



UMA ABORDAGEM BASEADA EM SIMULATED ANNEALING PARA A CONSTRUÇÃO DE MODELOS DE VELOCIDADE INICIAL PARA A FWI

Rafael Mendes da Silva

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Claudio Luis de Amorim
Valmir Carneiro Barbosa
Cristiana Barbosa Bentes

Rio de Janeiro
Maio de 2023

UMA ABORDAGEM BASEADA EM SIMULATED ANNEALING PARA A
CONSTRUÇÃO DE MODELOS DE VELOCIDADE INICIAL PARA A FWI

Rafael Mendes da Silva

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadores: Claudio Luis de Amorim
Valmir Carneiro Barbosa
Cristiana Barbosa Bentes

Aprovada por: Prof. Claudio Luis de Amorim
Prof. Valmir Carneiro Barbosa
Prof. Cristiana Barbosa Bentes
Prof. Diego Leonel Cadette Dutra
Prof. José Luis Drummond Alves
Prof. Maria Clicia Stelling de Castro
Prof. Lúcia Maria de Assumpção Drummond

RIO DE JANEIRO, RJ – BRASIL
MAIO DE 2023

Silva, Rafael Mendes da

Uma abordagem baseada em simulated annealing para a construção de modelos de velocidade inicial para a FWI/Rafael Mendes da Silva. – Rio de Janeiro: UFRJ/COPPE, 2023.

XIV, 56 p.: il.; 29,7cm.

Orientadores: Claudio Luis de Amorim

Valmir Carneiro Barbosa

Cristiana Barbosa Bentes

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2023.

Referências Bibliográficas: p. 53 – 56.

1. Inversão Sísmica.
2. Modelagem Estocástica.
3. Simulated Annealing. I. Amorim, Claudio Luis de *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*Ao meu pai, a quem nossa
sociedade desigual não deu a
mesma oportunidade que eu tive.*

Agradecimentos

Embora não seja uma pessoa religiosa, creio na existência de uma inteligência suprema, causa primária de todas as coisas do universo, e agradeço a ela por ter permitido dentre os seus desígnios que eu desenvolvesse este trabalho. Trabalhar com pesquisa científica é muito gratificante, pois acredito que a pesquisa, a busca pelo conhecimento, é umas das atividades mais transformadoras da nossa sociedade.

Gostaria de agradecer aos meus pais, que me ensinaram o valor da honestidade e do trabalho, em particular o meu pai, que muito trabalhou para que eu tivesse a oportunidade que ele não teve.

Quero agradecer ao Cláudio, ao Valmir e à Cristiana, meus orientadores. Foram companheiros desta jornada, que teve dias muito difíceis, mas vocês estavam sempre presentes, eu nunca vou me esquecer disso.

Agradeço também à empresa Geo Imaging Solutions, em particular ao seu CEO e meu amigo, Carlos Fernandez, e meus amigos Bruno Kaelin e Alejo. Vocês sempre acreditaram neste trabalho, sempre acreditaram em mim e me apoiaram, não há palavras pra descrever o que fizeram por mim.

Gostaria de agradecer ao nosso país, que com todas as críticas que possam ser feitas, criou uma política de Estado que permite que eu, oriundo de uma família humilde, tivesse a possibilidade de me graduar, fazer mestrado e agora doutorado, em instituições de altíssimo nível.

Agradeço aos meus amigos, que me incentivaram, compreenderam a minha ausência. Talvez vocês não saibam, mas são a família que me foi permitido escolher.

Agradeço aos professores da COPPE/UFRJ, pela formação que me proporcionaram. Também ao COMPASSO/LCP (Laboratório de Computação Paralela), ao LNCC (Laboratório Nacional de Computação Científica) e ao NACAD (Núcleo Avançado de Computação de Alto Desempenho), pela infra-estrutura oferecida para a execução dos experimentos necessários a este trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

UMA ABORDAGEM BASEADA EM SIMULATED ANNEALING PARA A CONSTRUÇÃO DE MODELOS DE VELOCIDADE INICIAL PARA A FWI

Rafael Mendes da Silva

Maio/2023

Orientadores: Claudio Luis de Amorim
Valmir Carneiro Barbosa
Cristiana Barbosa Bentes

Programa: Engenharia de Sistemas e Computação

Atualmente, a maior parte das descobertas de reservas de Óleo e Gás estão localizadas em áreas com estruturas geológicas complexas, impondo diversos desafios para a exploração. Métodos estado da arte baseados na equação de onda, como migração reversa no tempo (Reverse Time Migration, RTM), são utilizados em face destes desafios, pois eles são capazes de gerar imagens em tais regiões. Entretanto, tais métodos requerem modelos de velocidade detalhados do subsolo. A Inversão de Forma de Onda Completa (FWI: Full Waveform Inversion) é utilizada para criar tais modelos de velocidade detalhados. Entretanto, como a FWI é baseada em métodos de gradiente, ela sofre da limitação de convergir para mínimos locais. Para evitar este problema, a FWI precisa iniciar de um modelo de velocidade localizado na mesma região de convexidade do mínimo global. Neste trabalho, propomos um framework simulated annealing para encontrar tal modelo inicial de velocidade. O processo iterativo do algoritmo de simulated annealing, entretanto, demanda um alto custo computacional. Primeiro, por causa do grande número de iterações necessário para explorar o espaço de variáveis. Segundo, pelo grande número de simulações necessárias para ajustar os parâmetros do simulated annealing. Enquanto a primeira limitação é uma preocupação comum de trabalhos anteriores, a segunda em geral é negligenciada, baseando-se em tentativa e erro. O nosso framework simulated annealing inclui várias estratégias para tratar ambos os problemas. Avaliamos a nossa abordagem utilizando o dado sísmico Marmousi, e obtivemos resultados promissores em comparação com trabalhos anteriores. O custo computacional para ajustar os parâmetros da simulação foi reduzido em uma ordem de grandeza. A quantidade

de iterações necessária para explorar o espaço de variáveis foi reduzida pela metade. Além disso, a FWI convergiu para um modelo de velocidade detalhado com excelente qualidade, utilizando o modelo de velocidade inicial gerado pelo método proposto.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

AN APPROACH BASED ON SIMULATED ANNEALING TO BUILD AN
INITIAL VELOCITY MODEL FOR FWI

Rafael Mendes da Silva

May/2023

Advisors: Claudio Luis de Amorim
Valmir Carneiro Barbosa
Cristiana Barbosa Bentes

Department: Systems Engineering and Computer Science

Oil field discoveries are located in geological areas with complex structures, imposing new challenges for exploration. Two-way wave equation methods such as reverse time migration (RTM) are able to improve the images in these regions, however they require detailed velocity models. Full Waveform Inversion (FWI) is a prominent method commonly used to create detailed velocity models of the subsurface. However, as it relies on gradient methods, it suffers from the limitation of getting trapped in local minima. To avoid this problem, FWI needs to start from an initial velocity model that lies in the same region of convexity as the global minimum. In this work, we propose a simulated annealing framework to find such an initial velocity model. The iterative process of simulated annealing, however, entails high computational cost. First, because of the large number of iterations needed to explore the variable space. Second, because of the large number of simulations needed to adjust the simulation parameters. While the first issue is a common concern of previous related work, the second issue is usually neglected, relying on trial and error. Our simulated annealing framework includes a number of strategies to tackle both issues. We evaluated our approach using the Marmousi seismic data, and obtained promising results when compared to previous work. The computational effort to fine tune the simulation parameters was reduced by one order of magnitude. The number of iterations needed to explore the variable space was reduced by half. In addition, FWI was able to find a detailed velocity model with high quality when using the initial velocity model generated by the method.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Acrônimos	xiv
1 Introdução	1
1.1 Abordagem Proposta	3
1.2 Resumo das Contribuições e Resultados	5
1.3 Organização do Texto	6
2 Revisão de Conceitos	7
2.1 Processamento Sísmico	7
2.1.1 Reflexões versus Refrações	9
2.1.2 Modelagem	9
2.1.3 Migração	10
2.1.4 Inversão	12
2.2 Simulated Annealing	13
3 Construção de Modelos de Velocidade com Simulated Annealing	16
3.1 Modelos de Velocidade Suaves	16
3.2 Formulação do Problema Inverso	17
3.3 Descrição do Framework	18
3.3.1 Amostragem da Distribuição a Priori	18
3.3.2 Cálculo da Função de Custo	21
3.3.3 Resumo do Framework das Iterações	24
3.3.4 Stepping	24
3.4 Execução Paralela das Simulações	30
4 Avaliação Experimental	33
4.1 Ambiente de Execução	33
4.2 Dados Experimentais	34

4.3	Exploração do Espaço de Variáveis	35
4.4	Análise de Convergência	35
4.4.1	Tamanho da Época	36
4.5	Análise da Probabilidade de Aceitação	37
4.6	Análise da Função de Custo	38
4.7	Análise de Desempenho	40
5	Trabalhos Relacionados	44
5.1	Geração de Amostras	44
5.2	Métodos Monte Carlo	45
5.2.1	Metropolis-Hastings	46
5.2.2	Simulated Annealing	46
5.3	Algoritmos Genéticos	47
6	Conclusões e Trabalhos Futuros	49
6.1	Conclusões	49
6.2	Trabalhos Futuros	51
	Referências Bibliográficas	53

Lista de Figuras

2.1	Esquema da propagação de ondas no subsolo.	8
2.2	Primeiro tiro do dado sísmico Marmousi.	8
2.3	Tiro do dado sísmico gerado pelo modelo de velocidade da Figura 2.1.	10
2.4	Etapas da migração reversa no tempo (RTM).	11
2.5	Em (a) e (b) temos propagação direta e reversa com dois tiros, em (c) as correlações dos dois tiros empilhadas mostrando o refletor.	11
2.6	Kirchoff versus RTM: pode-se observar melhor definição das laterais do corpo de sal com a RTM (cortesia de Conoco-Philips).	12
2.7	Tomografia versus FWI: pode-se observar que a resolução do modelo de velocidade é bem maior no caso da FWI(cortesia de GeoImaging Solutions Inc.)	13
2.8	Em (a) é mostrado o dado sísmico de entrada para a FWI e em (b) é mostrado o modelo de velocidade inicial utilizado.	14
2.9	Em (a) temos o resultado da FWI, enquanto em (b) é mostrado o modelo de velocidade original.	14
3.1	O modelo suave de velocidade (a) é usado como modelo inicial da FWI, que converge para o resultado apresentado em (b).	17
3.2	Amostra gerada pelo Snesim (a) e a mesma amostra após aplicada a suavização.	19
3.3	Em (a) é mostrada a amostra original, em (b) pode ser vista a seleção da área a ser perturbada, e em (c) o resultado da perturbação, gerando a amostra proposta.	20
3.4	A velocidade suavizada (a) é usada para gerar o dado sísmico (b), que não apresenta reflexões, apenas refrações e a onda direta, ao contrário da velocidade real (c) que gera as reflexões mostradas no dado sísmico (d).	22
3.5	A imagem migrada (a) foi computada pela RTM usando o modelo de velocidades suave da Figura 3.4a. Nela pode-se observar a similaridade com a densidade (b).	23

3.6	Contraste de densidade (a) é usado para gerar as reflexões do dado sísmico (b).	24
3.7	Workflow de uma iteração do simulated annealing.	25
3.8	(a) mostra o tamanho de A_k decrementado exponencialmente de A_0 para A_{N-1} e (b) mostra a distância Euclideana entre cada par adjacente de amostras do modelo de velocidade.	27
3.9	Histogramas normalizados de ΔC_k para a simulação com 4×10^3 iterações e $A_k = 0.8A_T$ (a) e para $A_k = 0.1A_T$ (b).	28
3.10	Amostra de velocidade suavizada baseada no dado Marmousi	31
3.11	Imagens parciais geradas pela RTM por cada um dos nós de computação.	31
3.12	Imagem empilhada (a) resultado das imagens mostradas na Figura 3.11. Contraste de densidade (b).	32
4.1	Modelo de velocidades do Marmousi (a) e a densidade correspondente.	34
4.2	Amostra inicial do modelo de velocidade (por categorias e suavizada) e a imagem de treinamento.	35
4.3	(a) Resultado da simulação com 2×10^4 iterações usando parâmetros $A_0 = 0.8A_T$, $A_{N-1} = 0.1A_T$, $T_0 = 120$, $T_{N-1} = 2 \times 10^{-6}$, e tamanho de época 600. (b) Convergência da FWI usando o modelo de velocidades em (a) como modelo inicial.	37
4.4	(a) Valores de p_k para todas as iterações da simulação. (b) Média dos valores de p_k para cada época, considerando apenas as iterações onde $\Delta C_k > 0$	38
4.5	(a) Decaimento exponencial do tamanho da área conforme a época. (b) e (c) mostra os valores da função de custo antes e depois da exploração, respectivamente. (d) Valores de ΔC_k para todas as iterações.	39
4.6	Valores mínimos da função de custo normalizados. Os pontos verdes representam as amostras onde C_{\min} é atualizado.	40
4.7	Tempo de execução normalizado para 1 a 10 nós de computação.	42
4.8	Percentual do tempo de execução total da simulação consumido por cada etapa das iterações.	42

Lista de Tabelas

4.1	Resumo dos experimentos para vários tamanhos de época.	37
-----	--	----

Acrônimos

FWI *Full Waveform Inversion.* 2

RTM *Reverse Time Migration.* 2

Capítulo 1

Introdução

Aplicações de processamento sísmico são de muito interesse por conta de sua importância para o setor de Óleo e Gás, que tem grande relevância econômica e estratégica. Tais aplicações estão presentes em diversas etapas da cadeia produtiva tais como: (1) a exploração, que tem como objetivo encontrar novas reservas; (2) caracterização dos reservatórios; e (3) acompanhamento das transformações nos reservatórios em decorrência da extração de óleo e gás (sísmica 4D).

De uma forma geral as aplicações sísmicas envolvem etapas de aquisição de dados sísmicos e seu posterior processamento. A aquisição sísmica consiste na geração de ondas de choque que se propagam pelo subsolo, e são refletidas ou refratadas, fazendo com que parte de sua energia retorne para a superfície, que pode ser em terra ou mar. Essa energia se apresenta na forma de vibrações na superfície, que são captadas por sensores que armazenam esta informação. Todo este conjunto de dados produzido constitui o que chamamos de dado sísmico. Estes dados são processados computacionalmente utilizando modelos matemáticos que descrevem a propagação das ondas no subsolo. Este processamento permite que se calcule o valor aproximado de diversas propriedades das rochas presentes no subsolo em uma dada área. Os valores destas propriedades nos diversos pontos do subsolo permitem determinar uma provável presença de hidrocarbonetos. Tipicamente, o resultado do processamento sísmico são imagens, onde cada ponto da imagem representa o valor de uma determinada propriedade física, em um determinado ponto do subsolo. A velocidade de propagação das ondas sísmicas no subsolo está entre as propriedades mais importantes a ser determinada, sendo portanto o foco de boa parte das tecnologias de processamento sísmico, sendo também o foco deste trabalho.

Atualmente, a maior parte das descobertas de Óleo e Gás estão localizadas em áreas com estruturas geológicas complexas, impondo diversos desafios para a exploração. Um exemplo de tais desafios é a camada pré-sal, onde as reservas se encontram geologicamente abaixo de uma camada de sal, limitando a eficácia de

métodos tradicionais de imageamento do subsolo baseados em traçado de raios [1]. Métodos baseados na equação de onda tais como Migração Reversa no Tempo (Reverse Time Migration, RTM) [2] são utilizados em face destas limitações, pois eles são capazes de gerar imagens em tais regiões complexas. Entretanto, estes métodos requerem modelos de velocidade detalhados do subsolo. A Inversão de Forma de Onda Completa [3, 4] (Full Waveform Inversion, FWI) pode ser utilizada para criar tais modelos de velocidade detalhados, utilizando a equação de propagação de onda e estimativa de parâmetros.

Na prática, a FWI é um método iterativo de casamento de dados baseado em gradiente, utilizado para ajustar o dado sísmico modelado pela equação de onda com o dado sísmico adquirido em campo. O objetivo da FWI é encontrar o mínimo local da função de custo definida por

$$M(m) = \|D_{\text{acq}} - D_{\text{mod}}(m)\|^2, \quad (1.1)$$

onde m representa as variáveis a serem ajustadas, que no nosso caso, se referem ao modelo de velocidade de propagação das ondas no subsolo. $M(m)$ é a função de custo definida pela distância quadrática entre o dado adquirido D_{acq} e o dado modelado D_{mod} . Este último obtido utilizando a modelagem sísmica, que tem como parâmetro de entrada o modelo de velocidade representado por m , estabelecendo assim uma relação direta entre m e $D_{\text{mod}}(m)$. Portanto, o que a FWI faz é ajustar os valores de velocidade de m , de forma que $D_{\text{mod}}(m)$ se aproxime de D_{acq} , ou em outras palavras, m se aproxime do modelo de velocidade real.

Como a FWI é baseada no método de gradiente, ela sofre de uma limitação conhecida destes métodos, a presença de mínimos locais, que a impede de convergir para o mínimo global. Uma abordagem para evitar o problema dos mínimos locais e encontrar diretamente o mínimo global é utilizar métodos de otimização globais baseados em Monte Carlo, conforme proposto por [5–7]. Entretanto, estas abordagens impõem uma grande quantidade de iterações Monte Carlo, aumentando a demanda computacional.

Alternativamente, pode-se tirar vantagem de uma propriedade da FWI relacionada ao modelo inicial de velocidade, para reduzir o custo computacional de tais métodos de otimização global. Neste caso, o método de Monte Carlo é utilizado para gerar um modelo inicial de velocidade que atenda o requisito de convergência da FWI: o modelo de velocidade inicial precisa estar na mesma região de convexidade do mínimo global. Em outras palavras, para que a FWI convirja para o mínimo global, o seu modelo inicial de velocidade precisa produzir traços sísmicos sintéticos cujo deslocamento com relação aos traços sísmicos reais no domínio de tempo, seja no máximo a metade do comprimento de onda correspondente à frequência média

do dado sísmico real [8]. Se o deslocamento for maior que isso, o procedimento de minimização se iniciará fora da região de convexidade do mínimo global, e irá portanto convergir para um mínimo local. Este problema é conhecido como saltos de ciclo (cycle-skipping). Trabalhos anteriores [9, 10] demonstraram que estes modelos iniciais de velocidade não precisam ser detalhados para garantir a convergência da FWI. É suficiente prover um modelo velocidade inicial com valores baixos de número de onda (contraste suave de velocidade), contendo apenas os macro eventos geológicos.

Já foi provado que métodos de otimização global, tais como simulated annealing [11–14] são efetivos para encontrar tais modelos iniciais com contraste suave de velocidade [15]. Entretanto, existe um alto custo computacional associado às simulações do simulated annealing. Primeiro, elas requerem um grande número de iterações para explorar o espaço de variáveis para problemas com muitas variáveis. Isto é especialmente verdadeiro para inversão sísmica, onde mesmo modelos de velocidade em 2D podem possuir milhares de variáveis. Segundo, é necessária uma grande quantidade de simulações para ajustar os parâmetros do simulated annealing. No nosso estudo, propomos uma abordagem composta de diversas técnicas que tratam ambos os aspectos do custo computacional do simulated annealing.

1.1 Abordagem Proposta

Neste trabalho, propomos um *framework* simulated annealing para encontrar um modelo inicial de velocidade para a FWI, de forma que ela convirja para o mínimo global. Além disso, propomos três seguintes técnicas para tratar o alto custo computacional do simulated annealing. Primeiro, como parte do nosso método de amostragem dos modelos de velocidade, propomos algumas técnicas para reduzir o número de iterações, reduzindo o espaço de variáveis e guiando sua exploração. Segundo, propomos uma nova técnica para o cálculo da função de custo baseada apenas em reflexões sísmicas, evitando a modelagem de dados sísmicos com afastamentos longos e processamento de modo, como requerido por refrações. Terceiro, identificamos a interdependência dos parâmetros do simulated annealing e propomos uma metodologia para calculá-los, reduzindo drasticamente o número de simulações necessárias para ajustar seus valores.

As técnicas para reduzir o número de iterações do simulated annealing são: (1) redução do tamanho do espaço de variáveis; (2) focar a exploração deste espaço em suas áreas mais relevantes; e (3) controlar o quão exploratória é a simulação dependendo da época do simulated annealing.

Para reduzir o tamanho do espaço de variáveis, representamos o modelo de velocidade utilizando duas malhas: granularidade fina e grossa, baseado no que foi pro-

posto por [16]. Utilizamos a malha grossa para a amostragem (dado que as amostras não apresentam grande contraste de velocidade) e a malha fina para os métodos de equação de onda (evitando dispersão numérica e falseamento de frequências). Assim, após a geração de uma amostra utilizando a malha grossa, ela é interpolada para a malha fina utilizando um método de interpolação baseado em média móvel. Em nossos experimentos com o dado Marmousi [17], a amostragem empregando a malha grossa reduziu centenas de vezes o tamanho do espaço de variáveis. Trabalhos anteriores reduziram o espaço de variáveis representando as camadas do subolo utilizando um conjunto de pontos [15] e corpos salinos como um conjunto de elipses [18]. Em ambos os casos, as abordagens requerem intervenção humana para marcação dos sedimentos e o topo dos corpos salinos, ao passo que nosso método é completamente automático.

Para focar nas áreas mais relevantes do espaço de variáveis, utilizamos conhecimento prévio da geologia em estudo. Este conhecimento prévio usualmente origina-se de outros métodos ou mesmo de uma área com geologia similar. Então, baseado no que foi proposto por [19, 20], empregamos uma imagem de treinamento para representar este conhecimento geológico prévio. Nesta tese propomos utilizar esta imagem de treinamento para representar os padrões relevantes a serem gerados nas amostras do modelo de velocidade. Desta forma, podemos evitar reproduzir padrões que não tem chance de estarem presentes no modelo de velocidade real. Enquanto ambos os trabalhos anteriores apenas suportam amostras com valores discretos, nosso método é capaz de gerar amostras mais complexas, compostas de valores contínuos em ponto flutuante. Em nossos experimentos, utilizamos uma imagem de treinamento muito simples, demonstrando que mesmo um conhecimento prévio simples sobre a geologia é capaz de auxiliar na convergência. Adicionalmente, a imagem de treinamento permite representar padrões mais complexos que os apresentados em trabalhos anteriores que assumem uma geologia formada por camadas verticalmente justapostas [15, 21–24].

Para controlar o quão exploratória a simulação é, estendemos o trabalho feito por [19], definindo para cada época do simulated annealing o tamanho do subconjunto do espaço de variáveis a ser amostrado. Este subconjunto é definido como uma subárea do modelo de velocidade, e apenas as variáveis dentro desta subárea são amostradas. Chamamos o tamanho desta subárea de *step* da época. Propomos um método eficiente para definir os tamanhos ótimos do step inicial e final. Adicionalmente, propomos uma estratégia específica para decrementar o tamanho do step a cada época e analisamos o seu impacto na exploração. Desta forma, garantimos que a simulação é bastante exploratória no início e restrita no final.

Outro importante aspecto do simulated annealing é o cálculo da função de custo. Com relação a isso, propomos uma nova estratégia que computa a função de custo

baseada apenas nas reflexões sísmicas, ao invés de refrações [16]. Este método nos permite reduzir o tempo gasto pela modelagem sísmica e evita o processamento da separação dos modos (ondas S e P). Também, reflexões são baseadas em dados sísmicos com afastamento curto, que têm boa disponibilidade. Geramos estas reflexões utilizando um modelo de velocidade suave e seu contraste de densidade correspondente. Propomos um método para calcular o contraste de densidade utilizando imagens migradas geradas pela RTM [2].

Com relação à redução no número de simulações necessárias para ajustar os parâmetros do simulated annealing, propomos uma metodologia que estima o valor de dois parâmetros fundamentais do simulated annealing: temperatura e tamanho da época. Ao invés de empregar abordagens baseadas em tentativa e erro, estudamos o comportamento da função de custo e baseado neste comportamento calculamos o valor da temperatura. Então, utilizamos os valores de temperatura e step para calibrar o tamanho da época. Nossos resultados mostram que esta metodologia é efetiva para calcular adequadamente a temperatura e o tamanho da época.

1.2 Resumo das Contribuições e Resultados

As contribuições desta tese são:

- Um framework simulated annealing para encontrar um modelo de velocidade com baixo número de onda que satisfaz o requisito de convergência da FWI.
- Um método de amostragem contendo três técnicas para reduzir o número de iterações das simulações, reduzindo o tamanho do espaço de variáveis, e controlando e guiando a exploração deste espaço.
- Um método de cálculo da função de custo baseado apenas em reflexões sísmicas. Isto evita a necessidade de dados sísmicos com afastamento longo como requeridos pelas refrações. Além disso, propomos uma técnica para criar tais reflexões utilizando o contraste de densidade obtido utilizando imagens migradas.
- Uma metodologia para calcular os valores de temperatura do simulated annealing, baseado em sua relação com o quão exploratória é a simulação, como definido pelo step.
- Uma metodologia para estimar o tamanho da época, baseada no step e na temperatura.
- Uma análise detalhada do impacto dos parâmetros do simulated annealing em sua convergência.

Nossos experimentos com o dado Marmousi demonstram que podemos eficientemente definir os parâmetros do simulated annealing e limitar o número de iterações necessárias para a convergência. Apenas 8×10^3 iterações foram necessárias para definir os parâmetros da simulação. Após a definição de tais parâmetros, a simulação completa convergiu com 2×10^4 iterações. O modelo suave de velocidade resultante foi suficiente para garantir a convergência da FWI. Se comparado a trabalhos anteriores com o dado Marmousi [16, 25], reduzimos por uma ordem de grandeza a quantidade de iterações para definir os parâmetros do simulated annealing e em 50% o número de iterações da simulação.

1.3 Organização do Texto

Apresentamos no Capítulo 2, uma revisão de conceitos que serão utilizados na pesquisa tais como: métodos de modelagem, inversão e migração sísmica. No capítulo 3, apresentamos a abordagem que desenvolvemos para geração de modelos de velocidade utilizando o método de simulated annealing. No capítulo 4 temos a avaliação experimental, bem como os resultados alcançados. No capítulo 5, apresentamos alguns trabalhos relacionados, e finalmente, no capítulo 6, apresentamos nossas conclusões e descrevemos algumas propostas de trabalhos futuros.

Capítulo 2

Revisão de Conceitos

Este capítulo destina-se a revisão de conceitos que são fundamentais para o entendimento dos tópicos desta tese, tais como modelagem, migração e inversão sísmicas.

2.1 Processamento Sísmico

A Figura 2.1 apresenta uma representação esquemática de uma área de subsolo que possui três camadas geológicas paralelas com densidades diferentes. Cada área geológica apresenta diferentes valores de velocidade de propagação de onda (2200 m/s, 2400 m/s e 2600 m/s). Pode ser visto também a fonte sísmica, que pode ser um caminhão vibratório (aquisição terrestre) ou uma bomba de ar acoplada a um navio (aquisição marinha). Analogamente, os receptores podem ser posicionados no solo ou localizados em bóias puxadas por uma embarcação. As setas azuis mostram alguns dos possíveis caminhos de propagação das ondas, sendo que a propagação delas pode ser direta, refletida ou refratada. O processo de aquisição sísmica consiste em armazenar a informação coletada pelos receptores gerando um dado sísmico como o da Figura 2.2, onde é mostrado o primeiro tiro do dado sísmico Marmousi. Dependendo da abordagem usada pode variar qual grandeza é representada no dado sísmico, podendo ser pressão, deslocamento, velocidade etc. Entretanto, o que sempre teremos em cada sensor é uma sequência no tempo de leituras de uma dada grandeza física. Esta sequência por sensor é o que chamamos de *traço sísmico*, que no caso deste trabalho são agrupados por uma fonte comum, formando um *tiro*. A mesma fonte sísmica é deslocada no processo de aquisição, gerando assim vários tiros, portanto um dado sísmico é um conjunto de tiros sísmicos.

Em termos gerais, o processamento sísmico tem como dado de entrada o dado sísmico, as características da fonte, a geometria da área geológica e alguns outros parâmetros específicos do problema. Com estes dados de entrada e usando modelos

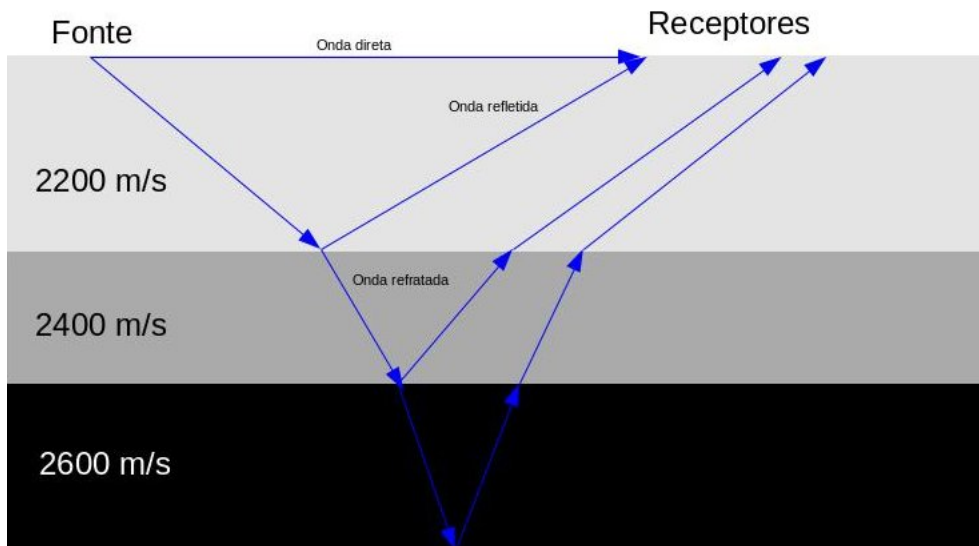


Figura 2.1: Esquema da propagação de ondas no subsolo.

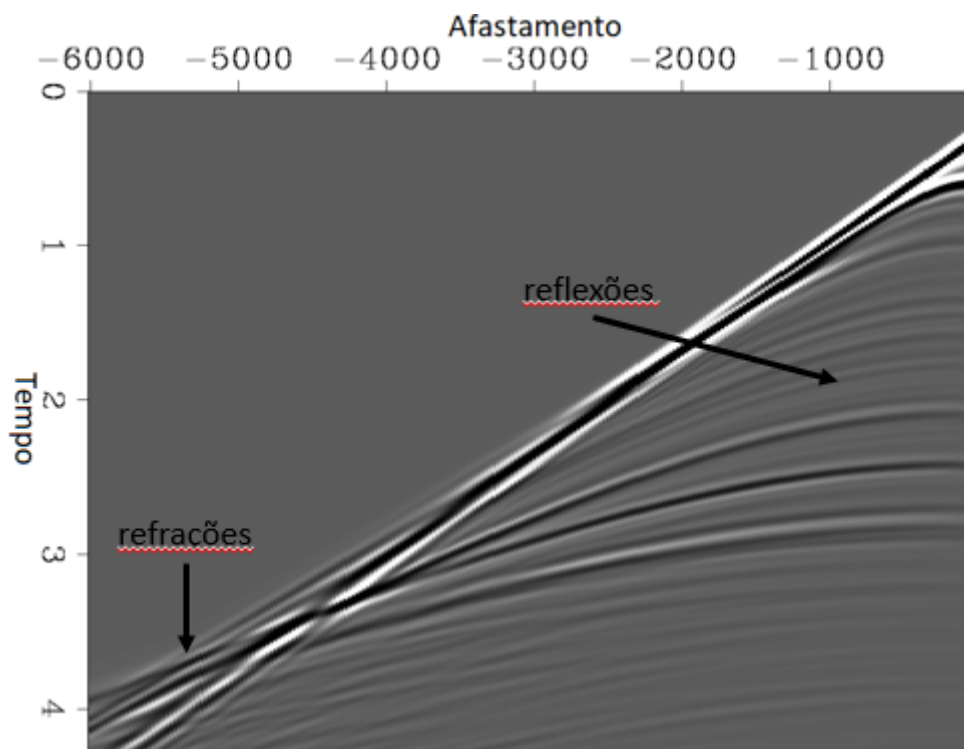


Figura 2.2: Primeiro tiro do dado sísmico Marmousi.

matemáticos, é possível obter informações sobre o subsolo, como por exemplo a espessura das camadas mostradas, bem como a sua velocidade de propagação de ondas.

2.1.1 Reflexões versus Refrações

O uso de refrações em processamento sísmico possui uma série de limitações quando comparado a reflexões.

Tanto as reflexões como as refrações são compostas por dois tipos de onda: P (compressional) e S (cisalhamento). Nas ondas P a direção da vibração do solo é a mesma direção da propagação da onda. Nas ondas S a direção da vibração é perpendicular a direção da propagação da onda.

Nas reflexões a quantidade de energia presente nas ondas S é muito pequena, podendo portanto ser desprezada sem nenhum prejuízo. No caso das refrações ambas as ondas, P e S, contribuem bastante para a energia sendo propagada. Os métodos de equação de onda utilizados neste trabalho, quando aplicados a refrações, necessitam que as componentes P e S sejam separadas em um processo chamado de *processamento de modo*. Este é um dos fatores que faz com que o uso das refrações demande maior tempo de processamento quando comparado com as reflexões.

Na Figura 2.2 podemos observar que as ondas refratadas, em sua maioria são observadas somente em longos afastamentos, no caso da Figura 2.2 de 5 km em diante. Por este motivo, há uma série de limitações no uso de ondas refratadas no processamento sísmico, quando comparado com o uso de ondas refletidas. Dados sísmicos com longos afastamentos não são tão disponíveis quanto dados com afastamentos curtos, além de serem maiores, demandando maior tempo de processamento.

A necessidade de dados com longos afastamentos para as refrações também limita ou mesmo inviabiliza sua aplicabilidade para camadas profundas. Isto porque, para tais cenários, o afastamento requerido pelas refrações é proibitivamente longo (dezenas de quilômetros).

2.1.2 Modelagem

O processo de modelagem tem o objetivo de modelar computacionalmente a propagação de onda no subsolo. A área do subsolo onde a propagação é modelada é representada computacionalmente por uma malha (2D ou 3D), onde se resolve a Equação 2.1 para cada ponto no espaço a cada instante de tempo. No nosso caso, utilizamos para isso um operador de diferenças finitas de ordem oito no espaço e quatro no tempo. O resultado deste processo é o dado sísmico sintético, cuja grandeza física sendo representada, no caso acústico, é a pressão. Na Figura 2.3 mostramos um tiro do dado sísmico gerado pela modelagem, utilizando o modelo de velocidades com três camadas apresentado na Figura 2.1.

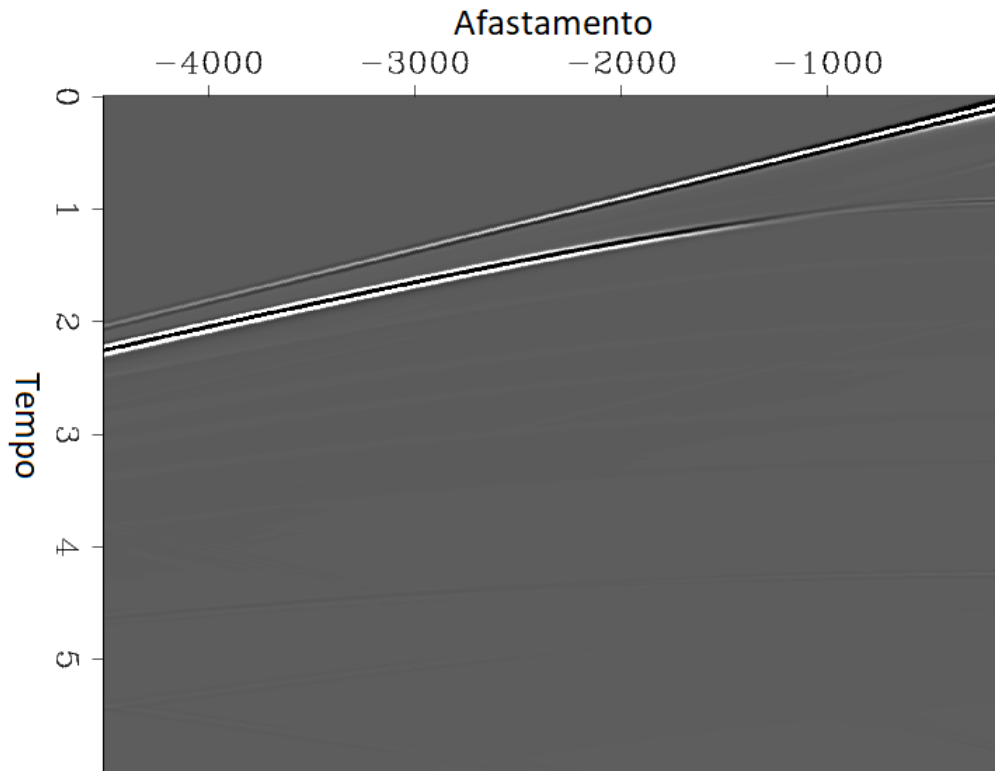


Figura 2.3: Tiro do dado sísmico gerado pelo modelo de velocidade da Figura 2.1.

2.1.3 Migração

A migração sísmica consiste em um processo de imageamento do subsolo, onde são representadas as estruturas presentes nas camadas do subsolo. A análise destas estruturas é um elemento fundamental do processo de investigação que determina a probabilidade da presença de hidrocarbonetos. A migração tem como dado de entrada o dado sísmico, que é uma informação em função do tempo, e o objetivo é gerar uma informação em função da posição no subsolo. No caso de reflexões, a informação de posição corresponde ao refletor, ou seja, o ponto no subsolo que gerou a reflexão que foi captada pelo sensor que registrou o dado sísmico.

Existem diversas técnicas de migração, das quais destacamos as baseadas em traçados de raios (Kirchoff) e as baseadas na equação de onda (RTM). Embora os métodos baseados em traçado de raios sejam mais eficientes computacionalmente, eles apresentam limitações para regiões com geologia complexa. Por exemplo, muitas descobertas recentes de campos de petróleo, como no Golfo do México e na bacia de Santos no Brasil, são localizadas em águas profundas e na presença de estruturas de sal complexas o que impõe desafios às técnicas de migração. Métodos tradicionais como o Kirchoff não apresentam bons resultados pois não tratam adequadamente grandes variações de velocidade lateral, tipicamente encontradas em campos pré-sal. Esta falta de acurácia coloca em cena métodos baseados na equação de onda,

como a RTM, que permite o imageamento em tais cenários, porém a um custo computacional muito mais alto [26–28].

Assim como na modelagem, a RTM resolve a equação de onda utilizando tipicamente um operador de diferenças finitas. Neste trabalho, utilizamos a equação de propagação de onda em sua versão acústica 2D dada por

$$\rho \frac{\partial^2 u}{\partial t^2} = v^2 \left(\frac{\partial}{\partial x} \left(\rho \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\rho \frac{\partial u}{\partial y} \right) \right), \quad (2.1)$$

onde u representa a pressão, v a velocidade, ρ a densidade, x, y as coordenadas no espaço e t o tempo.

Adotamos a equação acústica por simplicidade, entretanto acreditamos que os resultados podem ser reproduzidos com outras versões da equação de onda, como a elástica e a visco-elástica, tanto em 2D quanto em 3D.

Neste caso a equação é resolvida na sua forma direta e reversa no tempo. Feito isso, os campos de onda gerados, direto e reverso, são correlacionados de forma que a operação de correlação cruzada tem como resultado a indicação da posição dos refletores, conforme mostrado na Figura 2.4.

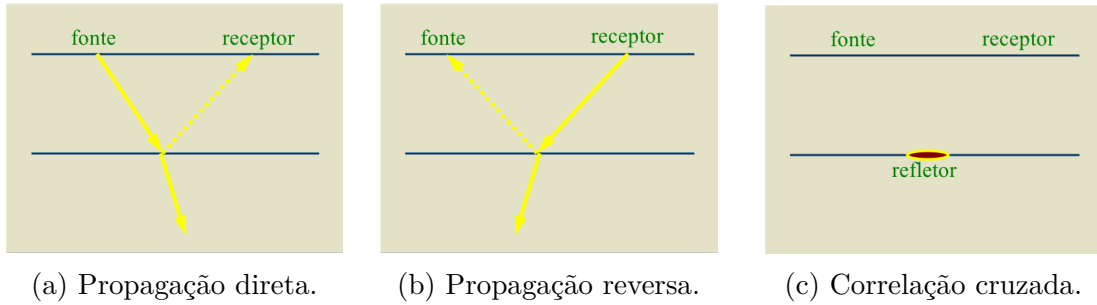


Figura 2.4: Etapas da migração reversa no tempo (RTM).

Quando efetuamos a correlação cruzada de vários tiros, temos vários pontos indicados como refletores, se empilharmos todas estas imagens podemos ter uma linha contínua que indica o contorno de um refletor, como mostrado na Figura 2.5.

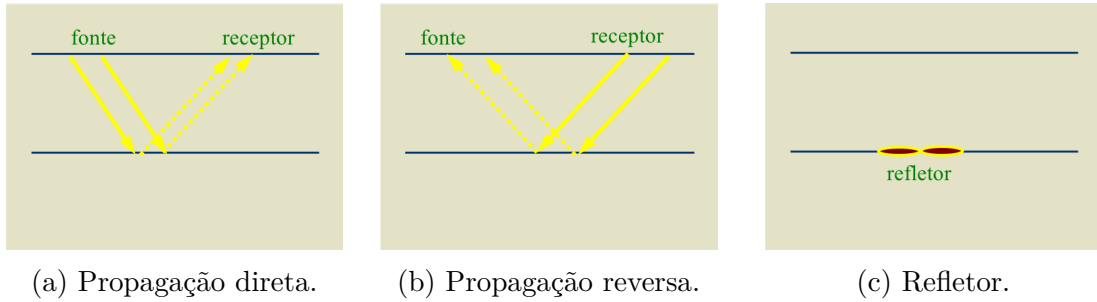


Figura 2.5: Em (a) e (b) temos propagação direta e reversa com dois tiros, em (c) as correlações dos dois tiros empilhadas mostrando o refletor.

Um exemplo da diferença de acurácia entre a migração Kirchoff e o RTM é mostrado na Figura 2.6, onde são mostradas duas imagens da mesma região geológica migradas com Kirchoff (a esquerda) e a RTM (a direita). Podemos observar que as laterais do corpo salino são melhor definidas com a RTM, o que mostra que esta técnica apresenta melhores resultados na presença de maiores contrastes laterais de velocidade.

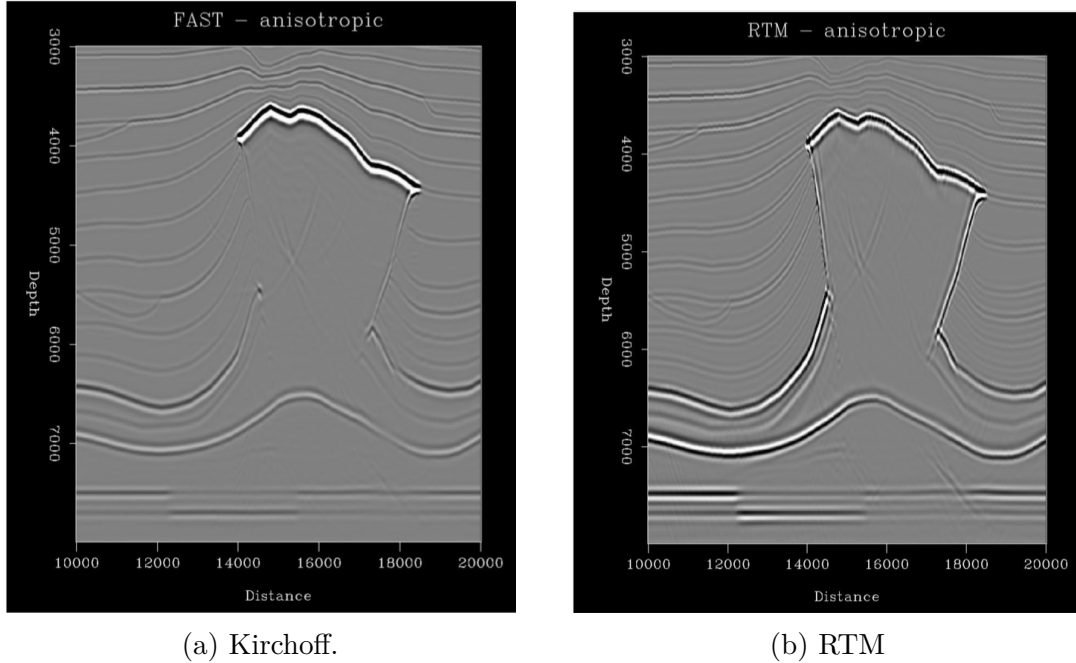


Figura 2.6: Kirchoff versus RTM: pode-se observar melhor definição das laterais do corpo de sal com a RTM (cortesia de Conoco-Philips).

2.1.4 Inversão

No contexto de processamento sísmico, a inversão consiste em derivar propriedades das rochas no subsolo a partir de um dado sísmico. Neste trabalho, utilizamos a inversão sísmica para obter a velocidade de propagação de ondas sísmicas no subsolo. Para tanto utilizamos a FWI para construir modelos de velocidade detalhados do subsolo.

A FWI é um método iterativo de casamento de dados baseado em gradiente, com o objetivo de ajustar os dados sintéticos modelados através da equação de onda, com os dados adquiridos em campo. Este casamento de dados é formulado como um problema de otimização que busca encontrar o mínimo global da função de custo mostrada na equação 2.2,

$$M(m) = \|D_{\text{acq}} - D_{\text{mod}}\|^2 \quad (2.2)$$

onde m é o vetor de parâmetros a ser estimado, neste trabalho representado por

cada ponto do modelo de velocidade. Já $M(m)$ é o *misfit*, que representa a diferença entre os dados adquiridos (D_{acq}) e os dados modelados (D_{mod}).

A FWI, assim como a RTM, resolve a equação de onda na forma direta e reversa no tempo, porém na FWI o que é propagado na parte reversa é a diferença entre o dado sísmico modelado e o adquirido em campo. Além disso, ao invés de uma correlação cruzada é calculado um vetor gradiente, que é utilizado na atualização do modelo de velocidade.

A título de comparação, mostramos na Figura 2.7 um modelo de velocidade gerado com a técnica de tomografia [29] (à esquerda) ao lado do mesmo modelo gerado com a FWI (à direita), onde podemos ver que a resolução da FWI é muito mais detalhada.

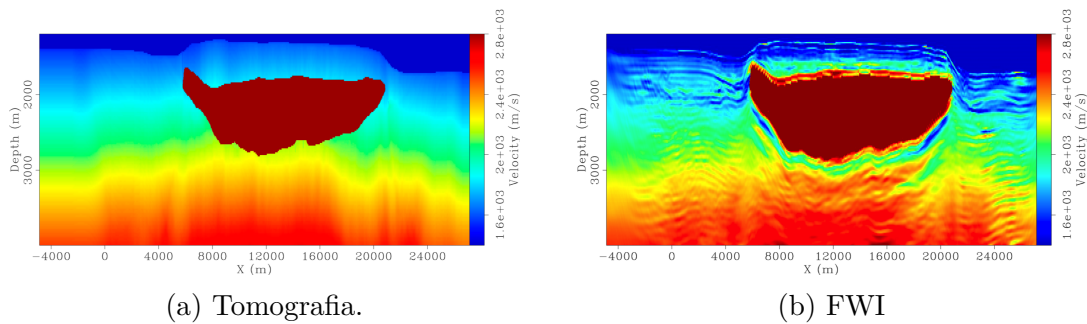


Figura 2.7: Tomografia versus FWI: pode-se observar que a resolução do modelo de velocidade é bem maior no caso da FWI (cortesia de GeoImaging Solutions Inc.)

As Figuras 2.8 e 2.9 mostram um exemplo mais detalhado do funcionamento da FWI para o dado sísmico Marmousi. Na Figura 2.8, podemos observar os dois principais dados de entrada da FWI: o dado sísmico e o modelo de velocidade inicial. Depois de algumas dezenas de iterações a FWI converge para o resultado mostrado na Figura 2.9a, que é bem próximo do modelo original de velocidade mostrado na Figura 2.9b.

2.2 Simulated Annealing

O simulated annealing [11] é uma variante do algoritmo de Monte Carlo, onde é gerada uma cadeia de Markov de N amostras de uma dada distribuição de probabilidades. A cada iteração do simulated annealing a amostra gerada pode ser aceita ou rejeitada de acordo com um critério probabilístico. Este critério depende dos valores da função de custo da amostra gerada e da anterior, e da variável *temperatura* que é decrementada conforme a iteração do simulated annealing. Caso a amostra gerada seja aceita ela substitui a amostra anterior se tornando a amostra atual, caso contrário a amostra anterior é mantida. Desta forma, o simulated annealing trata-se

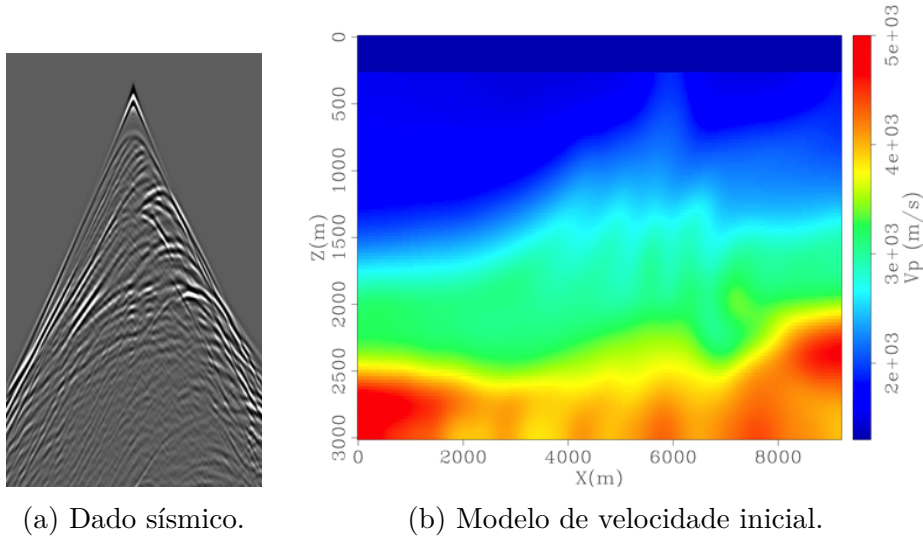


Figura 2.8: Em (a) é mostrado o dado sísmico de entrada para a FWI e em (b) é mostrado o modelo de velocidade inicial utilizado.

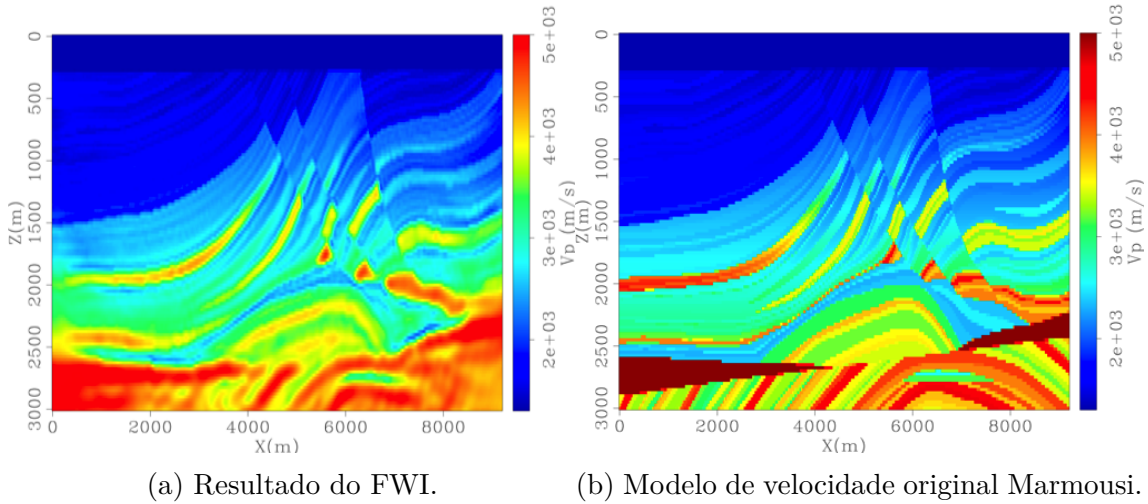


Figura 2.9: Em (a) temos o resultado da FWI, enquanto em (b) é mostrado o modelo de velocidade original.

de um algoritmo de otimização global que busca encontrar o valor mínimo de uma dada função de custo, para um dado espaço de variáveis.

Considerando uma simulação com N amostras, representadas por m_0, \dots, m_{N-1} , e fazendo $k = 0, \dots, N - 1$, o algoritmo de simulated annealing funciona como a seguir:

1. Gerar uma realização inicial das variáveis, m_0 . Seja $k = 0$.
2. Amostrando a distribuição a priori, gerar uma proposta de amostra do modelo de velocidade, m_{prop} , baseada em m_k .
3. A amostra proposta, m_{prop} , torna-se a próxima amostra, m_{k+1} , com probabi-

lidade

$$p_k = \min\{1, e^{-\Delta C_k/T_k}\}. \quad (2.3)$$

Caso contrário (i.e., com probabilidade $1 - p_k$), $m_{k+1} = m_k$. Na equação 2.3, $\Delta C_k = C_{\text{prop}} - C_k$, onde C_{prop} e C_k são os valores de função de custo da amostra proposta e da atual, respectivamente. T_k é o valor atual da temperatura do simulated annealing. Seja $k = k + 1$.

4. Repetir as etapas 2 e 3 enquanto $k < N$.

Estas etapas criam uma cadeia de Markov de amostras, onde cada amostra da iteração $k + 1$ é gerada a partir da amostrada da iteração k utilizando a distribuição de probabilidades predefinida. Quando temos $C_{\text{prop}} > C_k$ a nova amostra é sempre aceita, caso contrário, será aceita com probabilidade p_k . A função de custo é dependente do problema, sendo no nosso caso representada pela diferença entre o dado sísmico real observado e o dado modelado, como mostrado na equação 2.2.

Como pode ser observado pela equação 2.3 quanto maior a temperatura maior a probabilidade de uma proposta que piora a função de custo ser aceita, e quanto menor a temperatura menor esta probabilidade. Assim, uma simulação com o simulated annealing possui uma temperatura inicial (T_0), uma temperatura final (T_N), uma estratégia de decaimento de temperatura e um tamanho de *época*. Uma época é uma sequência de iterações na qual a temperatura permanece constante, sendo a mesma decrementada no final de cada época. Este controle da temperatura permite que a simulação seja muito exploratória no início, tornando-se progressivamente mais rigorosa com o número de iterações, o que é uma vantagem em relação a métodos anteriores como o Monte Carlo tradicional e o *Metropolis-Hastings* [30].

Capítulo 3

Construção de Modelos de Velocidade com Simulated Annealing

Neste capítulo, apresentamos a nossa abordagem para a construção de modelos iniciais de velocidade com baixo valor de número de onda (contraste suave de velocidade) que atendam aos requisitos de convergência da FWI.

3.1 Modelos de Velocidade Suaves

Modelos de velocidade suaves constituem a solução do problema inverso da sísmica com relação aos macro eventos geológicos. Neste trabalho, tais modelos são utilizados como ponto de partida da FWI, cuja convergência requer que o conjunto de variáveis que os compõe se encontre na mesma região de convexidade do mínimo global da função de custo. Um exemplo de modelo de velocidade suave é mostrado na Figura 3.1a, sendo o mesmo obtido a partir da suavização por média móvel, vertical e horizontal, do modelo de velocidade do dado sísmico Marmousi, mostrado na Figura 2.9b. Tal modelo de velocidade suave é utilizado como modelo inicial da FWI, que converge para o resultado mostrado na Figura 3.1b. Neste caso o modelo inicial de velocidade é obtido a partir do modelo real, que obviamente não estaria disponível na prática. Utilizamos este exemplo apenas como forma de apresentar as características do modelo de velocidade inicial que pretendemos estimar usando simulated annealing.

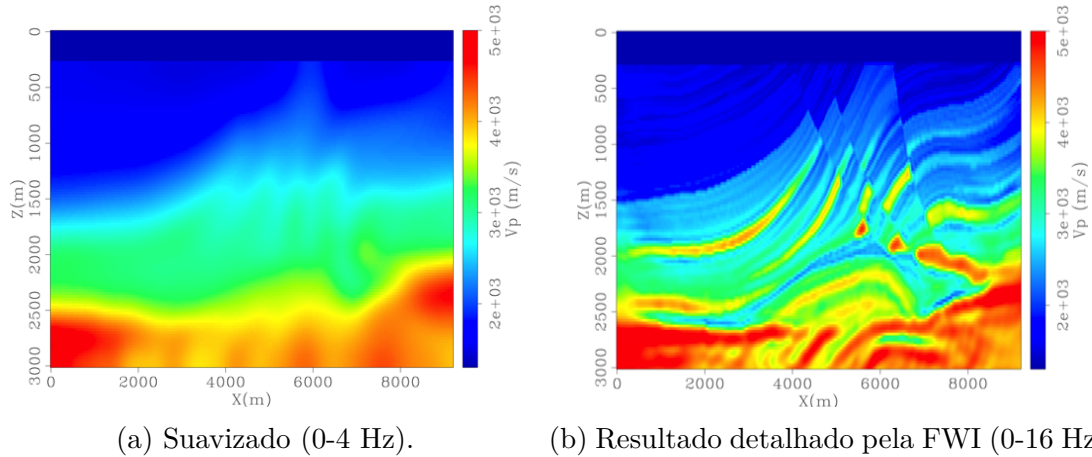


Figura 3.1: O modelo suave de velocidade (a) é usado como modelo inicial da FWI, que converge para o resultado apresentado em (b).

3.2 Formulação do Problema Inverso

O problema inverso na sismica tem como objetivo derivar as propriedades do subsolo utilizando o dado sísmico adquirido em campo. No nosso caso, derivamos o modelo de velocidade, representado por uma matriz 2D de variáveis. Assumimos a disponibilidade do dado sísmico correspondente, uma informação prévia sobre a geologia, e uma relação de modelagem direta entre o modelo de velocidades e o dado sísmico. Conforme demonstrado em [12, 13], a solução global para o problema inverso da sismica consiste em resolver a equação Bayesiana

$$\sigma(m) = c\rho(m)L(m), \quad (3.1)$$

onde m representa o vetor de parâmetros sendo estimado, que no nosso caso é o modelo de velocidades, $\sigma(m)$ representa a distribuição de probabilidade a *posteriori*, L representa a função de verossimilhança, ρ a distribuição a *priori*, e c é a constante de normalização.

No nosso estudo, o problema inverso possui um grande número de variáveis, dado que cada ponto do modelo de velocidade é um variável. Sendo assim, mesmo modelos em duas dimensões podem conter milhões de variáveis. Para problemas com tal quantidade de variáveis, a chamada maldição da dimensionalidade juntamente com a grande não linearidade das soluções, torna muito difícil obter de forma exata a distribuição a posteriori representada por σ . Por isso, propomos uma solução aproximada baseada na amostragem de σ utilizando o método simulated annealing tendo ρ como distribuição a priori. Assim, a solução do problema inverso corresponde à amostra com o menor valor de função de custo dentre todas geradas pelo simulated annealing. Para tanto, é gerada uma cadeia de Markov de N amostras da

distribuição a priori, podendo ser aceitas ou rejeitadas de acordo com o critério de aceitação do simulated annealing, discutido na seção 2.2. No nosso estudo, a função de custo é calculada pela diferença quadrática entre o dado sísmico modelado e o dado sísmico real, conforme mostrado na equação 2.2.

3.3 Descrição do Framework

Nesta seção vamos descrever nossas contribuições, conforme foram listadas na seção 1.2. Inicialmente abordaremos as etapas do framework proposto, a amostragem e o cálculo da função de custo, onde se encontram boa parte das nossas contribuições. Depois disso, descrevemos a stepping, que é o mecanismo que estamos propondo para controlar a abrangência da exploração do espaço de variáveis. Finalmente, apresentamos a forma como fazemos o resfriamento da temperatura do simulated annealing, onde apresentamos o método que criamos para calcular os valores de temperatura inicial e final.

3.3.1 Amostragem da Distribuição a Priori

Utilizamos o método de Gibbs [20] para amostrar nossa distribuição a priori baseado em simulação sequencial. Como parte do nosso método de amostragem, propomos três técnicas para reduzir a quantidade de iterações do simulated annealing: (1) redução do tamanho do espaço de variáveis; (2) focar a exploração em suas áreas mais relevantes; e (3) controle do quão exploratória é a simulação, dependendo da época do simulated annealing.

Reduzimos o tamanho do espaço de variáveis usando uma malha regular grossa para gerar as amostras do modelo de velocidade baseado no que foi proposto em [16]. Isso é possível porque as amostras do modelo de velocidade têm contraste suave. Assim, depois que uma amostra é gerada utilizando esta malha grossa, ela é interpolada para a malha fina requerida pelos métodos de equação de onda usados no cálculo da função de custo.

Para guiar a exploração do espaço de variáveis focando em suas áreas mais relevantes, nosso método emprega a técnica de amostragem baseada em estatística de múltiplos pontos proposta por [31]. Os autores propõem neste trabalho, gerar padrões no espaço das amostras usando uma distribuição a priori obtida a partir de uma imagem de treinamento. No nosso estudo, propomos utilizar esta imagem de treinamento para representar o conhecimento prévio da área geológica de interesse. Assim, esta informação guia a busca no espaço de variáveis evitando amostras irrelevantes, o que demandaria um número maior de iterações do simulated annealing

para convergir.

Na abordagem proposta por [31], a imagem de treinamento é percorrida cada vez que uma variável precisa ser amostrada. Isto significa que a imagem é percorrida MN vezes, onde M é a quantidade de pontos do modelo de velocidade e N é a quantidade de iterações, impondo um alto custo computacional. Como forma de diminuir este custo, em [32] é proposta uma abordagem onde a imagem de treinamento é percorrida apenas uma vez, no início da simulação, e todos os padrões contidos na imagem são armazenados em uma árvore de busca. Os padrões presentes nesta árvore são usados para gerar as amostras a cada iteração. A implementação dos autores desta abordagem é chamada de simulador Snesim.

No nosso trabalho, utilizamos o simulador Snesim para a geração de amostras, entretanto o mesmo suporta apenas valores discretos para cada célula da amostra: tais valores são chamados de categorias. Como utilizamos modelos de velocidade compostos de valores contínuos, após uma amostra ser gerada com categorias há uma segunda etapa para a aplicação da suavização, que transforma as variáveis representadas em categorias para valores em ponto flutuante. O processo de suavização utiliza uma média móvel nas duas direções, vertical e horizontal, e através dele obtemos o modelo de velocidade suave. A Figura 3.2a mostra uma amostra com valores de categorias gerada pelo Snesim, e a Figura 3.2b mostra sua versão suavizada.

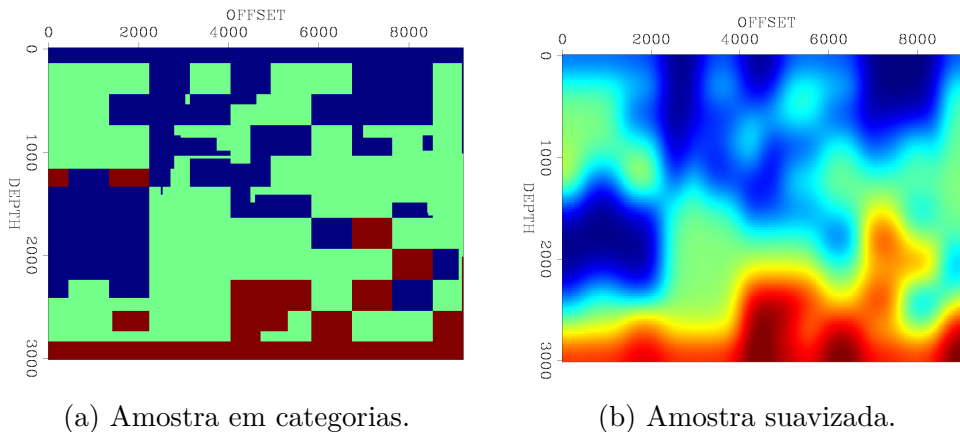


Figura 3.2: Amostra gerada pelo Snesim (a) e a mesma amostra após aplicada a suavização.

Considerando uma simulação com N amostras, representadas por m_0, \dots, m_{N-1} , e fazendo $k = 0, \dots, N - 1$, amostramos a distribuição a priori como a seguir:

1. Gerar uma amostra inicial das variáveis, m_0 . Ela pode ser gerada usando a técnica regular de amostragem da distribuição a priori, ou algo que o usuário considere razoável (por exemplo, camadas horizontais). Seja $k = 0$.

2. Escolher um subconjunto de variáveis de m_k aleatoriamente. Denotamos este subconjunto por m_{sub} .
3. Amostrando a distribuição a priori, gerar uma proposta de amostra do modelo de velocidades, m_{prop} , por

$$m_{\text{prop}} = \rho(m_{\text{sub}} | \widetilde{m}_{\text{sub}}), \quad (3.2)$$

onde $\widetilde{m}_{\text{sub}}$ é o subconjunto de variáveis não pertencentes a m_{sub} .

4. A amostra proposta, m_{prop} , torna-se a próxima amostra, m_{k+1} , com probabilidade

$$p_k = \min\{1, e^{-\Delta C_k/T_k}\}. \quad (3.3)$$

Caso contrário (i.e., com probabilidade $1 - p_k$), $m_{k+1} = m_k$. Na equação 3.3, $\Delta C_k = C_{\text{prop}} - C_k$, onde C_{prop} e C_k são os valores de função de custo da amostra proposta e da atual, respectivamente. T_k é o valor atual da temperatura do simulated annealing. Seja $k = k + 1$.

5. Repetir as etapas 2 a 4 enquanto $k < N$. A amostra com o menor valor de C_k para $k = 0, \dots, N$ é a solução desejada.

Estas etapas criam uma cadeia de Markov de amostras do modelo de velocidade, onde cada amostra da iteração $k + 1$ origina-se de uma perturbação em uma subárea da amostra da iteração k . A Figura 3.3 mostra um exemplo deste processo. Na Figura 3.3a, temos a amostra atual, a Figura 3.3b mostra a amostra atual com a subárea a ser amostrada em preto (m_{sub}), e a Figura 3.3c mostra a nova amostra gerada. A posição da subárea é definida aleatoriamente por uma distribuição de probabilidade uniforme.

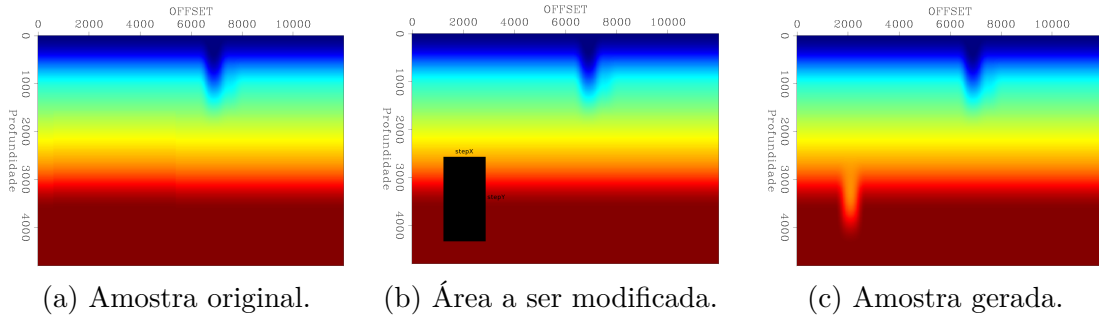


Figura 3.3: Em (a) é mostrada a amostra original, em (b) pode ser vista a seleção da área a ser perturbada, e em (c) o resultado da perturbação, gerando a amostra proposta.

Propomos utilizar o tamanho da subárea para controlar o quão exploratória é a simulação. Este tamanho é calculado baseado na época da simulação. O mesmo tem

inicialmente um valor alto para a primeira época e é decrementado para cada época, atingindo o menor valor na última época. Chamamos este processo de *stepping*. Ele permite fazer com que a simulação seja muito exploratória no início e restrita no final. Na seção 3.3.4 descrevemos em mais detalhes o processo de stepping.

3.3.2 Cálculo da Função de Custo

A função de custo é representada pela diferença quadrática entre o dado sísmico modelado e o adquirido em campo (equação 2.2). Em [16], é proposto o uso de ambos os tipos de ondas, refratada e refletida, para calcular o valor da função de custo. A utilização de ondas refratadas apresenta três desvantagens (subseção 2.1.1): (1) elas requerem dados sísmicos com longos afastamentos, que nem sempre estão disponíveis; (2) quando disponíveis, tais dados sísmicos demandam processamento de um maior volume de dados, além do processamento de modo (separação entre ondas P e S), portanto aumentando a demanda computacional; (3) sua aplicabilidade para camadas profundas é limitada ou mesmo inviável. Para evitar estas desvantagens, neste trabalho propomos um método de cálculo da função de custo baseado apenas nas reflexões. Entretanto, modelos de velocidade suaves não são capazes de gerar estas reflexões devido à ausência de grandes contrastes de velocidade. Um exemplo disso é mostrado na Figura 3.4, para o dado Marmousi. Na Figura 3.4a temos um modelo de velocidade suavizado obtido a partir do modelo de velocidade original do dado Marmousi, que é mostrado na Figura 3.4c. As Figuras 3.4b e 3.4d mostram o primeiro tiro dos dados sísmicos gerados pelo modelo de velocidade suavizado e o original, respectivamente. Podemos observar claramente a diferença entre estes dados sísmicos, onde o que foi gerado a partir da velocidade suavizada (Figura 3.4b) não apresenta reflexões, que por outro lado, estão presentes no dado sísmico produzido pelo modelo de velocidade original (Figura 3.4d). Propomos a seguir um método para gerar tais reflexões em modelos de velocidade suaves usando o contraste de densidade.

Contraste de Densidade

Para a geração do contraste de densidades, primeiro geramos uma imagem migrada usando a RTM em baixa frequência (0–20 Hz), aplicada ao dado adquirido e a amostra do modelo de velocidade. Então, calculamos o contraste de densidade ($\Delta\rho$) usando esta imagem migrada. Empregamos o contraste de densidade e a amostra do modelo de velocidade como parâmetros da modelagem sísmica, gerando um dado sísmico sintético com reflexões. Uma ilustração da relação entre a imagem migrada e o contraste de densidade é mostrado na Figura 3.5 para o dado sísmico Marmousi.

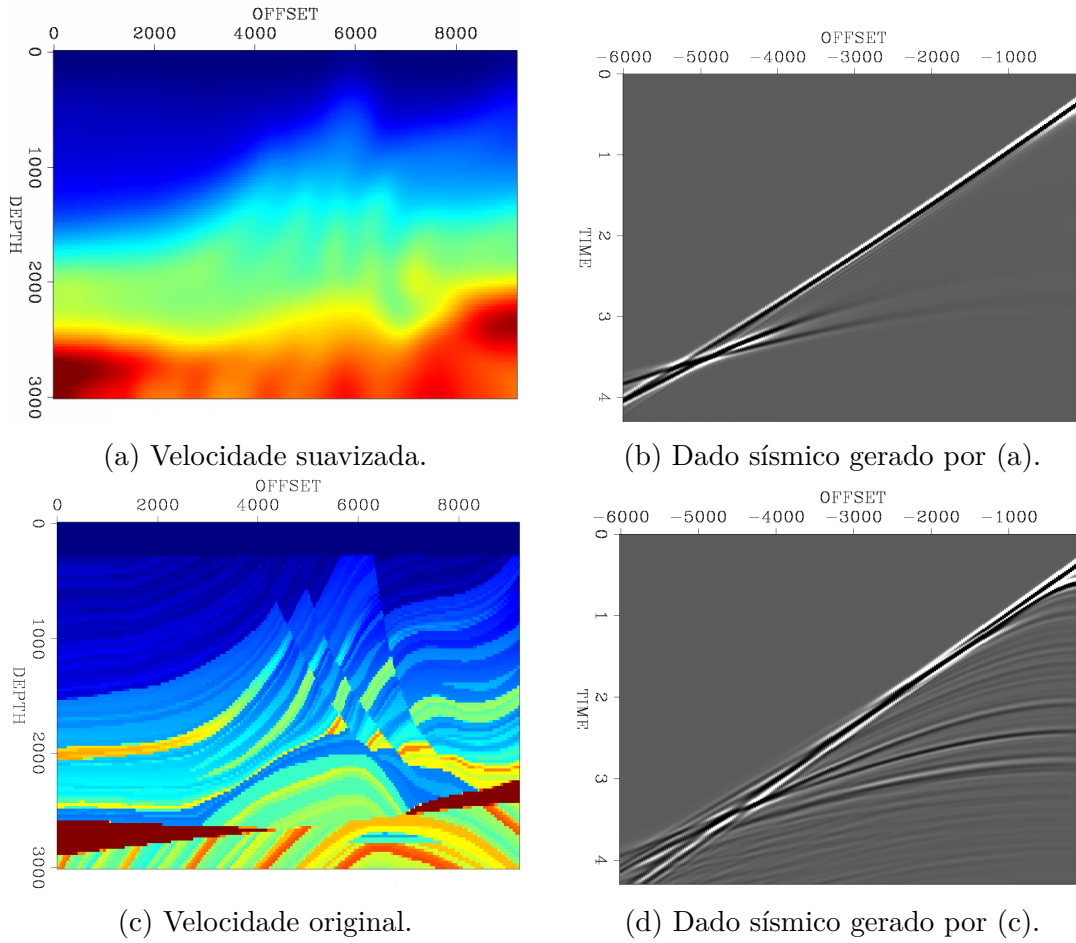


Figura 3.4: A velocidade suavizada (a) é usada para gerar o dado sísmico (b), que não apresenta reflexões, apenas refrações e a onda direta, ao contrário da velocidade real (c) que gera as reflexões mostradas no dado sísmico (d).

Fizemos a migração do dado Marmousi usando o modelo de velocidade suave da Figura 3.4a. Comparando a imagem migrada resultante mostrada na Figura 3.5a com a densidade mostrada na Figura 3.5b, podemos observar as estruturas da subsuperfície apropriadamente representadas (a densidade fixa na parte superior refere-se à camada de água).

A imagem migrada representa a refletividade da subsuperfície, e sua relação com a velocidade e a densidade é

$$R = \frac{\rho_2 V_2 - \rho_1 V_1}{\rho_2 V_2 + \rho_1 V_1}, \quad (3.4)$$

onde R representa a refletividade, ρ_1 e V_1 representam a densidade e a velocidade em um ponto qualquer do subsolo, e ρ_2 e V_2 a densidade e a velocidade em um ponto adjacente. A partir da equação 3.4 propomos que o contraste de densidade pode ser obtido conforme a seguir. Para um modelo suave de velocidades assumimos $V_2 \approx V_1$,

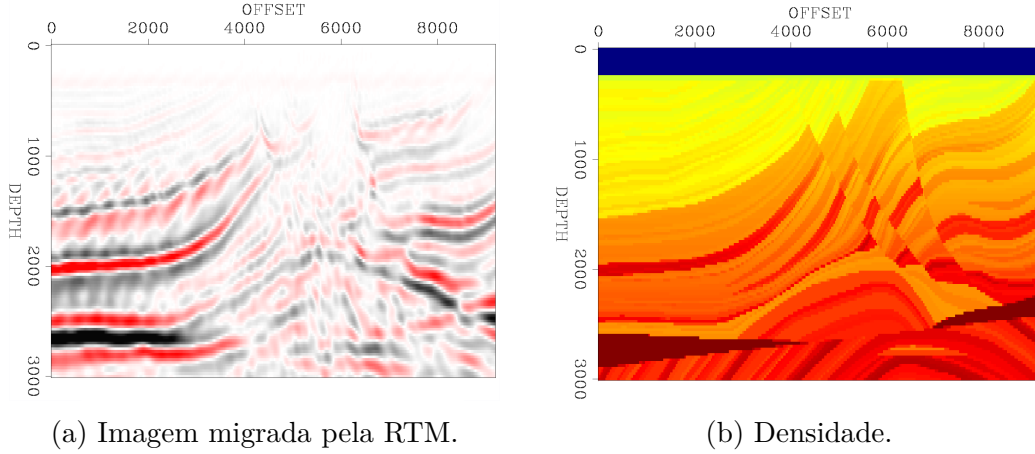


Figura 3.5: A imagem migrada (a) foi computada pela RTM usando o modelo de velocidades suave da Figura 3.4a. Nela pode-se observar a similaridade com a densidade (b).

obtendo

$$R \approx \frac{\rho_2 V_1 - \rho_1 V_1}{\rho_2 V_1 + \rho_1 V_1} = \frac{\rho_2 - \rho_1}{\rho_2 + \rho_1}. \quad (3.5)$$

Fazendo $\rho_2 = \rho_1 + \Delta\rho$, isto é simplificado para

$$R \approx \frac{\rho_1 + \Delta\rho - \rho_1}{\rho_1 + \Delta\rho + \rho_1} = \frac{\Delta\rho}{2\rho_1 + \Delta\rho}. \quad (3.6)$$

Fazendo $\rho_1 = \rho$, e para $\Delta\rho \ll \rho$, obtemos

$$\Delta\rho \approx 2\rho R. \quad (3.7)$$

Como estamos apenas interessados no contraste de densidade, podemos usar

$$\Delta\rho \propto R \quad (3.8)$$

para um fator apropriado. Portanto, temos

$$\Delta\rho = fR, \quad (3.9)$$

onde a variável f é um fator de escala para ajustar o intervalo de valores da refletividade para a mesma ordem de magnitude da densidade, que é em geral entre 1 kg/m³ and 3 kg/m³. A Figura 3.6a mostra o contraste de densidade resultante da aplicação da equação 3.9 usando a imagem migrada da Figura 3.5a. Usando este contraste de densidade e o modelo de velocidade suave como parâmetros da modelagem sísmica, geramos o dado sísmico sintético contendo reflexões mostrado na Figura 3.6b. Este dado sísmico sintético é usado para calcular a função de custo de acordo com a equação 2.2. Este método é utilizado para calcular o valor da função

de custo de cada amostra gerada pelo simulated annealing.

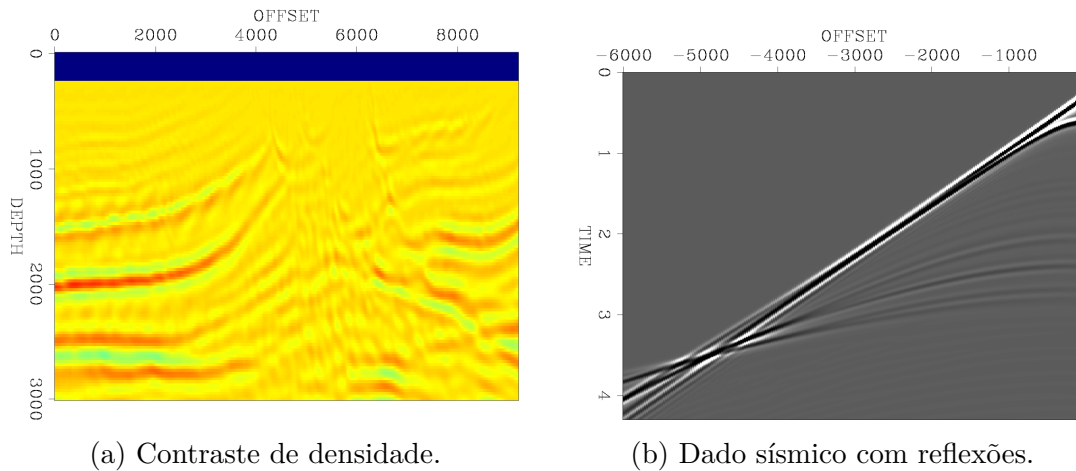


Figura 3.6: Contraste de densidade (a) é usado para gerar as reflexões do dado sísmico (b).

3.3.3 Resumo do Framework das Iterações

Os métodos de amostragem e cálculo da função de custo descritos fazem parte das etapas de cada iteração das simulações do nosso framework simulated annealing. Na Figura 3.7, apresentamos um workflow resumindo estas etapas. Uma amostra do modelo de velocidade é gerada, então ela é usada com o dado sísmico original para produzir uma imagem migrada. O contraste de densidade é calculado a partir desta imagem. Usando a amostra do modelo de velocidade e o contraste de densidade, é modelado o dado sísmico sintético correspondente. Então, o dado sintético e o original são utilizados para calcular o valor da função de custo, que é usada para calcular a probabilidade p_k (equação 3.3). Esta probabilidade define se a amostra proposta substitui a amostra atual ou não.

Todo o framework simulated annealing foi implementado utilizando a linguagem *Python*, ao passo que o processamento sísmico (RTM, modelagem etc) utilizou ferramentas já desenvolvidas anteriormente [9, 26]. Com relação a tais ferramentas, nosso framework não impõe que nenhuma implementação específica seja utilizada, podendo os nossos resultados serem reproduzidos com quaisquer ferramentas de RTM e modelagem disponível.

3.3.4 Stepping

O tamanho da subárea (conforme exemplificado preto na Figura 3.3), define o quão exploratória é a simulação na época correspondente àquela iteração. O processo de stepping tem como objetivo controlar este tamanho para cada época, para que a simulação seja bastante exploratória no início e focada no final.

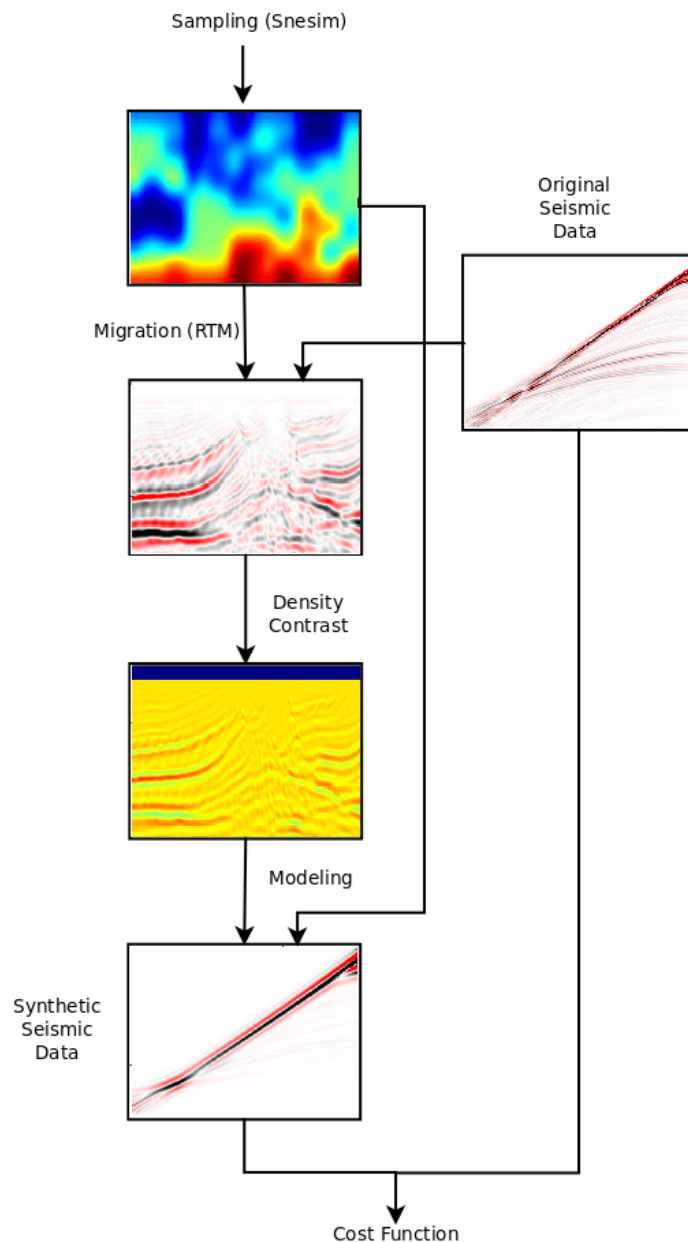


Figura 3.7: Workflow de uma iteração do simulated annealing.

A simulação possui N iterações, correspondendo a uma sequência não crescente A_0, \dots, A_{N-1} de tamanhos de subárea (com $A_{N-1} < A_0$). Cada um corresponde a uma amostra diferente do modelo de velocidade. Em nossa abordagem, o tamanho da subárea permanece constante para uma dada época e é decrementado ao fim da época. Então, as iterações de uma simulação são divididas em $S = N/E$ épocas, onde E é o número de iterações de uma época. Por simplicidade, assumimos que E é um divisor de N (i.e., N/E é inteiro). O tamanho A_k é decrementado a cada época para garantir que a step da exploração é a maior possível no início da simulação e a

menor no final. Decrementamos A_k exponencialmente como

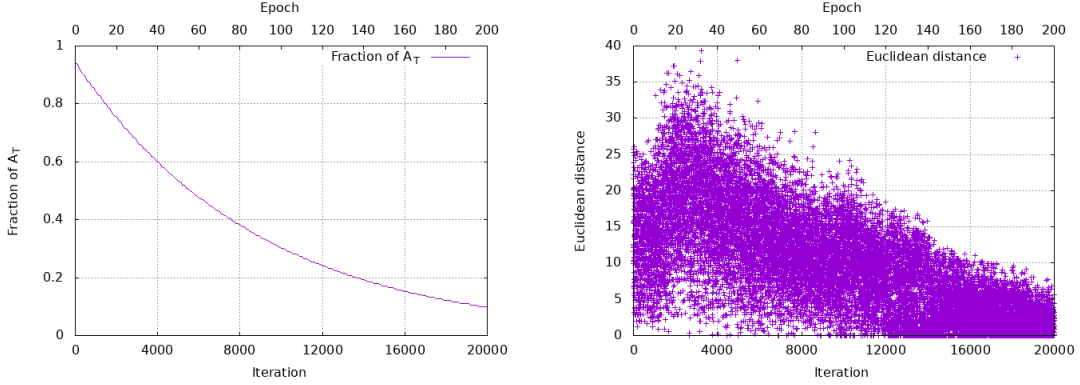
$$A_k = A_0 \left(\frac{A_{N-1}}{A_0} \right)^{\lfloor k/E \rfloor / (S-1)}. \quad (3.10)$$

A escolha apropriada dos valores para A_0 e A_{N-1} é fundamental para a convergência do simulated annealing. É possível definir estes valores experimentalmente, executando diversas simulações em uma abordagem de tentativa e erro, porém isso demandaria uma grande quantidade de simulações. Propomos evitar este grande número de simulações de ajuste fino conforme a seguir.

Existe uma relação de compromisso entre o tamanho A_k e o quão exploratória a simulação é. O valor de A_k define o tamanho do subconjunto de variáveis a ser amostrada (retângulo preto na Figura 3.3). Conseqüentemente, ele também define o tamanho de seu dado condicional, representado pela subárea fora de A_k ($\widetilde{m}_{\text{sub}}$, na equação 3.2). Denotando por A_T a área total do modelo de velocidades e \widetilde{A}_k a área fora de A_k , temos $A_T = A_k + \widetilde{A}_k$. Considerando que A_T é constante, se nós temos $A_k \rightarrow A_T$, conseqüentemente temos $\widetilde{A}_k \rightarrow 0$. Isto faz a simulação menos exploratória, pois a quantidade de dados condicionais ($\widetilde{m}_{\text{sub}}$) se torna muito pequena, limitando a quantidade de amostras diferentes que podem ser geradas por ele. O oposto também é verdadeiro, pois $\widetilde{A}_k \rightarrow A_T$ leva a $A_k \rightarrow 0$, também fazendo a simulação menos exploratória pois isso diminui a quantidade de variáveis amostradas. Assim, há um valor ótimo A_0 tal que $A_T > A_0 > 0$ que maximiza a exploração. Por outro lado, A_{N-1} deve ser definido de forma a focar a exploração no final da simulação, pois isso refina o modelo de velocidade encontrado nas épocas anteriores. Assim, temos A_0 e A_{N-1} tais que $A_T > A_0 > A_{N-1} > 0$.

Escolhemos A_0 e A_{N-1} gerando uma cadeia de Markov de amostras do modelo de velocidade, usando nossa técnica de amostragem da distribuição a priori. Como apenas geramos as amostras, o tempo de execução é desprezível quando comparado ao tempo de uma simulação completa (menos de 1%). Denotamos por N_A o número de amostras geradas e tais amostras por s_0, \dots, s_{N_A-1} . Então, calculamos a distância Euclidiana de cada par (s_k, s_{k+1}) com o inteiro k no intervalo $[0, N_A - 2]$. Assumimos que as etapas mais exploratórias correspondem àquelas com os maiores valores de distância Euclidiana. Similarmente, as etapas menos exploratórias correspondem aos menores valores. Portanto, definimos A_0 como a área que corresponde à época com os maiores valores da distância Euclidiana, e A_{N-1} à época com os menores valores desta distância. A Figura 3.8 mostra nossa abordagem aplicada ao dado Marmousi. Neste experimento, geramos uma cadeia de Markov com $N_A = 2 \times 10^4$ amostras, onde $A_0 = 0.95A_T$ e $A_{N-1} = 0.1A_T$, e usamos uma época $E = 100$. A Figura 3.8a mostra o decaimento exponencial de A_k para cada uma das N_A iterações, onde o eixo vertical mostra o valor de A_k como uma fração da área total A_T . A Figura

3.8b mostra no eixo vertical a distância Euclideana entre cada par de amostras adjacentes s_k e s_{k+1} , com k no intervalo $[0, N_A - 2]$.



(a) Tamanho de A_k como percentual de A_T . (b) Distância Euclideana entre amostras adjacentes do modelo de velocidade.

Figura 3.8: (a) mostra o tamanho de A_k decrementado exponencialmente de A_0 para A_{N-1} e (b) mostra a distância Euclideana entre cada par adjacente de amostras do modelo de velocidade.

Na Figura 3.8b, podemos observar o trade-off relacionado a A_k . A simulação é menos exploratória para A_k no intervalo $[0.90A_T, 0.95A_T]$, que para valores de A_k próximos a $0.8A_T$. Por outro lado, $A_k = 0.1A_T$ garante pequenos steps de exploração conforme requerido no final da simulação. Baseado nesses resultados, usamos $A_0 = 0.8A_T$ e $A_{N-1} = 0.1A_T$ para nossas simulações.

Resfriamento

O termo resfriamento se refere ao processo de decaimento da temperatura do simulated annealing. Ele controla a parte de exploração da simulação. A temperatura (T_k) é parte do cálculo de p_k como mostrado na equação 3.3. Altos valores de temperatura fazem $p_k \rightarrow 1$, enquanto baixos valores de temperatura fazem $p_k \rightarrow 0$. Similarmente a A_0 e A_{N-1} , os valores extremos da temperatura, T_0 e T_{N-1} , são fundamentais para a convergência das simulações. Propomos calcular os valores de T_0 e T_{N-1} baseando-se nos valores de A_0 e A_{N-1} , como a seguir.

Executando o nosso algoritmo de simulated annealing com $p_k = 1$ (sempre aceitando a amostra) e mantendo A_k fixo para todas as iterações nos permite medir a correlação entre um valor fixo de A_k e sua correspondente distribuição de valores de ΔC_k . Portanto, usando os valores já definidos de A_0 e A_{N-1} , podemos gerar as distribuições de valores de ΔC_k correspondentes a $k = 0$ e $k = N - 1$. Como mostrado na Figura 3.8b, os valores ótimos de step para o dado Marmousi são $A_0 = 0.8A_T$ e $A_{N-1} = 0.1A_T$. Usando estes valores para gerar a distribuição de ΔC_k para a primeira e a última época, obtemos o resultado mostrado na Figura 3.9, onde são

mostrados os histogramas normalizados dos valores de ΔC_k . Ambas as distribuições contém 4×10^3 valores. A Figura 3.9a mostra os resultados para a primeira época ($A_k = A_0 = 0.8A_T$) e a Figura 3.9b para a última época ($A_k = A_{N-1} = 0.1A_T$). Nota-se que os valores de ΔC_k em cada distribuição são impactados por A_k . Para $A_k = A_0 = 0.8A_T$, temos uma faixa mais larga de valores de ΔC_k quando comparado com $A_k = A_{N-1} = 0.1A_T$.

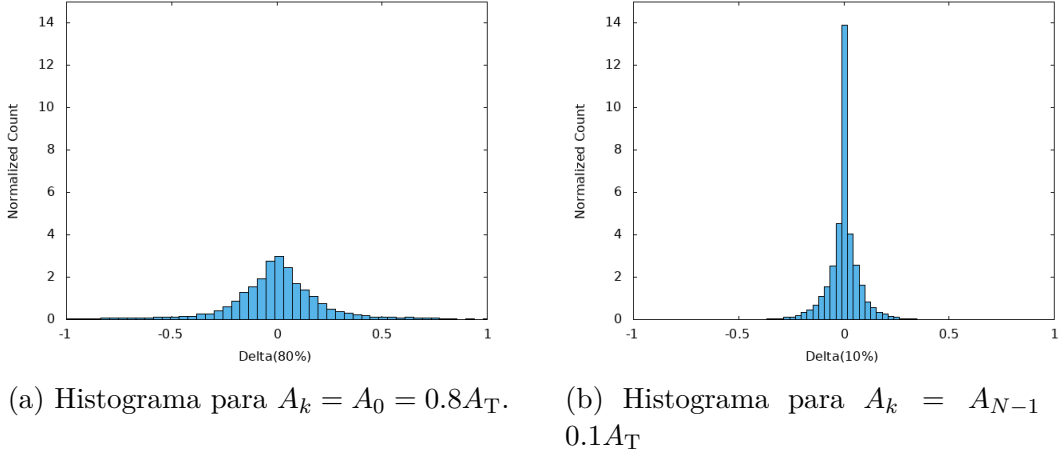


Figura 3.9: Histogramas normalizados de ΔC_k para a simulação com 4×10^3 iterações e $A_k = 0.8A_T$ (a) e para $A_k = 0.1A_T$ (b).

Do critério de aceitação do simulated annealing (equação 3.3), para $\Delta C_k \leq 0$ temos uma decisão determinística de aceitar a amostra. Quando $\Delta C_k > 0$, o objetivo é que p_k seja próximo de 1 para $k = 0$ e próximo de 0 para $k = N - 1$. Isto pode ser controlado fazendo decisões cruciais relacionadas à distribuição dos valores de ΔC_k para a primeira e a última época. Assim, os valores de T_0 and T_{N-1} podem ser determinados apropriadamente.

Similarmente ao stepping, assumimos uma sequência não crescente T_0, \dots, T_{N-1} de temperaturas (com $T_{N-1} < T_0$), subdividida em $S = N/E$ épocas. A temperatura é decrementada exponencialmente, como definida por

$$T_k = T_0 \left(\frac{T_{N-1}}{T_0} \right)^{\lfloor k/E \rfloor / (S-1)}. \quad (3.11)$$

Baseado no que foi observado, uma decisão pode ser feita a respeito do valor máximo de $\Delta C_k > 0$ para a primeira época que deve ser aceito (deixá-lo ilimitado requeriria que T_0 fosse muito alta). Também, precisamos decidir o valor máximo para a última época que deve ser tolerado (não tolerar nenhum valor positivo requeriria que T_{N-1} fosse muito pequena). Denotamos estes dois valores por ΔC_0^+ e ΔC_{N-1}^+ , respectivamente, e fazemos $\epsilon > 0$ um valor próximo a zero. Mirando $p_0 > 1 - \epsilon$ e $p_{N-1} < \epsilon$,

segue-se que precisamos ter

$$T_0 > \frac{-\Delta C_0^+}{\ln(1 - \epsilon)} \quad (3.12)$$

e

$$T_{N-1} < \frac{-\Delta C_{N-1}^+}{\ln(\epsilon)}. \quad (3.13)$$

Encontramos ΔC_0^+ e ΔC_{N-1}^+ usando a distribuição de valores de ΔC_k correspondentes ao tamanhos de subárea A_0 e A_{N-1} , respectivamente, como mostrado na Figura 3.9. Considerando apenas os valores positivos de ΔC_k em ambos os casos e definindo μ como um inteiro, escolhemos ΔC_0^+ como o μ -ésimo maior valor da distribuição ΔC_k para $A_0 = 0.8A_T$ (Figura 3.9a). Similarmente, escolhemos ΔC_{N-1}^+ como o μ -ésimo menor valor da distribuição de ΔC_k para $A_{N-1} = 0.1A_T$ (Figura 3.9b). Tendo os valores de ΔC_0^+ e ΔC_{N-1}^+ , podemos usá-los diretamente para calcular T_0 e T_{N-1} pelas equações 3.12 e 3.13.

O Tamanho da Época

Assumindo uma quantidade fixa de iterações, N , quanto maior a época, menor a quantidade de épocas. Definimos α e τ como a taxa de decrescimento da subárea e a temperatura para épocas adjacentes, respectivamente. Para um valor de k correspondente à última iteração antes de uma nova época iniciar temos

$$\alpha = \frac{A_{k+1}}{A_k} \quad (3.14)$$

e

$$\tau = \frac{T_{k+1}}{T_k}. \quad (3.15)$$

Ambos α e τ podem também ser calculados usando os valores extremos de subárea e temperatura, e o número de épocas S . Para a subárea, pelas equações 3.10 and 3.14 temos

$$\alpha = \left(\frac{A_{N-1}}{A_0} \right)^{1/(S-1)}. \quad (3.16)$$

Para a temperatura, pelas equações 3.11 e 3.15 temos

$$\tau = \left(\frac{T_{N-1}}{T_0} \right)^{1/(S-1)}. \quad (3.17)$$

As equações 3.16 e 3.17 representam o trade-off relacionado ao tamanho da época E (dado que $S = N/E$) para um número fixo de iterações. Para tamanhos maiores de época, os parâmetros do simulated annealing permanecem os mesmos por mais iterações, mas sua taxa de decaimento entre épocas adjacentes é maior (menores valores de α e τ). Escolhendo valores ótimos de α e τ nos permite calcular o

tamanho da época a partir dos valores inicial e final da subárea e temperatura usando as equações 3.16 e 3.17.

3.4 Execução Paralela das Simulações

O simulated annealing necessita de um elevado número de iterações para convergir, podendo este número de iterações chegar a dezenas de milhares ou mesmo milhões, no caso de problemas 3D. Por este motivo, propomos uma abordagem paralela do nosso algoritmo, como forma de tirar proveito do poder computacional oferecido pelos supercomputadores atuais. No nosso caso, as iterações permanecem sendo executadas sequencialmente, porém criamos uma versão paralela de duas das etapas mostradas no workflow da Figura 3.7, a migração (RTM) e a modelagem, que representam mais de 70% do tempo de execução de uma iteração, conforme veremos adiante nos nossos resultados.

Como o processamento de cada tiro sísmico é independente dos demais, o conjunto de tiros é dividido em subconjuntos de igual tamanho, e cada um desses subconjuntos é processado por um nó de computação. Sendo assim, tanto a RTM como a modelagem são executadas por cada nó de computação apenas no que se refere ao subconjunto de tiros sísmicos associados ao nó. Seja N_T o número de tiros do dado sísmico e M o número de nós computacionais. Por simplicidade, assumimos que N_T é um múltiplo de M , logo, cada nó computacional é responsável pelo processamento de $M_T = N_T/M$ tiros sísmicos adjacentes. Denotando por t_0, \dots, t_{N_T-1} os tiros sísmicos e m_0, \dots, m_{M-1} os nós computacionais, cada nó computacional m_i é responsável pelo processamento dos tiros sísmicos $t_{i*M_T}, \dots, t_{(i+1)*M_T-1}$.

A cada iteração, uma amostra do modelo de velocidade é gerada pelo nó m_0 e enviada a todos os demais nós m_1, \dots, m_{M-1} . Na sequência, todos os nós geram a imagem parcial migrada pela RTM correspondente a área coberta por seus respectivos tiros, utilizando a amostra do modelo de velocidade gerada. As imagens parciais geradas pelos nós m_1, \dots, m_{M-1} são então enviadas para o nó m_0 que as empilha. O processo de empilhamento das imagens parciais consiste em tratá-las como matrizes 2D e efetuar a soma elemento a elemento, originando a imagem empilhada. A partir da imagem empilhada o nó m_0 calcula localmente o contraste de densidade e o envia para os demais nós. Assim, usando a amostra do modelo de velocidade e o contraste de densidade, cada nó de computação executa a modelagem sísmica correspondente ao subconjunto de tiros sísmicos que lhe foi associado. Feito isso, cada nó de computação calcula uma função de custo parcial, baseada nos dados modelados localmente e seus respectivos tiros sísmicos do dado original. Uma vez calculados os valores das funções de custo parciais, os nós m_1, \dots, m_{M-1} enviam estes valores para o m_0 , que faz o somatório de todos, obtendo o valor da função de

custo total, que é utilizada pelo critério de aceitação do simulated annealing.

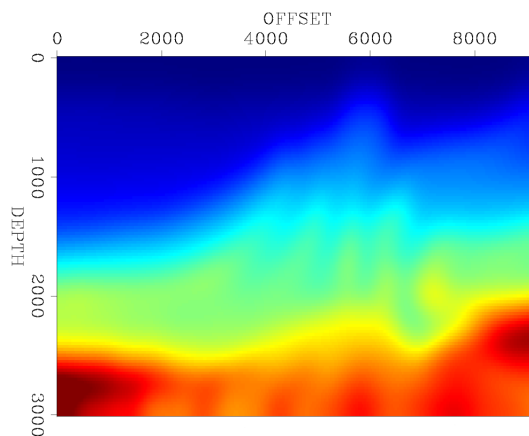
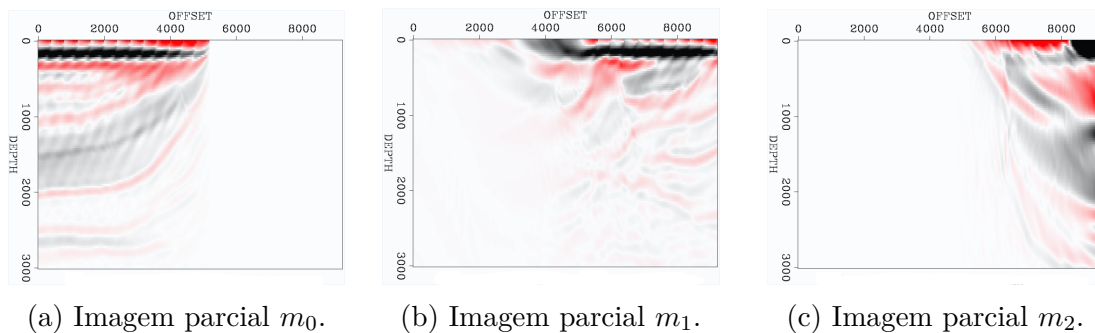


Figura 3.10: Amostra de velocidade suavizada baseada no dado Marmousi

Por exemplo, suponhamos que em uma determinada iteração do algoritmo é gerada uma amostra como a da Figura 3.10, e que o processamento é feito por 3 nós computacionais (m_0, m_1, m_2). A Figura 3.11 mostra a imagem migrada parcial gerada pela RTM por cada um dos nós de computação.



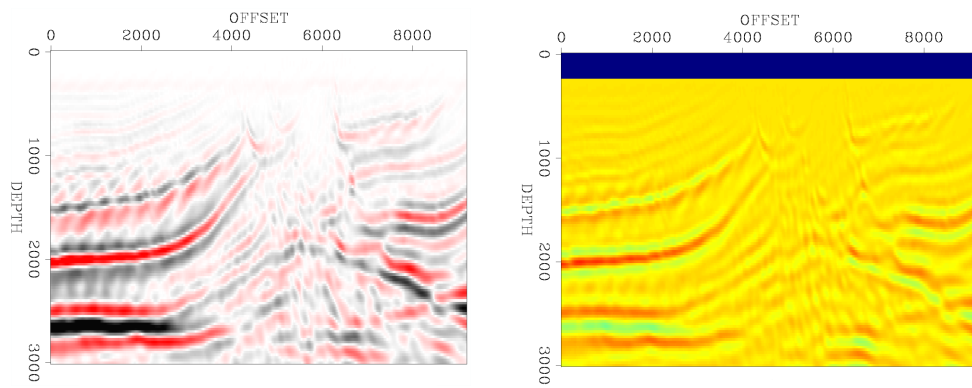
(a) Imagem parcial m_0 .

(b) Imagem parcial m_1 .

(c) Imagem parcial m_2 .

Figura 3.11: Imagens parciais geradas pela RTM por cada um dos nós de computação.

No exemplo da Figura 3.11 as imagens parciais dos nós m_1 e m_2 são enviadas para o nó m_0 , que é responsável por empilhá-las, originando a imagem empilhada mostrada na Figura 3.12a, através da qual é calculado o contraste de densidade da Figura 3.12b. Com esta paralelização conseguimos diminuir consideravelmente o tempo de execução de cada iteração, conforme apresentado na seção de resultados.



(a) Imagem migrada empilhada.

(b) Contraste de densidade.

Figura 3.12: Imagem empilhada (a) resultado das imagens mostradas na Figura 3.11. Contraste de densidade (b).

Capítulo 4

Avaliação Experimental

Nesta seção mostramos os resultados do nosso estudo utilizando o dado sísmico Marmousi. Avaliamos diversos parâmetros que influenciam a convergência das simulações, tais com o número de iterações, temperatura, step e época.

4.1 Ambiente de Execução

As simulações foram executadas em 10 nós de computação, em que cada nó de computação possui dois processadores 2695v2 Ivy Bridge. Cada processador possui 12 cores e executa a 2.4GHz, totalizando 24 cores por nó de computação, e 240 cores para cada simulação, utilizando 10 nós de computação. Os nós computacionais são interligados por uma rede *infiniband*, que também é utilizada no acesso ao armazenamento de dados central. Esta infraestrutura foi oferecida por dois supercomputadores: o Santos Dumont, do LNCC (Laboratório Nacional de Computação Científica); e o Lobo Carneiro, do NACAD/UFRJ (Núcleo Avançado de Computação de Alto Desempenho).

Utilizamos a versão paralela do framework descrita na seção 3.4. O paralelismo intra nó foi implementado nas ferramentas de RTM e modelagem utilizando OpenMP, juntamente a otimizações de acesso a memória (*cache blocking*, *domain decomposition*), apresentadas em um nosso trabalho anterior [26]. O paralelismo inter nó demanda trocas de mensagens, para as quais utilizamos diretamente a biblioteca de *sockets* oferecida pela linguagem Python: no início da simulação é criada uma conexão TCP/IP entre o nó m_0 (seção 3.4) e os demais nós de computação.

4.2 Dados Experimentais

A Figura 4.1a mostra o modelo de velocidade Marmousi, e a Figura 4.1b mostra o modelo de densidade. Tanto o modelo de densidade como o de velocidade têm 3 km de profundidade e 9 km em afastamento, com uma grade de 25 m, o que gera aproximadamente 4.5×10^4 pontos (variáveis). O dado sísmico tem originalmente 340 registros de tiros com 6 km de afastamento, com receptores separados por 25 m. A origem da fonte de cada tiro é separada por 50 m e a amostragem de tempo é de 2 milissegundos. Em nossos experimentos, utilizamos apenas um subconjunto de 40 tiros separados igualmente (i.e., decimados por um fator de 8). As ferramentas da modelagem sísmica e da RTM empregadas são baseadas em um operador de diferenças finitas de oitava ordem no espaço e quarta ordem no tempo.

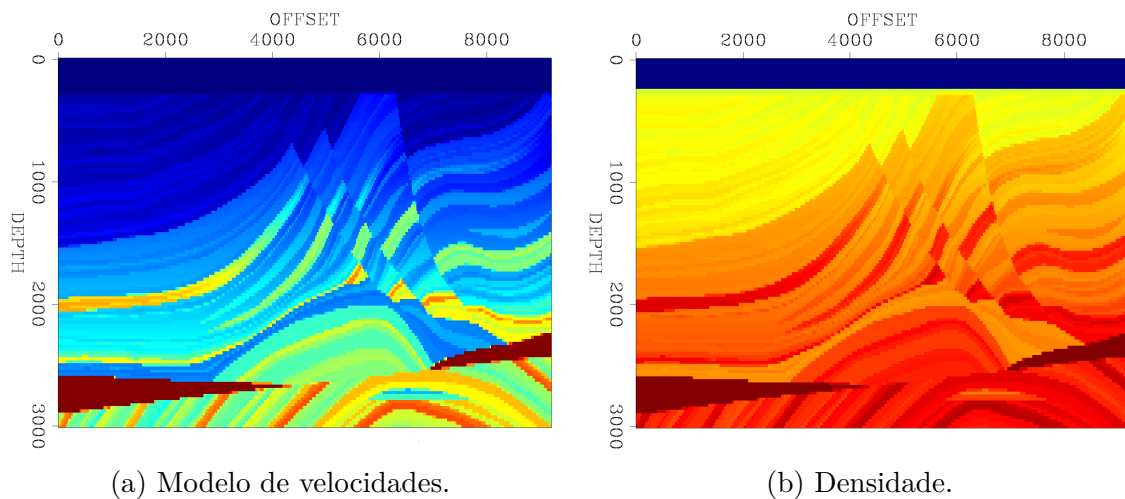


Figura 4.1: Modelo de velocidades do Marmousi (a) e a densidade correspondente.

Para o cálculo da função de custo, dado que estamos interessados em modelos de velocidade suaves e queremos evitar o salto de ciclos, filtramos o conteúdo de frequência do dado sísmico modelado e do adquirido em campo para baixas frequências de acordo com o seguinte *trade-off*. Se a frequência é muito baixa, a maior parte da energia representada no dado sísmico é removida. Isso afeta a efetividade da função de custo, impedindo a convergência da simulação. Por outro lado, se a frequência é muito alta, teremos saltos de ciclos. Para o dado Marmousi, a maior parte da energia concentra-se em torno da frequência de 3 Hz. Portanto, primeiramente aplicamos um filtro de baixa frequência a 4 Hz para o dado sísmico original e o modelado, então usamos esse dado filtrado para calcular a função de custo.

4.3 Exploração do Espaço de Variáveis

A malha original do dado Marmousi possui 121×369 pontos, mas para a amostragem do modelo de velocidades utilizamos uma malha grossa com 11×11 pontos, resultado da interpolação da malha original. Este processo de interpolação reduz o tamanho do espaço de variáveis centenas de vezes, mitigando a maldição da dimensionalidade.

Em nossos experimentos, configuramos o simulador Snesim para usar 3 possíveis valores de categorias (1500, 3000, and 4600) para cada célula das amostras do modelo de velocidade. A Figura 4.2a mostra a primeira amostra do modelo de velocidade da nossa simulação baseado em categorias. Ele é representado por três camadas horizontais de mesma espessura, cada uma com uma velocidade correspondendo a um valor de categoria. A estratégia de suavização utilizada é uma média móvel com 5 pontos de largura em ambas as direções (horizontal and vertical) aplicada quatro vezes. A Figura 4.2b mostra a velocidade resultante quando aplicamos a suavização. A imagem de treinamento é mostrada na Figura 4.2c, e consiste de uma imagem com 11×11 pontos, com os mesmos valores de categorias usados nas amostras. Ressaltamos que ambas a velocidade inicial e a imagem de treinamento são triviais. Isto mostra que as simulações convergem mesmo com um conhecimento prévio limitado sobre a geologia.

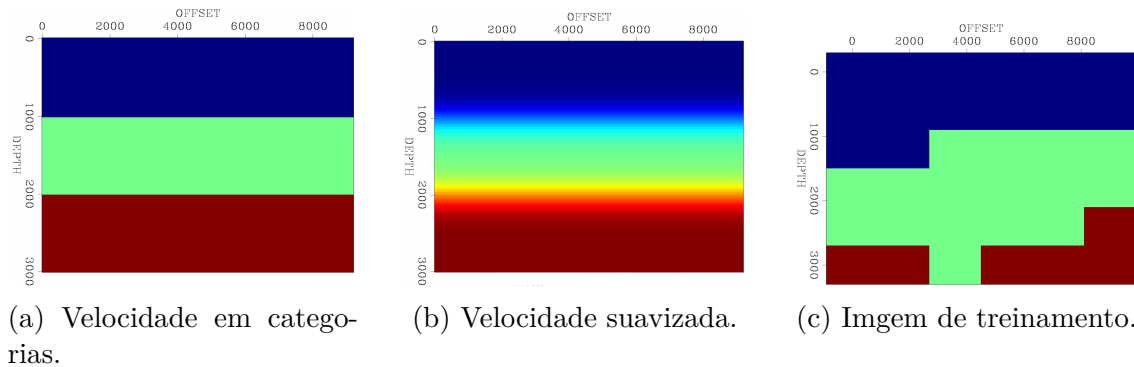


Figura 4.2: Amostra inicial do modelo de velocidade (por categorias e suavizada) e a imagem de treinamento.

Utilizamos em nossos experimentos $A_0 = 0.8A_T$ e $A_{N-1} = 0.1A_T$, e calculamos T_0 e T_{N-1} como proposto previamente. Usamos $\mu = 4$ (0.1% de 4×10^3 iterações, da Figura 3.9), resultando em $T_0 = 120$ e $T_{N-1} = 2 \times 10^{-6}$.

4.4 Análise de Convergência

Como mostrado na Figura 4.2a, nossas simulações iniciam a partir de um modelo de velocidade bem simples, que é aprimorado nas iterações do simulated annealing. Este processo iterativo faz com que a simulação convirja para um modelo de velo-

cidade a ser usando como ponto de partida da FWI. Tais simulações necessitam de um critério de parada, que pode ser baseado no número de iterações, nos valores da função de custo, no percentual de amostras aceitas por época, entre outros critérios. Neste trabalho adotamos como critério de parada um número fixo de iterações, determinado experimentalmente. Para tanto, fizemos experimentos utilizando simulações com 5×10^3 , 10^4 e 2×10^4 iterações, e uma época de 400 iterações. Utilizamos a amostra do modelo de velocidade com o menor valor de função de custo de cada simulação como modelo inicial da FWI. A convergência da FWI para o mínimo global só foi obtida a partir de modelos iniciais que foram resultado das simulações com 2×10^4 iterações. Entretanto, mesmo assim apenas duas simulações de um total de três geraram modelos de velocidade inicial que permitiram a convergência da FWI. Uma forma de mitigar esta incerteza seria aumentar o número de iterações da simulação, entretanto isto implicaria em um maior custo computacional. Ao invés disso, optamos por analisar o impacto do tamanho da época, de forma a definir um valor para a mesma, que garanta a convergência da FWI iniciando pelo modelo de velocidade resultado da simulação.

4.4.1 Tamanho da Época

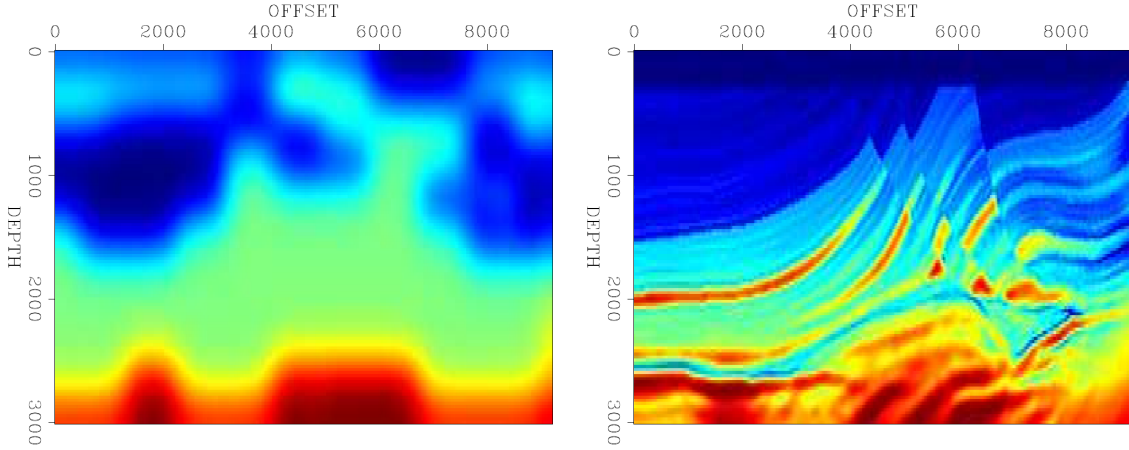
A teoria por trás do simulated annealing não relaciona o tamanho da época com a convergência. Entretanto, trabalhos anteriores [11, 33, 34] demonstram que, na prática, o tamanho da época impacta na convergência. Então, decidimos avaliar experimentalmente diversos tamanhos de época para encontrar um tamanho que maximiza a probabilidade da convergência. Executamos diversas simulações usando os parâmetros apresentados previamente para diferentes tamanhos de época, usando o mesmo tamanho de época para o stepping e o resfriamento. Para cada tamanho de época, repetimos a mesma simulação três vezes, com 2×10^4 iterações. A Tabela 4.1 mostra os resultados deste experimento, onde a última coluna mostra o número de simulações cujo modelo de velocidade resultante permitiu a convergência da FWI utilizando tal modelo como ponto de partida. O melhor resultado é para o tamanho de época igual a 600. Para este tamanho de época executamos mais duas simulações, que também geraram modelos de velocidade inicial que garantiram a convergência da FWI.

Os resultados da Tabela 4.1 mostram que os valores ótimos para α e τ para o dado Marmousi são 93% e 57.5%, respectivamente. Precisamos investigar mais para avaliar se outros dados sísmicos possuem valores ótimos similares. Entretanto, acreditamos que o uso dos valores de α e τ para calcular o tamanho da época é um método promissor, dado que ele permite calcular o tamanho da época diretamente através dos valores de A_0 , A_{N-1} , T_0 e T_{N-1} .

Tabela 4.1: Resumo dos experimentos para vários tamanhos de época.

Tamando da época E	α (%)	τ (%)	# convergiu
400	95.2	69.4	2
500	94.6	63.2	2
600	93.0	57.5	3
700	92.5	52.2	2
800	91.3	47.4	1

A Figura 4.3a mostra a amostra do modelo de velocidades que minimiza o valor da função de custo após 2×10^4 iterações. A Figura 4.3b mostra a convergência da FWI utilizando o modelo inicial da Figura 4.3a. Este resultado mostra que o nosso framework de simulated annealing é efetivo para encontrar um modelo de velocidade contendo os macro eventos geológicos, permitindo que a FWI convirja para um resultado final de alta qualidade.



(a) Resultado do simulated annealing.

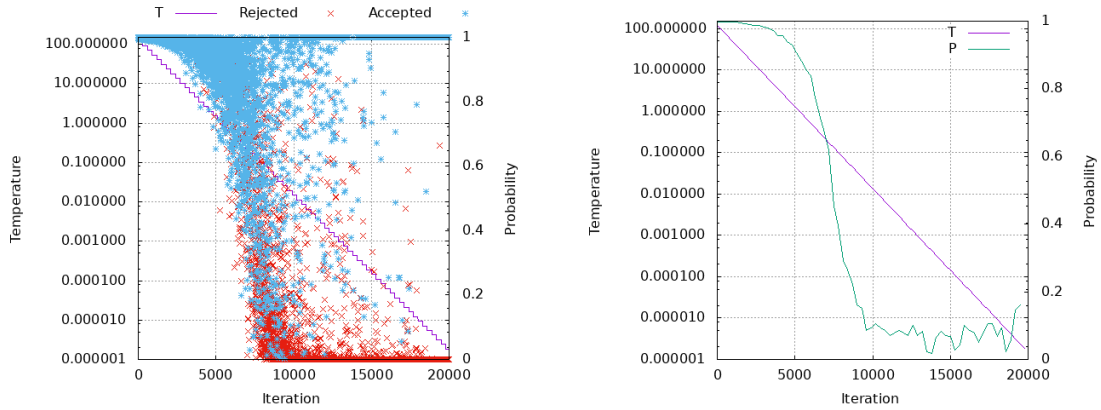
(b) Resultado da FWI.

Figura 4.3: (a) Resultado da simulação com 2×10^4 iterações usando parâmetros $A_0 = 0.8A_T$, $A_{N-1} = 0.1A_T$, $T_0 = 120$, $T_{N-1} = 2 \times 10^{-6}$, e tamanho de época 600. (b) Convergência da FWI usando o modelo de velocidades em (a) como modelo inicial.

4.5 Análise da Probabilidade de Aceitação

Conforme mencionado anteriormente, para as iterações onde $\Delta C_k \leq 0$, a simulação deterministicamente aceita a amostra proposta do modelo de velocidade. Assim, em nossa análise do comportamento de p_k , apenas consideramos iterações onde $\Delta C_k > 0$, dado que nestes casos temos uma decisão probabilística de aceitar ou rejeitar uma nova amostra. Para a simulação cujo resultado é o modelo de velocidade mostrado na Figura 4.3a, os valores de T_k para todas as iterações são mostrados, em escala logarítmica, no eixo vertical esquerdo da Figura 4.4a. Os valores correspondentes de

p_k são mostrados no eixo vertical direito. Os pontos azuis representam a aceitação da amostra proposta e os pontos vermelhos representam a rejeição. Como esperado, no início da simulação a maioria das amostras é aceita porque a temperatura está muito alta. Por outro lado, no final da simulação a maioria das amostras é rejeitada por que a temperatura está muito baixa. A Figura 4.4b mostra a média dos valores de p_k para cada época. Em ambas figuras, podemos observar que a simulação é bastante exploratória na primeira metade e consideravelmente menos exploratória na segunda metade, o que é fundamental para a convergência.



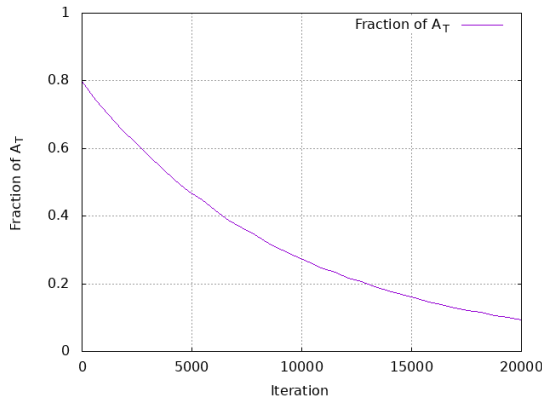
(a) Valores de p_k pra cada iteração. (b) Média de valores de p_k para cada época.

Figura 4.4: (a) Valores de p_k para todas as iterações da simulação. (b) Média dos valores de p_k para cada época, considerando apenas as iterações onde $\Delta C_k > 0$.

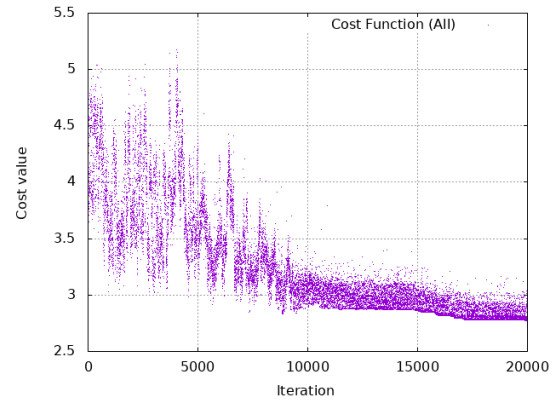
4.6 Análise da Função de Custo

Na Figura 4.5, apresentamos o efeito do stepping e do critério de aceitação do simulated annealing nos valores da função de custo e de ΔC_k . Para isso utilizamos os dados referentes às iterações da simulação apresentada na Figura 4.3. A Figura 4.5a mostra um decaimento exponencial de $0.8A_T$ para $0.1A_T$, e a Figura 4.5b mostra os valores da função de custo antes da aplicação probabilidade da exploração (equação 3.3). Neste caso, plotamos cada valor de C_k para todas as amostras, independente de terem sido aceitas ou rejeitadas. Isto mostra que, por causa do stepping, a faixa de valores na distribuição da função de custo é larga no início da simulação e estreita no final. A Figura 4.5c mostra os valores da função de custo após a aplicação da probabilidade da exploração. Neste caso, o valor da função de custo corresponde à amostra proposta quando ela é aceita ou corresponde à amostra anterior quando a amostra é rejeitada. Na segunda metade da simulação, a faixa de valores da função de custo é ainda mais estreita que na Figura 4.5b por duas razões. Primeiro, porque há mais amostras rejeitadas do que aceitas, logo na maioria das vezes temos $C_{k+1} = C_k$. Segundo, porque o modelo de velocidade já foi melhorado pelas iterações

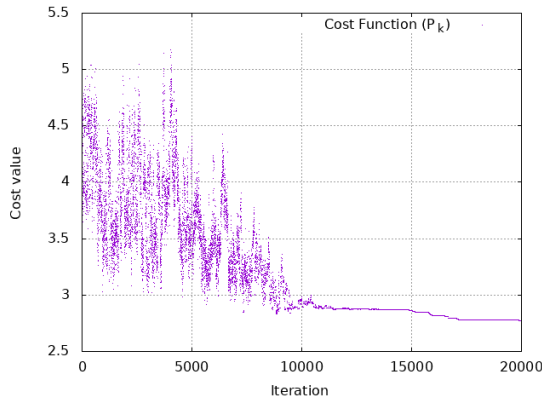
anteriores, ou seja, mesmo quando a simulação melhora o modelo atual a diferença no valor da função de custo é menor que nas iterações iniciais. A Figura 4.5d mostra os valores de ΔC_k , indicando: (1) uma faixa mais larga de valores na primeira metade da simulação; (2) a segunda metade da simulação é dominada por valores positivos. O processo de stepping explica (1). Por outro lado, na segunda metade da simulação, o modelo de velocidade já foi melhorado de tal forma que é mais difícil encontrar uma amostra com menor valor de função de custo. Isso explica a dominância de valores positivos de ΔC_k .



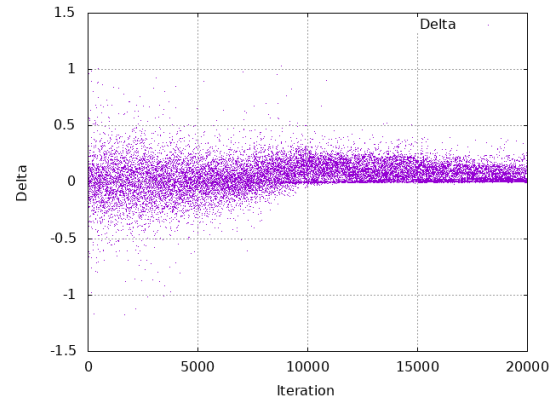
(a) Valores da subárea.



(b) Valores da função de custo antes da exploração.



(c) Valores da função de custo depois da exploração.



(d) Valores de ΔC_k para todas as iterações.

Figura 4.5: (a) Decaimento exponencial do tamanho da área conforme a época. (b) e (c) mostra os valores da função de custo antes e depois da exploração, respectivamente. (d) Valores de ΔC_k para todas as iterações.

Na nossa implementação, para cada iteração mantemos o registro do valor mínimo da função de custo até a iteração corrente, denotado por C_{\min} . Este valor é atualizado a cada amostra k tal que $C_k < C_{\min}$. Na Figura 4.6, mostramos no eixo vertical os valores normalizados de C_{\min} para cada iteração da simulação. Os pontos verdes representam o valor de cada atualização de C_{\min} , para a simulação mostrada na Figura 4.3. O comportamento da função de custo da Figura 4.5c e a dominância

dos valores positivos de ΔC_k mostrados na Figura 4.5d são responsáveis pelo efeito que vemos na Figura 4.6. Podemos observar que o valor de C_{\min} ao final da segunda metade (iteração 10^4) é muito próximo do valor de C_{\min} no final da simulação. A partir desta observação, testamos se a simulação poderia ser interrompida na iteração 10^4 . Para isto, executamos a FWI usando como modelo inicial de velocidade a amostra com o menor valor de função de custo até a iteração 10^4 . Como a FWI não convergiu neste teste, concluímos que a segunda metade da simulação ainda é crucial para a convergência da FWI, embora ela não decresce o valor da função de custo tanto quanto a primeira metade.

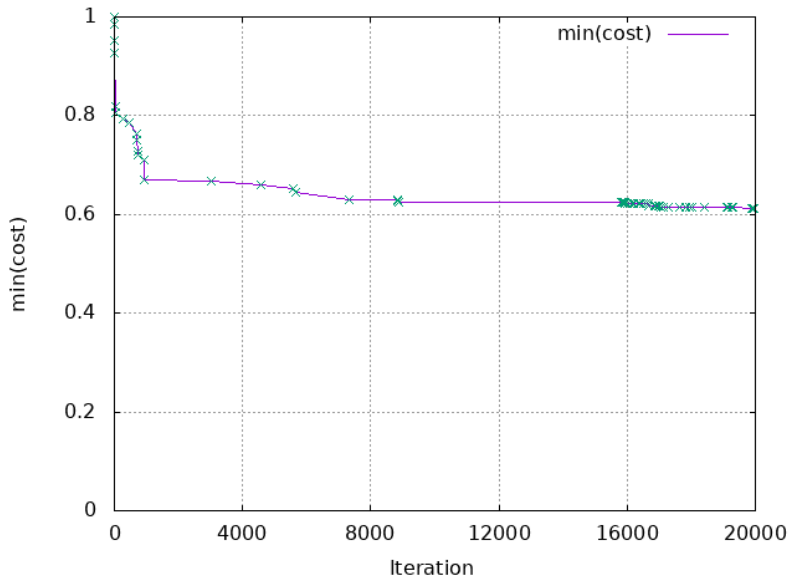


Figura 4.6: Valores mínimos da função de custo normalizados. Os pontos verdes representam as amostras onde C_{\min} é atualizado.

4.7 Análise de Desempenho

O tempo total de execução de uma simulação depende de seu número de iterações e do tempo de execução de cada iteração. Além disso, precisamos considerar que há etapas iniciais de ajuste fino para definir os parâmetros da simulação. Em alguns casos, estas etapas iniciais podem consumir mais tempo de computação que a própria simulação.

Devido à falta de um benchmark padrão largamente aceito, é difícil comparar o desempenho da nossa abordagem com outras. Elas são avaliadas usando dados sísmicos diferentes, ou mesmo quando utilizam o mesmo dado sísmico sua configuração é diferente (ex. granularidade da malha, número de tiros, profundidade, etc.).

O trabalho apresentado em [16] é o mais próximo do nosso para fins de comparação. O objetivo dos autores é encontrar um modelo inicial de velocidade suave

para a FWI usando dado Marmousi em seus experimentos. Ao invés de usar o simulated annealing, eles propuseram uma abordagem baseada em algoritmo genético. A abordagem deles emprega reflexões e refração para o cálculo da função de custo. Entretanto, os resultados mostram que a maior parte dos dados vem das refrações, que demandam dados sísmicos com longos afastamentos e processamento de modo. Também, para camadas profundas, a informação presente nas refrações é limitada quando comparada com as reflexões. Provavelmente este é o motivo pelo qual os autores cortaram o modelo de velocidade do Marmousi na profundidade de 2 km, enquanto nós utilizamos o modelo completo com 3 km de profundidade. Eles usaram uma malha grossa irregular com 16% mais pontos que no nosso caso, aumentando o tamanho do espaço de variáveis. A simulação deles requer 4×10^4 iterações para convergir, duas vezes mais que a nossa. Como não foi reportado o tempo de execução das simulações não podemos fazer nenhuma comparação utilizando esta métrica.

Em outro trabalho anterior dos mesmos autores [25], eles descrevem como calibraram os parâmetros do algoritmo genético. Foram executadas 3×10^6 avaliações de modelos de velocidade. Isto é uma ordem de grandeza maior que a dos nossos experimentos, mesmo considerando as simulações necessárias para calibrar o tamanho da época.

Além de reduzir a quantidade de iterações em comparação com trabalhos anteriores, também tratamos o tempo de execução de cada iteração paralelizando suas etapas de maior custo computacional: a modelagem e a RTM. Esta paralelização foi feita em dois níveis: intra-nó e inter-nó. No caso intra-nó, utilizamos versões da modelagem e da RTM que desenvolvemos anteriormente [26], utilizando OpenMP e diversas outras otimizações para processadores multicore. Com relação a paralelização inter-nó, utilizamos a versão paralela do nosso framework apresentada na seção 3.4. Cada uma de nossas simulações executou em cerca de 60 horas. Este tempo de execução é similar ao que tipicamente é observado na resolução de problemas em processamento sísmico, embora acreditemos que possa ser reduzido em uma ordem de grandeza através do uso de GPUs.

A Figura 4.7 mostra o tempo de execução normalizado para a execução com um nó de computação, de uma iteração do simulated annealing para diferentes quantidades de nós de computação. Neste experimento, variamos o número de nós de computação de 1 a 10. Podemos observar a diminuição do tempo de computação de cada iteração com o aumento no número de nós até um total de 10. Considerando a quantidade de tiros que é modelada e migrada a cada iteração, assim como o tamanho do modelo de velocidade do Marmousi, aumentar a quantidade de nós de computação além de um total de 10 não trará ganhos expressivos em termos de tempo de execução. Isto acontece porque, neste caso, a comunicação e sincronização se torna o gargalo, já que uma iteração possui três pontos de sincronismo: (1) envio

da amostra do modelo de velocidade; (2) recebimento das imagens migradas parciais de cada nó e envio do contraste de densidade; (3) recebimento dos valores da função de custo parcial de cada nó. Acreditamos que dados sísmicos maiores apresentarão maior escalabilidade.

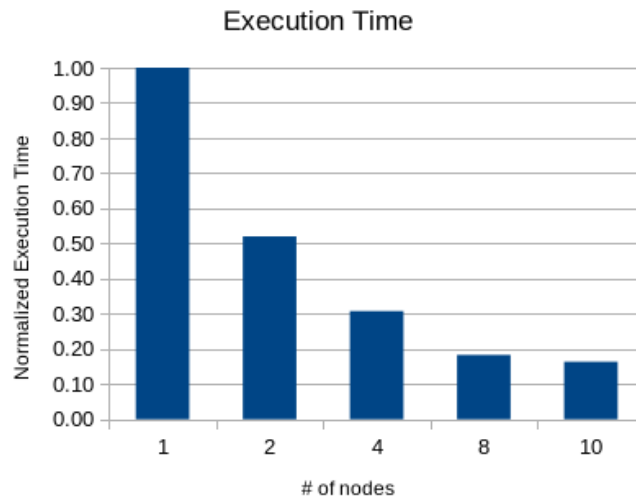


Figura 4.7: Tempo de execução normalizado para 1 a 10 nós de computação.



Figura 4.8: Percentual do tempo de execução total da simulação consumido por cada etapa das iterações.

Considerando que cada uma das etapas do nosso framework demanda um tempo de processamento da ordem de segundos, utilizamos uma medição de tempo cuja precisão fosse compatível com esta ordem de grandeza. Portanto, instrumentamos o código executado pelo nó computacional responsável pela distribuição do processamento dos tiros sísmicos, utilizando para isto a função *time* da linguagem Python, para medir o tempo de execução da cada uma das etapas do nosso framework. Assim, ao final de cada simulação, temos o tempo total gasto por cada etapa considerando

todas as iterações. Isto nos permite mostrar na Figura 4.8 o percentual do tempo total de execução de uma simulação, utilizando 10 nós de computação, gasto pelas principais etapas do nosso framework:

- Amostragem do modelo de velocidade
- Geração de imagens pela RTM
- Cálculo do contraste de densidade
- Modelagem sísmica
- Cálculo do custo global somando os custos parciais

A modelagem e a RTM juntas são responsáveis por quase 70% do tempo total de execução de cada iteração. Em ambos os casos, o tempo de execução é dominado pelo operador de diferenças finitas utilizado na resolução da equação de onda, que faz uso intenso da CPU. O cálculo do contraste de densidade é responsável por 30% do tempo de execução da simulação, porém neste caso o tempo é dominado pela comunicação. Esta comunicação consiste em uma comunicação síncrona, onde os nós m_1, \dots, m_{M-1} enviam suas imagens migradas parciais para o nó m_0 , que calcula o contraste de densidade e envia o mesmo de volta para os demais nós. O sincronismo desta comunicação consiste em um gargalo, limitando a escalabilidade. Na seção 6.2 sobre trabalhos futuros propomos alternativas de otimização do cálculo do contraste de densidade, como forma de evitar este custo de sincronização e comunicação.

Capítulo 5

Trabalhos Relacionados

5.1 Geração de Amostras

O Snesim utiliza uma árvore de busca para gerar as amostras, evitando o custoso processo de varrer a imagem de treinamento para a geração de cada ponto da amostra do modelo de velocidades. Entretanto com o surgimento das GPUs, a abordagem original baseada no percurso da imagem de treinamento proposta em [31] ganha novo destaque, dado que as GPUs são extremamente eficientes em tarefas que envolvam percorrer uma imagem, além da economia de memória evitando a criação de grandes árvores de busca. Esta abordagem é proposta em [35], onde é desenvolvida uma versão do Snesim baseada em GPUs apresentando bons resultados. Entretanto, os experimentos foram realizados em dados relativamente pequenos se comparados aos que apresentamos, estando bem longe do tamanho tipicamente visto em aplicações de geofísica aplicada a Óleo e Gás. Sendo assim, os autores não precisaram lidar com o problema de movimentação dos dados entre a memória da CPU e a da GPU, pois os dados cabiam totalmente na memória da GPU. Isto não seria possível em uma aplicação real. Esta versão GPU é baseada na implementação apresentada em [36], que apresenta um código mais moderno implementado em C++ com orientação a objetos, e suporta o Snesim tanto na versão com árvore de busca, como numa versão mais simplificada que utiliza uma lista encadeada no lugar de uma árvore.

Uma limitação dos geradores de amostra abordados é que eles não levam em consideração a posição da amostra na imagem. É comum haver padrões que são específicos a uma determinada área do modelo geológico, dificultando o estabelecimento de uma imagem de treinamento única e geral o bastante, para reproduzir todos os padrões necessários. Este problema é abordado em [37], onde se propõe a divisão da imagem de treinamento em imagens menores, chamadas classes, onde cada uma dessas imagens tem sua própria árvore de busca. Esta abordagem não só

ataca o problema de padrões específicos a cada região da imagem, como também diminui o tempo de simulação, já que as árvores de busca são menores. Entretanto, o maior desafio neste caso é detectar automaticamente os limites entre as classes, como forma de obter a divisão adequada da imagem total em imagens menores.

As amostras geradas pelo Snesim podem conter um conjunto pequeno de valores predefinidos, as categorias. Como as amostras de modelos de velocidade são representadas em ponto flutuante, utilizamos um método de interpolação baseado em média móvel para gerar tais modelos a partir de amostras representadas em categorias. Entretanto, há trabalhos na literatura que propõem a geração das amostras diretamente em ponto flutuante. O *Simpat*, proposto em [38], percorre a imagem de treinamento representada em ponto flutuante uma única vez e gera uma base de dados com seus padrões. Durante a simulação, de acordo com algum critério de similaridade, o padrão presente na base de dados mais próximo da vizinhança do ponto sendo simulado é atribuído à área sendo simulada. Esta abordagem permite não só o uso de valores contínuos, como também a simulação de diversos pontos simultaneamente. Entretanto, o processo de busca na base de dados pode ser extremamente lento. Estas limitações no *Simpat* são tratadas em [39] que propõe o *Filtersim*. A abordagem do *Filtersim* introduz o conceito de filtro, no qual de acordo com algum algoritmo de classificação, os padrões são agrupados. Sendo assim, não só a dimensionalidade é reduzida, como também a busca se torna mais eficiente, dado que a base de dados passa a conter apenas o elemento que representa um dado grupo. O *Filtersim* também possui sua implementação utilizando GPUs conforme mostrado em [40].

5.2 Métodos Monte Carlo

Em [21], é apresentada uma abordagem de inversão sísmica baseada no método Monte Carlo padrão, que é aplicada ao dado da base de Sorrento (sudeste do Colorado). A geologia é parametrizada por 100 camadas justapostas, onde cada camada é caracterizada pelas impedâncias da onda e sua densidade. A distribuição a priori é obtida através dos logs de poços de petróleo localizados na região. O método de Monte Carlo é utilizado para fazer a amostragem da distribuição a posteriori, com o objetivo de maximizar a função de verossimilhança. Esta abordagem equivale a minimizar a função de custo calculada a partir da diferença entre o dado modelado e o dado observado. O trabalho apresenta bons resultados para o dado sísmico utilizado como teste, porém impõe a limitação de considerar a geologia como camadas justapostas de mesma espessura. Além disso, necessita da presença de *logs* de poços de petróleo na região, o que nem sempre é possível principalmente na exploração de novas áreas.

5.2.1 Metropolis-Hastings

Em [19], os autores propõem um framework de inversão de um dado GPR (Ground Penetrating Radar) utilizando Metropolis-Hastings, onde a distribuição a priori é gerada com uma amostragem Gibbs utilizando Snesim, conforme abordado em [41]. O dado GPR utilizado nos testes é obtido entre poços perfurados, onde o objetivo é identificar canais curvilíneos entre os poços, e não um dado sísmico como o nosso. Além disso, o dado utilizado pelos autores é consideravelmente menor e são usadas apenas duas categorias.

Em [22], os autores utilizam o método Metropolis-Hastings para realizar o imageamento de dois dados sísmicos *Moho* e *W reflector*, onde são estimadas a espessura e o coeficiente de reflexão das camadas geológicas. Como distribuição a priori foram utilizadas a gaussiana para os coeficientes de reflexão e a exponencial para a espessura, e ambas foram definidas a partir da análise de dados de uma outra região com geologia semelhante. No nosso caso, é a imagem de treinamento que contém esta informação a priori, portanto o mesmo pode ser feito, bastando criar uma imagem de treinamento baseada nas estruturas da região de onde se quer trazer o conhecimento a priori.

5.2.2 Simulated Annealing

Em [15], é proposta uma técnica para estimar o modelo de velocidade utilizando o *Very Fast Simulated Annealing* (VFSA) [14]. Tal qual o nosso trabalho, o objetivo dos autores é gerar um modelo inicial de velocidade que reproduza os macro eventos da geologia, e usá-lo como modelo inicial da FWI. A estratégia utilizada para diminuir a dimensionalidade do problema foi representar como variáveis apenas as interfaces entre as camadas, sendo cada uma formada por um conjunto de pontos. Esta estratégia diminui muito a quantidade de variáveis amostradas, mas ao mesmo tempo restringe o modelo geológico que apenas pode ser representado por camadas horizontais. Além disso, é preciso fazer o delineamento destas camadas manualmente. O nosso trabalho é capaz de representar estruturas mais complexas utilizando os padrões presentes na imagem de treinamento, além de ser completamente automático.

Uma abordagem similar a que foi descrita acima é apresentada em [23], onde existe uma quantidade predeterminada de camadas, sendo que as variáveis estimadas são a velocidade e a espessura de cada camada, e as amostras são geradas por uma distribuição de Cauchy. Entretanto, esta técnica tem sua aplicabilidade restrita a uma geologia muito simples. Também em [24], é utilizado um modelo em camadas horizontais, onde a dimensionalidade do problema é dada pelas variáveis espessura, velocidade e densidade de cada camada horizontal.

O fluxo de execução do simulated annealing é um processo sequencial, sendo isso uma desvantagem pois os sistemas de alto desempenho são essencialmente paralelos. Em [42], é proposto um fluxo de execução especulativa que permite a execução paralela de diferentes iterações. Neste caso, antes da avaliação da função de custo, os dois possíveis caminhos (aceitar a nova amostra ou rejeitar) são iniciados especulativamente até um certo nível configurável pelo usuário. Com isso, temos uma árvore de execução, onde com o avanço da simulação os caminhos corretos são mantidos e os errados descartados. Assim, o passeio aleatório avança mais rápido, porém a um custo computacional maior, decorrente dos ramos da árvore que são descartados.

5.3 Algoritmos Genéticos

Os algoritmos genéticos (AG) [14, 43] constituem uma classe de métodos de otimização onde a evolução do sistema é uma analogia à forma como ocorre a evolução a biológica. Diferentemente do que acontece com o simulated annealing, o objetivo do AG é maximizar a função de *saúde* do sistema, que no nosso caso, representa quão bom é o casamento dos dados observados com os dados modelados.

Os parâmetros sendo estimados são representados como uma cadeia de bits de tamanho n , totalizando 2^n combinações. As cadeias de bits são alteradas pelas seguintes operações: seleção, cruzamento e mutação, sendo que a cada iteração é avaliada a saúde do sistema, que é o valor que a simulação busca maximizar. Um exemplo de função de saúde proposto em [14] para problemas geofísicos é,

$$F(m) = \frac{u_o \otimes u_s(m)}{(u_o \otimes u_s)^{1/2} u_s(m) \otimes u_s(m)^{1/2}}, \quad (5.1)$$

onde u_o é o dado observado, u_s é o dado simulado e \otimes é o operador de correlação.

Outro elemento chave é a forma como é determinada a probabilidade de um indivíduo (amostra da simulação) ser selecionado para a próxima geração (iteração do algoritmo). Uma forma clássica de calcular esta probabilidade é

$$p_i(m_i) = \frac{F(m_i)}{\sum_{j=1}^n F(m_j)}, \quad (5.2)$$

onde p_i é a probabilidade de i ser selecionado, $F(m_i)$ é a saúde do indivíduo m_i e $\sum_{j=1}^n F(m_j)$ é o somatório da saúde de todos os indivíduos da população.

Comparamos os nossos resultados com o trabalho baseado em algoritmos genéticos proposto em [16, 25] que também utilizam o dado sísmico Marmousi. Os autores utilizam reflexões e refrações no cálculo da função de custo, enquanto nós utilizamos apenas reflexões, o que nos permite utilizar dados com menor afastamento e evitar o processamento de modo. Além disso, o operador de diferenças finitas utilizado é

de ordem quatro no espaço e ordem dois no tempo, o que o torna menos acurado que o nosso, que é de ordem oito no espaço e quatro no tempo. Conforme demonstrado na seção de resultados, a nossa abordagem consegue convergir utilizando duas vezes menos iterações, além de ser capaz de calibrar os parâmetros fundamentais da simulação utilizando um número de iterações duas ordens de grandeza menor. Atribuímos o melhor resultado da nossa abordagem a capacidade de guiar e controlar a exploração do espaço de variáveis, e calcular os parâmetros da simulação evitando tentativa e erro.

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

Criamos um framework simulated annealing para gerar um modelo de velocidades com baixos números de onda para ser usado como modelo de velocidades inicial da FWI. A idéia geral é iniciar de um modelo bastante simples e atualizá-lo de acordo com os padrões presentes em uma imagem de treinamento. Dado o alto custo computacional do simulated annealing, devido ao número de iterações de cada simulação e ao ajuste fino de seus parâmetros, também propusemos diversas técnicas para reduzir este custo computacional.

Nossas contribuições foram eficazes para garantir a geração de um modelo de velocidade inicial que atende o critério de convergência da FWI. Além disso, elas permitiram que este objetivo fosse atingido com um número muito menor de iterações se comparado a trabalhos anteriores. Tais contribuições incluem o framework em si, uma série de mecanismos que controla a abrangência da exploração do espaço de variáveis e uma abordagem que guia a exploração deste espaço utilizando uma imagem de treinamento. Também criamos um mecanismo que permite calcular os valores de temperatura inicial e final do simulated annealing. Na sequência apresentamos nossas conclusões finais a respeito de cada uma dessas contribuições.

Um componente chave do custo computacional do simulated annealing é o tamanho do espaço de variáveis. Com relação a isso demonstramos que quando amostremos modelos de velocidade suaves, é possível reduzir o tamanho do espaço de variáveis por centenas de vezes usando uma malha grossa para geração de amostras. Uma vez gerada uma amostra, ela é interpolada para a malha fina através de um método bastante simples baseado em média móvel, para ser utilizada pelos métodos de equação de onda.

Além de reduzir o espaço de variáveis, criamos um mecanismo que determina a

sua abrangência baseado na amostragem de uma subárea de tamanho variável do mesmo. Chamamos este mecanismo de stepping. Criamos também uma métrica que permite calcular o tamanho da subárea do espaço de variáveis, que maximiza a abrangência da exploração. De acordo com esta métrica, o tamanho de subárea mais exploratório corresponde ao que gera os maiores valores de distância euclidiana, entre amostras de iterações adjacentes. Isto garante uma exploração bastante abrangente no início da simulação.

Outro componente chave é a efetividade com a qual o espaço de variáveis é explorado e explotado durante as simulações. Para isto, criamos um mecanismo que utiliza a informação prévia da geologia para guiar a exploração do espaço de variáveis. Este mecanismo se mostrou bem eficaz para este propósito. Esta informação em geral está disponível, ou pelo menos pode ser inferida a partir de áreas geológicas similares. Demonstramos que a representação apropriada desta informação em uma imagem de treinamento, permite ao mecanismo de amostragem focar na geração dos padrões de velocidades mais relevantes. Em nossos experimentos, conseguimos isso usando uma imagem de treinamento muito simples, o que reforça nossa conclusão que esta técnica pode facilmente ser aplicada a outras áreas geológicas, mesmo quando a informação prévia é muito limitada.

Criamos um mecanismo de cálculo da função de custo baseado apenas em reflexões, e avaliamos cada modelo de velocidade utilizando este mecanismo. Dado que modelos de velocidades suaves não produzem tais reflexões, propomos uma técnica para produzi-las. Usando imagens migradas pela RTM calculamos o contraste de densidade. Então, utilizamos este contraste como o parâmetro de densidade da ferramenta de modelagem produzindo as reflexões necessárias. Demonstramos que esta técnica pode gerar uma função de custo acurada que não requer nem dados com longos afastamentos nem processamento de modo. Também, esta técnica pode representar melhor na função de custo as camadas mais profundas do subsolo, quando comparado com trabalhos anteriores. A distribuição dos valores de função de custo em nossas simulações mostra que nosso método para o cálculo da função de custo, usando apenas reflexões, é efetivo para representar o quanto uma amostra difere da anterior. Isto é essencial para calcular os valores de ΔC_k usados na exploração.

Avaliamos o impacto dos principais parâmetros do simulated annealing e sua correlação: step, temperatura e tamanho da época. Demonstramos que a escolha apropriada dos mesmos, assim como o critério de decaimento da step e da temperatura, são fundamentais para garantir uma exploração abrangente no início da simulação e restrita no final, sendo isso fundamental para a convergência. Adicionalmente, analisamos em detalhe a correlação entre estes parâmetros. Utilizando esta análise criamos uma metodologia para calcular os valores destes parâmetros evitando tentativa e erro. Isto permitiu reduzir em duas ordens de grandeza, a quantidade de

iterações de ajuste fino de parâmetros comparado a trabalhos anteriores. Até onde sabemos fomos os primeiros a analisar detalhadamente esta correlação de parâmetros. A metodologia proposta é genérica e pode ser diretamente aplicada a outros dados sísmicos.

Nossos experimentos com o dado sísmico Marmousi mostram que podemos eficientemente definir os parâmetros do simulated annealing e limitar o número de iterações para convergir. Apenas 8×10^3 iterações foram necessárias para definir os parâmetros da simulação. Com estes parâmetros definidos, após uma exploração inicial abrangente nosso método foca nas áreas mais relevantes do espaço de variáveis. A simulação completa converge usando 2×10^4 iterações. Há duas razões para este número reduzido de iterações: (1) a imagem de treinamento evita a exploração de padrões irrelevantes; (2) a combinação do stepping e do resfriamento limita a aceitação de amostras com maior valor de função de custo. Quando comparado com trabalhos anteriores usando o dado sísmico Marmousi, reduzimos por duas ordens de grandeza o número de iterações para definir os parâmetros do simulated annealing e em 50% a quantidade de iterações da simulação.

Usando 10 nós de computação com duas CPUs de 12 núcleos cada, o tempo de execução de cada simulação é cerca de 60 horas, o que é bastante razoável considerando o poder de computação utilizado. Nosso framework pode ser estendido de forma a tirar proveito de um ambiente massivamente paralelo de uma GPU, na mesma direção proposta por outros autores. O poder de computação de uma GPU nos permitiria prover modelos de velocidade iniciais para dados sísmicos maiores e mais complexos. Deve ser observado que o nosso framework pode utilizar qualquer ferramenta de modelagem e RTM. Portanto, nenhum código proprietário é necessário para reproduzir nossos resultados.

6.2 Trabalhos Futuros

A realização de testes com outros dados sísmicos com diferentes tamanhos, configurações de sedimentos, camadas de sal, pode aprimorar as técnicas propostas para definir os parâmetros das simulações. Atualmente estamos trabalhando em outro dado sísmico, composto por sedimentos dispostos em camadas sobrepostas e curvilíneas. Uma vez que tivermos aplicado com êxito nossas técnicas a este dado sísmico, vislumbramos como próximo passo a aplicação de tais técnicas em dados 3D. O processamento de dados 3D é atualmente uma realidade na indústria de Óleo e Gás, sendo portanto um caminho natural deste trabalho. Neste cenário, os principais desafios são: (1) um espaço de variáveis maior; (2) tempo de processamento da RTM e modelagem maior; (3) maior quantidade de dados gerada, tanto pela amostragem como pela RTM e modelagem. Enfrentar estes desafios torna ainda mais importante

o uso de formas de controlar a exploração do espaço de variáveis. Além disso, são necessárias técnicas que garantam a escalabilidade, dado que uma quantidade muito maior de nós de computação é necessária para processar dados 3D.

Há na literatura diversas propostas de decaimento da temperatura do simulated annealing, tais como hiperbólica, linear e logarítmica. Embora estas outras formas de decaimento já se encontrem implementadas no nosso algoritmo, por uma questão de limitação de disponibilidade de recursos computacionais as mesmas ainda não foram avaliadas, porém, pretendemos avaliar o seu impacto futuramente.

Nossos resultados com o simulated annealing mostram que a chance de uma amostra ser aceita ou rejeitada é bastante previsível na maior parte da simulação. No início, mais de 90% das amostras é aceita enquanto no final mais de 90% delas é rejeitada. Acreditamos que esta previsibilidade possibilita a execução especulativa das iterações do algoritmo, onde várias delas são executadas em paralelo assumindo especulativamente um resultado de aceitação ou rejeição das amostras. Esta abordagem aumenta consideravelmente o paralelismo da simulação, podendo diminuir bastante seu tempo de execução mediante a disponibilidade de mais nós de computação.

A versão do simulador Snesim usada neste trabalho utiliza a abordagem que armazena os padrões da imagem de treinamento em uma árvore de busca, sendo tal abordagem adequada para CPUs tradicionais. Embora, conforme mostrado na Figura 4.8, a amostragem dos modelos de velocidade seja responsável por menos de 2% do tempo de execução, acreditamos que este percentual pode aumentar para simulações com amostras com mais valores de categorias, padrões mais complexos na imagem de treinamento ou dados 3D. Por isso, propomos como trabalho futuro uma abordagem baseada em GPUs para o que foi proposto originalmente em [31], onde a imagem de treinamento é varrida para cada ponto da amostra sendo gerada.

O cálculo do contraste de densidade limita a escalabilidade do nosso algoritmo, isso porque o mesmo é feito de forma centralizada no nó m_0 , e é feito broadcast do resultado para os demais nós. Para mitigar este problema uma alternativa é utilizar um contraste de densidade parcial, ou seja, calcular o mesmo localmente em cada nó de computação utilizando sua respectiva imagem migrada parcial. Assim, cada nó calcula o contraste de densidade correspondente aos tiros associados a ele, e usaria este contraste de densidade também na modelagem de seu subconjunto de tiros, e gerando assim seu respectivo valor de função de custo parcial. Esta proposta foi implementada e encontra-se em fase de avaliação.

Referências Bibliográficas

- [1] YILMAZ, O. *Seismic Data Analysis*, v. 10, *Investigations in Geophysics*. Society Of Exploration Geophysicists, jan. 2001. ISBN: 1560800941. doi: 10.1190/1.9781560801580. Disponível em: <<http://dx.doi.org/10.1190/1.9781560801580>>.
- [2] BAYSAL, E., KOSLOFF, D., SHERWOOD, W. “Reverse-time migration”, *Geophysics*, v. 48, pp. 1514–1524, 1983.
- [3] TARANTOLA, A. “Inversion of seismic reflection data in the acoustic approximation”, *Geophysics*, v. 49, pp. 1259–1266, 1984.
- [4] VIRIEUX, J., OPERTO, S. “An overview of full-waveform inversion in exploration geophysics”, *Geophysics*, v. 74, n. 6, 2009.
- [5] BISWAS, R., SEN, M. K. “2D Full-Waveform Inversion and Uncertainty Estimation using the Reversible Jump Hamiltonian Monte Carlo”, *SEG Technical Program Expanded Abstracts*, 2017.
- [6] ZHAO, Z., SEN, M. K. “A gradient-based Markov chain Monte Carlo method for full-waveform inversion and uncertainty analysis”, *Geophysics*, v. 86, pp. R15–R30, 2021.
- [7] BISWAS, R., SEN, M. K. “Transdimensional 2D Full-Waveform Inversion and Uncertainty Estimation”. 2022. Disponível em: <<https://arxiv.org/abs/2201.09334>>.
- [8] MARTÍNEZ-SANSIGRE, A., RATCLIFFE, A. “A probabilistic QC for cycle-skipping in full waveform inversion”, *SEG Technical Program Expanded Abstracts*, 2014.
- [9] GUITTON, A., AYENI, G., DÍAZ, E. “Constrained full-waveform inversion by model reparameterization”, *Geophysics*, v. 77, pp. R117–R127, 2012.
- [10] WANG, H., SINGH, S. C., AUDEBERT, F., et al. “Inversion of seismic refraction and reflection data for building long-wavelength velocity models”, *Geophysics*, v. 80, pp. R81–R93, 2015.

- [11] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P. “Optimization by simulated annealing”, *SCIENCE*, v. 220, n. 4598, pp. 671–680, 1983.
- [12] MOSEGAARD, K., TARANTOLA, A. “Monte Carlo sampling of solutions to inverse problems”, *Journal of Geophysical Research*, v. 100, pp. 431–447, 1995. doi: 10.1029/94JB03097.
- [13] MOSEGAARD, K., SAMBRIDGE, M. “Monte Carlo analysis of inverse problems”, *Inverse Problems*, v. 18, pp. R29–R54, 2002. doi: 10.1088/0266-5611/18/3/201.
- [14] SEN, M. K., STOFFA, P. L. *Global Optimization Methods in Geophysical Inversion*. 2013.
- [15] DATTA, D., SEN, M. K. “Estimating a starting model for full-waveform inversion using a global optimization method”, *Geophysics*, v. 81, pp. R211–R223, 2016.
- [16] MAZZOTTI, A., BIENATI, N., STUCCHI, E., et al. “Two-grid genetic algorithm full-waveform inversion”, *The Leading Edge*, v. 35, pp. 1010–1096, 2016.
- [17] VERSTEEG, R. “Velocity model determination on a synthetic complex data set”, *The Leading Edge*, v. 13, n. 9, pp. 927–936, 1994.
- [18] DATTA, D., SEN, M. K., LIU, F., et al. “Full-waveform inversion of salt models using shape optimization and simulated annealing”, *Geophysics*, v. 84, n. 5, pp. R793–R804, 2019.
- [19] KNUD SKOU CORDUA, T. M. H., MOSEGAARD, K. “Monte Carlo full-waveform inversion of crosshole GPR data using multiple-point geostatistical a priori information”, *GEOPHYSICS*, , n. 2, pp. H19–H31, MARCH-APRIL 2012. doi: 10.1190/GEO2011-0170.1.
- [20] HANSEN, T., CORDUA, K., MOSEGAARD, K. “Inverse problems with non-trivial priors: Efficient solution through sequential Gibbs sampling”, *Computational Geosciences*, v. 16, 06 2012. doi: 10.1007/s10596-011-9271-1.
- [21] GOUBEIA, W., SCALES, J. “Bayesian seismic waveform inversion: Parameter estimation and uncertainty analysis”, v. 103, 01 1998.
- [22] MOSEGAARD, K., SINGH, S., SNYDER, D., et al. “Monte Carlo analysis of seismic reflections from Moho and the W reflector”, v. 102, pp. 2969–2982, 01 1997.

- [23] TRAN, K., R. HILTUNEN, D. “Two-Dimensional Inversion of Full Waveforms Using Simulated Annealing”, v. 138, pp. 1075–1090, 09 2012.
- [24] AGARWAL, R., SAIN, K. “Application of VFSA for AVO Inversion of seismic reflection data”, 06 2016.
- [25] SAJEVA, A., ALEARDI, M., STUCCHI, E., et al. “Estimation of acoustic macro models using a genetic full-waveform inversion: Applications to the Marmousi model”, v. 81, pp. R173–R184, 07 2016.
- [26] MENDES, R., KAELIN, B. “Multicore Optimizations for Elastic Reverse Time Migration”, *76th EAGE Conference and Exhibition*, 2014. doi: 10.3997/2214-4609.20141136.
- [27] ORTIGOSA, F., ARAYA-POLO, M., RUBIO, F., et al. “Evaluation of 3D RTM on HPC Platforms”. In: *SEG Las Vegas Annual Meeting*, 2008.
- [28] BROWN, S. “Performance comparison of finite-difference modeling on Cell, FPGA and multi-core computers”. In: *SEG San Antonio Annual Meeting*, 2007.
- [29] JONES, I. *An Introduction to Velocity Model Building*. EAGE, 2010. ISBN: 9789073781849. Disponível em: <<https://books.google.com.br/books?id=Dac4bwAACAAJ>>.
- [30] HASTINGS, W. K. “Monte Carlo sampling methods using Markov chains and their applications”, *Biometrika*, v. 57, n. 1, pp. 97–109, 1970. doi: 10.1093/biomet/57.1.97. Disponível em: <<http://dx.doi.org/10.1093/biomet/57.1.97>>.
- [31] FELIPE B. GUARDIANOR, SRIVASTAVA, M. *Multivariate Geostatistics: Beyond Bivariate Moments*. Soares, A (Ed.), Geostatistics-Troia, 1. Kluwer Academic Publications, Dordrecht, 1993.
- [32] STREBELLE, S. *Sequential simulation drawing structures from training images*. Tese de Doutorado, Stanford University, 2000.
- [33] GEMAN, S., GEMAN, D. “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. PAMI-6, pp. 721–741, 1984.
- [34] JOHNSON, D. S., ARAGON, C. R., MCGEOCH, L. A., et al. “Optimization by simulated annealing: An experimental evaluation; Part I, graph partitioning”, *Operations research*, v. 37, pp. 865–892, 1989.

- [35] TAO HUANG, DE-TANG LU, X. L. L. W. “GPU-based SNESIM implementation for multiple-point statistical simulation”, *Computers & Geosciences*, , n. 54, pp. 75–87, 2013. ISSN: 0098-3004. doi: <https://doi.org/10.1016/j.cageo.2012.11.022>.
- [36] VU, T. M. H. L. T., BACH, T. “MPSLIB: A C++ class for sequential simulation of multiple-point statistical models”, *SoftwareX*, pp. 127–133, 2016. doi: <https://doi.org/10.1016/j.softx.2016.07.001>.
- [37] BOUCHER, A. “Considering complex training images with search tree partitioning”, *Computers & Geosciences*, pp. 1151–1158, 2009. doi: <https://doi.org/10.1016/j.cageo.2008.03.011>.
- [38] ARPAT, G., CAERS, J. “Conditional Simulation with Paterns”, *Mathematical Geology*, , n. 39, pp. 177–203, 2007. ISSN: 0882-8121. doi: <https://doi.org/10.1007/s11004-006-9075-3>.
- [39] WU, J. B. A., T, Z. “A SGeMS code for pattern simulation of continuous and categorical variables: FILTERSIM”, *Computers & Geosciences*, , n. 34, pp. 1863–1876, December 2008. ISSN: 0098-3004. doi: <https://doi.org/10.1016/j.cageo.2007.08.008>.
- [40] PEJMAN TAHMASEBIA, MUHAMMAD SAHIMIB, G. M., HEZARKHANIA, A. “Accelerating geostatistical simulations using graphics processing units (GPU)”, *Computers & Geosciences*, , n. 46, pp. 51–59, September 2012. ISSN: 0098-3004. doi: <https://doi.org/10.1016/j.cageo.2012.03.028>.
- [41] HANSEN, T., CORDUA, K., MOSEGAARD, K. “Inverse problems with non-trivial priors: Efficient solution through sequential Gibbs sampling”, v. 16, 06 2012.
- [42] PEREDO, O., ORTIZ, J. M. “Parallel implementation of simulated annealing to reproduce multiple-point statistics”, *Computers Geosciences*, v. 37, n. 8, pp. 1110 – 1121, 2011. ISSN: 0098-3004. doi: <https://doi.org/10.1016/j.cageo.2010.10.015>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0098300410003407>>.
- [43] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York, Addison-Wesley, 1989.