



UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA  
DEPARTAMENTO DE MATEMÁTICA APLICADA



BERNARDO CORREIA PRADOS NASCIMENTO

**Revisão e aprimoramento da tomografia  
Common Reflection Point (CRP)**

Campinas  
20/11/2025

BERNARDO CORREIA PRADOS NASCIMENTO

**Revisão e aprimoramento da tomografia  
Common Reflection Point (CRP)\***

Monografia apresentada ao Instituto de Matemática,  
Estatística e Computação Científica da Universidade  
Estadual de Campinas como parte dos requisitos para  
obtenção de créditos na disciplina Projeto Supervis-  
ionado, sob a orientação do Prof. Jorge Henrique  
Faccipieri Junior.

---

\*Este trabalho foi financiado pela PETROBRAS, projeto 115604-25.

## **Resumo**

O presente projeto visa, sobretudo, ao estudo exploratório da Tomografia CRP - técnica de inversão sísmica para imageamento de subsuperfícies -, sob o fim de identificar possíveis fragilidades, e lacunas para aprimoramento. Nesse sentido, o desenvolvimento do projeto contou com uma reimplementação computacional autoral da técnica, lançando mão de noções de paralelismo, além de técnicas de otimização não-linear.

## **Abstract**

The current project aims primarily at the exploratory study of the CRP Tomography - seismic inversion technique for subsurface imaging -, in order to identify possible weaknesses, and loopholes for refinement. In this regard, the project's development includes a brand new authorial computational implementation, taking advantage of parallel coding notions, as well as nonlinear optimization techniques.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>6</b>
<b>2</b>	<b>Modelo Matemático</b>	<b>7</b>
2.1	Problema de inversão . . . . .	7
2.2	Dados observados e variáveis de modelo . . . . .	8
2.3	Método de Gauss-Newton . . . . .	12
2.4	Traçamento de raios . . . . .	14
2.5	Algoritmo da Tomografia CRP . . . . .	15
<b>3</b>	<b>Restruturação</b>	<b>16</b>
3.1	Paralelização do Traçamento de Raios por CPU . . . . .	16
3.2	Geradores de experimentos sintéticos . . . . .	18
3.3	Módulos auxiliares: regularização e interpolação . . . . .	19
3.4	Reimplementação da Tomografia CRP . . . . .	19
3.5	Aplicação de <i>backtracking</i> . . . . .	20
3.6	Imposição de restrições naturais . . . . .	22
<b>4</b>	<b>Testes sintéticos</b>	<b>23</b>
4.1	Metodologia . . . . .	23
4.2	Resultados . . . . .	24
<b>5</b>	<b>Conclusão</b>	<b>27</b>

# 1 Introdução

A caracterização precisa da subsuperfície terrestre é um desafio central nas geociências aplicadas, particularmente na exploração de hidrocarbonetos. Entre as abordagens mais promissoras para esse fim, destacam-se os métodos de tomografia sísmica, os quais visam reconstruir propriedades do meio - como o modelo de velocidades, e as superfícies refletoras - mediante a resolução de um problema de minimização.

Sob essa perspectiva, a técnica de tomografia baseada em pontos de reflexão comum (*common reflection points*, no inglês) apresenta vantagens significativas. Ao organizar os dados sísmicos em famílias CRP – conjunto de pares fonte-receptor que compartilham um ponto de reflexão comum das ondas primárias –, de modo a adicionar mais informações relacionadas a um mesmo ponto em profundidade, obtém-se maior acurácia na modelagem da geometria dos refletores. Essa metodologia, introduzida na tese de doutorado de Ignácio [2021], é especialmente relevante por agregar mais informações durante a resolução do problema de inversão, mediante ligeiras simplificações do modelo físico, suscitando subsequente vantagem computacional durante os cálculos numéricos. Por conseguinte, a tese apresenta uma ferramenta em potencial para inversão sísmica.

Contudo, apesar de promissora, a formulação teórica e a implementação concreta da técnica demonstram fragilidades que restringem sua aplicabilidade, bem como escalabilidade, em cenários complexos. Por essa ótica, este trabalho tem como objetivo revisar e aprimorar a Tomografia CRP, propondo melhorias na formulação matemática, na estratégia de otimização e na implementação computacional. A saber, as principais contribuições incluem: (i) imposição de positividade (valores maiores ou iguais a zero) e limites físicos ao modelo de velocidades durante o processo de otimização; (ii) adoção de um esquema de *backtracking* baseado na minimização de cúbicas visando à aceleração da convergência do método de Gauss–Newton a um ponto estacionário; e (iii) reimplementação modular, paralelizada e com maior adaptabilidade ao usuário, do código original. Os novos códigos foram escritos em Python, em razão da sua mais fácil compreensão e abstração, bem como pelo seu rico acervo em bibliotecas numéricas e de paralelismo gratuitas – diferentemente da implementação original, feita em MATLAB.

## 2 Modelo Matemático

Nesta seção, é apresentado o modelo matemático do problema de inversão que fundamenta a Tomografia CRP. Para tanto, definem-se as componentes do processo de inversão; explicitam-se as respectivas correspondências na Tomografia CRP; ademais, especifica-se o algoritmo de otimização não linear adotado na técnica; e, então, apresenta-se brevemente a função modeladora de dados, que reside no traçamento de raios sísmicos. Por fim, descreve-se o algoritmo geral da técnica.

### 2.1 Problema de inversão

Enquanto técnica tomográfica, o método em análise objetiva a determinação de variáveis, relacionadas ao modelo geológico, as quais otimizam um problema de quadrados mínimos. Genericamente, a partir de um conjunto de dados invariantes, e de um palpite inicial para as variáveis minimizadoras, estas são iterativamente atualizadas conforme uma rotina de otimização - neste caso, o método de Gauss-Newton. Desse modo, mitiga-se o resíduo entre os dados invariantes, e os dados sintéticos a cada iteração da rotina.

Em termos matemáticos, formula-se o problema de inversão de quadrados mínimos a partir dos seguintes itens:

- Dados observados  $\mathbf{d}^{obs}$ : vetor real  $n$ -dimensional cujas entradas consistem em um conjunto de dados observados através de um experimento, e cedidos pelo usuário.
- Variáveis de modelo  $\mathbf{m}$ : vetor real  $M$ -dimensional de parâmetros que se deseja inverter de fato. É sobre ele que o processo de otimização, propriamente dito, ocorre.
- Dados sintéticos  $\mathbf{d}^{syn}(\mathbf{m})$ : vetor real  $n$ -dimensional, gerado a partir de  $\mathbf{m}$  mediante uma função de modelagem - neste caso, o traçamento de raios ulteriormente apresentado. Idealmente, procura-se gerar, por contínuas atualizações de  $\mathbf{m}$ , dados sintéticos que “se aproximem” o máximo possível daqueles observados e invariantes, isto é,  $\mathbf{d}^{obs}$ .
- Função de erro  $S(\mathbf{m})$  : uma função de  $\mathbb{R}^M$  para  $\mathbb{R}_+$ , a qual quantifica a discrepância entre os dados sintéticos e os observados, em função de um vetor de parâmetros de

modelo qualquer. A saber, em sua forma mais simples para o problema de quadrados mínimos, pode-se expressá-la como

$$S(\mathbf{m}) = \frac{1}{2} \Delta \mathbf{d}^T(\mathbf{m}) W_D^{-1} \Delta \mathbf{d}(\mathbf{m}),$$

em que  $\Delta \mathbf{d}(\mathbf{m}) = \mathbf{d}^{syn}(\mathbf{m}) - \mathbf{d}^{obs}$ , e  $W_D^{-1}$  é a inversa da matriz diagonal de covariância de dimensão  $n \times n$  - definida previamente pelo usuário.

- Esquema de atualização: pilar do processo de inversão, a rotina de atualização provê as sucessivas modificações sobre o modelo, necessárias para minimizar a função de custo  $S(\mathbf{m})$ . Ou seja, a partir de um palpite inicial  $\mathbf{m}_0$ , o esquema de atualização gera uma sequência de vetores modelo  $\mathbf{m}_k$ , para  $k = 1, 2, \dots$  que idealmente convergem em direção ao modelo real. Dessa forma, a cada iteração  $k$ , obtém-se

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha_k \Delta \mathbf{m}_k ,$$

em que  $\alpha_k$  corresponde a um escalar o qual controla a intensidade do passo de atualização  $\Delta \mathbf{m}_k$  - este, gerado segundo uma rotina de otimização. Em seguintes subseções, explora-se a formulação matemática do método de Gauss-Newton, a rotina de otimização utilizada na Tomografia CRP.

## 2.2 Dados observados e variáveis de modelo

Nesta subseção, explicitam-se as variáveis que compõem, de fato, os vetores de dados invariantes, assim como o vetor de modelo para o processo de inversão da Tomografia CRP.

**Dados observados:** Para cada ponto em profundidade, gera-se um vetor que agrupa informações referentes à família CRP - ou seja, conjunto de pares fonte-receptor associados a esse mesmo ponto refletor. Para os propósitos da Tomografia CRP, um único par-receptor é caracterizado inequivocamente por: um tempo de viagem dupla  $t_{sr}$  isto é, o tempo transcorrido entre a emissão da onda pela fonte, e sua respectiva captação pelo receptor; e também por um vetor  $(s, r, p_s, p_r)^T$ , em que  $s$  e  $r$  são as coordenadas espaciais

da fonte e receptor respectivamente;  $p_s$  e  $p_r$  são as componentes horizontais dos vetores de vagarosidade na posição da fonte e do receptor, nessa ordem.

Por conseguinte, dada uma família CRP  $f$ , dotada de  $N_f$  pares fonte-receptor, associa-se a ela um vetor de dados invariantes  $\mathbf{d}_f^{obs} = [t_f, (s, r, p_s, p_r)_{f_i}^T]_{i=1}^{N_f}$ , em que  $t_f$  é a soma dos tempos de viagem dupla de cada par  $f_i$ , ou seja,  $t_f = \sum_{i=1}^{N_f} (t_{sr})_i$ . Vale ressaltar que a compressão dos tempos de viagem a uma única variável  $t_f$  para cada família CRP  $f$  permite a redução do custo computacional da técnica, ao passo que possibilita a agregação de múltiplos pares fonte-receptor, sem impactar radicalmente, durante o processo de inversão, sobre o tamanho da matriz jacobiana - posteriormente apresentada.

A Figura 1, logo abaixo, exemplifica a definição de famílias CRP.

Agora, para um modelo com  $N$  pontos de reflexão comum, compõem-se os dados observados pelo (grande) vetor formado a partir da concatenação individual dos diversos  $\mathbf{d}_f^{obs}$ , logo, obtém-se  $\mathbf{d}^{obs} = [\mathbf{d}_f^{obs}]_{f=1}^N$ , vetor  $n$ -dimensional. Note que  $n = N + \sum_{f=1}^N 4N_f$ .

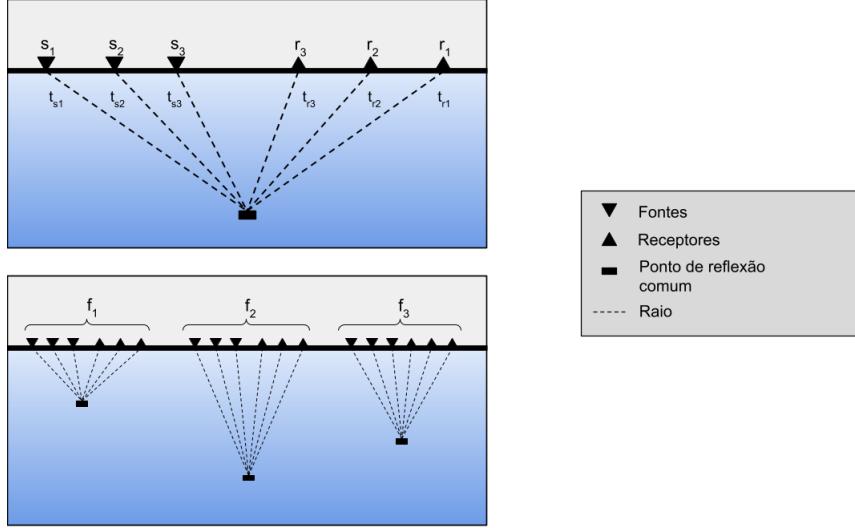


Figura 1: Esquema ilustrativo de duas configurações de famílias CRP. Em cima, o detalhe de uma única família com 3 pares fonte-receptor. Em baixo, uma configuração com 3 pontos de reflexão comum, a cada qual estão associados 3 pares fonte-receptor.

**Variáveis de modelo:** Pode-se definir o vetor de modelo  $\mathbf{m}$  como a concatenação entre as variáveis de modelagem relacionadas ao modelo de velocidade ( $\mathbf{m}^{vel}$ ), e aquelas relacionadas ao traçamento de raios ( $\mathbf{m}^{ray}$ ). Daí, tem-se  $\mathbf{m} = [\mathbf{m}^{vel}, \mathbf{m}^{ray}]$ .

Sendo assim, o modelo do modelo de velocidade consiste em um vetor  $M$ -dimensional

cujas entradas correspondem a escalares reais positivos, representando a velocidade da onda para cada ponto do espaço em análise. Esses valores constituem os coeficientes de interpolação das bases de B-spline. Em suma, atesta-se que

$$\mathbf{m}^{vel} = [v_{ij}] \quad \text{para } i = 1, 2, \dots, M_x \quad \text{e} \quad j = 1, 2, \dots, M_z \quad ,$$

em que  $M_x$  e  $M_z$  correspondem ao número de nós de interpolação, previamente definido, ao longo dos eixos  $x$  e  $z$ , nessa ordem. Note que, em consonância com a dimensionalidade postulada, deve-se ter que  $\mathbf{m}^{vel}$  tem dimensão dada por  $M_x \cdot M_z$ .

Desse modo, dada uma malha computacional com  $M_x$  linhas e  $M_z$  colunas, a cada iteração interpola-se o modelo atualizado, possibilitando a avaliação da velocidade para qualquer ponto  $(x, z)$  dentro dos limites físicos da malha, de maneira que:

$$c(x, z) = \sum_{i=1}^{M_x} \sum_{j=1}^{M_z} c_{ij} \beta_i(x) \beta_j(z) \quad ,$$

onde  $\beta_i(x)$  e  $\beta_j(z)$  correspondem a  $i$ -ésima e  $j$ -ésima bases de interpolação em  $x$  e em  $z$ , respectivamente. O procedimento supracitado garante a geração de modelos suaves ao empregar interpolação por cúbicas - duas vezes diferenciáveis. Tal característica é essencial para aplicação da função de modelagem, a qual preconiza a propagação de ondas em meios geológicos suaves. Ademais, para uma explicação completa acerca da construção de B-splines, recomenda-se DeBoor [1978].

No que tange à  $\mathbf{m}^{ray}$ , pode-se defini-lo como a coleção de  $N$  vetores modelo referentes a cada família CRP  $f$  munida de  $N_f$  pares fonte-receptor. Note que  $N$  equivale ao número de pontos refletores cedidos pelo usuário durante a definição dos dados invariantes. Dessa forma, tem-se:

$$\mathbf{m}^{ray} = [\mathbf{m}_f^{ray}]_{f=1}^N \quad ,$$

em que cada  $\mathbf{m}_f^{ray}$  é definido por

$$\mathbf{m}^{ray} = [(x, z)^T, (\theta_s, \theta_r)_{f_i}^T]_{i=1}^{N_f} \quad .$$

Para cada  $f$ , o vetor  $m_f^{ray}$  encapsula a posição aproximada  $(x, z)$  do ponto refletor que identifica a família em análise, além de que armazena os ângulos de incidência ( $\theta_s$ ) e de reflexão ( $\theta_r$ ) do raio associado a cada par fonte-receptor  $i$ . Complementarmente, deve-se notar que  $\mathbf{m}$  é vetor  $M$ -dimensional, tal que  $M = M_x \cdot M_z + 2\left(N + \sum_{f=1}^N N_f\right)$ . Ilustra-se, na Figura 2, a separação entre os conjuntos de dados.

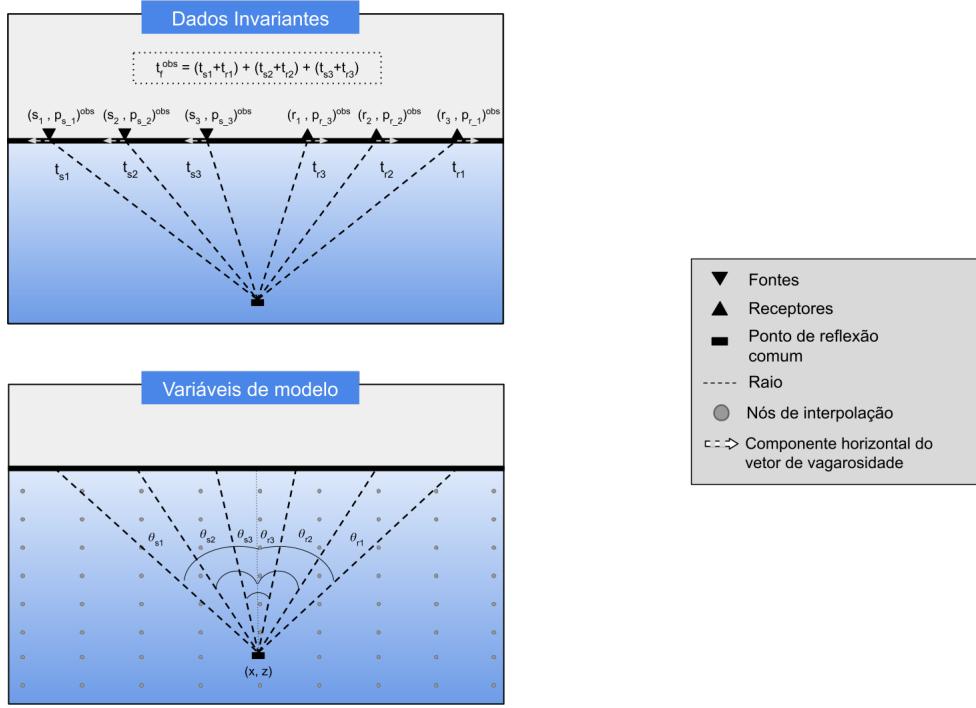


Figura 2: Esquema ilustrativo dos dados invariantes (imagem superior) e das variáveis de modelo (imagem inferior). No exemplo, utiliza-se apenas uma família CRP, dotada de 3 pares fonte-receptor.

Assim sendo, munido das definições acima descritas, atesta-se que a partir de um  $\mathbf{m}^{vel}$  arbitrário, cada vetor  $\mathbf{m}_f^{ray}$  caracteriza  $N_f$  pares de raios que partem da aproximação do ponto refletor comum em direção à superfície. A partir do traçamento de raios, geram-se para cada família CRP, um vetor de dados sintéticos  $\mathbf{d}_f^{syn}$  - análogo, em termos de entradas, ao vetor de dados invariantes  $\mathbf{d}_f^{obs}$ . Consequentemente, obtém-se  $\mathbf{d}_f^{syn} = [t_f^{syn}, (s, r, p_s, p_r)_{f_i}^{syn}]_{i=1}^{N_f}$ , cujas entradas possuem o mesmo significado físico daquelas que compõem  $\mathbf{d}_f^{obs}$ , a despeito de serem parâmetros simulados pela função de modelagem. Daí, a reunião dessas estruturas de dados define o dado sintético  $\mathbf{d}^{syn}$ , isto é,

$$\mathbf{d}^{syn} = [\mathbf{d}_f^{syn}]_{f=1}^{N_f} .$$

Idealmente, busca-se a igualdade entre os dados sintéticos e os observados, ou seja  $\mathbf{d}^{syn}(\mathbf{m}) = \mathbf{d}^{obs}$ , a fim de extrair as variáveis de modelo físico  $\mathbf{m}$  que descrevem a subsuperfície.

### 2.3 Método de Gauss-Newton

Enquanto método de otimização de segunda ordem, Gauss-Newton emprega informações acerca da derivada segunda, a partir de uma aproximação da Hessiana associada à função objetivo, para estimar a direção do passo de atualização das variáveis do modelo (Nocedal and Wright [2006]).

Formalmente, na  $k$ -ésima iteração do processo de otimização de uma função  $S$  duas vezes diferenciável, determina-se a direção  $\Delta\mathbf{m}_k$  através da equação matricial

$$H_S(\mathbf{m}_k) \Delta\mathbf{m}_k = -\nabla S(\mathbf{m}_k) , \quad (1)$$

em que  $H_S$  consiste na matriz hessiana associada à  $S(\mathbf{m}_k)$ . Da expressão, nota-se que o gradiente de  $S$ , avaliado em  $\mathbf{m}_k$ , pode ser expresso por

$$\nabla S(\mathbf{m}_k) = J(\mathbf{m}_k)^T W_D^{-1} \Delta\mathbf{d}(\mathbf{m}_k) , \text{ com } J(\mathbf{m}_k) = \frac{\partial \mathbf{d}^{syn}}{\partial \mathbf{m}}|_{\mathbf{m}=\mathbf{m}_k} .$$

Complementarmente, empregando-se a aproximação de Gauss-Newton para a hessiana, obtém-se

$$H_S(\mathbf{m}_k) \approx J(\mathbf{m}_k)^T W_D^{-1} J(\mathbf{m}_k).$$

Dessa maneira, a Equação (1) para atualização do passo à cada iteração torna-se:

$$J(\mathbf{m}_k)^T W_D^{-1} J(\mathbf{m}_k) \Delta\mathbf{m}_k = -J(\mathbf{m}_k)^T W_D^{-1} \Delta\mathbf{d}(\mathbf{m}_k) .$$

Como demonstrado por Ignácio em sua tese (páginas 49 e 50), pode-se manipular a última expressão, de modo à conformá-la ao formato de uma equação normal do problema de mínimos quadrados. Nesse sentido, tem-se o sistema normal

$$[J(\mathbf{m}_k)^T W_D^{-1/2}]^T [J(\mathbf{m}_k)^T W_D^{-1/2}] \Delta \mathbf{m}_k = -[J(\mathbf{m}_k)^T W_D^{-1/2}]^T [W_D^{-1/2} \Delta \mathbf{d}(\mathbf{m}_k)],$$

cuja solução está associada, por sua vez, à equação matricial a seguir:

$$[J(\mathbf{m}_k)^T W_D^{-1/2}] \Delta \mathbf{m}_k = W_D^{-1/2} \Delta \mathbf{d}(\mathbf{m}_k). \quad (2)$$

Em termos numéricos, o sistema linear obtido em (2), inerente à atualização de passo durante as iterações tomográficas, simboliza um dos principais entraves para a performance computacional, em razão de seu mal condicionamento. Na implementação original da Tomografia CRP, empregou-se o método da decomposição em valores singulares (SVD), cujo dispêndio computacional é compensado por sua acurácia - ao menos para formulação em 2 dimensões.

No que concerne à matriz Jacobiana, vale apontar que possui um número de linhas igual à  $\sum_{f=1}^N (4N_f + 1)$  – em razão do número total de variáveis em  $\mathbf{d}^{syn}$  -; e um número de colunas equivalente à  $M + \sum_{f=1}^N (2N_f + 2)$ , devido ao número de variáveis contidas em  $\mathbf{m}$ . Por conseguinte, a matriz pode ser representada por:

$$J(\mathbf{m}) = \begin{bmatrix} \frac{\partial \mathbf{t}_{syn}}{\partial \mathbf{x}} & \frac{\partial \mathbf{t}_{syn}}{\partial \mathbf{z}} & \frac{\partial \mathbf{t}_{syn}}{\partial \theta_s} & \frac{\partial \mathbf{t}_{syn}}{\partial \theta_r} & \frac{\partial \mathbf{t}_{syn}}{\partial \mathbf{v}} \\ \frac{\partial \mathbf{p}_s^{syn}}{\partial \mathbf{x}} & \frac{\partial \mathbf{p}_s^{syn}}{\partial \mathbf{z}} & \frac{\partial \mathbf{p}_s^{syn}}{\partial \theta_s} & \frac{\partial \mathbf{p}_s^{syn}}{\partial \theta_r} & \frac{\partial \mathbf{p}_s^{syn}}{\partial \mathbf{v}} \\ \frac{\partial \mathbf{p}_r^{syn}}{\partial \mathbf{x}} & \frac{\partial \mathbf{p}_r^{syn}}{\partial \mathbf{z}} & \frac{\partial \mathbf{p}_r^{syn}}{\partial \theta_s} & \frac{\partial \mathbf{p}_r^{syn}}{\partial \theta_r} & \frac{\partial \mathbf{p}_r^{syn}}{\partial \mathbf{v}} \\ \frac{\partial \mathbf{s}^{syn}}{\partial \mathbf{x}} & \frac{\partial \mathbf{s}^{syn}}{\partial \mathbf{z}} & \frac{\partial \mathbf{s}^{syn}}{\partial \theta_s} & \frac{\partial \mathbf{s}^{syn}}{\partial \theta_r} & \frac{\partial \mathbf{s}^{syn}}{\partial \mathbf{v}} \\ \frac{\partial \mathbf{r}^{syn}}{\partial \mathbf{x}} & \frac{\partial \mathbf{r}^{syn}}{\partial \mathbf{z}} & \frac{\partial \mathbf{r}^{syn}}{\partial \theta_s} & \frac{\partial \mathbf{r}^{syn}}{\partial \theta_r} & \frac{\partial \mathbf{r}^{syn}}{\partial \mathbf{v}} \end{bmatrix} \quad (3)$$

Computacionalmente, a Jacobiana da Tomografia CRP se beneficia, a despeito da tradicional Estereotomografia (Billette and Lambaré [1998]), ao associar diversos pares fonte-receptor (e seus raios) a um mesmo ponto em profundidade, e a um mesmo valor total de viagem. Tais “acoplamentos” permitem a inversão das famílias CRP em conjunto, em detrimento de uma análise individual para cada tempo de viagem e ponto de reflexão

- o que acarretaria no incremento da dimensionalidade da Jacobiana.

## 2.4 Traçamento de raios

Ao longo das iterações do processo de inversão, faz-se necessário aplicar a função de modelagem sobre cada  $\mathbf{m}_k$ , a fim de computar o erro entre os dados invariantes e os aproximados. Dessa forma, emprega-se o traçamento de raios: um sistema de equações diferenciais que simula a propagação de um ponto sobre a frente de onda, em função do comprimento de arco descrito, ou em função do tempo de viagem da onda. Em símbolos, dada uma velocidade do meio  $c(\mathbf{r})$ , em que  $\mathbf{r} = (x, z)$  é o vetor de posição (caso bidimensional), o traçado de raios é constituído pelas seguintes equações diferenciais em termos do comprimento de arco  $ds$ :

$$\begin{cases} \frac{d\mathbf{r}}{ds} = \frac{\mathbf{p}}{\sigma(\mathbf{r})} \\ \frac{dT}{ds} = \sigma(\mathbf{r}) \\ \frac{d\mathbf{p}}{ds} = \nabla\sigma(\mathbf{r}) \end{cases} \quad (4)$$

em que  $T$  é o tempo de viagem da onda;  $\sigma(\mathbf{r}) = \frac{1}{c(\mathbf{r})}$  consiste na vagarosidade do meio; bem como  $\mathbf{p}$  representa o vetor de vagarosidade, de modo que vale a igualdade  $\|\mathbf{p}(\mathbf{r})\|_2 = \sigma(\mathbf{r})$ . Em conexão com as variáveis de modelo, vale ressaltar que um raio incidente sobre um refletor, cuja inclinação em relação à vertical é dada por  $\theta$ , está submetido a um vetor de vagarosidade  $\mathbf{p} = \sigma(\mathbf{r}) \cdot [\sin \theta, \cos \theta]$ .

Dado um modelo de velocidade e uma condição inicial, isto é, valores de  $\mathbf{r}, T$  e  $\mathbf{p}$  (ou inclinação  $\theta$ ) em  $s = 0$ , pode-se computar novos valores iterativamente com base em métodos numéricos para problemas de valor inicial (PVI). Na implementação original, tal qual na atual, emprega-se o método de Euler avançado de primeira ordem, em razão de sua eficiência computacional (veja LeVeque [2007]).

Complementarmente, merece destaque o fato de que as derivadas parciais que compõem a Jacobiana na Equação (3) podem ser calculadas, mediante resultados da teoria paraxial de raios (Cerveny [2005]), individualmente para cada traçado. Esse fato possibilita, durante o processo de inversão, a simultaneidade da modelagem de  $\mathbf{d}^{syn}(\mathbf{m}_k)$  e o cálculo de  $\nabla S(\mathbf{m}_k)$ .

## 2.5 Algoritmo da Tomografia CRP

Enfim, uma vez dotados de todas as partes constituintes do problema de inversão, é possível apresentar o algoritmo da técnica tomográfica. Vale frisar que o seguinte enunciado diz respeito à versão original proposta e implementada por Ignácio em sua tese.

### Algoritmo da Tomografia CRP

**Passo 0** Dados de entrada:  $\mathbf{d}^{obs}$ , além de,  $W_D^{-1} \in \mathbb{R}^{n \times n}$ ,  $M_x$  e  $M_z \in \mathbb{N}$ ,  $v_0 \in \mathbb{R}_+$ .

**Passo 1** Inicialização: Calcule  $\mathbf{m}_0^{vel}$  e  $\mathbf{m}_0^{ray}$ , daí defina  $\mathbf{m}_0 = \mathbf{m}_0^{vel} \cup \mathbf{m}_0^{ray}$ ,  $k = 0$ ,  
 $\mathbf{m}_k = \mathbf{m}_0$ .

**Passo 2** Modelagem dos dados: Gere o modelo de velocidade  $c(x, z)$  por interpolação, compute  $\mathbf{d}^{syn}(\mathbf{m}_k)$  através do traçamento de raios, e então calcule  $S(\mathbf{m}_k)$  e  $\nabla S(\mathbf{m}_k)$ .

**Passo 3** Critério de parada: Se  $\|\nabla S(\mathbf{m}_k)\|_2 = 0$ , FIM. Senão, vá para o Passo 4.

**Passo 4** Direção de passo: Calcule o passo de variação  $\Delta\mathbf{m}_k$  a partir da Equação (2).

**Passo 5** Tamanho de passo: Encontre o menor inteiro não negativo  $j$  tal que  $\alpha_k = (\frac{1}{2})^j$  satisfaça a condição

$$S(\mathbf{m}_k + \alpha_k \Delta\mathbf{m}_k) \leq S(\mathbf{m}_k).$$

**Passo 6** Atualização: Defina  $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \Delta\mathbf{m}_k$ ,  $k = k + 1$  e volte para o Passo 2.

Vale apontar que, no Passo 1 do algoritmo, o cálculo de  $\mathbf{m}_0^{vel}$  e  $\mathbf{m}_0^{ray}$  é feita da seguinte forma: a princípio, define-se  $\mathbf{m}_0^{vel} = v_0 \cdot \mathbf{1}_{M_x \times M_z}$ , em que  $\mathbf{1}_{M_x \times M_z}$  é a matriz unitária de dimensões  $M_x \times M_z$ , e então se interpola  $c(x, z)$ . Dessa maneira, o modelo de velocidade inicial será constante no espaço, simulando um cenário em que não se tem nenhuma informação a priori do modelo geológico. Em seguida, para cada fonte e receptor de uma família  $f$ , aplica-se o traçamento de raios em direção à subsuperfície, a partir das condições iniciais de posição e vagarosidade definidas por  $\mathbf{d}^{obs}$  (refletindo o vetor de vagarosidade), até que se alcance  $t_f/(2N_f)$  segundos de propagação. Dessa forma, ao fim do traçado,

obtêm-se os respectivos ângulos de incidência e emergência  $(\theta_s, \theta_r)_{f_i}|_{i=1}^{N_f}$ . Ademais, define-se o ponto refletor relativo à  $f$  como o centroíde dos pontos finais de propagação de todos os raios da família - tal como é exibido na Figura 3. Por conseguinte,  $\mathbf{m}_{0_f}^{ray}$  é dado por:

$$\mathbf{m}_{0_f}^{ray} = \left( \frac{\sum_{i=1}^{N_f} (x_s + x_r)_i}{2N_f}, \frac{\sum_{i=1}^{N_f} (z_s + z_r)_i}{2N_f}, (\theta_s, \theta_r)_i|_{i=1}^{N_f} \right), \quad \forall f = 1, 2, \dots, N.$$

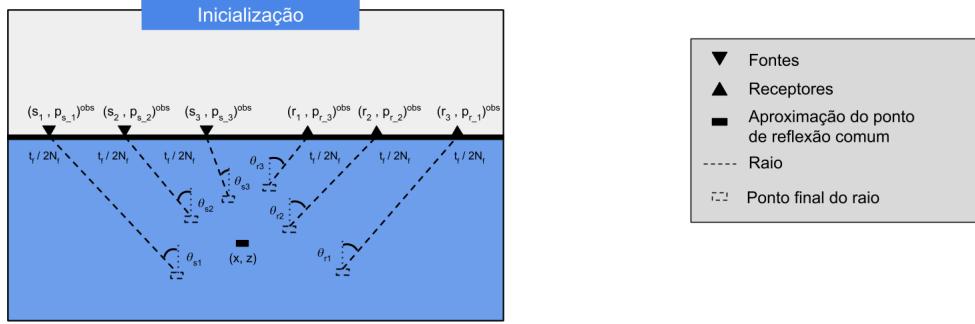


Figura 3: Esquema ilustrativo do processo de inicialização do algoritmo. No exemplo, utiliza-se uma família CRP com 3 pares fonte-receptor.

### 3 Restruturação

Nesta seção, são apresentadas as principais modificações desenvolvidas, tanto em termos de implementação quanto em termos de reformulação algorítmica. Dessa forma, as contribuições autorais são apresentadas gradualmente, partindo da implementação da função modeladora; em seguida, passando pela geração de dados sintéticos; daí apresentam-se módulos auxiliares; e então reimplementa-se o algoritmo original da técnica. Em seguida, são propostas e implementadas modificações teóricas para o problema de inversão, a fim de incorporar informações a priori sobre o modelo geológico.

#### 3.1 Paralelização do Traçamento de Raios por CPU

Pilar da técnica tomográfica, a função modeladora desfruta do fato de ser intuitivamente paralelizável: cada traçamento de raio (associado a uma fonte ou a um receptor) pode ser concebido como um PVI independente dos demais. Portanto, viabiliza-se a resolução simultânea de múltiplos problemas de valor inicial, de modo a acelerar a

modelagem.

Ao contrário do MATLAB - por onde a técnica foi originalmente implementada -, Python possui robustas bibliotecas de paralelismo gratuitas. No presente trabalho, lançou-se mão de Numba (Lam et al. [2015]), a qual permite a fácil distribuição de tarefas entre múltiplos núcleos de CPU. Com o propósito de ilustrar a significativa discrepância entre os tempos de computação requeridos para ambas metodologias (paralela e serial), apresentam-se os resultados de um simples experimento abaixo.

No experimento, foram definidas uma malha de pontos com dimensão  $400 \times 400$ , velocidade uniforme  $c(x, z) = 1 \text{ m/s}$ , e índice da posição da fonte  $(200, 400)$  - ou seja, a fonte estava no nível mais profundo do modelo. Daí, foram propagados em direção à superfície um número  $n$  crescente de raios, a saber  $n = 10, 20, \dots, 100$ , até que se alcançasse 100 segundos no Sistema de EDO's (4). Para cada conjunto de  $n$  raios, foram tomados ângulos iniciais  $\theta$  a passos uniformes no intervalo  $[\frac{\pi-1}{2}, \frac{\pi+1}{2}]$ . Ademais, fixou-se o passo de discretização de Euler para  $h = 0.01$ . A partir dessa configuração, incrementou-se  $n$  e computaram-se os PVI's em *loop*. Em experimentos separados, utilizou-se a abordagem serial e depois paralelizada em 5 núcleos.

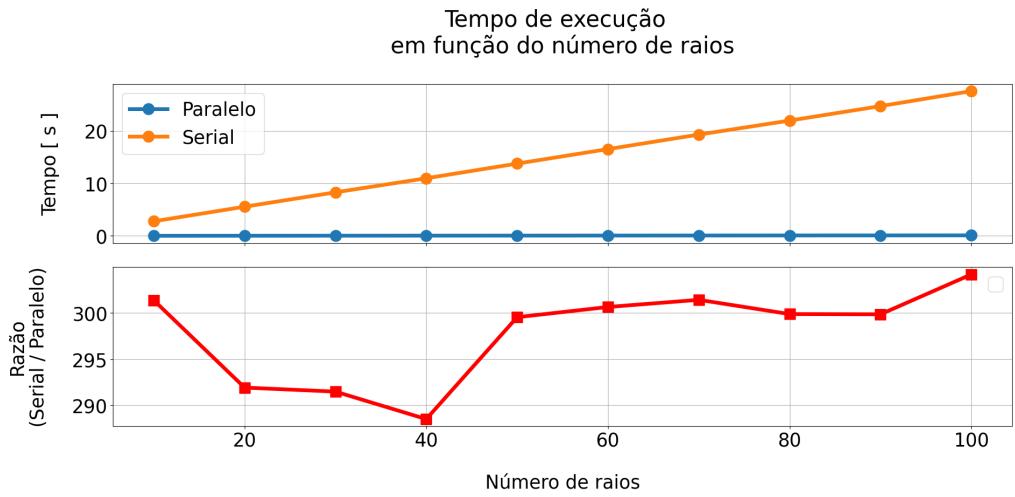


Figura 4: Gráfico exibindo o perfil do tempo de execução para a resolução serial e paralela em função do número de raios. Na figura de baixo, exibe-se o ganho relativo entre implementações.

Como demonstra a Figura 4, a distribuição de tarefas provê significativa redução do tempo de execução, uma vez que se compara o desempenho em paralelo (linha azul), e

o serial (linha laranja). Meramente lúdico, o cenário proposto pelo experimento - isto é, o traçamento de até 100 raios - ainda se demonstra simples em relação àquele trabalhado na inversão tomográfica: milhares de sistemas de raios são computados. Portanto, conclui-se a grande importância do paralelismo na Tomografia CRP.

A implementação das funções relativas ao traçamento de raios, e sua paralelização, foi feita em dois arquivos distintos: `dif_eq_solver.py` - voltado à resolução numérica do Sistema de EDO's de raios - e `ray_solver.py` - destinado a métodos auxiliares de manipulação dos raios e configuração de condições iniciais.

Enfim, também destaca-se o fato de que, durante o processo de inversão da Tomografia, o traçamento de raios é feito em simultâneo ao cálculo da matriz Jacobiana - agravando o tempo de modelagem do sistema físico para cada raio. Assim, a paralelização de ambas as etapas acarreta significativamente, na implementação final, a otimização do tempo de execução.

### 3.2 Geradores de experimentos sintéticos

Palco para validação e experimentação da Tomografia, experimentos sintéticos preconizam a geração de dados que simulem a extração de informações geofísicas. A esse respeito, a implementação original da técnica resumia em um único arquivo a geração específica - pouco adaptada ao usuário - de apenas um modelo geológico sintético. Em contrapartida, ao longo deste projeto buscou-se o desenvolvimento de um programa com mais alto nível de personalização do espaço físico e dos dados invariantes gerados.

Sendo assim, no arquivo `construct_NIP.py`, foi desenvolvido um programa para construção de interfaces refletoras - as quais simulam pontos CRP. A princípio, o código possibilita gerar curvas refletoras trigonométricas, ou retas, cada qual com um conjunto de condições iniciais arbitrárias para os raios emergentes. Sob essa lógica, a construção de dados invariantes para inversão é feita mediante aplicação de traçamento de raios sobre esses PVI's pré definidos. Na Figura 5 abaixo, um exemplo que ilustra a geração de dados observados.

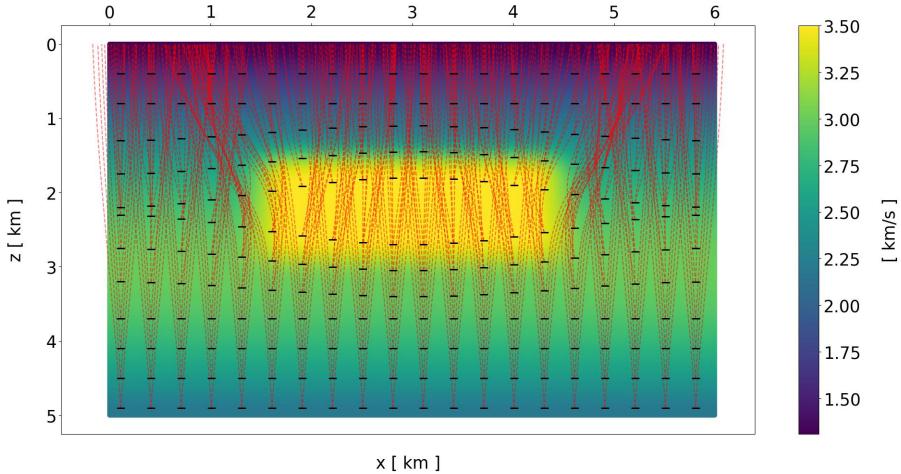


Figura 5: Exemplo de modelo físico sintético gerado a partir do programa supracitado. Nele, cada ponto refletor está associado a apenas um par de raios. Na imagem, as linhas tracejadas em vermelho representam os raios traçados a partir dos PVI's associados. Já as linhas pretas correspondem à posição real dos pontos CRP. Ao fundo, o modelo de velocidade com grande variação central.

### 3.3 Módulos auxiliares: regularização e interpolação

Originalmente, Ignácio complementa o desenvolvimento do problema de inversão com a adição de um termo de regularização sobre a função objetivo  $S$  (para mais detalhes, veja o capítulo 5, seção 3 da tese). Em suma, o termo regularizador envolve a construção de uma matriz  $R$ , invariante ao longo do processo de inversão. Visto que sua definição envolve certa arbitrariedade do usuário, destinou-se sua criação a um arquivo único, `construct_reg_mat.py`. Na implementação original, ela era extensivamente calculada no mesmo arquivo da inversão em si.

Ademais, dando continuidade à modularização dos códigos, criou-se o arquivo `construct_bspline.py`, o qual tem como objetivo a construção das bases de interpolação em duas dimensões a partir do número de nós e de seus valores fornecidos pelo usuário.

### 3.4 Reimplementação da Tomografia CRP

A partir das modificações apresentadas, foi possível reimplementar a Tomografia CRP, cujo programa se encontra no arquivo `tmg_crp.py`. Adicionalmente, no que tange ao processo de minimização, refinou-se o critério de aceitação de passo de atualização a partir da Condição de Armijo. Essencialmente, modificou-se o Passo 5 do Algoritmo da

seguinte maneira :

**Passo 5 (Condição de Armijo)** Tamanho de passo: Encontre o menor inteiro não negativo  $j$  tal que  $\alpha_k = (\frac{1}{2})^j$  satisfaça a condição

$$S(\mathbf{m}_k + \alpha_k \Delta \mathbf{m}_k) \leq S(\mathbf{m}_k) + \eta \alpha_k \nabla^T S(\mathbf{m}_k) \Delta \mathbf{m}_k ,$$

em que se fixou  $\eta = 1 \cdot 10^{-4}$  ao longo do projeto. Essa restrição enriquece o valor de decréscimo da função objetivo, a fim de evitar que sejam tomados passos  $\alpha_k$  grandes demais ou improdutivos (Friedlander et al. [2019]).

Dessa forma, foi possível comparar a performance entre o código original e o atual. Tomando como base o teste sintético apresentado no repositório da tese, registrou-se o tempo de execução e o valor da função objetivo ao final da otimização realizada ao longo de 15 iterações. Os resultados, exibidos na Tabela 1, revelam uma aceleração de 620%, aliada à obtenção de um minimizador local quase 3 vezes menor, através da nova implementação.

Original		Autoral	
Tempo (min)	Resíduo final	Tempo (min)	Resíduo final
107,47	1199,42	14,84	401,70

Tabela 1: Resultados obtidos por duas implementações distintas da Tomografia CRP

### 3.5 Aplicação de *backtracking*

Dando continuidade às modificações algorítmicas na Tomografia CRP, buscouse o aprimoramento da etapa associada à definição de  $\alpha_k$ , substituindo a mera divisão iterativa  $\alpha_k = (\frac{1}{2})^j$ . Em termos práticos, empregou-se a técnica de *backtracking* proposta por Nocedal e Wright. O algoritmo de redução de passo lança mão do conhecimento acerca de  $S(\mathbf{m}_k)$ ,  $\nabla S(\mathbf{m}_k)$  e  $S(\mathbf{m}_k + \Delta \mathbf{m}_k)$ ,  $\nabla S(\mathbf{m}_k + \Delta \mathbf{m}_k)$  para aproximar a função  $\phi(\alpha) = S(\mathbf{m}_k + \alpha \Delta \mathbf{m}_k)$  por uma cúbica.

Idealmente, o algoritmo visa alcançar  $\alpha_k = \text{argmin}\{\phi(\alpha)\}$ , via aproximação pelo mínimo global do polinômio

$$\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + \phi'_0\alpha + \phi_0 ,$$

com

$$a = \frac{2(\phi_0 - \phi_t) + \alpha_t(\phi'_0 + \phi'_t)}{\alpha_t^3} \quad \text{e} \quad b = \frac{3(\phi_t - f_0) - \alpha_t(2\phi'_0 + \phi''_t)}{\alpha_t^2} \quad ,$$

em que  $\phi_0 = \phi(0)$  e  $\phi_t = \phi(\alpha_t)$ , sendo  $\alpha_t$  a magnitude de passo anterior. Essa estratégia de busca tem como objetivo aprimorar a modelagem da função  $\phi(\alpha)$  - sobretudo para casos em que esta possui variações de sinal na curvatura.

Ao longo de 18 iterações, otimizou-se o mesmo problema abordado na subseção anterior, a fim de comparar o desempenho entre as duas estratégias de *backtracking*. Observou-se um aumento na velocidade média do algoritmo quando aplicada a minimização de cúbicas, ainda que o custo final tenha ficado aquém daquele resultante pela abordagem inicial - como exposto na Tabela 2.

Sucessivas divisões por $\frac{1}{2}$		Minimização de cúbicas	
Tempo (min)	Resíduo final	Tempo (min)	Resíduo final
19,87	261,79	15,29	5522,09

Tabela 2: Resultados obtidos por duas abordagens distintas de *backtracking*

A velocidade alcançada pela nova implementação é justificada pela redução do número de buscas por  $\alpha_k$ , visto que foram feitas, em média, cerca de 2 buscas por iteração. Em contrapartida, a estratégia original está sujeita, em muitas iterações, a performar cerca de 10 buscas de Armijo. Vale ressaltar que cada busca preconiza a rejeição de um tamanho de passo anterior, exigindo, de modo subsequente, a aplicação da função de modelagem para cada raio do sistema - lógica que acarreta no atraso da conclusão de uma iteração. Quanto à perda de acurácia, supõe-se que ela ocorre em razão do cálculo de passos pequenos demais durante a minimização das aproximações polinomiais, de modo a não gerar significativo decréscimo sobre a função objetivo. Isto posto, o problema de minimização auxiliar durante a busca por passo ainda constitui amplo palco de investigação, bem experimentação - como a possível mescla de estratégias em tempo de execução. Trabalhos futuros sobre Tomografia CRP devem abordar tais lacunas.

### 3.6 Imposição de restrições naturais

Nas subseções anteriores, partiu-se de uma aproximação inicial baseada no processo de inicialização sugerido por Ignácio. Não obstante, e a despeito do sucesso na inversão do problema anterior, a sucessiva atualização do modelo de velocidade gerou modelos não-físicos, isto é, foram construídos modelos com valores negativos. Tal fato joga luz sobre uma intrínseca fragilidade da Tomografia CRP: não existem restrições de sinal associadas ao problema de atualização.

Sob essa ótica, o problema se torna mais claro quando se submetem aproximações iniciais não homogêneas ao algoritmo. Em testes feitos tanto com a implementação original, quanto com a autoral, observou-se a estagnação do processo logo na primeira iteração. Isso ocorre pois o traçamento de raios não suporta velocidades negativas, logo, a modelagem resulta em aproximações inverossímeis que não chegam a produzir dados sintéticos que reduzam a função custo. Dessa forma, uma vez esgotado o limite de buscas de Armijo, encerram-se as iterações tomográficas.

A fim de contornar esse empecilho - sem reformular a Tomografia CRP a partir de um problema de otimização com restrições -, optou-se por uma mudança de variáveis que introduzisse, a partir de conhecimento *a priori* sobre o sistema, os vínculos necessários para assegurar a física da modelagem. Daí, pode-se definir a velocidade  $v \in \mathbb{R}_+$  em cada nó de interpolação em função de  $u \in \mathbb{R}$  tal que

$$v(u) = \alpha - (\alpha - \beta) \cdot \frac{1}{1 + u^2} \quad ,$$

onde  $\alpha$  é o maior valor de velocidade estimado para o modelo, e  $\beta$  o menor. Note que  $\lim_{u \rightarrow \pm\infty} v(u) = \alpha$ , assim como  $\lim_{u \rightarrow 0} v(u) = \beta$ . Assim, pode-se otimizar a variável  $u$  irrestritamente, de modo a garantir, naturalmente, a viabilidade dos valores de  $v$  associados.

Assim sendo, a substituição proposta suscita a adição de um termo para o cálculo do Jacobiano da Tomografia CRP (3). Pela regra da cadeia, atesta-se que

$$\frac{\partial \cdot}{\partial u} = \frac{\partial \cdot \partial v}{\partial v \partial u} = \frac{\partial \cdot}{\partial v} \cdot 2(\alpha - \beta) \frac{u}{(1 + u^2)^2} \quad .$$

Logo, é necessário, tão somente, a multiplicação das derivadas em relação à velocidade

pelo termo  $\frac{\partial v}{\partial u}$  explicitado acima. A implementação dessa versão encontra-se no arquivo `test_tmg_crp.py`, enquanto a versão atualizada do traçamento de raios, incorporando a troca de variáveis, reside em `dif_eq_solver.py`. Com o propósito de avaliar o desempenho da técnica tomográfica munida das modificações propostas, foram realizados dois testes sintéticos - cuja metodologia e cujos resultados, são descritos a seguir.

## 4 Testes sintéticos

A partir dos arquivos presentes no repositório da tese, e através do módulo de geração de experimentos sintéticos supracitado, foram realizados dois experimentos numéricos, visando à analise de performance da implementação autoral da Tomografia CRP. De modo sucinto, objetivava-se o funcionamento pleno do Algoritmo quando submetido a modelos iniciais não homogêneos, e respeitando as restrições físicas do problema de inversão. Os modelos geológicos utilizados foram: i) Modelo de Caixa e ii) Modelo de Pré-sal, apresentados a seguir na Figura 6.

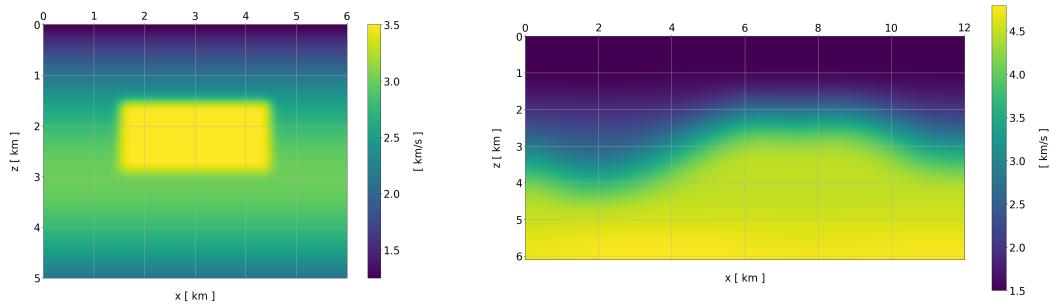


Figura 6: modelos de velocidade dos experimentos sintéticos. À esquerda, o modelo de Caixa - o mesmo apresentado por Ignácio em sua tese. À direita, o modelo de Pré-sal.

### 4.1 Metodologia

Para os experimentos, inicializou-se  $\mathbf{m}^{vel}$  a partir de uma amostragem uniforme dos valores de velocidade reais, e então aplicou-se um filtro de suavização aos modelos. Ademais, definiu-se o número máximo de iterações como 20, com até 20 buscas de Armijo. Em termos de métricas, além de avaliar a porcentagem de redução da função objetivo, e o tempo de execução ao fim do processo de inversão, também buscou-se avaliar a qualidade relativa dos pontos focais de cada par fonte-receptor, ou seja, especificamente a apro-

ximação dos pontos refletores à medida que se incrementava a profundidade (coordenada  $z$ ) associada.

Sob essa visão, para cada experimento, aplicou-se o traçamento de raios para baixo, a partir do conjunto final de dados sintéticos. Então, tomou-se como aproximação do ponto CRP o centroide dos pontos finais de cada raio. Para cada família, computou-se a máxima distância entre o refletor e os pontos finais, pois o valor obtido representa, nesta análise, o raio do círculo focal associado ao CRP. Complementarmente, tomou-se a porcentagem entre o raio de confiança e o nível de profundidade do refletor. Consequentemente, menores porcentagens indicam maior precisão (porém não necessariamente acurácia), sobretudo para níveis mais fundos - visto que esses pontos representam maior dificuldade para a inversão.

## 4.2 Resultados

**Modelo de Caixa :** Pode-se dizer que a Tomografia CRP mostrou-se estável através das modificações sugeridas, visto que todas as iterações foram feitas sem estagnação durante as buscas de Armijo ou durante a modelagem por raios. Para esse experimento, os resultados referentes ao tempo de execução, e à redução da função objetivo em relação ao valor inicial, foram registrados na Tabela (3).

Tempo (min)	Redução da função objetivo
23,43	50,77%

Tabela 3: Resultados obtidos pela inversão do Modelo de Caixa

No que tange aos resultados acerca dos raios de confiança, recorreu-se à análise do histograma representado na Figura 7. Torna-se evidente a alta precisão focal dos raios, uma vez que a maioria das famílias apresentou região de factibilidade com raio menor que 2% da profundidade do refletor correspondente - isto é, para refletores na faixa dos 5 km, o raio de confiança foi menor que 100 m.

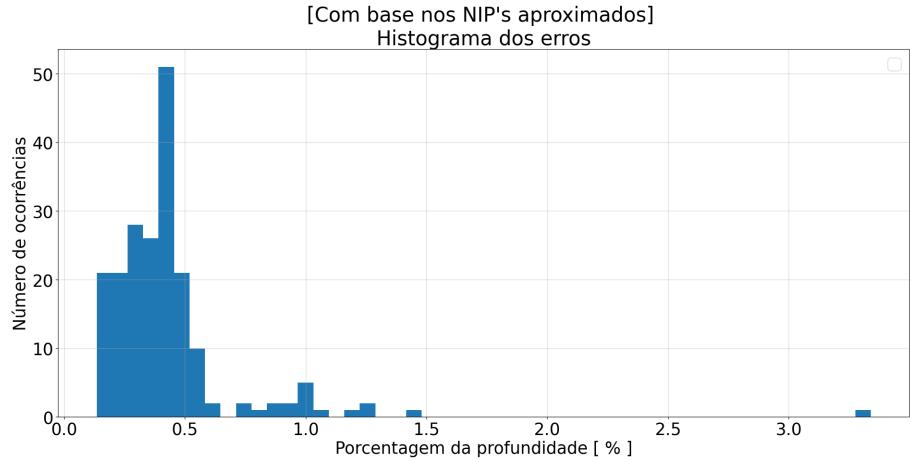


Figura 7: Associado ao Modelo da Caixa, histograma da concentração das famílias em função da porcentagem da profundidade do refletor associado.

Por fim na Figura 8, nota-se que o modelo de velocidade final alcançado preserva, superficialmente, o perfil geral do Modelo de Caixa - a saber, variação periódica em toda sua extensão, a não ser pela porção retangular central de alta velocidade. Contudo, nota-se o surgimento de um artefato nas últimas faixas de profundidade (entre 3,5 km até 5 km ao longo do eixo  $z$ ). A presença dessa anomalia sugere mais atenção ao desenvolvimento do problema de otimização.

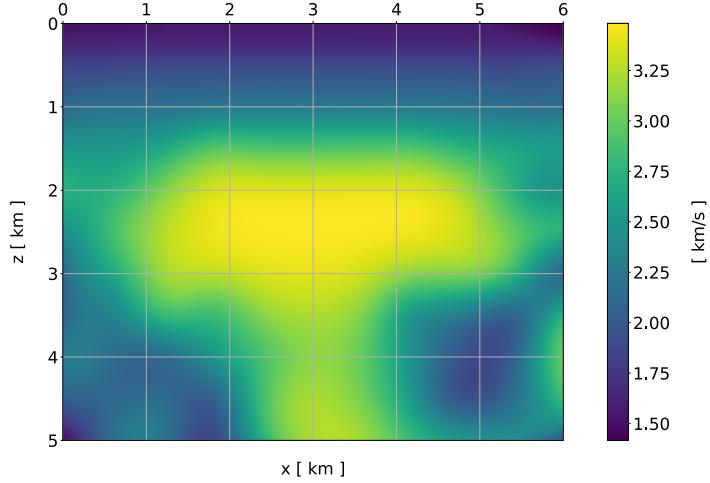


Figura 8: Modelo de velocidade final alcançado pela inversão do problema da Caixa.

**Modelo de Pré-sal:** Quanto ao segundo experimento sintético, novamente atesta-se a estabilidade do algoritmo reformulado a partir da inicialização não homogênea. Ademais, constatou-se uma redução da função objetivo similar ao caso anterior (aproximadamente

50%) - apesar do tempo de execução ter caído pela metade, como indicado na Tabela 4). A maior rapidez de convergência está intimamente atrelada ao menor número de buscas de Armijo realizadas ao longo das iterações.

Tempo (min)	Redução da função objetivo
11,84	48,73%

Tabela 4: Resultados obtidos pela inversão do Modelo de Pré-sal

Similarmente ao experimento anterior, registrou-se elevada precisão focal - como ilustrado pelo histograma da Figura 9. A maioria das famílias apresentou raios de confiança menores que 1% da profundidade respectiva ao refletor.

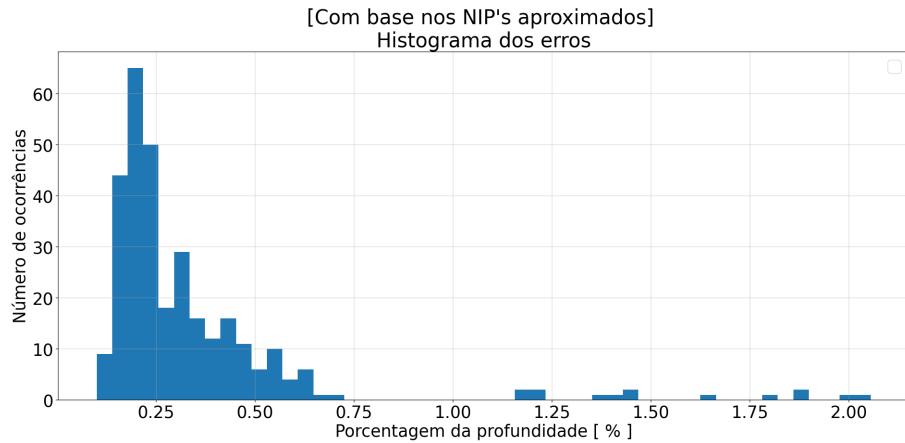


Figura 9: Associado ao Modelo do Pré-sal, histograma da concentração das famílias em função da porcentagem da profundidade do refletor associado.

Enfim na Figura 10, observa-se alta fidelidade entre o modelo de velocidade real e o alcançado pelas iterações tomográficas. Contudo, vale destacar, novamente, a presença de um artefato inesperado nos níveis mais profundos (entre 4,5 km e 6 km ao longo do eixo z).

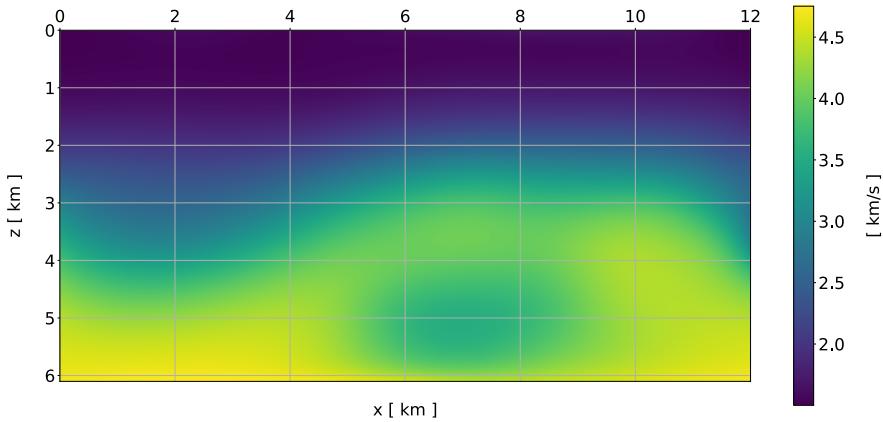


Figura 10: Modelo de velocidade final alcançado pela inversão do problema do Pré-sal.

## 5 Conclusão

O presente trabalho, acredita-se, cumpriu seu objetivo central de realizar um estudo exploratório inicial da Tomografia CRP, evidenciando tanto suas potencialidades quanto suas fragilidades estruturais. A nova implementação desenvolvida ao longo do projeto demonstrou-se significativamente superior à versão original — tanto em termos de desempenho de otimização quanto em tempo de execução — validando a relevância das intervenções propostas.

Primeiramente, a reestruturação completa do código, com paralelização explícita por CPU e modularização em Python, permitiu reduzir substancialmente o tempo de execução e aumentar a estabilidade da técnica. A nova implementação possibilitou, inclusive, avaliar casos inviáveis na formulação original, como inicializações não homogêneas, suscitando limitações e discussões antes inacessíveis à análise.

Em segundo lugar, este estudo demonstrou que a Tomografia CRP, em sua formulação original, carece de mecanismos explícitos de controle físico sobre o modelo de velocidades. A introdução de uma mudança de variáveis que garante positividade e agrupa informação a priori ao modelo constituiu um passo fundamental para tornar o processo de otimização fisicamente consistente, eliminando a geração de modelos não-físicos e prevenindo a irremediável estagnação do algoritmo.

Terceiro, a substituição do método tradicional de busca de passo por uma

estratégia baseada na minimização de cúbicas trouxe ganhos relevantes de desempenho, reduzindo drasticamente o número de avaliações da função de modelagem ao longo da otimização. Ao mesmo tempo, o comportamento observado — passos excessivamente pequenos e ineficientes — evidencia, sobretudo, a importância da formulação de uma estratégia de *backtracking* robusta e adaptativa às diferentes etapas da otimização - possivelmente pela mescla de aproximações polinomiais durante a busca de Armijo. Outra possibilidade, tal como Ignácio propunha, a resolução do subproblema de minimização por heurísticas de otimização global.

Por fim, os experimentos sintéticos demonstraram que, apesar da boa capacidade de focalização dos pontos refletores e da redução consistente da função objetivo, artefatos persistem nos níveis mais profundos da malha. Esse comportamento indica que a técnica, embora promissora, ainda apresenta limitações vinculadas tanto à formulação matemática quanto ao condicionamento do problema inverso. Assim, conclui-se que o método ainda pode ser amplamente aprimorado, seja por: regularizações mais sofisticadas sobre a função objetivo - aqui não abordadas -; esquemas de otimização que desfrutem de outras aproximações para a matriz Hessiana (como BFGS); algoritmos para solução de sistemas lineares que superem os altos custos proporcionados pela SVD. Além, é claro, da transposição para arquiteturas GPU - catalisando o desempenho computacional.

No conjunto, este projeto fornece não apenas uma implementação mais confiável e eficiente da Tomografia CRP, mas também um diagnóstico claro das principais lacunas teóricas e computacionais que precisam ser abordadas em trabalhos subsequentes. Dessa forma, estabelece-se um primeiro passo em direção à sofisticação e à compreensão da resolução de problemas de inversão na Geofísica Computacional.

## Referências

- F. Billette and G Lambaré. Velocity macro-model estimation from seismic reflection data by stereotomography. *Geophysical Journal International*, 1998.
- V. Cerveny. *Seismic ray theory*. Cambridge university press, 2005.
- DeBoor. *A practical guide to splines*. Springer, 1978.
- Ana Friedlander, Nataša Krejić, and Nataša Krklec Jerinkić. *Lectures on Fundamentals of Numerical Optimization*. University of Novi Sad Faculty of Sciences, 2019.
- Gustavo Barroso Dias Ignácio. *Tomografia CRP*. PhD thesis, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2021.
- S. K. Lam, A. Pitrou, and S. Seibert. A technical paper introducing the numba jit compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 2015.
- R.J LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, 2007.
- Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, 2006.