



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

RELATÓRIO DE PROCESSAMENTO SÍSMICO

Processamento sísmico com dado sintético

LUCAS DE CASTRO COSTA

Belém
2021

Sumário

1	OBJETIVO	1
2	PARTE TEÓRICA	2
2.1	MÉTODO SÍSMICO DE REFLEXÃO	2
2.2	MODELAGEM DA ESTRUTURA GEOLÓGICA	2
2.3	PARÂMETROS DE AQUISIÇÃO	3
2.4	PRÓXIMOS PASSOS DO PROCESSAMENTO SÍSMICO	4
3	METODOLOGIA	5
3.1	CONSTRUINDO MODELO DE CAMADAS PLANAS E AQUISIÇÃO .	5
4	RESULTADOS E DISCUSSÕES	7
4.1	ANÁLISE DE VELOCIDADE	7
4.2	APLICANDO RUÍDO NOS DADOS	7
4.3	APLICANDO CORREÇÕES	7
5	CONCLUSÃO	18
	REFERÊNCIAS	19
	APÊNDICES	20
A–	SHELL SCRIPT MODELAGEM TRIMODEL	21
B–	SHELL SCRIPT AQUISIÇÃO TRISEIS	23
C–	SHELL SCRIPT CDP	26
D–	SHELL SCRIPT RUIDO DE GAUSS	27
E–	SHELL SCRIPT ANÁLISE DE VELOCIDADE	28
F–	SHELL SCRIPT CORREÇÃO NMO	32
G–	SHELL SCRIPT EMPILHAMENTO	36
H–	SHELL SCRIPT MIGRAÇÃO NO TEMPO	37

1 OBJETIVO

O objetivo geral deste trabalho é realizar passos de um processamento sísmico. Para isso, será utilizado um modelo sintético e os seguintes passos serão reproduzidos: aquisição, CDP, modelo de velocidade, correção NMO, empilhamento e, por fim, migração temporal. Esses procedimentos serão realizados a partir do pacote *Seismic Unix*. Com isso, será possível desenvolver melhor o conhecimento sobre os passos do processamento sísmico e adquirir maior familiaridade com a parte de programação.

2 PARTE TEÓRICA

2.1 MÉTODO SÍSMICO DE REFLEXÃO

A teoria que descreve a propagação de uma onda mecânica é a base da sismologia e sísmica de exploração. Essa teoria está baseada na resolução das equações diferenciais parciais desde o modelo mais simples – onda acústica em um meio com densidade constante – até o modelo com uma complexidade maior. Saber a solução da equação da onda para um dado conjunto ou distribuição de fontes em um dado meio é crucial em várias aplicações dentro da sísmica de exploração: estudos de iluminação para um levantamento sísmico, amarração poço-sísmica, migração RTM, FWI, por exemplo.

O método sísmico de reflexão, consiste, basicamente, em obter informações da subsuperfície através da propagação de ondas produzidas por fontes artificiais em superfície e o posterior registro dessas ondas em receptores (geofones), que estão em superfície.

As ondas geradas pelas fontes propagam-se na subsuperfície, sendo transmitidas, refletidas e difratadas por interfaces geológicas que delimitam as camadas de rochas de diferentes propriedades físicas (Zakaria et al., 2000). As ondas registradas nos receptores são provenientes das reflexões das interfaces, o sinal registrado pelos receptores possuem informações sob as ondas propagadas e suas mudanças conforme o trajeto do percurso da onda (Yilmaz, 2001).

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

2.2 MODELAGEM DA ESTRUTURA GEOLÓGICA

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

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

década de 50 (Edwards, 1988).

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

$$\nabla^2 P(\mathbf{x}, t) - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} P(\mathbf{x}, t) = 0, \quad (2.1)$$

O pacote Seismic Unix (SU) oferece subrotinas de modelagem e processamento de dados sísmicos, pode-se dividir em três grandes etapas este relatório:

- Criação do modelo pela subrotina *trimodel*.
- Aquisição do modelo (simulação do experimento sísmico).

Para a modelagem da estrutura geológica foi usada a subrotina *trimodel* do pacote *Seismic Unix* (Cohen, 2017), esta rotina se baseia na triangulação de Delaunay. Este método sofisticado de geração de gráficos e imagens digitais, em duas ou três dimensões, é amplamente utilizado para modelar a superfície de objetos de diferentes complexidades.

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

$$s(x, z) = s_0 + (x - x_0) \frac{ds}{dx} + (z - z_0) \frac{ds}{dz} \quad (2.2)$$

2.3 PARÂMETROS DE AQUISIÇÃO

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

Utilizamos a subrotina *triseis* do *Seismic Unix* que tem por objetivo gerar os traços sísmicos referente a resposta do modelo de camadas curvas. A subrotina se baseia na teoria de raios e gera os sismogramas sintéticos de feixes Gaussianos, a partir de um modelo triangularizado preenchido (triângulos) com os valores de vagarosidade. Os parâmetros requeridos pela subrotina são:

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

2.4 PRÓXIMOS PASSOS DO PROCESSAMENTO SÍSMICO

Com o modelo de velocidade pronto e a aquisição feita, deve-se realizar os seguintes passos: análise da velocidade, correção de deslocamento normal (NMO), empilhamento e migração.

Após a aquisição dos dados de reflexão sísmica, os mesmos dados são processados, de forma que o produto final seja a seção sísmica, a ser interpretada por geofísicos e geólogos. Existem três etapas principais no processamento de dados sísmicos: deconvolução, empilhamento e migração, em sua ordem usual de aplicação (YLMAS, 1987). A deconvolução remove os efeitos da *wavelet*, que é a onda gerada pela fonte sísmica, do traço sísmico registrado nos receptores de superfície. Com a deconvolução aumenta a resolução temporal do traço sísmico. Após a deconvolução, ocorre o empilhamento que é um procedimento de compressão, de forma que o volume de dados é reduzido a uma seção sísmica empilhada. Isso é feito aplicando a correção de deslocamento normal (NMO) aos traços sísmicos classificados em grupos ou famílias de pontos médios comuns (CMPs) e, em seguida, os traços são somados ao longo do eixo de deslocamento. Um importante parâmetro necessário para o empilhamento é a chamada velocidade de empilhamento, que por sua vez é obtida por meio de uma análise de velocidade ou de um processo estatístico de maximização da consistência. Finalmente, a migração é uma etapa que elimina difrações e mapeia os eventos em uma seção empilhada para suas posições corretas de subsuperfície. Para obter a imagem empilhada, os dados são transformados de coordenadas fonte-receptor em famílias CMP. Uma família CMP consiste em vários traços sísmicos que têm diferentes posições de fonte e receptor, mas todos têm o mesmo ponto médio (Ortega et al., 2020).

3 METODOLOGIA

Neste capítulo, serão elencados os passos utilizados para o desenvolvimento do processamento sísmico realizado. Todos os passos foram realizados usando o pacote *Seismic Unix*.

3.1 CONSTRUINDO MODELO DE CAMADAS PLANAS E AQUISIÇÃO

Para o início do trabalho, foi feito um modelo sintético para realizar o processamento sísmico. Será realizado no formato de camadas planas (3.1). Este é um modelo com três camadas com velocidades diferentes, sendo o total da distância de 6 km e uma profundidade de 2 km.

A primeira camada tem 800 metros de profundidade e possui a velocidade nesta região de 1508 m/s. A segunda camada tem 800 metros de profundidade e possui velocidade nesta região de 1715 m/s. A última camada tem 400 metros de profundidade e possui velocidade nesta região de 1925 m/s. A Figura 3.1 foi gerada a partir do apêndice A–.

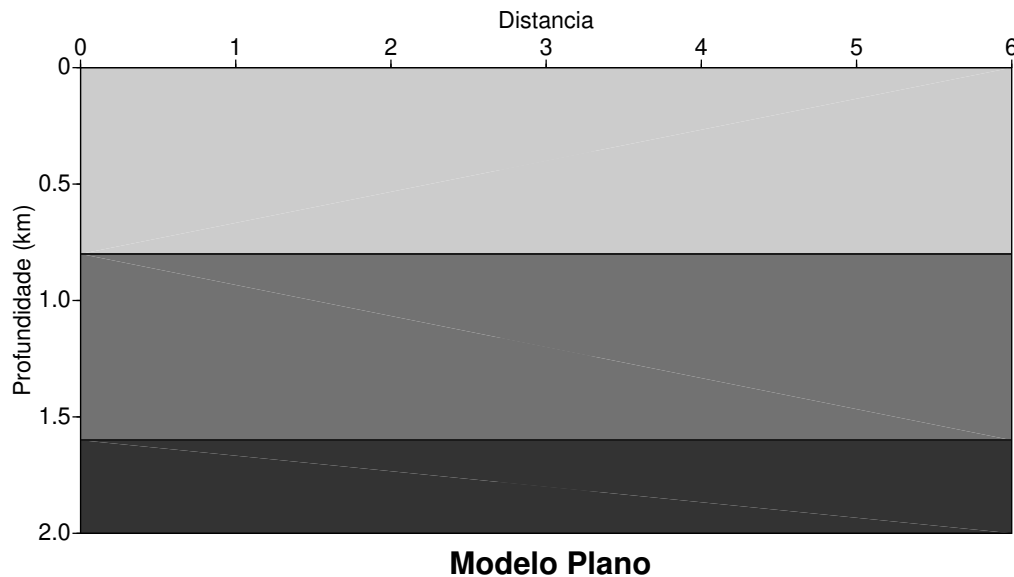


Figura 3.1: Modelo 2D de camadas planas.

A partir deste modelo, será feita a aquisição dos dados. Foram utilizados 40 fontes. Estas foram espaçadas em 50 metros. Enquanto que foram usados 60 geofones para cada tiro, gerando um total de 2400 traços. O intervalo de deslocamentos de geofone é entre -1475m e 1475m. As fontes estão localizadas entre a distância de 2 km até 3.95 km, enquanto o geofone está entre 0.525km e 5.425km.

Com estes resultados, será feita uma avaliação das velocidades para os CMPs e, se

necessário, utilizar a correção NMO. Após estes passos, será feito o empilhamento e a migração temporal. Estes resultados serão apresentados na próxima seção.

4 RESULTADOS E DISCUSSÕES

Neste capítulo, serão mostrados os resultados gráficos dos processos do processamento sísmico. Será apresentada de duas formas: uma com ruído e outra sem ruído, com o intuito de comparar e analisar a funcionalidade dos processos que retiram o ruído.

4.1 ANÁLISE DE VELOCIDADE

Após a aquisição e separar os CMPs, foram escolhidos CMPs com maior quantidade de traços para analisar a velocidade. A partir disto, foi possível gerar a Figura 4.1. Nesta Figura, mostra o exemplo de uma seção sísmica. Neste caso, observa-se que tem hiperboles. Para fazer a correção da velocidade, serão escolhidas velocidades do mapa de semblance (Fig. 4.2) para fazer a correção NMO (Fig. 4.3). Com esses passos, será possível horizontalizar as curvas. Além disso, foi possível gerar um gráfico com a velocidade constante empilhada (Fig. 4.4).

4.2 APLICANDO RUÍDO NOS DADOS

Para os próximos passos, foi aplicado ruído de Gauss nos dados de CMPs e, como comparativo, foram geradas as imagens da seção CMP 53 com o intuito de comparar as diferenças (Fig. 4.5, Fig. 4.6, Fig. 4.7 e Fig. 4.8).

Para comparar melhor esses testes, é possível analisar as Figuras 4.2 e 4.6. Nota-se que no primeiro mapa semblance está bem dividida a mudança de camadas (parte verde), enquanto que na segunda não está tão claro. Para mudar isso, é preciso aplicar correções para eliminar ao máximo o ruído.

4.3 APLICANDO CORREÇÕES

Após escolher as velocidades para cada família de CMP com o intuito de corrigir a velocidade foi gerada a Figura 4.9. Nesta imagem mostra uma correção das hiperboles encontradas nas seções sísmicas, tendo uma horizontalização das velocidades. Após a correção, foi aplicado o empilhamento (Fig. 4.10), reunindo todas as informações e dados corrigidos. Por fim, foi aplicada a migração, sendo um processo que elimina difrações e mapeia os eventos em uma seção empilhada, gerando a Figura 4.11.

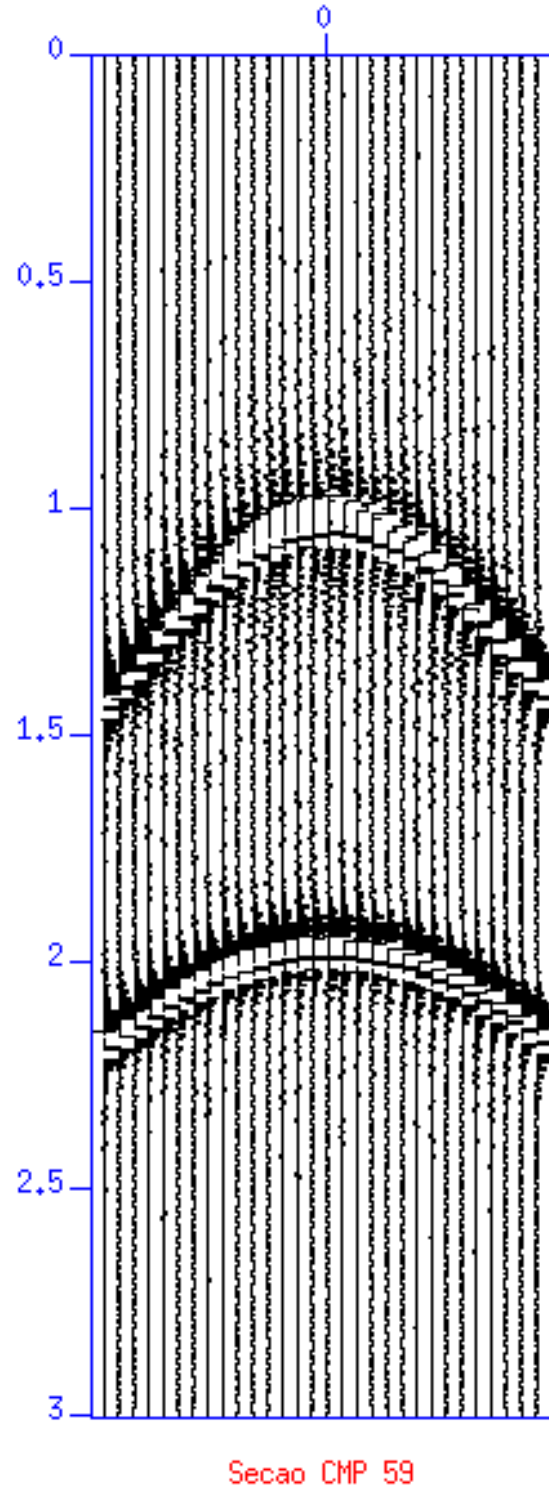


Figura 4.1: Seção CMP 59.

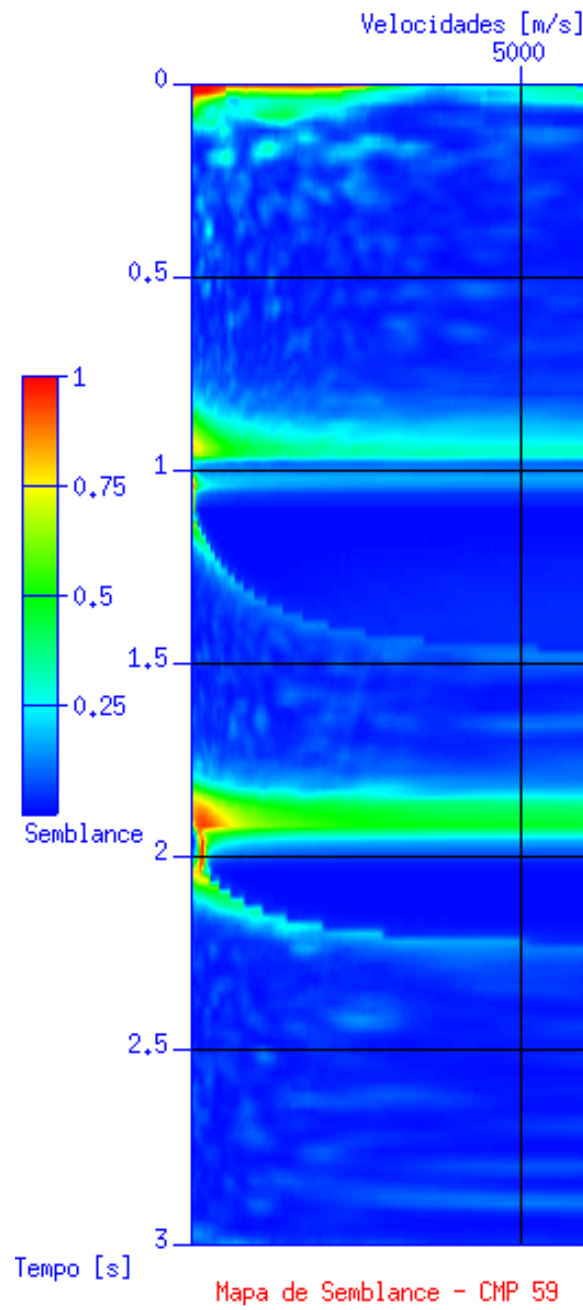


Figura 4.2: Mapa de Semblance - CMP 59.

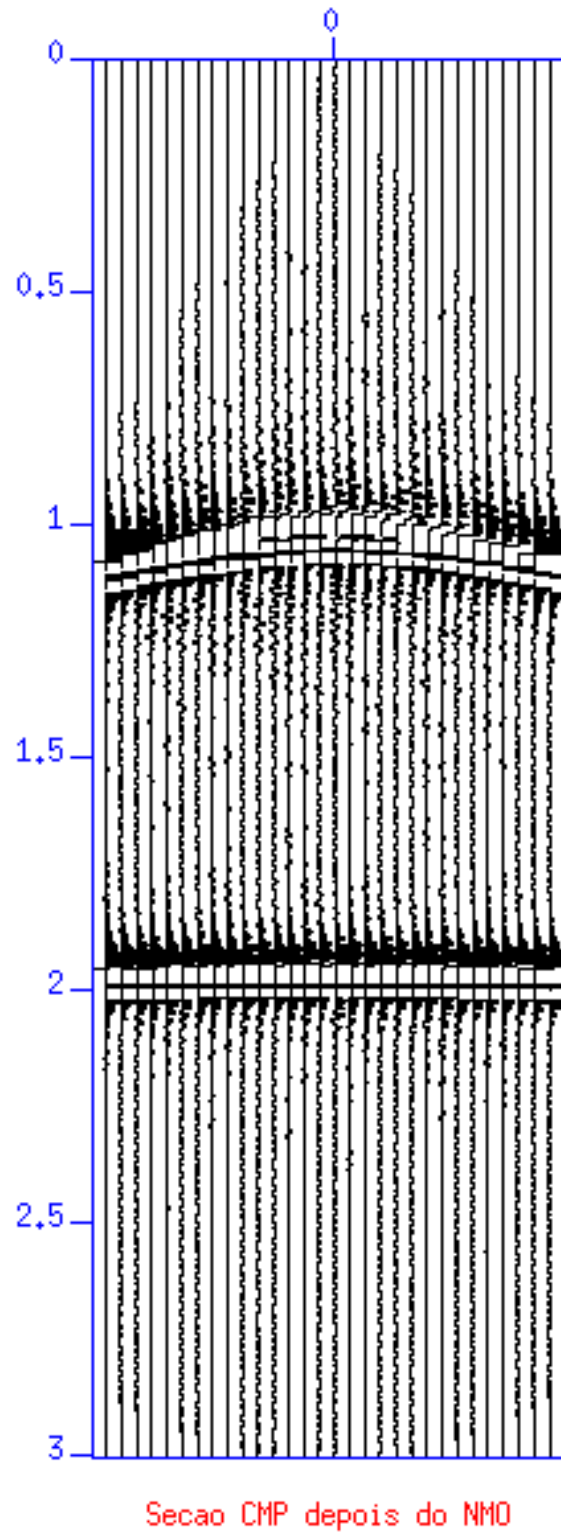


Figura 4.3: Seção CMP 59 após correção NMO.

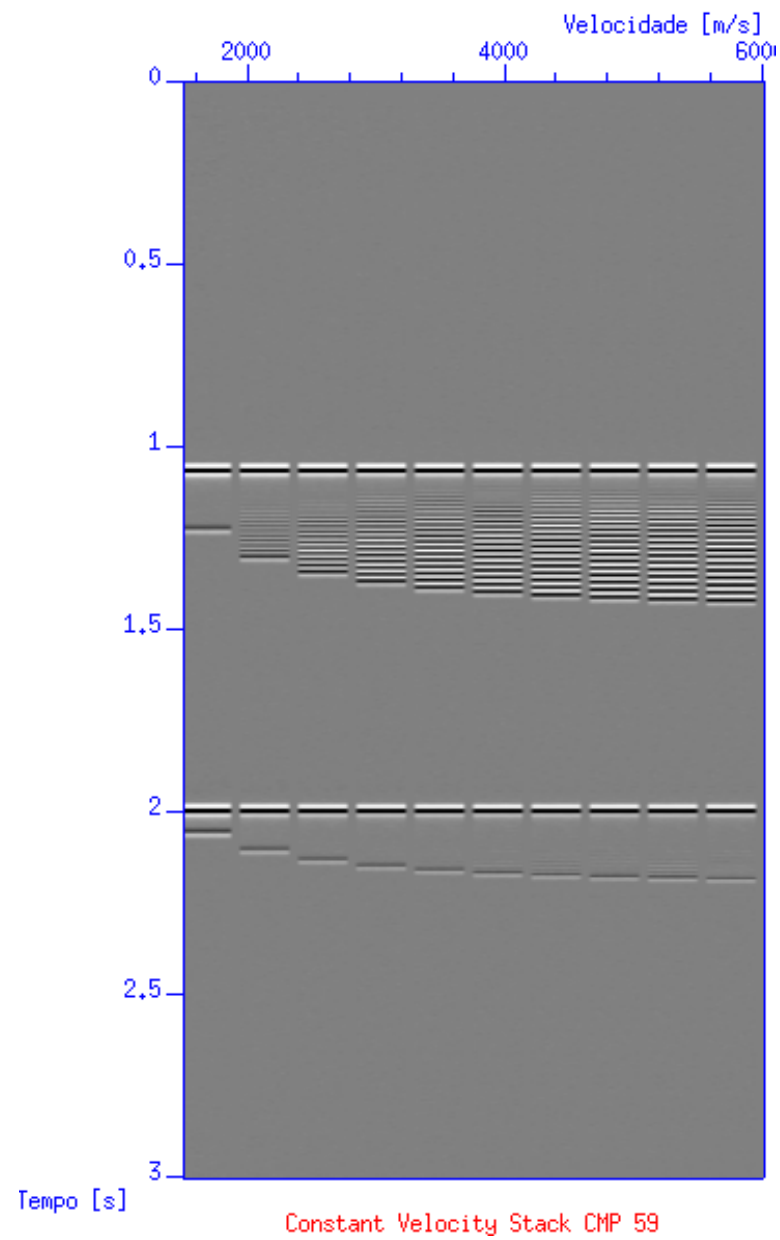


Figura 4.4: Velocidade constante empilhada - CMP 59.

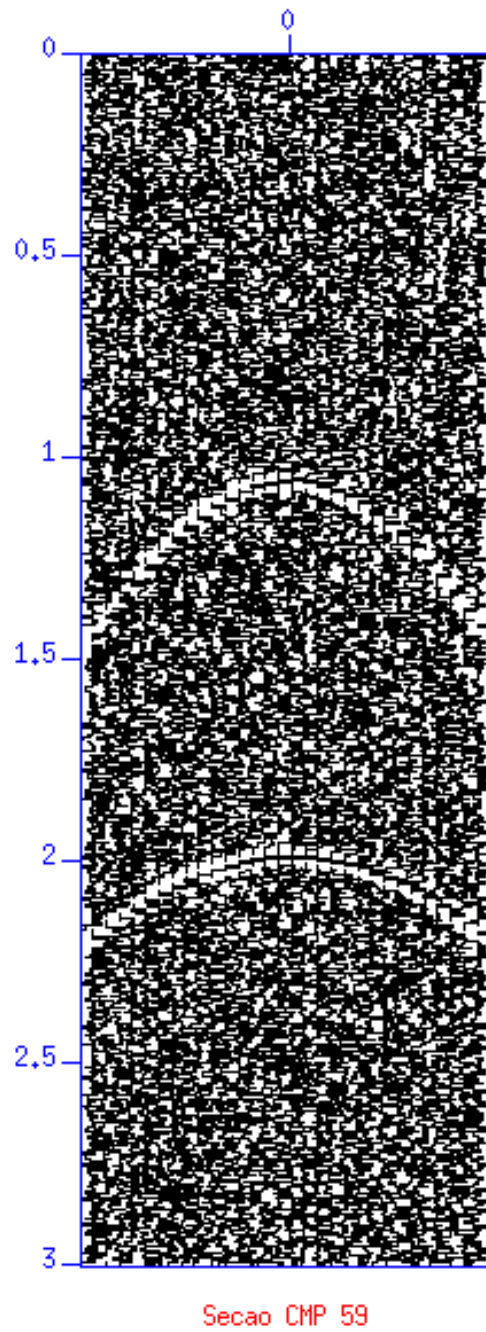


Figura 4.5: Seção CMP 59 com ruído.

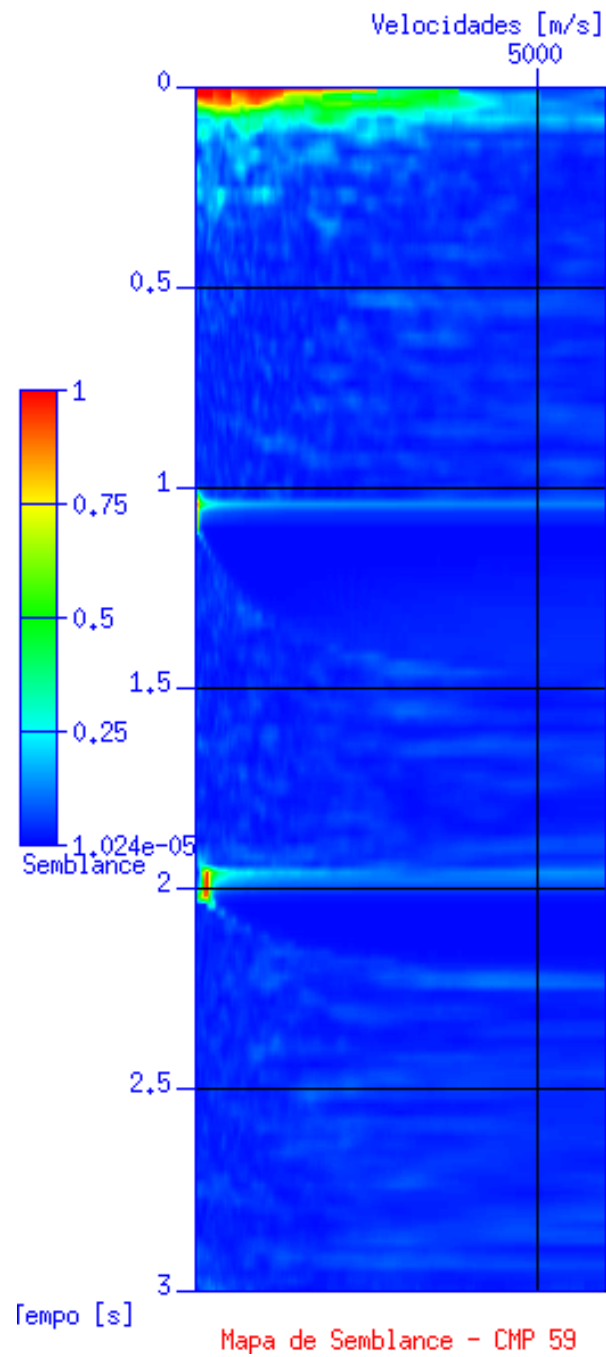


Figura 4.6: Mapa de Semblance com ruído - CMP 59.

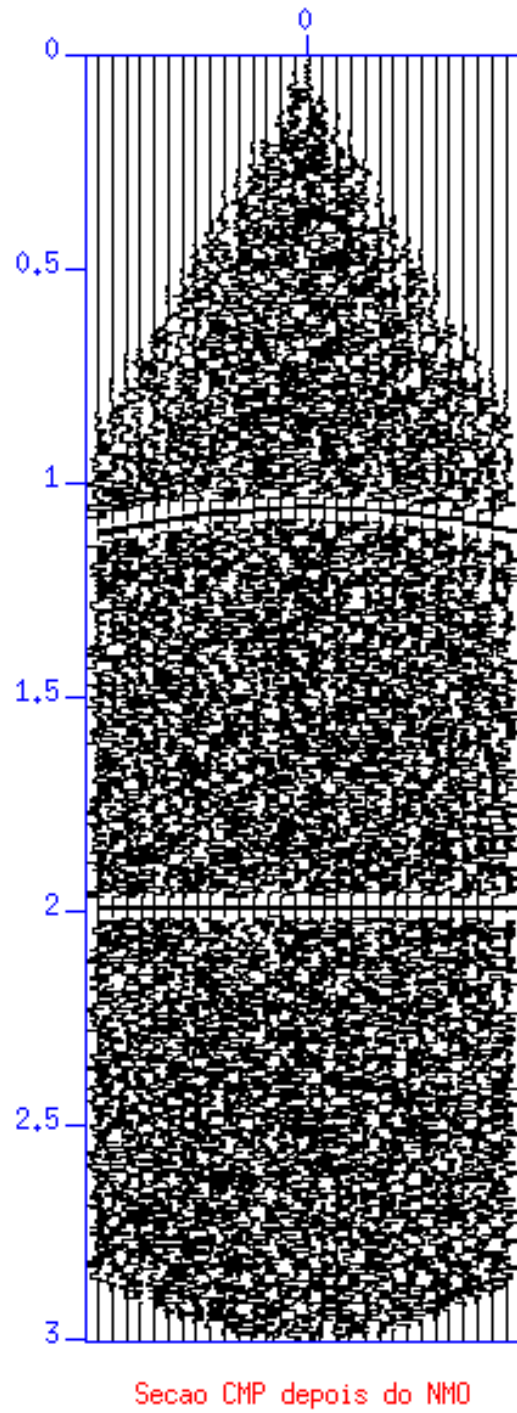


Figura 4.7: Seção CMP 59 com ruído após correção NMO.

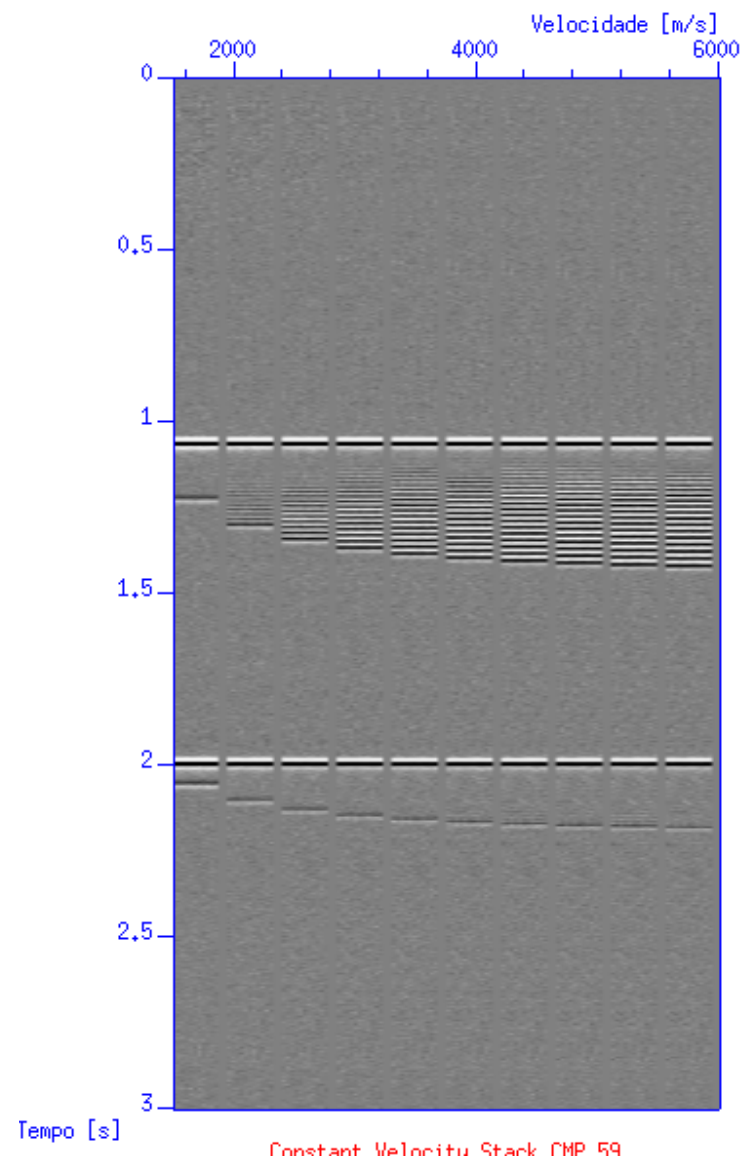


Figura 4.8: Velocidade constante empilhada com ruído- CMP 59.

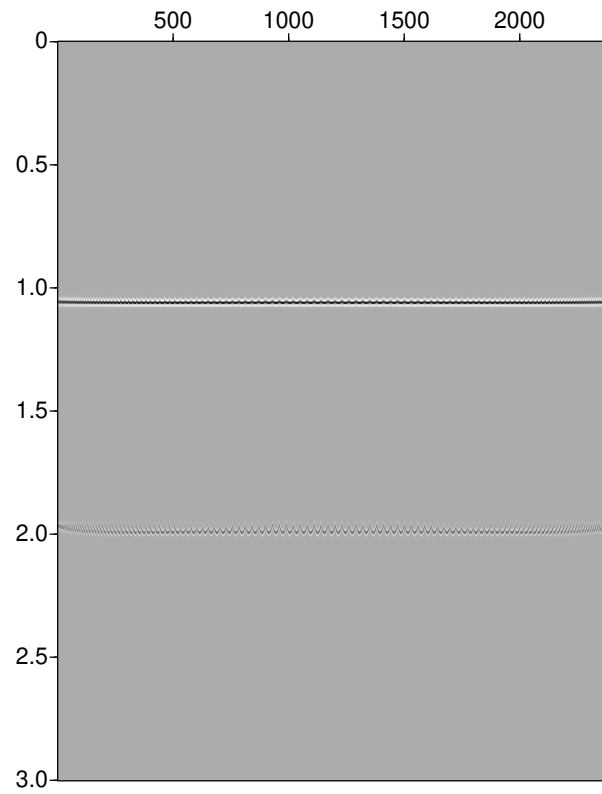


Figura 4.9: Correção NMO.

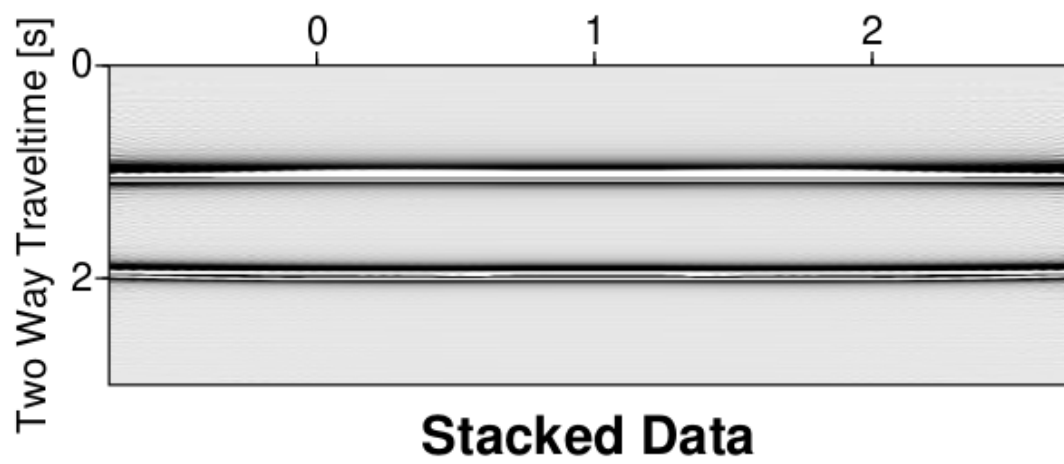


Figura 4.10: Empilhamento do modelo de camadas planas.

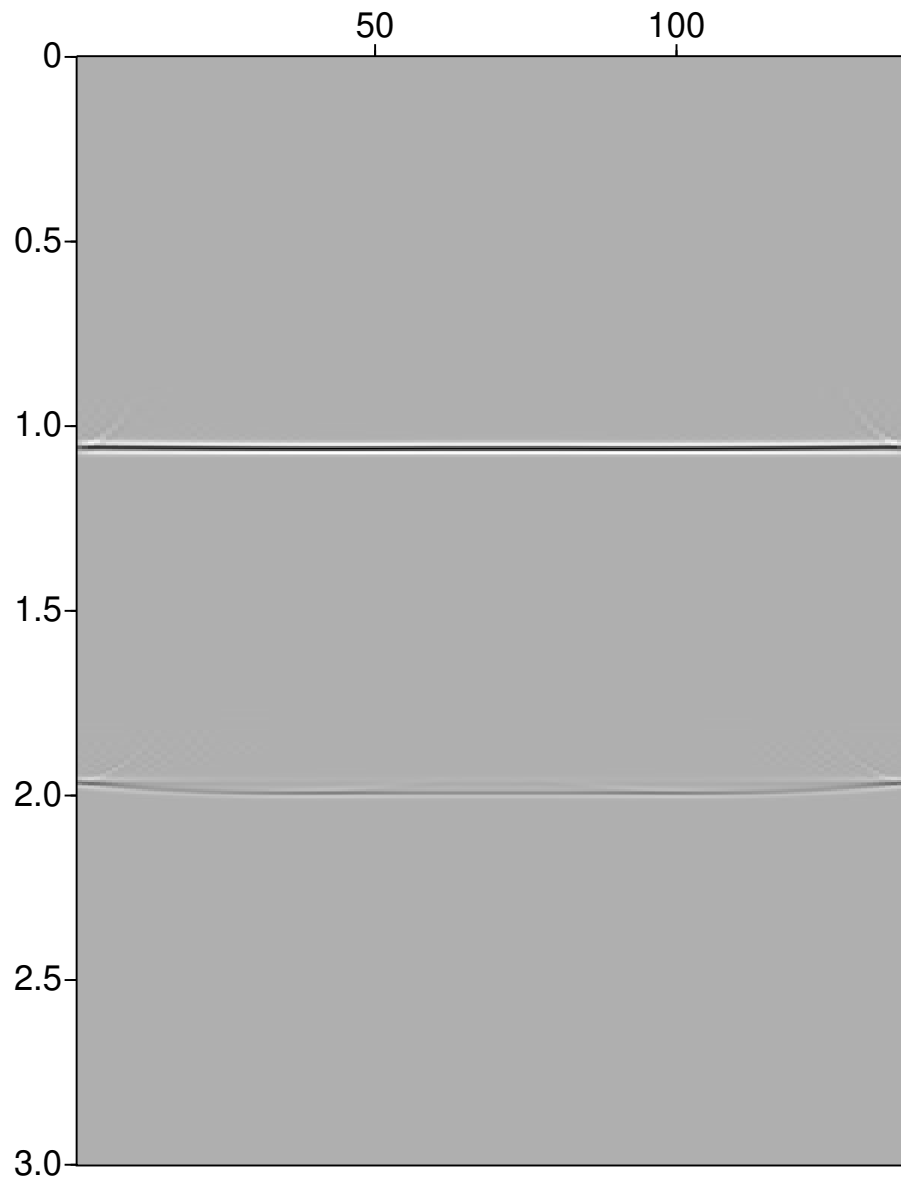


Figura 4.11: Migração temporal dos dados.

5 CONCLUSÃO

Após a finalização deste trabalho, concluiu-se que o processamento sísmico teve resultados satisfatórios. É possível afirmar isso, pois cada um dos passos (criação do modelo 2D de camadas planas, aquisição, separar em CDP, análise de velocidade, correção NMO, empilhamento e migração) tiveram os resultados esperados e condizentes com a teoria.

Além disso, foi importante para conhecer mais da teoria envolvendo processamento sísmico e adquirir mais conhecimento em relação a parte de programação do pacote *Seismic Unix*.

REFERÊNCIAS

- Cohen, J. K.; Stockwell, J. J. W., 2017, Cwp/su: Seismic un*x package - a free open seismic processing, research, and educational software package.: Colorado School of Mines: Center for Wave Phenomena.
- Edwards, S., 1988, Uses and abuses of seismic modeling: *Geophysics*, **7**, no. 4, 42–46.
- Forel, D., T. Benz and W. D. Pennington, 2005, Seismic data processing with seismic un*x: Society of Exploration Geophysicists.
- Ortega, F. G., A. Bassrei, E. N. Gomes and A. G. OLIVEIRA, 2020, Non-hyperbolic velocity analysis of seismic data from jequitinhonha basin, northeastern brazil: *Anais da Academia Brasileira de Ciências*, **92**.
- Yilmaz, Ö., 2001, Seismic data analysis: Processing, inversion, and interpretation of seismic data: Society of exploration geophysicists.
- YLMAZ, O., 1987, Seismic data processing: Seg publication: Tulsa, OK.
- Zakaria, A., J. Penrose, F. Thomas and X. Wang, 2000, The two dimensional numerical modeling of acoustic wave propagation in shallow water: Presented at the Australian Acoustical Society Conference (Acoustics).

APÊNDICES

A– SHELL SCRIPT MODELAGEM TRIMODEL

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

```
#!/bin/sh
#File: model.sh

# Set messages on
set -x

#Experiment Number
#num=1

datafile=camadas_planas.dat

trimodel xmin=0 zmin=0 xmax=6 zmax=2 \
  1 xedge=0,6 \
    zedge=0,0 \
    sedge=0,0 \
  2 xedge=0,6 \
    zedge=0.8,0.8 \
    sedge=0,0 \
  3 xedge=0,6 \
    zedge=1.6,1.6 \
    sedge=0,0 \
  4 xedge=0,6 \
    zedge=2,2 \
    sedge=0,0 \
  sfill=1,0.3,0,0,0.44,0,0 \
  sfill=1,1,0,0,0.34,0,0 \
  sfill=1,1.8,0,0,0.27,0,0 \
  kedge=1,2,3,4 \
  >$datafile

# Cria um Arquivo *.PS do Modelo
spsplot <$datafile >camadas_planas.eps \
  title=" Modelo Plano" \
  titlecolor=black axescolor=black gridcolor=white \
  labelz="Profundidade (km)" labelx="Distancia" legend=1\
  xbeg=0.0 xend=6.0 \
  zbeg=0 zend=2.0 \
  dxnum=1.0 dznum=0.5 \
```

```
gridx= solid gridz= dash\  
gedge=0.1 gtri=2.0 gmin=0.2 gmax=0.8 \  
wbox=10.0 hbox=5.0 &  
  
sxplot < $datafile -geom 1350x450+0+0 edgecolor=black tricolor=none ...  
title="Modelo em profundidade" \  
label1="Profundidade (km)" label2="Distancia (km)" &  
  
#evince modelo.eps &  
  
exit
```


B- SHELL SCRIPT AQUISIÇÃO TRISEIS

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

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

# Assign values to variables
nangle=131
fangle=-65
langle=65
nt=751
dt=0.004

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

# Name input model file
inmodel=camadas_planas.dat

# Name output seismic file
outseis=camadas_planas.su

# Remove survey file
rm -f survey.txt
# Name survey file
survey=survey.txt

#=====
# Create the seismic traces with "triseis"
#   i-loop = 40 source positions
#   j-loop = 60 geophone positions (split-spread)
#           per shot position
#   k-loop = layers 2 through 24
#           (do not shoot layers 1 and 25)

echo " --Begin looping over triseis."

i=0
while [ "$i" -ne "40" ]
```

do

```

fs=`bc -l <<-END
$i * 0.05
END`
sx=`bc -l <<-END
$i * 50 + 2000
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 + 525
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 "4" ]
do
triseis < $inmodel  xs=2,3.95 xg=0.525,5.425  zs=0,0 zg=0,0 \
nangle=$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,tracl,fldr,tracl,tracl,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 > $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!"

sugethw <camadas_planas.su key=trac1,flidr,tracf,sx,gx,offset> teste.txt

exit

```

C– SHELL SCRIPT CDP

Este shell script foi escrito para fazer o *plot* das imagens dos *shots* para verificar se aquisição está correta.

```
#!/bin/sh

#Vai plotar imagens dos shots para verificar se esta ok a aquisicao.

infile=camadas_planas.su
#templ=camada_plana_cdp1.su
outfile=camadas_planas_cdp.su

#Inserir a informacoes referentes aos cdps
suchw<$infile key1=cdp key2=gx key3=sx b=1 c=1 d=2 > templ

#Renumeracao dos cdps
suchw <templ key1=cdp key2=cdp key3=cdp a=-1237 b=1 d=25 | susort > ...
    $outfile cdp offset

#rm temp*

sugethw <$outfile key=tracf, fldr, sx, gx, offset, cdp > info.txt

sukeycount <$outfile key=cdp> fold.txt

exit
```

D– SHELL SCRIPT RUIDO DE GAUSS

Este shell script foi escrito para adicionar um ruído do tipo Gauss a seção sísmica.

```
#!/bin/sh

#Vai plotar imagens dos shots para verificar se esta ok a aquisicao.

infile=camadas_planas.su
#templ=camada_plana_cdp1.su
outfile=camadas_planas_cdp.su

#Inserir a informacoes referentes aos cdps
suchw<$infile key1=cdp key2=gx key3=sx b=1 c=1 d=2 > templ

#Renumeracao dos cdps
suchw <templ key1=cdp key2=cdp key3=cdp a=-1237 b=1 d=25 | susort > ...
    $outfile cdp offset

#rm temp*

sugethw <$outfile key=tracf, fldr, sx, gx, offset, cdp > info.txt

sukeycount <$outfile key=cdp> fold.txt

exit
```

E- SHELL SCRIPT ANÁLISE DE VELOCIDADE

O objetivo deste programa é de fazer análise de velocidades.

```
#!/bin/sh
#
# O objetivo deste programa e de fazer analise de velocidades.
#
# set -x

cmps=ruido_cdp.su

if [ ! -f $cmps ]
then    echo "Sort to CMP first!"
        pause EXIT
        exit
fi    # encerra "if"

echo "Analise de Velocidades"

rm -f panel.* picks.* par.* tmp*

#-----
# Definicao de variaveis:
#-----

indata=$cmps
outdata=velan_ruido.data #vpick

nt=751
dt=0.004

nv=420      # Numero de velocidades
dv=10       # Intervalo
fv=1500     # Primeira velocidade

>$outdata   # Arquivo vazio
>par.cmp    # Arquivo vazio

#-----
# Analise de velocidade Interativo
#-----

echo "Entre o numero de CMPs para o Picking:" >/dev/tty
```

```

read nrpicks

i=1
while [ $i -le $nrpicks ]
do
    echo "Especifique o numero de CMP $i" >/dev/tty
    read picknow
    echo "Preparando a posicao $i de $nrpicks para Picking "
    echo "A posicao e CMP $picknow "

#-----
# Plotando o CDP:
#-----

    suwind <$indata key=cdp min=$picknow \
        max=$picknow >panel.$picknow
    suxwigb <panel.$picknow xbox=312 ybox=10 \
        wbox=300 hbox=600 \
        title="Secao CMP $picknow" \
        perc=94 key=offset verbose=0 &

#-----
# Preparando o CVS (Constant Velocity Stack)
#-----

    >tmp1
    j=1
    k=`expr $picknow + 10`
    l=`bc -l <<-END
    $dv * $nv / 120
    END`

    suwind <$indata key=cdp min=$picknow \
        max=$k >tmp0

    while [ $j -le 10 ]
    do
        vel=`bc -l <<-END
        $fv + $dv * $j * $nv / 10
        END`

        sunmo <tmp0 vnmo=$vel |

        sustack >>tmp1
        sunull ntr=2 nt=$nt dt=$dt >>tmp1
        j=`expr $j + 1`
    done

```

```

suximage <tmp1 xbox=834 ybox=10 wbox=400 hbox=600 \
    title="Constant Velocity Stack CMP $picknow" \
    label1="Tempo [s]" label2="Velocidade [m/s]" \
    f2=$fv d2=$l verbose=0 mpicks=picks.$picknow \
    perc=99 n2tic=5 cmap=rgb0 &

#-----
# Plotando o mapa de Semblance:
#-----

echo "  Coloque o ponteiro do mouse sobre o mapa de Semblance ou sobre"
echo "  os paineis CVS e pressione 's' para salvar a velocidade ...
    marcada."
echo "  Pressione 'q' no mapa de semblance depois de marcar todas ...
    as velocidades"

suvelan <panel.$picknow nv=$nv dv=$dv fv=$fv |
suximage xbox=1 ybox=10 wbox=300 hbox=600 \
    units="Semblance" f2=$fv d2=$dv \
    label1="Tempo [s]" label2="Velocidades [m/s]" \
    title="Mapa de Semblance - CMP $picknow" cmap=hsv2 \
    legend=1 units=Semblance verbose=0 gridcolor=black \
    grid1=solid grid2=solid mpicks=picks.$picknow

sort <picks.$picknow -n |
mkparfile string1=tnmo string2=vnmo >par.$i

#-----
# Plotando a secao corrigida de NMO e o perfil de velocidades
#-----

>tmp2
echo "cdp=$picknow" >>tmp2
cat par.$i >>tmp2

sunmo <panel.$picknow par=tmp2 |
suxwigb title="Secao CMP depois do NMO" xbox=1 ybox=10 \
    wbox=300 hbox=600 verbose=0 key=offset perc=94 &

    sed <par.$i '
s/tnmo/xin/
s/vnmo/yin/
    ' >par.uni.$i
unisam nout=$nt fxout=0.0 dxout=$dt \
    par=par.uni.$i method=mono |
xgraph n=$nt nplot=1 dl=$dt fl=0.0 \
    label1="Tempo [s]" label2="Velocidade [m/s]" \

```



```

        title="Funcao Velocidade CMP $picknow" \
        -geometry 200x600+934+10 style=seismic &

echo "Deseja salvar as velocidades marcadas? (y/n) " >/dev/tty
read response

rm tmp*

case $response in
n) i=$i echo "Removendo as velocidades marcadas" ;;
y) i=`expr $i + 1`
    echo "$picknow $i" >>par.cmp ;;
*) echo "Digite y para yes e n para nao";;
esac

done

#-----
# Criando o arquivo de saida para as velocidades
#-----

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

i=0
while [ $i -le $nrpicks ]
do
    cat par.$i >>$outdata
    i=`expr $i + 1`
done

rm -f panel.* picks.* par.* tmp*

exit

```

F- SHELL SCRIPT CORREÇÃO NMO

O objetivo deste shell script é de fazer a correção NMO nos dados CMP.

```
#!/bin/sh
#
# O objetivo deste programa e de fazer analise de velocidades.
#
# set -x

cmps=ruido_cdp.su

if [ ! -f $cmps ]
then    echo "Sort to CMP first!"
        pause EXIT
        exit
fi      # encerra "if"

echo "Analise de Velocidades"

rm -f panel.* picks.* par.* tmp*

#-----
# Definicao de variaveis:
#-----

indata=$cmps
outdata=velan_ruido.data #vpick

nt=751
dt=0.004

nv=420      # Numero de velocidades
dv=10       # Intervalo
fv=1500     # Primeira velocidade

>$outdata   # Arquivo vazio
>par.cmp    # Arquivo vazio

#-----
# Analise de velocidade Interativo
#-----

echo "Entre o numero de CMPs para o Picking:" >/dev/tty
```

```

read nrpicks

i=1
while [ $i -le $nrpicks ]
do
    echo "Especifique o numero de CMP $i" >/dev/tty
    read picknow
    echo "Preparando a posicao $i de $nrpicks para Picking "
    echo "A posicao e CMP $picknow "

#-----
# Plotando o CDP:
#-----

    suwind <$indata key=cdp min=$picknow \
        max=$picknow >panel.$picknow
    suxwigg <panel.$picknow xbox=312 ybox=10 \
        wbox=300 hbox=600 \
        title="Secao CMP $picknow" \
        perc=94 key=offset verbose=0 &

#-----
# Preparando o CVS (Constant Velocity Stack)
#-----

    >tmp1
    j=1
    k=`expr $picknow + 10`
    l=`bc -l <<-END
    $dv * $nv / 120
    END`

    suwind <$indata key=cdp min=$picknow \
        max=$k >tmp0

    while [ $j -le 10 ]
    do
        vel=`bc -l <<-END
        $fv + $dv * $j * $nv / 10
        END`

        sunmo <tmp0 vnmo=$vel |

        sustack >>tmp1
        sunull ntr=2 nt=$nt dt=$dt >>tmp1
        j=`expr $j + 1`
    done

```

```

suximage <tmp1 xbox=834 ybox=10 wbox=400 hbox=600 \
    title="Constant Velocity Stack CMP $picknow" \
    label1="Tempo [s]" label2="Velocidade [m/s]" \
    f2=$fv d2=$l verbose=0 mpicks=picks.$picknow \
    perc=99 n2tic=5 cmap=rgb0 &

#-----
# Plotando o mapa de Semblance:
#-----

echo "  Coloque o ponteiro do mouse sobre o mapa de Semblance ou sobre"
echo "  os paineis CVS e pressione 's' para salvar a velocidade ...
    marcada."
echo "  Pressione 'q' no mapa de semblance depois de marcar todas ...
    as velocidades"

suvelan <panel.$picknow nv=$nv dv=$dv fv=$fv |
suximage xbox=1 ybox=10 wbox=300 hbox=600 \
    units="Semblance" f2=$fv d2=$dv \
    label1="Tempo [s]" label2="Velocidades [m/s]" \
    title="Mapa de Semblance - CMP $picknow" cmap=hsv2 \
    legend=1 units=Semblance verbose=0 gridcolor=black \
    grid1=solid grid2=solid mpicks=picks.$picknow

sort <picks.$picknow -n |
mkparfile string1=tnmo string2=vnmo >par.$i

#-----
# Plotando a secao corrigida de NMO e o perfil de velocidades
#-----

>tmp2
echo "cdp=$picknow" >>tmp2
cat par.$i >>tmp2

sunmo <panel.$picknow par=tmp2 |
suxwigb title="Secao CMP depois do NMO" xbox=1 ybox=10 \
    wbox=300 hbox=600 verbose=0 key=offset perc=94 &

    sed <par.$i '
s/tnmo/xin/
s/vnmo/yin/
    ' >par.uni.$i
unisam nout=$nt fxout=0.0 dxout=$dt \
    par=par.uni.$i method=mono |
xgraph n=$nt nplot=1 dl=$dt fl=0.0 \
    label1="Tempo [s]" label2="Velocidade [m/s]" \

```

```

        title="Funcao Velocidade CMP $picknow" \
        -geometry 200x600+934+10 style=seismic &

echo "Deseja salvar as velocidades marcadas? (y/n) " >/dev/tty
read response

rm tmp*

case $response in
n) i=$i echo "Removendo as velocidades marcadas" ;;
y) i=`expr $i + 1`
    echo "$picknow $i" >>par.cmp ;;
*) echo "Digite y para yes e n para nao";;
esac

done

#-----
# Criando o arquivo de saida para as velocidades
#-----

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

i=0
while [ $i -le $nrpicks ]
do
    cat par.$i >>$outdata
    i=`expr $i + 1`
done

rm -f panel.* picks.* par.* tmp*

exit

```

G– SHELL SCRIPT EMPILHAMENTO

O objetivo deste shell script é de fazer o empilhamento dos dados.

```
#!/bin/sh

#-----
# Empilhamento de dados
#-----

sustack <nmo_rui.data >stack_rui.data
supsimage < stack_rui.data > stack_rui.eps
supsimage <stack_rui.data >empilhamento.ps \
title="Stacked Data" labell="Two Way Traveltime [s]" \
wbox=6 hbox=2 perc=95 x2beg=-0.735 d2=0.025 f2=-0.735 &
exit
```

H– SHELL SCRIPT MIGRAÇÃO NO TEMPO

O objetivo deste shell script é de fazer a migração temporal dos dados empilhados.

```
#!/bin/sh

#-----
# Migrate Stacked Data...
#-----

sumigps <stack_rui.data >migration_rui.data \
tmig=0.0 \
vmig=2000 dx=50
supsimage < migration_rui.data > Migracao_dos_dados.eps
exit
```