

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE GEOCIÊNCIAS
DEPARTAMENTO DE GEOLOGIA E GEOFÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM DINÂMICA DOS OCEANOS E DA
TERRA

HANS CASTRO EINSIEDLER

**COMPARAÇÃO DE MÉTODOS DE OTIMIZAÇÃO PARA
SOLUÇÃO DO PROBLEMA DE INVERSÃO DA FORMA DE
ONDA COMPLETA**

NITERÓI

2019

HANS CASTRO EINSIEDLER

COMPARAÇÃO DE MÉTODOS DE OTIMIZAÇÃO PARA
SOLUÇÃO DO PROBLEMA DE INVERSÃO DA FORMA DE
ONDA COMPLETA

Dissertação apresentada à Universidade Federal Fluminense como requisito parcial do Programa de Pós-Graduação em Dinâmica dos Oceanos e da Terra para a obtenção do título de Mestre na área de Geologia e Geofísica.

Orientador: Prof. Dr. Wagner Moreira Lupinacci

NITERÓI - RJ

2019

HANS CASTRO EINSIEDLER

Dissertação apresentada à Universidade Federal Fluminense como requisito parcial do Programa de Pós-Graduação em Dinâmica dos Oceanos e da Terra para a obtenção do título de Mestre na área de Geologia e Geofísica.

Aprovada em 31 de Outubro de 2019.

Comissão Examinadora:

_____ Pro

f. xxxx

_____ Pro

f. Dr. Marco Antonio Cetale Santos (GISIS/GGO/UFF)

_____ Pro

f. Dr. Rodrigo Bijani (GIECAR/GGO/UFF)

_____ Pro

f. Dr. Wagner Moreira Lupinacci - Orientador (GIECAR/GGO/UFF)

AGRADECIMENTOS

A presente dissertação de mestrado não poderia chegar a bom porto sem o precioso apoio de várias pessoas. Em primeiro lugar, não posso deixar de agradecer ao meu orientador, Professor Doutor Wagner Moreira Lupinacci, por toda a paciência, empenho e sentido prático com que sempre me orientou neste trabalho. Muito obrigado por me ter corrigido quando necessário sem nunca me desmotivar. Agradeço aos funcionários do instituto e da sala de computação, que foram sempre solícitos. Por último, quero agradecer à minha família e amigos pelo apoio incondicional que me deram, especialmente a minha esposa que sempre me motivou para conclusão desta dissertação.

EPÍGRAFE

“A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo”. (Albert Einstein).

Resumo

A inversão da forma de onda completa (FWI - do inglês *full-waveform inversion*) é atualmente uma das principais ferramentas para determinar modelos de velocidades de subsuperfície com alta resolução. A estimativa do melhor modelo de velocidade utilizando o método FWI requer um alto custo computacional e não garante solução única, pois o problema envolvendo o método FWI é classificado como um problema inverso não linear, sendo resolvido de forma iterativa, calculando a função objetivo, correspondente a diferença entre os dados observados e preditos por um modelo matemático. Nesta dissertação é comparado o custo computacional e convergência do método FWI utilizando dois algoritmos de otimização: o gradiente projetado espectral não-monótono (NSPG – do inglês *Non-monotone Spectral Projected Gradient*) e o *Limited-Broyden–Fletcher–Goldfarb–Shanno* (L-BFGS) que é um algoritmo da família do método Quasi-Newton. O método NSPG de otimização possui baixo custo computacional e exige um baixo armazenamento de memória, limitando o domínio de busca para solução do método FWI, com um decréscimo não monótono da função objetivo. Já o método (L-BFGS) possui decréscimo monótono da função objetivo utilizando pré-condicionamento do passo, muito utilizado na otimização do método FWI. Para comparar as otimizações L-BFGS e NSPG do método FWI foram realizados experimentos utilizando o software The Julia Devito Inversion framework (JUDI). Os resultados mostraram que apesar do método NSPG possuir um custo computacional menos elevado, o método L-BFGS se mostrou mais preciso com valor da função de erro menor quando comparado com o mesmo número de iterações do método NSPG.

Palavras-chave: inversão da forma de onda; otimização e FWI.

Abstract

Full-waveform inversion (FWI) is currently one of the main tools for determining high-resolution subsurface velocity models. The estimation of the best velocity model using the FWI method requires a high computational cost and does not guarantee a single solution, since the problem involving the FWI method is classified as a highly non-linear inverse problem, being iteratively solved by calculating the objective function, corresponding to the difference between the data observed and calculated by a mathematical model. In this dissertation the computational cost and convergence of the FWI method will be compared using the optimization algorithms: The non-monotonous spectral projected gradient (NSPG) and Limited-Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm of the Quasi-Newton method family. The NSPG optimization method has a low computational cost and requires a low memory storage, limiting the search domain to FWI method solution, with a non-monotonic decrease of the objective function. The method (L-BFGS) has monotonous decay of the objective function using step preconditioning, which is widely used in the optimization of the FWI method. To compare the L-BFGS and NSPG optimizations of the FWI method, experiments were performed using the Julia Devito Inversion framework (JUDI) software. The results showed that although the NSPG method had a lower computational cost, the L-BFGS method proved to be more accurate with a lower error function value when compared to the same number of iterations.

Key words: full wave form, optimization and FWI

Sumário

1	Introdução	12
2	Revisão teórica	14
2.1	Modelagem sísmica	14
2.1.1	Equação da onda acústica	16
2.1.2	Diferenças finitas	20
2.1.3	Estabilidade numérica	24
2.2	Migração reversa no tempo.....	25
2.3	Inversão completa da forma da onda (FWI)	29
2.3.1	Método L-BFGS.....	33
2.3.2	Algoritmo L-BFGS.....	33
2.3.3	Método Non-motone Spectral Projected Gradient (NSPG).....	35
2.3.4	Algoritmo NSPG	36
3	Metodologia.....	37
4	Resultados	41
5	Conclusão	52
6	Apêndice (Scripts).....	53
7	Bibliografia	60

Lista de Figuras

Figura 1: Definição de modelo com contraste de velocidade acústica podendo ser utilizado na modelagem sísmica com a fonte(vermelho) e os receptores(verde) na superfície.....	15
Figura 2: Wavelet de Ricker com frequência de pico de 10Hz.	16
Figura 3: Estêncil 2-2 diferença finita.	23
Figura 4: Cone de iluminação progressivo, formado a partir da progressão no tempo do stencil.	24
Figura 5: Ponto de espalhamento em subsuperfície x_scatt único local onde o produto escalar das ondas propagadas descendentes e retropropagadas ascendentes é diferente de zero.	29
Figura 6: Esboço da função objetivo não-linear $E(m)$ para um caso de 2 parâmetros (m_1 e m_2). $E(m+\Delta m)$ é um parabolóide em função de Δm . Em vermelho, derivadas nulas em cada parâmetro dão o ponto de mínimo.....	31
Figura 7: Campo de velocidade overthrust da SEG/EAGE.	38
Figura 8: Tiro obtido a partir da modelagem utilizando o campo de velocidade overthrust da SEG/EAGE.....	38
Figura 9: Modelo inicial utilizado nos experimentos com FWI.....	39
Figura 10: Modelo inicial utilizado nos experimentos com velocidade constante de 3500 km/s.....	40
Figura 11: Tiro com ruído adicionado ao dado. Razão sinal/ruído de 70%.....	40
Figura 12: Campo de velocidade inicial suavizado para o teste com velocidade inicial suavizada e dados observados sem ruído.	41
Figura 13: Campo de velocidade verdadeiro.....	41
Figura 14: Campo de velocidade final obtido com o método SPG para o teste com velocidade inicial suavizada e dados observados sem ruído.	42
Figura 15: Campo de velocidade final obtido com o método SGD para o teste com velocidade inicial suavizada e dados observados sem ruído.	42
Figura 16: Campo de velocidade final obtido com o método L-BFGS para o teste com velocidade inicial suavizada e dados observados sem ruído.	42
Figura 17: Campo de velocidade inicial com velocidade constante para o teste com velocidade inicial constante e dados observados sem ruído.....	43

Figura 18: Campo de velocidade final obtido com o método SGD para o teste com velocidade inicial constante e dados observados sem ruído.....	43
Figura 19: Campo de velocidade final obtido com o método NSPG para o teste com velocidade inicial constante e dados observados sem ruído.....	43
Figura 20: Campo de velocidade final obtido com o método L-BFGS para o teste com velocidade inicial constante e dados observados sem ruído.....	44
Figura 21: Campo de velocidade inicial suavizado para o teste com velocidade inicial suavizada e dados observados com ruído.....	44
Figura 22: Campo de velocidade final obtido com o método SGD para o teste com velocidade inicial suavizada e dados observados com ruído.....	44
Figura 23: Campo de velocidade final obtido com o método NSPG para o teste com velocidade inicial suavizada e dados observados com ruído.....	45
Figura 24: Campo de velocidade final obtido com o método L-BFGS para o teste com velocidade inicial suavizada e dados observados com ruído.....	45
Figura 25: Dado modelado utilizando as velocidades obtidas do experimento 1. a) SGD; b) NSPG e c) L-BFGS.....	47
Figura 26: Diferença entre os dados modelados da figura 25 e o dado modelado com a velocidade real para os métodos a) SGD, b) NSPG e c) L-BFGS.....	47
Figura 27: Dado modelado utilizando as velocidades obtidas do experimento 2. a) SGD; b) NSPG e c) L-BFGS.....	48
Figura 28: Diferença entre os dados modelados da figura 27 e o dado modelado com a velocidade real para os métodos a) SGD, b) NSPG e c) L-BFGS.....	48
Figura 29: Dado modelado utilizando as velocidades obtidas do experimento 3. a) SGD; b) NSPG e c) L-BFGS.....	49
Figura 30: Diferença entre os dados modelados da figura 29 e o dado modelado com a velocidade real para os métodos a) SGD, b) NSPG e c) L-BFGS.....	49
Figura 31: Função objetivo para os métodos SPG, SGD e L-BFGS para o caso de velocidade inicial suavizada e dados sem ruído.....	50
Figura 32: Função objetivo para os métodos SPG, SGD e L-BFGS para o caso de velocidade inicial constante e dados sem ruído.....	51
Figura 33: Função objetivo para os métodos SPG, SGD e L-BFGS para o caso de velocidade inicial suavizada e dados com ruído.....	51

1 Introdução

Devido à alta não-linearidade e não unicidade da solução, o principal desafio do método FWI (TARANTOLA, 1984a) é evitar convergência a mínimos locais e ao mesmo tempo manter o custo computacional a um nível aceitável (DOS SANTOS; PESTANA, 2012).

Uma forma de evitar mínimos locais é utilizar uma abordagem multiescala do problema (BUNKS et al., 1995) ou restringir o domínio da solução procurada. Já o problema do custo computacional pode ser amenizado usando novos métodos de procura de minimização do funcional de erro ou utilizar paralelização de algoritmos com placas gráficas (WANG et al., 2011).

O método de resolução Nonmonotone Spectral Projected Gradient (ZEEV; SAVASTA; CORES, 2006), utilizado nesta dissertação, visa minimizar ambos os problemas citados, restringindo o domínio do parâmetro elástico procurado, evitando o cálculo da matriz Hessiana e pre-condicionando o comprimento do passo de procura da solução do funcional de erro para reduzir o custo computacional.

Já o método L-BFGS, também avaliado nesta dissertação, é um algoritmo de otimização da família do método Quasi-Newton similar ao algoritmo (BFGS), só que utilizando uma quantidade de memória RAM limitada. Como no algoritmo original BFGS, o L-BFGS usa uma estimativa da matriz Hessiana para direcionar sua procura pela solução, a diferença está no armazenamento, enquanto o BFGS armazena uma aproximação densa da matriz Hessiana, o L-BFGS armazena apenas alguns vetores que representam uma aproximação implícita. Devido ao seu requerimento linear de memória, o método L-BFGS é particularmente bem ajustável a otimização de problemas com número grande de variáveis.

Diante destas considerações, esta dissertação visa responder a seguinte pergunta: A otimização NSPG do método FWI que utiliza decrescimento não monótono da função objetivo é mais eficaz que a otimização L-BFGS amplamente utilizada para solução do método FWI?

Com o objetivo de responder esta pergunta será analisado o tempo de processamento, convergência e valor do funcional de erro do método FWI utilizando os algoritmos NSPG e L-BFGS.

Os resultados analisados foram gerados através de experimentos utilizando o The Julia Devito Inversion framework (JUDI). O método FWI foi utilizado para encontrar o campo de velocidade em um meio acústico com malha regular a partir do campo de velocidade suavizado e constante e de dados sintéticos do modelo overthrust da SEG/EAGE com e sem ruído.

2 Revisão teórica

2.1 Modelagem sísmica

A modelagem sísmica é o processo de reproduzir, através de modelos matemáticos, a propagação de uma onda em um meio, decorrente duma perturbação mecânica causada por uma fonte sísmica, localizada num ponto da subsuperfície da Terra.

Para a realização da modelagem sísmica são necessários os seguintes procedimentos:

- Uma representação da terra utilizando um campo de propriedades físicas e seus contrastes como, por exemplo, o campo de velocidades acústicas (Figura 1);
- Uma representação matemática de uma fonte sísmica, a ser inserida num ponto da terra de forma a simular uma perturbação mecânica. Normalmente, esse ponto se encontra na superfície (Figura 1);
- Uma relação física que descreva a propagação da onda sísmica gerada pela fonte. Exemplo: equação da onda acústica;
- Uma solução numérica dessa relação física, que permita uma implementação computacional estável. Alguns exemplos são: método de diferenças finitas, método pseudo-espectral, método de expansão rápida.

A técnica de modelagem sísmica utilizada no presente trabalho consiste em resolver numericamente a equação da onda com o objetivo de obter sismogramas sintéticos referentes a um dado modelo geológico de subsuperfície.

A equação da onda é descrita pela derivada parcial de segunda ordem para o deslocamento vetorial u , experimentado por um ponto ou partícula em um meio sólido devido à passagem de uma onda. Uma vez que u é conhecido (depois de resolver a equação da onda), uma geometria de aquisição pode ser simulada e os sismogramas sintéticos serem produzidos. Existe muitos métodos com suas variantes, que são comumente usados para modelar traços sintéticos, como, por

exemplo, usando traçamento dos raios (LIU et al., 2012), o método das diferenças finitas (ROBERTSSON; BLANCH; SYMES, 1994). Nesta dissertação foi utilizado o método de diferenças finitas.

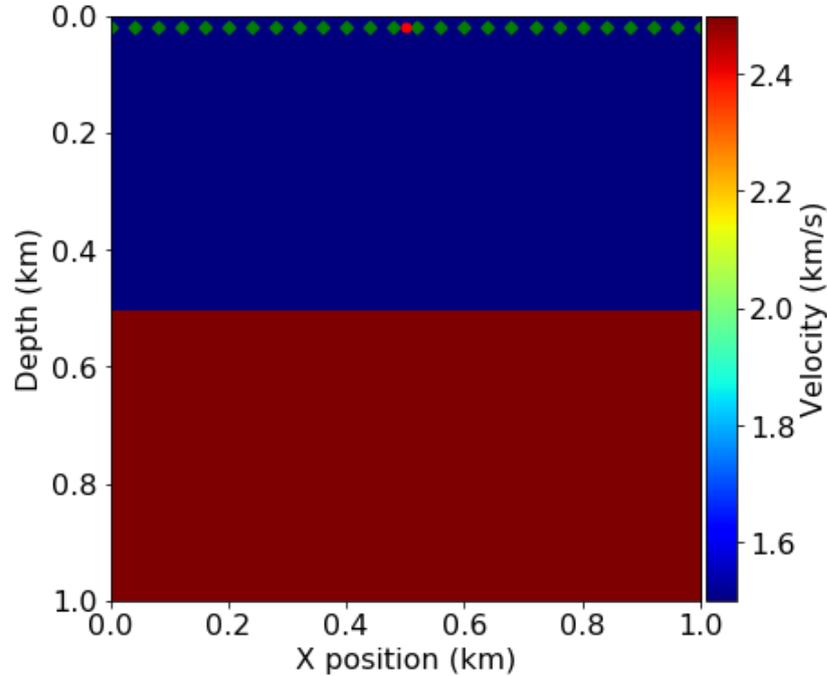


Figura 1: Definição de modelo com contraste de velocidade acústica podendo ser utilizado na modelagem sísmica com a fonte(vermelho) e os receptores(verde) na superfície.

Na formulação acústica da modelagem sísmica, utilizada nesta dissertação, a representação da subsuperfície é feita por uma malha 2D regular, que assume a forma de uma matriz (z, x) , onde cada ponto desta malha assume um valor de pressão no decorrer do tempo $p(z, x, t)$ e um valor de velocidade $v(z, x)$. No tempo inicial, o campo de pressão é igual a zero para toda malha $p(z, x, 0) = 0$, exceto no ponto (z_s, x_s) em que se introduz a fonte sísmica $s(t)$, ou seja, $p(z_s, x_s, 0) = s(t)$. O pulso sísmico $w(t)$ considerado ideal para a modelagem é um delta de Dirac (*spike*) no tempo $t = 0$. Contudo, é necessário simular um pulso sísmico real com uma duração temporal, para que possa fazer a propagação do mesmo em um modelo de subsuperfície. Tomando como exemplo o pulso sísmico do tipo Ricker com a frequência máxima de 10Hz (*Figura 2*), o pulso tem um período de $T_{pulso} = \frac{1}{10} = 100 \text{ ms}$, o que significa que o tempo para que toda a forma do pulso seja inserida no modelo seja de 100 ms.

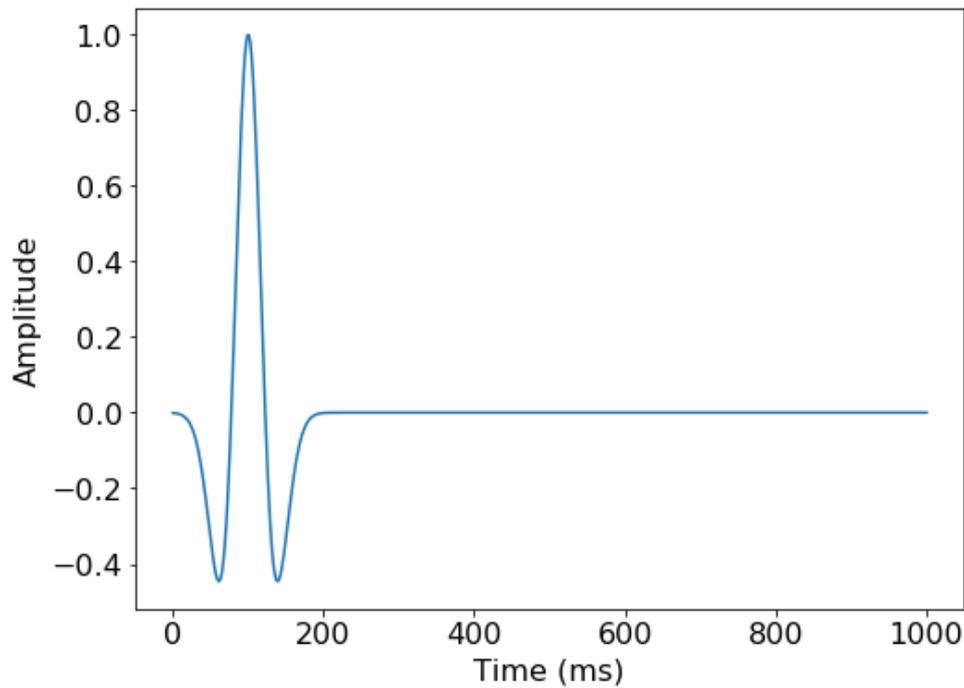


Figura 2: Wavelet de Ricker com frequência de pico de 10Hz.

Como o objetivo desta dissertação é simular o ambiente real de aquisição sísmica, serão considerados dados observados gerados de forma sintética, cuja fonte sísmica é acionada na superfície, ou seja, $p(z = 0, x, t)$.

2.1.1 Equação da onda acústica

A FWI é baseada na solução do problema direto, que consiste na simulação da propagação de ondas sísmicas através de um modelo de subsuperfície e no cálculo dos sismogramas sintéticos.

A propagação de ondas sísmicas pode ser modelada usando a equação da onda elástica:

$$\rho(x)\ddot{u}(x, t) - \nabla \cdot \sigma(x, t) = f(x, t), \quad x \in G \subset R^3, \quad t \in [t_0, t_1] \subset R, \quad (1)$$

Esta equação relaciona o deslocamento da partícula u na subsuperfície com sua densidade de massa $\rho(x)$, o tensor de tensão σ e uma fonte de força externa f . A Eq. (1) é baseada na segunda lei de Newton, balanceando momento de deslocamento da partícula $\rho(x)\ddot{u}(x, t)$, forças resultantes da tensão interna $\nabla \cdot \sigma(x, t)$ e as forças externas $f(x, t)$, esta última representada pela fonte de onda

sísmica. Para um maior detalhamento de como a Eq. (1) pode ser obtida são recomendados os seguintes trabalhos (DAHLEN; TROMP, 1998), (KENNETT, 2001) e (AKI; RICHARDS, 2002).

Na superfície ∂G as componentes normais do tensor de stress σ desaparecem, ou seja:

$$\sigma \cdot n|_{x \in \partial G} = 0, \quad (2)$$

na qual n é o vetor unitário em ∂G . A Eq. (2) é a condição de contorno para superfície livre. O deslocamento da partícula u e o campo de velocidade $v = \dot{u}$ precisam satisfazer a condição inicial de estado igual a zero antes de $t = t_0$, quando a força externa f começa a atuar:

$$u|_{t \leq t_0} = v|_{t \leq t_0} = 0. \quad (3)$$

Por conveniência é escolhido $t_0 = 0$. Para obter uma descrição mais completa, o tensor de tensão σ precisa ser relacionado ao campo de deslocamento da partícula u . Para isso, é assumido que σ depende linearmente do tensor de deformação ($\varepsilon := \frac{1}{2}(\nabla u + \nabla u^T)$):

$$\sigma(x, t) = \int_{-\infty}^{\infty} \dot{C}(x, t - t') \varepsilon(x, t') dt', \quad (4)$$

A Eq. (4) define a reologia viscoelástica linear. O tensor de quarta ordem \dot{C} é o tensor elástico. Como a tensão atual não depende da deformação futura, o tensor elástico é definido como causal:

$$C(t)|_{t \leq t_0} = 0. \quad (5)$$

A simetria do tensor de deformação ε , a conservação de momento angular e a relação de C com a energia interna (ver, por exemplo, Aki & Richards, 2002) permitem que os componentes de C satisfaçam a seguinte relação de simetria:

$$C_{ijkl} = C_{klij} = C_{jikl}. \quad (6)$$

As simetrias do tensor elástico reduzem o seu número de componentes independentes para 21 e permite escrever a Eq. (4) diretamente em termos do gradiente de deslocamento:

$$\sigma(x, t) = \int_{-\infty}^{\infty} \dot{C}(x, t - t') \nabla u(x, t') dt'. \quad (7)$$

O número de componentes de tensores elásticos independentes não nulos - também referidos como parâmetros elásticos ou módulos elásticos - determinam as propriedades anisotrópicas do meio. Por exemplo, um cristal triclinico como o plagioclásio requer todos os 21 parâmetros independentes para sua descrição completa e são necessários apenas três parâmetros elásticos para descrever cristais com simetria cúbica como granada (BABUSKA; CARA, 2012). Em uma escala macroscópica, a Terra pode ser descrita aproximado como um corpo isotrópico, onde os componentes dos tensores elásticos são combinações lineares de apenas dois módulos elásticos, as constantes de Lamé λ e μ :

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \delta_{ik} \delta_{jl} + \mu \delta_{il} \delta_{jk}. \quad (8)$$

O parâmetro μ relaciona a deformação com as tensões de cisalhamento e é chamado de módulo de cisalhamento. Como λ não tem um significado físico, é comumente usado em seu lugar o módulo de incompressibilidade (*bulk modulus*, $k = \lambda + \frac{2}{3}\mu$) que relaciona a deformação com o escalar de pressão, através da lei de Hooke para o caso 2D:

$$p(z, x, t) =: -k(z, x) \nabla \cdot u(z, x, t). \quad (9)$$

A Eq. (9) apresenta uma relação linear da entre a pressão e a velocidade da partícula u . Incluindo um termo fonte $q(z, x, t)$ adimensional (área injetada por unidade de área) tem-se:

$$p(z, x, t) =: -k(z, x) \nabla \cdot u(z, x, t) - q(z, x, t). \quad (10)$$

A segunda lei de Newton aplicada ao elemento de área perturbado, após algumas considerações algébricas, resulta em:

$$-\nabla p(z, x, t) = \rho(z, x) \frac{\partial^2 u(z, x, t)}{\partial t^2} \quad (11)$$

onde ρ é a densidade do meio. Adicionando uma força externa à Eq. (11), que deve ter dimensão de força por unidade de área, temos:

$$f(z, x, t) - \nabla p(z, x, t) = \rho(z, x) \frac{\partial^2 u(z, x, t)}{\partial t^2}. \quad (12)$$

Pode-se então eliminar u das equações (10) e (12), dividindo Eq. (12) por p , e calculando o seu divergente ($\nabla \cdot$); diferenciando a Eq. (10) duas vezes no tempo $\frac{\partial^2}{\partial t^2}$ e subtraindo as equações (12) de (10) de forma a eliminar u , resulta na equação da onda acústica:

$$\nabla \cdot \left[\frac{1}{\rho(z, x, t)} \nabla p(z, x, t) \right] - \frac{1}{k(z, x)} \frac{\partial^2 p(z, x, t)}{\partial t^2} = -s(z, x, t), \quad (13)$$

na qual:

$$s(z, x, t) = \frac{\partial^2 q(z, x, t)}{\partial t^2} - \nabla \cdot \left[\frac{f(z, x, t)}{\rho(z, x)} \right]. \quad (14)$$

Eq. (14), que expressa a fonte s de perturbação de pressão em termos das fontes q e f , das leis de Hooke e Newton, tem um papel teórico na solução da equação da onda para meios não homogêneos (GISOLF; VERSCHUUR, 2010). Nesta dissertação, essas fontes não serão mais abordadas, e $s(z; x; t)$ será expressada em termos de um pulso sísmico e de sua posição num modelo 2D, como é mostrado a seguir.

Assumindo um meio de densidade constante e sabendo que a velocidade de propagação de uma onda longitudinal de pressão (onda compressional) é dada por $c_0(z, x) = \sqrt{k(z, x)/\rho(z, x)}$ (KLIMENTOS, 1991), obtém-se a equação que segue:

$$\frac{1}{c_0^2(z, x)} \frac{\partial^2 p(z, x, t)}{\partial t^2} = \nabla^2 p(z, x, t) - s(z, x, t). \quad (15)$$

A Eq. (15) é conhecida como equação de onda acústica 2D para o caso de densidade constante. De forma a tornar a nomenclatura mais prática, a partir deste ponto é utilizado o termo “equação da onda acústica”.

Um dos componentes de suma importância na teoria do imageamento sísmico é o campo de onda de uma fonte pontual. Esse é o tipo de fonte utilizada neste trabalho. A equação de uma onda acústica, na presença de uma fonte pontual localizada em (z_s, x_s) é dada por:

$$\frac{1}{c_0^2(z, x)} \frac{\partial^2 p(z, x, t)}{\partial t^2} = \nabla^2 p(z, x, t) - w(t) \delta(z - z_s, x - x_s). \quad (16)$$

na qual $w(t)$ é a wavelet da fonte. Um tipo de wavelet muito usual e também utilizada neste trabalho é a Ricker, que é definida matematicamente como:

$$w(t) = (1 - 2t_r^2) e^{-t_r^2}, \quad (17)$$

na qual:

$$t_r = \pi f_{peak} \left(t - \frac{1}{f_{peak}} \right). \quad (18)$$

Analisando a transformada de Fourier da wavelet de Ricker é possível compreender os conceitos de frequência de pico e frequência máxima. A frequência de pico f_{peak} é o componente monofrequência de maior amplitude da wavelet. A frequência máxima f_{max} é o último componente monofrequência que possui correlação (amplitude relevante) com a wavelet. Segundo (DABLAIN, 1986), as duas frequências guardam a seguinte relação $2f_{peak} \approx f_{max}$.

Para proceder com a modelagem sísmica acústica é necessário resolver a Eq. (15). Para isto, podem ser utilizadas uma variedade de técnicas numéricas de solução de equação da onda, tais como, diferenças finitas, pseudo-spectral e método de expansão rápida. O software JUDI faz uso da solução com diferenças finitas, que é o tópico abordado na próxima seção.

2.1.2 Diferenças finitas

A equação da onda acústica 2D para um meio sem variação de densidade como na Eq. (15) é equivalente a:

$$\frac{\partial p^2(z, x, t)}{\partial x^2} + \frac{\partial p^2(z, x, t)}{\partial z^2} - \frac{1}{c_0(z, x)^2} \frac{\partial^2 p(z, x, t)}{\partial t^2} = s(x, z, t). \quad (19)$$

A equação da onda contínua (Eq. (19)) e sua solução pode ser discretizada em uma grade uniformemente amostrada nos domínios do espaço e do tempo de modo que:

$$(x, z, t) \leftrightarrow (i\Delta x, j\Delta z, t\Delta t), \quad (20)$$

$$p(x, z, t) \leftrightarrow p_{ij}^t, \quad (21)$$

$$s(x, z, t) \leftrightarrow s_{ij}^t, \quad (22)$$

$$c(x, z) \leftrightarrow c_{ij}, \quad (23)$$

na qual i, j, t são números inteiros e $\Delta x, \Delta z, \Delta t$ são a amostragem no espaço e tempo. No software JUDI foi utilizado o espaçamento de *grid* horizontal igual ao vertical, ou seja, $\Delta x = \Delta z$.

Uma aproximação de derivadas por diferenças finitas pode ser feita através do truncamento de séries de Taylor (ARAUJO; PESTANA, 2010). O objetivo é achar uma aproximação para a derivada segunda em z de $p(z; x; t)$. As variáveis $(x; t)$ foram omitidas de forma a tornar a demonstração mais didática. Então, partindo da série de Taylor para $p(z + \Delta z)$, aproximada em torno do ponto $p(z)$ obtém-se:

$$p(z + \Delta z) = p(z) + \frac{\partial p(z)}{\partial z} \Delta z + \frac{\partial^2 p(z)}{\partial z^2} \frac{\Delta z^2}{2!} + \frac{\partial^3 p(z)}{\partial z^3} \frac{\Delta z^3}{3!} + \dots \quad (24)$$

Em seguida, fazendo a série de Taylor para $p(z - \Delta z)$, aproximada em torno do ponto $p(z)$:

$$p(z - \Delta z) = p(z) - \frac{\partial p(z)}{\partial z} \Delta z + \frac{\partial^2 p(z)}{\partial z^2} \frac{\Delta z^2}{2!} - \frac{\partial^3 p(z)}{\partial z^3} \frac{\Delta z^3}{3!} + \dots \quad (25)$$

Somando as equações (24) e (25) tem-se:

$$p(z + \Delta z) + p(z - \Delta z) = 2p(z) + 2 \frac{\partial^2 p(z)}{\partial z^2} \frac{\Delta z^2}{2!} + \dots \quad (26)$$

Desprezando os termos de ordem superior a 2 (ou maiores que $O(\Delta z^2)$), a derivada segunda em z poder ser aproximada por:

$$\frac{\partial p^2(z)}{\partial z^2} \approx \frac{p(z + \Delta z) - 2p(z) + p(z - \Delta z)}{\Delta z^2}. \quad (27)$$

Esta é a aproximação por diferenças finitas centrada de segunda ordem, visto que a série foi truncada na segunda ordem e que a aproximação foi feita utilizando os pontos vizinhos ao ponto centrado em $p(z)$. Utilizando a mesma analogia, pode-se obter a aproximação por diferenças finitas centrada de segunda ordem para x e t . Existem outras formas de aproximações, tais como, diferenças progressivas e regressivas, todavia, a diferença centrada apresenta uma precisão superior se comparada com as outras formas (BURDEN; FAIRES, 2010). Mudando o campo de pressão em (i, j, t) e substituindo as derivadas de segunda ordem na Eq. (19) pela aproximação de diferenças finitas, centrada de segunda ordem, obtida na Eq. (27) para x , z e t , tem-se:

$$\frac{\partial p^2(z)}{\partial t^2} \leftrightarrow \frac{[p_{ij}^{t+1} - 2p_{ij}^t + p_{ij}^{t-1}]}{(\Delta t^2)}, \quad (28)$$

$$\frac{\partial p^2(z)}{\partial x^2} \leftrightarrow \frac{[p_{i+1j}^t - 2p_{ij}^t + p_{i-1j}^t]}{(\Delta x^2)}, \quad (29)$$

e:

$$\frac{\partial p^2(z)}{\partial z^2} \leftrightarrow \frac{[p_{ij+1}^t - 2p_{ij}^t + p_{ij-1}^t]}{(\Delta z^2)}. \quad (30)$$

Substituindo as equações (28), (29) e (30) na Eq. (19) e manipulando os termos é obtido o passo no tempo do campo de pressão $p(x, z, t)$ dado por:

$$p_{ij}^{t+1} = 2p_{ij}^t - p_{ij}^{t-1} + a[p_{i+1j}^t - 2p_{ij}^t + p_{i-1j}^t] + a[p_{ij+1}^t - 2p_{ij}^t + p_{ij-1}^t] - (\Delta x)^2 a s_{ij}^t, \quad (31)$$

na qual $a = (c_{ij} \frac{\Delta t}{\Delta x})^2$ e as condições iniciais são dadas por $p(x, z, t = 0)$ e $\frac{\partial p(z, x, t)}{\partial t}$ para todo (x, z) , de tal forma que a fonte do campo de onda é gerado pelo termo s_{ij}^t . A Figura 3 ilustra a representação gráfica da diferença finita para a Eq. (31) de onda acústica 2D. O valor futuro do campo de pressão no ponto (i, j) (círculo tracejado) é calculado utilizando o valor presente e passado do campo de pressão no ponto (i, j) e na vizinhança do ponto no presente. Na literatura, esta configuração recebe o nome de *stencil* e pode ser deslocada dentro do plano t para calcular todos os valores de pressão no plano $t + 1$.

A *Figura 4* apresenta o cone de iluminação progressivo de diferenças finitas (alguns *stencils* são representados) para a equação da onda de um ponto em profundidade. A transformação de coordenadas de t para $-t$, ou seja, reversão no tempo, não irá mudar a forma da equação da onda (Eq. (19)). Isso significa que a resposta da equação da onda reversa no tempo será a mesma que o original, exceto pela mudança no sinal da variável temporal. Desenvolvendo um pouco mais o raciocínio nesta linha, um cone de iluminação regressivo pode ser gerado utilizando equações de diferenças finitas para resolver a regressão no tempo. Tais cones regressivos são usados para criar a história anterior do campo de ondas, o qual é conhecido como extrapolação reversa no tempo da equação da onda. Este conceito será importante para o desenvolvimento do problema de *reverse time migration* (RTM) como será mostrado a seguir.

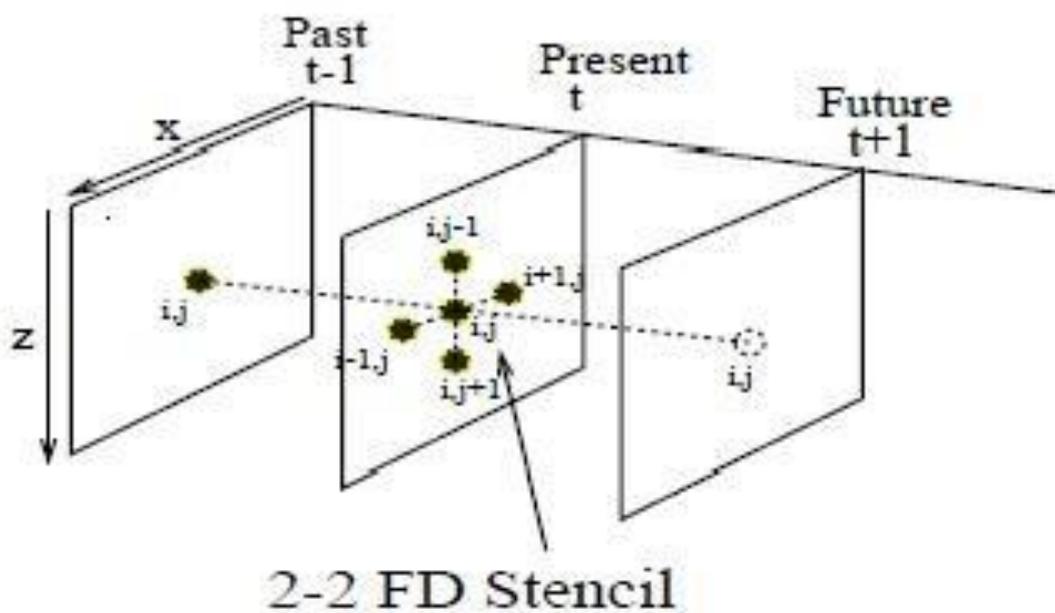


Figura 3: Estêncil 2-2 diferença finita.

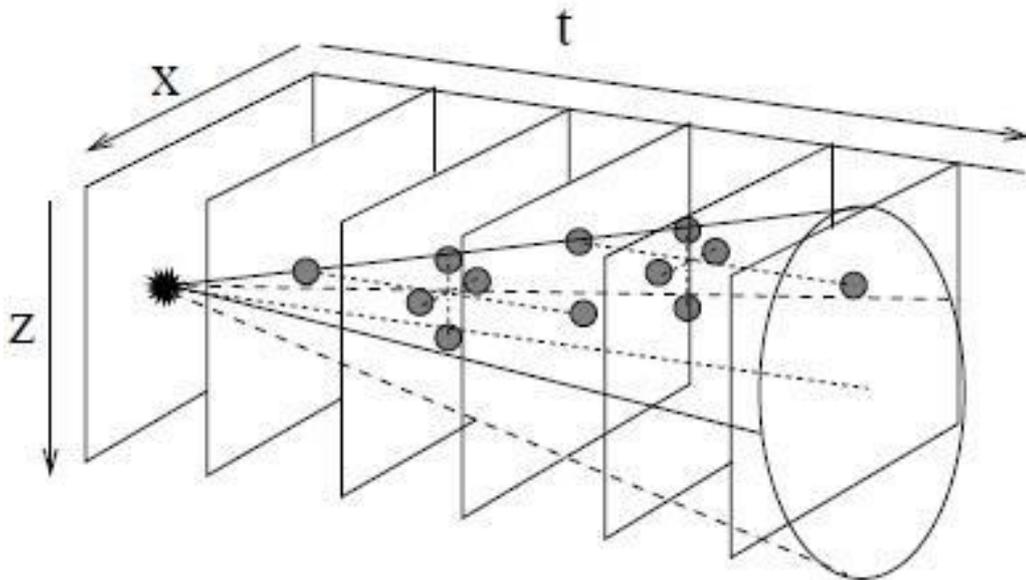


Figura 4: Cone de iluminação progressivo, formado a partir da progressão no tempo do stencil.

2.1.3 Estabilidade numérica

A aproximação do campo de onda utilizando a equação da onda acústica com densidade constante é feita através do método de diferenças finitas, para aproximar as derivadas espaciais e temporais. A aproximação consiste em substituir as quantidades infinitesimais $(\frac{\partial}{\partial x}, \frac{\partial}{\partial z}, \frac{\partial}{\partial t})$ por quantidades finitas $(\Delta x, \Delta z, \Delta t)$. Esta substituição pode gerar resultados indesejáveis, que não correspondem ao comportamento descrito pela equação da onda acústica. Um dos resultados é a instabilidade numérica, que ocorre quando as amplitudes da onda modelada crescem de forma anômala na extrapolação, gerando um campo de onda *NaN* (*Not a number*, resultado da tentativa de divisão por zero) e/ou *Inf* (*Infinity*, quando o resultado de uma operação gera um número maior que a precisão do processador).

(LINES; SLAWINSKI; BORDING, 1999) desenvolveram heurísticamente uma forma aproximada para determinar se os espaçamentos $(\Delta x, \Delta z, \Delta t)$ fazem a extrapolação do campo de onda ser estável:

$$\frac{c_0 \Delta t}{h} \leq \sqrt{\frac{a_1}{a_2}} \quad (32)$$

na qual c_0 é a velocidade sísmica do meio, neste caso substituída por $\max(c_0)$ para garantir a estabilidade em todos os pontos, h é o intervalo máximo do *grid* $\max(\Delta x, \Delta z)$, a_1 a soma dos pesos do operador de diferenças finitas temporal e a_2 a soma dos pesos dos operadores de diferenças finitas espaciais. Esta soma é definida por:

$$a = \sum_{m=-M}^M |w_m|, \quad (33)$$

na qual M é a metade da ordem do operador escolhido. Por exemplo, para o operador de segunda ordem $w = (1, -2, 1)$, resultando $a = 4$. Como exposto, a aproximação da derivada temporal é de segunda ordem, desta forma, modificando a Eq. (32), tem-se:

$$\frac{c_0 \Delta t}{h} \leq \frac{2}{\sqrt{a_2}} \quad (34)$$

Isolando o termo Δt na equação tem-se a condição para estabilidade dependendo da velocidade máxima do modelo, espaçamento e tamanho do operador de diferenças finitas:

$$\Delta t \leq \frac{2 \max(\Delta x, \Delta z)}{\max(c_0) \sqrt{a_2}}. \quad (35)$$

Falta resolver a_2 que envolve dois operadores espaciais em (x, z) , o que leva a equação modificada de (LINES; SLAWINSKI; BORDING, 1999):

$$a = \sum_{m=-M}^M |w_{mz}| + |w_{mx}|. \quad (36)$$

Usando um operador espacial de segunda ordem, temos $a_2 = 8$, logo a Eq. (35) se transforma em:

$$\Delta t \leq \frac{2 \max(\Delta x, \Delta z)}{\max(c_0) 2\sqrt{2}}. \quad (37)$$

2.2 Migração reversa no tempo

As equações para o algoritmo de migração reversa no tempo (do inglês, *reverse time migration* - RTM) são apresentadas neste capítulo. Atualmente, essa

migração é muito usada pela indústria do petróleo para produzir imagens abaixo de corpos salinos, sendo também utilizada para migrar iterativamente resíduos para o *Least Square Migration* (LSM). Em comparação com a migração Kirchhoff, a RTM considera todas as chegadas no campo de onda, incluindo primárias e múltiplas, se o modelo de velocidade for suficientemente preciso. Isso pode levar a uma resolução muito melhor na imagem, mas ao custo de ser mais sensível a erros no modelo de velocidade usado para a migração. Para frequências inferiores a 20 Hz, a RTM é o método de migração preferido para imagens abaixo de geologia complexa, como diápiros de sal (ROY*; CHAZALNOEL, 2011).

Os métodos de migração tradicionais pós-empilhamento e pré-empilhamento não lidam com o efeito de múltiplas do dado sísmico. Este é um problema particularmente pernicioso abaixo de corpos de sal, que tendem a desfocar a incidência e reflexão do campo de ondas. Para contornar este problema, (MCMECHAN, 1983), (WHITMORE, 1983), e (BAYSAL; KOSLOFF; SHERWOOD, 1983) introduziram o conceito de migração reversa no tempo. Ao invés de usar o traçamento de raio para calcular a função de Green, busca-se uma solução da equação de onda por diferenças finitas. Se a velocidade for conhecida com precisão, então, todos os tipos de evento de propagação de onda são contabilizados, incluindo múltiplas, difrações e ondas convertidas. O benefício pode ser, se o modelo de velocidade é conhecido com precisão, um aumento significativo na razão sinal/ruído da imagem migrada, melhora na imagem de refletores com mergulho íngreme e habilidade para ver eventos a abaixo de corpos com alto contraste de velocidade.

O algoritmo de RTM pode ser obtido a partir do algoritmo de imageamento generalizado, encontrado pelo método de otimização do gradiente do funcional de erro, podendo ser descrito em 3 passos:

1 - Funcional de erro: assumindo que existe um harmônico de energia emanando de uma fonte s e sendo registado em um geofone g . Por questões de simplicidade o meio de propagação é considerado como uma aproximação acústica de tal forma que o espectro do campo de pressão observado pode ser representado

por $P(s)^{obs}$, onde o pulso sísmico da fonte desconhecido $w(t)$ tem o espectro fornecido por $W(\omega)$. O funcional de erro ϵ é dado por:

$$\epsilon = \frac{1}{2} \int_{\omega} d\omega \sum_g \sum_s |\Delta P(s)|^2. \quad (38)$$

na qual $\Delta P(s)$ é o dado residual definido por $\Delta P(s) = P(s)^{calc} - P(s)^{obs}$ e $P(s)^{calc}$ é o campo de pressão calculado pela solução numérica da equação da onda acústica. Nesta etapa, um modelo de velocidade inicial com aproximação acurada do modelo geológico em subsuperfície é necessário. A integral da Eq. (38) é feita sobre o domínio da frequência, enquanto os somatórios são realizados sobre as coordenadas das fontes e receptores. O campo de pressão modelado é obtido pela solução por diferenças finitas da equação da onda acústica no domínio espaço-tempo ou espaço-frequência.

2 - Gradiente de vagarosidade e a derivada de Fréchet: utilizando algum método gradiente descendente, o modelo de vagarosidade $s(x, z) = \frac{1}{c^2}$ pode ser atualizado iterativamente por:

$$s(x, z)^{(i+1)} = s(x, z)^{(i)} - \alpha \gamma(x, z)^i, \quad (39)$$

na qual α é o comprimento do passo e $\gamma(x, z)^i$ é o gradiente do funcional de erro definido como:

$$\gamma(x, z)^i = \frac{\partial \epsilon^i}{\partial s(x, z)}, \quad (40)$$

$$\gamma(x, z)^i = \frac{1}{2} \int_{\omega} d\omega \sum_g \sum_s \text{Real} \left[\Delta P(s)_i \frac{\partial P(s)_i}{\partial s(x, z)} \right], \quad (41)$$

na qual o dado residual $\Delta P(s)_i = P(s)_i - P(s)^{obs}$ é atualizado a cada iteração. O operador $\text{Real}[\]$ se refere a parte real do número complexo entre colchete, deste ponto em diante esta notação é descartada, pois a soma no domínio da frequência é simétrica e a parte imaginária do somatório é antissimétrica em ω .

Assumindo que a perturbação no campo de pressão P com relação a vagarosidade $s(x, y)$ é a derivada de Fréchet, assim, sua substituição transforma a Eq. (41) do gradiente em:

$$\gamma(x)^i = 2s(x) \int_{\omega} d\omega \omega^2 \sum_g \sum_s W(\omega)^* \Delta P(s)_i G(x|g)_i^* G(x|s)_i, \quad (42)$$

na qual $G(x, x')_i$ é a função de Green calculada para um modelo de vagarosidade específico.

3-Atualização do modelo de vagarosidade: Substituindo a Eq. (42) na Eq. (39) tem-se a equação de atualização da vagarosidade:

$$s(x, z)^{(i+1)} = s(x, z)^{(i)} - \alpha 2s(x) \int_{\omega} d\omega \omega^2 \sum_g \sum_s W(\omega)^* \Delta P(s)_i G(x|g)_i^* G(x|s)_i, \quad (43)$$

Estes 3 passos são repetidos até a convergência de $s(x, z)$ e os algoritmos de migração RTM e de inversão FWI podem ser obtidos da Eq. (43).

Reagrupando a Eq. (43) para $i = 0$ temos:

$$\begin{aligned} m(x)^{mig} &= s(x)^{(1)} - s(x)^{(0)} \\ &= \alpha 2s(x) \int_{\omega} d\omega \omega^2 \sum_g \sum_s \Delta P(s) W(\omega)^* G(x|g)^* G(x|s), \end{aligned} \quad (44)$$

na qual $m(x)^{mig}$ é a imagem migrada.

A Eq. (44) pode ser manipulada em uma forma mais interpretável:

$$m(x)^{mig} = - \int_{\omega} d\omega \omega^2 \sum_g [W(\omega) G(x|s)] * [G(g|x) * d'(g|s)], \quad (45)$$

$$m(x)^{mig} = [w(t) * g(x, t|s, 0)] \otimes \sum_g g(g, -t|x) * \ddot{d}(g, t|s, 0)|_{t=0}, \quad (46)$$

$$m(x)^{mig} = \sum_t D(x, t) U(x, t), \quad (47)$$

na qual, convenientemente, $s(x) = \frac{1}{2\pi}$ assume uma fonte única. Com $d'(s) = \Delta P(s)$ representando os traços que contém somente ondas refletidas ascendentes. A Eq.

(47) representa a imagem RTM pré-empilhamento $m(x)^{mig}$, formada no ponto de imagem experimental x , assumindo a correlação temporal com atraso zero entre o campo de fonte descendente $D(x,t)$ e as reflexões retropropagadas $U(x,t)$ (MCMECHAN, 1983);(WHITMORE, 1983).

O termo $D(x,t) = w(t) * g(x,t|s,0)$ representa o campo de onda descendente, enquanto as reflexões retropropagadas ascendentes são dadas por $U(x,t) = \sum_g g(g,-t|x) * \ddot{d}(g,t|s,0)|_{t=0}$.

Para uma fonte impulsiva, essas reflexões retropropagadas coincidem com $D(x,t)$ somente quando $x = x_{scatt}$ visto no cubo espaço-tempo da *Figura 5*. Isto significa que o único produto escalar diferente de zero $D(x,t)U(x,t)$ está na posição de espalhamento.

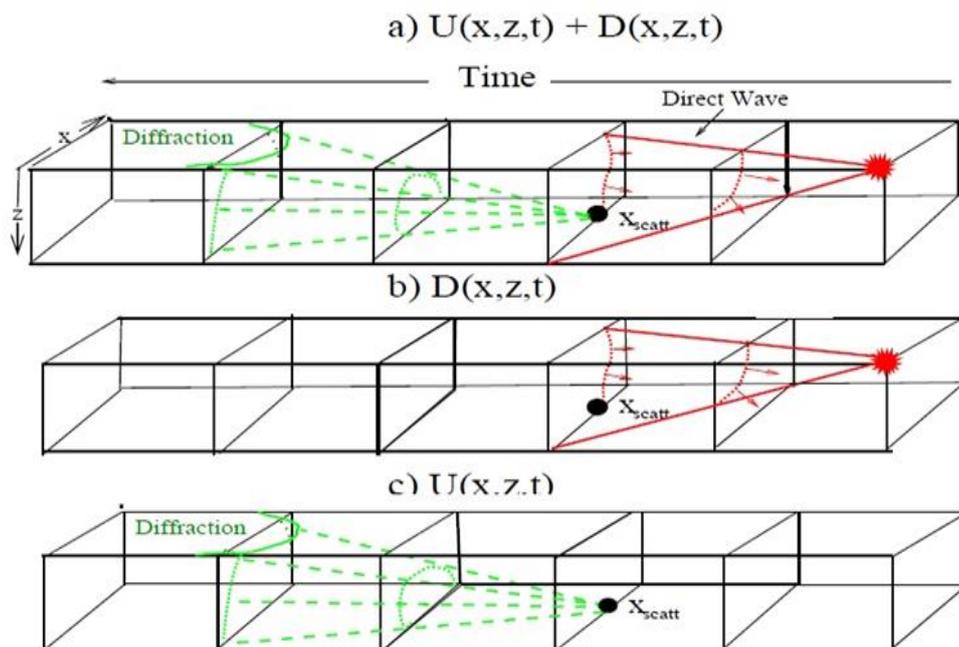


Figura 5: Ponto de espalhamento em subsuperfície x_{scatt} único local onde o produto escalar das ondas propagadas descendentes e retropropagadas ascendentes é diferente de zero.

2.3 Inversão completa da forma da onda (FWI)

Neste capítulo é introduzido a teoria da inversão completa da forma da onda (do inglês, *full waveform inversion* - FWI) utilizando notação matricial com base no artigo (VIRIEUX; OPERTO, 2009). É dada ênfase aos pontos essenciais para

entendimento do problema de FWI utilizado nesta dissertação, também são apresentados os métodos L-BFGS e NSPG para resolução do problema de FWI.

O problema de FWI consiste em, através de dados observados u , obter um modelo que descreva um parâmetro físico em subsuperfície m utilizando um operador G , que relaciona os dados observados (medidos) e o modelo. Escrevendo na forma matricial, tem-se:

$$u = G(m), \quad (48)$$

No método inverso, o modelo m é encontrado obtendo o operador inverso G^{-1} de forma que m seja encontrado por $m = G^{-1}(u)$. Contudo, o problema proposto para a solução da equação da onda é não linear, impossibilitando encontrar analiticamente G^{-1} .

A forma alternativa para resolver o problema é utilizar a formulação de mínimos quadrados para o problema, que consiste em achar o melhor m , em iterações sucessivas a partir de um modelo inicial m_0 , que minimize uma função objetivo de erro do tipo (TARANTOLA, 1984b):

$$E(m) = (d_{obs} - d_{cal})^T (d_{obs} - d_{cal}) \frac{1}{2}, \quad (49)$$

na qual d_{obs} é o dado observado, d_{cal} o dado calculado pelo modelo.

Na modelagem sísmica, os dados observados d_{obs} podem ser considerados como o campo de pressão medido ao longo do tempo, apenas na superfície. O modelo inicial m_0 consiste em uma estimativa inicial do modelo de velocidade, dado pelo conhecimento *a priori* da geologia da área de estudo. Neste trabalho são utilizados dados sintéticos e o modelo inicial é dado pela suavização do modelo verdadeiro de vagarosidade.

Conforme mencionado, a formulação de mínimos quadrados pode ser utilizado para resolver o problema de FWI. Esta formulação consiste na expansão por série de Taylor de segunda ordem do funcional de erro em torno do modelo inicial m_0 dado por:

$$\begin{aligned}
E(m_0 + \Delta m) &= E(m_0) + \sum_{j=1}^N \frac{\partial E(m_0)}{\partial m_j} \Delta m_j + \sum_{l=1}^N \sum_{j=1}^N \frac{\partial^2 E(m_0)}{\partial m_l \partial m_j} \Delta m_l \Delta m_j \\
&\quad + O(\Delta m^3) \rightarrow, \\
E(m_0 + \Delta m) &\approx E(m_0) + \sum_{j=1}^N \frac{\partial E(m_0)}{\partial m_j} \Delta m_j + \sum_{l=1}^N \sum_{j=1}^N \frac{\partial^2 E(m_0)}{\partial m_l \partial m_j} \Delta m_l \Delta m_j, \quad (50)
\end{aligned}$$

sendo N o número total de parâmetros do modelo. Os termos $O(\Delta m^3)$ são desprezados. Observe que a Eq. (50) descreve um parabolóide em Δm . O mínimo da aproximação em torno de m_0 é o ponto no qual $\frac{\partial E(m)}{\partial m_i} = 0$. Para $i = 1, \dots, N$ (ver *Figura 6* para o caso de 2 parâmetros).

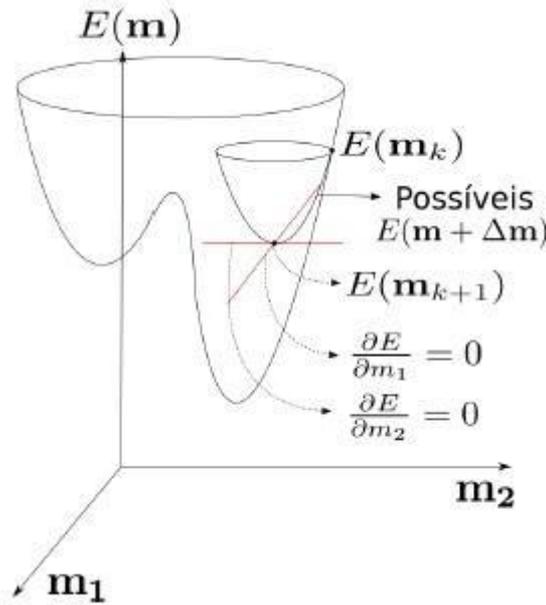


Figura 6: Esboço da função objetivo não-linear $E(m)$ para um caso de 2 parâmetros (m_1 e m_2). $E(m+\Delta m)$ é um parabolóide em função de Δm . Em vermelho, derivadas nulas em cada parâmetro dão o ponto de mínimo.

Derivando a Eq. (50) com relação a m_i , tem-se:

$$\frac{\partial E(m_0 + \Delta m)}{\partial m_i} = \frac{\partial E(m_0)}{\partial m_i} + \sum_{j=1}^N \frac{\partial^2 E(m_0)}{\partial m_i \partial m_j} \Delta m_j + \sum_{l=1}^N \sum_{j=1}^N \frac{\partial^3 E(m_0)}{\partial m_i \partial m_l \partial m_j} \Delta m_l \Delta m_j \quad (51)$$

e:

$$\frac{\partial E(m_0)}{\partial m_i} + \sum_{j=1}^N \frac{\partial^2 E(m_0)}{\partial m_i \partial m_j} \Delta m_j + \sum_{l=1}^N \sum_{j=1}^N \frac{\partial^3 E(m_0)}{\partial m_i \partial m_l \partial m_j} \Delta m_l \Delta m_j = 0 \quad (52)$$

Como as derivadas terceiras são nulas para a aproximação de Taylor de segunda ordem:

$$\sum_{j=1}^N \frac{\partial^2 E(m_0)}{\partial m_i \partial m_j} \Delta m_j = -\frac{\partial E(m)}{\partial m_i} \quad (53)$$

Organizando para todos os m_i em notação matricial tem-se:

$$\begin{bmatrix} \frac{\partial^2 E(m_0)}{\partial m_1^2} & \frac{\partial^2 E(m_0)}{\partial m_1 \partial m_2} & \dots & \frac{\partial^2 E(m_0)}{\partial m_1 \partial m_N} & \vdots & \vdots \\ \vdots & \frac{\partial^2 E(m_0)}{\partial m_N \partial m_1} & \frac{\partial^2 E(m_0)}{\partial m_N \partial m_2} & \dots & \frac{\partial^2 E(m_0)}{\partial m_N^2} & \vdots \end{bmatrix} [\Delta m_1 \ : \ \Delta m_N] \quad (54)$$

$$= - \begin{bmatrix} \frac{\partial E(m)}{\partial m_1} \\ \vdots \\ \frac{\partial E(m)}{\partial m_N} \end{bmatrix},$$

$$\frac{\partial^2 E(m_0)}{\partial m^2} \Delta m = -\frac{\partial E(m_0)}{\partial m}, \quad (55)$$

$$\Delta m = \left[\frac{\partial^2 E(m_0)}{\partial m^2} \right]^{-1} \left(-\frac{\partial E(m_0)}{\partial m} \right), \quad (56)$$

com $\frac{\partial^2 E(m_0)}{\partial m^2}$ e $-\frac{\partial E(m_0)}{\partial m}$ sendo, respectivamente, a matriz hessiana e o gradiente da função objetivo. Escrevendo de uma forma mais didática:

$$m = m_0 - H^{-1} \nabla E(m_0), \quad (57)$$

Para um sistema linear de equações, a Eq. (57) satisfaz o problema. Contudo, o problema FWI é não linear e sua solução precisa ser calculada iterativamente através de atualizações do gradiente e da hessiana em cada iteração.

O cálculo do gradiente e da hessiana possuem um alto custo computacional. Para contornar este problema existem diversos métodos de aproximação para o cálculo de ∇E e H^{-1} . Nesta dissertação são avaliados os métodos NSPG e L-BFGS descritos a seguir. A partir deste ponto é adotada a nomenclatura H_k para aproximação iterativa da matriz inversa hessiana; H_0 para o chute inicial da matriz inversa hessiana; $s_k = m_{k+1} - m_k$; $y_k = \nabla E_{k+1} - \nabla E_k$.

2.3.1 Método L-BFGS

O método L-BFGS para problemas de otimização não linear de grande porte, utiliza atualizações da matriz hessiana, calculadas a um custo razoável.

O método L-BFGS pode ser compreendido como uma variação do método quasi-Newton BFGS, com restrição na armazenagem de informações precedentes para o cálculo da matriz H_k , pois a atualização da matriz H_k , geralmente densa, é realizada armazenando um número restrito m de pares de vetores $\{s_k, y_k\}$ anteriores. Este método é computacionalmente menos custoso, mais robusto e de fácil implementação, mas ele possui taxa de convergência baixa. O usuário deve especificar o número m de correções que serão realizadas utilizando o método BFGS e definir a matriz H_0 , que precisa ser esparsa, simétrica, definida positiva e se aproximar da hessiana de E . Durante as primeiras m iterações $k < m$, o método L-BFGS é idêntico ao método BFGS. Para $k \geq m$, H_k é obtida usando informações dadas pelas m iterações anteriores utilizando o método BFGS. A seguir uma descrição detalhada do algoritmo é apresentada.

2.3.2 Algoritmo L-BFGS

O algoritmo começa com uma estimativa inicial do valor ideal m_0 e prossegue iterativamente para refinar o modelo inicial com uma sequência de estimativas m_1, m_2, \dots, m_n . As derivadas da função $g_k = \nabla E(m_k)$ são usadas como um direcionador chave do algoritmo para identificar a direção da descida mais íngreme e também para formar uma estimativa da matriz Hessiana (segunda derivada) de $E(m)$.

O método L-BFGS compartilha muitas características com outros algoritmos quase-Newton, mas é diferente em como a multiplicação matriz-vetor $d_k = -H_k g_k$ é calculado, na qual d_k é a aproximação do vetor direcional descendente na iteração k . Existem várias abordagens publicadas usando um histórico de atualizações para formar esse vetor de direção d_k (FABIEN-OUELLET; GLOAGUEN; GIROUX, 2017), (LEI; BOHLEN; ZHANG, 2018). Nesta dissertação é abordada a “recursão de dois loops”(NOCEDAL, 1980).

Este método usa a fórmula inversa da matriz hessiana:

$$H_{k+1} = V_k^t H_k V_k + \rho_k s_k s_k^t, \quad (58)$$

na qual:

$$\rho_k = \frac{1}{y_k^t s_k}, \quad (59)$$

e:

$$V_k = I - \rho_k s_k s_k^t, \quad (60)$$

Assim pode-se calcular o vetor direcional descende como segue. É assumido que as m iterações estão armazenadas na forma $\{s_k, y_k\}$.

Armazena informações da Hessiana das iterações $< m$

$$q = g_k$$

$$\text{for } i = k - 1, k - 2, \dots, k - m$$

$$\alpha_i = \rho_i s_i^t q$$

$$q = q - \alpha_i y_i$$

$$\gamma_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$$

$$H_0^k = \gamma_k I$$

$$z = H_0^k q$$

Desenvolvimento do L-BFGS para iterações $> m$

$$\text{for } i = k - m, k - m + 1, \dots, k - 1$$

$$\beta_i = \rho_i y_i^t z$$

$$z = z + s_i(\alpha_i - \beta_i)$$

Ao fim da execução do algoritmo, a direção de busca será dada pelo vetor $-z$. O fator γ_k serve como operador de escala que tenta aproximar os valores de H_k do inverso da hessiana.

2.3.3 Método Non-motone Spectral Projected Gradient (NSPG)

Como mencionado, a Eq. (57) pode ser escrita na forma iterativa para estimativa do parâmetro de vagarosidade que minimiza o funcional de erro (Eq. (49)) como segue:

$$m_{k+1} = m_k + \alpha_k \nabla E_k, \quad (61)$$

na qual α_k é o comprimento do passo para a iteração k .

O NSPG (BIRGIN; MARTÍNEZ; RAYDAN, 2015) gera as iterações como na Eq. (61) com o comprimento do passo dado por:

$$\alpha_k = \frac{h_{k-1}^T h_{k-1}}{h_{k-1}^T y_{k-1}}, \quad (62)$$

com:

$$h_{k-1} = m_k - m_{k-1}, \quad (63)$$

$$y_{k-1} = \nabla E(m_k) - \nabla E(m_{k-1}), \quad (64)$$

A escolha do comprimento do passo está relacionada aos autovalores da matriz hessiana da função objetivo, o que não garante decréscimo da função residual a cada iteração contrastando com os métodos clássicos de otimização descendentes.

Baseado nas propriedades do SPG, (BIRGIN; MARTÍNEZ; RAYDAN, 2015) fez uso da condição de procura em linha não monótona (GRIPPO; LAMPARIELLO; LUCIDI, 1986) que preserva o benefício do método SPG e permite convergência para qualquer chute inicial dada por:

$$\nabla E(m_{k-1}) \leq \{ \max_{0 \leq j \leq \{k, M-1\}} \nabla E(m_{k-1}) \} + \gamma \alpha_k \nabla E(m_k)^T d_k, \quad (65)$$

na qual M é um inteiro não negativo e γ é um número pequeno, para mais detalhes ver (BIRGIN; MARTÍNEZ; RAYDAN, 2015).

O método NSPG pode ser aplicado de forma restrita em um conjunto convexo de velocidade Ω , minimizando a função residual sujeita a esta restrição. O método

do gradiente projetado espectral não monótono (NSPG) é uma extensão do método de gradiente espectral, o qual lida com restrições no conjunto de soluções viáveis, projetando o gradiente da iteração atual em um conjunto restrito de soluções, logo, na Eq. (61), a direção é dada por:

$$d_k = P_{\perp}(m_k - \alpha_k \nabla E(m_k)) - m_k, \quad (66)$$

na qual P_{\perp} é a projeção ortogonal no domínio Ω . Caso o componente da iteração atual $m_k - \alpha_k \nabla E(m_k)$ não esteja em Ω , então os passos do algoritmo fazem com que o componente seja igual ao correspondente mais próximos dos limites de velocidade Ω .

O algoritmo NSPG para minimizar a função objetivo restrita a um domínio de velocidade levando em conta as equações (65) e (66) pode ser visto a seguir.

2.3.4 Algoritmo NSPG

1 Define os limites de restrição da velocidade

2 Detecta se o ponto atual é estacionário: se $\|P_{\perp}(m_k - \alpha_k \nabla E(m_k)) - m_k\| = 0$, caso positivo algoritmo para declarando que m_k é estacionário,

Caso contrário:

3 procura em linha não monótona

3.1 Calcula $d_k = P_{\perp}(m_k - \alpha_k \nabla E(m_k)) - m_k$, fazendo $\lambda = 1$

3.2 $m_+ = m_k + \lambda d_k$.

3.3 Se $\nabla E(m_+) \leq \{\max_{0 \leq j \leq \{k, M-1\}} \nabla E(m_{k-1})\} + \gamma \langle \lambda d_k, \nabla E(m_k) \rangle$

Com M inteiro positivo e $\gamma \in (0,1)$, então $\lambda_k = \lambda$,

$m_{k+1} = m_+$,

$h_k = m_{k+1} - m_k$,

$y_k = \nabla E(m_{k+1}) - \nabla E(m_k)$,

Vai para o passo 4.

Se 3.3 não é satisfeita, então

$\lambda_{new} \in [\sigma_1\lambda, \sigma_2\lambda]$, com $0 < \sigma_1 < \sigma_2 < 1$,

$\lambda = \lambda_{new}$, vai para o passo 3.2

4. Calcula $b_k = \langle h_k, y_k \rangle$.

Se $b_k \leq 0$, então $\alpha_{k+1} = \alpha_{max}$

Senão, calcula $\alpha_k = \langle h_k, h_k \rangle$ e define $\alpha_{k+1} = \min\{\alpha_{max}, \max\{\alpha_{min}, \frac{\alpha_k}{b_k}\}\}$.

Normalmente, a condição 2 não é satisfeita. O usuário precisa definir o valor abaixo do qual o ponto é considerado estacionário $\|P_{\perp}(m_k - \alpha_k \nabla E(m_k)) - m_k\| \leq diff$.

3 Metodologia

Visando comparar os métodos de otimização da FWI NSPG e L-BFGS foi utilizado o software The Julia Devito Inversion framework (JUDI) para realização de experimentos que pudessem comparar a eficiente da solução e custo computacional da FWI. JUDI é um framework para modelagem e inversão sísmica de larga escala, desenhado para viabilizar traduções rápidas de algoritmos para códigos eficientes e rápidos, dimensionados para problemas de porte 3D demandados pela indústria. O foco do pacote reside na modelagem sísmica bem como na otimização de equações diferenciais parciais restritas, tal como inversão da forma da onda completa (FWI) e imageamento Least-Square Reverse Time Migration (LS-RTM). No JUDI a equação da onda é resolvida com Devito (LOUBOUTIN et al., 2019), uma linguagem Python de domínio específico para cálculos automatizados de diferenças finitas (LUPORINI et al., 2018).

Para os experimentos foi utilizado o modelo sintético Overthrust SEG/EAGE (AMINZADEH, F., J. BRAC, 1996)(Figura 7), o qual servirá de base para comparação de qualidade dos modelos de campo de velocidade gerados pelos diferentes métodos de solução da FWI usados nesta dissertação. O modelo possui quatrocentos pontos no eixo das abscissas espaçados de 25 metros e cento e vinte pontos no eixo das ordenadas também espaçados de 25 metros com malha regular.

A partir deste modelo foram geradas famílias de tiros com dados amostrados em 4ms, arranjo *split-spread* com cento e oitenta receptores espaçados de vinte e cinco metros e dois segundos de dados registrados (*Figura 8*). Estes dados também serviram de referência para comparação da qualidade dos resultados.

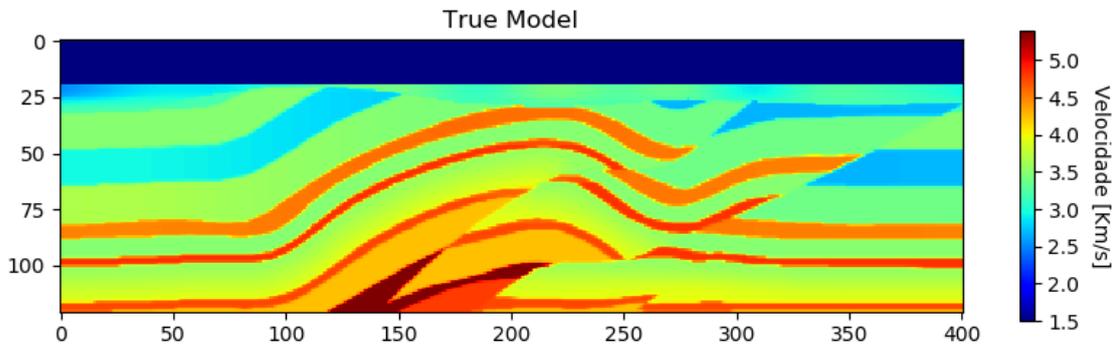


Figura 7: Campo de velocidade overthrust da SEG/EAGE.

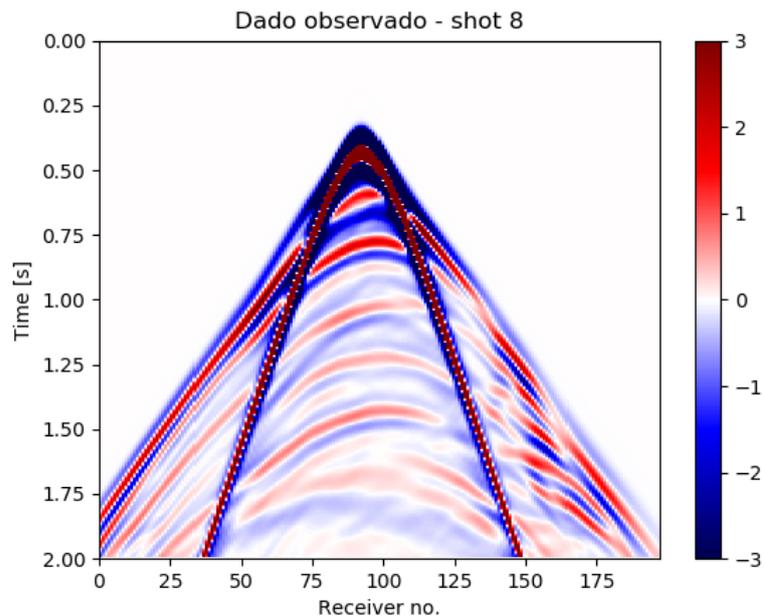


Figura 8: Tiro obtido a partir da modelagem utilizando o campo de velocidade overthrust da SEG/EAGE.

Para realizar os experimentos, o *framework* JUDI foi implementado em computador com ambiente Debian, processador Intel(R) Core (TM) i5-2430M CPU @ 2.40GHz, dois núcleos de processamento, memória (RAM) 4,00 GB e sistema operacional de 64 bits.

O *framework* JUDI utiliza linguagem de programação auxiliar Julia (BEZANSON *et al.*, 2017), uma linguagem de programação dinâmica de alto nível projetada para atender os requisitos da computação de alto desempenho numérico e científico, sendo também eficaz para a programação de propósito geral. O compilador desta linguagem foi instalado para possibilitar o uso do JUDI, assim como o pacote Devito já mencionado.

Após instalação das bibliotecas e compiladores necessários para implementação do JUDI, testes com os diferentes métodos de resolução da FWI, tais como SPG, L-BFGS e SGD, foram executados. Os *scripts* utilizados para execução encontram-se no anexo A. O Devito possui programação multi-processo de memória compartilhada (via OpenMP), devido a esta característica foi possível submeter os processos de forma compartilhada entre os processadores do sistema.

Foram realizadas três configurações para execução dos testes: 1) Utilizando campo de velocidade inicial suavizado pela média dos pontos vizinhos (*Figura 9*), e o dado observado sem ruído (*Figura 7*); 2) Utilizando o campo de velocidade constante (3500 km/s) (*Figura 10*) e o dado observado sem ruído; 3) Utilizando campo de velocidade inicial suavizado pela média dos pontos, e o dado observado com ruído (*Figura 11*). Foi adicionado ruído gaussiano ao dado observado limitando a adição do ruído aos dados com frequências definidas pelo polígono de frequência 20,30,40 e 50 e amplitudes de 0,1,1 e 0.

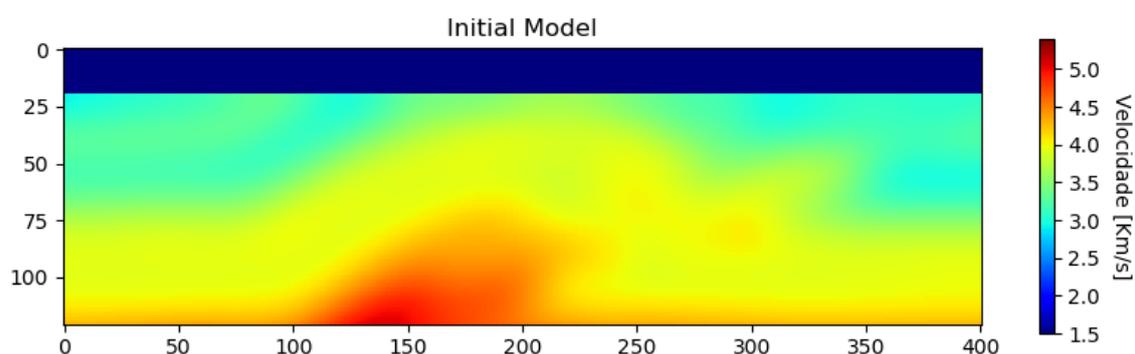


Figura 9: Modelo inicial utilizado nos experimentos com FWI.

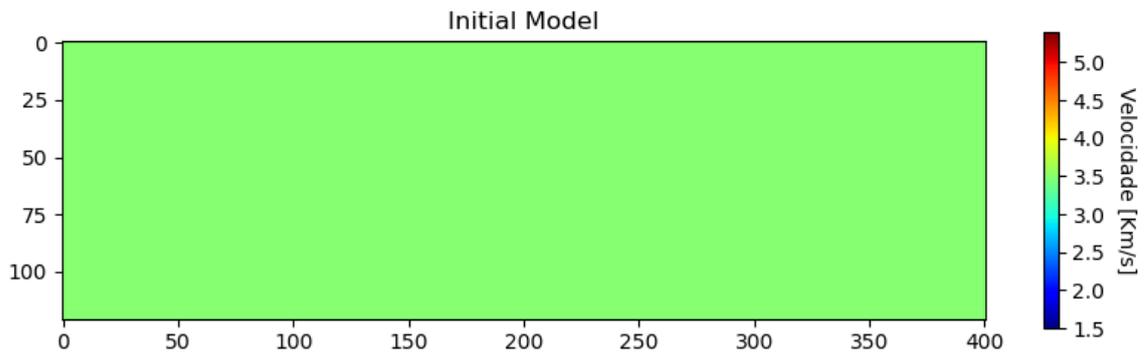


Figura 10: Modelo inicial utilizado nos experimentos com velocidade constante de 3500 km/s.

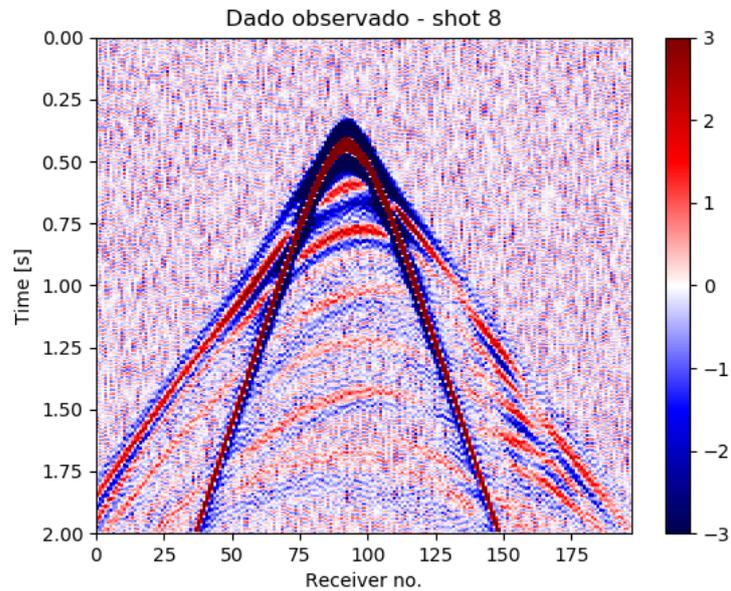


Figura 11: Tiro com ruído adicionado ao dado. Razão sinal/ruído de 70%.

Para comparar os resultados obtidos com os métodos de solução da FWI, foram utilizadas as famílias de tiros e campo de velocidade do dado modelado, que foram comparados com os dados preditos pelo FWI. A eficiência entre os métodos também foi comparada utilizando o tempo de execução dos métodos NSPG, L-BFGS e o gradiente descendente (SGD). O valor do funcional de erro destes métodos também foram comparados. Os resultados obtidos serão apresentados na próxima sessão.

4 Resultados

Os campos de velocidade inicial suavizado, verdadeiro e final para os métodos SPG, SGD e L-BFGS com dados sem ruído são apresentados nas *Figura 12*, *Figura 13*, *Figura 14*, *Figura 15* e *Figura 16*, respectivamente. O campo de velocidade obtido com o método L-BFGS é o que mais se aproxima do modelo real. O resultado final com os métodos NSPG e SGD apresentaram resultados parecidos.

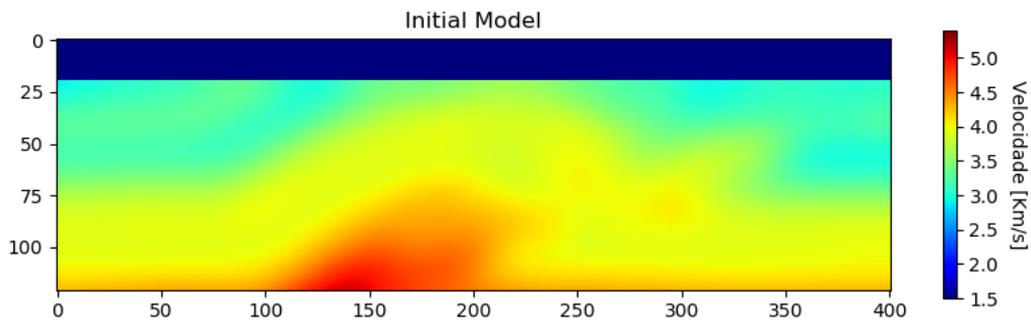


Figura 12: Campo de velocidade inicial suavizado para o teste com velocidade inicial suavizada e dados observados sem ruído.

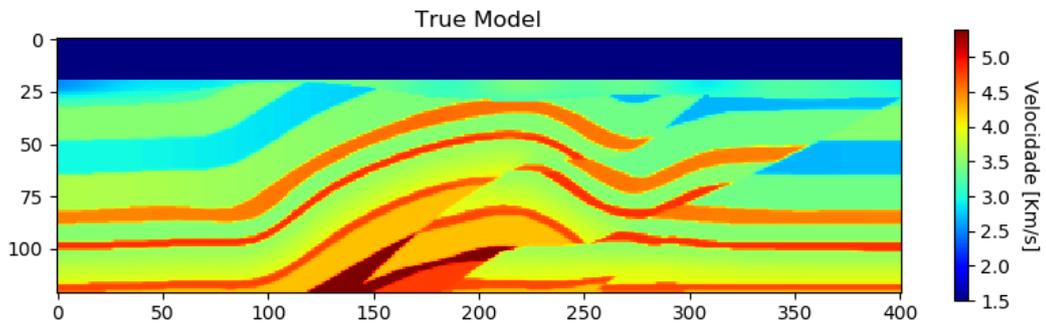


Figura 13: Campo de velocidade verdadeiro.

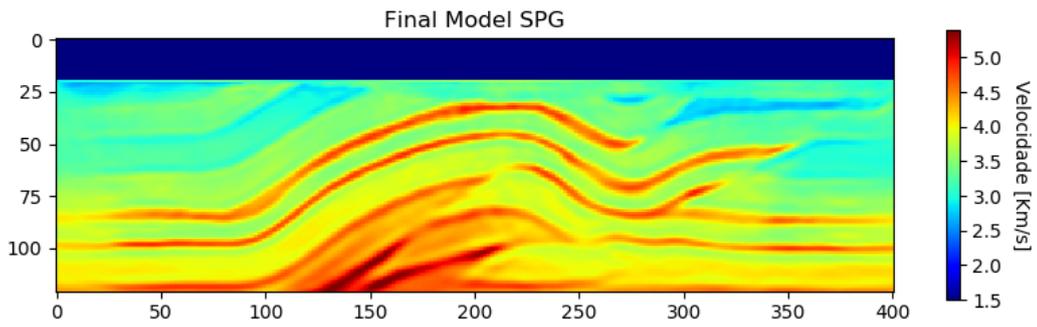


Figura 14: Campo de velocidade final obtido com o método SPG para o teste com velocidade inicial suavizada e dados observados sem ruído.

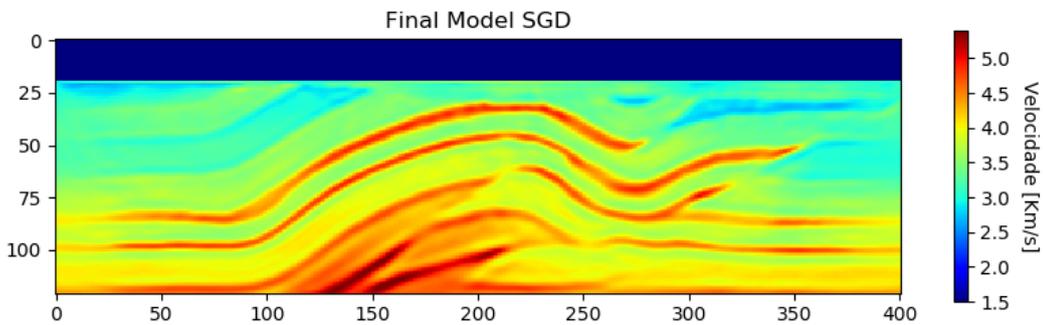


Figura 15: Campo de velocidade final obtido com o método SGD para o teste com velocidade inicial suavizada e dados observados sem ruído.

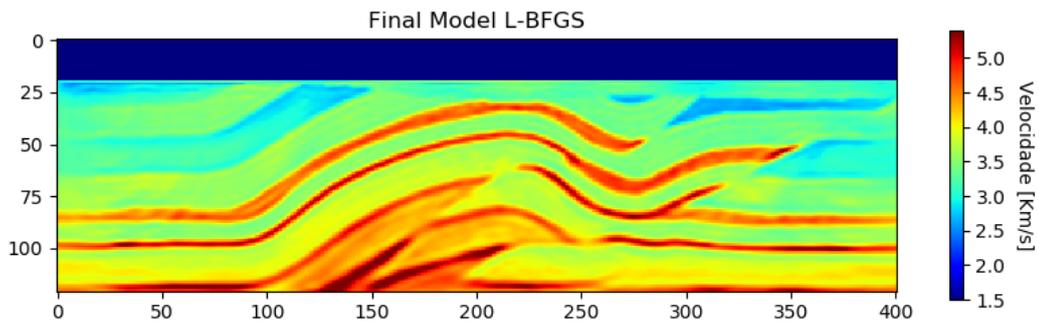


Figura 16: Campo de velocidade final obtido com o método L-BFGS para o teste com velocidade inicial suavizada e dados observados sem ruído.

Os campos de velocidade inicial constante e finais para os métodos SGD, SPG e L-BFGS com dados sem ruído são apresentados nas *Figura 17*, *Figura 18*, *Figura 19*, *Figura 20*, respectivamente. O campo de velocidade não convergiu para

o modelo real nos três casos estudados. Os resultados finais foram completamente diferentes para os três métodos de otimização.

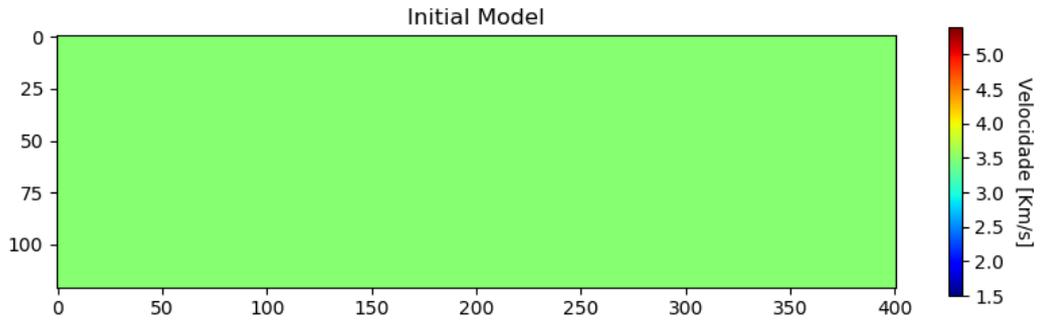


Figura 17: Campo de velocidade inicial com velocidade constante para o teste com velocidade inicial constante e dados observados sem ruído.

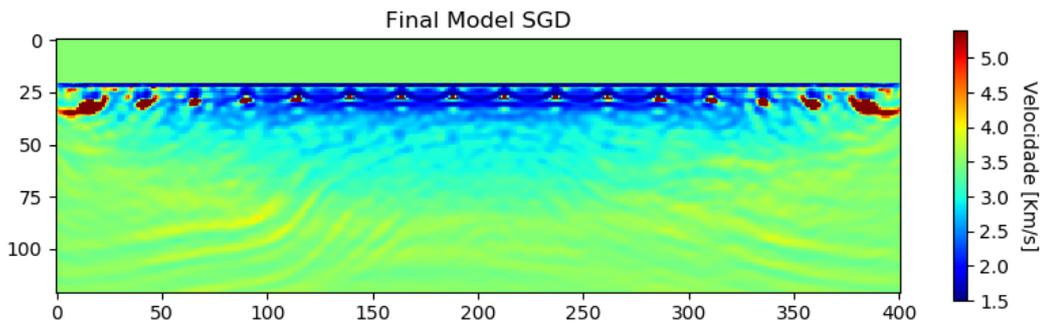


Figura 18: Campo de velocidade final obtido com o método SGD para o teste com velocidade inicial constante e dados observados sem ruído.

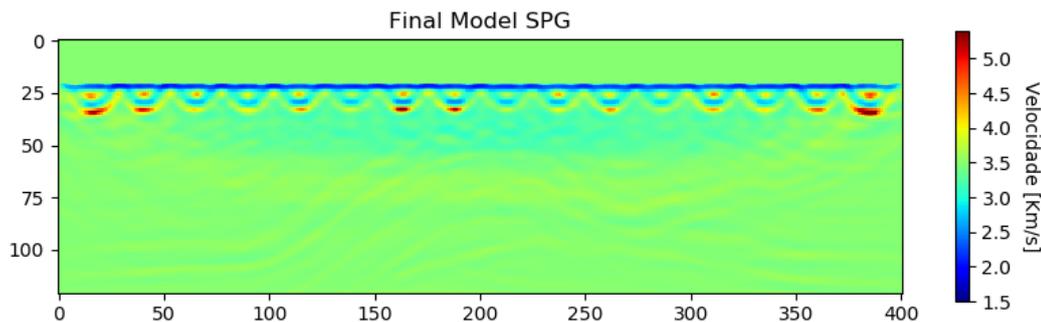


Figura 19: Campo de velocidade final obtido com o método NSPG para o teste com velocidade inicial constante e dados observados sem ruído.

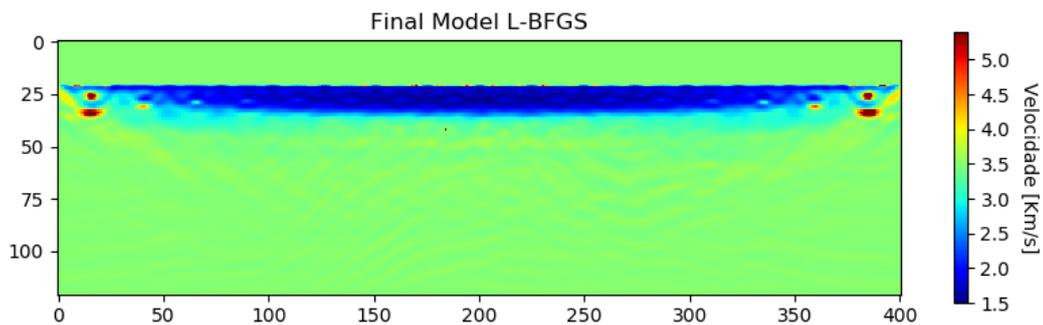


Figura 20: Campo de velocidade final obtido com o método L-BFGS para o teste com velocidade inicial constante e dados observados sem ruído.

Os campos de velocidade inicial suavizada e finais para os métodos SGD, SPG e L-BFGS com dados com ruído são apresentados nas Figura 21, Figura 22, Figura 23 e Figura 24, respectivamente. O Método L-BFGS apresentou o melhor resultado final, com o campo de velocidade mais próximo do modelo real. Os resultados com os métodos SGD e NSPG apresentaram resultados similares.

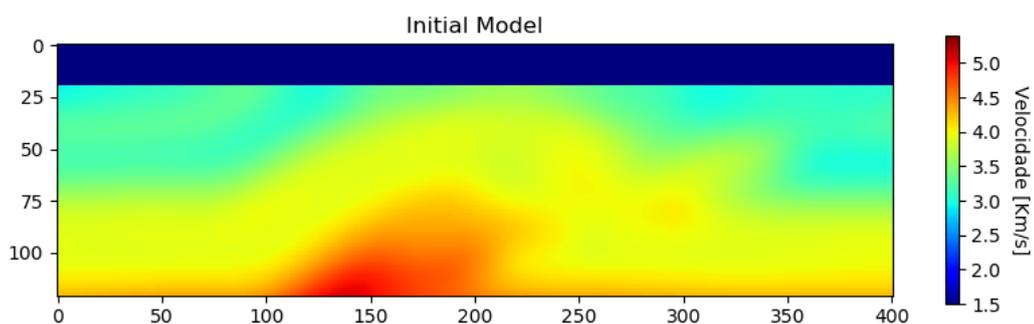


Figura 21: Campo de velocidade inicial suavizado para o teste com velocidade inicial suavizada e dados observados com ruído.

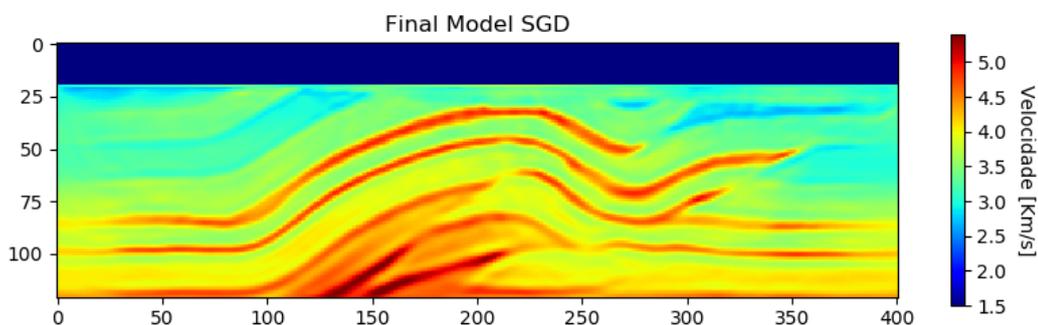


Figura 22: Campo de velocidade final obtido com o método SGD para o teste com velocidade inicial suavizada e dados observados com ruído.

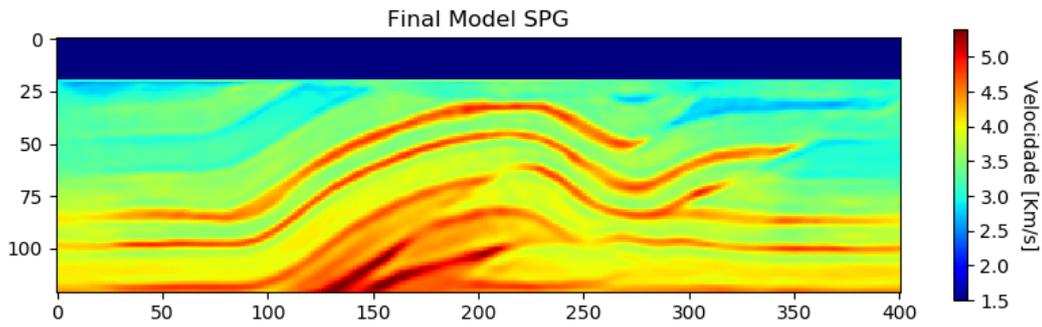


Figura 23: Campo de velocidade final obtido com o método NSPG para o teste com velocidade inicial suavizada e dados observados com ruído.

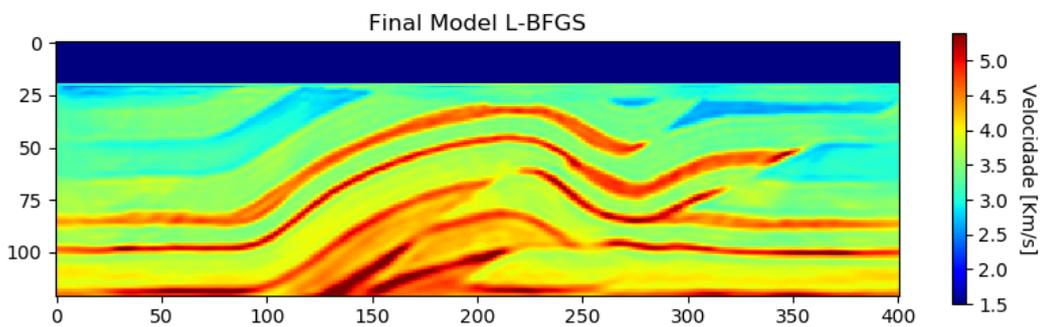


Figura 24: Campo de velocidade final obtido com o método L-BFGS para o teste com velocidade inicial suavizada e dados observados com ruído.

A *figura 25* apresentam os dados modelados utilizando as velocidades obtidas pela FWI com os métodos SGD, NSPG e L-BFGS, respectivamente, para o experimento com velocidade inicial suavizada e dados observados sem ruído. A *Figura 26* apresenta a diferença entre os dados da *figura 25* e o modelado com o campo de velocidade real.

A *figura 27* apresentam os dados modelados utilizando as velocidades obtidas pela FWI com os métodos SGD, NSPG e L-BFGS, respectivamente, para o experimento com velocidade inicial constante e dados observados sem ruído. A *Figura 28* apresenta a diferença entre os dados da *figura 27* e o modelado com o campo de velocidade real.

A *figura 29* apresentam os dados modelados utilizando as velocidades obtidas pela FWI com os métodos SGD, NSPG e L-BFGS, respectivamente, para o experimento com velocidade inicial suavizada e dados observados com ruído. A

Figura 30 apresenta a diferença entre os dados da figura 29 e o modelado com o campo de velocidade real.

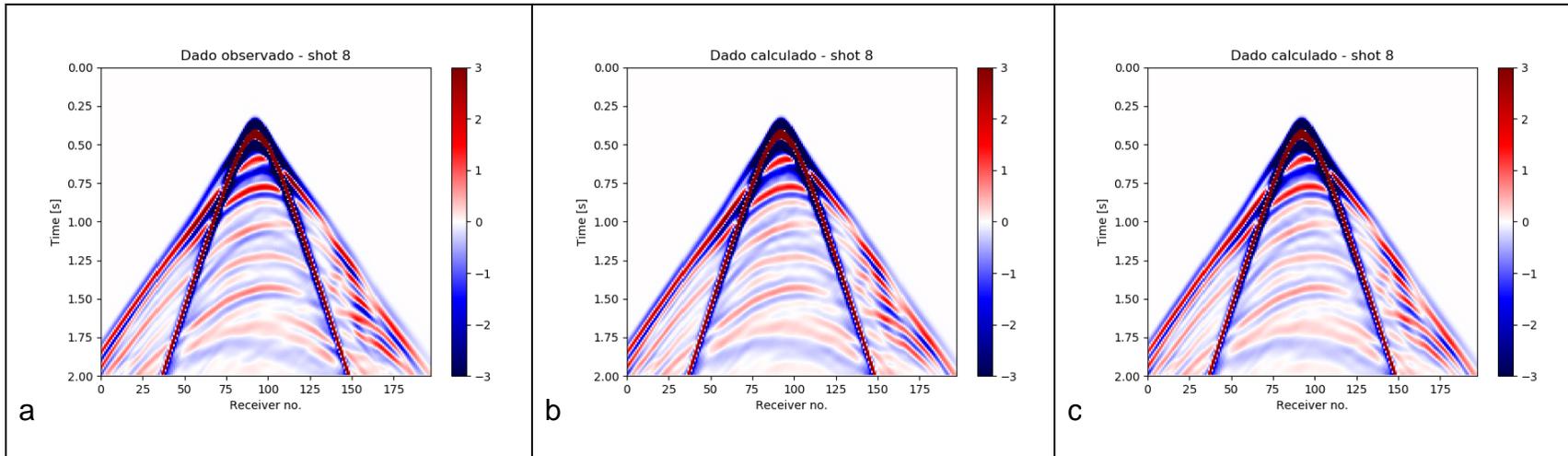


Figura 25: Dado modelado utilizando as velocidades obtidas do experimento 1. a) SGD; b) NSPG e c) L-BFGS.

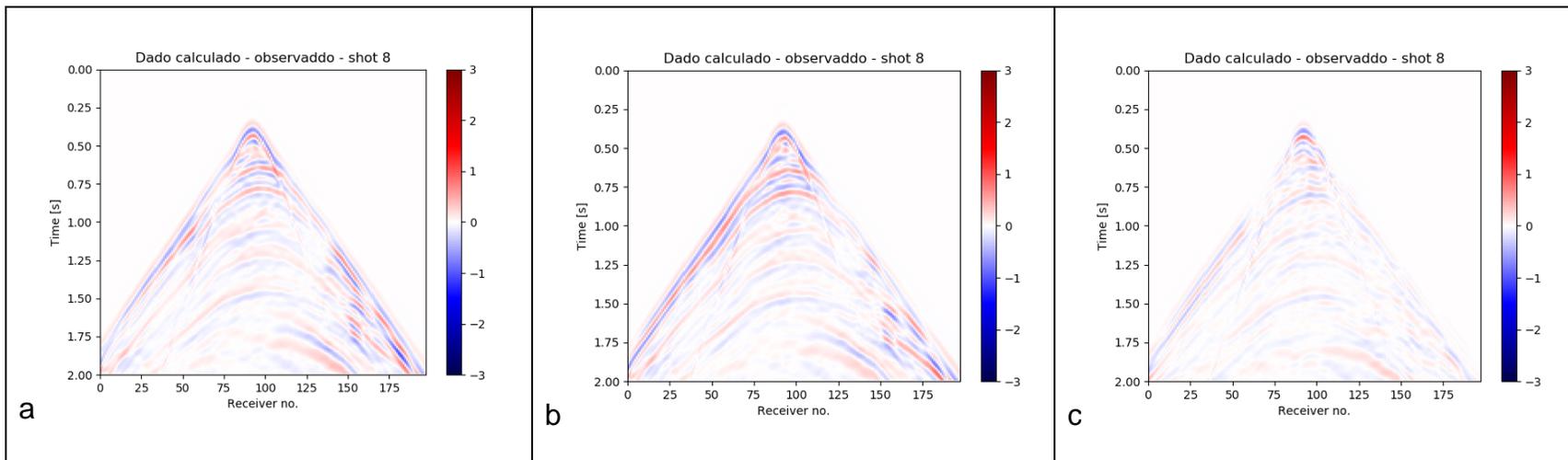


Figura 26: Diferença entre os dados modelados da figura 25 e o dado modelado com a velocidade real para os métodos a) SGD, b) NSPG e c) L-BFGS.

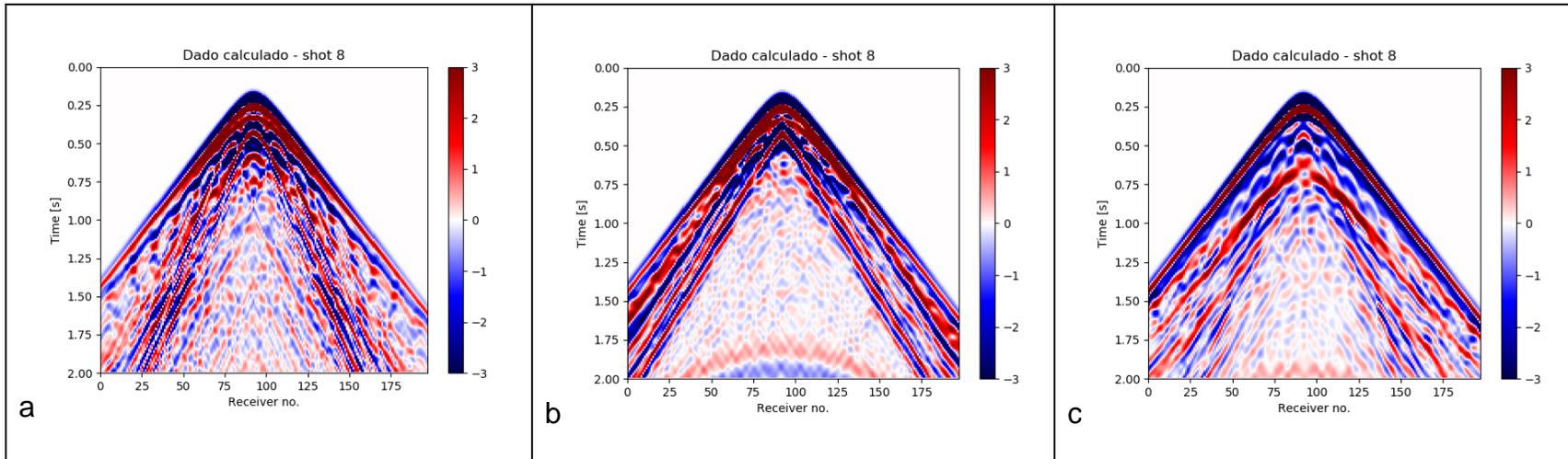


Figura 27: Dado modelado utilizando as velocidades obtidas do experimento 2. a) SGD; b) NSPG e c) L-BFGS.

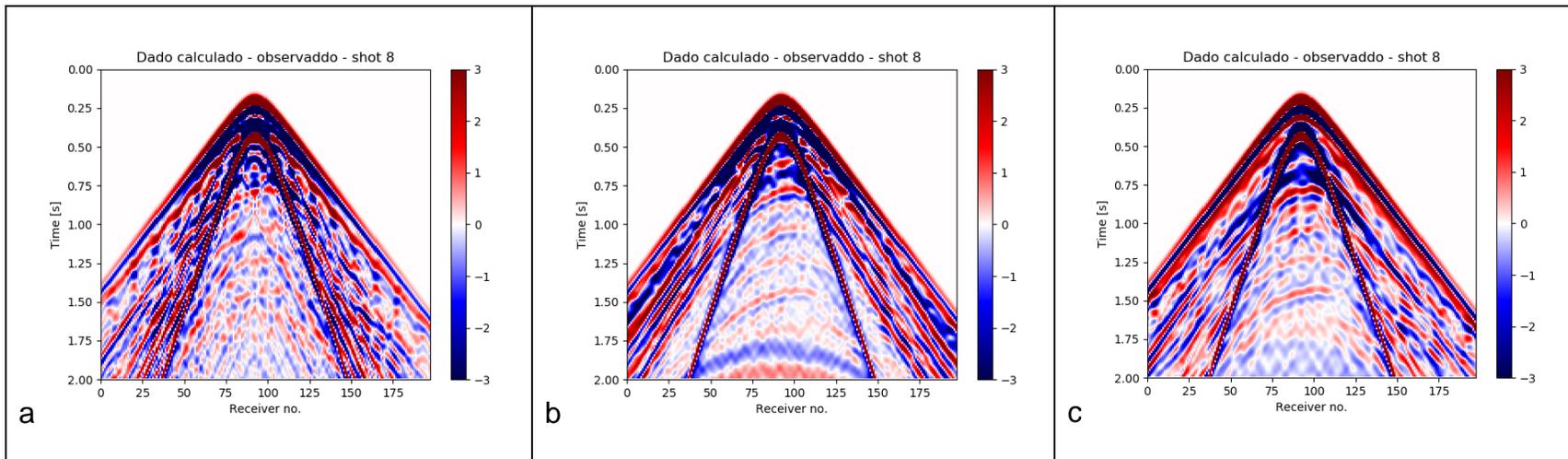


Figura 28: Diferença entre os dados modelados da figura 27 e o dado modelado com a velocidade real para os métodos a) SGD, b) NSPG e c) L-BFGS.

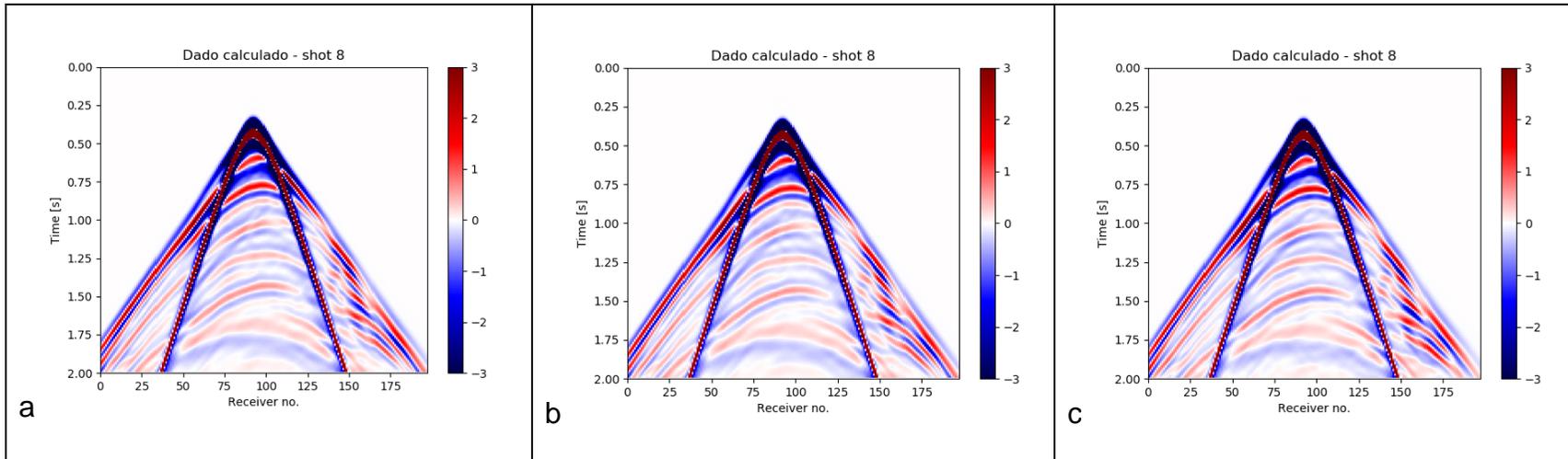


Figura 29: Dado modelado utilizando as velocidades obtidas do experimento 3. a) SGD; b) NSPG e c) L-BFGS.

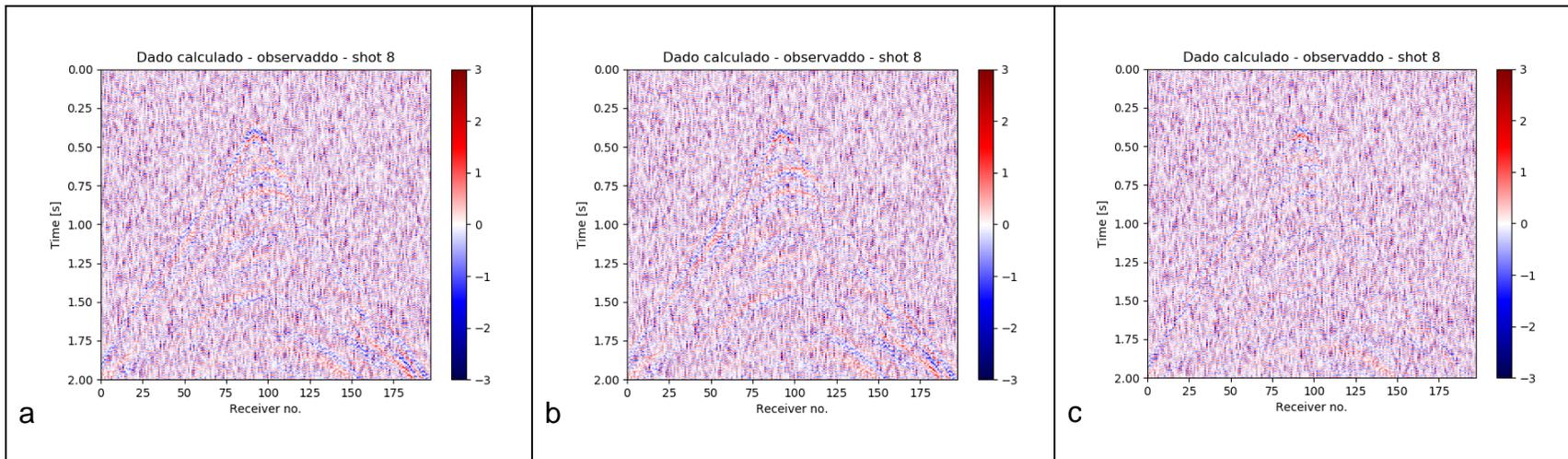


Figura 30: Diferença entre os dados modelados da figura 29 e o dado modelado com a velocidade real para os métodos a) SGD, b) NSPG e c) L-BFGS.

A nível de comparação dos métodos, foi plotado o gráfico do valor da função objetivo para cada iteração do modelo para os métodos NSPG, L-BFGS e SGD nas três configurações dos testes. Os resultados são apresentados na *Figura 31*, *Figura 32* e *Figura 33* para os casos 1, 2 e 3 respectivamente. No caso 1 (*Figura 31*), apesar do método NSPG ter o melhor chute inicial, obteve o segundo melhor resultado. No caso 2 (*Figura 32*) os métodos SGD e NSPG apresentaram resultados similares durante todas as iterações e de melhor qualidade quando comparado ao método L-BFG, no entanto o melhor resultado final ficou sendo o do método L-BFGS. No caso 2 (*Figura 33*) o método NSPG apresentou o melhor resultado dos três métodos, no entanto não convergiu para o modelo real de velocidade.

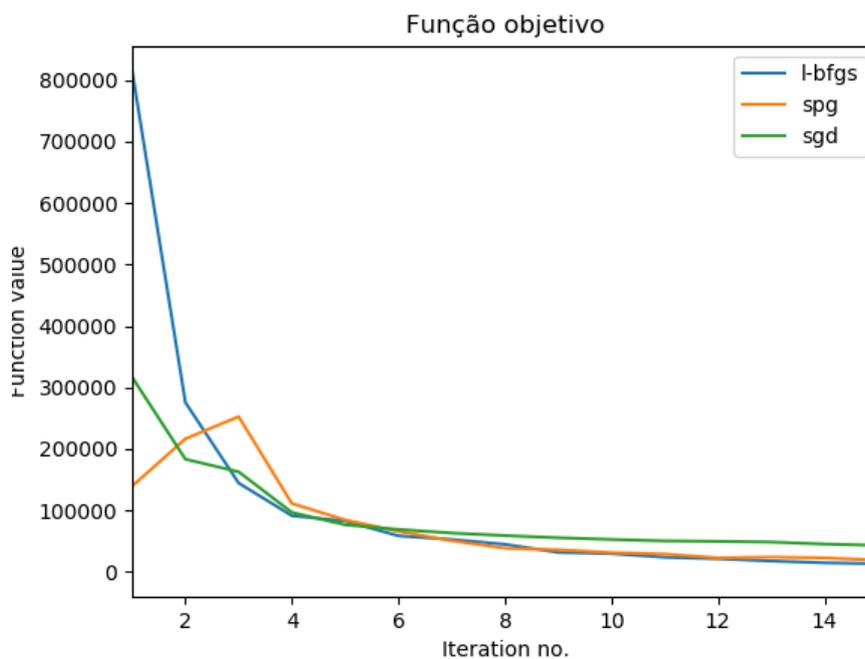


Figura 31: Função objetivo para os métodos SPG, SGD e L-BFGS para o caso de velocidade inicial suavizada e dados sem ruído.

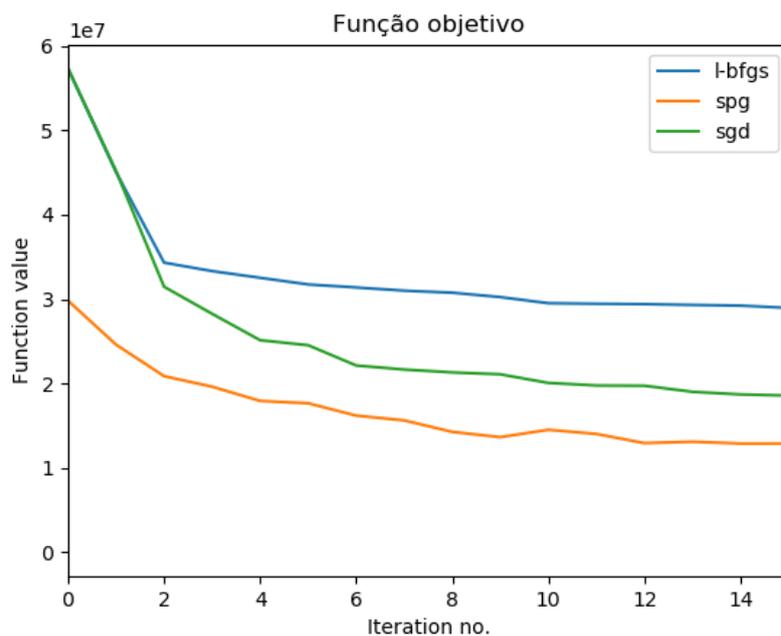


Figura 32: Função objetivo para os métodos SPG, SGD e L-BFGS para o caso de velocidade inicial constante e dados sem ruído.

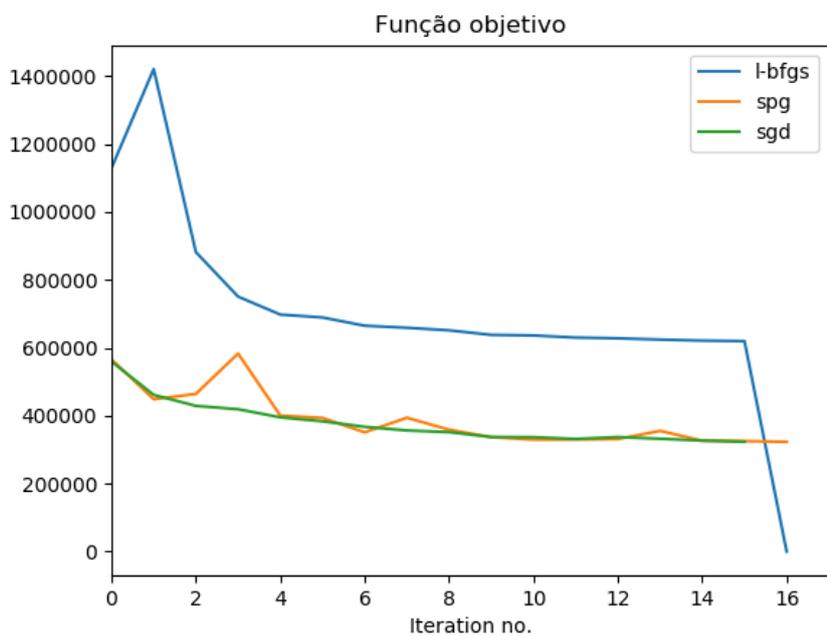


Figura 33: Função objetivo para os métodos SPG, SGD e L-BFGS para o caso de velocidade inicial suavizada e dados com ruído.

Para comparar o custo computacional dos métodos foi elaborada a Tabela 1. Com a memória alocada e tempo de execução para cada método de otimização do FWI para as três configurações de experimentos.

Tabela 1: Desempenho computacional dos métodos de resolução da FWI.

	Método	Memória alocada	Tempo (segundos)
Vel. suavizada e dados sem ruído.	L-BFGS	6.999 GiB	1516
	NSPG	6.121 GiB	697
	SGD	45.498 GiB	4043
Vel. suavizada e dados com ruído.	L-BFGS	6.998 GiB	3067
	SPG	6.589 GiB	1782
	SGD	78.479 GiB	7948
Vel. constante e dados sem ruído.	L-BFGS	6.567 GiB	1755
	NSPG	5.867 GiB	754
	SGD	43.124 GiB	3665

5 Conclusão

Esta dissertação fez uma análise do problema de FWI abordando os tópicos utilizados para otimização do problema de inversão da forma completa da onda.

Os resultados experimentais apresentaram fatos conclusivos sobre a eficácia do método NSPG comparado ao método L-BFGS e SGD. O método NSPG possui o menor custo computacional dos três métodos de otimização nos três experimentos realizados, comparando o tempo de execução e memória alocada do métodos de otimização. Nos experimentos com o campo de velocidade suavizado e dados observados sem ruído, o método NSPG apresentou menor custo computacional, mas obteve o segundo melhor resultado, próximo do resultado alcançado pelo método L-BFGS.

Utilizando o campo de velocidade inicial constante e dados observados sem ruído, nenhum dos três métodos convergiu para um modelo satisfatório, apesar do método NSPG possuir o menor valor da função objetivo.

Para o experimento utilizando campo de velocidade inicial suavizado e simulação de dados observados com ruído, o resultados do método NSPG esteve

muito próximo do método SGD no decorrer das iterações, contudo o método L-BFGS obteve o melhor campo de velocidade final.

Os experimentos mostraram que o método NSPG possui o menor valor da função objetivo para primeira iteração dentre os métodos de otimização analisados.

Nos experimentos foram utilizados dados sintéticos para conclusão dos resultados, novos experimentos com dados reais precisam ser realizados a nível de comparação dos métodos para obter um resultado mais conclusivo.

6 Apêndice (Scripts)

Método SGD

```
using Pkg; Pkg.activate("JUDI")

using Statistics, Random, LinearAlgebra

using JUDI.TimeModeling, JUDI.SLIM_optim, HDF5, SeisIO, PyPlot

# Load starting model

n,d,o,m0 = read(h5open("../data/overthrust_model.h5","r"), "n", "d", "o", "m0")

model2 = Model((n[1],n[2]), (d[1],d[2]), (o[1],o[2]), m0)

# Bound constraints

v0 = sqrt.(1f0 ./ model2.m)

vmin = ones(Float32,model2.n) .* 1.3f0

vmax = ones(Float32,model2.n) .* 6.5f0

vmin[:,1:21] .= v0[:,1:21] # keep water column fixed

vmax[:,1:21] .= v0[:,1:21]

# Slowness squared [s^2/km^2]

mmin = vec((1f0 ./ vmax).^2)
```

```

mmax = vec((1f0 ./ vmin).^2)

# Load data
block = segy_read("../data/overthrust_shot_records.segy")
d_obs = judiVector(block)

# Set up wavelet
src_geometry = Geometry(block; key="source")
wavelet = ricker_wavelet(src_geometry.t[1],src_geometry.dt[1],0.008f0) # 8 Hz wavelet
q = judiVector(src_geometry,wavelet)

##### FWI
#####

# Optimization parameters
niterations = 16
batchsize = 16
fhistory_SGD = zeros(Float32,niterations)

# Projection operator for bound constraints
proj(x) = reshape(median([vec(mmin) vec(x) vec(mmax)]; dims=2),model2.n)

# Main loop
@time begin
for j=1:niterations

# get fwi objective function value and gradient

```

```

i = randperm(d_obs.nsrc)[1:batchsize]
fval2, gradient = fwi_objective(model2,q[i],d_obs[i])
println("FWI iteration no: ",j,"; function value: ",fval2)
fhistory_SGD[j] = fval2

# linesearch
step = backtracking_linesearch(model2, q[i], d_obs[i], fval2, gradient, proj; alpha=1f0)

# Update model and bound projection
model2.m = proj(model2.m + reshape(step,model2.n))
end
end

#figure(); imshow(sqrt.(1f0./adjoint(model2.m))); title("FWI with SGD")

```

Método L-BFGS

```

using Pkg; Pkg.activate("JUDI")
using Statistics, Random, LinearAlgebra
using JUDI.TimeModeling, HDF5, NLOpt, SeisIO

# Load starting model
n,d,o,m0 = read(h5open("../data/overthrust_model.h5","r"), "n", "d", "o", "m0")
model1 = Model((n[1],n[2]), (d[1],d[2]), (o[1],o[2]), m0)

# Bound constraints
v0 = sqrt.(1f0 ./ model1.m)
vmin = ones(Float32, model1.n) .* 1.3f0

```

```

vmax = ones(Float32, model1.n) .* 6.5f0

# Slowness squared [s^2/km^2]
mmin = vec((1f0 ./ vmax).^2)
mmax = vec((1f0 ./ vmin).^2)

# Load data
block = segy_read("../data/overthrust_shot_records.segy")
d_obs = judiVector(block)

# Set up wavelet
src_geometry = Geometry(block; key="source")
wavelet = ricker_wavelet(src_geometry.t[1], src_geometry.dt[1], 0.008f0) # 8 Hz
wavelet
q = judiVector(src_geometry, wavelet)

##### FWI
#####

# optimization parameters
batchsize = 16
count = 0
fvals_lbfgs = zeros(17)

# NLOpt objective function
println("No. ", "fval1", " ", "norm(gradient)")
function f!(x,grad)

```

```

# Update model
model1.m = convert(Array{Float32, 2}, reshape(x, model1.n))

# Select batch and calculate gradient
i = randperm(d_obs.nsrc)[1:batchsize]
fval1, gradient = fwi_objective(model1, q[i], d_obs[i])

# Reset gradient in water column to zero
gradient = reshape(gradient, model1.n); gradient[:, 1:21] .= 0f0
grad[1:end] = vec(gradient)
global count; count += 1; fvals_lbfgs[count] = fval1
println(count, " ", fval1, " ", norm(grad))
return convert(Float64, fval1)
end

# Optimization parameters
opt = Opt{:LD_LBFGS, prod(model1.n)}
lower_bounds!(opt, mmin); upper_bounds!(opt, mmax)
min_objective!(opt, f!)
maxeval!(opt, 16)
@time begin
(minf, minx, ret) = optimize(opt, vec(model1.m))
end

Método NSPG
using Pkg; Pkg.activate("JUDI")
using Statistics, Random, LinearAlgebra
using JUDI.TimeModeling, JUDI.SLIM_optim, HDF5, SeisIO

```

```

# Load starting model
n,d,o,m0 = read(h5open("../data/overthrust_model.h5","r"), "n", "d", "o", "m0")
model0 = Model((n[1],n[2]), (d[1],d[2]), (o[1],o[2]), m0)

# Bound constraints
v0 = sqrt.(1f0 ./ model0.m)
vmin = ones(Float32,model0.n) .* 1.3f0
vmax = ones(Float32,model0.n) .* 6.5f0
vmin[:,1:21] .= v0[:,1:21] # keep water column fixed
vmax[:,1:21] .= v0[:,1:21]

# Slowness squared [s^2/km^2]
mmin = vec((1f0 ./ vmax).^2)
mmax = vec((1f0 ./ vmin).^2)

# Load data
block = segy_read("../data/overthrust_shot_records.segy")
d_obs = judiVector(block)

# Set up wavelet
src_geometry = Geometry(block; key="source")
wavelet = ricker_wavelet(src_geometry.t[1],src_geometry.dt[1],0.008f0) # 8 Hz wavelet
q = judiVector(src_geometry,wavelet)

```

```

#####
#####

```

FWI

```

# Optimization parameters

#srand(1) # set seed of random number generator

fevals = 16

batchsize = 8

fvals_spg = zeros(17)

# Objective function for minConf library

count = 0

function objective_function(x)

    model0.m = reshape(x,model0.n);

    # fwi function value and gradient

    i = randperm(d_obs.nsrc)[1:batchsize]

    fval, grad = fwi_objective(model0, q[i], d_obs[i])

    grad = .125f0*grad/maximum(abs.(grad)) # scale for line search

    global count; count+= 1; fvals_spg[count] = fval

    return fval, vec(grad)

end

# Bound projection

ProjBound(x) = median([mmin x mmax]; dims=2)

# FWI with SPG

options = spg_options(verbose=3, maxIter=fevals, memory=3)

@time begin

```

```

x, fsave, funEvals= minConf_SPG(objective_function, vec(model0.m), ProjBound,
options)

end

```

7 Bibliografia

AKI, K.; RICHARDS, P. Quantitative Seismology, 2nd Ed. 1 jan. 2002.

AMINZADEH, F., J. BRAC, AND T. K. 3-D Salt and Overthrust Seismic Models. **AAPG Bulletin**, p. 376–391, 1996.

ARAUJO, E. S.; PESTANA, R. C. Análise dos métodos de diferenças finitas e expansão dos raios na migração reversa no tempo. **Revista Brasileira de Geofísica**, v. 28, p. 723–739, 2010.

BABUSKA, V.; CARA, M. **Seismic Anisotropy in the Earth**. [s.l.] Springer Netherlands, 2012.

BAYSAL, E.; KOSLOFF, D. D.; SHERWOOD, J. W. C. Reverse time migration. **Geophysics**, v. 48, n. 11, p. 1514–1524, 1983.

BIRGIN, E. G.; MARTÍNEZ, J. M.; RAYDAN, M. Spectral Projected Gradient Methods: Review and Perspectives. **Journal of Statistical Software**, v. 60, n. 3, 2015.

BUNKS, C. et al. Multiscale seismic waveform inversion. **GEOPHYSICS**, v. 60, n. 5, p. 1457–1473, 1 set. 1995.

BURDEN, R. L.; FAIRES, J. D. **Numerical Analysis**. [s.l.] Cengage Learning, 2010.

DABLAIN, M. A. The application of high-order differencing to the scalar wave equation. **GEOPHYSICS**, v. 51, n. 1, p. 54–66, 1 jan. 1986.

DAHLEN, F. A.; TROMP, J. **Theoretical Global Seismology**. [s.l.] Princeton University Press, 1998.

DOS SANTOS, A. W. G.; PESTANA, R. C. A pseudo-analytical method to solve the acoustic wave equation for modeling and RTM. In: **SEG Technical Program Expanded Abstracts 2012**. SEG Technical Program Expanded Abstracts. [s.l.] Society of Exploration Geophysicists, 2012. p. 1–5.

FABIEN-OUELLET, G.; GLOAGUEN, E.; GIROUX, B. A stochastic L-BFGS approach for full-waveform inversion. In: **SEG Technical Program Expanded Abstracts 2017**. SEG Technical Program Expanded Abstracts. [s.l.] Society of Exploration Geophysicists, 2017. p. 1622–1627.

GISOLF, D.; VERSCHUUR, E. **The principles of quantitative acoustical imaging**.

[s.l.] EAGE Publ., 2010.

GRIPPO, L.; LAMPARIELLO, F.; LUCIDI, S. A Nonmonotone Line Search Technique for Newton's Method. **Siam Journal on Numerical Analysis - SIAM J NUMER ANAL**, v. 23, p. 707–716, 1 ago. 1986.

KENNETT, B. **The Seismic Wavefield Volume I: Introduction and Theoretical Development**. [s.l.: s.n.], v. 83

KLIMENTOS, T. The effects of porosity-permeability-clay content on the velocity of compressional waves. **GEOPHYSICS**, v. 56, n. 12, p. 1930–1939, 1 dez. 1991.

LEI, T.; BOHLEN, T.; ZHANG, W. Full-waveform inversion using truncated Newton optimization. In: **SEG Technical Program Expanded Abstracts 2018**. SEG Technical Program Expanded Abstracts. [s.l.] Society of Exploration Geophysicists, 2018. p. 1384–1388.

LINES, L. R.; SLAWINSKI, R.; BORDING, R. P. A recipe for stability of finite-difference wave-equation computations. **GEOPHYSICS**, v. 64, n. 3, p. 967–969, 1 maio 1999.

LIU, F. et al. 3-D Time-domain Full Waveform Inversion of a Valhall OBC dataset. In: **SEG Technical Program Expanded Abstracts 2012**. SEG Technical Program Expanded Abstracts. [s.l.] Society of Exploration Geophysicists, 2012. p. 1–5.

LOUBOUTIN, M. et al. Devito (v3.1.0): An embedded domain-specific language for finite differences and geophysical exploration. **Geoscientific Model Development**, v. 12, n. 3, p. 1165–1187, 2019.

LUPORINI, F. et al. Architecture and performance of Devito, a system for automated stencil computation. v. 1, n. 1, 2018.

MCMECHAN, G. A. MIGRATION BY EXTRAPOLATION OF TIME-DEPENDENT BOUNDARY VALUES*. **Geophysical Prospecting**, v. 31, n. 3, p. 413–420, 1983.

NOCEDAL, J. Updating Quasi-Newton Matrices With Limited Storage. **Mathematics of Computation**, v. 35, n. 151, p. 773–782, 1980.

ROBERTSSON, J. O. A.; BLANCH, J. O.; SYMES, W. W. Viscoelastic finite-difference modeling. **GEOPHYSICS**, v. 59, n. 9, p. 1444–1456, 1 set. 1994.

ROY*, A.; CHAZALNOEL, N. RTM technology for improved salt imaging in the Santos Basin, Brazil. In: **12th International Congress of the Brazilian Geophysical Society & EXPOGEF, Rio de Janeiro, Brazil, 15-18 August 2011**. SEG Global Meeting Abstracts. [s.l.] Brazilian Geophysical Society, 2011. p. 1812–1816.

TARANTOLA, A. Inversion of seismic data in acoustic approximation. **Geophysics**, v. 49, p. 1259–1266, 1984a.

TARANTOLA, A. in the acoustic approximation. v. 49, n. 8, p. 1259–1266, 1984b.

VIRIEUX, J.; OPERTO, S. An overview of full-waveform inversion in exploration

geophysics. **Geophysics**, v. 74, n. 6, p. WCC1–WCC26, 2009.

WANG, B. et al. CUDA?based acceleration of full waveform inversion on GPU. In: **SEG Technical Program Expanded Abstracts 2011**. SEG Technical Program Expanded Abstracts. [s.l.] Society of Exploration Geophysicists, 2011. p. 2528–2533.

WHITMORE, N. D. Iterative depth migration by backward time propagation. In: **SEG Technical Program Expanded Abstracts 1983**. SEG Technical Program Expanded Abstracts. [s.l.] Society of Exploration Geophysicists, 1983. p. 382–385.

ZEEV, N.; SAVASTA, O.; CORES, D. Non-monotone spectral projected gradient method applied to full waveform inversion. **Geophysical Prospecting**, v. 54, n. 5, p. 525–534, 2006.