

Karine Ribeiro Pereira

Absorção sísmica e métodos de correção

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-graduação em Matemática da PUC-Rio como requisito parcial para obtenção do grau de Mestre em Matemática.

Orientador : Prof. Thomas Lewiner
Co-Orientador: Prof. Osvaldo de Oliveira Duarte

Rio de Janeiro
Março 2015



Karine Ribeiro Pereira

Absorção sísmica e métodos de correção

Dissertação apresentada ao Programa de Pós-graduação em Matemática da PUC-Rio como requisito parcial para obtenção do grau de Mestre em Matemática. Aprovada pela Comissão Examinadora abaixo assinada.

Thomas Lewiner

Orientador

Pontifícia Universidade Católica do Rio de Janeiro

Osvaldo de Oliveira Duarte

Co-Orientador

Universidade Petrobras

Prof. Luiz Antônio Canelas Palermo

CENPES, Petrobras

Prof. Pedro Mario Cruz e Silva

Pontifícia Universidade Católica do Rio de Janeiro

Prof. Carlos Tomei

Pontifícia Universidade Católica do Rio de Janeiro

Prof. Alex Lúcio Ribeiro de Castro

Pontifícia Universidade Católica do Rio de Janeiro

José Eugenio Leal

Coordenador do Centro Técnico Científico

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 25 de março de 2015

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Karine Ribeiro Pereira

Graduada bacharel em matemática pela Universidade Federal Fluminense.

Ficha Catalográfica

Ribeiro, Karine

Absorção sísmica e métodos de correção / Karine Ribeiro Pereira; orientador: Thomas Lewiner; co-orientador: Osvaldo de Oliveira Duarte. — Rio de Janeiro : PUC-Rio, Departamento de Matemática, 2015.

v., 67 f: il. ; 29,7 cm

1. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Matemática.

Inclui referências bibliográficas.

1. Matemática – Dissertação 2. Processamento de sinais; 3. Transformada de Fourier; 4. Absorção sísmica; 5. Filtros; 6. Convolução; 7. Processamento sísmico. I. Lewiner, Thomas. II. Duarte, Osvaldo. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Matemática. IV. Título.

CDD: 510

À minha família.

Agradecimentos

Ao final desta jornada de dois anos, gostaria de agradecer às instituições e às pessoas que contribuíram para a conclusão deste trabalho. Agradeço à Vice-Reitoria para Assuntos Acadêmicos da PUC-Rio pela concessão da bolsa de isenção de taxas escolares e à Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP) pela liberação para cursar o mestrado e pela disponibilização dos dados sísmicos solicitados.

Agradeço também ao Professor Thomas Lewiner, que aceitou orientar este trabalho, ensinou-me muitas coisas e sempre foi muito paciente e disponível. Ao professor Osvaldo Duarte, que sugeriu o tema da dissertação e fez grandes contribuições.

Aos membros da banca, Professor Alex Lúcio Ribeiro de Castro, Professor Carlos Tomei, Luiz Antônio Canelas Palermo e Pedro Mário Cruz e Silva.

Gostaria de agradecer também aos meus pais, Ivone e Itamar, à minha irmã Lana, aos meus avós e aos meus amigos, em especial Renata da Silva, Amanda Carneiro, Luiz Carlos de Souza Junior, Mina Saito, André Sena, Luciana Araújo, Danielle Leal, Evandro dos Santos, Rafael Simon e Pedro Kenzo, que tiveram paciência e compreenderam as minhas ausências.

Agradeço ainda a todas as pessoas que contribuíram de alguma forma para este trabalho: Pedro Cabrera, que me ajudou a entender o processamento sísmico, Raphael Victor Aleixo Vasconcelos e Renato Lopes Silveira, que deram importantes dicas para o processamento dos dados, Sérgio Henrique Almeida, Cláudio Jorge Martins de Souza e Gustavo de Freitas Tinoco por disponibilizar os dados sísmicos, Gustavo Amaral, que me deu uma aula sobre filtros recursivos, Raphael Moura, meu ex-chefe na ANP e grande incentivador deste mestrado, e Nivaldo Medeiros, que deu muitas dicas e sugeriu leituras que ajudaram a compor este trabalho.

Por fim, agradeço especialmente a Marcos Cintra, que foi muito importante durante essa caminhada.

Resumo

Ribeiro, Karine; Lewiner, Thomas; Duarte, Osvaldo. **Absorção sísmica e métodos de correção**. Rio de Janeiro, 2015. 67p. Dissertação de Mestrado — Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho tem como objetivo analisar o problema das perdas por absorção dos dados de reflexão sísmica, bem como testar três métodos disponíveis na literatura para sua correção. Utilizamos a modelagem da absorção apresentada por Romanelli Rosa, com a noção de frequência instantânea, e analisamos os seguintes métodos de correção: a Compensação Q, o método de Varela *et al.* e o método de Duarte, que é um filtro recursivo. Observamos que o método de Duarte é computacionalmente mais rápido que os demais. Ainda assim, podemos utilizar a Transformada de Fourier para torná-lo mais rápido nos casos em que a recursão é interrompida em uma etapa M , menor que o número de amostras N do dado sísmico e maior que $\ln N$. Por fim, testamos o desempenho dos métodos em uma linha de reflexão sísmica marítima da Bacia de Sergipe-Alagoas, fornecida pela Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP). A linha foi reprocessada, com a correção das perdas por absorção aplicada antes do empilhamento, para cada método estudado. Para comparar os resultados, o dado também foi processado sem correção da absorção. Verificamos que houve um aumento da resolução das camadas geológicas de subsuperfície em todos os métodos testados em comparação com o dado sem correção, porém o método de Duarte mostrou-se mais rápido que os demais.

Palavras-chave

Processamento de sinais; Transformada de Fourier; Absorção sísmica; Filtros; Convolução; Processamento sísmico.

Abstract

Ribeiro, Karine; Lewiner, Thomas; Duarte, Osvaldo. **Seismic absorption and correction methods**. Rio de Janeiro, 2015. 67p. Dissertação de Mestrado — Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

This work aims at analyzing the problem of losses by absorption in seismic reflection data and test three correction methods available in the literature. We use the modeling of the absorption presented by Romanelli Rosa, with the concept of instantaneous frequency, and analyzed the following correction methods: Q compensation, Varela's method and Duarte's method, which is a recursive filter. We note that Duarte's method is computationally faster than the others. However, we can use the Fourier Transform to make it faster in cases where the recursion is interrupted at a step M , smaller than the number of samples N in the seismic data and greater than $\ln N$. Finally, we test the performance of the methods in a marine seismic line in Sergipe-Alagoas Basin, provided by the Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP). The line was reprocessed, with the correction of the losses by absorption applied before stacking, for each method studied. In order to compare the results, the data was also processed without correction of absorption. We observe an increase in the resolution of the geological subsurface in all methods tested in comparison with the data without correction. We also observe the computational advantage of Duarte's method.

Keywords

Signal processing; Fourier Transform; Seismic absorption; Filters; Convolution; Seismic processing.

Sumário

1	Introdução	10
2	Principais conceitos	13
2.1	Sinais	13
2.2	Sistemas	15
2.3	Transformada de Laplace	17
2.4	Transformada Z	18
2.5	Transformada de Fourier de sinais de tempo contínuo	19
2.6	Transformada de Fourier de sinais de tempo discreto	20
2.7	<i>Fast Fourier Transform</i> - FFT	22
2.8	Filtros	24
2.9	Decibéis	25
2.10	Frequência de Nyquist	26
2.11	Absorção sísmica	26
3	Modelagem e correção da absorção sísmica	27
3.1	A modelagem da absorção	27
3.2	Compensação Q	32
3.3	O método de Varela	34
4	O método de Duarte	38
4.1	Apresentação do método	38
4.2	Comparação com o método de Varela	42
4.3	Melhorando o desempenho do algoritmo	44
5	Experimentos	48
6	Conclusão	54
	Referências Bibliográficas	55
A	Apêndice	57
A.1	Mudança de formato	57
A.2	Visualização do dado	57
A.3	Correção de amplitude	58
A.4	Filtragem	59
A.5	Deconvolução	59
A.6	Balanceamento de amplitudes	60
A.7	Ordenar o dado por CMP	60
A.8	Análise de velocidade e correção NMO	62
A.9	Empilhamento	62
A.10	Migração	66

*Não há lugar para a sabedoria onde não há
paciência.*

Santo Agostinho.

1

Introdução

A sismica de reflexão é um método muito utilizado pela indústria do petróleo com objetivo de obter informações sobre as estruturas e a composição da subsuperfície da Terra, a fim de identificar áreas com maior probabilidade de existência de óleo e gás. O método consiste em três etapas básicas: aquisição, processamento e interpretação.

Na aquisição sísmica, um impulso é gerado por uma fonte artificial, que pode ser um canhão de ar ou explosivos. Esse pulso se propaga e, em cada interface gera uma reflexão que pode ser captada na superfície por receptores especiais (hidrofones ou geofones) e registrada por um sismógrafo.

Nas aquisições marítimas, um navio-sísmico puxa canhões de ar e cabos flutuadores nos quais estão presos os hidrofones, conforme a figura 1.1. Nos levantamentos terrestres, são utilizados explosivos para gerar a onda e os geofones são fixados na superfície, ao longo de uma linha. A figura 1.2 representa uma aquisição sísmica terrestre.

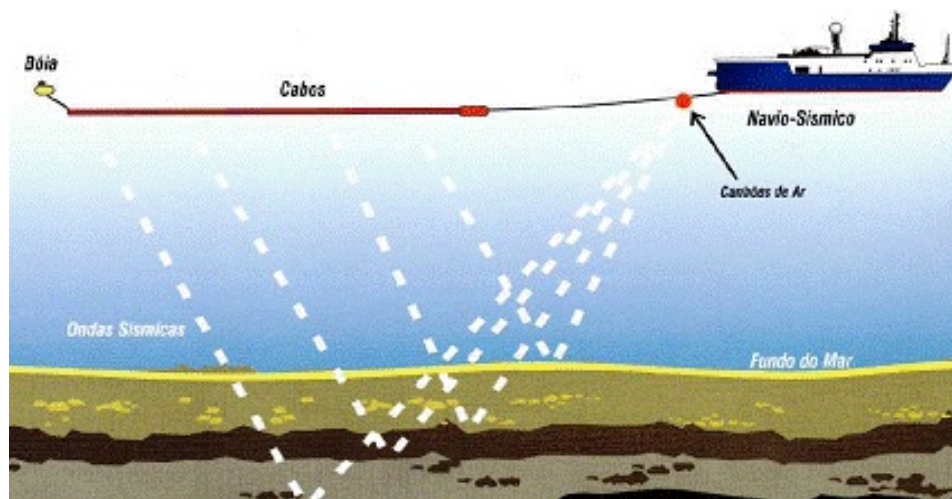


Figura 1.1: Aquisição sísmica no mar. Fonte: <http://www1.oceanica.ufrj.br>

Uma vez que os dados já foram registrados, é preciso processá-los para gerar uma imagem na qual é possível identificar as camadas de rochas e as estruturas geológicas de subsuperfície. No processamento, são realizadas diversas etapas, tais como filtragem, deconvolução, análise de velocidade,

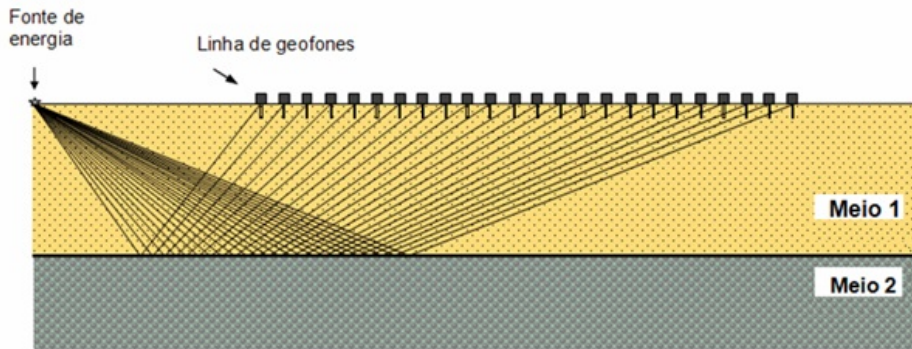


Figura 1.2: Aquisição sísmica terrestre. Fonte: tecnopetrobr.blogspot.com.br/-2008/11/ssmica-de-reflexo.html

empilhamento, dentre outras [13, 7]. Não existe um passo a passo único para processar um dado sísmico, mas algumas etapas estão geralmente presentes.

Por fim, a imagem obtida com o processamento dos dados é interpretada [13, 9], revelando as estruturas geológicas e, juntamente com informações obtidas com outros métodos de prospecção, estima-se o modelo geológico da subsuperfície. A partir dessa análise, são identificadas as regiões onde pode haver óleo ou gás.

Um dos problemas do método de reflexão sísmica é que, durante a propagação da onda, parte da energia é transformada em calor devido ao atrito com as rochas. Esse fenômeno é chamado de absorção sísmica e consiste na atenuação da amplitude, que aumenta com o tempo de reflexão e com a frequência, e na dispersão, que causa uma mudança de fase diferente para cada componente da onda.

Tomaremos como base a modelagem compilada por Romanelli Rosa [8] para a absorção sísmica, que consiste na multiplicação, no domínio da frequência, do sinal sem absorção por uma exponencial que depende da frequência e da profundidade, que será medida em tempo de propagação. Este modelo será discutido no capítulo 3. Alguns conceitos importantes para a discussão do fenômeno de absorção sísmica e dos métodos de correção serão apresentados no capítulo 2.

Romanelli Rosa [8] e Varela *et al.* [12] descrevem métodos para corrigir as perdas por absorção. O método chamado de Compensação Q é igual à modelagem do fenômeno de absorção, porém com o sinal da exponencial trocado. Vamos observar que, embora tenha um desempenho muito bom com relação à recuperação do sinal, esse algoritmo é bastante lento, com ordem de complexidade $O(N^2 \ln N)$, onde N é o número de amostras do sinal. Já o método de Varela consiste em uma soma de convoluções com uma função $g(t)$ específica. Segundo Romanelli Rosa [8], em geral podemos considerar até

cinquenta parcelas no método de Varela. Se M é o número de parcelas do somatório e N é o número de amostras do sinal, então o algoritmo de Varela tem ordem de complexidade $O(MN^2)$. Se tivermos $M < \ln N$, então o algoritmo de Varela será mais rápido que a Compensação Q.

No capítulo 4, apresentaremos o método de correção desenvolvido por Duarte [2], que utiliza um filtro recursivo. Trata-se de uma convolução recursiva do sinal de entrada com uma função que vale $\alpha = 1 - \beta$ na primeira amostra e $\beta = -\frac{1}{Q}$ na segunda. Ao compará-lo ao método de Varela, truncando o somatório e a função $g(t)$ no segundo termo, é possível obter um filtro muito parecido com o de Duarte. Alterando os valores dos coeficiente α e β de modo que eles sejam condizentes com a função $g(t)$, percebe-se uma melhora na correção em dados sintéticos.

Além disso, o algoritmo de Duarte possui um parâmetro $GAIN$, que limita o ganho máximo a ser aplicado aos dados durante o processamento. O objetivo do algoritmo é desfazer as perdas por absorção. Porém, se o ganho da amplitude for muito grande, haverá também um aumento do ruído que está presente no dado, podendo deixá-lo dado excessivamente ruidoso. Assim, o parâmetro $GAIN$ está relacionado com o corte da iteração do algoritmo de Duarte, isto é, com a etapa M em que a recursão será interrompida.

Deste modo, se N é o número de amostras do sinal, a ordem de complexidade do algoritmo de Duarte é de $O(MN)$. No pior caso, quando $M = N$, teremos $O(N^2)$, o que já representa uma melhora significativa em relação aos métodos de Romanelli Rosa e Varela, embora ainda seja um pouco lento.

Propomos então uma alteração no algoritmo de Duarte. Até a amostra M , continuaremos com o mesmo procedimento e teremos ordem de complexidade $O(M^2)$ nesta etapa. Conforme Hsu [5], Oppenheim [6] e Tomei [11], a partir dessa amostra, utilizaremos a Transformada de Fourier para torná-lo mais rápido, porém com o mesmo resultado, e teremos ordem de complexidade $O(N \ln N)$. Esse novo algoritmo será mais rápido que o algoritmo original de Duarte se $\ln N < M < N$.

No capítulo 5, vamos comparar os quatro métodos de correção usando os dados sísmicos fornecidos pela Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP). No apêndice, mostraremos como esses dados foram processados no *software Seismic Unix* [1].

2

Principais conceitos

Neste capítulo, serão apresentados os principais conceitos utilizados ao longo deste trabalho, tais como as noções de sinais e sistemas, a Transformada de Laplace, a Transformada Z, a Transformada de Fourier, os diferentes tipos de filtros e o fenômeno de absorção sísmica.

Os principais conceitos podem ser encontrados em Hsu [5], Smith [10], Oppenheim [6] e Duarte [3].

2.1

Sinais

Um **sinal** é uma função que representa uma quantidade ou variável física que contém informações sobre o comportamento ou natureza de um fenômeno.

Tipicamente, representaremos um sinal por uma função $x(t)$ de uma variável independente t , que normalmente será o tempo. Em um caso geral, um sinal pode ser uma função de várias variáveis.

Quando t é uma variável contínua, dizemos que o sinal é de tempo contínuo. Se t assume valores apenas em um conjunto discreto, dizemos que o sinal é de tempo discreto e o representamos na forma de sequência $x[n]$.

Dado um sinal de tempo contínuo $x(t)$, podemos amostrá-lo em alguns instantes, digamos $x(t_0)$, $x(t_1)$, $\dots$, $x(t_n)$, etc e depois tomamos a sequência $x_k = x[k] = x(t_k)$, que é um sinal de tempo discreto.

Exemplos de Sinais:

a) Degrau Unitário:

a.1) Caso contínuo

$$u(t) = \begin{cases} 1, & \text{se } t \geq 0 \\ 0, & \text{se } t < 0 \end{cases} \quad (2.1)$$

A Figura 2.1 representa o gráfico do sinal degrau unitário com tempo contínuo.

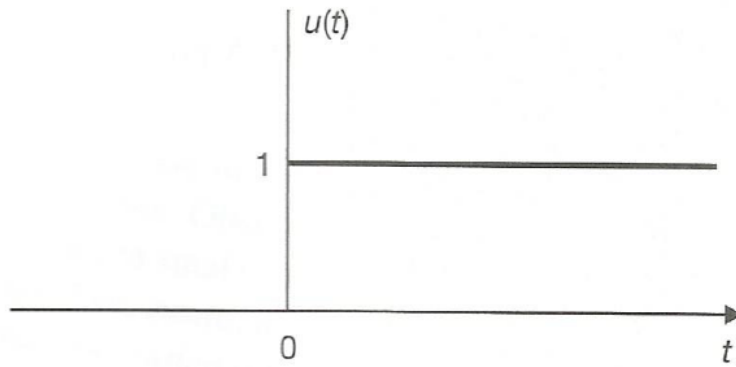


Figura 2.1: Degrau unitário. Fonte: Hsu [5]

a.2) Caso discreto

$$u[n] = \begin{cases} 1, & \text{se } n \geq 0 \\ 0, & \text{se } n < 0 \end{cases} \quad (2.2)$$

A Figura 2.2 representa o gráfico do sinal degrau unitário com tempo discreto.

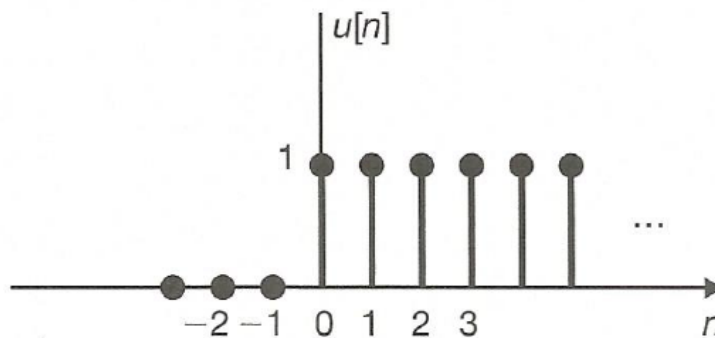


Figura 2.2: Degrau unitário discreto. Fonte: Hsu [5]

b) Impulso Unitário:

b.1) Caso discreto

$$\delta[n] = \begin{cases} 1, & \text{se } n = 0 \\ 0, & \text{se } n \neq 0 \end{cases} \quad (2.3)$$

A Figura 2.3 representa o gráfico do sinal impulso unitário de tempo discreto.

b.2) Caso contínuo: Vamos extrapolar o caso discreto e, como um abuso de conceito, consideramos

$$\delta(t) = \begin{cases} 0, & \text{se } t \neq 0 \\ \infty, & \text{se } t = 0 \end{cases} \quad (2.4)$$

tal que $\int_{-\infty}^{\infty} \delta(t) dt = 1$.

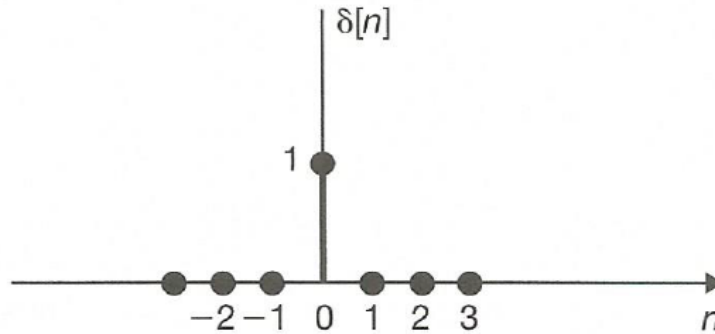


Figura 2.3: Impulso unitário discreto. Fonte: Hsu [5]

Observação 1: $\delta(t - \tau)$, com $\tau \in \mathbb{R}$, nada mais é do que o deslocamento de τ do impulso unitário. Deste modo, $\int_{-\infty}^{\infty} \delta(t - \tau) d\tau = 1$.

Por outro lado, fixando t e tomando τ variando em $\mathbb{R}$, temos que $\delta(t - \tau)$ vale zero para todo $\tau \neq t$. Assim, $x(t)\delta(t - \tau) = x(\tau)\delta(t - \tau)$.

Portanto, temos:

$$\begin{aligned} x(t) &= x(t) \int_{-\infty}^{\infty} \delta(t - \tau) d\tau \\ &= \int_{-\infty}^{\infty} x(t) \delta(t - \tau) d\tau \\ &= \int_{-\infty}^{\infty} x(\tau) \delta(t - \tau) d\tau \end{aligned} \quad (2.5)$$

Observação 2: Um sinal também pode ser representado no domínio da frequência $x(\omega)$ ou $x[\omega]$, onde ω é a frequência. A Transformada de Fourier, que será discutida mais a frente, transforma um sinal no domínio do tempo em um sinal no domínio da frequência.

2.2

Sistemas

Sistema é um modelo matemático de um processo físico que relaciona o sinal de entrada com o sinal de saída.

Quando os sinais de entrada e de saída forem de tempo contínuo, o sistema também será de tempo contínuo. Se os sinais de entrada e saída forem de tempo discreto, o sistema será dito de tempo discreto.

Um sistema S é linear se, dados um escalar α e dois sinais de entrada x_1 e x_2 com $S\{x_1\} = y_1$ e $S\{x_2\} = y_2$, temos $S\{x_1 + x_2\} = y_1 + y_2$ e $S\{\alpha x_1\} = \alpha y_1$.

Dado um sistema S , diremos que S é invariante no tempo quando, dado $\tau \in \mathbb{R}$ ou $k \in \mathbb{Z}$, tivermos $S\{x(t - \tau)\} = y(t - \tau)$ ou $S\{x[n - k]\} = y[n - k]$, onde $y = S\{x\}$.

Um sistema linear invariante no tempo (LTI) é um caso bastante interessante, uma vez que basta conhecer a resposta ao impulso unitário para determinar a saída para qualquer sinal de entrada.

A **convolução** de dois sinais de tempo contínuo $x(t)$ e $y(t)$ é definida por:

$$z(t) = (x * y)(t) = \int_{-\infty}^{\infty} x(\tau)y(t - \tau)d\tau \quad (2.6)$$

A convolução de dois sinais de tempo discreto $x[n]$ e $y[n]$ é definida por:

$$z[n] = (x * y)[n] = \sum_{k=-\infty}^{\infty} x[k]y[n - k] \quad (2.7)$$

Sejam $x(t)$, $y(t)$ e $z(t)$ sinais de tempo contínuo e $x[n]$, $y[n]$ e $z[n]$ sinais de tempo discretos. A operação de convolução possui as seguintes propriedades:

a) Comutatividade:

$$\begin{aligned} (x * y)(t) &= (y * x)(t) \\ (x * y)[n] &= (y * x)[n] \end{aligned}$$

b) Associatividade:

$$\begin{aligned} (\{x * y\} * z)(t) &= (x * \{y * z\})(t) \\ (\{x * y\} * z)[n] &= (x * \{y * z\})[n] \end{aligned}$$

c) Distributividade:

$$\begin{aligned} (x * \{y + z\})(t) &= (x * y)(t) + (x * z)(t) \\ (x * \{y + z\})[n] &= (x * y)[n] + (x * z)[n] \end{aligned}$$

Seja S um sistema LTI de tempo contínuo e seja $h(t) = S\{\delta(t)\}$. A partir da equação 2.5, temos:

$$\begin{aligned} y(t) &= S\{x(t)\} \\ &= S\left\{\int_{-\infty}^{\infty} x(\tau)\delta(t - \tau)d\tau\right\} \\ &= \int_{-\infty}^{\infty} x(\tau)S\{\delta(t - \tau)\}d\tau \\ &= \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau \\ &= (x * h)(t) \end{aligned} \quad (2.8)$$

O caso discreto é análogo ao caso contínuo, isto é, $y[n] = S\{x[n]\} = (x * h)[n]$, onde $h[n] = S\{\delta[n]\}$.

2.3

Transformada de Laplace

Dado um sinal de tempo contínuo $x(t)$, a **Transformada de Laplace** do sinal é dada por:

$$X(s) = \mathcal{L}\{x\} = \int_{-\infty}^{\infty} x(t)e^{-st}dt \quad (2.9)$$

onde $s \in \mathbb{C}$ é tal que a integral converge.

Sejam $x(t)$ e $y(t)$ dois sinais de tempo contínuo, $X(s) = \mathcal{L}\{x\}$, $Y(s) = \mathcal{L}\{y\}$, a , b e t_0 números reais. A Transformada de Laplace tem as seguintes propriedades:

- a) $\mathcal{L}\{ax + by\} = aX(s) + bY(s)$
- b) Se $y(t) = x(t - t_0)$, $\mathcal{L}\{y\} = e^{-st_0}X(s)$
- c) Se $y(t) = e^{s_0 t}x(t)$, $\mathcal{L}\{y\} = X(s - s_0)$
- d) Se $y(t) = x(at)$, $\mathcal{L}\{y\} = \frac{1}{|a|}X\left(\frac{s}{|a|}\right)$
- e) Se $y(t) = x(-t)$, $\mathcal{L}\{y\} = X(-s)$
- f) Se $y = \frac{dx(t)}{dt}$, $\mathcal{L}\{y\} = sX(s)$
- g) Se $y(t) = -tx(t)$, $\mathcal{L}\{y\} = \frac{dX(s)}{ds}$
- h) Se $y(t) = \int_{-\infty}^t x(\tau)d\tau$, $\mathcal{L}\{y\} = \frac{1}{s}X(s)$
- i) $\mathcal{L}\{(x * y)\} = X(s)Y(s)$

Se S é um sistema LTI e $S\{x\} = y(t)$, vimos na seção 2.2 que $y(t) = (x * h)(t)$, onde $h(t) = S\{\delta\}$. Assim, $Y(s) = \mathcal{L}\{(x * h)\} = X(s)H(s)$. Portanto,

$$H(s) = \frac{Y(s)}{X(s)} \quad (2.10)$$

Deste modo, $H(s)$ caracteriza o sistema LTI S para qualquer função X e é chamada de função de transferência do sistema.

A tabela 2.1 apresenta as Transformadas de Laplace de algumas funções.

Função	Transformada de Laplace	Região de Convergência
$\delta(t)$	1	$\mathbb{C}$
$u(t)$	$\frac{1}{s}$	$Re(s) > 0$
$t^k u(t)$	$\frac{k!}{s^{k+1}}$	$Re(s) < 0$
$e^{-at} u(t)$	$\frac{1}{s+a}$	$Re(s) > -Re(a)$
$\text{sen}(\omega t) u(t)$	$\frac{\omega}{s^2 + \omega^2}$	$Re(s) > 0$
$\text{cos}(\omega t) u(t)$	$\frac{s}{s^2 + \omega^2}$	$Re(s) > 0$

Tabela 2.1: Transformadas de Laplace

2.4

Transformada Z

Dado um sinal de tempo discreto $x[n]$, a **Transformada Z** do sinal é dada por:

$$X(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k} \quad (2.11)$$

onde $z \in \mathbb{C}$ e z é tal que o somatório converge.

A Transformada Z é equivalente à Transformada de Laplace para sinais de tempo discreto. Assim, de modo análogo à seção 2.3, dado S um sistema LTI de tempo discreto, tal que $S\{x\} = y$, vimos na seção 2.2 que $y[n] = (x * h)[n]$, onde $h[n] = S\{\delta\}$. Então, $Y(z) = X(z)H(z)$, sendo $X(z)$, $Y(z)$ e $H(z)$ as Transformadas Z de $x[n]$, $y[n]$ e $h[n]$ respectivamente.

$$\text{Logo, } H(z) = \frac{Y(z)}{X(z)}.$$

A função $H(z)$ caracteriza o sistema S de tempo discreto e é chamada de função de transferência.

A tabela 2.2 mostra a Transformada Z de algumas funções.

Função	Transformada Z	Região de Convergência
$\delta[n]$	1	$\mathbb{C}$
$u[n]$	$\frac{z}{z-1}$	$ z > 1$
$a^n u[n]$	$\frac{z}{z-a}$	$ z < a $
$e^{-at}u(t)$	$\frac{1}{s+a}$	$Re(s) > -Re(a)$
$sen[\Omega n]u[n]$	$\frac{(sen\Omega)z}{z^2 - (2cos\Omega)z + 1}$	$ z > 1$
$cos[\omega n]u[n]$	$\frac{z^2 - (cos\Omega)z}{z^2 - (2cos\Omega)z + 1}$	$ z > 1$

Tabela 2.2: Transformadas Z

2.5

Transformada de Fourier de sinais de tempo contínuo

A **Transformada de Fourier** de um sinal $x(t)$ de tempo contínuo é dada por:

$$X(\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-i\omega t} dt \quad (2.12)$$

onde ω é a frequência angular.

Podemos observar que a Transformada de Fourier de um sinal de tempo contínuo é igual à Transformada de Laplace do mesmo sinal quando tomamos $s = i\omega$.

A Transformada Inversa de Fourier é dada por:

$$x(t) = \mathcal{F}^{-1}\{X(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{i\omega t} d\omega \quad (2.13)$$

A tabela 2.3 lista alguns pares de Transformadas de Fourier.

Função	Transformada de Fourier
$\delta(t)$	1
1	$2\pi\delta(\omega)$
$e^{i\omega_0 t}$	$2\pi\delta(\omega - \omega_0)$
$e^{-at}u(t)$	$\frac{1}{s + a}$
$u(t)$	$\pi\delta(\omega) + \frac{1}{i\omega}$
$e^{-at}u(t), a > 0$	$\frac{1}{a + i\omega}$

Tabela 2.3: Transformadas de Fourier

2.6

Transformada de Fourier de sinais de tempo discreto

A **Transformada de Fourier** de um sinal $x[n]$ de tempo discreto é dada por:

$$X(\Omega) = \mathcal{F}\{x[n]\} = \sum_{k=-\infty}^{\infty} x[k]e^{-i\Omega k} \quad (2.14)$$

A Transformada de Fourier de um sinal de tempo discreto é igual à Transformada Z do sinal quando $z = e^{i\Omega}$.

A Transformada Inversa de Fourier de um sinal de tempo discreto é dada por:

$$x[n] = \mathcal{F}^{-1}\{X(\Omega)\} = \frac{1}{2\pi} \int_{2\pi} X(\Omega)e^{i\Omega n} d\Omega \quad (2.15)$$

A tabela 2.4 mostra alguns pares de Transformadas de Fourier.

Agora, consideremos um sinal de tempo discreto $x[n]$ com comprimento finito, isto é, definido apenas para $n = 0, 1, \dots, N - 1$.

A **Transformada de Fourier Discreta** (DFT) do sinal de tempo discreto e comprimento finito $x[n]$ é dada por:

$$X[n] = \sum_{k=0}^{N-1} x[k]e^{-i\frac{2\pi}{N}kn} \quad (2.16)$$

com $n = 0, 1, \dots, N - 1$.

Função $x[n]$	Transformada de Fourier $X(\Omega)$
$\delta[n]$	1
1	$2\pi\delta(\Omega)$
$a^n u[n], a < 1$	$\frac{1}{1 - ae^{-i\Omega}}$
$\text{sen}[\Omega_0 n]$	$-i\pi[\delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0)], \Omega , \Omega_0 \leq \pi$
$\text{cos}[\Omega_0 n]$	$\pi[\delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0)], \Omega , \Omega_0 \leq \pi$

Tabela 2.4: Transformadas de Fourier

A DFT inversa é dada por:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{i\frac{2\pi}{N}kn} \quad (2.17)$$

com $n = 0, 1, \dots, N-1$.

Teorema 2.1 *Sejam $x[n]$ e $y[n]$ dois sinais de tempo discreto, tais que $x[n] = y[n] = 0$ se $n < 0$ ou se $n \geq N \in \mathbb{N}$. Sejam $X[n]$ e $Y[n]$ as Transformadas de Fourier Discretas (DFTs) de $x[n]$ e $y[n]$ respectivamente. Então,*

$$\mathcal{F}\{x[n] * y[n]\} = \mathcal{F}\{x[n]\} \mathcal{F}\{y[n]\} = X[n]Y[n] \quad (2.18)$$

com $n = 0, 1, \dots, N-1$.

Prova: Seja $z[n] = (x * y)[n]$. Então, $z[n] = \sum_{k=0}^n x[k]y[n-k]$.

Como $x[n] = y[n] = 0$ se $n < 0$ ou se $n \geq N$, temos que $z[n] = \sum_{k=0}^{N-1} x[k]y[n-k]$.

Além disso, $Z[n] = \mathcal{F}\{z[n]\} = \sum_{l=0}^{N-1} z[l] e^{-i\frac{2\pi}{N}ln}$.

Assim, temos

$$Z[n] = \sum_{l=0}^{N-1} \left(\sum_{k=0}^{N-1} x[k]y[l-k] \right) e^{-i\frac{2\pi}{N}ln}$$

Como os somatórios são finitos, podemos trocar a ordem do somatório:

$$Z[n] = \sum_{k=0}^{N-1} x[k] \left(\sum_{l=0}^{N-1} y[l-k] e^{-i\frac{2\pi}{N}ln} \right)$$

$$\begin{aligned}
&= \sum_{k=0}^{N-1} x[k] e^{-i \frac{2\pi}{N} kn} Y[n] \\
&= \left(\sum_{k=0}^{N-1} x[k] e^{-i \frac{2\pi}{N} kn} \right) Y[n] \\
&= X[n] Y[n]
\end{aligned}$$

□

2.7

Fast Fourier Transform - FFT

A **Transformada Rápida de Fourier** (*Fast Fourier Transform - FFT*) é um algoritmo bastante eficiente para calcular a DFT apresentada na seção anterior. Se utilizarmos a equação (2.16) para calcular a DFT, teremos um algoritmo com ordem de complexidade $O(N^2)$, onde N é o número de amostras do sinal.

O algoritmo da FFT também calcula a DFT de um sinal, porém é mais rápido, com ordem de complexidade $O(N \ln N)$.

A seguir, vamos descrever o funcionamento deste algoritmo.

Primeiro, vamos supor que $N = 2^M$ para algum $M \in \mathbb{N}$. Caso isso não ocorra, basta acrescentar zeros ao sinal até que o número de amostras seja uma potência de 2. Deste modo, dado um sinal $x[n]$, temos:

$$\begin{aligned}
X[n] = FFT(x, N) &= \sum_{k=0}^{N-1} x[k] e^{-i \frac{2\pi n}{N} k} \\
&= \sum_{k=0, k \text{ par}}^{N-1} x[k] e^{-i \frac{2\pi n}{N} k} + \sum_{k=0, k \text{ impar}}^{N-1} x[k] e^{-i \frac{2\pi n}{N} k} \\
&= \sum_{m=0}^{\frac{N}{2}-1} x[2m] e^{-i \frac{2\pi n}{N} 2m} + \sum_{l=0}^{\frac{N}{2}-1} x[2l+1] e^{-i \frac{2\pi n}{N} (2l+1)}
\end{aligned}$$

Considerando $x_0[m] = x[2m]$ e $x_1[l] = x[2l+1]$, com $m, l = 0, 1, \dots, \frac{N}{2}-1$, temos:

$$\begin{aligned}
X[n] &= \sum_{m=0}^{\frac{N}{2}-1} x_0[m] e^{-i \frac{2\pi n}{N} m} + e^{-i \frac{2\pi n}{N}} \sum_{l=0}^{\frac{N}{2}-1} x_1[l] e^{-i \frac{2\pi n}{N} l} \\
&= FFT\left(x_0, \frac{N}{2}\right) + e^{-i \frac{2\pi n}{N}} FFT\left(x_1, \frac{N}{2}\right) \\
&= X_0[n] + e^{-i \frac{2\pi n}{N}} X_1[n]
\end{aligned} \tag{2.19}$$

Porém, X_0 e X_1 possuem apenas $\frac{N}{2}$ amostras e X possui N amostras. Podemos considerar X_0 e X_1 periódicas e, assim, $X_0[k + \frac{N}{2}] = X_0[k]$ e $X_1[k + \frac{N}{2}] = X_1[k]$ para $k = 0, 1, \dots, \frac{N}{2} - 1$.

Além disso, podemos observar que:

$$\begin{aligned} e^{-i\frac{2\pi}{N}(k+\frac{N}{2})} &= e^{-i\frac{2\pi}{N}k} e^{-i\frac{2\pi}{N}\frac{N}{2}} \\ &= e^{-i\frac{2\pi}{N}k} e^{-i\pi} \\ &= -e^{-i\frac{2\pi}{N}k} \end{aligned} \quad (2.20)$$

Portanto, $X[k] = X_0[k] + e^{-i\frac{2\pi}{N}k} X_1[k]$, se $k < \frac{N}{2}$ e $X[k] = X_0[k] - e^{-i\frac{2\pi}{N}k} X_1[k]$ se $k \geq \frac{N}{2}$.

Para calcular a FFT de x_0 e de x_1 , dividiremos as amostras em pares e ímpares novamente, o que resultará na soma de duas FFTs de $\frac{N}{4}$ amostras. Seguiremos este processo até que reste apenas uma amostra. Neste caso, se $x[0]$ é a única amostra, a DFT será

$$X[0] = \sum_{k=0}^0 x[k] e^{-i\frac{2\pi}{1}0k} = x[0] \quad (2.21)$$

Por recursão, aplicando a FFT para calcular X_0 e X_1 , obtemos a complexidade $O(N \ln N)$.

A figura 2.4 mostra o algoritmo da FFT.

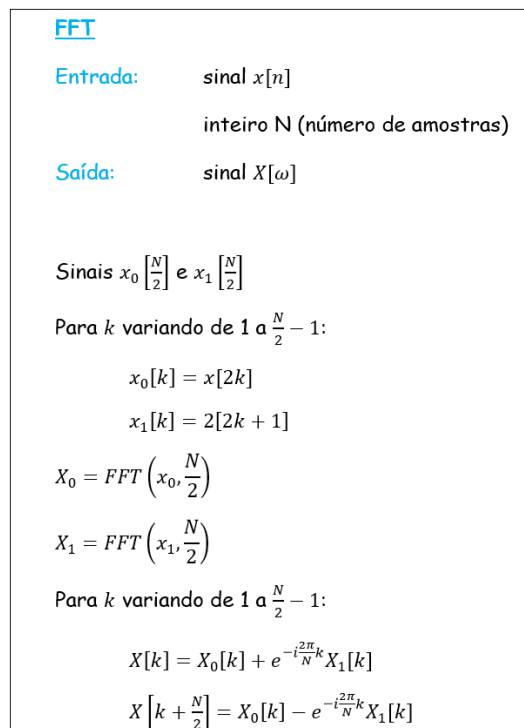


Figura 2.4: Algoritmo para calcular a FFT

2.8

Filtros

Dado um sinal $x(t)$, um **filtro** é qualquer modificação desse sinal, gerando um novo sinal $y(t)$. Em particular, todo filtro é um sistema.

Quanto à resposta em frequência, dentre os filtros lineares invariantes no tempo podemos destacar os seguintes tipos:

a) Filtro passa-baixa: permite a passagem de frequências abaixo da frequência de corte (ω_c) e atenua a amplitude das demais frequências. A figura 2.5(a) representa a resposta em frequência deste filtro.

b) Filtro passa-alta: permite a passagem de frequências acima da frequência de corte (ω_c) e atenua a amplitude das frequências menores que ω_c . A figura 2.5(b) representa a resposta em frequência deste filtro.

c) Filtro passa-faixa: permite a passagem de frequências situadas em uma faixa delimitada pela frequência de corte inferior (ω_{CI}) e a frequência de corte superior (ω_{CS}). As frequências abaixo ou acima desta faixa são atenuadas. A figura 2.5(c) representa este caso.

d) Filtro rejeita-faixa: atenua as frequências situadas em uma faixa delimitada pela frequência de corte inferior (ω_{CI}) e a frequência de corte superior (ω_{CS}) e permite a passagem das frequências abaixo ou acima desta faixa. A figura 2.5(d) representa este caso.

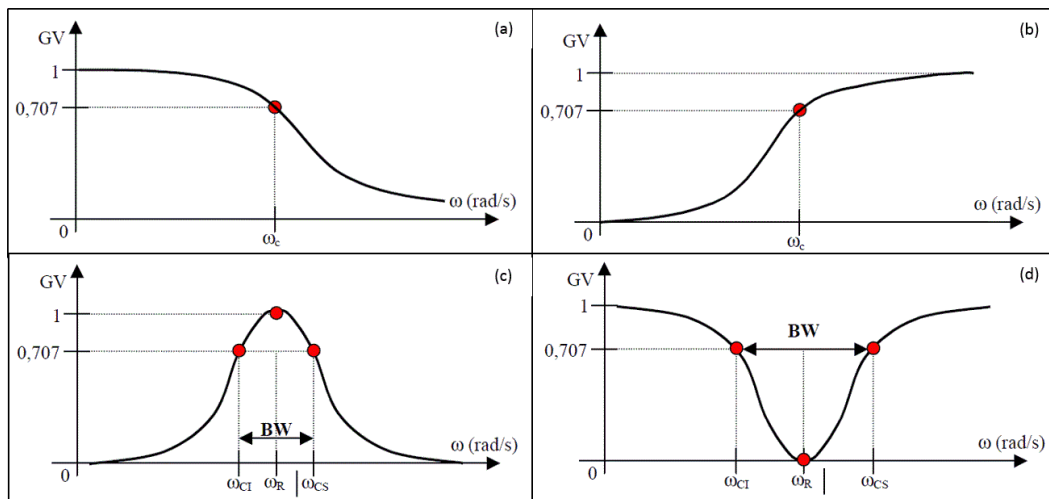


Figura 2.5: (a) Filtro passa-baixa; (b) Filtro passa-alta; (c) Filtro passa-faixa; (d) Filtro rejeita-faixa. Fonte: <http://www.peb.ufrj.br/cursos/COB781/Ref-03.pdf>

Os filtros digitais podem ser implementados de dois modos: por convolução (*Finite Impulse Response*) e por recursão (*Infinite Impulse Response*).

Quando o filtro é LTI, ele pode ser representado pela convolução do sinal de entrada com a resposta ao impulso unitário, como vimos na seção 2.2. Esse

tipo de filtro, gerado por uma convolução, tem a resposta ao impulso finita e por isso é chamado de *Finite Impulse Response* (FIR).

Dado um sinal de entrada $x[n]$, a forma geral de um filtro FIR com sinal de saída $y[n]$ é:

$$y[n] = \sum_{k=0}^N b_k x[n-k] \quad (2.22)$$

Por outro lado, os filtros recursivos possuem resposta ao impulso infinita e são chamados de *Infinite Impulse Response* (IIR).

Um filtro IIR aplicado a um sinal de entrada $x[n]$ é da forma

$$y[n] = \sum_{i=0}^P a_i x[n-i] + \sum_{j=1}^Q b_j y[n-j] \quad (2.23)$$

onde $y[n]$ é o sinal de saída.

Isso significa que para determinar a n -ésima amostra do sinal de saída, é preciso conhecer as $n-1$ amostras anteriores, ou seja, trata-se de um filtro recursivo.

Filtros FIR podem ter um desempenho qualitativo melhor que filtros IIR, mas são bem mais lentos.

2.9

Decibéis

Um **bel** é uma maneira de expressar a razão entre dois sinais e significa que a potência foi alterada em 10 vezes. Um **decibel** (dB) é um décimo de um bel. Assim, uma amplificação de 30dB gera um sinal de saída cuja potência é de 10^3 vezes a potência do sinal de entrada. Portanto, temos:

$$\begin{aligned} 10^{\frac{dB}{10}} &= \frac{P_2}{P_1} \\ \frac{dB}{10} &= \log_{10} \left(\frac{P_2}{P_1} \right) \\ dB &= 10 \log_{10} \left(\frac{P_2}{P_1} \right) \end{aligned} \quad (2.24)$$

onde P_1 e P_2 são as potências de entrada e saída respectivamente.

Porém, quando trabalhamos com sinais, em geral usamos a amplitude ao invés da potência. Como a amplitude é proporcional à raiz quadrada da potência, temos:

$$\begin{aligned} dB &= 10 \log_{10} \left(\frac{A_2}{A_1} \right)^2 \\ &= 20 \log_{10} \left(\frac{A_2}{A_1} \right) \end{aligned} \quad (2.25)$$

onde A_1 e A_2 são as amplitudes de entrada e saída respectivamente.

2.10

Frequência de Nyquist

O sinal sísmico é um sinal de tempo contínuo, mas a informação é registrada como um sinal de tempo discreto. Nos levantamentos sísmicos, a amostragem do sinal é geralmente de 2ms ou 4ms.

A partir de um sinal de tempo discreto, tentamos reconstruir o sinal de tempo contínuo. Porém, existem limitações. Usando amostragem regular e reconstrução por funções trigonométricas, a maior frequência que pode ser recuperada a partir de um sinal de tempo discreto é chamada de **Frequência de Nyquist**, dada por:

$$f_{Nyq} = \frac{1}{2\Delta t} \quad (2.26)$$

onde Δt é o intervalo de amostragem.

Assim, quanto menor o intervalo de amostragem, maior será a frequência de Nyquist e mais próximo do dado original estaremos.

2.11

Absorção sísmica

A **absorção sísmica** refere-se à perda de energia que ocorre durante a propagação da onda sísmica. Essa perda decorre da transformação de parte da energia em calor, devido ao atrito com as partículas das rochas. A absorção consiste em dois fenômenos: atenuação e dispersão.

A atenuação é a redução da amplitude do sinal e tem um efeito cumulativo no tempo. A atenuação aumenta com o tempo e com a frequência.

Toda onda pode ser decomposta em um somatório de senóides, onde cada senóide tem uma frequência. Quando a onda se propaga nas diferentes camadas de rocha, cada componente pode ter uma velocidade de propagação diferente, fazendo com que algumas componentes se propaguem com mais rapidez que outras. Deste modo, as componentes sofrem diferentes deslocamentos de tempo, causando o fenômeno de dispersão.

Esse deslocamento de tempo nada mais é do que uma mudança de fase no espectro de amplitude da onda. De fato, como visto na seção 2.3, uma translação de $(t - t_0)$ no tempo representa uma multiplicação por e^{-st_0} da Transformada de Laplace, ou seja, uma multiplicação por $e^{-i\omega t_0}$ na Transformada de Fourier do sinal.

3

Modelagem e correção da absorção sísmica

Neste capítulo, vamos apresentar um algoritmo que modela o fenômeno de absorção sísmica. Este modelo foi descrito por Romanelli Rosa [8] em seu livro e trata-se da reunião dos conceitos de uma vasta literatura. Destacamos que Romanelli Rosa não é o autor do método, embora o tenha descrito de forma excelente. Com base nessa modelagem, vamos apresentar dois métodos para corrigir os efeitos da absorção: a Compensação Q e o método desenvolvido por Varela *et al.* [12]. Vamos observar que os algoritmos apresentam bons resultados, mas podemos obter um método computacionalmente mais rápido, que será apresentado no próximo capítulo.

3.1

A modelagem da absorção

Consideremos um sinal sísmico $s(t)$, que não sofreu os efeitos da absorção. Esse sinal se propaga ao longo das camadas de rochas da subsuperfície e quanto maior a profundidade, maior é a perda causada por absorção. Ao invés de medir essa profundidade em metros, vamos medi-la em tempo de propagação, ou seja, no instante τ , o sinal terá se propagado até determinada profundidade e terá sofrido o efeito da absorção correspondente.

O método descrito por Romanelli Rosa [8] para simular a absorção consiste, primeiro, em estimar a frequência inicial do sinal pela sua Transformada de Fourier, chamada usualmente de frequência instantânea. Isso corresponde à profundidade zero ou, se considerarmos o equivalente em tempo da profundidade, ao instante $\tau = 0$. Depois, tomamos um intervalo de tempo $\Delta\tau$ e, para cada $n \in \mathbb{N}$, calculamos a Transformada de Fourier do sinal na profundidade $(\tau + n\Delta\tau)$.

Se em cada um desses instantes calcularmos a Transformada Inversa de Fourier, obteremos o sinal $a(t)$, que é a simulação dos efeitos da absorção no sinal $s(t)$.

Romanelli Rosa [8] descreve a absorção pela seguinte equação:

$$S(\tau + \Delta\tau, \omega) = S(\tau, \omega) \exp\left(-\frac{\omega\Delta\tau}{2Q}\right) \exp\left(-i\frac{\omega\Delta\tau}{\pi Q} \ln\left(\frac{\omega}{\omega_0}\right)\right) \exp(i\omega\Delta\tau) \quad (3.1)$$

onde:

ω é a frequência;

ω_0 é a frequência de referência;

Q é o fator de qualidade;

τ é o equivalente em tempo da profundidade;

$\Delta\tau$ é o intervalo entre os valores sucessivos de τ ;

$S(\tau, \omega)$ é a frequência instantânea na profundidade τ ;

$S(\tau + \Delta\tau, \omega)$ é a frequência instantânea na profundidade $\tau + \Delta\tau$.

Na equação (3.1), o fator $\exp\left(-\frac{\omega\Delta\tau}{2Q}\right)$ refere-se ao fenômeno da atenuação e o fator $\exp\left(-i\frac{\omega\Delta\tau}{\pi Q} \ln\left(\frac{\omega}{\omega_0}\right)\right) \exp(i\omega\Delta\tau)$ refere-se à dispersão.

A frequência de referência ω_0 pode ser escolhida arbitrariamente, desde que seja constante para todos os instantes. Note que uma escolha diferente resultaria apenas numa translação de fase constante. Neste trabalho, ω_0 será a frequência de Nyquist.

Seja $A(\tau, \omega) = \exp\left(-\frac{\omega\tau}{2Q}\right) \exp\left(-i\frac{\omega\tau}{\pi Q} \ln\left(\frac{\omega}{\omega_0}\right)\right)$. Então, a equação (3.1) pode ser reescrita como:

$$S(\tau + \Delta\tau, \omega) = S(\tau, \omega) A(\Delta\tau, \omega) \exp(i\omega\Delta\tau) \quad (3.2)$$

Assim, dado um sinal $s(t)$ sem efeito da absorção, escolhemos $\Delta\tau$ pequeno e obtemos o sinal que simula a absorção $a(t)$ aproximando a frequência instantânea pela Transformada de Fourier em janelas do próprio sinal:

$$1) a(0) = s(0)$$

$$2) \text{ Calculamos } S(\tau = 0, \omega) = \mathcal{F}\{s(t)\}(\omega)$$

3) Dado $t_0 \in \mathbb{R}_+$, existe $n \in \mathbb{N}$ tal que $n\Delta\tau \leq t_0 < (n+1)\Delta\tau$. Então, fazemos o seguinte:

$$3.1) \text{ Calculamos } S(n\Delta\tau, \omega) = S(\tau = 0, \omega) A(n\Delta\tau, \omega) \exp(i\omega n\Delta\tau)$$

$$3.2) \text{ Calculamos } s_n(t) = \mathcal{F}^{-1}\{S(n\Delta\tau, \omega)\}(t)$$

$$3.3) a(t_0) = s_n(0)$$

Ora, se tomarmos

$$s'_n(t) = \mathcal{F}^{-1}\{S(\tau = 0, \omega) A(n\Delta\tau, \omega)\}(t) \quad (3.3)$$

teremos $s_n(t) = s'_n(t + n\Delta\tau)$. Deste modo, $a(t_0) = s_n(0) = s'_n(n\Delta\tau)$.

O método estima as frequências instantâneas com a Transformada de Fourier, a fim de simular a absorção sísmica.

Quando trabalhamos com um sinal discreto, podemos tomar $\Delta\tau$ tão pequeno quanto o intervalo de amostragem. Assim, para cada amostra do sinal, calculamos o efeito da absorção, ou seja, se $s[n]$ é o sinal sem absorção, calculamos $a[n]$ da forma abaixo:

$$1) a[0] = s[0]$$

$$2) S[\omega] = \mathcal{F}\{s[n]\}[\omega]$$

3) Seja N o número de amostras do sinal. Para k variando de 1 a $N - 1$, calculamos $S_k[\omega] = S[\omega]A_k[\omega]$, onde $A_k[\omega] = A(k\Delta\tau, \omega)$.

$$4) a[k] = s_k[k], \text{ onde } s_k[n] = \mathcal{F}^{-1}\{S_k[\omega]\}[n]$$

Na figura 3.1, é possível comparar o sinal $s(t) = \text{sen}(100\pi t)$ na cor vermelha e o sinal com o efeito da absorção na cor azul. Neste caso, utilizamos intervalo de amostragem $\Delta t = 0.001s$, t variando de 0 a $0.5s$ e $Q = 180$. Como temos apenas uma frequência, não existe dispersão.

Na figura 3.2, fazemos o mesmo para o sinal $r(t) = \text{sen}(50\pi t)$. Observamos que quanto maior a frequência da senóide, maior será a absorção.

A figura 3.3 mostra o espectro de amplitude do sinal $s(t)$ e a figura 3.4 representa o espectro de amplitude após a simulação da absorção.

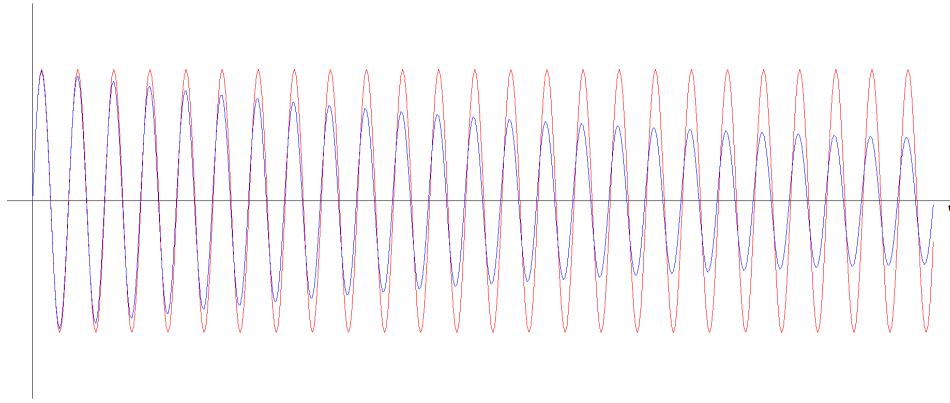


Figura 3.1: Simulação dos efeitos da absorção no sinal $s(t)$

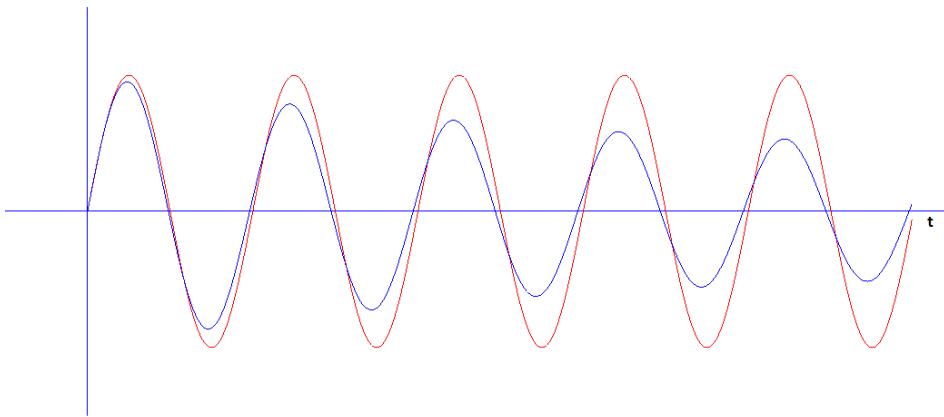


Figura 3.2: Simulação dos efeitos da absorção na senóide $r(t)$



Figura 3.3: Espectro de amplitude do sinal $s(t)$

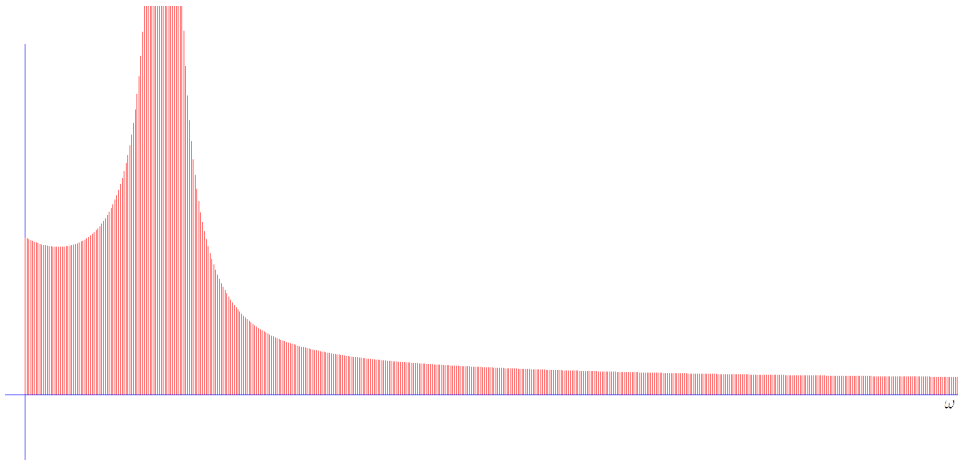


Figura 3.4: Espectro de amplitude após a simulação da absorção

3.2

Compensação Q

A partir desta modelagem, Romanelli Rosa [8] descreve um método para corrigir as perdas por absorção, a Compensação Q. Este método é muito parecido com a modelagem do fenômeno vista no item anterior.

Na verdade, o procedimento é o mesmo, porém os sinais dos expoentes da equação 3.1 são trocados, ou seja, a equação que define a Compensação Q é dada por:

$$S(\tau + \Delta\tau, \omega) = S(\tau, \omega) \exp\left(\frac{\omega\Delta\tau}{2Q}\right) \exp\left(i\frac{\omega\Delta\tau}{\pi Q} \ln\left(\frac{\omega}{\omega_0}\right)\right) \exp(-i\omega\Delta\tau) \quad (3.4)$$

Dado um sinal $s(t)$ que sofreu absorção, seguimos as etapas abaixo para obter um sinal $c(t)$ corrigido:

$$1) c(0) = s(0)$$

$$2) \text{ Calculamos } S(\tau = 0, \omega) = \mathcal{F}\{s(t)\}(\omega)$$

3) Dado $t_0 \in \mathbb{R}_+$, existe $n \in \mathbb{N}$ tal que $n\Delta\tau \leq t_0 < (n+1)\Delta\tau$. Então, fazemos o seguinte:

$$3.1) \text{ Calculamos } S(n\Delta\tau, \omega) = S(\tau = 0, \omega) A(-n\Delta\tau, \omega) \exp(-i\omega n\Delta\tau)$$

$$3.2) \text{ Calculamos } s_n(t) = \mathcal{F}^{-1}\{S(n\Delta\tau, \omega)\}(t)$$

$$3.3) c(t_0) = s_n(0)$$

Como já mencionado anteriormente, podemos tomar

$$s'_n(t) = \mathcal{F}^{-1}\left\{S(\tau = 0, \omega) \exp\left(\frac{\omega n\Delta\tau}{2Q}\right) \exp\left(i\frac{\omega n\Delta\tau}{\pi Q} \ln\left(\frac{\omega}{\omega_0}\right)\right)\right\}(t) \quad (3.5)$$

e teremos $c(t_0) = s'_n(n\Delta\tau)$.

Como o dado sísmico é registrado como um sinal discreto, utilizaremos a mesma ideia da seção anterior para obter um sinal $c[n]$ com a correção da absorção, ou seja:

$$1) c[0] = s[0]$$

$$2) S[\omega] = \mathcal{F}\{s[n]\}[\omega]$$

3) Seja N o número de amostras do sinal. Para k variando de 1 a $N-1$, calculamos $S_k[\omega] = S[\omega] A_{-k}[\omega]$, onde $A_{-k}[\omega] = A(-k\Delta\tau, \omega)$ e $\Delta\tau$ é o intervalo de amostragem.

$$4) c[k] = s_k[k], \text{ onde } s_k[n] = \mathcal{F}^{-1}\{S_k[\omega]\}[n]$$

Na figura 3.5(a), representamos o sinal $s(t) = \sin(100\pi t)$ na cor vermelha, na figura 3.5(b), mostramos o sinal sob efeito da absorção na cor azul, a

figura 3.5(c) representa a Compensação Q aplicada ao sinal da figura 3.5(b) na cor verde e, por fim, a figura 3.5(d) sobrepõe os três casos anteriores.

Podemos observar que a curva verde sobrepõe a curva vermelha fazendo com que ela praticamente não apareça na figura. Isso significa que, neste caso, a compensação Q recuperou o sinal.

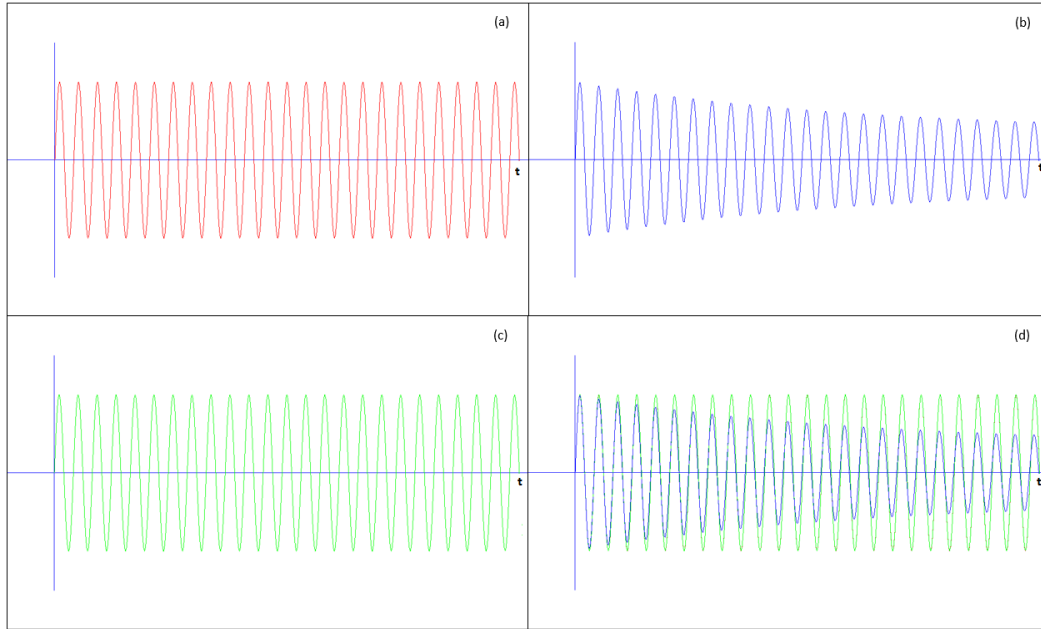


Figura 3.5: Simulação dos efeitos da absorção e da Compensação Q

A figura 3.6 mostra o sinal $p(t) = \sin(50\pi t) + \sin(70\pi t)$ em vermelho e a simulação da absorção em azul. A figura 3.7 sobrepõe o sinal $p(t)$ em vermelho, a absorção em azul e a compensação Q em verde. É possível observar que mesmo nesse caso, a Compensação Q recupera muito bem o sinal original, em parte porque as frequências são suficientemente distintas para evitar *aliasing*.

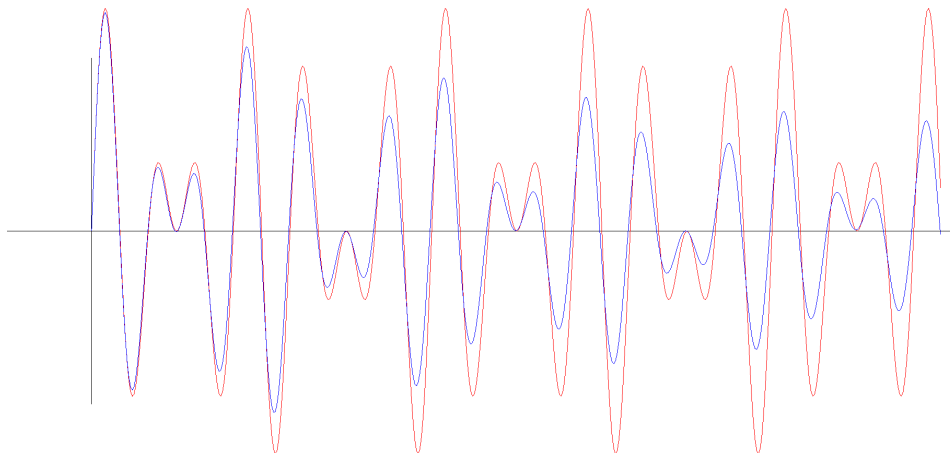


Figura 3.6: Simulação dos efeitos da absorção no sinal $p(t)$

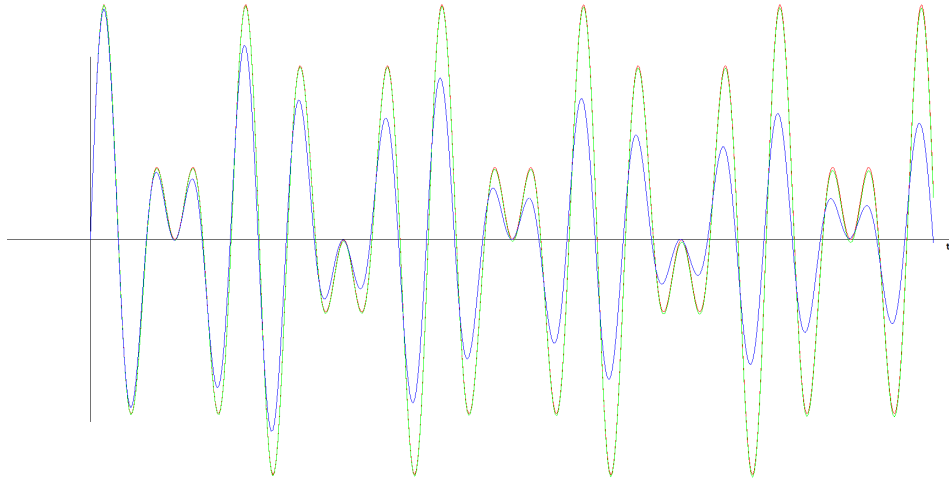


Figura 3.7: Simulação dos efeitos da absorção no sinal $p(t)$

O problema deste algoritmo é que ele é lento, tem ordem de complexidade $O(N^2 \ln N)$. De fato, para cada uma das N amostras, precisamos fazer uma Transformada Inversa de Fourier, que tem ordem de complexidade $O(N \ln N)$ quando utilizamos a FFT, que é o algoritmo mais eficiente.

3.3

O método de Varela

O método de correção do fenômeno de absorção proposto por Varela *et al.* [12] considera a modelagem descrita no item 3.1, porém usa a convolução ao invés de usar a Transformada de Fourier. Assim, este algoritmo funciona no domínio do tempo.

A compensação Q é dada pelo fator

$$U(\Delta\tau, \omega) = \exp\left(\frac{\omega\Delta\tau}{2Q}\right) \exp\left(i\frac{\omega\Delta\tau}{\pi Q} \ln\left(\frac{\omega}{\omega_0}\right)\right) \quad (3.6)$$

Fazendo $\omega = 2\pi f$, onde f é a frequência em Hertz, temos:

$$U(\Delta\tau, f) = \exp\left(\frac{\pi f \Delta\tau}{Q}\right) \exp\left(i\frac{2f \Delta\tau}{Q} \ln\left(\frac{f}{f_0}\right)\right) \quad (3.7)$$

Podemos reescrever $U(\Delta\tau, f)$ do seguinte modo:

$$U(\Delta\tau, f) = \exp\left(\frac{\pi \Delta\tau}{Q} G(f)\right) \quad (3.8)$$

onde $G(f) = f + i\frac{2f}{\pi} \ln\left(\frac{f}{f_0}\right)$.

Considerando $y = G(f)$ e tomando a série de Taylor em torno de $y = 0$, temos:

$$\begin{aligned}
U(\Delta\tau, f) &= 1 + \frac{\pi\Delta\tau}{Q}y + \frac{1}{2!} \left(\frac{\pi\Delta\tau}{Q} \right)^2 y^2 + \dots \\
&= 1 + \frac{\pi\Delta\tau}{Q}G(f) + \frac{1}{2!} \left(\frac{\pi\Delta\tau}{Q} \right)^2 G(f)^2 + \dots \quad (3.9)
\end{aligned}$$

Agora, vamos calcular a Compensação Q como foi mostrado na seção anterior, utilizando $U(\Delta\tau, f)$ da equação (3.9), $\tau = 0$ e $\Delta\tau = t_0$, onde t_0 é um instante qualquer.

Assim, para cada $t_0 \in \mathbb{R}_+$ fixo, temos $S(t_0, f) = S(0, f)U(t_0, f)$. Tomando a Transformada Inversa de Fourier, pelo Teorema 2.1, temos que:

$$s_{t_0}(t) = s(t) + \frac{\pi t_0}{Q}(s * g)(t) + \frac{1}{2!} \left(\frac{\pi t_0}{Q} \right)^2 (s * g * g)(t) + \dots \quad (3.10)$$

onde $g(t) = \mathcal{F}^{-1}\{G(f)\}(t)$

Segundo Varela *et al.* [12], $g(t)$ é dada por:

$$g(t) = \begin{cases} \frac{1}{4}, & \text{se } t = 0 \\ \frac{-2}{(\pi t)^2}, & \text{se } t = 1, 3, 5, \dots \\ 0, & \text{caso contrário} \end{cases} \quad (3.11)$$

Porém a equação (3.10) representa a correção da absorção apenas no instante t_0 , ou seja, se $c(t)$ é a correção do dado $s(t)$, temos:

$$c(t_0) = s_{t_0}(t_0) = s(t_0) + \frac{\pi t_0}{Q}(s * g)(t_0) + \frac{1}{2!} \left(\frac{\pi t_0}{Q} \right)^2 (s * g * g)(t_0) + \dots \quad (3.12)$$

Na figura 3.8, apresentamos o funcionamento desse algoritmo.

```

// M = número de parcelas do somatório
// x = dado de entrada
// y = dado de saída
// conv = convolução

void varela(int M, const double *x, double *y)
{
    double z[N] ;
    double inc = (PI)/Q ;
    for(int i = 0; i < N; ++i)
    {
        z[i] = x[i] ;
        y[i] = x[i] ;
    }

    for (int k = 1; k <= M; ++k)
    {
        double w[N] ;
        conv(z, g, w);
        for (int i = 0; i < N; ++i)
        {
            z[i] = (double)i*(inc/k) * w[i] ;
            y[i] += z[i] ;
        }
    }
}

void conv(const double *x, const double *g, double *z)
{
    for (int k = 0; k < N; ++k)
    {
        z[k] = 0.0 ;
        for(int j = 0; j <= k; ++j)
        {
            z[k] += x[j]*g[k-j] ;
        }
    }
}

```

Figura 3.8: Algoritmo de Varela

Como a convolução tem ordem de complexidade $O(N^2)$, onde N é o número de amostras do sinal, esse algoritmo tem ordem de complexidade $O(MN^2)$, onde M é o número de parcelas que serão consideradas no somatório da equação (3.12). Ao citar este modelo, Romanelli Rosa [8] afirma que o valor de M que gera o melhor custo benefício é, em geral, menor que 50. Deste modo, podemos considerar M uma constante menor que 50 e teremos ordem de complexidade $O(N^2)$, isto é, mais rápido que a Compensação Q.

Na figura 3.9 apresentamos a senóide $s(t) = \text{sen}(100\pi t)$ (a), a simulação da absorção (b), a correção com o método de Varela (c) e a sobreposição dos três casos (d). Utilizamos $M = 50$.

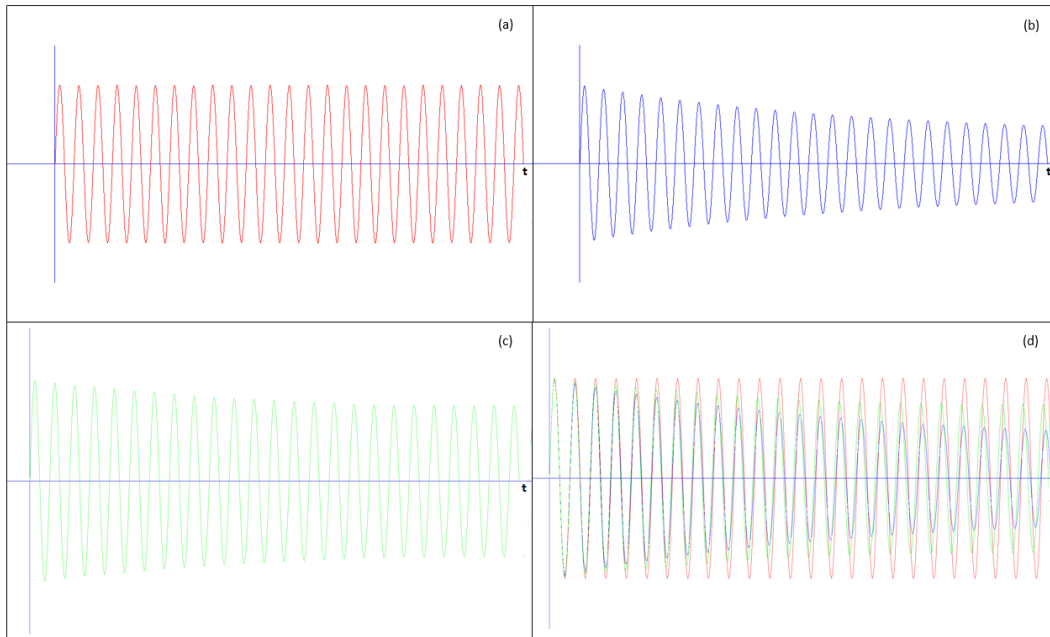


Figura 3.9: Simulação dos efeitos da absorção e da correção de Varela com $M = 50$

A figura 3.3 representa o espectro de amplitude do sinal $s(t)$ e a figura 3.4 mostra o espectro da simulação da absorção. A figura 3.10 a seguir apresenta o espectro de amplitude do sinal após a correção de Varela.

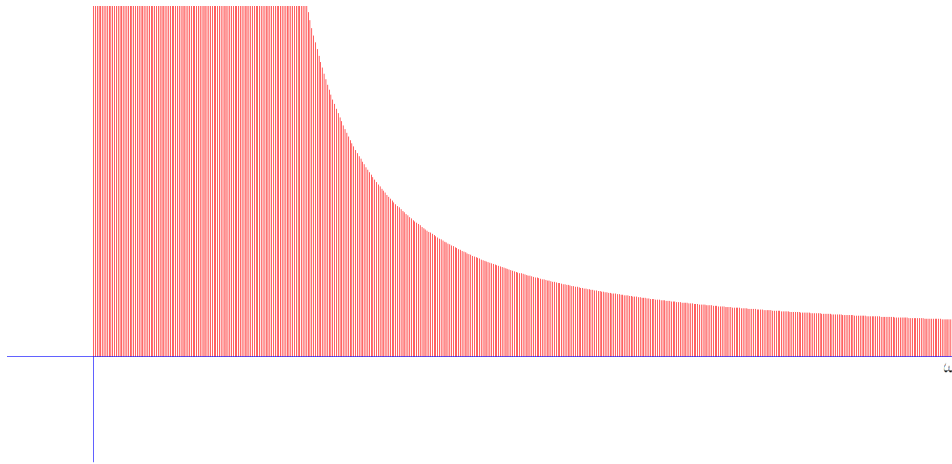


Figura 3.10: Espectro de amplitude do sinal $s(t)$

Podemos observar que, embora possa ser mais rápido que a compensação Q , a correção não foi tão precisa nesse exemplo sintético. Apesar disso, existe um ganho em relação ao dado com efeito da absorção.

4

O método de Duarte

Agora vamos apresentar o algoritmo proposto por Duarte [2] para corrigir as perdas por absorção. A seguir, comparamos este método ao algoritmo de Varela, apresentado no capítulo anterior, propomos uma alteração nos parâmetros α e β do algoritmo de Duarte [2] e, por fim, fazemos uma alteração no algoritmo para torná-lo mais rápido.

4.1

Apresentação do método

O método de Duarte consiste em uma convolução recursiva do sinal de entrada $x[t]$ com uma função $a[t]$ tal que

$$a[t] = \begin{cases} \alpha, & \text{se } t = 0 \\ \beta, & \text{se } t = 1 \\ 0, & \text{caso contrário} \end{cases} \quad (4.1)$$

onde $\beta = -\frac{1}{Q}$, $\alpha = 1 - \beta$ e Q é o fator que qualidade.

O algoritmo, apresentado na figura 4.1, consiste em um método iterativo que toma o sinal de entrada e faz uma convolução com $a[t]$. Na iteração de ordem k , fazemos a convolução do resultado do passo anterior, a partir da amostra k , com a função $a[t]$. Esse processo é interrompido na etapa de ordem M .

Seja $y[t]$ o sinal de saída. Dado um sinal $x[t]$, a primeira amostra $x[0]$ não sofre alteração, isto é, $y[0] = x[0]$. A seguir, fazemos a convolução de $x[t]$ com $a[t]$ para $t \geq 1$ e obtemos um sinal intermediário no processo, que chamaremos de $x_1[t]$:

$$x_1[t] = x[t]a[0] + x[t-1]a[1] = x[t]\alpha + x[t-1]\beta \quad \text{para todo } t \geq 1 \quad (4.2)$$

Depois, tomamos $y[1] = x_1[1]$, ou seja:

$$y[1] = x_1[1] = x[1]\alpha + x[0]\beta \quad (4.3)$$

```

// X = dado de entrada
// N = número de amostras
// Q = Fator de qualidade
// GAIN = ganho desejado

void duarte (float Q, int N, float GAIN, float *X)
{
    float beta = -1.0/Q ;
    float alpha = 1.0 - beta ;

    float M = (GAIN/20.0)/log10(1.0 + fabs(beta)) ; //corte na iteração

    if(M > N)
    {
        M = N ;
    }

    float xb = X[0] ;

    for(int j = 0; j < M ; ++j)
    {
        xb=X[j];
        for(int i = j ; i < N; ++i)
        {
            float xa = X[i] ;
            X[i] = xa*alpha + xb*beta ;
            xb = xa ;
        }
    }
}

```

Figura 4.1: Algoritmo proposto por Duarte

Na terceira etapa, fazemos a convolução do novo sinal $x_1[t]$ a partir de $t = 2$ com $a[t]$ e obtemos $x_2[t]$:

$$\begin{aligned}
 x_2[t] &= x_1[t]a[0] + x_1[t-1]a[1] \\
 &= (x[t]a[0] + x[t-1]a[1])a[0] + (x[t-1]a[0] + x[t-2]a[1])a[1] \\
 &= \alpha^2 x[t] + 2\alpha\beta x[t-1] + \beta^2 x[t-2] \quad \text{para todo } t \geq 2.
 \end{aligned} \tag{4.4}$$

Deste modo, $y[2] = x_2[2] = \alpha^2 x[2] + 2\alpha\beta x[1] + \beta^2 x[0]$.

Como o algoritmo não faz mais convoluções a partir da etapa de número $M + 1$, temos que:

$$y[k] = \begin{cases} \sum_{i=0}^k \binom{k}{i} \alpha^{k-i} \beta^i x[k-i] & \text{se } k \leq M \\ \sum_{i=0}^M \binom{M}{i} \alpha^{M-i} \beta^i x[k-i] & \text{se } k > M \end{cases} \quad (4.5)$$

Isso significa que $y[k] = (x * \underbrace{a * \dots * a}_{M \text{ vezes}})[k]$, se $k \leq M$ e $x[k] = (x * \underbrace{a * \dots * a}_{M \text{ vezes}})[k]$, se $k > M$.

Além disso, o método de Duarte funciona muito bem como seu próprio inverso quando trocamos o sinal de Q . Segundo Duarte, quando Q é negativo, o algoritmo simula as perdas por absorção e quando Q é positivo, ele faz a correção.

A figura 4.2(a) apresenta o gráfico de uma função que vale 1 nos pontos 0, 2, 4 e 6, vale -1 nos pontos 1, 3, 5 e 7 e zero nos demais pontos. Tomando $Q = -150$, utilizamos o algoritmo acima para simular a absorção nesta função (figura 4.2(b)). Por fim, na figura 4.2(c), utilizamos $Q = 150$ para corrigir as perdas da figura (b). Verificamos que o resultado final é bastante parecido com a função original.

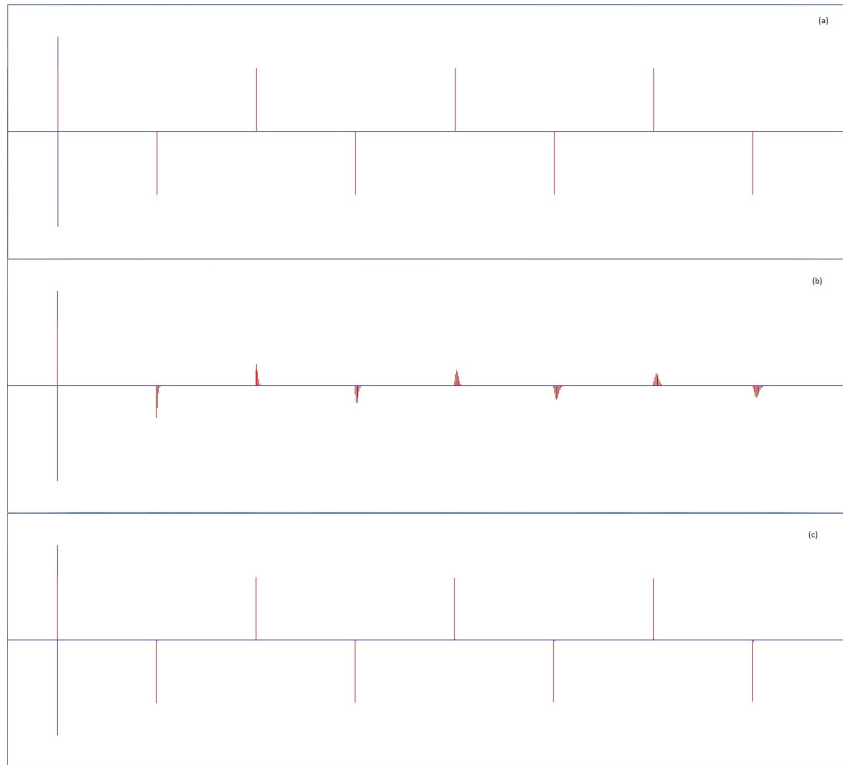


Figura 4.2: Função refletividade, simulação da absorção e correção pelo método de Duarte

Na figura 4.3, representamos a senóide $s(t) = \text{sen}(100\pi t)$ na cor vermelha, a senóide sob efeito da absorção na cor azul e a correção proposta por Duarte na cor verde. Percebemos que a curva verde não representa um ganho tão grande em relação à curva azul.

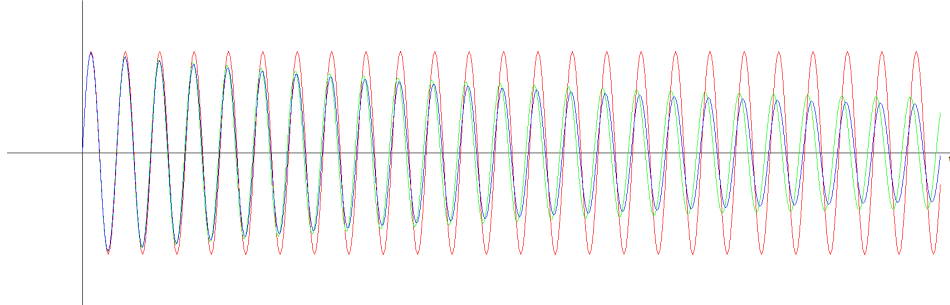


Figura 4.3: Simulação dos efeitos da absorção na senóide $s(t)$ e a correção utilizando o método de Duarte

Veremos na próxima seção que esse resultado pode melhorar bastante se alterarmos os valores de α e β .

A figura 4.4 representa o espectro de amplitude da correção de Duarte, isto é, o espectro de amplitude da curva verde da figura 4.3.

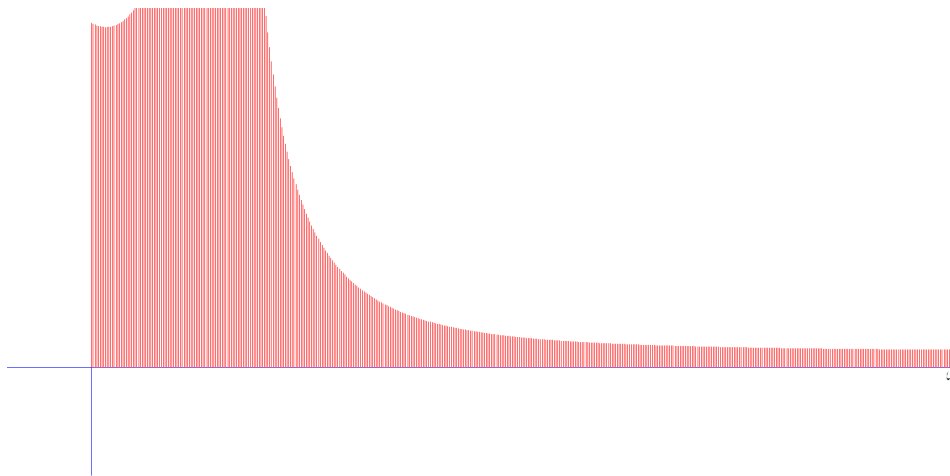


Figura 4.4: Espectro de amplitude da correção de Duarte

Agora, vamos observar o modo como M é calculado. Se queremos um ganho máximo de $GAIN$ (onde $GAIN$ medido em decibéis), temos da equação (2.25) que

$$GAIN = 20 \log_{10} \left(\frac{A_2}{A_1} \right) \quad (4.6)$$

Se $x[t]$ é o sinal de entrada, a maior razão entre as amplitudes de entrada e saída ocorrerá nas amostras a partir de M . Assim, tomamos $A_1 = x[t]$ e $A_2 = y[t] = \sum_{k=0}^M \binom{M}{k} \alpha^{M-k} \beta^k x[t-k]$.

Como $\beta = -\frac{1}{Q}$, temos que β é muito próximo de zero quando Q é grande. Então, podemos supor $A_2 = \alpha^M x[t]$. Portanto, $GAIN = 20 \log_{10} \left(\frac{\alpha^M x[t]}{x[t]} \right)$.

Segue que $M = \frac{\left(\frac{GAIN}{20} \right)}{\log_{10} \alpha}$.

Como $\alpha = 1 - \beta$ e, no caso da correção, Q é maior que zero, temos que $\alpha = 1 + \frac{1}{Q} = 1 + |\beta|$.

Deste modo, $M = \frac{\left(\frac{GAIN}{20} \right)}{\log_{10}(1 + |\beta|)}$.

Por fim, observamos que o método de Duarte é um filtro recursivo (IIR) transladado. De fato, se considerarmos $r[n]$ com $N+1$ amostras tal que $r[0] = 0$, $r[1] = x[0]$ e $r[k] = \alpha r[k-1] + \beta r[k-2]$ para todo $k \geq 2$, obtemos uma forma muito parecida com equação (2.23) onde $a_k = 0$ para todo k , $b_1 = \alpha$ e $b_2 = \beta$. Por fim, para obter o algoritmo de Duarte, basta tomar $y[n] = z[n+1]$.

Observação: Embora a equação $y[n] = \sum_{i=0}^n \binom{n}{i} \alpha^{n-i} \beta^i x[n-i]$ seja muito parecida com a equação que define um filtro FIR, o algoritmo de Duarte não é FIR uma vez que os coeficientes b_k da equação (2.22) são fixos e isso não ocorre na equação acima, pois $\binom{n}{i}$ depende da amostra n .

4.2

Comparação com o método de Varela

Se $M = N$, o algoritmo de Duarte fica muito parecido com o algoritmo de Varela. De fato, se tomarmos a equação (3.9) do algoritmo de Varela truncada no segundo termo, teremos:

$$U(\Delta\tau, f) = 1 + \frac{\pi\Delta\tau}{Q} G(f) \quad (4.7)$$

Tomamos também $g[t]$ truncada no segundo termo, isto é:

$$g[t] = \begin{cases} \frac{1}{4} & \text{se } t = 0 \\ \frac{-2}{\pi^2} & \text{se } t = 1 \\ 0 & \text{caso contrário} \end{cases} \quad (4.8)$$

Podemos fazer isso porque, segundo Varela, os demais termos de $U(\Delta\tau, f)$ e de $g[t]$ são muito próximos de zero. Deste modo, $S(\tau + \Delta\tau, f) = S(\tau, f) \left[1 + \frac{\pi\Delta\tau}{Q} G(f) \right]$, onde S é a frequência instantânea do sinal. Sejam $x[t]$ o sinal de entrada, $X(f)$ a Transformada de Fourier de $x[t]$ e $G(f)$ a Transformada de Fourier de $g[t]$.

Quando $\tau = \Delta\tau = 0$, temos $S(0, f) = X(f)$ e segue que $s_0[t] = \mathcal{F}^{-1}\{X(f)\}[t] = x[t]$. Portanto, $s_0[0] = x[0]$, ou seja, a primeira amostra não sofre alteração.

Quando $\tau = 0$ e $\Delta\tau = 1$, obtemos $S(1, f) = S(0, f) \left[1 + \frac{\pi}{Q}G(f)\right]$, ou seja, $S(1, f) = X(f) + \frac{\pi}{Q}X(f)G(f)$. Assim, $s_1[t] = \mathcal{F}^{-1}\{S(1, f)\}[t] = x[t] + \frac{\pi}{Q}(x * g)[t]$.

Como $g[t] = 0$ para $t \geq 2$, então para todo $t \geq 1$, $s_1[t]$ é dada por:

$$\begin{aligned} s_1[t] &= x[t] + \frac{\pi}{Q}(x[t]g[0] + x[t-1]g[1]) \\ &= \left(1 + \frac{\pi}{4Q}\right)x[t] + \left(\frac{-2}{\pi Q}\right)x[t-1] \end{aligned} \quad (4.9)$$

Em particular, $s_1[1] = \left(1 + \frac{\pi}{4Q}\right)x[1] + \left(\frac{-2}{\pi Q}\right)x[0]$.

Para $\tau = 1$ e $\Delta\tau = 1$, obtemos $S(2, f) = S(1, f) \left[1 + \frac{\pi}{Q}G(f)\right]$.

Assim, podemos calcular $s_2[t] = \mathcal{F}^{-1}\{S(2, f)\}[t]$ do seguinte modo:

$$\begin{aligned} s_2[t] &= s_1[t] + \frac{\pi}{Q}(s_1[t]g[0] + s_1[t-1]g[1]) \\ &= s_1[t] \left(1 + \frac{\pi}{4Q}\right) + s_1[t-1] \left(\frac{-2}{\pi Q}\right) \\ &= \left(1 + \frac{\pi}{4Q}\right)^2 x[t] + 2 \left(1 + \frac{\pi}{4Q}\right) \left(\frac{-2}{\pi Q}\right) x[t-1] + \left(\frac{-2}{\pi Q}\right)^2 x[t-2] \end{aligned} \quad (4.10)$$

Tomando $\gamma = \left(1 + \frac{\pi}{4Q}\right)$ e $\lambda = \left(\frac{-2}{\pi Q}\right)$, temos que

$$s_2[2] = \gamma^2 x[2] + 2\gamma\lambda x[1] + \lambda^2 x[0] \quad (4.11)$$

Seguindo com o raciocínio $S(\tau, f) = S(\tau-1, f) \left[1 + \frac{\pi}{Q}G(f)\right]$ e tomando $s_k[k]$ como o sinal corrigido no instante $\tau = k$, temos que

$$c[t] = \sum_{i=0}^t \binom{t}{i} \gamma^{t-i} \lambda^i x[t-i] \quad (4.12)$$

onde c é o sinal corrigido e $c[t] = s_t[t]$.

Portanto, o algoritmo de Varela, com as modificações em $U(\Delta\tau, f)$ e em $g[t]$, é equivalente ao algoritmo de Duarte quando $M = N$, $\alpha = \gamma$ e $\beta = \lambda$.

Quando simulamos a absorção com o método apresentado por Romanelli Rosa e depois corrigimos com o método de Duarte, percebemos que a correção não é tão boa quanto gostaríamos.

Por outro lado, quando alteramos os valores de α e β para $\left(1 + \frac{\pi}{4Q}\right)$ e

$\left(\frac{-2}{\pi Q}\right)$ respectivamente, conseguimos um resultado bem melhor. A figura 4.5 representa a senóide $s(t) = \sin(100\pi t)$ em vermelho, os efeitos da absorção na cor azul e a correção com a alteração dos valores de α e β na cor verde.

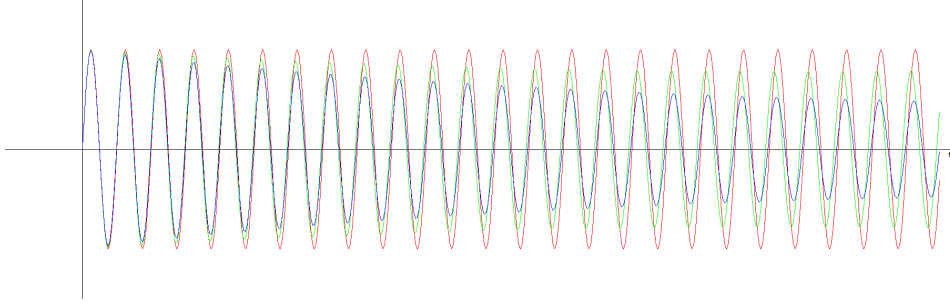


Figura 4.5: Simulação dos efeitos da absorção em uma senóide e a correção utilizando o método de Duarte com alteração de α e β

Percebemos que houve um ganho significativo na correção a partir da mudança nos valores das variáveis α e β , chegando mais próximo da senóide original. A figura 4.6 compara as figuras 4.3 e 4.5, isto é, compara o método de Duarte com $\alpha = 1 - \beta$ e $\beta = -\frac{1}{Q}$ e o método de Duarte com $\alpha = 1 + \frac{\pi}{4Q}$ e $\beta = -\frac{2}{\pi Q}$.

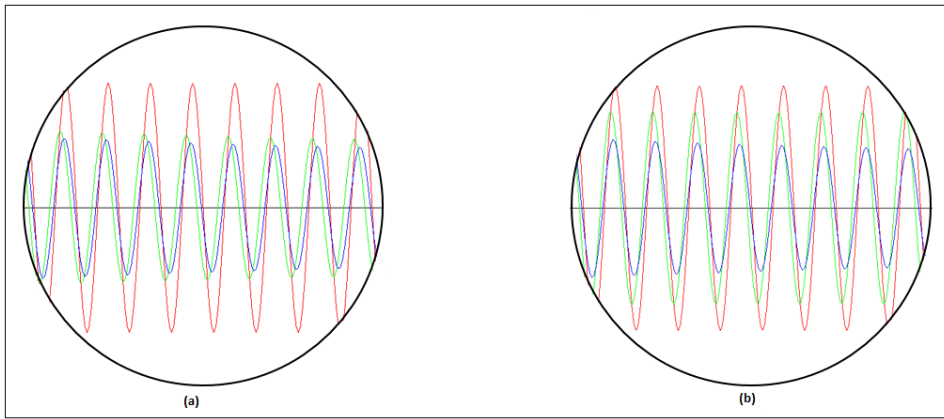


Figura 4.6: Comparação entre o método de Duarte original (a) e o método de Duarte com alteração de α e β (b)

4.3

Melhorando o desempenho do algoritmo

O algoritmo de Duarte tem ordem de complexidade $O(MN)$ e, no pior caso, isto é, quando $M = N$, temos ordem de complexidade $O(N^2)$.

Agora, vamos utilizar a transformada de Fourier para tornar o método mais eficiente. De acordo com o Teorema 2.1, a Transformada de Fourier de uma convolução de dois sinais é o produto das Transformadas de Fourier dos

sinais. Como a convolução tem ordem de complexidade $O(N^2)$ e o produto tem ordem de complexidade $O(N)$, vale a pena aplicar a Transformada de Fourier, com $O(N \ln N)$ e fazer a multiplicação termo a termo.

Porém, veremos que não é vantajoso fazer isso para todas as convoluções do algoritmo, mas apenas nas amostras $y[k]$ com $k > M$.

De acordo com a equação 4.5, para $k \leq M$, podemos calcular $y[k]$ do seguinte modo: $y[k] = \sum_{j=0}^k \binom{k}{j} \alpha^{k-j} \beta^j x[k-j]$.

Calculando a Transformada Z no instante k , obtemos:

$$\begin{aligned} Y(z) &= \sum_{j=0}^k \binom{k}{j} \alpha^{k-j} \beta^j z^{-j} X(z) \\ &= X(z) (\alpha + \beta z^{-1})^k \end{aligned} \quad (4.13)$$

Como a Transformada de Fourier é igual à Transformada Z quando $z = e^{i\frac{2\pi}{N}\omega}$, temos que $Y(\omega) = X(\omega) (\alpha + \beta e^{-i\frac{2\pi}{N}\omega})^k$, ou seja, para cada $k \in \mathbb{N}$, teríamos que calcular $(\alpha + \beta e^{-i\frac{2\pi}{N}\omega})^k$, multiplicar pela Transformada de Fourier do sinal de entrada e obter $Y(\omega)$ naquele instante. Então, teríamos que calcular a Transformada Inversa de Fourier de $Y(\omega)$ para cada k , o que se parece bastante com a Compensação Q e com a noção de frequência instantânea discutida no capítulo anterior.

Neste caso, teríamos ordem de complexidade $O(N^2 \ln N)$, que é pior que a ordem de complexidade do algoritmo de Duarte.

Porém, se $k > M$, temos que $y[k] = \sum_{j=0}^M \binom{M}{j} \alpha^{M-j} \beta^j x[k-j]$.

Vamos calcular a Transformada Z:

$$\begin{aligned} Y(z) &= \sum_{j=0}^M \binom{M}{j} \alpha^{M-j} \beta^j z^{-j} X(z) \\ &= X(z) (\alpha + \beta z^{-1})^M \end{aligned} \quad (4.14)$$

Assim, para todo $k > M$, a Transformada de Fourier do sinal de saída pode ser obtida multiplicando cada termo da Transformada de Fourier do sinal de entrada por $(\alpha + \beta e^{-i\frac{2\pi}{N}\omega})^M$, ou seja, $Y(\omega) = X(\omega) (\alpha + \beta e^{-i\frac{2\pi}{N}\omega})^M$. Deste modo, bastaria uma única Transformada Inversa de Fourier para obter $y[k]$ para todo $k > M$.

Alterando o algoritmo de Duarte a partir da amostra $M+1$, incluindo a Transformada de Fourier, o novo algoritmo terá ordem de complexidade $O(M^2)$

na primeira parte e ordem de complexidade $O(N \ln N)$ na segunda parte. Se M for suficientemente menor que N , teremos um algoritmo mais rápido.

A figura 4.7 mostra o funcionamento deste novo algoritmo proposto.

```
void duarte_modificado (double Q, double GAIN, fftw_complex *X)
{
    float alpha = 1 + PI/(4.0*Q) ;
    float beta = -2.0/(PI*Q) ;
    int M = (int) ((GAIN/20.0)/log10(1.0 + fabs(beta))) ;

    if(M >= N)
    {
        M = N ;

        float xb = X[0][0] ;

        for(int j = 0; j <= N ; ++j)
        {
            xb = X[j][0] ;

            for(int i = j ; i <= N ; ++i)
            {
                float xa = X[i][0] ;
                X[i][0] = xa*alpha + xb*beta ;
                xb = xa ;
            }
        }
    }
    else
    {
        fftw_complex X_[M] ;
        fft(N, X, X_) ;

        float xb = X[0][0] ;

        for(int j = 0; j <= M ; ++j)
        {
            xb = X[j][0] ;

            for(int i = j ; i <= M ; ++i)
            {
                float xa = X[i][0] ;
                X[i][0] = xa*alpha + xb*beta ;
                xb = xa ;
            }
        }

        fftw_complex Y_[N] ;
        for (int k = 0; k < N; ++k)
        {
            Y_[k][0] = A[k][M].real()*X_[k][0] - A[k][M].imag()*X_[k][1] ;
            Y_[k][1] = A[k][M].real()*X_[k][1] + A[k][M].imag()*X_[k][0] ;
        }

        fftw_complex Yprov[N] ;
        fft_inv(N, Y_, Yprov) ;

        for (int k = M+1; k < N; ++k)
        {
            X[k][0] = Yprov[k][0] ;
        }
    }
}
```

Figura 4.7: Algoritmo de Duarte com as modificações sugeridas

É importante ressaltar que para que esse algoritmo seja mais rápido que o algoritmo original de Duarte, é preciso que os dados referentes a $\left(\alpha + \beta e^{-i\frac{2\pi}{N}k}\right)^M$ já estejam armazenados em uma tabela no programa. Como k e M são inteiros, podemos criar uma tabela $A_{kM} = (\alpha + \beta e^{-i\frac{2\pi}{N}k})^M$ para diferentes valores de k e M .

Montar a tabela exige um esforço computacional de ordem $O(NM)$, mas esse trabalho só será feito uma vez e depois os dados serão armazenados no programa para serem apenas acessados pelo algoritmo modificado.

A figura 4.8 apresenta a senóide $s(t) = \text{sen}(100\pi t)$ em vermelho, a simulação da absorção em azul e a correção com o método de Duarte modificado da figura 4.7

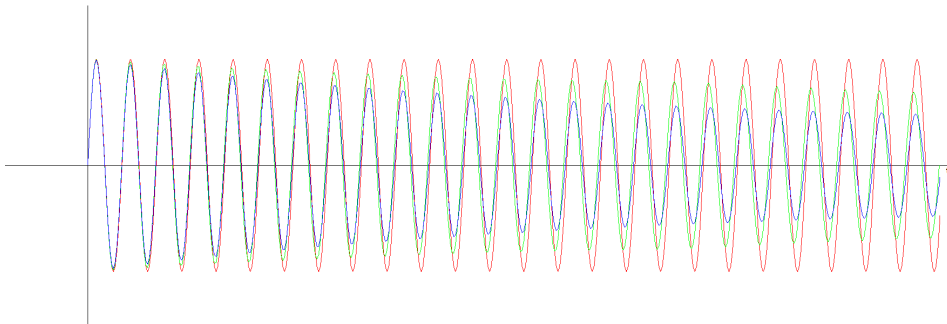


Figura 4.8: Correção da absorção com o método de Duarte modificado

A figura 4.9 compara o desempenho do método de Duarte com $\alpha = 1 + \frac{\pi}{4Q}$ e $\beta = -\frac{2}{\pi Q}$ e o método de Duarte modificado com FFT. Observamos que, de

fato, é possível tornar o algoritmo mais rápido utilizando Transformada de Fourier, porém com o mesmo resultado.

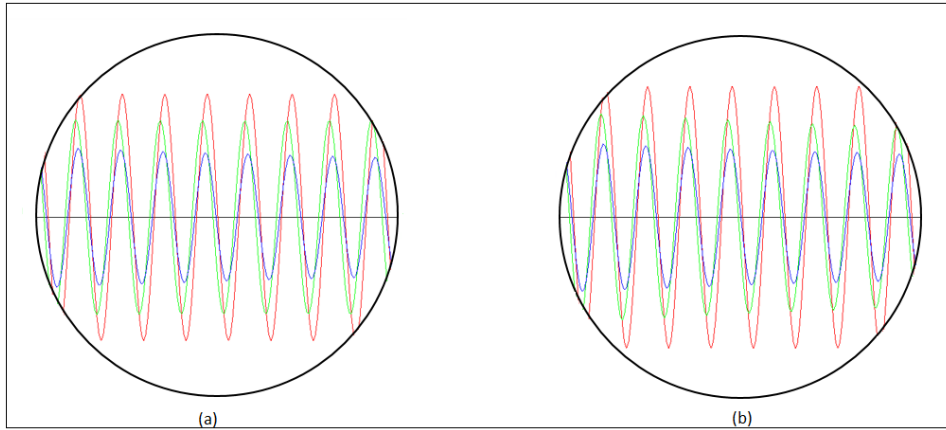


Figura 4.9: Comparação entre o método de Duarte com α e β modificados (a) e o método de Duarte com FFT (b)

A figura 4.10 representa o espectro de amplitude da correção (curva verde da figura 4.8).

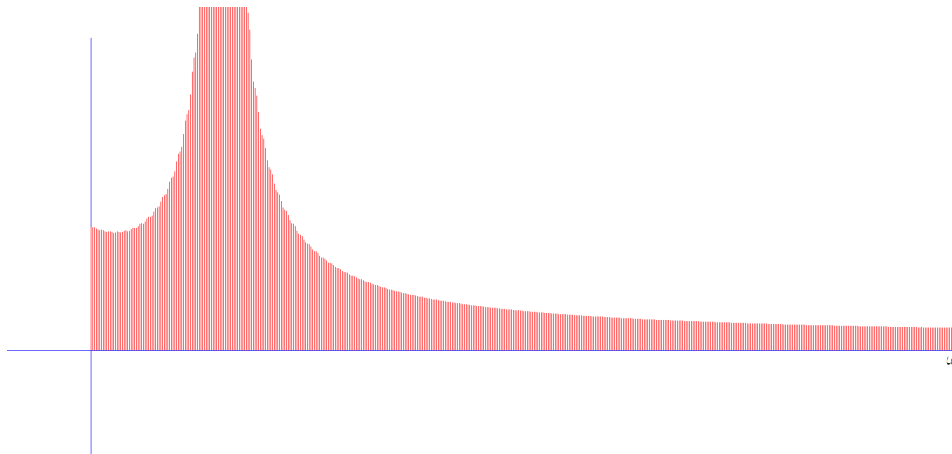


Figura 4.10: Espectro de amplitude da correção da absorção com o método de Duarte modificado

5 Experimentos

Neste capítulo, utilizamos os dados sísmicos fornecidos pela Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP) para comparar o desempenho dos diferentes métodos de correção da absorção sísmica apresentados anteriormente.

Com o objetivo de verificar a eficiência dos métodos apresentados, aplicamos a Compensação Q, o método de Varela, o método de Duarte e o método de Duarte modificado à linha sísmica 0261_VB00_2D_BM_SEAL.VB00-52.Field_Data.9.sgy.

Seguem os dados da linha utilizada:

Levantamento: VB00_2D_BM_SEAL

Segmento: VB00-52

Bacia: Sergipe-Alagoas

Período de amostragem: $4ms$

Tipo de fonte: canhão de ar

Comprimento dos registros: $12s$

Número de traços: 653.490

Número de amostras por traço: 3001

Para podermos visualizar melhor o ganho em cada modelo, ampliamos uma parte do dado e vamos comparar os resultados. Utilizamos apenas o trecho entre os CMPs 3000 e 3500 e no intervalo de tempo entre $4s$ e $9s$. Deste modo, cada traço do dado possui 1251 amostras.

O dado foi processado conforme fluxograma da figura 5.1. No caso da figura 5.2, que mostra o dado original, a etapa de correção da absorção não fez parte do processamento. O ideal seria aplicar o algoritmo de correção no começo do processamento, mas optamos por incluir esta etapa após a correção NMO para garantir que a diferença entre as figuras 5.2, 5.3, 5.4, 5.5 e 5.6 deve-se apenas aos métodos de correção e não às demais etapas do processamento.

A figura 5.3 apresenta o dado com a Compensação Q, a figura 5.4 mostra o dado com a correção de Varela, a figura 5.5, o dado com a correção de Duarte e a figura 5.6 é o dado após a aplicação do método de Duarte modificado, com $\alpha = 1 + \frac{\pi}{4Q}$, $\beta = -\frac{2}{\pi Q}$ e utilizando a Transformada de Fourier. Utilizamos

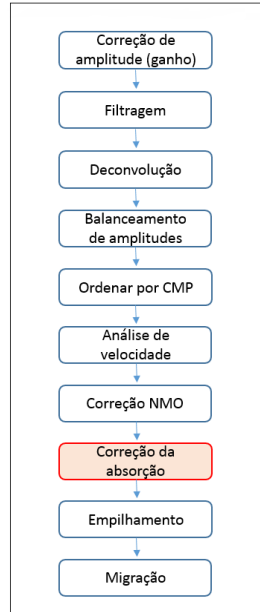


Figura 5.1: Fluxograma do processamento

$Q = 200$ para todos os casos, $M = 50$ no algoritmo de Varela e $GAIN = 60$ no algoritmo de Duarte e no algoritmo modificado.

Como o número de amostras é $N = 1251$ e a variável $GAIN$ é 60, neste caso não faz diferença utilizar o algoritmo com a Transformada de Fourier porque a variável M , que determina a etapa em que a iteração será interrompida, será maior que o número de amostras.

De fato, para $GAIN = 60$ e $Q = 200$, temos que $M = \frac{\frac{60}{20}}{\log_{10}(1 + \frac{2}{\pi Q})}$, isto é, $M = 2173 > N$.

Assim, o tempo necessário para aplicar o filtro no dado será o mesmo que o algoritmo original de Duarte. A única diferença será nos valores de α e β .

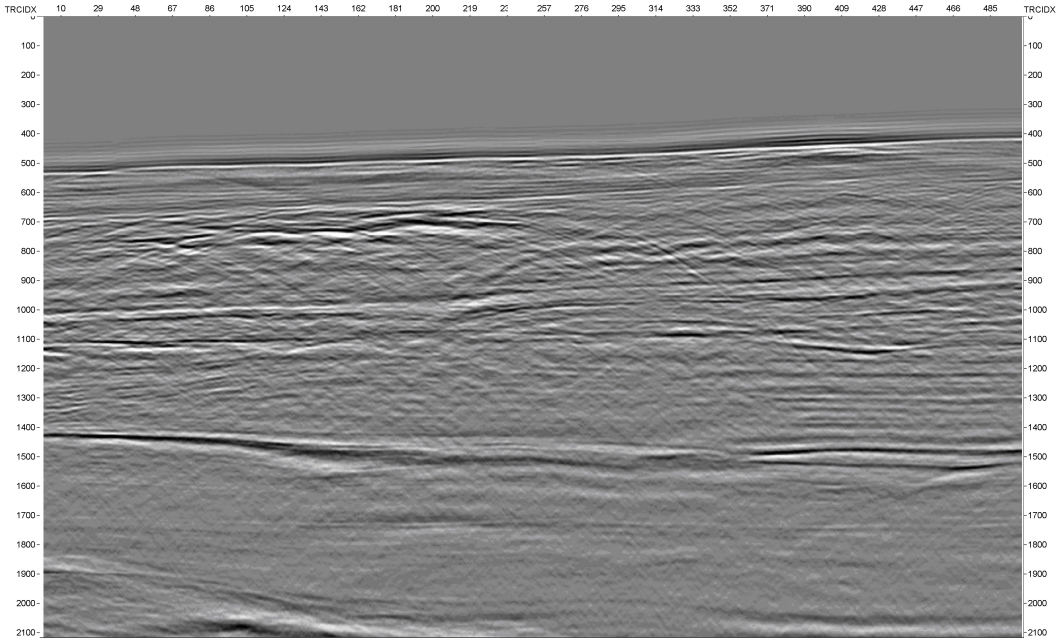


Figura 5.2: Dado original

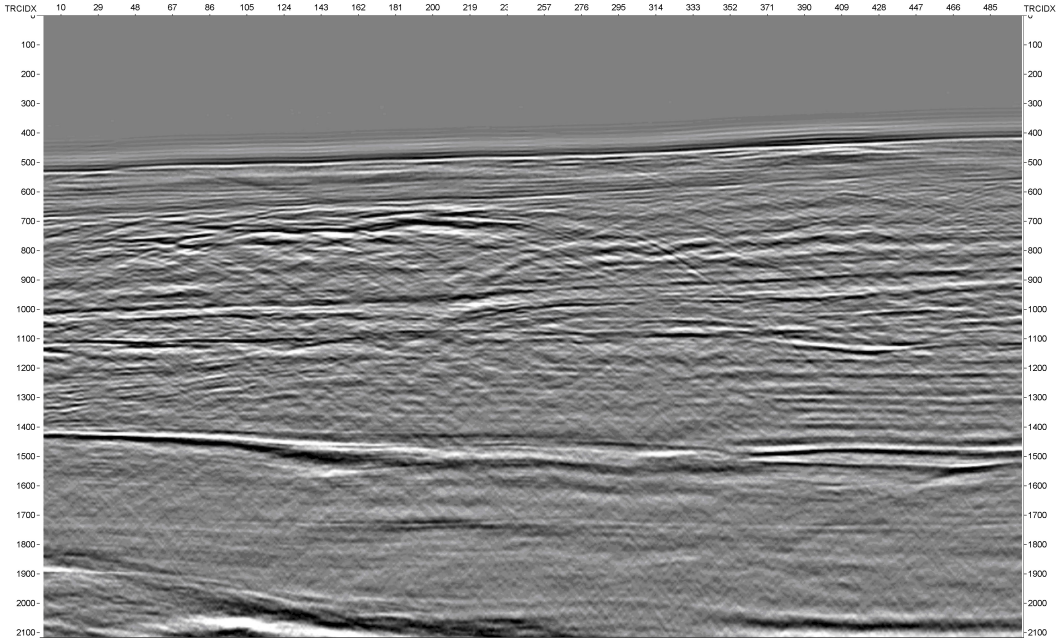


Figura 5.3: Dado com aplicação da Compensação Q com $Q = 200, 0$

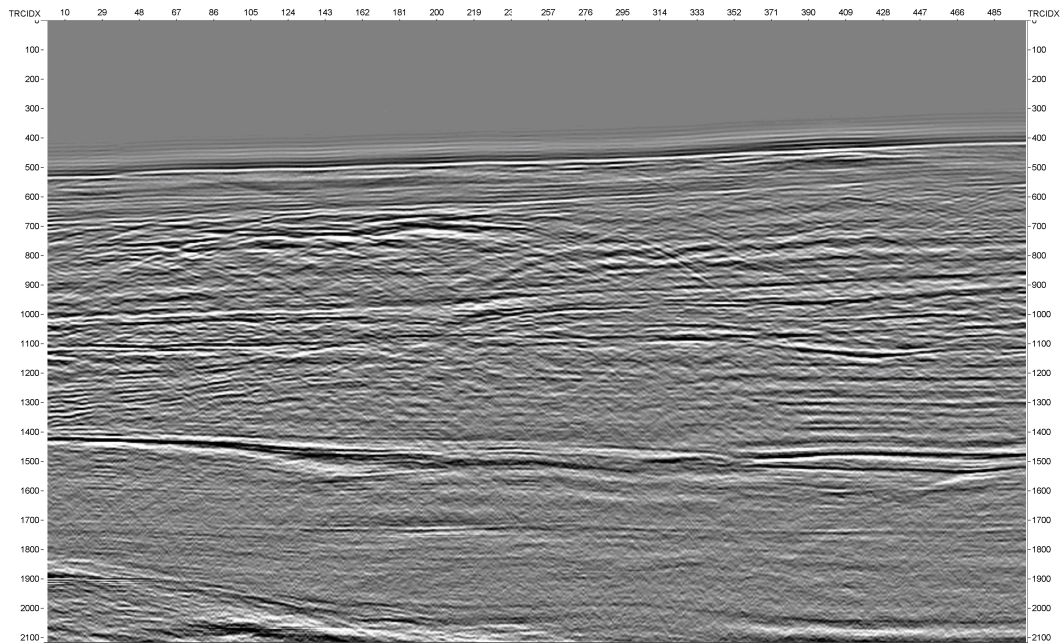


Figura 5.4: Dado com aplicação do método de Varela com $Q = 200,0$ e $M = 50$

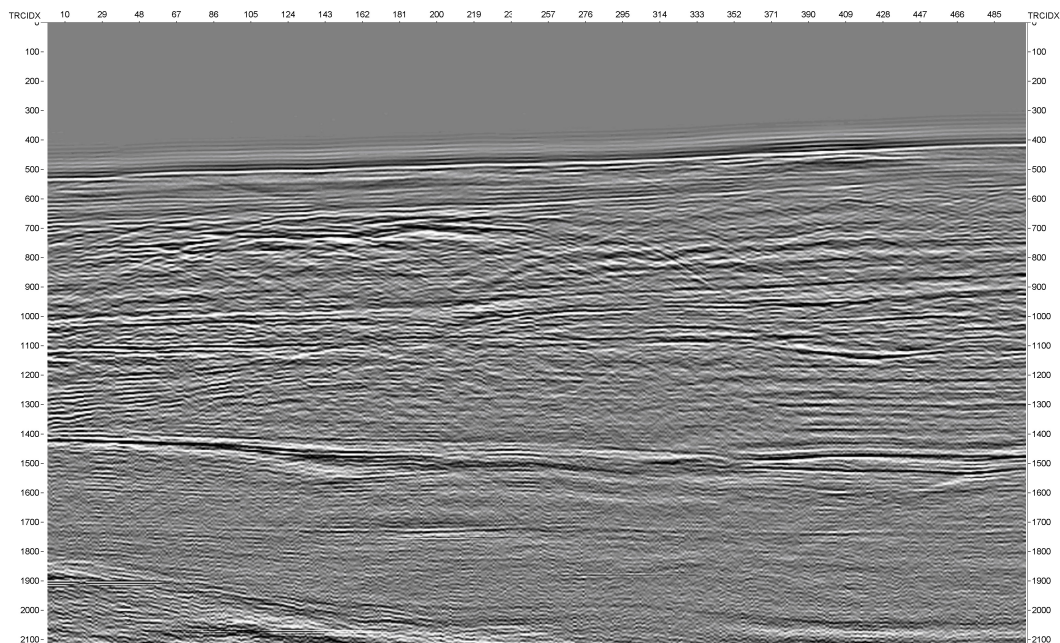


Figura 5.5: Dado com aplicação do método de Duarte com $Q = 200,0$ e $GAIN = 60,0$

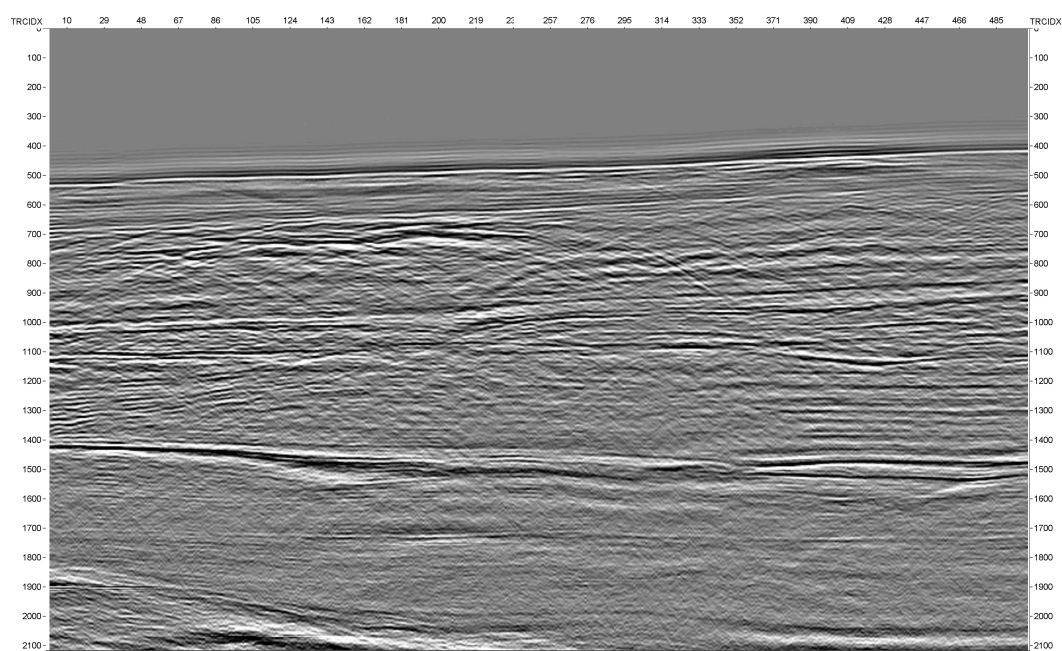


Figura 5.6: Dado com aplicação do método de Duarte modificado com $Q = 200,0$ e $GAIN = 60,0$

Observamos que os métodos apresentaram bons resultados, com aumento significativo da resolução das camadas de subsuperfície.

Agora, vamos avaliar os métodos do ponto de vista computacional. Percebemos que o método de Duarte é muito mais rápido que a Compensação Q e que o método de Varela. A fim de diferenciar o método de Duarte do método modificado proposto no capítulo anterior, utilizaremos $GAIN = 20$.

$$\text{Para } Q = 200 \text{ e } GAIN = 20, \text{ temos } M = \frac{\frac{20}{20}}{\log_{10}(1 + \frac{2}{\pi Q})} = 724 < N$$

A tabela 5.1 mostra o tempo gasto para aplicar cada uma das correções aos primeiros 500 traços do dado sísmico (dado com tamanho de 2,5 MB).

Método de correção	Tempo (s)
Compensação Q	350,86
Varela	107,61
Duarte	0,96
Duarte modificado	0,89

Tabela 5.1: Tempo gasto para aplicar as correções

Embora todos os métodos de correção da absorção apresentem um bom resultado, é preciso usá-los com cuidado. Por se tratar de filtros passa-alta, dependendo do valor de Q e do ganho $GAIN$, podemos destruir o dado, pois o ruído também terá um ganho.

6

Conclusão

Com base nos experimentos do capítulo 5, podemos concluir que a Compensação Q , o método de Varela e o método de Duarte apresentaram bons resultados quando aplicados em dados reais, aumentando a resolução da seção sísmica. Porém, do ponto de vista computacional, o método de Duarte mostrou-se bem mais rápido que os demais. Além disso, quando alteramos o algoritmo de Duarte, aplicando a Transformada de Fourier conforme a seção 4.3, conseguimos torná-lo ainda mais rápido.

Neste trabalho utilizamos o mesmo fator Q para todo o dado. Porém, as rochas possuem propriedades diferentes e o fator Q correspondente deve variar de acordo com as camadas de rochas por onde a onda se propaga. Deste modo, deixamos como trabalho futuro a alteração do algoritmo de Duarte de modo a permitir a variação do fator Q . A ideia é separar o algoritmo em janelas e em cada uma dessas janelas, utilizar o valor do fator Q mais adequado.

Referências Bibliográficas

- [1] COHEN, J. K.; STOCKWELL, J. J. W.. **Cwp/su: Seismic un*x: an open source software package for seismic research and processing**, 2007. Center for Wave Phenomena, Colorado School of Mines.
- [2] DUARTE, O. O.. **Um método simples para a correção das perdas por absorção**. Anais do III Congresso Internacional da Sociedade Brasileira de Geofísica, p. 226–231, 1993.
- [3] DUARTE, O. O.. **Dicionário enciclopédico inglês-português de geofísica e geologia**. SBGf, 2012. Quinta Edição.
- [4] FOREL, D.; BENZ, T. ; PENNINGTON, W. D.. **Seismic Data Processing with Seismic Unix: A 2D Seismic Data Processing Primer**. Society of Exploration Geophysicists, 2005.
- [5] HSU, H. P.. **Sinais e sistemas**. Bookman, 2012.
- [6] OPPENHEIM, A. V.; S.WILLSKY, A. ; NAWAB, H.. **Signals & Systems**. Hardcover, 1996. Second Edition.
- [7] PALERMO, L. A. C.. **Migração reversa no tempo: uma abordagem multifocal**. Dissertação de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2002. (em Português).
- [8] ROSA, A. L. R.. **Análise do sinal sísmico**. Sociedade Brasileira de Geofísica, 2010.
- [9] SILVA, P. M. C. E.. **Visualização volumétrica de horizontes em dados sísmicos 3D**. Tese de doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004. (em Português).
- [10] SMITH, S. W.. **The scientist and engineer's guide to digital signal processing**. California Technical Publishing, 1999. Second Edition.
- [11] TOMEI, C.. **Computação científica e equações diferenciais**, 2013. notas de aula.

- [12] VARELA, C. L.; ROSA, A. L. R. ; ULRYCH, T. J.. **Modeling of attenuation and dispersion**. Geophysics, 58:1167–1173, 1993.
- [13] YILMAZ, O.. **Seismic data analysis: processing, inversion and interpretation of seismic data**. Society of Exploration Geophysicists, 2001. Second Edition.

A

Apêndice

No capítulo 5, foram apresentados dados sísmicos processados. Agora, vamos apresentar o modo como esses dados foram processados no *software Seismic Unix* e o passo a passo para chegar às imagens que mostramos anteriormente.

A linha sísmica utilizada neste trabalho é a 0261_VB00_2D_BM_SEAL.VB00-52.Field_Data.9.sgy, que foi fornecida pela Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP). O levantamento sísmico foi realizado na Bacia Sergipe-Alagoas, no nordeste do Brasil.

A.1

Mudança de formato

Em geral, os dados sísmicos são gravados em formato SEG Y (.sgy), porém o *Seismic Unix* utiliza um formato próprio (.su). Assim, o primeiro passo necessário para trabalhar com o dado é alterar o formato de sgy para su.

Para isso, utilizamos o comando abaixo:

```
segypread tape=dado.sgy verbose=1 endian=0 | segyclean >dado.su
```

Este comando gera o dado *dado.su*, que utilizaremos a partir de agora.

A.2

Visualização do dado

Para visualizar o dado, usamos o comando **suximage** ou o comando **suxwigb**:

```
suximage perc=99 <dado.su suxwigb perc=99 <dado.su
```

Se quisermos visualizar apenas uma parte do dado, por exemplo, do traço de número 1 ao de número 2000, utilizamos os comandos acima combinados com o comando **suwind**, conforme abaixo:

```
suwind <dado.su key=trac1 min=1 max=2000 | suximage perc=99
```

Neste trabalho, porém utilizamos outro *software* para visualização dos dados, o *SeiSee*. A figura A.1 mostra os primeiros 2000 traços.

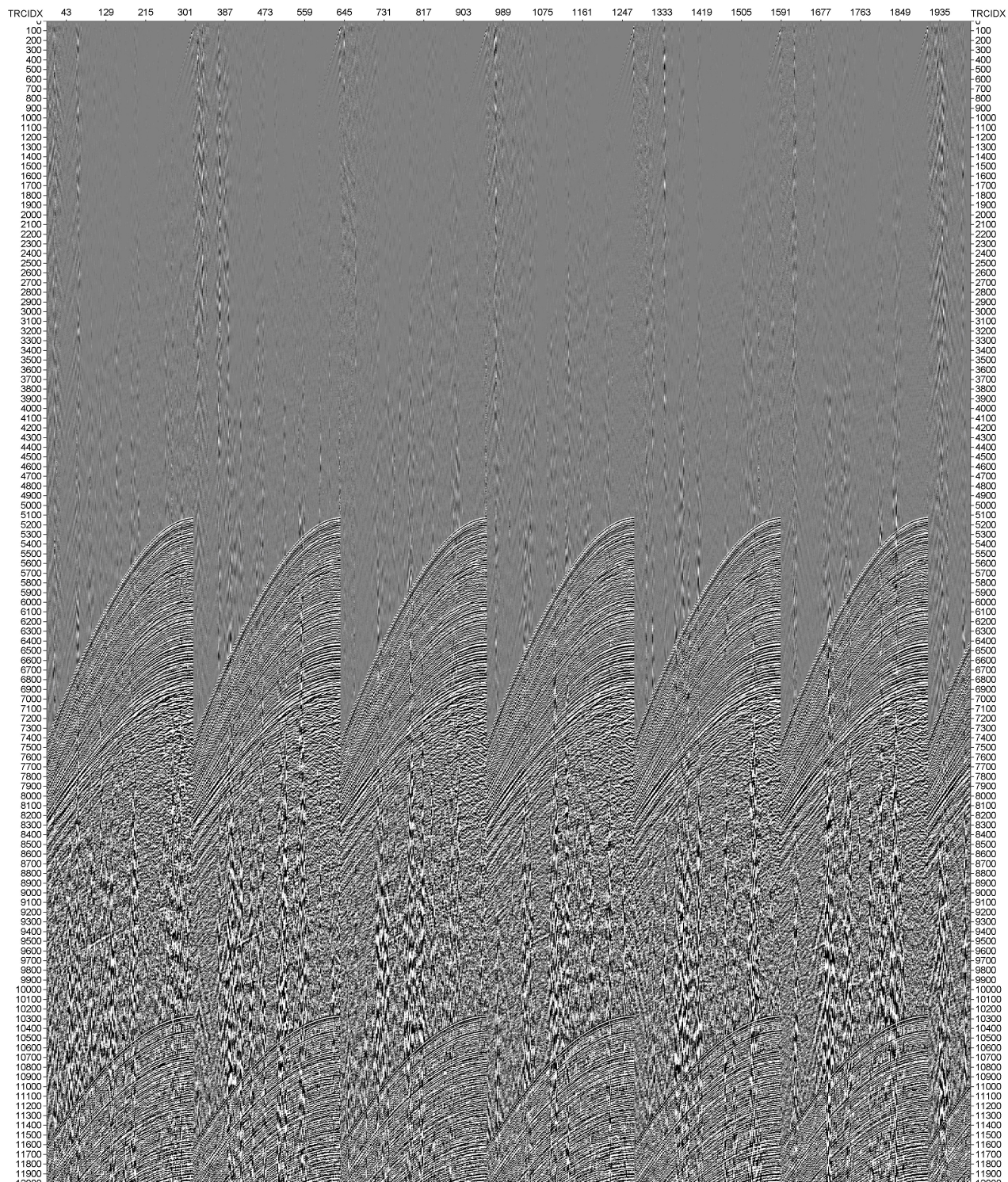


Figura A.1: Traços 1 a 2000 do dado sísmico

A.3

Correção de amplitude

O dado sísmico sofre uma perda de amplitude com o tempo. Essa perda está relacionada a vários fatores, tais como a absorção sísmica, discutida nos capítulos anteriores, e a divergência esférica e as reflexões parciais.

A divergência esférica é a causa da perda de energia e, conseqüentemente, de amplitude relacionada ao fato de que a área da frente de onda aumenta com o tempo, fazendo com que a energia por unidade de área diminua.

As reflexões parciais que ocorrem ao longo da propagação da onda também contribuem para a redução da energia, que acaba sendo dividida entre a onda refratada e a onda refletida.

A fim de minimizar este problema, aplicamos um ganho de amplitude. Existem diferentes métodos de ganho. Aqui, usamos o método t^2 , onde a cada instante t , a amplitude é multiplicada por t^2 .

O comando do *Seismic Unix* para fazer essa correção é:

```
sugain <dados.su tpow=2.0 >dados_gain.su
```

A.4

Filtragem

Em geral, utiliza-se um filtro passa faixa porque os dados sísmicos possuem ruídos de baixa frequência, como o *ground roll* na superfície, e de alta frequência. Segundo Yilmaz [13], a faixa de frequência de dados sísmicos normalmente está compreendida entre 10 e 70Hz.

Utilizamos o seguinte comando para aplicar o filtro passa faixa de quatro pontos:

```
sufilter <dados_gain.su f=10,20,70,80 amps=0,1,1,0 >dados_filter.su
```

A.5

Deconvolução

A deconvolução é utilizada para aumentar a resolução, dentre outras coisas, removendo ruídos como as múltiplas. As múltiplas ocorrem quando a onda é refletida no mais de uma vez no mesmo refletor, em especial quando a onda está sendo refletida em um outro refletor mais abaixo.

No *Seismic Unix*, utilizamos o seguinte comando:

```
supef <dados_filter.su minlag=0.03 maxlag=0.5 pnoise=0.001  
mincorr=0 maxcorr=0.4 >dados_dec.su
```

A.6

Balanceamento de amplitudes

Segundo Yilmaz [13], após a deconvolução, devemos balancear as amplitudes. O balanceamento é um escalonamento invariante no tempo.

Fazemos isso usando o mesmo comando **sugain** visto anteriormente, porém com outros parâmetros.

```
sugain <dados_dec.su qbal=1 >dados_bal.su
```

A.7

Ordenar o dado por CMP

Durante a aquisição sísmica, o navio se move, idealmente entre dois disparos do canhão de ar, alterando a posição da fonte e dos receptores. Assim, um mesmo ponto de reflexão da onda é amostrado várias vezes. Esse ponto é chamado de *Common-Midpoint* (CMP) ou *Common-Depth-Point* (CDP) e está localizado no ponto médio entre o receptor e a fonte, na profundidade do refletor.

A figura A.2 representa um esquema da aquisição sísmica com as posições da fonte, dos receptores e dos CMPs.

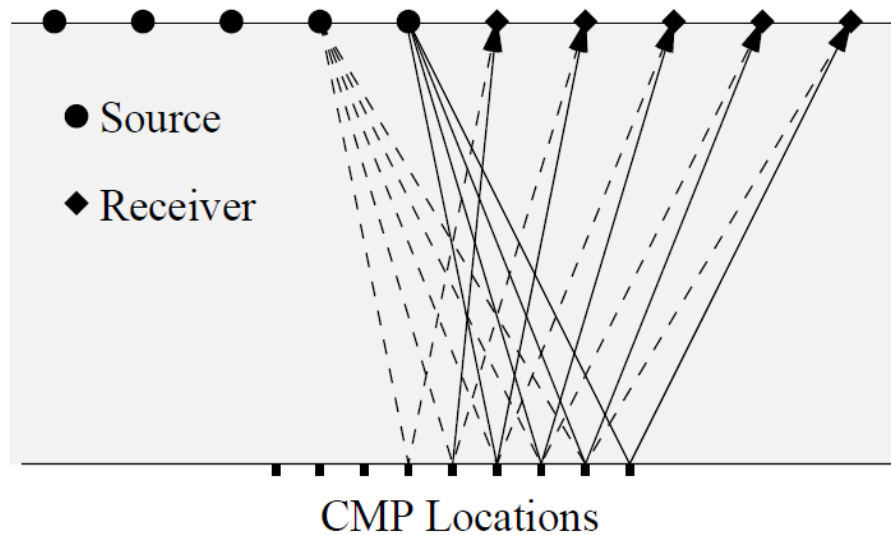


Figura A.2: *Common-Midpoints*

Assim, teremos vários traços que representam o mesmo CMP. Cada traço terá informações diferentes sobre este ponto, uma vez que possuem diferentes *offsets* e tempo de propagação. *Offset* é a distância entre a fonte e o receptor que registrou o sinal de retorno.

Agora, vamos organizar o dado primeiro por ordem de CMP e depois de *offset*. O *Seismic Unix*, utiliza a notação CDP.

```

susort <dados_bal.su >dados_cdp.su cdp offset

```

A figura A.3 mostra os traços de número 1 a 2000 após a ordenação por *Common-Midpoint*.

PUC-Rio - Certificação Digital Nº 1312483/CB

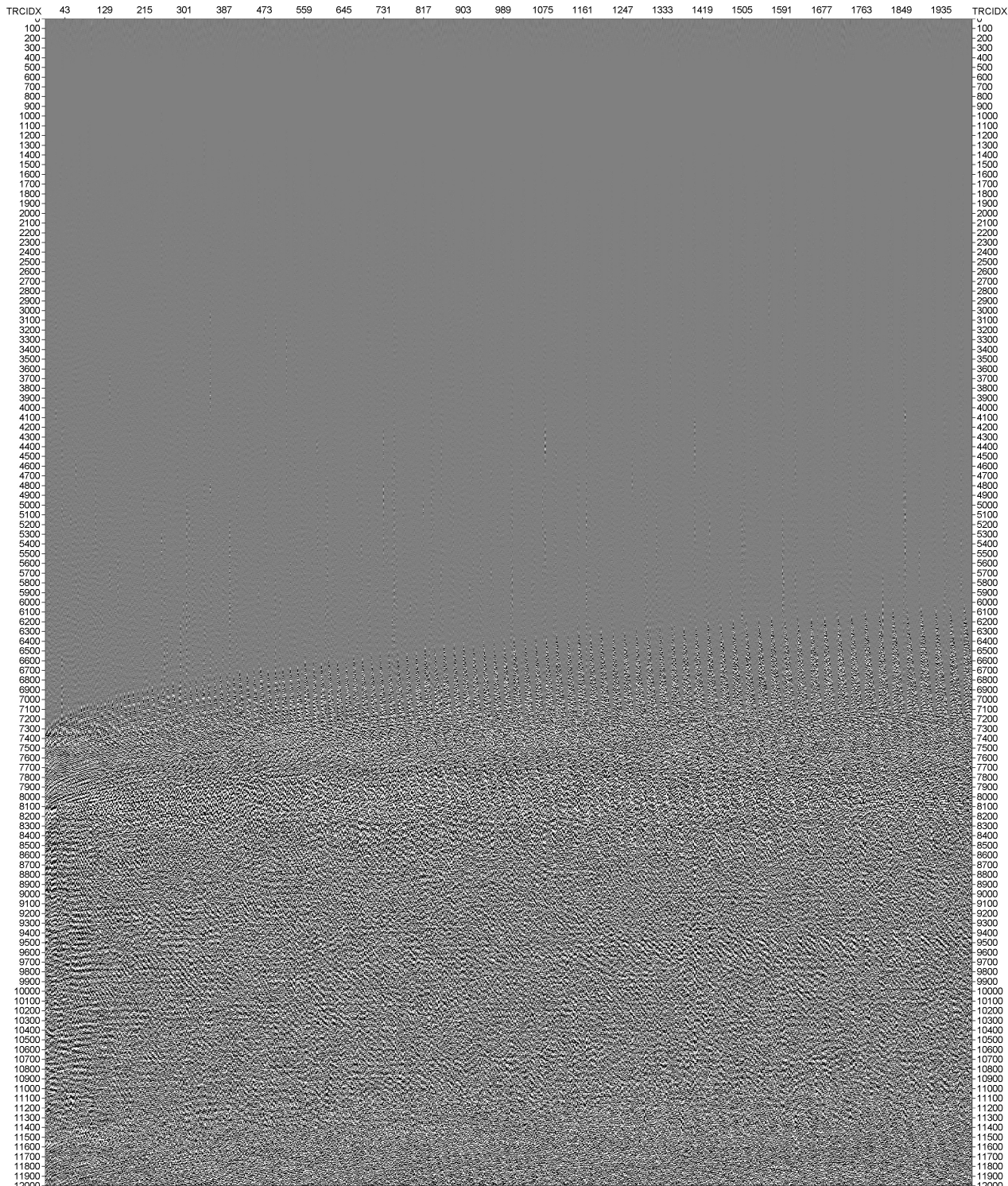


Figura A.3: Traços 1 a 2000 do dado sísmico após ordenar por CMP

A.8

Análise de velocidade e correção NMO

Dado um grupo de traços com o mesmo CMP, o tempo de propagação da reflexão, considerado como função do *offset*, segue trajetórias em formato de hipérbole. A correção NMO (*Normal Moveout*) tem como objetivo eliminar as hipérbolas, eliminando o efeito do *offset*.

A correção NMO utiliza um campo de velocidades para corrigir os traços. Esse campo de velocidade é obtido a partir da análise de velocidade. Neste trabalho, utilizamos o script **iva** apresentado por Benz [4] para plotar o *semblance* para alguns grupos de CMP (figura A.4) e escolher os pontos de maior coerência entre tempo e velocidade.

A seguir, usamos o seguinte comando para aplicar a correção NMO.

```
sunmo <dado_cdp.su par=vpick >dado_nmo.su
```

No caso acima, *vpick* é um arquivo de texto gerado durante a análise de velocidade. Este arquivo contém os números dos CMPs para os quais foi feita a análise, os tempos e as velocidades correspondentes aos pontos selecionados no *semblance*.

A figura A.5 mostra a os traços da figura A.3 após a correção NMO.

A.9

Empilhamento

Nesta etapa, os traços referentes a um mesmo CMP são somados, gerando um único traço. A amplitude do traço resultante é dividida pelo número de traços que foram somados. Esse processo é chamado de empilhamento e o comando utilizado é:

```
sustack <dado_nmo.su >dado_stk.su
```

A figura A.6 mostra o dado sísmico após o empilhamento.

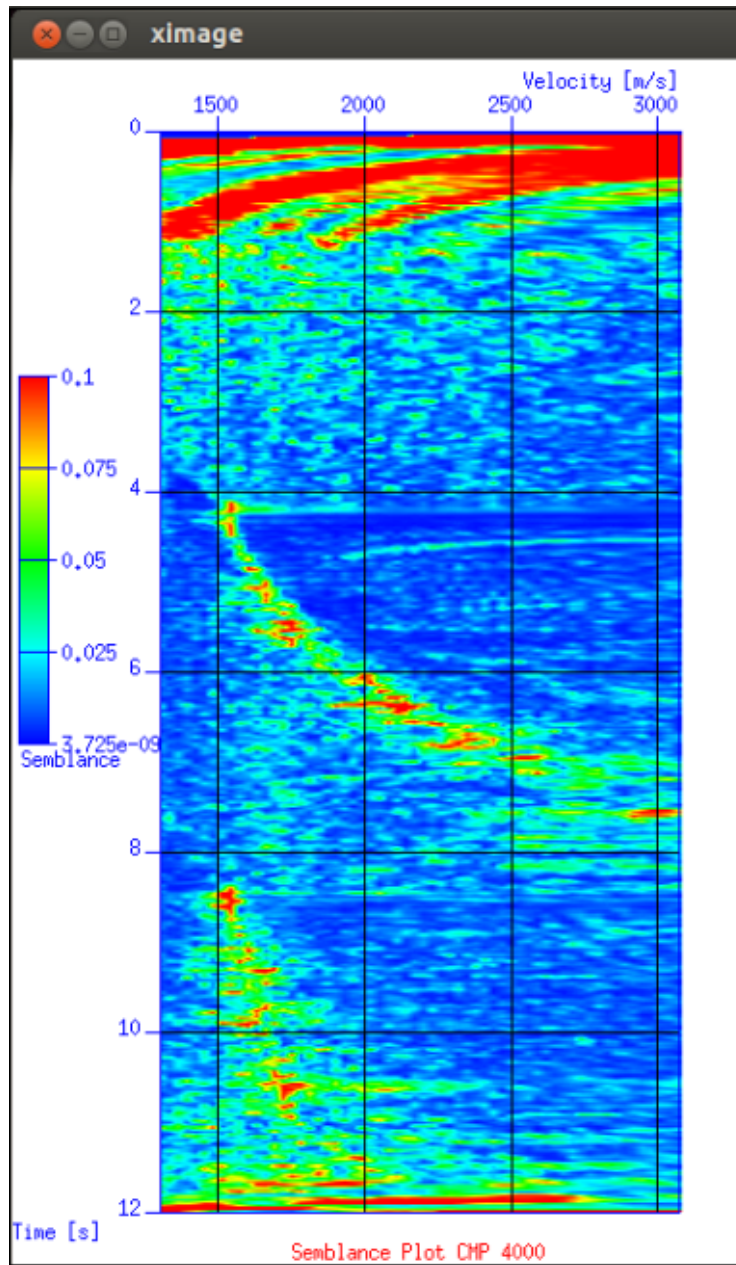


Figura A.4: *Semblance* do CMP 4000

PUC-Rio - Certificação Digital Nº 1312483/CB

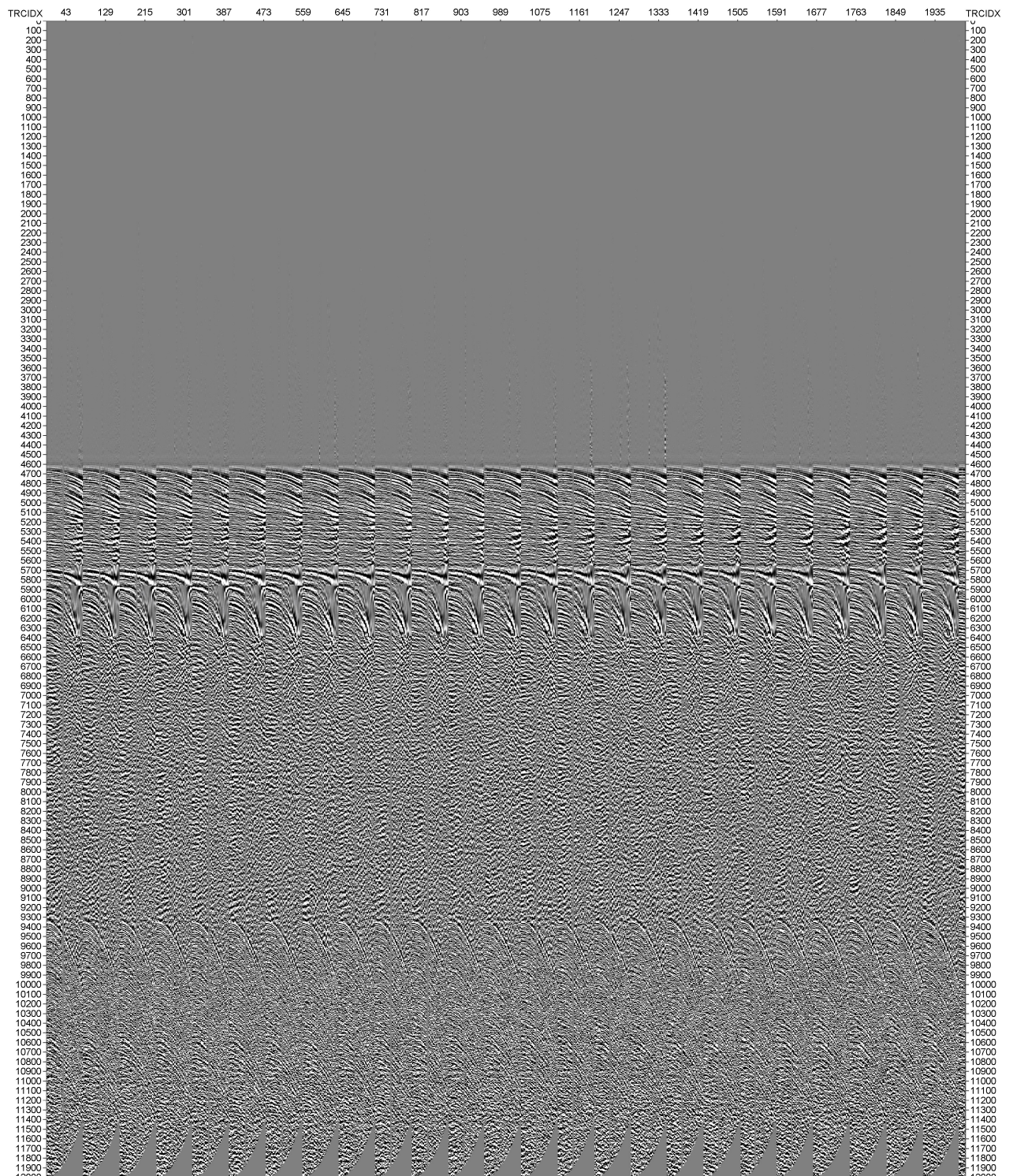


Figura A.5: Traços 1 a 2000 do dado sísmico após a correção NMO

PUC-Rio - Certificação Digital Nº 1312483/CB

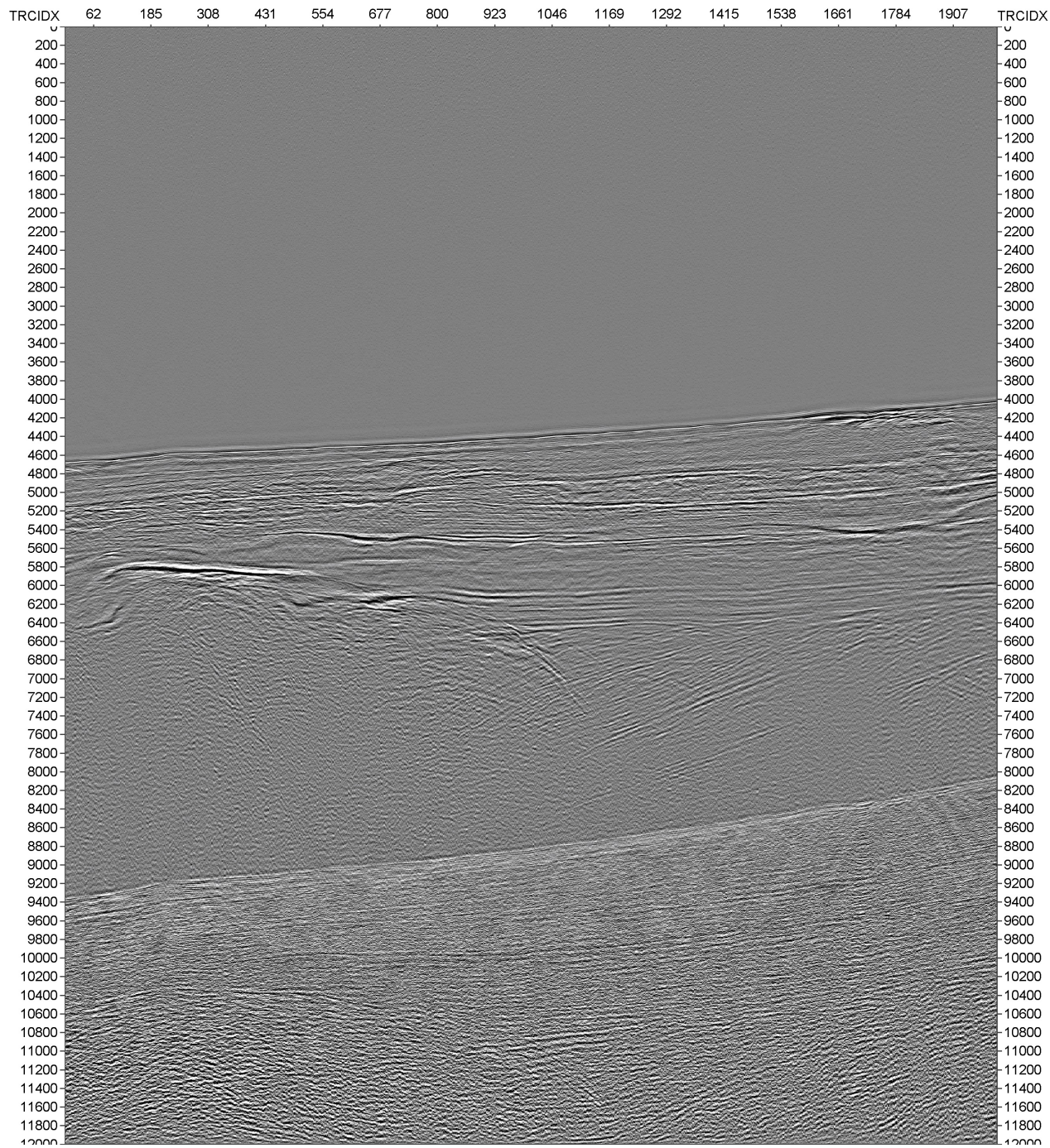


Figura A.6: Dado após o empilhamento

A.10 Migração

Quando calculamos a posição do CMP, consideramos que o refletor é horizontal. Porém, na maioria das vezes, isso não acontece. Deste modo, a posição real do CMP é diferente da que foi calculada. A migração tem como objetivo corrigir esse dado.

O comando utilizado é:

```
sumigps <dado_stk.su tmig=4.18563, 4.49845, 4.90809, 5.42945,  
5.71248, 6.13701, 6.27108, 6.36045, 6.44238, 6.62858 vmig=1440.2,  
1498.42, 1556.64, 1715.42, 1741.89, 1916.55, 1980.07, 2080.63,  
2149.44, 2276.47 dx=12.5 >dado_mig.su
```

Os parâmetros tmig e vmig são os tempos e as velocidades selecionadas na análise de velocidade com uma redução de 5% na velocidade. O parâmetro dx é a distância entre os CMPs. Como a distância entre os receptores é de 25 metros, a distância entre os CMPs é de metade desse valor.

A figura A.7 mostra a seção sísmica empilhada da figura A.6 após a migração.

PUC-Rio - Certificação Digital Nº 1312483/CB

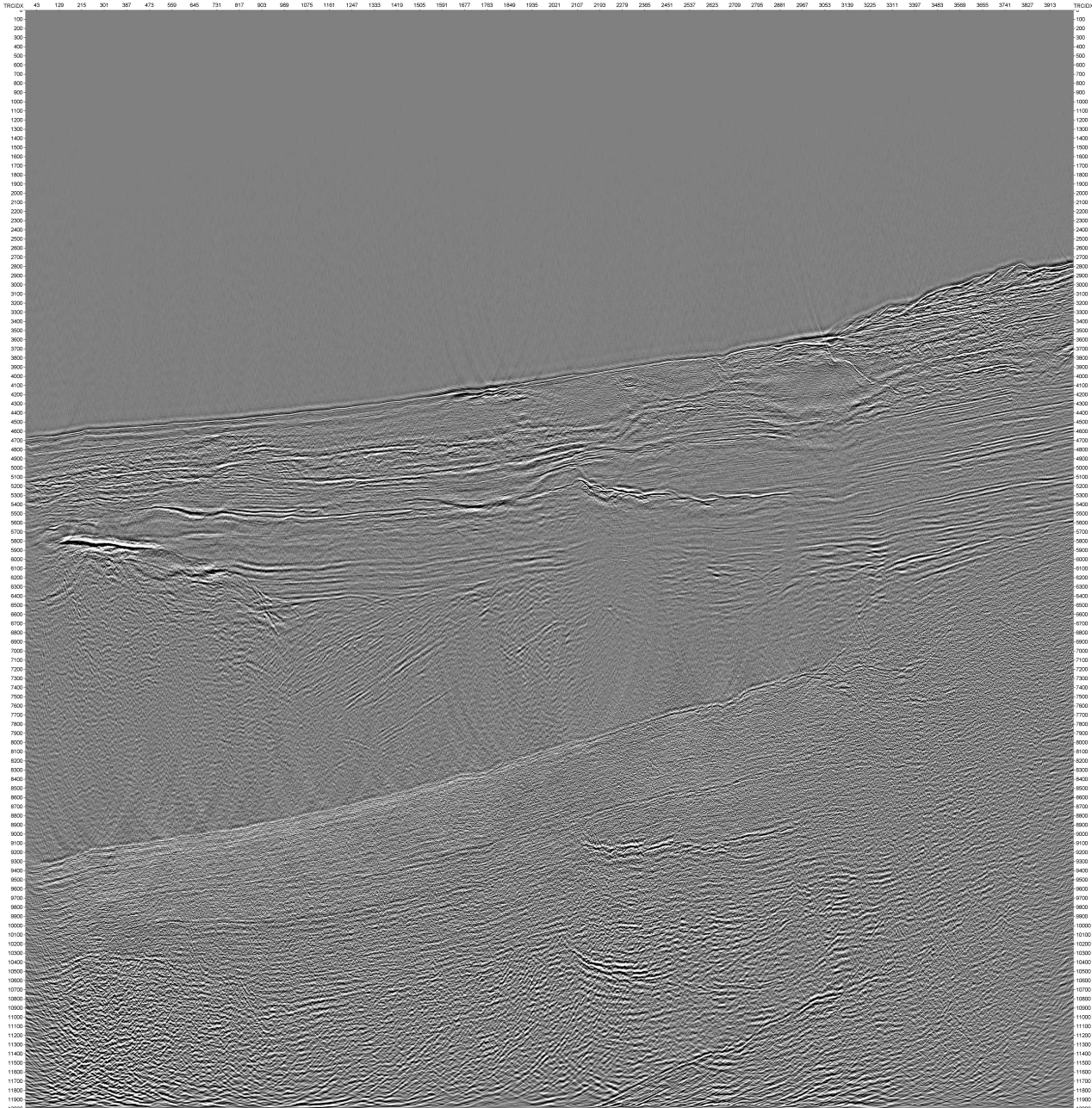


Figura A.7: Dado após a migração