



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM GEOFÍSICA

RELATÓRIO DE PROCESSAMENTO SÍSMICO

**Modelagem e Processamento Sísmico de Dado
Sintético.**

MURILO SANTIAGO VALE RODRIGUES

Belém
2019

MURILO SANTIAGO VALE RODRIGUES

**Modelagem e Processamento Sísmico de Dado
Sintético.**

Relatório de ativades relativo à disciplina de Processamento de Sísmico.

Orientador: Prof. Dr. João Carlos Ribeiro Cruz

Belém
2019

Sumário

1	INTRODUÇÃO	1
2	MODELAGEM E PROCESSAMENTO SÍSMICO	2
2.1	MODELAGEM DA ESTRUTURA GEOLÓGICA	2
2.2	GEOMETRIA E PARÂMETROS DE AQUISIÇÃO	7
2.3	TRANSFORMADA DE RADON LINEAR	11
2.4	TRANSFORMADA F-K	21
2.5	FILTROS DE CORTE NA FREQUÊNCIA	22
2.6	COBERTURA MÁXIMA DOS CDP'S	37
2.7	ANÁLISE DE VELOCIDADE E CORREÇÃO NMO	41
2.8	EMPILHAMENTO	46
3	MIGRAÇÃO	48
3.1	MIGRAÇÃO KIRCHHOFF NO TEMPO	48
3.2	MIGRAÇÃO DIFERENÇAS FINITAS EM PROFUNDIDADE	55
3.3	OUTRAS METODOLOGIAS DE MIGRAÇÃO EM PROFUNDIDADE .	60
	REFERÊNCIAS	64
	APÊNDICES	66
	APÊNDICES	67
A	SHELL SCRIPT MODELAGEM TRIMODEL	67
B	SHELL SCRIPT MODELO DE VELOCIDADE EXATO	69
C	SHELL SCRIPT AQUISIÇÃO TRISEIS	72
D	SHELL SCRIPT GEOMETRIA	74
E	SHELL SCRIPT PLOT AFASTAMENTO MÍNIMO . .	75
F	SCRIPT MATLAB TRANSFORMADA DE RADON LI- NEAR	76
G	SCRIPT MATLAB FILTRAGEM FK	79
H	SHELL SCRIPT MÁXIMA COBERTURA	87
I	SHELL SCRIPT PLOT DOS CDP'S	89
J	SHELL SCRIPT RUÍDO NA SEÇÃO SÍSMICA	90
K	SHELL SCRIPT ANÁLISE DE VELOCIDADE ITERA- TIVA	91
L	SHELL SCRIPT MODELO DE VELOCIDADE	102
M	SHELL SCRIPT CORREÇÃO NMO	105
N	SHELL SCRIPT EMPILHAMENTO	107

O	SHELL SCRIPT MIGRAÇÃO KIRCHHOFF	108
P	SHELL SCRIPT MIGRAÇÃO DIFERENÇAS FINITAS	109
Q	SHELL SCRIPT OUTRAS MIGRAÇÕES	110

1 INTRODUÇÃO

Este relatório tem como objetivo geral apresentar as etapas de modelagem e processamento sísmico de um dado sísmico sintético utilizando os softwares Seismic Unix (SU) e Matlab.

Apresentaremos um fluxograma (figura 1.1) do processamento de dados sísmicos sintéticos, com a aplicação do método de imageamento sísmico CDP, através do software livre Seismic Unix.

O Seismic Unix é um software de processamento de dados sísmicos de código aberto desenvolvido pelo *Wave Field Research Center* (CWP) da Escola de Minas do Colorado, usado principalmente para ensino e pesquisa em exploração sísmica.

Desta forma deseja-se obter uma melhor imagem da subsuperfície através das técnicas empregadas ao decorrer deste trabalho. A seguir, apresenta-se um fluxograma típico de processamento de um dado sísmico. Algumas etapas do processamento são opcionais, e a sequência pode ser modificada para se adequar ao trabalho desejado.

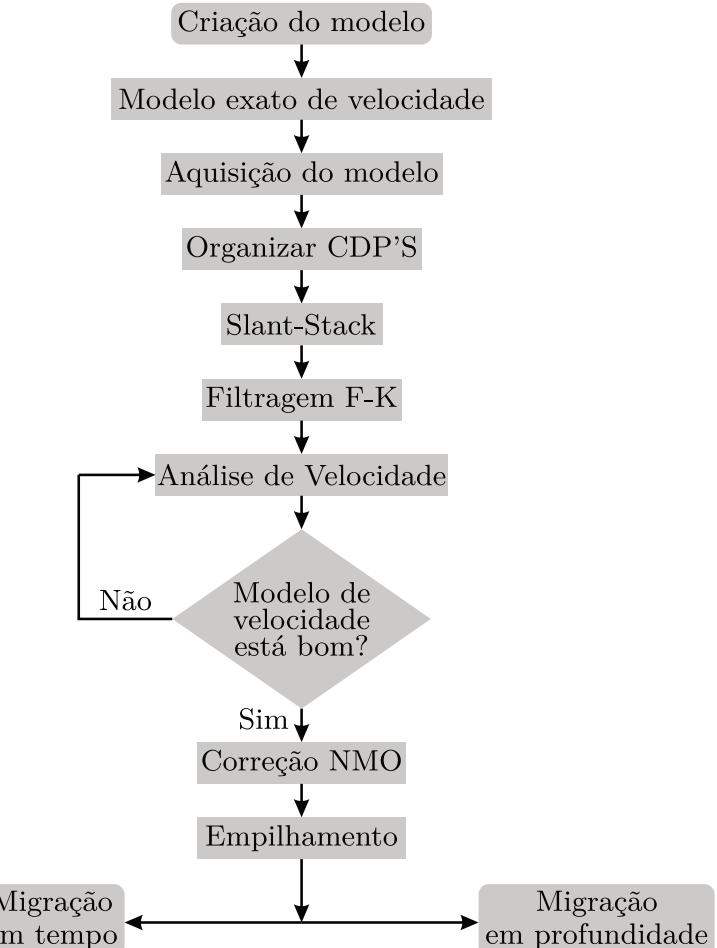


Figura 1.1: Fluxograma do processamento adotado neste trabalho.

2 MODELAGEM E PROCESSAMENTO SÍSMICO

O método sísmico de reflexão, consiste, basicamente, em obter informações da subsuperfície através da propagação de ondas produzidas por fontes artificiais em superfície e o posterior registro dessas ondas em receptores (geofones), que estão em superfície. As ondas geradas pelas fontes propagam-se na subsuperfície, sendo transmitidas, refletidas e difratadas por interfaces geológicas que delimitam as camadas de rochas de diferentes propriedades físicas. As ondas registradas nos receptores são provenientes das reflexões das interfaces, o sinal registrado pelos receptores possuem informações sob as ondas propagadas e suas mudanças conforme o trajeto do percurso da onda.

O método sísmico baseia-se na propagação do campo de onda, ou seja, na evolução temporal e espacial da frente de onda através de um meio com propriedades distintas e específicas, nas quais afetam diretamente a energia sísmica em propagação, fazendo-se necessário o estudo da equação da onda para compreender os fenômenos físicos envolvidos.

O campo de onda propagado é representado através de raios, que são retas perpendiculares as frentes de ondas (esféricas) com origem na fonte sísmica (perturbação).

O que se deseja no experimento sísmico é estimativa do tempo de trânsito (τ) ao longo de uma trajetória qualquer referente a pares fonte-receptor, este pode ser calculado baseada no traçamento de raios e na lei de Snell para cada interface atravessada. Neste cálculo é empregada a equação iconal (equação 2.1). Esta equação depende do conhecimento da distribuição das velocidades intervalares do modelo.

$$(\nabla\tau)^2 = \frac{1}{v^2} \quad (2.1)$$

2.1 MODELAGEM DA ESTRUTURA GEOLÓGICA

O objetivo da modelagem de uma seção sísmica é a construção de um modelo que represente a subsuperfície de forma coerente geologicamente. A modelagem sísmica pode ser feita de forma direta e inversa a primeira é realizada quando se parte de um modelo geológico “a priori” onde conhecendo os parâmetros (velocidade, densidade), é gerada a resposta da energia de amplitude das ondas propagadas sob a condição da geometria das interfaces e camadas, registradas no sismograma sintético. Já na modelagem inversa, tem-se a resposta sísmica da subsuperfície e a partir da resposta, tenta-se então estimar os parâmetros sísmicos para construir um modelo geológico conciso a essas propriedades estimadas.

A modelagem direta, ou seja, o processo através do qual um modelo geológico de subsuperfície, em uma, duas ou três dimensões, é transformado em um registro sísmico sintético de dimensão correspondente, foi primeiramente usado por exploracionistas na

década de 50 (Edwards, 1988).

A modelagem sísmica consiste de um sistema de equações diferenciais parciais (geralmente equações da onda acústicas, elásticas, visco-elásticas, etc.) acompanhadas das condições de contorno (comportamento nas interfaces e bordas do modelo) e condições iniciais (caracterização da emissão de energia pela fonte, tempo de propagação requerido, etc.). Na modelagem acústica na ausência de fontes internas, o sistema de equações diferenciais que expressa a resposta de um modelo geológico a um campo de ondas incidente, é constituído de equações da onda do tipo:

$$\nabla^2 P(\mathbf{x}, t) - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} P(\mathbf{x}, t) = 0, \quad (2.2)$$

O pacote Seismic Unix (SU) oferece programas de modelagem e processamento de dados sísmicos. Para a modelagem da estrutura geológica foi usada o programa *trimodel* do pacote SU, esta rotina se baseia na triangulação de Delaunay. Este método sofisticado de geração de gráficos e imagens digitais, em duas ou três dimensões, é amplamente utilizado para modelar a superfície de objetos de diferentes complexidades. Foi desenvolvido em 1934 pelo matemático russo Boris Nikolaevich, e consiste em representar o objeto através de uma malha de triângulos que cumprem a condição de Delaunay: “O interior da circunferência que circunscreve cada triângulo deve ser vazia” (Hale e Cohen, 1991).

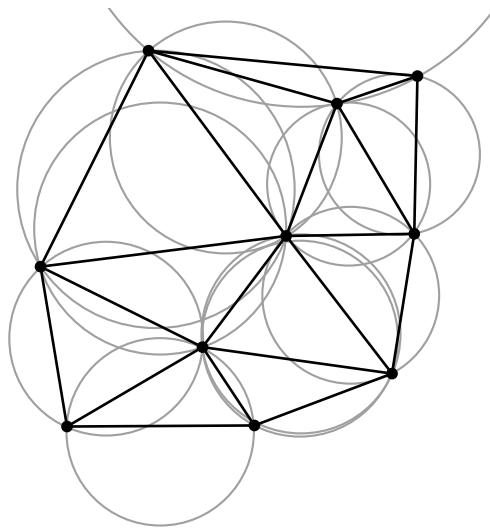


Figura 2.1: Triangulação de Delaunay no plano com as circunferências visíveis. Fonte: Wikipédia

O programa *trimodel* do SU cria um modelo triangularizado, como mostra a figura (2.2) a partir do modelo de velocidade da figura (2.3). A velocidade é introduzida na forma de vagarosidade ($sloth = 1/v$), o método realiza o traçamento dos raios baseado na equação iconal (Forel et al., 2005). A vagarosidade ao quadrado das regiões (triângulos) é determinada pela equação (2.3), onde o usuário define como é a variação da velocidade dentro de cada camada, na forma:

$$s(x, z) = s_0 + (x - x_0) \frac{ds}{dx} + (z - z_0) \frac{ds}{dz} \quad (2.3)$$

Cada par de x-z ainda é um ponto em uma camada. Na equação (2.3) x , x_0 , z e z_0 são os pontos finais e iniciais na horizontal e na vertical onde ocorrerá a variação da velocidade, esta variação terá um gradiente na direção x correspondente a ds/dx e na direção z de magnitude ds/dz .

A descrição da subsuperfície requer uma representação matemática (modelagem numérica ou modelagem sísmica), as ondas sísmicas se propagação com diferentes velocidades em subsuperfície em função dos diferentes tipos de rochas em subsuperfície. A tabela 2.1 apresenta os valores utilizados na construção do modelo de velocidade do modelo de camadas curvas (ver figura 2.3).

Camada	Velocidade da onda sísmica (m/s)	Vagarosidade (m/s) $^{-1}$
1	1600	0.39
2	1800	0.31
3	1900	0.28
4	2000	0.25
5	1350	0.55
6	1500	0.44
7	2500	0.16

Tabela 2.1: Tabela de valores de vagarosidade e velocidade do modelo de camadas curvas.

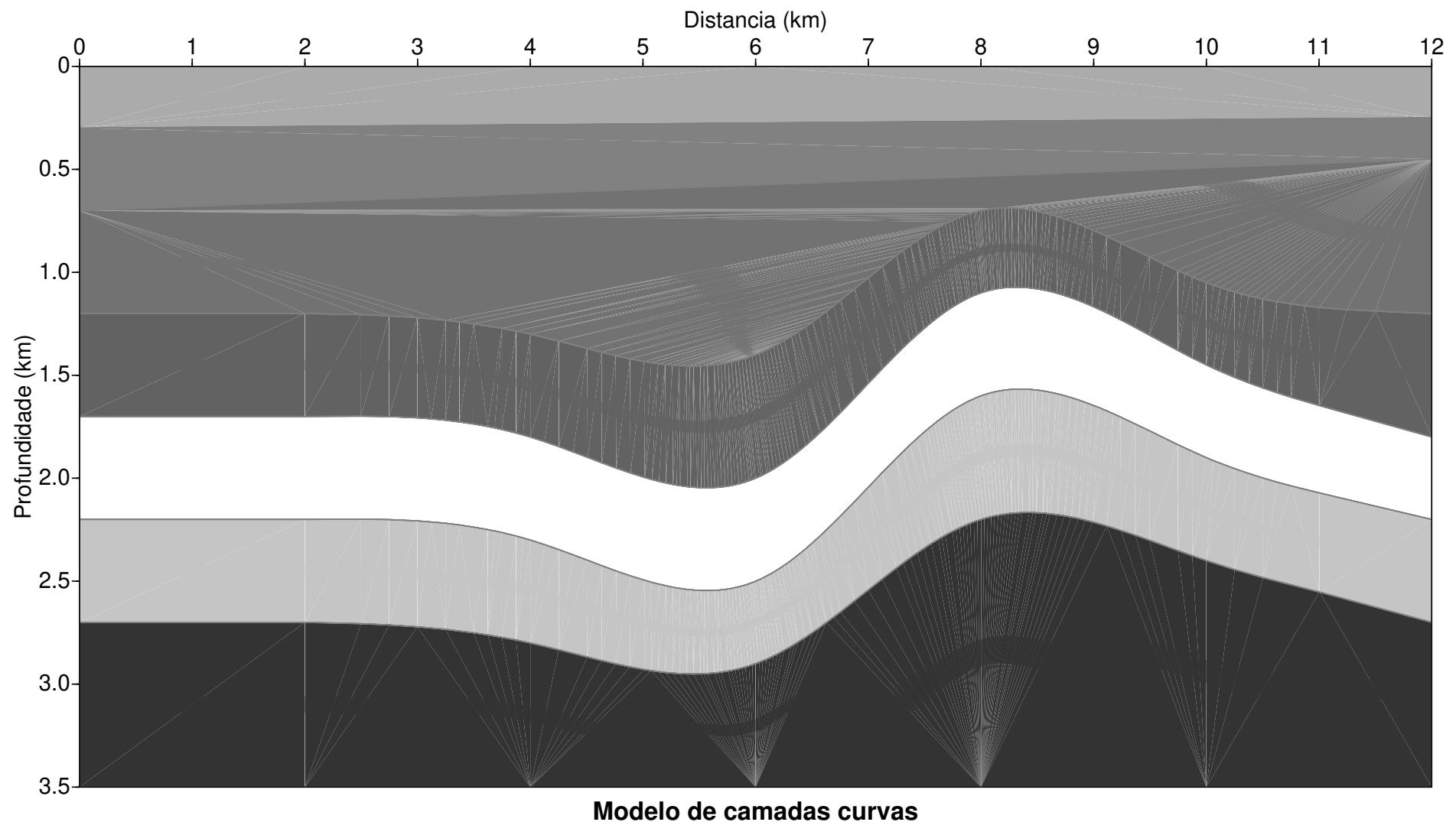


Figura 2.2: Modelo 2D de camadas curvas.

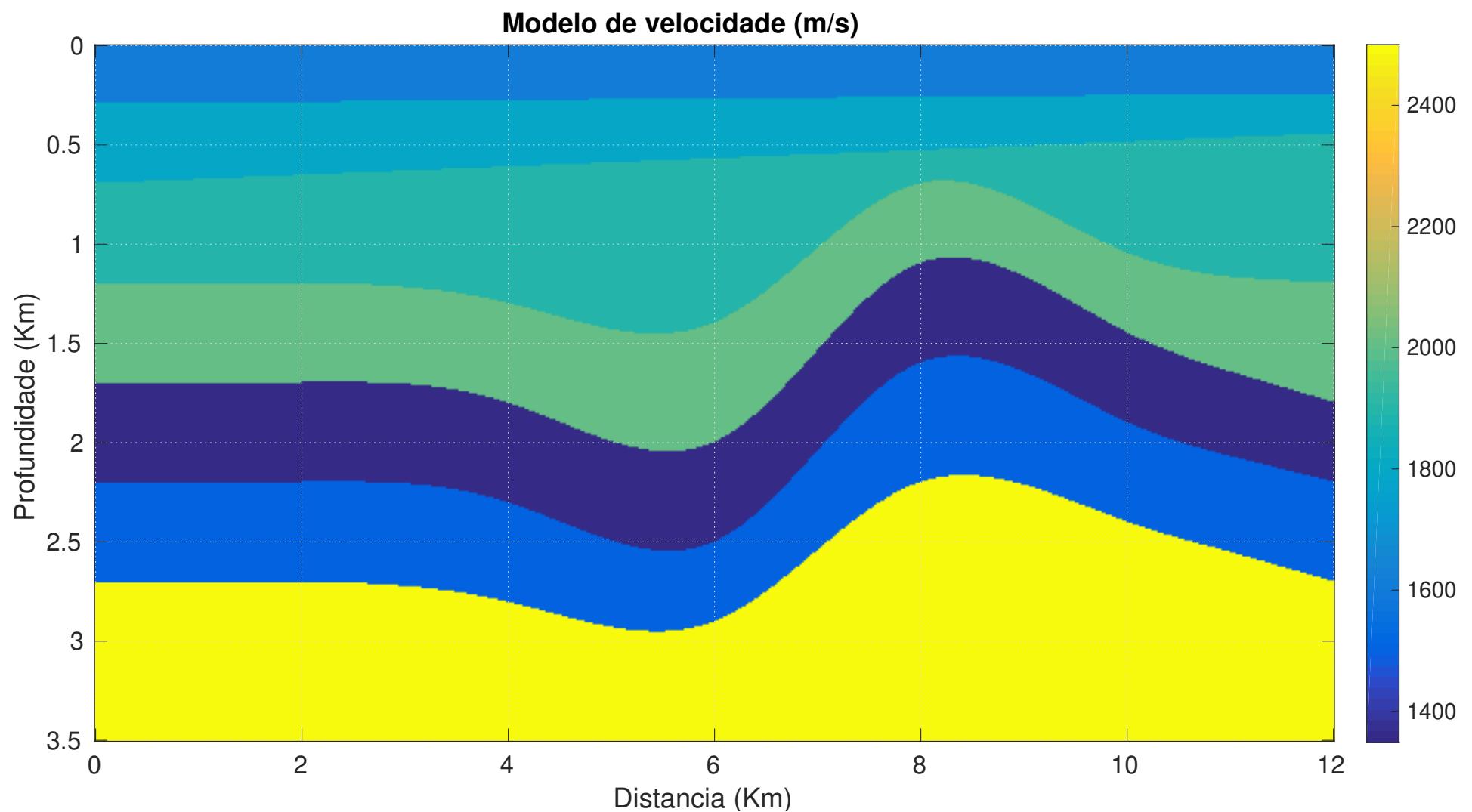


Figura 2.3: Modelo 2D de velocidade das camadas referente a Figura 2.2.

A escolha de um fluxo de processamento a ser utilizado não é única para todo e qualquer experimento sísmico de reflexão. As etapas deste fluxo devem ser criteriosamente analisados de acordo com o objetivo desejado. Desta maneira, as etapas de um fluxo de processamento convencional são completamente diferentes das etapas de um processamento sísmico visando análise da variação da amplitude com o offset (AVO), por exemplo. Sendo assim, o fluxo a ser seguido dependerá da qualidade do dado adquirido em levantamento, das ferramentas, disponíveis (software e hardware), da experiência de quem processa o dado, do tempo disponível para o processamento das imagens e do objetivo que espera ser alcançado. (Soares, 2011)

2.2 GEOMETRIA E PARÂMETROS DE AQUISIÇÃO

O termo Geometria é utilizado para designar a fase do processamento na qual são inseridas as informações a respeito de localização de cada fonte (tiro) e cada receptor (arranjo de hidrofones ou geofones), através de suas coordenadas. Essas informações serão inseridas no header dos traços sísmicos, permitindo que todas as etapas posteriores possam ser realizadas.

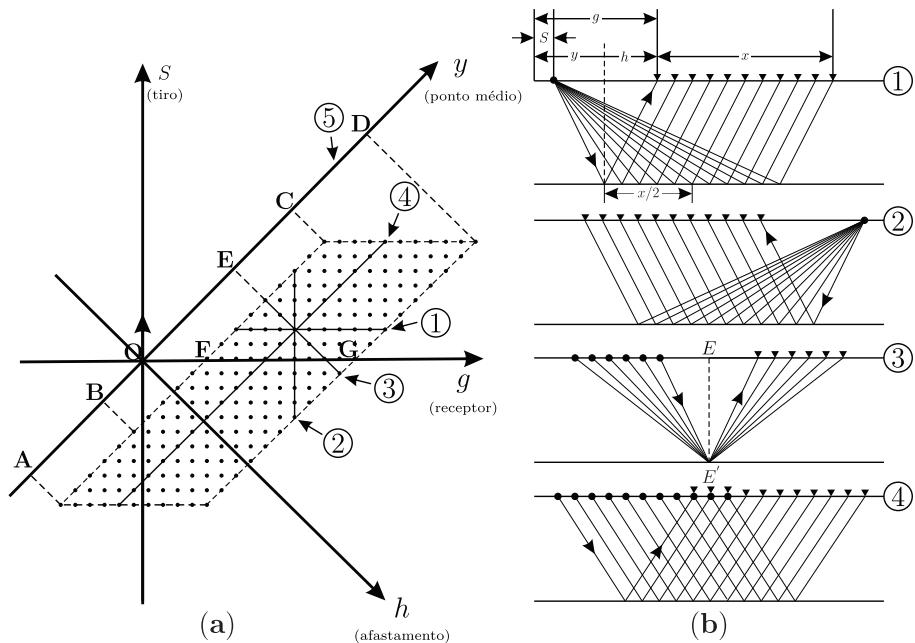


Figura 2.4: Geometria de aquisição, para vários tiros com a orientação cartesiana para obter uma melhor visualização das famílias comuns. Um dado sísmico pode ser organizado em família de tiro comum (1); receptor comum (2); ponto médio comum (3); e (4) afastamento comum (offset comum). Figura adaptada de (Yilmaz e Doherty, 2000)

Ponto comum em profundidade. Nome de uma técnica de levantamento de reflexão sísmica idealizada por W. Harry Mayne, em 1962, na qual os pontos de subsuperfície são registrados, redundantemente, com diferentes distâncias fonte-receptor. Esta técnica de aquisição trás a vantagem de se poder aumentar a razão sinal-ruído através da soma dos

traços com o mesmo CMP, designada por *stack*. Quando os traços com o mesmo CMP são agrupados formam um *CMP gather*.

Para se obter um sismograma sintético a partir de uma modelagem sísmica é preciso definir, inicialmente, a geometria de aquisição de dados e os seus parâmetros de aquisição. Neste sistema devem ser estabelecidas: a quantidade de receptores, a distância entre a fonte e o primeiro receptor e a distância entre os demais receptores, conforme a tabela 2.2.

As informações a respeito dos parâmetros de aquisição são importantes para todas as etapas de processamento do dado. Utilizamos o programa *triseis* do *Seismic Unix* (Anexo B) que tem por objetivo gerar os traços sísmicos referente a resposta do modelo de camadas curvas. O programa se baseia na teoria de raios e gera os sismogramas sintéticos de feixes Gaussianos, a partir de um modelo triangularizado preenchido (triângulos) com os valores de vagarosidade. Os parâmetros requeridos pelo programa são:

- xs: coordenadas da fonte
- zs: profundidade da fonte em superfície
- xg: coordenada dos receptores em superfície
- zg: profundidade dos receptores

A operação do programa funciona dentro de três loops sucessivos. Isto ocorre de forma que o *i-loop* referente às posições da fonte depende da conclusão do *j-loop* que descreve cada posição de receptor, que por sua vez, depende de *k-loop*, que mapeia cada refletor separadamente. Portanto, todos os raios referente ao primeiro refletor (localizados na interface 2), são armazenados nos receptores, para o primeiro tiro. Em seguida todos os raios varrem o segundo refletor (interface 3) e são armazenados em cada receptor novamente ainda para o primeiro tiro. Quando todos os refletores são mapeados é que ocorre o loop da posição da fonte varrendo a dimensão do modelo.

O arranjo utilizado “split-spread” simétrico figura 2.5 possui o mesmo número de receptores (geofones) equiespaçados linearmente em ambos os lados da coordenada de tiro. A distância entre a fonte e o primeiro receptor é conhecida como afastamento mínimo. A figura 2.6 mostra o afastamento mínimo entre fonte e receptor.

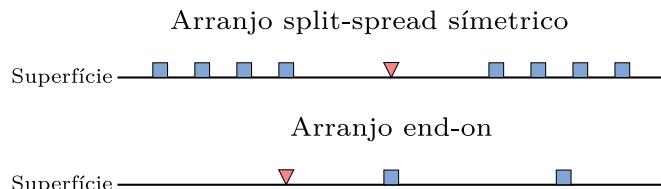


Figura 2.5: Arranjo split spread simétrico e end-on, o arranjo split spread em relação a posição da fonte fornece cobertura simétrica sobre a fonte. Em vermelho a representação da fonte e em azul os receptores (geofones).

Após a aquisição do dado sísmico é necessário carregar e organizar a geometria de aquisição, o SU permite adicionar e/ou modificar o header pelo programa *suchw* (ver Anexo). O *suchw* calcula valores de uma chave do header, segundo a equação (2.4), utilizando os valores de chaves pré-existentes (*offset*, *cdp*). Neste programa carregamos a chave *CDP* e organizamos os CDP's.

$$key3 = \frac{(a + b * key1 + c * key2)}{d} \quad (2.4)$$

Chaves	Descrição	Valores
trac1	Contagem sequêncial dos traços.	1 18528 (1 - 18528)
tracf	Número do receptor.	1 96 (1 - 96)
fldr	Número do tiro.	1 193 (1 - 193)
sx	Posição do tiro.	0 9600 (0 - 9600)
gx	Posição do receptor	-1175 10800 (-1175 - 10800)
offset	Afastamento fonte – receptor.	-1175 1200 (-1175 - 1200)
cdp	Ponto comum em profundidade.	1 10788 (1 - 10788)
trid	Identificação do tipo de traço sísmico.	1
dt	Intervalo de amostragem.	4000
ns	Número de amostras.	1200

Tabela 2.2: Principais chaves do header de um arquivo SU, de acordo com o programa *surange*. Estas informação são referentes ao arranjo split-spread.

Chaves	Descrição	Valores
trac1	Contagem sequêncial dos traços.	1 18528 (1 - 18528)
tracf	Número do receptor.	1 96 (1 - 96)
fldr	Número do tiro.	1 193 (1 - 193)
sx	Posição do tiro.	0 9600 (0 - 9600)
gx	Posição do receptor	50 14400 (50 - 14400)
offset	Afastamento fonte – receptor.	50 4800 (50 - 4800)
cdp	Ponto comum em profundidade.	25 12000 (25 - 12000)
trid	Identificação do tipo de traço sísmico.	1
dt	Intervalo de amostragem.	4000
ns	Número de amostras.	1200

Tabela 2.3: Principais chaves do header de um arquivo SU, de acordo com o programa *surange*. Estas informação são relativas ao arranjo end-on.

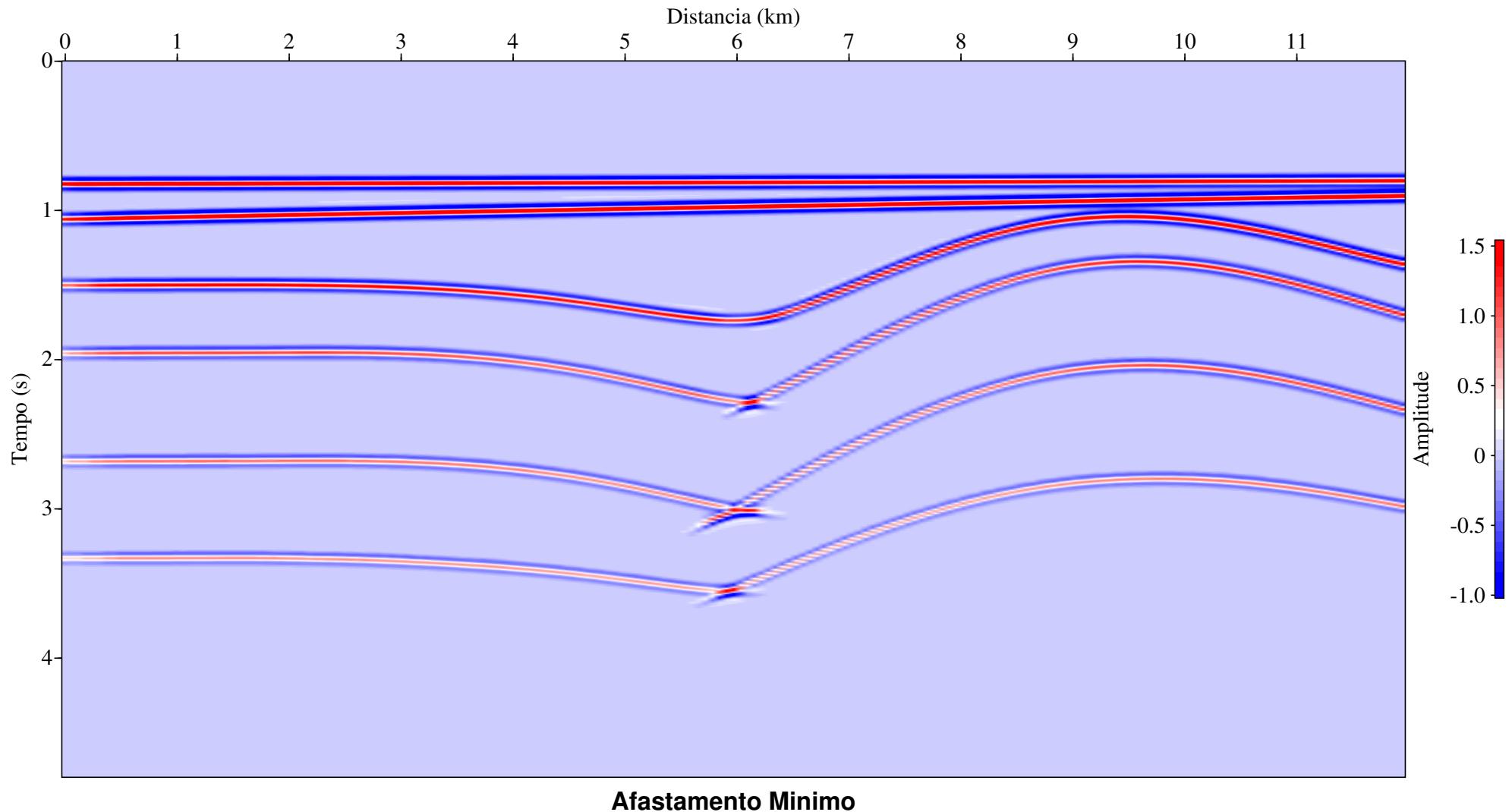


Figura 2.6: Afastamento mínimo entre fonte e receptor.

2.3 TRANSFORMADA DE RADON LINEAR

O nome da técnica (Radon) provém do matemático austríaco Johann Radon, responsável, em 1917, pela implementação dos fundamentos matemáticos para a reconstituição de imagens tomográficas. A transformada de Radon Linear (chamada também de transformada ($\tau - p$) ou “Slant Stack - empilhamento oblíquo”): Consiste de uma soma ao longo de trajetórias (retas) aplicada ao dado sísmico. A metodologia utilizada nesse trabalho para realizar a transformada de Radon linear, está no domínio da frequência.

Agora examinamos os aspectos físicos da construção de uma família de pilha inclinada “slant-stack gather”. Cada traço nesta família representa uma onda plana que se propaga em um determinado ângulo da vertical. Na realidade, quando uma fonte de dinamite explode, a energia se propaga em todos os ângulos (Figura 2.7). A energia refletida chega a diferentes grupos de receptores em ângulos diferentes devido ao deslocamento entre os locais da fonte e do receptor. Quanto mais afastada ou mais rasa a interface refletora, mais oblíquo o ângulo da próxima frente de onda (Yilmaz e Doherty, 2000).

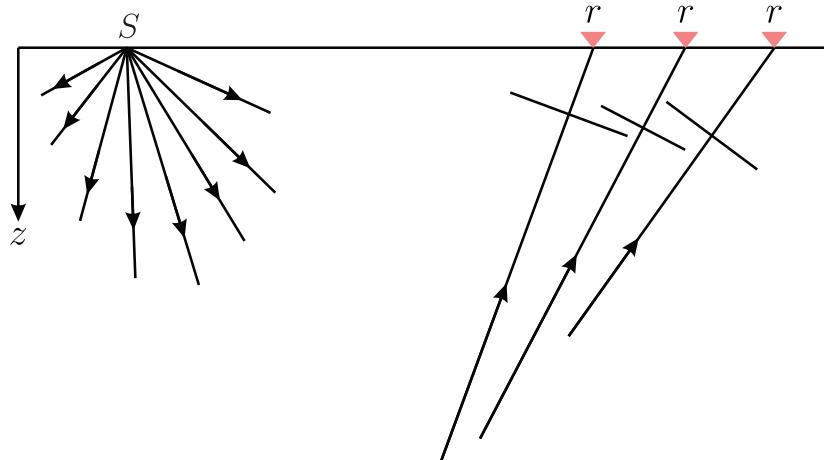


Figura 2.7: Uma fonte sísmica gera ondas que se propagam em todas as direções, as ondas que se propagam em diferentes direções são registradas em diferentes locais do receptor.

Como um auxílio na definição de um esquema para a construção da família de pilha inclinada, primeiro considere como as ondas planas podem ser geradas. Imagine uma linha de fontes pontuais, como mostra a Figura 2.8. Suponha que essa linha de fontes seja ativada para que todos os pontos da linha sejam excitados simultaneamente e cada ponto gere um campo de ondas esférico.

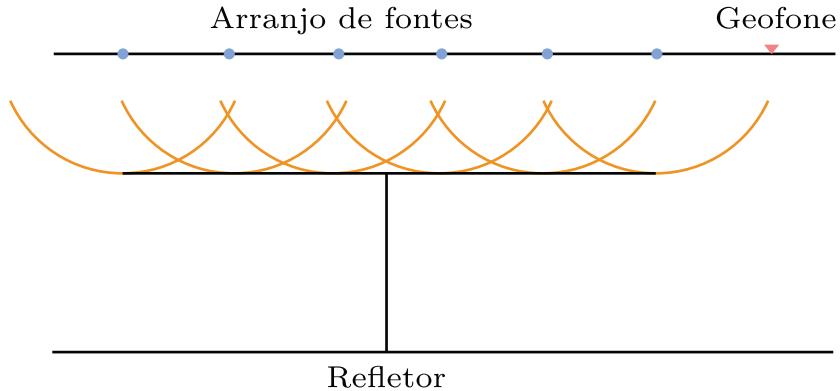


Figura 2.8: Uma fonte sísmica gera ondas que se propagam em todas as direções, as ondas que se propagam em diferentes direções são registradas em diferentes locais do receptor.

A alguma distância da superfície, as frentes de onda esféricas se sobrepõem e resultam em uma onda plana que viaja verticalmente para baixo. Essa onda plana reflete a partir de uma interface e é gravada por um receptor na superfície. (Na verdade, existem tipos de fontes, como Geoflex e Primacord, que aproximam as fontes de linhas curtas.)

Em vez de uma onda plana que se propaga verticalmente, uma onda plana que viaja no ângulo desejado a partir da vertical pode ser gerada usando a mesma linha de fontes pontuais, como ilustrado na Figura 2.9. Para fazer isso, as fontes pontuais devem ser ativadas sucessivamente, começando em uma extremidade da linha com um atraso de tempo igual entre elas. Quando uma fonte pontual específica é ativada, a frente de onda gerada a partir da localização da fonte anterior já terá percorrido uma certa distância para a Terra. Quando todas as frentes de onda esféricas geradas pelas várias fontes se sobrepõem, o resultado é um plano inclinado em frente de onda. Essa onda plana se propaga, reflete a partir de uma interface e é gravada por um receptor na superfície.

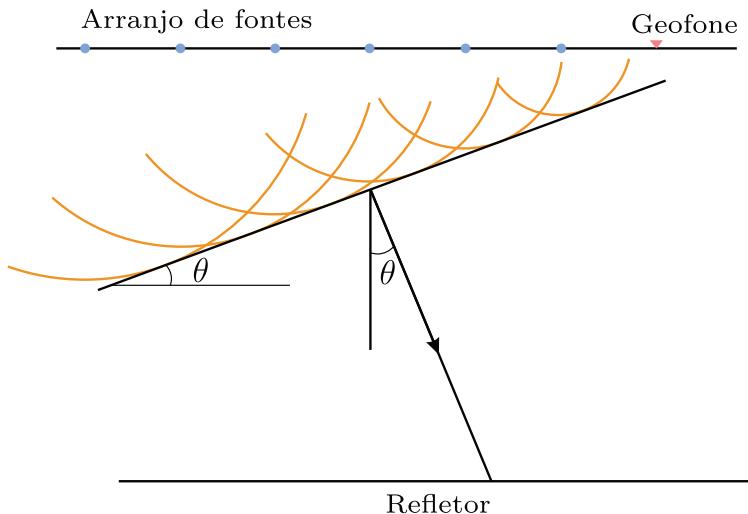


Figura 2.9: Uma onda plana que se propaga no ângulo θ da vertical é gerada utilizando múltiplos disparos (começando da esquerda) em intervalos de tempo apropriados.

A quantidade de inclinação da frente de onda, definida pelo ângulo de propagação da onda plana, pode ser controlada. Considere a geometria da frente de onda do caminho de raio na Figura 2.10. No momento em que a frente de onda gerada no local de origem S_1 atinge o ponto A na subsuperfície, a fonte de ponto no local S_2 deve ser excitada para que o ângulo desejado seja atingido. Defina a distância entre S_1 e S_2 como Δx e a velocidade média com a qual as ondas viajam como v . Se levar algum tempo para a frente de onda ir de S_1 a A , usando o triângulo S_1AS_2 , o ângulo de inclinação θ da onda plana é dada por

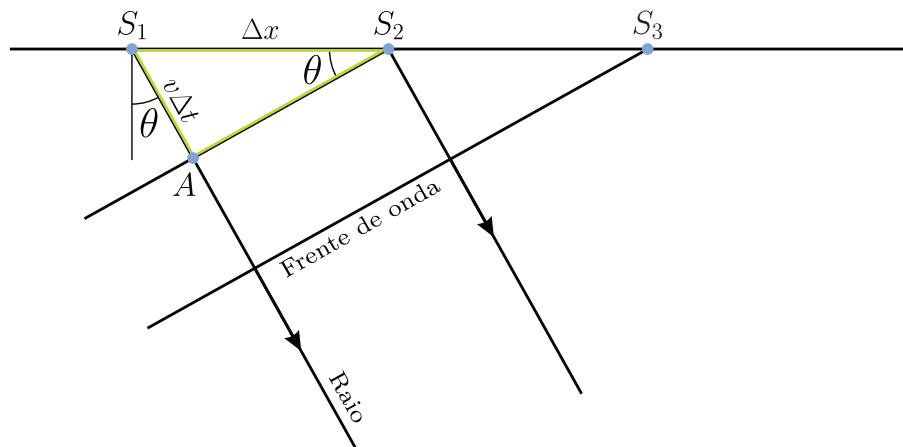


Figura 2.10: Cálculo do intervalo de tempo entre os disparos (S) para gerar a onda plana oblíqua na Figura 2.9.

$$\sin \theta = \frac{v\Delta t}{\Delta x}. \quad (2.5)$$

O local da fonte ativa deve, portanto, viajar com a velocidade fornecida

$$\frac{\Delta x}{\Delta t} = \frac{v}{\sin \theta} \quad (2.6)$$

ao longo da direção horizontal, e a fonte do ponto na localização S_2 deve ser excitada para que possamos pegar a frente da onda em S_1 quando ela atingir o ponto A na frente da onda na subsuperfície. A velocidade ($v / \sin \theta$) com a qual a localização da fonte deve se mover é chamada *velocidade de fase horizontal*.

A partir das experiências ilustradas nas Figuras 2.8 e 2.9, observe que uma onda plana que se propaga em um ângulo da vertical pode ser gerada por:

1. Colocando uma linha de fontes pontuais na superfície da Terra.
2. Excitando as fontes pontuais em sucessão com um atraso de tempo.
3. Sobrepondo as respostas que estão na forma de frentes de onda esféricas.

A resposta sobreposta é gravada em um único receptor (Figura 2.9). Essa resposta está na forma de uma onda plana que é refletida em uma interface. Superposição significa somar sobre o eixo do tiro para um determinado local do receptor. Usando o princípio da reciprocidade, a soma também pode ser realizada sobre o eixo do receptor para um determinado local do tiro.

Acabamos de discutir como uma família tiro comum com um único campo de ondas pode ser decomposta em seus componentes de ondas planas. Substituindo o eixo do tiro na Figura 2.10 pelo eixo receptor, a geometria do caminho do raio na Figura 2.11 resulta. O atraso de tempo associado à onda plana que viaja no ângulo θ na vertical e é dado por

$$\Delta t = \frac{\sin \theta}{v} \Delta x \quad (2.7)$$

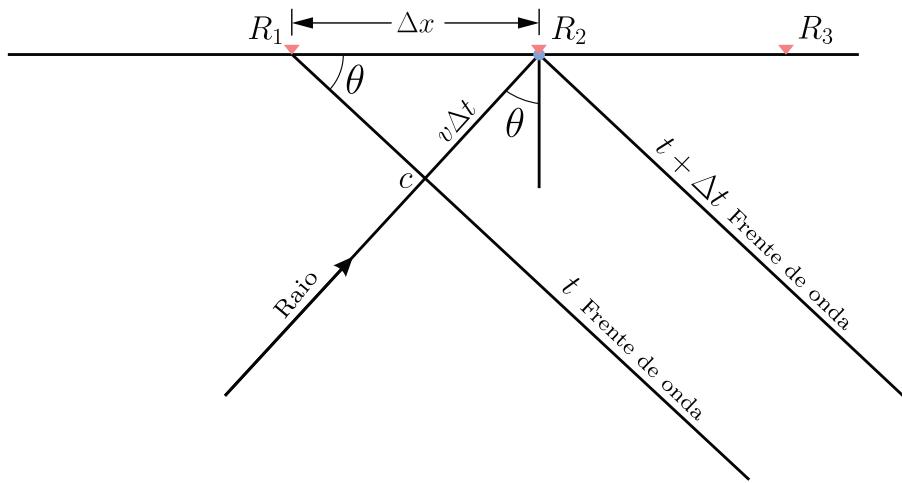


Figura 2.11: O princípio da reciprocidade aplicado à geometria da Figura 2.10 para substituir os disparos (S) pelos receptores (R).

A lei de Snell diz que a quantidade $\sin \theta/v$, que é o inverso da velocidade da fase horizontal, é constante

A lei de Snell diz que a quantidade $\sin \theta/v$, que é o inverso da velocidade da fase horizontal, é constante ao longo de um caminho de raio em um meio em camadas (Figura 2.12). Essa constante é chamada de parâmetro de raio p . A equação (2.7) é reescrita como

$$\sin \theta = \frac{v \Delta t}{\Delta x} \quad (2.8)$$

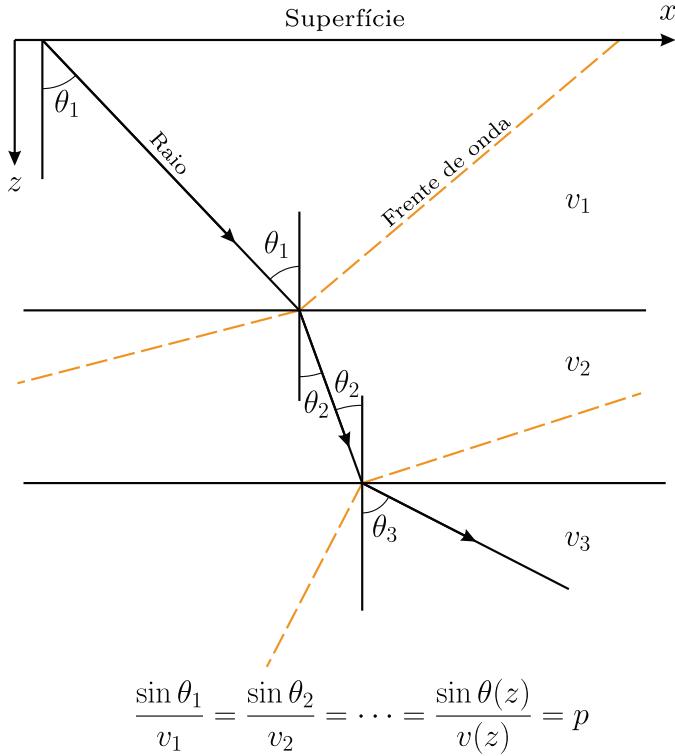


Figura 2.12: Se o parâmetro de raio p for especificado, o raio poderá ser traçado em um modelo de camadas horizontalmente com uma função de velocidade conhecida $v(z)$.

O ângulo de propagação da onda plana é controlado ajustando o valor de p . Se o parâmetro de raio p for especificado, o raio poderá ser traçado em um modelo de camadas horizontalmente com uma função de velocidade conhecida $v(z)$. Definindo $p = 0$ corresponde a uma onda plana que se propaga verticalmente. Dado o parâmetro de raio p e a função de velocidade $v(z)$ para o modelo de camadas, a família de caminhos de raios associados a um valor de p específico pode ser rastreada conforme mostrado na Figura 6.3-7. Uma onda plana que viaja em uma terra em camadas é chamada de onda Snell (Claerbout, 1978). Esse tipo de onda plana muda sua direção de propagação nos limites de cada camada, de acordo com a lei de Snell (Figura 2.12). Para um único valor p , observe que o sinal é gravado com muitas compensações (offsets) (Figura 2.13).

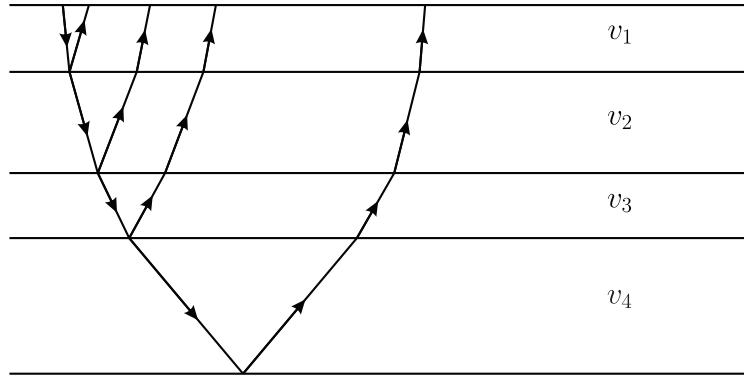


Figura 2.13: Alguns caminhos de raio para um dado valor p , correspondendo a um único rastreamento (traço) no domínio $\tau - p$.

Em geral, os receptores de todas as afastamentos registram ondas planas de muitos valores de p . Para decompor uma captura de tiro em seus componentes de onda plana, todas as amplitudes de traço na coleta devem ser somadas ao longo de vários caminhos inclinados, cada um com um atraso de tempo único definido pela equação (2.8).

Enquanto não houver mergulho, os tempos de propagação em um tiro comum e em um ponto médio comum são indistinguíveis. Como um agrupamento CMP não é um campo de onda único, a decomposição de ondas planas parece não se aplicar aos agrupamentos de CMP. No entanto, a equivalência de um agrupamento CMP e um agrupamento de tiros comuns em um modelo de camadas horizontais fornece uma justificativa para aplicar a decomposição de ondas planas nos dois tipos de agrupamentos (gathers).

TRANSFORMADA SLANT-STACK

Normalmente, duas etapas são usadas na síntese de ondas planas somando amplitudes no domínio de deslocamento ao longo de caminhos inclinados (Figura 2.14). Primeiro, uma correção de movimento linear (LMO) é aplicada aos dados por meio de uma transformação de coordenadas definida por (Claerbout, 1978)

$$\tau = t - px, \quad (2.9)$$

onde p é o parâmetro de raio, x é o deslocamento, t é o tempo de viagem duplo e τ é o tempo de interceptação em $p = 0$. Em seguida, os dados são somados sobre o eixo de deslocamento (offset) por

$$S(p, \tau) = \sum_x P(x, \tau + px), \quad (2.10)$$

onde $S(p, \tau)$ representa uma onda plana com parâmetro de raio $p = \sin\theta/v$. Repetindo a correção LMO para uma faixa de valores de p e realizando a soma na equação (2.10), é construída uma coleta de pilha inclinada completa. Uma pilha inclinada reúne, na prática, alternativamente é referido como uma coleta $\tau - p$; consiste em todos os componentes de mergulho dentro da faixa especificada de valores de p nos dados de deslocamento originais.

O mapeamento do domínio $t - x$ para o domínio $\tau - p$ é reversível (Thorson, 1978). Primeiro, aplique a correção de movimento linear inverso (LMO) aos dados no domínio $\tau - p$

$$t = \tau + px. \quad (2.11)$$

Em seguida, some os dados no domínio $\tau - p$ sobre o parâmetro de raio eixo p para obter

$$P(x, t) = \sum_p S(p, t - px). \quad (2.12)$$

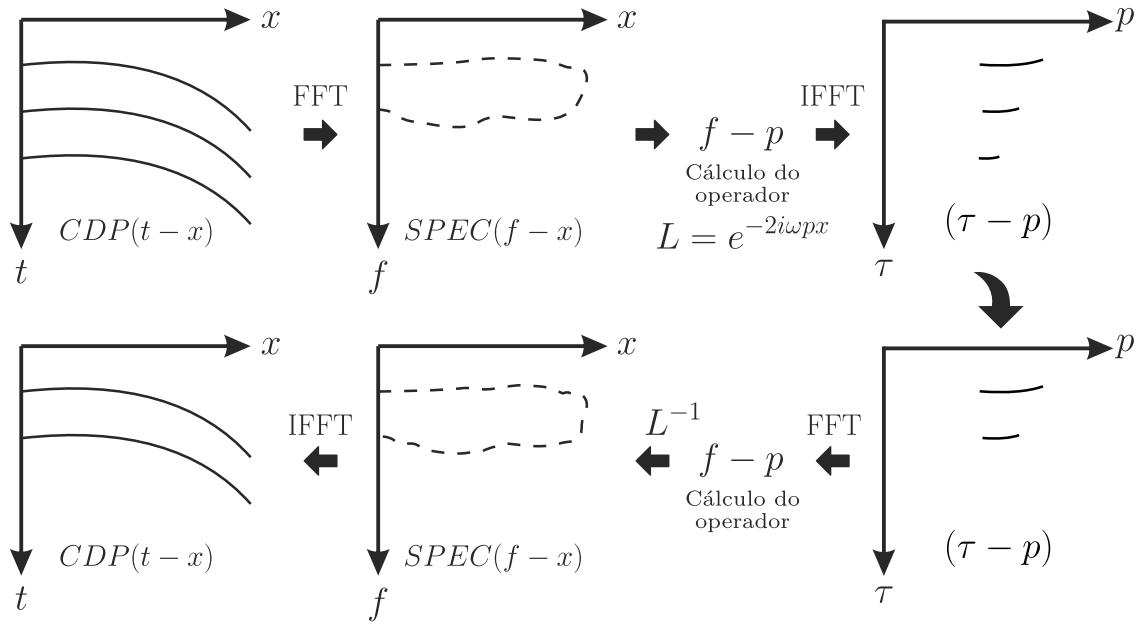


Figura 2.14: Etapas para a construção do empilhamento oblíquo.

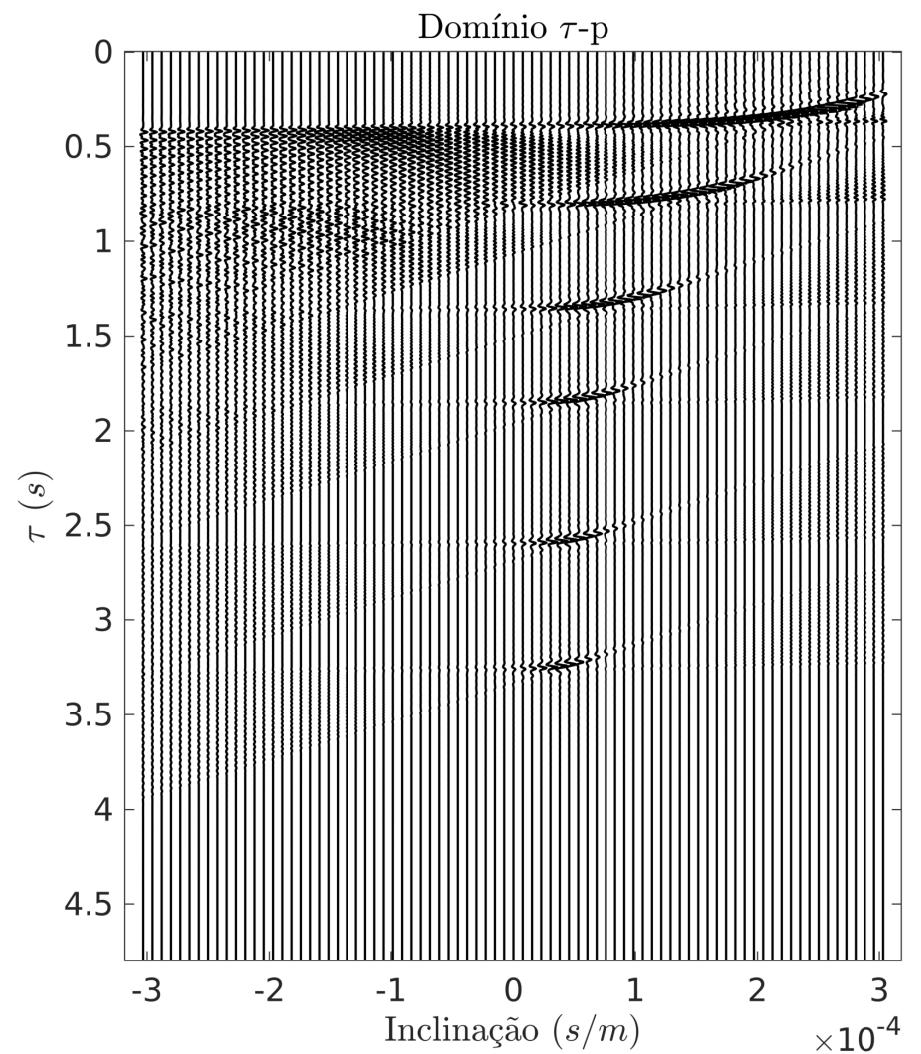
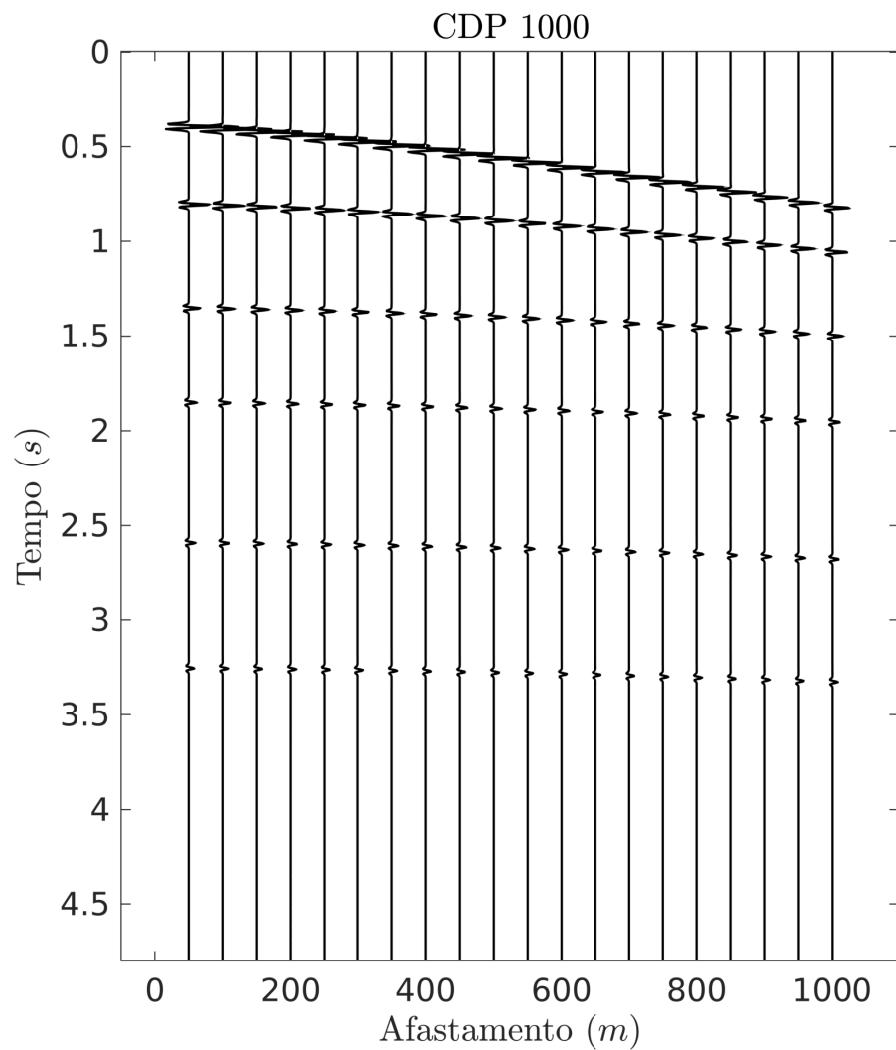


Figura 2.15: Transformada de Radon Linear direta do CDP-1000. (a) CDP-1000 no domínio (x, t) e (b) CDP-1000 no domínio (τ, p) .

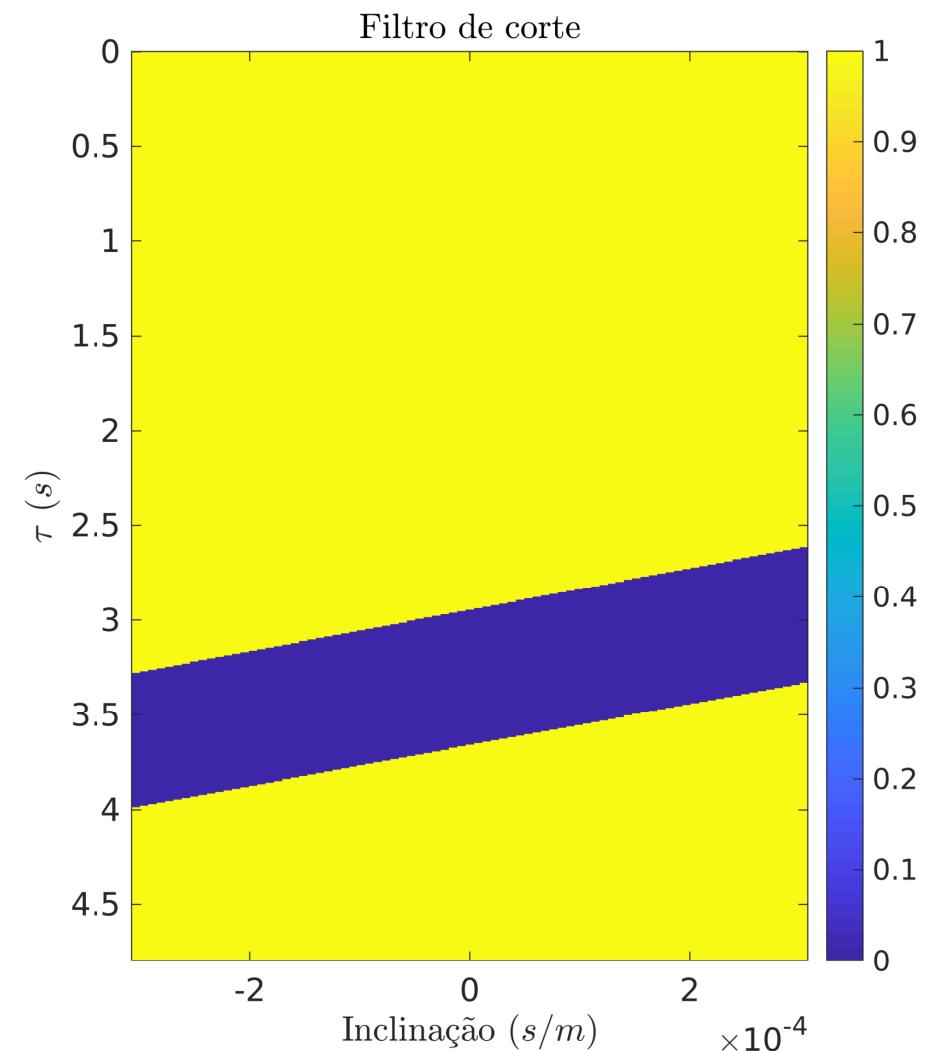
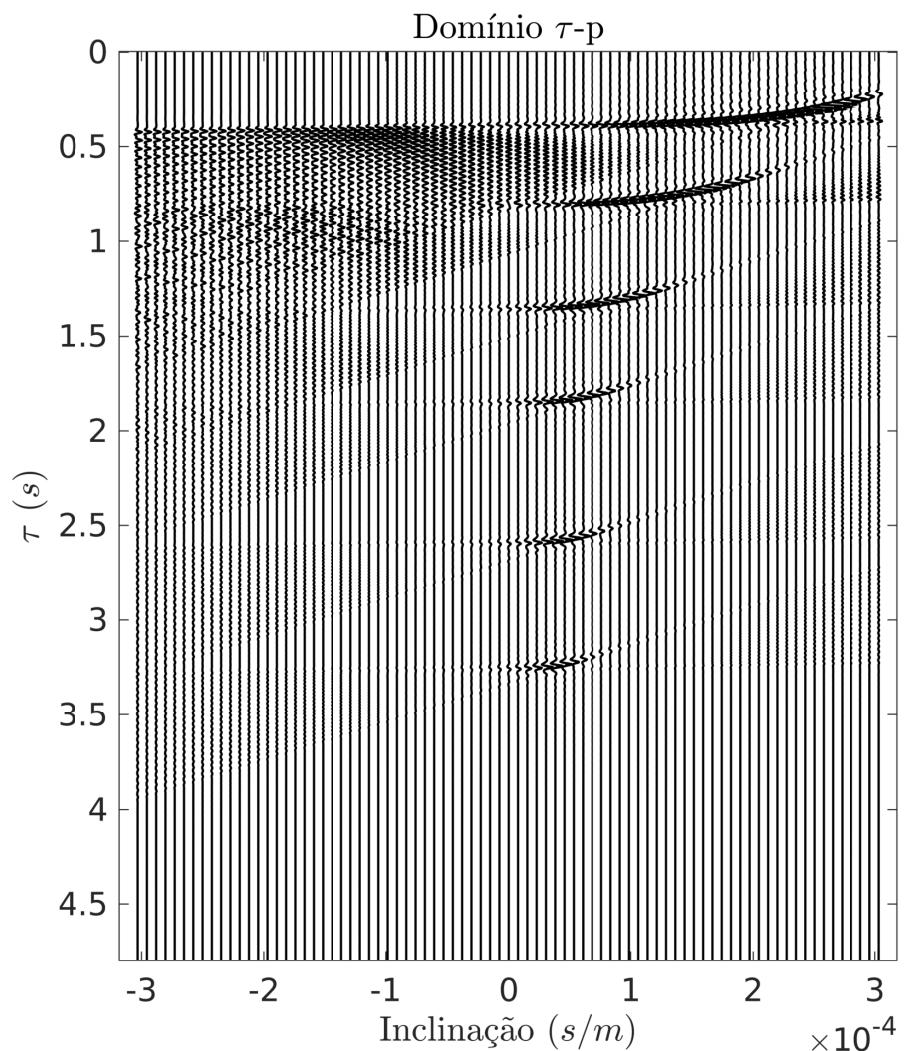


Figura 2.16: Transformada de Radon Linear direta do CDP-1000. (a) CDP-1000 no domínio (τ, p) . (b) Filtro de corte aplicado no CDP-1000.

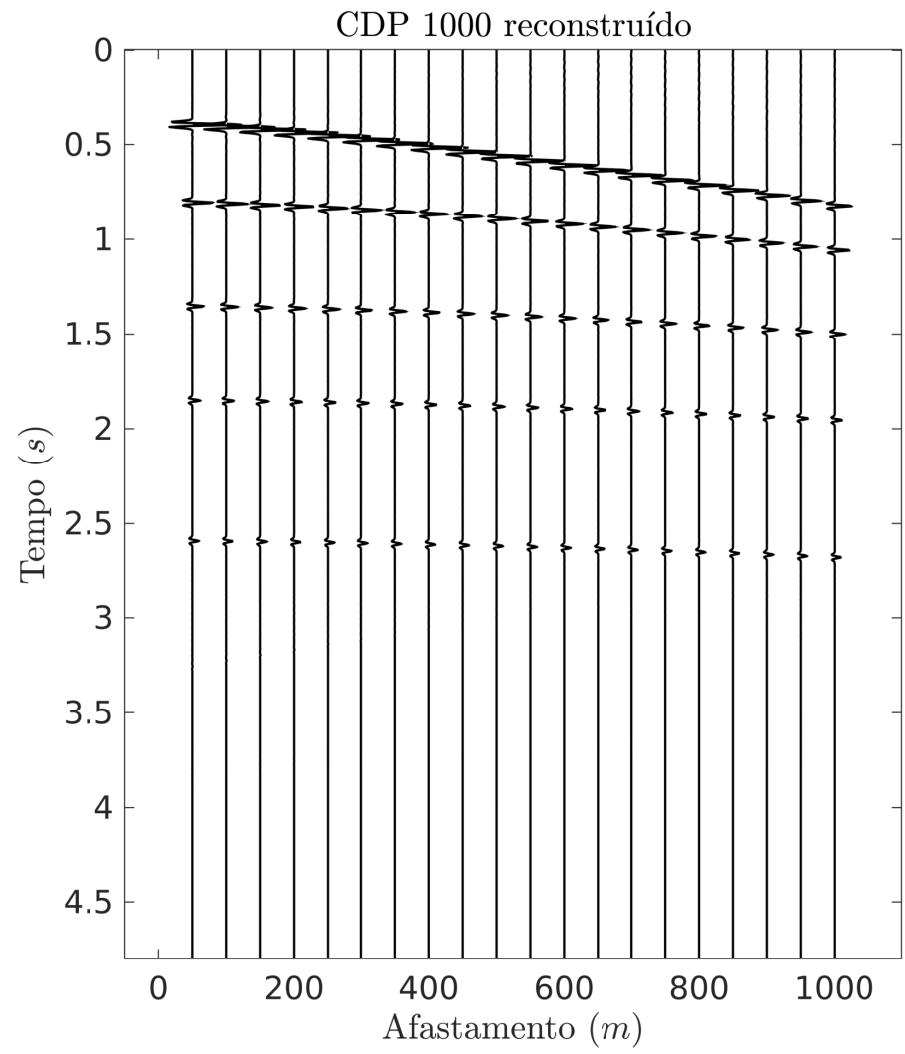
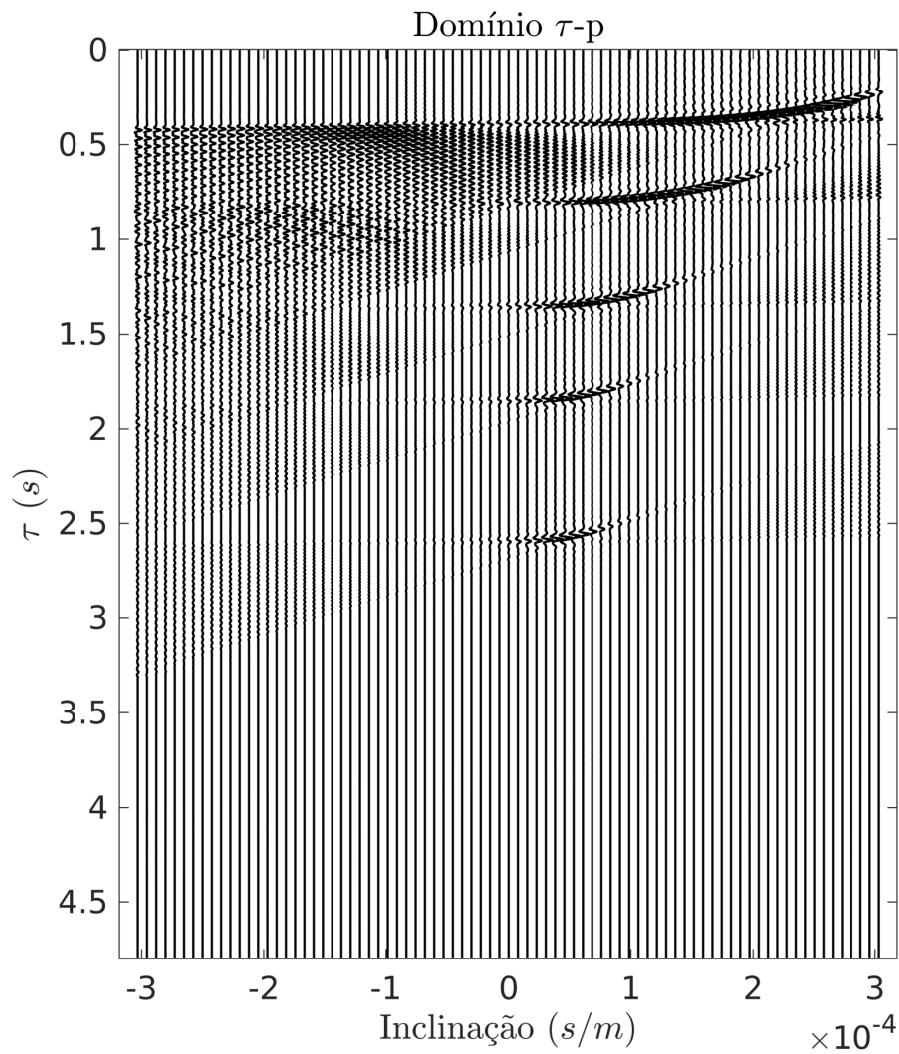


Figura 2.17: Transformada de Radon Linear inversa do CDP-1000. (a) CDP-1000 no domínio (τ, p) e (b) Reconstrução do CDP-1000 no domínio (x, t)

2.4 TRANSFORMADA F-K

A transformada de Fourier (TF) constitui um dos maiores produtos da física e da matemática, e é indispensável na teoria e no processamento de sinais por várias razões. Uma das primeiras é que a sua base é estruturada no conceito de *frequência*, o que permite uma compreensão melhor do fenômeno sendo estudado, e que adiciona um complemento ao sinal temporal (espacial) que é inicialmente usado para análise. Esta condição é muito comum, e os exemplos são vários na física de ondas (como na acústica, sísmica, eletromagnetismo, vibrações, ótica, etc.), bem como em outras áreas onde processos periódicos são importantes e regem os fenômenos de interesse (como na biomedicina, biologia, astronomia, economia, etc.). (Leite, 2015)

Para tornar a descrição mais prática, define-se o par das integrais direta, $G(\omega)$, e inversa, $g(t)$, em 1D nas seguintes formas ilimitadas (o que se faz para um lado $t, \omega \rightarrow +\infty$ se faz para o outro $t, \omega \rightarrow -\infty$):

$$G(\omega) = k_1 \int_{-\infty}^{\infty} g(t) e^{-i\omega t} dt, \quad G(\omega) = F\{g(t)\}, \quad (i = \sqrt{-1}), \quad (2.13)$$

$$g(t) = k_2 \int_{-\infty}^{\infty} G(\omega) e^{+i\omega t} d\omega, \quad g(t) = F^{-1}\{G(\omega)\}, \quad (\omega = 2\pi f). \quad (2.14)$$

A escolha dos coeficientes k_1 e k_2 dependem do usuário ou do problema em estudo. O requerimento é que o produto $k_1 k_2 = 1/2\pi$. Se $k_1 = k_2$, então, $k_1 = 1/\sqrt{2\pi}$. Se $k_1 = 1$, então, $k_2 = 1/2\pi$.

Para o caso 2D, temos a TF direta na forma da equação (2.15) e TF inversa na equação (2.16)

$$h(t, x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(f_t, f_x) e^{+i2\pi(f_t t + f_x x)} df_t df_x \quad (2.15)$$

$$H(f_t, f_x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(t, x) e^{-i2\pi(f_t t + f_x x)} dt dx \quad (2.16)$$

Para estas equações se tem as relações: $f_t = \frac{1}{T}$, $f_x = \frac{1}{\lambda} = \frac{1}{vT}$, de onde se estabelece a relação $f_t = vf_x$; isto é, nestas condições a frequência temporal, f_t , e a frequência espacial, f_x , são acopladas através de uma relação linear $f_t = vf_x$, onde v é o parâmetro de inclinação. Também, usando a vagarosidade $s = \frac{1}{v}$ se tem a forma $f_x = sf_t$. A figura 2.18 mostra a relação entre estas quantidades, com a descrição da propagação de uma onda plana, e a relação denominada de dispersão que envolve as frequências espaciais e a temporal.

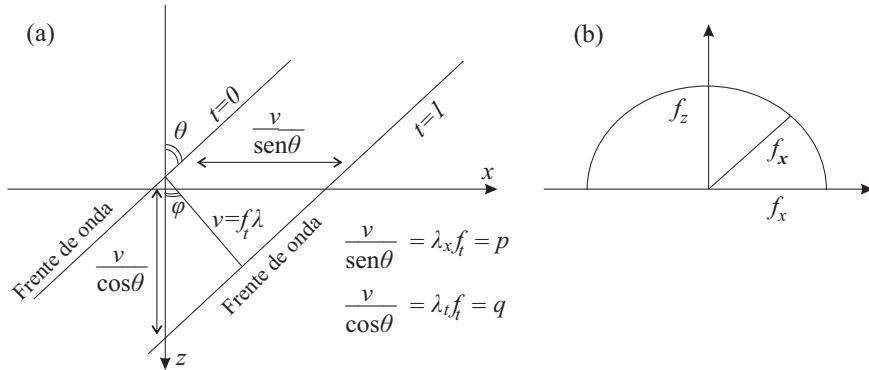


Figura 2.18: Descrição física da relação entre a frente de onda e a equação da dispersão. (a) Propagação da onda plana. (b) Relação entre as componentes f_x e f_z e a total f_x para f_t constante.

2.5 FILTROS DE CORTE NA FREQUÊNCIA

O princípio a ser aplicado é o de que os eventos possam ser separados no domínio da frequência, $f_t - f_x$, de alguma forma, pelo menos parcial. Partindo deste princípio, a figura 2.19 ilustra a composição espectral, e serve para a análise da relação frequência temporal-espacial (número de onda) do conteúdo de velocidades comumente presente em experimentos de sísmica de exploração.

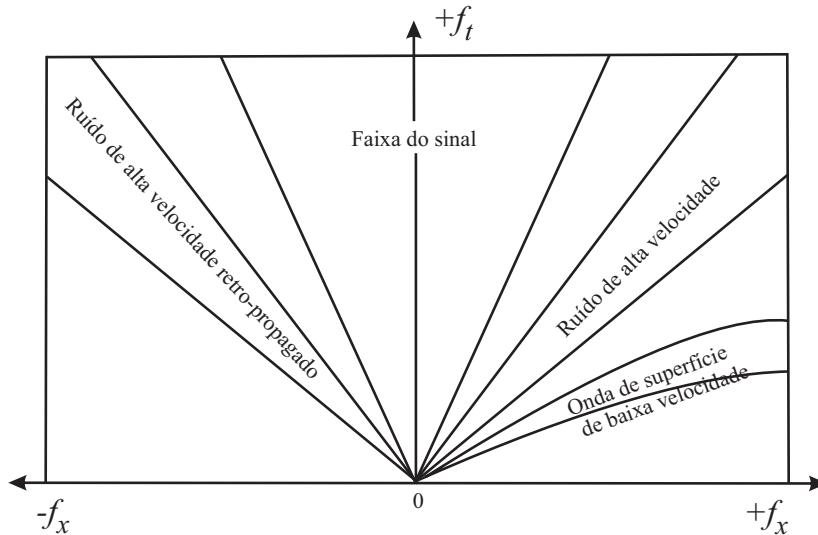


Figura 2.19: Ilustração da composição espectral de uma seção sísmica (frequência temporal-espacial). Se destacam as linhas de sinal de informação, ruídos de velocidade alta e baixa, e ruido das ondas de superfície de baixa frequência. As linhas inclinadas separam os eventos mergulhantes no domínio $f_t - f_x$.

A figura 2.20 ilustra o problema que se encontra na tentativa de se aplicar um filtro, $H(f_t, f_x)$, convencional simples, o que se ilustra com a figura 2.21, que tem por finalidade esboçar a relação linear simples (outra forma qualquer pode ser desenhada) no domínio espectral para o caso prático de dados no domínio do discretizado; as frequências de

Nyquist estão definidas: $f_{Nt} = \frac{1}{2\Delta t}$, $f_{Nx} = \frac{1}{2\Delta x}$ e $f_{Nt} = vf_{Nx} = \frac{v}{2\Delta x}$. Neste caso, v é a velocidade e s é a vagarosidade. Observe-se na figura 2.21 as faixas de passagem e de rejeição lineares, onde está superposto uma ilustração de uma linha de contorno típica (forma qualquer) do espectro de um dado real. A relação entre as linhas de corte (passagem-rejeição) linear e espectro real não é simples, o que se deseja é que as linhas sejam o mais alinhado (paralelo) possível.

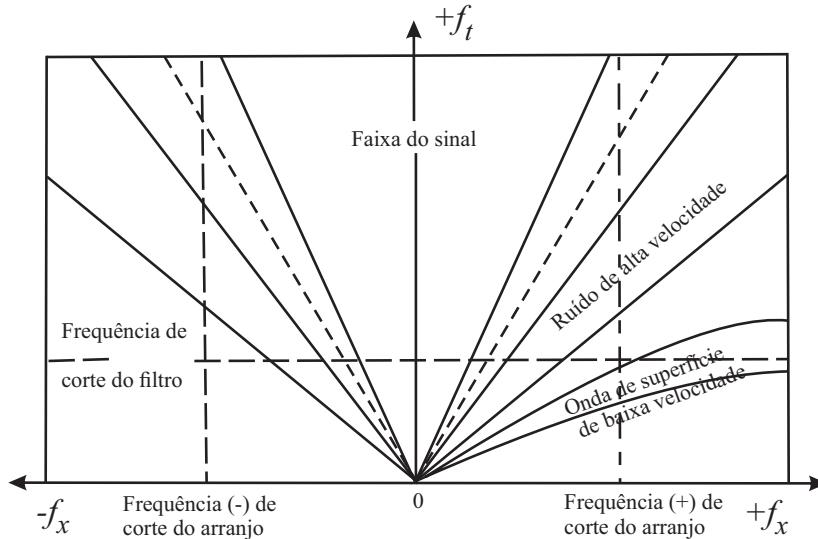


Figura 2.20: Construção idealizada do filtro para separação da composição espectral de uma seção sísmica (frequência temporal-espacial) correspondente à figura 2.19. Se destacam as linhas de sinal de informação, ruídos de velocidade alta e baixa, e ruido das ondas de superfície de baixa frequência.

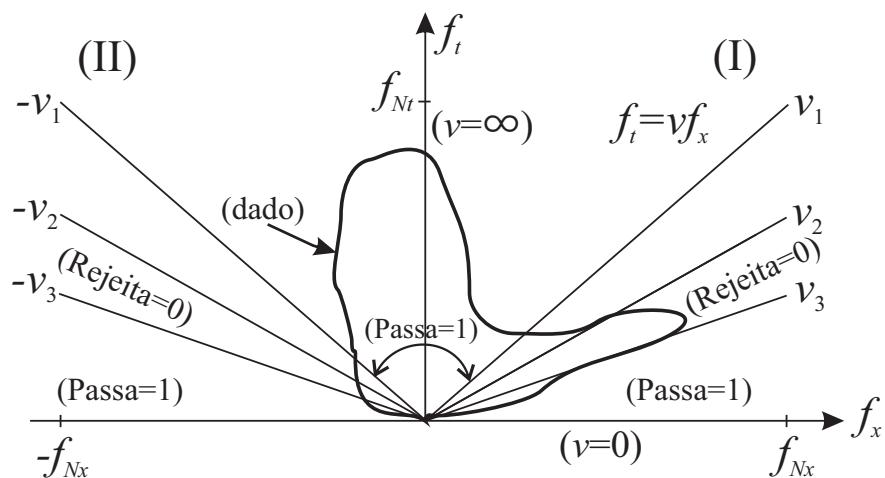


Figura 2.21: Divisão do domínio da frequência 2D em quadrantes, ilustração do espectro de (dado) real, $H_R(f_t, f_x)$, e desenho do filtro com a forma Em-leque de Passagem-Rejeição (EL-P-R). Para completar o filtro é necessário refletir os quadrantes: I para III e II para IV.

A figura 2.20 esboça a tentativa de se preservar uma banda de frequências mais ampla possível. Para isto se analisa a geometria do arranjo de registro (padrão de distribuição dos sensores nas estações, padrão de tiro, etc.), e ao passo que as dimensões geométricas do arranjo aumenta, a linha de corte vai no sentido de f_x menor (comprimento de onda maior). Ao passo que as dimensões do arranjo aumenta, o sistema atenua a informação de eventos de mergulho (linear) e ruído. E se as dimensões continuam aumentando, o corte passa para o leque do sinal de informação (Faixa do sinal).

Após a aquisição do dado sísmico figura (2.6) no *SU* com o programa *triseis* utilizou-se o tiro 50 para a filtragem F-K no *matlab*, onde foi implementado os plots da seção tiro figuras 2.24, 2.25, 2.26 e 2.27.

O ruído é intrínseco ao registro do traço sísmico, seja como consequência do meio geológico ou do próprio equipamento de registro utilizado. Para simular a entrada de ruído na seção sísmica, assumimos o pressuposto de que ele equivale a um arranjo de números aleatórios de média zero.

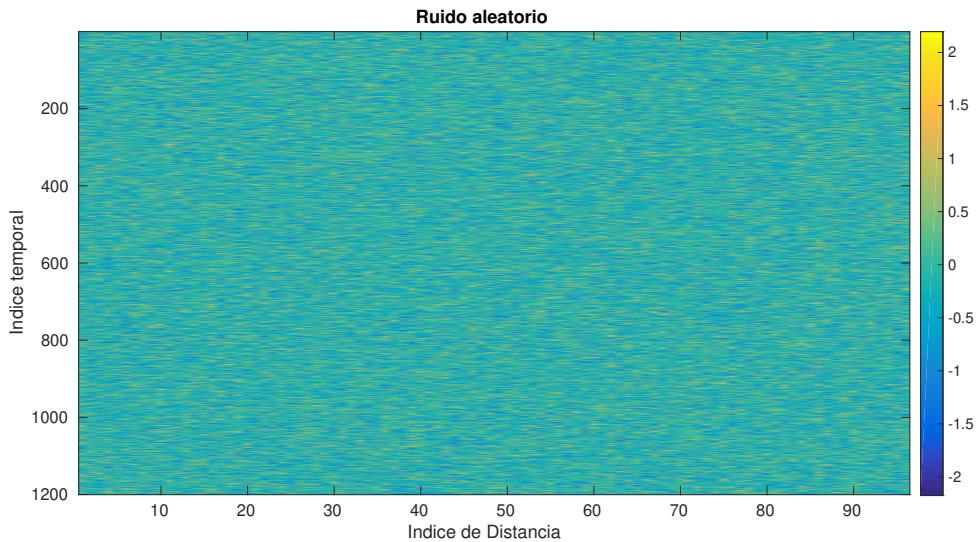


Figura 2.22: Ruído aleatório gerado no matlab

Nas seções sísmicas se precisa aplicar uma função de ganho devido à diminuição da amplitude com a distância-tempo, denominado de espalhamento geométrico, divergência esférica, e às vezes atenuação tempo-distância. A função ganho empregada corresponde a uma amplificação do sinal no tempo para que se possa ver e interpretar a composição do sinal, que na sísmica é composto de eventos do tipo ruído, reflexões, refrações, múltiplas e difrações. Isto é, a correção de espalhamento geométrico por uma função ganho é aplicada para compensar o decaimento de energia da divergência da frente de onda em qualquer fase do processamento de uma seção sísmica. Ressalta-se que a aplicação de um ganho é uma operação destrutiva, ou não-destrutiva; o ganho aplicado é classificada como quase não-destrutivo devido ao ponto zero anulado. A função ganho aplicada é dada por

$$g(t) = t^a e^{-bt}, \quad (2.17)$$

onde os valores utilizados para os parâmetros foram $a = 2$ e $b = 0,5$.

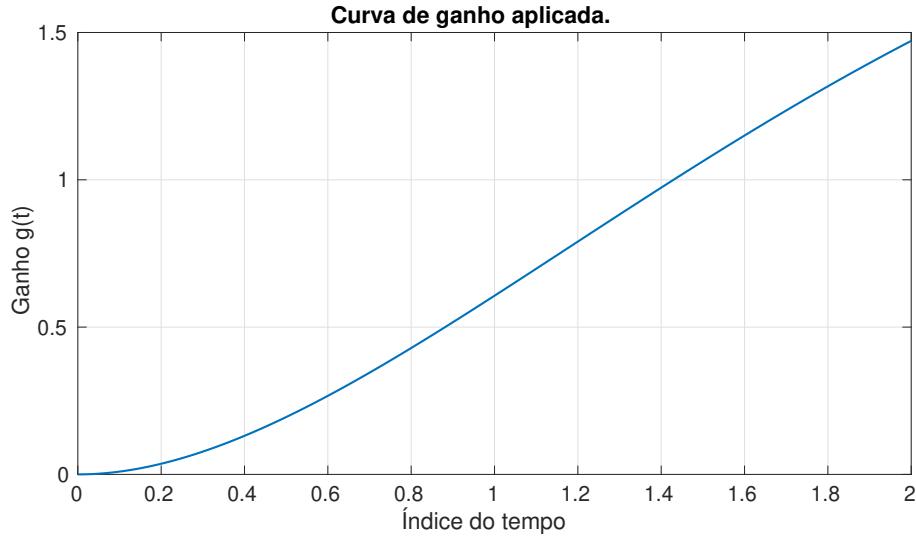


Figura 2.23: Função ganho empregada.

A figura 2.23 mostra a variação do ganho, onde se observa que ela é crescente, mas com um intervalo de amplificação entre zero e 1,5, que se apresenta de uma forma diferente, mas consistente no que se refere a colocar os valores num intervalo menor.

Para que se possa realizar uma filtragem, a informação a ser recuperada deve ser separável do não-desejado, no domínio-f , ou no domínio-t.

A figura 2.28 mostra o espectro de amplitude da seção tiro 50 com ruído gerada a partir da figura 2.26.

O termo filtro é reservado para a operação de multiplicação no domínio da frequência, e pode zerar a parte não desejada do espectro do sinal. Através da definição do espectro de amplitude e de fase podemos construir o operador $H(f) = A(f)e^{\theta(f)}$, que representa o filtro na frequência. Todo e qualquer processo de filtragem de um sinal pode ser representado na seguinte forma:

$$S_H(f) = G(f)H(f) \quad (2.18)$$

Onde S_H representa o sinal filtrado no domínio da frequência, $G(f)$ representa o sinal a ser filtrado e $H(f)$ o filtro a ser aplicado. Ou no domínio tempo na forma de convolução:

$$s_h(t) = s(t) * h(t) \quad (2.19)$$

As figuras (2.29), (2.30) e (2.31) ilustram a geometria de passagem e rejeição de uma forma geral.

Após a filtragem da seção tiro 50 (ver figura 2.32) nota-se que a maior parte da componente de ruído (informação não desejável) foi removida com a filtragem F-K. O corte no espectro de amplitude de maneira iterativa pelo matlab permite maior precisão na eliminação do ruído. Comparando a figura 2.32 com 2.26, respectivamente a seção tiro 50 filtrada sem ganho e seção tiro 50 com ruído.

Em seguida a figura 2.33 mostra a atuação do ganho na seção tiro 50 filtrada, o ganho aumenta com o tempo ou seja não apenas os eventos desejáveis são amplificados como também a componente de ruído. Comparando as duas figura 2.33 com 2.27 onde respectivamente representam a seção tiro 50 filtrada com ganho e seção tiro 50 com ruído e ganho.

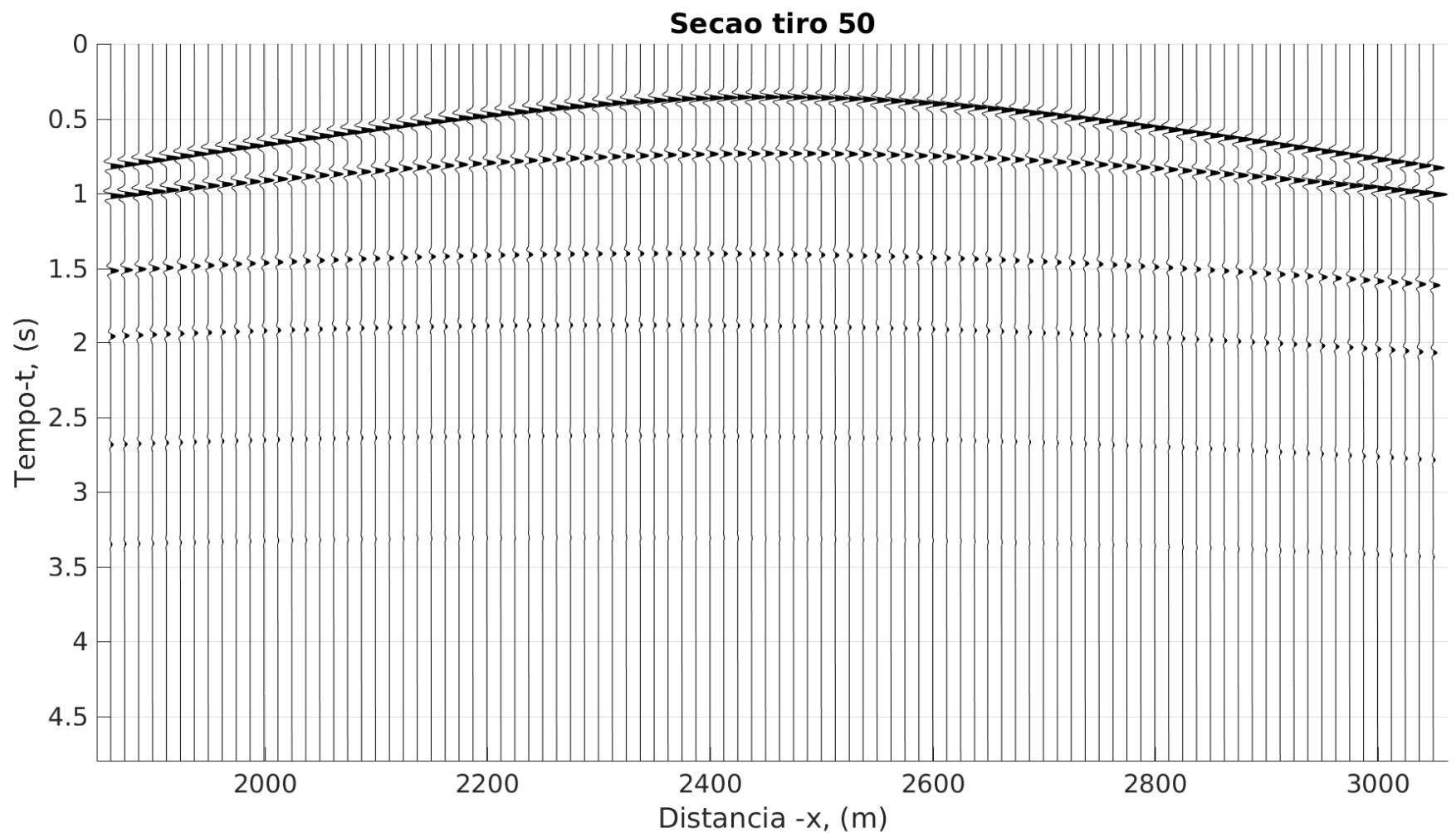


Figura 2.24: Seção tiro 50 sem ganho do modelo sintético, figura (2.2)

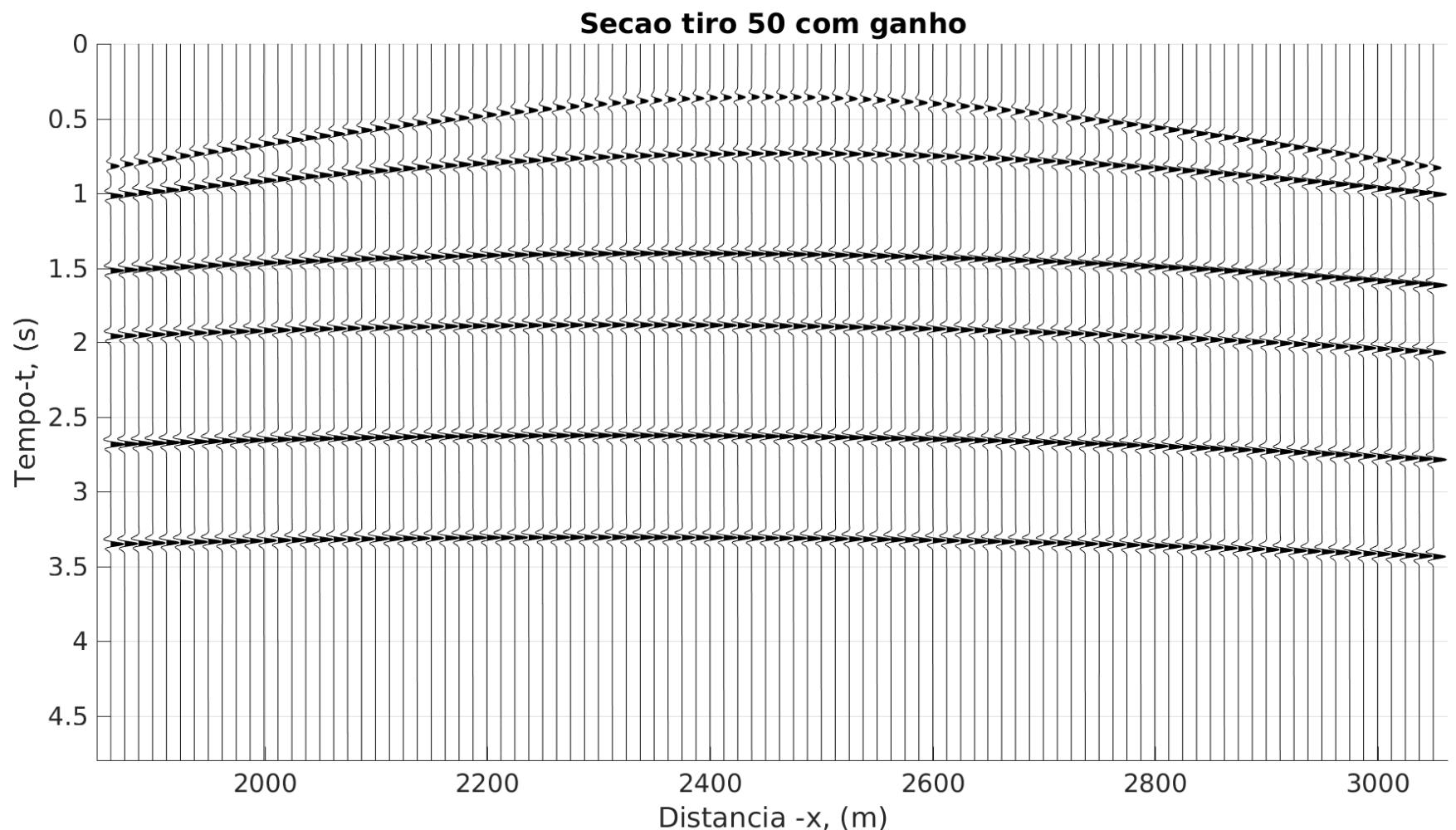


Figura 2.25: Seção tiro 50 com ganho, os eventos com maior tempo foram realizados.

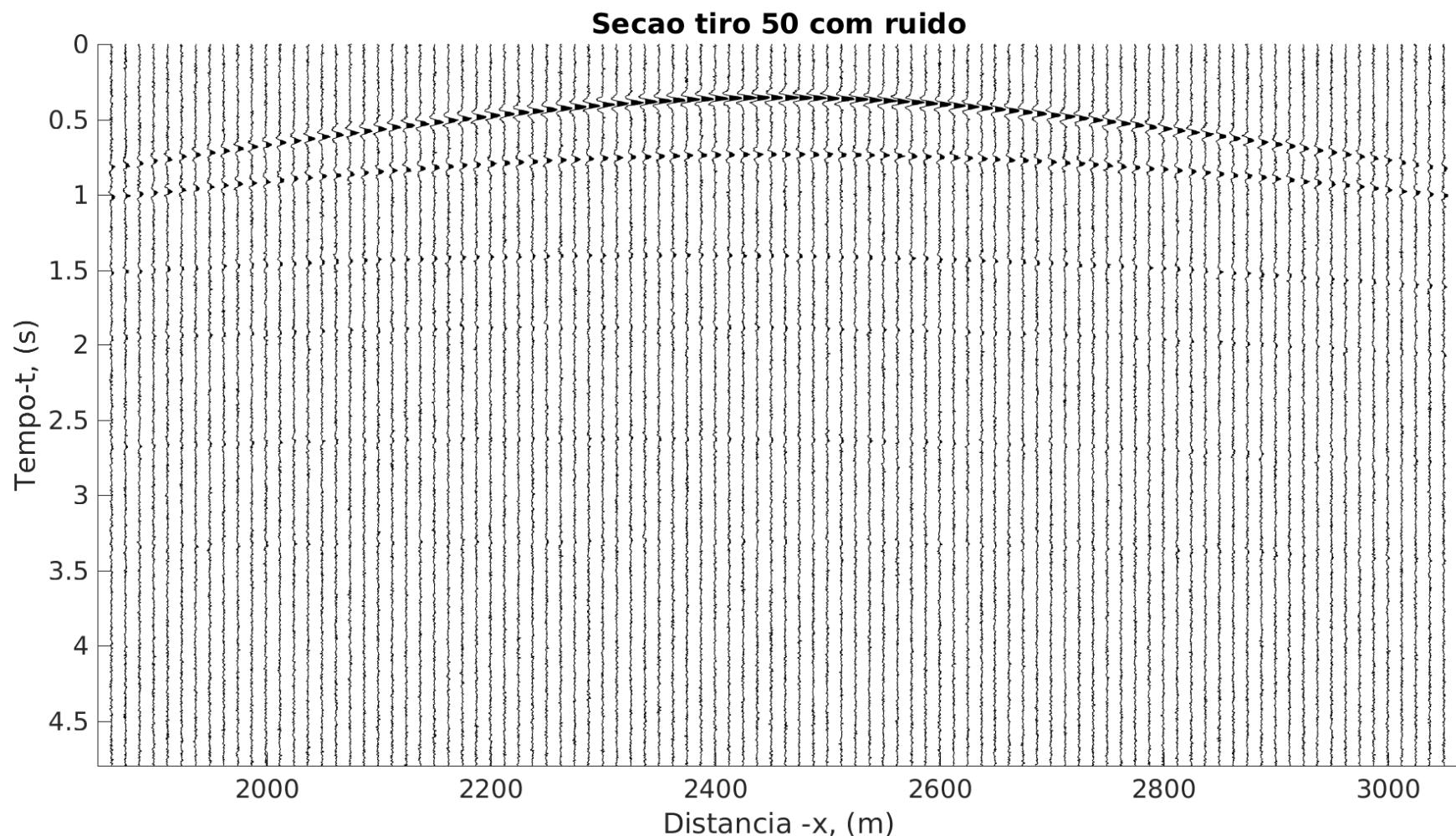


Figura 2.26: Seção tiro 50 com ruído aleatório, o ruído foi gerado no matlab e somado a seção tiro 50.

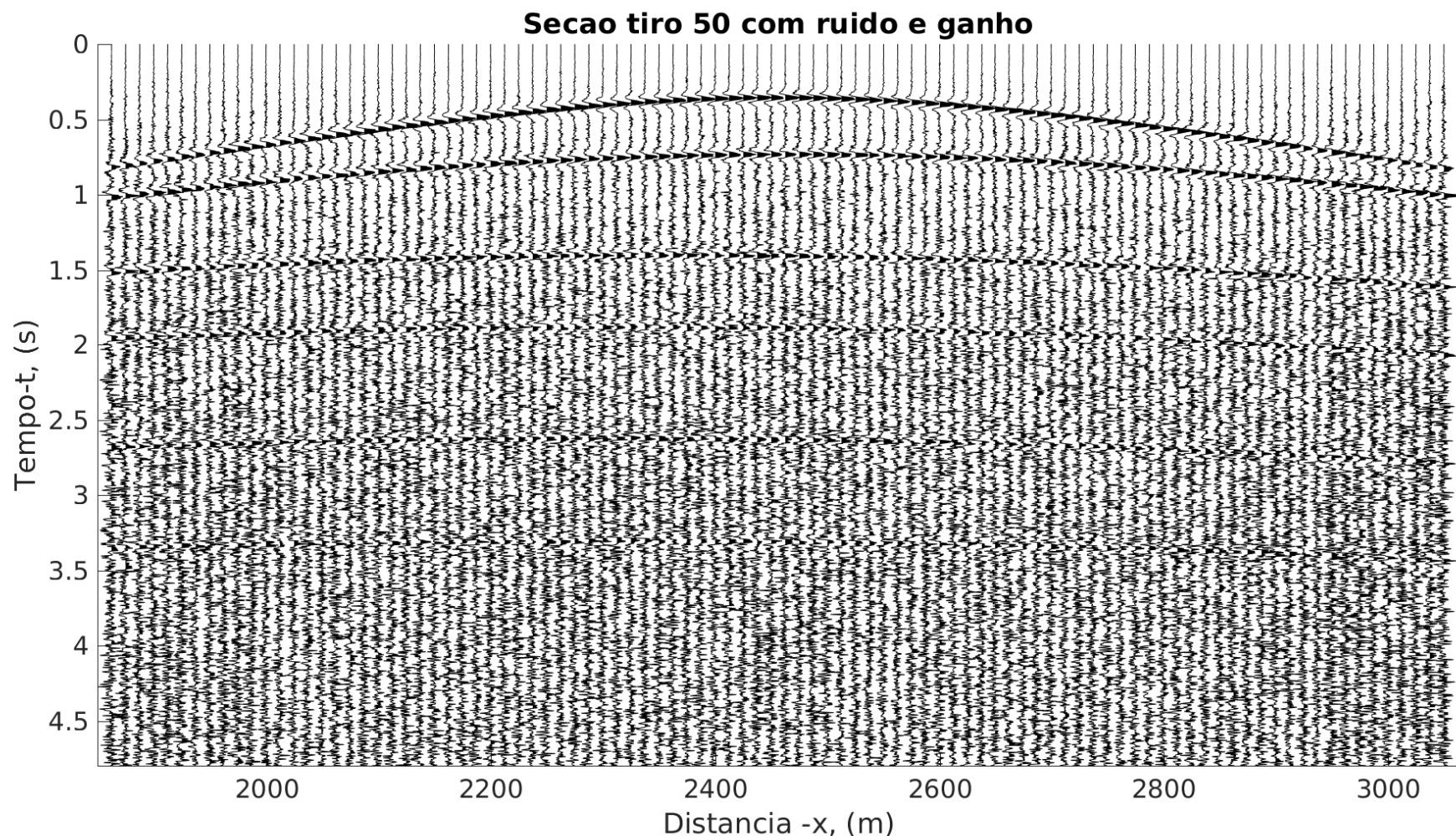


Figura 2.27: Seção tiro 50 com ruído e ganho, a informação e o ruído foram amplificados, os eventos de maior tempo foram “mascarados” com a amplificação da função ganho.

Espectro de amplitude da secao tiro 50 com ruído

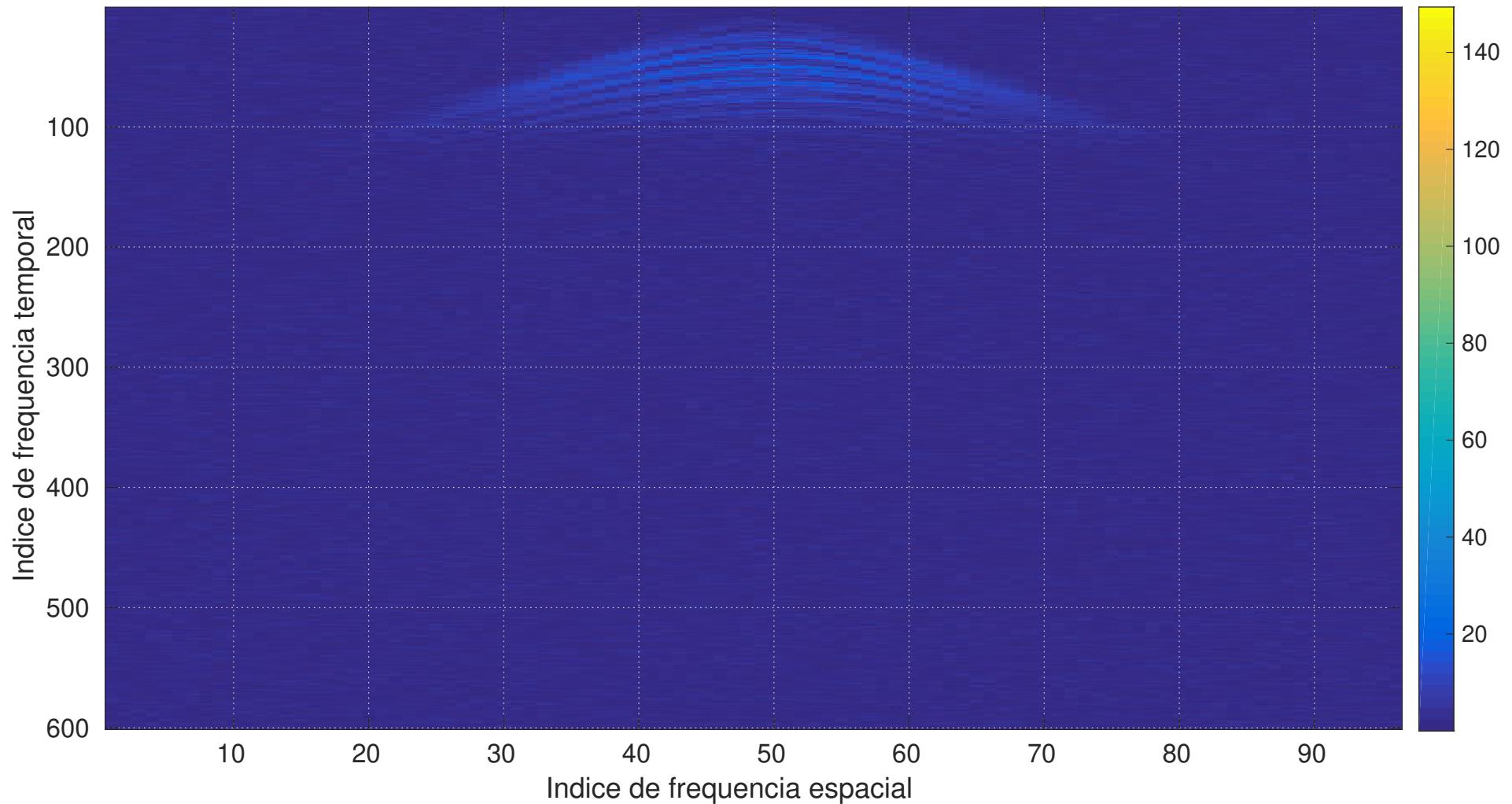


Figura 2.28: Espectro da seção tiro 50 com ruído, este espectro não apresenta falseamento de informação tanto em f_t (frequência temporal) quanto em f_k (frequência espacial).

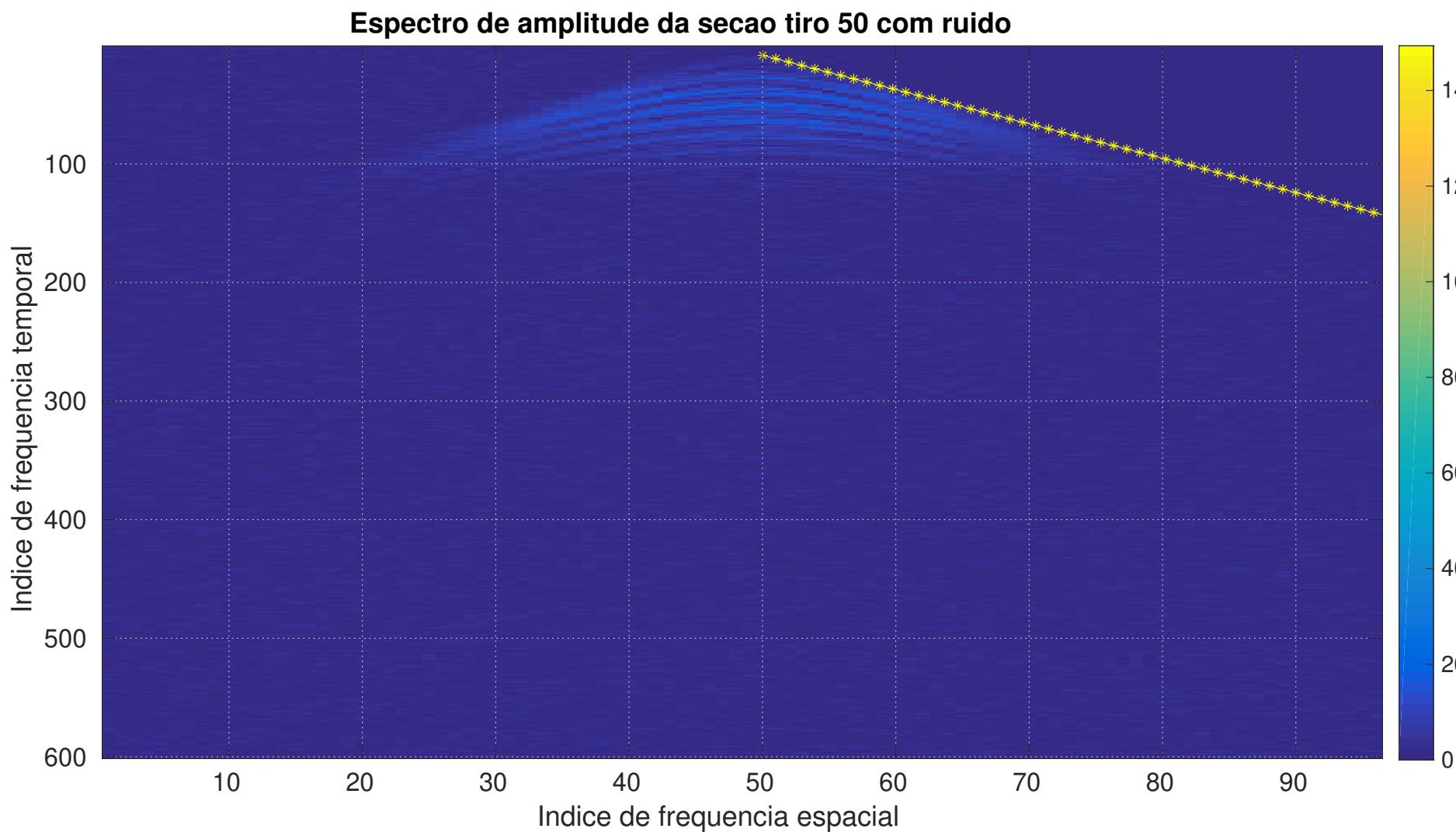


Figura 2.29: Seção tiro 50 com ruído após o primeiro corte (linha amarela) no espectro de amplitude.

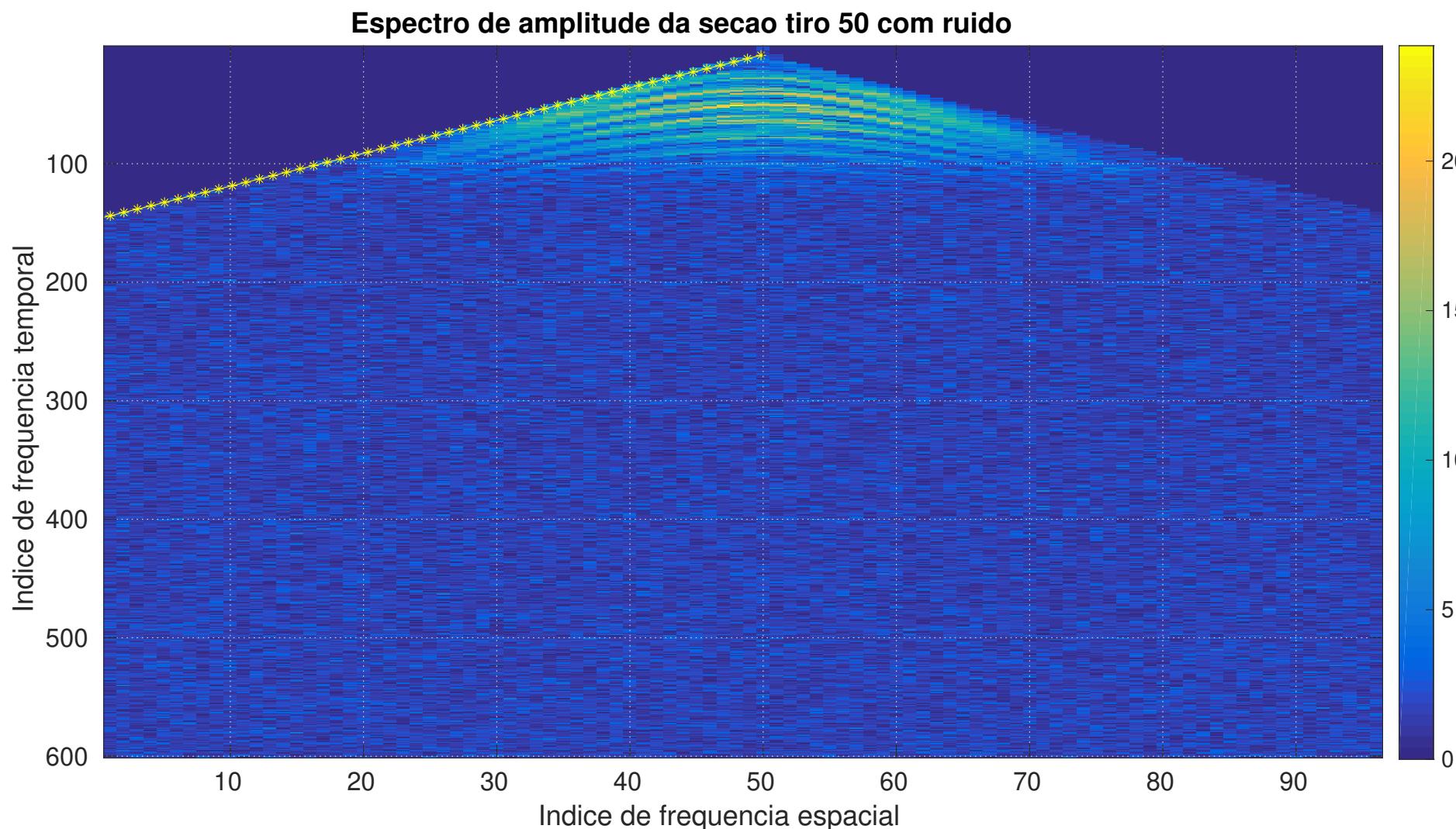


Figura 2.30: Seção tiro 50 com ruído após o segundo corte (linha amarela) no espectro de amplitude.

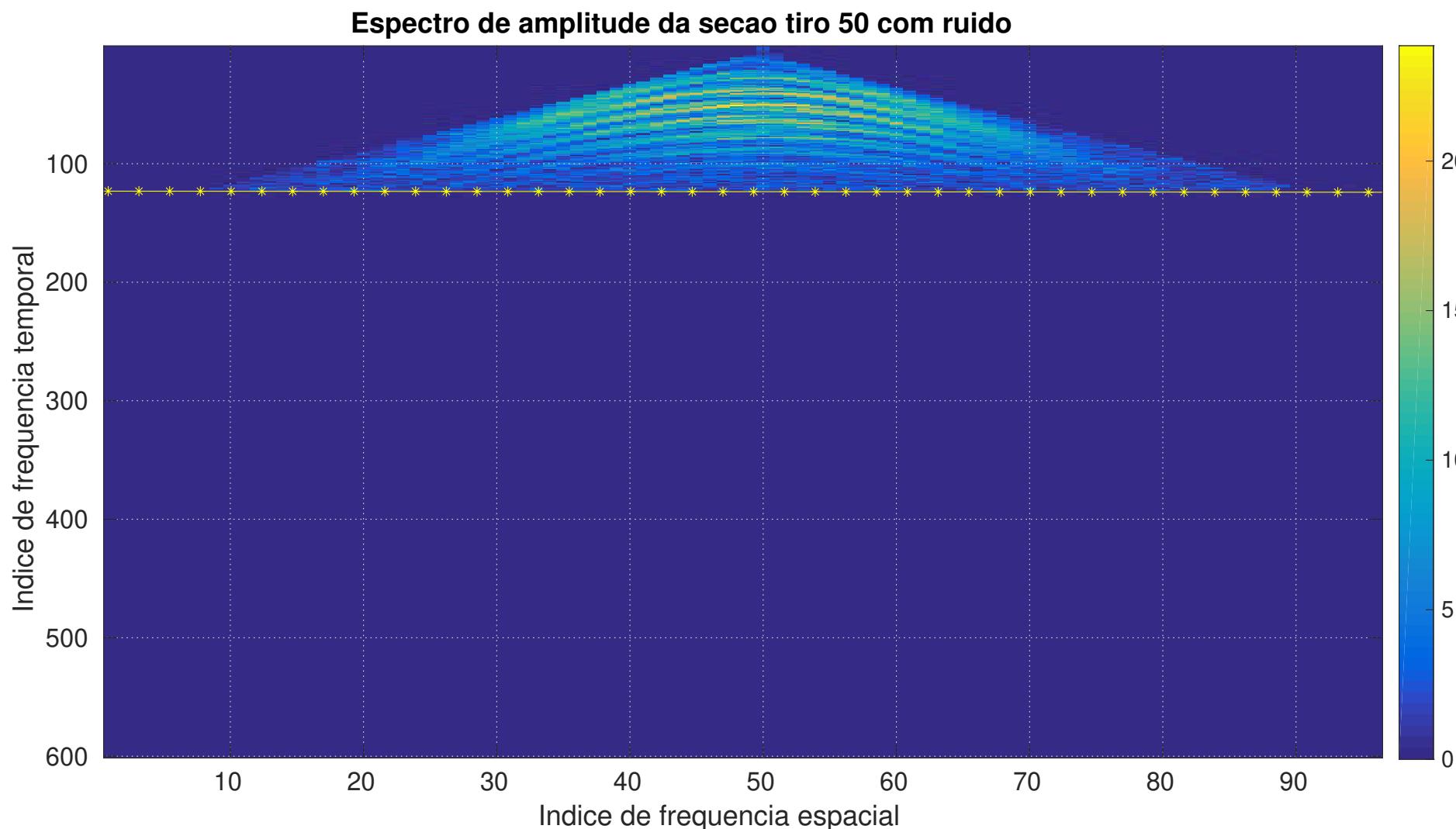


Figura 2.31: Seção tiro 50 com ruído após o terceiro corte (linha amarela) no espectro de amplitude.

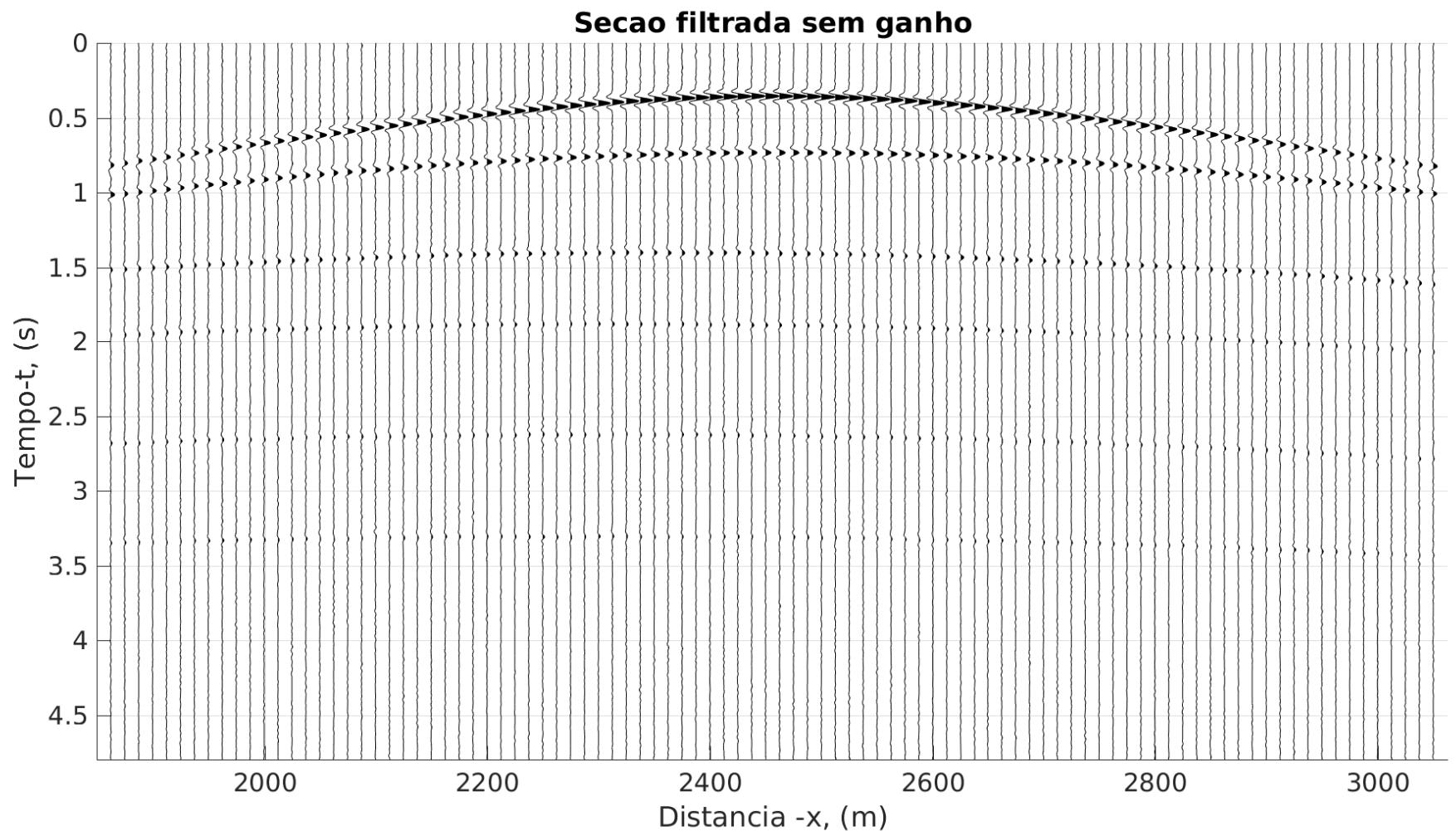


Figura 2.32: Seção tiro 50 filtrada do ruído sem ganho, comparando a figura (2.26) nota-se que a componente do ruído foi em grande parte removida após a filtragem F-K.

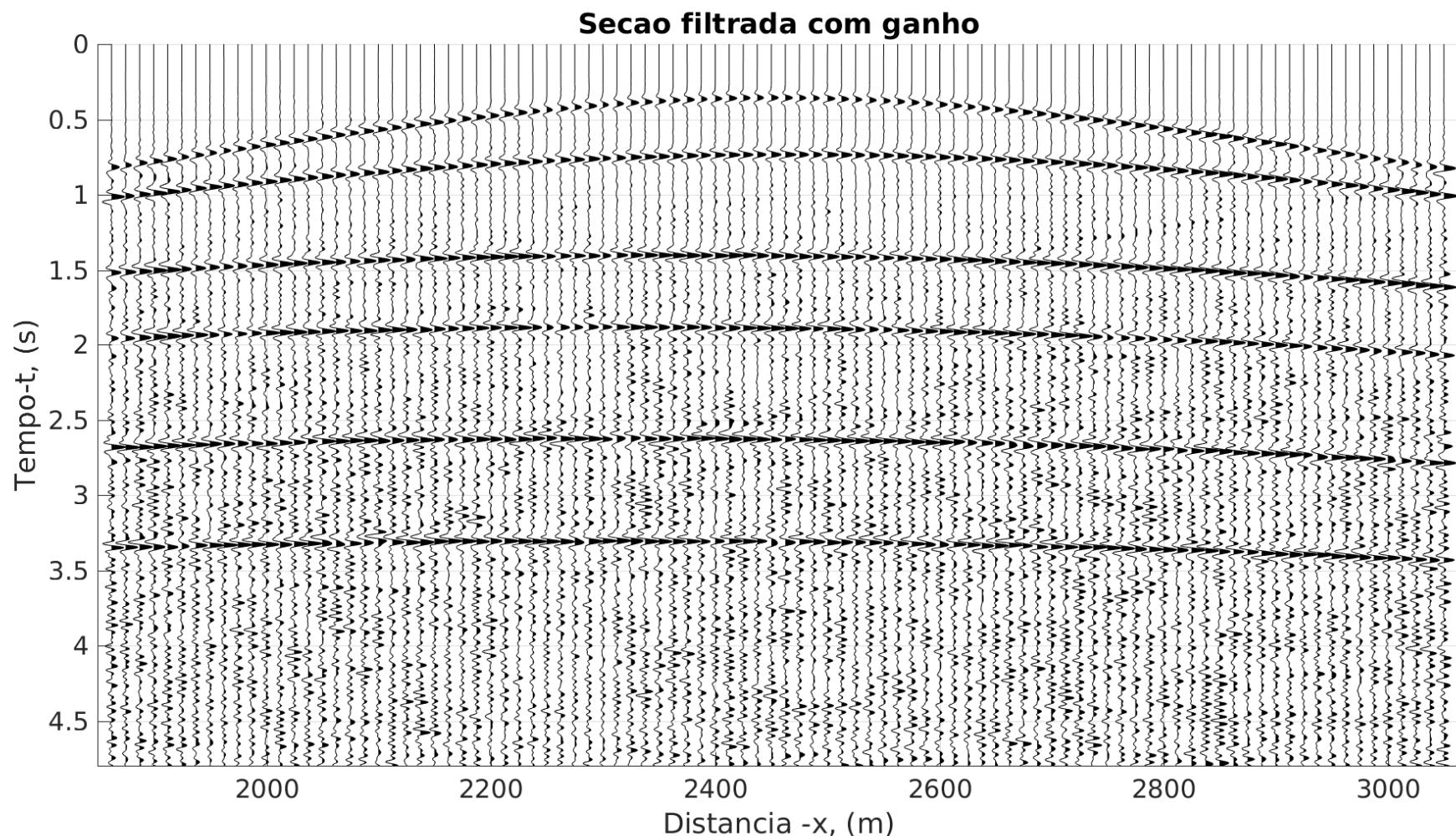


Figura 2.33: Seção tiro 50 filtrada do ruído com ganho, comparando a figura (2.27) nota-se que os eventos com maior tempo foram recuperados. O ganho amplificou tanto a informação de interesse (eventos) quanto o ruído.

2.6 COBERTURA MÁXIMA DOS CDP'S

Após o carregamento da geometria, uma forma de saber se a aquisição foi satisfatória é fazer plots de seções de afastamento constante, tais como a seção de afastamento mínimo (ver figura 2.6), médio e máximo, visualizar alguns CDPs para conferir as coordenadas ou qualquer família desejada.

Uma importante etapa para processos posteriores é a visualização da família de CDPs para analisar quais possuem cobertura máxima. Esta etapa está diretamente relacionada a com a análise de velocidade do dado sísmico. Para iniciar a análise de velocidade, o dado de entrada deve estar organizado (sorteado) em famílias de CDPs. O programa susort (ver anexo) é um programa do SU que faz esse sorteamento de forma rápida.

A figura 2.4 mostra a superposição das coordenadas tiro-receptor (s, g), ponto médio-afastamento (y, h) e geometrias do traçamento de raios para vários tipos de “coleta” (gather). As coordenadas (y, h) foram rotacionadas 45° em relação às coordenadas (s, g). A área pontilhada representa a cobertura usada no registro do perfil sísmico ao longo do eixo do ponto médio, Oy . Cada ponto representa um traço sísmico com o eixo do tempo perpendicular ao plano do papel.

O comprimento do cabo de gravação é representado pelo segmento FG e o comprimento da linha é AD . O número de pontos ao longo do eixo de afastamento (seção transversal 3) é igual à cobertura máxima CDP. A cobertura máxima diminui nas extremidades do perfil (segmentos AB e CD). A cobertura máxima completa ao longo da linha está nos pontos médios do segmento BC . O diagrama da 2.4 é conhecido como gráfico de empilhamento e é útil ao configurar a geometria de uma linha para pré-processamento. Se faltar um tiro ou um receptor ruim, os pontos médios afetados são facilmente identificados. Para a maioria das geometrias de gravação, a cobertura máxima n_f para um CDP é fornecida por:

$$\begin{aligned} n_f &= \frac{n_g \Delta g}{2 \Delta s}, \\ &= \frac{96 * 25}{2 * 50}, \\ &= 24 \end{aligned} \tag{2.20}$$

Onde Δg e Δs são os intervalos do grupo de receptores e dos disparos, respectivamente, e n_g é o número de canais de gravação.

O gráfico de cobertura máxima (figura 2.34) pode ser gerado a partir do programa sukeycount onde o parâmetro de entrada é a chave CDP e o dado sísmico (ver anexo), com o auxilio do matlab gera-se o gráfico de cobertura máxima que indica quais CDPs serão posteriormente utilizados na análise de velocidade.

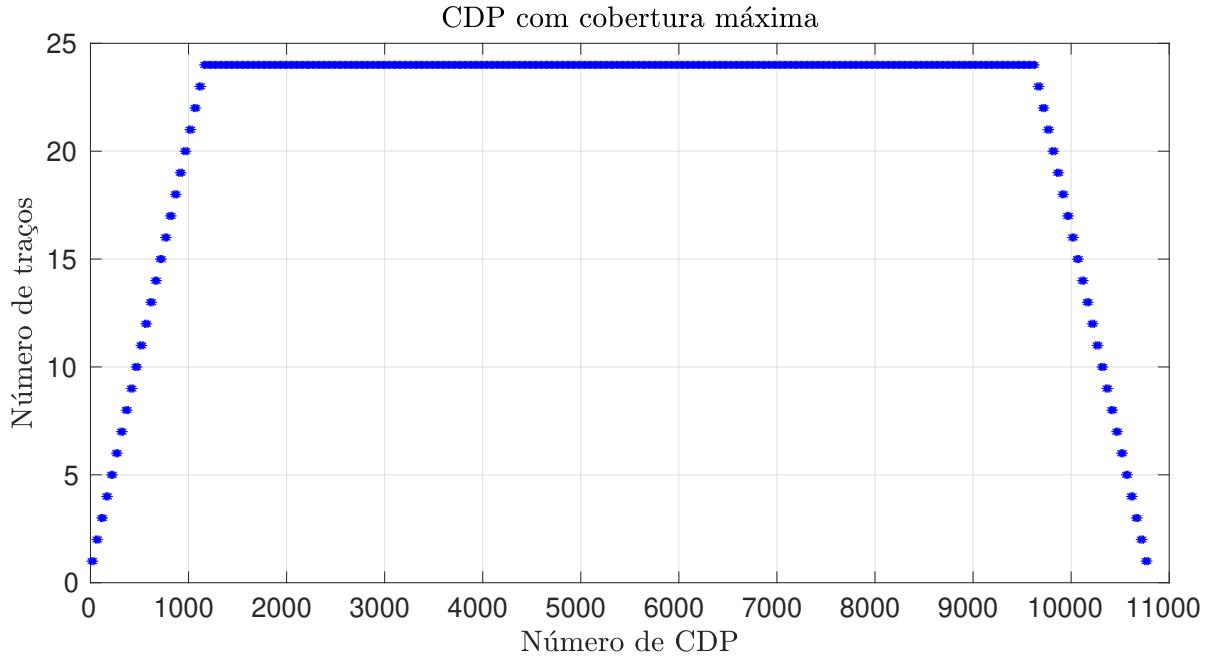
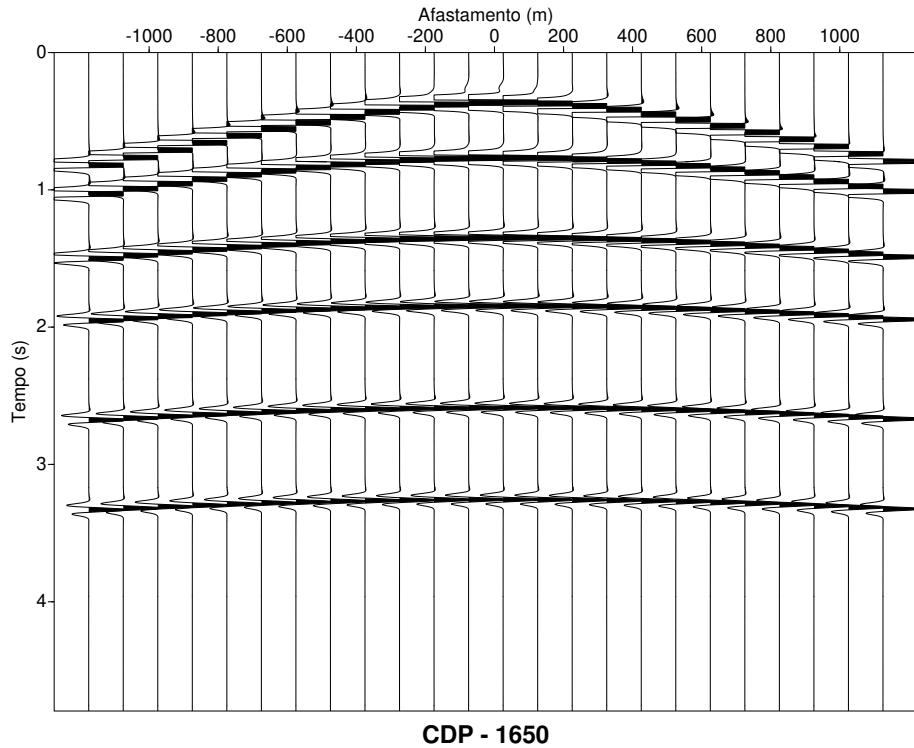


Figura 2.34: Máxima cobertura de família CDPs entre o CDP de número 1150 e 9650. No intervalo de cobertura máxima o número de traços é 24 por CDP.

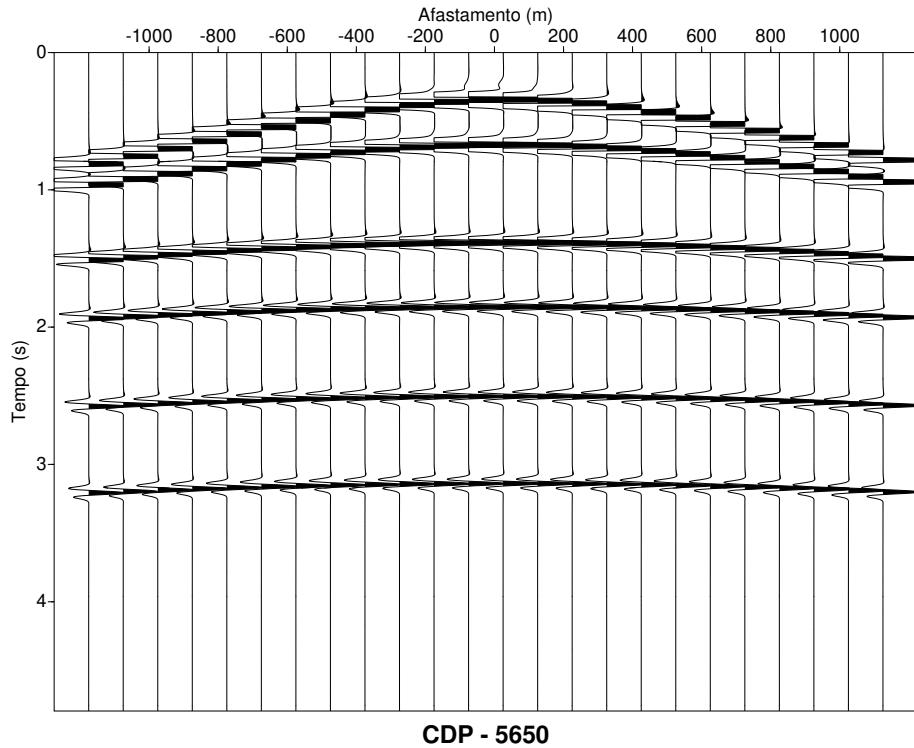
A equação que descreve o tempo de trânsito para uma reflexão de uma família CDP tem a seguinte forma:

$$t^2 = t_0^2 + \frac{x^2}{v^2} \quad (2.21)$$

- x = Corresponde ao afastamento entre fonte-receptor.
- t_0 = Tempo (duplo) de ida e volta do sinal até o refletor no afastamento fonte-receptor igual a zero ($x = 0$).
- v = Velocidade de propagação estimada do meio.

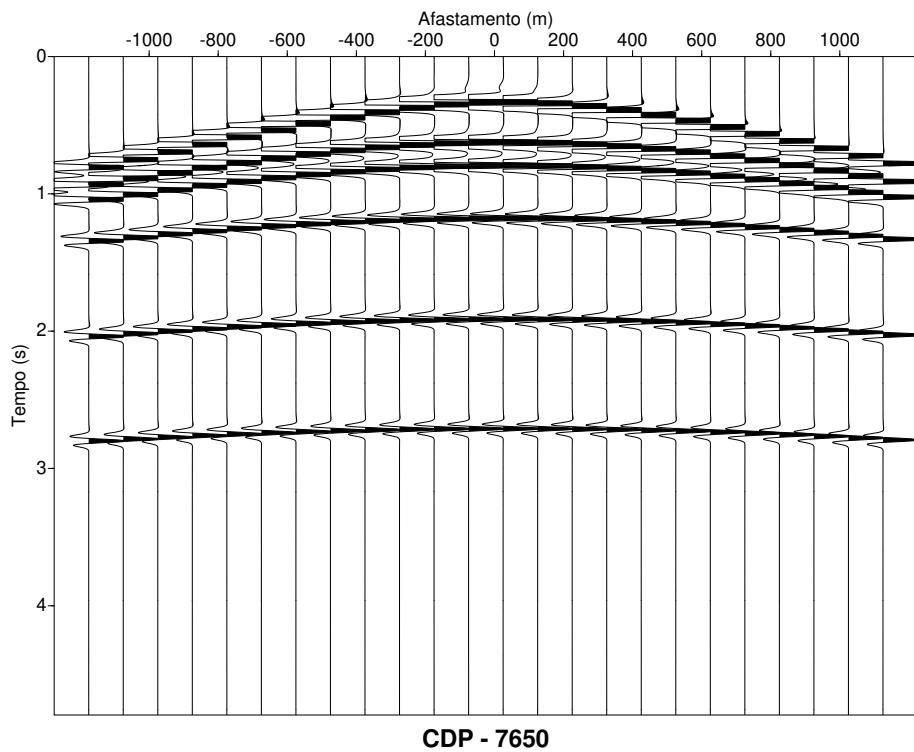


(a) CDP - 1650

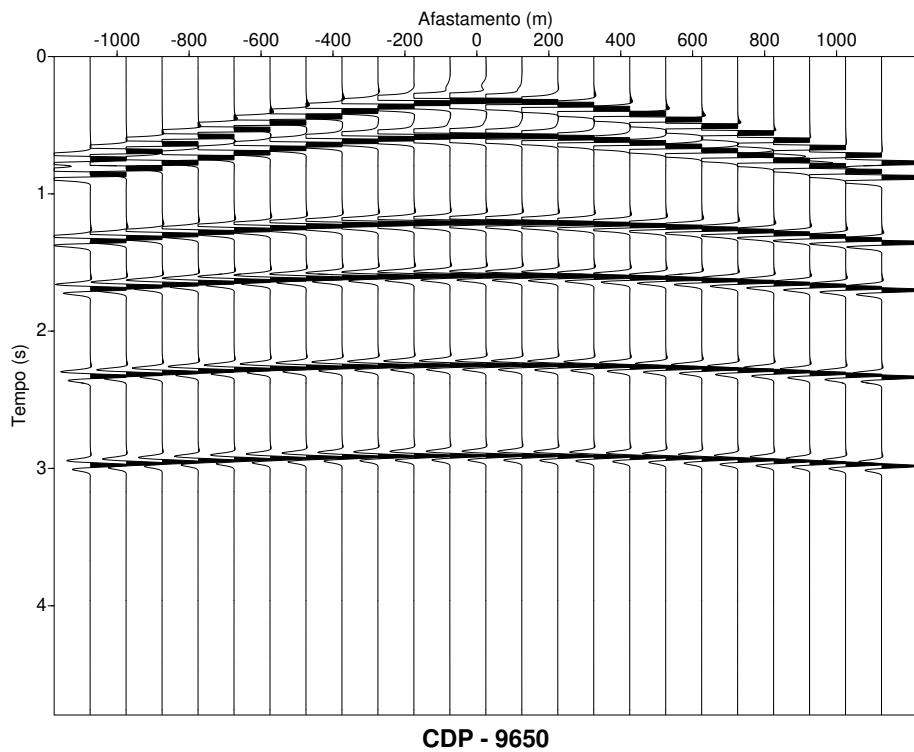


(b) CDP - 5650

Figura 2.35: CMP é um ponto em subsuperfície amostrado por diversos pares fonte-receptor. Esse conjunto de pares fonte-receptor recebe a denominação de família CDP ou CMP. A técnica foi inicialmente descrita para o caso de camadas horizontais plano-paralelas e com a velocidade do meio variando apenas no sentido vertical. Desde que foram estabelecidos no começo da década de 1950, os métodos common depth point (CDP), assim como os diferentes tipos de métodos de migração, são considerados a principal referência para o processamento de dados sísmicos.



(a) CDP - 7650



(b) CDP - 9650

Figura 2.36: A técnica CDP é uma maneira empregada em levantamento de reflexão sísmica para resolver problemas de baixa qualidade do imageamento de áreas com baixa razão sinal/ruído. Nesta técnica os pontos de subsuperfície são registrados redundantemente com diferentes afastamentos fonte-receptor. A seção empilhada é obtida sobrepondo-se os traços redundantes, após a correção de sobre-tempo normal NMO.

2.7 ANÁLISE DE VELOCIDADE E CORREÇÃO NMO

A análise de velocidades é uma técnica que está diretamente relacionada ao sucesso do processamento sísmico. Quanto melhor for a estimativa do campo de velocidades mais representativa será o imageamento da subsuperfície.

Há diversos tipos de velocidades definidas nos trabalhos sísmicos: velocidade intercalar, velocidade RMS, velocidade de NMO e velocidade de empilhamento são apenas algumas das mais importantes. Todas elas apresentam a sua importância, porém a velocidade de empilhamento é a de maior importância no contexto de análise de velocidade para imageamento.

Com o dado organizado em famílias CDP (figura 2.4), sabe-se que um CDP contém traços de um mesmo ponto de subsuperfície imageado várias vezes, com diferentes trajetos. Então utilizando informações, do tempo de trânsito para os diferentes trajetos e o afastamento fonte - receptor, é possível estimar a velocidade do meio. (Yilmaz e Doherty, 2000).

A análise de velocidade não é realizada com apenas um CDP, e sim com um conjunto de CDPs (CDPs de máxima cobertura) que são então interpoladas para os CDPs restantes.

A realização da análise de velocidade deve ser feita preferencialmente por um intérprete. A etapa de picagem dos pontos envolve a seleção valores de velocidade em função do tempo, para ser utilizada em processos subsequentes. Quando realizada a análise de velocidade, busca-se sempre obter uma melhor imagem no empilhamento, em que são tolerados erros na interpretação da velocidade. Assim, precisa-se de bastante tempo para realizar essa tarefa, que apresenta grande potencial para erros, especialmente quando o realizador da tarefa não conhece muito a respeito da geologia do local.

A exatidão e resolução dos valores da velocidade de empilhamento dependem dos fatores de aquisição do dado, tais como tamanho do afastamento (offset), cobertura, banda de frequências da aquisição, razão sinal-ruído entre outros.

Para o trabalho em questão, algumas ferramentas foram utilizadas para auxiliar na análise de velocidade. Estas ferramentas estão descritas a seguir.

A estimativa da distribuição de velocidade (modelo de velocidade) na seção sísmica usa o dado organizado em família CDP, a medida de coerência *semblance*, faz-se a correção sobre-tempo-normal (NMO), realiza-se então o empilhamento da seção e posteriormente a aplica-se a técnica de migração. A correção NMO é baseada no modelo de camada plano-horizontal, cujo tempo de trânsito de reflexão de uma onda primária tem forma hiperbólica definida pelo sobre-tempo-normal em relação ao afastamento $x = 0$, Δt_{NMO} , e expressa por:

$$\Delta t_{NMO}(h, t(0), v_{NMO}) = t(x) - t(0); \quad (2.22)$$

que a partir da Eq. (2.21) fica na forma:

$$\Delta t_{NMO}(h, t(0), v_{NMO}) = t(0) \left\{ \left[1 + \left(\frac{x}{t(0)v_{NMO}} \right)^2 \right]^{\frac{1}{2}} - 1 \right\}. \quad (2.23)$$

Na aplicação de Δt_{NMO} se busca a horizontalizar os eventos hiperbólicos em relação à $t(0)$ ajustando o parâmetro v_{NMO} . A estimativa de velocidade é realizada acoplada ao mapa semblance, $S(v_{NMO}, t_0; x_0)$, que mede a coerência no intervalo $[0, 1]$, e é dada por (Sguazzero e Vesnauer, 1987):

$$S(v_{NMO}, t_0; x_0) = \frac{\sum_{t=t_0-\delta t/2}^{t_0+\delta t/2} \left[\frac{1}{N_h} \sum_{h=h_0}^{\delta h} \bar{u}[t(h, v_{NMO}); x_0] \right]^2}{\sum_{t=t_0+\delta t/2}^{t_0+2\delta t/2} \frac{1}{N_h} \sum_{h=h_0}^{\delta h} [\bar{u}[t(h, v_{NMO}); x_0]]^2}, \quad (2.24)$$

onde $\bar{u}[t(h, v_{NMO}); x_0]$ é a amplitude do traço ao longo da trajetória de empilhamento, \sum_t e \sum_h definem as janelas temporal-espacial dentro da qual se ajusta a curva que melhor representa o evento de reflexão, e N_h é o número de traços envolvidos. Os pares (v_{NMO}, t_0) devem ser marcados no mapa semblance em conjunto com a análise dos eventos de reflexão, e cada evento é relacionado a um par que melhor o horizontaliza (ver figura 2.37), assim se forma o modelo de velocidades NMO utilizado, primeiramente, no empilhamento e, posteriormente, na migração.

Para o dado de arranjo split-spread, a análise de velocidade foi feita em 18 CDPs (1150, 1650, 2150, 2650, 3150, 3650, 4150, 4650, 5150, 5650, 6150, 6650, 7150, 7650, 8150, 8650, 9150, e 9650), ou seja, a partir do CDP de cobertura máxima 1150 foi dado um espaçamento regular de 500 CDPs até chegar ao CDP 9650, pois o CDP 10788 (último CDP de cobertura máxima) não está incluso para este intervalo.

Para cada CDP é gerado um semblance (medida de coerência multicanal), (Sheriff e Geldart, 1995), onde as regiões em vermelho representam as velocidades mais coerentes para horizontalizar os refletores. Sendo assim, o usuário deve apontar o cursor do mouse na região onde achar que corresponde a velocidade mais representativa, e pressionar a letra “s” no teclado. Este comando salvará os picks correspondentes aos pares de velocidades v_{NMO} e tempos t_{NMO} .

Em seguida, depois de ter salvado todos os picks de um determinado CDP é necessário pressionar a tecla “q”, para finalizar o processo de picagem.

Após o processo de picagem onde escolhe-se os pares de velocidades v_{NMO} e tempos t_{NMO} pode-se então aplicar a correção de sobre-tempo-normal. O programa que realiza a correção de NMO no SU em um dado sísmico é o sunmo (ver anexo). Os parâmetros de entrada para a correção NMO no SU são os seguintes:

- t_{NMO} - Tempo NMO correspondente para velocidades v_{NMO} .
- v_{NMO} - velocidades NMO correspondentes aos tempo t_{NMO} .
- CDP - CDPs para os quais v_{NMO} e t_{NMO} são especificados.

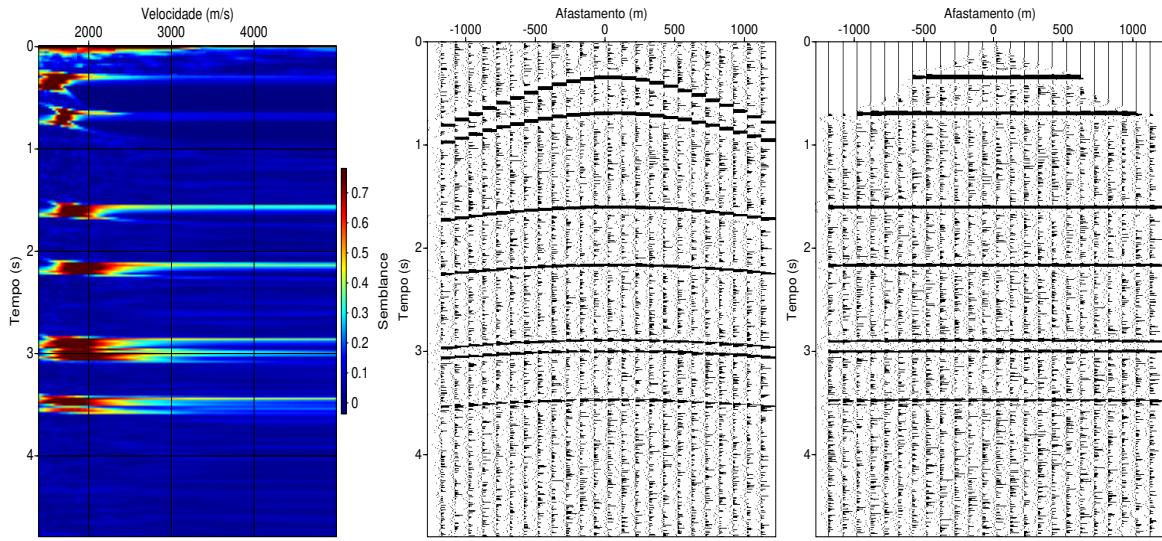


Figura 2.37: CMP - 4650: mapa semblance (à esquerda), antes da correção NMO (centro) e após a correção NMO (à direita).

Conhecido os pares de velocidades v_{NMO} e tempos t_{NMO} de todos os CDPs analisados no processo de picagem gera-se os modelos de velocidade NMO (ver anexo). A figura 2.38 e figura 2.39 mostram respectivamente os modelos de velocidade NMO e velocidade NMO suavizado em tempo utilizado no empilhamento do dado.

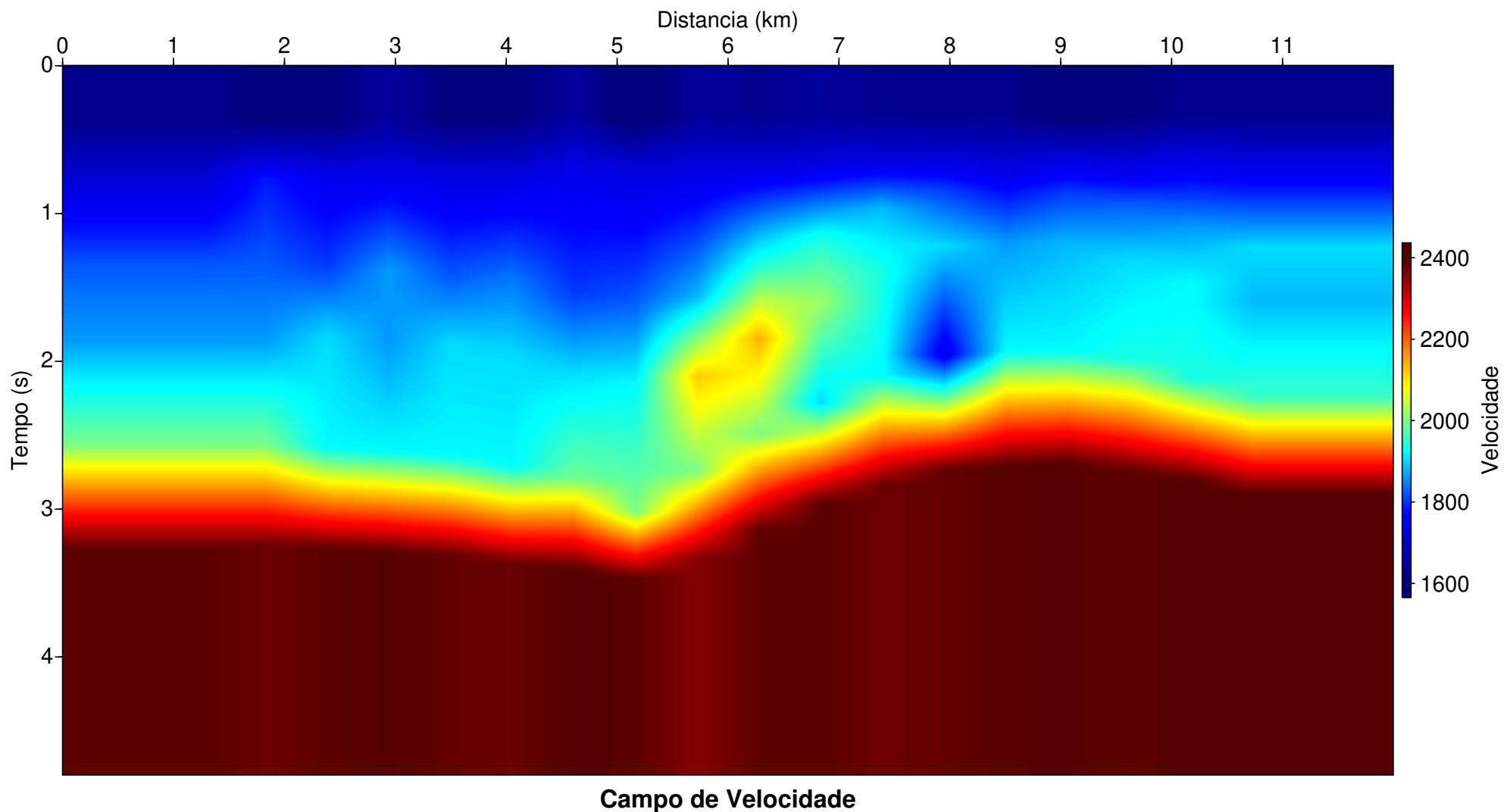


Figura 2.38: Modelo de velocidade em tempo não suavizado obtido no SU. A cor azul escuro está relacionada as camadas de baixas velocidades e a cor vermelha escura as camadas de altas velocidades (camadas mais próximas do embasamento consequentemente mais profundas).

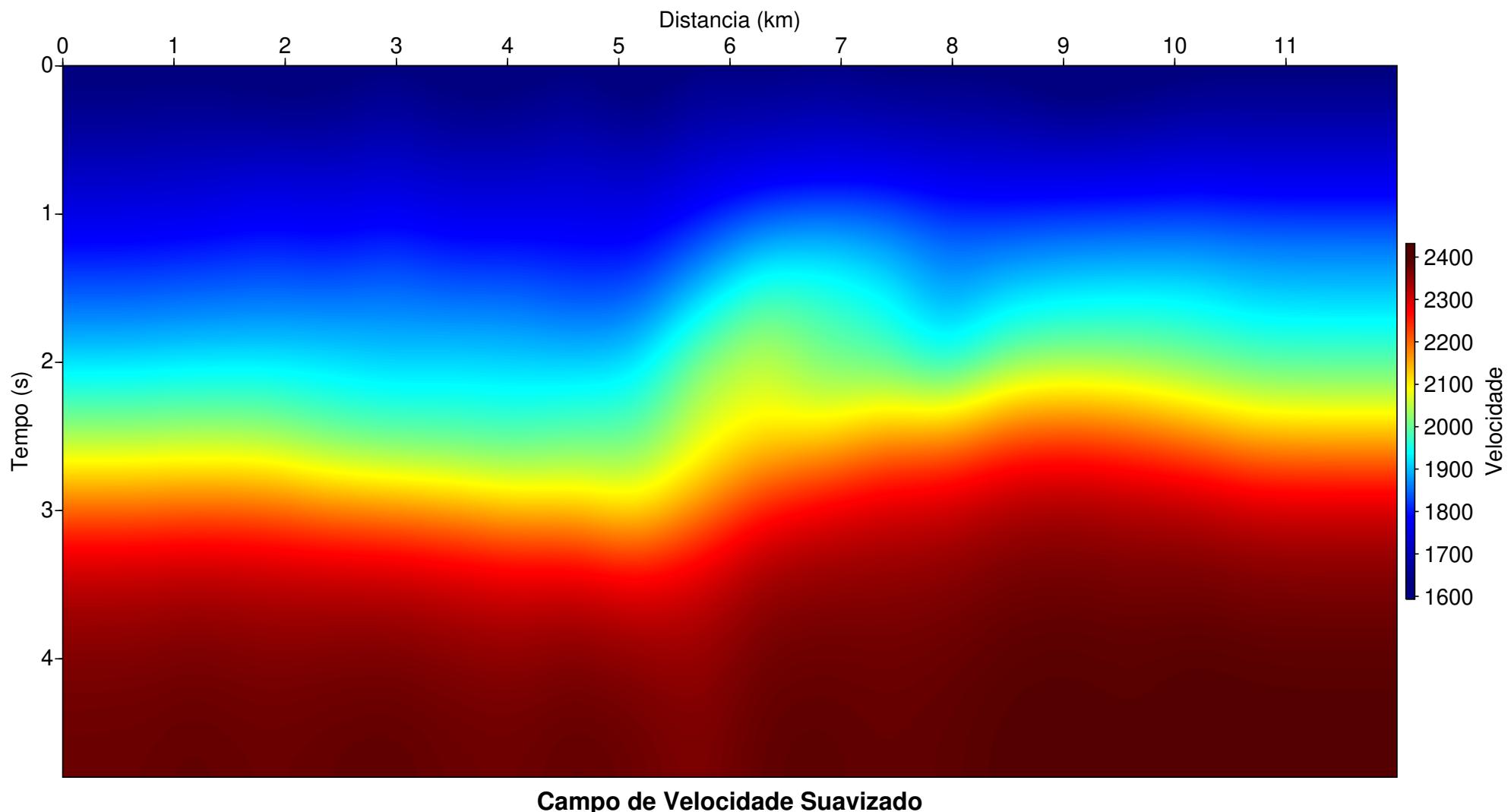


Figura 2.39: Modelo de velocidade em tempo suavizado obtido no SU. A cor azul escuro está relacionada as camadas de baixas velocidades e a cor vermelha escura as camadas de altas velocidades (camadas mais próximas do embasamento consequentemente mais profundas).

2.8 EMPILHAMENTO

O objetivo do empilhamento sísmico é simular um experimento sísmico de incidência normal, ou seja simular uma aquisição onde fonte e receptores (geofones) estivessem na mesma posição, este procedimento só é possível quando o dado de entrada esta organizado em família CDP. O empilhamento sísmico consiste na soma de todos os traços das famílias CDP, após a correção de NMO, que gerará um traço resultante. Essa etapa é realizada após a correção de NMO para que os eventos estejam horizontalizados e possam ser somados de forma construtiva. A soma construtiva recebe o nome de STACK e a imagem obtida é chamada de seção empilhada. (Vasconcellos, 2009)

Para que os traços da seção obtenham maior coerência dos eventos registrados o valor da amplitude somada e dividida pelo número de traços. Os primeiros eventos são somados em fase (construtivamente) e os outros eventos coerentes (ruídos) serão somados de forma destrutiva. Através do processo de empilhamento o ruído aleatório é atenuado em \sqrt{N} , onde N é o número de canais empilhados em conjunto.

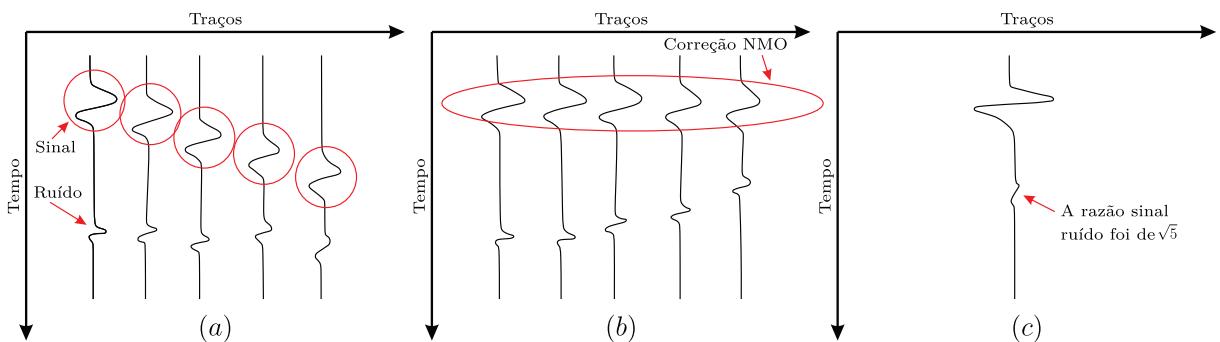


Figura 2.40: a) Cinco canais antes do empilhamento com ruído, b) Os cinco canais após a correção NMO, c) Canais já empilhados e com atenuação do ruído de \sqrt{N} . Onde N é o no de canais. Figura adaptada de (Vasconcellos, 2009)

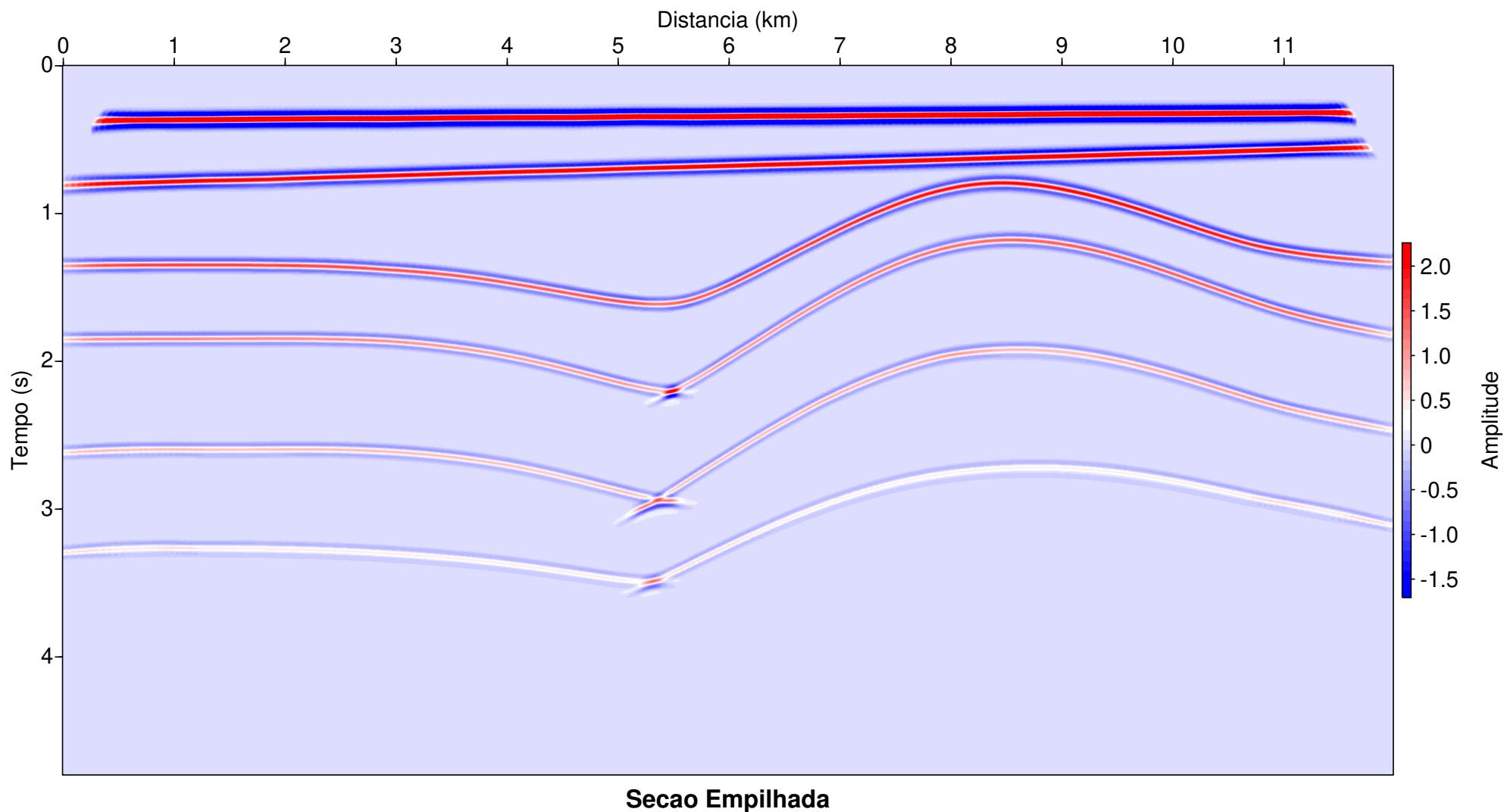


Figura 2.41: Seção empilhada

3 MIGRAÇÃO

A finalidade da migração é mover os eventos de reflexão registrados para a posição real na seção sísmica, de modo que a análise visual tenha uma correspondência direta com a estrutura do subsolo geológico. Esta operação pode ser realizada no domínio do tempo (domínio sísmico), ou do espaço denominada de profundidade (domínio geológico). Devido a esta proposta fundamental, migração é também denominada de imageamento.

A migração pós-empilhamento no tempo é um processo de imageamento de seções sísmicas empilhadas, e objetivo é de focalizar a energia das difrações e reposicionando os refletores para a posição correta, o que resulta em aumentar a continuidade e a resolução lateral dos refletores.

A migração no tempo leva em consideração o modelo “refletor em explosão”, o que admite variações laterais suaves em subsuperfície.

A migração das seções afastamento-nulo foi realizada com o método denominado de Migração Kirchhoff, onde utilizamos o programa *sumigtk* (ver anexo) do pacote CWP/SU (Cohen, 2017).

3.1 MIGRAÇÃO KIRCHHOFF NO TEMPO

A migração Kirchhoff parte da solução da equação de onda na forma escalar dada por

$$\nabla^2 P(\mathbf{r}, t) - \frac{1}{v^2} \frac{\partial^2 P(\mathbf{r}, t)}{\partial t^2} = -4\pi q(\mathbf{r}, t), \quad (3.1)$$

onde $P(\mathbf{r}, t)$ é o campo de onda (amplitude), v a velocidade constante do meio, $-4\pi q(\mathbf{r}, t)$ representa a fonte, e $\mathbf{r} = \mathbf{r}(x, y, z)$ representa o ponto de observação.

A solução completa para a equação (3.1), para um volume sísmico V_0 delimitado por uma superfície S_0 , é expressa pelo teorema de Green (Schneider, 1978), o que é dada por

$$P(\mathbf{r}, t) = \frac{1}{4\pi} \int_{t_0} dt_0 \int_{S_0} dS_0 \left[G \frac{\partial}{\partial n} P(\mathbf{r}_0, t_0) - P(\mathbf{r}_0, t_0) \frac{\partial}{\partial n} G \right], \quad (3.2)$$

onde $\mathbf{n} = n\hat{\mathbf{n}}$ é o vetor normal à superfície S_0 , que inclui a superfície de aquisição A_0 e a superfície de forma semi-esférica A' que é extrapolada para o infinito de forma que sua contribuição seja desprezível (ver Figura 3.1).

Desta forma, os valores na fronteira se reduzem a uma integral na superfície de aquisição, e à função auxiliar de Green, $G(\mathbf{r}, t | \mathbf{r}_0, t_0)$, que consiste da resposta do meio a uma fonte pontual em \mathbf{r}_0 . Neste caso a obtenção da função de Green utiliza o método das imagens, que para uma superfície plana localiza a imagem no ponto \mathbf{r}'_0 , e para um semi-espacôo ela é dada por

$$G(\mathbf{r}, t | \mathbf{r}_0, t_0) = \frac{\delta(t - t_0 - \frac{R}{v})}{R} - \frac{\delta(t - t_0 - \frac{R'}{v})}{R'}, \quad (3.3)$$

onde

$$R = [(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2]^{\frac{1}{2}}, \quad (3.4)$$

$$R' = [(x - x_0)^2 + (y - y_0)^2 + (z + z_0)^2]^{\frac{1}{2}}. \quad (3.5)$$

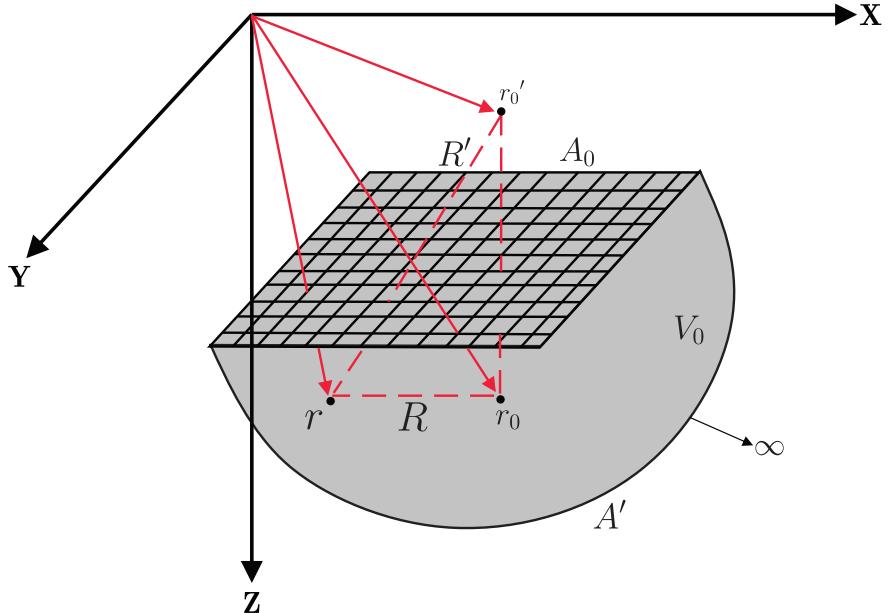


Figura 3.1: Meio escalar (3D) com volume V_0 delimitado pela fronteira $S_0 = A_0 + A'$, com um ponto fonte incluso em \mathbf{r}_0 , sua imagem em \mathbf{r}'_0 , e um ponto de observação em \mathbf{r}' .
Fonte: Modificado de (Schneider, 1978).

Na construção sísmica do problema, o campo $P(\mathbf{r}_0, t)$ é medido na fronteira $S_0 = A_0 + A'$, onde a função de Green se anula ($G = 0$), e é uma forma de eliminar a componente $\frac{\partial P(\mathbf{r}_0, t_0)}{\partial n}$ na integral, e se admite que \mathbf{n} corresponde à vertical \mathbf{z} ; ou seja, $n \rightarrow z_0$ (disto vem o conceito de migração em profundidade). Sendo assim, a equação (3.2) é simplificada à forma

$$P(\mathbf{r}, t) = \frac{1}{2\pi} \int_{t_0} dt_0 \int_{A_0} dA_0 \left\{ P(\mathbf{r}_0, t_0) \frac{\partial}{\partial z_0} \left[\frac{\delta(t - t_0 - \frac{R}{v})}{R} \right] \right\}, \quad (3.6)$$

sendo denominada de integral de Kirchhoff para a migração em profundidade.

Resolvendo a parte temporal da equação (3.6) que contém a função Delta de Dirac, e

depois substituindo convenientemente $\frac{\partial}{\partial z_0}$ por $\frac{\partial}{\partial z}$, resulta em

$$P(\mathbf{r}, t) = -\frac{1}{\pi} \frac{\partial}{\partial z} \int_{A_0} dA_0 \frac{P(\mathbf{r}_0, t - \frac{R}{v})}{R} \quad (3.7)$$

Esta representação mostra que a equação (3.1) é solução da equação da onda em virtude da forma $\frac{f(t - \frac{R}{v})}{R}$ do integrando.

Uma forma de representar uma seção sísmica empilhada é através do experimento físico denominado *refletor-em-explosão*. Neste modelo onde os receptores são localizados numa superfície de aquisição, e as fontes ao longo das interfaces refletoras onde são acionadas simultaneamente no instante de tempo $t = 0$, de onde o campo produzido se propaga até a superfície de aquisição segundo o Princípio de Huygens (ver Figura 3.2). Neste modelo, as equações ao longo desta seção devem ter suas velocidades alteradas ou o tempo pelo fator 1/2, e o resultado é a velocidade de migração (ver Figura 3.3).

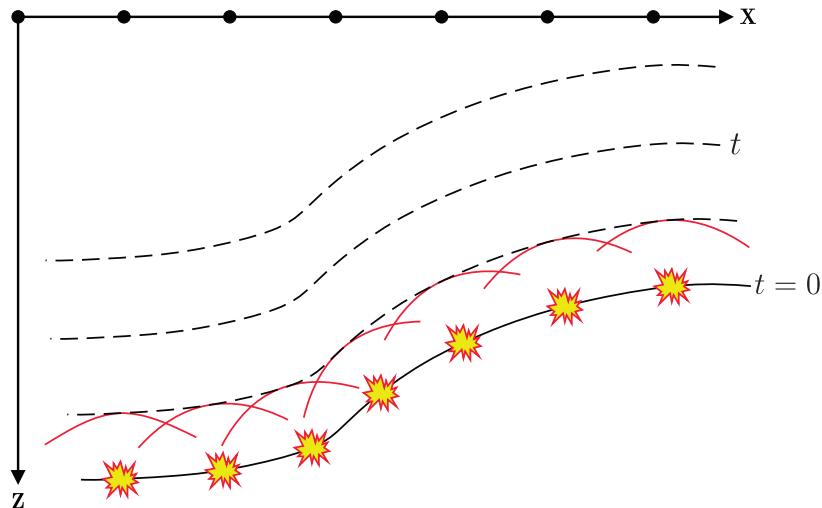


Figura 3.2: Representação do modelo refletor-em-explosão (em 2D para facilitar o desenho). As fontes estão localizadas nas interfaces refletoras são acionadas simultaneamente. O campo produzido se propaga de acordo com o Princípio de Huygens até a superfície de aquisição $z = 0$. Fonte: Modificado de (Schneider, 1978)

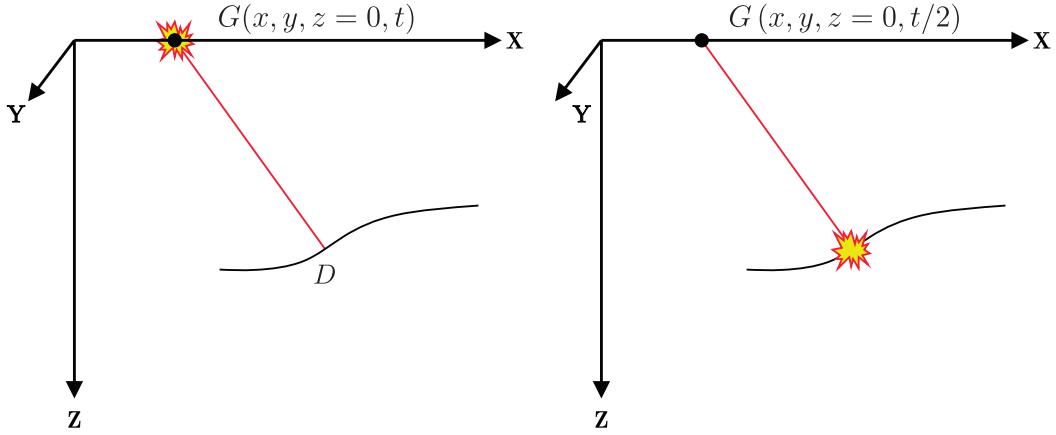


Figura 3.3: Modelo afastamento-nulo (esquerda) e do *refletor-em-explosão* (direita). Na esquerda o campo de onda parte da superfície no instante $t = 0$, reflete em D e retorna a superfície onde é registrado no tempo de percurso duplo (desce-e-sobe), t . Na direita se tem outra forma de representar o afastamento-nulo equivalente, onde o campo de onda parte de um ponto da subsuperfície em $t = 0$ e é registrado na superfície no tempo $t/2$. O conceito diz que a velocidade do campo de onda no modelo afastamento-nulo (esquerda) é a metade da velocidade do campo de onda no modelo do *refletor-em-explosão* (direita). Fonte Modificado de (Schneider, 1978).

Dada uma seção empilhada pode-se continuar em profundidade o dado registrado $P(x, y, z = 0, t)$, o que é representado pela equação

$$P(x, y, z, t) = -\frac{1}{2\pi} \frac{\partial}{\partial z} \int_{A_0} dA_0 \frac{P(x, y, z = 0, t + \frac{R}{v})}{R}. \quad (3.8)$$

Usualmente se deseja modelar a magnitude para a fonte-em-explosão proporcional à refletividade da interface, e para isto se calcula a equação (3.8) para os pontos em subsuperfície no tempo $t = 0$ de acionamento das fontes. Fixando $t = 0$ e calculando a integral para toda área de interesse (x, y, z) , ou seja,

$$P(x, y, z, t = 0) = -\frac{1}{2\pi} \frac{\partial}{\partial z} \int_{x_0} \int_{y_0} dx_0 dy_0 \frac{P(x, y, z = 0, \frac{R}{v})}{R}. \quad (3.9)$$

Este processo é denominado migração pós-empilhamento (seções afastamento-nulo, ou seções de incidência normal) baseado na integral de Kirchhoff, ou simplesmente migração Kirchhoff. A equação (3.9) corresponde ao modelo da seção migrada (continuada) conhecida como *condição de imagem* que mapeia o campo no domínio (x, y, z, t) para o domínio (x, y, z) , e que também pode ser mapeado para o domínio $(x, y, z, t = \frac{z}{v})$ correspondente à uma continuidade em profundidade (subsuperfície). A figura 3.4 exibe a relação entre o dado de entrada e o de saída no processo de mapeamento.

A entrada é um traço empilhado registrado no plano $z = 0$, e a saída é um traço em uma posição (x, y) plotado em função de z e do tempo $t = \frac{z}{v}$. Como os refletores se distribuem de cima para baixo através de sucessivas posições, imagea-se um ponto em

cada uma destas etapas, se calculando a integral para cada um ponto para o tempo $t = 0$. Exemplificando, o receptor r_1 em z_1 mapea um valor nulo para $t = 0$ devido o receptor não estar no ponto de reflexão, da mesma forma o valor é anulado para o receptor r_2 em z_2 . O valor numérico desta integral não será nulo quando o receptor estiver muito próximo ou em cima do refletor, como ocorre em z_3 , quando isto ocorre um valor não-nulo é relacionado a este determinado ponto na subsuperfície, este mapeamento é a última etapa do processo da migração Kirchhoff, e assim produzirá uma imagem migrada (por continuação para baixo).

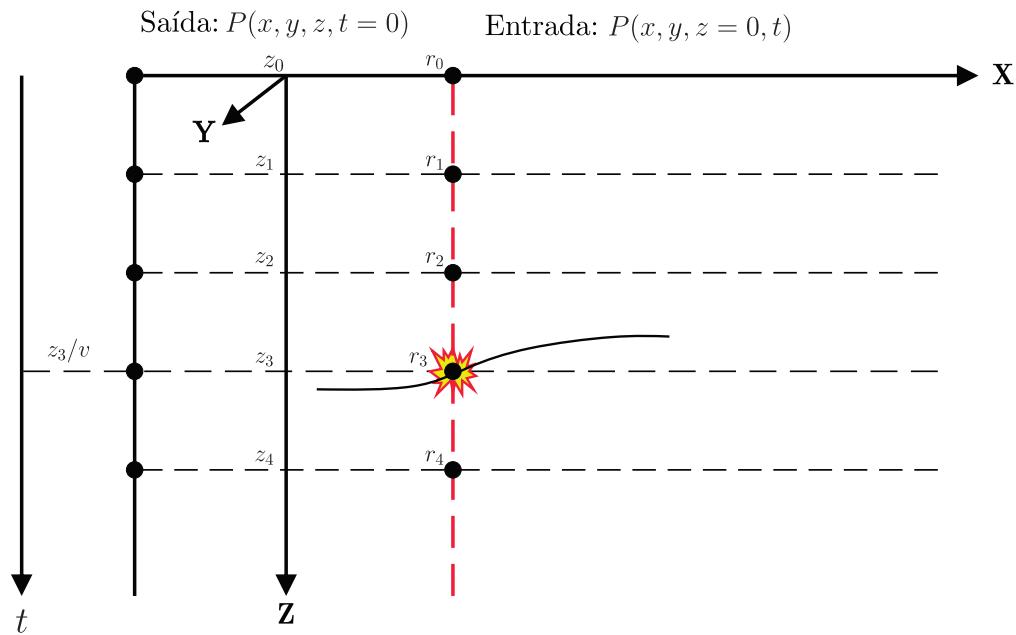


Figura 3.4: A relação entre o dado de entrada $P(x, y, z = 0, t)$ e o dado de saída $P(x, y, z, t = 0)$. O mapeamento do campo de onda sísmico (x, y, z, t) para $(x, y, z, t = \frac{z}{v})$. Observe que o modelo de velocidade admite apenas $v = v(z)$, mas na realidade $v = v(x, y, z)$. Fonte: Modificado de (Schneider, 1978).

Sendo assim, segundo o modelo do *refletor-em-explosão* e o processo de migração, a equação (3.9) é entendida como o processo que permite conhecer o valor do campo no tempo $t = 0$ a partir de seus valores registrados pelos receptores no tempo t . Ou também, como um processo de continuação do campo de onda $P(\mathbf{r}_0, t_0)$, conhecido na fronteira A_0 , para $P(\mathbf{r}, t = 0)$ em um ponto na subsuperfície.

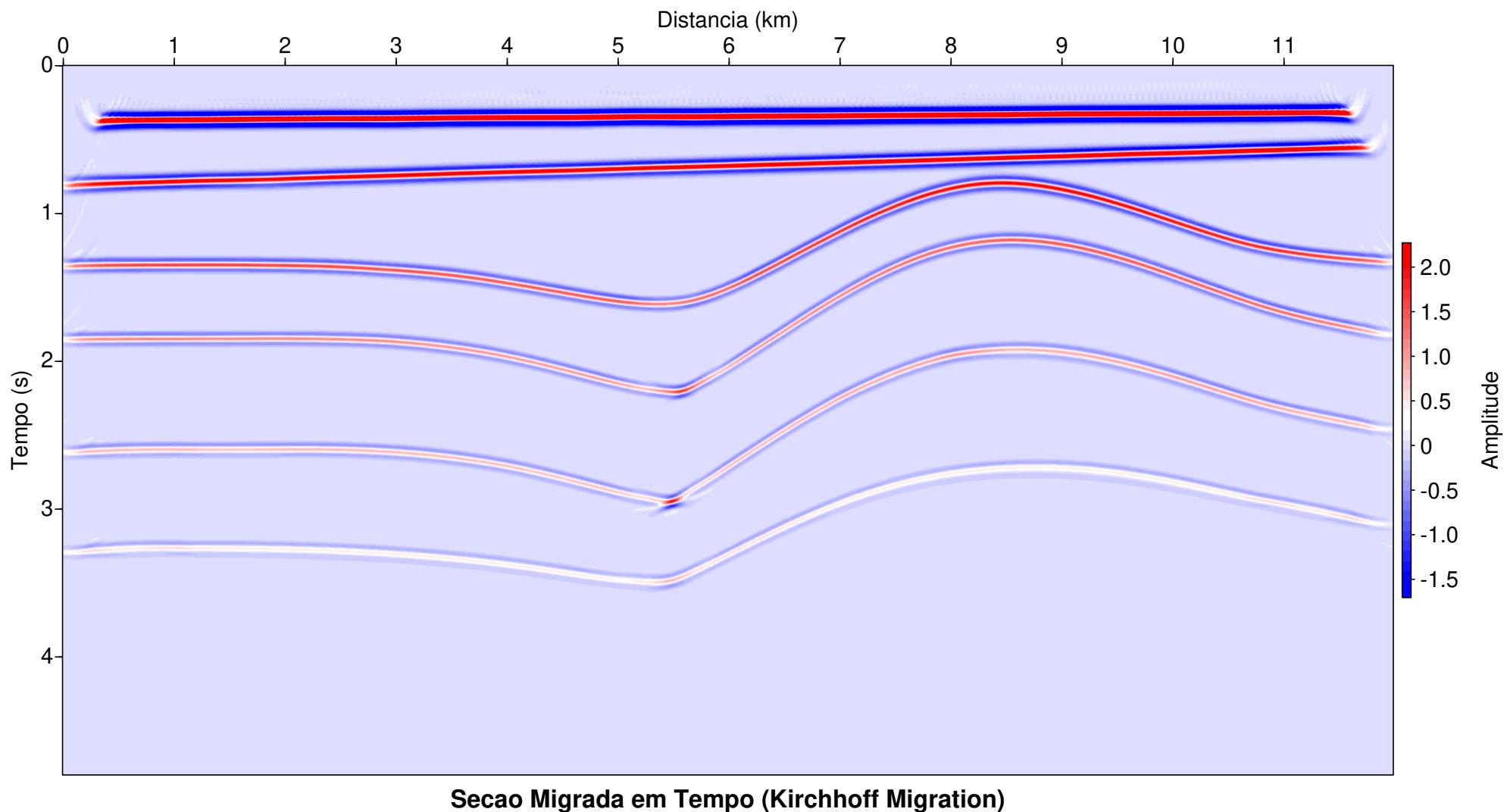
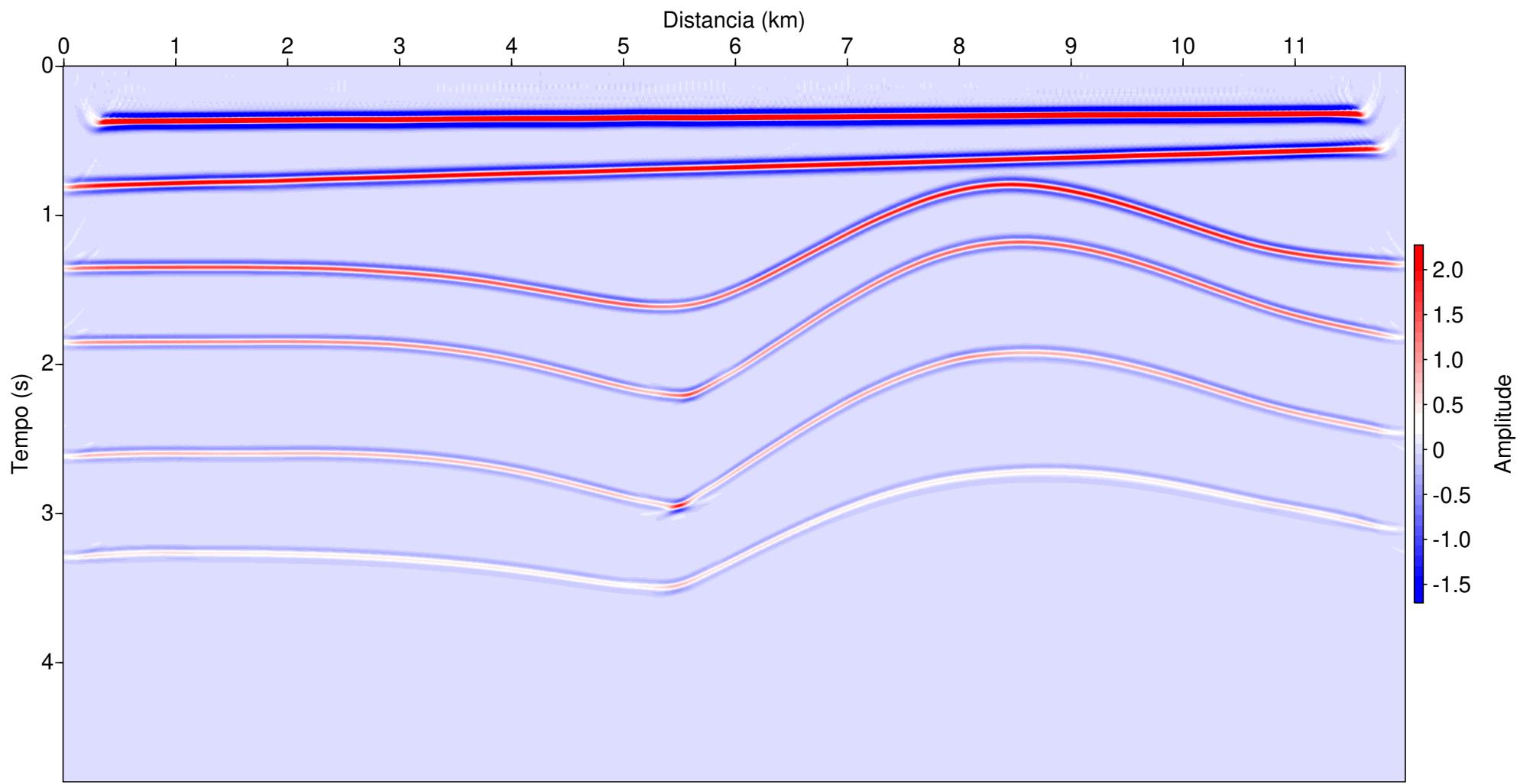


Figura 3.5: Seção migrada em tempo pelo método Kirchhoff utilizando o modelo de velocidade suavizado.



Secao Migrada em Tempo (Kirchhoff Migration) - M.E

Figura 3.6: Seção migrada em tempo pelo método Kirchhoff utilizando o modelo exato de velocidade (figura 2.3).

3.2 MIGRAÇÃO DIFERENÇAS FINITAS EM PROFUNDIDADE

Migração é o processo que cria uma imagem da subsuperfície a partir de dados sísmicos registrados em superfície, idealmente reposicionando os dados registrados em sua verdadeira posição geológica em subsuperfície. Existem duas abordagens principais para executar a migração: migração em tempo e migração em profundidade, que podem ser feitas após o processo de empilhamento ou antes do empilhamento.

A migração por diferenças finitas foi desenvolvida e popularizada por J. F. Claerbout, da Universidade de Stanford, e agora é amplamente utilizada no processamento sísmico. Para a maioria das seções, a migração de diferenças finitas fornece resultados comparáveis aos obtidos pela migração convencional de Kirchhoff e, onde os eventos não estão diminuindo muito, uma aparência mais limpa geralmente é aparente. No entanto, existem duas limitações práticas para o método, e elas ocorrem em regiões de mergulho muito acentuado e onde há uma grande variação da velocidade na direção lateral. (Hood, 1978)

O ponto de partida para a migração das ondas é normalmente a equação da onda escalar. Em um sistema de coordenadas em que o tiro e o receptor são coincidentes (ou seja, aproximadamente em uma pilha CDP), a velocidade deve ser reduzida pela metade (Sherwood, 1976), de modo que nessas coordenadas a equação da onda seja:

$$\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial z^2} = \frac{4}{c^2} \frac{\partial^2 P}{\partial t^2} \quad (3.10)$$

onde $P(x, z, t)$ é a pressão registrada nos hidrofones, x, z, t são as coordenadas horizontal, vertical e de tempo, respectivamente, e $c(x, z)$ é a velocidade de ondas.

A equação de onda unidirecional que governa a propagação de ondas ao longo da direção do eixo z é (no domínio da frequência):

$$\frac{\partial P}{\partial z} = i \left(\frac{4\omega^2}{c^2} + \frac{\partial^2}{\partial x^2} \right)^{1/2} P = iSP \quad (3.11)$$

onde ω é a frequência angular.

A chave para o sucesso de várias aproximações parabólicas da equação (3.11) reside na precisão com a qual o termo raiz quadrada S é aproximado. Claerbout (1970) deriva um conjunto de aproximações de várias ordens, e na figura 3.7 as aproximações de segunda e terceira ordem para a raiz quadrada são mostradas. Estes são chamados as aproximações de 15° e 45° , respectivamente.

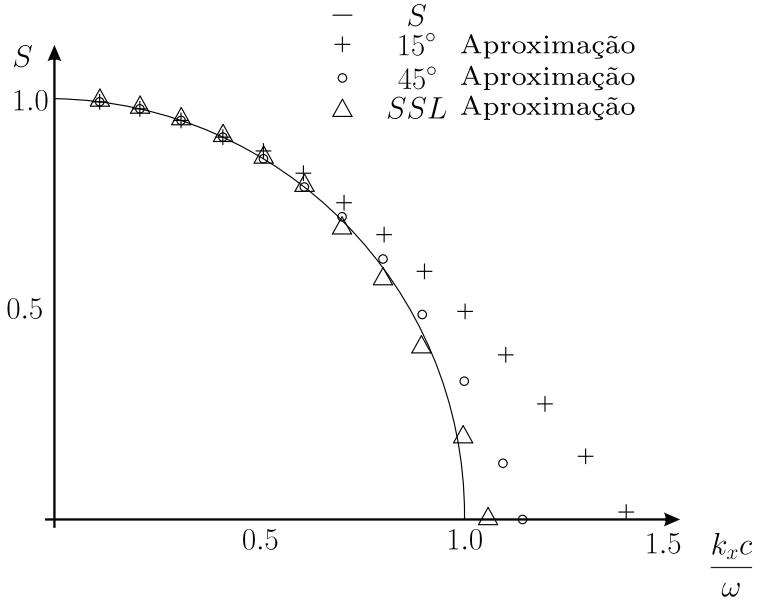


Figura 3.7: Aproximações para o termo S da equação (3.11). Figura modificada de (Hood, 1978).

É possível usar aproximações de quarta ordem ou superiores a S , mas há várias desvantagens computacionais decorrentes do fato de que as equações de diferenças finitas resultantes não têm mais uma forma tridiagonal direta. Isso causa um aumento progressivo nos custos computacionais. Além disso, os erros introduzidos pelo método das diferenças finitas podem ser mais graves do que os causados pela equação da onda parabólica, de modo que, a menos que o método das diferenças finitas seja abandonado em favor de outras técnicas, parece haver poucas perspectivas de aproximações de alta ordem que atinjam qualquer melhoria notável (Hood, 1978).

Voltando agora à equação (3.11), uma equação de 15° é derivada para referência posterior. Seja c uma velocidade constante. Nós temos então

$$\begin{aligned} \frac{\partial P}{\partial z} &= i \frac{2\omega}{c} \left[I - I + \left(\frac{\bar{c}}{c} \right)^2 + \left(\frac{\bar{c}}{2\omega} \right)^2 \frac{\partial^2}{\partial x^2} \right]^{1/2} P \\ &= i \frac{\omega}{\bar{c}} \left[I + \left(\frac{\bar{c}}{c} \right)^2 + \left(\frac{\bar{c}}{2\omega} \right)^2 \frac{\partial^2}{\partial x^2} \right] P \end{aligned} \quad (3.12)$$

Esta pode ser colocada em um período de tempo retardado, configurando $P = P' \exp(i2\omega z/\bar{c})$ portanto:

$$\frac{\partial P'}{\partial z} = i \frac{\omega}{\bar{c}} \left[\left(\frac{\bar{c}}{c} \right)^2 - I + \left(\frac{\bar{c}}{2\omega} \right)^2 \frac{\partial^2}{\partial x^2} \right] P' \quad (3.13)$$

Restaurando (3.13) o domínio do tempo e eliminando os primos, obtemos:

$$\frac{\partial^2 P}{\partial t \partial z} = \frac{\partial^2 P}{\partial t^2} \left(\frac{\bar{c}}{c^2} - \frac{1}{\bar{c}} \right) - \frac{\bar{c}}{4} \frac{\partial^2 P}{\partial z^2} \quad (3.14)$$

Um caso especial disso ocorre se as variações de velocidade lateral são insignificantes; nesse caso, negligenciando os efeitos de transmissão, a seguinte equação pode ser derivada:

$$\frac{\partial^2 P}{\partial z \partial t} = -\frac{c(z)}{4} \frac{\partial^2 P}{\partial x^2} \quad (3.15)$$

Seja $P_{j,k,n}$ o valor de P no ponto de malha $P(t_j, x_k, z_n)$. Os seguintes operadores de diferença podem ser definidos:

$$D_x P_{j,k,n} = (P_{j,k+1,n} - P_{j,k,n}) \frac{I}{\Delta x} \quad \left(\approx \frac{\partial P}{\partial x} \right),$$

$$D_{xx} P_{j,k,n} = (P_{j,k+1,n} - 2P_{j,k,n} + P_{j,k-1,n}) \frac{I}{\Delta x^2} \quad \left(\approx \frac{\partial^2 P}{\partial x^2} \right),$$

organizando desta forma

$$A_1 P_{j,k,n} = \frac{1}{2} [(I - \theta) P_{j,k,n} + \theta P_{j,k,n+1} \\ + (I - \theta) P_{j+1,k,n} + \theta P_{j+1,k,n+1}],$$

e

$$A_2 P_{j,k,n} = \alpha P_{j,k+1,n} + (I - 2\alpha) P_{j,k,n} + \alpha P_{j,k-1,n}$$

A equação (3.15) pode ser expressa em notação de diferencial como:

$$\left(A_2 D_z D_t + \frac{c(z)}{4} D_{xx} A_1 \right) P_{j,k,n} = 0 \quad (3.16)$$

Para estabilidade da equação (3.16) $0 \leq \alpha \leq \frac{1}{4}$ e $0.5 \leq \theta \leq I$ é necessário. Valores de α na faixa $\frac{1}{12} \leq \alpha \leq \frac{1}{6}$ parece ser amplamente aceito (Claerbout, 1985).

Para inclusão dos efeitos da variação da velocidade lateral o método utilizado envolve uma conversão da seção em “tempo” para uma seção “profundidade”. Assim, em vez de usar o sistema de coordenadas deslocadas no tempo usual:

$$t' = t + \frac{z}{c}$$

uma coordenada de profundidade d é usada definida por:

$$d = \frac{c_a t}{2} + z,$$

onde $c_a t = \int_0^t c_{int} dt$

c_a é comumente referido como a velocidade média e c_{int} como a velocidade do intervalo. Desde que a velocidade do intervalo varie lentamente em função da x e z .

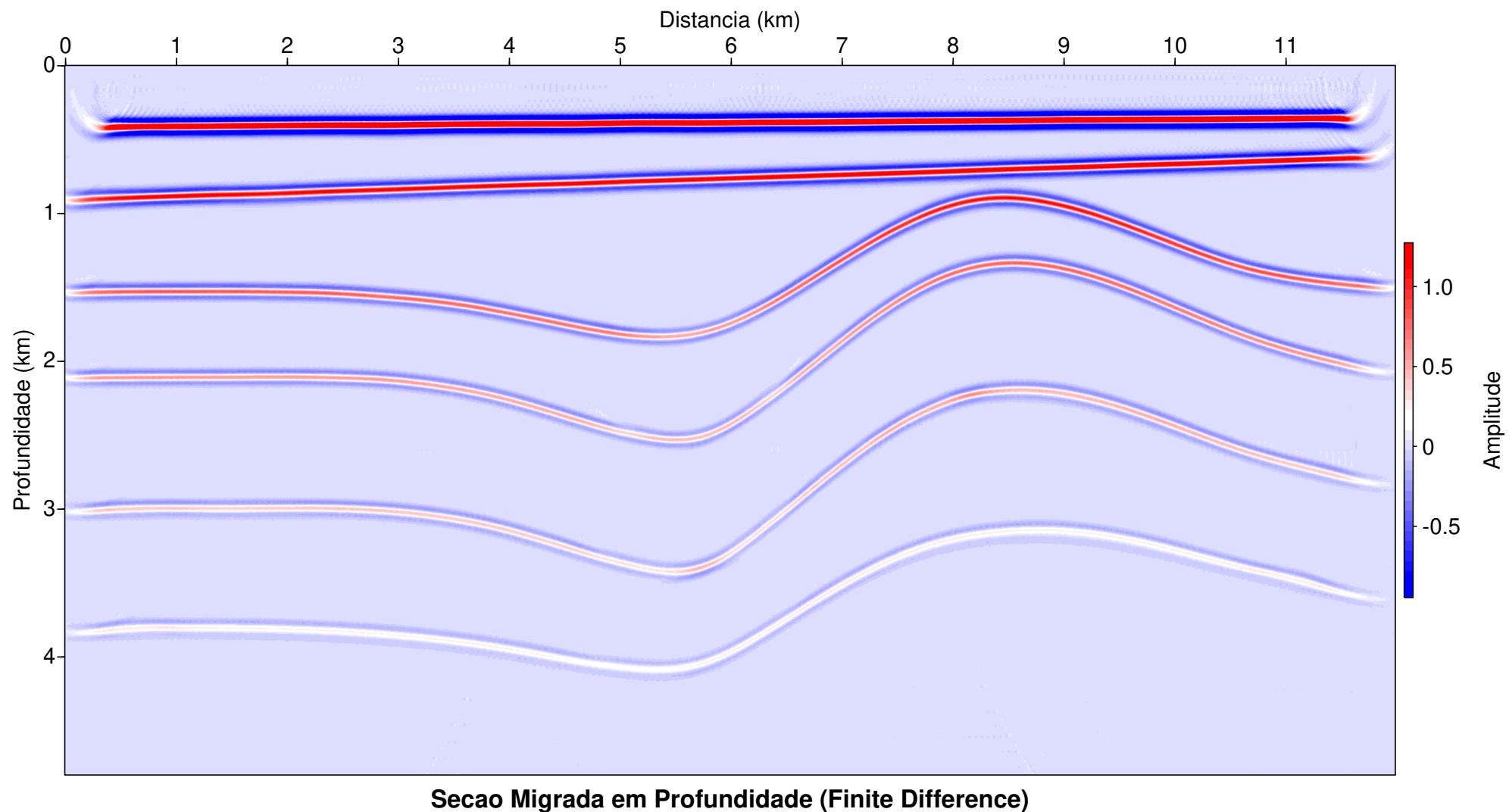


Figura 3.8: Seção migrada em tempo pelo método Diferenças Finitas utilizando o modelo de velocidade suavizado.

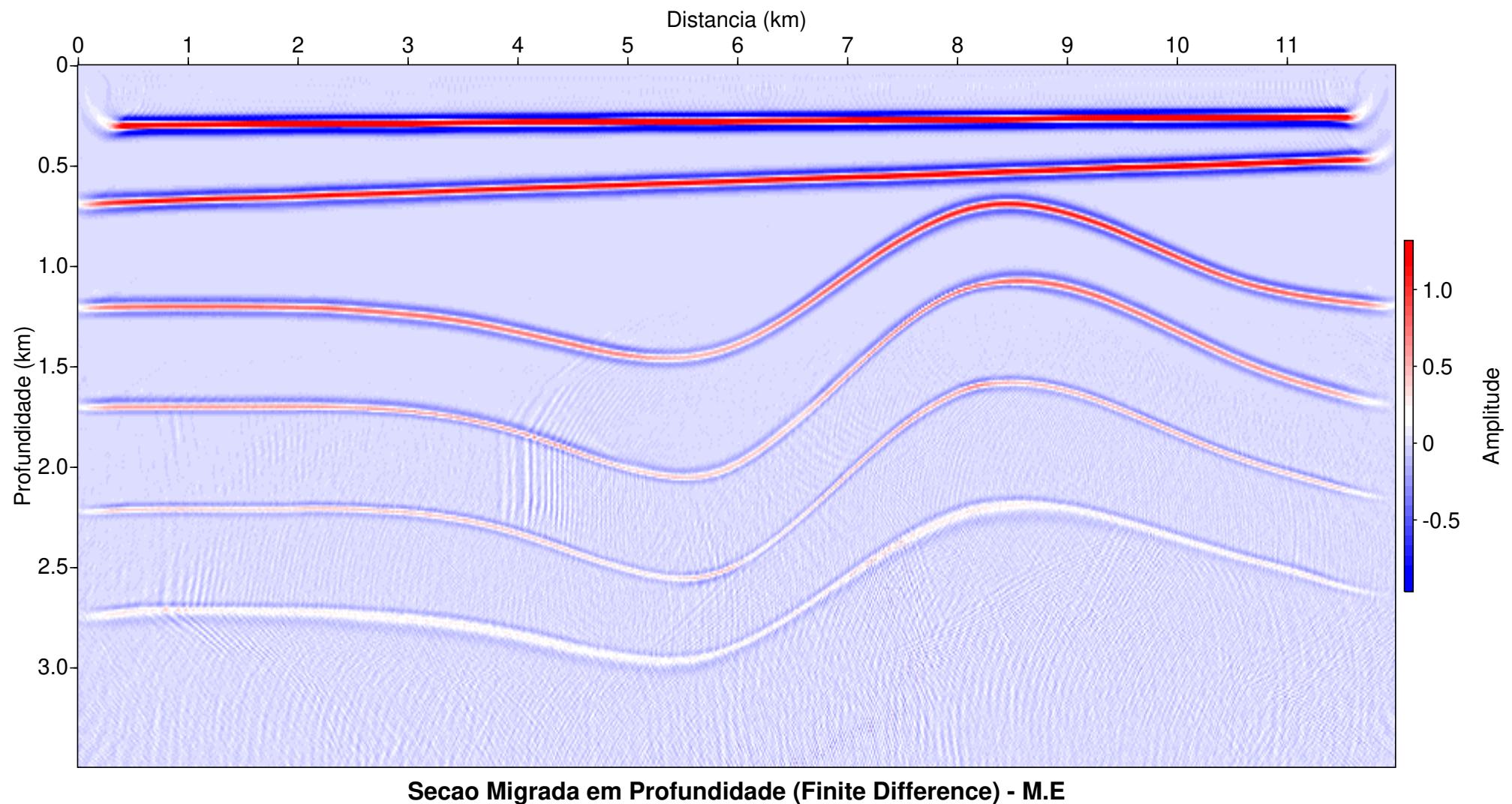


Figura 3.9: Seção migrada em tempo pelo método Diferenças Finitas utilizando o modelo exato de velocidade (figura 2.3).

3.3 OUTRAS METODOLOGIAS DE MIGRAÇÃO EM PROFUNDIDADE

Nesta seção propõe-se utilizar outras metodologias de migração em profundidade para comparação com o dado migrado por diferenças finitas.

A equação de onda escalar unidirecional é a base para algoritmos de migração comuns. Esses algoritmos não modelam explicitamente várias reflexões, ondas convertidas, ondas de superfície ou ruído. Qualquer energia presente na entrada de dados para migração é tratada como reflexões primárias. Os algoritmos de migração podem ser classificados em três categorias principais: (Yilmaz e Doherty, 2000)

1. Aqueles que se baseiam na solução integral para o equação de onda escalar,
2. Aqueles que são baseados nas soluções de diferenças finitas, e
3. Aqueles baseados em implementações de número de onda de frequência.

Qualquer que seja o algoritmo, deve:

1. Lidar com quedas acentuadas com precisão suficiente,
2. Lidar com variações de velocidade lateral e vertical, e
3. Ser implementado com eficiência.

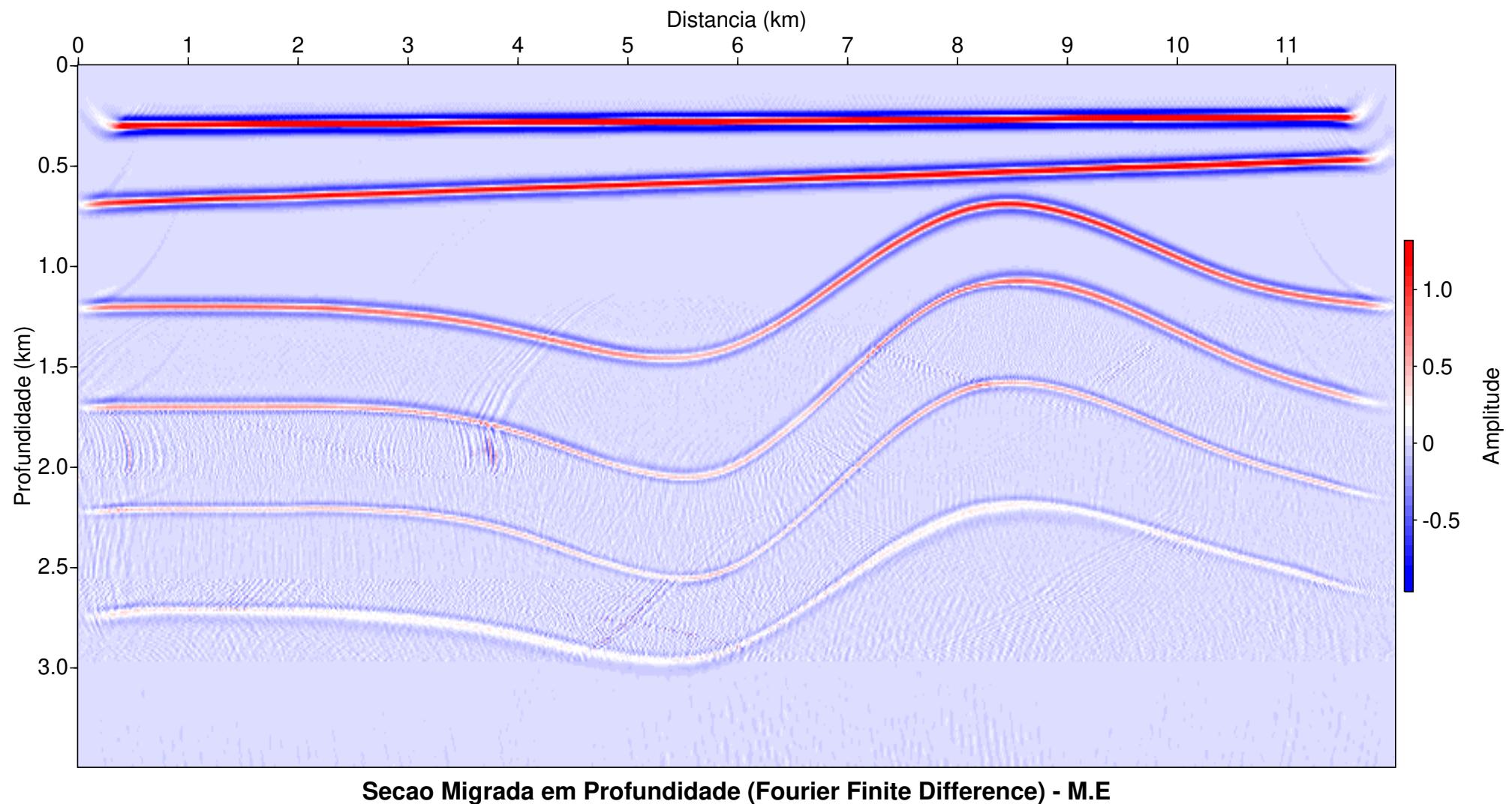


Figura 3.10: Seção migrada em tempo pelo método FFD utilizando o modelo exato de velocidade (figura 2.3).

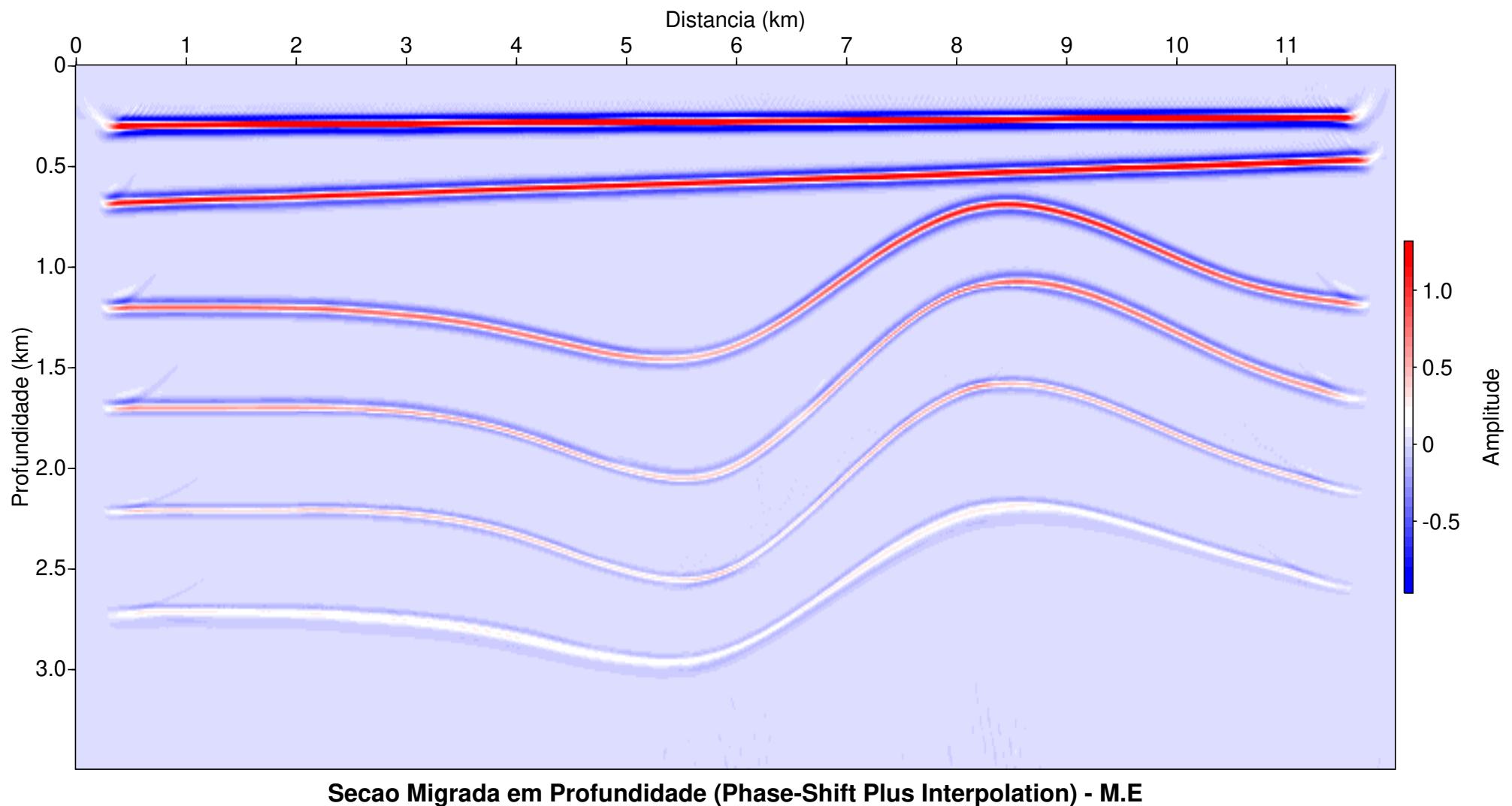


Figura 3.11: Seção migrada em tempo pelo método PSPI utilizando o modelo exato de velocidade (figura 2.3).

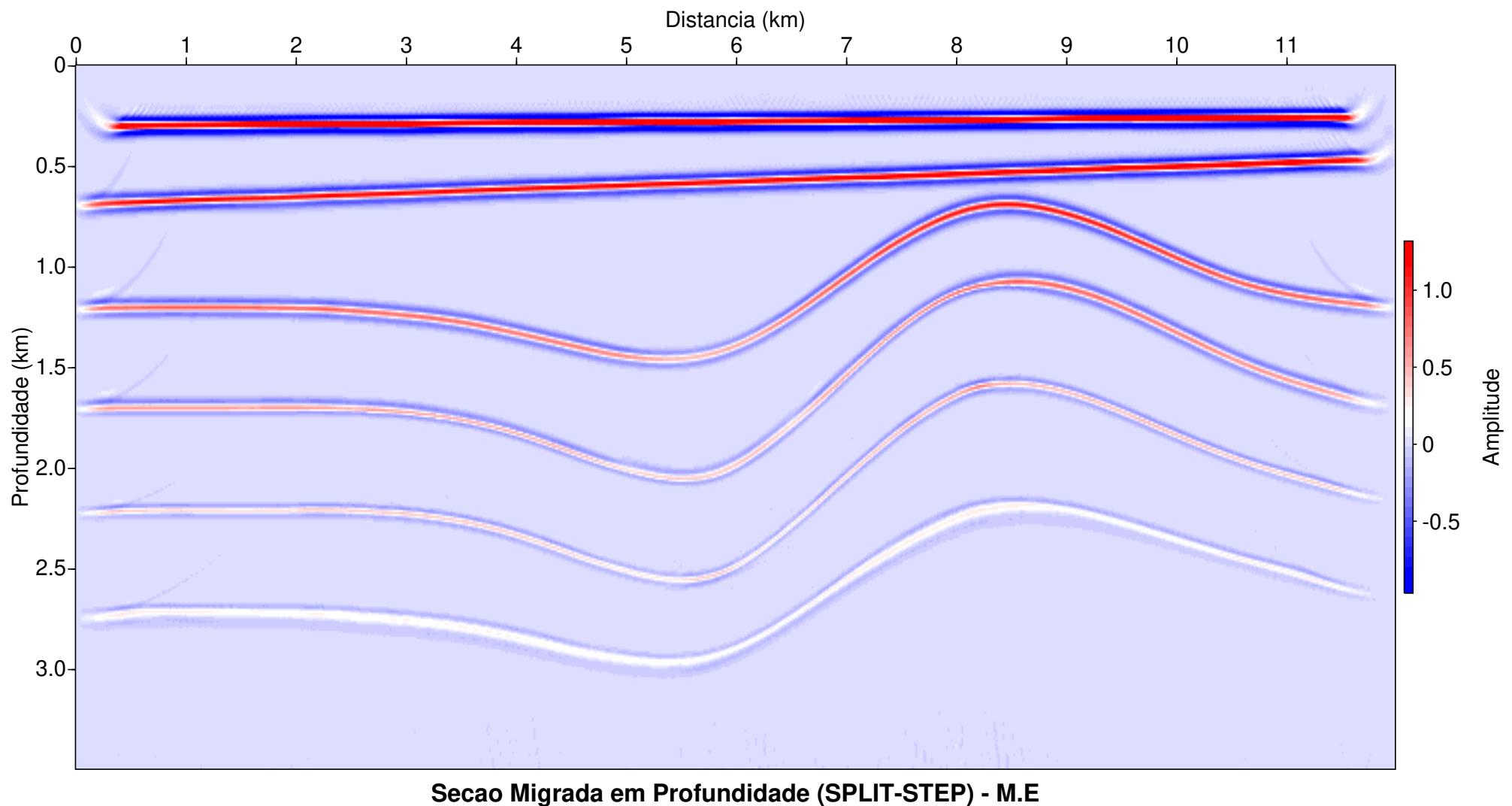


Figura 3.12: Seção migrada em tempo pelo método SPLIT STEP utilizando o modelo exato de velocidade (figura 2.3).

REFERÊNCIAS

- Claerbout, J. F., 1978, Snell waves: Stanford exploration project report.
- , 1985, Fundamentals of geophysical data processing (blackwell scientific publications, 19, illustrated edition ed.: Blackwell Science Inc.
- Cohen, J. K.; Stockwell, J. J. W., 2017, Cwp/su: Seismic un*x package - a free open seismic processing, research, and educational software package.: Colorado School of Mines: Center for Wave Phenomena.
- Edwards, S., 1988, Uses and abuses of seismic modeling: *Geophysics*, **7**, no. 4, 42–46.
- Forel, D., T. Benz and W. D. Pennington, 2005, Seismic data processing with seismic un*x: Society of Exploration Geophysicists.
- Hale, D. and J. K. Cohen, 1991, Triangulated models of the earth's subsurface: Colorado School of Mines, Center of Wave Phenomena, Report, **107**.
- Hood, P., 1978, Finite difference and wave number migration: *Geophysical Prospecting*, **26**, no. 4, 773–789.
- Leite, L. W. B., 2015, Conceitos da análise espectral de sinais em geofísica.: Instituto Nacional de Ciência e Tecnologia de Geofísica do Petróleo, INCT-GP.
- Schneider, W. A., 1978, Integral formulation for migration in two and three dimensions: *The Leading Edge*, **43**, no. 1.
- Sguazzer, P. and A. Vesnaver, 1987, A comparative analysis of algorithms for stacking velocity estimation: Blackwell Scientific Publications.
- Sheriff, R. E. and L. P. Geldart, 1995, Exploration seismology: Cambridge university press.
- Sherwood, D. L. L. R. R. J., 1976, The wave equation applied to migration: *Geophysical Prospecting*, **24**, no. 2, 380–399.
- Soares, A. G. O., 2011, Modelamento e Processamento de Dados Sísmicos Usando o Software Livre Seismic Unix. (Monografia - Bacharelado em Geofísica, UFRN - Universidade Federal do Rio Grande do Norte.).
- Thorson, J., 1978, Reconstruction of slant stacks: Stanford exploration project report.
- Vasconcellos, G. D. d., 2009, Processamento de dados sísmicos sintéticos de um modelo de sal utilizando o software spw. (Monografia - Bacharelado em Geofísica, Uff - Universidade Federal Fluminense, Niteroi, Rio de Janeiro.).

Yilmaz, Ö. and S. Doherty, 2000, Seismic data analysis: Processing, inversion, and interpretation of seismic data, 2nd ed.: Society of Exploration, volume **Vol2.** of Investigations in Geophysics, No. 10.

APÊNDICES

A SHELL SCRIPT MODELAGEM TRIMODEL

Este shell script foi escrito para utilizar a subrotina *trimodel* do pacote Seismic Unix (SU). O script gera a figura (2.2) do modelo de camadas curvas.

```
#!/bin/sh
trimodel xmin=0 zmin=0 xmax=12.0 zmax=3.5 \
1 xedge=0,2,4,6,8,10,12 \
zedge=0,0,0,0,0,0,0 \
sedge=0,0,0,0,0,0,0 \
2 xedge=0,12 \
zedge=0.3,0.25 \
sedge=0,0 \
3 xedge=0,12 \
zedge=0.7,0.45 \
sedge=0,0 \
4 xedge=0,2,4,6,8,10,12 \
zedge=1.2,1.2,1.3,1.4,0.7,1.05,1.2 \
sedge=0,0,0,0,0,0,0 \
5 xedge=0,2,4,6,8,10,12 \
zedge=1.7,1.7,1.8,2.0,1.1,1.45,1.8 \
sedge=0,0,0,0,0,0,0 \
6 xedge=0,2,4,6,8,10,12 \
zedge=2.2,2.2,2.3,2.5,1.6,1.9,2.2 \
sedge=0,0,0,0,0,0,0 \
7 xedge=0,2,4,6,8,10,12 \
zedge=2.7,2.7,2.8,2.9,2.2,2.4,2.7 \
sedge=0,0,0,0,0,0,0 \
8 xedge=0,2,4,6,8,10,12 \
zedge=3.5,3.5,3.5,3.5,3.5,3.5,3.5 \
sedge=0,0,0,0,0,0,0 \
sfill=0,0.1,0,0,0.39,0,0 \
sfill=0,0.4,0,0,0.31,0,0 \
sfill=0,0.9,0,0,0.28,0,0 \
sfill=0,1.5,0,0,0.25,0,0 \
sfill=0,2.0,0,0,0.55,0,0 \
sfill=0,2.4,0,0,0.44,0,0 \
sfill=0,3.0,0,0,0.16,0,0 \
kedge=1,2,3,4,5,6,7,8 \
>geomodel.bin

# Create a PS display of the model
spsplot < geomodel.bin > vagarosidade.eps gedge=0.5 gtri=2.0 gmin=0.2 ...
gmax=1.0 \
title="Modelo de camadas curvas" titlesize=20 labelz="Profundidade ... \
(km)" \
```

```
labelx="Distancia (km)" labelsize=18 dxnum=1.0 dznum=0.5 wbox=15 hbox=8  
exit
```

B SHELL SCRIPT MODELO DE VELOCIDADE EXATO

Este shell script foi escrito para utilizar a subrotina *tri2uni* do pacote Seismic Unix (SU) em conjunto com o Matlab. O script gera a figura (2.3) do modelo de camadas curvas.

```
#! /bin/sh
#-----
# Gera o modelo de velocidade exato em conjunto com o matlab (executa o ...
# matlab
# durante a conversao do modelo de vagarosidade para modelo uniforme de ...
# veloci
# dade)
# Suaviza o modelo gerado no matlab de acordo com os parametros r1 e r2.
#-----
input_model=geomodel.bin                      # binario gerado no trimodel
output_model=geomodel.tri                      # binario gerado na execucao
input_model2=modelo_velocidade.bin              # binario gerado no matlab
output_model2=model_velocidade_suaviado.bin    # binario gerado no SU

n1=350      # dimensao em z
n2=1200     # dimensao em x
d1=0.010    # incremento em z (350x0.010 = 3.5km dimensao em z do modelo)
d2=0.010    # incremento em x (1200x0.010 = 12km dimensao em x do modelo)
f1=0.0      # primeiro valor de amostra na dimensao 1
f2=0.0      # primeiro valor de amostra na dimensao 2
r1=5        # paramentro de suvizacao do modelo na direcao 1
r2=5        # paramentro de suvizacao do modelo na direcao 2

# comando para converter modelo de vagarosidade triangularizado ...
# (trimodel) para
# modelo uniforme.
tri2uni<$input_model n1=$n1 n2=$n2 d1=$d1 d2=$d2 f1=$f1 f2=$f2 ...
>$output_model

clear # limpa a tela na execucao do matlab
matlab -nodesktop -r "run $PWD/sloth2velocity.m" #-nodisplay

# suaviza o modelo de velocidade gerado no matlab
smooth2<$input_model2 n1=$n1 n2=$n2 r1=$r1 r2=$r2 >$output_model2
#-----
# figuras
outeps1=output_model.eps

psimage n1=$n1 <$output_model units="Vararosidade" perc=99 f2=0.0 ...
d2=0.010 \
```

```

d1=$d1 label1="Profundidade (km)" label2="Distancia (km)" \
title='Modelo de Vagarosidade convertido (Uniforme)' \
titlesize=20 legend=1 lstyle=vertright width=15.0 height=8.0 > $outeps1
#-----
outeps1=output_model2.eps

psimage n1=$n1 <$output_model2 units="Velocidade" perc=99 f2=0.0 ...
d2=0.010 \
d1=$d1 label1="Profundidade (km)" label2="Distancia (km)" \
title='Modelo de Vagarosidade Suavizado (Uniforme)' \
titlesize=20 legend=1 lstyle=vertright width=15.0 height=8.0 > $outeps1

clear
echo "arquivo compilado com sucesso !!!"
exit

```

SCRIPT DE CONVERSÃO DO MODELO DE VAGAOROSIDADE

Script executado para conversão do modelo de vagarosidade em velocidade. Este script de conversão é executado no shell script acima automaticamente.

```

clear;
clc;
close;

% Este script converte os valores de slot (S=1/V^2, V=velocidade em m/s)

% Este comando faz a leitura do arquivo binario no matlab, onde [350,1200]
% sao as dimensoes da matriz gerada no comando tri2uni.

H=fopen('geomodel.tri','r');
S=fread(H,[350,1200],'float');
fclose(H);

% Este comando faz o calculo de conversao de slot para velocidade em m/s
A=1./S;
B=sqrt(A);
V=1000.*B;

x=0:0.010:12000; % distancia em x
z=0:0.010:3500; % distancia em z

% Rho = zeros(350,1200);
% for i=1:1200
%     for j=1:350

```

```

%      if(V(j,i)==1500)
%          Rho(j,i)=1;
%
%      else
%          Rho(j,i)=(0.452+0.4788*V(j,i))/1000;
% %         Rho(j,i)=1.741*(V(j,i)^0.25);    % criacao do modelo de densidade.
%
%      end
%
%      end
%
%  end

figure(1)
imagesc(x,z,V)
grid on;
colorbar;
xlabel('Distancia (m)')
ylabel('Profundidade (m)')
title('Modelo de velocidade (m/s)')
set(gca,'FontSize',15);
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'modelo_de_velocidade','epsc')
saveas(1,'modelo_de_velocidade.png')

%
% figure(2)
% imagesc(x,z,Rho)
% grid on;
% colorbar;
% xlabel('Distancia (Km)')
% ylabel('Profundidade (Km)')
% title('Modelo de densidade')
% set(gca,'FontSize',15);
% set(gcf, 'Position', get(0, 'Screensize'));
% saveas(gcf,'modelo_de_densidade','epsc')
% saveas(2,'modelo_de_densidade.png')

% Este comando salva em formato binario o arquivo de velocidade ja ...
% convertido em m/s
J=fopen('modelo_velocidade.bin','w');
fwrite(J,V,'float');
fclose(J);
%
% Z=fopen('modelo_densidade.bin','w');
% fwrite(Z,Rho,'float');
% fclose(Z);
exit;

```

C SHELL SCRIPT AQUISIÇÃO TRISEIS

Este shell script foi escrito para utilizar a subrotina *triseis* do pacote Seismic Unix (SU). O script gera a seção (binário) para o processamento.

```
#!/bin/sh
/bin/rm -f temp*
# Assign values to variables
nangle=101
fangle=-65
langle=65
nt=1200 dt=0.004
datafile=geomodel.bin
seisfile=seis.su
#-----
# Shooting the seismic traces...
#-----

echo " ----Begin looping over triseis."
# Loop over shotpoints
i=0
while [ "$i" -ne "193" ]
do
    fs=`echo "$i * 0.05" | bc -l`
    sx=`echo "$i * 50" | bc -l`
    fldr=`echo "$i + 1" | bc -l`

    # Loop over receivers
    j=0
    while [ "$j" -ne "96" ]
    do
        fg=`echo "$i * 0.05 + $j *0.025" | bc -l`
        gx=`echo "$i * 50 + $j * 25 -1175" | bc -l`
        offset=`echo "$j * 25 -1175" | bc -l`
        tracl=`echo "$i * 96 + $j + 1" | bc -l`
        tracf=`echo "$j + 1" | bc -l`
        echo "sx=$sx gx=$gx flder=$fldr offset=$offset ...
trace_number=$tracl fs=$fs fg=$fg"

        # Loop over reflectors
        k=2
        while [ "$k" -ne "8" ]
        do
            triseis <$datafile xs=1.2,10.80 zs=0,0 \
            xg=0.025,11.975 zg=0,0 \
            nangle=$nangle fangle=$fangle langle=$langle \
```

```

kreflect=$k krecord=1 fpeak=12 lscale=0.5 \
ns=1 fs=$fs ng=1 fg=$fg nt=$nt dt=$dt |
suaddhead nt=$nt |
sushw key=dt,tracl,tracr,fldr,tracf,trid,offset,sx,gx \
a=4000,$trawl,$tracr,$fldr,$tracf,1,$offset,$sx,$gx >> temp$k

k=`expr $k + 1`
done

j=`expr $j + 1`
done

i=`expr $i + 1`
done

echo " ----End looping over triseis."

#-----
# Sum contents of the temp files...
#-----
echo " ----Sum files."
susum temp2 temp3 >tempa
susum tempa temp4 >tempb
rm -f tmpa
susum tempb temp5 >tempa
rm -f tmpb
susum tempa temp6 >tempb
rm -f tmpa
susum tempb temp7 >$seisfile
rm -f tmpb
#-----
# Clean up temp files...
#-----
echo " ----Remove temp files."
rm temp*
# Report output file
echo " ----Output file : $seisfile has been generated successfully ."
# Finishing shell script
echo " ----Finish!"
exit

```

D SHELL SCRIPT GEOMETRIA

Este shell script foi escrito para utilizar principalmente a subrotina *suchw* e *susort* do pacote Seismic Unix (SU). O script organiza a geometria da aquisição.

```
#!/bin/sh
# organizando geometria (inserindo chave cdp)

input1=seis.su          # binario da aquisicao (triseis)
outtemp=seis2.su         # binario convertido pra versao do usuario
outtemp2=seis_cdp.su     # temporario gerado na execucao
outtemp3=seis_cdp2.su    # temporario gerado na execucao
outtemp4=seis_cdp_sort.su # temporario gerado na execucao
#-----
# descomentar caso o binario desejado der erro na execucao
# converte o dado para a versao do su instalada na maquina
# ns= numero de amostras no tempo (ver script de aquisicao para ...
# descobrir ns)
#suswapbytes < $input1 > $outtemp ns=1200
#mv $outtemp $input1
#-----
# adicionando chave cdp ao dado su
suchw < $input1 key1=cdp key2=sx key3=gx a=0 b=1 c=1 d=2 > $outtemp2
# removendo cdps negativos (parametro a)
suchw key1=cdp key2=cdp key3=cdp a=588 b=1 c=0 d=1 < $outtemp2 >$outtemp3
# sobrescrevendo arquivos
mv $outtemp3 $outtemp2
# sort
susort < $outtemp2 > $outtemp4 cdp offset
# sobrescrevendo arquivos
mv $outtemp4 $outtemp2

# informacoes do dado com chave cdp
surange < $outtemp2

exit
```

E SHELL SCRIPT PLOT AFASTAMENTO MÍNIMO

Este shell script foi escrito para utilizar principalmente a subrotina *supsimage* do pacote Seismic Unix (SU). O script gera a figura (2.6) do modelo de camadas curvas.

```
#!/bin/sh
# Gera uma figura de uma seção em afastamento mínimo

file1=seis_cdp.su
eps1=seis_minoffset

dt=0.004
tmin=0.0
tmax=4.5

#wrgb=0,0,1.0 grgb=1.0,1.0,1.0 brgb=1.0,0,0
#wrgb=0.0,0.0,1.0 grgb=1.0,1.0,1.0 brgb=1.0,0.0,0.0

suwind key=offset min=-1175 max=-1175 < $file1 | supsimage d1=$dt f2=0 ...
d2=0.062176166 \
width=15.0 height=8.0 perc=99 legend=1 lx=17.5 units="Amplitude" ...
wrgb=0,0,1.0 grgb=1.0,1.0,1.0 brgb=1.0,0,0 \
labelfont=Times-Roman label1="Tempo (s)" label2="Distancia (km)" \
title="Afastamento Minimo" titlesize=20 > $eps1.eps

#suwind < seis.su key=offset min=-1175 max=-1175 > offset_min.su

#suwind < seis.su key=offset min=-1175 max=-1175 | supswigb > ...
wigb_plot.eps \
perc=99 nbpi=150 va=5 wbox=15.0 hbox=8.0 labelfont=Times-Roman \
label1="Tempo (s)" label2="CDP -597 ao 9012" title="Afastamento minimo ...
(1175 m)" titlesize=20

#12000/193 = 62.176165803 distancia em x do modelo/numero de tiros do ...
modelo (fldr)
#62.176165803/1000 = 0.062176166 distancia em km
```

F SCRIPT MATLAB TRANSFORMADA DE RADON LINEAR

Este script em matlab foi escrito para realizar a transformada de Radon linear. O script gera 2 figuras referentes ao CDP-1000.

```
% -----
% SLANT-STACK - LINEAR RADON TRANSFORM (TAU-P)
%
% Last updated date: 16-12-2019.
% -----
clear;
clc;
close all;

% Geometry parameters
dt = 0.004;
nt = 1200;
dx = 50;
nx = 20;
xmin = 50;

% Temporal and spatial axes
t = (0:nt-1)*dt;
x = xmin + (0:nx-1)*dx;

% Input Data: CDP gather
a = fopen('cdp_1000.bin','r');
cdp = fread(a,[nt,nx],'float');
fclose(a);
%
% -----
%
% LRT
[radon,tau,p] = LRT_forward(cdp,t,x);

figure(1)
radon_ida=figure(1);
subplot(1,2,1)
wigb(cdp,1,x,t)
hTitle=title('CDP 1000', 'Interpreter','latex','fontsize',16);
xlabel('Afastamento ($m$)', 'Interpreter','latex','fontsize',14)
ylabel('Tempo ($s$)', 'Interpreter','latex','fontsize',14)
set(hTitle, 'FontSize',16)
set(gca,'fontsize',14);
%
subplot(1,2,2)
```

```
wigb(radon,1,p,tau)
hTitle=title('Dom\'inio $\tau$-p', 'Interpreter','latex','fontsize',16);
xlabel('Inclina\c{c}\o (\$/m\$)', 'Interpreter','latex','fontsize',14)
ylabel('$\tau$ (\$/m\$)', 'Interpreter','latex','fontsize',14)
set(gcf,'color','w', 'Position', get(0, 'Screensize'));
set(hTitle, 'FontSize',16)
set(gca, 'fontsize',14);
export_fig('radon_ida','-pdf','-png');
% -----
%
hold on
npoints = 4;
idx_tau = zeros(1,npoints);
idx_p = zeros(1,npoints);

% Coordinate selection
for i = 1:npoints
    [idx_p(i), idx_tau(i)] = ginput(1);
end

np = length(p);
% Upper cut
tau_cut1 = linspace(idx_tau(1),idx_tau(2),np);
p_cut1 = linspace(idx_p(1),idx_p(3),np);

% Lower cut
tau_cut2 = linspace(idx_tau(3),idx_tau(4),np);
p_cut2 = linspace(idx_p(2),idx_p(4),np);

% Cut-off filter
mask = ones(nt,np);

for ip = 1:np
    for it = 1:nt
        if (tau_cut1(ip) <= tau(it) && tau_cut2(ip) >= tau(it)) && ...
            (p_cut1(ip) <= p(ip) && p_cut2(ip) >= p(ip))
            mask(it,ip) = 0;
    end
end
end

figure(3)
Cut_off_Filter=figure(3);
subplot(1,2,1)
wigb(radon,1,p,tau)
hTitle=title('Dom\'inio $\tau$-p', 'Interpreter','latex','fontsize',16);
```

```

xlabel('Inclina\c{c}\o (\$s/m\$)', 'Interpreter', 'latex', 'fontsize', 14)
ylabel('$\tau (\$s\$)', 'Interpreter', 'latex', 'fontsize', 14)
set(hTitle, 'FontSize', 16)
set(gca, 'fontsize', 14);
%
subplot(1,2,2)
imagesc(p,tau,mask)
colorbar;
hTitle=title('Filtro de corte', 'Interpreter', 'latex', 'fontsize', 16);
xlabel('Inclina\c{c}\o (\$s/m\$)', 'Interpreter', 'latex', 'fontsize', 14)
ylabel('$\tau (\$s\$)', 'Interpreter', 'latex', 'fontsize', 14)
set(hTitle, 'FontSize', 16)
set(gca, 'fontsize', 14);
set(gcf, 'color', 'w', 'Position', get(0, 'Screensize'));
export_fig('Cut_off_Filter', '-pdf', '-png');

% -----
radon = radon.*mask;
% Inverse LRT
[inv_cdp,t,x] = LRT_backward(radon,tau,p,x);

figure(2)
radon_volta=figure(2);
subplot(1,2,1)
wigb(radon,1,p,tau)
hTitle=title('Dom\''inio $\tau$p', 'Interpreter', 'latex', 'fontsize', 16);
xlabel('Inclina\c{c}\o (\$s/m\$)', 'Interpreter', 'latex', 'fontsize', 14)
ylabel('$\tau (\$s\$)', 'Interpreter', 'latex', 'fontsize', 14)
set(hTitle, 'FontSize', 16)
set(gca, 'fontsize', 14);
%
subplot(1,2,2)
wigb(inv_cdp,1,x,t)
hTitle=title('CDP 1000 reconstru\'ido', ...
    'Interpreter', 'latex', 'fontsize', 16);
xlabel('Afastamento (\$m\$)', 'Interpreter', 'latex', 'fontsize', 14)
ylabel('Tempo (\$s\$)', 'Interpreter', 'latex', 'fontsize', 14)
set(hTitle, 'FontSize', 16)
set(gca, 'fontsize', 14);
set(gcf, 'color', 'w', 'Position', get(0, 'Screensize'));
export_fig('radon_volta', '-pdf', '-png');
% -----

```

G SCRIPT MATLAB FILTRAGEM FK

Este script em matlab foi escrito para realizar a filtragem $f - k$. O script gera 12 figuras para o modelo de camadas curvas. Os parâmetros de entrada são os seguintes:

```
%-----
% Codigo: Filtragem no dominio f-k de uma secao tiro
%-----
% Objetivo: Adicionar ruido e ganho no dado e realizar filtragem iterativa
% no dominio f-k de uma secao tiro.
% Atualizacao: 26 de Junho de 2019.
% Programador: Murilo Santiago Vale Rodrigues (murilovj@gmail.com)
%-----
clc;
close all;
clear;
tic;
%
% slCharacterEncoding('UTF-8'); % acentuacao nos graficos
%-----
% Desenho do filtro no dominio da frequencia
%-----
nPt=1200; % Num. pontos tempo do filtro Em-leque.
nPz=96; % Num. pontos distancia do filtro Em-leque Tiro 50.

A=fopen('tiro_50.bin','r'); % leitura do arquivo binario
pw_int=fread(A,[nPt,nPz],'float'); % criacao da matriz
fclose(A);

Dtt=0.004; % intervalo temporal.
fator= 0.25; % fator de ruido.
%-----
% Desenho do filtro no dominio da frequencia
%-----
agc_gate=2; % gate agc
coef_a = 2; % coeficiente a da funcao ganho.
coef_b = 0.5; % coeficiente b da funcao ganho.
% pw_int_gain=gain1(pw_int,Dtt,'agc',agc_gate,1);
pw_int_gain=gain1(pw_int,Dtt,'time',[coef_a,coef_b],1);
%-----
t=(0:nPt-1)*Dtt; % vetor tempo
dist=1862:12.5:3050; % distancia para o tiro 50
%dist=[1:96];
%-----
% Gerando ruido aleatorio
%-----
```

```

ruido=fator*(2*randn(nPt,nPz)-0.5);
pw_int_ruido=pw_int+ruido;
%pw_int_ruido=pw_int_gain+ruido;
coefa_ruido=1.1;
coefb_ruido=0.3;
secao_tiro=gain1(pw_int,Dtt,'time',[1,1],1);
pw_int_ruido_gain=gain1(pw_int_ruido,Dtt,'time',[coefa_ruido,coefb_ruido],1);
%-----
%                                         Figuras
%-----

figure(1)
% wigb(pw_int,1,dist,t)
wiggle(t,dist,secao_tiro,'vK')
grid on
title('Secao tiro 50','fontsize',15,'fontweight','b')
xlabel('Distancia -x, (m)')
ylabel('Tempo-t, (s)')
set(gca,'FontSize',15); % tamanho da fonte dos axis
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig1','epsc')
saveas(1,'fig1.png')
%
figure(2)
% wigb(pw_int_gain,1,dist,t)
wiggle(t,dist,pw_int_gain,'vK')
grid on
title('Secao tiro 50 com ganho','fontsize',15,'fontweight','b')
xlabel('Distancia -x, (m)')
ylabel('Tempo-t, (s)')
set(gca,'FontSize',15); % tamanho da fonte dos axis
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig2','epsc')
saveas(2,'fig2.png')
%
figure(3)
% wigb(pw_int_ruido,1,dist,t)
wiggle(t,dist,pw_int_ruido,'vK')
title('Secao tiro 50 com ruido','fontsize',15,'fontweight','b')
xlabel('Distancia -x, (m)')
ylabel('Tempo-t, (s)')
set(gca,'FontSize',15); % tamanho da fonte dos axis
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig3','epsc')
saveas(3,'fig3.png')
%
figure(4)
% wigb(pw_int_ruido_gain,1,dist,t)

```

```
wiggle(t,dist,pw_int_ruido_gain,'vK')
title('Secao tiro 50 com ruido e ganho','fontsize',15,'fontweight','b')
xlabel('Distancia -x, (m)')
ylabel('Tempo-t, (s)')
set(gca,'FontSize',15); % tamanho da fonte dos axis
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig4','epsc')
saveas(4,'fig4.png')
%%%
%----- Aplicacao do filtro Fk -----
%
ntpad=nPt;
nx=1;

nxpad=1;
while nxpad < nx
    nxpad=2*nxpad;
end

percent=0;
ishift=1;

% [spec,f_int,kx_int]=fktran(pw_int,t,dist,ntpad,nxpad,percent,ishift);
% [spec,f_int,kx_int]=fktran(pw_int_ruido,t,dist,ntpad,nxpad,percent,ishift);

modulo_spec=sqrt((real(spec)).^2 +(imag(spec)).^2);

[nf,nk]=size(modulo_spec);

zz=1:nf; % vetor profundidade (indice j-z)
xx=1:nk; % vetor distancia (indice i-x)

modulo_spec_mute=modulo_spec;
spec_1=spec;

figure(5)
imagesc(modulo_spec)
colorbar;
grid on;
title('Espectro de amplitude da secao tiro 50 com ...
       ruido','fontsize',15,'fontweight','b')
xlabel('Indice de frequencia espacial')
ylabel('Indice de frequencia temporal')
set(gca,'FontSize',15);
```

```

set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig5','epsc')
saveas(5,'fig5.png')

num_point=2;

hold on
indice_j=zeros(1:num_point);
indice_i=zeros(1:num_point);
for i = 1:num_point
    [indice_j(i),indice_i(i)] = ginput(1);
    plot(indice_j(i),indice_i(i),'m+', 'LineWidth',2)% ko is a ...
        circlur point. you can use '*' point.
end

nPontos=50;
ii = linspace(indice_i(1),indice_i(2),nPontos); % vetor velocidade
jj = linspace(indice_j(1),indice_j(2),nPontos); % vetor velocidade

mute=[jj;ii];
[n, n] = size(mute);

%-----%
%                               Corte na matriz
%-----%
for i=1:nk                  % loop sobre o exio x
    for j=1:nf                % loop sobre o exio z
        for k=1:n                % loop sobre o vetor mute
            if(zz(j)≤mute(2,k)) && (xx(i)≥mute(1,k)) % se a coordenada em ...
                z e x for menor o arquivo mute
                modulo_spec_mute(j,i)=0.0;
                spec_1(j,i)=0.0;

            end
        end
    end
end
%-----%

modulo_spec_mute_aux1=modulo_spec_mute;
spec_21=spec_1;

figure(6)
imagesc(modulo_spec_mute)
hold on;
plot(mute(1,:),mute(2,:),'-*y')
colorbar;

```

```

grid on;
title('Espectro de amplitude da secao tiro 50 com ...
    ruido','fontsize',15,'fontWeight','b')
xlabel('Indice de frequencia espacial')
ylabel('Indice de frequencia temporal')
set(gca,'FontSize',15);
hold off;
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig6','epsc')
saveas(6,'fig6.png')
%
num_point=2;
%
hold on
for i = 1:num_point
    [indice_j(i),indice_i(i)] = ginput(1);
    plot(indice_j(i),indice_i(i),'m+','LineWidth',2)% ko is a ...
        circlur point. you can use '*' point.
end
%
nPontos=50;
ii = linspace(indice_i(1),indice_i(2),nPontos); % vetor velocidade
jj = linspace(indice_j(1),indice_j(2),nPontos); % vetor velocidade
%
mute=[jj;ii];
[n,n] = size(mute);

%-----%
%                               Corte na matriz
%-----%

for i=1:nk          % loop sobre o exio x
    for j=1:nf          % loop sobre o exio z
        for k=1:n          % loop sobre o vetor mute
            if(zz(j)≤mute(2,k)) && (xx(i)≤mute(1,k))      % se a coordenada ...
                em z e x for menor o arquivo mute
            % if(zz(j)≥mute(2,k)) && (xx(i)≥mute(1,k))      % se a ...
                coordenada em z e x for menor o arquivo mute
                modulo_spec_mute_aux1(j,i)=0.0;
                spec_21(j,i)=0.0;
            end
        end
    end
end

modulo_spec_mute_aux=modulo_spec_mute_aux1;
spec_2=spec_21;

```

```

%
figure(7)
imagesc(modulo_spec_mute_aux1)
hold on;
plot(mute(1,:),mute(2,:),'-*y')
colorbar;
grid on;
title('Espectro de amplitude da secao tiro 50 com ...
    ruido','fontsize',15,'fontWeight','b')
xlabel('Indice de frequencia espacial')
ylabel('Indice de frequencia temporal')
set(gca,'FontSize',15);
hold off;
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig7','epsc')
saveas(7,'fig7.png')
%
num_point=2;

hold on
for i = 1:num_point
    [indice_j(i),indice_i(i)] = ginput(1);
    plot(indice_j(i),indice_i(i),'m+','LineWidth',2)% ko is a ...
        circlur point. you can use '*' point.
end

nPontos=50;
ii = linspace(indice_i(1),indice_i(2),nPontos); % vetor velocidade
jj = linspace(indice_j(1),indice_j(2),nPontos); % vetor velocidade

mute=[jj;ii];
[m,n] = size(mute);

% i=1;
% j=1;
% k=1;
%-----
%-----                                     Corte na matriz
%-----

for i=1:nk                         % loop sobre o exio x
    for j=1:nf                       % loop sobre o exio z
        for k=1:n                      % loop sobre o vetor mute
            if(zz(j)>=mute(2,k)) && (xx(i)<=mute(1,k))      % se a coordenada ...
                em z e x for menor o arquivo mute
            % if(zz(j)>=mute(2,k)) && (xx(i)<=mute(1,k))      % se a coordenada ...
                em z e x for menor o arquivo mute

```

```

        modulo_spec_mute_aux(j,i)=0.0;
        spec_2(j,i)=0.0;
    end
end
end
end

figure(8)
imagesc(modulo_spec_mute_aux)
hold on;
plot(mute(1,:),mute(2,:),'-*y')
colorbar;
grid on;
title('Espectro de amplitude da secao tiro 50 com ...
    ruido','fontsize',15,'fontweight','b')
xlabel('Indice de frequencia espacial')
ylabel('Indice de frequencia temporal')
set(gca,'FontSize',15);
hold off;
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig8','epsc')
saveas(8,'fig8.png')
%
filt_spec=spec_2;
nfpad=0;
nkpad=0;
%
[pw_ifk,t_fk,z_fk]=ifktran(filt_spec,f_int,kx_int,nfpad,nkpad,percent);
%
figure(9)
% wigb(pw_ifk,1,dist,t) %pw_ifk(1:1001,:),1,dist,t(1:1001) % corte
wiggle(t,dist,pw_ifk,'vK')
grid on;
title('Secao filtrada sem ganho','fontsize',15,'fontweight','b')
xlabel('Distancia -x, (m)')
ylabel('Tempo-t, (s)')
set(gca,'FontSize',15); % tamanho da fonte dos axis
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig9','epsc')
saveas(9,'fig9.png')
%
coef_a = 2; % coeficiente a da funcao ganho.
coef_b = 0.5; % coeficiente b da funcao ganho.
pw_ifk_gain=gain1(pw_ifk,Dtt,'time',[coef_a,coef_b],1); % ganho na ...
    secao filtrada
%
figure(10)

```

```

% wigb(pw_ifk_gain,1,dist,t) %pw_ifk(1:1001,:),1,dist,t(1:1001)
wiggle(t,dist,pw_ifk_gain,'vK')
grid on;
title('Secao filtrada com ganho','fontsize',15,'fontweight','b')
xlabel('Distancia -x, (m)')
ylabel('Tempo-t, (s)')
set(gca,'FontSize',15); % tamanho da fonte dos axis
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig10','epsc')
saveas(10,'fig10.png')
%
figure(11)
imagesc(ruido);
colorbar;
title('Ruido aleatorio','fontsize',15,'fontweight','b')
xlabel('Indice de Distancia')
ylabel('Indice temporal')
set(gcf, 'Position', get(0, 'Screensize'));
set(gca,'FontSize',15); % tamanho da fonte dos axis
saveas(gcf,'fig11','epsc')
saveas(11,'fig11.png')
%
%-----%
%       Funcao ganho no tempo: g(t)=t^(a) * e^(-b*t)
%-----%
%
t_gain=0:0.01:2;      % vetor tempo
curva_gainho = t_gain.^coef_a.*exp(-coef_b.*t_gain);
figure(12)
plot(t_gain,curva_gainho,'LineWidth',2);
title('Curva de ganho aplicada','fontsize',15,'fontweight','b')
xlabel('Tempo-t, (s)')
ylabel('Ganho g(t)')
set(gca,'FontSize',15); % tamanho da fonte dos axis
grid on;
set(gcf, 'Position', get(0, 'Screensize'));
saveas(gcf,'fig12','epsc')
saveas(12,'fig12.png')

toc;

```

H SHELL SCRIPT MÁXIMA COBERTURA

Este shell script foi escrito para utilizar principalmente a subrotina *sukeycount* do pacote Seismic Unix (SU). O script gera a figura (2.23) do modelo de camadas curvas.

```
#!/bin/bash
#-----
# Sorteamento por cdp. Para usar no Matlab
#-----
data=seis_cdp.su

sukeycount < $data key=cdp > cdp.txt

exit
```

SCRIPT DE PLOTAGEM NO MATLAB

Após a execução do shell script “máxima cobertura” é gerado um arquivo *cdp.txt* que é a entrada da plotagem no matlab. Este arquivo precisa conter apenas os dois vetores colunas que serão carregados no matlab, é necessário excluir o texto gerado no arquivo de saída *cdp.txt*.

```
clc;
close all;
clear;
tic;
%-----
% Carregar o arquivo txt ou dat
%-----
cdp=load('cdp.txt');
%
figure(1)
plot(cdp(:,1),cdp(:,2),'*b')
grid on;
title('CDP com cobertura m\'axima', ...
    'Interpreter','latex','fontsize',20,'fontweight','b')
xlabel('N\'umero de ... ...
    CDP','Interpreter','latex','fontsize',15,'fontweight','b')
ylabel('N\'umero de ... ...
    tra\c{c}os','Interpreter','latex','fontsize',15,'fontweight','b')
xlim([0 11000]);
set(gca,'FontSize',20); % tamanho da fonte dos axis
set(gcf,'color','w', 'Position', get(0, 'Screensize'));
export_fig('CDP_iluminados','-pdf','-png');
```

```
% saveas(gcf,'CDP_iluminados.eps','epsc')  
% saveas(l,'CDP_iluminados.png')
```

```
toc;
```

I SHELL SCRIPT PLOT DOS CDP'S

Este shell script foi escrito para utilizar principalmente a subrotina *supswigp* do pacote Seismic Unix (SU). O script gera as figuras (2.24) e (2.25) do modelo de camadas curvas.

```
#! /bin/sh
#-----
# Mostra as secoes CDP
# entre os CDPs 1150 e 9650 com espacamento de 500 CDPs
#-----

input=seis_cdp.su

for number in $(seq 1150 500 9650)
do
    echo "CMP - $number"
    suwind < $input key=cdp min=$number max=$number > cmp$number.su
#    suxwigb < cmp$number.su title=" CMP - $number " key=offset \
#        label1="Tempo (s)" label2="Afastamento (m) " \
#        perc=94  &
#
#    supswigp < cmp$number.su title=" CMP - $number " key=offset \
#        label1="Tempo (s)" label2="Afastamento (m) " \
#        perc=94 key=offset verbose=0 wbox=11.8 hbox=9.0 > ...
#        cmp$number.eps
#    evince cmp$number.eps &
done

exit
```

J SHELL SCRIPT RUÍDO NA SEÇÃO SÍSMICA

Este shell script foi escrito para utilizar principalmente a subrotina *suaddnoise* do pacote Seismic Unix (SU).

```
#! /bin/sh
#-----
# Adiciona ruido a secao sismica
# scale = (1/sn) * (absmax_signal/sqrt(2))/sqrt(energy_per_sample)
#-----
indata=seis_cdp.su          # dado de entrada
outdata=seis_noise.su        # dado com ruido
sn=50                         # razao sinal/ruido
noisetype=gauss               # tipo de ruido
seed=from_clock                # semente de partida

suaddnoise <$indata >$outdata sn=$sn noise=$noisetype seed=$seed

# cdp 1150 sem ruido
suwind < $indata key=cdp min=1150 max=1150 | suximage perc=99 &
# cdp 1150 com ruido
suwind < $outdata key=cdp min=1150 max=1150 | suximage perc=99 &

exit
```

K SHELL SCRIPT ANÁLISE DE VELOCIDADE ITERATIVA

Este shell script foi escrito para utilizar principalmente a subrotina *suvelan* do pacote Seismic Unix (SU).

```
#!/bin/sh
# File: iva.scr
#       Run this script to start script iva.sh

xterm -geom 80x20+10+545 -e iva.sh

#! /bin/sh
# File: iva.sh
# Run script iva.scr to start this script
# Este script gera a figura do semblance e o cmp antes e apos a ...
# correcao nmo em formato eps.

# Set messages on
# set -x

#=====
# USER AREA -- SUPPLY VALUES
#-----
# CMPs for analysis

cmp1=9650

numCMPs=1

#-----
# File names

indata=seis_cdp_sort_noise.su # SU format
outpicks=CDP_9650.dat         # ASCII file

#-----
# display choices

myperc=97          # perc value for plot
plottype=0         # 0 = wiggle plot, 1 = image plot

#-----
# Processing variables

# Semblance variables
nvs=360  # number of velocities
```

```

dvs=10 # velocity intervals
fvs=1400 # first velocity

# CVS variables
fc=1350 # first CVS velocity
lc=3000 # last CVS velocity
nc=10 # number of CVS velocities (panels)
XX=11 # ODD number of CMPs to stack into central CVS

#=====
# HOW SEMBLANCE (VELAN) VELOCITIES ARE COMPUTED

# Last Vel = fvs + (( nvs-1 ) * dvs ) = lvs
#      5000 = 500 + (( 99-1 ) * 45 )
#      3900 = 1200 + (( 100-1 ) * 27 )

# Compute last semblance (velan) velocity
lvs=`bc -l << -END
$fvs + (( $nvs - 1 ) * $dvs )
END` 

#-----
# HOW CVS VELOCITIES ARE COMPUTED

# dc = CVS velocity increment
# dc = ( last CVS vel - first CVS vel ) / ( # CVS - 1 )
# m = CVS plot trace spacing (m = d2, vel units)
# m = ( last CVS vel - first CVS vel ) / ( ( # CVS - 1 ) * XX )

# j=1
# while [ j le nc ]
# do
#   vel = fc + { [ ( lc - fc ) / ( nc-1 )] * ( j-1 ) }
#   j = j + 1
# done
# EXAMPLE:
#   vel = 1200 + ( (( 3900 - 1200 ) / ( 10-1 )) * ( 1-1 ) )
#   vel = 1200 + ( (( 3900 - 1200 ) / ( 10-1 )) * ( 2-1 ) )
#
#           .
#           .
#           .
#   vel = 1200 + ( (( 3900 - 1200 ) / ( 10-1 )) * (11-1 ) )

#=====

```

```

# FILE DESCRIPTIONS

# tmp0 = binary temp file for input CVS gathers
# tmp1 = binary temp file for output CVS traces
# tmp2 = ASCII temp file for managing picks
# tmp3 = binary temp file for stacked traces
# tmp4 = ASCII temp file for "wc" result (velan)
# tmp5 = ASCII temp file for stripping file name from tmp4 (velan)
# tmp6 = ASCII temp file to avoid screen display of "zap"
# tmp7 = ASCII temp file for picks
# tmp8 = binary temp file for NMO (flattened) section
# panel.$picknow = current CMP windowed from line of CMPs
# picks.$picknow = current CMP picks arranged as "t1 v1"
#                                     "t2 v2"
#                                     etc.
# par.# (# is a sequential index number; 1, 2, etc.)
#       = current CMP picks arranged as
#         "tnmo=t1,t2,t3,..." 
#         "vnmo=v1,v2,v3,..." 
# par.uni.# (# is a sequential index number; 1, 2, etc.)
#       = current CMP picks arranged as
#         "xin=t1,t2,t3,..." 
#         "yin=v1,v2,v3,..." 
#       for input to xgraph to display velocity profile
# par.cmp = file of CMP number and sequential index number;
#           for example: "40 1"
#                         "60 2"
#                         etc.
# par.0 = file "par.cmp" re-arranged as
#         "cdp=#,#,#,etc." NOTE: # in this line is picked CMP
#         "#=1,2,3,etc."   NOTE: # in this line is "#"
# outpicks = concatenation of par.0 and all par.# files.

#=====
echo "
echo " *** INTERACTIVE VELOCITY ANALYSIS ***"
echo "

#-----
# Remove old files. Open new files
rm -f panel.* picks.* par.* tmp*

> $outpicks # Write empty file for final picks
> par.cmp    # Write empty file for recording CMP values

#-----

```

```

# Get ns, dt, first time from seismic file
nt=`sugethw ns < $indata | sed 1q | sed 's/.ns=//'` 
dt=`sugethw dt < $indata | sed 1q | sed 's/.dt=//'` 
ft=`sugethw delrt < $indata | sed 1q | sed 's/.delrt=//'` 

# Convert dt from header value in microseconds
# to seconds for velocity profile plot
dt=`bc -l << -END
    scale=6
    $dt / 1000000
END` 

# If "delrt", use it; else use zero
if [ $ft -ne 0 ] ; then
    tstart=`bc -l << -END
        scale=6
        $ft / 1000
    END` 
else
    tstart=0.0
fi

#-----
# Initialize "repick" -- for plotting previous picks on velan
repick=1 # 1=false, 0=true

#-----
# BEGIN IVA LOOP
#-----


i=1
while [ $i -le $numCMPs ]
do

# set variable $picknow to current CMP
eval picknow=\$cmp\$i

if [ $repick -eq 1 ] ; then
    echo " "
    echo "Preparing CMP $i of $numCMPs for Picking "
    echo "Location is CMP $picknow "
fi

#-----
# Plot CMP (right)
#-----
```

```

suwind < $indata \
        key=cdp min=$picknow max=$picknow > panel.$picknow
if [ $repick -eq 1 ] ; then
  if [ $plottype -eq 0 ] ; then
    suxwigb < panel.$picknow xbox=634 ybox=10 wbox=300 hbox=450 \
              title="CMP gather $picknow" \
              label1=" Time (s)" label2="Offset (m)" key=offset \
              perc=$myperc verbose=0 &
  else
    suximage < panel.$picknow xbox=634 ybox=10 wbox=300 hbox=450 \
              title="CMP gather $picknow" \
              label1=" Time (s)" \
              perc=$myperc verbose=0 &
  fi
else
  if [ $plottype -eq 0 ] ; then
    suxwigb < panel.$picknow xbox=946 ybox=10 wbox=300 hbox=450 \
              title="CMP gather $picknow" \
              label1=" Time (s)" label2="Offset (m)" key=offset \
              perc=$myperc verbose=0 &
  else
    suximage < panel.$picknow xbox=946 ybox=10 wbox=300 hbox=450 \
              title="CMP gather $picknow" \
              label1=" Time (s)" \
              perc=$myperc verbose=0 &
  fi
fi

#      suwind <$indata key=cdp min=$picknow \
#                      max=$picknow >panel.$picknow
#      suxwigb <panel.$picknow xbox=312 ybox=10 \
#                      wbox=300 hbox=600 \
#                      title="Secao PMC $picknow" \
#                      perc=94 key=offset verbose=0 &
#      supswigb < panel.$picknow label2='Afastamento (m)' label1='Tempo ...
#                           (s)' \
#                      perc=94 key=offset verbose=0 wbox=7.0 hbox=9.0 > ...
#                      CMP_$picknow.eps

#-----
# Constant Velocity Stacks (CVS) (middle-left)
# Make CVS plot for first pick effort.
# If re-picking t-v values, do not make this plot.
#-----


# repick: 1=false, 0=true

```

```

if [ $repick -eq 1 ] ; then

# number of CMPs - 1; for windowing
X=`expr $XX - 1` 

# Window CMPs around central CMP (+/- X/2). Write to tmp0
k1=`expr $picknow - $X / 2` # Window from CMP to CMP - X/2
k2=`expr $picknow + $X / 2` # Window from CMP to CMP + X/2
suwind < $indata key=cdp min=$k1 max=$k2 > tmp0

# Calculate CVS velocity increment
# dc = ( last CVS vel - first CVS vel ) / ( # CVS - 1 )
dc=`bc -l << -END
( $lc - $fc ) / ( $nc - 1 )
END` 

# Calculate trace spacing for CVS plot (m = d2, vel units)
# m = ( last CVS vel - first CVS vel ) / ( ( # CVS - 1 ) * XX )
m=`bc -l << -END
( $lc - $fc ) / ( ( $nc - 1 ) * $XX )
END` 

# CVS velocity loop
j=1
while [ $j -le $nc ]
do

    vel=`bc -l << -END
    $fc + $dc * ( $j - 1 )
    END` 

# uncomment to print CVS velocities to screen
##     echo " vel = $vel"

    sunmo < tmp0 vnmo=$vel verbose=0 |
    sustack >> tmp1

    j=`expr $j + 1`
done

# Compute lowest velocity for annotating CVS plot
# loV = first CVS velocity - ( ( CMP range - 1 ) / 2 ) * vel inc
loV=`bc -l << -END
$fc - ( $X / 2 ) * $m
END` 

suximage < tmp1 xbox=322 ybox=10 wbox=300 hbox=450 \

```



```

        grid1=solid grid2=solid mpicks=picks.$picknow \
        curve=tmp7 npair=$npair curvecolor=white
#      supsimage bclip=0.5 wclip=0.0 f2=$fvs d1=0.004 d2=$dvs \
#          label1="Tempo (s)" label2="Velocidade (m/s) " ...
mpicks=picks.$picknow \
#          grid1=solid grid2=solid gridcolor=black hbox=1200 ...
wbox=500 \
#          legend=1 lstyle=horibottom units=Semblance verbose=0 \
#          cmap= hsv2 nc=10 curvecolor=white > Semblance_$picknow.eps

else

suvelan < panel.$picknow nv=$nvs dv=$dvs fv=$fvs |
suximage xbox=10 ybox=10 wbox=300 hbox=600 perc=99 \
units="semblance" f2=$fvs d1=0.02 d2=$dvs n2tic=5 \
title="Semblance Plot CMP $picknow" cmap=hsv2 \
label1=" Time (s)" label2="Velocity (m/s)" \
legend=1 units=Semblance verbose=0 gridcolor=black \
grid1=solid grid2=solid mpicks=picks.$picknow
fi

suvelan < panel.$picknow nv=$nvs dv=$dvs fv=$fvs > teste

supsimage < teste units="Semblance" perc=98 \
f2=$fvs d1=0.02 d2=$dvs label1="Tempo (s)" label2="Velocidade ... \
(m/s) " \
legend=1 lstyle=vertright units=Semblance gridcolor=black \
grid1=solid grid2=solid mpicks=picks.$picknow > ...
Semblance_$picknow.eps

mycolorsu Semblance_$picknow.eps Semblance_color_$picknow.eps matlab

#-----
# End first set of plots
#=====

#-----
# Manage picks (1): Prepare picks for sunmo
#-----

sort < picks.$picknow -n |
mkparfile string1=tnmo string2=vnmo > par.$i
echo "cdp=$picknow" >> tmp2
cat par.$i >> tmp2

#=====
# Begin second set of plots

```

```

#-----#
#-----#
# Flattened seismic data (NMO) plot (middle-right)
#-----#


sunmo < panel.$picknow par=tmp2 verbose=0 > tmp8

supswigb < tmp8 label2='Afastamento (m)' label1='Tempo (s)' perc=94 ...
key=offset verbose=0 wbox=7.0 hbox=9.0 > CMP_After_NMO_$picknow.eps
if [ $plottype -eq 0 ] ; then
suxwigb < tmp8 xbox=634 ybox=10 wbox=300 hbox=450 \
title="CMP $picknow after NMO" \
label1=" Time (s)" label2="Offset (m)" \
verbose=0 perc=$myperc key=offset &
else
suximage < tmp8 xbox=634 ybox=10 wbox=300 hbox=450 \
title="CMP $picknow after NMO" \
label1=" Time (s)" \
verbose=0 perc=$myperc &
fi

#-----#
# Stack window (right)
#-----#


j=1
while [ $j -le 8 ]
do

# Append stack trace into tmp3 multiple times
sustack < tmp8 >> tmp3

j=`expr $j + 1`
done

suxwigb < tmp3 xbox=946 ybox=10 wbox=200 hbox=450 \
title="CMP $picknow repeat stack trace" \
label1=" Time (s)" d2num=50 key=cdp \
verbose=0 perc=$myperc &

#-----#
# Manage picks (2): Prepare picks for vel profile
#-----#


sed < par.$i '
s/tnmo/xin/

```

```

s/vnmo/yin/
    ' > par.uni.$i

#-----
# Velocity profile (left)
#-----

unisam nout=$nt fxout=$tstart dxout=$dt \
    par=par.uni.$i method=mono |
xgraph n=$nt nplot=1 d1=$dt f1=$tstart x2beg=$fvs x2end=$lvs \
    label1=" Time (s)" label2="Velocity (m/s)" \
    title="CMP $picknow Stacking Velocity Function" \
    -geometry 300x450+10+10 -bg white style=seismic \
    grid1=solid grid2=solid linecolor=2 marksize=1 mark=0 \
    titleColor=black axesColor=blue &

#-----
# Dialogue with user: re-pick ?
#-----


echo " "
echo " t-v PICKS CMP $picknow"
echo "-----"
cat picks.$picknow
echo " "
echo " Use the velocity profile (left),"
echo "     the NMO-corrected gather (middle-right),"
echo "     and the repeated stack trace (right)"
echo "     to decide whether to re-pick the CMP."
echo " "
echo "Picks OK? (y/n) " > /dev/tty
read response

rm tmp*

# "n" means re-loop. Otherwise, continue to next CMP.
case $response in
n*)
    i=$i
    echo " "
    echo "Repick CMP $picknow. Overlay previous picks."
    repick=0
    cp picks.$picknow tmp7
    ;;
*)
    echo "$picknow $i" >> par.cmp
    i=`expr $i + 1`
```

```
repick=1
echo "-- CLOSING CMP $picknow WINDOWS --"
zap xwigb > tmp6
zap ximage > tmp6
zap xgraph > tmp6
;;
esac

done

#-----
# Create velocity output file
#-----

mkparfile < par.cmp string1=cdp string2=# > par.0

i=0
while [ $i -le $numCMPs ]
do
  sed < par.$i 's/$/ \\\'/g' >> $outpicks
  i=`expr $i + 1`
done

#-----
# Remove files and exit
#-----
echo " "
echo " The output file of t-v pairs is \"$outpicks"
pause
rm -f panel.* picks.* par.* tmp* teste*
exit
```

L SHELL SCRIPT MODELO DE VELOCIDADE

Este shell script foi escrito para utilizar principalmente as subrotinas *unisam2* e *smooth2* do pacote Seismic Unix (SU).

```
#!/bin/sh
# UTILIZA O CAMPO DE VELOCIDADE COM TNMO E VNMO

# INFORMACOES DE ENTRADA
VELDIR=.
INVEL=velan.dat # nome do arquivo txt com os pares de tempo e velocidade
OUTVEL1=interp1d-lms.bin
OUTVEL2=campovel.bin
output=campovel_suavizado.bin
output2=campovel_suavizado.eps

# INFORMACOES DE ENTRADA
nsamp=1200
dt=0.004
fsamp=0
ncdpin=18      # numero de cdp analisados
fcdpin=1150    # numero do 1o. cdp da analise de vel.
dcdpin=500     # Δ entre cdp utilizado na analise vel.
ncdpout=10788  # numero total de cdps do dado
fcdpout=1      # numero do 1o. cdp de saida
dcdpout=1
p1=250         # parametro de suavizacao em n1 (eixo- vertical).
p2=450         # parametro de suavizacao em n2 (eixo- horizontal).

nlines=`grep -v cdp $INVEL | grep -v "#" | wc -l | awk '{print $1}'` 
currline=1

>$OUTVEL1

# INTERPOLACAO ESPACIAL
while [ $currline -le $nlines ]
do
nextline=`expr $currline + 1`
ncolx=`grep -v cdp $INVEL | grep -v "#" | head -$currline | tail -1 ...
| wc -c`
lcolx=`expr $ncolx - 2`
ncoly=`grep -v cdp $INVEL | grep -v "#" | head -$nextline | tail -1 ...
| wc -c`
lcoly=`expr $ncoly - 2`
Xin=`grep -v cdp $INVEL | grep -v "#" | head -$currline | tail -1 | ...
cut -c6-$lcolx`
```

```

Yin=`grep -v cdp $INVEL | grep -v "#" | head -$nextline | tail -1 | ...
cut -c6-$lcoly` 

unisam xin=$Xin yin=$Yin nout=$nsamp dxout=$dt fxout=$fsamp ...
method=linear >>$OUTVEL1
currline=`expr $currline + 2`
done
# INTERPOLACAO TEMPORAL
unisam2 nx1=$nsamp dx1=$dt fx1=$fsamp n1=$nsamp d1=$dt f1=$fsamp ...
nx2=$ncdpin dx2=$dcdpin fx2=$fcdpin n2=$ncdpout d2=$dcdpout ...
f2=$fcdpout method1=linear method2=linear <$OUTVEL1 >$OUTVEL2

rm $OUTVEL1

ximage n1=$nsamp < $OUTVEL2 d1=$dt f2=$fcdpout d2=$dcdpout cmap=hsv2 ...
legend=1 title='sem suavizacao' &

# suavizando o modelo de velocidade

smooth2 n1=$nsamp n2=$ncdpout r1=$p1 r2=$p2 < $OUTVEL2 > $output

ximage n1=$nsamp < $output d1=$dt cmap=hsv2 legend=1 title='suavizado'

exit

```

SCRIPT PLOT DO MODELO DE VELOCIDADE

```

#!/bin/sh
#-----
output=campovel.bin
output2=campovel2.eps
#output=campovel_suavizado.bin
#output2=campovel_suav.eps
#-----
# INFORMACOES DE ENTRADA
nsamp=1200 # numero de amostras temporais nt
dt=0.004   # intervalo de amostragem
#-----
psimage <$output n1=$nsamp units="Velocidade" perc=98 \
d1=$dt f2=0.0 d2=0.001112347 label1="Tempo (s)" ...
label2="Distancia (km) " \
title='Campo de Velocidade' titlesize=20 legend=1 ...
lstyle=vertright width=15.0 height=8.0 > $output2
mycolorsu $output2 campovel_2.eps matlab
#12000/10788 = 1.112347052 distancia em x/numero de cdps

```

```
#1.112347052/1000 = 0.001112347 distancia em km  
exit
```

M SHELL SCRIPT CORREÇÃO NMO

Este shell script foi escrito para utilizar principalmente a subrotina *sunmo* do pacote Seismic Unix (SU).

```
#!/bin/sh
#-----
indata=seis_cdp.su      # dado de entrada
outdata=seis_cdp_nmo.su # dado corrigido NMO
#-----
sunmo < $indata \
cdp=1150,1650,2150,2650,3150,3650,4150,4650,5150,5650,6150,6650,7150,7650,8150,8650,9150 \
tnmo=0.39758,0.795161,1.35539,1.85236,2.5662,3.26197 \
vnmo=1619.39,1719.11,1818.83,1858.72,1999,2390.22 \
tnmo=0.379508,0.768053,1.33732,1.85236,2.5662,3.24389 \
vnmo=1599.44,1778.94,1818.83,1858.72,1998.94,2380.17 \
tnmo=0.379508,0.759017,1.34635,1.83429,2.57524,3.26197 \
vnmo=1599.44,1739.06,1798.89,1898.61,1919,2389 \
tnmo=0.361437,0.759017,1.37346,1.87043,2.60234,3.28907 \
vnmo=1639.33,1739.06,1858.72,1858.72,1918.89,2395 \
tnmo=0.388544,0.722873,1.42767,1.87947,2.62945,3.30715 \
vnmo=1599.44,1719.11,1818.83,1898.61,1918.94,2385.06 \
tnmo=0.352401,0.731909,1.44575,1.96079,2.70174,3.36136 \
vnmo=1599.44,1719.11,1838.78,1898.61,1919,2381 \
tnmo=0.343365,0.722873,1.53611,2.06922,2.78306,3.40654 \
vnmo=1639.33,1739.06,1798.89,1898.61,1989.89,2398 \
tnmo=0.361437,0.704801,1.59032,2.15055,3.00896,3.46076 \
vnmo=1579.5,1719.11,1818.83,1918.56,1998.5,2390.94 \
tnmo=0.352401,0.68673,1.56321,2.09633,2.73788,3.32522 \
vnmo=1639.33,1719.11,1878.67,2118,1998.28,2360.61 \
tnmo=0.334329,0.677694,1.3825,1.84333,2.48488,3.13546 \
vnmo=1619.39,1719.11,1958.44,2137.94,1998.33,2389.89 \
tnmo=0.352401,0.659622,1.16563,1.59936,2.27705,2.95474 \
vnmo=1639.33,1719.11,1938.5,2018.28,1898.67,2390.06 \
tnmo=0.334329,0.632514,0.966843,1.39153,2.0873,2.82824 \
vnmo=1619.39,1719.11,1878.67,1938.5,1918.89,2379.06 \
tnmo=0.343365,0.632514,0.831304,1.21985,1.95176,2.72885 \
vnmo=1619.39,1719.11,1798.89,1898.61,1739.06,2385 \
tnmo=0.325293,0.623478,0.813232,1.19274,1.92465,2.70174 \
vnmo=1619.39,1699.17,1759,1858.72,1919.11,2393.06 \
tnmo=0.343365,0.623478,0.849376,1.21081,1.92465,2.68367 \
vnmo=1579.5,1719.11,1798.89,1878.67,1919.17,2395.22 \
tnmo=0.334329,0.614442,0.930699,1.31021,1.9879,2.73788 \
vnmo=1599.44,1699.17,1818.83,1898.61,1939.06,2389.22 \
tnmo=0.334329,0.59637,1.09335,1.45478,2.12344,2.80113 \
```

```
vnmo=1619.39,1719.11,1858.72,1918.56,1938.94,2399.17 \
tnmo=0.343365,0.587335,1.22888,1.59032,2.24091,2.89149 \
vnmo=1619.39,1699.17,1898.61,1878.67,1959.06,2395.89 \ > $outdata
```

N SHELL SCRIPT EMPILHAMENTO

Este shell script foi escrito para utilizar principalmente a subrotina *sustack* do pacote Seismic Unix (SU).

```
#! /bin/sh
#-----
# Empilhamento
#-----
indata=seis_cdp_nmo.su # dado corrigido NMO
outdata=seis_stack.su   # dado de saida empilhado

sustack < $indata > $outdata key=cdp verbose=1

exit
```

SCRIPT PLOT DA SEÇÃO EMPILHADA

```
#! /bin/sh

output=seis_stack.su
output2=seis_stack.eps
dt=0.004

supsimage <$output units="Velocidade" perc=99 \
           d1=$dt f2=0 d2=0.01388 label1="Tempo (s)" label2="Distancia ...
           (km) " \
           wrgb=0,0,1.0 grgb=1.0,1.0,1.0 brgb=1.0,0,0 \
           title='Secao Empilhada' titlesize=20 legend=1 ...
           units="Amplitude" \
           lstyle=vertright width=15.0 height=8.0 > $output2 #grid1=solid ...
           grid2=solid

#=12000/864 comprimento do modelo / numero de tracos
#=13,88888889 em metros
#=13,88888889/1000 divido por mil para ficar em km
#=0.01388889

exit
```

O SHELL SCRIPT MIGRAÇÃO KIRCHHOFF

Este shell script foi escrito para utilizar principalmente a subrotina *sumigtk* do pacote Seismic Unix (SU).

```
#! /bin/sh
#-----
# Migracao Kirchhoff em tempo
#-----

input=seis_stack.su
output=seis_stack_mig.su
dxcdp=25
vfile=campovel_suavizado.bin

sumigtk < $input > $output dxcdp=$dxcdp vfile=$vfile verbose=1

exit
```

P SHELL SCRIPT MIGRAÇÃO DIFERENÇAS FINITAS

Este shell script foi escrito para utilizar principalmente a subrotina *sumigfd* do pacote Seismic Unix (SU).

```
#! /bin/sh
# Migracao pos-stack em profundidade metodo de diferencias finitas

wbox=500
hbox=300

#nt=501
dt=0.004
ft=0.0
nxm=864
dxm=13.889
xm0=0.0
nz=1200
dz=2.9

input=seis_stack.su
output=migfd2.su

vfile=campovell.bin
vfilet=modelovelt.bin

transp n1=$nz n2=$nxm <$vfile >$vfilet
#ximage n1=600 n2=350 d1=20 d2=10 < modelovelt.bin perc=99 legend=1 &

sumigfd < $input nz=$nz dz=$dz dx=$dxm vfile=$vfilet >$output

suximage < $input perc=99 f1=$ft d1=$dt f2=$xm0 d2=$dxm \
wbox=$wbox hbox=$hbox \
label1="Tempo [s]" label2="CDP" title="Secao ZO" &

suximage < $output perc=99 f1=$ft d1=$dz f2=$xm0 d2=$dxm \
wbox=$wbox hbox=$hbox \
label1="Profundidade [m]" label2="CDP" title="Secao Migrada ..." \
FD -- NMO + MUTE + STACK" &

exit 0
```

Q SHELL SCRIPT OUTRAS MIGRAÇÕES

Este shell script foi escrito para utilizar as subrotinas de migrações do pacote Seismic Unix (SU).

```
#!/bin/sh

dt=0.004
ft=0.0
nxm=864
dxm=13.889
xm0=0.0
nz=350
dz=10

input=seis_stack.su
output_migpspi=seis_Mig_PSPI_depth.su
output_migsplit=seis_Mig_SPLIT_STEP_depth.su
output_migffd=seis_Mig_FFD_depth.su

vfile=modelo_velocidade.bin
vfilet=modelo_velocidade_tranps.bin

#-----
#Transpondo o modelo de velocidade em profundidade

transp n1=$nz n2=$nxm <$vfile >$vfilet

#-----
#Migracao em profundidade PSPI (Phase-Shift Plus Interpolation)

sumigpspi <$input nz=$nz dz=$dz dt=$dt dx=$dxm vfile=$vfilet > ...
$output_migpspi

#-----
#Migracao em profundidade SPLIT-STEP

sumigspli < $input nz=$nz dz=$dz dx=$dxm vfile=$vfilet > $output_migspli

#-----
#Migracao em profundidade FFD (Fourier Finite Difference)

sumigffd < $input nz=$nz dz=$dz dx=$dxm vfile=$vfilet > $output_migffd

#-----
#Visualizar a secao ZO empilhada
```

```
suximage < $input perc=99 f1=$ft d1=$dt d2=$d xm cmap=rgb1 \
    label1="Time [s]" label2="Distance [m]" title="ZO Section" &

#-----
#Visualizar secao migrada PSPI (Phase-Shift Plus Interpolation)

suximage < $output_migpspi perc=99 label1="Depth [m]" n1=$nz d1=$dz ...
    n2=$nxm d2=$d xm \
    cmap=rgb1 label2="Distance [m]" title="PSPI Migrated Section" &

#-----
#Secao migrada SPLIT-STEP

suximage < $output_migsplit perc=99 label1="Depth [m]" d2=$d xm \
    cmap=rgb1 label2="Distance [m]" title="SPLIT-STEP Migrated ...
    Section" &

#-----
#Secao migrada FFD (Fourier Finite Difference)

suximage < $output_migffd perc=95 label1="Depth [m]" d2=$d xm \
    cmap=rgb1 label2="Distance [m]" title="FFD Migrated Section" &

#-----

exit
```