

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE GEOCIÊNCIAS
DEPARTAMENTO DE GEOLOGIA E GEOFÍSICA



DANIEL DA SILVA FERREIRA

**PROCESSAMENTO DE DADOS SÍSMICOS SINTÉTICOS E REAIS COM
SEISMIC UNIX**

MONOGRAFIA

CURSO DE GRADUAÇÃO EM GEOFÍSICA

**Niterói
12/2022**

DANIEL DA SILVA FERREIRA

PROCESSAMENTO DE DADOS SÍSMICOS SINTÉTICOS E REAIS COM SEISMIC UNIX

Monografia apresentada à Universidade Federal Fluminense como requisito parcial do Curso de Graduação em Geofísica para a obtenção do título de Bacharel em Geofísica.

Orientador

Prof. Dr. Luiz Alberto Santos

**Niterói
12/2022**

Ficha catalográfica automática - SDC/BIG
Gerada com informações fornecidas pelo autor

F383p Ferreira, Daniel da Silva
 Processamento de dados sísmicos sintéticos e reais com
 Seismic Unix / Daniel da Silva Ferreira. - 2022.
 77 f.

 Orientador: Luiz Alberto Santos.
 Trabalho de Conclusão de Curso (graduação)-Universidade
 Federal Fluminense, Instituto de Geociências, Niterói, 2022.

 1. Processamento sísmico. 2. Seismic Unix. 3. Migração.
 4. Modelagem. 5. Produção intelectual. I. Santos, Luiz
 Alberto, orientador. II. Universidade Federal Fluminense.
 Instituto de Geociências. III. Título.


CDD - XXX

DANIEL DA SILVA FERREIRA

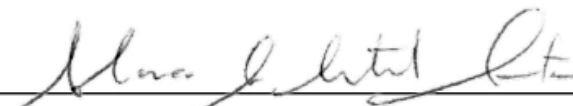
**PROCESSAMENTO DE DADOS SÍSMICOS SINTÉTICOS E REAIS COM SEISMIC
UNIX**

Monografia apresentada à Universidade Federal
Fluminense como requisito parcial do Curso de
Graduação em Geofísica para a obtenção do
título de Bacharel em Geofísica.

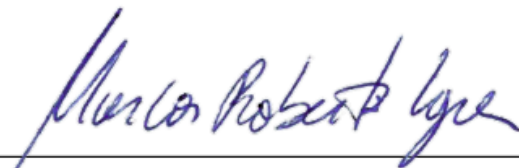
Aprovada em 20/12/2022 pela banca examinadora abaixo:



Prof. Dr. Luiz Alberto Santos
UFF / DOT / GISIS / PETROBRAS



Prof. Dr. Marco Antonio Cetale Santos
UFF / DOT / GISIS



Dr. Marcos Roberto Fetter Lopes
UFF / GISIS

*Dedico este trabalho aos meus pais, minhas irmãs,
Samara e Larissa, e a todos os meus amigos!*

Agradecimentos

Por meio dessas palavras, venho agradecer de coração a todos que me auxiliaram de alguma forma até aqui. Não posso dizer que foi fácil, mas com a grande ajuda de Deus, meu amigo fiel, eu pude chegar até este momento.

Agradeço à minha família de sangue, meus pais, minhas irmãs que me incentivaram em todos os momentos em que estive cabisbaixo, pensando em desistir. Obrigado por cada abraço, sorriso e conselho que me guiaram e me deram forças até hoje.

Agradeço também aos meus amigos que sempre ouviram meus desabafos e me ajudaram da forma que podiam me animando, e me fortalecendo com suas leais amizades.

Quero agradecer de forma especial ao meu orientador, Dr. Luiz Alberto Santos, que me acompanha desde o estágio. Obrigado por todo o ensino, investimento e paciência. Obrigado por me estender a mão, e me guiar na estrada no momento mais difícil da graduação para mim.

Meu agradecimento especial ao Dr. Marco Cetale pela confiança e por permitir meu ingresso ao grupo do GISIS, onde pude aprender tanto em tão pouco tempo. Todos sempre solícitos e dispostos a sanarem as minhas dúvidas praticamente 24h por dia. Agradeço de coração aos meus companheiros do projeto Pelotas: Guilherme Lenz, Marcos Fetter, Ursula Belem, Rodrigo Stern, Roberto Miyamoto, Rodrigo Piccoli, Pedro Albuquerque, Nathalia Campos. Como aprendi com todos vocês nesses últimos meses.

Agradeço também aos demais integrantes do GISIS, Hugo Eduardo Araújo, Paulo Alves, Gustavo e Fernanda que também me ajudaram e me incentivaram em inúmeros momentos.

Agradeço da mesma forma à Universidade Federal Fluminense por toda a infraestrutura, e a todos os funcionários e professores nesses anos de graduação.

”Não fui eu que lhe ordenei? Seja forte e corajoso! Não se apavore, nem se desanime, pois o Senhor, o seu Deus, estará com você por onde você andar”.
Josué 1:9

Resumo

FERREIRA, Daniel da Silva. **PROCESSAMENTO DE DADOS SÍSMICOS SINTÉTICOS E REAIS COM SEISMIC UNIX**. MONOGRAFIA, Universidade Federal Fluminense. Niterói, p. 77. 2022.

O processamento dos dados adquiridos em um levantamento sísmico é de suma importância, visto que uma melhor interpretação da geologia em subsuperfície pode ser realizada a partir das imagens obtidas. Este trabalho apresenta um fluxo básico de processamento de dados sísmicos realizado com o Seismic Unix. A pesquisa é baseada na modelagem, aquisição de uma linha 2D de dados sísmicos sintéticos e seu consequente processamento. Do mesmo modo, uma linha 2D de dados coletados perto da costa do Japão, sobre a fossa de Nankai, é processada. Verificou-se que os dois tipos de dados analisados neste trabalho apresentaram seus próprios desafios. Houve problemas de imageamento de refletores profundos do dado modelado e a migração utilizada, por sua vez, possui muitas limitações. Como a função t-v foi projetada para se ajustar em um certo intervalo de CMPs, a seção foi mal-migrada em outras áreas. Com isso, não foi obtido o melhor imageamento para toda a seção. No dado de Nankai, o uso dos algoritmos de Stolt e Gazdag gerou imagens semelhantes: a linha processada contém muitas difrações mal colapsadas, principalmente, na superfície do fundo do mar. A partir desses resultados, pode-se concluir que o SU, aliado com Shell Scripts, é uma ferramenta útil e possibilita um contato inicial com tarefas básicas do processamento sísmico. Todavia, a falta de conhecimento das ferramentas disponíveis do software, e experiência em processamento sísmico limitam os resultados que podem ser alcançados. Por sua vez, uma nova configuração de aquisição de dados sísmicos do modelo gerado poderia resolver o problema de imageamento de refletores profundos. E, tanto o produto do processamento do dado sintético quanto do dado real podem ser melhorados com uma migração variável no tempo e no espaço, lidando assim com a variação lateral de velocidade. Uma migração em profundidade, e aplicação de filtros e ganhos podem melhorar a imagem final também. Vale salientar que, apesar das dificuldades relatadas anteriormente, foram obtidos resultados semelhantes aos contidos na literatura e no artigo embaixadores deste trabalho.

Palavras-chave: processamento sísmico; Seismic Unix; migração.

Abstract

The processing of data acquired in a seismic survey is very important, since a better interpretation of subsurface geology can be performed from the images obtained. This work presents a basic flow of seismic data processing performed with Seismic Unix. The research is based on modeling, acquisition of a 2D line of synthetic seismic data and its subsequent processing. Likewise, a 2D line of data collected near from the coast of Japan, over the Nankai Trough, is processed. Checked that the two types of data analyzed in this paper presented their own challenges. There were problems in imaging deep reflectors of the modeled data and the migration used, in turn, has many limitations. As the t-v function was designed to fit a certain range of CMPs, the section was poorly migrated in other areas. As a result, the best image was not obtained for the entire section. In the Nankai data, the use of the Stolt and Gazdag algorithms generated similar images: the processed line contains many poorly collapsed diffractions, mainly on the seabed surface. From these results, it can be concluded that the SU, combined with Shell Scripts, is a useful tool and allows an initial contact with basic tasks of seismic processing. However, the lack of knowledge of the available tools of software, and experience in seismic processing limit the results that can be achieved. In turn, a new seismic data acquisition configuration of the generated model could solve the deep reflector imaging problem. And, both the processing product of the synthetic data and the real data can be improved with a migration variable in time and space, thus dealing with the lateral variation of velocity. A deep migration, and applying filters and gains can improve the final image as well. It is worth mentioning that, despite the difficulties previously reported, results similar to those contained in the literature and in the article that underlie this work were obtained.

Keywords: seismic processing; seismic unix; migration.

Lista de ilustrações

Figura 1 – Exemplo de Semblance utilizado na etapa de análise de velocidade.	20
Figura 2 – Correção NMO.	21
Figura 3 – Esquema de aplicação da correção NMO.	21
Figura 4 – CMPs antes e após a correção NMO.	22
Figura 5 – Hipérbole de reflexão sobrecorrigida após a correção NMO.	22
Figura 6 – Hipérbole de reflexão subcorrigida após a correção NMO.	23
Figura 7 – Representação esquemática de empilhamento 2D.	24
Figura 8 – Esquema de migração de um refletor sísmico.	25
Figura 9 – Seção <i>zero-offset</i> contendo efeito bow-tie antes e após a migração.	25
Figura 10 – Fluxograma utilizado para a migração Stolt de velocidade constante no domínio f-k.	27
Figura 11 – Fluxograma do método de migração Gazdag.	28
Figura 12 – Etapas do trabalho.	30
Figura 13 – Fluxo de processamento do dado sintético.	32
Figura 14 – CMP <i>fold</i> do modelo criado.	32
Figura 15 – Mapa mostrando o local de aquisição da linha sísmica 2D.	36
Figura 16 – CMP <i>fold</i> dos dados de Nankai.	37
Figura 17 – Fluxo de processamento dos dados de Nankai.	38
Figura 18 – Modelo sintético gerado.	40
Figura 19 – Velocidade em m/s de cada camada do modelo gerado.	40
Figura 20 – Exemplos de shot gathers do modelo criado.	41
Figura 21 – Ferramentas de análise de velocidade.	41
Figura 22 – Perfil de velocidade do CMP 60 analisado.	42
Figura 23 – Exemplo de aplicação da correção NMO ao CMP dos dados de Nankai.	43
Figura 24 – Dado sintético empilhado.	43
Figura 25 – Dado sintético migrado.	44
Figura 26 – Espectro de amplitudes do dado original.	45
Figura 27 – Espectro de amplitudes do dado após a aplicação dos filtros.	46
Figura 28 – Amplitude medida do dado original.	46
Figura 29 – Amplitude medida do dado após a aplicação do ganho <i>tpow</i> de 1.8.	47
Figura 30 – Amplitude medida do dado após a aplicação do ganho <i>tpow</i> de 2.2.	47
Figura 31 – Exemplo de correção NMO aplicado aos dados de Nankai.	48
Figura 32 – Dado de Nankai após o empilhamento.	49
Figura 33 – Dado de Nankai empilhado e migrado com <i>Stolt</i>	50
Figura 34 – Dado de Nankai empilhado e migrado com <i>Phase-Shift Gazdag</i> . .	50

Figura 35 – Comparação entre os resultados obtidos por Moore et al. (1990) e este trabalho.	52
---	----

Lista de tabelas

Tabela 1 – Parâmetros da Migração <i>Stolt</i> :	34
Tabela 2 – Variação em dB do dado antes e após a aplicação de ganho.	46

Lista de abreviaturas e siglas

UFF	Universidade Federal Fluminense
GGO	Departamento de Geologia e Geofísica
GISIS	Grupo de Imageamento e Inversão Sísmica
NMO	Normal Move Out
CMP	Common Mid Point
SU	Seismic Unix
RMS	Root Mean Square

Lista de símbolos

- Γ Letra grega Gama
- Λ Lambda
- ζ Letra grega minúscula zeta
- ∈ Pertence

Sumário

1	INTRODUÇÃO	16
2	REVISÃO BIBLIOGRÁFICA	18
2.1	Processamento Sísmico	18
2.1.1	Análise de velocidades	18
2.1.2	Correção NMO	20
2.1.3	Empilhamento	23
2.1.4	Migração	24
2.1.5	<i>Software</i> de processamento <i>Seismic Unix</i> - SU	29
3	METODOLOGIA	30
3.1	Dado sintético	30
3.1.1	Modelagem	30
3.1.2	Análise de velocidades	31
3.1.3	T-V Picks, NMO, Empilhamento	33
3.1.4	Migração	34
3.2	Dados de Nankai	35
3.2.1	Área de estudo	35
3.2.2	Exame, Reamostragem, Ordenação e Ganho	36
3.2.3	Análise de velocidade, NMO, Empilhamento, Migração	38
4	RESULTADOS	40
4.1	Dado Sintético	40
4.1.1	Modelagem e análise de velocidades	40
4.1.2	T-V picks, NMO, Empilhamento	42
4.1.3	Migração	43
4.2	Dados de Nankai	45
4.2.1	Exame, Reamostragem, Ordenação e ganho	45
4.2.2	Análise de velocidade, NMO, Empilhamento e Migração	48
5	DISCUSSÕES	51
6	CONCLUSÃO	53
	REFERÊNCIAS	54
	ANEXOS	56
	ANEXO A – PROGRAMA DO SEISMIC UNIX	57
A.1	Modelagem	57
A.2	Aquisição de linha 2D de dados sísmicos do modelo criado	59
A.3	Visualização de shots	62

A.4	Controle de qualidade dos pares tnmo-vnmo escolhidos	63
A.5	Correção NMO	65
A.6	Visualização dos dados após a correção NMO	67
A.7	Criação de CVP's com migração Stolt	70
A.8	Visualização dos CVP's	72
A.9	Migração Stolt	76

1 Introdução

Dentre os métodos geofísicos, o método sísmico é um dos mais utilizados, principalmente pela indústria de óleo e gás por engendrar boas imagens das estruturas em subsuperfície, sendo dividido em três grandes etapas: aquisição, processamento e interpretação. No primeiro, fontes sísmicas, com configuração específica, são utilizadas para gerar ondas que se propagam nas rochas em subsuperfície, que quando refletidas retornam e são gravadas por um arranjo de receptores. Uma vez registrados, esses dados necessitam de processamento (ARAUJO, 2022). Esse é o escopo deste trabalho e será melhor descrito posteriormente. O terceiro estágio do método é a interpretação dos produtos obtidos com o processamento. A partir deles, os dados sísmicos podem ser interpretados segundo a geologia, e utilizados na caracterização de reservatórios.

De acordo com Yilmaz (2001 apud Boletim SBGf, 2012), o processamento sísmico pode ser desmembrado nas seguintes etapas: reorganização dos dados, geometria, edição de traços, aplicação de filtros (espaciais, espectrais e outros), deconvolução, análise de velocidade e migração. Resumidamente, Rosa (2018) descreve o processamento como a aplicação de uma sequência de técnicas com a finalidade de se estimar a geometria e as propriedades elásticas das estruturas responsáveis pelas reflexões sísmicas registradas nos sismogramas.

Este trabalho teve como objetivo empregar um fluxo básico de processamento sísmico de linha 2D utilizado por Forel (2005) com o *Seismic Unix*. Por meio de *Shell Scripts* (arquivo de texto executável que permite executar uma sequência de comandos), foram feitas a modelagem de um dado sintético, aquisição de dados sísmicos provenientes dele e consequente processamento. Além disso, foi realizado também o processamento de uma linha 2D de dados reais fornecida pelo Prof. Greg Moore da Universidade do Havaí. Os dados foram coletados perto da costa do Japão, sobre a fossa de Nankai, onde a placa das Filipinas está subductando abaixo da Eurásia. A linha utilizada aqui e o estudo completo da área estão presentes em Moore et al. (1990).

O presente trabalho foi organizado da seguinte forma: Fundamentação teórica,

onde são descritos os conceitos embasadores do projeto e os métodos utilizados; Metodologia, em que o fluxo de processamento sísmico aplicado e os parâmetros adotados são explicados; Resultados, onde são apresentados por meio de explicações e figuras os principais produtos do trabalho; Discussões e Conclusões.

A partir dos resultados, pode-se concluir que o SU, aliado com Shell Scripts, é uma ferramenta útil e possibilita um contato inicial com tarefas básicas do processamento sísmico. A falta de conhecimento das ferramentas disponíveis do software e experiência em processamento sísmico, no entanto, limitam os resultados que podem ser alcançados. O problema de imageamento de refletores profundos, por sua vez, poderia ser resolvido através de uma nova configuração de aquisição de dados sísmicos do modelo gerado. E, tanto o produto do processamento do dado sintético quanto do dado real podem ser melhorados com uma migração variável no tempo e no espaço, lidando assim com a variação lateral de velocidade. Apesar das dificuldades relatadas, foram obtidos resultados semelhantes aos contidos na literatura e no artigo que embasaram este trabalho. No final, há uma seção que contém os *Shell Scripts* utilizados, que poderão servir de auxílio para aqueles que também queiram utilizar o *Seismic Unix* para o processamento de dados sísmicos.

2 Revisão Bibliográfica

2.1 Processamento Sísmico

O processamento dos dados adquiridos em um levantamento sísmico é de suma importância, visto que permite que seja feita uma melhor interpretação da geologia em subsuperfície a partir do produto final obtido. Cada dado possui o seu próprio fluxo de processamento. Uma das etapas do tratamento de um dado terrestre, por exemplo, é resolver o ruído causado pelas ondas Rayleigh, conhecidas como *ground roll*. Essa, no entanto, não está presente em levantamentos marinhos, sendo uma das principais diferenças no processamento de ambos os dados.

Outro aspecto refere-se à complexidade do processamento. Há inúmeras etapas que podem ser aplicadas como filtragens, edição e silenciamento de traços, demultiplexação, correções, análises espectrais e outros. Também pode ser feito um processamento mais básico, focado na análise de velocidades, correção NMO (*Normal Move Out*), empilhamento e migração, que foi o seguido por este trabalho. A seguir, serão apresentados alguns conceitos importantes (não nos aprofundaremos matematicamente) do fluxo adotado para um entendimento melhor dos objetivos e resultados logrados.

2.1.1 Análise de velocidades

A análise de velocidade tem como principal objetivo determinar as velocidades que promovem a devida correção NMO dos dados para maximizar o empilhamento dos eventos considerados como reflexões primárias (TELFORD; GELDART; SHERIFF, 1990). O empilhamento atenua ruídos aleatórios e as múltiplas, produzindo uma seção de afastamento zero.

A velocidade das formações pode ser obtida, diretamente, a partir de perfis sônicos nos poços em função da profundidade e, indiretamente, a partir da análise de velocidade de dados sísmicos multicanal. Existem diferentes tipos de velocidades na exploração

sísmica, de acordo com a área de aplicação (DONDURUR, 2018).

Uma característica intrínseca das rochas em subsuperfície é a velocidade intervalar que é a média entre duas interfaces. Considerando que dois refletores estão localizados nas profundidades h_2 e h_1 fornecendo tempos de viagem t_2 e t_1 , respectivamente, a velocidade intervalar da camada entre tais profundidades é definida como

$$V_{int} = \frac{h_2 - h_1}{t_2 - t_1} \quad (2.1)$$

A grande importância da obtenção de uma função de velocidade confiável é extrair a velocidade que fornece uma seção de empilhamento de maior qualidade, que está relacionada de forma direta às velocidades NMO e RMS. A primeira é utilizada antes do empilhamento (Figura 3), podendo ser definida como

$$V_{nmo} = \sqrt{\frac{x^2}{t(x)^2 - t(0)^2}} \quad (2.2)$$

onde o tempo de chegada de uma reflexão no afastamento x e o tempo no afastamento zero dessa reflexão são dados por $t(x)$ e $t(0)$ respectivamente. Vale a pena ressaltar que a morfologia do fundo do mar e as estruturas em subsuperfície afetam a velocidade NMO.

Já a velocidade média quadrática (V_{rms}) leva em consideração uma subsuperfície que consiste de n número de camadas e suas velocidades intervalares correspondentes V_1, V_2, \dots, V_n . É definida como

$$V_{rms}^2 = \frac{V_1^2 t_1 + V_2^2 t_2 + \dots + V_n^2 t_n}{t_1 + t_2 + \dots + t_n} = \frac{\sum V_i^2 t_i}{\sum t_i} \quad (2.3)$$

onde os tempos de viagem unidirecionais do sinal para cada camada são dados por t_1, t_2, \dots, t_n . Essa é a velocidade obtida durante a etapa de análise de velocidade dos dados sísmicos. As velocidades NMO e RMS aproximam-se para deslocamentos curtos em um meio com camadas plano-paralelas horizontais e sem variação lateral de velocidade intervalar em subsuperfície (DONDURUR, 2018; YILMAZ, 2001).

Uma ferramenta muito útil para a análise de velocidade é o gráfico de *Semblance* que auxilia nas escolhas de pares de tempo-velocidade (t-v) ao revelar quantos refletores estão presentes no dado e as velocidades NMO das ondas refletidas por eles (ALVES, 2021; ROBINSON; CORUH, 1988). As escolhas dos pares, geralmente, são feitas nos picos de energia plotados pelo *Semblance* (Figura 1).

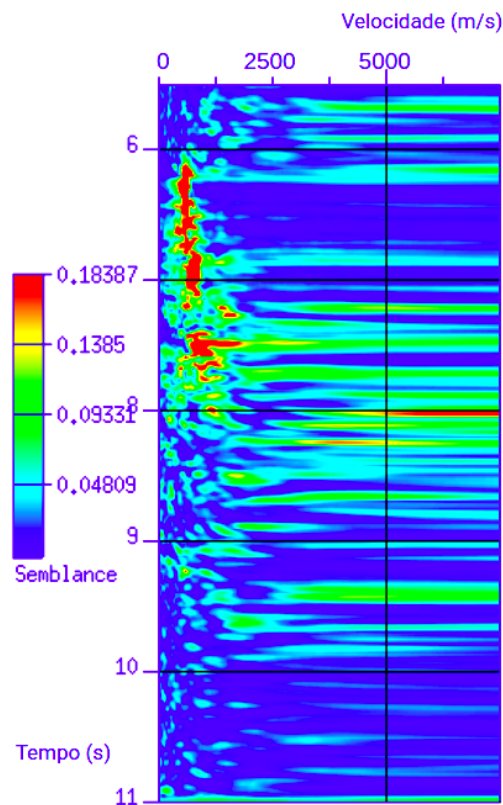


Figura 1 – Exemplo de Semblance utilizado na etapa de análise de velocidade.

2.1.2 Correção NMO

Uma etapa muito importante também dentro do processamento é a correção NMO. Numa coleta de tiro ou ponto de profundidade comum (CDP), os sinais de reflexão são registrados mais cedo pelos receptores localizados próximos ao ponto de origem do que as reflexões percebidas em locais distais (DONDURUR, 2018). Tomando uma única camada horizontal de velocidade constante, espera-se que a curva do tempo de viagem de reflexão em função do offset seja caracterizada por uma hipérbole (equação 2.2). A diferença temporal entre o tempo de viagem em um determinado *offset* e em uma configuração zero *offset* é chamada de NMO (YILMAZ, 2001).

Portanto, supondo que, em uma coleta de CMP, os tempos de viagem de reflexão em função do *offset* seguem trajetórias hiperbólicas, a correção NMO consiste na aplicação de um *shift* ou deslocamento de tempo, utilizando as velocidades quadráticas médias (RMS) obtidas da análise de velocidade (YILMAZ, 2001). Com isso, os tempos normais de deslocamento de cada hipérbole de reflexão em cada coleta de CMP são removidos (DONDURUR, 2018) (equação 2.2).

Em suma, de acordo com Dondurur (2018, p.460): “a correção NMO é um processo que remove o efeito no tempo devido ao afastamento entre a fonte e os receptores de modo que todas as chegadas de reflexão sejam reduzidas a $t(0)$, tempo de afastamento zero vertical bidirecional, em uma coleta CDP.” (Figura 2), entre a superfície e a interface.

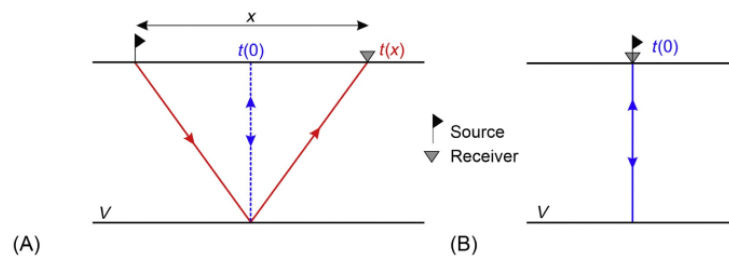


Figura 2 – Em (A), observamos a trajetória de um raio para uma reflexão registrada no afastamento (*offset*) x antes da correção NMO. Em (B), temos o caminho do raio para uma reflexão após a aplicação da correção NMO, que remove as distâncias de deslocamentos entre as fontes e os receptores, e realiza a redução dos tempos de chegada ao correspondente tempo de deslocamento zero $t(0)$. Fonte: (DONDURUR, 2018).

Caso seja utilizado um campo de velocidade NMO correto, espera-se, após a correção, que as reflexões registradas em todos os traços sejam reposicionadas em seu tempo de chegada de afastamento zero (Figura 3).

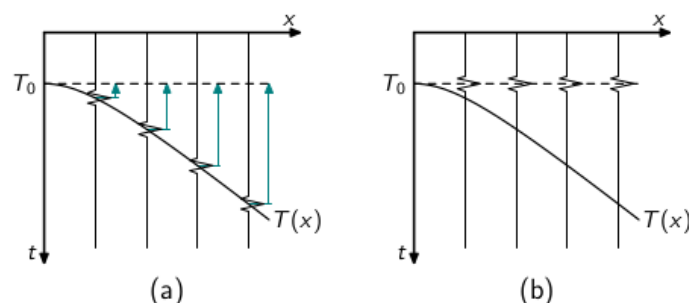


Figura 3 – Temos em (a) um esquema de seção CMP. O deslocamento que precisa ser aplicado a cada traço, prescrito pela curva $T(x)$, com a devida velocidade são indicados pelas setas azuis. Em (b) temos a correção NMO correspondente. Fonte: (BILOTI, 2022).

Em virtude disso, as reflexões primárias, antes com trajetórias hiperbólicas, se

apresentarão com um aspecto horizontalizado, achatadas, fornecendo um sub-sistema isotrópico, superfície composta por camadas horizontais (DONDURUR, 2018) (FIGURA 4).

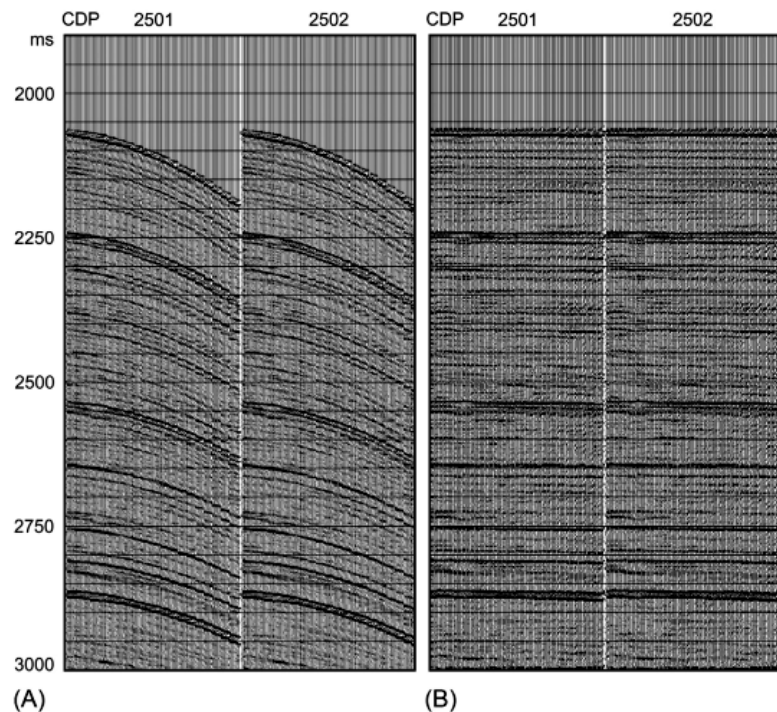


Figura 4 – Em (A) temos duas coletas sucessivas de CDP. Já em (B), temos ambas as coletas após a correção NMO. Fonte: (DONDURUR, 2018)

Se a velocidade empregada na correção NMO for mais lenta, a hipérbole de reflexão é supercorrigida, sendo caracterizada por uma concavidade para cima (Figura 5). Já no caso em que a hipérbole é subcorrigida, resultado de uma velocidade NMO mais rápida, a concavidade é para baixo (DONDURUR, 2018) (Figura 6).

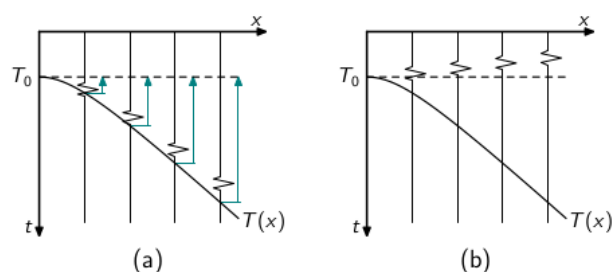


Figura 5 – Em (a), temos um esquema de uma seção CMP, contendo uma curva hiperbólica que foi traçada utilizando uma velocidade mais lenta que a ideal para o evento. Temos em (b) a correção NMO correspondente. Pode-se perceber que a hipérbole de reflexão foi sobrecorrigida. Fonte: (BILOTI, 2022).

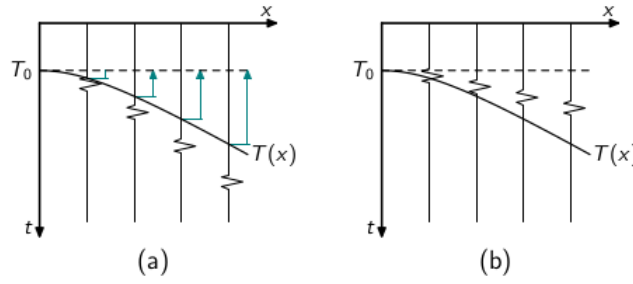


Figura 6 – Em (a), observamos um esquema de uma seção CMP, com uma hipérbole que foi traçada por meio de uma velocidade mais alta que a ideal para o evento. Já em (b), a correspondente correção NMO. Neste caso, a hipérbole de reflexão foi subcorrigida. Fonte: (BILOTI, 2022).

2.1.3 Empilhamento

Uma etapa bastante importante também do processamento é o empilhamento. Nesse, todos os traços em CDPs 2D ou 3D são somados gerando um único traço empilhado para cada coleta de CDP. A seção criada por todos os traços empilhados exibidos lado a lado, após a correção NMO, é denominada seção empilhada (ALVES, 2021) (Figura 7). Tendo em vista que cada coleta de CDP é transformada em um único traço empilhado, pode-se inferir que o número de CDPs em uma linha 2D ou em um volume 3D é justamente igual ao número de traços na seção empilhada (DONDURUR, 2018). Matematicamente, o empilhamento pode ser descrito por

$$S(t) = \frac{1}{N} \sum_{i=1}^N A_i(t) \quad (2.4)$$

onde N é o número de traços envolvidos, A_i é o valor da amplitude do i -ésimo traço na coleta CDP e S é dito o traço empilhado.

Tal passo do processamento sísmico, primeiramente, melhora a razão sinal-ruído. As amplitudes de reflexão primária em fase quando somadas têm uma interferência construtiva, reforçando o sinal. Todavia, as componentes de ruído aleatório, nos CDPs corrigidos por NMO, estão fora de fase e são inconsistentes traço a traço. Logo, quando somadas, atenuam-se.

Um segundo efeito considerável do empilhamento sobre os dados é referente às reflexões primárias e múltiplas. Enquanto as múltiplas permanecem fora de fase e

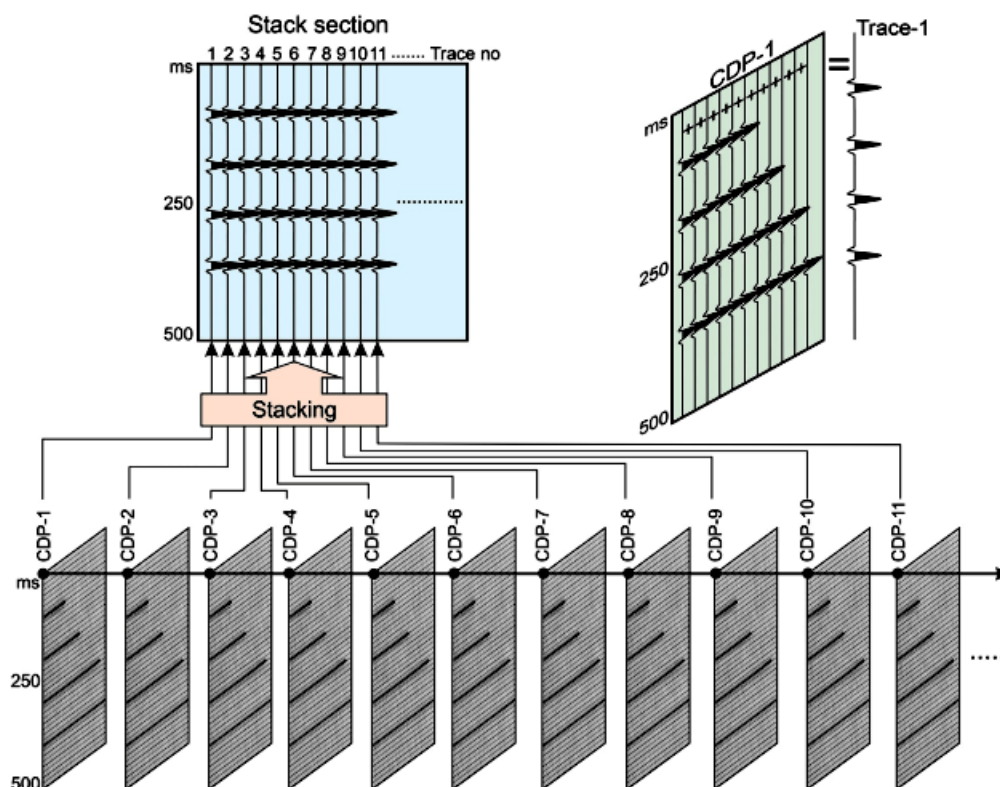


Figura 7 – Representação esquemática de empilhamento 2D. Após a correção NMO, os traços de um CDP são somados produzindo um traço empilhado (canto superior direito). Cada CDP (abaixo) fornece um único traço empilhado que ao serem exibidos lado a lado geram a seção empilhada da linha sísmica (canto superior esquerdo). Fonte: (DONDURUR, 2018).

ainda são hiperbólicas, as primárias apresentam-se em fase e achatadas nos CDPs após a correção NMO. Desta forma, após o empilhamento, as múltiplas são atenuadas e é fornecida uma análise de velocidade precisa para reflexões primárias (DONDURUR, 2018).

2.1.4 Migração

A migração é uma etapa do processamento sísmico em que as reflexões são movidas para as suas verdadeiras posições em subsuperfície na seção sísmica, tanto no tempo quanto no espaço (Figura 8), e a energia das difrações é colapsada de volta aos seus pontos de espalhamento (DONDURUR, 2018; YILMAZ, 2001). Este processo possui como principal objetivo melhorar a resolução espacial do dado e assemelhar a seção sísmica à seção transversal geológica abaixo da linha sísmica (ARAUJO, 2022).

A figura 8 ilustra o processo de migração de um refletor C-D que, em uma seção ZO

(*zero-offset*), teria os tempos de reflexão registrados logo abaixo dos receptores A e B, entre os tempos marcados por C' e D'. O papel da migração é exatamente reposicionar os pontos C' e D' e os pontos compreendidos entre eles em suas verdadeiras posições.

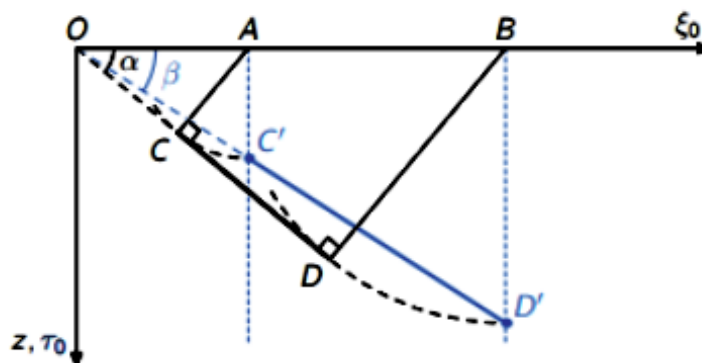


Figura 8 – Ilustração do evento sísmico antes da migração, com difrações nas terminações dos refletores sísmicos e um falso posicionamento do refletor (linha C'D'). Após a migração, as difrações são colapsadas e o refletor sísmico é posicionado em sua verdadeira localização (linha CD). Fonte: (BILOTI, 2022).

Por meio disso, os planos de falhas ficam muito mais claros e os efeitos de gravata de borboleta (*bow-tie*) são resolvidos ao serem convertidos em anticlinais e sinclinais, enquanto os refletores horizontais não são afetados (DONDURUR, 2018; YILMAZ, 2001) (Figura 9).

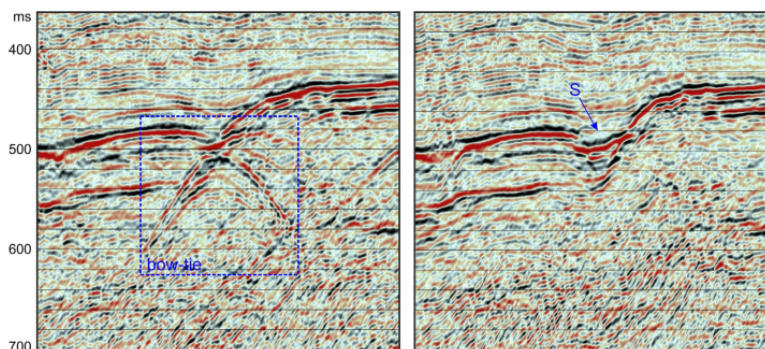


Figura 9 – Seção *zero-offset* contendo efeito bow-tie antes (à esquerda) e após a migração (à direita). Note que a migração corrige tal efeito e converte-o no sinclinal S. Fonte: (DONDURUR, 2018).

Antes de migrar o dado sísmico, é de suma importância que todos os eventos nele presentes sejam reflexões primárias, sendo livre de ruídos, ondas S e múltiplas energias. Outra premissa é o conhecimento da velocidade sísmica. Normalmente, na migração em tempo, utiliza-se o campo de velocidade RMS oriundo da análise de velocidade para correção NMO (ARAUJO, 2022; DONDURUR, 2018).

A migração pode ser realizada no tempo e na profundidade, aplicada antes ou após o empilhamento e baseada em diferentes algoritmos com suas próprias especificações, vantagens e desvantagens. A migração no tempo e pós-empilhamento, por exemplo, não é apropriada para dados em que os gradientes laterais de velocidade são significativos. Uma vez realizada em tal contexto, não produz a verdadeira imagem de subsuperfície (YILMAZ, 2001).

Neste trabalho foram realizadas a migração Stolt e Phase Shift (Gazdag) (no domínio do tempo), após o empilhamento. Ambas são denominadas técnicas espectrais pois são aplicadas no domínio frequência angular-número de onda ($\omega - k_x$, ou $\omega - k_x - k_z$) (ROSA, 2018). No entanto, a última utiliza o deslocamento de fase no domínio de Fourier e a primeira a transformada de Fourier em duas ou três dimensões. Essas duas técnicas possuem a dificuldade comum em lidar com variação lateral no campo de velocidades, todavia, quando aplicadas a meios homogêneos são relativamente rápidas em comparação a outros algoritmos (DONDURUR, 2018; ROSA, 2018).

Comparada às demais, a migração Stolt é mais eficiente computacionalmente, podendo produzir resultados aceitáveis de mergulhos estruturais de até 90°. Contudo, normalmente não lida bem com variações verticais e horizontais de velocidade e requer um meio de velocidade constante. Para resolver ou contornar tal deficiência, Stolt (1978), por meio de uma constante denominada fator de estiramento Stolt, tornou o algoritmo aplicável em áreas de geologia relativamente complexa marcadas por variações direcionais da velocidade (DONDURUR, 2018). Outra proposta de resolução do problema heterogêneo foi idealizada por Beasley et al (1988).

Considerando que a velocidade média é constante, pode-se expressar a migração como um mapeamento direto (STOLT, 1978) de frequência ω para k_z (número de onda vertical) dado pela equação:

$$P(k_x, k_z, t = 0) = \left[\frac{v}{2} \frac{K_z}{\sqrt{k_x^2 + k_z^2}} \right] \quad (2.5)$$

$$P(k_x, k_z, t = 0) = P[k_x, 0, \omega = \frac{v}{2} \sqrt{k_y^2 + k_z^2}] \quad (2.6)$$

na qual a seção de deslocamento zero é dada por $P(k_x, z = 0, \omega)$ e $P(k_x, k_z, t = 0)$ é a seção migrada no domínio do número de onda de frequência. O fluxograma do

algoritmo Stolt é descrito na Figura M.

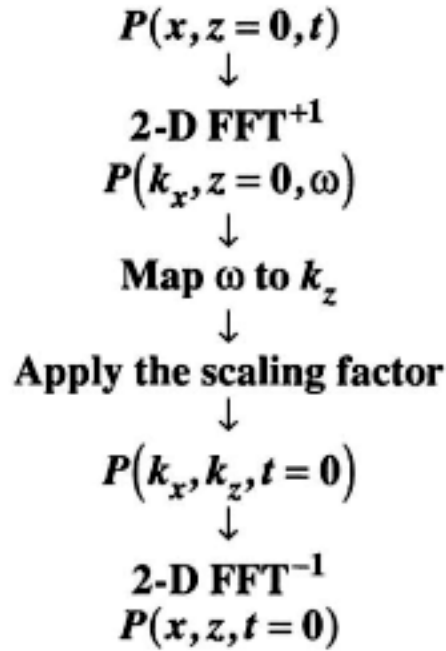


Figura 10 – Fluxograma utilizado para a migração Stolt de velocidade constante no domínio f-k.

Note que, primeiramente, aplica-se a Transformada de Fourier 2D ao campo de onda de entrada $P(x, z = 0, t)$ associado ao CMP empilhado para obter $P(k_x, z = 0, \omega)$. Em seguida, mapeia-se o campo de onda de ω a k_z utilizando a relação de dispersão dada por

$$\omega = \frac{v}{2} \sqrt{k_x^2 + k_z^2} \quad (2.7)$$

O terceiro passo é aplicar o fator de escala S (equação 2.8) como parte do mapeamento. Para obter $P(k_x, k_z, t = 0)$ Invoca-se o princípio de imagem configurando $t = 0$ e, por último, a Transformada de Fourier Inversa é aplicada resultando na seção migrada $P(x, z, t = 0)$ (YILMAZ, 2001).

$$S = \frac{v}{2} \frac{k_z}{\sqrt{k_x^2 + k_z^2}} \quad (2.8)$$

Por sua vez, o método de deslocamento de fase (GAZDAG, 1978) pode lidar com variações verticais de velocidade e, assim como o método exposto anteriormente, produz bons resultados até mergulhos estruturais de 90°, contanto que não haja

variação lateral de velocidade. Gazdag e Sguazzero (1984) expandiram o método para minimizar essa deficiência. O fluxograma adotado pelo método é exibido pela figura 11 (YILMAZ, 2001).

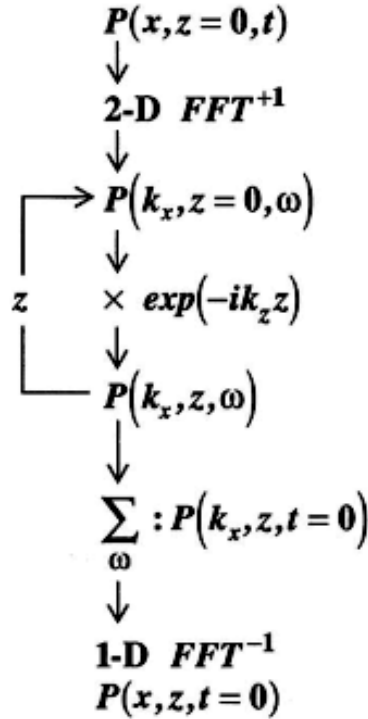


Figura 11 – Fluxograma do método de migração Gazdag.

Note que inicia-se o método trabalhando com a seção empilhada (uma aproximação para a seção de deslocamento zero $P(x, z = 0, t)$) e executa-se a transformada de Fourier 2D obtendo como resultado $P(k_x, z = 0, \omega)$. Em seguida, para cada frequência ω , o campo de onda transformado $P(k_x, z, \omega)$ é extrapolado com um operador de deslocamento de fase $\exp(-ik_z \Delta z)$ com o intuito de se obter o campo de onda $P(k_x, z + \Delta z, \omega)$ em profundidade $z + \Delta z$. Um novo operador de extrapolação com a velocidade definida para aquele valor de z é calculado a cada passo z .

Para obter a seção migrada $P(k_x, k_z, t = 0)$ no domínio da transformada, invoca-se o princípio de imageamento configurando $t = 0$ em cada etapa da extrapolação. Os dois passos anteriores são repetidos para continuação descendente e imageamento, respectivamente, para todos os passos de profundidade com o objetivo de se obter a seção migrada no domínio da transformada $P(k_x, z, t = 0)$. A última etapa consiste na aplicação da transformada inversa na direção x para se obter a seção migrada $P(x, z, t = 0)$.

2.1.5 *Software* de processamento *Seismic Unix* - SU

O *software Seismic Unix*, criado por Cohen & Stockwell (2002) é um pacote de ferramentas gratuito para o processamento de dados sísmicos. É mantido e atualizado de forma regular pelo *Center for Wave Phenomena* (CWP) na *Colorado School of Mines* (CSM), podendo ser encontrado em: <http://www.cwp.mines.edu/cwpcodes/>. O SU pode ser baixado e instalado em praticamente qualquer sistema *Unix* e *Linux*.

O software não demanda aprofundado conhecimento de programação. Faz-se necessário somente o conhecimento de comandos básicos no terminal do *Unix*. O livro de Forel (2005) disponibiliza um capítulo específico para introdução a estes comandos e exemplos de *Shell Scripts*. Essa combinação entre pacotes do SU e *Shell Scripts* foi adotada por este trabalho por possibilitar que várias etapas de processamento sejam rodadas em uma simples compilação.

3 Metodologia

O presente trabalho seguiu as etapas de processamento contidas no livro “*Seismic Data Processing with Seismic Unix*”, escrito por David Forel, Thomas Benz e Wayne D. Pennington. A obra concede ao leitor uma breve introdução ao ambiente *Linux* e descreve a metodologia adotada pelos autores para a modelagem e processamento de dados sintéticos e, posterior, processamento de dois dados reais: fossa de Nankai, no Japão, e Taiwan (Figura 12). Nela estão contidos os *scripts* utilizados nas etapas de modelagem, análise de velocidade, correção NMO, empilhamento, e migração Stolt e Gazdag pós-empilhamento. O trabalho consistiu em reproduzir tal fluxo de processamento desde a modelagem dos dados sintéticos até as migrações dos dados de Nankai com os dois tipos de migração citados na fundamentação teórica.

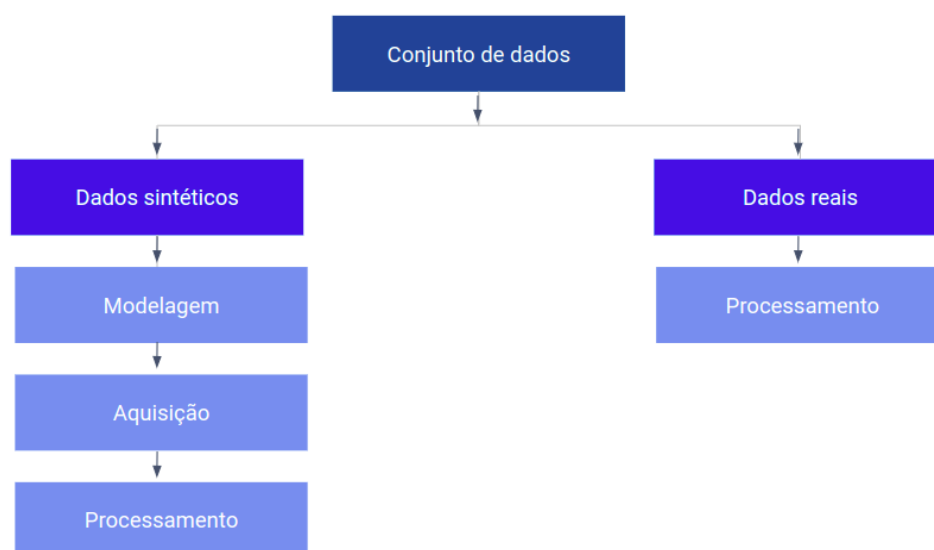


Figura 12 – Etapas de processamento do livro *Seismic Data Processing with Seismic Unix* seguidas por este trabalho.

3.1 Dado sintético

3.1.1 Modelagem

Por meio do script *model4.sh*, foi desenvolvido um modelo geológico, que foi utilizado posteriormente para adquirir uma linha 2D de dados sísmicos. Nesse script, foi

utilizado o programa *trimodel* que permite ao usuário definir as dimensões do modelo, os limites das camadas (interfaces) e suas respectivas velocidades.

Para adquirir uma linha 2D de dados sísmicos por meio do modelo geológico, criado anteriormente, utilizou-se o script *acq4.sh*. Nele configura-se as variáveis *nangle* (número de ângulos ou raios que emanam da fonte), *fangle* (primeiro ângulo do leque de raios que emanam da fonte) e *langle* (último ângulo do leque de raios que emanam da fonte). Esses parâmetros auxiliam na aquisição de conjuntos de dados sísmicos completos com o modelo. Outros parâmetros configurados concernentes à aquisição foram:

- 200 tiros igualmente espaçados em intervalos de 50m;
- 60 *split-spread traces* foram registrados a partir de cada local de tiro;
- espaçamento do geofone foi de 50m;
- faixa de afastamentos de geofone foi -1475m a +1475m;
- locais de tiro variaram de 0 km a 9,95 km e
- localizações dos geofones variaram de -1,475 km a 11,425 km.

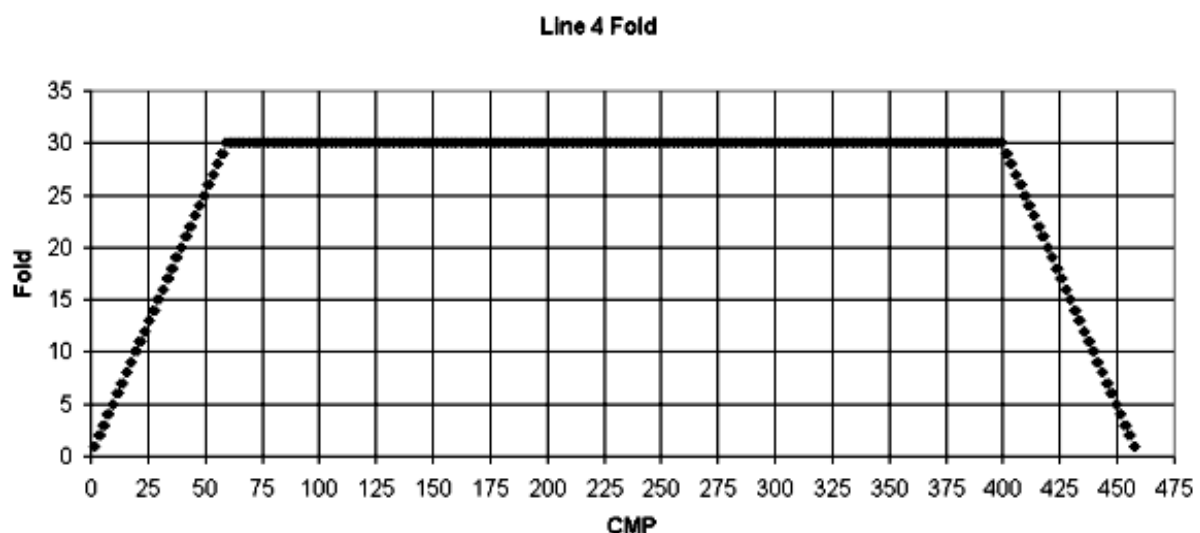
O conjunto final de dados possui 12.000 traços (200 tiros X 60 geofones por tiro) e foi salvo no arquivo *seis4.su*. Os *shot gathers*, por sua vez, podem ser visualizados fazendo o uso do script *showshot.sh*.

3.1.2 Análise de velocidades

Uma vez que a linha 2D de dados sísmicos já havia sido adquirida, a próxima etapa foi classificar os *shot gathers* em CMP *gathers* e realizar a análise de velocidade em CMPs selecionados. Por intermédio do script *sort2cmp.sh* foram ordenados 458 CMP *gathers*, no entanto, os primeiros e os últimos *gathers* não são *full fold* como pode ser visto na Figura 14. Os CMPs podem ser visualizados individualmente com o uso de *showcmp.sh*. Como os dados sintéticos aqui apresentados são livres de ruídos, pode-se prosseguir de *shot gathers* para a migração rapidamente. Essa e as demais etapas do processamento dos dados sintéticos podem ser visualizadas na figura 13.



Figura 13 – Fluxo de processamento do dado sintético.

Figura 14 – CMP *fold* do modelo criado.

Para a análise de velocidade foram escolhidos dezoito CMPs *full fold* em um intervalo regular ao longo da linha: 60, 80, ..., 380, 400. O intuito desse passo é determinar uma função de velocidade de tempo ($t-v$) que nivela as reflexões em cada CMP analisado para, desta forma, reforçar as reflexões primárias no processo de empilhamento posterior. Para tal fim, foram utilizadas duas ferramentas incorporadas em um script de análise de velocidade interativa (*iva.sh*): gráfico de *Semblance* (*velan*) e CVS (*Constant Velocity Stacks*). Essa última pode somente ser feita para uma linha de CMPs e oferece ao usuário uma visão simplificada da seção final empilhada. O melhor resultado final empilhado está atrelado a uma maior continuidade e às mais altas amplitudes dos eventos. Já aquela permite fazer a escolha dos pares de tempo e

velocidade que horizontalizam as reflexões. Na Figura 21 é exemplificado o uso das duas ferramentas para a análise de velocidade do CMP 60. O mesmo processo se repete para os 17 CMPs escolhidos.

Após as escolhas dos pares t-v, é essencial que o controle de qualidade seja feito, por isso, o mesmo script permite ao usuário conferir a qualidade das velocidades selecionadas por intermédio de três gráficos:

- um gráfico do perfil de velocidade,
- um gráfico do CMP corrigido por NMO, e
- um gráfico do rastreamento de pilha [stack trace] repetido oito vezes.

Espera-se que o perfil de velocidade seja suave e mostre velocidades geralmente crescentes com a profundidade. Já o CMP após a correção de NMO deve apresentar reflexões primárias horizontais. Uma reflexão primária curvada para cima demonstra que a velocidade escolhida é muito lenta (sobrecorrigida) e uma reflexão primária curvada para baixo revela que a velocidade escolhida é muito rápida (sub-corrigida). O script também oferece uma quarta ferramenta de QC (*Quality Control*) quando é decidido fazer a análise novamente em um CMP: as suas escolhas feitas anteriormente são colocadas no velan. Isso possibilita que diferentes ou antigas escolhas sejam feitas com facilidade.

3.1.3 T-V Picks, NMO, Empilhamento

Visto que um arquivo txt de escolhas de velocidade de tempo (t-v), em uma certa quantidade de CMPs, foi criado após a análise de velocidade, faz-se necessária a verificação da qualidade dessas escolhas. Para isso, foi utilizado o script *tvQC.sh* em que é verificado se o número de pares *tnmo-vnmo* corresponde ao número de CMPs e se cada entrada da série temporal (conjunto de valores *tnmo*) aumenta com o tempo. Essas são premissas importantes para o programa *sunmo* contido no script *nmo4.sh* que realiza a correção de movimento normal (NMO). Uma vez corretas, o script aplica a correção aos CMPs selecionados e interpola entre eles para cobrir todo o dado. Os dados após a correção NMO podem ser visualizados através do script *iview.sh*. O

empilhamento, processo em que os traços são somados gerando um único traço de configuração *zero-offset*, foi realizado com apenas um comando.

3.1.4 Migração

A migração é um processo no qual os refletores são reposicionados em suas verdadeiras localizações em subsuperfície (DONDURUR, 2018; YILMAZ, 2001). Para aplicá-la ao dado sintético analisado neste trabalho, utilizou-se painéis de velocidade constante de migração (CVP) com migração *Stolt*.

O programa *sustolt*, presente no script *migcvp.sh*, foi utilizado por ser rápido. Ele possui alguns parâmetros importantes que precisam ser configurados antes da migração. Os primeiros dois parâmetros são *cdpmin* e *cdpmax*, que permitem ao usuário aplicar DMO (correção de mergulho; NMO corrigido para leitos de imersão em dados de pré-empilhamento) a uma faixa específica de CMPs de entrada. Ambos são obrigatórios e não possuem um valor padrão. O terceiro parâmetro, *dxcdp*, não tem um valor padrão e é obrigatório. Refere-se à distância entre CMPs adjacentes que pode ser adquirida por meio da geometria da aquisição dos dados. O quarto e quinto parâmetros, *tmig* e *vmig*, referem-se a tempos correspondentes às velocidades RMS respectivamente. O sexto parâmetro, *smig*, fator de estiramento (configurado como 0,6 se V_{rms} está aumentando). O sétimo parâmetro, *vscale*, é um fator de escala aplicado a todas as velocidades. E os últimos, *lstaper* e *lbtaper*, referem-se a *length of side tapers (traces)* e *length of bottom taper (samples)* respectivamente. Neste trabalho, foi escolhida a configuração presente na tabela 1.

Tabela 1 – Parâmetros da Migração *Stolt*:

Parâmetro	Valor escolhido
<i>cdpmin</i>	80
<i>cdpmax</i>	150
<i>dxcdp</i>	70.71
<i>tmig</i>	-
<i>vmig</i>	-
<i>smig</i>	1.0
<i>vscale</i>	1.9
<i>lstaper</i>	20
<i>lbtaper</i>	100

Para determinar o valor correto do parâmetro *vscale*, utilizou-se o script *migcvp.sh* contendo o programa *sustolt* que migra diversas vezes uma seção empilhada, cada vez com uma velocidade constante. O produto de cada migração é chamado de “painel” sendo utilizados vinte no total (primeiro painel com velocidade de 1200 m/s até o último painel de 5000 m/s com incremento de 200 m/s). A ideia desta etapa é utilizar os painéis para determinar o valor correto de *vscale* que colapsa a difração do canal de areia com velocidade de 3015 m/s conhecida a priori pela modelagem.

Somente a parte do modelo entre os CMPs 80 e 150 foi migrada, exatamente para examinar a difração do canal presente próximo ao CMP 110. O valor ideal de *vscale* foi definido quando a melhor imagem de migração para a velocidade em torno de 3000 m/s, que sabe-se que é a verdadeira velocidade do canal, foi encontrada. Após testes, o valor de *vscale* foi definido como 1.9. Uma vez tendo o valor correto, pode-se migrar todo o modelo. O script *iviewcvp.sh* possibilita a visualização dos *CVPs* de migração.

Para enfim fazer a migração do modelo, foi utilizado o script *migStolt.sh* que tem uma única função de velocidade variável no tempo, que foi aplicada em toda a seção. A função t-v foi projetada para se ajustar melhor aos dados próximo ao CMP 110, onde a difração do canal de areia está localizada. Foram selecionadas velocidades em momentos diferentes tendo como base os painéis de migração de velocidade constante. Descobriu-se que a velocidade de migração de 1500 m/s era apropriada em 0,70 segundos; maiores velocidades de migração forneceram imagens apropriadas em momentos posteriores.

3.2 Dados de Nankai

3.2.1 Área de estudo

A linha 2-D real de dados sísmicos processada neste trabalho foi fornecida pelo Prof. Greg Moore da Universidade do Havaí. Os dados foram coletados perto da costa do Japão, sobre o vale de Nankai, onde a placa das Filipinas está se subductando sob a Eurásia (Figura 15). Os dados “Nankai” foram coletados pela Universidade do Texas, pela Universidade de Tulsa e pela Universidade de Tóquio. Com base neste conjunto de dados, foi publicado um artigo (MOORE et al., 1990) no qual esta linha é denominada

NT62-8.

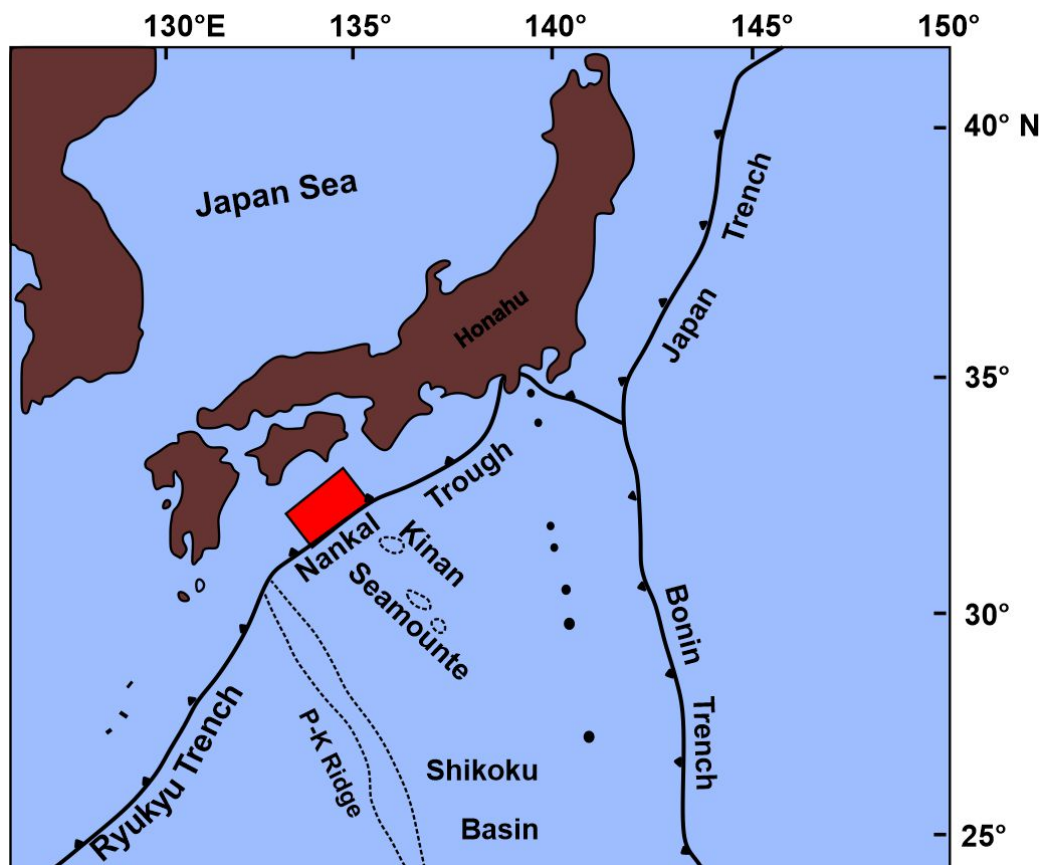


Figura 15 – Mapa mostrando o local de aquisição da linha sísmica 2D (retângulo vermelho). Adaptado de: (MOORE et al., 1990).

3.2.2 Exame, Reamostragem, Ordenação e Ganho

Os dados sísmicos de Nankai disponíveis são um arquivo de *shot gathers* e um arquivo empilhado (esse último foi utilizado na publicação do artigo em 1990). Ao examiná-los, observa-se que o primeiro possui 326 *shot gathers* e o comprimento do traço é de 11 segundos visto que o número de amostras por traço (ns) é 5500 e o intervalo de amostragem de traço (dt) é de 2.000 microssegundos (5.500 amostras/traço x 0,002 segundos/amostra). Possui 401 CMPs. Já o comprimento do traço do arquivo empilhado é de 9 segundos (2250 amostras por traço com intervalo de 4 ms entre elas) e possui 2869 CMPs. Nota-se que, neste trabalho, somente uma parcela do dado utilizado na publicação será processado. Vale a pena também ressaltar que os dados foram coletados em águas muito profundas com aproximadamente 4500 metros de lâmina d'água. Seguindo o padrão do arquivo empilhado, foi feita uma nova amostragem de 2 ms para 4 ms, utilizando todos os 11 segundos de dado.

Uma primeira etapa muito importante antes do processamento é o exame dos dados. Por isso alguns *shot gathers* foram examinados para averiguar se havia algum receptor ruim que poderia trazer dano a todo o dado. Após a análise, não foi encontrado nenhum traço ruidoso, provavelmente isso é resultado de um pré-processamento realizado em que os traços com ruídos foram excluídos. Essa e as demais etapas do fluxo de processamento podem ser visualizadas na figura 17.

Diferente dos dados sintéticos analisados anteriormente, dados reais exigem inspeção, análise de frequência, aplicação de ganho, dentre outras etapas antes de dar prosseguimento à análise de velocidade. Primeiramente, então, foi feita uma filtragem *anti-alias* com início em 85 Hz e frequências completamente zero a 95 Hz, com segurança abaixo da frequência de *Nyquist* para dados de 4 ms: 125 Hz. Com o auxílio do script *Nsort2cmp.sh*, os dados foram filtrados, reamostrados e ordenados de *shot gather* para *CMP gather* gerando 401 CMPs. A Figura 16 revela que somente os CMPs à direita do CMP 933 são full fold.

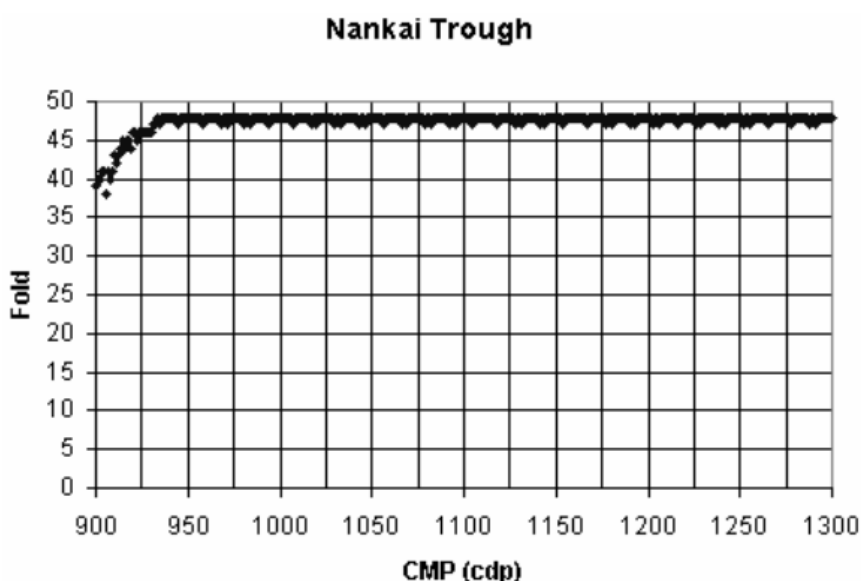


Figura 16 – CMP *fold* dos dados de Nankai.

Uma última etapa ainda se fez necessária: a correção da divergência esférica. Para se determinar qual o ganho (parâmetro *tpow*) correto que deveria ser aplicado em todos os CMPs, foram escolhidos dois CMPs específicos para uma análise criteriosa: 1055 e 1280. Por meio de um script que produz três gráficos para controle de qualidade, foram feitos testes com valores distintos de *tpow* (1.8, 2.2, 2.5). Foi dada uma ênfase no exame do terceiro display. O ideal era que abaixo de seis segundos (camada de sedimentos) uma linha de amplitude vertical fosse encontrada. Após os testes, foi escolhido o valor 2.1 por tratar bem os dados e, por fim, a correção da divergência

esférica foi aplicada a todo o dado com auxílio de somente uma linha de comando no terminal.



Figura 17 – Fluxo de processamento dos dados de Nankai.

3.2.3 Análise de velocidade, NMO, Empilhamento, Migração

Para realizar então a análise de velocidade foram escolhidos 15 CMPs *full fold*: 933, 958, 983,...,1233, 1258, 1283. O valor do parâmetro *perc* foi alterado para 95 (acomodação necessária para dados reais). Os painéis CVS, por sua vez, variam de 1000 m/s a 6000 m/s. Ao analisar o dado empilhado de Nankai, observa-se que a camada de água tem aproximadamente 6 segundos (tempo bidirecional), então, para otimizar a análise, os arquivos CMPs foram janelados, dando enfoque as reflexões contidas entre os tempos de 5,5 segundos e 11 segundos por intermédio do comando:

```
suwind < Ncdps4g.su tmin = 5,5 tmax = 11 > Ncdps4g5511.su
```

Uma vez que não foram feitas escolhas a partir do tempo zero, o arquivo de seleção de saída teve que ser modificado adicionando o tempo zero e a velocidade

correspondente a este tempo às séries *tnmo* e *vnmo* para assim aplicar a correção NMO. Essa, por sua vez, foi realizada com o auxílio do script *Nnmo.sh* que aplica NMO e empilhamento à linha 2D de CMPs obtidos. Comparando o arquivo empilhado gerado com o arquivo empilhado disponibilizado, observa-se que as imagens não são correspondentes. Isso se deve ao fato do segundo arquivo não ser somente empilhado, porém, também ser migrado.

Para realizar a migração *Stolt*, foi utilizado o script *Nmigcwp.sh* (uma adaptação do script anterior utilizado nos dados sintéticos para os dados de Nankai). O parâmetro *dxcdp* foi configurado com o valor 16.667 (distância relatada no artigo de Moore et al, 1990). Tendo em vista que as velocidades RMS não eram conhecidas para esses dados, *vscale* foi definido com valor 1.0. O processamento não foi realizado por velocidade, e sim, por imagem. A primeira velocidade dos painéis foi de 1400 m/s e a última de 2000 m/s. Utilizando o script *Nviewcwp.sh*, pode-se visualizar os painéis gerados que foram janelados entre 5,5 e 8 segundos uma vez que não havia refletores significativos abaixo de 7,8 segundos.

Uma segunda migração foi feita com o programa *sugazmig*, uma versão SU da migração de mudança de fase (*Phase Shift Migration*) de Jenó Gazdag para dados de deslocamento zero. De fácil aplicação, *sugazmig* necessita somente dos pares t-v e do valor *dx*, nome dado pelo programa à distância do compartimento do CDP. Tendo como base as migrações *Stolt* anteriores adotadas, as velocidades são uma aproximação. Vale ressaltar que para *sugazmig*, as velocidades são velocidades intervalares. Para exibir o arquivo migrado, foi utilizado o seguinte comando:

```
suwind < Ngazmig.su tmin=5,5 tmax=8  
| suximage perc=95 title="Nankai stack + sugazmig" &
```


4 Resultados

4.1 Dado Sintético

4.1.1 Modelagem e análise de velocidades

O modelo criado, através da utilização do script de modelagem, possui quatro camadas um tanto horizontais que se estendem por todo o modelo, um canal enterrado (entre as interfaces 5 e 6) e um pinchout (entre as interfaces 7 e 8) (Figura 18). As camadas são homogêneas e isotrópicas cada uma tendo somente uma única velocidade acústica (onda P) (Figura 19).

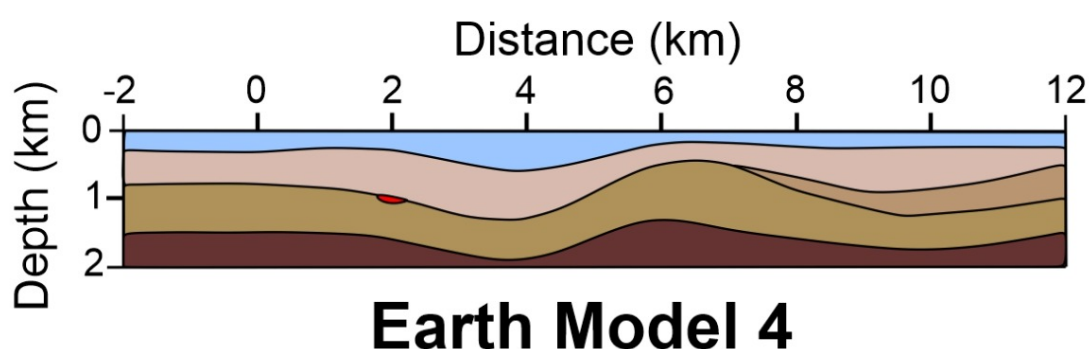


Figura 18 – Modelo com quatro camadas um tanto horizontais que se estendem por todo o modelo, um canal enterrado (em aproximadamente 2 km entre as camadas 2 e 3) e um pinchout (de 7 km até 12 km, entre as camadas 2 e 3).

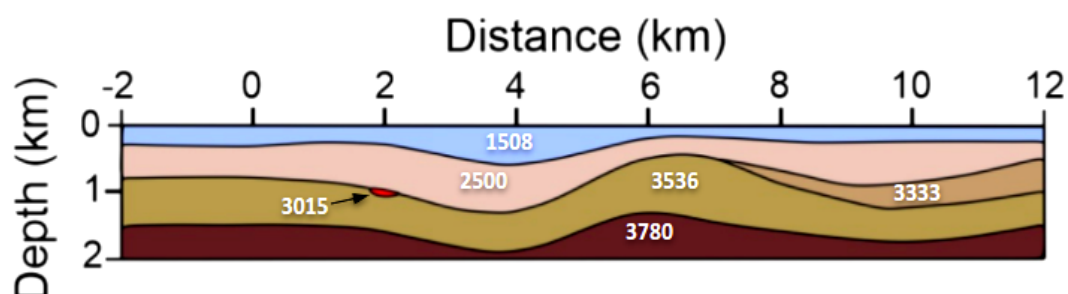


Figura 19 – Velocidade em m/s de cada camada do modelo gerado.

Os exemplos de *shots*, após adquirir a linha 2D de dados sísmicos por meio do modelo geológico, podem ser vistos na figura 20. Na figura 21 é exemplificado o uso

das duas ferramentas para a análise de velocidade do CMP 60. Já a figura 22 mostra um perfil de velocidade obtido após a análise de velocidades neste mesmo CMP.

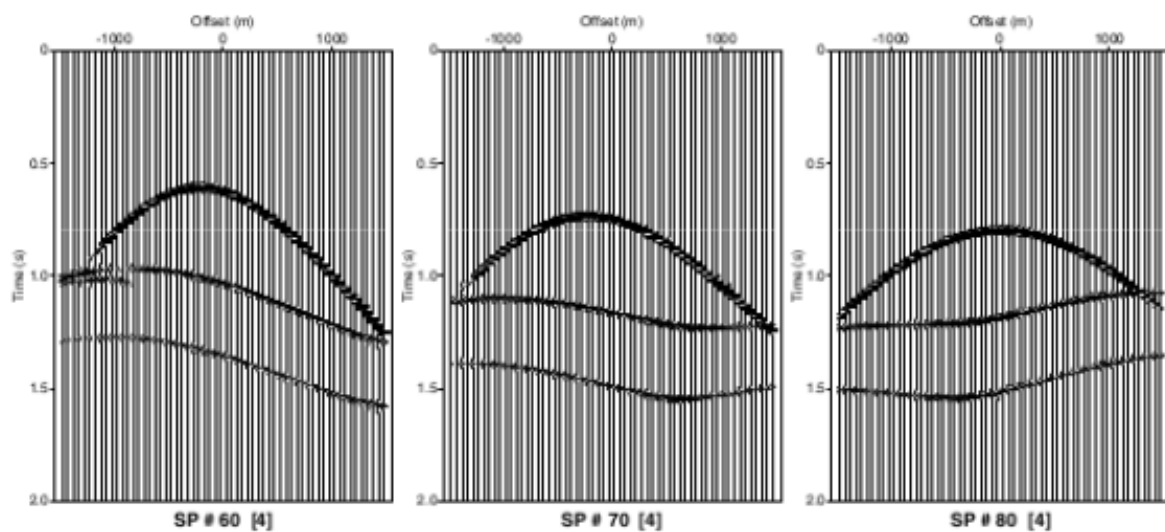


Figura 20 – Exemplos de shot gathers do modelo criado.

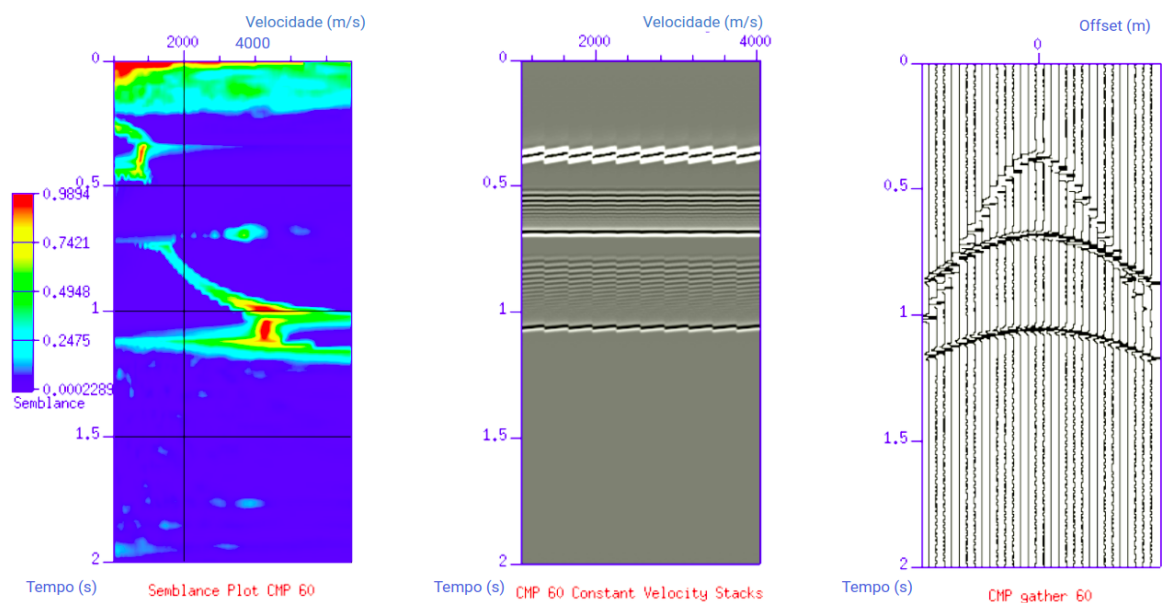


Figura 21 – Ferramentas para análise de velocidade. À esquerda: Velan. Centro: CVS plot. À direita: Plot do CMP 60 analisado.

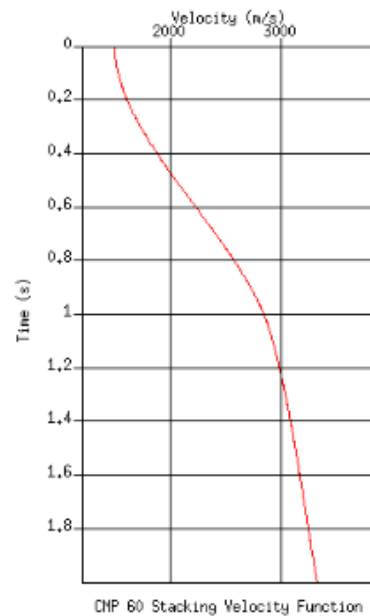


Figura 22 – Perfil de velocidade do CMP 60 analisado.

4.1.2 T-V picks, NMO, Empilhamento

Após a análise de velocidade (subseção 3.2.3) e o controle de qualidade das escolhas realizadas, a próxima etapa foi a aplicação da correção NMO na qual as reflexões antes hiperbólicas são horizontalizadas conforme mostrado na figura 23. Em seguida o dado corrigido por NMO foi o empilhado (Figura 24).

Observando as porções com certa declividade de cada refletor, percebe-se que nesses locais a análise de velocidade deve ser feita com mais frequência, uma vez que a interpolação realizada pelo programa pode gerar distorções ou imagens difusas, fora de foco. Outro fato observado é que essa análise faz-se necessária também frequentemente em locais onde a velocidade muda rapidamente (vertical ou lateralmente), como no canal que criou a difração localizada por volta do CMP 110 e em aproximadamente 0,8 segundos. Examinando ainda o dado empilhado, nota-se que há um *gap* do refletor mais profundo entre os CMPs 310 e 390. Isso se deve a caminhos de raios diferentes que fazem com que a camada inferior não retorne reflexões no intervalo.

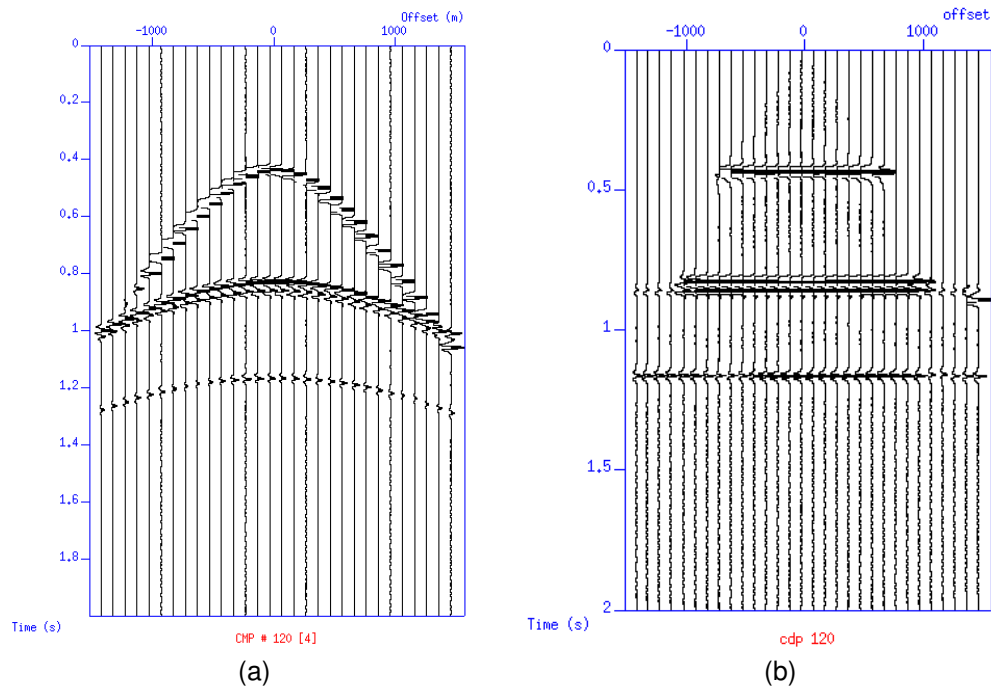


Figura 23 – Exemplo de aplicação da correção NMO. a) CMP antes e b) após a correção. Fonte: o autor.

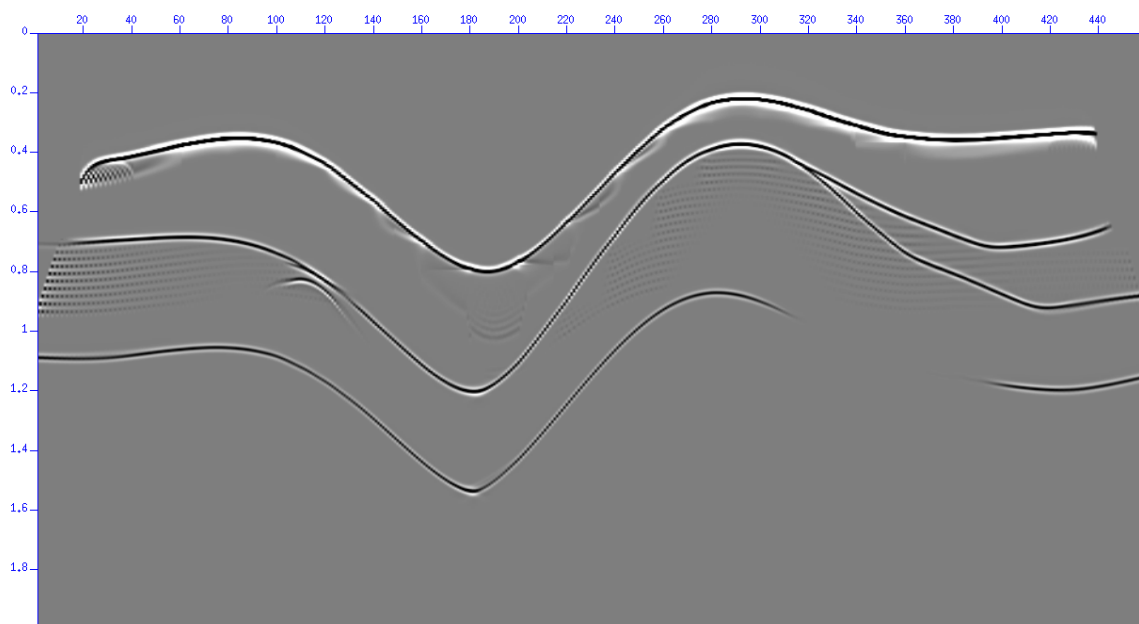


Figura 24 – Dado sintético empilhado.

4.1.3 Migração

A migração aplicada ao dado sintético foi a migração *Stolt* como visto na seção de Metodologia. O programa *sustolt* foi utilizado para produzir “painéis” de velocidade constante para observamos o colapso das difrações. Quando estas eram colapsadas

por uma determinada velocidade, concluíamos que essa era a velocidade correta da camada. Por fim, utilizou-se o mesmo programa para migrar o dado após a escolha das velocidades e tempos.

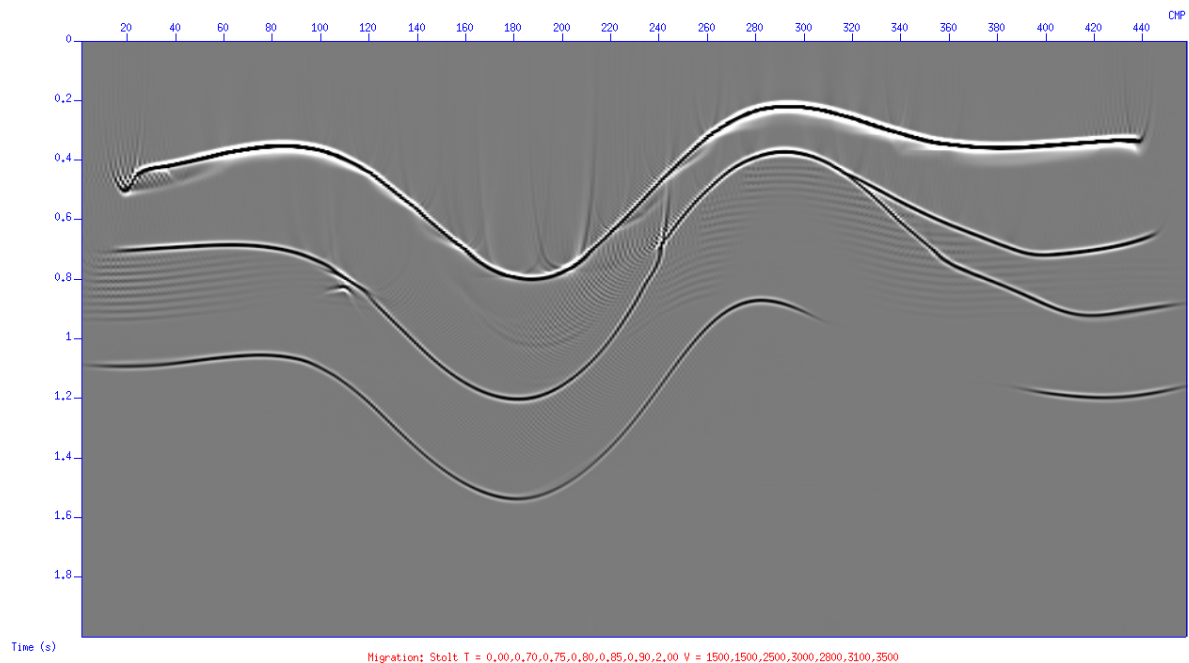


Figura 25 – Dado sintético migrado.

Comparando o dado empilhado (Figura 24) com o dado migrado (Figura 25) observa-se que:

- a difração do canal de areia por volta do CMP 110 foi colapsada (reduzida);
- os sinclinais inferiores próximos ao CMP 180, antes estreitos, foram suavizados;
- lembremos que a função de velocidade utilizada na migração era adequada para o canal de areia. Para o CMP 240, por exemplo, não foi apropriada, gerando uma distorção no segundo refletor, resultado da brusca mudança de velocidade de 1500 m/s para 2500 m/s em aproximadamente 0,7 segundos.

4.2 Dados de Nankai

4.2.1 Exame, Reamostragem, Ordenação e ganho

Após o exame dos dados de Nankai e posterior ordenação para a família de CMPs, foi aplicada uma filtragem *anti-alias* com início em 85 Hz e frequências completamente zero a 95 Hz no dado original (Figura 26), com segurança abaixo da frequência de *Nyquist* para dados de 4 ms: 125 Hz. Ao analisar o espectro de amplitudes do dado original, foi decidido remover frequências que aparentam ser ruído aleatório. Com isso, um filtro trapezoidal de 16 - 21 Hz também foi utilizado com o intuito de eliminar os ruídos de baixa frequência. Conforme mostrado no espectro de amplitudes da figura 27, tanto os ruídos aleatórios quanto as altas frequências foram eliminadas.

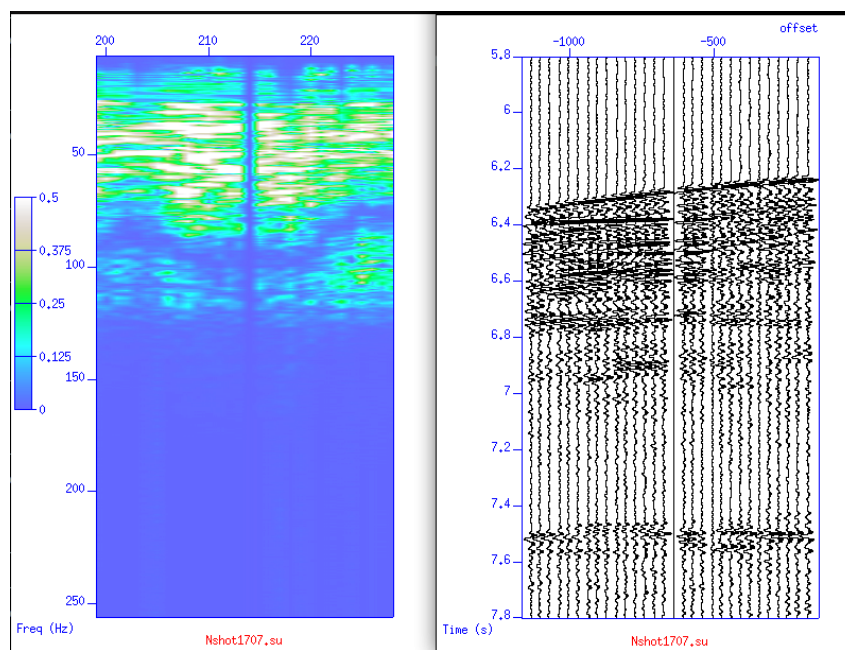


Figura 26 – Espectro de amplitudes do dado original.

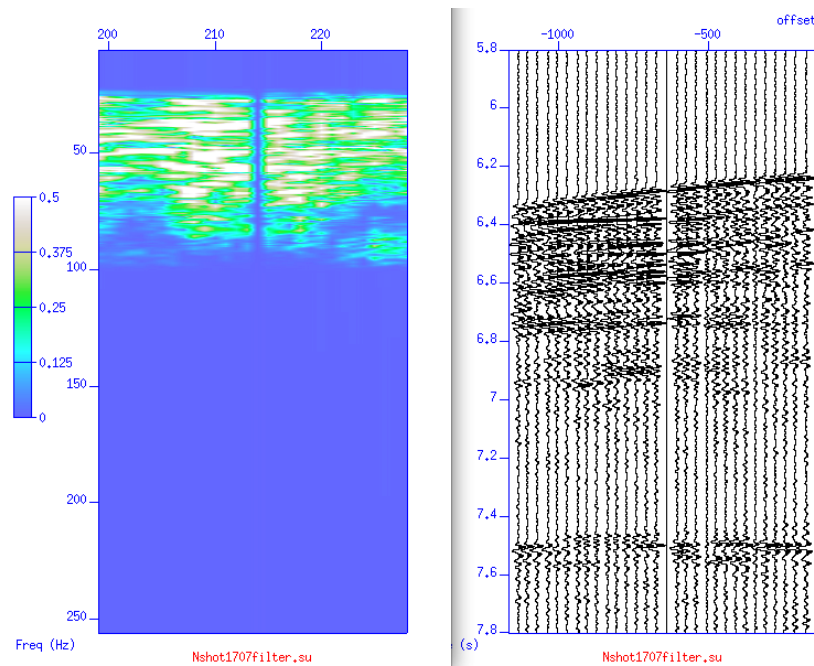


Figura 27 – Espectro de amplitudes do dado após a aplicação dos filtros.

Em seguida, foi feita a correção da divergência esférica utilizando o valor de $tpow$ mais otimizado após testes com valores diferentes descritos na tabela 2.

Tabela 2 – Variação em dB do dado antes e após a aplicação de ganho.

	Sem ganho	$tpow = 1.8$	$tpow = 2.2$
Máximo dB	-6.28	21.82	28.06
Mínimo dB	-65.97	-135.95	-143.9

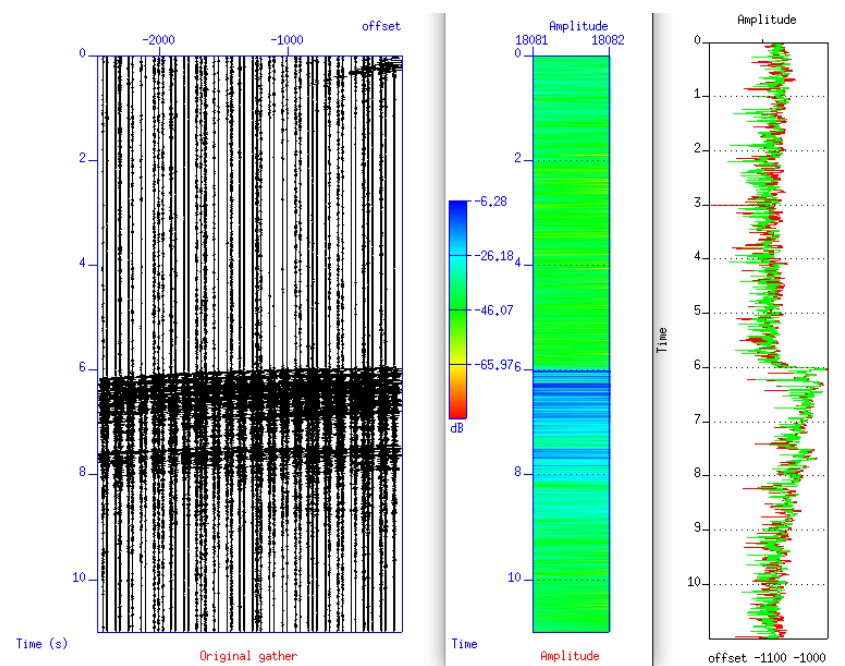


Figura 28 – Amplitude medida do dado original.

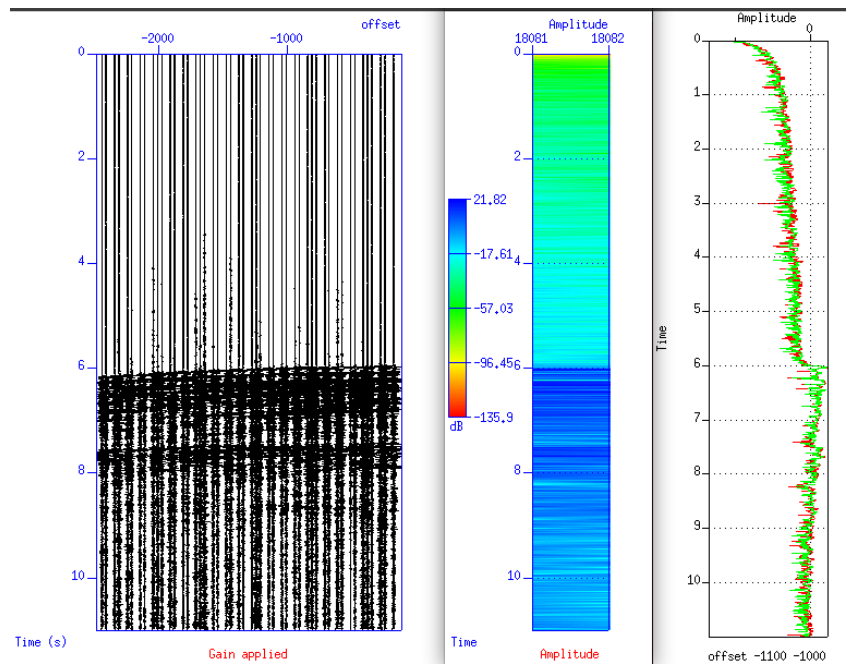


Figura 29 – Amplitude medida do dado após a aplicação do ganho $tpow$ de 1.8.

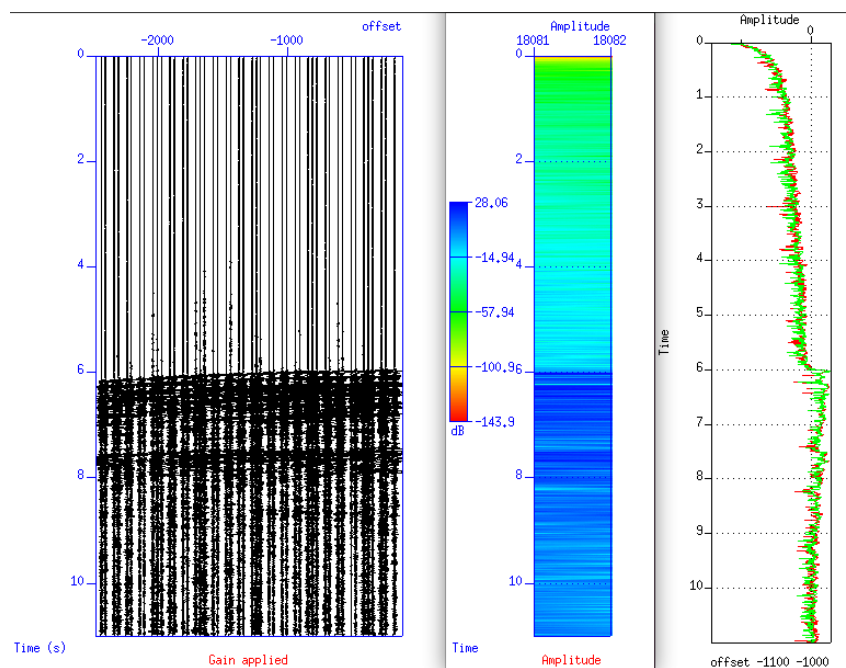


Figura 30 – Amplitude medida do dado após a aplicação do ganho $tpow$ de 2.2.

Ao comparar o dado original e o dado com o valor de $tpow$ escolhido, observa-se que houve um ganho de amplitude dos traços acentuando eventos reflexivos em áreas mais profundas (refletores localizados aproximadamente em 6.2 segundos e 7.7 segundos).

4.2.2 Análise de velocidade, NMO, Empilhamento e Migração

A próxima etapa do processamento foi a análise de velocidade para horizontalização dos refletores após a correção NMO. O CMP 1280, por exemplo, pode ser visto antes e após a aplicação de tal correção na figura 31.

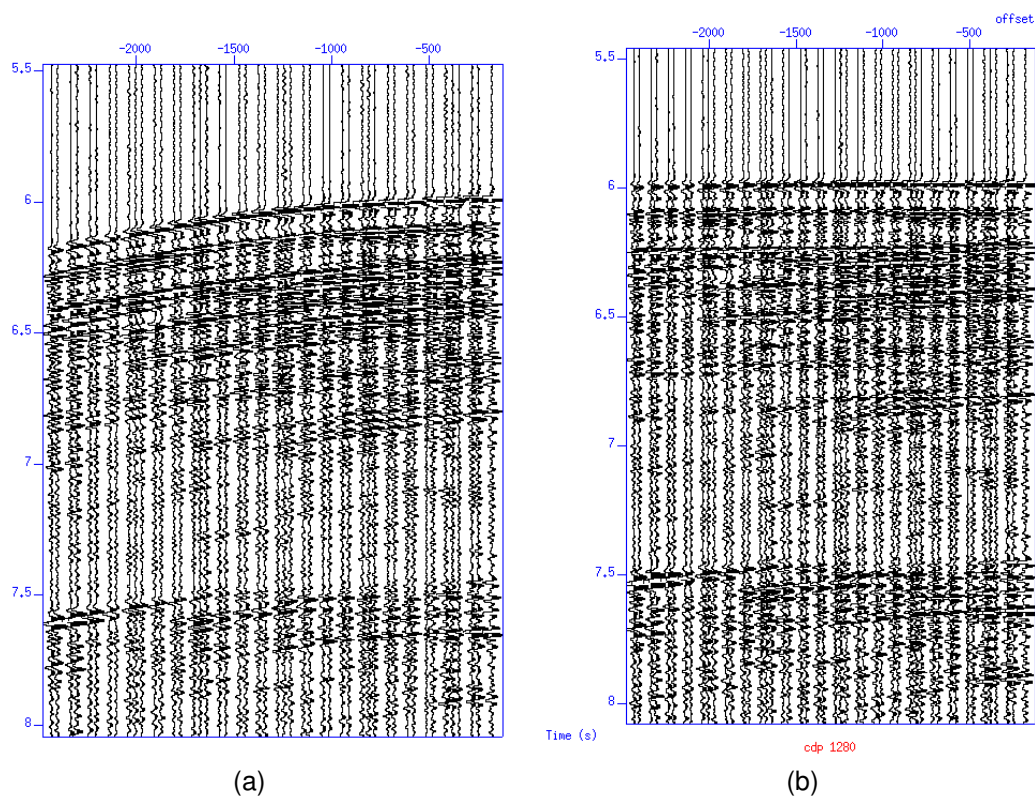


Figura 31 – Exemplo de correção NMO aplicado aos dados de Nankai. Em a) CMP antes e em b) após a correção. Fonte: o autor.

Uma vez tendo uma boa horizontalização dos refletores, o próximo passo foi o empilhamento para gerar uma seção pronta para ser migrada (Figura 32). Nota-se que o dado empilhado apresenta muitas difrações no fundo do mar e em todos os refletores subjacentes. Após 7.8 segundos não há reflexões significativas, por isso, janelamos a visualização entre 5.5 segundos e 8 segundos, como já afirmado anteriormente na Metodologia.

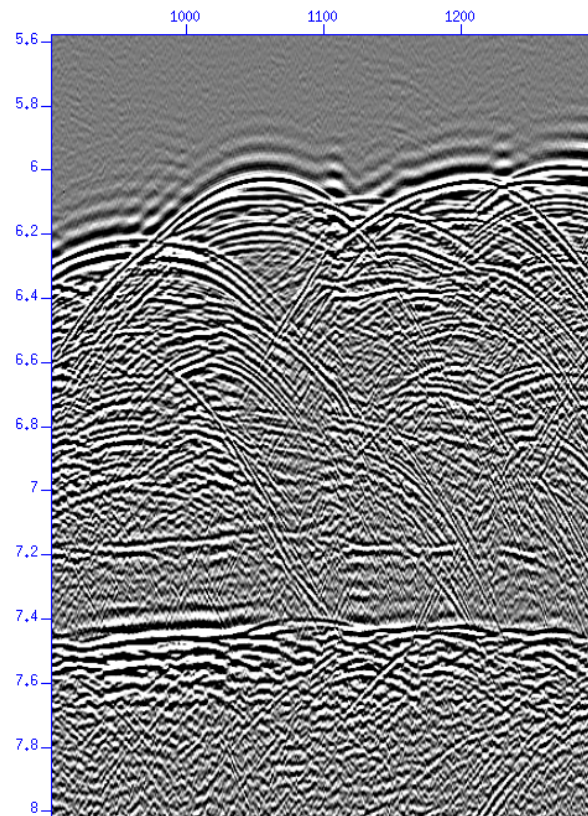


Figura 32 – Dado de Nankai após o empilhamento.

A próxima etapa do trabalho foi a migração realizada utilizando dois algoritmos diferentes apresentados na seção de Metodologia e seus resultados podem ser vistos nas Figuras 33 e 34. Para uma melhor análise, foram postos lado a lado o dado empilhado e a posterior migração realizada sobre ele.

Na figura 33, observa-se que, mesmo após a migração, a seção sísmica ainda apresenta muitas difrações mal colapsadas e vários artefatos foram gerados no fundo do mar.

Assim como a migração *Stolt* realizada anteriormente, a migração feita utilizando o algoritmo de *Gazdag* não solucionou todas as difrações que são vistas no fundo do mar. No entanto, os refletores de aproximadamente 7.5 segundos são melhor visualizados.

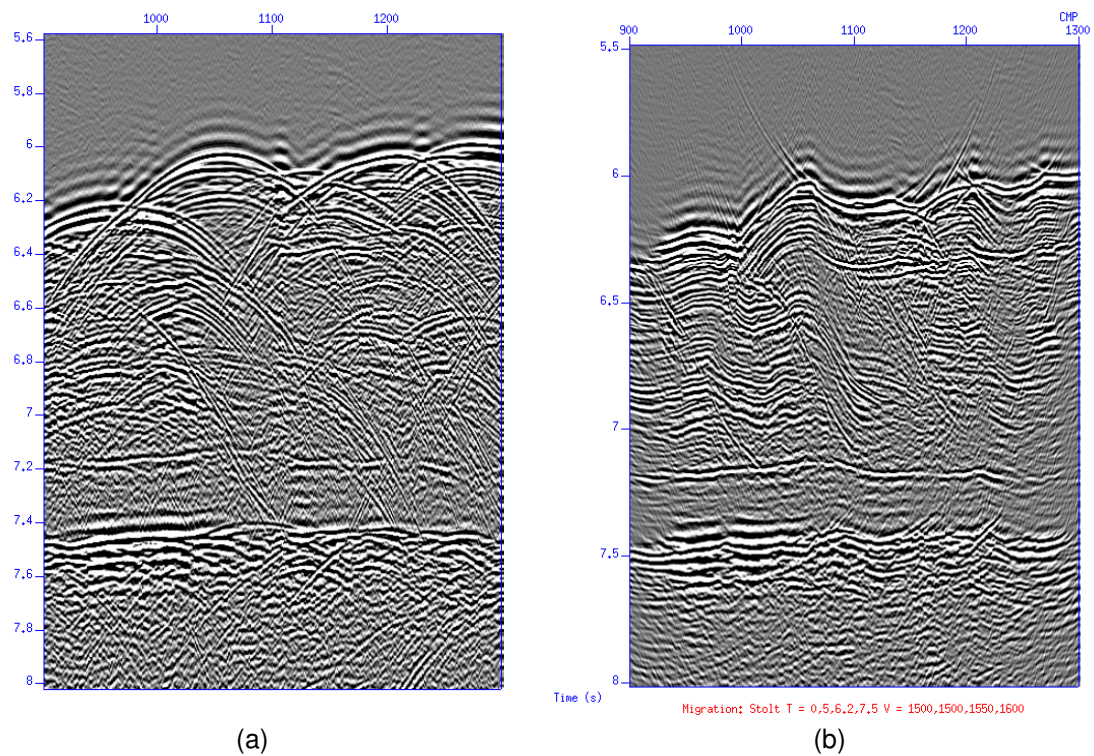


Figura 33 – Em a) temos o dado de Nankai empilhado e em b) temos o dado migrado com *Stolt*. Fonte: o autor.

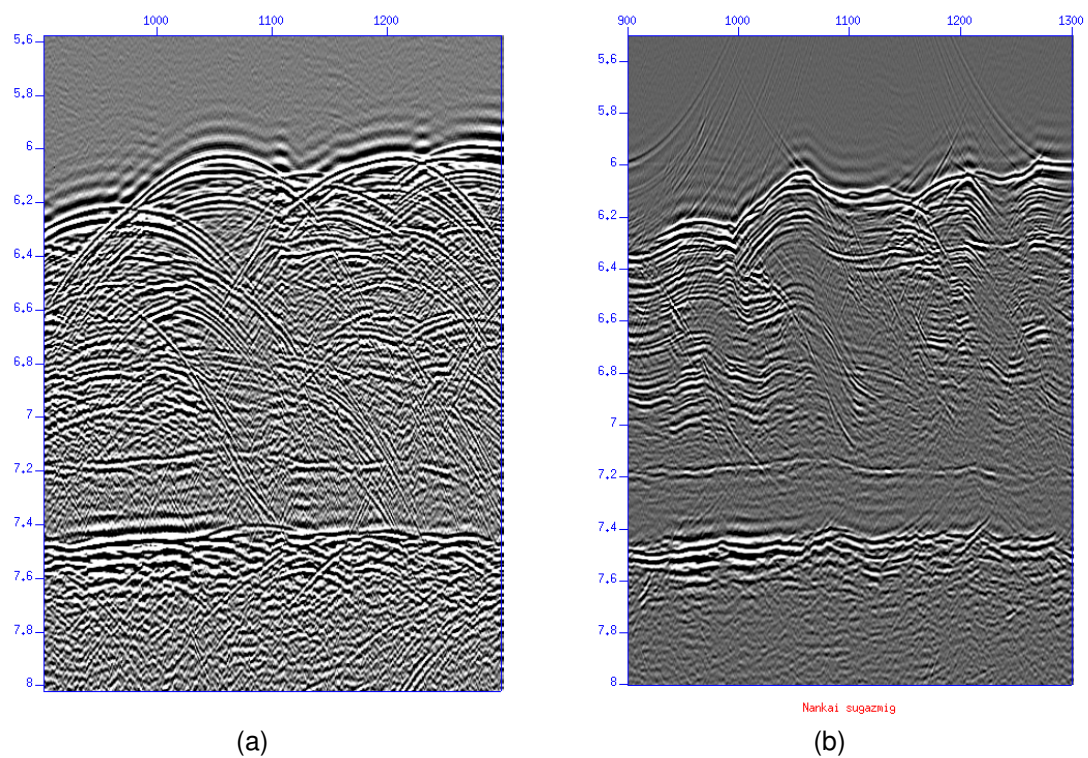


Figura 34 – Em a) temos o dado de Nankai empilhado e em b) temos o dado migrado com *Phase-Shift Gazdag*. Fonte: o autor.

5 Discussões

Os dois tipos de dados sísmicos processados neste trabalho apresentaram seus próprios desafios. Na etapa de aquisição através do modelo geológico criado, faz-se necessária uma nova configuração de raios emitidos pela fonte para que não haja mais o *gap* do refletor mais profundo entre os CMPs 310 e 390.

A interpolação espaço-temporal dos perfis de velocidade, por conta dos mergulhos acidentados das camadas, demandou cuidadosa observação para que os refletores não apresentassem sub ou sobrecorreção após a correção NMO.

Outro desafio foi a determinação do valor do parâmetro *vscale*, que demandou muitos testes para que fosse adequado para a velocidade modelada. A migração utilizada, por sua vez, tem muitas limitações. Por se tratar de uma migração 1D, o script possui uma única função de velocidade variável no tempo. Independente do mergulho dos refletores, foi aplicada em toda a seção sísmica. Como a função t-v foi projetada para se ajustar melhor na localização da difração do canal de areia (CMP 110), a seção foi mal-migrada em outras áreas. Com isso, não foi obtido o melhor imageamento para todo o dado. Para resolução destas limitações, o ideal seria a aplicação de uma migração variável tanto no tempo quanto no espaço.

O dado de Nankai, por sua parte, teve suas próprias peculiaridades. Como visto na seção de resultados, o filtro trapezoidal e o ganho aplicado para correção da divergência esférica melhoraram o espectro de amplitudes. No entanto, outras etapas do pré-processamento são limitadas no SU. À semelhança do dado sintético, a interpolação dos perfis de velocidade precisou ser observada com atenção, fazendo a análise de velocidade em CMPs próximos.

Diferentemente da migração do modelo gerado, em que foram utilizadas velocidades modeladas e um valor de *vscale* após testes, na migração aplicada ao dado real, não podíamos seguir tal metodologia por não conhecermos a verdadeira velocidade de uma camada específica. O uso dos algoritmos de Stolt e Gazdad gerou imagens semelhantes: a linha processada contém muitas difrações mal colapsadas, principalmente, na superfície do fundo do mar. Forel (2005) afirmou que por ser uma

geologia abaixo de 4500 metros de água, a análise de velocidade dos dados demanda certa paciência e conhecimento a priori das velocidades locais.

Na figura 35 temos a comparação entre o produto da migração obtida por este trabalho e o produto da migração realizada por Moore et al. (1990), contida no livro de Forel (2005). Não levando em conta os “sorrisos” de migração mutados acima do fundo do mar no dado original publicado por Moore et al. (1990) para melhorar sua aparência, o resultado obtido foi semelhante.

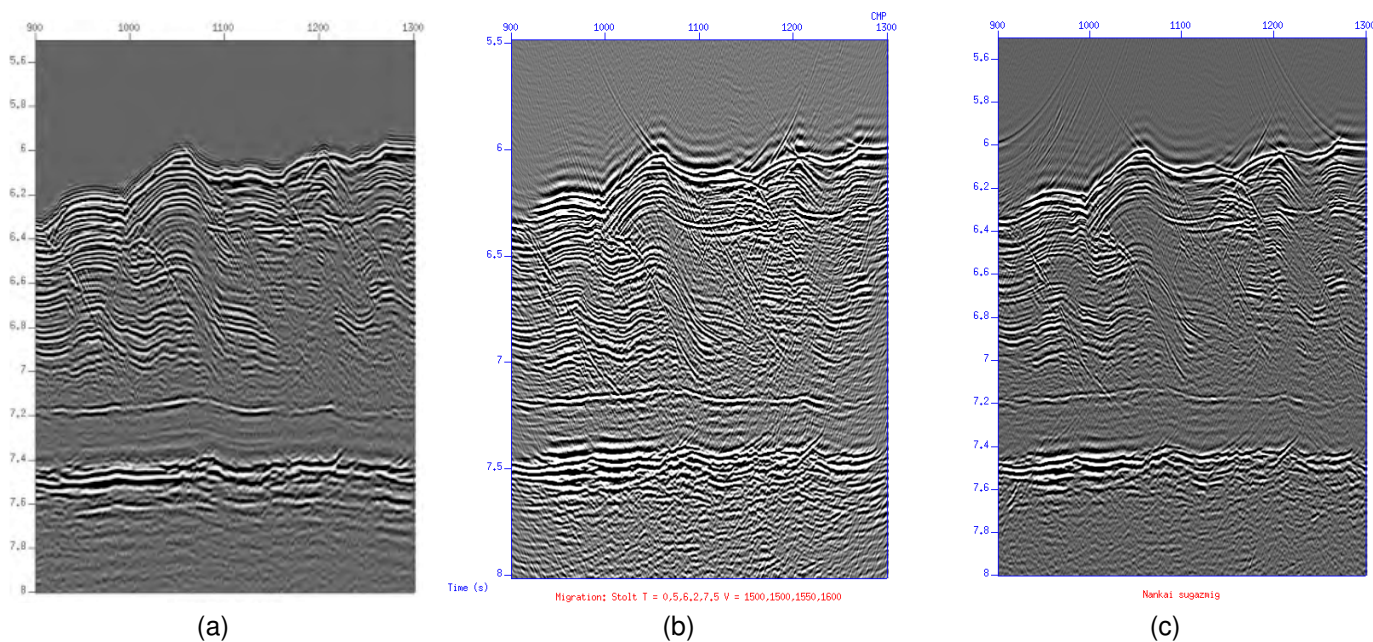


Figura 35 – Comparação entre os resultados obtidos por Moore et al. (1990) e este trabalho. Em a) temos a linha migrada por Moore, em b) Migrada com *Stolt* e em c) com *Phase-Shift Gazdad* neste trabalho. Fonte: o autor.

6 Conclusão

Uma vez que o tema deste trabalho foi o processamento sísmico utilizando o *Seismic Unix*, iremos dividir as conclusões em três grandes blocos: primeiro, falaremos sobre o programa SU; em um segundo e terceiro momentos, discorreremos sobre o dado sintético e o dado real de Nankai.

Referente ao SU, notou-se que a utilização do mesmo aliado com *Shell Scripts* possibilitou um contato inicial com tarefas básicas do processamento sísmico. Seria de suma importância que os alunos tivessem um contato com ele antes de mergulharem no uso de *softwares* poderosos do ramo de processamento. Todavia, a falta de conhecimento das ferramentas disponíveis do SU, e experiência em processamento sísmico limitam os resultados que podem ser alcançados.

Analisaremos nesta parte o dado modelado. Como relatado na seção de resultados e discussões, esse apresentou problemas de imageamento de refletores profundos, e a seção migrada com Stolt revelou as limitações da técnica. Para o primeiro obstáculo, faz-se necessária uma nova configuração de aquisição de dados sísmicos do modelo gerado. Dessa forma, espera-se que os raios provenientes da nossa fonte alcancem tais interfaces para otimização do imageamento. Concernente ao segundo impasse, o ideal é que seja feito um outro tipo de migração com algoritmos que lidem com variações espaciais e temporais.

Assim como o dado sintético, o dado real apresentou um resultado migrado que pode ser melhorado com uma migração variável no tempo e no espaço lidando assim com a variação lateral de velocidade. Também pode ser feita uma migração em profundidade e aplicação de filtros e ganhos podem melhorar a imagem também. Vale salientar que foram obtidos os mesmos resultados de Forel (2005) e Moore et al. (1990).

Referências

- ALVES, P. H. B. *Construção de modelos baseados na análise de velocidade de dados sísmicos da margem sudoeste da Inglaterra: uma comparação de dados empilhados*. Niterói: [s.n.], 2021. Disponível em: http://www.geofisica.uff.br/sites/default/files/projetofinal/2021_paulo_henrique_bastos_alves.pdf.
- ARAUJO, H. E. d. *Comparação dos métodos de migração Kirchhoff e migração reversa no tempo utilizando dados sintéticos do pré-sal*. Niterói: [s.n.], 2022. Disponível em: http://www.geofisica.uff.br/sites/default/files/projetofinal/2022_hugo_eduardo_de_araujo.pdf.
- BEASLEY, C. J. et al. Cascaded f-k migration: Removing the restrictions on depth-varying velocity. *Geophysics*, v. 53, p. 875–1011, 1988. ISSN 1942-2156.
- BILOTI, R. *Processamento Sísmico: Notas de aula*. [S.l.]: GGC / Unicamp, 2022. 180 p.
- DONDURUR, D. *Acquisition and Processing of Marine Seismic Data*. [S.l.]: Elsevier, 2018. 597 p.
- FOREL, D.; BENZ, T.; PENNINGTON, W. D. *Seismic Data Processing with Seismic Unix - A 2D Seismic Data Processing Primer*. Tulsa: Society of Exploration Geophysicists (SEG), 2005. 291 p.
- GAZDAG, J. Wave equation migration with the phase-shift method. *Geophysics*, v. 43, p. 1337–1556, 1978. ISSN 1942-2156.
- GAZDAG, J.; SGUAZZERO, P. Migration of seismic data by phase shift plus interpolation. *Geophysics*, v. 49, p. 105–203, 1984. ISSN 1942-2156.
- MOORE, G. F. et al. Structure of the nankai trough accretionary zone from multichannel seismic reflection data. *Journal of Geophysical Research*, v. 95, p. 8753–8765, 1990.
- ROBINSON, E. S.; CORUH, C. *Basic Exploration Geophysics*. 1. ed. [S.l.]: Wiley, 1988. 562 p.
- ROSA, A. L. R. *Análise do Sinal Sísmico*. 2. ed. Rio de Janeiro: Sociedade Brasileira de Geofísica (SBGf), 2018. 713 p.
- (SBGF, S. B. de G. Processamento de Dados Sísmicos: avanços em busca da qualidade. *Boletim SBGf*, Rio de Janeiro, n. 3, p. 13–19, 2012. ISSN 2177-9090.
- STOCKWELL, J.; COHEN, J. K. The new su user's manual. *Center for Wave Phenomena, Colorado School of Mines, Golden, USA*, v. 3, 2002.
- STOLT, R. H. Migration by fourier transform. *Geophysics*, v. 43, p. 23–48, 1978.
- TELFORD, W. M.; GELDART, L. P.; SHERIFF, R. E. *Applied Geophysics*. 2. ed. [S.l.]: Cambridge University Press, 1990. 760 p.

YILMAZ, Ö. *Seismic data analysis: Processing, inversion, and interpretation of seismic data*. Tulsa: Society of Exploration Geophysicists (SEG), 2001. v. 1. 2065 p.

Anexos

ANEXO A – Programa do Seismic Unix

A.1 Modelagem

```

#!/bin/sh
# File: model4.sh

# Set messages on
set -x

# Experiment Number
num=4

# Name output binary model file
modfile=model${num}.dat

# Name output encapsulated Postscript image file
psfile=model${num}.eps

# Remove previous .eps file
rm -f $psfile

trimodel xmin=-2 xmax=12.0 zmin=0 zmax=2.0 \
1 xedge=-2,12 \
  zedge=0,0 \
  sedge=0,0 \
2 xedge=-2.,0.00,2.0,4.0,6.0,8.00,10.0,12.0 \
  zedge=0.3,0.32,0.3,0.6,0.2,0.25,0.25,0.25 \
  sedge=0,0,0,0,0,0,0,0 \
3 xedge=-2.,0.0,2.0,4.0,6.0,7.0,7.5,8.00,8.50,9.0,9.50,10.,10.5,11.0,11.5,12. \
  zedge=0.8,0.8,1.0,1.3,0.5,0.5,0.7,0.9,1.02,1.15,1.25,1.25,1.2,1.15,1.08,1.0 \
  sedge=0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 \
4 xedge=-2.,0.0,2.0,4.0,6.0,7.00,8.0,9.0,10.0,11.0,12. \

```

```

    zedge=1.5,1.5,1.6,1.9,1.3,1.45,1.6,1.7,1.75,1.65,1.5 \
    sedge=0,0,0,0,0,0,0,0,0,0,0 \
5 xedge=1.80,2.00,2.20 \
    zedge=0.98,1.00,1.03 \
    sedge=0,0,0 \
6 xedge=1.80,2.00,2.20 \
    zedge=0.98,1.05,1.03 \
    sedge=0,0,0 \
7 xedge=7.0,7.5,8.0,8.5,9.0,9.50,10.0,11.0,12. \
    zedge=0.5,0.6,0.7,0.8,0.9,0.91,0.88,0.72,0.5 \
    sedge=0,0,0,0,0,0,0,0,0 \
8 xedge=7.0,7.5,8.0,8.50,9.00,9.50,10.0,10.5,11.0,11.5,12.0 \
    zedge=0.5,0.7,0.9,1.02,1.15,1.25,1.25,1.20,1.15,1.08,1.00 \
    sedge=0,0,0,0,0,0,0,0,0,0,0 \
9 xedge=-2,12 \
    zedge=2,2 \
    sedge=0,0 \
kedge=1,2,3,4,5,6,7,8,9 \
sfill=0.0,0.20,0,0,0.44,0,0 \
sfill=0.0,0.50,0,0,0.16,0,0 \
sfill=0.0,1.20,0,0,0.08,0,0 \
sfill=0.0,1.80,0,0,0.07,0,0 \
sfill=2.0,1.02,0,0,0.11,0,0 \
sfill=10.,1.00,0,0,0.09,0,0 > $modfile
##                x,z

# Create Encapsulated PostScript (EPS) image of model
spsplot < $modfile > $psfile \
    gedge=0.5 gtri=2.0 gmin=0.0 gmax=5.0 \
    title="Earth Model $num" \
    labelz="Depth (km)" labelx="Distance (km)" \
    wbox=5.25 hbox=0.75 dxnum=2.0 dznum=1.0

# Exit politely from shell
exit

```

A.2 Aquisição de linha 2D de dados sísmicos do modelo criado

```
#!/bin/sh
# File: acq4.sh
# Set messages on
##set -x

# Assign values to variables
nangle=201 fangle=-65 langle=65 nt=501 dt=0.004

# Model
num=4
echo " --Model number = $num"

# Name input model file
inmodel=model$num.dat

# Name output seismic file
outseis=seis${num}.su

# Remove survey file
rm -f survey${num}.txt
# Name survey file
survey=survey${num}.txt

#=====
# Create the seismic traces with "triseis"
#      i-loop = 200 source positions
#      j-loop = 60 geophone positions (split-spread)
#                      per shot position
```

```
#      k-loop = layers 2 through 8
#                               (do not shoot layers 1 and 9)

echo " --Begin looping over triseis."

i=0
while [ "$i" -ne "200" ]
do

    fs='bc -l <<-END
    $i * 0.05
    END'
    sx='bc -l <<-END
    $i * 50
    END'
    fldr='bc -l <<-END
    $i + 1
    END'

    j=0
    while [ "$j" -ne "60" ]
    do

        fg='bc -l <<-END
        $i * 0.05 + $j * 0.05
        END'
        gx='bc -l <<-END
        $i * 50 + $j * 50 - 1475
        END'
        offset='bc -l <<-END
        $j * 50 - 1475
        END'
        tracl='bc -l <<-END
        $i * 60 + $j + 1
        END'
```

```

    tracf='bc -l <<-END
    $j + 1
    END'

    echo " Sx=$sx Gx=$gx fldr=$fldr Offset=$offset tracl=$tracl\
    fs=$fs fg=$fg"
    echo " Sx=$sx Gx=$gx fldr=$fldr Offset=$offset tracl=$tracl\
    fs=$fs fg=$fg" >> $survey

    k=2
    while [ "$k" -ne "9" ]
    do

        triseis < $inmodel  xs=0,9.95 xg=-1.475,11.425 zs=0,0 zg=0,0 \
            nangle=$nangle fangle=$fangle langle=$langle \
            kreflect=$k krecord=1 fpeak=40 lscale=0.5 \
            ns=1 fs=$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,$tracl,$tracl,$fldr,$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
susum tempb temp5 > tempc
susum tempc temp6 > tempd
susum tempd temp7 > tempe
susum tempe temp8 > $outseis

# Remove temp files
echo " --Remove temp files."
rm -f temp*

# Report output file
echo " --Output file      ** $outseis **"

# Exit politely from shell script
echo " --Finished!"
exit
```

A.3 Visualização de shots

```
#!/bin/sh

# showshot.sh: Window one "field record" from file seis#.su
#
#               where # represents the model number.
# Outputs: wiggle image of the shot gather
#
#               .eps file of the shot gather
# Use:          showshot.sh    model    shot
# Example: showshot.sh        3         20

# Set messages on
set -x

# Window one "field record" to a temporary file
suwind < seis$1.su key=fldr min=$2 max=$2 > temp$1$2.su
```

```
# Make wiggle plot
suxwigb < temp$1$2.su title="SP # $2 [$1]" key=offset \
        label1=" Time (s)" label2="Offset (m)" \
        x2beg=-1500 x2end=1500 perc=99 &

# Create .eps image of a shot gather
supswigp < temp$1$2.su title="SP # $2 [$1]" key=offset \
        label1="Time (s)" label2="Offset (m)" \
        x2beg=-1500 x2end=1500 perc=99 > shot$1$2.eps &

# Remove temporary gather
rm -f temp$1$2.su

# Exit politely from shell
exit
```

A.4 Controle de qualidade dos pares tnmo-vnmo escolhidos

```
#!/bin/sh
# File: tvQC.sh
#           Test tnmo-vnmo series
#   Input: The same cdp-tnmo-vnmo file that
#           would be input to sunmo
# Output: Screen messages
# Use: tvQC.sh

# Set debugging on
##set -x

tvnmoqc mode=1 \
cdp=60,80,100,120,140,160,180,200,220,\
```



```
240,260,280,300,320,340,360,380,400 \
tnmo=0.0158311,0.390501,0.686016,1.06069,1.99472 \
vnmo=1511.85,1511.85,2640.45,2922.6,3249.3 \
tnmo=0.0158311,0.353562,0.701847,1.0343,1.99472 \
vnmo=1482.15,1511.85,2625.6,2774.1,2982 \
tnmo=0.0158311,0.364116,0.765172,0.875989,1.08707,1.98945 \
vnmo=1511.85,1526.7,2595.9,2714.7,2625.6,2699.85 \
tnmo=0.0211082,0.437995,0.844327,0.949868,1.16623,1.98945 \
vnmo=1497,1526.7,2551.35,2432.55,2685,2685 \
tnmo=0.0105541,0.569921,1.00792,1.30343,1.99472 \
vnmo=1497,1541.55,2595.9,2848.35,3011.7 \
tnmo=0.0158311,0.686016,1.12401,1.04591,1.98945 \
vnmo=1482.15,1541.55,2729.55,3160.2,3293.85 \
tnmo=0.0158311,0.78628,1.19789,1.5409,1.99472 \
vnmo=1482.15,1511.85,3011.7,3694.8,3798.75 \
tnmo=0.0105541,0.765172,1.1029,1.45119,1.98945 \
vnmo=1497,1511.85,2937.45,3665.1,3858.15 \
tnmo=0.0158311,0.627968,0.907652,1.25594,1.99472 \
vnmo=1497,1556.4,2595.9,3056.25,3353.25 \
tnmo=0.0158311,0.469657,0.686016,1.07124,1.98417 \
vnmo=1482.15,1556.4,2521.65,2818.65,3026.55 \
tnmo=0.0158311,0.3219,0.501319,0.923483,1.98417 \
vnmo=1482.15,1541.55,2462.25,2670.15,2774.1 \
tnmo=0.0158311,0.226913,0.395778,0.870712,1.98417 \
vnmo=1482.15,1511.85,2521.65,2699.85,2848.35 \
tnmo=0.0105541,0.23219,0.390501,0.918206,1.99472 \
vnmo=1482.15,1526.7,2566.2,2878.05,3234.45 \
tnmo=0.0158311,0.258575,0.46438,1.99472 \
vnmo=1497,1526.7,2699.85,3368.1 \
tnmo=0.0158311,0.311346,0.538259,0.612137,1.98945 \
vnmo=1482.15,1526.7,2759.25,2952.3,3442.35 \
tnmo=0.0211082,0.353562,0.622691,0.744063,1.99472 \
vnmo=1467.3,1511.85,2774.1,3056.25,3457.2 \
tnmo=0.0211082,0.369393,0.691293,0.812665,1.99472 \
vnmo=1482.15,1511.85,2878.05,2982,3279 \
```

```
tnmo=0.0211082,0.348285,0.728232,0.881266,1.20844,1.98945 \  
vnmo=1482.15,1511.85,2833.5,3026.55,3145.35,3323.55  
  
# Exit politely from shell  
exit
```

A.5 Correção NMO

```
#!/bin/sh  
# File: nmo4.sh  
#           Apply NMO (flatten) 2-D line of CMPs  
# Input (1): 2-D line of CMPs  
# Output (1): NMO-corrected 2-D line of CMPs  
# Use: nmo4.sh  
#  
# NMO correction is interpolated between named CMPs.  
  
# Set debugging on  
set -x  
  
# Name data sets  
indata=cmp4.su  
outdata=nmo4.su  
  
sunmo < $indata \  
cdp=60,80,100,120,140,160,180,200,220,\  
240,260,280,300,320,340,360,380,400 \  
tnmo=0.0158311,0.390501,0.686016,1.06069,1.99472 \  
vnmo=1511.85,1511.85,2640.45,2922.6,3249.3 \  
tnmo=0.0158311,0.353562,0.701847,1.0343,1.99472 \  
vnmo=1482.15,1511.85,2625.6,2774.1,2982 \  
tnmo=0.0158311,0.364116,0.765172,0.875989,1.08707,1.98945 \  
vnmo=1511.85,1526.7,2595.9,2714.7,2625.6,2699.85 \  

```

```
tnmo=0.0211082,0.437995,0.844327,0.949868,1.16623,1.98945 \  
vnmo=1497,1526.7,2551.35,2432.55,2685,2685 \  
tnmo=0.0105541,0.569921,1.00792,1.30343,1.99472 \  
vnmo=1497,1541.55,2595.9,2848.35,3011.7 \  
tnmo=0.0158311,0.686016,1.12401,1.44591,1.98945 \  
vnmo=1482.15,1541.55,2729.55,3160.2,3293.85 \  
tnmo=0.0158311,0.78628,1.19789,1.5409,1.99472 \  
vnmo=1482.15,1511.85,3011.7,3694.8,3798.75 \  
tnmo=0.0105541,0.765172,1.1029,1.45119,1.98945 \  
vnmo=1497,1511.85,2937.45,3665.1,3858.15 \  
tnmo=0.0158311,0.627968,0.907652,1.25594,1.99472 \  
vnmo=1497,1556.4,2595.9,3056.25,3353.25 \  
tnmo=0.0158311,0.469657,0.686016,1.07124,1.98417 \  
vnmo=1482.15,1556.4,2521.65,2818.65,3026.55 \  
tnmo=0.0158311,0.3219,0.501319,0.923483,1.98417 \  
vnmo=1482.15,1541.55,2462.25,2670.15,2774.1 \  
tnmo=0.0158311,0.226913,0.395778,0.870712,1.98417 \  
vnmo=1482.15,1511.85,2521.65,2699.85,2848.35 \  
tnmo=0.0105541,0.23219,0.390501,0.918206,1.99472 \  
vnmo=1482.15,1526.7,2566.2,2878.05,3234.45 \  
tnmo=0.0158311,0.258575,0.46438,1.99472 \  
vnmo=1497,1526.7,2699.85,3368.1 \  
tnmo=0.0158311,0.311346,0.538259,0.612137,1.98945 \  
vnmo=1482.15,1526.7,2759.25,2952.3,3442.35 \  
tnmo=0.0211082,0.353562,0.622691,0.744063,1.99472 \  
vnmo=1467.3,1511.85,2774.1,3056.25,3457.2 \  
tnmo=0.0211082,0.369393,0.691293,0.812665,1.99472 \  
vnmo=1482.15,1511.85,2878.05,2982,3279 \  
tnmo=0.0211082,0.348285,0.728232,0.881266,1.20844,1.98945 \  
vnmo=1482.15,1511.85,2833.5,3026.55,3145.35,3323.55 \  
    > $outdata  
  
# Exit politely from shell  
exit
```

A.6 Visualização dos dados após a correção NMO

```
#!/bin/sh
# File: iview.sh
#           View seismic gathers from a 2-D line

# Set messages on
##set -x

#=====
# USER AREA -- SUPPLY VALUES

# input seismic data
indata=nmo4a.su # SU format

# plot choices
myperc=98      # perc value for plot
plottype=0     # 0 = wiggle plot, 1 = image plot
Wplot=400      # Width of plot (pixels)
Hplot=500      # Height of plot (pixels)

# processing variables
sortkey=cdp    # sort key (usually fldr or cdp)
firsts=10      # first sort (fldr or cdp) value
lasts=458      # last sort (fldr or cdp) value
increment=10   # sort key increment
tracekey=offset # trace label key

#=====
# file descriptions

# tmp1 = binary temp file for input gather
```

```
#-----

echo " "
echo " *** VIEWER ***"
echo " "
echo "          INPUT: $indata"

#-----

# Remove old temporary file
rm -f tmp*

#-----

# BEGIN LOOP
#-----

i=$firsts
while [ $i -le $lasts ]
do

    echo " "
    echo "Reading gather $i of $indata"
    echo "First gather = $firsts  Last gather = $lasts"

    suwind < $indata key=$sortkey min=$i max=$i > tmp1

    if [ $plottype -eq 0 ] ; then
        suxwigg < tmp1 xbox=10 ybox=10 wbox=$Wplot hbox=$Hplot \
            title="$sortkey $i" \
            label1=" Time (s)" label2="$tracekey" key=$tracekey \
            perc=$myperc verbose=0 &
    else
        suximage < tmp1 xbox=10 ybox=10 wbox=$Wplot hbox=$Hplot \
            title="$sortkey $i" \
            label1=" Time (s)" \
```

```
perc=$myperc verbose=0 &

fi

echo " "
echo "Press Return or Enter to see next gather"
echo " Or enter \"x\" to exit"
> /dev/tty
read response

case $response in
  [xX])
    zap xwigb > tmp1
    zap ximage > tmp1
    i='expr $lasts + 1'
    ;;
  *)
    zap xwigb > tmp1
    zap ximage > tmp1
    i='expr $i + $increment'
    ;;
esac

done

#-----
# END LOOP
#-----

# Exit
echo " "
echo " ==> Closing iview"
pause
rm -f tmp*
exit
```

A.7 Criação de CVP's com migração Stolt

```
#!/bin/sh
# File: migcvp.sh
#       Create one panel for each migration velocity
#       Each panel has the same "fldr" value
#       The migration velocity is in key "offset"
#       Total number of panels is in key "nvs"

# Set messages on
##set -x

#=====
# USER AREA -- SUPPLY VALUES
#-----

# Seismic files
indata=stack4.su      # SU format
outdata=migcvp.su     # migration Constant Velocity Panels

# Migration variables
cdpmin=80             # Start CDP value
cdpmax=150            # End CDP value
dxcdp=70.71          # distance between adjacent CDP bins (m)
smig=1.0              # stretch factor (0.6 typical if vrms increasing)
                        # [the "W" factor] (Default=1.0)
vscale=1.9            # scale factor to apply to velocities (Default=1.0)
lstaper=20            # length of side tapers (traces) (Default=0)
lbtaper=100           # length of bottom taper (samples) (Default=0)

# Velocity panel variables
firstv=1200           # first velocity value
lastv=5000            # last velocity value
increment=200         # velocity increment
```

```

numVtest=100      # use to limit number of velocity panels
                  # otherwise, use very large value (100)

#=====

# Compute number of velocity panels

numV='bc -l << -END
( ( $lastv - $firstv ) / $increment ) + 1
END'

if [ $numVtest -lt $numV ] ; then
    numV=$numVtest
fi

#-----

# FILE DESCRIPTIONS
# tmp1 = binary temp file of input data

#-----

cp $indata tmp1
migV=$firstv
echo " "

#-----

# Loop through Migration Constant Velocity Panels
#      Each panel has the same "fldr" value
#      Panel migration velocity is in key "offset"
#      Total number of panels (numV) is in key "nvs"
#-----

i=1
while [ $i -le $numV ]

```



```

do

    echo "          iteration number = $i          Velocity = $migV"

    suwind < tmp1 key=cdp min=$cdpmin max=$cdpmax |
        sushw key=fldr a=$i |
        sushw key=offset a=$migV |
        sushw key=nvs a=$numV |

    sustolt cdpmin=$cdpmin cdpmax=$cdpmax dxcdp=$dxcdp \
        tmig=0 vmig=$migV smig=$smig vscale=$vscale \
        lstaper=$lstaper lbtaper=$lbtaper \
        >> $outdata

    i='expr $i + 1'
    migV='expr $migV + $increment'

done

#-----
# Remove files and exit
#-----

echo " "
echo "          Output file = $outdata"
echo " "

rm -f tmp*
exit

```

A.8 Visualização dos CVP's

```
#!/bin/sh
```

```
# File: iviewcvp.sh
#      View seismic panels from a migration line of
#      Constant Velocity Panels -- key fldr
#      Migration velocity of each panel is in key offset
#      Total number of panels is in key nvs

# Set messages on
##set -x

#=====
# USER AREA -- SUPPLY VALUES
#-----

# Input seismic data
indata=migcvp_teste.su # SU format

# Plot choices
myperc=98      # perc value for plot
plottype=1     # 0 = wiggle plot, 1 = image plot
Wplot=300      # Width of plot (pixels)
Hplot=500      # Height of plot (pixels)

#=====
# Processing variables
sortkey=fldr   # sort key [Do Not Change]
firsts=1       # first sort (fldr) value:
# lasts       # last sort (fldr) value: key nvs
increment=1    # sort key (fldr) increment
tracekey=cdp   # trace label key

#-----

# file descriptions

# tmp1 = binary file of input panel
# tmp2 = ASCII file to reduce "zap" screen messages
```

```
#-----

echo " "
echo " *** MIGRATION CONSTANT VELOCITY PANEL VIEWER ***"
echo " "
echo "      INPUT: $indata"

#-----

# Remove old temporary files
rm -f tmp*

#-----

# Get first trace - total number of panels in key nvs

lasts='sugethw nvs < $indata | sed 1q | sed 's/.*nvs=/'''

#-----

# BEGIN PANEL LOOP
#-----

i=$firsts
while [ $i -le $lasts ]
do

    suwind < $indata key=$sortkey min=$i max=$i > tmp1

# Retrieve migration velocity from key offset of first trace
migV='sugethw offset < tmp1 | sed 1q | sed 's/.*offset=/'''
migVel='expr $migV' # Ensure migration velocity is numeric

echo " "
echo "Reading panel $i of $indata, Velocity $migVel"
echo "First panel = $firsts      Last panel = $lasts"
```

```

if [ $plotttype -eq 0 ] ; then
    suxwigb < tmp1 xbox=10 ybox=10 wbox=$Wplot hbox=$Hplot \
        title="$sortkey $i    Velocity $migVel" \
        label1=" Time (s)" label2="$tracekey" key=$tracekey \
        perc=$myperc verbose=0 &
else
    suximage < tmp1 xbox=10 ybox=10 wbox=$Wplot hbox=$Hplot \
        title="$sortkey $i    Velocity $migVel" \
        label1=" Time (s)" label2="$tracekey" key=$tracekey \
        perc=$myperc verbose=0 &
fi

echo " "
echo "Press Return or Enter to see next panel"
echo " Or enter \"x\" to exit"
> /dev/tty
read response

case $response in
    [xX])
        zap xwigb > tmp2
        zap ximage > tmp2
        i='expr $lasts + 1'
        ;;
    *)
        zap xwigb > tmp2
        zap ximage > tmp2
        i='expr $i + $increment'
        ;;
esac

done

#-----
# END PANEL LOOP

```

```
#-----

# Exit
echo " "
echo "      ==> Closing iview"
pause
rm -f tmp*
exit
```

A.9 Migração Stolt

```
#!/bin/sh
# File: migStolt.sh
#      Stolt migration stacked data
#  Input: stack data
#  Output: plot of migrated data
#      Use: migStolt.sh
# Example: migStolt.sh

# Velocities are stacking (Vrms)
# Here, we use the false assumption that stacking
#      and migrating velocities do not change laterally.

# smig = stretch factor (0.6 typical if vrms increasing)
# vscale = scale factor to apply to velocities

# Set messages on
set -x

time=0.00,0.70,0.75,0.80,0.85,0.90,2.00
vels=1500,1500,2500,3000,2800,3100,3500

sustolt < stack4.su \
```

```
cdpmin=1 cdpmax=458 dxcdp=70.71 \  
tmig=$time vmig=$vels \  
smig=0.6 vscale=1.9 lstaper=20 lbtaper=100 |  
  
suximage xbox=10 ybox=10 wbox=800 hbox=400 \  
title="Migration: Stolt T = $time V = $vels" \  
label1=" Time (s)" label2="CMP" key=cdp \  
perc=99 verbose=0 &  
  
# Exit politely from shell  
exit
```