**BTA/TGP/11-16 Title: Focusing one dimensional models by using the 'Single-sided' autofocusing theory of Rose for layered materials**

**Date:** 2011-08-19 **Student name V.M. de Waard**

Department of Geotechnology

Delft University of Technology

| | | |
|---|---|---|
| Title | : | Focusing one dimensional models by using the 'Single-sided' autofocusing procedure of Rose for layered materials |
| Author(s) | : | V.M. de Waard |
| Date | : | 2011-08-19 |
| Professor(s) | : | |
| Supervisor(s) | : | Dr.ir. E.C. Slob |
| TA Report number | : | BTA/TGP/11-16 |


Postal Address : Section for Applied Geophysics
Department of Applied Earth Sciences
Delft University of Technology
P.O. Box 5028
The Netherlands
Telephone : (31) 15 2781328 (secretary)
Telefax : (31) 15 2781189

# Contents

## 0.1 Abstract

In this thesis the theory of Rose about 'Single-sided' autofocusing is implemented in a computer code and tested on a few Matlab models. First the theory of single-sided reflection profiles is treated, then the construction of three models and the corresponding reflection profiles are shown. After that the equivalence between Rose's theory and the Marchenko equation is shown. Next the theory is implemented in a Matlab file, and finally the theory is tested on the reflection profiles of the models described earlier. The main result is that the theory of Rose is indeed capable of focusing a reflection profile, thus reducing the reflections from a two-way travel time to a one-way travel time and removing the multiples.

# Chapter 1

# Reflections in a layered material

In a layered medium when only one side is accessible an image of the inaccessible part can be created out of reflections. Reflections are generated when a signal is sent forward through the medium and it crosses a layer boundary. In general a velocity and density difference between the layers is present, also known as acoustic impedance, and causes a partial reflection of the signal. The other part of the signal is transmitted deeper into the medium. At the next boundary, the same principle occurs again. When multiple layers are present it is also possible that reflections are reflected back and forth between two or more interfaces, resulting in multiple arrivals at the surface of the same interface. A schematic representation of this process is visible in figure 1.1.

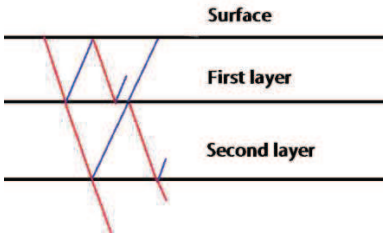


Figure 1.1: Schematic picture of subsurface reflections. Along the red lines waves are moving down, and along the blue lines waves are moving up. Process of multiple forming is clearly visible.

A reflection profile consists of all first and multiple arrivals from the interfaces. With this reflection profile an accurate image of the subsurface can be created even if the velocity profile is not known. This is the topic of chapter 2. The reflection profile can be modeled in a mathematical program like Matlab. This will be shown in the next section.

## 1.1 Construction of reflection profiles

The reflection profile is constructed as follows. When just one layer boundary is present, the reflection $R$ is equal to the reflection coefficient $r$ plus a time delay that accounts for the travel time from source to reflector and back to receiver, which in the frequency domain looks like $e^{-2i\omega t}$,

where $t = d/c$, and $d$ denotes layer thickness, while $c$ denotes velocity. When the medium consists of two boundaries, or one layer, then $R$ is the sum of all reflections and transmission that arrive eventually at the surface. Let's define the transmission as $t^+ = 1 + r$ when passing a boundary from above to below. The transmission from below to above is $t^- = 1 - r$. So, in a case of two boundaries and the source and receiver are above the upper reflector then in the frequency domain $R(d = 0, \omega)$ is the sum of the reflection on the upper reflector $e^{-iwt_0} r_0 e^{-iwt_0}$ plus the reflection that passes through the upper reflector, bounces back on the lower reflector, and again passes through the upper reflector $e^{-iwt_0}(1 + r_0)e^{-iwt_1} r_1 e^{-iwt_1}(1 - r_0)e^{-iwt_0}$ plus all the reflections that are multiples, so reflections that bounce forth and back between the upper and lower reflector before they arrive at the receiver. In this formula is $r_0$ the reflection coefficient of the upper boundary, and $r_1$ the reflection coefficient of the lower boundary. The travel time, $t_1$ is defined as $d_1/c_1$, $d_1$ is the thickness of the layer and $c_1$ is the wave velocity in the layer; $r_0$ is defined as: $\frac{Z_1 - Z_0}{Z_1 + Z_0}$, $r_1 = \frac{Z_2 - Z_1}{Z_2 + Z_1}$, and so on. Hereby is $Z$ the acoustic impedance and $Z_0 = \rho_0 c_0$. Figure 1.2 shows the situation described above. In the upper left corner a unit amplitude signal is sent in, and it gets modified and delayed by transmission and reflection.

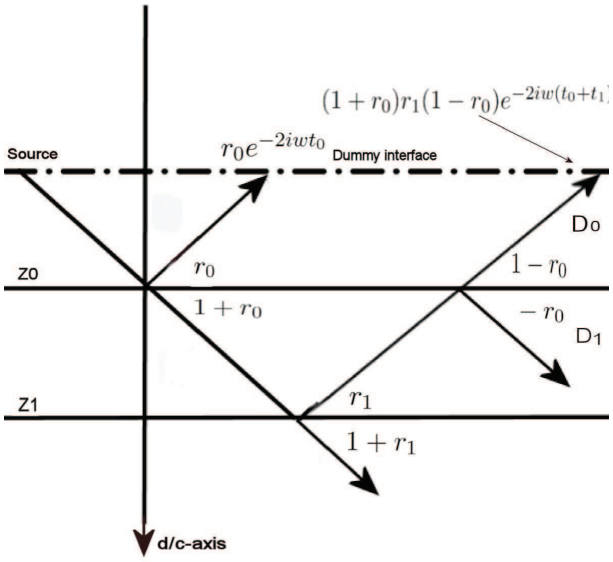


Figure 1.2: Schematic representation of a reflection profile for primary arrivals in a two boundary situation.

When all the reflections are summed together $R(d = 0, \omega)$ becomes $r_0 e^{-2iwt_0} + (1 - r_0^2) r_1 e^{-2iw(t_0+t_1)} - (1 - r_0^2) r_1^2 r_0 e^{-2iw(t_0+2t_1)} + \cdots$. Here the first three arrivals are visible and the dots represent the theoretically infinite sequence of positive and negative reflections that are multiples of the first and second reflection. This can be rewritten to:

$$R(d = 0, \omega) = e^{-2iwt_0} \left( r_0 + e^{-2iwt_1}(1 - r_0^2) r_1 \sum_{m=0}^{n} (-r_0 r_1)^m e^{-2iwmt_1} \right), \tag{1.1}$$

$$R(d = 0, \omega) = e^{-2iwt_0} \left( r_0 + \frac{(1 - r_0^2) r_1 e^{-2iwt_1}}{1 + r_0 r_1 e^{-2iwt_1}} \right), \tag{1.2}$$

$$R(d=0,\omega) = e^{-2iwt_0}\frac{r_0 + r_1 e^{-2iwt_1}}{1 + r_0 r_1 e^{-2iwt_1}}. \tag{1.3}$$

The last formula is equal to the following formula from Goupillaud [1] with a time delay. This time delay disappears when $R(d=0,\omega)$ is evaluated with both source and receiver located at the interface instead of above the interface.

$$R_0 = \frac{r_0 + r_1 e^{-2i\omega t_1}}{1 + r_0 r_1 e^{-2i\omega t_1}}. \tag{1.4}$$

Now let's add another layer on top of the last medium, $r_0$ is again the top boundary and $r_2$ is now the lowest boundary. The reflection of the two lowest interfaces is of course:

$$R_1 = \frac{r_1 + r_2 e^{-2i\omega t_2}}{1 + r_1 r_2 e^{-2i\omega t_2}}. \tag{1.5}$$

Now the total reflection profile, using that $R_1$ is the signal that returns from reflector $r_1$, becomes:

$$R_0 = \frac{r_0 + R_1 e^{-2i\omega t_1}}{1 + r_0 R_1 e^{-2i\omega t_1}}. \tag{1.6}$$

See figure 1.3 for an illustration of the concept described above.

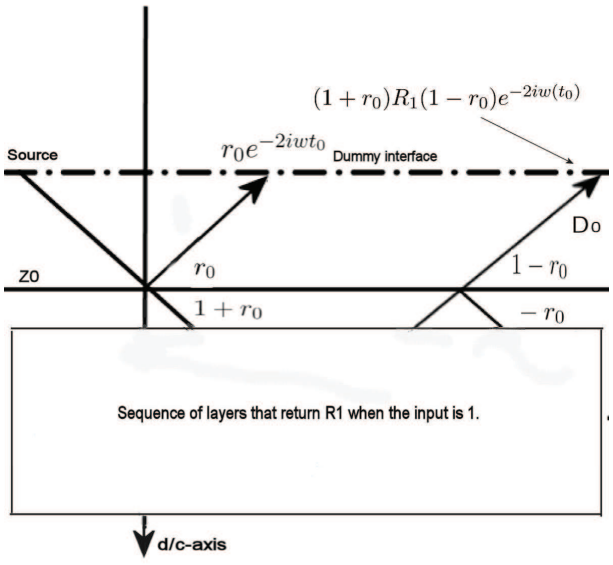


Figure 1.3: Schematic representation of a reflection profile when the lower layers return $R_1$ if the input is equal to 1.

Eventually for a model of N layers and the lowest layer is a half space; the global reflection coefficient R can be built up from bottom to top with the following recursive formula valid for any layer $D_n$,

$$R_n = \frac{r_n + R_{n+1} e^{-2i\omega t_{n+1}}}{1 + r_n R_{n+1} e^{-2i\omega t_{n+1}}}. \tag{1.7}$$

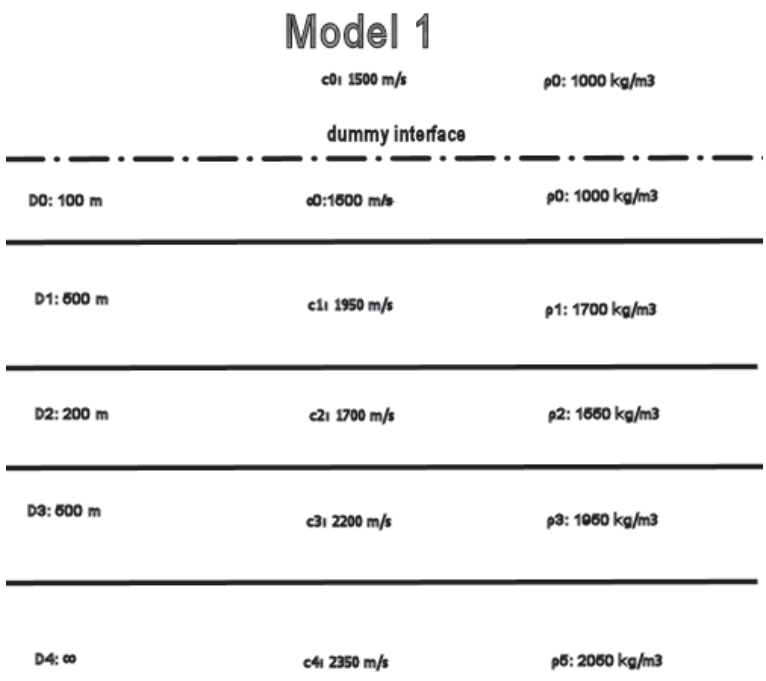


Figure 1.4: Schematic representation of model 1. Model 1 contains five layers.

In equation (1.7) is $R_N = 0$ because in the lower half-space no reflections occur, therefore $n$ goes from $N-1$ to 0, $r_{N-1}$ is the reflection coefficient of the lowest boundary present; $r_0$ is the reflection coefficient of the top boundary. Now the reflection profile follows from a convolution between $R_0$ and the signal that is sent through the medium. If the signal is equal to a delta pulse, the reflection profile is equal to $R_0$, else it will be a filtered version. A model, model 1, is constructed to create a reflection profile according to equation (1.7) and a convolution with an incident pulse. Figure 1.4 gives a schematic representation of model 1. Model 1 contains five layers, whereby the top layer and the bottom layer are half-spaces. The dashed-dotted line is a dummy interface, the imaginary source and receiver are placed on this level. The incident pulse used is a delta pulse. The result of the convolution between model 1 and the delta pulse is visible in figure 1.5 which contains the total reflection series. Compare the reflection series in figure 1.5 with the model in figure 1.4. Here can be seen that there are four layer boundaries in figure 1.4 and also four large events plus a number of smaller events in figure 1.5. The large events are primary reflections, and the smaller events are multiples. The first primary reflection corresponds with the interface between layer 0 and layer 1. The second primary reflection corresponds with the interface between layer 1 and layer 2, and so on.

With a reflection profile as in figure 1.5 it is possible to solve the inverse scattering problem. The inverse scattering problem is the problem to determine the properties of an object, in this case the internal constitution, from measured data like reflection profiles. The problem can be solved by finding a solution for the Marchenko equation. The Marchenko equation gives the exact solution

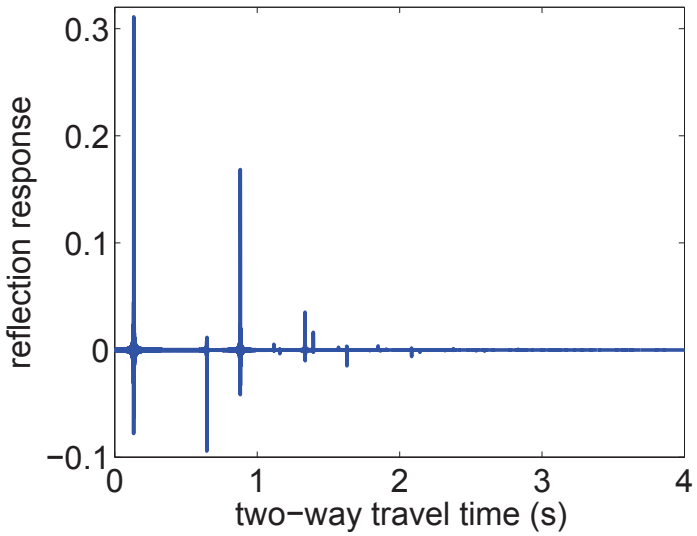


Figure 1.5: The reflection profile of the first model. This reflection series contains primary reflections and multiples.

of the inverse scattering problem. The equation can be solved by finding a direct solution, which is computationally expensive for large models if solved by direct matrix inversion, or with some iteration scheme; like the solution of Rose that will be treated in chapter 2:

$$0 = R(t + t_f) + w(t, t_f) + \int_{-\infty}^{t_f} dt' R(t + t') w(t', t_f). \tag{1.8}$$

Equation (1.8) is the Marchenko equation [2] and the exact solution of the inverse scattering problem can be determined by solving this equation by integrating $w(t, t_f)$ over all t, the acoustic impedance can be obtained as a function of one-way travel time.

## 1.2 Numerical models

To test the theory of Rose, three models have been created. The first model is already known, now for the second and third model the total reflection series and a schematic figure follows where the number of layers and properties are shown.
The second model consists of eight layers as can be seen in figure 1.6, whereby the top and bottom layer are half-spaces and two are rather thin. Eighth layers are present, so there are seven interfaces in model 2 and seven primary reflections are expected. This is also visible in figure 1.7, but it is hard to see. The primary reflections from the top and bottom interfaces of the two thin layers, their reflections are very close together, which makes it hard to distinguish the seven primary reflections, or to see which events are primaries and which are multiple reflections. There are four larger spikes pointed upwards. Again the first reflection corresponds to the boundary between layer 0 and layer 1, then directly after this event a smaller spike mostly pointed downwards is visible, which corresponds to the interface between layer 1 and layer 2. After the downward spike a very small spike pointed

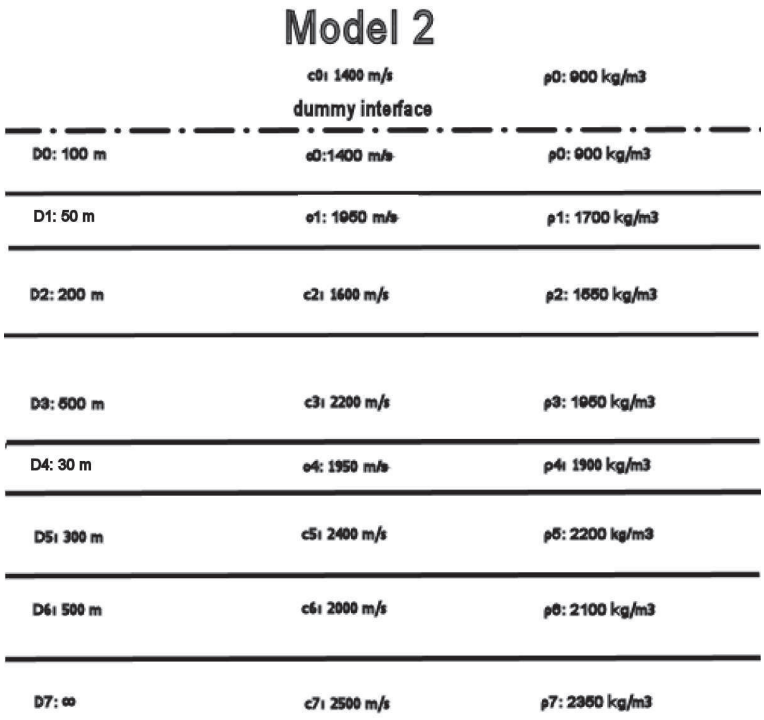


Figure 1.6: Schematic representation of model 2. Model 2 contains eight layers.

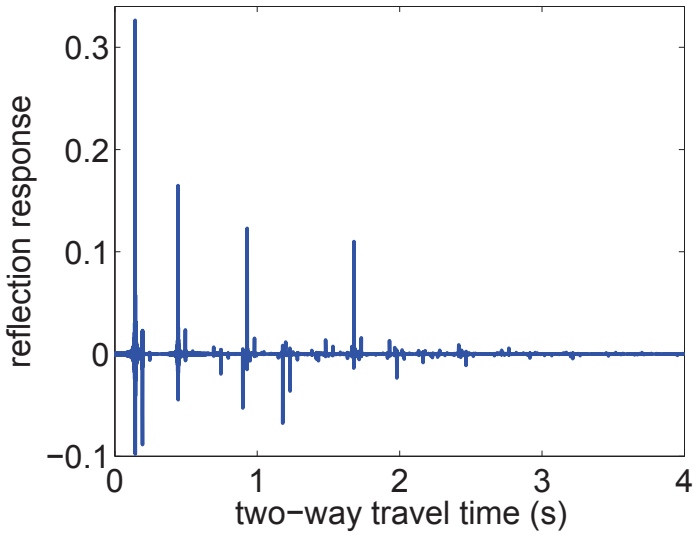


Figure 1.7: Reflectionprofile of model 2.

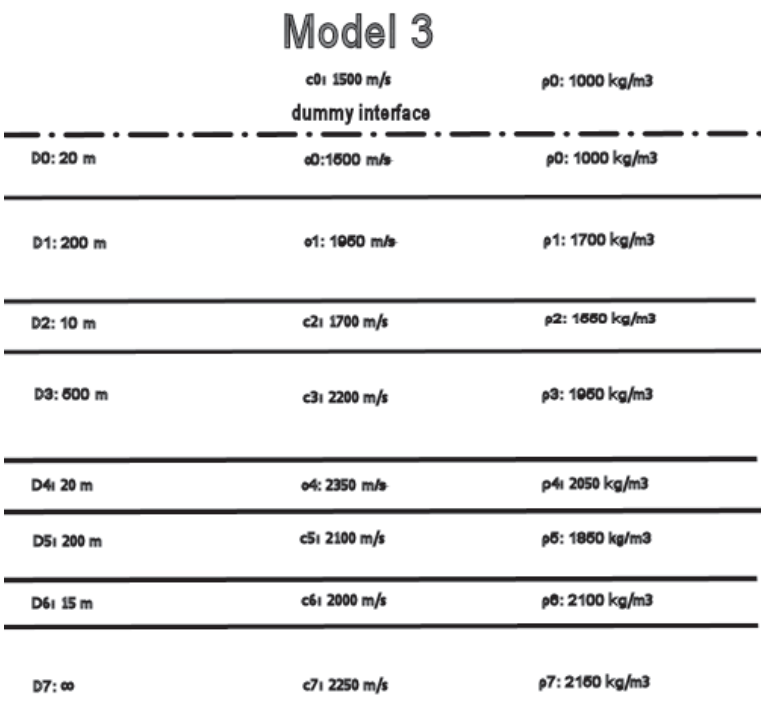


Figure 1.8: Schematic representation of model 3. Model 3 contains eight layers total.

downwards is a multiple of layer 1. The second large event pointed upwards corresponds to the boundary between layer 2 and layer 3. The third large event, if watched carefully, consists of two spikes. First a smaller spike downwards, and then a larger spike upwards. These spikes correspond to the interfaces between layers 3, 4, and 5. Next there are two spikes downwards, and one of those is a multiple and the other corresponds to the boundary between layers 5 and 6. Finally the last spike upwards corresponds to the boundary between layers 6 and 7.
Model three contains eigth layers, two layers are half-spaces and four layers are very thin, which makes it hard to distinguish the individual reflections of the thin layers in the total reflection series. See figure 1.8 for the overview. The reflection profile of model three can be analyzed in the same way as model two. Only four large events are visible, but event two, three and four consist of two spikes. Which brings the total amount of reflections up to seven primary reflections. The smaller spikes between event two and three are multiples.

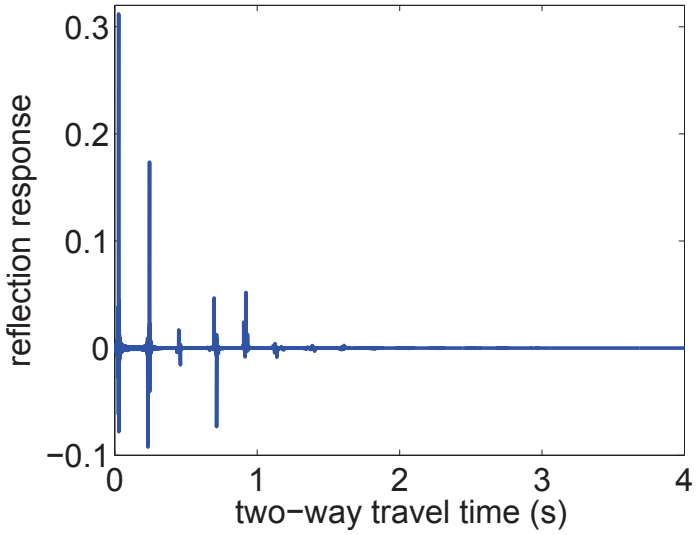


Figure 1.9: Reflectionprofile of model 3.

# Chapter 2

# Theory of Rose and implementation in Matlab

## 2.1 'Single-sided' autofocusing

When using sound as a method of image analyses some problems arise. One of them is the loss of focus. In the ideal situation it is possible to focus at any place in a medium so a good reflection image can be created. An important problem with focusing in a medium is the disturbance by variations in the velocity profile. Prada and Fink et al. [3, 4, 5] discovered a form of autofocusing for three dimensional acoustics. Iteratively retransmitting an array's own time-reversed signal causes it to focus on the strongest scatterer in range. A limitation in this procedure is that only the focus time is known, and not the focus point. Furthermore only the strongest scatterers will be visible. Aktosun and Rose [6] theoretically proved that it is possible for the one dimensional variable velocity Helmholtz equation to focus at any place in the medium. They proved that one can focus to any point in one dimension by constructing an incident pulse that consists of a delta function wavefront followed by the time-reversed solution of Marchenko's equation. Now Rose [2] proposed an automatic iterative procedure for one dimensional single-sided focusing at a specified time. The advantage is that only the input and output needs to be measured, no equations need to be solved. First a short description of the method will follow, and then the equivalence with the Marchenko equation will be shown.

## 2.2 Theory of Rose

The iterative scheme to solve equation (1.8) that Rose proposes is as follows: For any $t_f$, note that $t_{fmax}$ is equal to $\frac{1}{2}t_{max}$, start the following procedure.

1. Send in $\delta(t + t_f)$ at $x = 0$.
2. Record the reflection response $R(t + t_f)$ at $x = 0$.
3. Truncate $R$ after $t = t_f$. $R$ becomes $R_0(t + t_f)$ which is zero for $t > t_f$.

4. Time-reverse $R_0(t+t_f)$ so it becomes $R_0(-t+t_f)$.
5. Subtract $R_0(-t+t_f)$ from $\delta(t+t_f)$.
6. Send in $\delta(t+t_f)-R_0(-t+t_f)$ at $x=0$. Step 6 is equal to step 1, so step 7 is step 2 etcetera.

The mathematical description follows now. The new signal is a convolution of the old signal with the reflection $R_0$.

$$\varphi_{out}(t) = \int_{-\infty}^{\infty} dt' R(t-t')\varphi_{in}(t'). \tag{2.1}$$

When $\varphi_{in}(t')$ is equal to a delta-function and the iteration takes place according to the description above, then equation (2.1) becomes:

$$\varphi^f_{out}(t;t_f) = R_0(t+t_f) - \int_{-t_f}^{t_f} dt' R_0(t+t')R_0(t'+t_f) + \int_{-t_f}^{t_f} dt'' \int_{-t_f}^{t_f} dt' R_0(t+t')R_0(t'+t'')R_0(t''+t_f) + \cdots . \tag{2.2}$$

Notice that the reflection signal $R$ is truncated for $t > t_f$, and because the function above is a correlation $t$ is limited at $-t_f$. So, the interval of the integration is limited to $-t_f < t < t_f$, therefore $R$ can be replaced by $R_0$. When summed, this is equal to:

$$\varphi^f_{out}(t;t_f) = R_0(t+t_f) - \int_{-t_f}^{t_f} dt' R_0(t+t')\varphi^f_{out}(t';t_f). \tag{2.3}$$

Now define $w(t,t_f) = -\varphi^f_{out}(t;t_f)$ and this equation is equal to equation (1.8). This shows the equality between Rose's theory and the Marchenko equation.

## 2.3 Implementation of Rose in Matlab

The most efficient way to implement Rose's procedure in Matlab is to use that a convolution in the time domain is the same as a multiplication in the frequency domain. The iterative procedure can therefore be programmed as a series of matrix multiplications and fast-Fourier transforms.

To implement Rose in Matlab it is necessary to transfer the equations from a continuous state to a discrete state. Let's evaluate equation (2.3), and replace $\varphi^f_{out}$ with $W$.

$$W(t,t_f) = R_0(t+t_f) - \int_{-t_f}^{t_f} R_0(t+t')W(t',t_f)dt. \tag{2.4}$$

Now since computers work with discrete data, let's rewrite the formula to a discrete equation. Define $t_f$ as $k\triangle t$, $t$ as $n\triangle t$, $t'$ as $m\triangle t$, and $-t_f < t, t' \leq t_f$.

$$W(n,k) = R_0(n+k) - \sum_{m=-n}^{k} R_0(n+m)W(m,k). \tag{2.5}$$

Note that $-k+1 \leq n, m \leq k$. This calculation can now be programmed in Matlab, however it can be rewritten to a more efficient state. When the equation is written like equation (2.5) the

calculation requires an additional for-loop because for each n, the sum of all the different m needs to be calculated. While in the frequency domain it is just a multiplication. Therefore it is faster to do a Fourier transform and make it a multiplication. Equation (2.5) can be rewritten to the following formula for every $k$.

$$W_1(n,k) = R_0(n+k) - \sum_{p=0}^{n+k} R_0(p) W_0(p-n,k). \tag{2.6}$$

Here is $p$ equal to $m+n$. This equation is equal to equation (2.5) after a certain amount of iterations when $W_1(n,k)$ is equal to $W_0(p-n,k)$. The equation is a convolution between $R_0(n)$ and $\delta(n+k)$, also between $R_0(n)$ and $W_0(p-n,k)$. So applying Fourier transforms gives:

$$W_1(n,k) = \mathrm{IFFT}(R_{0f}(n) * (e^{+iwk} - \mathrm{FFT}(W_0(m,k)))). \tag{2.7}$$

In this equation is $R_{0f}$ the fast Fourier transform of $R_0$, FFT stands for fast-Fourier transform, and IFFT for inverse fast-Fourier transform. There are various ways to program equation 2.7, next is shown how it is done in this thesis.

First define a variable $R_c(1:4t_{fmax}) = 0$, $R_c(1:3t_{fmax}) = R(1:3t_{fmax})$, whereby $R$ the first convolution is between $R_0$ and the original signal. Let $R_{0f}$ be the fast-Fourier transform of $R_c$, and $W_0(1) = R_c(-t_f+1)$, $W_0(2) = R_c(-t_f+2)$ etcetera. Since this is for a computer program, $R_c(n<0)$ does not exist, therefore is $R_c(0) = R_c(4t_{fmax})$,$R_c(-1) = R_c(4t_{fmax}-1)$ and so on, and finally define $W_0(t_f : 4t_{fmax} - t_f + 1) = 0$. Basically the meaningful data is loaded into $W_0(-t_f : t_f - 1)$, the signal length is $2t_f$, and all other traces zero. The data from $R(1:2t_f)$ is shifted $t_f + 1$ places to the left in $W_0$. So the data containing traces are: $W_0(4t_{fmax} - t_f : 4t_{fmax})$ and $W_0(1 : t_f - 1)$. Now the iterative process is as follows:

$$W_1 = \mathrm{IFFT}(R_{0f}(e^{-iw(4t_{fmax}-t_f+1)} - \mathrm{FFT}(\mathrm{circshift}(\mathrm{fliplr}(W_0),1)))), \tag{2.8}$$

$$W_1(t_f : 4t_{fmax} - t_f + 1) = 0, \tag{2.9}$$

$$W_0 = W_1. \tag{2.10}$$

Repeat this until $W_1$ is accurate enough, then $a = \mathrm{fftshift}(W_0$ and $K(tf,:) = a(t_{fmax}+1 : 3t_{fmax})$. Repeat the whole procedure for a new focus time $t_f$ until the whole medium is focused. The definition of accurate is that $\sqrt{(\sum (W_0 - W_1))^2/(\sum (W_0))^2}$ is smaller than $10^{-3}$. FFT stands for fast-Fourier transform, IFFT for inverse fast-Fourier transform, fftshift swaps the left and right half of a vector, circshift$(A,1)$ shifts all numbers in vector A in a circular way one place to the right; fliplr flips a matrix in the left right dimension. Note that fliplr and circshift together flip a matrix in such a way that if it is a function that $t$ becomes $-t$ with the original zero point still in place. In matrix $K$ the signal is focused at $t = t_f$. The code as it is programmed in Matlab is in the last section of the appendix. The next three figures are plots of $K$ for $t_f = 0.2;\ 0.4;\ 0.6$ of the first model. These figures all show a large amount of zero's, the first figure shows a lot more zero's than the later figure, because the length of the signal that contains data increases when $t_f$ increases, but the total length of the signal is constant.

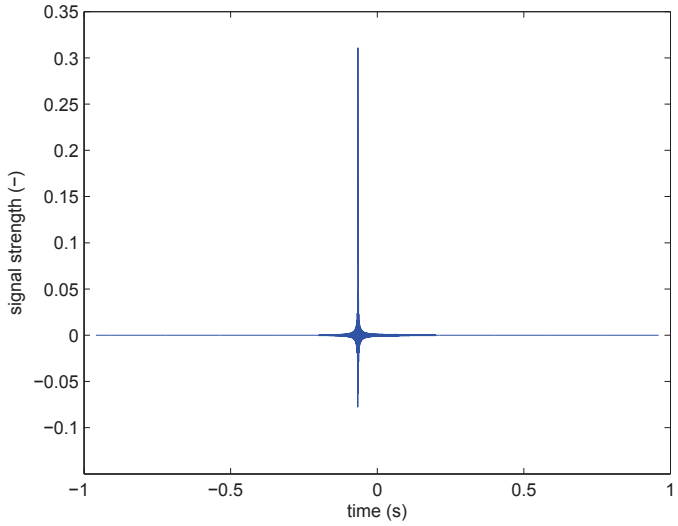


Figure 2.1: $K$ with $t_f = 0.2$ seconds.

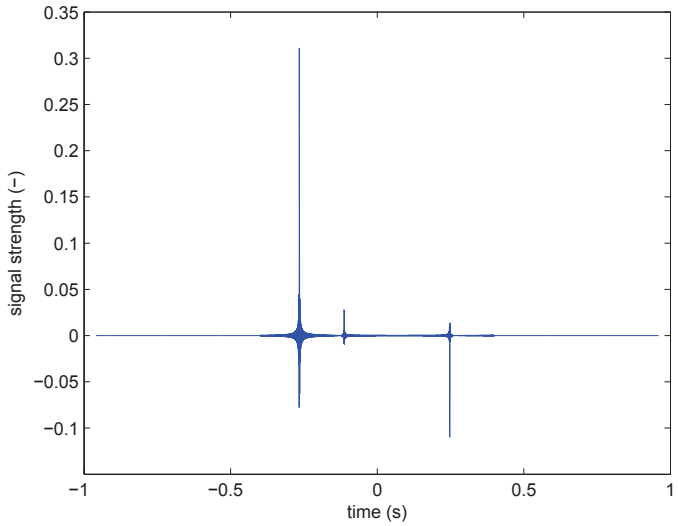


Figure 2.2: $K$ with $t_f = 0.4$ seconds.

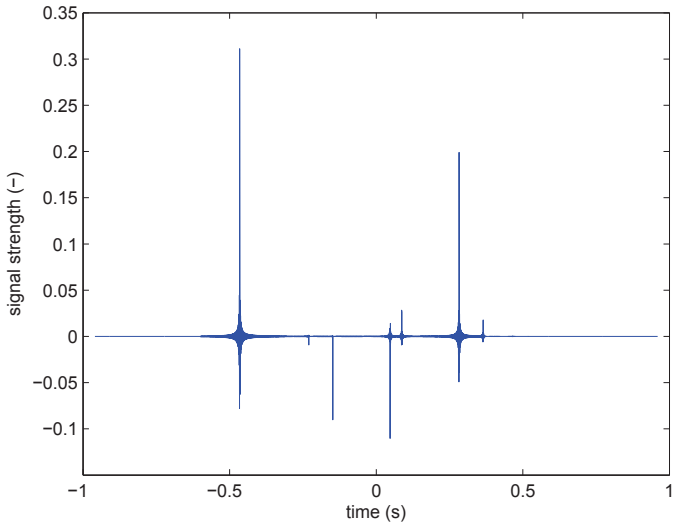


Figure 2.3: $K$ with $t_f = 0.6$ seconds.

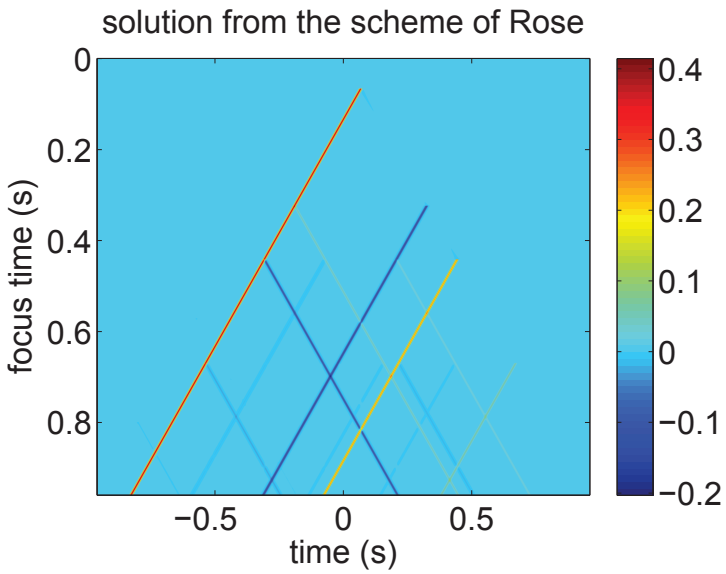


Figure 2.4: Overview of all $W_1$ for model 1.

When comparing figure 2.3 with figure 1.5 it turns out that $K$ with a focus time of 0.6 seconds contains more reflections than the original reflection profile. However when every input in matrix $K$ is plotted in a 2D figure, see figure 2.4, it is clear that the autofocused data is along the $t = t_f$ line. The reflection profile only focuses at $t_f$, every other part has a different meaning. Therefore to get a focused profile for all t, all the data along the $t = t_f$ line needs to be collected in one vector. Remember that in the iteration scheme the reflections are ignored after $t_f$ and the incident pulse crosses the origin at $t = -t_f$, so the reflections can only appear within the $t = t_f$ and $-t = t_f$ lines. The colored lines indicate the reflections, visible in figures 2.1, 2.2, and 2.3.

# Chapter 3

# Results of the three models

The theory of Rose, implemented in a Matlab file, is used on the three models described chapter 1. The results follow in this chapter. The purpose of focusing is to bring back the reflections from two-way travel time to one-way travel time and removing the multiples. For all the models and the procedure of Rose the corresponding Matlab-files can be found in the appendix. In the following figures the primary reflection series in one-way travel time is plotted against the auto-focused reflection series. Therefore the difference between the plotted reflection series should be zero if the autofocusing procedure functions correctly. The difference in appearance of the reflection series in figures 3.1, 3.2, 3.3 and in figures 1.5, 1.7, 1.9 is just a cosmetic difference. The data of the pictures in this chapter is multiplied by the second derivative of a Gaussian pulse.

Figure 3.1 corresponds to model 1. In this figure it is clearly visible that the focused reflection series matches exactly with the primary reflection series. All the multiple reflections are removed by the iteration scheme.
Figure 3.2 shows the reflection series of model 2. Also the match between the autofocused series and the primary reflection series is perfect. Even the seven primary reflections can be distinguished, the reflections from the top and bottom interfaces of the thin layers are overlapping.
In figure 3.3 the reflection series of model 3 are visible. Again the two reflection series match perfectly, and the reflections of the thin layers are clearly visible.

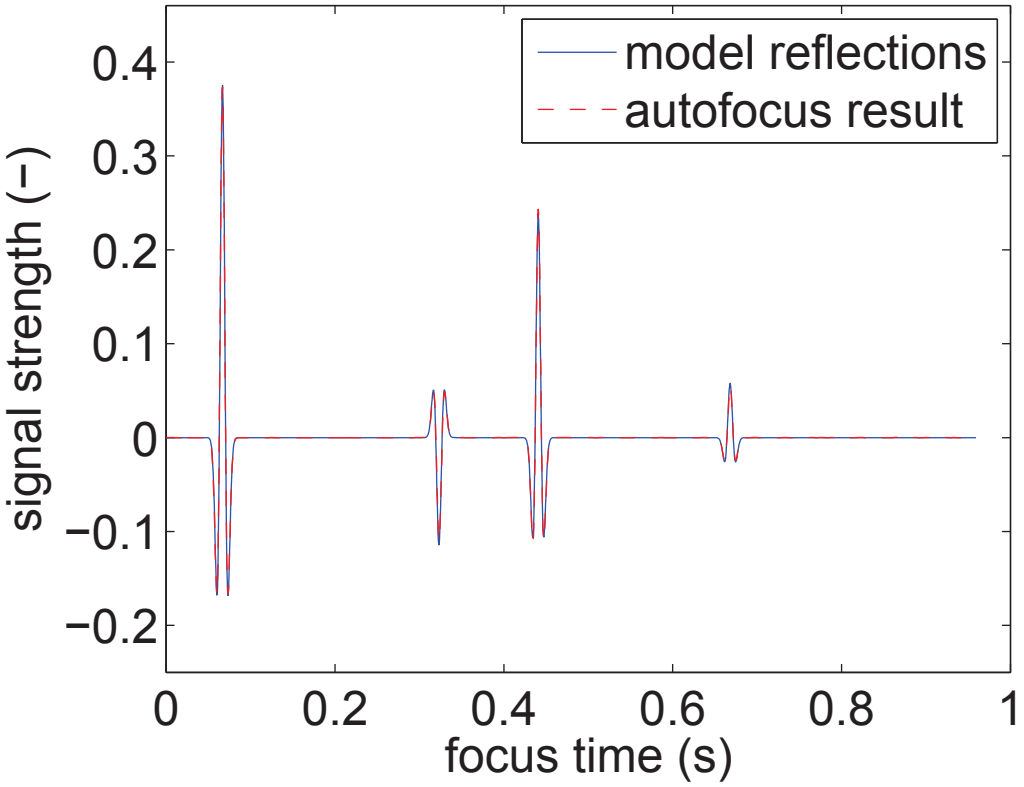


Figure 3.1: Focused reflection series of the first model plotted against the primary reflection series in one-way travel time.

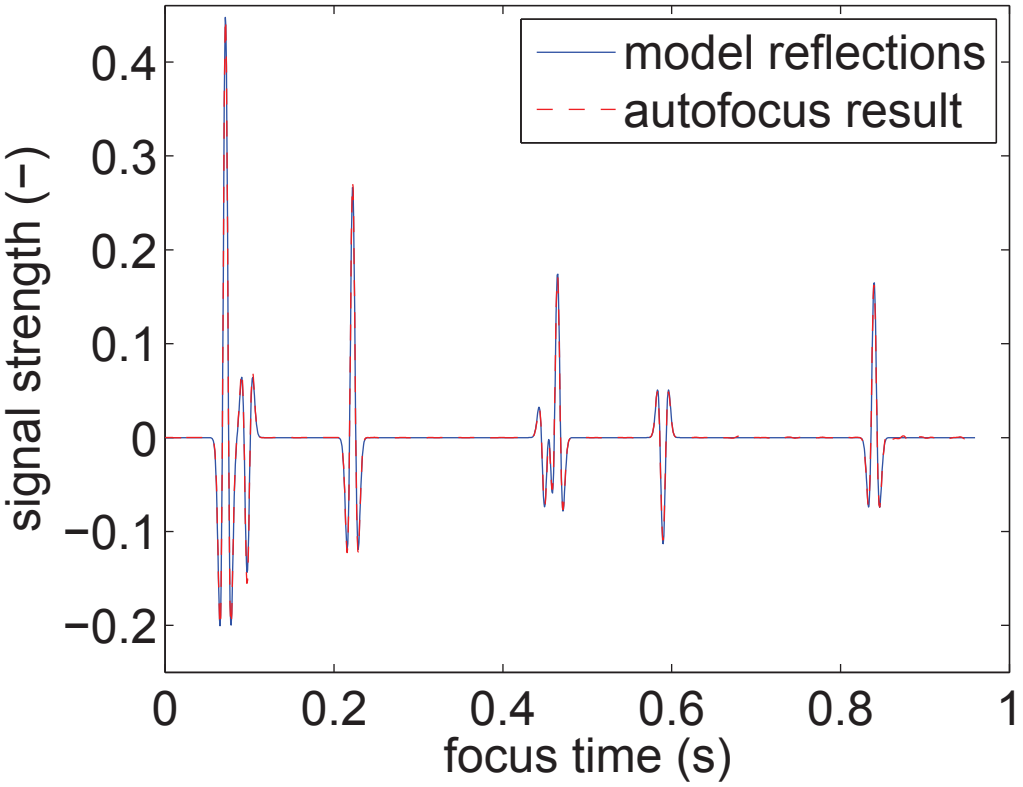


Figure 3.2: Focused reflection series of the second model plotted against the primary reflection series in one-way travel time.

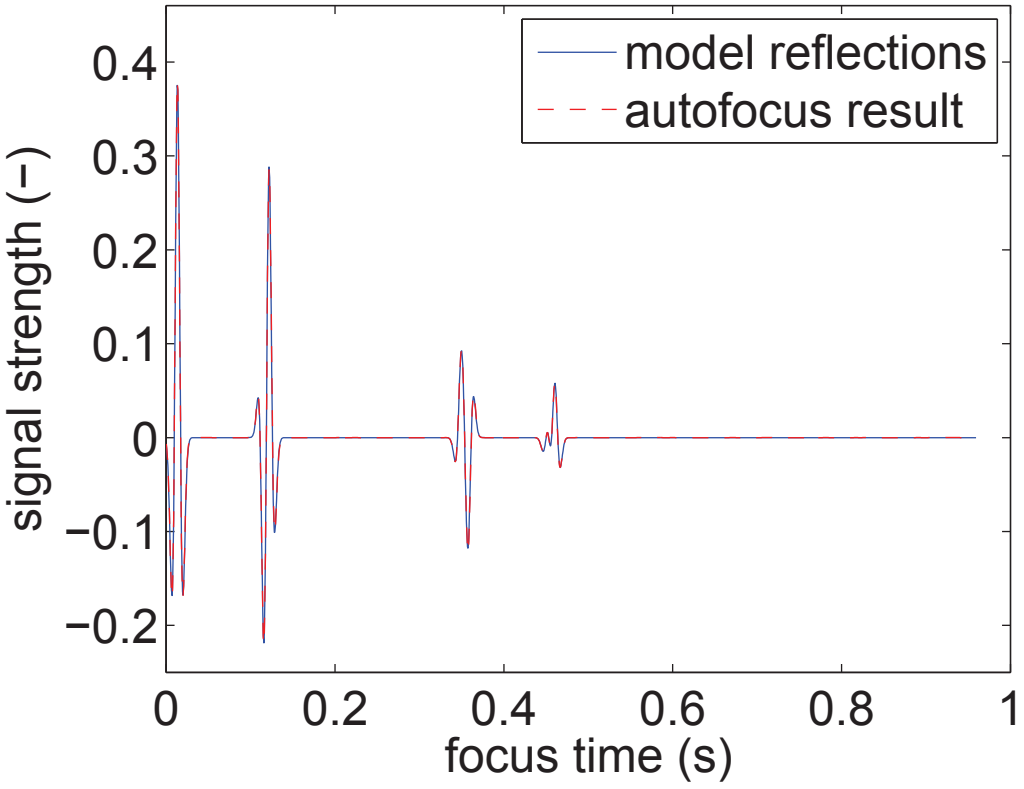


Figure 3.3: Focused reflection series of the third model plotted against the primary reflection series in one-way travel time.

# Chapter 4

# Conclusion and Discussion

## 4.1 Conclusion

The theory of Rose can indeed focus the tested models very accurate, thereby reducing the travel time from two-way travel time to one-way travel time and removing all the multiple reflections. Autofocusing the data does not result in a loss of details. Reflections of two layers that are close together are afterwards still recognizable as two different reflections. Furthermore the properties of the layered medium remain unknown, but the interfaces are now known as a function of one-way travel time and the local reflection coefficients are known. From the local reflection coefficients, the impedance of the layers can be determined if the impedance of the half space where the source and receiver are located is known.

## 4.2 Discussion

The autofocusing procedure of Rose is only solved for the delta pulse as incident pulse. Nothing can be told about the behavior of the focusing procedure for a Gaussian pulse for instance, especially the behavior for certain thin layers, thinner than the length of the incident pulse, when they are focused by the autofocusing procedure. According to Rose [2] it is possible to retrieve the acoustic impedance because autofocusing measures the solution to Marchenko's equation which is the hardest step of the 1D inverse scattering theory, and the velocity profile can be retrieved by solving an ordinary differential equation [6]. For a next investigation these are some ideas that can probably be solved relatively easy.

## 4.3 Acknowledgment

I thank Dr.ir. E.C. Slob for his guidance.

# Bibliography


[1] P. Goupillaud, "An approach to inverse filtering of near-surface layer effects from seismic records," *Geophysics*, vol. 26, no. 6, p. 754, 1961.

[2] J. Rose, "'single-sided' autofocusing of sound in layered materials," *Inverse problems*, vol. 18, pp. 1923–1934, 2002.

[3] P. C. and F. M., "Eigenmodes of the time reversal operator: a solution to selective focusing in multiple-target media," *Wave Motion*, vol. 20, pp. 151–63, 1994.

[4] T. J.-L. Prada C. and F. M., "The iterative time-reversal process: analysis of convergence," *J. Acoust. Soc. Am.*, vol. 97, p. 62, 1995.

[5] S. D. Prada C., Manneville S. and F. M., "Decomposition of the time reversal operator: application to detection and selective focusing on two scatterers," *J. Acoust. Soc. Am.*, vol. 99, p. 2067, 1996.

[6] T. Aktosun and J. Rose, "Wave focusing on the line," *J. Math. Phys.*, vol. 43, p. 37171, 2002.

# Chapter 5

# Appendix

## 5.1 The first model

```
clear all
close all
clc

c0=1500;
rho0=1000;
kap0=1/(rho0*c0^2);
nz=1001;
vel=[1500 1950 1700 2200 2350];
rho=[1000 1700 1550 1950 2050];
h=[100 500 200 500 1e100];
nl=length(h);
kap=1./(rho.*vel.^2);
Z=sqrt(kap./rho);
r=(Z(1:nl-1)-Z(2:nl))./(Z(1:nl-1)+Z(2:nl));
r(nl)=0;
nx=512;
x3=linspace(0,2e3,nx);
dz=x3(2);
t3=zeros(1,nx);
impx(1,1:nx)=1/sqrt(Z(1));
nh=round(h/dz);
t3(1:nh(1))=(1:nh(1))*dz/vel(1);

Rose
```

## 5.2 The second model

```
clear all
close all
clc
c0=1400;
rho0=900;
kap0=1/(rho0*c0^2);
nz=1001;
vel=[1400 1950 1600 2200 1950 2400 2000 2500];
rho=[900 1700 1550 1950 1900 2200 2100 2350];
h=[100 50 200 500 30 300 500 1e100];
nl=length(h);
kap=1./(rho.*vel.^2);
Z=sqrt(kap./rho);
r=(Z(1:nl-1)-Z(2:nl))./(Z(1:nl-1)+Z(2:nl));
r(nl)=0;
nx=512;
x3=linspace(0,2e3,nx);
dz=x3(2);
t3=zeros(1,nx);
impx(1,1:nx)=1/sqrt(Z(1));
nh=round(h/dz);
t3(1:nh(1))=(1:nh(1))*dz/vel(1);

Rose
```

## 5.3 The third model

```
clear all
close all
clc

c0=1500;
rho0=1000;
kap0=1/(rho0*c0^2);
nz=1001;
vel=[1500 1950 1700 2200 2350 2100 2000 2250];
rho=[1000 1700 1550 1950 2050 1880 2100 2150];
h=[20 200 10 500 20 200 15 1e100];
nl=length(h);
kap=1./(rho.*vel.^2);
Z=sqrt(kap./rho);
```

```
r=(Z(1:nl-1)-Z(2:nl))./(Z(1:nl-1)+Z(2:nl));
r(nl)=0;
nx=512;
x3=linspace(0,2e3,nx);
dz=x3(2);
t3=zeros(1,nx);
impx(1,1:nx)=1/sqrt(Z(1));
nh=round(h/dz);
t3(1:nh(1))=(1:nh(1))*dz/vel(1);

Rose
```

## 5.4 Rose

```
for il=2:nl-1
    t3(sum(nh(1:il-1))+1:sum(nh(1:il)))=t3(sum(nh(1:il-1)))+ ...
        (1:nh(il))*dz/vel(il);
    impx(sum(nh(1:il-1))+1:sum(nh(1:il)))=1/sqrt(Z(il));
end
t3(sum(nh(1:il))+1:nx)=t3(sum(nh(1:il)))+(1:nx-sum(nh(1:il)))*dz/vel(nl);
impx(sum(nh(1:il))+1:nx)=1/sqrt(Z(nl));
fc=60;
nf=16*8192;
freq=linspace(0,800,nf);
df=freq(2)-freq(1);
freq=freq-1i*df;
fc=60;
wav=2*sqrt(1/pi)*(freq/fc).^2.*exp(-(freq/fc).^2)/fc;
R0=zeros(1,nf);
P=zeros(1,nf);
for ii=nl-1:-1:1
    gam=2i*pi*freq/vel(ii+1);
    R0=(r(ii)+R0.*exp(-2*gam*h(ii+1)))./(1+r(ii)*R0.*exp(-2*gam*h(ii+1)));
    P=(r(ii+1)+P).*exp(-gam*h(ii+1));
end
gam=2i*pi*freq/vel(1);
R0t=2*real(ifft(R0.*exp(-2*gam*h(1)),2*nf));
P=(r(1)+P).*exp(-gam*h(1));
Pt=2*real(ifft(P.*wav,2*nf))*df*2*nf;
t=(0:nf-1)/(2*nf*df);
dt=t(2)-t(1);
Pt=Pt(1:nf).*exp(2*pi*t*df);
R0t=R0t(1:nf).*exp(2*pi*t*df);
```

```
figure(3)
plot(t,R0t(1:nf))
axis([0 4 -0.25 0.75])
kids=get(gca,'children');
set(kids,'linewidth',2)
set(gca,'fontsize',18)
xlabel('two-way travel time (s)')
ylabel('reflection response')
title('convolved with Gaussian')
nfc=nf/128+nf/256;
nt=2*nfc;
k=zeros(nfc,nt);
ds=1/(2*nt*dt);
s=2i*pi*fftshift(-nt:nt-1)*ds;
tic
img=zeros(1,nfc);
for tf=1:nfc
    Rc=zeros(1,2*nt);
    Rc(1:nt+nfc)=R0t(1:nt+nfc);
    R0f=fft(Rc);
    w0=Rc;
    w0=circshift(w0,[0 -tf+1]);
    w0(tf:2*nt-tf+1)=0;
    diff=1;
    ii=0;
    while diff > 1e-3
        ii=ii+1;
        if floor(ii/30)==ii/30
            disp(['ii= ',num2str(ii,4)])
        end
        w1=real(ifft(R0f.*(exp(s*(tf-1)*dt)-fft(circshift(fliplr(w0),[0 1])))));
        w1(tf:2*nt-tf+1)=0;
        diff=sqrt(sum(abs(w0-w1).^2)/sum(abs(w0).^2));
        w0=w1;
    end
    tmp=fftshift(w0);
    k(tf,:)=tmp(nt-nfc+1:nt+nfc);
end
df=1/(2*nfc*dt);
freq=df*fftshift(-nfc:nfc-1);
fc=60;
% wav=2*sqrt(1/pi)*(freq/fc).^2.*exp(-(freq/fc).^2)/fc;
wav=sqrt(1/pi)*exp(-(freq/fc).^2)/(2*fc);
for tf=1:nfc
```

```
    tk(tf,:)=2*real(ifft(wav.*fft(k(tf,:)),2*nfc))*2*nfc*df;
end
figure(1)
imagesc((-nfc+1:nfc)*dt,t(1:nfc),tk(:,1:2*nfc))
set(gca,'fontsize',18)
xlabel('time (s)')
ylabel('focus time (s)')
title('solution from the scheme of Rose')
colorbar
for tf=1:nfc
    img(tf)=k(tf,nfc+tf-1)+k(tf,nfc+tf-2);
end
toc
fimg=fft(img,2*nfc);
fimg=fimg(1:nfc);
freq=df*(0:nfc-1);
fc=60;
% wav=2*sqrt(1/pi)*(freq/fc).^2.*exp(2i*pi*freq*dt-(freq/fc).^2)/fc;
wav=2*sqrt(1/pi)*(freq/fc).^2.*exp(2i*pi*freq*dt/sqrt(2)-(freq/fc).^2)/fc;
pimg=2*real(ifft(wav.*fimg,2*nfc))*2*nfc*df;
figure(2)
plot(t(1:nfc),Pt(1:nfc),'b',t(1:nfc),pimg(1:nfc),'r--')
set(gca,'fontsize',18)
xlabel('focus time (s)')
ylabel('signal strength (-)')
legend('model reflections','autofocus result')
```