



UNIVERSIDADE
ESTADUAL DE LONDRINA

TIAGO DE SOUZA SOTANA

**PARALELIZAÇÃO DE CÓDIGOS COMPUTACIONAIS PARA SOLUÇÃO
DA EQUAÇÃO DE TRANSPORTE SEM SIMETRIA AZIMUTAL**

Londrina - Pr
2013

TIAGO DE SOUZA SOTANA

**PARALELIZAÇÃO DE CÓDIGOS COMPUTACIONAIS PARA
SOLUÇÃO DA EQUAÇÃO DE TRANSPORTE SEM SIMETRIA
AZIMUTAL**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Londrina como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Jacques Duílio Brancher.

Londrina
2013

TIAGO DE SOUZA SOTANA

**PARALELIZACAO DE CÓDIGOS COMPUTACIONAIS PARA SOLUÇÃO DA
EQUACÃO DE TRANSPORTE SEM SIMETRIA AZIMUTAL**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Londrina como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Jacques Duílio Brancher.

BANCA EXAMINADORA

Prof. Dr. Jacques Duílio Brancher Universidade
Estadual de Londrina - UEL

Prof. Dr. Alan Salvany Felinto Universidade
Estadual de Londrina - UEL

Prof. Dr. Paulo Laerte Natti Universidade
Estadual de Londrina - UEL

Prof. Dr. Sylvio Barbon Junior Universidade
Estadual de Londrina - UEL

Londrina, 30 de Janeiro de 2013.

*Aos meus pais, Antonio e Suely,
a minha namorada Marília,
e a todos que me apoiaram
até a conclusão deste trabalho.*

Agradecimentos

Agradeço primeiramente a Deus, por ter me dado o dom maior, a vida.

Aos meus pais, Antonio e Suely, e à minha namorada, Marília, por todo apoio e incentivo durante toda minha caminhada.

Ao meu orientador, Jacques, pela amizade, companheirismo, confiança e por todas as broncas e incentivos, pois tenho certeza que foram muito úteis para que eu chegasse até aqui.

Aos amigos Marc Antonio, Ricardo Inácio, Rafael Herrera, Márcio Santos, Estevan Costa e Fabio Matsunaga pelo apoio, compreensão, conversas e risadas que dividimos durante esses últimos anos.

SOTANA, Tiago de Souza. **Paralelização de Códigos Computacionais para Solução da Equação de Transporte sem Simetria Azimutal**. 2013. 70 f. Dissertação (Mestrado) - Universidade Estadual de Londrina, Londrina. 2013.

Resumo

O método LTS_N vem sendo extensamente utilizado para a solução da equação de transporte de partículas. Inúmeros trabalhos foram desenvolvidos, tendo sempre como foco a obtenção de resultados numéricos para a equação. Dentre os principais, pode-se citar: o dependente do tempo e em 2 e 3 dimensões. Neste trabalho foi feita a paralelização do método LTS_N para a solução da equação sem simetria azimutal. Os algoritmos que foram paralelizados foram propostos por Brancher[1] e Segatto[2]. A principal motivação para o desenvolvimento deste trabalho foi buscar melhorias na performance de ambos, identificando pontos fortes, fracos e também melhorias nos resultados já existentes na literatura. Para este fim, utilizou-se o compilador Fortran 90, e também a biblioteca de álgebra linear LAPACK e o MPI, que foi utilizado na paralelização propriamente dita. Ambos os algoritmos implementaram o mesmo método porém com características diferentes. Brancher utilizou uma técnica de inversão recursiva de matrizes, enquanto Segatto utilizou a técnica da diagonalização para inverter a mesma matriz com Dummy Node. Como principais resultados, são apresentadas as implementações em paralelo de ambos os algoritmos, e também resultados numéricos mais precisos para os graus de anisotropia $L = 8$, $L = 82$ e $L = 299$ e inéditos para $L = 1999$, que também representam um avanço significativo no processo de solução de transporte.

Palavras-Chave: Computação paralela. Equação de Transporte. Interface de Passagem de Mensagem. Boltzmann.

SOTANA, Tiago de Souza. **Parallelization of Computer Codes for the Solution of Transport Equation Without Azimuthal Symmetry**. 2013. 70 p. Dissertation (Master's degree) - State University of Londrina, Londrina. 2013.

Abstract

The LTS_N method has been extensively used to solving transport equation of particles. Countless studies have been developed, always focused on obtaining numerical results for the equation. Among the principal works, can be cited: time dependent and 2 and 3 dimensions. This work was done the parallelization of the LTS_N method for the solution of the equation without azimuthal symmetry. The algorithms that were parallelized were proposed by Brancher[1] and Segatto[2]. The main motivation for this work was the development in order to seek improvements in both performance and identify strengths, weaknesses and also improvements in results in the existing international literature. For this purpose, we used the compiler Fortran 90, and also a library of linear algebra LAPACK and MPI, which was used in parallelization itself. Both algorithms implemented the same method but with different characteristics. Brancher used a recursive matrix inversion technique while Segatto uses the diagonalization technique to invert the matrix and also uses Dummy Node. The main results are presented in parallel implementation of both algorithms and also more accurate numerical results for the degree of anisotropy $L = 8$, $L = 82$ and $L = 299$, and unpublished for $L=1999$, which also represent a significant advance in the process of transport solution.

Keyword: Parallel Computing. Transport Equation. Message Passing Interface. Boltzmann.

Lista de Figuras

- 4.1 Fluxograma do algoritmo sequencial para cálculo da equação de transporte. . p. 39
- 4.2 Fluxograma do algoritmo paralelo para cálculo da equação de transporte. . . p. 40
- 5.1 Gráfico Processos x Tempo para Anisotropia 82 e técnica DNI na Máquina 1. p. 42
- 5.2 Gráfico Processos x Tempo para Anisotropia 299 e técnica DNI na Máquina 1. p. 42
- 5.3 Gráfico Processos x Tempo para Anisotropia 82 e técnica de Inversão Recursiva na Máquina 1. p. 43
- 5.4 Gráfico Processos x Tempo para Anisotropia 299 e técnica de Inversão Recursiva na Máquina 1. p. 43
- 5.5 Gráfico Processos x Tempo para Anisotropia 82 e técnica DNI na Máquina 2. p. 44
- 5.6 Gráfico Processos x Tempo para Anisotropia 299 e técnica DNI na Máquina 2. p. 45
- 5.7 Gráfico Processos x Tempo para Anisotropia 299 e técnica DNI na Máquina 3. p. 46
- 5.8 Gráfico Processos x Tempo para Anisotropia 82 e técnica de Inversão Recursiva na Máquina 3. p. 46
- 5.9 Gráfico Processos x Tempo para Anisotropia 299 e técnica de Inversão Recursiva na Máquina 3. p. 47
- 5.10 Gráfico Processos x Tempo para Anisotropia 1999. p. 49

Lista de Tabelas

1	Problemas Executados	p. 41
2	Máquinas utilizadas para testes	p. 41
3	Aumento de Velocidade Ideal e Obtido na Máquina 1	p. 44
4	Aumento de Velocidade Ideal e Obtido na Máquina 2	p. 45
5	Aumento de Velocidade Ideal e Obtido na Máquina 3	p. 47
6	Resultados do Problema para Anisotropia 1999, $\omega = 0.9$, $\mu_0 = 1$, $x_0 = 64$ e $N = 2000$	p. 48
7	Aumento de Velocidade Ideal e Obtido para o Problema de Anisotropia 1999	p. 49
8	Menores Valores para Obter um Resultado Satisfatório	p. 50
9	Valores dos Parâmetros dos Problemas Executados	p. 56
10	Resultado parcial, $M = 0$, para o problema com Anisotropia 8	p. 57
11	Resultado parcial, $M = 1$, para o problema com Anisotropia 8	p. 58
12	Resultado parcial, $M = 2$, para o problema com Anisotropia 8	p. 59
13	Resultado parcial, $M = 3$, para o problema com Anisotropia 8	p. 60
14	Resultado parcial, $M = 4$, para o problema com Anisotropia 8	p. 61
15	Resultado parcial, $M = 5$, para o problema com Anisotropia 8	p. 62
16	Resultado parcial, $M = 6$, para o problema com Anisotropia 8	p. 63
17	Resultado parcial, $M = 7$, para o problema com Anisotropia 8	p. 64
18	Resultado parcial, $M = 8$, para o problema com Anisotropia 8	p. 65
19	Resultado para o problema com anisotropia 82, $\omega = 0.9$, $\mu_0 = 0.5$, $x_0 = 1$ e $N = 1000$	p. 66
20	Resultado para o problema com anisotropia 82, $\omega = 0.9$, $\mu_0 = 1$, $x_0 = 1$ e $N = 1000$	p. 67

21	Resultado para o problema com anisotropia 299, $\omega = 0.9$, $\mu_0 = 1$, $x_0 = 64$ e $N = 1000$	p.68
----	--	------

Lista de Siglas e Abreviaturas

BLAS *Basic Linear Algebra Subprograms*

EISPACK *Eigensystem Package*

LAPACK *Linear Algebra Package*

LINPACK *Linear Equation Software Package*

LTS_N *Laplace Transform S_N*

MPI *Message Passing Interface*

MIMD *Multiple Instruction/Multiple Data*

MISD *Multiple Instruction/Single Data*

NUMA *Non-Uniform Memory Access*

SMP *Symetric MutiProcessors*

S_N *Sistema de Ordenadas Discretas*

SISD *Single Instruction/Single Data*

SIMD *Single Instruction/Multiple Data*

SVD *Singular Value Decomposition*

UMA *Uniform Memory Access*

Lista de Símbolos

1. Caracteres Arábicos

A:	Matriz numérica de ordem N .
$a_{i,j}$:	Elementos da matriz A .
$a_{i,j}^m$:	Elementos da matriz A^m .
$\text{Adj}(S_k)$:	Matriz adjunta da matriz S_k .
$\mathbf{B}(x)$:	Matriz transformada inversa de $(s\mathbf{I} - \mathbf{A})^{-1}$.
$\mathbf{B}^m(x)$:	Matriz transformada inversa de $(s\mathbf{I} - \mathbf{A}^m)^{-1}$.
$\mathbf{C}^m(x)$:	Vetor convolução da matriz $B^m(x)$ com o vetor $Q^m(x)$.
D:	Matriz diagonal dos autovalores de \mathbf{A} .
$\det(\mathbf{S}_k)$:	Determinante da matriz \mathbf{S}_k .
d_i :	Elementos da matriz \mathbf{D} .
f :	Intensidade de radiação incidente em $x = 0$.
$f(\mu)$:	Intensidade de radiação incidente na fronteira $x = 0$.
g :	Intensidade de radiação incidente em $x = x_0$.
$g(\mu)$:	Intensidade de radiação incidente na fronteira $x = x_0$.
$\mathbf{H}(x)$:	Vetor convolução da matriz $\mathbf{B}(x)$ com o vetor fonte.
I:	Matriz identidade.
\mathcal{L} :	Operador da transformada de Laplace.
\mathcal{L}^{-1} :	Operador da transformada inversa Laplace.

N :	Ordem da quadratura de Gauss.
$P_l(\mu)$:	Polinômios de Legendre.
$P_l^m(\mu)$:	Funções associados de Legendre.
$p(\cos \theta)$:	Função de fase.
$\mathbf{Q}(x)$:	Vetor fonte.
$\mathbf{Q}(x, \mu)$:	Termo fonte na posição x e direção μ .
$\overline{\mathbf{Q}}(s)$:	Transformada de Laplace do vetor $\mathbf{Q}(x)$.
$\mathbf{Q}^m(x)$:	Termo fonte na posição x e na direção μ_m .
$\overline{\mathbf{Q}}^m(s)$:	Termo fonte transformado na direção μ_m .
s :	Parâmetro complexo proveniente da transformada de Laplace.
$(s\mathbf{I} - \mathbf{A})$:	Matriz LTS_N .
\mathbf{S}_k :	Sequência de matrizes.
\mathbf{T} :	Matriz triangular superior.
\mathbf{U} :	Matriz unitária.
\mathbf{U}^T :	Matriz transposta da matriz unitária.
\mathbf{V} :	Vetor coluna.
\mathbf{X} :	Matriz dos autovetores de \mathbf{A} .
x :	Espessura ótica.
x_0 :	Espessura da placa.
w_k :	Pesos da quadratura de Gauss-Legendre.

2. Caracteres Gregos

β_l :	Coefficientes da expansão em Polinômios de Legendre.
-------------	--

β_l^m :	Coeficientes da expansão em Polinômios de Legendre da função de fase.
γ_{M+N} :	Direções da nova quadratura.
μ :	Variável angular.
μ_k :	Raízes do polinômio de Legendre.
ω :	Termo de espalhamento simples do meio (albedo).
ω_n :	Termo de espalhamento simples do meio (albedo) na direção μ .
$\Psi(x)$:	Vetor intensidade de radiação.
$\Psi_1(x)$:	Intensidade de radiação das direções positivas de μ .
$\Psi_2(x)$:	Intensidade de radiação das direções negativas de μ .
$\Psi^m(x)$:	Vetor intensidade de radiação.
$\bar{\Psi}(s)$:	Intensidade de radiação transformada.
$\bar{\Psi}^m(s)$:	Intensidade de radiação transformada na direção μ_m .
$\psi(x, \mu)$:	Intensidade de radiação na direção μ e na posição x .
$\psi(x, \mu, \varphi)$:	Intensidade de radiação.
$\psi^i(x, \mu)$:	Componentes de Fourier.
θ :	Ângulo polar.
φ :	Ângulo azimutal.
φ_0 :	Ângulo de incidência de radiação solar.
ζ_M :	Direções associadas a peso nulo.

Sumário

1	Introdução	p. 16
2	Formulação do Problema	p. 19
2.1	Equação de Boltzmann	p. 19
2.2	Equação Transporte com Simetria Azimutal	p. 20
2.3	Equação Transporte sem Simetria Azimutal	p. 23
2.4	Método LTS_N	p. 26
2.5	Métodos de Inversão da Matriz LTS_N	p. 27
2.5.1	Inversão Recursiva usando Decomposição de Schur	p. 27
2.5.2	Método da Diagonalização	p. 30
2.6	Técnica de Interpolação Angular	p. 31
2.6.1	DNI	p. 31
3	Ferramentas	p. 33
3.1	Histórico do Paralelismo	p. 33
3.2	Hardware Paralelo	p. 33
3.2.1	Classificação de Flynn	p. 34
3.2.2	Classificação quanto ao Modelo de Comunicação	p. 34
3.3	Message Passing Interface	p. 35
3.4	LAPACK	p. 36
4	Detalhamento da Implementação	p. 37
4.1	Algoritmo Sequencial	p. 38

4.2	Algoritmo Paralelo	p. 38
5	Resultados e Discussão	p. 41
5.1	Desempenho dos Algoritmos em Paralelo	p. 41
5.2	Anisotropia 1999	p. 46
5.3	Cálculo dos Erros dos Problemas	p. 49
6	Conclusões	p. 51
6.1	Trabalhos Futuros	p. 51
	Referências	p. 52
	Apêndices	p. 55
	Apêndice A – Resultados Obtidos em Comparação com a Literatura	p. 56

1 Introdução

A equação de Boltzmann descreve a distribuição de partículas, como neutrons, elétrons, íons, moléculas e ondas eletromagnéticas, que fluem em um meio material, levando em conta seu movimento e suas interações com o meio [3] [4]. Assim, pode ser usada para descrever os mais diferentes fenômenos físicos como difusão da luz através de atmosferas planetárias, distribuição de neutrons no núcleo de um reator, fluxo de tráfego em uma auto-estrada, entre outras.

O método LTS_N consiste em resolver a aproximação S_N da equação de transporte, aplicando a transformada de Laplace no sistema de equações ordinárias, S_N , obtendo um sistema de N equações e N incógnitas. Esse sistema é resolvido de forma analítica para a intensidade de radiação que é encontrada invertendo-se a matriz LTS_N e calculando a transformada inversa de Laplace.

O primeiro trabalho sobre o método é de Vilhena e Barichello [5], que o desenvolveram para solucionar problemas em uma dimensão [6]. Essa formulação foi estendida para duas [7] e três dimensões [8]. Cabe ressaltar que a solução LTS_N aproxima a solução exata de Case [9], quando N tende ao infinito [10].

A formulação desenvolvida era válida apenas para N até 8. Do ponto de vista computacional era impraticável a utilização do método para valores maiores que 8, então Brancher et al. [11] propuseram um método recursivo para inverter a matriz LTS_N e obtiveram resultados satisfatórios para $N=100$. Como a maioria dos casos estudados possuíam autovalores reais, Simch et al. [12] apresentaram uma generalização do método para solucionar problemas de transporte com autovalores complexos.

Para aumentar a abrangência do método, Marona [13] modificou o método para resolução de problemas isotrópicos quando o parâmetro albedo é unitário e Rost [14] definiu uma solução analítica usando a decomposição espectral para transformar a matriz LTS_N em uma matriz diagonal.

Observa-se que o método LTS_N já foi aplicado a uma vasta quantidade de

problemas de transporte de partículas, portanto a matemática e a física envolvendo o tema já foram extensamente estudadas. O grande problema associado a ele é o alto custo computacional, quando implementado, o tempo para encontrar-se a solução aumenta exponencialmente. Na prática, isto significa que para valores de N grandes, acima de 500, o tempo de processamento torna-se proibitivo.

Uma das soluções para a melhoria de desempenho dos algoritmos, é a paralelização do mesmo, em que partes pré-definidas do código são executados em diversos processadores ao mesmo tempo, assim é obtido um aumento no desempenho. Para que um algoritmo seja paralelizável, é necessário que ele possa ser decomposto em subproblemas que possam ser executados satisfatoriamente ao mesmo tempo. Depois de decomposto, o algoritmo já pode ser paralelizado, bastando apenas escolher o modelo de paralelismo a ser adotado.

Um dos modelos conhecidos é a troca de mensagens. Este método é largamente utilizado em sistema de memória distribuída, em que cada processador endereça sua própria memória e a troca de dados entre os processos é feita por mensagens. Um padrão de biblioteca que usa a troca de mensagens é o MPI (*Message Passing Interface*), que define instruções para criar, gerenciar e realizar a comunicação entre os processos.

O padrão MPI é o mais utilizado para programação paralela em sistemas de memória distribuída [15]. As duas maiores vantagens na utilização desse padrão são a portabilidade e a funcionalidade. A portabilidade se refere ao fato de que o MPI pode ser executado em diferentes plataformas, que suportem o padrão, sem que seja necessário realizar alterações no código. A funcionalidade vem da existência de mais de 300 rotinas definidas para atender todas as necessidades dos usuários [16].

Considerando-se o acima exposto, o objetivo desse trabalho é melhorar o desempenho de duas técnicas aplicadas ao algoritmo LTS_N para N grandes (maiores que 500) por meio da paralelização. Dessa forma pode-se aumentar o valor de N , conseguir um resultado mais próximo da solução exata de [9] e ainda diminuir o tempo de processamento.

Para alcançar esse objetivo, o trabalho foi organizado da seguinte maneira: o capítulo 2 apresenta a formulação do problema. Descreve-se a equação de Boltzmann, o problema com e sem simetria azimutal, o método LTS_N e também as abordagens sobre os métodos de inversão e de interpolação. O capítulo 3 apresenta o motivo da paralelização, as características do paralelismo, do MPI e da LAPACK. No capítulo 4 é abordado o detalhamento da implementação, analisando a estrutura do algoritmo e mostrando a diferença entre o algoritmo sequencial e o paralelo. O capítulo 5 mostra os resultados comparando os obtidos

com os da literatura, seguido pelos resultados obtidos ao executar os problemas com os dois algoritmos e também uma análise do desempenho obtido. Também são apresentados resultados para um novo grupo de problemas, ainda não existente na literatura, com anisotropia de grau 1999. Para finalizar, o capítulo 6 contém a conclusão do trabalho e as sugestões para trabalhos futuros.

2 Formulação do Problema

Como já foi definido por vários autores na literatura, dentre eles [21], [14] e [13], foram apresentadas as equações de transporte com e sem simetria e a aplicação do método LTS_N . Também foram apresentados dois métodos de inversão da matriz LTS_N , a diagonalização e o método recursivo baseado na decomposição de Schur. A inversão recursiva foi mostrada por Segatto et al. [22] e Marona [13]. Além disso, foi mostrado como calcular a intensidade de radiação em direções diferentes das N calculadas utilizando a técnica de Dummy Node [14].

2.1 Equação de Boltzmann

A equação de transporte de Boltzmann, desenvolvida por Ludwig Boltzmann, descreve o comportamento estatístico de um fluido fora do equilíbrio termodinâmico [23] [24]. Ela pode ser utilizada para determinar como as quantidades físicas variam, como calor e momento, quando um fluido está em transporte. Outras propriedades características dos fluidos são viscosidade, condutividade térmica e elétrica. A equação de Boltzmann pode ser escrita, de forma geral, como

$$\frac{df}{dt} = \left(\frac{df}{dt}\right)_{\text{força}} + \left(\frac{df}{dt}\right)_{\text{difusão}} + \left(\frac{df}{dt}\right)_{\text{colisão}}, \quad (2.1)$$

onde o termo força corresponde à força exercida nas partículas por uma influência externa, o termo difusão representa a difusão das partículas e o termo colisão representa as forças agindo entre as partículas em colisões.

Modelando os termos de força e difusão, reescreve-se (2.1) como:

$$\frac{df}{dt} + \frac{\mathbf{p}}{m} \cdot \nabla f + \mathbf{F} \cdot \frac{df}{d\mathbf{p}} = \left(\frac{df}{dt}\right)_{\text{colisão}}, \quad (2.2)$$

enquanto o termo colisão tem a forma

$$\left(\frac{df}{dt}\right)_{\text{colisão}} = \int \int g I(g, \Omega) [f(\mathbf{p}'_A, t) f(\mathbf{p}'_B, t) - f(\mathbf{p}_A, t) f(\mathbf{p}_B, t)] d\Omega d^3 \mathbf{p}_A, \quad (2.3)$$

onde \mathbf{p}_A e \mathbf{p}_B são os momentos de duas partículas antes da colisão e \mathbf{p}'_A e \mathbf{p}'_B são os momentos depois da colisão, satisfazendo a conservação do momento, ou seja,

$$g = |\mathbf{p}_B - \mathbf{p}_A| = |\mathbf{p}'_B - \mathbf{p}'_A|$$

sendo g a magnitude do momento relativo e $I(g, \Omega)$ a seção transversal diferencial da colisão.

2.2 Equação Transporte com Simetria Azimutal

A partir da equação principal, definida em (2.1), é possível descrever a equação de transporte linear bidimensional, monoenergética, estacionária e com simetria azimutal, como:

$$\mu \frac{d}{dx} \psi(x, \mu) + \psi(x, \mu) = \frac{\omega}{2} \sum_{l=0}^L \beta_l P_l(\mu) \int_{-1}^1 P_l(\mu') \psi(x, \mu') d\mu' + Q(x, \mu) \quad (2.4)$$

com as condições de contorno

$$\psi(0, \mu) = f(\mu), \quad \text{se } \mu > 0$$

e

$$\psi(x_0, \mu) = g(\mu), \quad \text{se } \mu < 0,$$

onde:

- x varia do começo ao fim da placa, ou seja, $x \in [0, x_0]$;
- $\mu = \cos \theta$, sendo que θ é o ângulo polar e varia entre $[-1, 1]$;
- $\omega \in [0, 1]$ é o termo de espalhamento do meio, conhecido também como albedo;
- $Q(x, \mu)$ é uma fonte externa;
- $\psi(x, \mu)$ é a intensidade de radiação em x na direção μ ;
- $f(\mu)$ e $g(\mu)$ são intensidades de radiação conhecidas nas fronteiras do domínio, nas direções positiva e negativa.

Para obter a aproximação S_N , primeiro é feita a aproximação do seu termo integral pela quadratura de Gauss-Legendre, conforme descrito na equação (2.5)

$$\int_{-1}^1 P_l(\mu')\psi(x, \mu')d\mu' \simeq \sum_{k=1}^N P_l(\mu_k)\psi_k(x)w_k \quad , \quad (2.5)$$

onde μ_k e w_k são raízes e pesos do polinômio de Legendre de grau N e $\psi_k(x) = \psi(x, \mu_k)$. Na sequência, aplica-se o método da colocação sobre a equação (2.4) considerando como função teste a distribuição delta de Dirac, usando como pontos de colocação as N raízes do polinômio de Legendre $P_N(\mu)$. Desta forma é obtida a seguinte equação

$$\frac{d}{dx}\psi_n(x) + \frac{\psi_n(x)}{\mu_n} = \frac{\omega}{2\mu_n} \sum_{l=0}^L \beta_l P_l(\mu_n) \cdot \sum_{k=1}^N P_l(\mu_k)\psi_k(x)w_k + \frac{Q_n(x)}{\mu_n} \quad (2.6)$$

para $n = 1, 2, \dots, N$ com as condições de contorno

$$\psi_n(0) = f_n, \quad \text{se } \mu_n > 0$$

$$\psi_n(x_0) = g_n, \quad \text{se } \mu_n < 0$$

onde $Q_n(x) = Q(x, \mu_n)$. Conforme é usual na literatura, as direções discretas são ordenadas de forma decrescente

$$\mu_N < \dots < \mu_{\frac{N}{2}+1} < 0 < \mu_{\frac{N}{2}} < \dots < \mu_1 \quad .$$

Reescrevendo as equações S_N descritas pela equação (2.6) na forma matricial

$$\frac{d}{dx}\Psi(x) - \mathbf{A}\Psi(x) = \mathbf{Q}(x) \quad , \quad (2.7)$$

onde A representa uma matriz quadrada de ordem N cujos elementos são:

$$a_{i,j} = \begin{cases} -\frac{1}{\mu_i} + \frac{\omega w_j}{2\mu_i} \left[\sum_{l=0}^L \beta_l P_l(\mu_i) P_l(\mu_j) \right] & \text{se } i = j \\ \frac{\omega w_j}{2\mu_i} \left[\sum_{l=0}^L \beta_l P_l(\mu_i) P_l(\mu_j) \right] & \text{se } i \neq j \end{cases} \quad (2.8)$$

$Q(x)$ é um vetor fonte de ordem N definido por:

$$Q(x) = \left[\frac{Q_1(x)}{\mu_1}, \frac{Q_2(x)}{\mu_2}, \dots, \frac{Q_N(x)}{\mu_N} \right]^T \quad (2.9)$$

e o vetor intensidade de radiação é um vetor de ordem N definido por:

$$\Psi(x) = \begin{bmatrix} \Psi_1(x) \\ \Psi_2(x) \end{bmatrix} = \begin{bmatrix} \psi_1(x) \\ \vdots \\ \psi_{\frac{N}{2}}(x) \\ \psi_{\frac{N}{2}+1}(x) \\ \vdots \\ \psi_N(x) \end{bmatrix} . \quad (2.10)$$

Para facilitar a notação, na equação (2.10), a intensidade de radiação é dividida em dois sub-vetores de ordem $N/2$, onde o primeiro $\Psi_1(x)$ representa a intensidade de radiação das direções positivas e o segundo $\Psi_2(x)$ representa a intensidade das direções negativas de μ . Dessa forma, as condições de contorno podem ser escritas como:

$$\begin{cases} \Psi_1(0) = f \\ \Psi_2(x_0) = g \end{cases}$$

onde $f = [f_1, f_2, \dots, f_{N/2}]^T$ e $g = [g_{N/2+1}, g_{N/2+1}, \dots, g_N]^T$.

O método LTS_N consiste, primeiramente, em aplicar a transformada de Laplace sobre a variável espacial desta equação, criando um sistema linear de equações algébricas de N incógnitas e N equações, representadas por:

$$(s\mathbf{I} - \mathbf{A})\bar{\Psi}(s) = \Psi(0) + \bar{\mathbf{Q}}(s) , \quad (2.11)$$

onde \mathbf{I} é uma matriz identidade de ordem N, o parâmetro s é complexo e a barra representa a transformada de Laplace, isto é, $\bar{\Psi}(s) = \mathcal{L}[\Psi(x)]$ e $\bar{\mathbf{Q}}(s) = \mathcal{L}[\mathbf{Q}(x)]$. Resolvendo a intensidade de radiação transformada da equação (2.11) temos:

$$\bar{\Psi}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\Psi(0) + (s\mathbf{I} - \mathbf{A})^{-1}\bar{\mathbf{Q}}(s) . \quad (2.12)$$

Agora para encontrar a intensidade radiação basta aplicar a transformada inversa de Laplace na equação (2.12), isto é:

$$\Psi(x) = \mathbf{B}(x)\Psi(0) + \mathbf{H}(x) \quad (2.13)$$

onde

$$\mathbf{B}(x) = \mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{A})^{-1}]$$

e

$$\mathbf{H}(x) = \mathbf{B}(x) * \mathbf{Q}(x)$$

onde o símbolo $*$ indica convolução.

Observando a equação (2.13), nota-se que é conhecida apenas as $N/2$ primeiras componentes do vetor inicial $\Psi(0)$, isto é, $\Psi_1(0)$. Para encontrar $\Psi_2(0)$, reescreve-se a solução definida pela equação (2.13) como:

$$\begin{bmatrix} \Psi_1(x) \\ \Psi_2(x) \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{11}(x) & \mathbf{B}_{12}(x) \\ \mathbf{B}_{21}(x) & \mathbf{B}_{22}(x) \end{bmatrix} \begin{bmatrix} \Psi_1(0) \\ \Psi_2(0) \end{bmatrix} + \begin{bmatrix} \mathbf{H}_1(x) \\ \mathbf{H}_2(x) \end{bmatrix} . \quad (2.14)$$

Fazendo $x = x_0$ em (2.14) e usando as $N/2$ últimas equações deste sistema, obtém-se

$$\Psi_2(x_0) = \mathbf{B}_{21}(x_0)\Psi_1(0) + \mathbf{B}_{22}(x_0)\Psi_2(0) + \mathbf{H}_2(x_0) . \quad (2.15)$$

Como os vetores $\Psi_1(0)$, $\Psi_2(x_0)$ e $\mathbf{H}_2(x_0)$ são conhecidos, calcula-se $\Psi_2(0)$ da forma

$$\Psi_2(0) = \mathbf{B}_{22}^{-1}(x_0) \left[\Psi_2(x_0) - \mathbf{B}_{21}(x_0)\Psi_1(0) - \mathbf{H}_2(x_0) \right] , \quad (2.16)$$

logo a solução da equação (2.13) está completamente definida.

2.3 Equação Transporte sem Simetria Azimutal

A equação de transporte sem simetria azimutal é descrita como:

$$\mu \frac{d}{dx} \psi(x, \mu, \varphi) + \psi(x, \mu, \varphi) = \frac{\omega}{4\pi} \int_{-1}^1 \int_0^{2\pi} p(\cos \theta) \psi(x, \mu', \varphi') d\varphi' d\mu' + Q(x, \mu, \varphi) \quad (2.17)$$

com as seguintes condições de contorno

$$\psi(0, \mu, \varphi) = \pi \delta(\mu - \mu_0) \delta(\varphi - \varphi_0) \quad \text{se } \mu > 0$$

e

$$\psi(x_0, \mu, \varphi) = 0 \quad \text{se } \mu < 0 ,$$

onde:

- $x \in [0, x_0]$ é espessura ótica;
- $p(\cos \theta)$ representa a função de fase, e θ é o angulo entre as direções do fóton antes e depois da colisão com o alvo;
- $\omega \in [0, 1]$ é o termo de espalhamento simples do meio, também conhecido como coeficiente de albedo;
- $Q(x, \mu, \varphi)$ é o termo de fonte.

- $\psi(x, \mu, \varphi)$ é a intensidade de radiação na posição x e na direção definida por μ ;
- as direções $\varphi \in [0, 2\pi]$ e $\mu \in [-1, 1]$ são, respectivamente, o ângulo azimutal e o cosseno do ângulo polar;

Expandindo a função de fase em uma série de polinômios de Legendre, tem-se:

$$p(\cos \theta) = \sum_{l=0}^L \beta_l P_l(\cos \theta) .$$

Usando o Teorema de Adição para os polinômios de Legendre, obtém-se:

$$p(\cos \theta) = \sum_{m=0}^L (2 - \delta_{0,m}) \sum_{l=m}^L \beta_l^m P_l^m(\mu) P_l^m(\mu') \cos[m(\varphi - \varphi')] ,$$

onde os coeficientes β_l^m são dados por,

$$\beta_l^m = \beta_l \frac{(l-m)!}{(l+m)!} , \quad (2.18)$$

sendo $\beta_0 = 1$ e $|\beta_l| < 2l + 1$, $l = 1, 2, \dots, L$ e

$$P_l^m(\mu) = (1 - \mu^2)^{\frac{m}{2}} \frac{d^m}{d\mu^m} P_l(\mu)$$

são as funções associadas de Legendre.

Para resolver o problema (2.17) foi utilizado a solução proposta por ??, onde a intensidade de radiação pode ser escrita como:

$$\psi(x, \mu, \varphi) = \psi^*(x, \mu, \varphi) + \pi \delta(\mu - \mu_0) \delta(\varphi - \varphi_0) e^{-x/\mu} . \quad (2.19)$$

Substituindo a equação (2.19) em (2.17) tem-se

$$\mu \frac{d}{dx} \psi^*(x, \mu, \varphi) + \psi^*(x, \mu, \varphi) = \frac{\omega}{4\pi} \int_{-1}^1 \int_0^{2\pi} p(\cos \theta) \psi^*(x, \mu', \varphi') d\varphi' d\mu' + Q(x, \mu, \varphi) \quad (2.20)$$

$$\psi^*(0, \mu, \varphi) = 0 \quad \text{se } \mu > 0$$

e

$$\psi^*(x_0, \mu, \varphi) = 0 \quad \text{se } \mu < 0$$

onde

$$Q(x, \mu, \varphi) = \frac{\omega}{4} \sum_{m=0}^L (2 - \delta_{0,m}) \sum_{l=m}^L \beta_l^m P_l^m(\mu) P_l^m(\mu_0) \cos[m(\varphi - \varphi_0)] e^{-x/\mu_0} .$$

Expandindo a intensidade de radiação em uma série de Fourier, obtém-se:

$$Q^*(x, \mu, \varphi) = \psi_c^0(x, \mu) + \sum_{i=1}^{\infty} \left[\psi_c^i(x, \mu) \cos[i(\varphi - \varphi_0)] + \psi_s^i(x, \mu) \sin[i(\varphi - \varphi_0)] \right] .$$

Tem-se então duas situações:

i) Multiplicar as equações resultantes por $\sin[k(\varphi - \varphi_0)]$, com $k = 1, 2, \dots$ e integrar em $\varphi \in [0, 2\pi]$, resultando na equação

$$\mu \frac{d}{dx} \psi_s^k(x, \mu) + \psi_s^k(x, \mu) = \frac{\omega}{2} \sum_{l=k}^L \beta_l^k P_l^k(\mu) \int_{-1}^1 P_l^k(\mu') \psi_s^k(x, \mu') d\mu' \quad (2.21)$$

para $k = 1, \dots, L$, ou na equação

$$\mu \frac{d}{dx} \psi_s^k(x, \mu) + \psi_s^k(x, \mu) = 0 \quad (2.22)$$

para $k = L + 1, L + 2, \dots$, e com as condições de contorno

$$\psi_s^k(0, \mu) = \psi_s^k(x_0, -\mu) = 0 \quad \text{se } \mu > 0$$

Mediante a ausência de fonte e as condições de contorno serem nulas, pode-se afirmar que a solução para os problemas diferenciais acima, tem solução nula, $\psi_s^k(x, \mu) = 0$, para $k = 1, 2, \dots$

ii) Multiplicar as equações resultantes por $\cos[k(\varphi - \varphi_0)]$, com $k = 0, 1, 2, \dots$, e integrar em $\varphi \in [0, 2\pi]$, resultando na equação

$$\mu \frac{d}{dx} \psi_c^k(x, \mu) + \psi_c^k(x, \mu) = \frac{\omega}{2} \sum_{l=k}^L \beta_l^k P_l^k(\mu) \int_{-1}^1 P_l^k(\mu') \psi_c^k(x, \mu') d\mu' + Q^k(x, \mu) \quad (2.23)$$

para $k = 0, 1, 2, \dots, L$, onde

$$Q^k(x, \mu) = \frac{\omega}{2} \sum_{l=k}^L \beta_l^k P_l^k(\mu) P_l^k(\mu_0) e^{-x/\mu}$$

ou na equação

$$\mu \frac{d}{dx} \psi_c^k(x, \mu) + \psi_c^k(x, \mu) = 0 \quad (2.24)$$

para $k = L + 1, L + 2, \dots$, e com as condições de contorno

$$\psi_c^k(0, \mu) = \psi_c^k(x_0, -\mu) = 0 \quad \text{se } \mu > 0$$

Novamente, pode-se afirmar que $\psi_c^k(x, \mu) = 0$, para $k = L + 1, L + 2, \dots$, restando assim, resolver o problema diferencial dado pela equação (2.23). Portanto, a solução do problema sem simetria azimutal dado na equação (2.17) pode ser reescrita como:

$$\psi(x, \mu, \varphi) = \sum_{m=0}^L \psi_c^m(x, \mu) \cos[m(\varphi - \varphi_0)] + \pi \delta(\mu - \mu_0) \delta(\varphi - \varphi_0) e^{-x/\mu} \quad (2.25)$$

onde as funções $\psi_c^m(x, \mu)$ são soluções do problema diferencial (2.23), para $0 \leq m \leq L$.

2.4 Método LTS_N

Para resolver o problema sem simetria azimutal foi utilizado o método da diagonalização, portanto, resolver esse problema é o mesmo que resolver $L+1$ problemas com simetria azimutal. A aproximação S_N é feita com a aproximação do termo integral por meio da Quadratura Gaussiana [27] de ordem N e aplicando o método da colocação usando como função teste a distribuição delta de Dirac nas N direções discretas, resultando em

$$\mu_n \frac{d}{dx} \psi_n^m(x) + \psi_n^m(x) = \frac{\omega}{2} \sum_{l=m}^L \beta_l^m P_l^m(\mu_n) \left[\sum_{k=1}^N P_l^m(\mu_k) \psi_k^m(x) w_k + Q_n^m(x) \right] \quad (2.26)$$

com condições de contorno homogêneas. A intensidade de radiação e o termo de fonte podem ser representados, respectivamente, como $\Psi_n^m(x) = \Psi_c^m(x, \mu_n)$ e $Q_n^m(x) = Q_c^m(x, \mu_n)$.

Dividindo a equação (2.26) por μ_n , aplicando a transformada de Laplace na variável espacial e fazendo uso da equação (2.18) para exprimir β_l^m , obtemos para $n = 1, 2, \dots, L$ as equações

$$\begin{aligned} & \left(s + \frac{1}{\mu_n} - \frac{\omega}{2\mu_n} \sum_{l=m}^L \beta_l^m \frac{(l-m)!}{(l+m)!} P_l^m(\mu_n) P_l^m(\mu_n) w_n \right) \bar{\Psi}_n^m(s) - \\ & \frac{\omega}{2\mu_n} \sum_{l=m}^L \beta_l^m \frac{(l-m)!}{(l+m)!} P_l^m(\mu_k) w_k \bar{\Psi}_k^m(s) = \psi_n^m(0) + \frac{\bar{Q}_n^m(s)}{\mu_n} \end{aligned} \quad (2.27)$$

Logo, obtem-se $L+1$ sistemas de equações na forma matricial

$$(s\mathbf{I} - \mathbf{A}^m) \bar{\Psi}^m(s) = \Psi^m(0) + \bar{Q}^m(s) \quad (2.28)$$

com $0 \leq m \leq L$, onde cada elemento $a^m(i, j)$ da matriz A^m , é definido por:

$$a_{i,j}^m = \begin{cases} s + \frac{1}{\mu_i} - \frac{\omega}{2\mu_i} \left[\sum_{l=m}^L \beta_l \frac{(l-m)!}{(l+m)!} P_l^m(\mu_i) w_j P_l^m(\mu_j) \right] & \text{se } i = j \\ -\frac{\omega}{2\mu_i} \left[\sum_{l=m}^L \beta_l \frac{(l-m)!}{(l+m)!} P_l^m(\mu_i) w_j P_l^m(\mu_j) \right] & \text{se } i \neq j \end{cases} \quad (2.29)$$

e os vetores $\bar{\Psi}^m(s)$, $\Psi^m(0)$ e $\bar{Q}^m(s)$ são escritos como:

$$\bar{\Psi}^m(s) = \begin{bmatrix} \bar{\psi}_1^m(s) \\ \bar{\psi}_2^m(s) \\ \vdots \\ \bar{\psi}_N^m(s) \end{bmatrix}, \quad \Psi^m(0) = \begin{bmatrix} \psi_1^m(0) \\ \psi_2^m(0) \\ \vdots \\ \psi_N^m(0) \end{bmatrix}, \quad \bar{Q}^m(s) = \begin{bmatrix} \frac{\bar{Q}_1^m(s)}{\mu_1} \\ \frac{\bar{Q}_2^m(s)}{\mu_2} \\ \vdots \\ \frac{\bar{Q}_N^m(s)}{\mu_N} \end{bmatrix}. \quad (2.30)$$

Resolvendo a equação 2.28 é obtido para cada valor de m , os fluxos transformados:

$$\bar{\Psi}^m(s) = (s\mathbf{I} - \mathbf{A}^m)^{-1} [\Psi^m(0) + \bar{Q}^m(s)]. \quad (2.31)$$

Pode-se obter o vetor intensidade $\Psi^m(x)$ em função do vetor $\Psi^m(0)$:

$$\Psi^m(x) = \mathbf{B}^m(x) \Psi^m(0) + \mathbf{C}^m(x) \quad (2.32)$$

onde as funções $\mathbf{B}^m(x)$ e $\mathbf{C}^m(x)$ são definidas como:

$$\mathbf{B}^m(x) = \mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{A}^m)^{-1}], \quad \mathbf{C}^m(x) = \mathbf{B}^m(x) * \bar{Q}^m(x), \quad (2.33)$$

sabendo que o operador $*$ indica convolução.

2.5 Métodos de Inversão da Matriz LTS_N

Neste trabalho são discutidos dois métodos. O primeiro é a inversão recursiva associada a decomposição de Schur [11] e o segundo é o método da diagonalização.

2.5.1 Inversão Recursiva usando Decomposição de Schur

O teorema de Schur [29], garante que toda matriz quadrada pode ser decomposta na forma

$$A = UTU^T, \quad (2.34)$$

onde U é uma matriz unitária, isto é, $U^{-1} = U^T$ e T é uma matriz triangular superior. Assim, podemos simplificar a inversão da matriz cheia, $(sI - A)^{-1}$, para a inversão de uma matriz triangular da seguinte maneira:

$$(s\mathbf{I} - \mathbf{A})^{-1} = (s\mathbf{U}\mathbf{U}^T - \mathbf{U}\mathbf{T}\mathbf{U}^T)^{-1} = (\mathbf{U}(s\mathbf{I} - \mathbf{T})\mathbf{U}^T)^{-1} = \mathbf{U}(s\mathbf{I} - \mathbf{T})^{-1}\mathbf{U}^T . \quad (2.35)$$

Assim, como U é uma matriz constante, então

$$\mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{A})^{-1}] = \mathbf{U}\mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{T})^{-1}]\mathbf{U}^T . \quad (2.36)$$

Observa-se que:

- A matriz $s\mathbf{I} - \mathbf{T}$ tem a forma:

$$s\mathbf{I} - \mathbf{T} = \begin{bmatrix} s - t_{11} & -t_{12} & \dots & -t_{1N-1} & -t_{1N} \\ 0 & s - t_{22} & \ddots & -t_{2N-1} & -t_{2N} \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \dots \\ 0 & \dots & 0 & 0 & s - t_{NN} \end{bmatrix} .$$

- O determinante é dado por:

$$\det(s\mathbf{I} - \mathbf{T}) = \prod_{i=1}^N (s - t_{ii}) .$$

- A inversa da matriz bloco pode ser calculada como:

$$\begin{bmatrix} \mathbf{C} & \mathbf{D} \\ \mathbf{0} & \mathbf{E} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{C}^{-1} & -\mathbf{C}^{-1}\mathbf{D}\mathbf{E}^{-1} \\ \mathbf{0} & \mathbf{E}^{-1} \end{bmatrix} ,$$

onde $\mathbf{0}$ é uma matriz nula e \mathbf{C}^{-1} e \mathbf{E}^{-1} existem.

Para fazer a inversão da matriz $s\mathbf{I} - \mathbf{T}$ é construída uma sequência de matrizes S_k de ordem k definidas por:

$$\mathbf{S}_k = \begin{bmatrix} s - t_{11} & -t_{12} & \dots & -t_{1k-1} & -t_{1k} \\ 0 & s - t_{22} & \ddots & -t_{2k-1} & -t_{2k} \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & s - t_{kk} \end{bmatrix} , \quad k = 2, \dots, N . \quad (2.37)$$

Analisando a fórmula acima, pode-se escrever a matriz S_k em função de S_{k-1} como:

$$\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_{k-1} & -\mathbf{V} \\ \mathbf{0} & [s - t_k k] \end{bmatrix}, \quad (2.38)$$

onde $\mathbf{0}$ é um vetor linha nulo de ordem $k - 1$ e $\mathbf{V} = [t_{1k}, \dots, t_{k-1k}]^T$ é um vetor coluna de ordem $k - 1$. Quando $k = 2$, \mathbf{S}_1 é uma matriz unitária, logo sua inversa é fácil de calcular. Para $k = 3$ temos

$$\mathbf{S}_2^{-1} = \begin{bmatrix} \mathbf{S}_1^{-1} & \frac{\mathbf{S}_1^{-1}\mathbf{V}}{s-t_{22}} \\ \mathbf{0} & \frac{1}{s-t_{22}} \end{bmatrix}. \quad (2.39)$$

Assim como \mathbf{S}_k , escreve-se \mathbf{S}_k^{-1} em função de \mathbf{S}_{k-1}^{-1} , ou seja,

$$\mathbf{S}_k^{-1} = \begin{bmatrix} \mathbf{S}_{k-1}^{-1} & \frac{\mathbf{S}_{k-1}^{-1}\mathbf{V}}{s-t_{kk}} \\ \mathbf{0} & \frac{1}{s-t_{kk}} \end{bmatrix}, \quad k = 2, \dots, N. \quad (2.40)$$

Para calcular a transformada inversa de Laplace, utilizando o Teorema de Heaviside, precisa calcular sua adjunta

$$\text{Adj}(\mathbf{S}_k) = \mathbf{S}_k^{-1} \det(\mathbf{S}_k) = \begin{bmatrix} (s - t_{kk}) \text{Adj}(\mathbf{S}_{k-1}) & \text{Adj}(\mathbf{S}_{k-1}) \mathbf{V} \\ \mathbf{0} & \det(\mathbf{S}_{k-1}) \end{bmatrix}, \quad k = 2, 3, \dots, N \quad (2.41)$$

Finalmente, o Teorema de Heaviside que garante que se tem uma função do tipo

$$F(s) = \frac{P(s)}{Q(s)},$$

onde o grau de $Q(s)$ é maior que o grau de $P(s)$, então

$$\mathcal{L}^{-1} \left\{ \frac{P(s)}{Q(s)} \right\} = \sum_{k=1}^n \frac{P(\alpha_k)}{Q'(\alpha_k)} e^{\alpha_k x} \quad (2.42)$$

sendo α_k as raízes de $Q(s)$.

Assim, pode-se escrever a matriz $\mathbf{B}(x)$ como sendo:

$$\mathbf{B}(x) = \mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{A})^{-1}] = \mathbf{U} \mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{T})^{-1}] \mathbf{U}^T = \mathbf{U} \left[\sum_{i=1}^N \frac{\text{Adj}(\mathbf{S}_N)}{\frac{d}{ds} \det(\mathbf{S}_N)|_{s=t_{ii}}} e^{t_{ii}x} \right] \mathbf{U}^T \quad (2.43)$$

Assim a descrição do método de inversão é concluído, tendo sua solução definida na equação (2.43).

2.5.2 Método da Diagonalização

O método da diagonalização possui uma aproximação um pouco diferente da apresentada anteriormente. Neste caso, a matriz \mathbf{A} pode ser decomposta espectralmente como

$$\mathbf{A} = \mathbf{XDX}^{-1}$$

onde \mathbf{D} é uma matriz diagonal formada pelos autovalores de \mathbf{A}

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_3 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & d_N \end{bmatrix} \quad (2.44)$$

e \mathbf{X} é a matriz que tem como colunas os autovetores de \mathbf{A} . Logo,

$$\begin{aligned} \mathbf{B}(x) &= \mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{A})^{-1}] = \mathcal{L}^{-1}[(s\mathbf{X}\mathbf{X}^{-1} - \mathbf{XDX}^{-1})^{-1}] = \\ &= \mathcal{L}^{-1}[(\mathbf{X}(s\mathbf{I} - \mathbf{D})\mathbf{X}^{-1})^{-1}] = \mathcal{L}^{-1}[\mathbf{X}(s\mathbf{I} - \mathbf{D})^{-1}\mathbf{X}^{-1}] . \end{aligned} \quad (2.45)$$

Como \mathbf{X} é uma matriz constante, reescreve-se a equação acima na forma

$$\mathbf{B}(x) = \mathbf{X}\mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{D})^{-1}]\mathbf{X}^{-1} . \quad (2.46)$$

Escrevendo a matriz diagonal $s\mathbf{I} - \mathbf{D}$ na forma matricial

$$s\mathbf{I} - \mathbf{D} = \begin{bmatrix} s - d_1 & 0 & 0 & \dots & 0 \\ 0 & s - d_2 & 0 & \dots & 0 \\ 0 & 0 & s - d_3 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & s - d_N \end{bmatrix} ,$$

onde os elementos d_i são os autovalores de \mathbf{A} . Assim a inversa fica

$$(s\mathbf{I} - \mathbf{D})^{-1} = \begin{bmatrix} \frac{1}{s-d_1} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{s-d_2} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{s-d_3} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \frac{1}{s-d_N} \end{bmatrix} . \quad (2.47)$$

Aplicando a transformada inversa de Laplace na matriz acima, tem-se

$$\mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{D})^{-1}] = \begin{bmatrix} e^{d_1x} & 0 & 0 & \dots & 0 \\ 0 & e^{d_2x} & 0 & \dots & 0 \\ 0 & 0 & e^{d_3x} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & e^{d_Nx} \end{bmatrix} = e^{\mathbf{D}x} . \quad (2.48)$$

Logo, a matriz $\mathbf{B}(x)$ pode ser escrita da seguinte forma

$$\mathbf{B}(x) = \mathbf{X}e^{\mathbf{D}x}\mathbf{X}^{-1} . \quad (2.49)$$

2.6 Técnica de Interpolação Angular

Como já foi visto, a formulação para encontrar a intensidade de radiação nas direções μ_i , onde $i = 1, 2, \dots, N$, já foi obtido. Porém para calcular a intensidade de radiação em uma direção diferente das N já encontradas, pode ser utilizado a interpolação polinomial. Aqui é mostrada a técnica DNI que consiste na inclusão de nós fictícios associados a pesos nulos da quadratura.

2.6.1 DNI

A técnica DNI é a inclusão de nós fictícios na formulação do problema, considerando nulo os pesos da quadratura nesse ponto. Assim, determina-se a solução angular de partículas em M novas direções discretas onde está querendo calcular a intensidade de radiação, sendo $\zeta_1 > \zeta_2 > \dots > \zeta_{\frac{M}{2}} > 0 > \zeta_{\frac{M}{2}+1} > \dots > \zeta_M$ as direções cujos pesos associados são nulos. Desta forma são redefinidas as direções como:

$$\begin{aligned} \gamma_1 &= \zeta_1 \longrightarrow \omega_1 = 0 \\ &\vdots \\ \gamma_{\frac{M}{2}} &= \zeta_{\frac{M}{2}} \longrightarrow \omega_{\frac{M}{2}} = 0 \\ \gamma_{\frac{M}{2}+1} &= \mu_1 \longrightarrow \omega_{\frac{M}{2}+1} = w_1 \\ &\vdots \\ \gamma_{\frac{M}{2}+N} &= \mu_N \longrightarrow \omega_{\frac{M}{2}+N} = w_N \\ \gamma_{\frac{M}{2}+N+1} &= \zeta_{\frac{M}{2}+1} \longrightarrow \omega_{\frac{M}{2}+N+1} = 0 \end{aligned}$$

$$\vdots \quad \vdots$$

$$\gamma_{M+N} = \zeta_M \longrightarrow \omega_{M+N} = 0$$

desta forma as $\frac{N+M}{2}$ primeiras direções da nova quadratura são positivas e as $\frac{N+M}{2}$ restantes são negativas. Assim, o método LTS_N é aplicado sobre o novo sistema de $M + N$ equações, fazendo com que a matriz LTS_N também tenha ordem $M + N$.

3 Ferramentas

Neste capítulo é descrita a motivação para se paralelizar os códigos computacionais, suas características, o padrão de biblioteca MPI e a biblioteca de álgebra linear LAPACK.

3.1 Histórico do Paralelismo

O interesse em computação paralela data da década de 50, com avanços nos supercomputadores através dos anos 60 até 90. Esses computadores eram multiprocessadores com memória compartilhada trabalhando paralelamente nos dados. No meio da década de 80, um novo tipo de computador paralelo foi apresentado quando a *Caltech Concurrent Computation* apresentou um supercomputador para aplicações científicas com 64 processadores Intel 8086. Essa arquitetura recebe o nome de *massively parallel processors* ou MPP.

No fim dos anos 80, os *clusters* vieram para competir com os MPPs. Um *cluster* é um tipo de computador paralelo que contém vários computadores conectados entre si através de uma rede. Atualmente, a computação paralela é baseada em processadores multi-núcleos. A maioria dos computadores e notebooks pessoais são equipados com processadores *dual-core* e até mesmo *quad-core*. As fabricantes de processadores começaram a melhorar o desempenho dos *chips* com a adição de novos núcleos, pois são mais eficientes em economizar energia e dissipar calor do que simplesmente aumentar a frequência dos processadores.

3.2 Hardware Paralelo

A arquitetura de hardware utilizada por um computador paralelo influencia de forma direta no funcionamento do mesmo. Por isso são apresentados duas classificações bastante utilizadas na arquitetura de computadores: a classificação de Flynn e quanto ao modelo de comunicação.

3.2.1 Classificação de Flynn

Embora seja bastante antiga, a classificação de Flynn [31] é a mais aceita. O processo computacional é visto como uma relação entre o fluxo de instruções e o fluxo de dados. Foram propostas quatro classes de computadores: SISD, SIMD, MISD, MIMD.

- **SISD**: Fluxo de instrução e de dados únicos. Um processador executa um conjunto de instruções em um conjunto de dados. Não é um computador paralelo.
- **SIMD**: Fluxo de instruções único e vários fluxos de dados. Múltiplos processadores executando simultaneamente o mesmo conjunto de instruções em diversos conjuntos de dados.
- **MISD**: Vários fluxos de instruções e fluxo de dados único. Múltiplos processadores executando diferentes conjuntos de instruções em um único conjunto de dados.
- **MIMD**: Vários fluxos de instruções e de dados. Múltiplos processadores executando diferentes conjuntos de instruções em diferentes conjuntos de dados.

É importante lembrar que essa classificação desconsidera os avanços nas técnicas empregadas no paralelismo, levando em conta apenas as instruções de processador.

3.2.2 Classificação quanto ao Modelo de Comunicação

A comunicação entre os processos é um grande problema a ser resolvido para que haja eficiência no uso do paralelismo. De forma geral, quanto maior a distância física entre dois processadores, mais lenta será a comunicação entre elas, por outro lado, quanto menor a distância, mais cara é a solução.

Como não existe uma solução ideal é preciso escolher quais características beneficiar e arquitetar o hardware em função da sua aplicação. As duas propostas que mais se destacam no quesito desempenho e viabilidade são: memória compartilhada e troca de mensagens.

- **Memória Compartilhada**: Consiste em compartilhar o espaço de memória através de conexão por meio do barramento e primitivas do tipo *load/store*. Existem dois tipos de multiprocessadores que compartilham a memória, os UMA, também conhecidos como SMP, onde todos os processadores possuem o mesmo tempo de acesso a qualquer posição

da memória, e os NUMA, onde os processadores tem diferentes tempos de acesso, dependendo de qual processador solicitou o acesso e qual a posição a ser acessada.

- **Troca de Mensagens:** Consiste em definir espaços de memória privados e compartilhar dados entre estes por meio de uma rede e de primitivas do tipo *send/receive*. Neste tipo de sistema não há a necessidade de fazer alterações estruturais nos processadores e permite uma grande escala de paralelismo, com até centenas de processadores. Porém como a comunicação é feita por meio de rede, a velocidade fica limitada à conexão da rede, que pode ser bem inferior à velocidade do barramento de memória.

3.3 Message Passing Interface

MPI é um padrão de comunicação entre processos em programação paralela. Essa comunicação é utilizada em sistema com memória distribuída, pois é necessário passar informações importantes de um processo ao outro, devido a cada processador ter sua própria memória independente, o que não acontece em um sistema com memória compartilhada, pois todos os processos acessam os mesmos endereços de memória.

O MPI é extensamente utilizado em computação paralela, por ser apenas um padrão e não uma implementação, com isso, ele é totalmente portátil, pois funciona em várias plataformas diferentes. A maioria das implementações do MPI é um conjunto de subrotinas chamadas diretamente pelas linguagens C, C++ e Fortran. Essa interface proporciona uma topologia virtual, sincronização e comunicação entre um conjunto de processos, que foram mapeados como nós de um cluster, processadores de um computador multi-processador ou núcleos de um multi-núcleo.

As funções definidas no MPI incluem comunicação ponto-a-ponto, troca de mensagens entre pares de processos, combinação de resultados parciais, sincronização de nós, e informações sobre a rede, tais como: número de processos, identidade do processador onde o processo foi mapeado, processos vizinhos acessíveis na topologia, entre outros. As rotinas de comunicação ponto-a-ponto incluem as formas síncrona, assíncrona e com buffer.

Apesar do MPI possuir centenas de funções, muitos algoritmos paralelos podem ser escritos com apenas seis funções básicas. São elas: `MPI_INIT`, `MPI_FINALIZE`, `MPI_SEND`, `MPI_RECV`, `MPI_COMM_SIZE`, `MPI_COMM_RANK`.

- **MPI_INIT:** Inicializa o ambiente de execução do MPI;
- **MPI_FINALIZE:** Finaliza o ambiente de execução;

- **MPI_SEND**: Envia uma mensagem;
- **MPI_RECV**: Recebe uma mensagem;
- **MPI_COMM_SIZE**: Determina o tamanho do grupo que está associado com um comunicador;
- **MPI_COMM_RANK**: Determina o *rank* de cada processo num comunicador.

Essas operações são utilizadas de forma totalmente simplificada. No começo do algoritmo iniciamos o MPI, com o `MPI_INIT`. Depois configuramos o tamanho do grupo e o *rank* de cada processo, com o `MPI_COMM_SIZE` e `MPI_COMM_RANK`, respectivamente. Assim uma implementação está configurada para funcionar paralelamente em qualquer arquitetura. Quando necessário trocar informações entre os processos, utilizamos `MPI_SEND` para enviar e `MPI_RECV` para receber mensagens. Para finalizar a execução do algoritmo, utilizamos o `MPI_FINALIZE` para encerrar a paralelização.

3.4 LAPACK

LAPACK é uma biblioteca para o cálculo de álgebra linear. Foi escrita em Fortran 90 e possui rotinas para solução de sistemas de equações lineares, problemas de autovalores e problemas de SVD. Possui rotinas associadas a fatoração de matrizes, como LU, Cholesky, QR, Schur, entre outras.

O objetivo principal da LAPACK era fazer um uso das bibliotecas EISPACK e LINPACK e executar satisfatoriamente em processadores paralelos e em memória compartilhada. Nessas máquinas, a LINPACK e a EISPACK fazem o acesso à memória de forma ineficiente, mas a LAPACK corrige esse problema fazendo com que o algoritmo use operações de matrizes em blocos. Esses blocos são otimizados para cada arquitetura de computador, levando em conta a hierarquia de memória.

As rotinas foram escritas para que o máximo possível de computação seja feita por chamadas a BLAS, que é uma biblioteca para realizar operações básicas de álgebra linear [33][34][35]. Ela foi desenvolvida para explorar a BLAS nível 3, que é um conjunto de especificações para subprogramas em Fortran que faz vários tipos de multiplicação matricial e calcula a solução de sistemas triangulares.

4 Detalhamento da Implementação

Nesse capítulo são apresentados as estruturas dos algoritmos e a implementação paralela de ambos.

Os dois algoritmos implementam o método LTS_N , cada um usando uma técnica diferente. Brancher et al. [11] e Segatto et al. [22] utilizaram uma maneira recursiva de inverter a matriz LTS_N , utilizando a Decomposição de Schur, melhorando o desempenho do algoritmo. Segatto [2] utilizou o DNI, que consiste na inserção de nós fictícios na quadratura, podendo ser encontradas novas intensidades de radiação nas novas direções discretas. Mesmo com características diferentes, ambos utilizam a mesma estrutura de cálculo definida no método clássico. A solução do método segue alguns passos pré-definidos que são representados no algoritmo pelas rotinas.

- **Gera Matriz A:** Calcula todos os elementos a_{ij} da matriz A, que são definidos na equação (2.29). A matriz A é a matriz mais importante a ser calculada, pois é a partir dela que todas as outras matrizes e vetores serão calculados;
- **Calcula Autovalores e Autovetores:** Calcula os autovalores e autovetores que serão usados no cálculo da matriz B. O cálculo é feito utilizando uma rotina da LAPACK chamada DGEEV. Essa rotina recebe como parâmetros principais os seguintes elementos JOBVL, JOBVR, N, A, WR, WI, VL, VR. JOBVL e JOBVR que indicam se os autovetores da esquerda e/ou da direita serão calculados. N é o tamanho da matriz. A é a matriz onde serão calculados os autovalores e autovetores, WR e WI são respectivamente a parte real e parte imaginária dos autovalores e VL e VR recebem, respectivamente, os autovalores da esquerda e da direita;
- **Calcula Matriz B:** Calcula a transformada inversa de Laplace para encontrar a matriz $B(x)$, conforme apresentado na equação (2.33);
- **Calcula Vetor Fonte:** Calcula o vetor $Q(x)$ que será usado na convolução. Esse vetor pode ser calculado como $Q(x) = \mathcal{L}^{-1}[Q(s)]$ e o vetor $Q(s)$ foi definido na equação

(2.30);

- **Calcula Convolução:** Calcula a convolução entre a matriz B e o vetor Fonte. Tendo calculado a matriz $B(x)$ e o vetor $Q(x)$, resolve-se a convolução entre os dois para ser utilizado no cálculo do fluxo angular. A convolução é definida na equação (2.33);
- **Calcula Fluxo Angular:** Calcula o fluxo angular para o determinado valor de L . O fluxo angular é calculado como sendo a soma entre a matriz $B(x)$ e a convolução, como definido na equação (2.32).

4.1 Algoritmo Sequencial

O algoritmo sequencial segue a estrutura definida anteriormente, seguindo todos os passos. Como o problema sem simetria azimutal é calculado como sendo $L + 1$ problemas com simetria, pode-se dizer que o algoritmo segue essa estrutura de cálculo para cada um dos problemas existentes. O problema calcula o fluxo para $L = 0$, depois para $L = 1$, e assim sucessivamente até realizar o cálculo para $L = N$. Para cada um desses subproblemas, o código calcula como sendo um problema diferente, então para cada um deles, uma matriz A é gerada, uma matriz B é gerada e assim por diante. Para ilustrar essa execução, o fluxograma do algoritmo é mostrado na Figura 4.1, mostrando claramente as $L + 1$ repetições na execução.

4.2 Algoritmo Paralelo

Conforme foi mostrado no algoritmo sequencial, o paralelo também segue a mesma estrutura. A vantagem de se utilizar o paralelismo, nesse caso, vem do problema ser dividido em subproblemas independentes. Essa característica é baseada no paradigma de programação "*divide and conquer*" que significa dividir e conquistar. Esse paradigma consiste em dividir um problema complexo em subproblemas mais simples para serem resolvidos e depois as soluções de cada subproblema é combinada para obter a solução do problema original.

No caso do paralelismo, cada subproblema é calculado em um processo /processador separado, não precisando esperar um cálculo finalizar para começar outro. Com essa característica, a execução do algoritmo acaba sendo mais rápida, pois vários problemas são executados ao mesmo tempo e conforme a quantidade de processos sendo executados paralelamente o ganho de desempenho pode variar. A Figura 4.2 ilustra o fluxograma desse algoritmo paralelo, mostrando claramente como cada problema independente é executado em um processo diferente.

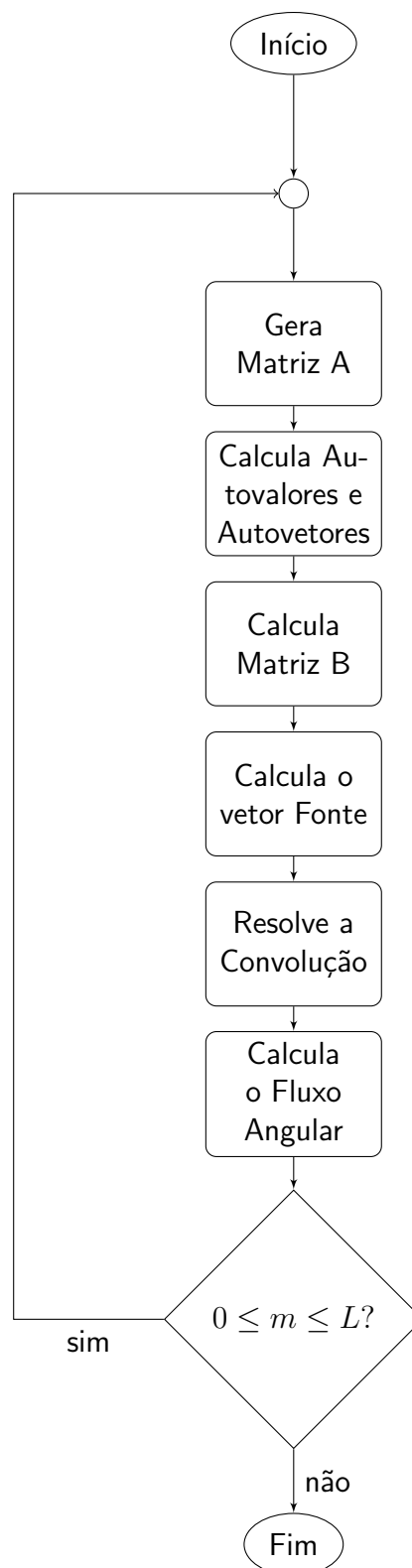


Figura 4.1: Fluxograma do algoritmo sequencial para cálculo da equação de transporte.

Como pode ser observado na Figura 4.2 cada processo executa um dos sub-problemas de forma independente, para então concluir o problema somando os resultados

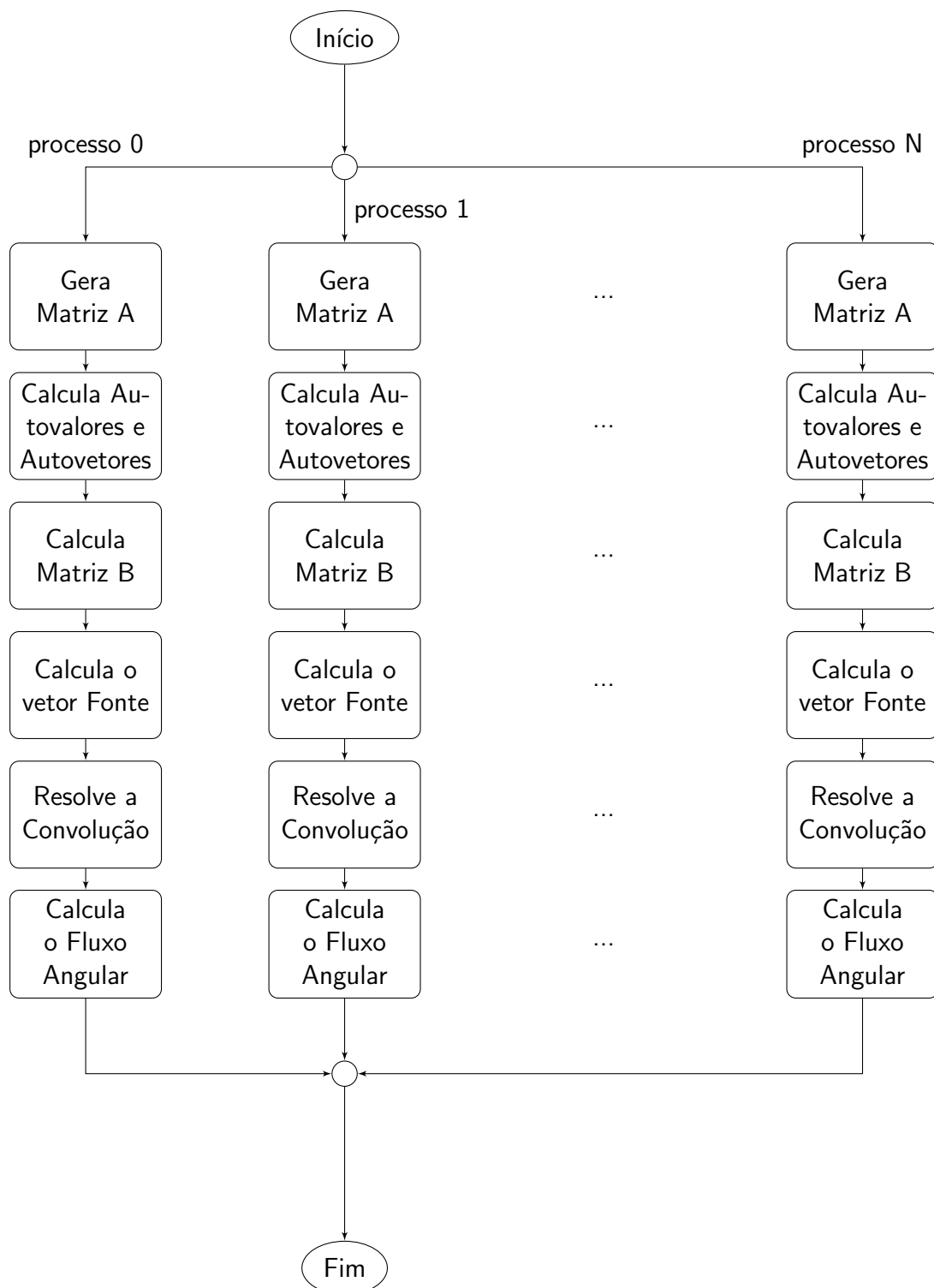


Figura 4.2: Fluxograma do algoritmo paralelo para cálculo da equação de transporte.

parciais obtidos por cada um dos processos.

5 Resultados e Discussão

5.1 Desempenho dos Algoritmos em Paralelo

Para avaliar o ganho de performance do paralelismo, cada algoritmo foi submetido a dois problemas conforme demonstrado pela Tabela 1, sendo executado em três máquinas de testes diferentes, conforme a Tabela 2.

Tabela 1: Problemas Executados

	ω	μ_0	x_0	L
Problema 1	0.9	0.5	1.0	82
Problema 2	0.9	1.0	64.0	299

Tabela 2: Máquinas utilizadas para testes

	Tipo	Processador	Memória
Máquina 1	Notebook	Intel Core i7 720QM @1.6GHz	16 GB
Máquina 2	Notebook	Intel Core i7 3612QM @2.1GHz	8 GB
Máquina 3	Servidor	2x AMD Opteron 6176 SE @2.3GHz	32 GB

Os problemas foram executados e obtiveram os resultados conforme pode ser observado nas Tabelas 19 e 21, no Apêndice A. Os resultados em ambas as técnicas são equivalentes, eles diferem apenas na precisão dos valores com o aumento ou diminuição do valor de N .

Os problemas foram executados, primeiramente, na Máquina 1, utilizando para a técnica DNI valor de $N = 2000$ e para a técnica de recursividade, $N = 200$. Essa diferença de grandeza entre as duas técnicas existe pelo fato de o algoritmo com recursividade ter sido implementado para processamento vetorial, assim instruções que nesta arquitetura são executadas em tempos muito baixos, no processador da máquina utilizada demandou uma grande quantidade de tempo, sendo inviável executar o mesmo com valores de N grandes. Nesse equipamento foram executados ambos os problemas primeiro sequencialmente para obter o tempo para comparar com as execuções paralelas, fazendo assim, a comparação dos resultados e análise do ganho de tempo.

Os problemas 1 e 2, utilizando o algoritmo com DNI, quando executados sequencialmente, levaram aproximadamente 5000 e 20000 segundos até concluir a execução. A diminuição nos tempos de execução conforme o aumento na quantidade de processos rodando simultaneamente pode ser observada pelas Figuras 5.1 e 5.2.

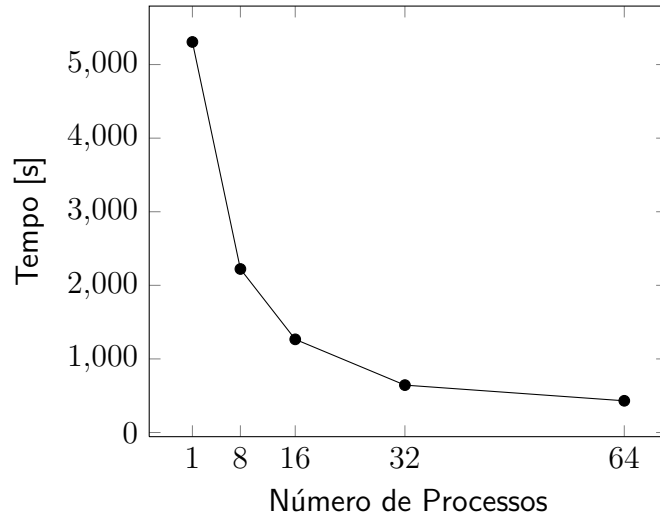


Figura 5.1: Gráfico Processos x Tempo para Anisotropia 82 e técnica DNI na Máquina 1.

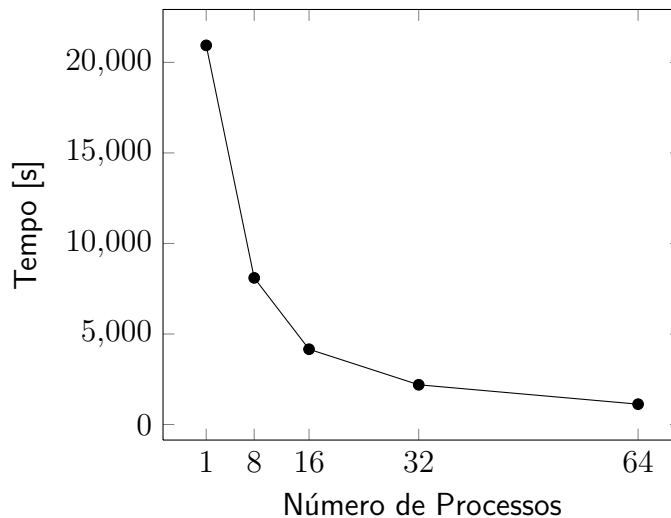


Figura 5.2: Gráfico Processos x Tempo para Anisotropia 299 e técnica DNI na Máquina 1.

Da mesma forma, os dois problemas utilizando o algoritmo recursivo, quando executados sequencialmente, levaram aproximadamente 400 e 1300 segundos. Nas execuções paralelas, pode-se observar a diminuição nos tempos conforme mostrado nas Figuras 5.3 e 5.4.

Conforme pode ser observado na Tabela 3, os ganhos obtidos diferem dos ganhos ideais. Isso se deve ao fato de que os algoritmos foram implementados e executados em um computador com memória compartilhada, e por isso, quando executado de forma sequencial, o barramento e a memória ficam disponíveis para apenas um processador, e conforme o

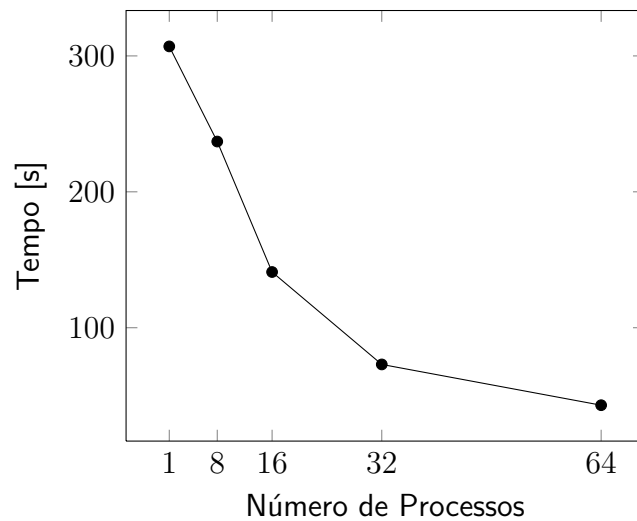


Figura 5.3: Gráfico Processos x Tempo para Anisotropia 82 e técnica de Inversão Recursiva na Máquina 1.

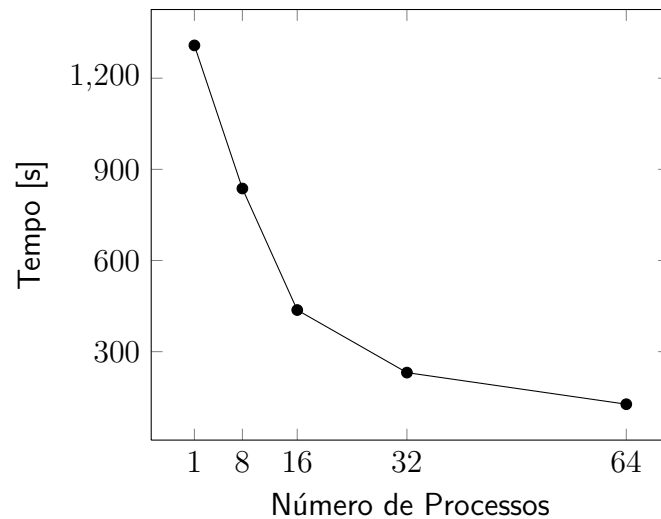


Figura 5.4: Gráfico Processos x Tempo para Anisotropia 299 e técnica de Inversão Recursiva na Máquina 1.

número de processos executando simultaneamente aumenta, há um aumento da concorrência utilizando o barramento e acessando a memória, por isso os ganhos acabam sendo inferiores.

Ao analisar a Tabela 3 pode-se observar que o ganho real é menor do que o ganho ideal pois são vários processos acessando a mesma memória. Porém se for analisado a relação entre os ganhos, vê-se que o algoritmo paralelo cumpre seu papel. Quando executado 8 processos em paralelo, que é o número de núcleos presentes no processador da Máquina 1 teste, o ganho é de 2.4 vezes. Porém, quando executado 16 processos em paralelo, observa-se que o ganho agora é aproximadamente o dobro do obtido para 8 processos em paralelo. O mesmo pode ser observado para 32 e 64 processos executados em paralelo. Conforme o número de processos é dobrado, o ganho aproximadamente dobra também.

Tabela 3: Aumento de Velocidade Ideal e Obtido na Máquina 1

Técnica	Anisotropia	Número de Processos	Ganho Ideal	Ganho Obtido
DNI	82	8	8	2.38
		16	16	4.19
		32	32	8.24
		64	64	12.31
DNI	299	8	8	2.58
		16	16	5.03
		32	32	9.33
		64	64	18.63
Recursividade	82	8	8	1.29
		16	16	2.17
		32	32	4.20
		64	64	7.13
Recursividade	299	8	8	1.56
		16	16	2.99
		32	32	5.66
		64	64	10.29

Na sequência, os algoritmos foram executados nas Máquinas 2 e 3. Na Máquina 2 foi executado apenas o algoritmo com dummy node, devido a quantidade de memória utilizada pelo algoritmo recursivo ser maior do que o apresentado na mesma. Os resultados obtidos pela Máquina 2 são apresentados nas Figuras 5.5 e 5.6

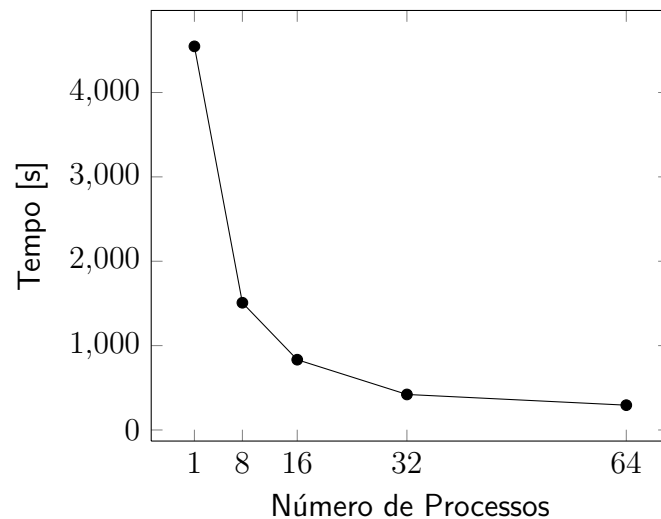


Figura 5.5: Gráfico Processos x Tempo para Anisotropia 82 e técnica DNI na Máquina 2.

Os resultados obtidos na máquina 2 são satisfatórios, levando aproximadamente 4500 segundos para o problema de anisotropia 82 e 15000 segundos para o problema de anisotropia 299. Este não possui resultados com 64 processos em paralelo devido a falta de memória na máquina para gerar as 64 instâncias. Os ganhos obtidos podem ser observados na

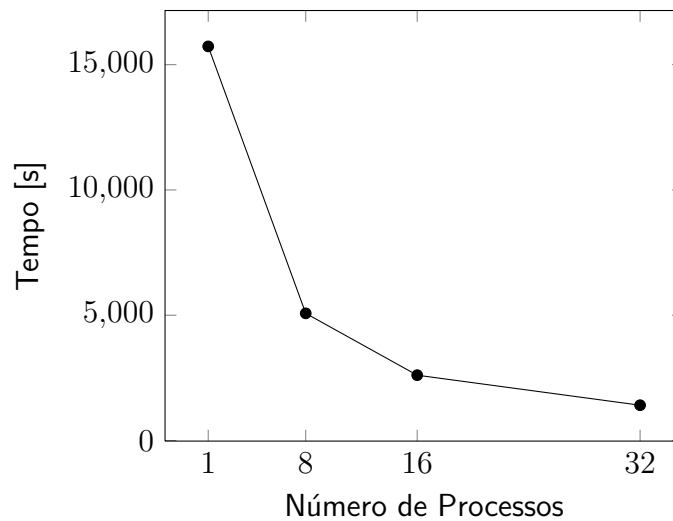


Figura 5.6: Gráfico Processos x Tempo para Anisotropia 299 e técnica DNI na Máquina 2.

Tabela 4: Aumento de Velocidade Ideal e Obtido na Máquina 2

Anisotropia	Número de Processos	Ganho Ideal	Ganho Obtido
82	8	8	3.01
	16	16	5.45
	32	32	10.80
	64	64	15.46
299	8	8	3.09
	16	16	6.03
	32	32	11.17

Tabela 4, onde o ganho para 8 processos é 3 vezes, e os ganhos, conforme dobram o número de processos, são aproximadamente o dobro do obtido para 8 processos.

Na Máquina 3, por ser um servidor e conseqüentemente mais poderoso que os dois notebooks envolvidos no experimento foi possível executar os dois problemas para os dois algoritmos. No algoritmo com DNI não há a necessidade de um valor de N maior que 2000, portanto manteve-se o mesmo, mas no algoritmo recursivo tentou-se utilizar N=500, porém o tempo tornou-se inviável para execução, pois não obteve resultado mesmo depois de várias horas em execução. O valor de N que obteve um tempo satisfatório foi N=300.

As Figuras 5.7, 5.8 e 5.9 apresentam os resultados obtidos nas execuções dos problemas na máquina 3. A Tabela 5 apresenta os ganhos obtidos nas execuções paralelas.

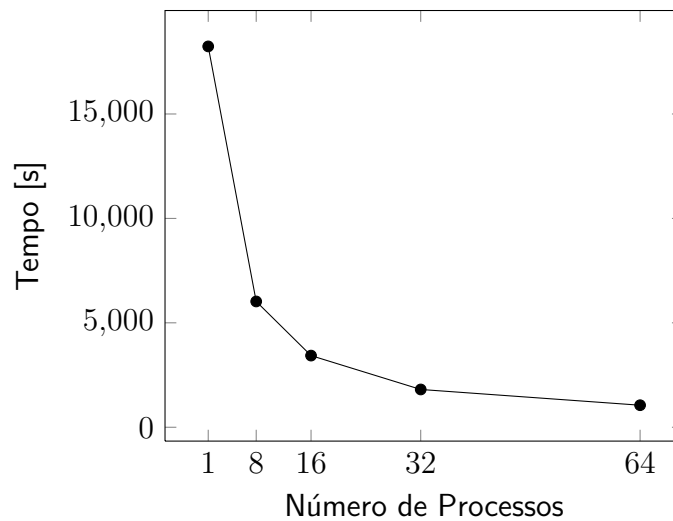


Figura 5.7: Gráfico Processos x Tempo para Anisotropia 299 e técnica DNI na Máquina 3.

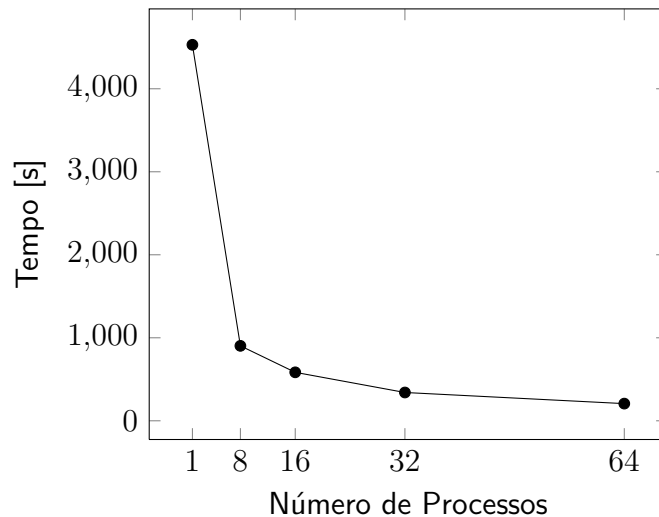


Figura 5.8: Gráfico Processos x Tempo para Anisotropia 82 e técnica de Inversão Recursiva na Máquina 3.

5.2 Anisotropia 1999

Como o desempenho do algoritmo melhorou, problemas até então inviáveis puderam ser calculados, como é o caso da anisotropia 1999. Problemas com esse grau de anisotropia ainda não possuem resultados obtidos na literatura, o que torna este, possivelmente, o primeiro resultado obtido para esse problema.

Para realizar esse cálculo, foi necessário calcular os valores de β_l , que foram obtidos através da fórmula

$$\beta_l = \frac{(2l+1)(L+1-l)}{(2l-1)(L+1+l)}\beta_{l-1}, \quad 0 \leq l \leq L \text{ e } \beta_0 = 1 \quad (5.1)$$

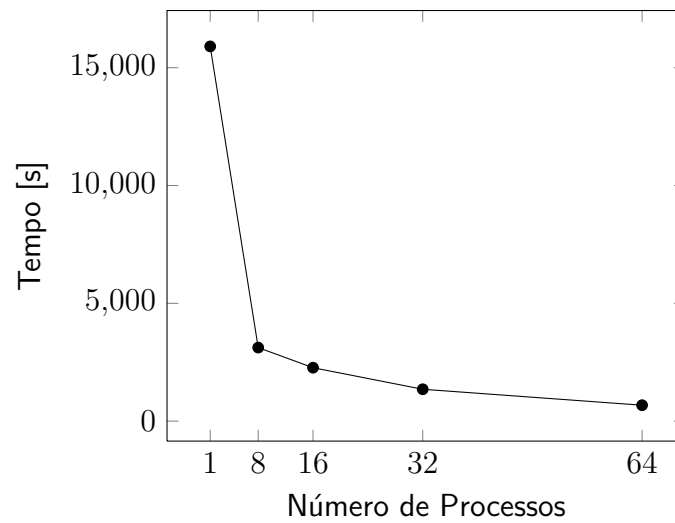


Figura 5.9: Gráfico Processos x Tempo para Anisotropia 299 e técnica de Inversão Recursiva na Máquina 3.

Tabela 5: Aumento de Velocidade Ideal e Obtido na Máquina 3

Técnica	Anisotropia	Número de Processos	Ganho Ideal	Ganho Obtido
DNI	299	8	8	3.02
		16	16	5.31
		32	32	10.08
		64	64	17.28
Recursividade	82	8	8	5.01
		16	16	7.75
		32	32	13.28
		64	64	21.88
Recursividade	299	8	8	5.09
		16	16	7.01
		32	32	11.76
		64	64	23.56

sabendo que L é o valor da anisotropia [39].

Tabela 6: Resultados do Problema para Anisotropia 1999, $\omega = 0.9$, $\mu_0 = 1$, $x_0 = 64$ e $N = 2000$

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	-4.374189(-7)	1.343883(-7)	1.408444(-7)	8.062501(-8)	1.190057(-8)	2.016133(-9)	
-0.9	-7.499347(-8)	-4.664413(-8)	-3.709401(-8)	-1.532433(-8)	6.581904(-10)	2.955615(-10)	
-0.8	7.451510(-10)	-1.223603(-8)	-1.551886(-10)	3.936930(-9)	-1.632668(-10)	-1.525729(-10)	
-0.7	5.303330(-7)	2.940312(-7)	1.556007(-7)	3.646515(-8)	-2.504109(-9)	-7.444666(-10)	
-0.6	-8.215435(-7)	-4.668808(-7)	-2.936253(-7)	-1.213337(-7)	-1.057757(-8)	-1.601333(-9)	
-0.5	-1.182286(-6)	-6.493910(-7)	-3.905677(-7)	-1.539726(-7)	-1.231518(-8)	-1.736194(-9)	
-0.4	2.375521(-7)	7.772230(-8)	5.124926(-9)	-2.620858(-8)	-5.746529(-9)	-9.347085(-10)	
-0.3	4.847764(-7)	2.702573(-7)	1.554069(-7)	5.248312(-8)	2.336039(-9)	3.042503(-10)	
-0.2	-8.241098(-8)	4.455765(-8)	6.888471(-8)	5.357195(-8)	8.646522(-9)	1.609039(-9)	
-0.1	1.810123(-7)	1.661663(-7)	1.294941(-7)	8.363547(-8)	1.530033(-8)	2.989010(-9)	
-0.0	-1.504758(-9)	3.719134(-7)	3.119852(-7)	1.778819(-7)	2.350536(-8)	4.585937(-9)	
0.0		3.719134(-7)	3.119852(-7)	1.778819(-7)	2.350536(-8)	4.585937(-9)	8.973645(-10)
0.1		9.758650(-7)	6.803304(-7)	2.975799(-7)	2.911592(-8)	7.047430(-9)	5.055802(-9)
0.2		9.422434(-7)	6.353980(-7)	2.568438(-7)	2.564665(-8)	1.532900(-8)	1.953502(-8)
0.3		-1.473360(-7)	-7.093893(-8)	-2.224963(-9)	1.647732(-8)	5.927734(-8)	8.509343(-8)
0.4		-6.564459(-7)	-5.005056(-7)	-1.646681(-7)	4.142800(-8)	3.158470(-7)	3.956213(-7)
0.5		2.806247(-7)	1.676886(-7)	4.337756(-8)	3.819729(-7)	1.925938(-6)	1.936449(-6)
0.6		-5.355292(-8)	1.884131(-7)	1.936519(-7)	4.205695(-6)	1.302757(-5)	1.000072(-5)
0.7		7.772791(-7)	4.269988(-7)	2.319297(-7)	5.661969(-5)	9.822771(-5)	5.482021(-5)
0.8		-1.284664(-6)	-1.088133(-6)	2.205096(-5)	9.563030(-4)	8.358067(-4)	3.213652(-4)
0.9		6.157383(-6)	2.955385(-4)	6.318720(-3)	2.145541(-2)	8.190622(-3)	2.034322(-3)
1.0		1.534874(2)	5.741555(1)	1.339336(1)	7.304404(-1)	9.571900(-2)	1.409427(-2)

Tabela 7: Aumento de Velocidade Ideal e Obtido para o Problema de Anisotropia 1999

Anisotropia	Número de Processos	Ganho Ideal	Ganho Obtido
1999	8	8	3.51
	16	16	6.89
	32	32	13.36

Seguindo a proposta do trabalho, este problema também foi paralelizado e o ganho de performance analisado. A Figura 5.10 mostra os tempos de execução para o problema sequencial e paralelo e a Tabela 7 mostra os ganhos obtidos de desempenho.

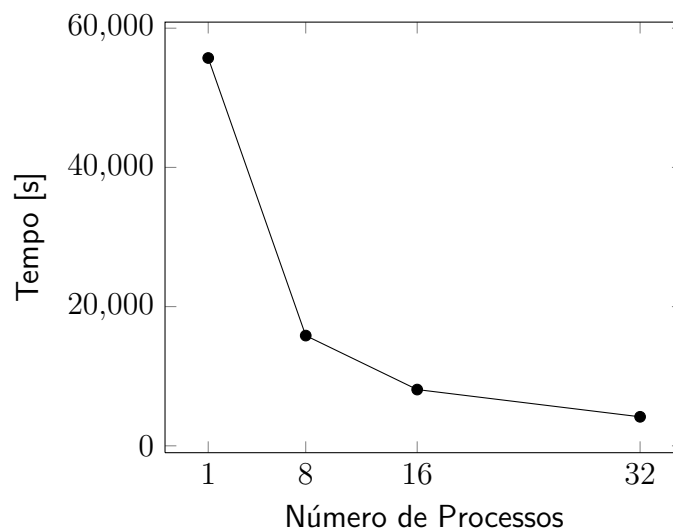


Figura 5.10: Gráfico Processos x Tempo para Anisotropia 1999.

5.3 Cálculo dos Erros dos Problemas

Sabendo que o resultado de cada problema é dado pela soma dos resultados dos $L+1$ problemas independentes, foi feito um algoritmo que calculava o erro entre o resultado final previamente calculado e o resultado parcial de cada iteração do problema. Calculando o erro de todas as posições da tabela utilizando a Equação (5.2)

$$e = \left\| \frac{R_f - R_p}{R_f} \right\| \quad (5.2)$$

onde R_f é o resultado final, obtido após a soma dos $L + 1$ problemas e R_p é o resultado parcial calculado até o momento.

Para calcular, foi utilizado um erro de 10^{-5} . A Tabela 8 mostra o menor valor parcial de l para que o resultado não obtenha erro maior que 10^{-5} .

Tabela 8: Menores Valores para Obter um Resultado Satisfatório

Anisotropia	Menor Valor l
8	7
82	51
299	213

Com isso, não é preciso calcular todos os subproblemas independentes para obter o resultado final, tornando a execução dos problemas mais rápida. Usando esse método em conjunto com a paralelização é possível obter resultados considerados ótimos em um período de tempo menor. Não foi feita a verificação para os valores de $L = 1999$.

6 Conclusões

Nesse trabalho foi implementada a paralelização dos dois algoritmos para solução da equação de transporte, um utilizando a técnica de inversão recursiva e outro utilizando a técnica de inclusão de *dummy nodes*. Ambos foram executados para diferentes problemas e números de processos variáveis afim de analisar o ganho no desempenho.

Considerando a arquitetura para a qual o algoritmo recursivo foi programado, não foi possível realizar sua execução com o valor de N grande, mesmo uma das máquinas sendo um servidor com desempenho relativamente superior as outras. Apesar disso, foi possível ver o ganho de performance de ambos os algoritmos em todas as máquinas de teste, mesmo que inferior ao ideal. Isso se deve ao fato de as máquinas de testes possuírem uma arquitetura de memória compartilhada, assim, quando a execução é feita de forma sequencial, toda a banda memória está disponível para apenas um processo e no caso das execuções em paralelo, a banda é compartilhada entre todos os processos, o que ocasionará concorrência.

Também foram executados os problemas presentes na literatura, e em comparação com esses resultados, os encontrados são mais precisos devido à utilização de um valor de N maior. Outro resultado interessante que se originou deste trabalho, foi o cálculo de um grupo de problemas com anisotropia 1999, que não possui resultados publicados, tornando esta uma contribuição importante.

Portanto, é possível concluir que a paralelização foi obtida de forma satisfatória para ambos os algoritmos, pois os resultados mostram um aumento no desempenho dos mesmos.

6.1 Trabalhos Futuros

O próximo passo desse trabalho, é expandir a formulação e os algoritmos para outros grupos de problemas mais complexos, testando a eficiência do algoritmo para estes problemas.

Referências

- 1 BRANCHER, J. *Formulação Analítica para Solução de Problema de Ordenadas Discretas para Valores de N Grandes*. Tese (Doutorado) — UFRGS, 1998.
- 2 SEGATTO, C. F. *Código computacional para Solução do método LTSN sem simetria azimutal utilizando Dummy Nodes*. 2005.
- 3 DUDERSTADT, J. J.; MARTIN, W. R. *Transport Theory*. [S.l.]: John Wiley & Sons, 1975.
- 4 DUDERSTADT, J.; HAMILTON, L. *Nuclear Reactor Analysis*. [S.l.]: John Wiley & Sons, 1976.
- 5 VILHENA, M. T.; BARICHELLO, L. B. A new analytical approach to solve neutron transport equation. *Kerntechnik*, v. 56, p. 334–336, 1991.
- 6 BARICHELLO, L. B.; VILHENA, M. T. A general analytical approach to the one group, one-dimensional transport equa. *Kerntechnik*, v. 58, p. 182–184, 1993.
- 7 ZABADAL, J. R.; BARICHELLO, L. B.; VILHENA, M. T. Solução analítica computacional da equação bidimensional de transporte de radiação. In: *Encontro Nacional de Física de Reatores e Termohidráulica*. Caxambú, MG: Brasil, 1993.
- 8 ZABADAL, J.; VILHENA, M.; BARICHELLO, L. Solution of the three-dimensional one-group discrete ordinates problems by the ltsn method. *Annals of Nuclear Energy*, v. 22, n. 2, p. 131 – 134, 1995.
- 9 CASE, K. Elementary solutions of the transport equation and their applications. *Annals of Physics*, v. 9, n. 1, p. 1 – 23, 1960.
- 10 PAZOS, R. P.; VILHENA, M. T. Convergence of the ltsn method: Approach of c_0 semigroups. *Progress in Nuclear Energy*, v. 34, n. 1, p. 77 – 86, 1999.
- 11 BRANCHER, J.; CARDONA, A.; VILHENA, M. T. de. A recursive method to invert the ltsn matrix. *Progress in Nuclear Energy*, v. 33, n. 4, p. 393 – 401, 1998.
- 12 SIMCH, M.; SEGATTO, C.; VILHENA, M. An analytical solution for the radiative transfer equation with polarization in a slab by the method. *Journal of Quantitative Spectroscopy and Radiative Transfer*, v. 97, n. 3, p. 424 – 435, 2006.
- 13 MARONA, D. V. *Solução LTSN da Equação de Transporte em Geometria Cartesiana Unidimensional para $c = 1$* . Dissertação (Mestrado) — UFRGS, 2007.
- 14 ROST, B. *Solução da Equação de Transferência Radiativa Unidimensional em Geometria Cartesiana em Nuvens pela Transformada de Laplace*. Dissertação (Mestrado) — UFRGS, 2008.

- 15 JIN, H. et al. High performance computing using mpi and openmp on multi-core parallel systems. *Parallel Computing*, v. 37, p. 562–575, 2011.
- 16 LI, G. et al. Formal specification of mpi 2.0: Case study in specifying a practical concurrent programming api. *Science of Computer Programming*, v. 76, p. 65–81, 2011.
- 17 CHENG, P.; GU, Y. An analysis of multicore specific optimization in mpi implementations. In: *IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*. Shangai, China: [s.n.], 2012. p. 1874–1878.
- 18 BASILI, V. R. et al. Understanding the high-performance-computing community: A software engineer's perspective. *IEEE Software*, v. 25, p. 29–36, 2008.
- 19 GOPALAKRISHNAN, G.; KIRBY, R. M. Formal methods for mpi programs. *Electronic Notes in Theoretical Computer Science*, v. 193, p. 19–27, 2007.
- 20 GROPP, W.; LUSK, E. The mpi message-passing interface standard: Overview and status. *Advances in Parallel Computing*, v. 10, p. 265–269, 1995.
- 21 SEGATTO, C. F. *Formulação LTSN para Problemas de Transporte sem Simetria Azimutal e Problemas Dependentes do Tempo*. Tese (Doutorado) — UFRGS, 1995.
- 22 SEGATTO, C. F.; VILHENA, M. T.; BRANCHER, J. D. The one-dimensional Itsn formulation for high degree of anisotropy. *J. Quant. Spectrosc. Radiat. Transfer*, v. 61, p. 39–43, 1999.
- 23 ARKERYD, L. On the boltzmann equation. part i: Existence. *Archive for Rational Mechanics and Analysis*, v. 45, p. 1–16, 1972.
- 24 ARKERYD, L. On the boltzmann equation. part ii: The full initial value problem. *Archive for Rational Mechanics and Analysis*, v. 45, p. 17–34, 1972.
- 25 BOLTZMANN, L. *Lectures on Gas Theory*. [S.l.]: Dover Publications, 1964.
- 26 CHANDRASEKHAR, S. *Radiative Transfer*. [S.l.]: Dover Publications, 1960.
- 27 GIL, A.; SEGURA, J.; TEMME, N. M. *Numerical Methods for Special Functions*. [S.l.]: Society for Industrial and Applied Mathematics, 2007.
- 28 BARICHELLO, L. B. *Formulação Analítica para Solução do Problema de Ordenada Discreta Unidimensional*. Tese (Doutorado) — UFRGS, 1992.
- 29 GOLUB, G. H.; LOAN, C. F. *Matrix Computations*. [S.l.]: Johns Hopkins University Press, 1989.
- 30 INTEL. *Parallel Computing: Background*. 2012.
- 31 FLYNN, M. J. Some computer organization and their effectiveness. *IEEE Transactions on Computers*, C-21, p. 948–960, 1972.
- 32 DETONI, G. G. *MTC: Modelo de Programação Paralela Baseado na Perspectiva Conexionista*. Dissertação (Mestrado) — UFRGS, 2010.

- 33 LAWSON, C. L. et al. Basic linear algebra subprograms for fortran usage. *ACM Transactions on Mathematical Software*, v. 5, p. 308–323, 1979.
- 34 DONGARRA, J. J. et al. An extended set of fortran basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, v. 14, p. 1–17, 1988.
- 35 DONGARRA, J. J. et al. A set of level 3 basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, v. 16, p. 1–17, 1990.
- 36 NETLIB. *LAPACK - Linear Algebra PACKage*. 2012. Disponível em: <<http://www.netlib.org/lapack>>.
- 37 NETLIB. *BLAS*. 2012. Disponível em: <<http://www.netlib.org/blas>>.
- 38 NESTEROV, O. A simple parallelization technique with mpi for ocean circulation models. *Journal of Parallel and Distributed Computing*, v. 70, p. 35–44, 2010.
- 39 SEGATTO, C. et al. A solution for the non-linear s_n radiative-conductive problem in a grey plane-parallel participating medium. *International Journal of Thermal Sciences*, v. 49, p. 1493–1499, 2010.
- 40 GARCIA, R. D. M.; SIEWERT, C. E. Benchmark results in radiative transfer. *Transport Theory and Statistical Physics*, v. 14, p. 437–483, 1985.

APÊNDICES

APÊNDICE A – Resultados Obtidos em Comparação com a Literatura

Como forma de comparar os resultados obtidos, alguns problemas presentes na literatura foram executados pelos algoritmos. Os problemas foram retirados de [40].

A Tabela 9 mostra os problemas apresentados em [40], onde ω é o termo de espalhamento, μ_0 é o cosseno do ângulo polar, x_0 é o comprimento da placa, L é o valor da anisotropia e N é o tamanho da quadratura utilizada. O primeiro problema tem seus resultados apresentados nas Tabelas 10 até 18. Os resultados dos problemas com anisotropia 82 são apresentados nas Tabelas 19 e 20, assim como o resultado para anisotropia 299 é apresentado na Tabela 21.

Tabela 9: Valores dos Parâmetros dos Problemas Executados

ω	μ_0	x_0	L	N
0.95	0.5	1	8	500
0.9	0.5	1	82	1000
0.9	1	1	82	1000
0.9	0.5	64	299	1000

Tabela 10: Resultado parcial, $M = 0$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	4.768049(-2)	4.419108(-2)	4.064657(-2)	3.370972(-2)	1.585714(-2)	5.452899(-3)	5.452899(-3)
-0.9	6.456409(-2)	6.037410(-2)	5.601367(-2)	4.728978(-2)	2.381196(-2)	9.010733(-3)	9.010733(-3)
-0.8	8.458719(-2)	7.967604(-2)	7.444230(-2)	6.375375(-2)	3.382852(-2)	1.369171(-2)	1.369171(-2)
-0.7	1.083491(-1)	1.027184(-1)	9.656756(-2)	8.374855(-2)	4.649158(-2)	1.988430(-2)	1.988430(-2)
-0.6	1.365037(-1)	1.302033(-1)	1.231273(-1)	1.080625(-1)	6.260459(-2)	2.817191(-2)	2.817191(-2)
-0.5	1.696762(-1)	1.628512(-1)	1.549185(-1)	1.376177(-1)	8.329176(-2)	3.947887(-2)	3.947887(-2)
-0.4	2.082286(-1)	2.012001(-1)	1.926253(-1)	1.733576(-1)	1.101239(-1)	5.536269(-2)	5.536269(-2)
-0.3	2.516757(-1)	2.450647(-1)	2.363092(-1)	2.158060(-1)	1.451439(-1)	7.862933(-2)	7.862933(-2)
-0.2	2.975206(-1)	2.924005(-1)	2.842403(-1)	2.637993(-1)	1.900348(-1)	1.145568(-1)	1.145568(-1)
-0.1	3.401208(-1)	3.384750(-1)	3.319832(-1)	3.128871(-1)	2.412167(-1)	1.706965(-1)	1.706965(-1)
-0.0	1.045519(-1)	1.439078(-1)	1.651655(-1)	1.885318(-1)	1.945119(-1)	1.687629(-1)	1.687629(-1)
0.0		1.439078(-1)	1.651655(-1)	1.885318(-1)	1.945119(-1)	1.687629(-1)	1.170335(-1)
0.1		1.556168(-1)	2.514247(-1)	3.388631(-1)	3.307021(-1)	2.698550(-1)	2.030712(-1)
0.2		9.243530(-2)	1.650787(-1)	2.618855(-1)	3.355167(-1)	3.015528(-1)	2.442402(-1)
0.3		6.707140(-2)	1.241431(-1)	2.103819(-1)	3.148046(-1)	3.097121(-1)	2.703012(-1)
0.4		5.301610(-2)	9.989308(-2)	1.753036(-1)	2.881820(-1)	3.029381(-1)	2.810651(-1)
0.5		4.371756(-2)	8.323769(-2)	1.492507(-1)	2.613037(-1)	2.884631(-1)	2.808217(-1)
0.6		3.678595(-2)	7.051887(-2)	1.283226(-1)	2.352585(-1)	2.697922(-1)	2.730285(-1)
0.7		3.114075(-2)	5.999045(-2)	1.103862(-1)	2.099385(-1)	2.484751(-1)	2.598217(-1)
0.8		2.622224(-2)	5.071636(-2)	9.419983(-2)	1.849817(-1)	2.251786(-1)	2.424530(-1)
0.9		2.171312(-2)	4.215361(-2)	7.900372(-2)	1.600341(-1)	2.001864(-1)	2.216665(-1)
1.0		1.742303(-2)	3.397151(-2)	6.431911(-2)	1.348183(-1)	1.736271(-1)	1.979321(-1)

Tabela 11: Resultado parcial, $M = 1$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.9	3.601794(-2)	3.367456(-2)	3.120629(-2)	2.627864(-2)	1.329022(-2)	5.197980(-3)	
-0.8	6.165931(-2)	5.785879(-2)	5.381592(-2)	4.567628(-2)	2.383371(-2)	9.729075(-3)	
-0.7	9.103548(-2)	8.572061(-2)	8.001115(-2)	6.842223(-2)	3.678194(-2)	1.562016(-2)	
-0.6	1.262530(-1)	1.192950(-1)	1.117415(-1)	9.628212(-2)	5.333972(-2)	2.355845(-2)	
-0.5	1.689097(-1)	1.601920(-1)	1.506088(-1)	1.308179(-1)	7.485593(-2)	3.448759(-2)	
-0.4	2.204242(-1)	2.099252(-1)	1.981880(-1)	1.736912(-1)	1.031419(-1)	4.994581(-2)	
-0.3	2.817129(-1)	2.696347(-1)	2.557774(-1)	2.264665(-1)	1.406329(-1)	7.268192(-2)	
-0.2	3.522235(-1)	3.391913(-1)	3.235023(-1)	2.896655(-1)	1.898676(-1)	1.078996(-1)	
-0.1	4.285609(-1)	4.162661(-1)	3.994054(-1)	3.613735(-1)	2.494432(-1)	1.635126(-1)	
-0.0	9.306851(-2)	1.337744(-1)	1.541652(-1)	1.737080(-1)	1.661642(-1)	1.340865(-1)	
0.0		1.337744(-1)	1.541652(-1)	1.737080(-1)	1.661642(-1)	1.340865(-1)	8.628795(-2)
0.1		2.120982(-1)	3.363560(-1)	4.366730(-1)	3.784844(-1)	2.795068(-1)	1.923671(-1)
0.2		1.276501(-1)	2.240928(-1)	3.440331(-1)	3.980378(-1)	3.263117(-1)	2.411861(-1)
0.3		9.314111(-2)	1.695458(-1)	2.784541(-1)	3.791522(-1)	3.428827(-1)	2.744661(-1)
0.4		7.343425(-2)	1.361084(-1)	2.315987(-1)	3.474675(-1)	3.370784(-1)	2.879028(-1)
0.5		5.979336(-2)	1.119985(-1)	1.947142(-1)	3.112155(-1)	3.174291(-1)	2.848624(-1)
0.6		4.897958(-2)	9.236362(-2)	1.628865(-1)	2.721568(-1)	2.880949(-1)	2.685569(-1)
0.7		3.944038(-2)	7.471777(-2)	1.331066(-1)	2.298413(-1)	2.503719(-1)	2.405026(-1)
0.8		3.018102(-2)	5.736556(-2)	1.029689(-1)	1.824128(-1)	2.033198(-1)	2.001294(-1)
0.9		2.005669(-2)	3.821721(-2)	6.900564(-2)	1.248170(-1)	1.418012(-1)	1.424603(-1)
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tabela 12: Resultado parcial, $M = 2$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.9	6.133675(-3)	5.632577(-3)	5.141457(-3)	4.227276(-3)	2.081091(-3)	8.511354(-4)	
-0.8	1.404721(-2)	1.291993(-2)	1.181249(-2)	9.746385(-3)	4.867575(-3)	2.029075(-3)	
-0.7	2.423737(-2)	2.232714(-2)	2.044704(-2)	1.693300(-2)	8.590095(-3)	3.657030(-3)	
-0.6	3.729754(-2)	3.441456(-2)	3.157237(-2)	2.625196(-2)	1.355543(-2)	5.912650(-3)	
-0.5	5.393620(-2)	4.985896(-2)	4.583212(-2)	3.828375(-2)	2.018510(-2)	9.068031(-3)	
-0.4	7.498504(-2)	6.946663(-2)	6.400302(-2)	5.374854(-2)	2.907364(-2)	1.357082(-2)	
-0.3	1.013654(-1)	9.414789(-2)	8.697336(-2)	7.349266(-2)	4.105372(-2)	2.021835(-2)	
-0.2	1.339679(-1)	1.248063(-1)	1.156243(-1)	9.834334(-2)	5.712585(-2)	3.052318(-2)	
-0.1	1.735375(-1)	1.623210(-1)	1.508034(-1)	1.289338(-1)	7.754072(-2)	4.687639(-2)	
-0.0	1.837781(-2)	2.695047(-2)	3.063056(-2)	3.323059(-2)	2.835191(-2)	2.102767(-2)	
0.0		2.695047(-2)	3.063056(-2)	3.323059(-2)	2.835191(-2)	2.102767(-2)	1.251716(-2)
0.1		9.138436(-2)	1.411978(-1)	1.743379(-1)	1.305058(-1)	8.620881(-2)	5.452457(-2)
0.2		5.478303(-2)	9.400050(-2)	1.383859(-1)	1.419969(-1)	1.054087(-1)	7.123694(-2)
0.3		3.942623(-2)	7.025481(-2)	1.110339(-1)	1.359662(-1)	1.127752(-1)	8.320776(-2)
0.4		3.032531(-2)	5.508183(-2)	9.038419(-2)	1.229047(-1)	1.102400(-1)	8.742052(-2)
0.5		2.375721(-2)	4.364665(-2)	7.328562(-2)	1.066760(-1)	1.011036(-1)	8.465468(-2)
0.6		1.836365(-2)	3.399062(-2)	5.795807(-2)	8.846510(-2)	8.728415(-2)	7.615248(-2)
0.7		1.353476(-2)	2.518339(-2)	4.341380(-2)	6.861367(-2)	6.978968(-2)	6.285790(-2)
0.8		8.961794(-3)	1.673823(-2)	2.909123(-2)	4.721603(-2)	4.918042(-2)	4.542688(-2)
0.9		4.479301(-3)	8.390169(-3)	1.467387(-2)	2.431991(-2)	2.581913(-2)	2.434091(-2)
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tabela 13: Resultado parcial, $M = 3$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.9	4.718812(-4)	4.279220(-4)	3.865042(-4)	3.123226(-4)	1.493034(-4)	6.129727(-5)	
-0.8	1.636998(-3)	1.484918(-3)	1.342017(-3)	1.086495(-3)	5.248748(-4)	2.189394(-4)	
-0.7	3.627937(-3)	3.292559(-3)	2.978070(-3)	2.416434(-3)	1.181171(-3)	5.013031(-4)	
-0.6	6.651238(-3)	6.040873(-3)	5.469425(-3)	4.449919(-3)	2.205502(-3)	9.552621(-4)	
-0.5	1.095582(-2)	9.960451(-3)	9.029682(-3)	7.370676(-3)	3.715628(-3)	1.650916(-3)	
-0.4	1.683604(-2)	1.532602(-2)	1.391537(-2)	1.140388(-2)	5.873204(-3)	2.700309(-3)	
-0.3	2.463607(-2)	2.246026(-2)	2.042907(-2)	1.681912(-2)	8.901963(-3)	4.299043(-3)	
-0.2	3.475356(-2)	3.173263(-2)	2.891223(-2)	2.391186(-2)	1.307992(-2)	6.815205(-3)	
-0.1	4.770495(-2)	4.362027(-2)	3.978924(-2)	3.299678(-2)	1.854681(-2)	1.084046(-2)	
-0.0	1.590522(-3)	2.400970(-3)	2.715320(-3)	2.888249(-3)	2.319082(-3)	1.651391(-3)	
0.0		2.400970(-3)	2.715320(-3)	2.888249(-3)	2.319082(-3)	1.651391(-3)	9.507397(-4)
0.1		2.659464(-2)	4.046958(-2)	4.859840(-2)	3.378697(-2)	2.121421(-2)	1.301557(-2)
0.2		1.591243(-2)	2.694676(-2)	3.877783(-2)	3.750522(-2)	2.662632(-2)	1.736224(-2)
0.3		1.128591(-2)	1.986953(-2)	3.076680(-2)	3.580141(-2)	2.859529(-2)	2.042722(-2)
0.4		8.438024(-3)	1.515507(-2)	2.440115(-2)	3.168290(-2)	2.748857(-2)	2.118069(-2)
0.5		6.319581(-3)	1.148832(-2)	1.894932(-2)	2.642420(-2)	2.429613(-2)	1.981805(-2)
0.6		4.567164(-3)	8.369973(-3)	1.403344(-2)	2.057005(-2)	1.972973(-2)	1.679974(-2)
0.7		3.043571(-3)	5.610013(-3)	9.517503(-3)	1.447249(-2)	1.433141(-2)	1.261408(-2)
0.8		1.716690(-3)	3.177915(-3)	5.439493(-3)	8.507284(-3)	8.636501(-3)	7.803042(-3)
0.9		6.328574(-4)	1.175446(-3)	2.025932(-3)	3.239647(-3)	3.354923(-3)	3.095824(-3)
1.0		0.0	0.0	0.0	0.0	0.0	0.0

Tabela 14: Resultado parcial, $M = 4$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.9	6.692578(-5)	6.022392(-5)	5.408382(-5)	4.334360(-5)	2.053040(-5)	8.514938(-6)	
-0.8	3.024942(-4)	2.723772(-4)	2.447942(-4)	1.965479(-4)	9.390142(-5)	3.939727(-5)	
-0.7	7.617193(-4)	6.864177(-4)	6.174715(-4)	4.968770(-4)	2.398511(-4)	1.020614(-4)	
-0.6	1.503583(-3)	1.356224(-3)	1.221339(-3)	9.854313(-4)	4.817661(-4)	2.086958(-4)	
-0.5	2.591465(-3)	2.340124(-3)	2.110141(-3)	1.708028(-3)	8.484549(-4)	3.762918(-4)	
-0.4	4.093629(-3)	3.701427(-3)	3.342750(-3)	2.716118(-3)	1.376985(-3)	6.308685(-4)	
-0.3	6.083241(-3)	5.508147(-3)	4.982689(-3)	4.066263(-3)	2.115986(-3)	1.016674(-3)	
-0.2	8.639010(-3)	7.831887(-3)	7.095262(-3)	5.814296(-3)	3.122067(-3)	1.615591(-3)	
-0.1	1.186590(-2)	1.076454(-2)	9.758979(-3)	8.013569(-3)	4.412702(-3)	2.554970(-3)	
-0.0	1.085634(-4)	1.599635(-4)	1.776992(-4)	1.840810(-4)	1.402437(-4)	9.662528(-5)	
0.0		1.599635(-4)	1.776992(-4)	1.840810(-4)	1.402437(-4)	9.662528(-5)	5.266686(-5)
0.1		6.425984(-3)	9.728861(-3)	1.157828(-2)	7.869328(-3)	4.869187(-3)	2.963653(-3)
0.2		3.728299(-3)	6.286423(-3)	8.981146(-3)	8.532916(-3)	5.980940(-3)	3.861843(-3)
0.3		2.539619(-3)	4.453705(-3)	6.852124(-3)	7.853490(-3)	6.206476(-3)	4.394817(-3)
0.4		1.802667(-3)	3.226021(-3)	5.163737(-3)	6.614547(-3)	5.686281(-3)	4.347859(-3)
0.5		1.262671(-3)	2.287717(-3)	3.752878(-3)	5.168645(-3)	4.713261(-3)	3.818188(-3)
0.6		8.356436(-4)	1.526641(-3)	2.546542(-3)	3.689682(-3)	3.512158(-3)	2.971817(-3)
0.7		4.935303(-4)	9.070206(-4)	1.531361(-3)	2.303320(-3)	2.264728(-3)	1.981701(-3)
0.8		2.325279(-4)	4.292642(-4)	7.314000(-4)	1.132098(-3)	1.141605(-3)	1.025750(-3)
0.9		6.200290(-5)	1.148624(-4)	1.971129(-4)	3.120966(-4)	3.211405(-4)	2.947823(-4)
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tabela 15: Resultado parcial, $M = 5$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.9	6.280628(-6)	5.640377(-6)	5.057803(-6)	4.044688(-6)	1.910711(-6)	7.93860(-7)	
-0.8	3.731241(-5)	3.353314(-5)	3.009401(-5)	2.411155(-5)	1.148574(-5)	4.824415(-6)	
-0.7	1.0745(77-4)	9.665568(-5)	8.682525(-5)	6.972102(-5)	3.355035(-5)	1.428491(-5)	
-0.6	2.294699(-4)	2.066075(-4)	1.858023(-4)	1.495992(-4)	7.289508(-5)	3.158243(-5)	
-0.5	4.151673(-4)	3.742346(-4)	3.369928(-4)	2.722000(-4)	1.347408(-4)	5.974515(-5)	
-0.4	6.753721(-4)	6.095801(-4)	5.497512(-4)	4.457394(-4)	2.251373(-4)	1.030903(-4)	
-0.3	1.019567(-3)	9.215041(-4)	8.324201(-4)	6.778223(-4)	3.513197(-4)	1.686508(-4)	
-0.2	1.455575(-3)	1.317058(-3)	1.191419(-3)	9.740589(-4)	5.207580(-4)	2.691373(-4)	
-0.1	1.992725(-3)	1.803876(-3)	1.632722(-3)	1.337341(-3)	7.328231(-4)	4.235135(-4)	
-0.0	3.358001(-6)	4.856036(-6)	5.324522(-6)	5.413151(-6)	3.977970(-6)	2.678977(-6)	
0.0		4.856036(-6)	5.324522(-6)	5.413151(-6)	3.977970(-6)	2.678977(-6)	1.390540(-6)
0.1		1.043596(-3)	1.577735(-3)	1.873090(-3)	1.265818(-3)	7.806031(-4)	4.745245(-4)
0.2		5.870869(-4)	9.887002(-4)	1.409730(-3)	1.333316(-3)	9.317561(-4)	6.004442(-4)
0.3		3.838984(-4)	6.724900(-4)	1.032825(-3)	1.179199(-3)	9.295690(-4)	6.570165(-4)
0.4		2.584692(-4)	4.620746(-4)	7.384285(-4)	9.426373(-4)	8.085922(-4)	6.172565(-4)
0.5		1.690904(-4)	3.060635(-4)	5.013262(-4)	6.882654(-4)	6.264053(-4)	5.067013(-4)
0.6		1.022909(-4)	1.867063(-4)	3.109988(-4)	4.492763(-4)	4.268976(-4)	3.607333(-4)
0.7		5.342044(-5)	9.809326(-5)	1.653939(-4)	2.480779(-4)	2.435166(-4)	2.128168(-4)
0.8		2.096747(-5)	3.867627(-5)	6.581497(-5)	1.016035(-4)	1.022967(-4)	9.180688(-5)
0.9		4.031126(-6)	7.462081(-6)	1.279010(-5)	2.020030(-5)	2.075486(-5)	1.903003(-5)
1.0		0.0	0.0	0.0	0.0	0.0	0.0

Tabela 16: Resultado parcial, $M = 6$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.9	3.646121(-7)	3.273384(-7)	2.934593(-7)	2.345967(-7)	1.107739(-7)	4.603599(-8)	
-0.8	2.904795(-6)	2.609761(-6)	2.341561(-6)	1.875436(-6)	8.929577(-7)	3.751320(-7)	
-0.7	9.727383(-6)	8.746866(-6)	7.855457(-6)	6.305815(-6)	3.032903(-6)	1.291462(-6)	
-0.6	2.278621(-5)	2.050978(-5)	1.844025(-5)	1.484214(-5)	7.228348(-6)	3.131890(-6)	
-0.5	4.378871(-5)	3.945952(-5)	3.552459(-5)	2.868446(-5)	1.419122(-5)	6.292495(-6)	
-0.4	7.409434(-5)	6.685595(-5)	6.028019(-5)	4.885807(-5)	2.466326(-5)	1.129279(-5)	
-0.3	1.145965(-4)	1.035417(-4)	9.350972(-5)	7.611515(-5)	3.942634(-5)	1.892487(-5)	
-0.2	1.656136(-4)	1.498027(-4)	1.354780(-4)	1.107186(-4)	5.915236(-5)	3.056649(-5)	
-0.1	2.271905(-4)	2.055802(-4)	1.860219(-4)	1.523030(-4)	8.339207(-5)	4.818220(-5)	
-0.0	4.798582(-8)	6.874546(-8)	7.475648(-8)	7.505290(-8)	5.379725(-8)	3.565896(-8)	
0.0		6.874546(-8)	7.475648(-8)	7.505290(-8)	5.379725(-8)	3.565896(-8)	1.779655(-8)
0.1		1.161115(-4)	1.754936(-4)	2.082549(-4)	1.406001(-4)	8.665944(-5)	5.267458(-5)
0.2		6.359698(-5)	1.070777(-4)	1.526212(-4)	1.442356(-4)	1.007470(-4)	6.490644(-5)
0.3		4.005498(-5)	7.015114(-5)	1.077051(-4)	1.228869(-4)	9.683301(-5)	6.842330(-5)
0.4		2.564523(-5)	4.583788(-5)	7.323046(-5)	9.342580(-5)	8.011206(-5)	6.114069(-5)
0.5		1.569772(-5)	2.840851(-5)	4.651965(-5)	6.383098(-5)	5.807548(-5)	4.696723(-5)
0.6		8.689784(-6)	1.585822(-5)	2.640828(-5)	3.813030(-5)	3.622043(-5)	3.060053(-5)
0.7		4.014352(-6)	7.370126(-6)	1.242358(-5)	1.862531(-5)	1.827788(-5)	1.597062(-5)
0.8		1.312170(-6)	2.420025(-6)	4.117153(-6)	6.353017(-6)	6.394727(-6)	5.737972(-6)
0.9		1.817103(-7)	3.363164(-7)	5.763203(-7)	9.098207(-7)	9.345713(-7)	8.567601(-7)
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tabela 17: Resultado parcial, $M = 7$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.9	1.270311(-8)	1.140408(-8)	1.022351(-8)	8.172554(-9)	3.858799(-9)	1.603695(-9)	1.603695(-9)
-0.8	1.429731(-7)	1.284471(-7)	1.152438(-7)	9.229923(-8)	4.394446(-8)	1.846141(-8)	1.846141(-8)
-0.7	5.796682(-7)	5.212195(-7)	4.680889(-7)	3.757349(-7)	1.807071(-7)	7.694913(-8)	7.694913(-8)
-0.6	1.536486(-6)	1.382936(-6)	1.243360(-6)	1.000714(-6)	4.873349(-7)	2.111535(-7)	2.111535(-7)
-0.5	3.210824(-6)	2.893282(-6)	2.604693(-6)	2.103086(-6)	1.040409(-6)	4.613269(-7)	4.613269(-7)
-0.4	5.750182(-6)	5.188249(-6)	4.677822(-6)	3.791297(-6)	1.913702(-6)	8.762441(-7)	8.762441(-7)
-0.3	9.224431(-6)	8.334265(-6)	7.526555(-6)	6.126216(-6)	3.173057(-6)	1.523077(-6)	1.523077(-6)
-0.2	1.360596(-5)	1.230652(-5)	1.112938(-5)	9.095016(-6)	4.858720(-6)	2.510674(-6)	2.510674(-6)
-0.1	1.879139(-5)	1.700314(-5)	1.538499(-5)	1.259562(-5)	6.896011(-6)	3.984282(-6)	3.984282(-6)
-0.0	3.591189(-10)	5.114524(-10)	5.525438(-10)	5.490746(-10)	3.855730(-10)	2.523104(-10)	2.523104(-10)
0.0		5.114524(-10)	5.525438(-10)	5.490746(-10)	3.855730(-10)	2.523104(-10)	1.215927(-10)
0.1		9.389913(-6)	1.419166(-5)	1.684005(-5)	1.136800(-5)	7.006308(-6)	4.258668(-6)
0.2		4.997866(-6)	8.414623(-6)	1.199308(-5)	1.133309(-5)	7.915609(-6)	5.099522(-6)
0.3		3.021410(-6)	5.291469(-6)	8.123823(-6)	9.268192(-6)	7.302854(-6)	5.160144(-6)
0.4		1.830883(-6)	3.272409(-6)	5.227797(-6)	6.669025(-6)	5.718418(-6)	4.364129(-6)
0.5		1.042548(-6)	1.886677(-6)	3.089373(-6)	4.238736(-6)	3.856398(-6)	3.118704(-6)
0.6		5.246060(-7)	9.573462(-7)	1.594190(-6)	2.301669(-6)	2.186310(-6)	1.847047(-6)
0.7		2.128038(-7)	3.906875(-7)	6.585484(-7)	9.872307(-7)	9.687850(-7)	8.464763(-7)
0.8		5.747045(-8)	1.059901(-7)	1.803139(-7)	2.782196(-7)	2.800380(-7)	2.512726(-7)
0.9		5.684565(-9)	1.052099(-8)	1.802852(-8)	2.845958(-8)	2.923297(-8)	2.679859(-8)
1.0		0.0	0.0	0.0	0.0	0.0	0.0

Tabela 18: Resultado parcial, $M = 8$, para o problema com Anisotropia 8

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.9	6.562539(-10)	5.891430(-10)	5.281524(-10)	4.221971(-10)	1.993461(-10)	8.284730(-11)	
-0.8	9.163687(-9)	8.232631(-9)	7.386371(-9)	5.915751(-9)	2.816527(-9)	1.183247(-9)	
-0.7	4.024932(-8)	3.619081(-8)	3.250161(-8)	2.608895(-8)	1.254724(-8)	5.342905(-9)	
-0.6	1.096629(-7)	9.870338(-8)	8.874126(-8)	7.142285(-8)	3.478184(-8)	1.507037(-8)	
-0.5	2.291897(-7)	2.065228(-7)	1.859228(-7)	1.501177(-7)	7.426373(-8)	3.292924(-8)	
-0.4	4.036465(-7)	3.641992(-7)	3.283680(-7)	2.661359(-7)	1.343346(-7)	6.150900(-8)	
-0.3	6.294364(-7)	5.686933(-7)	5.135775(-7)	4.180233(-7)	2.165128(-7)	1.039268(-7)	
-0.2	8.944761(-7)	8.090461(-7)	7.316578(-7)	5.979137(-7)	3.194142(-7)	1.650526(-7)	
-0.1	1.181312(-6)	1.068890(-6)	9.671636(-7)	7.918093(-7)	4.335065(-7)	2.504651(-7)	
-0.0	1.555000(-12)	2.111100(-12)	2.232500(-12)	2.162400(-12)	1.458400(-12)	9.310000(-13)	4.128000(-13)
0.0		2.111100(-12)	2.232500(-12)	2.162400(-12)	1.458400(-12)	9.310000(-13)	4.128000(-13)
0.1		5.285934(-7)	7.988994(-7)	9.479824(-7)	6.399360(-7)	3.944025(-7)	2.397311(-7)
0.2		2.632924(-7)	4.432892(-7)	6.318032(-7)	5.970294(-7)	4.169938(-7)	2.686421(-7)
0.3		1.476375(-7)	2.585604(-7)	3.969581(-7)	4.528727(-7)	3.568388(-7)	2.521392(-7)
0.4		8.207071(-8)	1.466878(-7)	2.343384(-7)	2.989403(-7)	2.563282(-7)	1.956218(-7)
0.5		4.224968(-8)	7.645820(-8)	1.251974(-7)	1.717745(-7)	1.562798(-7)	1.263846(-7)
0.6		1.882527(-8)	3.435390(-8)	5.720655(-8)	8.259355(-8)	7.845374(-8)	6.627949(-8)
0.7		6.545609(-9)	1.201708(-8)	2.025611(-8)	3.036579(-8)	2.979835(-8)	2.603628(-8)
0.8		1.428357(-9)	2.634249(-9)	4.481459(-9)	6.914737(-9)	6.959911(-9)	6.244983(-9)
0.9		9.885690(-11)	1.829641(-10)	3.135219(-10)	4.949190(-10)	5.083671(-10)	4.660321(-10)
1.0		0.0	0.0	0.0	0.0	0.0	0.0

Tabela 19: Resultado para o problema com anisotropia 82, $\omega = 0.9$, $\mu_0 = 0.5$, $x_0 = 1$ e $N = 1000$

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	2.281907(-2)	2.141713(-2)	1.999208(-2)	1.715746(-2)	9.347227(-3)	4.025141(-3)	
-0.9	4.111263(-2)	3.862776(-2)	3.605598(-2)	3.087040(-2)	1.645421(-2)	6.810496(-3)	
-0.8	6.499816(-2)	6.129713(-2)	5.739573(-2)	4.939996(-2)	2.660773(-2)	1.102124(-2)	
-0.7	9.994450(-2)	9.474211(-2)	8.912096(-2)	7.735085(-2)	4.261315(-2)	1.793271(-2)	
-0.6	1.509935(-1)	1.440828(-1)	1.363428(-1)	1.196575(-1)	6.811692(-2)	2.950648(-2)	
-0.5	2.247679(-1)	2.162630(-1)	2.061689(-1)	1.834793(-1)	1.090676(-1)	4.931244(-2)	
-0.4	3.293348(-1)	3.201812(-1)	3.080443(-1)	2.788888(-1)	1.752098(-1)	8.410539(-2)	
-0.3	4.725347(-1)	4.655195(-1)	4.529536(-1)	4.187182(-1)	2.820817(-1)	1.472385(-1)	
-0.2	6.568330(-1)	6.583900(-1)	6.494950(-1)	6.152065(-1)	4.514659(-1)	2.661827(-1)	
-0.1	8.703204(-1)	8.945236(-1)	8.974451(-1)	8.721714(-1)	6.975140(-1)	4.915541(-1)	
-0.0	2.278249(-1)	4.203132(-1)	5.297611(-1)	6.555032(-1)	7.129739(-1)	6.163642(-1)	
0.0		4.203132(-1)	5.297611(-1)	6.555032(-1)	7.129739(-1)	6.163642(-1)	4.153662(-1)
0.1		6.075450(-1)	9.985909(-1)	1.384726(0)	1.411808(0)	1.158056(0)	8.764679(-1)
0.2		5.088813(-1)	9.072216(-1)	1.437900(0)	1.826302(0)	1.612724(0)	1.291436(0)
0.3		5.499938(-1)	9.998970(-1)	1.648253(0)	2.307865(0)	2.155872(0)	1.805491(0)
0.4		6.386149(-1)	1.169254(0)	1.957780(0)	2.873006(0)	2.778217(0)	2.401041(0)
0.5		6.342703(-1)	1.168691(0)	1.983203(0)	3.033330(0)	3.034260(0)	2.710544(0)
0.6		4.182597(-1)	7.780869(-1)	1.346197(0)	2.183952(0)	2.295057(0)	2.153686(0)
0.7		2.048831(-1)	3.858103(-1)	6.839325(-1)	1.193344(0)	1.332003(0)	1.326821(0)
0.8		8.647518(-2)	1.651498(-1)	3.010133(-1)	5.695831(-1)	6.786077(-1)	7.199338(-1)
0.9		3.136636(-2)	6.090098(-2)	1.146120(-1)	2.373355(-1)	3.032066(-1)	3.434777(-1)
1.0		5.071147(-3)	1.011917(-2)	2.004358(-2)	4.760842(-2)	6.734925(-2)	8.375799(-2)

Tabela 20: Resultado para o problema com anisotropia 82, $\omega = 0.9$, $\mu_0 = 1$, $x_0 = 1$ e $N = 1000$

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	2.796971(-2)	2.658160(-2)	2.517778(-2)	2.234066(-2)	1.375105(-2)	6.703964(-3)	
-0.9	3.018010(-2)	2.874264(-2)	2.727625(-2)	2.428233(-2)	1.503694(-2)	7.327908(-3)	
-0.8	3.144738(-2)	3.007040(-2)	2.864076(-2)	2.566176(-2)	1.609604(-2)	7.854949(-3)	
-0.7	3.438352(-2)	3.305539(-2)	3.164036(-2)	2.860568(-2)	1.831402(-2)	9.002530(-3)	
-0.6	3.913056(-2)	3.789076(-2)	3.651329(-2)	3.342763(-2)	2.209762(-2)	1.106184(-2)	
-0.5	4.563815(-2)	4.461734(-2)	4.338418(-2)	4.040337(-2)	2.800867(-2)	1.451438(-2)	
-0.4	5.351151(-2)	5.298564(-2)	5.214115(-2)	4.968634(-2)	3.685410(-2)	2.023079(-2)	
-0.3	6.154150(-2)	6.199097(-2)	6.197820(-2)	6.088591(-2)	4.961624(-2)	2.982748(-2)	
-0.2	6.695624(-2)	6.905650(-2)	7.049972(-2)	7.203732(-2)	6.669702(-2)	4.630060(-2)	
-0.1	6.552959(-2)	7.000436(-2)	7.343262(-2)	7.859061(-2)	8.446299(-2)	7.361107(-2)	
-0.0	2.603121(-2)	3.724644(-2)	4.474627(-2)	5.641092(-2)	7.839947(-2)	8.533561(-2)	
0.0		3.724644(-2)	4.474627(-2)	5.641092(-2)	7.839947(-2)	8.533561(-2)	6.985178(-2)
0.1		2.249487(-2)	3.951819(-2)	6.265315(-2)	9.625453(-2)	1.087185(-1)	1.081889(-1)
0.2		1.310062(-2)	2.534005(-2)	4.677293(-2)	9.159169(-2)	1.138147(-1)	1.242115(-1)
0.3		1.019419(-2)	2.027014(-2)	3.947193(-2)	8.746711(-2)	1.166197(-1)	1.357117(-1)
0.4		9.529076(-3)	1.906769(-2)	3.777034(-2)	8.833246(-2)	1.225027(-1)	1.482677(-1)
0.5		1.026377(-2)	2.050238(-2)	4.064932(-2)	9.641853(-2)	1.358167(-1)	1.675159(-1)
0.6		1.252928(-2)	2.490941(-2)	4.906552(-2)	1.153361(-1)	1.622302(-1)	2.007004(-1)
0.7		1.741707(-2)	3.441511(-2)	6.708099(-2)	1.539815(-1)	2.135630(-1)	2.616716(-1)
0.8		2.856216(-2)	5.602034(-2)	1.076968(-1)	2.384853(-1)	3.225492(-1)	3.869203(-1)
0.9		6.163310(-2)	1.197630(-1)	2.261237(-1)	4.761027(-1)	6.197033(-1)	7.177450(-1)
1.0		3.281233(-1)	6.290646(-1)	1.156315(0)	2.248393(0)	2.741471(0)	2.977658(0)

Tabela 21: Resultado para o problema com anisotropia 299, $\omega = 0.9$, $\mu_0 = 1$, $x_0 = 64$ e $N = 1000$

μ	$x = 0$	$x = x_0/20$	$x = x_0/10$	$x = x_0/5$	$x = x_0/2$	$x = 3x_0/4$	$x = x_0$
-1.0	2.097727(-1)	8.661258(-2)	4.134351(-2)	9.511052(-3)	1.082671(-4)	2.578581(-6)	
-0.9	1.330570(-1)	7.991605(-2)	4.164227(-2)	9.888586(-3)	1.130038(-4)	2.691950(-6)	
-0.8	1.558568(-1)	8.497950(-2)	4.375163(-2)	1.041519(-2)	1.192699(-4)	2.841720(-6)	
-0.7	1.224771(-1)	8.304484(-2)	4.531465(-2)	1.107138(-2)	1.273460(-4)	3.034597(-6)	
-0.6	1.061316(-1)	8.381669(-2)	4.770553(-2)	1.190914(-2)	1.375608(-4)	3.278425(-6)	
-0.5	1.003734(-1)	8.726272(-2)	5.110253(-2)	1.296576(-2)	1.503023(-4)	3.582457(-6)	
-0.4	9.363773(-2)	9.188200(-2)	5.540277(-2)	1.427545(-2)	1.660311(-4)	3.957683(-6)	
-0.3	8.611250(-2)	9.778759(-2)	6.073020(-2)	1.588262(-2)	1.852981(-4)	4.417252(-6)	
-0.2	7.847247(-2)	1.052146(-1)	6.725240(-2)	1.784107(-2)	2.087666(-4)	4.976984(-6)	
-0.1	6.762522(-2)	1.140416(-1)	7.512467(-2)	2.021526(-2)	2.372391(-4)	5.656025(-6)	
-0.0	3.037370(-2)	1.240097(-1)	8.455349(-2)	2.308409(-2)	2.716921(-4)	6.477668(-6)	
0.0		1.240097(-1)	8.455349(-2)	2.308409(-2)	2.716921(-4)	6.477668(-6)	7.644754(-8)
0.1		1.365976(-1)	9.587516(-2)	2.654481(-2)	3.133201(-4)	7.470402(-6)	1.364178(-7)
0.2		1.509471(-1)	1.094183(-1)	3.071801(-2)	3.635919(-4)	8.669256(-6)	1.743703(-7)
0.3		1.682651(-1)	1.257333(-1)	3.575573(-2)	4.243240(-4)	1.011754(-5)	2.152568(-7)
0.4		1.902171(-1)	1.456221(-1)	4.185340(-2)	4.977766(-4)	1.186917(-5)	2.616738(-7)
0.5		2.201906(-1)	1.704162(-1)	4.927099(-2)	5.867810(-4)	1.399160(-5)	3.157824(-7)
0.6		2.637126(-1)	2.024313(-1)	5.837253(-2)	6.949098(-4)	1.656995(-5)	3.799335(-7)
0.7		3.312696(-1)	2.460919(-1)	6.971435(-2)	8.267102(-4)	1.971236(-5)	4.569296(-7)
0.8		4.460374(-1)	3.108659(-1)	8.426871(-2)	9.880297(-4)	2.355747(-5)	5.502435(-7)
0.9		6.767158(-1)	4.226834(-1)	1.041739(-1)	1.186503(-3)	2.828467(-5)	6.642525(-7)
1.0		6.947941(1)	8.975663(0)	2.230855(-1)	1.432686(-3)	3.412862(-5)	8.046186(-7)

Se for feita a comparação entre os resultados apresentados nas Tabelas 10 até 21 e os resultados apresentados nas Tabelas 1 a 9, 13, 14 e 20 do artigo [40], pode-se notar um aumento na precisão dos dados, devido a utilização de valores de N maiores que os utilizados anteriormente.

Trabalhos publicados pelo Autor

Tiago de Souza Sotana, Cynthia Feijó Segatto, Marco Túlio Vilhena, Jacques Duílio Brancher, **Um código em paralelo para o cálculo da equação de transporte**, Revista do CCEI, 2013. (Em publicação).