio Grande do Sul, Brasil

Fevereiro, 2018

Bruno Venâncio Barbosa

Desenvolvimento de uma interface gráfica para o tratamento de problemas de transporte de nêutrons

Trabalho de Conclusão de Curso, Matemática Aplicada Bacharelado, submetido por Bruno Venâncio Barbosa junto ao Instituto de Matemática, Estatística e Física da Universidade Federal do Rio Grande.

Universidade Federal do Rio Grande - FURG

Instituto de Matemática, Estatística e Física - IMEF

Curso de Matemática Aplicada Bacharelado

Orientador: Dr. João Francisco Prolo Filho

Rio Grande, Rio Grande do Sul, Brasil

Fevereiro, 2018

Universidade Federal do Rio Grande – FURG
Instituto de Matemática, Estatística e Física
Curso de Bacharelado em Matemática Aplicada
Av. Itália km 8 Bairro Carreiros
Rio Grande-RS CEP: 96.201-900 Fone (53)3233.5411
e-mail: imef@furg.br Sítio: www.imef.furg.br

Ata de Defesa de Monografia

Aos vinte dias do mês de fevereiro de 2018, às 14h, no mini auditório da Escola de Engenharia, no Campus Carreiros, foi realizada a apresentação pública da defesa do Trabalho de Conclusão de Curso pelo acadêmico **Bruno Venâncio Barbosa**, sob a orientação do Prof. Dr. João Francisco Prolo Filho, deste instituto, e intitulada **Desenvolvimento de uma interface gráfica para o tratamento de problemas de transporte de nêutrons**. Para participar da banca avaliadora, junto ao orientador, foram convidados o Prof. Dr. Matheus Jatkoske Lazo e a Prof.ª Dr.ª Bárbara Denicol do Amaral Rodriguez, docentes do IMEF/FURG. Concluídos os trabalhos de apresentação e arguição, a candidata foi: (X) aprovado por unanimidade; ( ) aprovado somente após satisfazer as exigências que constam na folha de modificações, no prazo fixado pela banca; ( ) reprovado. Na forma regulamentar, foi lavrada a presente ata, que é abaixo assinada pelos membros da banca, na ordem acima relacionada.

João Francisco Prolo Filho

Prof. Dr. João Francisco Prolo Filho
Orientador

Matheus Jatkoske Lazo

Prof. Dr. Matheus Jatkoske Lazo
Membro

B Denicol Rodriguez

Prof.ª Dr.ª Bárbara Denicol do Amaral Rodriguez
Membro

Este trabalho é dedicado a todas as pessoas que contibuiram em minha caminhada.

Agradecimentos

Aos meus Pais, João e Glauce Barbosa, não só pelo apoio, mas pelo que sou.

Ao meu Avô Candido que sempre me ajudou e incentivou a nunca desistir.

A todo o pessoal do laboratório de Estatística Ambiental, aos professores Juliano, Kinas e Raquel, amigos Hans, Carlos, Carol, Liana, Laura e Marcus.

Ao João Francisco Prolo Filho que me orientou neste trabalho e pela paciência disposta, também por ter se tornado um grande amigo.

Aos colegas do meio acadêmico e amigos.

À FURG(PDE/EPEC) e FAPERGS pelo apoio financeiro.

Resumo

Neste trabalho será apresentada uma abordagem para a solução da equação de transporte de nêutrons baseada no Método de Ordenadas Discretas Analítico (método ADO), bem como o desenvolvimento de um software para auxiliar no estudo das classes de problemas que envolvem meios homogêneos e heterogêneos em geometria cartesiana unidimensional, com regime estacionário e monoenergético, considerando efeitos de isotropia e anisotropia linear no espalhamento, tendo fontes de nêutrons nos extremos do domínio. No processo, uma discretização da variável angular é realizada, de forma que a equação integro-diferencial de transporte é transformada em um sistemas de EDO's. Propondo que as soluções sejam escritas em termos de exponenciais decrescentes, obtêm-se um sistemas algébrico no qual, após algumas manipulações matemáticas, um problema de autovalores de ordem reduzida é obtido. Para a determinação de alguns coeficientes que surgem nos cálculos, condições de contorno são utilizadas. Após a completa determinação das soluções, certas quantidades de interesse podem ser analisadas através de expressões analíticas na variável espacial. Para isto, um software em linguagem *Python* foi implementado, cuja interface gráfica possibilita a imediata visualização de gráficos e obtenção de perfis numéricos, o que auxilia na análise dos resultados. Para validação do código e do método, no qual testes foram simulados e os resultados foram comparados com a literatura. A demais, disponibiliza-se ao longo do texto o código em *Python* referente a a geração da interface gráfica.

Palavras-chaves: transporte unidimensional de nêutrons; Método de Ordenadas Discretas Analítico (ADO); Python.

Lista de ilustrações

Lista de tabelas

Lista de símbolos

a	dimensão do domínio [0,a] (cm)
A_α, D_α	matrizes do problema de autovalores na camada α
$A_{j,\alpha}$	coeficiente da solução homogênea na camada α
E	energia (J)
N	ordem da quadratura
$\vec{r}$	vetor posição (m)
t	tempo (s)
$\vec{v}$	vetor velocidade (m/s)
$U_\alpha, V_\alpha,$	funções auxiliares na camada α
x	variável espacial a direção i
y	variável espacial a direção j
z	variável espacial a direção k
w_k	pesos da quadratura
M	número de camadas homogêneas

Lista de símbolos

α	índice referente a camada homogênea
$\nu_{\alpha,j}$	constante de separação na camada α
μ	direção de propagação das partículas
λ_α	autovalores do problema homogêneo na camada α
Φ_α	autofunção do problema homogêneo na camada α
ϕ_α	fluxo escalar na variável x na camada α (N/$cm^2.s$)
Ψ_α	fluxo angular na camada α ($W/cm^2.Sr$)
Ψ_α^h	solução homogênea do problema na camada α
$\sigma_{t,\alpha}$	seção de choque macroscópica total na camada α (cm^{-1})
$\sigma_{a,\alpha}$	seção de choque macroscópica de absorção na camada α (cm^{-1})
$\sigma_{s0,\alpha}$	seção de choque macroscópica de espalhamento isotrópica na camada α (cm^{-1})
$\sigma_{s1,\alpha}$	seção de choque macroscópica de espalhamento anisotrópico linear na camada α (cm^{-1})
$\vec{\Omega}$	vetor unitário que define a direção do nêutron

Sumário

Introdução

Idealizada por Ludwig Boltzmann no século XIX para descrever a dinâmica de gases rarefeitos, e logo após utilizada para transferência radioativa [9, 15], a Equação de Boltzmann (EB) é uma equação integro-diferencial não-linear que expressa o comportamento estatístico das partículas. Devido a sua não linearidade, há grandes dificuldades na sua resolução, encorajando o uso de versões mais simplificadas (modelos) dela como, por exemplo, a Equação Linearizada de Boltzmann (ELB) e a Equação de Transporte (ET). A Equação de Transporte tem sido muito utilizada na modelagem matemática de problemas em inúmeras áreas de pesquisa, tais como sondas nucleares para prospecção de petróleo [1, 2, 18], em aerodinâmica [19], nas aplicações envolvendo vegetação para estudos da taxa de fotossíntese [17], na segurança e detecção de transporte de materiais nucleares [14, 13], entre outros.

Tendo em mente sua versatilidade na aplicação em problemas envolvendo partículas, a Equação de Transporte obteve importante papel na área de energia nuclear, contribuindo para o avanço no estudo de reatores, através da determinação de grandezas relacionadas ao transporte de nêutrons no núcleo [9, 15] e as características de blindagens necessárias [12].

No contexto de geração de energia por fontes nucleares, a Equação de Transporte de nêutrons é um modelo construído a partir da ideia de que há um balanço de partículas que entram e saem do meio hospedeiro, onde são assumidas algumas hipóteses com intuito de simplificar o problema matemático e resolver determinada classe de problema sem perder a influência física desejada. Obtido o modelo desejado, a resolução pode ser feita por métodos numéricos e/ou analíticos.

O Método das Ordenadas Discretas tradicional, apresentado por Wick [20] e complementado por Chandrasekhar [9], destaca-se entre os métodos determinísticos, devido a sua aplicabilidade em problemas unidimensionais [3], tratamento de problemas em diferentes geometrias [6, 7] e em meios heterogêneos [16], estando sempre associado a um menor custo computacional. Sua utilização pressupõe a discretização do termo integral por uma quadratura numérica, o que converte a Equação de Transporte integro-diferencial em um sistema de equações diferenciais cuja solução pode ser obtida tanto por formulações numéricas quanto analíticas, na qual será dado destaque aqui ao Método de Ordenadas Discretas Analítico (método ADO) [5].

O Método estudado aqui se baseia no Método ADO que se caracteriza não só por propôr uma solução homogênea em termos de exponenciais decrescentes, resultando em um tratamento analítico da variável espacial, mas também por determinar as constantes

de separação a partir de um problema de autovalores de tamanho reduzido. O fato de não necessitar de métodos iterativos ou de interpolação faz com que esta formulação tenha um menor custo do ponto de vista computacional. Além disso, a analiticidade das soluções em termos da variável espacial, aliado a capacidade de lidar com esquemas de quadratura arbitrários de diversas ordens, garantem uma maior confiança dos resultados quanto a precisão.

Além do avanço das técnicas de otimização dos métodos matemáticos, teve-se também um grande crescimento na capacidade de processamento computacional. Como consequência, tem-se obtido resultados com maior rapidez e precisão, sendo possível solucionar problemas cada vez mais complexos e encorajando o surgimento de novas formulações para sua resolução. A modelagem computacional de problemas realísticos têm permitido uma melhor análise nos processos, o que tem auxiliado na tomada de decisões.

No período da segunda guerra mundial, os computadores eram utilizados por equipes com conhecimento técnico para cálculos científicos. Entretanto, atualmente servem para inúmeros propósitos, do lazer ao trabalho, e estão sendo amplamente manuseados por pessoas comuns sem conhecimento de programação. Por causa dessa expansão, houve a necessidade do desenvolvimento de softwares com interfaces gráficas amigáveis para facilitar e/ou otimizar a utilização pelo usuário.

Softwares podem ser descritos como uma sequência de instruções a serem seguidas e/ou executadas, desenvolvidos em uma ou até mesmo múltiplas linguagens de programação. Para o qual será dada ênfase aqui a linguagem *Python* [11], por ser de alto nível orientada a objeto e tipagem dinâmica, que vem se sobressaindo em diferentes áreas, graças a facilidade de aprendizado e por sua poderosa biblioteca que auxilia na criação de websites, desenvolvimento de jogos, analise estatística, entre outros. A grande quantidade de material existente na literatura que pode auxiliar os programadores que estão iniciando e/ou que desejam aprender sobre uma determinada subrotina também encorajou e possibilitou a criação do aplicativo.

Com o uso dos pacotes *Numpy* e *Tkinter* [10, 11] que compõe a biblioteca básica do *Python*, conseguiu-se neste trabalho o desenvolver de forma fácil e rápida um código computacional capaz de resolver toda uma classe de problemas unidimensionais de transporte de nêutrons atráves de uma interface gráfica (GUI) amigável ao usuário. Utilizando de uma lista de comando pré-definidos para operações matemáticas, elaboração de botões, janelas e exibição gráfica, o software foi criado de forma que o usuário possa facilmente inserir os parâmetros de entrada, visualizar a saída dos resultados e fazer a análise gráfica, tornando muito simples a sua utilização.

Sendo assim, destaca-se como objetivos deste trabalho a utilização da linguagem *Python* para a implementação de uma formulação baseada no Método ADO, com enfoque no desenvolvimento de uma interface gráfica para utilização do software criado,

bem como fornecer resultados comparáveis com a literatura e análise gráfica dos fenómenos estudados. Para isto, buscou-se aqui apresentar o estudo feito para a classe de problemas de transporte de nêutrons monoenergéticos para um meio heterogêneo em geometria cartesiana unidimensional, considerando espalhamento isotrópico e anisotrópico linear, utilizando condições de contorno prescritas.

Para atingir tais objetivos este trabalho está organizado da seguinte forma: no capitulo 1 é apresentada à Equação de Transporte de Nêutrons e a construção do modelo simplificado, no qual será aplicado o Método ADO. No capitulo 2, é explicado a parte do desenvolvimento da interface gráfica onde parte do código é apresentada. No capitulo 3 faz-se a comparação numérica com a literatura, estuda-se os efeitos da isotropia/anisotropia nos perfis de fluxo escalar e discute-se os resultados .

1 Fundamentação Matemática

A Equação de Transporte [8] corresponde à um modelo matemático capaz de descrever o comportamento médio da população de partículas em um meio hospedeiro. Sua dedução basicamente corresponde ao balanço entre as partículas que migram para dentro ou para fora do espaço de fase e, na sua forma integro-diferencial, é descrita como

$$\frac{1}{v}\frac{\partial}{\partial t}\Psi(\vec{r},\vec{\Omega},E,t)+\vec{\Omega}.\nabla\Psi(\vec{r},\vec{\Omega},E,t)+\sigma_t(\vec{r},E)\Psi(\vec{r},\vec{\Omega},E,t)=Q(\vec{r},\vec{\Omega},E,t)+\int_0^\infty\int_{4\pi}\sigma(\vec{r},E')f(\vec{r},\vec{\Omega}',E'\rightarrow\vec{\Omega},E)\Psi(\vec{r},\vec{\Omega}',E',t)d\vec{\Omega}'dE', \quad (1.1)$$

onde cada termo da Equação (1.1) será descrito por:

- v velocidade escalar,
- $\Psi(\vec{r},\vec{\Omega},E,t)=vN(\vec{r},\vec{\Omega},E,t)$, fluxo angular de nêutrons definido em termos da densidade angular, que representa o número médio de nêutrons, localizado em, $\vec{r}$(vetor posição) e que migram na direção $\vec{\Omega}=\dfrac{\vec{v}}{v}$, (vetor unitário que define a direção de movimento do nêutron), com energia E e tempo t,
- Variação do fluxo angular no espaço de fase com o relação ao tempo,

$$\frac{1}{v}\frac{\partial}{\partial t}\Psi(\vec{r},\vec{\Omega},E,t),$$

- Perda dos nêutrons:

  Por migração,

$$\vec{\Omega}.\nabla\Psi(\vec{r},\vec{\Omega},E,t),$$

  Por colisão,

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

  $\sigma_t(\vec{r},E)$ é a seção de choque macroscópica total definida como sendo a probabilidade de interação entre nêutrons e os núcleos dos átomos.

- Ganhos dos nêutrons:

  Por fonte interna,

$$Q(\vec{r},\vec{\Omega},E,t),$$

  Por migração,

$$\int_0^\infty\int_{4\pi}\sigma(\vec{r},E')f(\vec{r},\vec{\Omega}',E'\rightarrow\vec{\Omega},E)\Psi(\vec{r},\vec{\Omega}',E',t)d\vec{\Omega}'dE',$$

sendo $f(\vec{r}, \vec{\Omega}', E' \rightarrow \vec{\Omega}, E)$ a probabilidade de um nêutron migrando na direção $\vec{\Omega}'$ e com energia E', emergir após uma colisão com um núcleo do meio, na direção $\vec{\Omega}$ e com energia E.

Devido às dificuldades em lidar com a Equação (1.1) diretamente, é comum que se construa modelos matemáticos, nos quais apenas as características físicas de interesse sejam mantidas. Estes modelos são basicamente representações aproximadas do fenômeno, cuja solução pode ser obtida a partir dos diversos métodos existentes na literatura.

1.1 Construção de um modelo simplificado

1) Assumindo que o fenômeno é independente do tempo, pode-se desconsiderar a variável tempo (regime estacionário):

$$\frac{\partial}{\partial t}\Psi(\vec{r}, \vec{\Omega}, E, t) = 0, \tag{1.2}$$

$$\Psi(\vec{r}, \vec{\Omega}, E, t) = \Psi(\vec{r}, \vec{\Omega}, E), \tag{1.3}$$

fazendo com que a Equação (1.1) sejá reescrita na forma,

$$\vec{\Omega}.\nabla\Psi(\vec{r}, \vec{\Omega}, E) + \sigma_t(\vec{r}, E)\Psi(\vec{r}, \vec{\Omega}, E) = Q(\vec{r}, \vec{\Omega}, E) + \int_0^\infty \int_{4\pi} \sigma(\vec{r}, E')f(\vec{r}, \vec{\Omega}', E' \rightarrow \vec{\Omega}, E)\Psi(\vec{r}, \vec{\Omega}', E')d\vec{\Omega}'dE'. \tag{1.4}$$

2) Geometria cartesiana unidimensional, desconsiderando a dependência da simetria azimutal e também redimencionando a função para uma variável espacial:

$$\vec{\Omega}(\mu, \eta, \xi) = \mu, \tag{1.5}$$

$$\vec{r}(x, y, z) = x, \tag{1.6}$$

transformando a Equação (1.4) em,

$$\mu\frac{d}{dx}\Psi(x, \mu, E) + \sigma_t(x, E)\Psi(x, \mu, E) = Q(x, \mu, E) + \frac{1}{2}\int_0^\infty \int_{-1}^1 \sigma(x, E')f(x, \mu', E' \rightarrow \mu, E)\Psi(x, \mu', E')d\mu'dE'. \tag{1.7}$$

3) Regime monoenergético (independência da energia):

$$\mu\frac{d}{dx}\Psi(x, \mu) + \sigma_t(x)\Psi(x, \mu) = Q(x, \mu) + \frac{1}{2}\int_{-1}^1 \sigma(x)f(x, \mu' \rightarrow \mu)\Psi(x, \mu')d\mu'. \tag{1.8}$$

4) Meio heterogêneo em camadas, onde as variáveis σ_t σ, f, independem de x, mas podem variar em cada região α, sendo:

$$\mu\frac{d}{dx}\Psi_\alpha(x, \mu) + \sigma_{t,\alpha}\Psi_\alpha(x, \mu) = Q_\alpha(x, \mu) + \frac{1}{2}\int_{-1}^1 \sigma_\alpha f_\alpha(\mu' \rightarrow \mu)\Psi_\alpha(x, \mu')d\mu'. \tag{1.9}$$

5) Espalhamentos :

$$\sigma_\alpha f_\alpha(\mu' \to \mu) = \sigma_{s0,\alpha} \quad \text{isotrópico,}$$
$$\sigma_\alpha f_\alpha(\mu' \to \mu) = \sigma_{s0,\alpha} + 3\sigma_{s1,\alpha}\mu\mu' \quad \text{anisotrópico linear.}$$

Seguindo uma anisotropia linear, onde a direção interfere na propriedade física, a Equação (1.9) é reescrita como:

$$\mu\frac{d}{dx}\Psi_\alpha(x,\mu) + \sigma_{t,\alpha}\Psi_\alpha(x,\mu) = Q_\alpha(x,\mu) + \\ + \frac{\sigma_{s0,\alpha}}{2}\int_{-1}^{1}\Psi_\alpha(x,\mu')d\mu' + \frac{3}{2}\sigma_{s1,\alpha}\mu\int_{-1}^{1}\mu'\Psi_\alpha(x,\mu')d\mu'. \quad (1.10)$$

Obtido o modelo matemático o qual será estudado, descrito pela Equação (1.10), onde $\sigma_{t,\alpha}$, $\sigma_{s0,\alpha}$ e $\sigma_{s1,\alpha}$ correspondem, respectivamente, as seções de choque macroscópicas total, espalhamento isotrópico e espalhamento anisotrópico linear, em cada camada $\alpha = 1, ..., M$, e aparecem na equação com o objetivo de indicarem as probabilidades com que ocorram os eventos. As variáveis $x \in [0, a]$ e $\mu \in [-1, 1]$ estabelecem a posição e a direção com que o fluxo angular $\Psi_\alpha(x,\mu)$ é medido, e $Q_\alpha(x,\mu)$ corresponde a fonte interna de nêutrons na posição x e direção μ. Vale ressaltar que neste trabalho não serão abordados fontes internas nos problemas estudados.

1.2 Discretização

Relacionado aos métodos de resolução determinísticos, principalmente para os ditos mistos (que envolvem aspectos numéricos e analíticos como o método ADO), a discretização do termo integral de colisão é um passo obrigatório. Para isto, a quadratura numérica de Gauss-Legendre será utilizada, de forma a aproximar os termos integrais pelos somatórios.

$$\int_{-1}^{1}\Psi_\alpha(x,\mu')d\mu' = \sum_{k=1}^{N} w_k\Psi_\alpha(x,\mu_k), \quad (1.11)$$
$$\int_{-1}^{1}\mu'\Psi_\alpha(x,\mu')d\mu' = \sum_{k=1}^{N} w_k\mu_k\Psi_\alpha(x,\mu_k), \quad (1.12)$$

nas quais as direções discretas μ_k estão associadas aos pesos w_k.

Esta discretização da variável angular (associada ao termo integral), transforma a equação de transporte integro-diferencial em um sistema de equações diferenciais ordinárias. Sendo assim, a versão discreta da Equação. (1.10) será dada por,

$$\mu_i\frac{d}{dx}\Psi_\alpha(x,\mu_i) + \sigma_{t,\alpha}\Psi(x,\mu_i) = \\ + \frac{\sigma_{s0,\alpha}}{2}\sum_{k=1}^{N} w_k\Psi_\alpha(x,\mu_k) + \frac{3}{2}\sigma_{s1,\alpha}\mu_i\sum_{k=1}^{N} w_k\mu_k\Psi_\alpha(x,\mu_k), \quad (1.13)$$

Para melhor compreensão do problema heterogêneo a Figura 1 ilustra, o domínio $[0, a]$ dividido em M regiões homogêneas e um ordenamento nas direções que será feito a seguir.

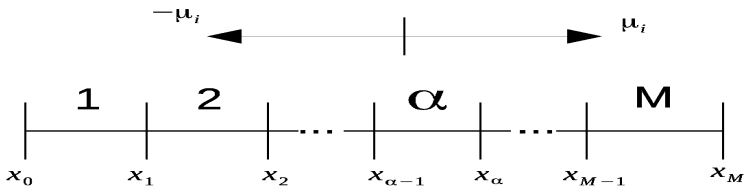

Figura 1 – Divisão do domínio em regiões.

1.3 Método baseado ADO

O método de Ordenadas Discretas Analítico (método ADO) [5] tem se mostrado uma boa alternativa no tratamento de diversos problemas em transporte de partículas e radiação, e tem-se destacado por conta de algumas características que o tornam muito atrativo, do ponto de vista computacional seja pela simplicidade na implementação, ou pela velocidade de resposta se comparado com outros métodos.

Este método consiste basicamente em propor que a solução homogênea seja da forma

$$\Psi_\alpha(x, \pm\mu_i) = \Phi_\alpha(\nu_\alpha, \pm\mu_i)e^{-x/\nu_\alpha}, \tag{1.14}$$

e

$$\frac{d}{dx}\Psi_\alpha(x, \mu_i) = -\frac{1}{\nu_\alpha}\Phi_\alpha(\nu_\alpha, \pm\mu_i)e^{-x/\nu_\alpha}, \tag{1.15}$$

para $i = 1,..., N/2$ e $\alpha = 1,..., M$.

O método apresentado aqui (e que seria indicado como ADO N), onde é feita apenas uma separação entre as direções positivas μ_i das direções negativas $-\mu_i$, com $i = 1, ..., N/2$. Dessa forma, é possível dividir a Equação (1.13) em dois sistemas de EDO's,

$$-\frac{\mu_i}{\nu_\alpha}\Phi_\alpha(\nu_\alpha, \mu_i) + \sigma_{t,\alpha}\Phi_\alpha(\nu_\alpha, \mu_i) = \frac{\sigma_{s0,\alpha}}{2}\sum_{k=1}^{N/2} w_k[\Phi_\alpha(\nu_\alpha, \mu_k) + \Phi_\alpha(\nu_\alpha, -\mu_k)] + \\ + \frac{3}{2}\sigma_{s1,\alpha}\mu_i\sum_{k=1}^{N/2} w_k\mu_k[\Phi_\alpha(\nu_\alpha, \mu_k) - \Phi_\alpha(\nu_\alpha, -\mu_k)] \tag{1.16}$$

e

$$\frac{\mu_i}{\nu_\alpha}\Phi_\alpha(\nu_\alpha, -\mu_i) + \sigma_{t,\alpha}\Phi_\alpha(\nu_\alpha, -\mu_i) = \frac{\sigma_{s0,\alpha}}{2}\sum_{k=1}^{N/2} w_k[\Phi_\alpha(\nu_\alpha, \mu_k) + \Phi_\alpha(\nu_\alpha, -\mu_k)] + \\ - \frac{3}{2}\sigma_{s1,\alpha}\mu_i\sum_{k=1}^{N/2} w_k\mu_k[\Phi_\alpha(\nu_\alpha, \mu_k) - \Phi_\alpha(\nu_\alpha, -\mu_k)], \tag{1.17}$$

para $i = 1,..., N/2$ e $\alpha = 1,..., M$.

Seguindo a metodologia proposta pelo método ADO, utilizam-se as funções auxiliares

$$U_\alpha(\nu_\alpha, \mu_i) = \Phi_\alpha(\nu_\alpha, \mu_i) + \Phi_\alpha(\nu_\alpha, -\mu_i), \tag{1.18}$$

$$V_\alpha(\nu_\alpha, \mu_i) = \Phi_\alpha(\nu_\alpha, \mu_i) - \Phi_\alpha(\nu_\alpha, -\mu_i), \tag{1.19}$$

tais que, somando as Equações. (1.17) e (1.16), depois substituindo as Equações (1.18) e (1.19), obtem-se

$$V_\alpha(\nu_\alpha,\mu_i)=\frac{\nu_\alpha}{\mu_i}[\sigma_{t,\alpha}U(\nu_\alpha,\mu_i)-\sigma_{s0,\alpha}\sum_{k=1}^{N/2}w_kU(\nu_\alpha,\mu_i)]. \tag{1.20}$$

De forma similar, sendo agora subtraída a Equação (1.17) da (1.16) obtem-se

$$\frac{\mu_i}{\nu_\alpha}U(\nu_\alpha,\mu_i)+\sigma_{t,\alpha}V(\nu_\alpha,\mu_i)=3\sigma_{s1,\alpha}\mu_i\sum_{k=1}^{N/2}\mu_kw_kV(\nu_\alpha,\mu_i) \tag{1.21}$$

para $i=1,...,\ N/2$ e $\alpha=1,...,\ M$.

As Equações (1.20) e (1.21), constituem relações entre $U_\alpha(\nu_\alpha,\mu_i)$ e $V_\alpha(\nu_\alpha,\mu_i)$, a qual serão utilizadas para a construção de um problema de autovalores. Assim substituindo a Equação (1.20) na (1.21), obtemos,

$$\begin{aligned}\frac{1}{\nu_\alpha^2}U_\alpha(\nu_\alpha,\mu_i)=&\frac{\sigma_{t,\alpha}^2}{\mu_i^2}U_\alpha(\nu_\alpha,\mu_i)\\&-\sum_{k=1}^{N/2}w_k[\frac{\sigma_{t,\alpha}\sigma_{s0,\alpha}}{\mu_i^2}+3\sigma_{t,\alpha}\sigma_{s1,\alpha}-3\sigma_{s0,\alpha}\sigma_{s1,\alpha}(\sum_{k=1}^{N/2}w_j)]U(\nu_\alpha,\mu_i),\end{aligned} \tag{1.22}$$

que pode ser descrito na forma matricial como

$$[D_\alpha-A_\alpha]U_\alpha=\lambda_\alpha U_\alpha \tag{1.23}$$

onde

$$\lambda_\alpha=\frac{1}{\nu_\alpha^2}, \tag{1.24}$$

e as matrizes de dimensão $N/2$ x $N/2$, são dadas por

$$D_\alpha=diag\left\{\frac{\sigma_{t,\alpha}^2}{\mu_1^2},\ldots,\frac{\sigma_{t,\alpha}^2}{\mu_{N/2}^2}\right\} \tag{1.25}$$

e

$$A_\alpha(i,k)=w_k\left[\frac{\sigma_{t,\alpha}\sigma_{s0,\alpha}}{\mu_i^2}+3\sigma_{t,\alpha}\sigma_{s1,\alpha}-3\sigma_{s0,\alpha}\sigma_{s1,\alpha}\left(\sum_{j=1}^{N/2}w_j\right)\right], \tag{1.26}$$

para $i,j,k=1,...,\ N/2$ e $\alpha=1,...,\ M$.

Solucionado o problema de autovalores, as constantes de separação são obtidas por meio da Equação (1.24) e com base nas Equações (1.18) e (1.19), serão somadas e subtraidas, para se obter as autofunções,

$$\Phi_\alpha(\nu_{j,\alpha},\mu_i) = \frac{1}{2}[U_\alpha(\nu_{j,\alpha},\mu_i) + V_\alpha(\nu_{j,\alpha},\mu_i)], \tag{1.27}$$

$$\Phi_\alpha(\nu_{j,\alpha},-\mu_i) = \frac{1}{2}[U_\alpha(\nu_{j,\alpha},\mu_i) - V_\alpha(\nu_{j,\alpha},\mu_i)], \tag{1.28}$$

para $i,j,k = 1,...,\ N/2$ e $\alpha = 1,...,\ M$.

Vale lembrar que para cada camada α, as constantes de separação $\pm\nu_{j,\alpha}$ são obtidas aos pares e todos os valores são reais.

A partir das Equações (1.27) e (1.28), para que as autofunções Φ_α constituam uma base linearmente independente, considera-se a simetria

$$\Phi_\alpha(\nu_{j,\alpha},\mu_i) = \Phi_\alpha(-\nu_{j,\alpha},-\mu_i), \tag{1.29}$$

$$\Phi_\alpha(\nu_{j,\alpha},-\mu_i) = \Phi_\alpha(-\nu_{j,\alpha},\mu_i). \tag{1.30}$$

Além disso, para os problemas estudados até o momento, fontes internas não foram usadas, fazendo com que a solução geral seja dada pelas equações,

$$\Psi_\alpha(x,\mu_i) = \sum_{j=1}^{N/2} A_{j,\alpha}\Phi_\alpha(\nu_{j,\alpha},\mu_i)e^{-(x-x_{\alpha-1})/\nu_{j,\alpha}} + \\ + A_{j+N/2,\alpha}\Phi_\alpha(-\nu_{j,\alpha},\mu_i)e^{-(x_\alpha-x)/\nu_{j,\alpha}}, \tag{1.31}$$

$$\Psi_\alpha(x,-\mu_i) = \sum_{j=1}^{N/2} A_{j,\alpha}\Phi_\alpha(\nu_{j,\alpha},-\mu_i)e^{-(x-x_{\alpha-1})/\nu_{j,\alpha}} + \\ + A_{j+N/2,\alpha}\Phi_\alpha(-\nu_{j,\alpha},-\mu_i)e^{-(x_\alpha-x)/\nu_{j,\alpha}}, \tag{1.32}$$

para $i = 1,...,\ N/2$ e $\alpha = 1,...,\ M$.

1.4 Condições de contorno e Sistema de acoplamento

Para se ter uma solução geral do problema de forma completa é necessário estabelecer condições de contorno as quais serão submetidos os fluxos angulares. Estas condições podem ser do tipo:

- Contorno discretas prescritas

$$\Psi_1(0, \mu_i) = F, \mu_i > 0, \text{ (a esquerda)} \tag{1.33}$$

$$\Psi_M(a, -\mu_i) = G, -\mu_i < 0 \text{ (a direita).} \tag{1.34}$$

onde $i = 1,..., N/2$

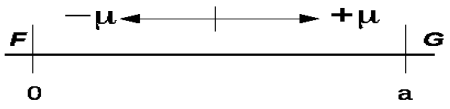

Figura 2 – Condições de contorno prescritas à esquerda e direita.

- Contorno reflexivas

$$\Psi_1(0, \mu_i) = \Psi_1(0, -\mu_i), \text{ (a esquerda)}$$

$$\tag{1.36}$$

$$\Psi_M(a, \mu_i) = \Psi_M(a, -\mu_i), \text{ (a direita)}$$

$$\tag{1.38}$$

onde $i = 1,..., N/2$

Lembra-se aqui que neste trabalho se deu atenção apenas a condições de contorno prescritas.

Vale ressaltar também que para garantir continuidade do fluxo entre as regiões de contato, é necessário utilizar condições descritas na forma

$$\Psi_\alpha(x_\alpha, \pm\mu_i) = \Psi_{\alpha+1}(x_\alpha, \pm\mu_i), \tag{1.39}$$

para $i = 1,..., N/2$ e $\alpha = 1,..., M$-1.

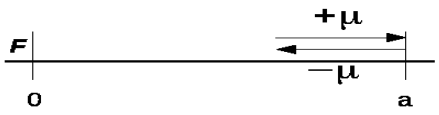

Figura 3 – Exemplo de condições de contorno reflexiva à direita.

Utilizando as condições de contorno e as de interface obtém se um sistema de dimensão MNxMN no qual os valores para os coeficientes $A_{j,\alpha}$ são a obtido como solução do problema. Com isso pode se estabelecer por completo as Equações (1.31) e (1.32), possibilitando a avaliação de algumas quantidades de interesse.

1.5 Grandeza estudada

Até o momento, a grandeza avaliada nos fenômenos estudados foi o fluxo escalar de partículas($n/cm^2.s$) nas camadas α, o qual é descrito pela equação,

$$\phi_\alpha(x) = \frac{1}{2}\int_{-1}^{1} \Psi_\alpha(x,\mu)d\mu, \tag{1.40}$$

que pode ser aproximada numericamente por,

$$\phi_\alpha(x) = \frac{1}{2}\sum_{k=1}^{N/2} w_k[\Psi_\alpha(x,\mu_k) + \Psi_\alpha(x,-\mu_k)], \tag{1.41}$$

$$\tag{1.42}$$

representando a média do fluxo angular em termos da variável direcional μ.

2 Implementação em *Python*

Criada na década de noventa por Guido Van Rossum, a linguagem de programação *Python* [11], tem como seus principais objetivos a produtividade e a facilidade na programação, destacando o fato dela ser de livre distribuição (sem custo financeiro) e por existir versões funcionais para *Windows*, *Linux* e *Mac* (sendo possível a execução pelo terminal nesses dois últimos). Contendo uma vasta biblioteca com inúmeras subrotinas de diversas aplicações como, Numpy e Scipy com enfoque matemático, PyQt e *Tkinter* no desenvolvimento de interfaces gráficas, Matplotlib e Plotly na criação de gráficos, também é possível o desenvolvimento de paginas web, aplicativos para smartphones, entre outros.

Apesar de ter se originado nos anos noventa, o *Python* começou a ganhar espaço nesses últimos anos, tendo uma comunidade de usuários extremamente ativa que vem auxiliando o desenvolvimento de diversas áreas de pesquisa, se destacando tanto no meio acadêmico quanto no profissional, isso tudo devido às inúmeras vantagens vistas.

A escolha da subrotina *Tkinter* neste trabalho, foi devido ao fácil manuseio e a grande quantidade de materiais didáticos, sendo suas rotinas simples e vastamente utilizadas para trabalhos não muito complexos. A criação de softwares auxiliam o estudo, pois facilita a entrada e saída dinâmica de dados, possibilitando uma rápida comparação numérica entre os diversos perfis obtidos no processo de validação.

2.1 Desenvolvimento

Devido as dificuldades encontradas na implementação e com intuito de auxiliar futuros trabalhos aqui será apresentado o código, ressaltando que só é visto o código da interface e não o do método ADO_N.

```
1  # encoding: utf-8
   # encoding: iso-8859-1
3  # encoding: win-1252
   # ============================================================
5  # = PROGRAMA QUE RESOLVE PROBLEMAS DE TRANSPORTE DE NEUTRONS
   # = UNIDIMENSIONAIS, COM ESPALHAMENTO ISOTROPICO, EM REGIME
7  # = ESTACIONARIO, SEM FONTE INTERNA, ATRAVES DO METODO ADO
   # ============================================================
9
   # Plotar o grafico na GUI
11 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,
       NavigationToolbar2TkAgg
```

```
13 from matplotlib.figure                    import  Figure
   import matplotlib.pyplot as plt
15 # ***Quando utilizado "as ..." tornando assim o plt um prefixo
   # para o comando sendo melhor entendido no decorrer do algoritmo.
17
   # Utilizado para exportar arquivo em csv (nao eh necessario usar ele,
19 # mas utilizo devido a facilidade que tenho e no auxilo  para o
   # trabalhado com bancos de dados )
21 import pandas as pd

23 # otimiza o codigo aumentando a performace,
   # entretanto esta em fase de teste
25 from multiprocessing import Pool
   import multiprocessing as mp
27
   # Pacote numerico, montagem de matrizes, tensores  e ...
29 from numpy   import *

31 # Usado para a solucao dos sistemas lineares
   from numpy   import linalg as la
33
   # Exite inumeros integradores numericos ja implementados
35 # ***recomendo a pesquisa no propio site do scipy
   from scipy   import integrate
37
   # Pacote ja mencionado que organiza a criacao e desenvolvimento da GUI
39 from Tkinter import *

41 # retorna uma pop up informando uma mensagem
   import tkMessageBox
43 import Pmw

45 # Pacote desenvolvido, pois a partir dos valores obtidos pela quadratura
   # de gauss-legendre no qual eh necessario realizar auteracoes.
47 # (a quadratura esta na subrotina scipy!)
   import quad
49
   #==inteface grafica==
51 #class Application:
   #         0    1     2          3          4
53 #condicoes basicas iniciais
   basics = 'N', 'nx',  'M'       ,'CE'       ,'CD'
55 #parametros fisicos
   fields = 'M', 'a' ,  'sigma_t' ,'sigma_s'  ,'sigma_s1'
57 #=============================================================

59 show_plot = False
```

```
61 # Quando utilizado def criamos uma funcao no qual x e y
   # sao as variaveis de entrada
63 def plot(x, y):
       fig = plt.figure()
65     ax1 = fig.add_subplot(1,1,1)
       ax1.plot(x,y)
67
       #criacao da grade onde k eh a cor preta, linestyle o estilo que
69     # eh tracejado e linewidth a espessura da linha
       ax1.grid(color='k',linestyle='--', linewidth=1)
71
       # Pode se utilizar min(x) e max(x) nos quais
73     # retorna o menor e o maior valor
       xmin = x.min() -0.5
75     xmax = x.max() +0.5
       #delimita o eixo x com os valores maximos e minimos
77     ax1.set_xlim([xmin,xmax])

79     #de forma similar ao eixo x, mas agora no eixo y
       ymin = y.min() -0.2
81     ymax = y.max() +0.2
       ax1.set_ylim([ymin,ymax])
83
       #legendas nos eixos x e y
85     plt.ylabel('fluxo escalar(n/cm^2.s)')
       plt.xlabel('x(cm)')
87
       # E fig o objeto de saida, no qual eh uma figura, mas podendo
89     # ser um numero, string, vetor, matriz e ....
       return fig
91
   def ajuste(coefs,M):
93     #auxilia na obtencao dos valores
       coef = zeros((M,4))
95     for j in xrange(4):
           for i in xrange(M):
97             coef[i,j] = coefs[i,j].get()

99     return coef

101    # Metodo ADO
       #coloquei apenas as linhas a qual acho importante informar
103 def ADO(entries,check_E,check_D,coefs,M):
       #coeficientes parametros fisicos
105    coef = ajuste(coefs,M)
```

```
107     #[:,1], com os dois pontos(:) na matriz na primeira posicao, pegamos
            todos os valores da 2 coluna no python comeca em 0.
        sigma_t = coef[:,1]
109     sigma_s = coef[:,2]
        sigma_s1 = coef[:,3]
111     a = coef[:,0]
        CE = float(entries['CE'].get())
113     CD = float(entries['CD'].get())
        n = int(entries['N'].get())
115     nx = int(entries['nx'].get())

117     #condicoes reflexivas de contorno na forma binaria
        check_E = int(check_E.get())
119     check_D = int(check_D.get())

121     #mensagem pop up de erro eh verificado todos os M meios
        for i in xrange(len(sigma_t)):
123         if ((sigma_s[i] >= sigma_t[i]) or (sigma_s1[i]>= sigma_t[i])):
              tkMessageBox.showerror("ERRO","entre com (sigma_t > sigma_s1) ou
                  (sigma_t > sigma_s)")
125           tkMessageBox.resizable(width=False, heidth=False)

127     #funcao criada
        u,w = quad.quadratura(n)
129
        #FINAL DO CODIGO DO ADO, mas ainda esta na funcao ADO,
131     # sendo agora plotado o grafico e exportando o arquivo em csv
        df = pd.DataFrame(matrix)
133     df.to_csv('fluxo_N(%d)_M(%s).csv' %(n,M),header = True,index = True,
            mode ='w',encoding='utf-8')

135     PHI_INPUT.insert(0.0,phi)                          #PRINT PHI NA GUI

137     #===GRAFICO OUTPUT===
        # Sendo plot a primeira funcao definida no inicio do codigo,
139     # no qual retorna a figura mencionada
        fig = plot(x, phi)                                 #PLOT GRAFICO
141     canvas = FigureCanvasTkAgg(fig, master=root)       #PLOT GRAFICO
        #janela onde eh exibido o grafico
143     canvas.show()
        canvas.get_tk_widget().place(x= 700, y = 0,height=500, width=500)
145     toolbar = NavigationToolbar2TkAgg(canvas, root)
        #toolbar onde podemos salvar o grafico, ampliar e reduzir
147     #***obs: podesse salvar o grafico ja ampliado
        toolbar.place(x= 700, y = 500)
149     toolbar.update()
        return x,phi
```

```
151 def makeform_basic(root,basics):
    #condicoes iniciais
153 # criacao de uma borda destacada
    COND_BASIC=Frame(root,width=370,height=100,bd=2,relief=GROOVE)
155 #coloracao da borda
    COND_BASIC.configure(background='gray92')#
157 #posicionamento
    COND_BASIC.place(x=10,y=10)
159 #Titulo e posicionamento dele
    Label(text='CONDICOES INICIAIS',background='gray92').place(x=20, y=1)
161 labels = []
    entries = {}
163
    #===Posicionamento===
165 for basic in basics: #forma alternativa de percorrer um vetor
        if basic in ('CE','CD'): #as CE e CD ficam a esquerda de N, nx e M.
167         aux_x = 110
            aux_y = 90
169     else:
            aux_x =20
171         aux_y = 0

173     labels = Label(root,text=basic+':', anchor='w') #eh possivel somar
            stings como no caso basic(sting) + ':'
        labels.configure(background='gray92')
175     labels.place(x=aux_x,y=20+(30*basics.index(basic) - aux_y ))
        #index.() retorna a posicao dentro do vetor
177
      #cria uma lacuna onde sera informado os valores de entrada
179     ent = Entry()
        ent.place(x=30+aux_x,y=20+(30*basics.index(basic) - aux_y ),width =
            50, height = 20)
181     ent.insert(END,0 )
        #com isso associa um valor aos strings no vetor basics
183     entries[basic] = ent

185 #check das condicoes reflexivas definida como uma variavel inteira
    check_E = IntVar()
187 #no qual ela eh 0
    check_E.set(0)
189 CBE = Checkbutton(root, variable = check_E,text="espelhamento esquerda"
        , onvalue=1, offvalue=0)
    CBE.configure(background='gray92')
191 CBE.place(x = 200 ,y = 20)


193
```

```
195     check_D = IntVar()
        check_D.set(0)
197     CBD = Checkbutton(root, variable = check_D,text="espelhamento direita",
            onvalue=1, offvalue=0)
        CBD.configure(background='gray92')
199     CBD.place(x = 200 ,y = 50)

201     #com isso eh retorna as condicoes basicas
        return (entries,check_E,check_D)
203
labels = []
205 def makeform_coef(root,basics,E,D):
        #estrutura sobre o frame principal o qual sera gerado a matriz de
            paramentros fisico
207     frame = Frame(root, width=350, height = 470,bg = 'gray92')
        frame.place(x = 10 , y = 150,)
209
        # .get() pegamos as variaveis necessarias
211     N = int(basics['N'].get())
        M = int(basics['M'].get())
213     nx = int(basics['nx'].get())

215     #Matriz onde sera informado os parametros fisicos de cada meio alpha
        rows = []
217     for i in xrange(M):
            cols  = []
219         for j in xrange(4):
                labels_1 = Label(frame,text=str(i+1),anchor='w')
221             labels_1.configure(background='gray92')
                labels_1.place(x=5,y=5+(30*i))
223
                aux_E = Entry(frame)
225             aux_E.place(x=30 + 60*j,y=5+(30*i),width = 50, height = 20)
                aux_E.insert(END, 0.0)
227             #append() adiciona o valor na ultima posicao do vetor
                cols.append(aux_E)
229             #adiciona o vetor no vetor
            rows.append(cols)
231
        #matrix() transformamos um vetor de dimensao 1 com inumeros vetores em
            cada posicao em uma matrix
233     coefs = matrix(rows)
        pool = Pool(processes=4)
235     #obs: primeiramente eh criado um botao falso mas agora e criado um real
            onde o comando envias as variaveis a funcao ADO
        b1 = Button(root, text='Calcular',command=(lambda e=ents_1: mp.Process(
            ADO(e,E,D,coefs,M))))
```

```
237     b1.place(x= 390,y = 10)

239 #===FUNCAO PRINCIPAL===
    if __name__ == '__main__':
241     rows = []
        cols  = []
243
        root = Tk() #==========ESTRUTURA PRINCIPAL==========
245     root.title('PROGRAMA QUE RESOLVE PROBLEMAS DE TRANSPORTE DE NEUTRONS(
            DREIZEHN)')

247     #tamanho da interface
        w = 1200
249     h = 600
        x = (root.winfo_screenwidth()/2) - (w/2)
251     y = (root.winfo_screenheight()/2) - (h/2)

253     root.fontepadrao = ('verdana', '10')            #fonte padrao e tamanho
        root.minsize(width=w, height=h)
255     root.maxsize(width=w, height=h)
        #impossibilita alterar o tamanho
257     root.resizable(width = FALSE, height = FALSE)

259     #definindo tamanho da pagina
        root.geometry('%dx%d+%d+%d' % (w, h, x, y))
261     root.configure(background='gray92')             #cor de fundo
        Label(text='PARAMETROS FISICOS',background='gray92').place(x=20, y=113)
263
        # linha dos nomes dos parametros fisicos
265     for field in fields:
            j = fields.index(field)
267         labels_2 = Label(root,text= field ,anchor='center')
            labels_2.configure(background='gray92')
269         aux_x =[17,59,99,159,219]
            labels_2.place(x=aux_x[j],y=130)
271
        #===BUSCA DADOS DE ENTRADA EM FIELDS===
273     ents_1,E,D = makeform_basic(root, basics)

275     #===B0TOES===
        #obs: nao existe nenhum comando associado
277     b1 = Button(root, text='Calcular',)
        b1.place(x= 390,y = 10)
279
        #botao de saida do root
281     b2 = Button(root, text='Quit', command=root.quit)
        b2.place(x= 470,y = 10)
```

```
283
    #eh necessario 2 botoes um com as condicoes basicas no qual a seguir
        criar a matriz onde eh informado os parametros
285 b3 = Button(root, text='INICIO',command=(lambda ents_1=ents_1:
        makeform_coef(root,ents_1,E,D) ) )
    b3.place(x= 300,y = 75,width = 50, height = 20)
287
    #===PRINT PHI===
289 #informa o fluxo escalar de particulas (eh necessario ajustar)
    PHI_INPUT = Text(root, width = 44, height = 35, takefocus=0)
291 PHI_INPUT.place(x= 380,y= 80)

293 #===PLOT GRAPH===
    #funcoes de plot que foram definidas
295 fig = plt.figure()
    f = Figure(figsize=(70,70), dpi=80)
297 a_fig = fig.add_subplot(111)

299 canvas = FigureCanvasTkAgg(fig, master=root)
    canvas.get_tk_widget().configure(background='gray92', highlightcolor='
        gray92', highlightbackground='gray92')
301 canvas.get_tk_widget().place(x= 700, y = 0,height=500, width=500)
    toolbar = NavigationToolbar2TkAgg(canvas, root)
303 toolbar.place(x= 700, y = 500)
    toolbar.update()
305
    #mantem a janela aberta
307 root.mainloop()
```

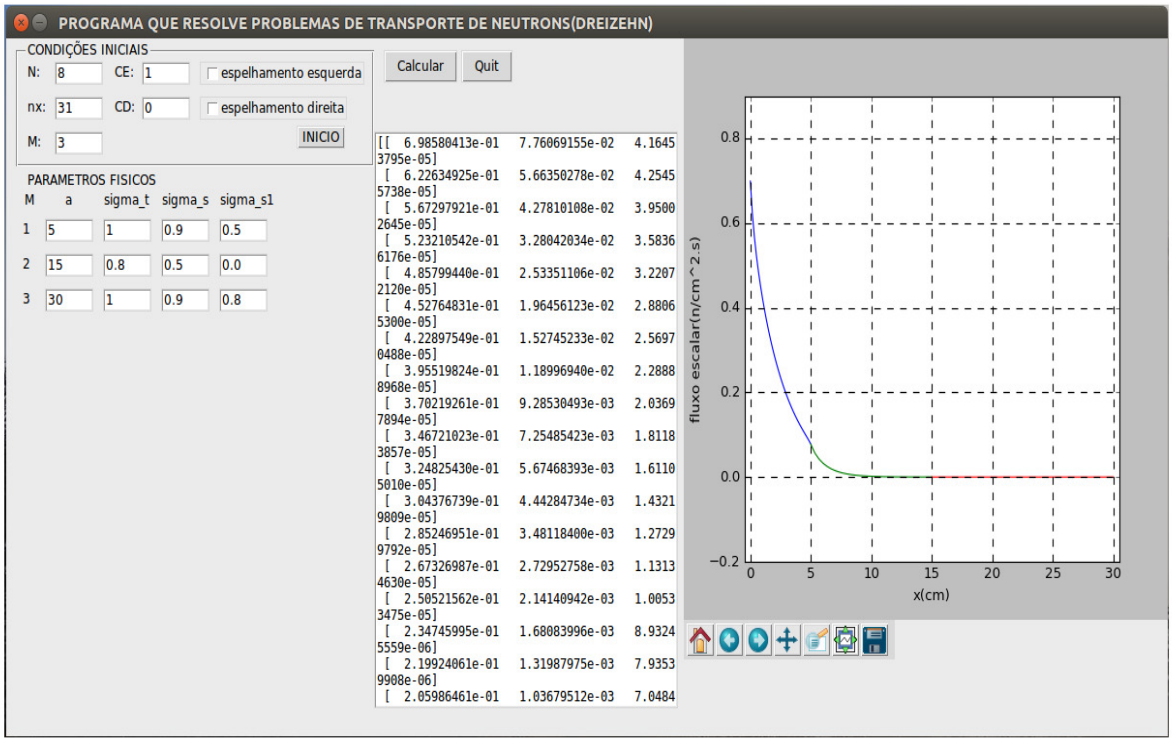

Figura 4 – Ambiente gráfico desenvolvido em *Python* utilizando a biblioteca *Tkinter*.

A partir deste código é possível desenvolver outras interfaces com aplicações em diversas áreas de estudos.

Além disso, pode-se fazer a visualização gráfica e obtenção completa dos perfis de fluxo escalar, podendo informar o grau da quadratura desejada, lembrando que não é necessário usar valores altos devido a rápida convergência.

O programa segue uma organização onde primeiramente são lidas as funções e depois executada a interface. Após a inserção dos dados de entrada (que compreendem os parâmetros físicos e as condições de contorno) executa-se o código e então, é fornecido (exportado em arquivo csv) o fluxo escalar na forma de perfil numérico e o gráfico associado é apresentado (sendo possível salvar a imagem).

3 Resultados Numericos

Resultados obtidos aqui serão comparados com a literatura a fim de validar o código e o método. Primeiramente serão abordados problemas homogêneos, considerando efeitos de espalhamento isotropico e anisotropico linear para diferentes ordens de quadratura.

Entre os métodos aqui comparados, explica-se abaixo aspectos relativos a sua aplicação,

- Método Degrau: o fluxo angular médio no interior de cada célula espacial é igual ao fluxo da próxima face na direção de varredura, utilizando de uma equação auxiliar da seguinte forma,

$$\Psi_{m,i} = \begin{cases} \Psi_{m,i+\frac{1}{2}} \text{ para } \mu_m > 0, \\ \Psi_{m,i-\frac{1}{2}} \text{ para } \mu_m < 0, \end{cases} \tag{3.1}$$

  para m = 1,..., N em cada célula i.

  i = nodo/célula, m = direção.

- Método Nodal Constante (CN): o fluxo angular médio é calculado de forma a preservar a solução analítica do problema de absorvedor puro. Os parâmetros $\theta_{m,i}$ são calculados de forma que as soluções analíticas gerais do problema S_N sejam preservadas no interior de cada célula espacial considerando a fonte de espalhamento como constante, utilizando de uma equação auxiliar da seguinte forma,

$$\Psi_{m,i} = \frac{(1+\theta_{m,i+\frac{i}{2}})\Psi_{m,i+\frac{i}{2}} + (1-\theta_{m,i-\frac{i}{2}})\Psi_{m,i-\frac{i}{2}}}{2} \tag{3.2}$$

  com

$$\theta_{m,i} = \coth(\frac{\alpha_{m,i}}{2}) - \frac{2}{\alpha_{m,i}}, \alpha_{m,i} = \frac{h_i \sigma_{t,i}}{\mu_m}.$$

  para m = 1,..., N em cada célula i.

- Método Diamond difference (DD): o fluxo angular médio é aproximado por uma média aritmética entre os fluxos angulares nas faces dos nodos, assumindo que o fluxo angular é uma reta no interior de cada célula espacial.

$$\Psi_{m,i} = \frac{\Psi_{m,i+\frac{1}{2}} + \Psi_{m,i-\frac{1}{2}}}{2}, m = 1...N, \tag{3.3}$$

  para m = 1,..., N em cada célula i.

- Método SGF: utilizasse a função de green em cada no nodo como sendo o valor do fluxo angular médio na direção m, apenas a uma unidade incidente do fluxo angular no canto do nodo na direção n, os coeficientes $\theta_{m,n}$ são calculados de forma a preservarem a componente homogênea da solução analítica da equação, assim os valores $\theta_{m,n}$ são calculados resolvendo o sistema de equações,

$$\frac{2a_m(v_k)v_k}{h_i\sigma_t} senh(\frac{h_i\sigma_t}{2v_k}) = e^{\frac{h_i\sigma_t}{2v_k}} \sum_{\mu_n>0} \theta_{m,n}a_n(v_k) + e^{\frac{h_i\sigma_t}{2v_k}} \sum_{\mu_n<0} \theta_{m,n}a_n(v_k). \tag{3.4}$$

  Os valores de $a_m(v_k)$ são autovetores com m= 1,...,N e v_k os autovalores obtidos a partir da análise espectral das equações. Os autovetores são dados pela expressão,

$$a_m(v_k) = \frac{c_0 v_k}{2(v_k - \mu_m)}, \tag{3.5}$$

  para $m = 1,...,N$ em cada célula i, e os autovalores são obtidos da solução da relação de dispersão

$$\frac{c_0 v_k}{2} \sum_{m=1}^{N} \frac{\omega_m}{v_k - \mu_m} = 1. \tag{3.6}$$

Ressalta-se que aqui serão apresentados também alguns resultados originais, nos quais são feitas variações de algum dos parâmetros para uma melhor compreensão dos seus efeitos.

Considerando como Problema 1 o caso de um domínio homogêneo (M=1) tendo condições de contorno prescritas do tipo $\Psi(0, \mu_i) = 1.0$ e $\Psi(100, -\mu_i) = 0.0$ para i = 1,...,$N/2$, seguindo os parâmetros descritos na Tabela 1, onde se faz uma comparação do método estudado aqui com os apresentados na Tabela 2.

Tabela 1 – Parâmetros físicos do problema.

Região	espessura (cm)	$\sigma_t(cm^{-1})$	$\sigma_{s0}(cm^{-1})$	$\sigma_{s1}(cm^{-1})$	N
1	$0 < x < 100$	1.0	0.99	0.8	4

Tabela 2 – Comparação do fluxo escalar para um dominio homogêneo segundo diferentes métodos

Métodos	0cm	50cm	100cm
Degrau	8.077971E-01	2.176060E-02	2.336640E-04
CN	8.163946E-01	1.802246E-02	1.516540E-04
DD	8.222555E-01	1.650595E-02	1.230556E-04
SGF	8.222603E-01	1.653832E-02	1.235337E-04
LTS_N	8.2226E-01	1.6538E-01	1.2353E-4
ADO_N	8.22255597E-01	1.65375974E-03	1.23528984E-14

Tabela 3 – convergência do Problema homogêneo

N	0cm	50cm	100cm
2	8.17255967e-01	1.69911848e-02	1.29181605e-04
4	8.22255597e-01	1.65375974e-02	1.23528984e-04
8	8.22836109e-01	1.64704922e-02	1.22501430e-04
16	8.22957039e-01	1.64564930e-02	1.22287556e-04
32	8.22985442e-01	1.64532042e-02	1.22237336e-04
64	8.22992365e-01	1.64524026e-02	1.22225097e-04
128	8.22994076e-01	1.64522045e-02	1.22222072e-04
256	8.22994501e-01	1.64521552e-02	1.22221319e-04
512	8.22994608e-01	1.64521429e-02	1.22221132e-04

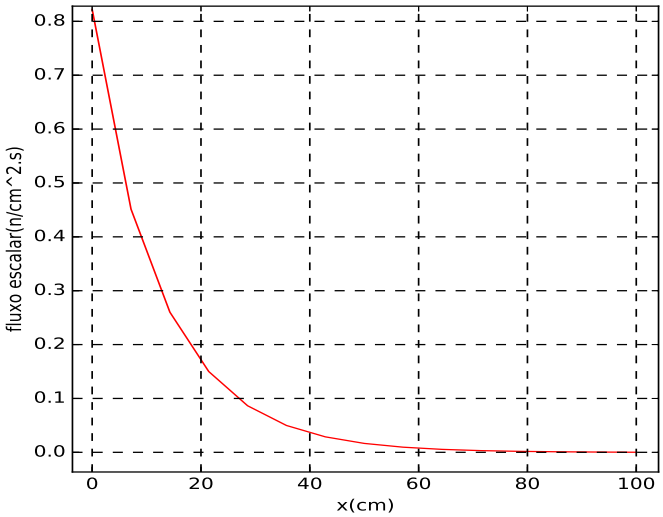

Figura 5 – Gráfico associado a Tabela 2

Os resultados obtidos pelo método estudado apresentam uma ótima concordância com a literatura,(Oliveira [4]), principalmente com os métodos SGF e LTS_N que tem um menor erro nos pontos 50 e 100 cm. A Figura 5 corresponde ao gráfico deste problema para uma melhor compreensão do fenômeno em meio homogêneo.

Sendo agora comparados resultados para o Problema 2, que trata de um meio heterogêneo em camadas, entre o Método ADO_N os Métodos SGF,LTS_N,[3], com condições de contorno prescritas $\Psi(0, \mu_i) = 1.0$ e $\Psi(100, -\mu_i) = 0.0$ para $i = 1,...,N/2$, e meio tem espessura $a = 100cm$, dividido em três camadas (M=3), com os parâmetros apresentados na Tabela 3, sendo utilizados 2, 4 e 8 pontos de quadratura.

Tabela 4 – Meio heterogêneo, espessura e parâmetros de cada região

Região	espessura(cm)	$\sigma_t(cm^{-1})$	$\sigma_{s0}(cm^{-1})$	$\sigma_{s1}(cm^{-1})$
1	$0 < x < 20$	1.0	0.9	0.8
2	$20 < x < 70$	0.6	0.4	0.3
3	$70 < x < 100$	1.0	0.9	0.8

Tabela 5 – Problema 2: comparação numérica dos resultados de fluxo escalar para os métodos SGF_N, LTS_N e ADO_N.

Método	0cm	20cm	70cm	100cm
SGF_2	0.58578E0	0.40531E-2	0.26614E-11	0.14190E-14
LTS_2	0.58578E0	0.40531E-2	0.17281E-9	0.14190E-14
ADO_2	0.585784502E0	0.405305902E-02	0.266144899E-11	0.141896882E-14
SGF_4	0.60819E0	0.41468E-2	0.31817E-10	0.24196E-13
LTS_4	0.60819E0	0.41468E-2	0.31816E-10	0.24196E-13
ADO_4	0.608187686E0	0.414859717E-03	0.318166173E-10	0.241960194E-13
SGF_8	0.6112E0	0.41033E-2	0.32316E-10	0.24323E-13
LTS_8	0.6112E0	0.41033E-2	0.32316E-10	0.24323E-13
ADO_8	0.611117479E0	0.410326160E-03	0.323162207E-10	0.243233334E-13

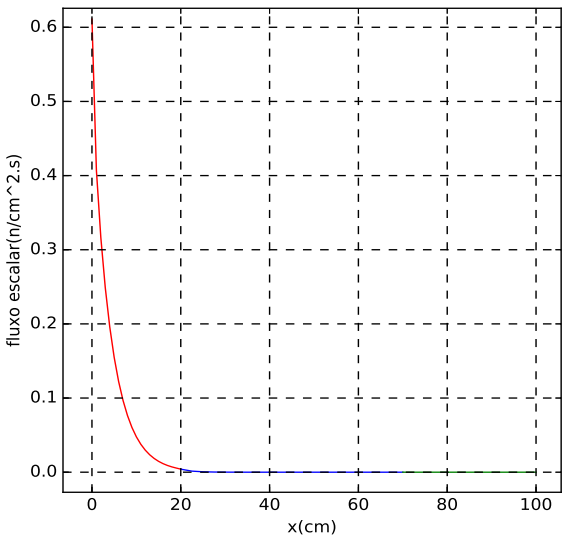

Figura 6 – Gráfico associado a Tabela 5.

Novamente, após a comparação dos resultados com a literatura, observa-se na Tabela 4 uma concordância excepcional chegando a cinco dígitos significativos, na comparação do ADO_N com os demais métodos, ressalta-se aqui que utilizou-se quadratura até $N = 8$ apenas para a comparação com a literatura, pois o programa é capaz de simular para valores maiores que N.

No Problema 3 será analizada a convergência dos resultados, considerando os fatores de espalhamento isotrópico e anisotrópico linear em meio heterogêneo, com condições de contorno prescritas $\Psi(0, \mu_i) = 1.0$ e $\Psi(40, -\mu_i) = 0.0$ para $i = 1,...,N/2$.

Tabela 6 – Meio heterogêneo com características físicas diferentes.

Região	espessura(cm)	$\sigma_t(cm^{-1})$	$\sigma_{s0}(cm^{-1})$	$\sigma_{s1}(cm^{-1})$
1	$0 < x < 5$	1.0	0.9	0.8
2	$5 < x < 15$	0.9	0.8	0.1
3	$15 < x < 20$	0.6	0.4	0.3
4	$20 < x < 40$	1.0	0.99	0.8

Tabela 7 – Problema 3: Análise de convergência do método estudado.

N	0	5	15	20	40
2	5.9968303e-01	2.2836672e-01	1.0401725e-03	1.9179842e-04	1.5329484e-05
4	6.2083102e-01	2.0864250e-01	1.1388227e-03	2.6081962e-04	2.3234407e-05
8	6.2353083e-01	2.0678765e-01	1.1294955e-03	2.6034514e-04	2.3047250e-05
16	6.2410048e-01	2.0639354e-01	1.1275168e-03	2.6011784e-04	2.3000443e-05
32	6.2423528e-01	2.0630078e-01	1.1270475e-03	2.6006004e-04	2.2989083e-05
64	6.2426828e-01	2.0627815e-01	1.1269326e-03	2.6004538e-04	2.2986274e-05
128	6.2427645e-01	2.0627255e-01	1.1269041e-03	2.6004168e-04	2.2985575e-05
256	6.2427849e-01	2.0627116e-01	1.1268970e-03	2.6004076e-04	2.2985400e-05
512	6.2427899e-01	2.0627081e-01	1.1268952e-03	2.6004052e-04	2.2985357e-05

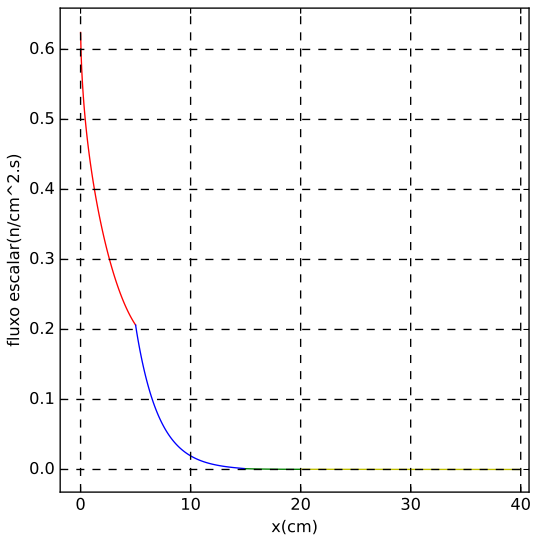

Figura 7 – Gráfico associado a Tabela 7, N = 64.

Devido ao elevado valor do fator de anisotrópia (σ_{s1}) presente nos Problemas 1 e 2, pode-se observar uma demora na convergência se comparado com a formulação *ADO* apresentada em outras referências [21]. Utilizando-se valores de N até 512, onde observou-se uma convergência de 4 à 5 dígitos significativos, o que comprova a dificuldade causada pela anisotrópia na convergência.

3.1 Efeitos da isotropia e anisotropia

Aqui serão apresentadas as diferenças dos efeitos entre os coeficientes de espalhamento isotrópico e anisotrópico, utilizando os gráficos e tabelas para uma melhor comparação,da variação destes coeficientes. Para isto, tratou-se como primeiro caso (Problema 4) como sendo que $\sigma_{s1} = 0.0$ e diferentes valores de σ_{s0} para estudar a influência da isotropia. No segundo caso (Problema 5) utilizou-se $\sigma_{s0} = 0.0$ e diferentes valores de σ_{s1} para estudar os efeitos da anisotropia linear. Para todos os testes considerou-se o meio como sendo homogêneo e os parâmetros são $\sigma_t = 1$, $N = 8$, $\Psi(0, \mu_i) = 2.0$ e $\Psi(4, -\mu_i) = 0.0$ para $i =$ 1,...,$N/2$ e $a \in [0, 4]$.

Tabela 8 – Perfil do fluxo escalar para o Problema 4.

σ_{s0}	0	1	2	3	4
0.1	1.0263336	0.1649148	0.04383275	0.01275161	0.00379312
0.5	1.1715550	0.28966650	0.09734324	0.03397235	0.01003153
0.8	1.3813380	0.56331191	0.26399235	0.12028093	0.03747838
0.9	1.5154393	0.79588434	0.44674992	0.23367199	0.07559687

Tabela 9 – Perfil do fluxo escalar para o Problema 5.

σ_{s1}	0	1	2	3	4
0.1	0.9798777	0.1546541	0.04214988	0.01264624	0.00414000
0.5	0.8718140	0.18667271	0.06961072	0.02839236	0.01398517
0.8	0.7243292	0.20483595	0.10934817	0.06520924	0.04864242
0.9	0.6372906	0.19995629	0.12983281	0.09489742	0.08432245

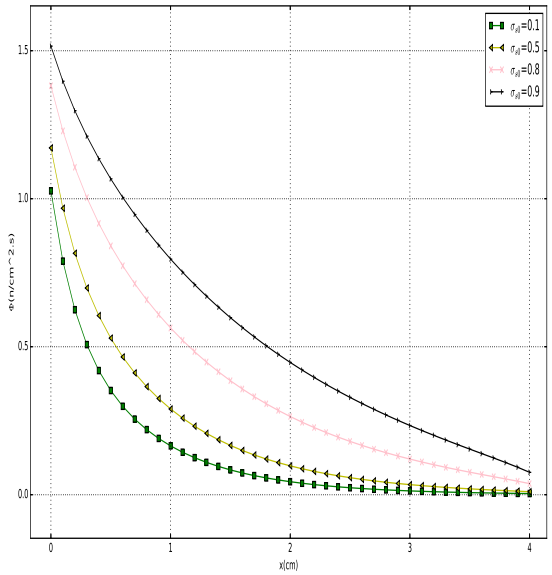

Figura 8 – Perfil do fluxo escalar baseado na Tabela 8.

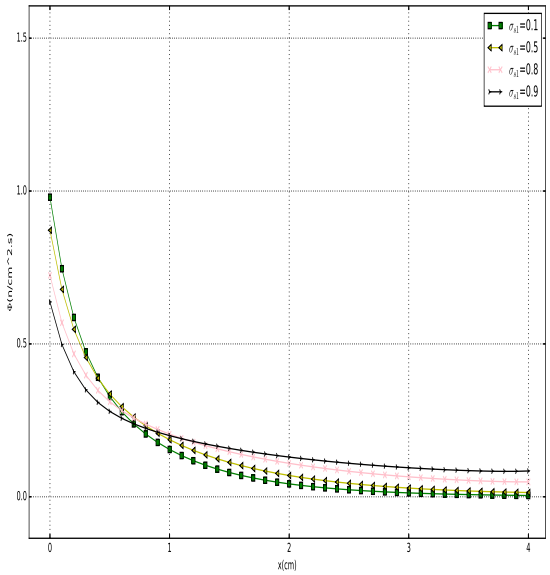

Figura 9 – Perfil do fluxo escalar baseado na Tabela 9.

Na Figura 8 pode ser observado o efeito do fator de isotropia, cujo aumento de seu valor acarreta também no aumento dos valores do perfil de fluxo escalar.

Por outro lado, como pode ser visto na Figura 9, a anisotropia contribui para o aumento dos perfis de fluxo escalar apenas longe da parede com a fonte. em comparação ao domínio todo, também observa-se que o aumento do fator de anisotrópia causa um achatamento do perfil de fluxo escalar e acarreta uma queda abrupta do seu valor nas proximidades da parede onde está a fonte.

Nas duas situações o comportamento exponencial (decrescente) se manteve, atingindo valores maiores nos perfis em que o espalhamento isotrópico é mais representativo.

4 Conclusões

O método apresentado aqui mostrou resultados bem similares aos encontrados na literatura e com comportamento físico coerente ao que se propôs com o modelo matemático, tanto em meio homogêneo quanto heterogêneo, seguindo os espalhamentos isotrópico e anisotrópico linear, para a classe de problemas propostos.

Com o objetivo de contribuir para o aprimoramento do método e para a área de transporte de nêutrons, foi implementado um código em linguagem *Python* que possibilitou o desenvolvimento de uma interface gráfica que de forma bem simples, permitiu ao usuário informar os parâmetros iniciais (ordem de quadratura, quantidade de meios, espessura das camadas e parâmetros físicos), calcular os perfis de fluxo escalar, possibilitando analisar numericamente ou por gráfico os resultados, o que auxiliou muito os estudos. Ainda, tem a versatilidade de salvar os resultados em arquivos externos e permite que os dados de entrada sejam facilmente alterados, se necessário.

O método apresenta um custo computacional baixo devido ao tratamento analítico da variável espacial, pelo problema de autovalores ter um tamanho reduzido se comparado com as outras formulações existentes na literatura e por não depender de esquemas iterativos ou métodos de interpolação. Utilizou-se um processador Intel I7 com 8Gb de Ram e o tempo de processamento variou de alguns segundos a poucos minutos, dependendo do valor de N e M.

A simplicidade do método apresentado, aliado a versatilidade da linguagem *Python*, facilitou o desenvolvimento do algoritmo para a resolução dos problemas propostos. Entretanto, apesar de toda a literatura disponível o desenvolvimento do ambiente gráfico foi um processo mais lento devido a não familiaridade com o pacote *Tkinter*.

Entendendo que os objetivos propostos para este trabalho foram atingidos, propõe-se como continuidade a implementação de soluções particulares para fontes internas no domínio, elevação do grau de anisotropia, cálculo de outras quantidades de interesse (refletância, transmitância), aparência e funcionalidade da interface.

Referências

1 Azmy, Y., and Sartori, E., Nuclear Computational Science: A Century in Review. Springer, 2010. Citado na página 10.

2 Badruzzaman, A., Computational methods in nuclear geophysics, Progress in Nuclear Energy, vol. 25, pp. 265–290, 1991. Citado na página 10.

3 Barichello, L. B., Formulação analítica para soluções do problema de ordenada discreta unidimensional. Tese de doutorado do Programa de Pós Graduação em Engenharia Mecânica, UFRGS, Porto Alegre, 1992. Citado 2 vezes nas páginas 10 e 33.

4 Oliveira, F. B. S., Problema Inverso de Reconstrução Analítica Aproximada da Solução da Equação Monoenergética de Transporte de Partículas Neutras em Geometria Unidimensional Cartesiana com Espalhamento Isotrópico. Tese de doutorado do programa de Pós-Graduação em Modelagem Computacional do Instituto Politécnico, UERJ, Rio de janeiro, 2007. Citado na página 33.

5 Barichello, L. B., Siewert, C. E., A discrete-ordinates solution for non-grey model with complete frequency redistribution JQSRT,vol 62 pp. 665-675, 1999. Citado 2 vezes nas páginas 10 e 17.

6 Barichello, L. B., Rodrigues P., Siewert, C. E., An analitical discrete-ordinates solution for dual-mode heat transfer in a cylinder, JQSRT, vol. 73 pp. 583-602, 2002. Citado na página 10.

7 Barichello, L. B., Cabrera L. C., and Prolo Fiho, J.F., An analytical approach for a nodal scheme of two-dimensional neutron transport problems, Annals of Nuclear Energy, v. 38, pp. 1310-1317, 2011. Citado na página 10.

8 Bell, G., Glasstone, S., Nuclear reactor theory. Robert E. Krieger Publishing Company, New York, 1979. Citado na página 13.

9 Chandrasekhar, S., Radiative Transfer, Oxford University Press, 1950. Citado na página 10.

10 Chaudhary, B., Tkinter GUI application development blueprints, Packt Publishing, 2015. Citado na página 11.

11 Borges, L. E., Python para desenvolvedores 2ª edição, Edição do Autor, 2010. Citado 2 vezes nas páginas 11 e 22.

12 Garcia, R. D. M., Métodos para solucão da equação de transporte de partículas integro-diferencial, Escola de Verão em Teoria de Transporte de Partículas Neutras, Puc - Porto Alegre, 2002. Citado na página 10.

13 Gunay, M., Sarer, B., and Han cerliogullari, A., Three-dimensional Monte Carlo calculation of gas production in structural material of APEX reactor for some evaluated data file, Annals of Nuclear Energy, vol. 55, pp. 292–296, 2013. Citado na página 10.

14 Hu, B. X., Wu, Y. W., Tian, W. X., Su, G. H., and Qiu, S. Z., Development of a transient thermal-hydraulic code for analysis of China Demonstration Fast Reactor, Annals of Nuclear Energy, vol. 55, pp. 302–311, 2013. Citado na página 10.

15 Modest, M., Radiative Heat Transfer, Academic Press, 2013. Citado na página 10.

16 Prolo Filho, J. F., Rodrigues, M. P., A closed form solution for a one-dimensional multi-layered neutron transport problem by Analytical Discrete Ordinates Method, Comparison of Four Thermo-Mechanical Models for Simulating Reactive Flow in Porous Materials, v. 372, pp. 50-59, 2017. Citado na página 10.

17 Shultis, J. K., and Myneni, R. B., Radiative transfer in vegetation canopies with anisotropic scattering, J. Quant. Spectrosc. Radiat. Trans., Vol. 39, pp. 115-129, 1988. Citado na página 10.

18 Schulz, D., Métodos analíticos e computacionais em geofísica nuclear, Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Programa de Pós-graduação em Matemática Aplicada, 2014. Citado na página 10.

19 Sun, Q., Boyd, I. D., Candler, G. V., Numerical simulation of gas flow over microscale airfoils, AIAA Thermophysics Conference, 35th, Anaheim, CA, vol. 16, pp. 171-179, 2002. Citado na página 10.

20 Wick, G. C., Uber ebene diffusion problem, Z. Phys., vol. 120, pp. 702-705, 1943. Citado na página 10.

21 Rodrigues, M. P., Tratamento de problemas de transporte unidimensionais por meio do método ADO, Dissertação de Mestrado, Universidade Federal do Rio Grande, Programa de Pós-graduação em Engenharia Oceânica, 2017. Citado na página 36.

APÊNDICE A

Dedução da proposta de solução utilizada na formulação do método ADO neste trabalho.

O decaimento exponencial na variável espacial que se encontra na solução da Eq.

$$\mu\frac{d}{dx}\Psi(x,\mu)+\sigma_t\Psi(x,\mu)=\frac{\sigma_{s0}}{2}\int_{-1}^{1}\Psi(x,\mu')d\mu' \tag{1}$$

pode ser deduzido através do método de separação de variáveis. no qual propõe-se que

$$\Psi(x,\mu)=X(x,\nu)\Phi(\nu,\mu), \tag{2}$$

onde ν é um parâmetro arbitrário. Desta maneira, a Eq. 1 é reescrita como

$$\mu\frac{d}{dx}X(x,\nu)\Phi(\nu,\mu)+\sigma_tX(x,\nu)\Phi(\nu,\mu)=\frac{\sigma_{s0}}{2}\int_{-1}^{1}X(x,\nu)\Phi(\nu,\mu)d\mu', \tag{3}$$

onde os termos constantes podem ser removidos do termo diferencial e integral, de acordo com

$$\mu\Phi(\nu,\mu)\frac{d}{dx}X(x,\nu)+\sigma_tX(x,\nu)\Phi(\nu,\mu)=\frac{\sigma_{s0}}{2}X(x,\nu)\int_{-1}^{1}\Phi(\nu,\mu)d\mu', \tag{4}$$

dividindo a Eq. (4) por $\mu X(x,\nu)\Psi(\nu,\mu)$, obtendo,

$$\frac{1}{X(x,\nu)}\frac{d}{dx}X(x,\nu)+\frac{\sigma_t}{\mu}=\frac{\sigma_{s0}}{2\Phi(\nu,\mu)}\int_{-1}^{1}\Phi(\nu,\mu)d\mu', \tag{5}$$

Portanto, pode-se fazer que,

$$\frac{1}{X(x,\nu)}\frac{d}{dx}X(x,\nu)+=\frac{\sigma_{s0}}{2\Phi(\nu,\mu)}\int_{-1}^{1}\Phi(\nu,\mu)d\mu'-\frac{\sigma_t}{\mu}=\frac{1}{\nu}, \tag{6}$$

tendo-se assim duas equações diferenciais que podem ser resolvidas separadamente, e ν como o parâmetro que relaciona ambas equações.

A solução para a expressão

$$\frac{d}{dx}X(x,\nu)=-\frac{1}{\nu} \tag{7}$$

pode ser obtida através de outra separação de variáveis, de modo que se obtenha

$$\frac{dX(x,\nu)}{X(x,\nu)}=-\frac{1}{\nu}dx \tag{8}$$

Consequentemente, pode-se integrar ambos os lados da Eq.(8) Assim

$$\int\frac{dX(x,\nu)}{X(x,\nu)}=\int\frac{-1}{\nu}dx \tag{9}$$

A solução destas integrais é dada como

$$ln(X(x,\nu)) = -\frac{x}{v} + C \tag{10}$$

onde é conveniente utilizar uma exponencial para simplificação, conforme

$$e^{ln(X(x,\nu))} = e^{-x/v}e^{C} \tag{11}$$

Dessa forma, pode-se concluir o decaimento exponencial na variável espacial, dado por

$$X(x,\nu) = Ce^{-x/v} \tag{12}$$

fazendo com que a solução possa ser escrita como

$$\Psi(x,\mu) = \Phi(\nu,\mu)e^{-x/v} \tag{13}$$