Projeto de Pesquisa de Pós-Doutorado

Inversão de Forma de Onda Completa Anisotrópica Usando Método de Elementos Finitos

Daiane Iglesia Dolci
daia.dolci@gmail.com

Supervisor: Prof. Dr. Bruno Souza Carmo
Universidade de São Paulo
Escola politécnica - Departamento de Engenharia de Engenharia Mecânica

30 de Janeiro de 2026

Conteúdo

1 Introdução

A Inversão de Forma de Onda Completa (FWI, do inglês *Full Waveform Inversion*) é uma ferramenta de alta resolução no imageamento sísmico de subsuperfície, permitindo a predição de modelos de parâmetros acústicos e elásticos através da minimização da desajustes entre dados sísmicos observados e simulados [16,18]. FWI depende intrinsicamente da precisão física da modelagem de ondas e da eficiência dos métodos numéricos empregados. Para detalhes adicionais sobre FWI, o leitor pode consultar os trabalhos de [**?**,13,18] que apresentam revisões sobre o método.

Um dos desafios centrais na aplicação da FWI a dados reais é a correta inversão de modelos anisotrópicos tridimensionais (3D) [19]. A suposição simplificada de um meio acústico ou isotrópico em regiões intrinsecamente anisotrópicas pode resultar em erros de imageamento [1, 2]. Quando na subsuperfície as camadas sedimentares são deformadas por tectonismo regional ou por forças locais (por exemplo, corpos de sal), a rotação correspondente pode inclinar o meio TI (transversalmente isotrópico), formando um meio transversalmente isotrópico inclinado (TTI).

A modelagem de meios TTI anisotrópicos transforma FWI em um problema de inversão de multiparametros, sujeito a fortes interdependências (*trade-offs*) entre os parâmetros do modelo [5]. Esses *trade-offs* ocorrem quando variações em diferentes parâmetros físicos produzem respostas sísmicas similares, gerando ambiguidades (ou *crosstalk*) que dificultam a estimativa única de cada propriedade. Isso torna a inversão significativamente mais desafiadora, exigindo estratégias de parametrização robustas para mitigar tais ambiguidades.

Métodos baseados em Diferenças Finitas (FDM) são amplamente utilizados devido à sua eficiência computacional e facilidade de implementação em domínios simples. No entanto, o FDM apresenta limitações na representação de geometrias complexas, dada a rigidez inerente às malhas estruturadas [8]. Por outro lado, o Método de Elementos Finitos (FEM) oferece maior flexibilidade geométrica em domínios topograficamente complexos [6]. Komatitsch e Vilotte [7] demonstraram a aplicabilidade de elementos espectrais para a propagação de ondas em meios heterogêneos. Mais recentemente, trabalhos como os de Modrak e Tromp [11] e Thrastarson et al. [17] exploraram o uso de FEM de alta ordem para FWI em escalas global e regional. Complementarmente, Keith et al. [15] implementaram solucionadores de elementos finitos de alta ordem triangulares (2D) e tetraédricos (3D) aplicados à FWI acústica. No entanto, a aplicação de solucionadores FEM 3D em escalas industriais impõe demandas severas de custo computacional e gerenciamento de memória.

Neste contexto, este projeto propõe o desenvolvimento de um *framework* computacional em Python para FWI anisotrópico TTI, utilizando a biblioteca *Firedrake* [14]. A abordagem adota a abstração matemática da *Unified Form Language* (UFL) [3] para a definição

simbólica das equações de onda na forma fraca, combinada com a geração automática de código otimizado. O diferencial da proposta reside na integração de Diferenciação Algorítmica (AD) via `pyadjoint` [10] para o cálculo automatizado de gradientes, aliado a estratégias ótimas de *checkpointing*, paralelismo de fontes de onda e inversão multiparamétrica.

2 Justificativa

O desenvolvimento de um solucionador baseado no Método de Elementos Finitos (FEM) de alta ordem, dotado de diferenciação automática e gestão eficiente de memória, é estratégico para o avanço do FWI em meios complexos. Embora ferramentas de FEM ofereçam precisão superior em malhas não estruturadas, a complexidade de implementação de códigos adjuntos (necessários para o cálculo de gradientes) muitas vezes limita a pesquisa de novas físicas, como a anisotropia TTI.

A utilização de AD [9] para automatizar a obtenção das equações adjuntas elimina erros de derivação manual e acelera o desenvolvimento. Contudo, o uso de AD em problemas de grande porte (3D) apresenta desafios significativos de desempenho e consumo de memória. Investigar o equilíbrio entre a sobrecarga da AD e a eficiência computacional — comparando-a com abordagens manuais — é uma contribuição fundamental deste projeto.

Além disso, a natureza multiparamétrica do FWI em meios TTI exige uma investigação rigorosa sobre os *trade-offs* entre parâmetros. Um *framework* flexível e extensível permitirá testar e integrar com AD as diferentes estratégias de parametrização e regularização para mitigar ambiguidades, algo difícil de realizar em códigos legados fechados. Portanto, a ferramenta proposta não apenas viabilizará simulações precisas, mas servirá como plataforma para o desenvolvimento metodológico em inversão sísmica.

3 Objetivos

O objetivo geral é implementar e verificar um *framework* computacional para FWI TTI anisotrópico baseado em elementos finitos de alta ordem, integrando diferenciação algorítmica para o cálculo automatizado de gradientes e paralelismo *ensemble* para o processamento eficiente de múltiplas fontes. Adicionalmente, o projeto avaliará a inversão multiparamétrica, desde modelos sintéticos simples até modelos de referência industriais 3D, implementando uma arquitetura flexível com DA e estratégias para mitigar o *trade-off* entre os parâmetros físicos.

4 Metodologia

4.1 Equação da Onda em Meios Anisotrópicos (TTI)

Assumindo um sistema de referência inercial, com base na segunda lei de Newton e na lei de Hooke generalizada, a equação da onda elástica anisotrópica de segunda ordem, no domínio do tempo e espaço, 3D, para um meio heterogêneo não atenuante pode ser escrita como:

$$\begin{cases} \rho(\mathbf{x},t)\,\ddot{v}_i(\mathbf{x},t) = \sigma_{ij,j}(\mathbf{x},t) + f_i(\mathbf{x},t), \\ \qquad \sigma_{ij}(\mathbf{x},t) = C_{ijkl}(\mathbf{x})\,v_{k,l}(\mathbf{x},t) + \dot{m}_{ij}(\mathbf{x},t). \end{cases} \tag{1.1}$$

onde utiliza-se a convenção de soma de Einstein para índices repetidos. Os valores possíveis para cada subíndice i, j, k e l são $\{x, y, z\}$, sendo z o eixo vertical. t representa o tempo e $\mathbf{x}$ o vetor posição espacial. A notação $_{,j}$ indica diferenciação espacial parcial em relação a j e $\dot{}$ indica derivada parcial temporal de primeira ordem. v_i é a i-ésima componente do campo de velocidade (unidade de comprimento por unidade de tempo) e σ_{ij} é a componente ij do tensor de tensão (força por unidade de superfície). ρ é a densidade (massa por unidade de volume) e C_{ijkl} são os componentes do tensor de elasticidade de quarta ordem (i.e. módulo elástico ou rigidez) introduzidos na lei de Hooke generalizada. Junto com a densidade, o módulo elástico descreve as propriedades mecânicas da subsuperfície. Note que o uso da lei de Hooke implica elasticidade linear, experimentalmente válida para pequenas deformações.

Os componentes do tensor de elasticidade são positivos e exibem simetrias ($C_{jikl} = C_{ijkl} = C_{ijlk} = C_{jilk}$) (ver, por exemplo, Aki e Richards, 2002) e podem ser representados de forma mais compacta por uma matriz 6×6 usando a notação de Voigt, reagrupando o primeiro e segundo pares de índices conforme explicado na Tabela 1. f_i e m_{ij} são os componentes de forças externas. f_i é a i-ésima componente de uma fonte de força de corpo externa (força por unidade de volume). m_{ij} são os componentes de uma fonte de densidade de momento externa (força por unidade de superfície, i.e. tensão). Note que, matematicamente, uma fonte de densidade de momento pode ser substituída por uma força de corpo.

Tabela 1: Mapeamento de índices tensoriais para notação de Voigt.

Índices Tensoriais (i, j) ou (k, l)	**Índices Cartesianos** (x, y, z)	**Índice Voigt** α ou β
11	xx	1
22	yy	2
33	zz	3
23, 32	yz, zy	4
13, 31	xz, zx	5
12, 21	xy, yx	6

4.1.1 Formulação Elasticidade TTI

Para meios com isotropia transversal vertical (VTI), o eixo de simetria do material coincide com o eixo vertical. A matriz de rigidez para um meio VTI, em notação de Voigt, é dada por:

$$\mathbf{C}_{\text{VTI}} = \begin{pmatrix} C_{11} & C_{11}-2C_{66} & C_{13} & 0 & 0 & 0 \\ C_{11}-2C_{66} & C_{11} & C_{13} & 0 & 0 & 0 \\ C_{13} & C_{13} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{pmatrix} \tag{1}$$

Os componentes da matriz C_{IJ} podem ser expressos em termos das velocidades de onda P e S verticais (V_{P0}, V_{S0}), densidade (ρ) e parâmetros de Thomsen $(\epsilon, \delta, \gamma)$.

Para meios com isotropia transversal inclinada (TTI), o eixo de simetria do material não coincide com o eixo vertical. As propriedades elásticas são definidas rotacionando o tensor de rigidez de um meio VTI. Seja R a matriz de rotação definida pelos ângulos de dip (θ) e azimute (ϕ). A relação entre o tensor de rigidez no sistema rotacionado $(\mathbf{C}_{\text{TTI}})$ e o sistema original $(\mathbf{C}_{\text{VTI}})$ é dada pela transformação de Bond. Em notação tensorial:

$$C_{ijkl}^{\text{TTI}} = R_{ip}R_{jq}R_{kr}R_{ls}C_{pqrs}^{\text{VTI}} \tag{2}$$

onde os índices repetidos implicam soma. Alternativamente, usando a notação matricial de Voigt e a matriz de transformação de Bond $\mathbf{M}$:

$$\mathbf{C}_{\text{TTI}} = \mathbf{M}\mathbf{C}_{\text{VTI}}\mathbf{M}^T \tag{3}$$

A matriz $\mathbf{M}$ é construída a partir dos elementos da matriz de rotação R e permite o cálculo direto das propriedades do meio TTI a partir dos parâmetros VTI e dos ângulos de orientação do eixo de simetria.

4.2 Ferramentas Computacionais

Os solucionadores de FWI serão implementados utilizando `spyro` [15], uma biblioteca Python projetada para modelagem de propagação de ondas sísmicas. `spyro` foi construída sobre o Firedrake [14], um sistema automatizado para resolução de equações diferenciais parciais (EDPs) usando elementos finitos, escrito em Python e de código aberto.

Firedrake emprega a Unified Form Language (UFL) [3] para descrever problemas variacionais de forma simbólica de alto nível. Esta representação simbólica permite operações automáticas como diferenciação e cálculo de operadores adjuntos (transpostos hermitianos). Estas capacidades simbólicas, combinadas com a biblioteca de diferenciação algorítmica `pyadjoint` [10], habilitam o Firedrake aos cálculos automatizados gradientes baseados no método adjunto sem necessidade de derivação manual das equações.

A escolha desta plataforma justifica-se por diversos fatores: (i) flexibilidade na especificação de problemas variacionais através de notação matemática de alto nível; (ii) suporte nativo a malhas não estruturadas e elementos de ordem arbitrária; (iii) paralelismo automático via MPI; (iv) diferenciação algorítmica integrada; e (v) comunidade ativa e documentação extensiva.

4.2.1 Modelagem de fontes e receptores

As fontes de onda podem ser expressas na forma fraca como:

$$\int_\Omega f_s v \, dx = \int_\Omega r(N) p(\mathbf{x}) \cdot v \, dx = r(N) \int_\Omega q(\mathbf{x}) v \, dx = r(N) w_s, \tag{4}$$

onde $r(N)$ denota a wavelet (por exemplo, a wavelet de Ricker) avaliada no tempo discreto N, e $w_s : V \to V^*$ é um funcional que mapeia do espaço de funções V para seu espaço dual V^*.

Em UFL, qualquer função em V^* é referida como uma `Cofunction`. Portanto, referiremos w_s como uma cofunção ao longo desta seção.

A cofunção w_s é não nula apenas em uma localização específica, $\mathbf{x}_s = \{x_s, z_s\}$ em 2D (ou $\mathbf{x}_s = \{x_s, y_s, z_s\}$ em 3D). Como discutido em [12], a localização $\mathbf{x}_s$ implica uma malha de nuvem de pontos, que, neste caso, consiste em um único ponto onde a fonte de onda está localizada. No Firedrake, tal malha de nuvem de pontos é referida como malha somente de vértices (vertex-only mesh). A construção da malha somente de vértices é baseada na localização da fonte $\mathbf{x}_s$ usando o seguinte script:

```
source_mesh = VertexOnlyMesh(mesh, source_location)
```

A malha somente de vértices está imersa dentro de uma `mesh` mestre. Aqui, ela corresponde à `mesh` usada nos solucionadores de equação de onda.

O objetivo então é interpolar uma fonte pontual, definida na `source_mesh`, para o espaço de funções V^* a fim de obter a fonte de onda w_s. Uma vez que o ponto da fonte é definido em uma malha somente de vértices, onde apenas um valor é atribuído a cada célula do elemento, construímos um espaço de funções, V_s, como um espaço de Galerkin descontínuo de grau polinomial 0. Em seguida, computamos a cofunção do ponto da fonte $s \in V_s^*$ como segue:

$$s = \int_{\Omega_v} v_s \, dx, \tag{5}$$

onde v_s é a função teste em V_s, e $\Omega_v \subset \Omega$ representa a malha somente de vértices.

Após calcular $s \in V_s^*$, interpolamos para o espaço dual V^*. Esta interpolação é formulada como uma operação de elementos finitos, denotada pelo operador $I_{V^*} : V_s^* \to V^*$. Para uma cofunção $s \in V_s^*$, o operador de interpolação I_{V^*} mapeia s para $w_s \in V^*$ como:

$$I_{V^*}(; s) = w_s, \quad w_s \in V^*.$$

A notação de ponto e vírgula indica que o operador é linear em w_s.

Os dados dos receptores também são obtidos através de um interpolador de elementos finitos $I_{V_r} : V \to V_r$, tal que $I_{V_r}(; u) = u_r$, onde u_r representa os valores dos dados nas localizações dos receptores $\mathbf{x}_r$. O espaço de funções V_r é construído em uma malha somente de vértices criada a partir da nuvem de pontos definida pelas localizações dos receptores $\mathbf{x}_r$.

Um aspecto crucial da modelagem de fontes e receptores é que ambos são operações diferenciáveis. Esta propriedade é essencial para cálculos de gradiente baseados no adjunto através de diferenciação algorítmica.

4.2.2 Gradiente adjunto automatizado

`spyro` implementa solucionadores de propagação de ondas diferenciáveis, isto é, solucionadores que utilizam exclusivamente a linguagem UFL para resolucionador as equações. Esta característica é essencial para obter cálculos de gradiente automatizados, uma vez que o processo de diferenciação automática realizado pelo módulo `firedrake.adjoint` utiliza a UFL [3] para diferenciação simbólica das operações dos solucionadores diretos. Além disso, o `firedrake.adjoint` integra a biblioteca `pyadjoint` [9], que fornece capacidades de diferenciação algorítmica (AD) para calcular gradientes baseados no método

adjunto.

A biblioteca `pyadjoint` gerencia automaticamente a execução dos solucionadores direto e adjunto. Durante a execução do solucionador direto, ela constrói um grafo computacional (*computational tape*) armazenando todas as operações realizadas, suas dependências e saídas. Este grafo é então percorrido em ordem reversa para gerar automaticamente o solucionador adjunto, que calcula os gradientes necessários para o processo de otimização. Esta abordagem elimina a necessidade de derivação e implementação manual das equações adjuntas, reduzindo significativamente a probabilidade de erros e facilitando extensões metodológicas.

O módulo `firedrake.adjoint` facilita a execução eficiente de FWI ao aproveitar o paralelismo espacial e de fontes sísmicas através de funcionais reduzidos em ensemble. Além disso, incorpora estratégias de checkpointing para otimizar o gerenciamento de memória durante o processo de inversão, permitindo aplicações em problemas de larga escala.

4.2.3 Funcional Reduzido em Ensemble: Paralelismo de Fonte e Espacial

Para usar `EnsembleReducedFunctional`, um objeto `ensemble` deve primeiro ser instanciado. Este objeto requer um comunicador global (`COMM_WORLD`) e especifica o número de processos usados para cada membro do ensemble. O trecho de código abaixo demons tra a configuração:

```
1 from firedrake import *
2 ensemble = Ensemble(COMM_WORLD, M)
```

O objeto `ensemble` gerencia dois comunicadores durante a execução de solucionadores direto e adjunto com paralelismo de fonte e espacial. O primeiro, referido como `ensemble_comm`, comunica os valores do funcional e gradientes entre fontes de onda distintas. O segundo, o comunicador espacial, cuida da comunicação dentro da malha espacialmente distribuída para cada fonte de onda.

Como ilustrado na Figura 1, considere a computação de propagação de onda para três fontes ($s = 0, 1, 2$). Os parâmetros de controle $\mathbf{m}$ são distribuídos entre os $N_s = 3$ membros de `ensemble_comm`, cada um particionando ainda a malha através de $M = 2$ processos. Esta configuração permite a computação simultânea da equação de onda direta $F(\mathbf{u}, \mathbf{m}, s)$, os valores do funcional J_s, e os gradientes $\nabla_{\mathbf{m}} \widehat{J}_s(\mathbf{m})$ para $s = 0, 1, 2$. Para cada fonte s, a malha é distribuída sobre dois processos, que se comunicam usando o comunicador espacial. Uma vez que os valores do funcional e gradientes são computados para todas as fontes, o `ensemble_comm` reúne esses resultados e os soma para obter o valor total do funcional e gradiente, que são então utilizados no processo de otimização.

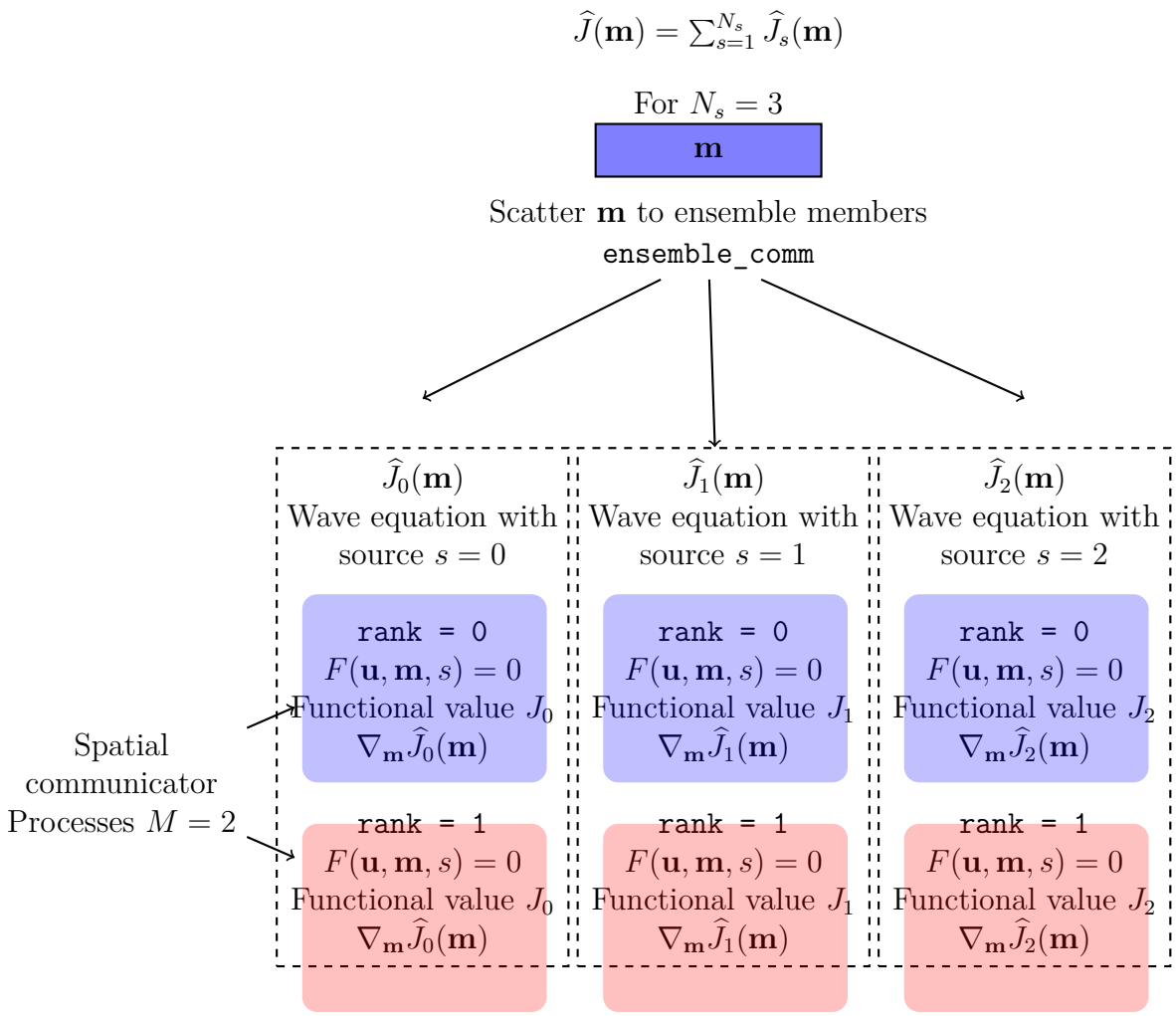

Figura 1: Ilustração da comunicação em ensemble e paralelismo no cálculo do funcional reduzido para múltiplas fontes em um fluxo de trabalho de FWI.

4.2.4 Checkpointing

Cálculos de gradiente baseados no adjunto são altamente intensivos em memória, especialmente para problemas de larga escala como FWI. Em EDPs dependentes do tempo, o gradiente adjunto tipicamente depende da solução direta e é computado em ordem temporal reversa. Isso requer primeiro resolucionador a EDP direta e armazenar as variáveis de estado diretas, levando a um requisito de memória proporcional ao número de passos de tempo, capaz de exceder a capacidade de um sistema computacional.

Gerenciar o uso de memória é portanto um desafio crítico em fluxos de trabalho de FWI. Uma estratégia para abordar esta questão é checkpointing, que envolve armazenar um número limitado de estados diretos intermediários. Estes checkpoints podem ser usados tanto para reiniciar a computação direta ou para a computação de gradientes baseados no adjunto. À medida que o cálculo adjunto avança em tempo reverso, a computação direta é progressivamente re-executada a partir do último estado armazenado disponível até o passo adjunto atual. Isso permite que menos estado direto seja armazenado, às custas de

um maior custo computacional de tempo, já que passos diretos são executados mais de uma vez.

A biblioteca de diferenciação algorítmica `pyadjoint` habilita checkpointing para cálculos de gradiente baseados no adjunto. Esta funcionalidade foi alcançada integrando o pacote `checkpoint_schedules` [4] no `pyadjoint`. O pacote `checkpoint_schedules` fornece uma gama de estratégias de checkpointing, permitindo que usuários (como `spyro`) alternem entre elas de forma transparente.

4.3 Análise e verificação de resultados

A verificação da implementação numérica será realizada em múltiplas etapas, seguindo metodologias estabelecidas na literatura de métodos numéricos e otimização inversa.

4.3.1 Análise de Erro Numérico

A análise dos resultados será inicialmente conduzida através da quantificação de erros numéricos utilizando parâmetros que se aproximam de modelos reais. Embora não haja solução analítica disponível para problemas complexos de FWI, utilizaremos dados de referência obtidos a partir de simulações de alta fidelidade (malhas refinadas e ordem de aproximação elevada) para avaliar a convergência da implementação numérica.

O erro será quantificado utilizando a norma L^2 relativa, definida como:

$$\epsilon_{L^2} = \frac{\|u_h - u_{ref}\|_{L^2(\Omega)}}{\|u_{ref}\|_{L^2(\Omega)}} = \frac{\sqrt{\int_\Omega |u_h - u_{ref}|^2 \, d\Omega}}{\sqrt{\int_\Omega |u_{ref}|^2 \, d\Omega}} \tag{6}$$

onde u_h representa a solução numérica com tamanho de elemento h, u_{ref} é a solução de referência de alta fidelidade, e Ω denota o domínio computacional.

Para verificar a taxa de convergência espacial do método de elementos finitos, utilizaremos a relação:

$$\epsilon_{L^2}(h) = Ch^p \tag{7}$$

onde p é a ordem de convergência esperada (dependente do grau polinomial do elemento finito) e C é uma constante. A taxa de convergência será estimada através do gráfico log-log:

$$\log(\epsilon_{L^2}) = p \log(h) + \log(C) \tag{8}$$

4.3.2 Verificação de Gradientes via Teste de Taylor

A corretude dos gradientes calculados via método adjunto será verificada utilizando o teste de Taylor. Para um funcional $J(\mathbf{m})$ (função objetivo da FWI) dependente do modelo $\mathbf{m}$,

a expansão de Taylor até segunda ordem em torno de $\mathbf{m}_0$ é:

$$J(\mathbf{m}_0 + \alpha\delta\mathbf{m}) = J(\mathbf{m}_0) + \alpha\nabla J(\mathbf{m}_0) \cdot \delta\mathbf{m} + \mathcal{O}(\alpha^2) \tag{9}$$

onde α é um escalar pequeno e $\delta\mathbf{m}$ é uma perturbação arbitrária.

Definindo o resíduo de primeira ordem:

$$R_1(\alpha) = |J(\mathbf{m}_0 + \alpha\delta\mathbf{m}) - J(\mathbf{m}_0) - \alpha\nabla J(\mathbf{m}_0) \cdot \delta\mathbf{m}| \tag{10}$$

Se o gradiente adjunto estiver correto, devemos observar que $R_1(\alpha) = \mathcal{O}(\alpha^2)$, ou seja, o erro decai com taxa quadrática quando $\alpha \to 0$. Em escala logarítmica:

$$\log(R_1(\alpha)) \approx 2\log(\alpha) + \text{const} \tag{11}$$

4.3.3 Verificação em Modelos de Referência

Na etapa final de verificação, serão realizadas simulações tridimensionais de FWI anisotrópico em modelos padrão da indústria de petróleo e gás (como os modelos EAGE Salt, Marmousi 3D, ou BP 2004 Benchmark), a fim de avaliar a capacidade do método em recuperar propriedades do meio subsuperficial a partir de dados sintéticos.

A qualidade da inversão será quantificada através de métricas específicas:

- **Redução do misfit**: Razão entre o valor inicial e final da função objetivo

$$R_{misfit} = \frac{J(\mathbf{m}_{final})}{J(\mathbf{m}_{inicial})} \tag{12}$$

- **Correlação entre modelos**: Coeficiente de correlação de Pearson

$$\rho = \frac{\text{cov}(\mathbf{m}_{true}, \mathbf{m}_{inv})}{\sigma_{\mathbf{m}_{true}}\sigma_{\mathbf{m}_{inv}}} \tag{13}$$

Estas métricas permitirão avaliar quantitativamente a eficácia da implementação e sua adequação para aplicações em dados reais.

5 Plano de Trabalho

5.1 Justificativa

A justificativa detalhada encontra-se na Seção 1 (Introdução e Justificativa) deste projeto. De forma sucinta, o projeto justifica-se pela necessidade de avançar nas técnicas de

imageamento sísmico em meios complexos (anisotrópicos), superando limitações de custo computacional e precisão através do uso de métodos de elementos finitos de alta ordem, diferenciação automática e paralelismo massivo. O desenvolvimento dessas ferramentas é estratégico para a caracterização de reservatórios em bacias sedimentares brasileiras.

5.2 Cronograma de Execução e Carga Horária

O projeto será executado ao longo de 24 meses, com dedicação integral de **40 horas semanais** (carga horária global estimada de 3.840 horas). O esquema de trabalho semanal seguirá o horário comercial padrão, distribuído entre desenvolvimento de código, execução de simulações, análise de dados e redação científica, conforme acordado com o supervisor Prof. Dr. Bruno Souza Carmo.

A distribuição temporal das atividades de pesquisa está detalhada na Tabela 2, organizada por trimestres.

Tabela 2: Cronograma de atividades do projeto (24 meses)

Atividade	**Trimestre**							
	1	2	3	4	5	6	7	8
1. Revisão bibliográfica e fundamentação teórica	X	X						
2. Implementação do solucionador e integração de AD	X	X	X					
3. Comparação AD vs. Adjunto Manual (Eficiência)			X	X				
4. Implementação de paralelismo *ensemble*				X	X			
5. Inversão multiparamétrica e mitigação de *trade-offs*					X	X	X	
6. Verificação em modelos 3D complexos							X	X
7. Disseminação (artigos, documentação, tutoriais)			X		X		X	X

5.3 Atividades de Disseminação e Interação

Além das atividades de pesquisa estritas, o pós-doutorando(a) engajar-se-á em atividades que visam maximizar o impacto acadêmico e social do projeto. Buscando promover a divulgação científica, participará de apresentações em seminários e de atividades de extensão envolvendo ferramentas de código aberto para simulação científica, com o objetivo de capacitar a comunidade externa no uso e no desenvolvimento de software científico aberto.

6 Considerações Finais

Este projeto propõe o desenvolvimento de um *framework* computacional inovador para FWI anisotrópico, integrando diferenciação algorítmica e paralelismo em ensemble.

Como resultado, espera-se avançar significativamente na eficiência e precisão da inversão sísmica em meios anisotrópicos, contribuindo para a compreensão das propriedades subsuperficiais e aprimorando as práticas de exploração geofísica. Pretende-se também disseminar os resultados através de publicações científicas e documentação técnica, promovendo o uso do *framework* pela comunidade acadêmica e industrial.

Referências

[1] Keiiti Aki and Paul G Richards. *Quantitative seismology*. University Science Books, Sausalito, California, second edition, 2002.

[2] Tariq Alkhalifah and R.-E. Plessix. A recipe for practical full-waveform inversion in anisotropic media: An analytical parameter resolution study. *Geophysics*, 79(3):R91–R101, 2014.

[3] Martin S Alnæs, Anders Logg, Kristian B Ølgaard, Marie E Rognes, and Garth N Wells. Unified Form Language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software (TOMS)*, 40(2):1–37, 2014.

[4] Daiane I. Dolci, James R. Maddison, David A. Ham, Guillaume Pallez, and Julien Herrmann. checkpoint_schedules: schedules for incremental checkpointing of adjoint simulations. *Journal of Open Source Software*, 9(95):6148, 2024.

[5] Daniel Köhn, Dorian De Nil, André Kurzmann, Alicja Przebindowska, and Thomas Bohlen. Influence of model parameterization for elastic full waveform tomography. *Geophysical Journal International*, 191(1):325–345, 2012.

[6] Dimitri Komatitsch and Jeroen Tromp. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical journal international*, 139(3):806–822, 1999.

[7] Dimitri Komatitsch and Jean-Pierre Vilotte. The spectral element method: an efficient tool for simulating the seismic response of 2d and 3d geological structures. *Bulletin of the seismological society of America*, 88(2):368–392, 1998.

[8] J Luo, H Zhu, T Nissen-Meyer, C Morency, and J Tromp. Seismic modeling on irregular meshes using the spectral-element method. *Geophysics*, 77(4):T301–T312, 2012.

[9] Sebastian K. Mitusch. An algorithmic differentiation tool for FEniCS. Master's thesis, University of Oslo, 2018.

[10] Sebastian K. Mitusch, Simon W. Funke, and Jørgen S. Dokken. dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. *Journal of Open Source Software*, 4(38):1292, 2019.

[11] Ryan T Modrak and Jeroen Tromp. Seismic waveform inversion best practices: regional, global and exploration test cases. *Geophysical Journal International*, 206(3):1864–1889, 2016.

[12] R. W. Nixon-Hill, D. Shapero, C. J. Cotter, and D. A. Ham. Consistent point data assimilation in firedrake and icepack. *Geoscientific Model Development*, 17(13):5369–5386, 2024.

[13] Stéphane Operto, Y Gholami, V Prieux, A Ribodetti, R Brossier, L Metivier, and Jean Virieux. A guided tour of multiparameter full-waveform inversion with multicomponent data: From theory to practice. *The Leading Edge*, 32(9):1040–1054, 2013.

[14] Florian Rathgeber, David A Ham, Lawrence Mitchell, Michael Lange, Fabio Luporini, Andrew TT McRae, Gheorghe-Teodor Bercea, Graham R Markall, and Paul HJ Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43(3):1–27, 2016.

[15] Keith J. Roberts, Alexandre Olender, Lucas Franceschini, Robert C. Kirby, Rafael S. Gioria, and Bruno S. Carmo. Spyro: a Firedrake-based wave propagation and full-waveform-inversion finite-element solver. *Geoscientific Model Development*, 15(23):8639–8667, 2022.

[16] Albert Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266, 1984.

[17] Solvi Thrastarson, Dirk-Philip van Herwaarden, and Andreas Fichtner. Inversion-based resolution analysis in full waveform inversion. *Geophysical Journal International*, 221(3):1596–1616, 2020.

[18] Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.

[19] Michael Warner, Andrew Ratcliffe, Tenice Nangoo, Joanna Morgan, Adrian Umpleby, Nikhil Shah, Vetle Vinje, Ivan Štekl, Lluís Guasch, Caroline Win, Graham Conroy, and Alexandre Bertrand. Anisotropic 3D full-waveform inversion. *Geophysics*, 78(2):R59–R80, 2013.