

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA

LESLY VIVIANE MONTÚFAR BERRIOS

**COMPUTAÇÃO PARALELA COM OPENMP APLICADA À
PREDIÇÃO DE PERDA DE PERCURSO UTILIZANDO EQUAÇÕES
PARABÓLICAS E O MÉTODO DAS DIFERENÇAS FINITAS**

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

LESLY VIVIANE MONTÚFAR BERRIOS

**COMPUTAÇÃO PARALELA COM OPENMP APLICADA À
PREDIÇÃO DE PERDA DE PERCURSO UTILIZANDO EQUAÇÕES
PARABÓLICAS E O MÉTODO DAS DIFERENÇAS FINITAS**

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Engenharia Eletrônica e de Telecomunicações.

Orientador: Lorenço Santos Vasconcelos

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Elétrica

Bacharelado em Engenharia Eletrônica e de Telecomunicações

Uberlândia, Brasil

2023

LESLY VIVIANE MONTÚFAR BERRIOS

**COMPUTAÇÃO PARALELA COM OPENMP APLICADA À
PREDIÇÃO DE PERDA DE PERCURSO UTILIZANDO EQUAÇÕES
PARABÓLICAS E O MÉTODO DAS DIFERENÇAS FINITAS**

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Engenharia Eletrônica e de Telecomunicações.

Trabalho aprovado. Uberlândia, Brasil, 2 de fevereiro de 2023:

Lorenço Santos Vasconcelos
Orientador

Igor Santos Peretta
Convidado 1

Gilberto Arantes Carrijo
Convidado 2

Uberlândia, Brasil
2023

Dedico este trabalho a meus pais que me ensinaram muito, especialmente sobre aprender com tudo que está ao meu redor; Dedico este trabalho a minha irmã que me ajudou criativamente a superar vários obstáculos; Dedico a todas as pessoas que me apoiaram em meu crescimento pessoal e acadêmico, sendo gentis e me ensinando muito.

AGRADECIMENTOS

Agradeço a todos que me apoiaram em minhas escolhas para que eu chegasse ao início deste novo caminho de ser uma engenheira. Aos meus pais, Sonia e Hernán, que me ensinaram que posso aprender com tudo ao meu redor, que me instigaram a crescer e me proporcionaram várias oportunidades para evoluir.

Agradeço a minha irmã, Eymi, que juntas nos divertimos bastante, graças a isso os desafios da universidade foram mais leves. Agradeço a todos os parentes que mesmo longe me apoiaram bastante, em especial minha abuelita Vicentina que sempre é bondosa comigo e a meu avô Policarpio que sempre nos acolhe bem.

Agradeço aos membros do Laboratório de Acionamentos Elétricos, pois lá foi o início de todo meu aprendizado na universidade, aprendi que existem tantas coisas legais que posso aprender e as possibilidades são imensas. Um agradecimento a todos os docentes da UFU e à instituição UFU, por essa oportunidade de aprender sobre vários temas distintos, que abriram minha mente, me fazendo valorizar ainda mais os pesquisadores. Em especial, agradeço à Professora Dra. Maria das Graças Pereira por sempre ser gentil e me trazer esse impulso inicial de sempre perguntar.

Agradeço ao Professor Dr. Lorenzo Santos Vasconcelos, que possui um vasto conhecimento e suas aulas sempre foram excelentes, trazendo uma base de conhecimento teórico muito forte e desafios em que aplicamos essa base em códigos MATLAB e C++. Sua orientação e apoio acadêmico é sempre excelente.

Agradeço a Henrique Santos de Lima, por ter me apoiado desde o início. Por ter me ensinado a programar no início da universidade, por termos aprendido muito juntos e por ter me convidado a colaborar neste projeto, sou muito grata. Além de que sempre está disposto a tirar dúvidas e compartilhar conhecimento.

“Se tiver medo de fazer algo, nunca vai conseguir fazer nada”

RESUMO

O objetivo deste trabalho é otimizar o tempo de processamento dos cálculos da predição de perda de percurso, utilizando a API OpenMP para computação paralela dos excessivos cálculos iterativos da resolução do método de equações parabólicas por meio do método de diferenças finitas. A implementação fez-se buscando que os cálculos fossem passíveis de ser paralelizados, para isso utilizou-se o algoritmo do *Parallel Cyclic Reduction* para a resolução do sistema tridiagonal e utilizaram-se estratégias para se aproveitar os blocos paralelos, mantendo assim o menor número possível de laços iterativos. Logo, o tempo reduzido no cálculo da perda de percurso, comparando-se com a referência em MATLAB de [Vasconcelos \(2017\)](#), permite agilidade de obtenção dos testes para determinar a melhor posição da antena transmissora de um projeto de telecomunicações, além de que o cálculo da intensidade de campo é essencial para a regulamentação dos serviços nos padrões da Agência Nacional de Telecomunicações (ANATEL) e do Ministério das Comunicações.

Palavras-chave: Computação paralela, OpenMP, equações parabólicas, diferenças finitas, terreno irregular.

ABSTRACT

The objective of this work is to optimize the processing time of the calculations of the path loss prediction, using the OpenMP API for parallel computation of the excessive iterative calculations of the resolution of the parabolic equations method through the finite difference method. The implementation was done so that the calculations could be parallelized, for this the algorithm of the *Parallel Cyclic Reduction* was used to solve the tridiagonal system and strategies were used to take advantage of the parallel blocks, thus maintaining the smallest possible number of iterative loops. Thus, the reduced time in the path loss calculation, compared to the MATLAB reference of [Vasconcelos \(2017\)](#), allows agility in obtaining the tests to determine the best position of the transmitting antenna of a telecommunications project, besides that the field strength calculation is essential for the regulation of services in the standards of the National Telecommunications Agency (ANATEL) and the Ministry of Communications.

Palavras-chave: Parallel computing, OpenMP, parabolic equations, finite differences, irregular terrain.

LISTA DE ILUSTRAÇÕES

Figura 1 – Onda eletromagnética	31
Figura 2 – Fronteira entre dois meios	34
Figura 3 – Propagação paraxial na troposfera	39
Figura 4 – Esquema de utilização do método PE	39
Figura 5 – Ângulo de propagação α	45
Figura 6 – Exemplificação da camada absorvedora	47
Figura 7 – Transformação para Terra plana	47
Figura 8 – Ilustração dos sistemas de coordenadas (x, y) mapeado e (v, w) não mapeado	49
Figura 9 – Ilustração do deslocamento de fase da frente de onda. Aqui, u_1^m, u_0^m , e u_{-1}^m referem-se a $u(x_m, \Delta z)$, $u(x_m, 0)$, e $u(x_m, -\Delta z)$, respectivamente, em que Δz é o passo de altura	50
Figura 10 – Quantidade de transistores em um chip	55
Figura 11 – A parede de potência	56
Figura 12 – Concorrência e paralelismo em processo de três <i>threads</i>	57
Figura 13 – Exemplo de três <i>threads</i> rodando simultaneamente em três diferentes núcleos	58
Figura 14 – Modelo de execução do OpenMP: <i>Fork-join</i>	59
Figura 15 – Distribuição do trabalho entre as <i>threads</i>	62
Figura 16 – Perfis de refração	66
Figura 17 – Algoritmo de redução cíclica	68
Figura 18 – Campo $u(x, z)$ em terreno sem relevo: resultados em MATLAB (referência) e em OpenMP, e o erro absoluto entre eles, sendo 10^{-16} a ordem de ponto flutuante	71
Figura 19 – Campo $u(x, z)$ em terreno com relevo: resultados em MATLAB (referência) e em OpenMP, e o erro absoluto entre eles, sendo 10^{-16} a ordem de ponto flutuante	72
Figura 20 – Campo $u(x, z)$ na fronteira inferior (solo): resultados em MATLAB (referência) e em OpenMP, e o erro absoluto entre eles, sendo 10^{-16} a ordem de ponto flutuante	73
Figura 21 – Campo $u(x, z)$ na fronteira superior (troposférica): resultados em MATLAB (referência) e em OpenMP, e o erro absoluto entre eles, sendo 10^{-16} a ordem de ponto flutuante	74
Figura 22 – Campo $u(x, z)$ para as condições especificadas pelo usuário	76
Figura 23 – Perda de percurso L_P para as condições especificadas pelo usuário	76
Figura 24 – Fator de propagação FP para as condições especificadas pelo usuário	77

Figura 25 – Perda de percurso L_P na máxima distância pedida pelo usuário	77
Figura 26 – Campo $u(x, z)$ para as condições especificadas pelo usuário	78
Figura 27 – Perda de percurso L_P para as condições especificadas pelo usuário . . .	78
Figura 28 – Fator de propagação FP para as condições especificadas pelo usuário .	79
Figura 29 – Perda de percurso L_P na máxima distância pedida pelo usuário	79

LISTA DE TABELAS

Tabela 1 – Grandezas Fundamentais do Campo Eletromagnético	24
Tabela 2 – Condições de contorno para os campos elétrico e magnético	34
Tabela 3 – Resultados do teste: sem relevo	71
Tabela 4 – Resultados do teste: com relevo	72
Tabela 5 – Resultados do teste: fronteiras	74
Tabela 6 – Resultados do teste: longas distâncias, altas frequências	75

LISTA DE ABREVIATURAS E SIGLAS

ANATEL	Agência Nacional de Telecomunicações
API	<i>Application Programming Interface</i> , ou Interface de Programação de Aplicação
OpenMP	<i>Open Multi-Processing</i> , ou Multi-processamento aberto
CUDA	<i>Compute Unified Device Architecture</i> , ou Arquitetura de Dispositivo de Computação Unificada
OpenCL	<i>Open Computing Language</i> , ou Linguagem de Computação Aberta
CPU	<i>Central Processing Unit</i> , ou Unidade de Processamento Central
GPU	<i>Graphics Processing Unit</i> , ou Unidade de Processamento Gráfico
RAM	<i>Random Access Memory</i> , ou Memória de Acesso Aleatório
PE	<i>Parabolic Equation</i> , ou Equações Parabólicas
WAPE	<i>Wide Angle Parabolic Equation</i> , ou Equações Parabólicas de Ângulo Amplo
PCR	<i>Parallel Cyclic Reduction</i> , ou Redução Cíclica Paralela
omp	OpenMP

LISTA DE SÍMBOLOS

ρ	Densidade volumétrica de carga, em C/m ³ .
ρ_s	Densidade superficial de carga, em C/m ² .
ρ_l	Densidade linear de carga, em C/m.
\mathbf{F}	Intensidade de força, em C/s ou A.
\vec{E}	Intensidade de campo elétrico, em V/m.
\vec{D}	Densidade de fluxo elétrico, em C/m ² .
\vec{H}	Intensidade de fluxo magnético, em A/m.
\vec{B}	Densidade de fluxo magnético, em T.
c	Velocidade da onda eletromagnética, igual a 299792458 m/s.
ϵ_0	Permissividade elétrica no espaço livre, aproximadamente 8.854187817620E-12 F/m.
μ_0	Permeabilidade magnética no espaço livre, igual a $4\pi \times 10^{-7}$ H/m.
ω	Frequência angular, em rad/s.
f	Frequência da onda eletromagnética, em Hz.
λ	Comprimento de onda, em m.
σ	Condutividade elétrica, em S/m.
ϵ	Permissividade elétrica absoluta do material, em F/m.
ϵ_r	Permissividade elétrica relativa do material, em F/m.
μ	Permeabilidade magnética, em H/m.
Q	Operador pseudodiferencial
a	Raio médio da Terra, igual a 6371008 m.

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Introdução	16
1.2	Motivação	17
1.3	Objetivos	18
1.4	Revisão Bibliográfica	19
1.5	Estrutura deste Trabalho	20
2	FUNDAMENTOS DO ELETROMAGNETISMO	22
2.1	Introdução	22
2.2	Fontes do campo eletromagnético	22
2.3	Campo elétrico	24
2.4	Campo magnético	26
2.5	Equações de Maxwell	27
2.5.1	Equações de Maxwell na forma pontual	28
2.5.2	Equações de Maxwell na forma integral	29
2.6	Ondas eletromagnéticas	29
2.7	Campos harmônicos no tempo e notação fasorial	30
2.7.1	Equações de Maxwell na forma fasorial	32
2.8	Condições de contorno	33
2.9	Equação da onda e a onda plana uniforme	35
3	MÉTODO DE EQUAÇÕES PARABÓLICAS	38
3.1	Introdução	38
3.2	Formulação do método	38
3.3	Aproximação parabólica	40
3.4	Equação de Claerbout	43
3.5	Condições de contorno para a WAPE	44
3.5.1	Fronteira inferior (solo)	44
3.5.2	Fronteira superior (troposférica)	46
3.6	Transformação para a Terra Plana	46
3.7	Transformação para terrenos irregulares	48
3.7.1	WAPE para terreno irregular	48
3.7.2	Condição de contorno da WAPE para terreno irregular	49
3.8	Esquema de resolução por diferença finitas	51
3.8.1	WAPE para terreno plano	51
3.8.2	WAPE para terreno irregular	52

3.9	Perda de percurso e fator de propagação	53
4	COMPUTAÇÃO PARALELA	55
4.1	Introdução	55
4.2	Processamento paralelo	56
4.3	OpenMP	58
4.4	Principais diretivas do OpenMP	60
5	METODOLOGIA PROPOSTA	63
5.1	Introdução	63
5.2	Metodologia de implementação no código	63
5.3	Passo de distância e altura <i>default</i>	65
5.4	Preparar o terreno	65
5.5	Perfis de refração	65
5.6	Propagar	66
5.6.1	Multiplicação da matriz tridiagonal	66
5.6.2	Algoritmo de Thomas	67
5.6.3	Redução Cíclica Paralela	67
6	RESULTADOS	69
6.1	Introdução	69
6.2	Teste 1: Sem relevo	70
6.3	Teste 2: Com relevo	72
6.4	Teste 3: Fronteiras	73
6.5	Teste 4: Longas distâncias e altas frequências	75
6.5.1	Resultados para terreno sem relevo	76
6.5.2	Resultados para terreno com relevo	78
7	CONCLUSÕES	80
7.1	Conclusões	80
7.2	Trabalhos futuros	80
	REFERÊNCIAS	82
	APÊNDICES	85
	APÊNDICE A – CÓDIGOS	86
A.1	Arquivo <i>parametros.json</i>	86
A.2	Cálculo do tamanho da camada absorvedora	88
A.3	Determinando o valor máximo de um veto de alta ordem	88

A.4	Interpolação	89
A.5	Passo de distância e altura <i>default</i>	89
A.6	Derivada por diferença central	90
A.7	Multiplicação da matriz tridiagonal	90
A.8	Algoritmo de Thomas	91
A.9	Algoritmo da PCR	91

1 INTRODUÇÃO

1.1 Introdução

Muitos dos sistemas de telecomunicações atuais envolvem enlaces de rádio. Alguns exemplos são: sistemas de difusão de rádio e televisão, a interface aérea de sistemas celulares, enlaces de micro-ondas, entre outros. Cada sistema possui sua peculiaridade no raio de cobertura, modulação e polarização. Em um sistema de radiodifusão de difusão, por exemplo, os receptores mais distantes da estação base estão, geralmente, a 20 ou 30 km dela. Enquanto que em um sistema celular, os raios de cobertura das células chegam a ser menores que 1 km.

Apesar dessas diferentes peculiaridades entre os sistemas, eles também possuem semelhanças: suas ondas eletromagnéticas propagam sobre uma superfície irregular e pela atmosfera terrestre, sujeitas às condições climáticas, às características atmosféricas e do terreno. Assim, a confiabilidade do enlace de comunicação pode ser significativamente afetada devido aos fatores do ambiente de propagação. Por isso, nas aplicações dos sistemas de comunicações, deve-se realizar estudos sobre a área de cobertura dos sistemas irradiantes e isso requer o cálculo de perda de percurso.

A perda de percurso de um sistema de comunicação terrestre pode sofrer interferência de dois tipos principais de desvanecimento por multipercurso: atmosférico e de solo. O multipercurso atmosférico é causado pela interferência de dois ou mais raios supra-refratados que chegam ao receptor por diferentes caminhos e pode conduzir a uma perda completa do sinal no receptor. Além disso, condições anômalas de propagação causadas por variações troposféricas podem fazer com que o horizonte de rádio seja modificado, favorecendo ou comprometendo a comunicação. Portanto, em geral, o comportamento desse tipo de multipercurso depende da refratividade da atmosfera ([VASCONCELOS, 2017](#)).

Por outro lado, o multipercurso de solo é devido à interferência entre os raios direto e refletido pelo solo e seu comportamento depende fortemente da geometria do terreno e das características do solo. Além disso, a propagação de ondas de rádio em terrenos irregulares pode encontrar uma variedade de obstáculos que bloqueiam a visada direta entre os terminais de transmissão e recepção ou que obstruem a primeira zona de Fresnel do enlace. Isso pode sombrear completamente o receptor, levando a perdas severas do sinal ([JANASWAMY, 1994](#)).

Existem diferentes métodos para cálculo de perda de percurso, cada um considerando diferentes fatores e com complexidades distintas. Um dos métodos mais generalistas e robustos, principalmente para aplicações de longa distância, é o método de equações

parabólicas, porém sua implementação comum exige alto tempo de processamento e uso de memória, podendo levar cerca 40 min para fornecer resultados de cenários relativamente pequenos. Apesar de existirem alguns *softwares* disponíveis, hoje, no mercado que oferecem soluções mais rápidas, eles utilizam métodos mais simples, com resultados aproximados, e às vezes não condizentes com a realidade dependendo das condições do ambiente (quantidade de obstáculos, frequência do sinal).

Por isso, buscando manter a precisão do robusto método de equações parabólicas, porém contornando o problema de alto de tempo de processamento e uso de memória, iniciou-se um trabalho em grupo com o objetivo de utilizar computação paralela para manter a precisão desse método, gerenciar as *threads* da Unidade de Processamento Central (CPU) e Unidades de Processamento Gráfico (GPU), de forma a otimizar o tempo de cálculo dessa solução e evitar aproximações, e, dessa forma, tornar a solução viável comercialmente. Assim, o trabalho em grupo consistiu em, baseando-se na tese de doutorado de Vasconcelos (2017), escrever um programa em C++ que implemente e resolva o método de equações parabólicas, usando o método de diferenças finitas, e otimizando os cálculos por meio de computação paralela. Para isso, foram utilizadas três *Application Programming Interface* (API) diferentes: OpenMP, OpenCL e CUDA. Assim, seria possível avaliar o desempenho de cada uma mediante da comparação entre si.

Cada API possui sua vantagem dependendo da máquina que o usuário possui. OpenMP é especialmente indicado para processadores *multicore*, é adequado para uma ampla gama de algoritmos executados em nós de múltiplos núcleos e chips, sistemas NUMA, GPUs, e outros dispositivos ligados a uma CPU (OPENMP.ORG, 2021). OpenCL e CUDA são amplamente utilizadas para colher o poder computacional das placas de vídeo, em particular das GPUs. Existindo a condição de que códigos em CUDA são executados somente em hardwares da NVIDIA e devem ser totalmente reescritos para que possam ser executados em GPUs da AMD ou Intel. Já o modelo de programação OpenCL, do Kronos Group, suporta multi-plataforma, e programação paralela em sistemas de processamento heterogêneos (EMBEDDED, 2014).

Entre as APIs, este trabalho em específico trata-se da avaliação de desempenho do OpenMP. Sendo assim, neste trabalho utiliza-se a API OpenMP para a implementação do método de equações parabólicas por diferenças finitas, a fim de permitir a execução dos cálculos em tempo otimizado e aproveitar ao máximo os recursos CPU da máquina.

1.2 Motivação

A predição e o cálculo de intensidade de campo elétrico (ou da perda de propagação) são imprescindíveis na instalação e manutenção de qualquer sistema de telecomunicações sem fio. O cálculo da atenuação é muito importante no projeto de tais sistemas, pois,

a partir das predições de intensidade de campo, pode-se determinar (ou predizer) várias características do seu desempenho, como a área de cobertura, os contornos interferentes, áreas de sombra, entre outros. Pode-se, ainda, fazer estudos e simulações da intensidade do sinal em uma dada área, alterando a posição do transmissor, a fim de estimar o melhor local para instalar uma estação base (VASCONCELOS, 2017).

Assim, a agilidade de obtenção dos testes para determinar a melhor posição da antena transmissora é essencial para a implementação rápida de um projeto e sua comercialização. Com esse fim, pode ser investido em *hardwares* potentes e caros, e realizar o cálculo com uma frequência de *clock* maior. No entanto, não somente a partir da utilização de dispositivos de *hardware* mais rápidos obtém-se o alto desempenho, mas também por meio de melhorias na arquitetura dos computadores e técnicas de processamento. Por isso, utilizando o processamento paralelo, por meio da arquitetura de *pipeline*, tem-se a redução do tempo de processamento ao executar cálculos simultâneos, permitindo a utilização de métodos robustos e bastantes iterativos em menos tempo e evitando aproximações.

A nível de *hardware*, a OpenMP é mais generalista, uma vez que é compatível com grande parte dos processadores disponíveis no mercado, além de também poder ser utilizada com a GPU, como apresentado por Klemm (2021). OpenCL também é portátil, porém o código é compilado e construído (*built*) em tempo de execução, o que significa que o enfileiramento de *kernels* é feito em tempo de execução, aumentando o tempo de processamento. CUDA e OpenMP não tem esse problema, mas, por outro lado, o desempenho do CUDA exige a execução em hardware NVIDIA que possua os *CUDA Cores*.

Vale ressaltar que o cálculo de intensidade de campo é necessário para a regulamentação dos serviços nos padrões da Agência Nacional de Telecomunicações (ANATEL) e do Ministério das Comunicações. Por isso, a motivação principal deste trabalho é otimizar o tempo de processamento da resolução do método de equações parabólicas utilizando uma API para processamento paralelo de maior disponibilidade de compatibilidade nos dispositivos comerciais: a OpenMP. Espera-se, assim, que usando técnicas de paralelismo seja possível aumentar a agilidade dos cálculos de intensidade de campo (ou predição de perda de percurso) sem comprometer a precisão do resultado final.

1.3 Objetivos

Com o objetivo de se vislumbrar ganhos em termos de agilidade e precisão no cálculo da predição de perda de percurso, deve-se atingir alguns objetivos específicos. São eles:

- Entendimento dos aspectos físicos e matemáticos que regem a propagação de ondas;

- Entendimento de aproximações utilizadas para a equação da onda e estabelecimento de condições de fronteira inferior e superior, para que seja viável numericamente e sejam incorporadas corretamente a refratividade atmosférica, a forma e as constantes elétrica do terreno;
- Entendimento do paralelismo executado pelas diretivas da OpenMP;
- Implementação computacional em C++ da referência disponibilizada por [Vasconcelos \(2017\)](#) (escrita em MATLAB), para a propagação do sinal em terreno com e sem relevo;
- Validação dos resultados numéricos obtidos pelo programa C++, por meio da comparação com os resultados da referência em MATLAB;
- Inserção de diretivas OpenMP para processar as tarefas iterativas paralelamente;
- Criação de estratégias e adaptação do código para tornar alguns cálculos passíveis de ser executados paralelamente;
- Testes e simulações comparativas para avaliação de desempenho da implementação em OpenMP;
- Criação e implementação de estratégias de redução do uso de memória RAM tomando em conta a saída desejada pelo usuário.

1.4 Revisão Bibliográfica

Tratando-se da busca por redução do custo computacional e tempo de processamento para analisar a propagação troposférica sobre terrenos irregulares, [Vasconcelos \(2017\)](#) propôs um novo modelo híbrido para predição de perda de percurso, em que aproveita a parte de menor custo do modelo de equações parabólicas e a incorpora a um modelo simples de difração por gume de faca, o modelo de Bullington.

[Ramalho \(2006\)](#) usa *Visual Basic for Application* (VBA) e o ArcView, que é um sistema de software para manipulação de informações geográficas GIS (*Geographic Information System*) do ambiente sob análise, para desenvolver o Preditor de Rádio Frequências do Inatel (PraFin), um sistema didático para apresentação de conceitos em intuições de ensino e alternativo em vista os elevados custos das ferramentas de predição de cobertura comerciais. Ademais, sobre ferramentas comerciais, pode-se citar a *CellPlanner* da empresa [CelPlan \(1992\)](#).

Já [Dias, Napolitano e Silveira \(2018\)](#), no contexto das comunicações táticas terrestres em V/UHF, apresentam uma ferramenta de predição de cobertura desenvolvida

como prova de conceito no escopo do futuro Planejador de Missões do RDS-Defesa. Com a crescente capacidade computacional disponível nos equipamentos de comunicações táticas, afirmaram ser pertinente se pensar em portar tal funcionalidade para um módulo compacto dedicado, ou mesmo para o próprio rádio, se definido por *software*.

1.5 Estrutura deste Trabalho

A estrutura deste trabalho foi dividida em oito capítulos, com os respectivos títulos:

1. Introdução (este capítulo);
2. Fundamentos do Eletromagnetismo;
3. Método de Equações Parabólicas;
4. Computação Paralela;
5. Metodologia Proposta;
6. Resultados;
7. Conclusões.

A Introdução do Capítulo 1 tem por objetivo delimitar e apresentar os objetos de estudo deste trabalho, bem como justificar e dar relevância para as questões pesquisadas. Além disso, ela ambienta o leitor à estrutura observada nos capítulos seguintes.

O Capítulo 2, Fundamentos do Eletromagnetismo, estabelece o entendimento da base da aplicação tratada neste trabalho, as ondas eletromagnéticas. Por isso, é citado e explicado o trabalho de Maxwell que descreve a onda eletromagnética como uma perturbação do campo elétrico e magnético. Em seguida, apresentada a equação da onda para ondas planas cujo campo eletromagnético é harmônico no tempo.

O Capítulo 3, Método de Equações Parabólicas, apresenta a aproximação parabólica, aliada à de Claerbout, como alternativa para a solução exata descrita por Maxwell, já que a adaptação é necessária para a computação de soluções numéricas. Feita a aproximação, que não apresenta erros significantes para a aplicação, é apresentado o esquema responsável por resolver a equação resultante, mediante o método de diferenças finitas para discretização do problema.

O Capítulo 4, Computação Paralela, explica como surgiu a necessidade de se otimizar a computação mediante o processamento paralelo ao invés de continuar com as pesquisas de aumento da frequência do *clock* como única forma de ganhar desempenho.

Apresenta a API OpenMP, cita suas vantagens, modelo de execução e exemplifica sua utilização em códigos interativos.

O Capítulo 5, Metodologia Proposta, mostra os passos executados pelo código em OpenMP para receber os parâmetros de entrada, paralelizar os cálculos iterativos e gerar uma saída de dados em tempo de processamento otimizado. Para isso, é explicado sobre algumas estratégias e algoritmos utilizados para reduzir o custo computacional.

O Capítulo 6, Resultados, mostra os resultados de estabilidade, comparando a solução OpenMP com a MATLAB, e de tempo, mostrando a eficiência da metodologia utilizada. Além disso, são apresentadas as figuras de perda de percurso para um caso próximo real, que é o objetivo deste trabalho.

Por fim, o Capítulo 7, Conclusões, são apresentadas as vantagens do OpenMP para esta aplicação, tendo em vista os resultados obtidos nos testes anteriormente executados. Além disso, são apontadas possíveis melhorias e trabalhos que dariam continuidade ao ganho de eficiência no tempo de processamento.

2 FUNDAMENTOS DO ELETROMAGNETISMO

2.1 Introdução

Para compreender o problema, é necessário compreender a propagação de ondas eletromagnéticas: como são formadas, emitidas por uma antena transmissora, para assim serem captadas por uma antena receptora. Dessa forma, é necessário o entendimento das ondas eletromagnéticas, que como o nome diz, surgem da interação entre dois campos: o elétrico e o magnético. Assim, neste capítulo, são apresentados conceitos essenciais do eletromagnetismo, que é o estudo dos efeitos das cargas elétricas em repouso ou em movimento.

2.2 Fontes do campo eletromagnético

Existem dois tipos de cargas: positiva (próton) e negativa (elétron), e ambas podem ser fontes de um campo eletromagnético. Além disso, deve ser obedecido o princípio da conservação da carga elétrica em qualquer circunstância, assim dizendo, as cargas elétricas podem se mover de um lugar a outro e podem ser redistribuídas conforme interagem com o ambiente (outras cargas) sobre a influência de um campo eletromagnético, mas algebricamente a soma das cargas positivas e negativas em um sistema fechado (isolado) sempre se mantém.

Além disso, uma propriedade importante é a densidade de volume de carga, ρ , definida como:

$$\rho = \lim_{\Delta v \rightarrow 0} \frac{\Delta q}{\Delta v} \quad (2.1)$$

em que:

- ρ é a densidade do volume de carga, em C/m³.
- Δv é um volume muito pequeno, em m³.
- Δq é a quantidade de carga contida em Δv , em C.

Em algumas situações físicas, a quantidade de carga Δq pode ser identificada como um elemento de superfície Δs ou um elemento de linha Δl , nesses casos é apropriado definir uma densidade de superfície de carga, ρ_s , ou uma densidade de linha de carga ρ_l :

$$\rho_s = \lim_{\Delta s \rightarrow 0} \frac{\Delta q}{\Delta s} \quad (C/m^2) \quad (2.2)$$

$$\rho_l = \lim_{\Delta l \rightarrow 0} \frac{\Delta q}{\Delta l} \quad (C/m) \quad (2.3)$$

Por último, define-se a corrente elétrica, I , a qual é a variação de carga no tempo:

$$I = \frac{dq}{dt} \quad (C/s \text{ ou } A) \quad (2.4)$$

Existem quatro campos vetoriais fundamentais no eletromagnetismo, mostrados na Tabela 1. Adicionados a eles, existem três constantes universais no modelo eletromagnético mostrado, que se referem às propriedades do espaço livre (vácuo). Elas são: velocidade da onda eletromagnética (incluindo a luz) no espaço livre, c ; permissividade elétrica do espaço livre, ϵ_0 ; e permeabilidade magnética do espaço livre, μ_0 . As Equações (2.5) a (2.7) definem os valores das constantes mencionadas.

$$c = 299792458 \quad (\text{m/s}) \quad (\text{no espaço livre}) \quad (2.5)$$

$$\epsilon_0 \approx 8,85418781762 \times 10^{-12} \quad (\text{F/m}) \quad (\text{no espaço livre}) \quad (2.6)$$

$$\mu_0 = 4\pi \times 10^{-7} \quad (\text{H/m}) \quad (\text{no espaço livre}) \quad (2.7)$$

A permissividade elétrica ϵ_0 pertence ao fenômeno elétrico, é a constante de proporcionalidade entre a densidade de fluxo elétrico \mathbf{D} e a intensidade do campo elétrico \mathbf{E} no espaço livre. Já μ_0 pertence ao fenômeno magnético, e é a constante de proporcionalidade entre a densidade de fluxo magnético \mathbf{B} e a intensidade do campo magnético \mathbf{H} no espaço livre, como mostrado nas Equações (2.8) e (2.9).

$$\vec{D} = \epsilon_0 \vec{E} \quad (\text{no espaço livre}) \quad (2.8)$$

$$\vec{H} = \frac{1}{\mu_0} \vec{B} \quad (\text{no espaço livre}) \quad (2.9)$$

Tabela 1 – Grandezas Fundamentais do Campo Eletromagnético

Símbolos e Unidades do Campo	Grandeza de campo	Símbolo	Unidade
Elétrico	Intensidade de campo elétrico	\vec{E}	V/m
	Densidade de fluxo elétrico	\vec{D}	C/m ²
Magnético	Intensidade de campo magnético	\vec{H}	A/m
	Densidade de fluxo magnético	\vec{B}	T

2.3 Campo elétrico

Um campo é uma distribuição espacial de uma quantidade escalar ou vetorial, que pode ou não estar em função do tempo. No campo gravitacional da Terra, por exemplo, ao representar a força da gravidade sobre uma unidade de massa, é ter um campo vetorial direcionado ao centro da Terra, tendo sua magnitude dependente da altitude dessa unidade de massa. Intensidades de campos elétricos e magnéticos são campos vetoriais (CHENG, 1989).

Considerando uma carga elétrica estacionária no espaço livre, tem-se que a intensidade de seu campo elétrico é definida como a força por unidade de carga que uma pequena “carga de teste” estacionária experimenta quando é colocada em uma região em que o campo elétrico existe. Isto é matematicamente mostrado na Equação (2.10).

$$E = \lim_{q \rightarrow 0} \frac{F}{q} \quad (2.10)$$

em que:

- \mathbf{E} é a intensidade do campo elétrico, em V/m.

- \mathbf{F} é a intensidade da força aplicada à “carga de teste”, em N.
- q é a “carga de teste”¹, em C.

Existem dois postulados da eletroestática no espaço livre, que são definidos nas Equações (2.11) e (2.12). A Equação (2.12) afirma que campos elétricos estáticos são irrotacionais, enquanto a Equação (2.11) implica que um campo elétrico estático não é solenoidal a menos que $\rho = 0$. Esses dois postulados são concisos, simples, e independentes de qualquer sistema de coordenadas, e podem ser usados para derivar todas as outras relações, leis e teoremas na eletroestática (CHENG, 1989).

$$\vec{\nabla} \cdot \vec{E} = \frac{\rho}{\epsilon_0} \quad (2.11)$$

$$\vec{\nabla} \times \vec{E} = 0 \quad (2.12)$$

Assim, usando o teorema da divergência, da Equação (2.11) tem-se a Equação (2.14). Essa equação é uma forma da Lei de Gauss, que diz que o fluxo do campo \vec{E} que passa em qualquer superfície fechada no espaço livre é igual à carga total envolvida nessa superfície dividida pela constante ϵ_0 .

$$\int_V \vec{\nabla} \cdot \vec{E} dv = \frac{1}{\epsilon_0} \int_V \rho dv \quad (2.13)$$

$$\oint_S \vec{E} \cdot d\vec{s} = \frac{q}{\epsilon_0} \quad (2.14)$$

Exemplificando, considere uma carga pontual no centro de uma superfície esférica S de raio r . A Equação (2.15) descreve seu campo. Assim, o fluxo elétrico que atravessa essa esfera é obtido integrando o campo \vec{E} , como mostrado na Equação (2.16). Além disso, como o fluxo independe da superfície, mostrado em (2.17), pode-se perceber também que se uma carga q está no exterior da superfície fechada, suas linhas de campo elétrico entram e saem da superfície ao mesmo tempo, assim, o campo dessa superfície é nulo.

$$\vec{E} = \frac{q}{4\pi\epsilon_0 r^2} \quad (2.15)$$

¹ A “carga de teste” não pode ser zero na prática, pois não pode ser inferior à carga de um elétron ($e^- = 1,6 \cdot 10^{-19}$ C).

$$\oint_S \vec{E} \cdot d\vec{A} = \phi_E \quad (2.16)$$

$$\phi_E = \oint_S \vec{E} \cdot d\vec{A} = \vec{E} \oint_S d\vec{A} = \frac{q}{4\pi\epsilon_0 r^2} 4\pi r^2 = \frac{q}{\epsilon_0} \quad (2.17)$$

2.4 Campo magnético

Já foi visto que as Equações (2.11) e (2.12) referem-se aos postulados fundamentais da eletrostática, sendo que (2.11) pode ser reescrito como (2.18). Isso porque, no meio material é conveniente definir o deslocamento elétrico, denotado por \vec{D} , que representa a carga por unidade de área que seria deslocada através de uma camada de condutor colocada sobre um campo elétrico. Essa grandeza, também conhecida como densidade do fluxo elétrico, é prontamente calculada pela relação $\vec{D} = \epsilon\vec{E}$, para quando o meio é linear e isotrópico², em que ϵ é a permissividade elétrica do material.

Somando-se às duas equações fundamentais do modelo eletrostático, mostradas das Equações (2.18) e (2.19), tem-se então os postulados fundamentais da magnetostática no espaço livre, definidas das Equações (2.20) e (2.21).

$$\vec{\nabla} \cdot \vec{D} = \rho \quad (2.18)$$

$$\vec{\nabla} \times \vec{E} = 0 \quad (2.19)$$

$$\vec{\nabla} \cdot \vec{B} = 0 \quad (2.20)$$

$$\vec{\nabla} \times \vec{B} = \mu_0 \vec{J} \quad (2.21)$$

em que \vec{J} é a densidade de corrente em A/m².

As linhas de um campo magnético diferem das linhas de campo elétrico \vec{E} , pois as linhas de um campo magnético \vec{B} formam curvas fechadas, enquanto as linhas de um campo elétrico começam e terminam em cargas elétricas positivas ou negativas. Por isso, o postulado da Equação (2.20), afirma que fluxo magnético em uma superfície fechada é sempre zero, uma vez que uma superfície fechada não pode envolver um “monopolo magnético”. Logo, não há cargas ou “monopolos magnéticos” análogos às cargas elétricas. Em vez disso, o campo magnético é gerado por uma configuração chamada dipolo, que são

² meio em que as propriedades de interesse independem da direção

mais bem representadas como correntes fechadas, mas que lembram cargas magnéticas positivas e negativas inseparáveis.

Integrando a Equação (2.21) sobre uma superfície aberta e aplicando o teorema de Stokes tem-se a Equação (2.22). Essa equação é uma das formas da lei circuital de Ampère, que afirma que a circulação da densidade de fluxo magnético no espaço livre ao redor de qualquer caminho é igual a μ_0 vezes a corrente total fluindo pela superfície definida por esse caminho fechado.

$$\oint_C \vec{B} \cdot d\vec{l} = \mu_0 I \quad (2.22)$$

2.5 Equações de Maxwell

Os fenômenos magnéticos e elétricos eram tratados isoladamente pelas leis de Ampère e Gauss. Entretanto, quando James Clerck Maxwell (1831-1879) percebeu a simetria da variação do fluxo de campo elétrico produzindo um campo magnético induzido, ele manipulou as equações de Ampère, Gauss, Faraday e Lenz, e, assim, sintetizou as quatro equações do eletromagnetismo que, em sua homenagem, são chamadas de “equações de Maxwell”. Ao fazer isso, Maxwell unificou conceitos, antes distintos, e elucidou o mistério sobre a natureza da luz.

Assim, as equações são:

- **1ª Equação de Maxwell (Lei de Gauss para a eletricidade):** Relaciona o fluxo elétrico através de qualquer superfície gaussiana³ fechada com a densidade de cargas elétricas contidas no interior do volume encerrado pela superfície gaussiana. Matematicamente, na Equação (2.25).
- **2ª Equação de Maxwell (Lei de Gauss para o magnetismo):** Afirma que não há cargas ou “monopolos magnéticos” análogos às cargas elétricas. Assim, qualquer linha de campo magnético que entra em um determinado volume ou material deve de alguma forma sair deste volume ou material. Em uma linguagem mais técnica, o fluxo magnético através de qualquer superfície gaussiana é zero, ou pode-se dizer que o campo magnético é um campo vetorial solenoidal⁴. Matematicamente, em (2.27).

³ são superfícies tridimensionais fechadas, arbitrárias e imaginárias

⁴ um campo vetorial com divergência zero: $\vec{\nabla} \cdot \vec{v} = 0$ e dado pelo rotacional de um outro campo vetorial, isto é $\vec{v} = \vec{\nabla} \times \vec{\psi}$ (CARMO, 2008).

- **3ª Equação de Maxwell (Lei de Ampère-Maxwell):** O campo elétrico variando no tempo gera um campo magnético. Matematicamente, em (2.23).
- **4ª Equação de Maxwell (Lei de Faraday):** O campo magnético variando no tempo gera um campo elétrico. Matematicamente, em (2.24).

2.5.1 Equações de Maxwell na forma pontual

Assim, as Equações (2.23) a (2.27) são as formas pontuais das Equações de Maxwell. Sendo que a densidade de corrente magnética, m , é apenas uma conveniência matemática, incluída nas equações para que sejam matematicamente completas. Isso porque a real fonte de corrente magnética é sempre um caminho fechado de elétrica ou algum tipo de dipolo magnético, não um fluxo de cargas (monopolos) magnéticas (que não existem).

$$\vec{\nabla} \times \vec{e} = -\frac{\partial \vec{b}}{\partial t} - \vec{m} \quad (2.23)$$

$$\vec{\nabla} \times \vec{h} = \frac{\partial \vec{d}}{\partial t} + \vec{j}_{total} \quad (2.24)$$

$$\vec{\nabla} \cdot \vec{d} = \rho \quad (2.25)$$

$$\vec{\nabla} \cdot \vec{b} = \rho_m = 0 \quad (2.26)$$

$$\vec{\nabla} \cdot \vec{j}_{total} = -\frac{\partial \rho}{\partial t} \quad (2.27)$$

em que:

- $\vec{e}(x, y, z, t)$ é o campo elétrico, em V/m.
- $\vec{h}(x, y, z, t)$ é o campo magnético, em A/m.
- $\vec{d}(x, y, z, t)$ é a densidade de fluxo elétrico, em C/m².
- $\vec{b}(x, y, z, t)$ é a densidade de fluxo magnético, em Wb/m² ou T.
- $\vec{m}(x, y, z, t)$ é a densidade de corrente magnética fictícia, em V/m².
- $\vec{j}_{total}(x, y, z, t)$ é a densidade de elétrica total, em A/m².
- $\rho(x, y, z, t)$ é a densidade de carga, em C/m³.
- $\rho_m(x, y, z, t)$ é a densidade de carga magnética fictícia, em Wb/m³.

2.5.2 Equações de Maxwell na forma integral

Além disso, pode se representá-las na forma integral como nas Equações (2.28) a (2.32).

$$\oint \vec{e} \cdot d\vec{l} = - \int_S \frac{\partial \vec{b}}{\partial t} \cdot d\vec{S} - i_m \quad (2.28)$$

$$\oint \vec{h} \cdot d\vec{l} = \int_S \frac{\partial \vec{d}}{\partial t} \cdot d\vec{S} + i_{total} \quad (2.29)$$

$$\oint_S \vec{d} \cdot d\vec{S} = \int_V \rho dV \quad (2.30)$$

$$\oint_S \vec{b} \cdot d\vec{S} = \int_V \rho_m dV = 0 \quad (2.31)$$

$$i_{total} = \oint_S \vec{j}_{total} \cdot d\vec{S} = - \int_V \frac{\partial \rho}{\partial t} dV \quad (2.32)$$

em que:

- $\vec{e}(x, y, z, t)$ é o campo elétrico, em V/m.
- $\vec{h}(x, y, z, t)$ é o campo magnético, em A/m.
- $\vec{d}(x, y, z, t)$ é a densidade de fluxo elétrico, em C/m².
- $\vec{b}(x, y, z, t)$ é a densidade de fluxo magnético, em Wb/m² ou T.
- $i_m(x, y, z, t)$ é a corrente magnética fictícia, em V.
- $i_{total}(x, y, z, t)$ é a corrente magnética total, em A.
- $\vec{j}_{total}(x, y, z, t)$ é a densidade de corrente magnética, em A/m².
- $\rho(x, y, z, t)$ é a densidade de carga, em C/m³.
- $\rho_m(x, y, z, t)$ é a densidade de carga magnética fictícia, em Wb/m³.

2.6 Ondas eletromagnéticas

Como lição final das Equações de Maxwell, principalmente das Equações (2.23) e (2.24), entende-se que: quando os campos são variáveis no tempo, um não pode existir sem o outro e vice-versa. Exemplificando, a variação de um campo magnético ocasiona o surgimento de campo elétrico induzido. Esse campo elétrico induzido também é variável e, de

acordo com a hipótese de Maxwell, gera um campo magnético induzido. Por sua vez, esse campo magnético induzido também é variável e irá causar um campo elétrico induzido, e assim sucessivamente. Portanto, os campos magnéticos e elétricos são interdependentes e se propagam em todas as direções, resultando em um campo eletromagnético.

A partir dos experimentos de Hans C. Oersted⁵ e Faraday, Maxwell percebeu que as perturbações no campo elétrico e magnético incorporavam um caráter ondulatório, transportavam energia e estavam sujeitas aos fenômenos de reflexão, refração, difração e interferência, como quaisquer outras ondas conhecidas. Por isso, essas perturbações foram denominadas ondas eletromagnéticas. Maxwell fez essa descoberta no ano 1865, mas somente 23 anos depois, Hertz conseguiu comprovar que existem semelhanças entre as ondas eletromagnéticas e as luminosas. Ele descobriu, utilizando um transmissor de ondas, por ele construído, que ambas se propagam a 300 mil quilômetros por segundo, constatando, que elas se refletem e se refratam, e sofrem os mesmos fenômenos de interferência e difração, podendo também ser polarizadas. O nome da unidade de medida da frequência dessas ondas é hertz (Hz), em sua homenagem.

Dessa forma, a Figura 1 ilustra o comportamento da onda eletromagnética, a qual é basicamente composta de dois campos variáveis no tempo: o elétrico (em vermelho) e o magnético (em azul), que são perpendiculares entre si e perpendiculares à direção de propagação. Sobre como surgem, elas podem ser produzidas por cargas elétricas aceleradas. Uma das maneiras para isso é proporcionar o movimento oscilatório de elétrons, como acontece em antenas e outra maneira é por meio de saltos de elétrons entre diferentes camadas orbitais, como acontece nos aparelhos de raios-X (SIQUEIRA, 2021).

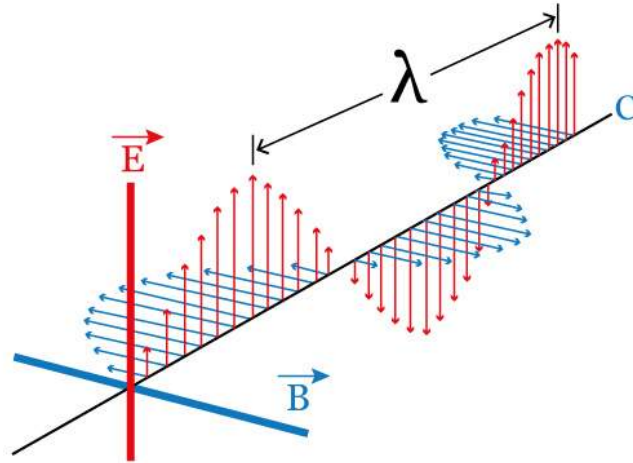
2.7 Campos harmônicos no tempo e notação fasorial

Na natureza, a maioria das aplicações possui campos e fontes eletromagnéticas com variações senoidais, sendo que em caso contrário pode ser modelada como a soma do efeito de várias fontes senoidais por meio de séries, integrais e transformadas de Fourier, se necessário. Assim, o problema pode ser simplificado (sem perda de generalidade) para o caso em que os campos possuem variações senoidais com o tempo. Esse tipo de campo é chamado de campo harmônico no tempo.

Como nos campos harmônicos no tempo já se sabe o comportamento do campo no tempo, é possível simplificar ainda mais o problema, eliminando a dependência no tempo das variáveis das Equações de Maxwell que descrevem a propagação. Para isso, utiliza-se

⁵ Oersted percebeu que a agulha imantada de uma bússola sofria deflexões quando estava próxima a um fio condutor por onde passava uma corrente elétrica. A única possibilidade para a mudança de direção da agulha da bússola era a presença de um campo magnético diferente daquele provocado pela Terra. Esse fato deu início ao eletromagnetismo.

Figura 1 – Onda eletromagnética.



Fonte: Rosa (2019).

o conceito de fasor⁶, aproveitando-se de que o campo é senoidal, para escrevê-lo na forma independente do tempo, $r\angle\phi$.

Exemplificando, considere o sinal $s(t)$ ⁷ como um sinal senoidal com amplitude A e fase ϕ , definido em (2.33). Nesse caso, o fasor seria $Ae^{j\phi}$, que é a parte que independe do tempo, e identifica sua amplitude e sua fase ($r\angle\phi$). Analogamente, considerando um campo harmônico vetorial \vec{v} , descrito em (2.34), seu fasor seria $\vec{A}(x, y, z) = |A(x, y, z)|e^{j\phi}$.

$$s(t) = Ae^{j\phi}e^{j\omega t} \quad (2.33)$$

$$\vec{v}(x, y, z, t) = \vec{A}(x, y, z) \cos(\omega t) \quad (2.34)$$

em que:

- \vec{A} é um vetor que pode ser complexo.
- ω é a frequência angular, em rad/s .
- $f = \frac{\omega}{2\pi}$ é a frequência de variação harmônica.

A vantagem de se usar essa notação é que, além de simplificar a representação dos campos, também reduz a complexidade de se aplicar derivada e integral. Conforme as Equações (2.35) a (2.40), tem-se que diferenciar uma função harmônica no tempo, equivale

⁶ amplitude complexa da representação analítica de um sinal

⁷ $s(t)$ não é um sinal físico real. $s(t)$ é a representação analítica de um sinal de valores reais.

a multiplicar seu fasor por $j\omega$ no domínio fasorial, enquanto que integrá-la equivale dividir seu fasor por $j\omega$ no domínio fasorial.

$$\frac{\partial v}{\partial t}(x, y, z, t) = \frac{\partial}{\partial t} \operatorname{Re} \left\{ \vec{A}(x, y, z, t) e^{j\omega t} \right\} \quad (2.35)$$

$$\frac{\partial v}{\partial t}(x, y, z, t) = \operatorname{Re} \left\{ \vec{A}(x, y, z, t) \frac{\partial}{\partial t} e^{j\omega t} \right\} \quad (2.36)$$

$$\frac{\partial v}{\partial t}(x, y, z, t) = \operatorname{Re} \left\{ j\omega \vec{A}(x, y, z, t) e^{j\omega t} \right\} \quad (2.37)$$

$$\int \vec{v}(x, y, z, t) dt = \int \operatorname{Re} \left\{ \vec{A}(x, y, z, t) e^{j\omega t} \right\} dt \quad (2.38)$$

$$\int \vec{v}(x, y, z, t) = \operatorname{Re} \left\{ \vec{A}(x, y, z, t) \int e^{j\omega t} dt \right\} \quad (2.39)$$

$$\int \vec{v}(x, y, z, t) = \operatorname{Re} \left\{ \frac{1}{j\omega} \vec{A}(x, y, z, t) e^{j\omega t} \right\} \quad (2.40)$$

2.7.1 Equações de Maxwell na forma fasorial

Assim, as equações de Maxwell podem ser reescritas para facilitar o manejo das equações durante os cálculos, como descrito nas Equações (2.41) a (2.45). Lembrando que este novo conjunto de equações é aplicável apenas para campos harmônicos no tempo.

$$\vec{\nabla} \times \vec{E} = -j\omega \vec{B} - \vec{M} \quad (2.41)$$

$$\vec{\nabla} \times \vec{H} = j\omega \vec{D} + \vec{J}_{total} \quad (2.42)$$

$$\vec{\nabla} \cdot \vec{D} = \rho \quad (2.43)$$

$$\vec{\nabla} \cdot \vec{B} = \rho_m = 0 \quad (2.44)$$

$$\vec{\nabla} \cdot \vec{J}_{total} = -j\omega \rho \quad (2.45)$$

em que:

- $\vec{E}(x, y, z)$ é o fasor vetorial do campo elétrico, em V/m.

- $\vec{H}(x, y, z, t)$ é o fasor vetorial do campo magnético, em A/m.
- $\vec{D}(x, y, z, t)$ é o fasor vetorial da densidade de fluxo elétrico, em C/m².
- $\vec{B}(x, y, z, t)$ é o fasor vetorial da densidade de fluxo magnético, em Wb/m² ou T.
- $\vec{M}(x, y, z, t)$ é o fasor vetorial da densidade de corrente magnética fictícia, em V/m².
- $\vec{J}_{total}(x, y, z, t)$ é o fasor vetorial da densidade de elétrica total, em A/m².
- $\rho(x, y, z, t)$ é o fasor escalar da densidade de carga, em C/m³.
- $\rho_m(x, y, z, t)$ é o fasor escalar da densidade de carga magnética fictícia, em Wb/m³.

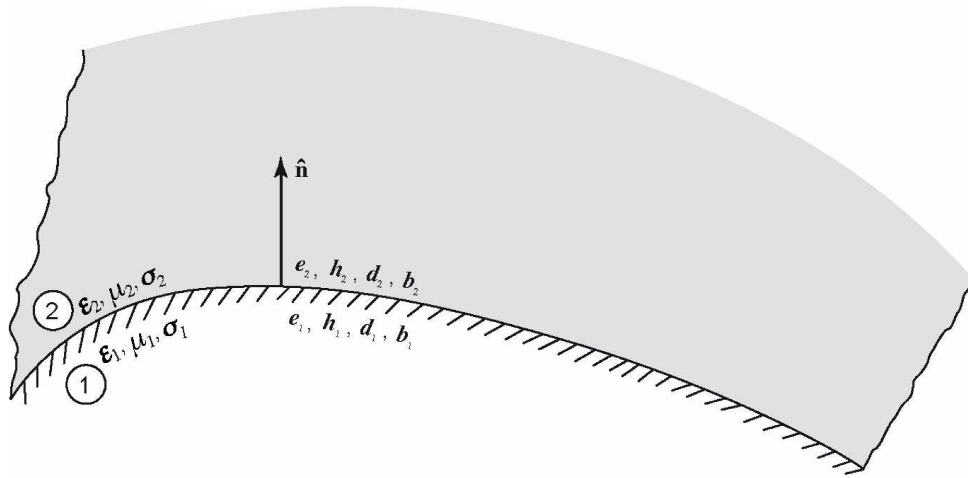
2.8 Condições de contorno

As equações de Maxwell na forma diferencial representam derivadas com respeito às coordenadas espaciais e, em pontos de descontinuidade, essas derivadas não possuem significado e não podem ser utilizadas para definir o comportamento dos campos ao longo das fronteiras. Nessas regiões, não se pode esperar que elas forneçam alguma informação válida (VASCONCELOS, 2017).

Considerando uma situação de fronteira entre dois meios ($i = 1, 2$), ilustrada na Figura 2, a Tabela 2 traz as condições de contorno para os campos elétrico e magnético, em que:

- ϵ_i é permissividade elétrica do meio i , em F/m.
- μ_i é a permeabilidade magnética do meio i , em H/m.
- σ_i é a condutividade elétrica do meio i , em S/m.
- \hat{n} é o vetor unitário normal à superfície de separação dos dois meios.
- m_s é a densidade linear de corrente magnética na superfície.
- j_s é a densidade linear de corrente elétrica na superfície.
- ρ_s é a densidade superficial de carga elétrica na superfície.
- ρ_{ms} é a densidade superficial de carga magnética na superfície.

Figura 2 – Fronteira entre dois meios.



Fonte: [BALANIS \(2012\)](#).

Tabela 2 – Condições de contorno para os campos elétrico e magnético.

	Geral	Meios com condutividade finita, sem fontes e sem cargas $\sigma_1, \sigma_2 \neq \infty$ $\mathbf{j}_s = 0$ $\rho_s = 0$ $\mathbf{m}_s = 0$ $\rho_{ms} = 0$	Meio 1 condutor elétrico perfeito $\mathbf{e}_1 = \mathbf{h}_1 = 0$ $\sigma_1 = \infty$ $\sigma_2 \neq \infty$ $\mathbf{m}_s = 0$ $\rho_{ms} = 0$	Meio 1 condutor magnético perfeito $\mathbf{e}_1 = \mathbf{h}_1 = 0$ $\mathbf{j}_s = 0$ $\rho_s = 0$
Campo elétrico tangencial	$-\hat{\mathbf{n}} \times (\mathbf{e}_2 - \mathbf{e}_1) = \mathbf{m}_s$	$\hat{\mathbf{n}} \times (\mathbf{e}_2 - \mathbf{e}_1) = 0$	$\hat{\mathbf{n}} \times \mathbf{e}_2 = 0$	$-\hat{\mathbf{n}} \times \mathbf{e}_2 = \mathbf{m}_s$
Campo magnético tangencial	$\hat{\mathbf{n}} \times (\mathbf{h}_2 - \mathbf{h}_1) = \mathbf{j}_s$	$\hat{\mathbf{n}} \times (\mathbf{h}_2 - \mathbf{h}_1) = 0$	$\hat{\mathbf{n}} \times \mathbf{h}_2 = \mathbf{j}_s$	$\hat{\mathbf{n}} \times \mathbf{h}_2 = 0$
Densidade de fluxo elétrico normal	$\hat{\mathbf{n}} \cdot (\mathbf{d}_2 - \mathbf{d}_1) = \rho_s$	$\hat{\mathbf{n}} \cdot (\mathbf{d}_2 - \mathbf{d}_1) = 0$	$\hat{\mathbf{n}} \cdot \mathbf{d}_2 = \rho_s$	$\hat{\mathbf{n}} \cdot \mathbf{d}_2 = 0$
Densidade de fluxo magnético normal	$\hat{\mathbf{n}} \cdot (\mathbf{b}_2 - \mathbf{b}_1) = \rho_{ms}$	$\hat{\mathbf{n}} \cdot (\mathbf{b}_2 - \mathbf{b}_1) = 0$	$\hat{\mathbf{n}} \cdot \mathbf{b}_2 = 0$	$\hat{\mathbf{n}} \cdot \mathbf{b}_2 = \rho_{ms}$

Fonte: [BALANIS \(2012\)](#).

A última coluna da tabela considera um meio condutor magnético perfeito. Em geral, um meio desse tipo é definido como um material no qual os campos elétrico e magnético variáveis no tempo são nulos. A componente tangencial do campo magnético é nula na superfície e cargas magnéticas se movem para a superfície do material, criando uma densidade de corrente magnética em uma camada muito fina na superfície. Embora esses materiais não existam fisicamente, eles geralmente são utilizados como equivalente para descrever algumas situações elétricas complexas ([VASCONCELOS, 2017](#)).

2.9 Equação da onda e a onda plana uniforme

Existem quatro principais tipos de meios nos quais as ondas eletromagnéticas podem propagar:

- espaço livre ($\sigma = 0$, $\epsilon = \epsilon_0$, $\mu = \mu_0$);
- dielétrico sem perdas ($\sigma = 0$, $\epsilon = \epsilon_r \epsilon_0$, $\mu = \mu_r \mu_0$);
- dielétrico com perdas ($\sigma \neq 0$, $\epsilon = \epsilon_r \epsilon_0$, $\mu = \mu_r \mu_0$);
- bons condutores ($\sigma \approx \infty$, $\epsilon = \epsilon_0$, $\mu = \mu_r \mu_0$ ou $\sigma \gg \omega \epsilon$).

O caso dos dielétricos com perdas é o mais geral, e os demais casos podem ser resolvidos a partir de simplificações dele. Assim, considerando esse caso mais geral, para um meio qualquer que está conduzindo ($\sigma \neq 0$), uma corrente flui com característica descrita na Equação (2.46) e, para campos harmônicos no tempo, a Equação (2.42) pode ser reescrita com na Equação (2.47).

$$\vec{J} = \sigma \vec{E} \quad (2.46)$$

$$\vec{\nabla} \times \vec{H} = (\sigma + j\omega\epsilon) \vec{E} = j\omega \left(\epsilon + \frac{\sigma}{j\omega} \right) \vec{E} = j\omega \epsilon_c \vec{E} \quad (2.47)$$

em que $\epsilon_c = \epsilon - j\frac{\sigma}{\omega}$ é a permissividade elétrica complexa, em F/m, que pode ser escrita como $\epsilon_c = \epsilon' - j\epsilon''$.

Assim, considerando as Equações de Maxwell, de (2.41) a (2.45), e a forma reescrita na Equação (2.47), têm-se as Equações (2.48) e (2.49). Elas são conhecidas como as equações vetoriais homogêneas de Helmholtz ou, simplesmente, como equações vetoriais da onda. Além disso, como γ é complexo, pode-se fazer $\gamma = \alpha + j\beta$, sendo que α e β são definidos pelas Equações (2.50) e (2.51).

$$\nabla^2 \vec{E} - \gamma^2 \vec{E} = 0 \quad (2.48)$$

$$\nabla^2 \vec{H} - \gamma^2 \vec{H} = 0 \quad (2.49)$$

em que $\gamma^2 = -\omega^2 \mu \epsilon + j\omega \mu \sigma$ é chamada constante de propagação da onda, em m^{-1} .

$$\alpha = \omega \sqrt{\frac{\mu\epsilon}{2} \left[\sqrt{1 + \left(\frac{\sigma}{\omega\epsilon}\right)^2} - 1 \right]} \quad (\text{Np/m}) \quad (2.50)$$

$$\beta = \omega \sqrt{\frac{\mu\epsilon}{2} \left[\sqrt{1 + \left(\frac{\sigma}{\omega\epsilon}\right)^2} + 1 \right]} \quad (\text{rad/m}) \quad (2.51)$$

Sem perda de generalidade, para efeito didático, pode-se assumir que o campo elétrico é função apenas de z e possui apenas uma componente em x , conforme a Equação (2.52). Os resultados podem ser estendidos para y . Além disso, substituindo (2.52) em (2.48) têm-se as Equações (2.53) a (2.55).

$$\vec{E} = E_x \hat{x} \quad (2.52)$$

$$(\nabla^2 - \gamma^2) [E_x \hat{x}] = 0 \quad (2.53)$$

$$\frac{\partial^2 E_x(z)}{\partial x^2} + \frac{\partial^2 E_x(z)}{\partial y^2} + \frac{\partial^2 E_x(z)}{\partial z^2} - \gamma^2 E_x(z) = 0 \quad (2.54)$$

$$\left[\frac{\partial^2}{\partial z^2} - \gamma^2 \right] E_x(z) = 0 \quad (2.55)$$

A Equação (2.55) é uma equação de onda escalar e uma equação diferencial homogênea e linear cuja solução é dada pela Equação (2.56).

$$E_x(z) = E_0 e^{-\gamma z} + E'_0 e^{+\gamma z} \quad (2.56)$$

em que E_0 e E'_0 são constantes reais.

Recuperando a dependência do tempo em (2.56), obtém-se o campo físico real $e_x(z, t)$, dado pela Equação (2.57).

$$\begin{aligned} e_x(z, t) &= \text{Re} \left\{ E_0 e^{-\gamma z + j\omega t} + E'_0 e^{+\gamma z + j\omega t} \right\} \\ &= \text{Re} \left\{ E_0 e^{-(\alpha + j\beta)z + j\omega t} + E'_0 e^{(\alpha + j\beta)z + j\omega t} \right\} \\ &= \text{Re} \left\{ E_0 e^{-\alpha z + j(\omega - \beta z)} + E'_0 e^{\alpha z + j(\omega + \beta z)} \right\} \\ &= E_0 e^{-\alpha z} \cos(\omega t - \beta z) + E'_0 e^{\alpha z} \cos(\omega t + \beta z) \end{aligned} \quad (2.57)$$

A partir do campo elétrico, determina-se o campo magnético a partir da equação de Maxwell em (2.58), para se chegar em (2.63), e assim recuperar o campo magnético físico real em (2.64).

$$\vec{\nabla} \times \vec{E} = -j\omega\mu\vec{H} \quad (2.58)$$

$$\nabla \times [E_x(z)\hat{x}] = -j\omega\mu\vec{H} \quad (2.59)$$

$$\frac{\partial E_x(z)}{\partial z}\hat{y} = -j\omega\mu\vec{H} \quad (2.60)$$

$$\vec{H} = \frac{j}{\omega\mu} \frac{\partial}{\partial z} [E_0 e^{-\gamma z} + E'_0 e^{+\gamma z}] \hat{y} \quad (2.61)$$

$$\vec{H} = \frac{-j\gamma}{\omega\mu} (E_0 e^{-\gamma z} - E'_0 e^{+\gamma z}) \hat{y} \quad (2.62)$$

$$H_y(z) = H_0 e^{-\gamma z} + H'_0 e^{+\gamma z} \quad (2.63)$$

em que $H_0 = -\frac{j\gamma}{\omega\mu} E_0$ e $H'_0 = \frac{j\gamma}{\omega\mu} E'_0$.

$$\begin{aligned} h_y(z, t) &= \text{Re} \left\{ H_0 e^{-\gamma z + j\omega t} + H'_0 e^{+\gamma z + j\omega t} \right\} \\ &= \text{Re} \left\{ H_0 e^{-(\alpha + j\beta)z + j\omega t} + H'_0 e^{(\alpha + j\beta)z + j\omega t} \right\} \\ &= \text{Re} \left\{ H_0 e^{-\alpha z + j(\omega t - \beta z)} + H'_0 e^{\alpha z + j(\omega t + \beta z)} \right\} \\ &= H_0 e^{-\alpha z} \cos(\omega t - \beta z) + H'_0 e^{\alpha z} \cos(\omega t + \beta z) \end{aligned} \quad (2.64)$$

As Equações (2.56), (2.57), (2.63) e (2.64) representam duas ondas planas, uma que propaga na direção positiva de z e outra na direção negativa de z . É importante notar que os campos elétrico e magnético estão no mesmo plano (xy ou $z = cte$), têm o mesmo valor em todo o plano e estão em quadratura espacial. Isso faz com que esse tipo de onda seja chamado de onda plana uniforme transversal eletromagnética.

3 MÉTODO DE EQUAÇÕES PARABÓLICAS

3.1 Introdução

A equação parabólica (do inglês, *Parabolic Equation - PE*) é uma aproximação da equação da onda que modela a propagação de energia em um cone centrado em uma direção de interesse, a direção paraxial. A aproximação parabólica foi introduzida por Leontovich e Fock, nos anos de 1940, para tratar o problema da difração de ondas de rádio ao redor da Terra, e continuou a ser utilizada em outras aplicações, tais como refração atmosférica e difração por obstáculos. Muitos anos depois, com o advento dos computadores digitais, a ideia da aproximação parabólica foi retomada, dessa vez para encontrar soluções numéricas. Um dos pesquisadores dessa área foi Claerbout, quem propôs uma aproximação interessante ao desenvolver códigos de diferenças finitas para aplicações geofísicas (LEVY, 2000).

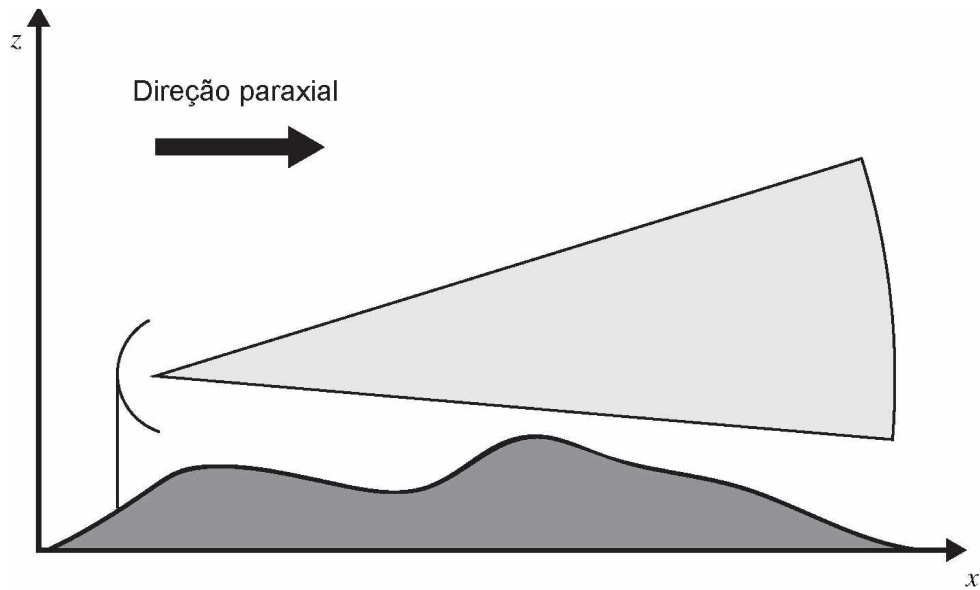
É um método que considera a refratividade atmosférica, a topografia e as características do solo da região de interesse. Além disso, ao contrário de métodos de raios comuns, o método PE é também válido nas regiões de sombra, e, acerca do cálculo computacional, sua principal vantagem é que a resolução passa a ser iterativa e não precisa ser calculada simultaneamente em todos os pontos, poupando memória. Ainda vale ressaltar que o método utilizado aqui é do tipo *1-way propagation*, ou seja, ignora o retroespalhamento, e para a aplicação deste trabalho em problemas de propagação a longa distância já produz resultados bastante satisfatórios. Assim, os métodos *1-way propagation*, que consideram o retroespalhamento, não são tratados aqui, já que são mais adequados, por exemplo, para aplicações de radar.

3.2 Formulação do método

Como já dito, o método PE simplifica a equação da onda de Helmholtz para uma equação parabólica, com o objetivo de trazer bons resultados na direção paraxial, representada por um cone no sentido da propagação de energia da onda, como ilustrado na Figura 3. Esse cone deve ser colocado em uma região onde o índice de refração da atmosfera possua variações suaves e na qual podem ser estabelecidas condições de contorno apropriadas nas suas fronteiras.

Devido ao método ser iterativo, a solução numérica de uma coluna é obtida a partir da coluna anterior e vai “marchando” até a máxima distância desejada pelo usuário. Para isso, é necessário executar o método em um domínio limitado por uma fronteira

Figura 3 – Propagação paraxial na troposfera.

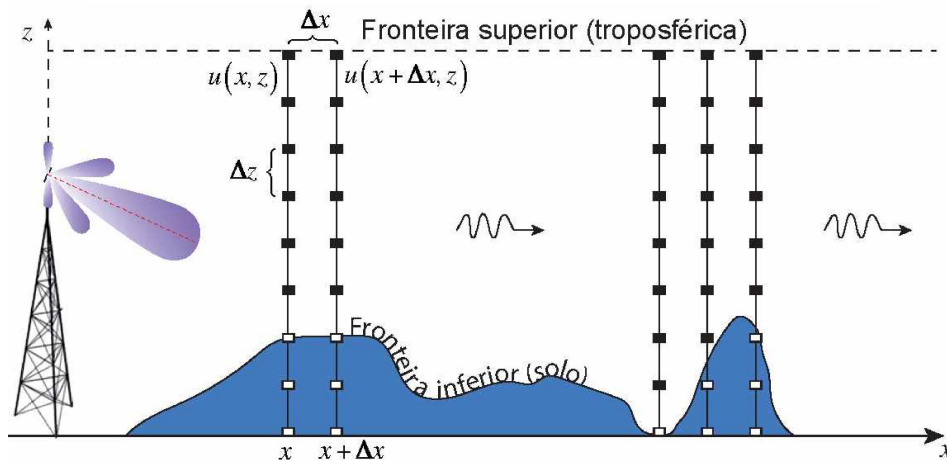


Fonte: Levy (2000).

troposférica superior sob condição de radiação e por uma fronteira inferior de terreno irregular sob condição de contorno apropriada, como apresentado na Figura 4.

Na maioria dos problemas, a fronteira inferior representa a interface atmosfera/solo e a fronteira superior deveria se estender ao infinito. Como é impossível ter uma matriz que ocupe uma memória infinita, são utilizadas técnicas que simulam essa condição, no caso, a existência de uma camada atmosférica de tamanho calculável que simula a absorção da onda eletromagnética e impede que sejam considerados componentes de sua reflexão. A fronteira superior será tratada com mais detalhes em seções adiante.

Figura 4 – Esquema de utilização do método PE.



Fonte: OZGUN et al. (2011).

Para o desenvolvimento genérico do método PE, são feitas as seguintes observações:

- assume-se que os campos são harmônicos no tempo com frequência angular ω e, por isso, é usada a notação fasorial;
- é destinado a resolver problemas bidimensionais;
- é usado o sistema cartesiano de coordenadas;
- por convenção, a direção positiva do eixo x é tomada como direção paraxial e a altitude é representada no eixo z ;
- os campos elétrico e magnético são independentes da coordenada transversa y (não são considerados efeitos de despolarização);
- todo problema pode ser decomposto em dois casos: problema horizontalmente polarizado ou problema verticalmente polarizado;
- na polarização horizontal, o campo elétrico \vec{E} possui somente uma componente diferente de zero, a E_y ;
- na polarização vertical, o campo magnético \vec{H} possui somente a componente H_y diferente de zero;
- para generalidade dos cálculos, trabalha-se com uma variável auxiliar ψ que dependendo da polarização utilizada como referência para os cálculos representa: ou o campo elétrico, como na Equação (3.1), ou o campo magnético, como em (3.2).

$$\psi(x, z) = E_y(x, z) \quad (3.1)$$

$$\psi(x, z) = H_y(x, z) \quad (3.2)$$

3.3 Aproximação parabólica

Equações parabólicas diferenciais são bem adequadas para métodos numéricos, já que usualmente possuem soluções que são numericamente bem comportadas. Por isso, suas soluções são computadas bem mais rápido do que se fossem usadas soluções exatas das equações de Maxwell. Assim, considerando o modelo de Terra esférica lisa (ou suave), a solução para ângulo amplo pode ser determinada usando a técnica de diferenças finitas aliada à forma modificada da PE padrão. O resultado é chamado de equação de Claerbout. Para chegar a ela, inicia-se na equação homogênea vetorial de Helmholtz, dada em (3.3).

$$\nabla^2 \vec{\psi} - \gamma^2 \vec{\psi} = 0 \quad (3.3)$$

em que o campo vetorial $\vec{\psi}$ possui apenas uma componente na direção y ($\vec{\psi} = \psi \hat{y}$).

Como $\vec{\psi} = \psi \hat{y}$, a expressão vetorial da Equação (3.3) se reduz à expressão escalar da Equação (3.4).

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial z^2} - \gamma^2 \psi = 0 \quad (3.4)$$

em que a constante de propagação é dada por $\gamma^2 = -\omega^2 \mu \epsilon + j\omega \mu \sigma$.

Considerando um meio sem perdas e não magnético ($\epsilon = \epsilon_0 \epsilon_r$, $\mu = \mu_0$, $\sigma = 0$), o termo γ pode ser reescrito como $\gamma^2 = -\omega^2 \mu_0 \epsilon_0 \epsilon_r$, e a Equação (3.4) se torna a Equação (3.6).

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial z^2} + \omega^2 \mu_0 \epsilon_0 \epsilon_r \psi = 0 \quad (3.5)$$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial z^2} + k_0^2 \epsilon_r \psi = 0 \quad (3.6)$$

em que $k_0^2 = \omega \sqrt{\mu_0 \epsilon_0}$ é o número de onda ou a constante de fase do vácuo.

Além disso, em um meio sem perdas e não magnético, o índice de refração pode ser dado por $n = \sqrt{\epsilon_r}$. Assim, obtém-se a Equação (3.7).

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial z^2} + k_0^2 n^2 \psi = 0 \quad (3.7)$$

Quando o índice de refração varia espacialmente, a Equação (3.7) não é exatamente válida, mas ainda pode ser uma boa aproximação se as variações do índice de refração forem suaves na escala do comprimento de onda. Então, considerando esse comportamento, a Equação (3.8) pode ser utilizada nesses casos. Vasconcelos (2017) ainda prova que a forma da Equação (3.8), equivalente a (3.9), pode ser usada tanto para polarização horizontal quanto para a vertical.

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial z^2} + k_0^2 n(x, z)^2 \psi = 0 \quad (3.8)$$

$$\frac{\nabla^2 \psi(x, z)}{\partial z^2} + k_0^2 n(x, z)^2 \psi = 0 \quad (3.9)$$

A aproximação parabólica consiste considerar que a componente de campo $\psi(x, z)$ pode ser fatorada em dois termos, como na Equação (3.10), em que o termo e^{jk_0x} é uma variação rápida de fase e $u(x, z)$ é uma variação lenta de amplitude. Isso só pode ser feito porque o interesse está nas variações do campo em escalas grandes com relação ao comprimento de onda.

$$\psi(x, z) = u(x, z) e^{jk_0x} \quad (3.10)$$

Grande parte de simplicidade da equação parabólica frente à equação elíptica vem dessa decomposição. Porém, a acurácia dessa aproximação é boa apenas para uma região paraxial dentro do cone mostrado na Figura 3. Para pontos distantes do cone, o erro entre a modelagem parabólica e elíptica aumenta consideravelmente (VASCONCELOS, 2017).

Substituindo a aproximação de (3.10) em (3.7), mas já tendo em mente que o índice de refração pode variar espacialmente e, nesse caso, os cálculos realizados adiante ainda são válidos, chega-se na Equação (3.12).

$$\frac{\partial^2 (ue^{jk_0x})}{\partial x^2} + \frac{\partial^2 (ue^{jk_0x})}{\partial z^2} + k_0^2 n^2 (ue^{jk_0x}) = 0 \quad (3.11)$$

$$\frac{\partial^2 u}{\partial x^2} + 2jk_0 \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial z^2} + k_0^2 (n^2 - 1) u = 0 \quad (3.12)$$

A Equação (3.12) pode ser fatorada de acordo com a Equação (3.13), por meio do fator pseudodiferencial Q .

$$\left(\frac{\partial}{\partial x} + jk_0(1 - Q) \right) \left(\frac{\partial}{\partial x} + jk_0(1 + Q) \right) u = 0 \quad (3.13)$$

em que:

$$Q = \sqrt{\frac{1}{k_0^2} \frac{\partial^2}{\partial z^2} + n^2(x, z)} = \sqrt{1 + Z} \quad (3.14)$$

$$Z = \frac{1}{k_0^2} \frac{\partial^2}{\partial z^2} + n^2(x, z) - 1 \quad (3.15)$$

Operadores pseudodiferenciais são formados por derivadas parciais e funções ordinárias das variáveis. A operação de raiz quadrada em Q existe para representar a composição de operações no sentido mostrado na Equação (3.16). Nesse caso, também assume-se

que expansões comuns para a operação de raiz quadrada podem ser aplicadas a Q (VASCONCELOS, 2017).

$$Q(Q(u)) = \frac{1}{k_0^2} \frac{\partial^2 u}{\partial z^2}(x, z) + n^2 u(x, z) \quad (3.16)$$

A Equação (3.13) representa tanto as ondas propagando para frente como para trás e a parte que representa as ondas propagando para frente é mostrada na Equação (3.17).

$$\left(\frac{\partial}{\partial x} + ik_0(1 - Q) \right) u = \left[\frac{\partial}{\partial x} + ik_0(1 - \sqrt{1 + Z}) \right] u = 0 \quad (3.17)$$

3.4 Equação de Claerbout

Da Equação (3.17), é possível obter a equação parabólica padrão usando a expansão de Taylor de primeira ordem de $\sqrt{1 + Z}$, isto é, $\sqrt{1 + Z} \approx 1 + Z/2$. A equação padrão é limitada a geometrias de propagação que não envolva caminhos de propagação com ângulos maiores que 15° a partir da direção paraxial. Para ângulos maiores, deve ser usada uma expansão mais precisa. Como a expansão de Taylor de $\sqrt{1 + Z}$ produz esquemas numéricos instáveis, é escolhido utilizar a aproximação de Padé, cuja forma está em (3.18) (HOLM, 2007).

$$\sqrt{1 + Z} \approx \frac{1 + aZ}{1 + bZ} \quad (3.18)$$

Utilizando $a = 3/4$ e $b = 1/4$, a Equação (3.18) se torna uma expansão de Taylor de segunda ordem. Assim, usando esses parâmetros, também escolhidos por Claerbout quando sugeriu essa aproximação, reescreve-se (3.14) em (3.19).

$$Q = \sqrt{\frac{1}{k_0^2} \frac{\partial^2}{\partial z^2} + n^2(x, z)} = \sqrt{1 + Z} = \frac{1 + aZ}{1 + bZ} \quad (3.19)$$

em que:

$$Z = \frac{1}{k_0^2} \frac{\partial^2}{\partial z^2} + n^2(x, z) - 1 \quad (3.20)$$

Desenvolvendo (3.17) usando os parâmetros de a e b , têm-se as Equações (3.21) a (3.26). Sendo que (3.26) pode ser reescrita como (3.27).

$$\frac{\partial u}{\partial x} = -jk_0(1 - Q)u \quad (3.21)$$

$$\frac{\partial u}{\partial x} = -jk_0 \left(1 - \sqrt{\frac{1}{k_0^2} \frac{\partial^2}{\partial z^2} + n^2(x, z)}\right) u = \sqrt{1 + Z} u \quad (3.22)$$

$$\frac{\partial u}{\partial x} = -jk_0(1 - \sqrt{1 + Z})u \quad (3.23)$$

$$\frac{\partial u}{\partial x} = -jk_0 \left(1 - \frac{1 + aZ}{1 + bZ}\right) u \quad (3.24)$$

$$(1 + bZ) \frac{\partial u}{\partial x} - jk_0 Z(a - b)u = 0 \quad (3.25)$$

$$\left(1 + \frac{Z}{4}\right) \frac{\partial u}{\partial x} - jk_0 \frac{Z}{2} u = 0 \quad (3.26)$$

$$\frac{\partial^3 u}{\partial x \partial z^2} - 2jk_0 \frac{\partial^2 u}{\partial z^2} + k_0^2(n^2 + 3) \frac{\partial u}{\partial x} - 2jk_0^3(n^2 - 1)u = 0 \quad (3.27)$$

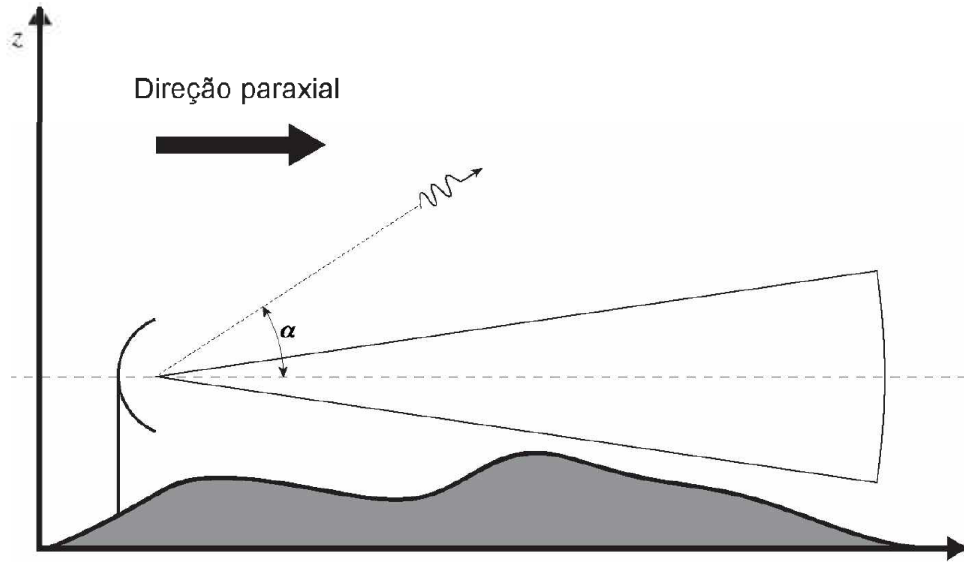
A Equação (3.26) é uma equação parabólica do tipo WAPE (de ângulo amplo). Nesse caso, o erro para uma onda plana propagando a um ângulo α da horizontal, mostrado na Figura 5, é da ordem de $\sin^6 \alpha$ e a solução é aceitável para ângulos de até 45° . Assim, a aproximação de Claerbout utiliza um esquema de Padé-(1,1) e os coeficientes $a = 3/4$ e $b = 1/4$ (KIRBY, 1986).

3.5 Condições de contorno para a WAPE

3.5.1 Fronteira inferior (solo)

A fronteira inferior é o solo, logo pode se utilizar as condições de contorno de Leontovich e considerando o terreno plano tem-se a Equação (3.28).

$$\frac{\partial \psi(x, 0)}{\partial z} + ik_0 \eta(x) \psi(x, 0) = 0 \quad (3.28)$$

Figura 5 – Ângulo de propagação α .


Fonte: Vasconcelos (2017).

em que $\eta(x)$ é

$$\eta(x) = \begin{cases} \sqrt{\epsilon(x) - 1} & \text{para a polarização horizontal} \\ \frac{\sqrt{\epsilon(x) - 1}}{\epsilon(x)} & \text{para a polarização vertical} \end{cases} \quad (3.29)$$

e $\epsilon(x)$ é a permissividade complexa relativa do solo.

Como $\psi(x, z) = e^{jk_0x}u(x, z)$, a condição de contorno para o campo $u(x, z)$ é a mesma que a de $\psi(x, z)$, representado em (3.30).

$$\frac{\partial u(x, 0)}{\partial z} = -ik_0\eta(x)u(x, 0) \quad (3.30)$$

Como o campo abaixo do terreno é desconhecido, deve ser usada uma aproximação unilateral para a derivada de (3.30) que será substituída em (3.27). Assim, considerando o passo de altura denotado por Δz , tem-se a Equação (3.31). Essa expressão melhor aproxima a derivada de segunda ordem para o ponto $(x, \Delta z/4)$ e não $(x, 0)$, mas isso não trará erros significativos.

$$\begin{aligned} \frac{\partial^2 u(x, 0)}{\partial z^2} &\approx \frac{2}{\Delta z} \left[\frac{\partial u(x, \Delta z/2)}{\partial z} - \frac{\partial u(x, 0)}{\partial z} \right] \\ &= \frac{2}{\Delta z} \left[\frac{\partial u(x, \Delta z/2)}{\partial z} + ik_0\eta(x)u(x, 0) \right] \end{aligned} \quad (3.31)$$

Assim para $z = 0$, ao substituir (3.31) em (3.27) obtém-se a Equação (3.32).

$$\begin{aligned} & \frac{2}{\Delta z} \left[\frac{\partial^2 u(x, \Delta z/2)}{\partial x \partial z} + ik_0 \eta(x) \frac{\partial u(x, 0)}{\partial x} + ik_0 \frac{\partial \eta(x)}{\partial x} u(x, 0) \right] - \dots \\ & \dots - 2ik_0 \frac{2}{\Delta z} \left[\frac{\partial u(x, \Delta z/2)}{\partial z} + ik_0 \eta(x) u(x, 0) \right] + \dots \\ & \dots + k_0^2 (n^2 + 3) \frac{\partial u(x, 0)}{\partial x} - 2ik_0^3 (n^2 - 1) u(x, 0) = 0 \end{aligned} \quad (3.32)$$

3.5.2 Fronteira superior (troposférica)

É de interesse simular a condição de Sommerfeld no infinito, em que toda energia incidente na fronteira superior escapa para o infinito. Para isso, pode-se usar uma camada absorvedora. Esse artifício consiste em estender o domínio de cálculo e aplicar um filtro à equação parabólica de forma que a energia seja absorvida e quase nenhuma energia seja refletida de volta para o domínio de interesse.

O filtro mais simples e eficaz utilizado é a janela de Hanning, dado pela Equação (3.33). Essa janela satisfaz $\phi(0) = 1$, $\phi(1) = 0$ e as derivadas nos pontos finais são zero, que assegura um casamento suave com o resto do domínio. Um exemplo, de como a camada absorvedora é aplicada ao sistema, é mostrado na Figura 6.

$$\phi(t) = \frac{1 + \cos(\pi t)}{2} \quad (3.33)$$

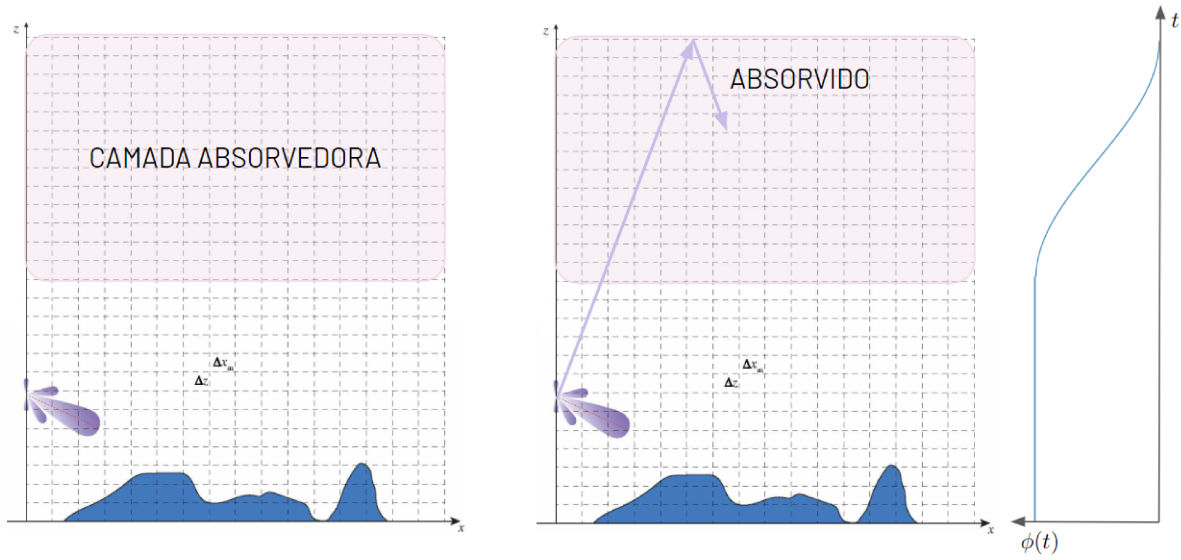
3.6 Transformação para a Terra Plana

A solução para a Terra plana é mais comum e menos custosa de se determinar, por isso, é de interesse realizar a transformação de Terra esférica para a Terra plana da Figura 7. Essa ideia foi introduzida por Pekeris (1946) e, mesmo com tal transformação, a equação da onda continua válida em baixas altitudes se o índice de refração n for substituído pelo índice de refração modificado m .

A modificação ideal de Pekeris é apresentada na Equação (3.34). Entretanto, não pode ser usada sem que haja uma distorção da equação da onda, já que essa não é uma transformação conformal¹. Por isso, ao invés de usar a transformação da Equação (3.34), define-se uma transformação dada pela Equação (3.35).

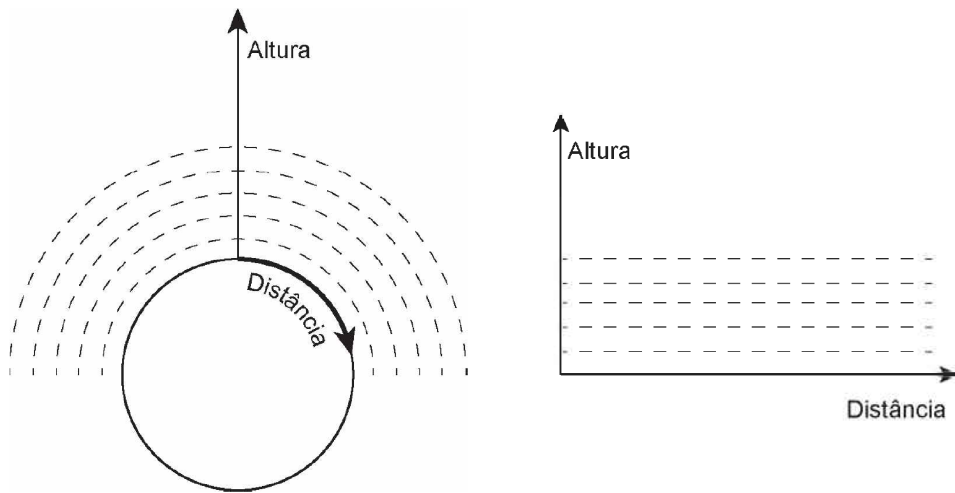
¹ transformação geométrica que preserva as relações angulares entre os vetores

Figura 6 – Exemplificação da camada absorvedora.



Fonte: Autora.

Figura 7 – Transformação para Terra plana.



Fonte: Levy (2000).

$$m(x, h) = n(x, h) + \frac{h}{a} \tag{3.34}$$

em que a é raio médio da Terra e h é a altitude acima da superfície da Terra.

$$\nabla^2(x, z) = e^{\frac{2z}{a}} \nabla^2(X, Z) \tag{3.35}$$

Substituindo o Laplaciano da Equação (3.35) na Equação (3.9) obtém-se a equação da onda no novo sistema de coordenadas na Equação (3.36).

$$\nabla^2\psi(x, z) + k_0^2\tilde{n}^2(x, z)e^{\frac{2z}{a}}\psi(x, z) = 0 \quad (3.36)$$

Definindo então o índice de refração modificado \tilde{m} na Equação (3.37), a equação da onda se torna (3.38).

$$\tilde{m}(x, z) = \tilde{n}(x, z)e^{\frac{z}{a}} \quad (3.37)$$

$$\nabla^2\psi(x, z) + k_0^2\tilde{m}^2(x, z)\psi(x, z) = 0 \quad (3.38)$$

3.7 Transformação para terrenos irregulares

3.7.1 WAPE para terreno irregular

Para considerar as variações relevo, pode-se fazer uma mudança de coordenadas de (x, y) para (v, w) , usando a técnica de Donohue e Kuttler (2000), como mostrado na Figura 8. Aplicando a transformação na coordenada de altura, obtém-se uma forma mais genérica da Equação (3.17), a Equação (3.39).

$$\left[\frac{\partial}{\partial x} + i\frac{\partial\theta}{\partial x} - T' \left(\frac{\partial}{\partial z} + i\frac{\partial\theta}{\partial z} \right) - i\sqrt{\left(\frac{\partial}{\partial z} + i\frac{\partial\theta}{\partial z} \right)^2 + k^2n^2} \right] u = 0 \quad (3.39)$$

em que $T' = dT/dv = dT/dx$ é a inclinação do terreno, $x = v$, $z = w - T(v)$, v é a correspondência para o novo sistema de coordenadas, e $T(v)$ é o perfil do terreno. A função $u(x, z)$ é relacionada a $\psi(x, z)$ por meio da equação $\psi(x, z) = e^{i\theta}u(x, z)$, em que o fator de fase é dado pela Equação (3.40).

$$\theta(x, z) = kzT'(x) + f(x) \quad (3.40)$$

Para um terreno linear por partes, tem-se a Equação (3.41), em que a inclinação do terreno $S = T'(x)$ é constante sobre cada elemento linear. Como $f(x)$ é uma função arbitrária, escolhe-se $f(x)$ que satisfaz (3.42), ou seja (3.43), que resulta em um esquema de diferenças finitas conhecido. Assim o fator de fase é dado por (3.44) (HOLM, 2007).

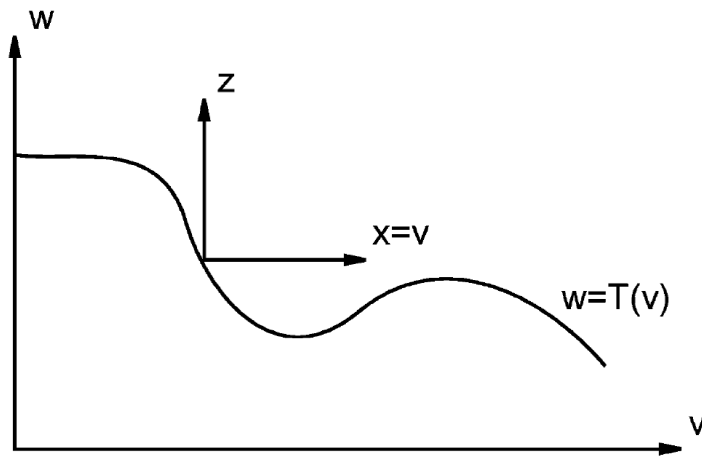
$$\theta(x, z) = kSz + f(x) \quad (3.41)$$

$$\frac{f'}{k} - \frac{S^2}{2} = 1 \quad (3.42)$$

$$f(x) = kx + 0.5kS^2x \quad (3.43)$$

$$\theta(x, z) = kSz + 0.5kS^2x + kx \quad (3.44)$$

Figura 8 – Ilustração dos sistemas de coordenadas (x, y) mapeado e (v, w) não mapeado.



Fonte: Holm (2007).

3.7.2 Condição de contorno da WAPE para terreno irregular

Um contorno com uma inclinação fixa S é descrita pela condição Leontovich pela Equação (3.45). Conseqüentemente, usando as aproximações de (3.46) e (3.48) desenvolvidas no artigo de Holm (2007), obtêm-se as Equações (3.50) e (3.51). Observa-se, então, que se tem o cenário da Figura 9.

$$\frac{\partial \psi(x, 0)}{\partial n} + \frac{ik_0 \eta(x) \psi(x, 0)}{\sqrt{1 + S^2}} = 0 \quad (3.45)$$

$$\frac{\partial u(x_m, 0)}{\partial n} \approx \frac{u_1^m - u_{-1}^m}{2\Delta z \sqrt{1 + S^2}} \quad (3.46)$$

$$\approx - \left(\eta + \frac{S^3}{2} \right) \frac{ik_0 u_0^m}{\sqrt{1 + S^2}} \quad (3.47)$$

$$\frac{\partial^2 u(x_m, 0)}{\partial n^2} \approx \frac{u_1^m - 2u_0^m + u_{-1}^m}{\Delta z^2 (1 + S^2)} \quad (3.48)$$

$$\approx - \left(\eta + \frac{S^3}{2} \right)^2 \frac{k_0^2 u_0^m}{(1 + S^2)} \quad (3.49)$$

$$\begin{cases} u_{-1}^m - 2ik_0 \Delta z \left(\eta + \frac{1}{2} S^3 \right) u_0^m \approx u_1^m \\ u_{-1}^m - 2u_0^m + k_0^2 \Delta z^2 \left(\eta + \frac{1}{2} S^3 \right)^2 u_0^m \approx -u_1^m \end{cases} \quad (3.50)$$

em que para um valor conhecido de u_1^m , tem-se:

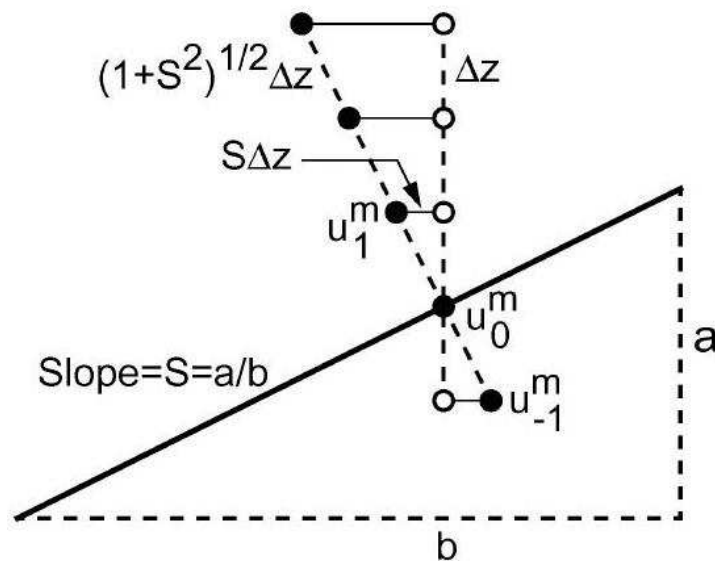
$$\begin{cases} u_0^m = \frac{1}{1 - \kappa + \kappa^2/2} u_1^m \\ u_{-1}^m = \frac{1 + \kappa + \kappa^2/2}{1 - \kappa + \kappa^2/2} u_1^m \end{cases} \quad (3.51)$$

em que $\kappa = ik_0 \Delta z (\eta + S^3/2)$. Além disso, a partir de (3.51), tem-se que u_0^m é calculada separada do esquema de diferenças, logo obtém-se uma solução dada pelas Equações (3.52) e (3.53).

$$cu_{j-1}^m + D_j^m u_j^m + cu_{j+1}^m = \tilde{c}u_{j-1}^{m-1} + \tilde{D}_j^m u_j^{m-1} + \tilde{c}u_{j+1}^{m-1} \quad (3.52)$$

$$cu_0^m + D_1^m u_1^m + cu_2^m = \tilde{c}u_0^{m-1} + \tilde{D}_1^m u_1^{m-1} + \tilde{c}u_2^{m-1} \quad (3.53)$$

Figura 9 – Ilustração do deslocamento de fase da frente de onda. Aqui, u_1^m , u_0^m , e u_{-1}^m referem-se a $u(x_m, \Delta z)$, $u(x_m, 0)$, e $u(x_m, -\Delta z)$, respectivamente, em que Δz é o passo de altura.



Fonte: Holm (2007).

3.8 Esquema de resolução por diferença finitas

3.8.1 WAPE para terreno plano

Usando a notação “ \sim ” para denotar, por exemplo dado $F = F(\Delta x)$, $\tilde{F} = F(-\Delta x)$. A partir das Equações (3.27) e (3.31), é possível montar as matrizes para determinação de u numericamente, mediante (3.54)

$$\mathbf{A}_m \mathbf{U}_m = \mathbf{B}_m \mathbf{U}_{m-1} \quad (3.54)$$

em que:

$$\mathbf{U}_m = \begin{pmatrix} u_0^m \\ u_1^m \\ \vdots \\ u_N^m \end{pmatrix} \quad (3.55)$$

é a componente de campo que se deseja determinar, sendo u_j^m referente a $u(x_m, z_j)$ no ponto $(x_m, z_j) = (m\Delta x, j\Delta z)$,

$$\mathbf{A}_m = \begin{pmatrix} \alpha_0^m & c & 0 & 0 & \cdots & 0 & 0 & 0 \\ c & \alpha_1^m & c & 0 & \cdots & 0 & 0 & 0 \\ 0 & c & \alpha_2^m & c & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & c & \alpha_{N-1}^m & c \\ 0 & 0 & 0 & 0 & \cdots & 0 & c & \alpha_N^m \end{pmatrix} \quad (3.56)$$

e

$$\mathbf{B}_m = \begin{pmatrix} \tilde{\alpha}_0^m & \tilde{c} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \tilde{c} & \tilde{\alpha}_1^m & \tilde{c} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \tilde{c} & \tilde{\alpha}_2^m & \tilde{c} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \tilde{c} & \tilde{\alpha}_{N-1}^m & \tilde{c} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \tilde{c} & \tilde{\alpha}_N^m \end{pmatrix} \quad (3.57)$$

A última equação pode ser reescrita como $\mathbf{B}_m = \tilde{\mathbf{A}}_m$. Além disso, considera-se α_j^m e c , definidos conforme as Equações (3.58) a (3.62).

$$\alpha_j^m = \begin{cases} D_j^m/2 + ikc\eta_m\Delta z & \text{for } j = 0 \\ D_j^m & \text{for } j = 1, 2, 3, \dots \end{cases} \quad (3.58)$$

$$D_j^m = c(a_j^m + b - 2) \quad (3.59)$$

$$c = 1 - ik\Delta x \quad (3.60)$$

$$a_j^m = k^2\Delta z^2 \left[\left(\frac{n_j^m + n_j^{m-1}}{2} \right)^2 - 1 \right] \quad (3.61)$$

$$b = \frac{4k^2\Delta z^2}{c} \quad (3.62)$$

em que:

- η_m refere-se a $\eta(x_{m-1})$ e está diretamente ligado às características do solo;
- n_j^m refere-se ao índice de refração $n(x_m, z_j)$;
- $\tilde{\alpha}_j^m$ pode ser escrito como na Equação (3.63).

$$\tilde{\alpha}_j^m = \begin{cases} \tilde{D}_j^m/2 + ik\tilde{c}\eta_m\Delta z & \text{for } j = 0 \\ \tilde{D}_j^m & \text{for } j = 1, 2, 3, \dots \end{cases} \quad (3.63)$$

3.8.2 WAPE para terreno irregular

O esquema de diferenças finitas pode iniciar da segunda linha, isto é, é descrita como em (3.64). A relação $\mathbf{B}_m = \tilde{\mathbf{A}}_m$ ainda é válida. Nesse caso, \mathbf{U}_m vai de u_1^m a u_N^m e não mais de u_0^m a u_N^m . Assim, após o esquema de diferenças, dado por (3.65) e (3.66), ser resolvido, o campo u_0^m do contorno é determinado por (3.67).

$$\mathbf{A}_m = \begin{pmatrix} \alpha_1^m & c & 0 & 0 & \cdots & 0 & 0 & 0 \\ c & \alpha_2^m & c & 0 & \cdots & 0 & 0 & 0 \\ 0 & c & \alpha_3^m & c & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & c & \alpha_{N-1}^m & c \\ 0 & 0 & 0 & 0 & \cdots & 0 & c & \alpha_N^m \end{pmatrix} \quad (3.64)$$

$$\alpha_j^m = \begin{cases} D_j^m + \frac{c}{1 - \kappa_m + \kappa_m^2/2} & \text{for } j = 1 \\ D_j^m & \text{for } j = 2, 3, 4 \dots \end{cases} \quad (3.65)$$

em que:

$$\kappa_m = ik_0 \Delta z \left(\eta_m + S_m^3/2 \right) \approx ik\eta_m \Delta z \quad (3.66)$$

$$u_0^m = \frac{u_1^m}{1 - \kappa_m + \kappa_m^2/2} \quad (3.67)$$

3.9 Perda de percurso e fator de propagação

Para aplicações de comunicação sem fio, um dos resultados mais importantes e desejados é a perda de percurso. Por isso, a resolução da equação parabólica em termos dos campos elétrico ou magnético ou da componente de campo ψ ou u deve ser convertida para uma perda de percurso equivalente. A perda de transmissão de um enlace de rádio em um ambiente é definida pela ITU como a razão entre a potência radiada pela antena transmissora e a potência disponível na antena receptora, considerando que não há perdas adicionais no circuito de comunicação. Além disso, é interessante desacoplar o efeito da propagação dos efeitos da antena, como na Equação (3.68) (VASCONCELOS, 2017).

$$L_{total} = L_B - G_T - G_R \quad (3.68)$$

em que G_T e G_R são os ganhos das antenas transmissora e receptora, respectivamente.

A perda básica pode ser calculada como na Equação (3.69) que se desenvolve em (3.70), uma vez que se assume que a distância entre os terminais é grande comparada ao comprimento de onda λ e a propagação ocorre no espaço livre, pois assim considera-se o fator de perda no espaço livre da equação de Friis - $\left(\frac{\lambda}{4\pi R}\right)^2$.

$$L_B = -10 \log_{10} \left(\frac{\lambda}{4\pi R} \right)^2 = 20 \log_{10} \left(\frac{4\pi R}{\lambda} \right) \quad (3.69)$$

$$L_B(x, z) = 20 \log_{10}(4\pi) + 20 \log_{10}(R) - 20 \log_{10}(\lambda) \quad (3.70)$$

$$PF(x, z) = L_B(x, z) - L_P(x, z) \quad (3.71)$$

Assim, Vasconcelos (2017) prova que tanto para polarização horizontal quanto para polarização vertical são válidas as Equações (3.72), para a perda de percurso, e (3.73) para o fator de propagação, obtida pela substituição de (3.72) em (3.71).

$$\begin{aligned}
 L_P(x, z) = & -20 \log_{10} |u(x, z)| + 20 \log_{10}(4\pi) + \dots \\
 & \dots + 10 \log_{10} \left[a e^{\frac{z}{a}} \sin \left(\frac{x}{a} \right) \right] - 30 \log_{10}(\lambda_0)
 \end{aligned} \tag{3.72}$$

$$\begin{aligned}
 PF(x, z) = & 20 \log_{10} |u(x, z)| - 10 \log_{10} \left[a e^{\frac{z}{a}} \sin \left(\frac{x}{a} \right) \right] + \dots \\
 & \dots + 10 \log_{10}(\lambda_0) + 10 \log_{10}(x^2 + z^2)
 \end{aligned} \tag{3.73}$$

4 COMPUTAÇÃO PARALELA

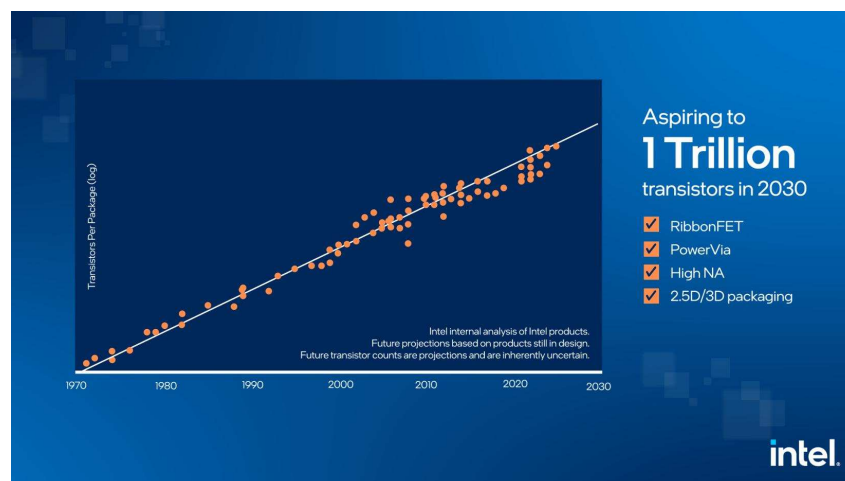
4.1 Introdução

Em 1965, o co-fundador da Intel, Gordon Earl Moore, elaborou um postulado que, baseando-se nos dados de evolução dos componentes e dispositivos eletrônicos da época, previa que o número de transistores em um circuito integrado de um processador dobraria a cada 18 meses. Defendia, assim, o investimento na miniaturização para viabilizar o desenvolvimento de novos e melhores computadores, como afirma [Moore \(1965\)](#):

Com o custo unitário caindo ao mesmo tempo que o número de componentes por circuitos aumenta, em 1975 a economia deve ditar a compressão de aproximadamente 65 000 componentes em um único chip de silício¹.

Como Moore previu, a Figura 10 confirma sua hipótese de 1965, uma vez que a própria Intel afirma que hoje existem chips com cerca de 100 Bilhões de transistores e que a estimativa é que em 2030 haja 1 Trilhão de transistores em um único chip. A área tecnológica está sempre buscando formas de evoluir, pois à medida que as dimensões físicas continuavam a encolher, a indústria adicionou novas áreas de inovação, incluindo ciência de materiais para acompanhar o ritmo.

Figura 10 – Lei de Moore, número de transistores por dispositivo: passado, presente, futuro.



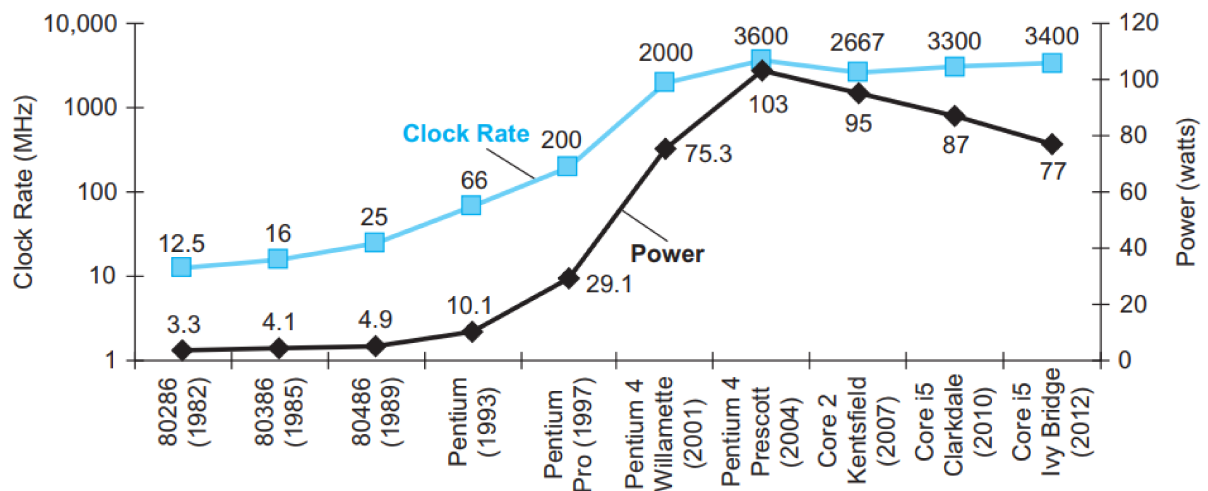
Fonte: [Kelleher \(2022\)](#).

¹ Citação original: “With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65 000 components on a single silicon chip.”

Assim, aparentemente, o desempenho, diretamente ligado à frequência de clock do processador, vem da evolução do *hardware*. No entanto, chegou um momento em que a quantidade de transistores era tão excessiva, que a fonte também precisaria acompanhar suas necessidades, levando o crescimento da potência necessária em uma ordem de potência de 1,75, como pode ser observado na Figura 11. Esse aumento de potência é principal desafio para o projeto de computadores, já que o circuito precisa refrigerado e para isso há um limite.

Para se ter ideia do cenário histórico mostrado, compara-se dois chips. O chip Intel 80386, fabricado em 1985, consumia cerca de 2 Watts. Já um modelo atual, o Intel Core i7 a um *clock* de 3,3 GHz consome 130 Watts (ABU-AMARA, 2019). Por isso, a importância de buscar desempenho melhorando também os *softwares*, e isso se faz aproveitando que, já que os computadores atuais possuem mais de um *core* de processador e mais de um processador lógico por *core*, é possível utilizar técnicas de processamento paralelo.

Figura 11 – A parede de potência.



Fonte: Abu-Amara (2019).

4.2 Processamento paralelo

Para se entender a importância de se paralelizar processos, nesse caso a nível de aplicação, define-se dois conceitos importantes:

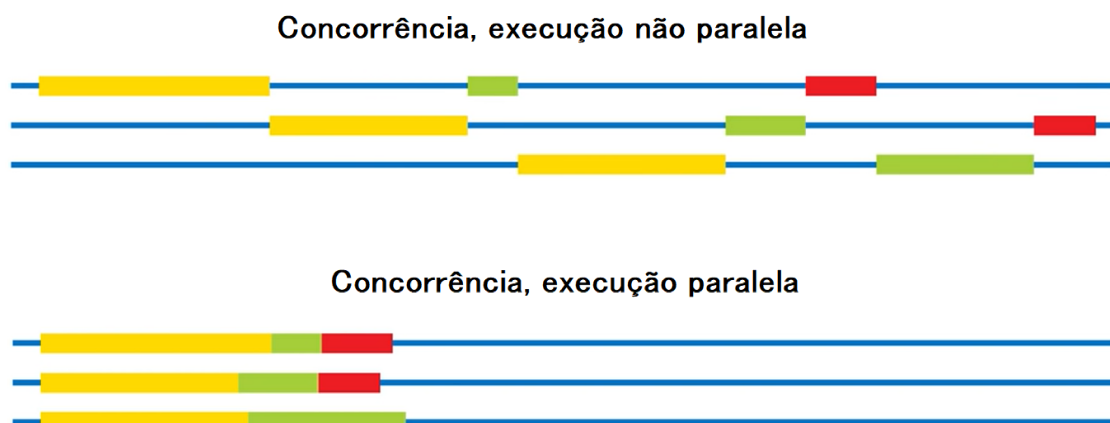
- **Concorrência:** um programa pode ser constituído por mais de uma tarefa concorrendo por recursos. Ligado ao *software*. É sobre lidar com várias tarefas ao mesmo tempo, porém a execução é sequencial.

- **Paralelismo:** uma aplicação é executada por um conjunto de processadores em um ambiente único (dedicado). Ligado ao *hardware*. É sobre fazer várias tarefas ao mesmo tempo, pois cada *core* do processador executa uma tarefa e fazem isso simultaneamente.

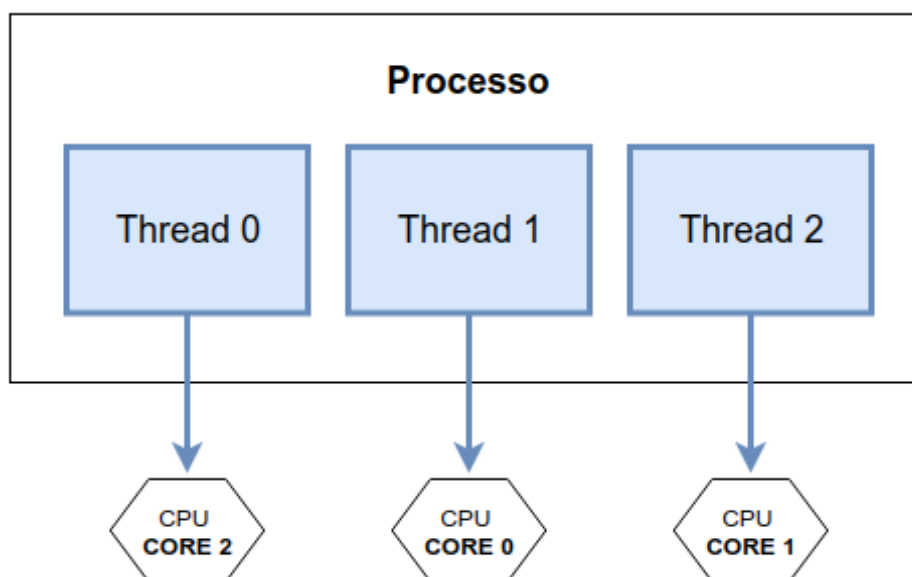
Assim, fazendo uma analogia, é como se na concorrência tivéssemos somente uma mesa de trabalho (recurso disponível) e cada trabalhador devesse esperar para usá-la, enquanto no paralelismo há mais de uma mesa, sendo possível que trabalhadores usem as mesas simultaneamente, portanto vão terminar suas respectivas tarefas mais rápido, o que significa redução de tempo de processamento. Ademais, é interessante definir uma *thread*: uma “linha” de execução dentro de um processo. Cada thread tem o seu próprio estado de processador e a sua própria pilha, mas compartilha a memória atribuída ao processo com as outras threads “irmãs” (filhas do mesmo processo).

Um exemplo é mostrado na Figura 12 e complementado pelo esquema da Figura 13. Nela, cada programa (amarelo, verde e vermelho) concorrem para ser executados por uma *thread*. No todo, no primeiro caso nenhum programa é computado ao mesmo tempo que outro, por isso diz-se que há concorrência, porém sem execução paralela. Já no segundo caso, o sistema não espera que uma tarefa seja computada para executar a próxima em outra *thread*, assim há concorrência e a execução das tarefas ocorrem paralelamente, reduzindo o tempo de processamento.

Figura 12 – Concorrência e paralelismo em processo de três *threads*.



Fonte: Autora.

Figura 13 – Exemplo de três *threads* rodando simultaneamente em três diferentes núcleos.

Fonte: [Tedesco \(2020\)](#).

Atualmente CPUs com mais de um *core* são comuns mesmo para os computadores portáteis de consumo básico. Isso mostra que o processamento paralelo não é apenas útil para a realização de cálculos avançados, mas que está se tornando comum em várias aplicações. Os ambientes de computação também estão cada vez mais diversificados, explorando as capacidades de uma gama de processadores multi-core, de unidades de processamento central (CPUs), de processadores de sinais digitais e hardware reconfigurável (FPGAs) e de unidades gráficas de processamento (GPUs). Essa heterogeneidade faz com que o processo de desenvolvimento eficiente de *software* apresente muitas ferramentas e arquiteturas, resultando em uma série de desafios para a comunidade da programação ([CORREA; SULZBACH, 2017](#)).

4.3 OpenMP

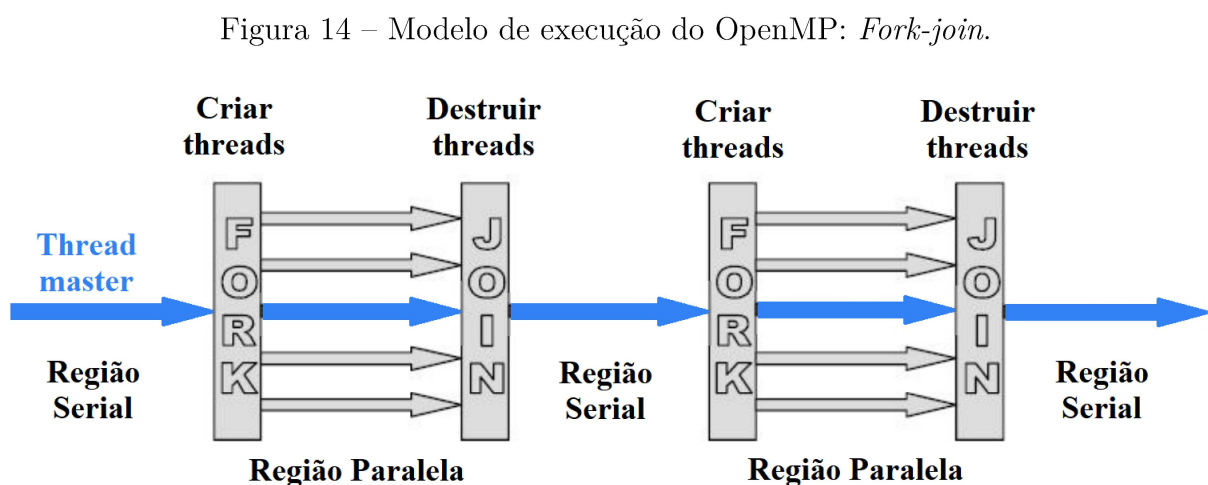
Uma das APIs que possibilitam a execução paralela é a OpenMP. Ela foi criada pela necessidade de padronização das diretivas para compartilhamento de memória. Cada fabricante fazia a sua própria, o que dificulta a portabilidade dos códigos. Só se justificam aqueles cujo método de paralelismo está intrinsecamente ligado ao potencial do *hardware* do fabricante, como no caso CUDA da NVIDIA. Assim, o fórum OpenMP foi iniciado pela Digital, IBM, Intel, KAI e SGI, e, atualmente, já inclui todos os grandes fabricantes. Então, o padrão OpenMP para Fortran foi liberado em outubro de 1997. Logo depois, o padrão OpenMP C/C++ foi liberado em outubro de 1998.

Essa API consiste em diretivas de compilação, rotinas de execução e variáveis de ambiente. Permite, assim, a utilização de um conjunto de diretivas que criam blocos de paralelização e distribuem o processamento entre os núcleos disponíveis. Entre suas vantagens estão:

- Bom desempenho e escalabilidade;
- Permite que o código seja paralelizado fácil e eficazmente;
- Portabilidade: suportado por vários compiladores e independe de fabricante;
- Usa memória compartilhada, assim cada *thread* pode ler e escrever em todas as posições de memória;
- Dados podem ser compartilhados ou privados;
- Há mecanismos de sincronização implícitos.

O modelo de execução do OpenMP é do tipo *Fork-Join* e suas etapas, ilustradas na Figura 14, são:

- inicia a execução com um processo (*thread master*);
- no início do construtor paralelo, cria um team-of-threads (*fork*);
- ao completar, o team-of-threads sincroniza numa barreira (*join*); implícita;
- apenas a *thread master* continua a execução.



Fonte: Adaptado de [Silva \(2012\)](#).

4.4 Principais diretivas do OpenMP

```
#pragma omp parallel <clause>
    bloco_codigo;
```

Indica que o `bloco_codigo` será executado em paralelo (SILVA, 2012), e as opções mais usadas para `<clause>` foram:

- `private(list)`, as variáveis da lista são privadas para cada *thread* do *host*.
- `firstprivate(list)`, permite que as variáveis privadas sejam inicializadas.
- `shared(list)`, as variáveis da lista são compartilhadas ou comuns para todas as *threads*. Por *default* todas as variáveis são *shared*.

A quantidade de *threads* que a região paralela utilizará pode ser especificada ou deixada como *default*, por meio de:

- cláusula `num_threads(int)`
- função `omp_set_num_threads()`
- variável de ambiente `OMP_NUM_THREADS`
- *default*, o omp distribui as threads entre a quantidade total de tarefas ou conforme as regras implementadas pelo programador

Além disso, o programador deve estabelecer regras de divisão do trabalho (conjunto de tarefas repetitivas) entre as *threads*, podendo utilizar os seguintes tipos de partilha:

- **for**, partilha as iterações de um ciclo pelos *threads* do time (*data parallelism*).
- **sections**, divide o trabalho em seções discretas, distintas, que são executadas pelos *threads*.
- **single** Serializa o código.

Como o problema da propagação é excessivamente iterativo devido às operações entre matrizes de alta ordem, as abordagens mais utilizadas são as mostradas abaixo:

EXEMPLO 1.

```
N = 8;
num_threads = 4;

#pragma omp parallel for private(f)
  for(i=0; i<N; i++){
    f = (c[i] - b[i]) * 0.5;
    a[i] = b[i]/c[i] * f;
  }
```

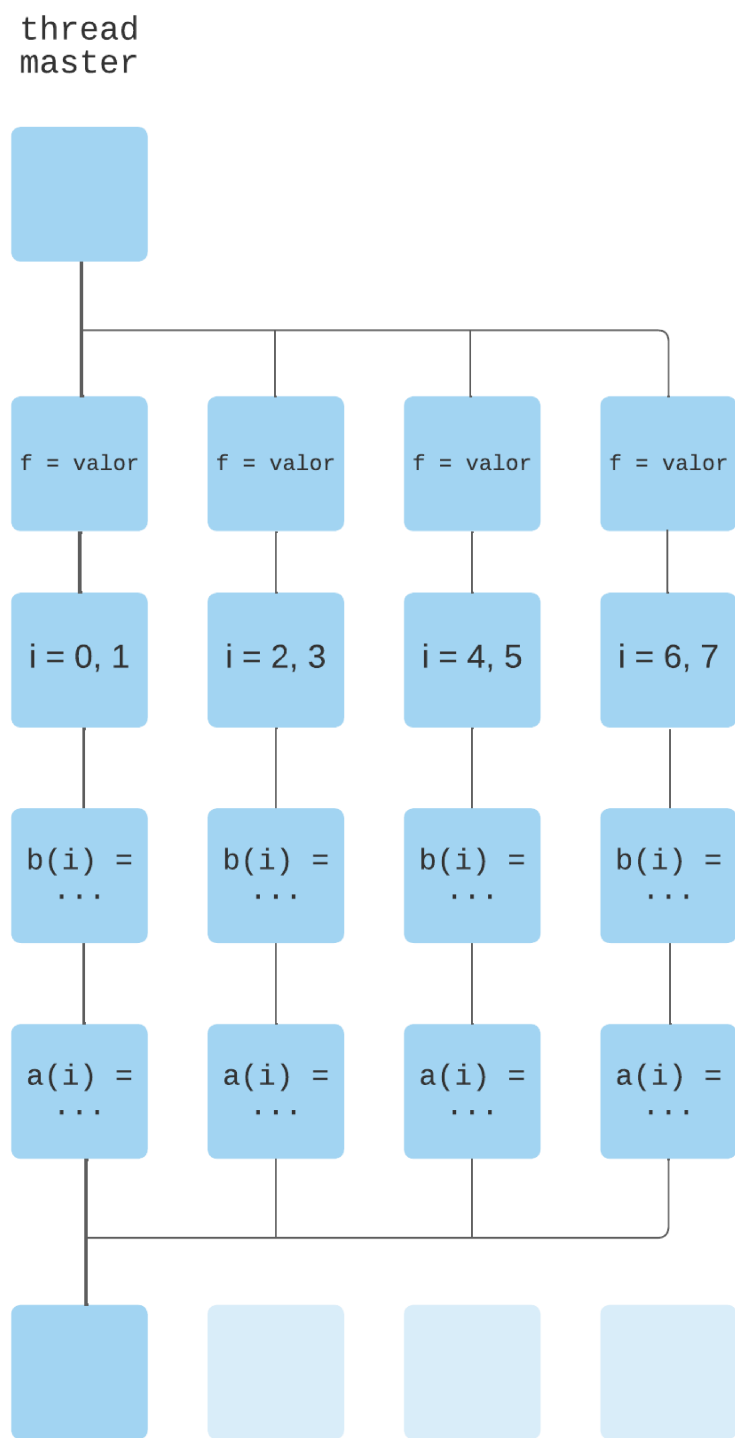
Nesse exemplo, é aberto um bloco paralelo em que suas *threads* são organizadas no modo *default* de modo a partilhar as iterações entre elas. A variável auxiliar f foi declarada como privada já que cada *thread* modificará seu valor de acordo com os próprios valores de i das iterações de que foi encarregada.

EXEMPLO 2.

```
#pragma omp parallel private(f)
{
  f = valor;
#pragma omp for
  for(i=0; i<N; i++){
    b[i] = b[i] + f;
    a[i] = b[i]/c[i];
  }
}
```

Já neste exemplo, o f mantém o mesmo valor em todas as iterações, mas essa abordagem como mostrada na seção de desenvolvimento deste trabalho pode ser usada para a determinação do valor máximo de um vetor. Um esquema mais intuitivo sobre a distribuição das *threads* é mostrado na Figura 15.

Figura 15 – Esquema da distribuição do trabalho entre as *threads*.



Fonte: Autora.

5 METODOLOGIA PROPOSTA

5.1 Introdução

Neste capítulo são apresentadas as etapas de implementação do método de equações parabólicas aliado ao método de resolução de diferenças finitas. Esse procedimento ocorreu seguindo as etapas:

1. Implementação computacional em C++ da referência disponibilizada por [Vasconcelos \(2017\)](#) (escrita em MATLAB), para a propagação do sinal em terreno com e sem relevo;
2. Validação dos resultados numéricos obtidos pelo programa C++, por meio da comparação com os resultados da referência em MATLAB;
3. Inserção de diretivas OpenMP para processar as tarefas iterativas paralelamente;
4. Criação de estratégias e adaptação do código para tornar alguns cálculos passíveis de ser executados paralelamente;

5.2 Metodologia de implementação no código

Para a implementação do método de equações parabólicas com resolução pelo método de diferenças finitas, descrito no Capítulo 3, desenvolveu-se conforme a seguinte metodologia:

1. **Entrada de parâmetros:** os parâmetros entradas são inseridos alterando o arquivo *parametros.json*. Foi escolhido esse tipo de entrada pela facilidade de ler e modificar esse tipo de arquivo, um exemplo de entrada é mostrado no Apêndice A.1. Como o objetivo do projeto total é a comparação das três APIs: OpenMP, OpenCL e CUDA, esse arquivo foi padronizado para que uma entrada gerasse a mesma saída para os três casos.
2. **Conversão de unidades:** é conveniente converter os parâmetros de entrada para o Sistema Internacional de Unidades (SI).
3. **Preparar o terreno:** caso o usuário selecione o modelo com relevo (denotado por **TERRA_COM_RELEVO**), então o arquivo do terreno é lido, e o terreno é interpolado de acordo com os parâmetros de passo de distância fornecidos pelo usuário.

4. **Modo de Otimização:** esse parâmetro deve ser avaliado para que o domínio de altura seja recalculado, se a escolha do usuário implique em sua diminuição ou aumento. Esse parâmetro \mathbf{N} é utilizado todas as outras etapas do cálculo, por isso, é importante redefini-lo agora. Isso porque para o algoritmo *Parallel Cyclic Reduction* (PCR) resolver o sistema linear contendo a matriz tridiagonal de alta ordem de forma paralela, ele exige que o sistema seja da ordem $2k - 1$, sendo k um número inteiro maior que 2.
5. **Janela absorvedora:** para simular a condição de Sommerfeld no infinito, é calculada quanto o domínio deve se estender para que usando o filtro de Hanning o sinal incidente na fronteira superior seja totalmente absorvido na simulação. Esse cálculo depende da frequência de operação da antena transmissora e, conforme Vasconcelos (2017), é implementado como no Apêndice A.2.
6. **Cálculo de α_{BC} :** é um parâmetro resultante da modelagem da fronteira inferior de impedância do solo, em que dada a especificação do tipo do solo em cada segmento, são calculadas as constantes do solo e então o parâmetro α_{BC} .
7. **Campo inicial:** é gerado a partir de três fontes à escolha do usuário: fonte gaussiana, fonte *rect*, fonte omnidirecional (*sinc*) (VASCONCELOS, 2017).
8. **Propagar:** São aproveitados laços iterativos em comum para o cálculo do índice de refração modificado ocorra simultaneamente ao cálculo das diagonais principais das matrizes tridiagonais A e B .
 - **Aplica *phase shift* no campo:** para incorporar as variações do terreno no cálculo.
 - **Aplica condições de contorno:** considerando a impedância do solo, é calculado os valores do limite inferior (primeira linha da matriz).
 - **Reconstrução das diagonais inferior e superior da matriz A :** a otimização usando o algoritmo PCR provoca alteração de todos os valores da matriz. A diagonal principal já é calculada levando em consideração o terreno. Já as diagonais inferior e superior são reatribuídas para evitar a propagação do erro.
 - **Multiplicação da matriz tridiagonal:** o primeiro passo para “marchar” a solução $\mathbf{A}_m \mathbf{U}_m = \mathbf{B}_m \mathbf{U}_{m-1}$, é calcular $V = B * U_0$, em que U_0 representa U_{m-1} .
 - **Resolve o sistema linear tridiagonal:** o segundo passo é resolver a $U_1 = A \setminus V$ para determinar U_m , representado por U_1 .
 - **Aplica o *phase shift* reverso e multiplica pela janela de Hanning:** retorna ao sistema de coordenadas cartesiano e aplica a janela de Hanning para obter o efeito da condição de Sommerfeld no infinito.

- **Exportar módulo de $u(x, z)$:** em termos de u , podem ser calculados e plotados os gráficos da perda de percurso e fator de propagação, por meio das Equações (3.72) e (3.73), respectivamente.

5.3 Passo de distância e altura *default*

Os passo *default* faz com que $delz$, o passo de altura, seja atribuído $\lambda_0/4$, e $delx$, o passo de distância seja calculado para haver melhor representação do terreno considerando a frequência da fonte. Isto é descrito no código do Apêndice A.5.

5.4 Preparar o terreno

Preparar o terreno para os cálculos significa adaptá-lo para colher seus valores nos pontos múltiplos de Δx , para isso faz-se a interpolação para determinação nesses pontos, e aproveita-se para guardar as inclinações do terreno que serão úteis à frente. Por isso, a interpolação é feita como no código do Apêndice A.4 e a derivada é calculada, pelo método de diferenças central, no código do Apêndice A.6.

5.5 Perfis de refração

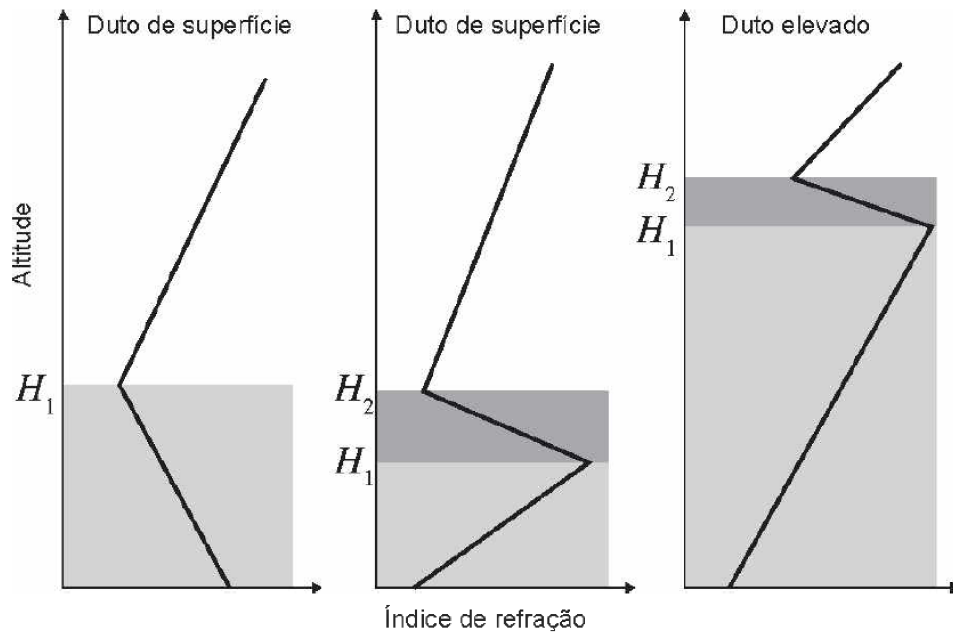
Para simular a atmosfera foram disponibilizadas cinco opções de perfil de refração para o usuário: Atmosfera padrão, Linear, Bilinear, Trilinear e Exponencial. O perfil da atmosfera padrão é um caso particular do perfil exponencial, dado pela Equação (5.1), em que a refratividade a nível do mar é $N_0 = 315 \times 10^{-6}$ N-unidades e altitude de escala é $1/h_0 = -0.136 \times 1e - 3$. Na Figura 16 estão representados os perfis dos casos bilinear e trilinear, assim, matematicamente, um perfil trilinear genérico é dado pela Equação (5.2).

$$n(z) = N_0 e^{\frac{z}{h_0}} + 1; \quad (5.1)$$

$$n(z) = \begin{cases} \frac{\partial n}{\partial z_{(1)}} z + n_0 & , z \leq H_1 \\ \frac{\partial n}{\partial z_{(2)}} z + H_1 \left(\frac{\partial n}{\partial z_{(1)}} - \frac{\partial n}{\partial z_{(2)}} \right) + n_0 & , H_1 < z \leq H_2 \\ \frac{\partial n}{\partial z_{(3)}} z + H_2 \left(\frac{\partial n}{\partial z_{(2)}} - \frac{\partial n}{\partial z_{(3)}} \right) + H_1 \left(\frac{\partial n}{\partial z_{(1)}} - \frac{\partial n}{\partial z_{(2)}} \right) + n_0 & , z > H_2 \end{cases} \quad (5.2)$$

em que n_0 é o índice de refração na superfície.

Figura 16 – Perfis de refração.



Fonte: Sevgi (2014).

5.6 Propagar

Para “marchar” a solução de propagação foram utilizadas algumas estratégias para otimizar o tempo de processamento, aproveitando que \mathbf{A}_m e \mathbf{B}_m são matrizes tridiagonais.

5.6.1 Multiplicação da matriz tridiagonal

Uma matriz tridiagonal é uma matriz quadrada cujos únicos elementos não nulos estão na diagonal principal e nas diagonais imediatamente acima e abaixo da principal, como a matriz do sistema da Equação (5.3). Assim, é possível fazer uma simplificação ao algoritmo de multiplicação de matrizes, resultando no código do Apêndice A.7.

$$\begin{bmatrix}
 b_1 & c_1 & & & & \\
 a_2 & b_2 & c_2 & & & \\
 & a_3 & b_3 & \ddots & & \\
 & & \ddots & \ddots & c_{n-1} & \\
 & & & a_n & b_n &
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 \vdots \\
 x_n
 \end{bmatrix}
 =
 \begin{bmatrix}
 d_1 \\
 d_2 \\
 d_3 \\
 \vdots \\
 d_n
 \end{bmatrix}
 \tag{5.3}$$

5.6.2 Algoritmo de Thomas

O método da matriz tridiagonal ou algoritmo de Thomas ou ainda TDMA (do inglês *tridiagonal matrix algorithm*) é o caso particular da eliminação gaussiana aplicada a matrizes tridiagonais. Esse algoritmo foi o primeiro a ser implementado e possui a desvantagem de que não pode ser paralelizado, como visto no código do Apêndice A.8.

O sistema da Equação (5.3) é dito tridiagonal, uma vez que é um sistema de equações lineares cuja matriz associada é tridiagonal. Por conveniência, a partir daqui, definiremos os elementos inexistentes na matriz a_1 e c_n como zero: $a_1 = c_n = 0$. Assim, o algoritmo para a solução do sistema tridiagonal (5.3) pelo algoritmo de Thomas é dada pelas expressões das Equações (5.4) e (5.5).

$$c'_i = \begin{cases} \frac{c_i}{b_i}, & i = 1 \\ \frac{a_i}{b_i - a_i d'_{i-1}}, & i = 2, 3, \dots, n-1 \end{cases} \quad (5.4)$$

$$d'_i = \begin{cases} \frac{d_i}{b_i}, & i = 1 \\ \frac{d_i - a_i d'_{i-1}}{b_i - a_i c'_{i-1}}, & i = 2, 3, \dots, n. \end{cases} \quad (5.5)$$

Finalmente a solução final é obtida por substituição reversa mostradas nas Equações (5.6) e (5.7).

$$x_n = d'_n \quad (5.6)$$

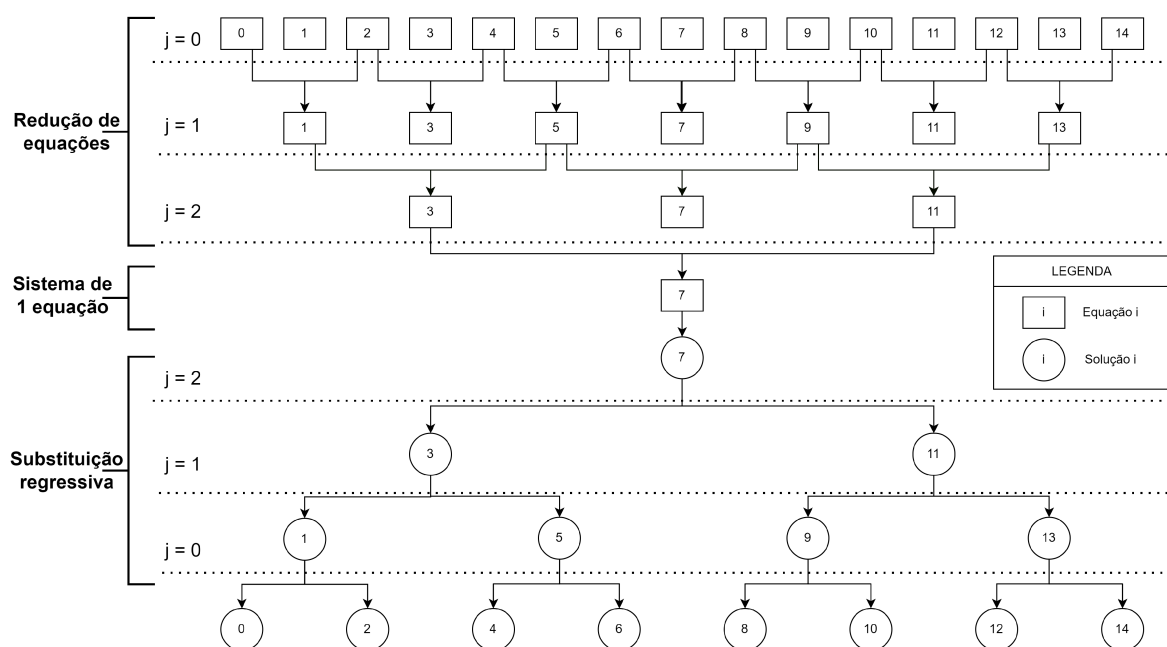
$$x_i = d'_i - c'_i x_{i+1}, \quad i = n-1, n-2, \dots, 1. \quad (5.7)$$

5.6.3 Redução Cíclica Paralela

Para paralelizar a resolução do sistema tridiagonal optou-se pelo algoritmo de Redução Cíclica (*Cyclic Reduction* - CR), cuja forma paralela é dita Redução Cíclica Paralela (*Parallel Cyclic Reduction* - PCR), implementado no código do Apêndice A.9. CR consiste em duas fases: compressão progressiva, então o valor de uma variável passa a ser conhecida, e substituição regressiva.

Um exemplo é apresentado na Figura 17. Na fase de compressão progressiva, o sistema é sucessivamente reduzido para um sistema menor com metade do número de incógnitas, até que um sistema de 2 incógnitas seja alcançado e seja resolvido trivialmente. Na fase de substituição regressiva, a outra metade das incógnitas é encontrada por substituindo os valores encontrados anteriormente nas equações (TAVAKKOL, 2019).

Figura 17 – Algoritmo de redução cíclica.



Fonte: Elaborado por Henrique Santos de Lima (2023).

6 RESULTADOS

6.1 Introdução

Neste capítulo, são realizados testes comparativos mostrando o tempo de processamento para execução do código MATLAB, usado como referência, e do implementado em C++ com paralelismo de tarefas usando OpenMP. O objetivo é mostrar o processamento em cenários distintos, variando os parâmetros de entrada do Apêndice A.1. Assim, com o intuito de apresentar o desempenho no cálculo do método, é utilizado o salvamento discreto dos dados resultantes. O salvamento discreto é mais rápido e poupa o uso de memória, permitindo trabalhar com matrizes muito maiores, ou seja, em cenários de grandes distâncias.

Sobre o dispositivo em que os testes ocorrem, têm-se as seguintes especificações:

- Processador: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz, 1992 Mhz, 4 Core(s), 8 Processadores Lógicos
- Memória física instalada (RAM): 16 GB
- Processador gráfico: NVIDIA GeForce MX150

Vale ressaltar que apesar do notebook *Samsung Style S51* possuir placa de vídeo, o OpenMP não acessa ela, já que as diretivas usadas são específicas para processamento paralelo no processador. Além disso, sobre os testes a serem realizados a seguir, o salvamento poderá ser Completo (C) ou Discreto (D), a polarização poderá ser Horizontal (H) ou Vertical (V), o erro máximo é medido com relação ao módulo campo $u(x, z)$ e padronizam-se alguns parâmetros:

- Altura máxima acima do terreno [m]: 500,
- Passo de distância paraxial [m]: 0, (ou seja, *DEFAULT*)
- Passo de altitude [m]: 0, (ou seja, *DEFAULT*)
- Método de Otimização da PCR: 1, (**AUMENTAR_N**)
- Fonte:
 - Tipo da fonte: 0, (**FONTE_GAUSSIANA**)
 - Largura de Feixe de 3dB das fontes [graus]: [2],

- *Beam tilt* das fontes [graus]": [1],
- Altura das fontes [m] acima do solo: [125]
- Refração:
 - Perfil: 4, (**EXPONENCIAL**)
 - Refratividade a nível do mar: 315,
 - Altura da escala da refratividade: -0.136
- Solo: tipo 5 (**SOLO_SECO**) em toda a extensão do terreno
- Para o salvamento discreto:
 - Passo de altura salva [m]: -1000, (que significa 1000 pontos)
 - Passo de distância salva [m]": -1000, (que significa 1000 pontos)

6.2 Teste 1: Sem relevo

Considerando o modelo sem relevo **TERRA_SEM_RELEVO_2**, tem-se os resultados da Tabela 3 e a Figura 18, em que:

- M: 334,
- N: 8192,
- Nuser: 1024,
- delx: 2.99792 m,
- delz: 0.749481 m

Observa-se que como a quantidade de pontos do caso completo (1024 x 334) é semelhante à quantidade de pontos salva (1000 x 333), o desempenho dos salvamentos completo e discreto são parecidos, sendo que para o salvamento discreto há uma etapa de processamento a mais. Apesar disso, será mostrado adiante como o salvamento discreto é eficaz para grandes distâncias, já matrizes completas do OpenMP para grandes distância são rejeitadas pelo MATLAB que reclama *Out of memory*, ou seja, o processamento da matriz de alta ordem não é possível devido a uma limitação de memória do dispositivo.

Esse teste objetiva-se testar a estabilidade da simulação sem relevo, logo comprova-se que a solução apresenta um erro insignificante na ordem de 10^{-6} , resultante do erro de ponto flutuante para dados do tipo *double*. Além disso, a eficiência do código OpenMP

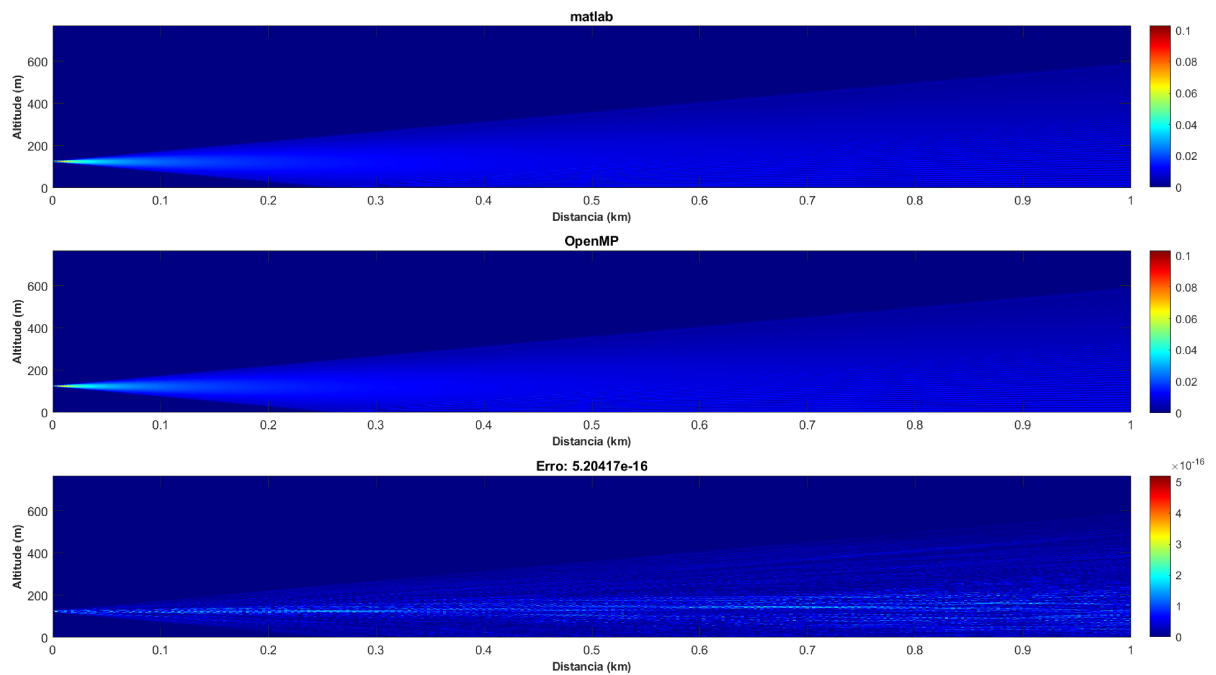
medida como a razão do tempo de processamento do MATLAB e do tempo de processamento do OpenMP indicou que em média o OpenMP foi 10,775 vezes mais rápido que o MATLAB.

Tabela 3 – Resultados do teste: sem relevo

Salv. (C/D)	Frequência (MHz)	Pol. (H/V)	Distância (m)	Tempo (s) MATLAB	Tempo (s) OpenMP	Eficiência OpenMP	Erro Máximo
C	100	H	1000	1.1472	0.0973	11.77	5e-16
D	100	H	1000	1.0306	0.0976	10.55	5e-16
C	100	V	1000	1.0354	0.0982	10.54	5e-16
D	100	V	1000	1.0368	0.1012	10.24	5e-16

Fonte: Autora.

Figura 18 – Campo $u(x, z)$ em terreno sem relevo: resultados em MATLAB (referência) e em OpenMP, e o erro absoluto entre eles, sendo 10^{-16} a ordem de ponto flutuante.



Fonte: Autora.

6.3 Teste 2: Com relevo

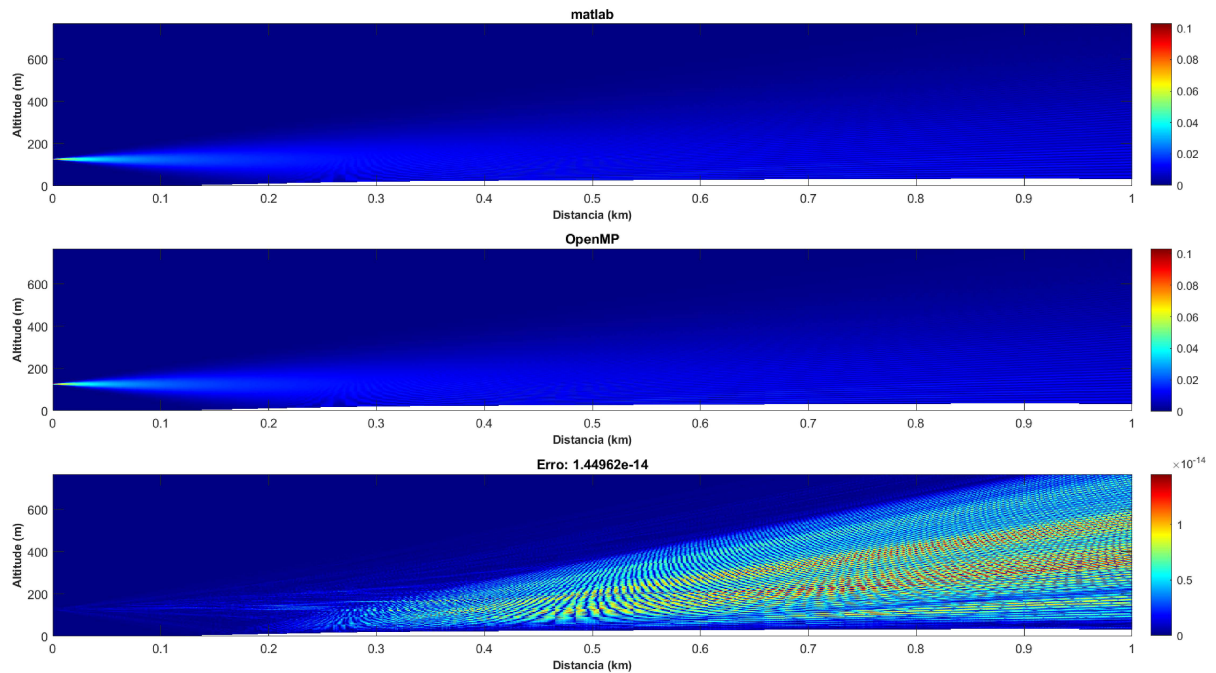
No teste com relevo, permanece a estabilidade da solução como observado no erro máximo da Tabela 4 e Figura 19, além de que em média o OpenMP foi 6,73 vezes mais rápido que MATLAB. Percebe-se que a eficiência reduziu um pouco devido às operações de *phase shift* para adaptação às variações do terreno, já que é necessário calcular cos e sin da relação de Euler, $e^{j\theta} = \cos \theta + j \sin \theta$.

Tabela 4 – Resultados do teste: com relevo

Salv. (C/D)	Frequência (MHz)	Pol. (H/V)	Distância (m)	Tempo (s) MATLAB	Tempo (s) OpenMP	Eficiência OpenMP	Erro Máximo
C	100	H	1000	8.0776	1.1457	7.04	1e-14
D	100	H	1000	7.7869	1.2376	6.29	1e-14
C	100	V	1000	7.8282	1.2124	6.45	7e-15
D	100	V	1000	7.8182	1.0949	7.14	7e-15

Fonte: Autora.

Figura 19 – Campo $u(x, z)$ em terreno com relevo: resultados em MATLAB (referência) e em OpenMP, e o erro absoluto entre eles, sendo que 10^{-16} a ordem de ponto flutuante.



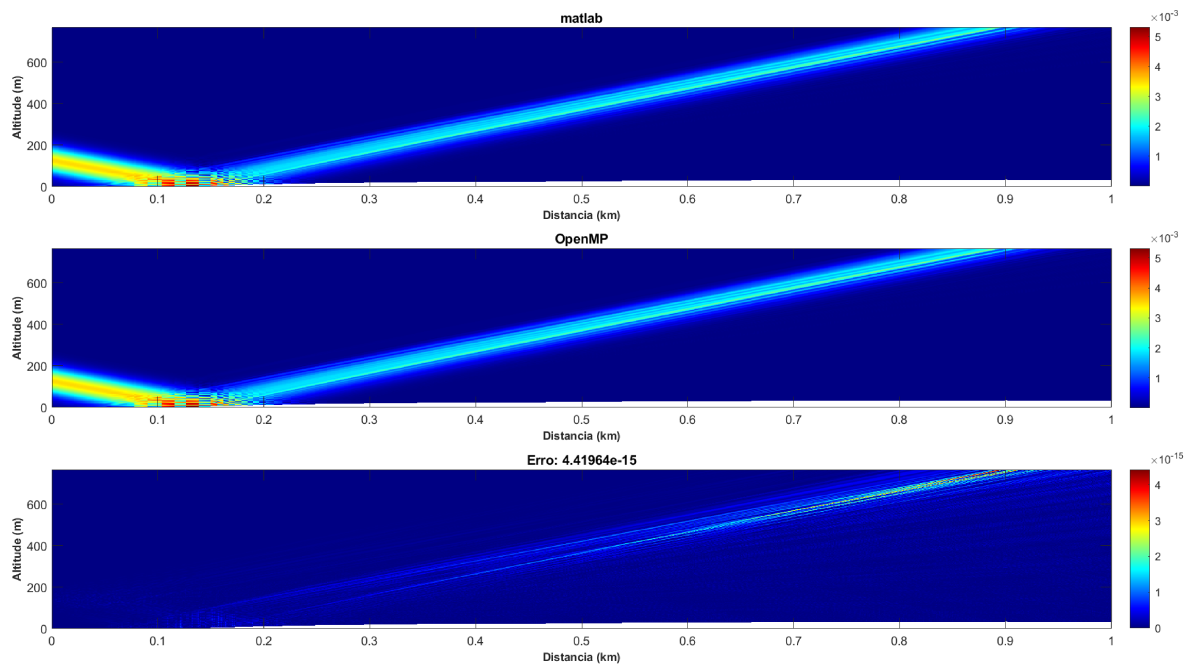
Fonte: Autora.

6.4 Teste 3: Fronteiras

As fronteiras superior e inferior devem fazer seu papel para evitar interferência de sinais inexistentes em um caso real e a propagação de erro, por isso realiza-se testes considerando uma antena diretiva com largura de feixe de 1° grau a uma inclinação de feixe de 85° graus. Recorda-se que o método utilizado representa bem ondas se propagando até 45° graus da direção paraxial. Então, este teste serve unicamente mostrar, nas Figuras 20 e 21, e Tabela 5, a estabilidade das condições de contorno aplicadas às fronteiras e não representam um caso real.

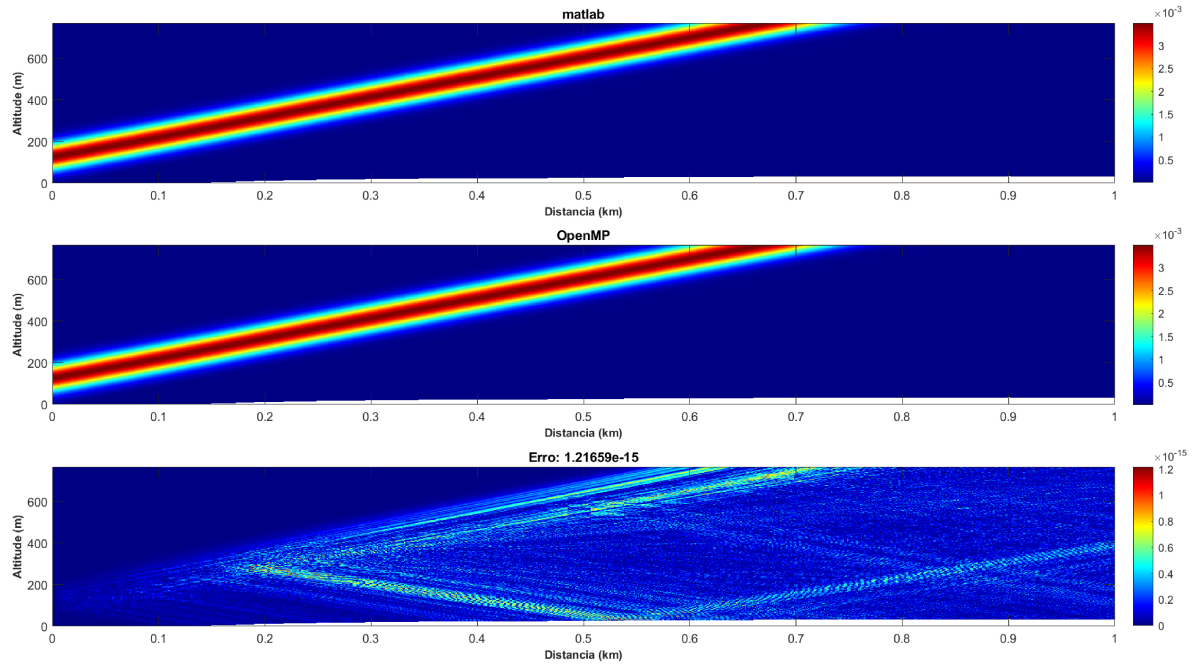
Isso porque quando ocorre uma reflexão no terreno, por exemplo, a onda refletida pode assumir uma direção com um ângulo maior que 45° e o campo passa a ser mal representado. Se a largura de feixe é muito grande, o campo contém muitas componentes fora da região válida, mas também tem a maioria das componentes na direção paraxial. Se o *tilt* (inclinação) é grande, a fonte vai criar as ondas em ângulos grandes também, aumentando o erro para com um caso de medições reais (VASCONCELOS, 2017).

Figura 20 – Campo $u(x, z)$ na fronteira inferior (solo): resultados em MATLAB (referência) e em OpenMP, e o erro absoluto entre eles, sendo 10^{-16} a ordem de ponto flutuante.



Fonte: Autora.

Figura 21 – Campo $u(x, z)$ na fronteira superior (troposférica): resultados em MATLAB (referência) e em OpenMP, e o erro absoluto entre eles, sendo 10^{-16} a ordem de ponto flutuante.



Fonte: Autora.

Tabela 5 – Resultados do teste: fronteiras

Ângulo (graus)	Frequência (MHz)	Pol. (H/V)	Distância (m)	Tempo (s) MATLAB	Tempo (s) OpenMP	Eficiência OpenMP	Erro Máximo
85	100	H	1000	8.0330	1.7449	4.60	4e-15
85	100	V	1000	8.0394	1.7402	4.61	3e-15
-85	100	H	1000	7.9990	1.7220	4.64	1e-15
-85	100	V	1000	7.9803	1.7184	4.64	1e-15

Fonte: Autora.

6.5 Teste 4: Longas distâncias e altas frequências

A Tabela 6 mostra os resultados para grandes distâncias e altas frequências, que resultam em matrizes de mais alta ordem do que as mostradas anteriormente. Os testes foram realizados para os dois modelos já mostrados: sem relevo (S) e com relevo (C). A estabilidade é mantida, sendo que a eficiência é maior no modelo sem relevo, uma vez que não realiza operações de *phase-shift* e *phase-shift* reverso.

Tabela 6 – Resultados do teste: longas distâncias, altas frequências.

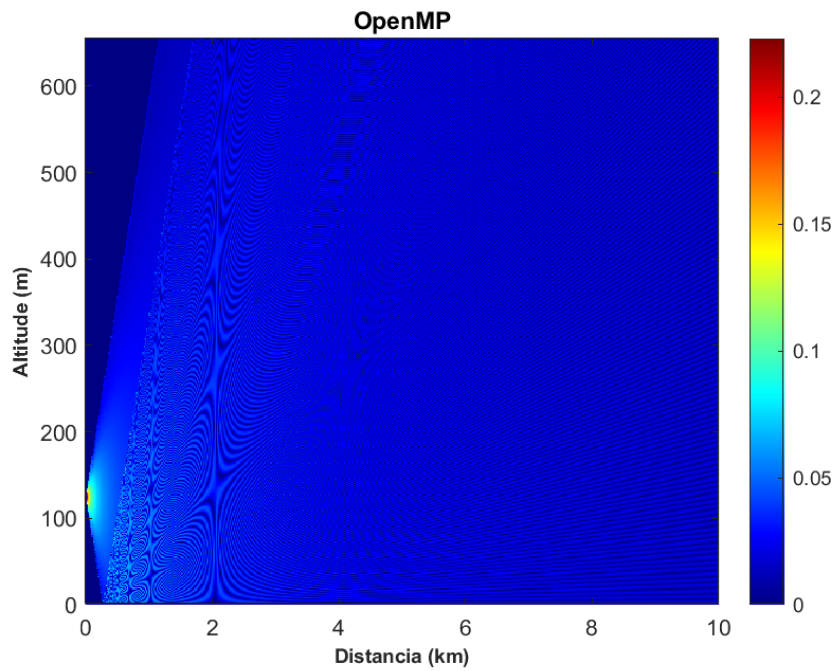
Modelo (S/C)	Frequência (MHz)	Pol. (H/V)	Distância (m)	Tempo (s) MATLAB	Tempo (s) OpenMP	Eficiência OpenMP	Erro Máximo
S	700	H	5000	106.90	15.09	7.08	2e-14
C	700	H	5000	2373.58	442.21	5.36	3e-12
C	700	H	10000	4762.14	894.06	5.32	7e-12
S	1875	H	10000	575.56	211.89	2.71	3e-14

Fonte: Autora

6.5.1 Resultados para terreno sem relevo

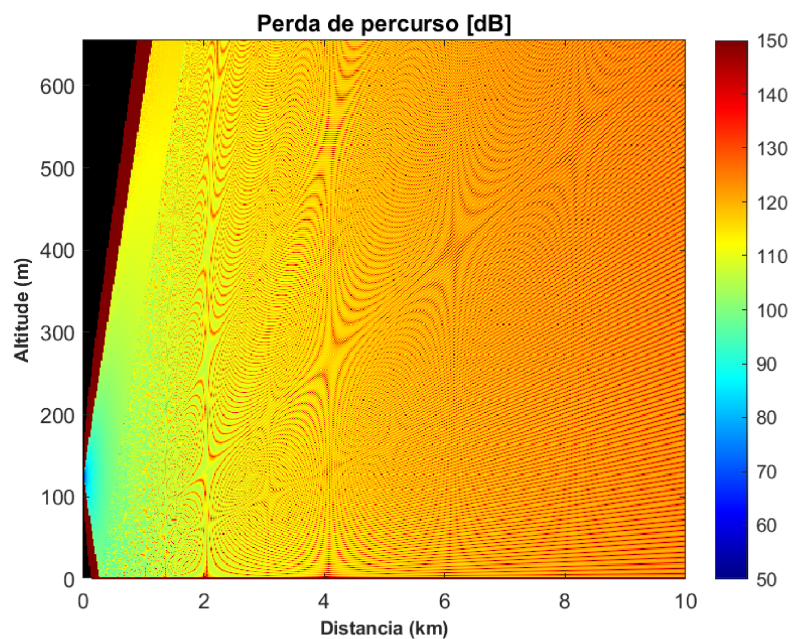
As Figuras 22 a 25 mostram, respectivamente, o módulo do campo $u(x, z)$, perda de percurso L_P , Fator de propagação FP e Perda de percurso L_P na máxima distância pedida pelo usuário, para o cenário sem relevo.

Figura 22 – Campo $u(x, z)$ para as condições especificadas pelo usuário.



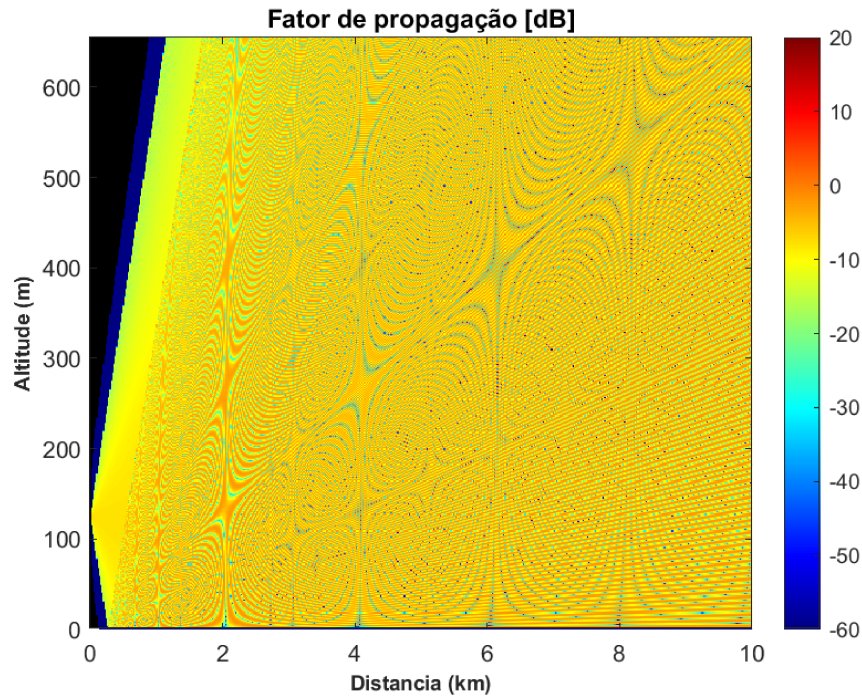
Fonte: Autora.

Figura 23 – Perda de percurso L_P para as condições especificadas pelo usuário.



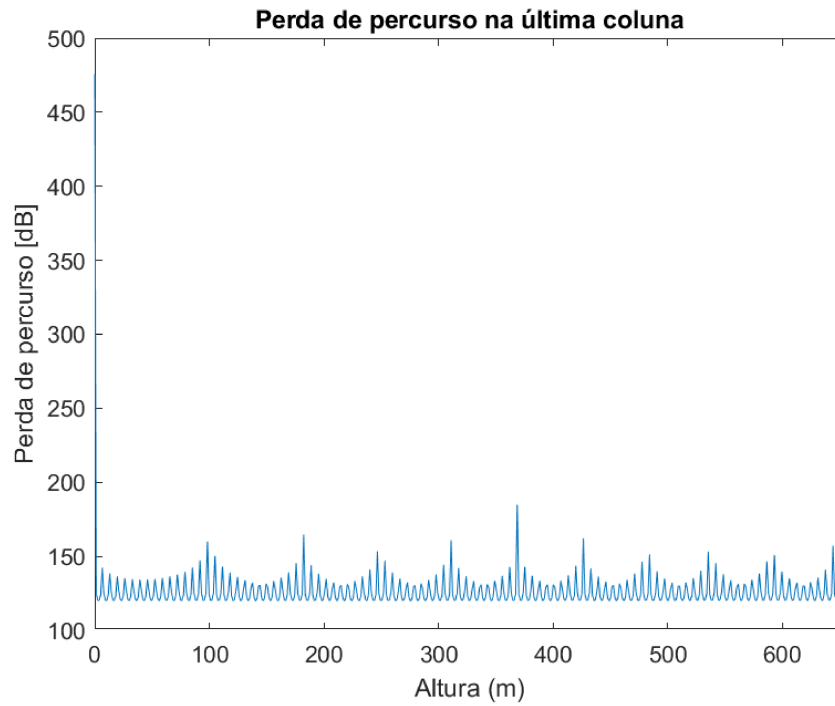
Fonte: Autora.

Figura 24 – Fator de propagação FP para as condições especificadas pelo usuário.



Fonte: Autora.

Figura 25 – Perda de percurso L_P na máxima distância pedida pelo usuário.

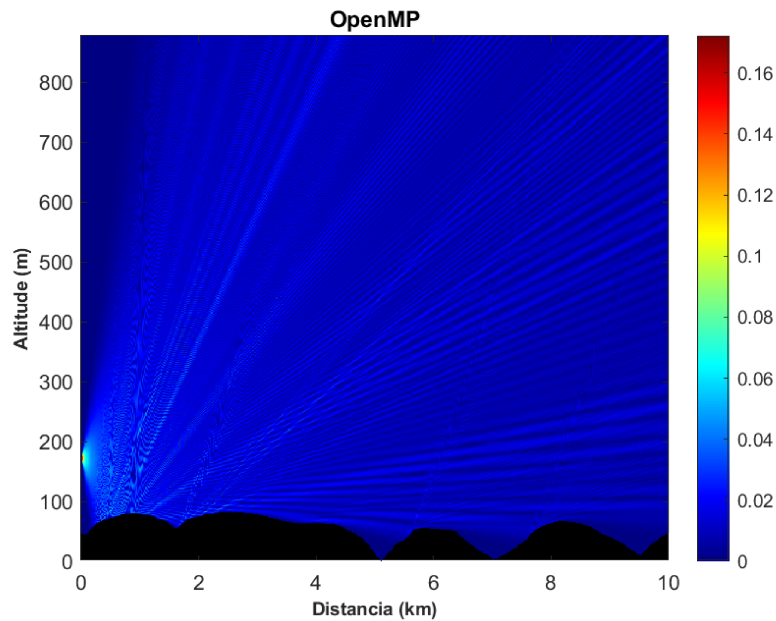


Fonte: Autora.

6.5.2 Resultados para terreno com relevo

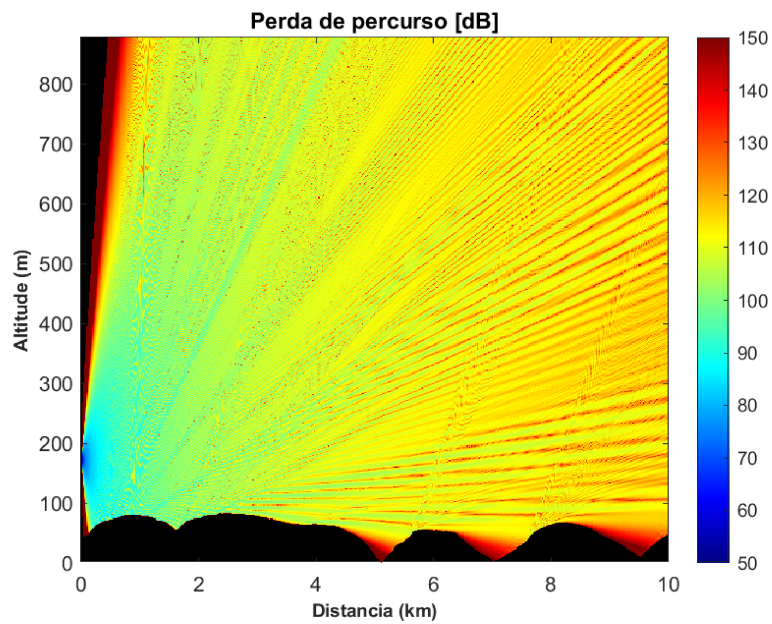
As Figuras 26 a 29 mostram, respectivamente, o módulo do campo $u(x, z)$, perda de percurso L_P , Fator de propagação FP e Perda de percurso L_P na máxima distância pedida pelo usuário, para o cenário com relevo.

Figura 26 – Campo $u(x, z)$ para as condições especificadas pelo usuário.



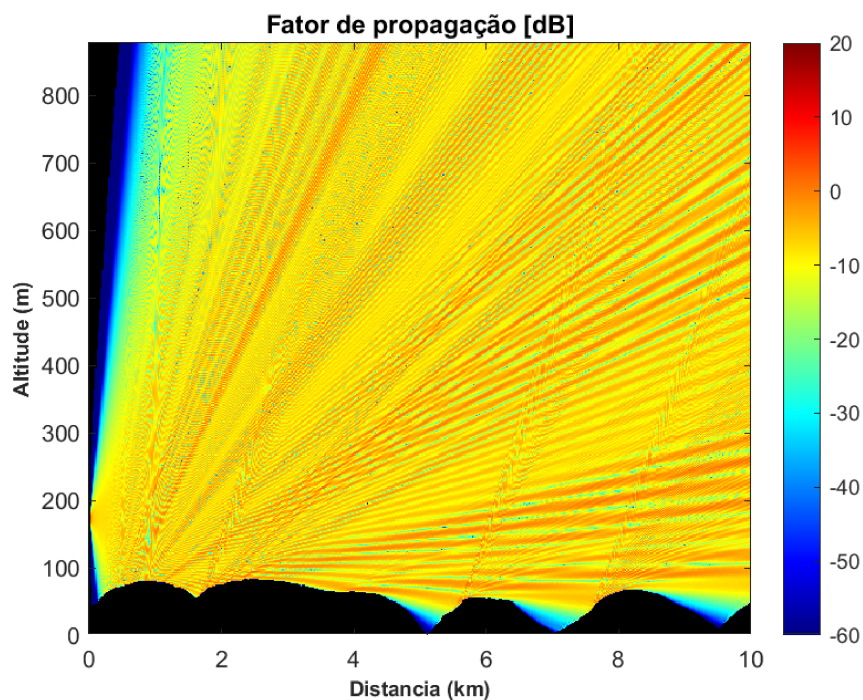
Fonte: Autora.

Figura 27 – Perda de percurso L_P para as condições especificadas pelo usuário.



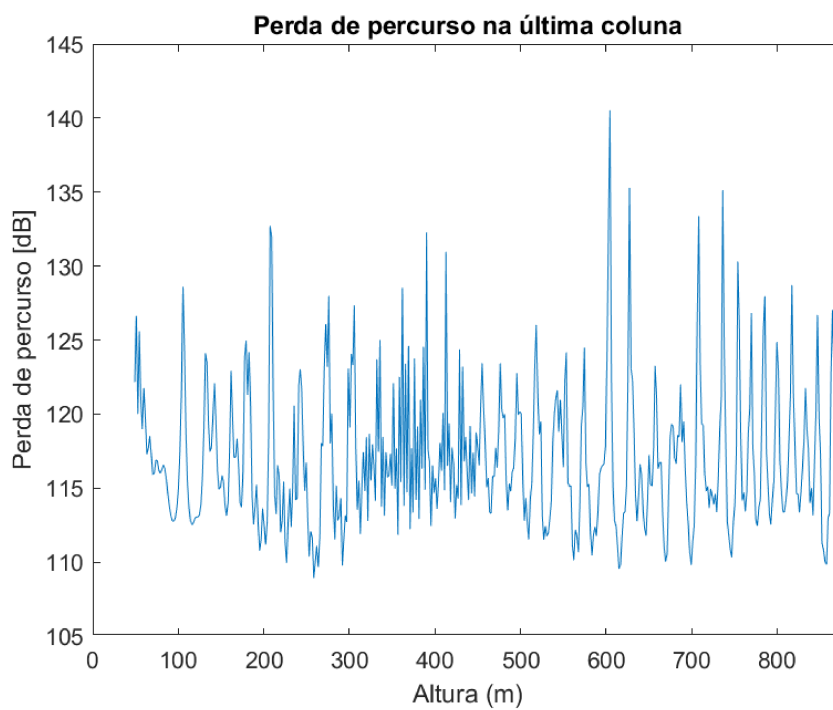
Fonte: Autora.

Figura 28 – Fator de propagação FP para as condições especificadas pelo usuário.



Fonte: Autora.

Figura 29 – Perda de percurso L_P na máxima distância pedida pelo usuário.



Fonte: Autora.

7 CONCLUSÕES

7.1 Conclusões

Os resultados mostram que a eficiência alcançada é aceitável, tendo em vista público alvo dessa aplicação. Uma vez que a principal vantagem do OpenMP sobre outras APIs de paralelização é a portabilidade, seu desempenho foi bem aproveitado. Assim, ao invés de realizar os cálculos da perda de percurso pelo método de equações parabólicas e aproximação computacional por diferenças finitas de forma sequencial, é mais proveitoso utilizar as *threads* comumente disponíveis nos dispositivos do mercado. Mesmo que hajam poucas *threads* disponíveis, o ganho de tempo já é significativo, sendo que o menor foi de 2,71 vezes, considerando um problema de matriz de altíssima ordem. Em minutos, esse caso foi de 9,58 min para 3,51 min. Além de que a implementação pode ser usada em servidores, em que a frequência de *clock* é maior e a quantidade de *threads* é muito maior que a do notebook utilizado nos testes.

É claro que é possível otimizar esse tempo, uma vez que aliado ao OpenMP podem se incorporar APIs de desempenho na GPU, que apesar de possuir menor frequência de *clock*, possui maior número de *threads* disponíveis. E esse foi escopo do projeto geral: comparar a utilização de três APIs no problema da perda de percurso, assim mesclar, se for vantajoso, a implementação delas no código. Já que os cálculos executados na GPU devem receber uma instrução da CPU ou dados vindo dela, para alguns desses cálculos é preferível que sejam executados pela CPU mesmo, evitando cópias que exijam mais esforço que a paralelização no processador, como é o caso do cálculo para determinação de valor máximo de um vetor. Além disso, a implementação da PCR nesse estudo de caso resultando em um erro absoluto próximo à ordem do erro de ponto flutuante faz com que o projeto também represente uma validação do esquema da PCR.

7.2 Trabalhos futuros

No desenvolvimento deste projeto e escrita deste trabalho foram pensadas em várias formas de otimizar o uso de memória e tempo de processamento, entre elas:

- Usar as *threads* da GPU por meio da API OpenMP, como apresentado por [Klemm \(2021\)](#).
- Possibilidade da OpenMP utilizar as *threads* da GPU automaticamente se a máquina do usuário tiver e o usuário optar por isso;

- Fazer redução cíclica do sistema para que atenda o requisito da PCR, que resolve apenas sistemas tridiagonais de ordem $2^k - 1$;
- Interatividade com mapas;
- Transição suave do índice de refração em função da distância para início de um novo tipo de solo;
- Simular a propagação em condições de chuva ou outras variações climáticas, dada a região geográfica do terreno analisado. Para assim prever quão será afeta a confiabilidade do enlace de comunicação;
- Problemas tridimensionais;
- Considerar o retroespalhamento, usando um método PE *2-way propagation* para satisfazer aplicações como radar e seção reta de alvos;
- Encontrar uma melhor solução computacional para truncamento de domínio na fronteira troposférica superior. Uma alternativa seria utilizar a *Perfectly Matched Layer* (PML) como apresentado por [Vasconcelos \(2017\)](#);
- Otimizar uso de memória usando somente o espaço de memória de uma matriz tridiagonal, uma vez $\mathbf{B}_m = \tilde{\mathbf{A}}_m$, significando que a maioria dos elementos a matriz \mathbf{B} é o conjugado de \mathbf{A} ;
- Realizar a multiplicação da matriz tridiagonal no mesmo escopo de resolução do sistema linear, ou seja, calcular diretamente $\mathbf{A}_m \mathbf{U}_m = \mathbf{B}_m \mathbf{U}_{m-1}$, dispensando a utilização da variável auxiliar V ;
- Fazer a modelagem de uma fonte real para representar o campo inicial, já que foram utilizados modelos de campos iniciais descritos analiticamente pela distribuição de Gauss, função *rect* e função *sinc* (fonte omnidirecional).

REFERÊNCIAS

- ABU-AMARA, M. *Performance COE 301 Computer Organization (slide 37)*. [S.l.], 2019. Disponível em: <<https://slideplayer.com/slide/14368741/>>. Acesso em: 15 jan. 2023. Citado na página 56.
- BALANIS, C. A. *Advanced Engineering Electromagnetics*. 2. ed. [S.l.]: John Wiley & Sons, 2012. 1018 p. Citado na página 34.
- CARMO, F. P. D. *A Equação de Poisson e a Decomposição de Helmholtz-Hodge com Operadores SPH: Decomposição de Helmholtz-Hodge*. [S.l.], 2008. Disponível em: <https://www.maxwell.vrac.puc-rio.br/12140/12140_6.PDF>. Acesso em: 24 jan. 2023. Citado na página 27.
- CELPLAN. *Ferramenta poderosa para planejamento, projeto e validação de redes sem fio*. 1992. Disponível em: <<https://celplan.com.br/solucoes/celplanner/>>. Acesso em: 26 jan. 2023. Citado na página 19.
- CHENG, D. K. *Field and Wave Electromagnetics*. [S.l.]: Tsinghua University Press, 1989. Citado 2 vezes nas páginas 24 e 25.
- CORREA, G. R.; SULZBACH, M. Programação paralela em cpu e gpu: Uma avaliação do desempenho das apis openmp, cuda, opencl e openacc. *Revista de Engenharia, Computação e Tecnologia*, v. 01, n. 01, p. 19–24, novembro 2017. Citado na página 58.
- DIAS, M. H. C.; NAPOLITANO, F. M. P.; SILVEIRA, A. C. da. Ferramenta de predição de cobertura para planejamento de comunicações táticas v/uhf. In: *XX Simpósio de Aplicações Operacionais em Áreas de Defesa (SIGE)*. São José dos Campos, SP: [s.n.], 2018. Acesso em: 26 jan. 2023. Disponível em: <https://www.researchgate.net/publication/328176216_Ferramenta_de_predicao_de_cobertura_para_planejamento_de_comunicacoes_taticas_VUHF>. Citado na página 19.
- DONOHUE, D. J.; KUTTLER, J. R. Propagation modeling over terrain using the parabolic wave equation. *IEEE Transactions on Antennas and Propagation*, IEEE, v. 48, n. 2, p. 260–277, 2000. Citado na página 48.
- EMBEDDED. *Comparison of OpenMP & OpenCL Parallel Processing Technologies*. [S.l.], 2014. Disponível em: <<https://www.embedded.com/comparison-of-openmp-opencl-parallel-processing-technologies/>>. Acesso em: 15 jan. 2023. Citado na página 17.
- HOLM, P. D. Wide-angle shift-map pe for a piecewise linear terrain—a finite-difference approach. *IEEE Transactions On Antenas and Propagation*, v. 55, n. 10, 2007. Citado 4 vezes nas páginas 43, 48, 49 e 50.
- JANASWAMY, R. *Efficient Parabolic Equation Solution of Radiowave Propagation in an Inhomogeneous Atmosphere and Over Irregular Terrain: Formulation*. Tese (Doutorado) — Naval Postgraduate School, 1994. Disponível em: <https://www.researchgate.net/publication/328176216_Ferramenta_de_predicao_de_cobertura_para_planejamento_de_comunicacoes_taticas_VUHF>.

[//core.ac.uk/download/pdf/36721547.pdf](https://core.ac.uk/download/pdf/36721547.pdf)>. Acesso em: 15 jan. 2023. Citado na página 16.

KELLEHER, A. *Lei de Moore – Agora e no Futuro*. [S.l.], 2022. Disponível em: <<https://www.intel.com.br/content/www/br/pt/newsroom/opinion/moore-law-now-and-in-the-future.html>>. Acesso em: 15 jan. 2023. Citado na página 55.

KIRBY, J. T. Large-angle parabolic equation methods. *International Conference on Coastal Engineering*, v. 20, n. 32, 1986. Citado na página 44.

KLEMM, M. *Intro to GPU Programming with the OpenMP API*. 2021. Disponível em: <<https://www.openmp.org/wp-content/uploads/2021-10-20-Webinar-OpenMP-Offload-Programming-Introduction.pdf>>. Acesso em: 27 jan. 2023. Citado 2 vezes nas páginas 18 e 80.

LEVY, M. *Parabolic equation methods for electromagnetic wave propagation*. [S.l.]: IEEE Electromagnetic Waves Series 45, 2000. Citado 3 vezes nas páginas 38, 39 e 47.

MOORE, G. E. Cramming more components onto integrated circuits. *Electronics*, p. 114–117, 1965. Citado na página 55.

OPENMP.ORG. *OpenMP 5.2 API Syntax Reference Guide*. [S.l.], 2021. Disponível em: <<https://www.openmp.org/wp-content/uploads/OpenMPRefCard-5-2-web.pdf>>. Acesso em: 15 jan. 2023. Citado na página 17.

OZGUN, O. et al. PETOOL: MATLAB - based one-way and two-way split-step parabolic equation tool for radiowave propagation over variable terrain. *Computer Physics Communications*, p. 2638–2654, 2011. Citado na página 39.

PEKERIS, C. Accuracy of the earth-flattening approximation in the theory of microwave propagation. *Physical Review*, American Physical Society, v. 70, p. 518–522, 1946. Citado na página 46.

RAMALHO, G. R. *Uma ferramenta de predição e avaliação de cobertura para sistema de comunicação ponto-área*. Dissertação (Mestrado) — Instituto Nacional de Telecomunicações (Inatel), Santa Rita do Sapucaí, Setembro 2006. Citado na página 19.

ROSA, J. *Ondas Eletromagnéticas*. [S.l.], 2019. Disponível em: <<https://www.educamaisbrasil.com.br/enem/fisica/ondas-eletromagneticas>>. Acesso em: 22 jan. 2023. Citado na página 31.

SEVGI, L. *Electromagnetic Modeling and Simulation*. [S.l.]: Wiley, 2014. 663 p. Citado na página 66.

SILVA, F. *Introdução ao OpenMP*. [S.l.], 2012. Disponível em: <http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5645/intro_openmp-Fernando-Silva.pdf>. Acesso em: 27 dez. 2022. Citado 2 vezes nas páginas 59 e 60.

SIQUEIRA, F. C. As equações de maxwell e as ondas eletromagnéticas. *Brazilian Journal of Development*, p. 93571–93589, 2021. Citado na página 30.

TAVAKKOL, S. *Interactive and Immersive Coastal Hydrodynamics*. Tese (Doutorado) — Faculty of the Graduate School, University of Southern California, 2019. Disponível em: <https://www.researchgate.net/publication/332240312_Interactive_and_Immersive_Coastal_Hydrodynamics>. Citado na página 67.

TEDESCO, K. *Concorrência, Paralelismo, Processos, Threads, programação síncrona e assíncrona*. 2020. Disponível em: <<https://www.treinaweb.com.br/blog/concorrenca-paralelismo-processos-threads-programacao-sincrona-e-assincrona>>. Acesso em: 26 jan. 2023. Citado na página 58.

VASCONCELOS, L. S. *Análise da propagação troposférica sobre terrenos irregulares em VHF e UHF utilizando equações parabólicas e o desenvolvimento de um novo modelo híbrido para predição de perda de percurso*. Tese (Doutorado) — Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, 2017. Disponível em: <<https://repositorio.ufu.br/handle/123456789/20157>>. Acesso em: 26 dez. 2022. Citado 18 vezes nas páginas 6, 7, 16, 17, 18, 19, 33, 34, 41, 42, 43, 45, 53, 54, 63, 64, 73 e 81.

Apêndices

APÊNDICE A – CÓDIGOS

A.1 Arquivo *parametros.json*

```
{
  "Diretorio de trabalho": "../terrenos",
  "Arquivo do perfil do terreno": "perfil1.ghl",
  "Modelo": 0,
  "Distancia (geodesica) maxima [m]": 1000,
  "Altura maxima acima do terreno [m]": 100,
  "Passo de distancia paraxial [m]": 1,
  "Passo de altitude [m]": 1,
  "Metodo de Otimizacao da PCR": 3,
  "Fonte": {
    "Frequencia [MHz]": 1000,
    "Tipo da fonte": 0,
    "Largura de Feixe de 3dB das fontes [graus]": [2],
    "Beam tilt das fontes [graus]": [1],
    "Altura das fontes [m] acima do solo": [25],
    "Polarizacao da fonte [H/V]": "H"
  },
  "Refracao": {
    "Perfil": 4,
    "Indice de refracao inicial": 1.0003,
    "Refractividade a nivel do mar": 315,
    "Altura da escala da refractividade": -0.136,
    "Alturas (em ordem de baixo pra cima) de limite na
      atmosfera": [50, 100],
    "Inclinacoes do perfil (para caso bilinear ou trilinear)"
      : [-1e-5, 1e-5, 1e-3]
  },
  "Saida de dados": {
    "Diretorio de saida": "OUTPUT",
    "Tipo de salvamento": 0,
    "PASSO DE ALTURA Salva [m]": -1000,
    "PASSO DE DISTANCIA Salva [m]": -2000,
    "Imagem de Saida": [5000,5000]
  },
  "Solo": [ {"tipo": 5, "posicao": 0},
    {"tipo": 0, "posicao": 100, "epsilon relativo": 1, "
      sigma": 1e15 } ]
}
```

No arquivo *parametros.json* é pedido que o usuário escolha alguns tipos, os quais são:

```
enum ModeloTerra {
    TERRA_SEM_RELEVO_2 = 0,
    TERRA_SEM_RELEVO_1 = 1,
    TERRA_COM_RELEVO = 2
};
```

```
enum TipoFonte {
    FONTE_GAUSSIANA = 0,
    FONTE_RECT = 1,
    FONTE_OMNI = 2
};
```

```
enum PerfisRefracao {
    ATMOSFERA_PADRAO = 0,
    LINEAR = 1,
    BILINEAR = 2,
    TRILINEAR = 3,
    EXPONENCIAL = 4
};
```

```
enum TiposSolo {
    DEFINIDO_PELO_USER = 0,
    MAR = 1,
    AGUA_DOCE = 2,
    SOLO_UMIDO = 3,
    SOLO_MEIO_UMIDO = 4,
    SOLO_SECO = 5,
    AGUA_PADRAO = 6
};
```

```
enum TipoSalvamentoCampo {
    SALVAR_COMPLETO = 0,
    SALVAR_DISCRETIZADO = 1,
    SALVAR_MEDIA = 2,
    SALVAR_MEDIAMOVEL = 3
};
```

```
enum ModoOtimizacao {
    N_MAIS_PROXIMO = 0,
    AUMENTAR_N = 1,
    DIMINUIR_N = 2,
    MANTER_N_ATUAL = 3 // nao otimiza
};
```


A.2 Cálculo do tamanho da camada absorvedora

O domínio `max_z` é entendido conforme a determinada a taxa em que a janela de altura deve crescer, `win_ratio`.

```
win_ratio = (freq <= 401e6) ? 8 : ((freq <= 1001e6) ? 4 : ((freq
    < 2001e6) ? 2 : 0.5));
max_z = (freq < 2001e6) ? max_z_user * win_ratio : max_z_user *
    (1 + win_ratio);
```

A.3 Determinando o valor máximo de um veto de alta ordem

Usando a estratégia apresentada na Seção 4.4, cada *thread* calcula o valor máximo para cada “pedaço” do vetor *v* e armazena no vetor *vetor*, para depois calcular o valor máximo que cada uma guardou, de forma sequencial. O valor mínimo é determinado analogamente.

```
double valor_maximo = invalid_max(v[0]);
#pragma omp parallel
{
    int i = omp_get_thread_num();
    vetor[i] = valor_maximo;
#pragma omp for
    for (int j = 0; j < n; ++j) {
        if (vetor[i] < v[j] && invalid_max(v[j])) vetor[i] =
            v[j];
    }
}
valor_maximo = vetor[0];

for (lli i = 1; i < cpu_n_threads; ++i) {
    if (valor_maximo < vetor[i]) valor_maximo = vetor[i];
}

double invalid_max(double v){return (!(isnan(v)||isinf(v)))? v :
    0;}
```

A.4 Interpolação

```
#pragma omp parallel for private(yr,t,j)
  for (lli i = 0; i < tam_out; ++i) {
    yr = 0.0;
    t = x_out[i] / delta_in; // mapeamento pra indice
    j = (lli) floor(t); // indice do x_in[i] // PRIMEIRO J
    t = t - (double) j; // t = x_out[i] / delta_in - floor(
      x_out[i] / delta_in)
    yr = y_in[j] * (1 - t);
    j++; // SEGUNDO J
    if (j < tam_in) { // y_out[tam_in - 1] = y_in[j1 ==
      tam_in - 1]
      yr += y_in[j] * t;
    }

    y_out[i] = yr;
  }
}
```

A.5 Passo de distância e altura *default*

```
delz = (delz == 0.) ? lambda0 / 4 : delz;

if (delx == 0.) { //DELX_DEFAULT
  delx = lambda0;
  M = (lli) floor(max_range / delx) + 1;
  range = Linspace(max_range, M);
  terrenoInterpolado = Vetor<double>(M);
  slopes = Vetor<double>(M);

  CPU::interp1(perfilTerreno.delta, perfilTerreno.host, range,
    terrenoInterpolado.host, perfilTerreno.z, M);
  CPU::derivada(slopes.host, terrenoInterpolado.host, delx, M);

  delx = (-0.1125 * 180. / propag_pi) * lambda0 * atan(CPU::
    maximoAbs(slopes.host, slopes.z)) + lambda0;
  delx = (delx < lambda0 / 20) ? (lambda0 / 20) : delx; //
  evita que delx seja desnecessariamente muito pequeno
}
```

```

terrenoInterpolado.release();
slopes.release();
}

```

A.6 Derivada por diferença central

```

// Calcula a derivada por diferenca central
//      t[i] - t[i-1] = t[i+1] - t[i] = delx
//      f'[i] = ((-y[i-1]) + (y[i+1])) / (2*delx);
// i = 0 ->
//      f'[0] = ((y[0]-y[2]) / (2) + (y[1]-y[0]) * (2)) / (
//      delx);
// i = n-1 ->
//      f'[n-1] = ((y[n-1]-y[n-2]) * (2) + (y[n-3]-y[n-1])
//      /(2)) / (delx);
// 0 1 ... n-3 n-2 n-1 ---->
//      f'[i] = ((y[i]-y[i-1]) / (t[i]-t[i-1]) * (t[i+1]-t[
//      i]) + (y[i+1]-y[i]) / (t[i+1]-t[i]) * (t[i]-t[i-1])) / ((t[i
//      +1]-t[i-1])));

double fs = 1. / delx;
derivada_y[0] = ((y[0] - y[2]) * 0.5 + (y[1] - y[0]) * (2)) *
    fs;
#pragma omp parallel for
    for (lli i = 1; i < n - 1; i++) {
        derivada_y[i] = ((-y[i - 1]) + (y[i + 1])) * fs * 0.5;
    }
derivada_y[n - 1] = ((y[n - 1] - y[n - 2]) * (2) + (y[n - 3]
    - y[n - 1]) * 0.5) * fs;

```

A.7 Multiplicação da matriz tridiagonal

```

/**
 * C = D*B
 * [c0]   |d0 u0   | [b0]
 * [c1] = |l1 d1 u1 | [b1]

```

```

* [c2]      |  12  d2| [b2]
*/
int CPU::multiplicaTridiagonal(Complexo *C, Complexo *dl,
    Complexo *d, Complexo *du, Complexo *B, lli n) {
    lli end = n - 1;
#pragma omp parallel for
    for (lli i = 1; i < end; ++i) {
        C[i] = dl[i] * B[i - 1] + d[i] * B[i] + du[i] * B[i + 1];
    }
    C[0] = d[0] * B[0] + du[0] * B[1];
    C[end] = dl[end] * B[end - 1] + d[end] * B[end];
    return 0;
}

```

A.8 Algoritmo de Thomas

```

// Determina o vetor auxiliar V
Complexo W{};
u[0] = A.d[0];
for (lli i = 1; i < nsystem; i++) {
    W = A.dl[i] / u[i - 1];
    u[i] = A.d[i] - W * A.du[i - 1];
    V[i] = V[i] - W * V[i - 1]; // Vi depende de V[i-1]
}

// Calcula as posicoes de u
u[A.n - 1] = V[A.n - 1] / A.d[A.n - 1];
for (lli i = A.n - 2; i >= 0; --i) {
    u[i] = (V[i] - A.du[i] * u[i + 1]) / u[i];
}

```

A.9 Algoritmo da PCR

```

long long int logSize = log2(Npcr + 1) - 1; // por isso 2^k-1
long long int index, i, j, offset, index1, index2, step;
T alpha, gamma;

```

```

    // cyclic comprime
#pragma omp parallel shared (dl, d, du, B, Npcr) private(i, step,
    j, offset, index1, index2, alpha, gamma)
    {
        for (i = 0; i < logSize; i++) {
            step = (1LL << (i + 1)); //pow(2, i + 1);

#pragma omp for
            for (j = step - 1LL; j < Npcr; j = j + step) { // pow
                (2, i + 1)
                offset = (1LL << i); //pow(2,i);
                index1 = j - offset; //pow(2, i);
                index2 = j + offset; //pow(2, i);
                alpha = dl[j] / d[index1];
                gamma = du[j] / d[index2];

                dl[j] = -dl[index1] * (alpha); //
                d[j] = d[j] - du[index1] * alpha - dl[index2] *
                    gamma;
                du[j] = -du[index2] * (gamma);
                B[j] = B[j] - B[index1] * alpha - B[index2] *
                    gamma;
            }
        }

#pragma omp master // #pragma omp atomic write
    {
        index = (Npcr - 1LL) / 2;
        x[index] = B[index] / d[index];
    }

    // cyclic expande

    for (i = logSize - 1; i >= 0; i--) {
        step = (1LL << (i + 1));

#pragma omp for
        for (j = step - 1LL; j < Npcr; j = j + step) {
            offset = (1LL << i);
            index1 = j - offset; //pow(2, i);
            index2 = j + offset; //pow(2, i);
            if (index1 - offset < 0) {

```

```
        x[index1] = (B[index1] - du[index1] * x[
            index1 + offset]) / d[index1];
    } else {
        x[index1] =
            (B[index1] - dl[index1] * x[index1 -
                offset] - du[index1] * x[index1 +
                    offset]) / d[index1];
    }
    if (index2 + offset >= Npcr) {
        x[index2] = (B[index2] - dl[index2] * x[
            index2 - offset]) / d[index2];
    } else {
        x[index2] =
            (B[index2] - dl[index2] * x[index2 -
                offset] - du[index2] * x[index2 +
                    offset]) / d[index2];
    }
}
}
```