There are diferent ways to get the source code for the Marche
a tagged snapshot from the GitHub repository. These tagged
The latest tagged version that can be cloned from the reposi
git clone -b '2.1.1' --single-branch git://github.com/Ja
and contains the full package. To get a version with the late
of the most current version
git clone git://github.com/JanThorbecke/OpenSource.git
The package extracts in OpenSource, which contains the following sub-director

- FFTlib: basic library for FFT's includes a wrapper for MK
- MDD: Multiple Dimensional Deconvolution to solve $Ax = b$ problems $x = \frac{bA^*}{AA^*}$ solve differe
- corrvir: seismic interferometry (correlation) for pass
- doc: documentation related to the code.
- extrap: recursive wavefeld depth extrapolation, includ
- extrap3d: 3D version of the above.
- fdacrtmc: RTM based on fdelmodc.
- fdelmodc: fnite diference modeling (visco)-acoustic, a
- fdelmodc3D: 3D version for acoustic media.
- fdemmodc: EM fnite diference code.
- marchenko: basic, plane-wave and MME implementations.
- marchenko3D: 3D version of the basic algorithm.
- raytime: eikonal solver.
- raytime3D: 3D eikonal solver.
- utils: basic (pre-) processing and additional programs f
- zfp: ZFP data compression library from Peter Lindstrom.

Besides the Marchenko algorithms the OpenSource package c
manual we will only describe the different Marchenko implemen
marchenko, marchenko3D, utils.
The INSTALL file in the ROOT directory contains guidelines how to
README file briefly explains the different code packages and how to r
papers. Section of this manual contains a brief (one-sentence) ex
Marchenko source code files in the source tree of this packag
The code is used by many different people and new options are b

1. To compile and link the code you first have to set the ROOT
   which can be found in the directory. README and INSTALL have to be ...
2. Check the compiler and CFLAGS options in the file Make_inc
   are using. The default options are set for a the GNU C-com
   g++ compiler is only needed to compile the MDD code. The c
   been compiled and tested with several versions of GNU, AM
3. If the compiler options are set in the Make_include file yo

   > make

   and the Makefile will execute the commands to compile and
   directories.

The compiled FFT and ZFP libraries will be placed in lib/ the executables
directory and the include files of the FFT and ZFP libraries i
To use the executables don't forget to include the pathname

bash:
export PATH='path_to_this_directory'/bin:$PATH:
csh:
setenv PATH 'path_to_this_directory'/bin:$PATH:

On Linux systems using the bash shell you can put export PATH='path_to_this_direc
setting in $HOME/.bashrc to set it every time you login.
Other useful make commands are:

- make clean: removes all object files, but leaves libraries and
- make realclean: removes also object files, libraries and execut

Important Note: The examples and demo scripts make use of SU programs. Please
make sure that SU is compiled with -DSU_XDR flag in $CWPROOT/Makefile.co
must NOT be set in compiling SU. The SU output files of fdelmodc
When the XDR flag is set in SU you have to convert the output fil
in the utils directory: basop, fconv, extendModel, makemod
SU programs.

### 1.3.1 Finite Difference for Seismic Interferometry

If the compilation has finished without errors and produced
run one of the demo programs by running

> ./fdelmodc_plane.scr

in the directory fdelmodc/demo. This /demo directory contains many script
different possibilities of the modeling program.

- To reproduce the Figures shown in the Geophysics manuscript "Finite-difference modeling experiments for seismic interferometry" (Thorbecke and Draganov, 2011) the scripts in FiguresPaper directory can be used. Please read the README i
  instructions and guidelines.

To clean-up all the produced output files in fdelmodc/FiguresPaper/ you can run the clean script in those directories.
An extensive manual of fdelmodc can be found in fdelmodc/fdelmodcManual.pdf

### 1.3.2 Marchenko: basic, plane-waves and MME

If the compilation has finished without errors and produced a
can run one of the demo programs by running a set of scripts t
of the directory marchenko/demo/oneD

- To reproduce the Figures shown in the Geophysics paper "Implementation of the March
  method" the scripts in marchenko/demo/oneD directory can be used. The READM
  directory gives more instructions and guidelines.
- To reproduce the Figures shown in the Scientific Reports "Virtual acoustic sources in i
  mogeneous media with single-sided access" the scripts in marchenko/demo/Sc
  directory can be used. The README in this directory gives
- To reproduce the Figures shown in the Geophysics paper "Implementation of the March
  Multiple Elimination algorithm" the scripts in marchenko/demo/mme can be use
  The README_PRIMARIES in this directory gives more instru
- To reproduce the Figures shown in the Geophysics paper "Implementation of the 3D Marc
  method" the scripts in marchenko3D/demo/marchenko3D can be used. The
  in this directory gives more instructions and guidelines
- To reproduce the Figures shown in the IEEE paper "Design, implementation and
  application of the Marchenko Plane-wave algorithm" the scripts in marchenko/demo/plan
  directory can be used. The README in this directory gives

A brief manual about the MME program 'marchenko_primaries'

### 1.3.3 MDD: target replacement

- To reproduce the Figure's shown in the paper "New ..." the scripts in MDD/demo/
  directory can be used. The README in this directory gives

0. DOI reference of this software release
   https://zenodo.org/badge/latestdoi/23060862
1. If the Finite Diference code has helped you in your resear
   your publications:

   Finite-diference modeling experiments for seismic inte
   Jan Thorbecke and Deyan Draganov, 2011, Geophysics, Vol
   H1–H18, doi: 10.1190/GEO2010-0039.1 Download: pdf
2. If the Marchenko code has helped you in your research pleas

   Implementation of the Marchenko method,
   Jan Thorbecke, Evert Slob, Joeri Brackenhof, Joost van
   Geophysics, Vol. 82, no. 6 (November-December); p. WB29–
   Download: pdf

3. If you used the code to construct homogeneous Green's fun
related publications:

Virtual sources and receivers in the real Earth: Conside
tions,
Brackenhof, J., Thorbecke, J., and Wapenaar, K., 2019, J
Earth, Vol. 124, 11,802-11,821. pdf-fle

Virtual acoustics in inhomogeneous media with single-si
Wapenaar, K., Brackenhof, J., Thorbecke, J., van der Neut
Scientifc Reports, Vol. 8, 2497. Download: pdf

4. When you are using the marchenko_primaries algorithm dev
the following papers:

Free-surface and internal multiple elimination in one s
tion,
Lele Zhang and Evert Slob 2019, Geophysics, Vol. 84, no.
10.1190/GEO2018-0548.1 Download: pdf

Free-surface and internal multiple elimination in one s
tion,
Jan Thorbecke, Lele Zhang, Kees Wapenaar and Evert Slob
(March-April); p. xxxx, doi: xxxx Download: pdf

5. When you are using the plane wave versions of marchenko
developed by Giovanni Meles please refer to the following

Virtual plane-wave imaging via Marchenko redatuming:
Meles, G.A., K. Wapenaar, and J. Thorbecke, 2018, Geophy
(1), p. 508–519.

6. If you use the fdacrtmc code of Max Holicki please refer t

Acoustic directional snapshot wavefeld decomposition,
Holicki, M., Drijkoningen, G., and Wapenaar, K., 2019, A
decomposition: Geophysical Prospecting, Vol. 67, 32-51 pdf

7. If you use the vmar code of Johno van IJsseldijk please re

Extracting small time-lapse traveltime changes in a res
ternal multiples after Marchenko-based target zone isol
Van IJsseldijk, J., van der Neut, J., Thorbecke, J., and W
(2), R135-R143 Download: pdf

-7- A reference to the extrapolation and migration progra

Design of one-way wavefeld extrapolation operators, usi
optimisation.
Jan Thorbecke, Kees Wapenaar, Gerd Swinnen, 2004, Geophy
pdf

In this section we describe in detail the implementational as
Marchenko method based on focusing functions. Although the
the treatment of amplitudes, and the initialisation steps o
The input of the method is a refection response without free
source wavelet. The output of an SRME scheme can (in princip
(smooth) background model is needed to calculate an initia
The Numerical Examples section demonstrates the use of the a
with the Marchenko technique.

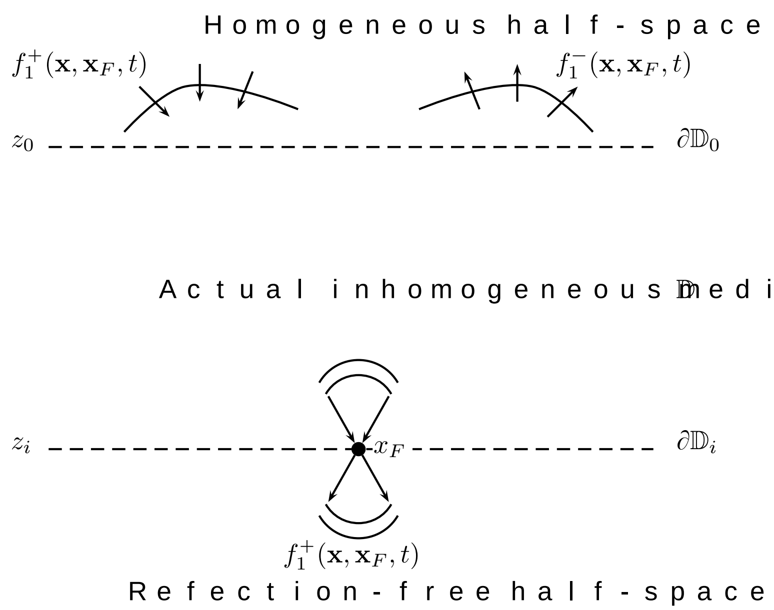



Figure 1: Downgoing and upgoing components of the $f_1$ focusing function of the 2D wave equation
truncated medium.

The Marchenko method is briefy introduced here aiming at an
understand the algorithm. The references mentioned in the
the derivation of this method. In an imaging medium at focal depth $z_i$ we introduce a focusing
function $f_1$. The truncated medium is identical to the actual medium above $z_i$ and refection-
free below this depth level (Figure 1). The actual and truncated media are
free above the surface $\partial\mathbb{D}_0$. We also introduce up- and downgoing parts of the focusing $f_1$
function (Slob et al., 2014).

$$f_1(\mathbf{x},\mathbf{x}_F,t) = f_1^+(\mathbf{x},\mathbf{x}_F,t) + f_1^-(\mathbf{x},\mathbf{x}_F,t),$$

where $x_F = (x_F, z_i)$ is a focal position on the boundary $\partial\mathbb{D}_i$, $\mathbf{x}$ an arbitrary point in the me
is time (see Figure 1). In our notation the frst argument represents th
argument stands for the focal point. The superscript $+$ denotes the downgoing $f_1^+$ feld at o
point $\mathbf{x}$, and the superscript $-$ the upgoing $f_1^-$ feld. Below $\partial\mathbb{D}_i$, $f_1^+$ continues
as a diverging downgoing feld into the refection-free half-
The $f_1^\pm$ focusing functions are defned to relate the up- and dow
medium with the refection response at the surface (Wapenaar et al., 2014).

$$G^+(\mathbf{x}_F,\mathbf{x}_R,t) = -\int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t} R(\mathbf{x}_R,\mathbf{x},t-t')f_1^-(\mathbf{x},\mathbf{x}_F,-t')\mathrm{d}t'\mathrm{d}\mathbf{x} + f_1^+(\mathbf{x}_R,\mathbf{x}_F,-t), \quad (1)$$

$$G^-(\mathbf{x}_F,\mathbf{x}_R,t) = \int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t} R(\mathbf{x}_R,\mathbf{x},t-t')f_1^+(\mathbf{x},\mathbf{x}_F,t')\mathrm{d}t'\mathrm{d}\mathbf{x} - f_1^-(\mathbf{x}_R,\mathbf{x}_F,t). \quad (2)$$

$R(\mathbf{x}_R,\mathbf{x},t)$ is the reflection response after surface multiple elim
the wavelet. The first argument of $R$ is the receiver location, the se
location, and the last argument is time. $R$ is a causal (t − 2) ... source wavefield at $\mathbf{x}_R$ of a vertical
force source $F_z$ at $\mathbf{x}$, or, via reciprocity, the particle velocity $V_z$ at a point source of volu
injection at $\mathbf{x}_R$ (Wapenaar et al., 2014a). This reflection response is related t

$$\frac{\partial R(\mathbf{x}_R,\mathbf{x},t)}{\partial t}=\frac{2}{\rho(\mathbf{x})}\frac{\partial G^s(\mathbf{x}_R,\mathbf{x},t)}{\partial z} \qquad (3)$$

with $G^s$ the Green's function of the scattered field only (it does
integration boundary $t'=0$ of the time integral in equations (1) and (2) accounts for the causal
reflection response $R(\mathbf{x}_R,\mathbf{x},t-t')$. Summing equations (1) and (2) using source-receiver re
the Green's function as (Wapenaar et al., 2014a)

$$G(\mathbf{x}_R,\mathbf{x}_F,t)=\int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t}R(\mathbf{x}_R,\mathbf{x},t-t')f_2(\mathbf{x}_F,\mathbf{x},t')\mathrm{d}t'\mathrm{d}\mathbf{x}+f_2(\mathbf{x}_F,\mathbf{x}_R,-t), \qquad (4)$$

The Green's function $G(\mathbf{x}_R,\mathbf{x}_F,t)$ represents the response to a virtual point
rate at $\mathbf{x}_F$ and pressure receiver at $\mathbf{x}_R$. The focusing function $f_2$ is defined as

$$f_2(\mathbf{x}_F,\mathbf{x},t)=f_1^+(\mathbf{x},\mathbf{x}_F,t)-f_1^-(\mathbf{x},\mathbf{x}_F,-t). \qquad (5)$$

Wapenaar et al. (2014a) introduced $f_2(\mathbf{x}_F,\mathbf{x},t)$ as a focusing function, which has its $\partial\mathbb{D}_0$.
Here we merely use $f_2$ as a compact notation for the combination of the $f_1^+$
and $f_1^-$, as defined in equation (5). Note that the time-reversed $f_1^-$ can be interpreted
as a downgoing function. Hence, $f_2(\mathbf{x}_F,\mathbf{x},t)$ is a downgoing
focusing function which is emitted at $\mathbf{x}$ and which focuses at $\mathbf{x}_F$ receiver
The argument change in equation (5) between $\mathbf{x}$ and $\mathbf{x}_F$ in the left-hand $f_2$ and the right-hand s
$f_1^{\pm}$ follows from the same logic in the original Wapenaar et al. (2014a) arguments a
the Numerical Examples section $f_2$ can be back-propagated into the
focuses at $\mathbf{x}_F$ and (Wapenaar et al., 2013) a (reciprocal) $f_2(\mathbf{x}_F,\mathbf{x},t)$ and downgoing
wave $p^+(\mathbf{x},\mathbf{x}_F,t)$ is given. Together with $p^-$ the field, the upgoing reflecti
at $\mathbf{x}$ from the focal point $\mathbf{x}_F$. The sum of $p^+$ and $p^-$ gives also the Green's function (4).
The $p^{\pm}$ functions are just a different notation of the Marchenko r
Green's functions in a convenient way. The $p^{\pm}$ functions are therefore used in the
compute the Green's function. From an educational point of v
understood by using the focusing functions only and we will
The above equations, on which the following implementation
Other papers derive similar equations (Wapenaar et al., 2013; Slob et al., 2014; Singh et al., normalis
2015; van der Neut et al., 2015b). The relationship between pressure- and fu
is explained in Wapenaar et al. (2014a).
The Marchenko algorithm estimates $f_1^+(\mathbf{x},\mathbf{x}_F,t)$ and $f_1^-(\mathbf{x},\mathbf{x}_F,t)$. In the Green's function equation
(1) and (2) are by themselves insufficient to determine the four unknowns: $f_1^{\pm}$, but onl
We can eliminate two unknowns by exploiting the fact that foc
different causality properties in the time-space domain. Ba
no energy arrives before $t_d(\mathbf{x}_R,\mathbf{x}_F)$, therefore the Green's function $G(\mathbf{x}_R,\mathbf{x}_F,t<t_d(\mathbf{x}_R,\mathbf{x}_F))$
is zero. This also holds for the up- and downgoing Green's fu

$$0=-\int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t}R(\mathbf{x}_R,\mathbf{x},t-t')f_1^-(\mathbf{x},\mathbf{x}_F,-t')\mathrm{d}t'\mathrm{d}\mathbf{x}+f_1^+(\mathbf{x}_R,\mathbf{x}_F,-t), \qquad (6)$$

$$0=\int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t}R(\mathbf{x}_R,\mathbf{x},t-t')f_1^+(\mathbf{x},\mathbf{x}_F,t')\mathrm{d}t'\mathrm{d}\mathbf{x}-f_1^-(\mathbf{x}_R,\mathbf{x}_F,t), \qquad (7)$$

where $t<t_d(\mathbf{x}_R,\mathbf{x}_F)$ in both equations above.
The combination of equations (6) and (7) is known as the Marchenko equation. The
basis of the iterative scheme, which estimates $f_1^+$ and $f_1^-$ (Wapenaar et al., 2014a) using fun
the relation

$$f_1^+(\mathbf{x},\mathbf{x}_F,t)=T^{inv}(\mathbf{x}_F,\mathbf{x},t), \qquad (8)$$

is used to derive an initial estimate $f_1^+$ that can start the inversion scheme (8). $T^{inv}(\mathbf{x}_F,\mathbf{x},t)$ is the inverse of the transmission response of the t the actual medium between $\partial\mathbb{D}_0$ and $\partial\mathbb{D}_i$, for a source at $\mathbf{x}$ (at $\partial\mathbb{D}_0$) and a receiver at $\partial\mathbb{D}_i$. It is assumed that this inverse transmission $T^{inv}$ can be composed as a direct wave follo coda (Van der Neut et al., 2015).

$$f_1^+(\mathbf{x},\mathbf{x}_F,t) = T_d^{inv}(\mathbf{x}_F,\mathbf{x},t) + M^+(\mathbf{x},\mathbf{x}_F,t), \tag{9}$$

where $M^+$ is the unknown scattering coda and $T_d^{inv}$ the direct arrival of the inverse tr In equation (9) the inverse of the direct arrival of the transmissio take the time-reversal of the direct arrival of the Green's $G_d(\mathbf{x},\mathbf{x}_F,-t)$.

$$f_1^+(\mathbf{x},\mathbf{x}_F,t) \approx G_d(\mathbf{x},\mathbf{x}_F,-t) + M^+(\mathbf{x},\mathbf{x}_F,t). \tag{10}$$

We thereby introduce an overall scaling error and an ofset t to transmission losses in the time-reversed direct arr $G_d(\mathbf{x},\mathbf{x}_F,-t)$ transmission response to a subsurface focal point $\mathbf{x}_F$ can be computed fr macromodel. As mentioned before, $G_d(\mathbf{x},\mathbf{x}_F,t)$ is the direct arrival $t_d$, $G_d(\mathbf{x},\mathbf{x}_F,t)$ is zero before $t<t_d$. The multiple scattering coda $M^+(\mathbf{x},\mathbf{x}_F,t)$ follows after the first arrival of $f_1^+$, and is zero for $t \leqslant -t_d$. It can also be shown that (Slob et al., 2014) zero for $t \geqslant +t_d$. Equations (6) and (7) are only valid for $t<t_d$. We therefore define a time-window fu

$$\theta(\mathbf{x}_R,\mathbf{x}_F,t) = \begin{cases} 1 & t < t_d^{\varepsilon} \\ \frac{1}{2} & t = t_d^{\varepsilon} \\ 0 & t > t_d^{\varepsilon} \end{cases}. \tag{11}$$

where $t_d^{\varepsilon}$ is the time of the direct arrival $t_d(\mathbf{x}_F,\mathbf{x}_R)$ minus a small positive constant $\varepsilon$ to exclude the wavelet in $G_d$. For example, a time-window that mutes times $t<-t_d^{\varepsilon}$ to zero and applied to equation (10) mutes $G_d(-t)$, but leaves $M^+$ intact. In the following we will use $\theta_t$ for $\theta(\mathbf{x}_R,\mathbf{x}_F,t)$. In the implemented Marchenko pr is an input parameter (see Appendix A for all input parameters) tapering in this mute-window to suppress high frequency art It is further assumed that it is possible to get an estimate of $G_d$ of the transmission response. Given the reflection response $R(\mathbf{x}_R,\mathbf{x},t)$ and the direct arrival $G_d(\mathbf{x},\mathbf{x}_F,t)$ from the focal point, the Marchenko algorithm solves for the coda $M^+(\mathbf{x},\mathbf{x}_F,t)$ of $f_1^+(\mathbf{x},\mathbf{x}_F,t)$ and the scattering co $f_1^-(\mathbf{x},\mathbf{x}_F,t)$.

The iterative solution of the Marchenko equations can now be with the following initialization of $M^+$;

$$M_0^+(\mathbf{x}_R,\mathbf{x}_F,t) = 0. \tag{12}$$

The subscript of $M_0^+$ defines the iteration number. By using (10) and substituting (12) into equation (7), we arrive at the initialisation of $f_1^-$:

$$f_{1,0}^-(\mathbf{x}_R,\mathbf{x}_F,t) = \theta_t \int_{\partial\mathbb{D}_0} \int_{t'=-\infty}^{t} R(\mathbf{x}_R,\mathbf{x},t-t')G_d(\mathbf{x},\mathbf{x}_F,-t')\mathrm{d}t'\mathrm{d}\mathbf{x}. \tag{13}$$

Equation (13) includes the previously defined $\theta_t$. Equation (10) and (6) are now both functio expressions for $f_1^+$. By combining these equations the $M^+$ part in (10) is. The remain iterative update for $M^+$ at step $k \geqslant 1$ is given by

$$M_k^+(\mathbf{x}_R,\mathbf{x}_F,-t) = \theta_t \int_{\partial\mathbb{D}_0} \int_{t'=-\infty}^{t} R(\mathbf{x}_R,\mathbf{x},t-t')f_{1,k-1}^-(\mathbf{x},\mathbf{x}_F,-t')\mathrm{d}t'\mathrm{d}\mathbf{x}. \tag{14}$$

Following the assumption (10) that it is possible to write $f_{1,k}^+$ as a direct field plus sca coda, the update of $f_{1,k}^+$ at step $k$ is given by

$$f_{1,k}^+(\mathbf{x}_R,\mathbf{x}_F,t) = G_d(\mathbf{x}_R,\mathbf{x}_F,-t) + M_k^+(\mathbf{x}_R,\mathbf{x}_F,t). \tag{15}$$

Using equation (7) and the expression for $f_1^+$ in equation (15), the update of $f_1^-$ at step $k$ is given by

$$f^-_{1,k}(\mathbf{x}_R, \mathbf{x}_F, t) = f^-_{1,0}(\mathbf{x}_R, \mathbf{x}_F, t) + \theta_t \int_{\partial\mathbb{D}_0} \int_{t'=-\infty}^{t} R(\mathbf{x}_R, \mathbf{x}, t - t') M^+_k(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x}. \quad (16)$$

This completes the defnition of the iterative Marchenko sche
are discussed in detail and illustrated with simple numeric

To compute focusing functions $f_1$ with the Marchenko method two ing

- Reflection data without free-surface multiples $R(\mathbf{x}_R, \mathbf{x}, t)$ ghosts a with sources $\mathbf{x}$ and receivers $\mathbf{x}_R$ over the same surface $\partial\mathbb{D}_0$, and a small enough sampling fo to avoid spatial aliasing.

- An estimate of the direct arrival between $\mathbf{x}_R$ and the focal point $\mathbf{x}_F$: $G_d(\mathbf{x}_R, \mathbf{x}_F, t)$, and derived from it the travel time $t_d(\mathbf{x}_R, \mathbf{x}_F, t)$. Note that $t_d$ can also be computed by another method, for example an eik

Given these two components the iterative method can be initi method can start.

The initialisation of the method is given by equations (12) and (13). The time-windowed expre $f^-_{1,0}(\mathbf{x}_R, \mathbf{x}_F, t)$ in equation (13) is renamed to:

$$-N_0(\mathbf{x}_R, \mathbf{x}_F, -t) = \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t-t') G_d(\mathbf{x}, \mathbf{x}_F, -t') \mathrm{d}t' \mathrm{d}\mathbf{x}. \quad (17)$$

At each iteration, the spatial integration and temporal convolution with $R$ is used to define new focus $N_i$ (suggested as a possible approach by Wapenaar et al. (2014b)). The $N_i$ terms are used to update the estimates of $f^+_1$ and $f^-_1$. Although these $N_i$ terms are strictly not needed to describe the method, they possible to the actual implementation.

For computational efficiency, the implementation of the convolution with $R$ is done in the Fourier spatial integration is carried out by summing the resulting receiver gather. The introduced $t < t^\epsilon_d$ time-window removes events after (17). Applying the mute-window is therefore a crucial and man without it the method would be incorrect.

Given these initialisations the first steps (14)-(16) in the algorithm can be computed. This first step, $k = 1$, involves two integrations of $R$ with the up- and downgoing $f_1$'s with

$$\begin{aligned} M^+_1(\mathbf{x}_R, \mathbf{x}_F, -t) &= \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t-t') f^-_{1,0}(\mathbf{x}, \mathbf{x}_F, -t') \mathrm{d}t' \mathrm{d}\mathbf{x} \quad (18) \\ &= -\theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t-t') N_0(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x} \\ &= N_1(\mathbf{x}_R, \mathbf{x}_F, -t), \\ f^+_{1,1}(\mathbf{x}_R, \mathbf{x}_F, t) &= G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + M^+_1(\mathbf{x}_R, \mathbf{x}_F, t) \\ &= G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + N_1(\mathbf{x}_F, \mathbf{x}_R, t), \quad (19) \\ f^-_{1,1}(\mathbf{x}_R, \mathbf{x}_F, t) &= f^-_{1,0}(\mathbf{x}_R, \mathbf{x}_F, t) + \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t-t') M^+_1(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x} \\ &= -N_0(\mathbf{x}_R, \mathbf{x}_F, -t) + \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t-t') N_1(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x}, \\ &= -N_0(\mathbf{x}_R, \mathbf{x}_F, -t) - N_2(\mathbf{x}_R, \mathbf{x}_F, -t), \quad (20) \\ f_{2,1}(\mathbf{x}_F, \mathbf{x}_R, t) &= G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + N_0(\mathbf{x}_R, \mathbf{x}_F, t) + N_1(\mathbf{x}_R, \mathbf{x}_F, t) + N_2(\mathbf{x}_R, \mathbf{x}_F, t). \quad (21) \end{aligned}$$

The first integration of $R$ in equation (18) is used to compute $f^+_1$ as shown in equation (19). The second integration (20) computes $f^-_1$. The output $f_2$, as introduced in equation (5), includes the results of all $R$ integration-convolutio

The next step for $k = 2$ results in the following updates:

$$M_2^+(\mathbf{x}_R, \mathbf{x}_F, -t) = \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t - t') f_{1,1}^-(\mathbf{x}, \mathbf{x}_F, -t') \mathrm{d}t' \mathrm{d}\mathbf{x} \quad (22)$$

$$= -\theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t - t') \{N_0(\mathbf{x}, \mathbf{x}_F, t) + N_2(\mathbf{x}, \mathbf{x}_F, t)\} \mathrm{d}t' \mathrm{d}\mathbf{x}$$

$$= N_1(\mathbf{x}_R, \mathbf{x}_F, -t) + N_3(\mathbf{x}_R, \mathbf{x}_F, -t),$$

$$f_{1,2}^+(\mathbf{x}_R, \mathbf{x}_F, t) = G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + M_2^+(\mathbf{x}_R, \mathbf{x}_F, t)$$

$$= G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + N_1(\mathbf{x}_R, \mathbf{x}_F, t) + N_3(\mathbf{x}_R, \mathbf{x}_F, t), \quad (23)$$

$$f_{1,2}^-(\mathbf{x}_R, \mathbf{x}_F, t) = f_{1,0}^-(\mathbf{x}_R, \mathbf{x}_F, t) + \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t - t') M_2^+(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x}$$

$$= -N_0(\mathbf{x}_R, \mathbf{x}_F, -t) + \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t - t') \{N_1(\mathbf{x}, \mathbf{x}_F, t) + N_3(\mathbf{x}, \mathbf{x}_F, t)\} \mathrm{d}t' \mathrm{d}\mathbf{x}$$

$$= -N_0(\mathbf{x}_R, \mathbf{x}_F, -t) - N_2(\mathbf{x}_R, \mathbf{x}_F, -t) - N_4(\mathbf{x}_R, \mathbf{x}_F, -t), \quad (24)$$

$$f_{2,2}(\mathbf{x}_F, \mathbf{x}_R, t) = G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + N_0(\mathbf{x}_R, \mathbf{x}_F, t) + N_1(\mathbf{x}_R, \mathbf{x}_F, t) + N_2(\mathbf{x}_R, \mathbf{x}_F, t) + N_3(\mathbf{x}_R, \mathbf{x}_F, t) + N_4(\mathbf{x}_R, \mathbf{x}_F, t). \quad (25)$$

From these updates it becomes clear that in equation (23) for $f_1^+$ $G_d$ and the odd $N_i$ terms are used and in equation (24) for $f_1^-$ the even $N_i$ terms are used. The function $f_2$ in equation (25) is build up from $G_d$ and the even and odd $N_i$.
In the implementation the $N_i$ terms are computed by

$$N_{-1}(\mathbf{x}_R, \mathbf{x}_F, -t) = G_d(\mathbf{x}, \mathbf{x}_F, -t'), \quad (26)$$

$$N_i(\mathbf{x}_R, \mathbf{x}_F, -t) = -\theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t - t') N_{i-1}(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x}, \quad (27)$$

and used to update the $f_1^+$, $f_1^-$ and $f_2$ functions. This makes the algorithm both h efficient. In summary, the updates for $M_m^+$, $N_i$ and the focusing functions for the $m$th iteration with $m \geq 1$ are simply:

$$M_m^+(\mathbf{x}_R, \mathbf{x}_F, t) = \sum_{l=0}^{m-1} N_{2l+1}(\mathbf{x}_R, \mathbf{x}_F, t), \quad (28)$$

$$f_{1,m}^+(\mathbf{x}_R, \mathbf{x}_F, t) = G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + \sum_{l=0}^{m-1} N_{2l+1}(\mathbf{x}_R, \mathbf{x}_F, t), \quad (29)$$

$$f_{1,m}^-(\mathbf{x}_R, \mathbf{x}_F, t) = -\sum_{l=0}^{m} N_{2l}(\mathbf{x}_R, \mathbf{x}_F, -t), \quad (30)$$

$$f_{2,m}(\mathbf{x}_F, \mathbf{x}_R, t) = G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + \sum_{l=0}^{2m} N_l(\mathbf{x}_R, \mathbf{x}_F, t). \quad (31)$$

In the provided program each computation of $N_i$ is a ... implementation is shown in Algorithm 1 and Algorithm 2. The ... in Figure ... $f_1^+$, $f_2$, and $N_i$ are done just before the iteration starts ... $N_i$ ... $f_1^-$. The even ... and $f_1^+$ respectively. The Green's function is computed by inserting $f_2$ ... (31) into equation (4)

$$G(\mathbf{x}_F, \mathbf{x}_R, t) = f_2(\mathbf{x}_F, \mathbf{x}_R, -t) + \int_{\partial\mathbb{D}_0} \int_{t'=-\infty}^{t} R(\mathbf{x}_R, \mathbf{x}, t - t') G_d(\mathbf{x}, \mathbf{x}_F, -t) \mathrm{d}t' \mathrm{d}\mathbf{x}$$
$$+ \sum_{l=0}^{2m} \int_{\partial\mathbb{D}_0} \int_{t'=-\infty}^{t} R(\mathbf{x}_R, \mathbf{x}, t - t') N_l(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x}. \quad (32)$$

In equation (32) the integration-convolution terms can be recognis ... $N_i$ terms. By storing the sum of these unmuted terms ... $p$ ... the Green's ... functions can be calculated as a summation of previously co

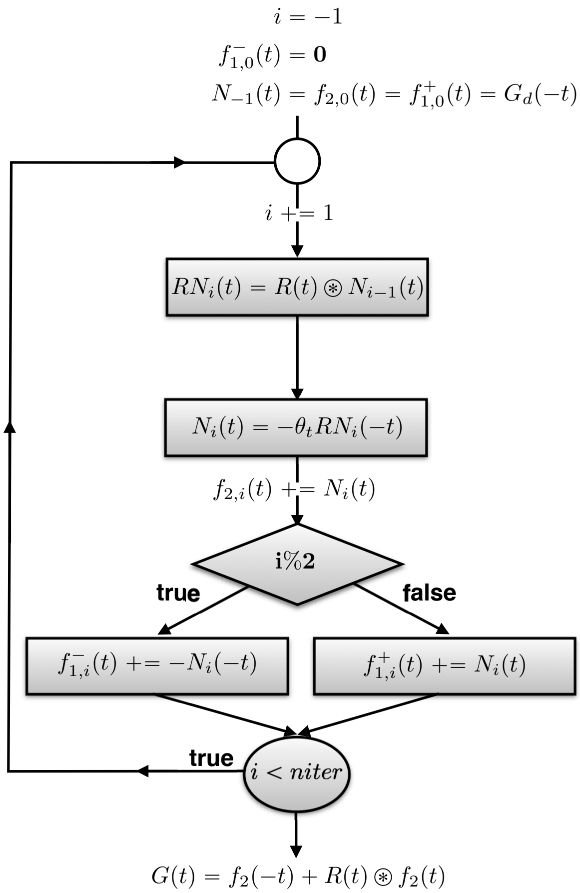



Figure 2: Flow chart of the Marchenko algorithm. In the notation t
a more compact notation. The symbol $\circledast$ represents the integration-convolutio

The program can compute the results of multiple focal points (Nfoc in Algorithm 1) in t
This is convenient for calculating the Marchenko results (e
of interest in one run. The computational advantage is that
only once to compute the results of multiple focal points.
are independent of each other. Hence, the code is OpenMP par
(Nfoc)
The function synthesis in Algorithm 1 computes the integration-convolution
term $N_i$ with $R$, in the frequency domain (Fourier $\mathcal{F}$). For the computation
of only one focal point, loading the required input data in
the computational work. The implementation has additional (1)
to compute the up- and downgoing Green's functions in equations (1) and (2)
computed $N_i$ fields to disk.

```
Main begin
  Reading SU-style input Data and Allocate arrays
  Initialisation
  Ni(t) = f2p(t) = f1plus(t) = G_d(-t)
  f1min(t) = pmin(t) = 0.0
  for iter ← 0 to niter do
    synthesis(Ref, Ni, iRN)
    Ni(t) = -iRN(-t)
    pmin(t) += iRN(t)
    applyMute(Ni, muteW)
    f2p(t) += Ni(t)
    if (iter % 2 == 0) then
      f1min(t) -= Ni(-t)
    else
      f1plus(t) += Ni(t)
    end
  end
  Green(t) = pmin(t) + f2p(-t)
end
synthesis(Ref, Ni, iRN)
begin
  iRN = 0
  ∀l,i : Fop(l,i,ω) = F_t{Ni(l,i,t)}
  for k ← 0 to nshots do
    #pragma omp parallel for
    for l ← 0 to Nfoc do
      for ω ← ω_min to ω_max do
        sum(ω) = 0
        for i ← 0 to nrecv do
          sum(ω) += Ref(i,k,ω) * Fop(l,i,ω)
        end
      end
      iRN(l,k,t) = F_t^-1{sum(ω)}
    end
  end
end
```

Algorithm 1: Marchenko algorithm as implemented in the provid

To use the Marchenko method with numerically modeled data i of the refection response are correct. This is certainly als of amplitude scaling is explained frst before discussing th In the summation of $N_1$ and $G_d$ to compute $f^+_{1,1}$ in equation (19), it is important that the a the measured refection data is accurate. A wrong amplitude of $R$ will result in a wrong $f^+_{1,1}$ and the scheme will not converge. This is illustrated with we introduce a wrong scaling factor $b$ in $R$ to compute $f^+_{1,1}$. Then the frst iterations will

$$-bN_0(\mathbf{x}_R, \mathbf{x}_F, -t) = \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} bR(\mathbf{x}_R, \mathbf{x}, t-t') G_d(\mathbf{x}, \mathbf{x}_F, -t') \mathrm{d}t' \mathrm{d}\mathbf{x},$$

$$-b^2N_1(\mathbf{x}_R, \mathbf{x}_F, -t) = \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} bR(\mathbf{x}_R, \mathbf{x}, t-t') bN_0(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x},$$

$$f^+_{1,1}(\mathbf{x}_R, \mathbf{x}_F, t) = G_d(\mathbf{x}_R, \mathbf{x}_F, -t) + b^2 N_1(\mathbf{x}_R, \mathbf{x}_F, t).$$

The update $f^+_{1,1}$ involves $a^2$ and for each next update $f^+_{1,m}$ the order of $a$ in $N_{2m+1}$ the update will grow with $a^{2m+2}$. A wrong amplitude of $G_d$ is not a problem as the Marchenko equa… the focusing function. An amplitude error can be factored ou…

$$-aN_0(\mathbf{x}_R, \mathbf{x}_F, -t) = \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t-t') aG_d(\mathbf{x}, \mathbf{x}_F, -t') \mathrm{d}t' \mathrm{d}\mathbf{x},$$

$$-aN_1(\mathbf{x}_R, \mathbf{x}_F, -t) = \theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t-t') aN_0(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x},$$

$$af^+_{1,1}(\mathbf{x}_R, \mathbf{x}_F, t) = aG_d(\mathbf{x}_R, \mathbf{x}_F, -t) + aN_1(\mathbf{x}_R, \mathbf{x}_F, t).$$

van der Neut et al. (2015) introduces an adaptive amplitude-correctio… amplitude error $R$ by solving the Marchenko equation in an explicit … errors can be adjusted by adaptive subtraction of the focus… better suited to apply (van der Neut et al., 2015; Staring et al., 2016). Brackenhoff (2016); Thomsen (2016) have developed estimation methodologie… factor $b$ to $R$. These methods compensate for an $R$, which is a main … first step to apply the Marchenko method on measured data. Br… use of the $G^-(\mathbf{x}, \mathbf{x}_F, t)$ that is identically zero below $\mathbf{x}_F$ of the deepest reflector. … The following will provide step by step directions how to co… amplitudes.

- The reflection data must be deconvolved (Mildner, 2017). The wavelet left (h… deconvolution is the reflection response of a zero-phase … frequencies $f_{min}$ and $f_{max}$. Since we are computing the reflection respo… lution and directly model the reflection response with a s… spectrum of amplitude 1.0:

$$s(t) = \int_{f_{min}}^{f_{max}} 1.0 \exp(-j2\pi ft)df \tag{33}$$

  The implemented flat wavelet spectrum has smooth transit… and from the maximum, frequency to avoid a very long wave… vided program makewave generate these waveforms and the provide… parameters used to calculate the source wavelet. Note, i… putation of the source wavelet in the frequency domain on… frequency $\Delta f$, when going from frequency to time with the Fou… wavelet used in the examples is shown in Figure 3. A shift of 0.3 seconds (t0=0.3 parameter in makewave) is added to the source wavelet to make it causal a… difference program. In the finite difference modeling of th… data is postponed with 0.3 seconds (rec_delay=0.3 parameter) to reset the peak of the wavelet back at the correct time.

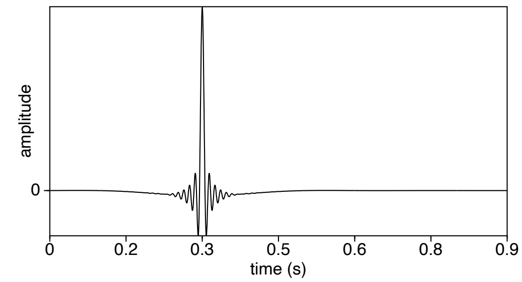
- In the finite-difference program $R(\mathbf{x}_R, \mathbf{x}_0, t)$ is modeled for an $F_z$ source of vertical force is… manual of the finite-difference modeling program about the… The receivers are placed at the same surface as the source…

- The amplitude scaling factor, in the finite-difference code for an $F_z$ source with signal $s(t)$, is defined in the update of $V_z$ as particle velocity

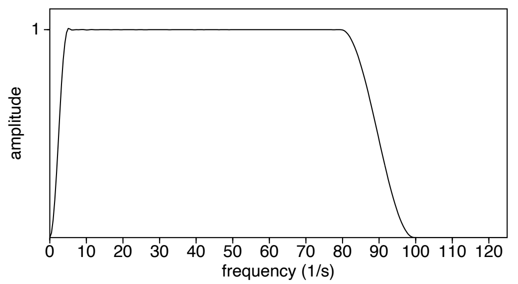
$$V_z(x, z, t + \Delta t) = V_z(x, z, t) - \frac{\Delta P(x, z, t)}{\rho \Delta z} + \frac{\Delta t}{\rho \Delta x^2} s(t). \tag{34}$$

  The discrete intervals $\Delta t, \Delta x = \Delta z$ are the steps in the finite-difference grid, $\rho$ the density at the injection point of the source. $\frac{\Delta P}{\Delta z}$ is the finite-differenc… of the first derivative $\partial_z$ of the pressure field.

- To compute $R$, from the Green's functions calculated by the fini… -2 is needed (e.g. Wapenaar et al. (2011)). This factor -2 is included in… program when it reads in $R$ the reflection response.




a) Source wavelet for modeling $R$.




b) Amplitude spectrum of source wavelet.

Figure 3: Source wavelet with a flat frequency spectrum between $f_{min}$ (= 5 Hz) and $f_{max}$ (= 80 Hz) used to model the reflection response.

- The time convolution in $R$ is implemented by a forward Fourier transfo quency domain, multiplication in the frequency domain, a main. In the numerical implementation of the convolution a factor $\Delta t$, from the convolution not with $\Delta x$ for the integration over $x$, must be included as well. Together wit factor $\frac{1}{N}$ of discrete Fourier transformations when going f time, with $N$ the number of time samples, the scale factor to co space integration in the frequency domain becomes:

$$\frac{\Delta x \Delta t}{N}.$$

### 3.2.1 Building up the Green's function

The Marchenko algorithm is illustrated with a 2-dimensiona 4. The numerical modeling is carried out with the finite-difference Thorbecke and Draganov, 2011) that is also included in the software package. The i the reflection response $R(x_R, x_S, t)$ is approximately a sinc-function with a fla as shown in Figure 3.
The full reflection matrix $R(x_R, x_S, t)$, for a fixed-spread geometry, can be const modeled shot (Figure 4c), since the model contains no lateral variatio geometry ranges from -2250 to 2250 meter with 5 meter distan The 5 meter distance is chosen to avoid spatial aliasing. We the time to compute the reflection response of a laterally varying medium is too for the desired reproducibility of the examples in this pap not make any assumption about the medium and can handle late demo directory of the Marchenko program contains also an exa (marchenko/demo/twoD).
The transmission response, recorded at the surface (Figure 4d). For a sou has been modeled with a zero-phase Ricker source wavelet s(t that the chosen source wavelet $G_d$ is zero-phase, otherwise the time rev algorithm would not work properly and the Marchenko scheme w to choose a source wavelet that decreases rapidly in time. T

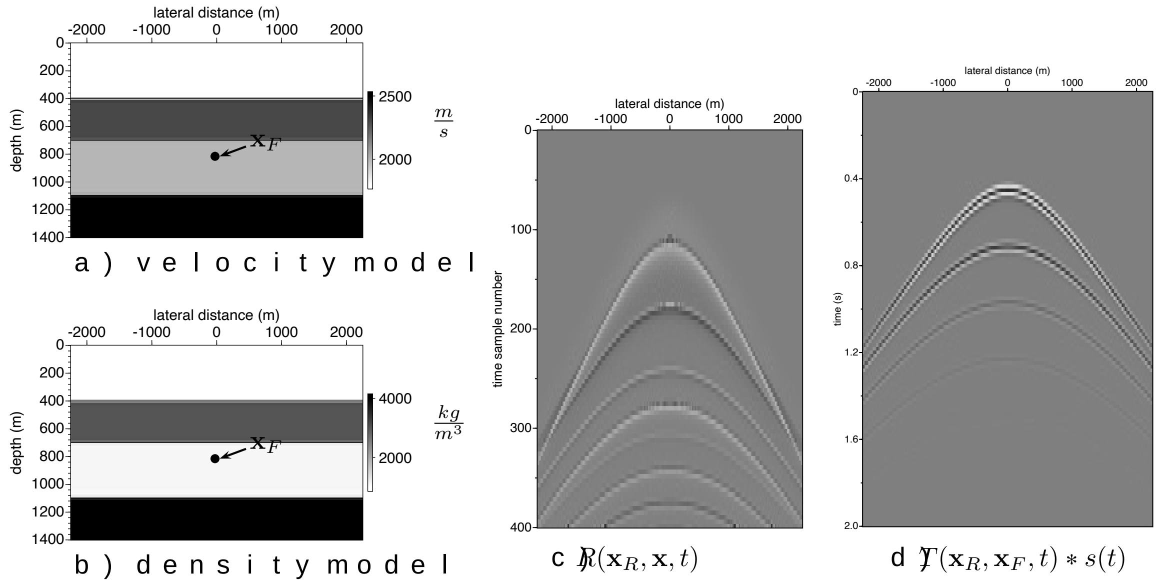



Figure 4: Four layer model with velocity (a) and density (b) contr... $\mathbf{x}(x=0,z=0)$ and receivers $\mathbf{x}_R(x=x_r,z=0)$ (c), and the transmission respons... $\mathbf{x}_F(x=0,z=900)$ (d). Note that the source wavelet ... $R$ (c) ... is given by ... (3) ... $T$ ... Figure ... (d) is modeled with a Ricker wavelet with a peak at 25 Hz.

between the direct arrival and the first (10) reflection ... in case of an overlap ... defined window $\theta_\epsilon$ in equation (11) ... through the overlapping events is not retrieved correctly.

The initialisation $f_{1,0}^-$ ... (equation 17) ... is computed ... Figure 5 ... Each ... record in $R(\mathbf{x}_R,\mathbf{x},t)$ is convolved with $G_d(\mathbf{x}_R,\mathbf{x}_F,-t)$, where $G_d(\mathbf{x}_R,\mathbf{x}_F,-t)$ shown in Figure 5 ... contains the time-reversal of the full transmission ... 4 ... By making use of ... $R(\mathbf{x}_R,\mathbf{x},t)=R(\mathbf{x}_R-\mathbf{x},0,t)$, the time-convolution result is integrated (s... $\mathbf{x}_R$) and results in one trace at position $\mathbf{x}$ in $N_0$.

In $-N_0(\mathbf{x},\mathbf{x}_F,-t)$ the dotted lines indicate the cut-of boundaries o... $\theta(\mathbf{x},\mathbf{x}_F,t)$. To suppress wrap-around events (from positive times ... window $\theta(\mathbf{x},\mathbf{x}_F,t)$, as introduced in (11), is symmetrized. Hence $\theta(\mathbf{x},\mathbf{x}_F,t)$ from here is zero for $t>t_d^\varepsilon$ and $t<-t_d^\varepsilon$, and unity for $-t_d^\varepsilon<t<t_d^\varepsilon$. For deep focal points o... also extend the time axis by padding zeros at the end of the a... wrap-around events in the time domain. In the Appendix the tr... in more detail.

The events before the top dotted line and the events after th... remaining events originate from the two refectors above the ... detailed explanation of the diferent events in the focusing ... Van der Neut et al. (2015) ... Staring et al. (2016) give a similar explanation in case free-surf... Marchenko method. This is the initial input for the next step to comput... $f_1^+$, given in equations (13) and (19).

The computation of $f_{1,1}^+$ involves the same time convolution and spatial ... illustrated in Figure 6. The result of the integration $-N_1(\mathbf{x},\mathbf{x}_F,-t)$ is, according to equation (19), time reversed, multiplied by $-1$ and added to $G_d(\mathbf{x},\mathbf{x}_F,-t)$ to get the first estimate $f_{1,1}^+$. Note, that the lower (causal) part of $\theta(\mathbf{x},\mathbf{x}_F,t)$ ... at the direct arrival time. This event at the direct arrival $t_d$ ... update of the Green's ... adjust the amplitude of the direct arrival ... Van der Neut et al. (2015b) ... The update of the direct arrival in the Green's function is expl...

Figure 7 shows the results of the first 4 iterations of the Marchen... the results of each convolution and integration ... $N_i$ ... weight ... $R$ ... from the focus ... (all with the same clipping factor) ... $N_i$ become smaller ...

next iteration.
The trace in the fifth column is a comparison between the ref[...] the computed Green's function (dotted black). In these trac[...] that some events are weakened by subsequent iterations: The [...] the reference Green's function.
To get a better understanding of the computation of the Gree[...] discussed in more detail. The initial $G_d$ (equation 26) of the metho[d] ... computed according to equation (32). This gives

$$f_{2,0}(\mathbf{x}_F,\mathbf{x}_R,t) = G_d(\mathbf{x}_R,\mathbf{x}_F,-t)$$

$$G_0(\mathbf{x}_R,\mathbf{x}_F,t) = G_d(\mathbf{x}_R,\mathbf{x}_F,-t) + \int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t} R(\mathbf{x}_R,\mathbf{x},t-t')G_d(\mathbf{x},\mathbf{x}_F,-t)\mathrm{d}t'\mathrm{d}\mathbf{x} + N_0(\mathbf{x}_R,\mathbf{x}_F,-t) \qquad (35)$$

Note that in (35) the result of the first integral is the unmuted $R$-convolution with $\theta_t$ [...]
The initial estimate of the Green's function is thus built u[p ...]

1. The direct arrival of the transmission response ($G_d(\mathbf{x}_R,\mathbf{x}_F,-t)$).
2. The integration $R$-convolution with $G_d$ of the (unmuted) top left panel of Figure 7.
3. A $\theta_t$ muted and multiplied by $-1$ version of the integral $R$-convolution of $G_d$: $N_0(\mathbf{x}_R,\mathbf{x}_F,-t) = -f^-_{1,0}(\mathbf{x}_R,\mathbf{x}_F,t)$, the second panel in the top row of Figure 7 multiplied by $-1$.

It is important to note that the result of the combination o[f ...] $f^-_{1,0}(t)$ (the events within the black-dotted lines) with the $R$-convolution of $G_d$ the unmu[ted ...]. This is the same as the inverse $\theta_t$ operator. To get to the first estimate of the Green's function $G_d(\mathbf{x}_R,\mathbf{x}_F,-t)$ is added to this result and gives the panel in Figure 7. In the next iteration we have

$$f_{2,1}(\mathbf{x}_F,\mathbf{x}_R,t) = G_d(\mathbf{x}_F,\mathbf{x}_R,-t) + N_0(\mathbf{x}_F,\mathbf{x}_R,t)$$

$$G_1(\mathbf{x}_F,\mathbf{x}_R,t) = G_0(\mathbf{x}_R,\mathbf{x}_F,t) + \int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t} R(\mathbf{x}_R,\mathbf{x},t-t')N_0(\mathbf{x},\mathbf{x}_F,t)\mathrm{d}t'\mathrm{d}\mathbf{x} + N_1(\mathbf{x}_R,\mathbf{x}_F,-t) \qquad (36)$$

Compared to the previous iteration two new terms are added:

1. The integration $R$-convolution with $N_0$ of the (unmuted $i=1$ in Figure 7 top panel fo[r ...]
2. The $\theta_t$ muted, time reversed, multiplied by $-1$ version of the integral $R$-convolution with $N_0$: $N_1(-t)$.

The combination of these two terms results in the subtractio[n ...] from the unmuted integral $R$-convolution of $N_0$.
Each next iteration follows this same pattern ($\theta_t$ and $G_d$) the events wi[thin ...] to update $f_1$ for the focusing function, the events ($\theta_t$, $G_d$) are used to update the Green's function. Application of $\theta_t$ is separating the focusing function from the Green's function.
There is one important remark to make: $T_d^{inv}$ for the first $f_1^+$ is not updated, whereas the direct arrival of the Green's function $G$ is updated. In the first iterati[on ...] Figure 7) the direct arrival in $G_0$ is the $G_d$. In the second iteration Figure 7 the amplitude of the direct arrival is $-N_1(-t)$ corrected by the e[...] the black-dotted line of the mute window $-N_1(-t)$ [...] direct arrival and decreases the Van der Neut et al. (2015) [...] trace $G_0$ the amplitude of the direct arrival (dotted line) is mu[ch ...] In $G_1$ the amplitudes of the direct arrival between reference a[nd ...] much closer. We do not expect that the scheme with $T_d^{inv} \approx G_d$ will arrive at the correct amplitudes; in order to achieve accurate amplit[udes ...] had to be used. $G_d$ and the correct [...] is an offset dependent scaling factor b[...]

computed Green's function. Thorbecke et al. (2013) showed that this estimate of the d
not have to be precise and can be based on a macro model. The r
of the computed Green's function is correct and shown in the
in Figure 7.

The iterative corrections of the amplitude of the Green's fu
mission losses. The result is that at $t = t_d$ the upgoing field that is r
to the local reflection coefficient at $z_f$ (Slob et al., 2014). In the next section we wi
good this correction is when the $f_2$ focusing function is emitted into the medi
Broggini (2014) also the energy before the direct arrival in the
convergence of the scheme. In the provided Marchenko progra
give the user the freedom to choose the number of iterations c
term ($\sqrt{\sum_{x,t} N_i^2(x,t)}$) is computed and printed for each iteration and ca
In each next iteration this energy should become smaller. Th
in Figure 8.

A comparison with the reference Green's function and the Ma
8 iterations is shown in Figure 9. The difference with the reference Green's f
middle part of the $x = 0$ pic. A small amplitude mismatch increases s
offset. Closer to the edge of the acquisition ($\pm 2250$ m) the difference with the ref
larger, because the full Fresnel zone is not included in the
present at earlier times, are also not captured by the limit ... $\sim 1.5$ s
the presence of higher wavenumbers becomes smaller, and th
decreases. To suppress artefacts from limited acquisition
the initial focusing operator and/or the reflection respons
effects on suppressing these artefacts. Depending on the sp
the finite aperture effect could slightly be attenuated. In
the non-tapered part adjacent to the tapered region and fin
usually smaller, amplitude mismatch is caused by the use of
transmission losses in $G_d$ (equation 10) instead of the inverse.

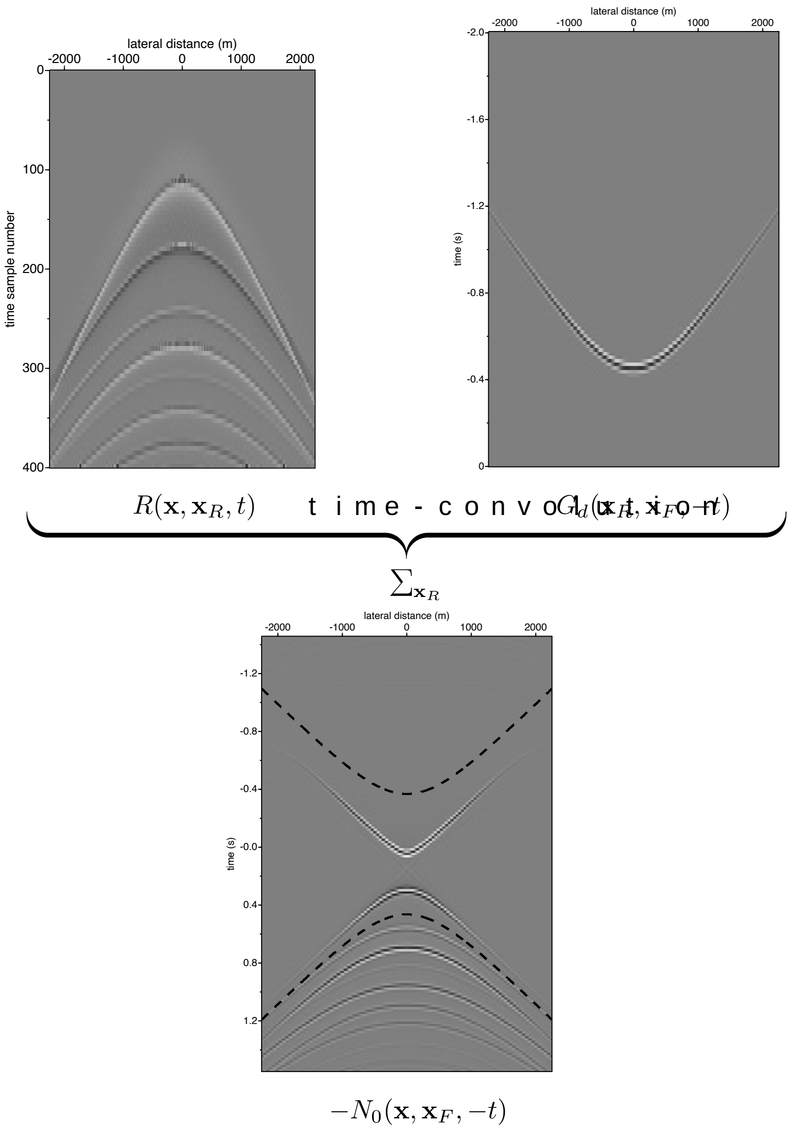



Figure 5: Initialisation step to compute $f^-_{1,0}(\mathbf{x}, \mathbf{x}_F, t) = -N_0(\mathbf{x}, \mathbf{x}_F, -t)$. After applying the time window $\theta(\mathbf{x}, \mathbf{x}_F, t) = \theta_t$ only events between the dotted lines remain in $N_0$. The mute for $t < 0$ is applied to mute the wrap-around events of the temporal convolution, $t < 0$ is only a practical solution and not needed from the theory. Note the panels; $R(t)$ is positive, $G_d(-t)$ is negative and $N_0(-t)$ positive for

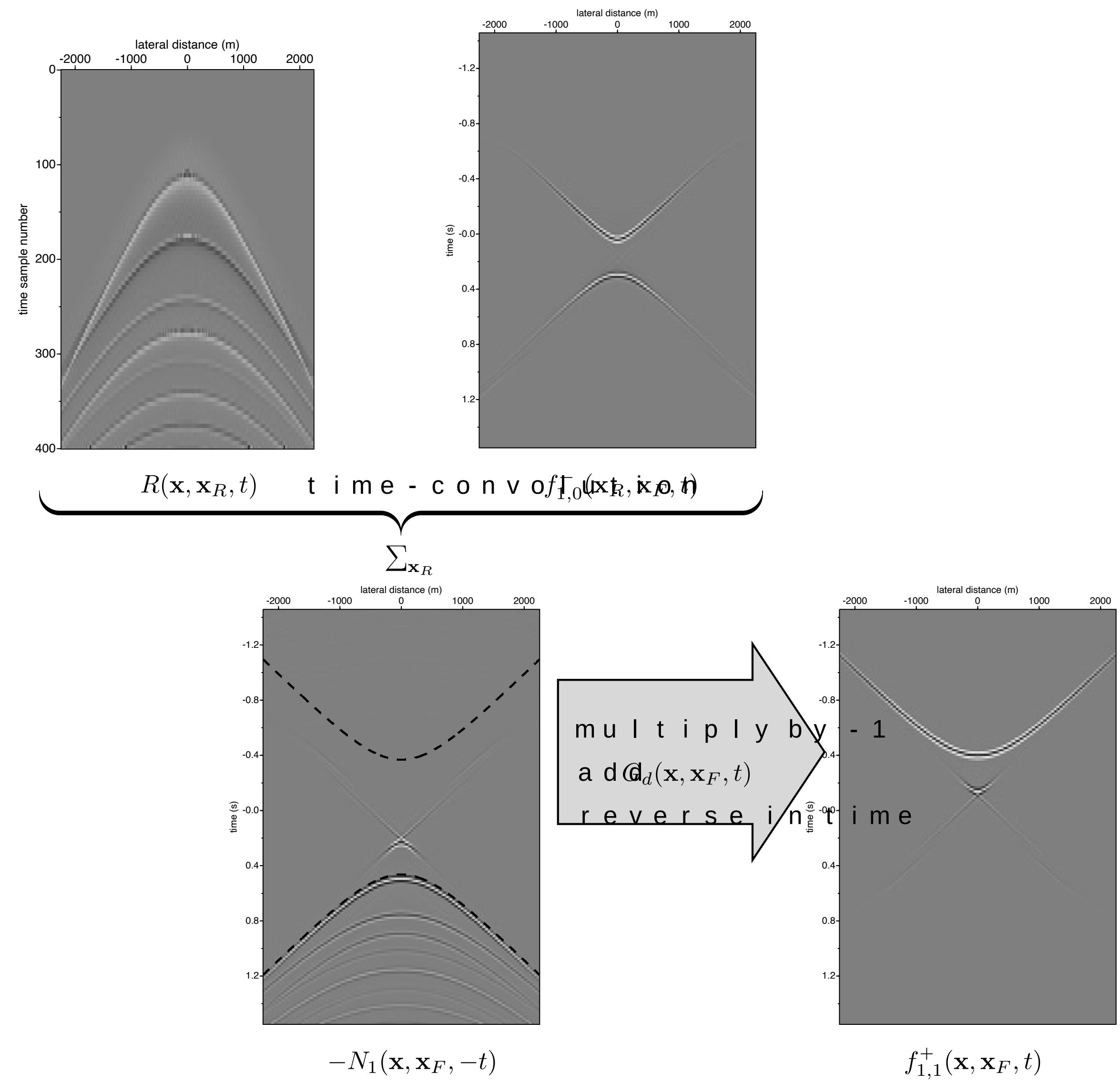



Figure 6: First iteration to compute $f^+_{1,1}(\mathbf{x},\mathbf{x}_F,t)$ from $f^+_{1,0}(\mathbf{x}_R,\mathbf{x}_F,t)$. In the summation of $G_d$ with $N_1$ it is important that the amplitudes of $R$ are correct.

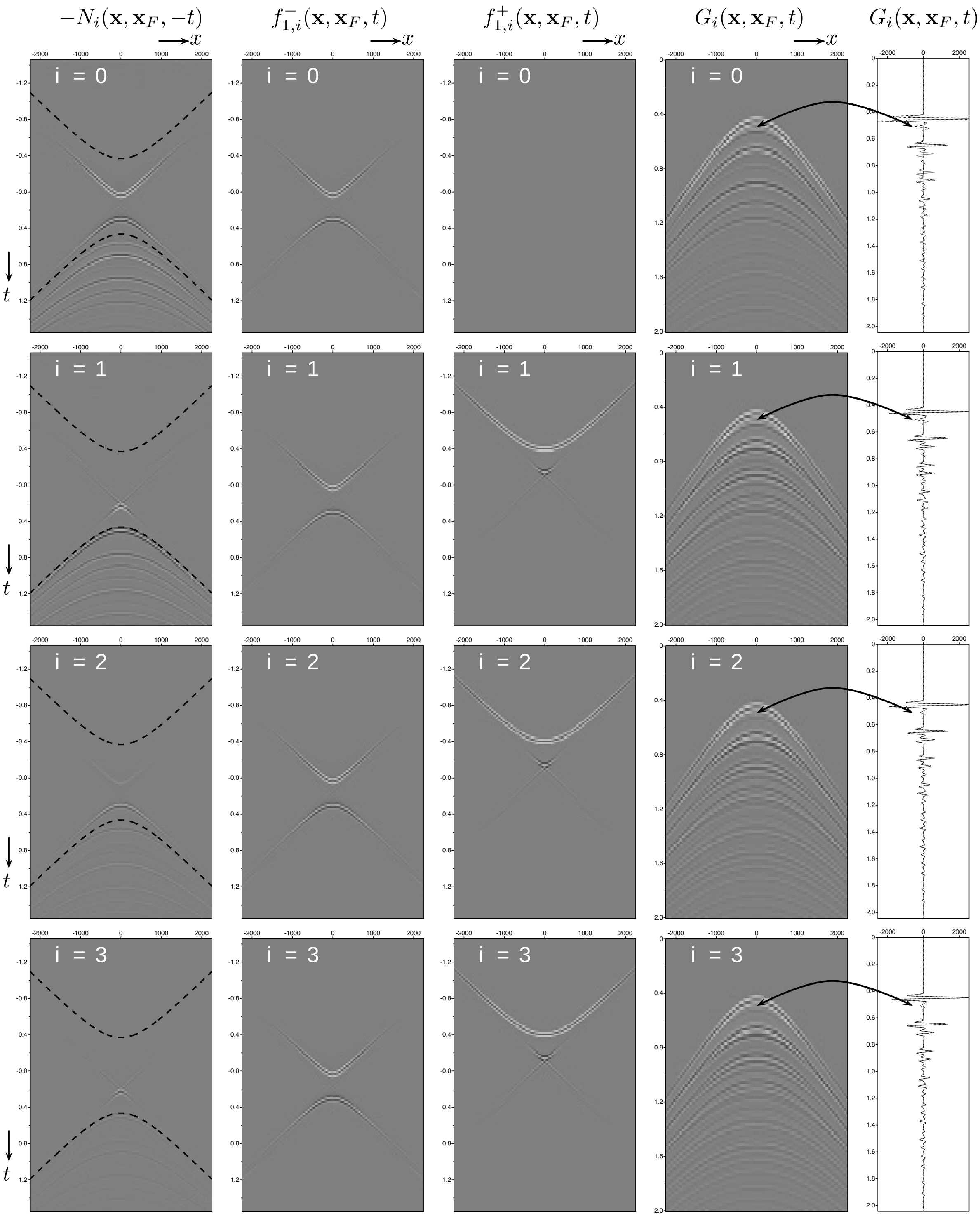



Figure 7: Four successive iterations of the Marchenko method. Th[...]
not belong to the Green's function, and [...] $f^-_{1,i}$ (the 2nd column) [...] ch[...]
from $i=1$ to $i=2$, while $f^+_{1,i}$ (the 3rd column) changes from $i=0$ to $i=1$, and from $i=2$ to $i=3$.
The clip level of $N_i$ and $G_i$ is the same for all panels. Labels of the horiz[...]
same for all panels, and are shown for the top and left panels[...]

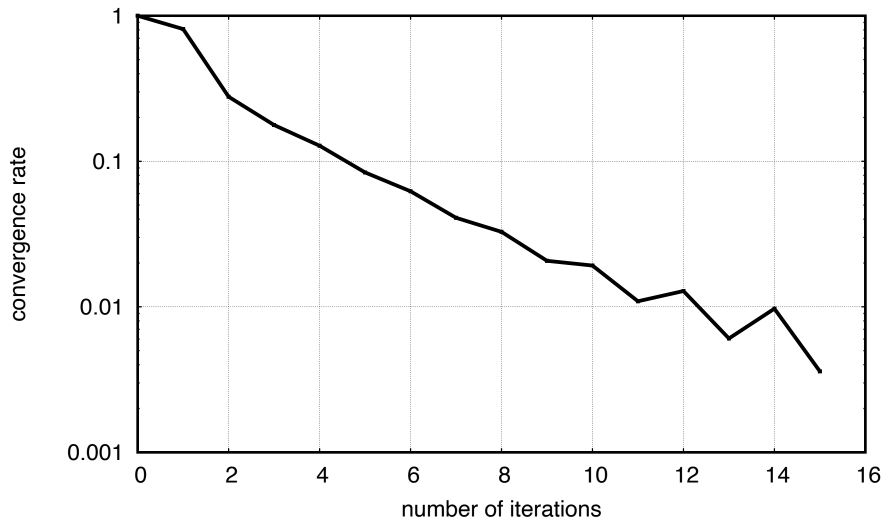



Figure 8: Logarithmic convergence rate of the marchenko/example for 16 iteration
bumps at the end of the curve are caused by limited aperture
magnitude smaller than the main events.

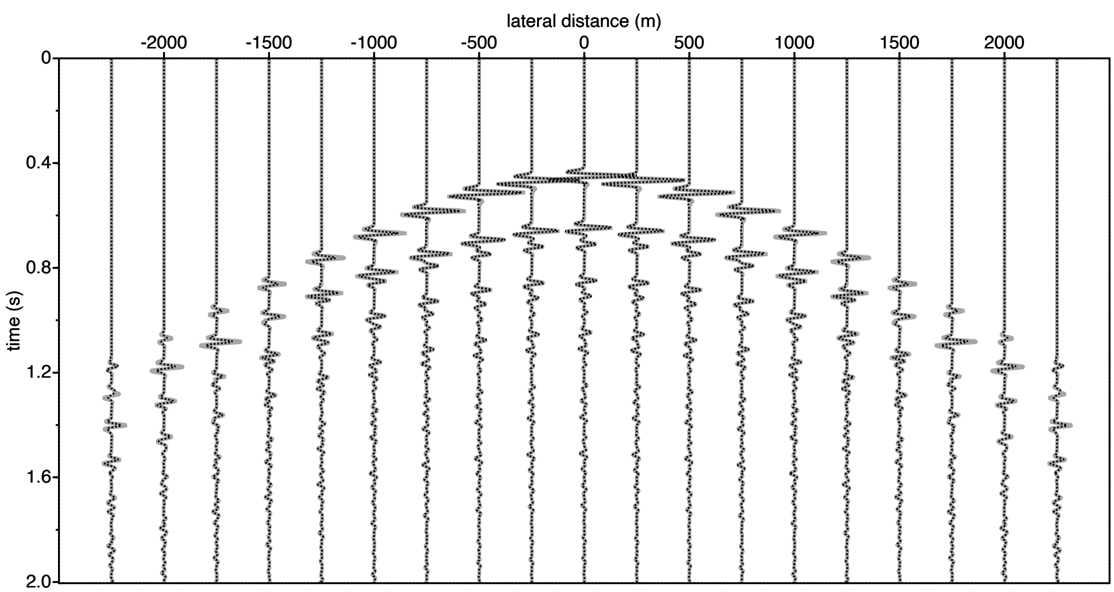



Figure 9: Comparison of the Marchenko computed Green's function
Green's function: solid-gray trace in the background is the
function computed with the Marchenko method.

### 3.2.2 Propagating focusing function

One of the properties of the focusing function $f_2(\mathbf{x}_F, \mathbf{x}, t)$ from equation (5) is that it will focu
$t=0$ at the focal point $\mathbf{x}_F$. This property can be demonstrated by emitting $f_2(\mathbf{x}_F, \mathbf{x}, t)$ from $\partial \mathbb{D}_0$ in
the medium and show that it has a focus at $\mathbf{x}_F$ at $t=0$ (Slob et al., 2014; Wapenaar
et al., 2017). If the transmission losses of $f_2$ have been correctly taken into
all internal multiples will be canceled at $t=0$. The left column of Figure 10 shows
snapshots at times $-0.30, -0.15, -0.03, -0.02$, and $0$. The snapshot at $t=0$ indeed shows only a fo
at the focal point. In the snapshots at $t=-0.03$ and $t=-0.02$ it is observed that events
internal multiples have opposite amplitude and travel towa $t=0$
The second column of Figure 10 shows snapshots at positive times, after th
$t=0$. After $t=0$ the focussed and dimmed events separate again and c

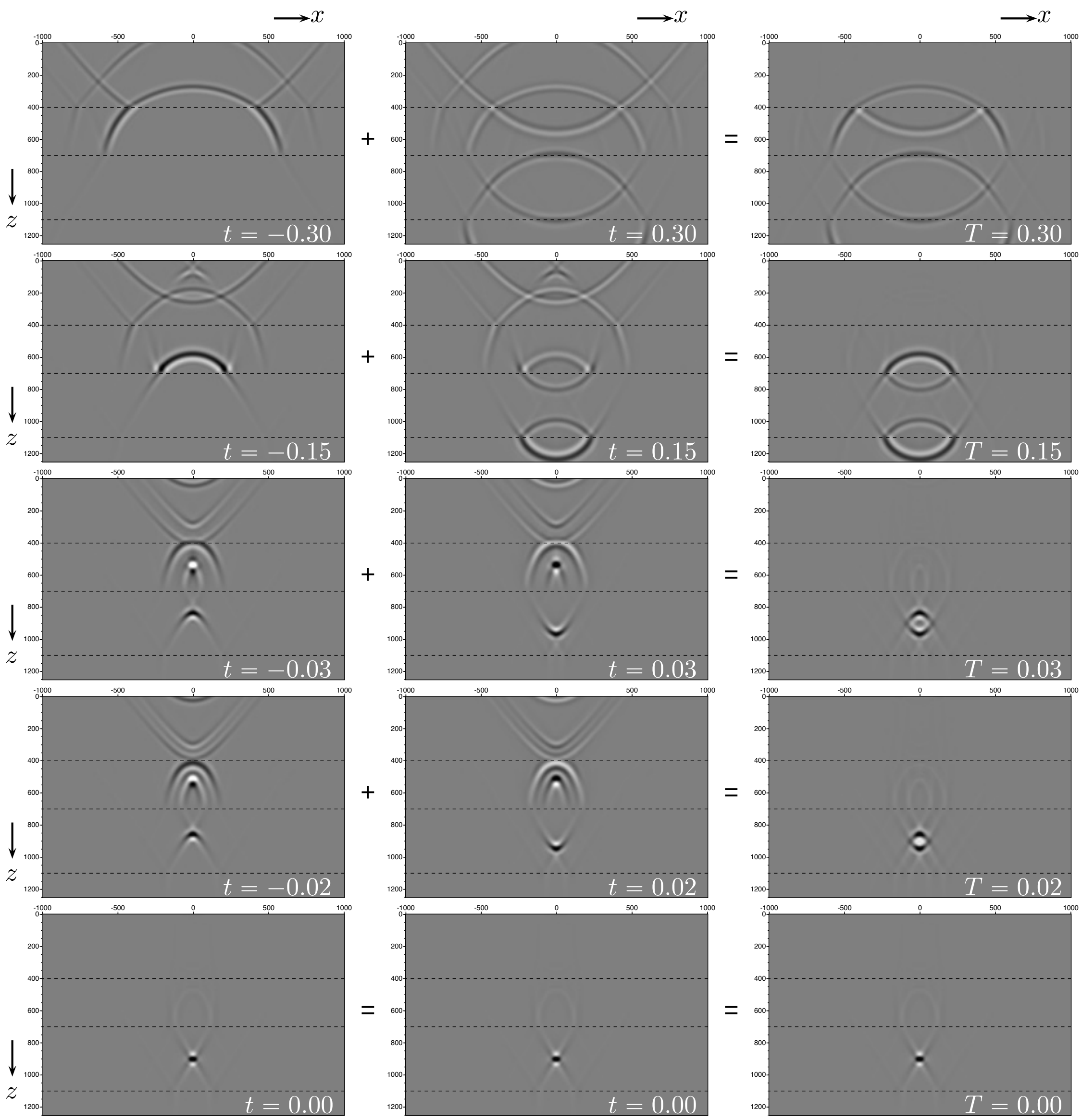



Figure 10: Snapshots of propagating $f_2$ through the actual medium. The
shows snapshots at a-causal times, the middle column snapsh
shows the addition of the acausal snapshots at negative tim
at positive times ($T$). Labels of the horizontal and vertical axes ar
shown for the top and left panels.

Adding the snapshots at negative times to the correspondin
snapshots of the homogeneous Green's function (Wapenaar, 2014) with virtual source $\mathbf{x}_A$.
The third column of Figure 10 shows these combined snapshots, where the s
negative times are summed, and represent the causal part of
snapshots can be interpreted as the response of a virtual sou
$\mathbf{x}_F$.

## marchenko

The self-doc of the program marchenko is shown by typing marchenko on the command line without any
You will then see the following list of parameters:
If you are not considering special cases, the default values
parameters have to be changed from their default values to g
The provided marchenko source-code package contains two ma

- fmute: picks the first arrival time from a transmission respo
- marchenko: solves for the focusing functions in the Marchen functions

The fmute program tracks the first arrival from a transmission res
Its main use is to separate the direct arrival ($G_d$) from the multiple coda ($M^+$). In the examples provided the transmission response
modeling and the direct arrival needs to be separated from
components $G_d(t)$ from $T(t)$ in Figure 4. The program fmute is not needed if a method is u
eikonal solver) that computes the direct arrival $G_d$ of the fmute program direc
is the input file_in for the marchenko program. The different parameters of fmute are shown in the
self-documentation of the program:

```
fmute - mute in time domain file_shot along curve of maximum amplitude i

fmute file_shot= {file_mute=} [optional parameters]

Required parameters:

  file_mute= ................ input file with event that defines the mut
  file_shot= ................ input data that is muted

Optional parameters:

  file_out= ................ output file
  above=0 ................. mute after(0), before(1) or around(2) the
  ........................ options 4 is the inverse of 0 and -1 the inv
  shift=0 ................. number of points above(positive) / below(n
  check=0 ................. plots muting window on top of file_mute: ou
  scale=0 ................. scale data by dividing through maximum
  hw=15 ................... number of time samples to look up and down
  smooth=0 ................ number of points to smooth mute with cosine
  verbose=0 ............... silent option; >0 display info
```

If file_mute is not provided, file_shot will be used instead to pick the first arr
The above option is explained in Figure 11 and separates in different ways the dir
from the coda. The above=4 and above=-1 options have also a truncation point a
time-axis, with the $t_d$, to mute the wrap-around events introduced b
discrete Fourier transform. Note that the lower end of the t
The above=2 option defines a pass gate around $t_d$ and is convenient to select the
the transmission response in case the first arrival also cont
To find the first arrival time in file_mute a simple tracking algorithm is implem
position equal to the source position the algorithm searche
assumed that this is the first arrival time at the source posi
true and it is therefore always good to enable check=1 and check the output
if the program has tracked the correct direct arrival time.

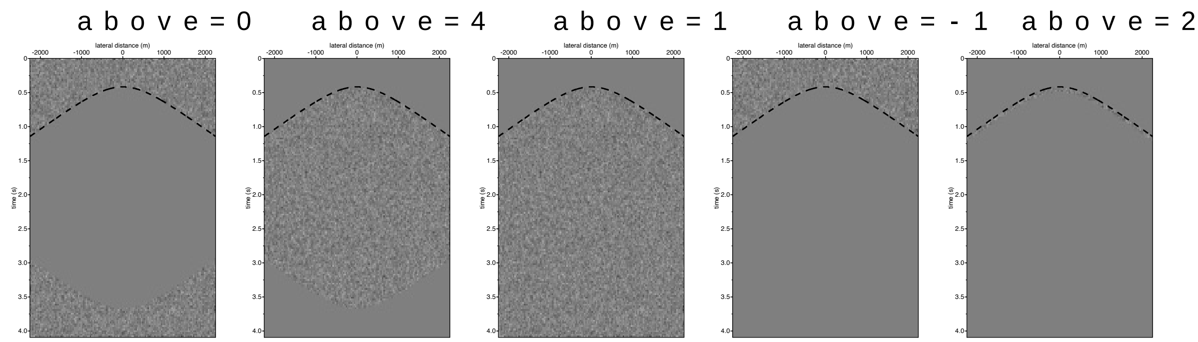



Figure 11: The different options of the parameter above in the marchenko program, illustra
with a shot panel consisting of noise.

maximum ($j_{max}$) in the source trace $i$ the algorithm looks in the neighbourhood ($i \pm 1$) for the maximum
It only searches for this maximum in a restricted time windo
trace is searched in the time window $j_{max} - n < t_{i-1} < j_{max} + n$, where $n$ is a number of sample
given as input parameter. If there are head-waves present
direct arrival, so it is good practice to choose a small $n$ (4-8 samples).

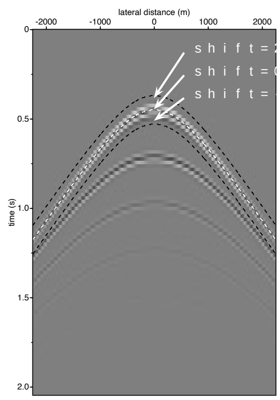



Figure 12: The shift parameter in the marchenko program.

The shift option represents $t_d^{\epsilon}$ and is needed to include the width of the
window. Figure 12 shows the effect of setting a negative or positive s
of the wavelet. With shift=-1 or a positive shift will mute the dire
will preserve the direct arrival.
The parameter smooth defines a transition zone (in samples) going fro
Using a few time-samples (3-5) for the smooth transition zon
direction of the taper, going from 1 to 0, is away from $\pm t_d$.
The marchenko program has the following parameters and options:

```
MARCHENKO - Iterative Green's function and focusing functions retrieva

marchenko file_tinv= file_shot= [optional parameters]

Required parameters:

  file_tinv= ............... direct arrival from focal point: G_d
  file_shot= ............... Reflection response: R

Optional parameters:

 INTEGRATION
  tap=0 .................... lateral taper focusing(1), shot(2) or both
```

```
  ntap=0 ................... number of taper points at boundaries
  fmin=0 ................... minimum frequency in the Fourier transform
  fmax=70 .................. maximum frequency in the Fourier transform
 MARCHENKO ITERATIONS
  niter=10 ................. number of iterations
 MUTE-WINDOW
  above=0 .................. mute above(1), around(0) or below(-1) the
  shift=12 ................. number of points above(positive) / below(n
  hw=8 ..................... window in time samples to look for maximum
  smooth=5 ................. number of points to smooth mute with cosine
  plane_wave=0 ............. enable plane-wave illumination function
  src_angle=0 .............. angle of plane source array
  src_velo=1500 ............ velocity to use in src_angle definition
 REFLECTION RESPONSE CORRECTION
  tsq=0.0 .................. scale factor n for t^n for true amplitude r
  Q=0.0 .................... Q correction factor
  f0=0.0 ................... ... for Q correction factor
  scale=2 .................. scale factor of R for summation of Ni with G
  pad=0 .................... amount of samples to pad the reflection ser
  reci=0 ................... 1; add receivers as shots 2; only use recei
  countmin=0 ............... 0.3*nxrcv; minumum number of reciprocal tr
 OUTPUT DEFINITION
  file_green= .............. output file with full Green function(s)
  file_gplus= .............. output file with G+
  file_gmin= ............... output file with G-
  file_f1plus= ............. output file with f1+
  file_f1min= .............. output file with f1-
  file_f2= ................. output file with f2 (=p+)
  file_pplus= .............. output file with p+
  file_pmin= ............... output file with p-
  file_iter= ............... output file with -Ni(-t) for each iteratio
  rotate=1 ................. 1: t=0 at nt/2 (middle) 0: t=0 at sample 0 f
  verbose=0 ................ silent option; >0 displays info
```

The number of iterations required for convergence depends o of events in the model; a complex model will need more iterat between 8 and 20. An automatic stopping criterion could be b $N_i$. This stopping criterion is not implemented to give the u iterations.

To suppress artefacts from a limited acquisition aperture, focusing operator (tap=1) and/or the reflection response (tap=2). In our experience these t limited efects on suppressing the fnite-acquisition relate

The mute-window parameters have the same meaning as in the fmute program.

The temporal convolution of events at positive times in the $R$ ... be shifted forward in time. Events at negative times in $R$ will be shifted backward in time. Inst Marchenko method it is important that these backward shifte focal points some events can be shifted to negative times. in the frequency domain, we make use of the periodic propert negative times wrap-around to the end of the discrete time a The reason to symmetrise $\theta_t$ is to suppress the unwanted time wrap-ar time-wrap-around efects can also be avoided by padding zeros to $R$, making the traces 2*nt long, where the last nt samples are zeros. The pad parameter adds zeros to the traces of $R$. Adding extra time samples will lead to longer comput use a symmetrised time window to suppress the unwanted efect

The scale parameter can be useful when the modeled data does not represents the previously mentioned scaling factor $\ell$ of the refection response ally, when the file_green is defined, output results of computed Green

Defining file_iter writes for each iteration $-N_i(-t)$ (for running the Algorithm 1) before applying the mute window. By setting verbose=2 the energy of the update term is printed out for each iteration and can be used

The code to reproduce all fgures in this paper can be found in the marchenko/demo/oneD. The README fle in that directory explains in detail how to run the s varying) model can be found in marchenko/demo/twoD. This example usually tak hours to complete the refection data modeling on a personal

In addition to the Marchenko programs, the package also cont
ence modeling code, that is used to model all the examples (Thorbecke
and Draganov, 2011). The directory utils contains programs to calculate a grid
source wavelet (makewave) as well as programs for basic processing step
In the next subsections all the parameters will be describe
how to use them.

Verbose The parameter verbose prints messages and produces additional f
the programs. Table 1 shows the kind of messages and the extra files pri
verbose. Those messages and files contain extra information for
different setting of the verbose parameter are:

| setting | messages printed to stdout |
|---|---|
| 0 | no messages only warnings |
| 1 | data information, source, receiver, paramete |
| 2 | + iteration convergence |
| 3 | + mute-window, OpenMP info |
| 4 | + shot gather processing |
| $>4$ | |

Table 1: The files and messages produced by different values o

The demo directory contains scripts which demonstrate the di
In the subsections below most demo script are explained and

In this section, we describe the implementation of both Mar... Transmission-compensated Marchenko Multiple Elimination ... eliminate internal multiple reflections without the need fo... Only a reflection response without source wavelet and free-s... as input. The paper is organised as follows: In the theory se... MME and T-MME schemes. In the implementation section the pr... step and this section provides a user's first step with the MM... of the algorithm is illustrated with a simple three-reflecto... This simple model is chosen to keep the number of events limi... can be followed more easily. The method is not limited to sim... to complicated 3D media as shown in Zhang et al. (2019b) for example

In this section we give a brief overview of the theory of both ... surface is located at $\partial\mathbb{D}_0$. The reflection response $R(\mathbf{x}_0', \mathbf{x}_0'', t)$ is measured with the source and receiver positions $\mathbf{x}_0'$ and $\mathbf{x}_0''$, which is free from free-surface rela... and source wavelet. The time is denoted $t$.

## Marchenko multiple elimination

As presented in Zhang et al. (2019), we give the equations of the Marchenko Mul... scheme as

$$R_t(\mathbf{x}_0', \mathbf{x}_0'', t=t_2) = R(\mathbf{x}_0', \mathbf{x}_0'', t=t_2) + \sum_{m=1}^{\infty} M_{2m}(\mathbf{x}_0', \mathbf{x}_0'', t=t_2, t_2), \tag{37}$$

with

$$\begin{aligned} M_{2m}(\mathbf{x}_0', \mathbf{x}_0'', t, t_2) = &\int_{t'=0}^{+\infty}\int_{\partial\mathbb{D}_0} R(\mathbf{x}_0''', \mathbf{x}_0', t') H(t-t'-\varepsilon)\mathrm{d}\mathbf{x}_0'''\mathrm{d}t' \times \\ &\int_{t''=0}^{+\infty}\int_{\partial\mathbb{D}_0} R(\mathbf{x}_0, \mathbf{x}_0''', t'') H(t'-t+t_2-t''-\varepsilon) \times \\ &\quad M_{2(m-1)}(\mathbf{x}_0, \mathbf{x}_0'', t-t'+t'', t_2)\mathrm{d}\mathbf{x}_0\mathrm{d}t'', \end{aligned} \tag{38}$$

and initialization

$$M_0(\mathbf{x}_0', \mathbf{x}_0'', t, t_2) = -(H(t+t_2-\varepsilon) - H(t+\varepsilon))R(\mathbf{x}_0', \mathbf{x}_0'', -t), \tag{39}$$

where $R_t$ denotes the retrieved dataset without internal multiples ... $t_2$ ... $H$ ... the Heaviside function, which is used to apply the time ... $(\varepsilon, t_2-\varepsilon)$ ... in the equations. $\varepsilon$ The constant is a small positive value which can b... time-duration in practice. The $M_0$ is the initialization ... at shot position $\mathbf{x}_0''$ for times $(\varepsilon, t_2-\varepsilon)$. The left-hand side of equation 37, $R_t$, is the same as the shot record, but without internal multiples Zhang and Slob (2019a). ... $t_2$ constant and independent of the source and receiver positions in the ... is carried out over the receiver coordinate for both integra... The second term in the right-hand side of equation 37 predicts all internal multiple r... Equation 39 indicates that the measured reflection response is the ... in equation 37. To retrieve a dataset without internal multiples, this process must be repeated for all times $t_2$.

Equation 38 contains the terms that correct for the internal multi[ples] in $R(\mathbf{x}'_0,\mathbf{x}''_0,t)$. To better explain the right-hand side of equation 38, we divide the expression into two pa[rts]

$$M_{2m}(\mathbf{x}'_0,\mathbf{x}''_0,t,t_2)=\int_{t'=0}^{+\infty}\int_{\partial\mathbb{D}_0}R(\mathbf{x}'''_0,\mathbf{x}'_0,t')H(t-t'-\varepsilon)\times M_{2m-1}(\mathbf{x}'''_0,\mathbf{x}''_0,t-t',t_2)\mathrm{d}\mathbf{x}'''_0\mathrm{d}t', \tag{40}$$

$$M_{2m-1}(\mathbf{x}'''_0,\mathbf{x}''_0,t-t',t_2)=\int_{t''=0}^{+\infty}\int_{\partial\mathbb{D}_0}R(\mathbf{x}_0,\mathbf{x}'''_0,t'')H(t'-t+t_2-t''-\varepsilon)\times M_{2(m-1)}(\mathbf{x}_0,\mathbf{x}''_0,t-t'+t'',t_2)\mathrm{d}\mathbf{x}_0\mathrm{d}t''. \tag{41}$$

Equation 40 is a time domain convolution of $R$ with $M$ integrated over the spatial coordinate $\mathbf{x}'''_0$, which is the receiver position $\mathbf{x}'_0$. Equation 41 is a time domain correlation of $R$ with $M$ integrated over the spatial coordinate $\mathbf{x}_0$, which is the receiver position $\mathbf{x}'''_0$. The Heaviside function in equation 40 is to exclude negative times $t-t'<\varepsilon$ and to impose the causality of $M_{2m-1}$. In the equation 41 the function $M_{2(m-1)}$ contains non-zero values for $t_2-\varepsilon$. This means these should not contribute to the integral and the Heaviside function is used to impose the integration for $t''-t+t'>t_2-\varepsilon$. Note that equations 40 and 41 perform very simil[ar] operations (they differ by a sign change) and are implemented 38 this function is applied two times. The convolution terms a[re] function and the correlation terms are the odd numbered oper[ations]. To better explain the method and for illustration purposes, $M$ 40 is defined as the field:

$$k^-_{1,i}(\mathbf{x}'_0,\mathbf{x}''_0,t,t_2)=R(\mathbf{x}'_0,\mathbf{x}''_0,t,t_2)-\sum_{m=1}^{i}\int_{t'=0}^{+\infty}\int_{\partial\mathbb{D}_0}R(\mathbf{x}'''_0,\mathbf{x}'_0,t')H(t-t'-\varepsilon)\times M_{2m-1}(\mathbf{x}'''_0,\mathbf{x}''_0,t-t',t_2)\mathrm{d}\mathbf{x}'''_0\mathrm{d}t'. \tag{42}$$

We can evaluate equation 42 also for $t\geqslant t_2-\varepsilon$ and the equation can be further split as follows

$$k^-_{1,i}(\mathbf{x}'_0,\mathbf{x}''_0,t,t_2)=\begin{cases}v^-_{1,i}(\mathbf{x}'_0,\mathbf{x}''_0,t,t_2) & t<t_2-\varepsilon\\ u^-_{1,i}(\mathbf{x}'_0,\mathbf{x}''_0,t,t_2) & t\geqslant t_2-\varepsilon\end{cases}, \tag{43}$$

where $v_1^-$ and $u_1^-$ are similar to the projected Green's function and Marchenko scheme Van der Neut and Wapenaar (2016). The minus superscript in 42 and 43 refers to upgoing wavefields at $\mathbf{x}'_0$. To solve the equivalent of equation 37 in 39, $v_1^-$ computation is not needed. The $M$ terms are used to solve the Marchenko e[quations] ... compute at $t_2$ with $u_1^-$. The solution in the MME scheme is $u_1^-$ at $t=t_2$, collected fro[m] ... Similarly for the summation of the even terms in equation 41, a field is defined

$$k^+_{1,i}(\mathbf{x}'_0,\mathbf{x}''_0,t,t_2)=\int_{t'=0}^{+\infty}\int_{\partial\mathbb{D}_0}R(\mathbf{x}'_0,\mathbf{x}_0,-t')v^-_{1,i}(\mathbf{x}_0,\mathbf{x}''_0,t-t',t_2)\mathrm{d}t'\mathrm{d}\mathbf{x}_0. \tag{44}$$

$$v^+_{1,i}(\mathbf{x}'''_0,\mathbf{x}''_0,t,t_2)=\sum_{m=1}^{i}\int_{t''=0}^{+\infty}\int_{\partial\mathbb{D}_0}R(\mathbf{x}_0,\mathbf{x}'''_0,t'')H(t'-t+t_2-t''-\varepsilon)\times M_{2(m-1)}(\mathbf{x}_0,\mathbf{x}''_0,t+t'',t_2)\mathrm{d}\mathbf{x}_0\mathrm{d}t'', \tag{45}$$

Equation 44 (or 45) can be further split in the time domain as follows;

$$k^+_{1,i}(\mathbf{x}'''_0,\mathbf{x}''_0,t,t_2)=\begin{cases}v^+_{1,i}(\mathbf{x}'''_0,\mathbf{x}''_0,t,t_2) & t<t_2-\varepsilon\\ u^+_{1,i}(\mathbf{x}'''_0,\mathbf{x}''_0,t,t_2) & t\geqslant t_2-\varepsilon\end{cases}, \tag{46}$$

where $v_1^+$ is similar to the projected focusing function in the reg[ular] ... Van der Neut and Wapenaar (2016). In $v_1^+$ the multiple annihilator is created and t[he] ... numerical examples 44 is only evaluated for $\varepsilon < t < t_2-\varepsilon$. The plus superscript in 44 refers to downgoing wavefields. To solve the MME, $v_1^+$ is not needed and is only defined for illustration purposes to explain the mechanism o[f]

Time $t_2$ is the instant two-way travel-time where the solution o
The primary reflection $\bar{v}_1$ is collected for every time $t_2$. This is a computational e
way, since only one sample is collected in the output. Never
implemented without any human interaction or model informa
one sample around $t_2$, and take bigger time steps, but the num
around $t_2$ must take into consideration the frequency bandwidth o
by examples in the detailed discussion of the implementatio
algorithm.
In this MME scheme the primary is collected from the origina
removes all overlapping internal multiples from earlier re
the physical reflection amplitude as present in the data.

## Transmission compensated Marchenko multiple elimination

Both internal multiple reflections and transmission losses
the Transmission-compensated Marchenko multiple eliminat Zhang et al. (2019).
The equation is given by

$$R_r(\mathbf{x}_0', \mathbf{x}_0'', t = t_2) = R(\mathbf{x}_0', \mathbf{x}_0'', t = t_2) + \sum_{m=1}^{\infty} \bar{M}_{2m}(\mathbf{x}_0', \mathbf{x}_0'', t = t_2, t_2), \tag{47}$$

with

$$\begin{aligned}\bar{M}_{2m}(\mathbf{x}_0', \mathbf{x}_0'', t, t_2) = &\int_{t'=0}^{+\infty} \int_{\partial \mathbb{D}_0} R(\mathbf{x}_0''', \mathbf{x}_0', t') H(t - t' + \varepsilon) \mathrm{d}\mathbf{x}_0''' \mathrm{d}t' \times \\ &\int_{t''=0}^{+\infty} \int_{\partial \mathbb{D}_0} R(\mathbf{x}_0, \mathbf{x}_0''', t'') H(t' - t + t_2 - t'' + \varepsilon) \\ &\qquad \bar{M}_{2(m-1)}(\mathbf{x}_0, \mathbf{x}_0'', t - t' + t'', t_2) \mathrm{d}\mathbf{x}_0 \mathrm{d}t''\end{aligned} \tag{48}$$

and

$$\bar{M}_0(\mathbf{x}_0', \mathbf{x}_0'', t, t_2) = -(H(t + t_2 + \varepsilon) - H(t + \varepsilon)) R(\mathbf{x}_0', \mathbf{x}_0'', -t), \tag{49}$$

where $R_r$ denotes the retrieved dataset without internal multip
primary reflections. The truncation window in equation 48 is different from the window in equation 38, in
which guarantees that the second term in equation 47 predicts both internal multiple
reflections and transmission losses in equation 48. The primary reflection se
$\bar{M}_{2(m-1)}$ does not have a contribution for $t''-t+t' > t_2+\varepsilon$ or values of t in equation 41 ...
$t_2$ is now part of the integration and $\bar{M}_{2m}$. Still, as is given in equation 49, ...
measured reflection response is the only input to solve the T
The primary reflection is, different than $\bar{v}_1$, in the MME scheme, h
transmission compensation. The $\bar{k}_1$, $\bar{v}_1$ are no longer needed, $\bar{M}_{2m}$. The ... of a
scheme is applied for every time instant $t_2$ and has the same advantages and disadva
scheme. In the T-MME scheme the amplitude of the primary is au
because it is the only way to predict and attenuate internal
We come back to this remark in the explanation of Figure 13.
Both MME and T-MME schemes only need the measured reflection response $R$ as input. The reflect
response $R$ needs to be deconvolved for the source-wavelet and the
must be removed. The output of a surface-related multiple el Verschuur et al. (1992)
scheme can meet these requirements. Diffracted and refracte
schemes and a detailed analysis about limitations can Zhang et al. (2019).

The basic Marchenko algorithm (MME) is explained in Algorithm 2. The arrays in this algorithm
stored in C-order; the last (most right) addressed dimensi
dimensions of these arrays $[. . .]$, are the arguments of a function ca
regular $(. . )$ brackets. The data input of the algorithm is $R$, the measured

```
Main begin
  Read SU-style input parameters
  Initialization, reading of input parameters and allocat
  READ( R[N_shots, iω, N_recv] )
  DD[N_recv, it] = F^{-1}{R*[j, iω, N_recv]}
  for ii ← istart to iend do
    M_0[N_recv, it] = { 0                  0 < it < n_t − ii + n_ε
                      { −DD[N_recv, it]    n_t − ii + n_ε ⩽ it < n_t
    k^-_{1,0}[N_shots, it] = DD[N_recv, n_t − it]
    v^+_{1,i}[N_shots, it] = 0
    for i ← 0 to n_i do
      synthesis(R, M_i, RM_i)
      M_{i+1}[N_shots, it] = RM_i[N_shots, n_t − it]
      if (i % 2 == 0) then
        M_{i+1}[N_shots, it] = 0;   ii − n_ε < it < n_t
        v^+_{1,i+1}[N_shots, it] = v^+_{1,i}[N_shots, it] + M_{i+1}[N_shots, it]
      else
        k^-_{1,i+1}[N_shots, it] = k^-_{1,i}[N_shots, it] − M_{i+1}[N_shots, n_t − it]
        M_{i+1}[N_shots, it] = 0;   0 < it < n_t − ii + n_ε
      end
    end
    R_t[j, N_shots, ii] = k^-_{1,n_i}[N_shots, ii]
  end
end
```

Algorithm 2: Basic Marchenko algorithm, without transmission l mented in the provided source code. The integer $it$ is the time sample number, $ii$ iterates from $istart$ to $iend$, represents $t = it * \Delta t$. The number of recorded time samples is $n_t$, and the time position of the source signal is $\varepsilon = n_\varepsilon * \Delta t$ and time $ii$ represents $t_2 = ii * \Delta t$. The number of receivers $N_{recv}$ and number of shots $N_{shots}$. The loop represents the number of Marchenko iterations $n_i$. Note that the samples for negative time $n_t - it$ are stored in $-t$.

reflection data must be properly pre-processed (Brackenhoff, 2019). As explained processing must take care of the following:

- Elimination of free-surface multiples. Note that there is also a very similar Marchenko algorit multiples (Ravasi, 2017). discusses a redatuming algorithm (Singh et al., 2015) simil and requires a smooth model (Zhang and Slob, 2019), while multiples does not need any model information.
- Sufficient (i.e. alias free) sampling in the spatial recei Note, there are Marchenko-based methods that can fill in m under the assumption that the (Wapenaar and van IJsseldijk, 2020)
- Compensate for dissipation.
- Shot amplitude regularization.
- Deconvolution for source wavelet.

Following Algorithm 2, the pre-processed reflection data is read from di $\mathcal{F}\{...\}$) to the frequency domain ($\omega$) (and all shots and receivers are stored step in the algorithm and the only significant data read. One ($j$) where we want to suppress the internal multiples from, is se step. This shot record is transformed back to time ($R^*$), called $DD$ in the time reversal

first loop in the algorithm loops over the selected number of
internal multiples. Typically this represents all samples
the number of samples to the first reflection event in the sel
$ii$ the iterative Marchenko algorithm is executed. The larges
in Thorbecke et al. (2017) is that time-truncation along the first arriva
subsurface) is replaced by a constant time-truncation and t
needed anymore. The initial $M_0$ is formed from the shot record $j$ from which we
would like to attenuate $(DD)$; $M_0$ is a copy of the time (reversed sh
equation 39) and set to zero from the $n_t - ii + n_\varepsilon$ samples, where $n_t$ is the total number
of samples in the shot record. The extra $n_\varepsilon$ samples take into account the time duratio
to exclude a possible $ii$. The initial $k^-_{1,0}$ is a complete (no time-m
is carried out) copy of the shot record that still contains a
With these two initializations the iterations of the Marche
updated field $M_i$ is computed by $RM_i$. This integration process is ca
produces the output $RM_i$, and is explained in more detail below. Depend
$i$, being odd or even, different time muting of $RM_i$ and two components are use
an update $M_{i+1}$. For even iterations the times between $ii - n_\varepsilon$ and $n_t$ are set to zero and for
iterations the times between $0$ and $ii + n_\varepsilon$ are set to zero. Only in the odd iterations $k^-_{1,i}$ is updated with thi
the unmuted $M_{i+1}$. In this update of $k^-_{1,i}$ the internal multiples after $ii$ are removed/attenuated. This i
update represented in equation 37, where the update $M_{2m}$ is in fact one even and one odd
the implemented Marchenko algorithm, and $M_{2m-1}$ in equation 42 in the notatio
In the regular redatuming Marchenko algorithm (Thorbecke et al., 2017) the truncation windows
the first arrival time of a focal point in the subsurface. I
rithm, the focal point is projected on the surface and the t
constant time. The flat time window has the big advantage th
data-informed. Meles et al. (2020) demonstrate that in the application of
algorithm to dipping plane waves a time truncation consiste
Depending on the position of strong reflectors typically, 10
time samples $ii$ in the selected shot record. The presence of strong r
convergence slow at large times (20). In these cases, it helps to first-order r
are attenuated with even $v_1^-$ and that are removed again later whe
multiple is finally removed by a converged multiple attenuato
all multiples are removed and hence all earlier higher-orde
Once the iterations are finished the update of $k^-_1$ at sample $ii$ is stored in output
sample $ii$ in the multiple eliminated $R_t$, the final output of the program that re
shot record with attenuated internal multiples. It is a com
equations for each sample $ii$ in the shot record (3). A faster algorithm (10-20x) implem
Algorithm 2.
In Algorithm 3, after the Marchenko equations are solved at $ii-1$, the next time sample $ii$
is initialized with the $ii-1$ result (Zhang and Slob, 2020b). The idea is that to re
the internal multiples at the next time sample there is no n
multiples that were already removed in the previous time-sa
attenuated multiples need to be attenuated plus one (or a fe
new time sample only the multiples have to be removed that we
deviation from the previous results and usually 2 iteratio
the next time sample $ii$. The initial $M_0$ in the fast algorithm is the difference be
$(DD)$ and $k^{-(ii-1)}_{1,n_i}$; the already estimated internal multiples at $ii-1$. The initial $k^-_{1,0}$ is from t
is the previous result $k^{-(ii-1)}_{1,n_i}$. With these initializations $RM_i$ contains only a sma
correction, since it is based on a converged previous resul
get the complete $M_{i+1}$, $DD$ is added to $RM_i$ (Zhang and Slob, 2020b).
In this fast algorithm only one pair of even-odd iterations i
could solve the equations only one time and use that result t
models of numerically modeled data this works fine indeed. Ho
of numerically modeled data and on field data we have to do a fu
and the speed-up of the faster algorithm is limited to one o
we would advise to begin with the basic algorithm and then ve

speed-up the computations. The reason for this limited use
data-sets and a large number of iterations, artifacts, for
get amplifed. The primary refections will still converge, b
in the algorithm and can diverge. In the iterative scheme ea
computed result based on for example 30 iterations. With 10
and can cause artifacts being amplifed to signal level.

In the algorithm we solve the Marchenko equation for every time $ii$. From theory we know
the first event at $ii$ is a primary refector (all multiple refectio
before sample $ii$ are removed by the scheme). Hence, at $ii$ a number of samples are free
of internal multiples. We could make the Marchenko steps with $ii$ at a result for
samples ($n_\varepsilon$ samples), since that is the time resolution we are a
$ii$. This can speed-up the code by $n_\varepsilon$ (typically 20) times. This is similar to the fa
without making any iterations and directly use the previous

Main begin
  Read SU-style input parameters
  Initialization, reading of input parameters and allocat
  READ $R[N_{shots}, i\omega, N_{recv}]$
  $DD[N_{recv}, it] = \mathcal{F}^{-1}\{R^*[j, i\omega, N_{recv}]\}$
  for $ii \leftarrow istart$ to $iend$ do
    $k^-_{1,0}[N_{shots}, it] = k^{-,(ii-1)}_{1,n_i}[N_{shots}, it]$
    $$M_0[N_{shots}, it] = \begin{cases} 0 & 0 < it < n_t - ii + n_\varepsilon \\ DD[N_{shots}, n_t - it] - k^-_{1,0}[N_{shots}, n_t - it] & n_t - ii + n_\varepsilon \leqslant it < n_t \end{cases}$$
    for $i \leftarrow 0$ to $n_i$ do
      synthesis($R, M_i, RM_i$)
      $M_{i+1}[N_{shots}, it] = RM_i[N_{shots}, n_t - it]$
      if ($i$ % 2 == 0) then
        $M_{i+1}[N_{shots}, it] = 0; \quad ii - n_\varepsilon < it < n_t$
      else
        $M_{i+1}[N_{shots}, it] = M_{i+1}[N_{shots}, it] - DD[N_{recv}, it]$
        $k^-_{1,i+1}[N_{shots}, it] = -M_{i+1}[N_{shots}, n_t - it]$
        $M_{i+1}[N_{shots}, it] = 0; \quad 0 < it < n_t - ii + n_\varepsilon$
      end
    end
    $R_t[j, N_{shots}, ii] = k^-_{1,n_i}[N_{shots}, ii]$
  end
end

Algorithm 3: Faster Marchenko algorithm that uses previous resu $ii - 1$ ($k^{-,(ii-1)}_{1,n_i}$) as input for the current time instant $ii$.

The synthesis process is shown in Algorithm 4 and computes the second integrant in the
equation 3.3. The synthesis function is a straightforward matrix
data are stored in such a way that the most inner loop, that s
shot, is contiguous in memory. To speed-up the computationa
the outer loop $N_{shots}$. An alternative implementation of the synthes
loop the outer loop and use a BLAS function to compute the matrix-vector
implementation will also be efficient when all shots are compu
matrix gemm function becomes the kernel of the synthesis process
the integration is carried out over the number of receivers
at the shot position. Thus for $N_{shots}$ the output matrices are computed.
From a computational point of view the transmission compens
the MME algorithm, except for the application of the time-t
the wavelet ($n_\varepsilon$) is applied in the opposite time direction for the T

```
synthesis(R[N_shots, iω, N_recv], M[N_shots, it], RM[N_shots, it])
begin
    Fop[iω, N_shots] = F{M[N_shots, it]}
    RM[N_shots, t] = 0
    #pragma omp parallel for
    for k ← 0 to N_shots do
        for iω ← ω_min to ω_max do
            for i ← 0 to N_recv do
                sum[iω] = sum[iω] + R[k, iω, i] * Fop[iω, i]
            end
        end
        RM[k, it] = F^{-1}{sum[iω]}
    end
end
```

Algorithm 4: Marchenko synthesis kernel. with $i\omega = i * \Delta\omega$ ($= \frac{2\pi}{n_t * \Delta t}$)

of $n_\varepsilon$ in the MME algorithm take into account ... of the wa
instant $it$ in the initialization $M_i$. Suppose that $it$ is the two-way traveltime ...
reflector (see Figure 13a). The reflection of the ... $M_i$ ... MME algorithm (Fi
13a), but in $\bar{M}_i$ ... T-MME algorithm (Figure 13b). In the T-MME algorithm the
event at instant $it$ ... $\bar{v}_1^-$ ... updated in the MME algorithm the r
from the original shot $v_1^-$. When ... $it$ ... between two ref
then there is no difference between the ... (Figure 13c) ... T-MME ... (Figure 13d) ... the MM
scheme.

To get to the T-MME scheme from ... Algorithm ... in the ... $+n_\varepsilon$ ... time ...
$-n_\varepsilon$. The $R_t$ ... $R_r$ ... that contains the transmission compensated pr

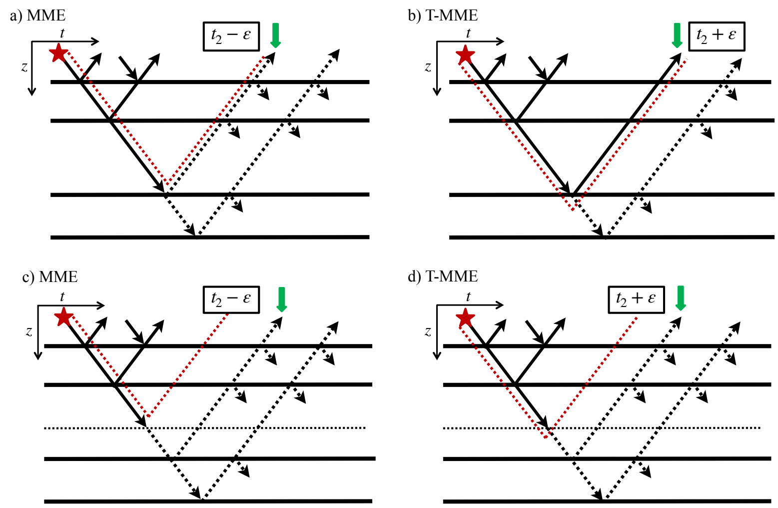
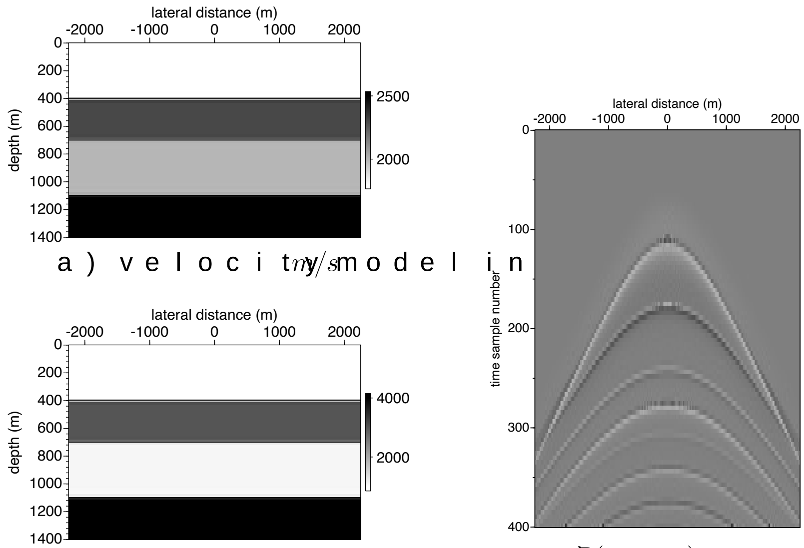



Figure 13: Comparison of the MME and and T-MME schemes. Figures a ...
$t_2$ equal to the two-way traveltime of the third refector. The ...
a red dotted line. The dotted lines ... $M_i$ ...
in $M_i$, after application of the time window. Figures c) and d)




a) velocity model in $m/s$

b) density model in $kg/m^3$

c) $R(\mathbf{x}_R, \mathbf{x}, t)$

Figure 14: Four layer model with velocity (a) and density (b) par
position $\mathbf{x} = (x = 0, z = 0)$ and receivers $\mathbf{x}_R = (x = x_r, z = 0)$ (c). The source wavelet $R(x)$ has a flat
fat frequency spectrum from 5 to 90 Hz.

The Marchenko algorithm is illustrated with a 1.5-dimension
14. The numerical modeling is carried out with a finite-difference (Thorbecke and
Draganov, 2011) that is also included in the software package. The
the reflection response $R(\mathbf{x}_0', \mathbf{x}_0'', t)$ is approximately a sinc-function with a f
between $f_{min}$ and $f_{max}$ ($5 - 90$ Hz) to represent a deconvolved source wavel
program for modeling $R(\mathbf{x}_0', \mathbf{x}_0'', t)$ a source of vertical force is chosen. The re
and measure the pressure field. A fixed spread acquisition is at $z = 0$ from $-2250$ to $2250$ m
distance between the 901 sources / receiver positions is $5$ m. The receivers have a ti
interval of $4$ ms and 1024 recorded time samples.

### 4.3.1 The first iterations

Figure 15 demonstrates the first iteration $n = 1$ to compute $M_1$ for a time sample-
number 276 ($t = 1.10$ s). The time sample 276 corresponds to the zero-
third reflector. In this first step $R$ and $M_0$ are correlated with each other
shot record. In our example we use $R_0(\mathbf{x}_R, \mathbf{x} = (0,0), t)$ (shot number 451) of the record;
Before the correlation is carried out the selected shot re
$276 - n_\varepsilon$, multiplied by -1 and time reversed $M_0(t)$ in equation 39 (in Figure we l
15 the shot record is convolved with a Ricker wavelet to redu
(deconvolved) wavelet $R$ (Figure 15a). The number $n_\varepsilon$ in this example is $n_\varepsilon = 20$ samples
excludes the reflection from $M_0$. In the middle of Figure 15 the shot $R$ correlated of
15a, where we used source receiver reciprocity $M_0(-t)$ (Figure 15b) correlated
to give the result. The events in Figure 15 in the first and second ref
first internal multiple between the first and second reflector
the auto-correlation of the $t = 0$ wavelet (reflection) at time zero
at the bottom of the panel). Note that the long train of event
15 can interfere with events at the end of the time axis. To o
pad the time axis with zeros before the transformation to th
computed.
The correlation result is time reversed for the first 400 samples in Figure 15e. The
three events in Figure 15d originate from correlation with the

the first three events ($r_1, r_2, m_1$) in the shot record. According to [41] to obtain the integr[...] an output $M_1$ the traces in Figure [15]d are summed together. The stationary [...] in Figure [15]d give a contribution in the result of the summation. Be[...] events (both in time and space) give unwanted contributions [...] The integration result is set to zero for samples larger than $276 - n_\epsilon$ and [...] (the middle $M_1$ trace shown in Figure [15]e). In Figure [15]e the truncation appears to be a[...] but that is the result of the shift upward in time with the arrival time [...] truncation at $276 - n_\epsilon$ is indicated with a dotted line. There are two [...] a few linear artifacts. The first hyperbolic event originates at $r_2^*.r_1$ and $m_1^*.r_2$, and the second hyperbolic event from the $c_2 = m_1^*.r_1$ internal [...] The linear events are unwanted artifacts due to truncation a[...] taper at the truncation boundaries in time and space.

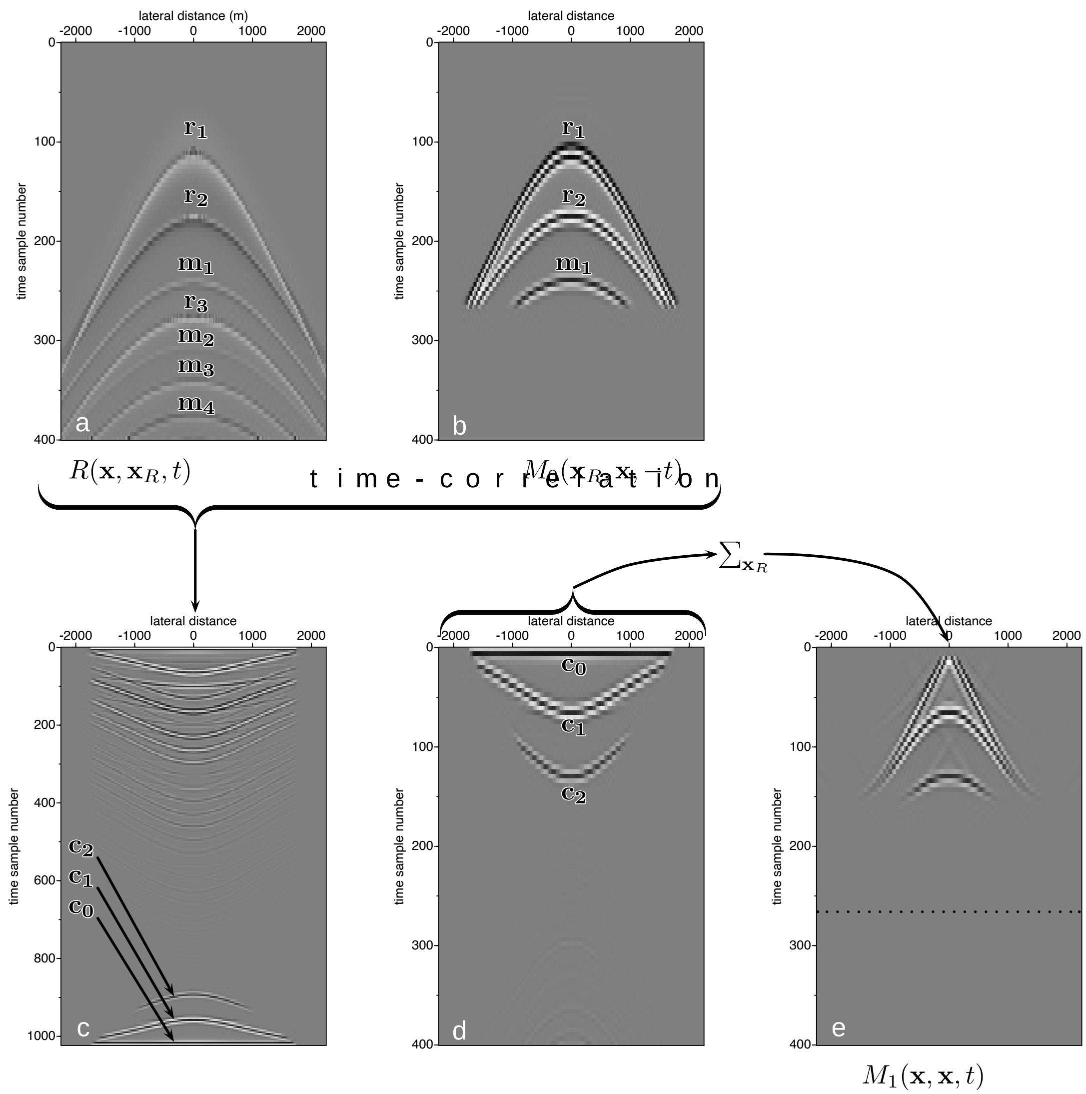



Figure 15: Computational steps to compute $M_1$ from $M_0$ at time sample number 276. The m[...] record $R$ is shown in (a); time $276 - n_\epsilon$, and convolved with a Ricke[...] it gives $M_0$ in (b). Time-correlation of (a) with (b) gives (c). Af[...] time window again gives (d). The traces in (d) are summed to [...] events above sample $276 - n_\epsilon$ will end-up in the $M_1$ (e). The muted window $t_\epsilon$ [...] later than $t=0$ is needed to mute the autocorrelation of the first event [...] $r_i$ the [...] reflector and $m_l$ the $l$'th multiple. In (c) the correlated events [...] $\mathbf{c_0} = \mathbf{r_1^*.r_1} + \mathbf{r_2^*.r_2} + \mathbf{m_1^*.m_1}$, $\mathbf{c_1} = \mathbf{r_2^*.r_1} + \mathbf{m_1^*.r_2}$, and $\mathbf{c_2} = \mathbf{m_1^*.r_1}$.

Figure 16 demonstrates the computation of $M_2$ from $M_1$ (Figure 15) [illegible] iteration according to equation 40 with $n=1$. The reflection data (Figure 16a) are convolved with $M_1$ (Figure 16b) that contains three main events; $a_1$ and two correlation artifacts ($c_1$ and $c_2$). [illegible] Convolving $M_1$ with the middle shot record $R$ gives Figure 16c. The hyperbolic events are now back at the same times as reflection events in $M_1$ [illegible] the sho[illegible] convolves with $R$ and introduces many (mostly linear) artifact[illegible] reversed in time (Figure 16d). After the integration over $x_R$ in equation 40 [illegible] coordin[illegible] becomes the middle of $M_2$ (Figure 16e). Most of the linear artifacts are re[illegible] to the destructive interference in the integration [illegible] $a_2 = r_1.a_1$ [illegible] $M_2$ [illegible] first term in the sum in equation 37 [illegible] $M_2$ [illegible]; last events in $M_2$, [illegible] time re[illegible] presented in Figure 16, will already attenuate the multiple event $m_1$.

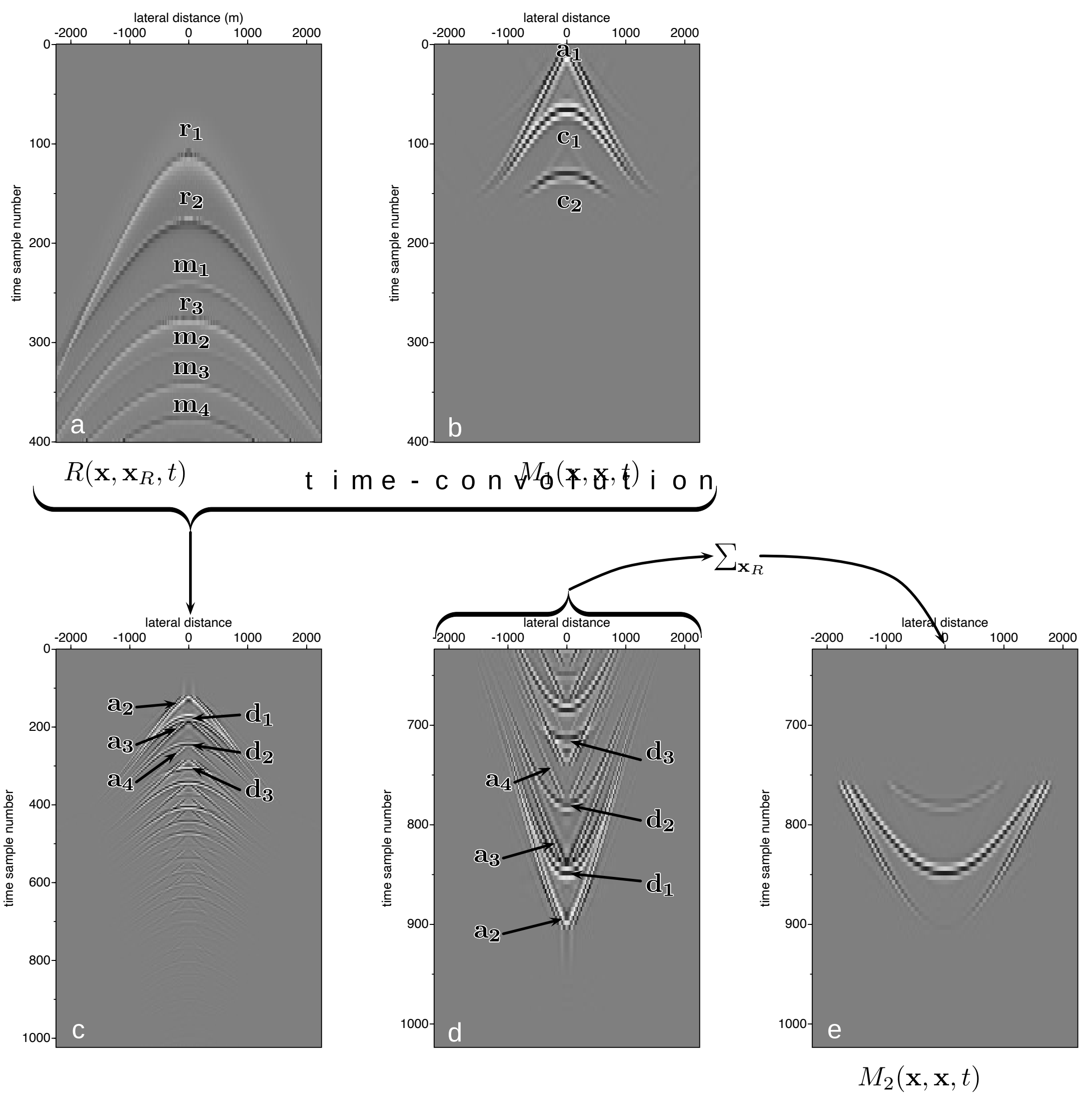
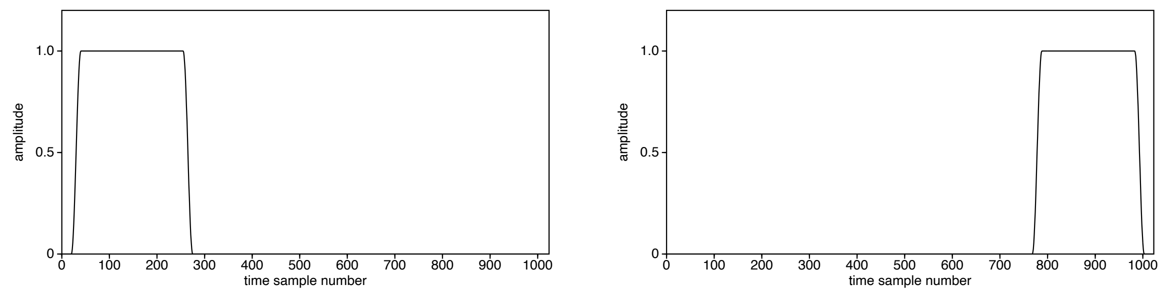



Figure 16: Computational steps for computing $M_2(\mathbf{x}, \mathbf{x}, t)$ at time sample number 276. Th[illegible] shot record $R$ is shown in (a) and $M_1$ truncated after sample 276 (Figure 15) [illegible] comput[illegible] (b). Time-convolution of (a) with (b) gives (c). After time-[illegible] again gives (d). The traces in (d) are summed together and o[illegible] $n_t - 276 + n_\varepsilon$ end-up in the middle of $M_2$ (e). The label $r_l$ indicates the $l$'th reflec[illegible] the $l$'th multiple $m_n$ and [illegible] artifact. The labeled events in $R$ (a) and from $M_1$ (b) are: $\mathbf{a_2} = \mathbf{r_1}.\mathbf{a_1}, \mathbf{a_3} = \mathbf{r_2}.\mathbf{a_1}, \mathbf{a_4} = \mathbf{m_1}.\mathbf{a_1}, \mathbf{d_1} = \mathbf{r_1}.\mathbf{c_1}, \mathbf{d_2} = \mathbf{r_1}.\mathbf{c_2} + \mathbf{r_2}.\mathbf{c_1}$ and $\mathbf{d_3} = \mathbf{r_2}.\mathbf{c_2} + \mathbf{m_1}.\mathbf{c_1}$.

To compute $M_4$ (in general odd number $i$), [illegible] $M_i$ [illegible] updates [illegible] shifted backward

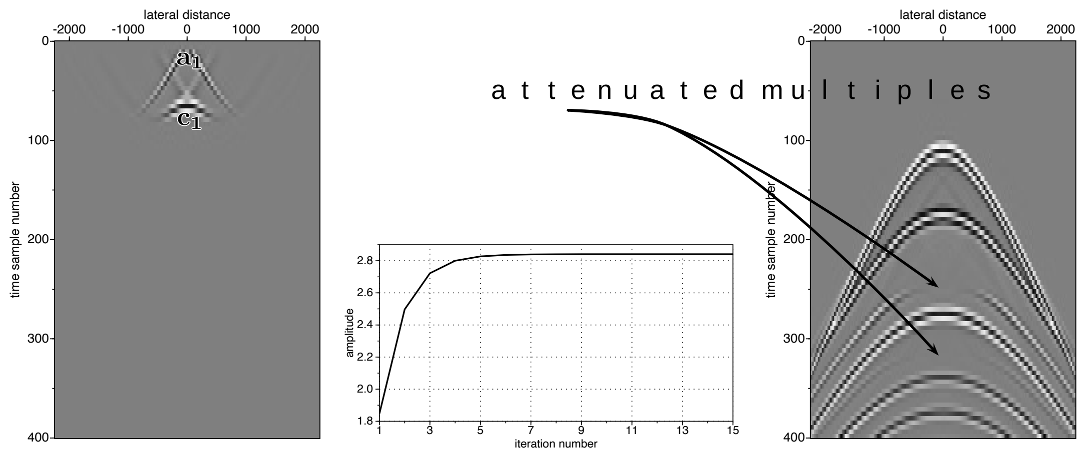
lation) with the time $M_0$. To compute the even numbered updates ($M_2$, $M_i$) of $M_1$, times to
events are shifted forward (convolution) in time. The even
the scheme. Each even $M_i$ iterate, and $M_1^+$ and the updated $M_i$ and $M_1^-$ updates
scheme reverts the time-axis for each iteration, hence the
switches also. These time windows are shown in Figure 17 for 276 samples. In these time window
smooth cosine shaped transition zone is used to reduce the t




a) Time-window for even iteration number   b) Time-window for odd iteration

Figure 17: Time-window functions in the Marchenko scheme with a
zone. This transition zone has $0.5 n_\varepsilon$ samples and $n_\varepsilon$ samples at the start of the

### 4.3.2 Multiple removal in action




a) $v_{1,1}^+(t)$   b) Convergence of $c_1$   c) $k_{1,30}^-(t)$

Figure 18: Creation of the event $c_1$ (that annihilates all the internal m
first and second reflector $a_1$ is originated in the initial $v_{1,1}^+(t)$ for Picture a) s
Marchenko iteration ($ii = 200$ samples) the number convergence of the maximum $c_1$
is shown as function of the iteration count. Figure c) shows $k_{1,30}^-(t)$ after 30
iterations.

The results in Figure 18 are partial solutions of the Marchenko equatio
$ii = 200$. After applying the time window to $M_0$, the $200 - n_\varepsilon$ samples i
no internal multiple reflections. The first event $M_0$ is between time sample 0 and sample 2
and $n_2$, but not $n_1$, see Figure 15. In the first iteration, according to equation (44),
extra event (Figure 18a) event $c_1$ is created to correct for the amplitude of $v_1^-$.
Note that the time window in Figure 18 appears around sample 80, but thi
applied on $M_0$ shifted by the correlation to negative times and ti
event $c_1$ converges to the amplitude that can annihilate the ampl
converge $v_1^+$ and the reflection data through equation (40) all multiples aris

between the first and second reflector (37). This scheme then removes the without ever having 'seen' $r_1$ and $r_2$ at the multiple, a free event that all the internal multiples between these reflectors (18); the artefacts $k_{1,30}(t)$ are removed. The ar computed according to equation (42); multiples that are already p are only partly removed because $r_2$ is only used as a small part of sample 200 and Repeat scheme for samples larger than 200 with $r_1$ and $r_2$ will cancel all the ... offsets for all internal multiples between $r_1$ and $r_2$.

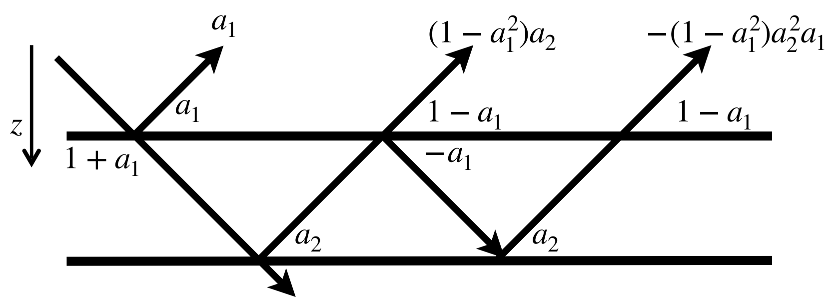



Figure 19: Sketch of the ray-paths and reflection and transmissio velocity and variable density model. The local reflection co $a_1$ and $a_2$ respectively.

For the investigation of the $v_1^+$ amplitudes, for the sake of argum reflection coefficient is a constant (15). The vertically reflections co $a_1, a_2$ for respectively $\mathbf{r}_1, \mathbf{r}_2$. We consider the situation in Figure 18, for times 200 that only creates $s_1^+$ on. We demonstrate that after sufficient i in $v_1^+$ has converged to an amplitude that can cancel the first-or multiples related to that event in $M_0$. For sample 200, only $r_1, r_2$ have reflectio amplitude:

$$r_1^a = a_1, \tag{50}$$

$$r_2^a = (1-a_1^2)a_2. \tag{51}$$

Figure 19 gives a sketch of the reflection paths and reflection and t reflector case. According to equation (41), $M_0$ is correlated with $R$ and integrated over the receiver coordinate. After applying the time window, $M_1$ ... $\mathbf{c}_1 = \mathbf{r}_2^*.\mathbf{r}_1$ in Figure 18 with amplitude:

$$c_{1,1}^a = a_1(1-a_1^2)a_2. \tag{52}$$

The second subscript of $c_{1,1}^a$ indicates the iteration number $R$. In the next iteration (according to equation (40)) after windowing, $\mathbf{r}_1.\mathbf{c}_1$ remains only at the event time of the reflection of the second reflector with amplitude

$$c_{1,2}^a = a_1^2(1-a_1^2)a_2. \tag{53}$$

In each next iteration, alternating (41) and (40), multiplication with $a_1$ is added, in general for iteration $i$ we have

$$c_{1,i}^a = (a_1)^i(1-a_1^2)a_2. \tag{54}$$

Summation of all $c_{1,i}^a$ iterations gives the final amplitude of the $c_1$ multiple $v_1^+$. The initialization of $v_1^+$ and the summation of the odd terms lead

$$\begin{aligned}
\sum_{i=0}^{n_i} c_{1,1+2i}^a &= \sum_{i=0}^{n_i}(a_1)^{1+2*i}(1-a_1^2)a_2, \\
&= a_1a_2 - a_1^3a_2 + a_1^3a_2 - a_1^5a_2 + a_1^5a_2 - a_1^7a_2 + \ldots \\
&\approx a_1a_2.
\end{aligned} \tag{55}$$

Application of $v_1^+$ to the data creates multiples $u_1^-$ and is shown in Figure 18. The
The first-order internal multiple from the first $c_1$ in Figure 18a, and the multi
amplitude $a_1 a_2$, we will meet each other in time just below the first re
annihilator cancels the first-order downgoing internal mult
To be able to cancel the first downgoing internal multiple the
as that event. The first $\mathbf{m}_1$ in Figure 15 has the amplitude

$$m_1^a = -(1 - a_1^2)a_2^2 a_1. \quad (56)$$

After convergence of the scheme $c_1$ ...
$\mathbf{r}_2$ or $R$ in the next iteration and $\mathbf{m}_1$ ... has the same amplitude as $m_1^a$

$$\begin{aligned} c_1^a r_2^a &= = a_1 a_2 . r_2^a, \\ &= (1 - a_1^2)a_2^2 a_1. \end{aligned} \quad (57)$$

This result is added to the data $\mathbf{m}_1$ as shown in Figure 18 and equal ...
37. Furthermore, $\mathbf{c}_1$ ... $\mathbf{m}_1$ ... after the annihilator of the seco
$\mathbf{c}_1$ will automatically annihilate all higher-order multiples
To complete the amplitude analysis, the $\bar{n}_1$ (amplitude of ...
equation 43 for the T-MME scheme) can be compared to equation 48 and ...
the even amplitude 54. The ... $\bar{a}_1$ ... reversed $DsD$ shot re
in Algorithm 2. The summation of $c_{1,i}^a$ ... the time of the second reflect
final amplitude for the $\bar{v}_1$ second reflector in

$$\begin{aligned} a_2 =& (1 - a_1^2)a_2 + \sum_{i=1}^{n_i} a_1^{2*i}(1 - a_1^2)a_2 \\ =& a_2 - a_1^2 a_2 + a_1^2 a_2 - a_1^4 a_2 + a_1^4 a_2 - a_1^6 a_2 + \ldots \\ \approx& a_2. \end{aligned} \quad (58)$$

This shows that the transmission compensated $v_1^+$ ... implemented ...
in the T-MME scheme. The approximation sign is due to a limit
implementation.
Figure 20 is obtained in the same way as Figure 18, but with high contrast layers. T
layers is the same as used in Figure 18, but the density contrast between the la
from a factor 3 (1000-3000) to 10 (150-1500). Compared to Figure
this high contrast model, the higher-order terms will have large
media and require more iterations for convergence.

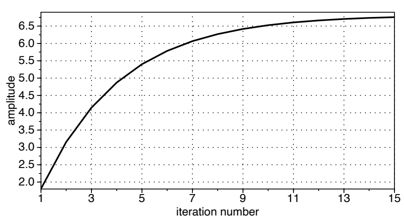
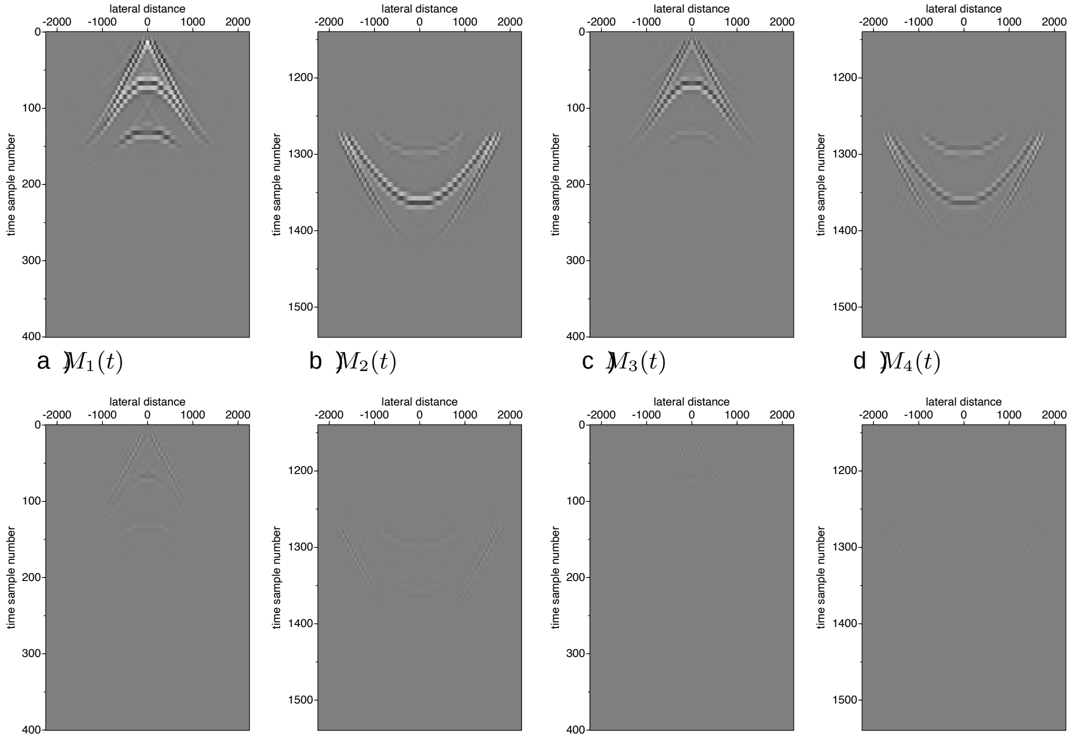



Figure 20: Convergence of the maximum amplitude of $c_1$ in Figure 18a) ... all the internal multiples between the first and second refle

### 4.3.3 Higher iteration counts

The first few iterations $M_i$ are shown in Figure 21. The truncation time is
at sample 276 and a first-order multiple of the second layer i ... $M_0$
after time-truncation. For higher numbers of iterations th
indicating that the scheme converges. All the updates show
amplitude of the events change during the iterations.



a) $M_1(t)$ b) $M_2(t)$ c) $M_3(t)$ d) $M_4(t)$

e) $M_{11}(t)$ f) $M_{12}(t)$ g) $M_{19}(t)$ h) $M_{20}(t)$

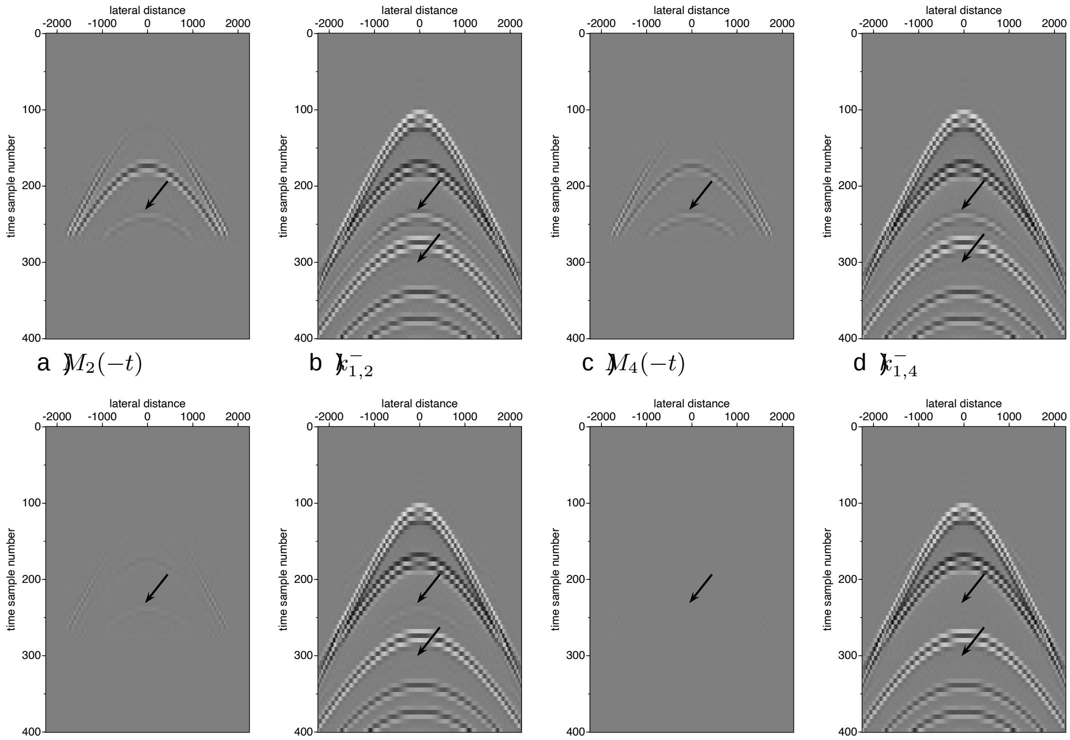
Figure 21: $M_i$ fields for a focal time sample $i = 276$, the zero-offset arrival of the t… figures are plotted with the same clipping factor.

In the odd iterations $k^-_{1,i}(t)$ is the function updated with $M_i(t)$ (4.2) … the second and four selected iterations. As shown in Figure 22 after two iterations all order multi… with incorrect amplitudes. In the following iterations the … because the removal of the first-order multiple improves. Af… (indicated with arrows) have further attenuated 22 … and are not … 22. The higher-order multiples do not have to be removed by $u_1^+$, but are removed automatically by removing the first-order multiple.

In Figure 22 one can observe that the first internal multiple (po… attenuated between $276-n_\varepsilon+1$ and … but it is not yet completely attenuated b… $276-n_\varepsilon$. The first $276-n_\varepsilon$ samples of $v_1^-$ contain the information on attenuation … is constructed, while $276-n_\varepsilon+1$ samples onward from $u_1^-$ … where the multiple is a… attenuated. The constant-time cross-section of $k^-_{1,i}$ at sample 276 is also … the final output $R_i$ at sample 276 of $u_1^-$ part (between samples … of $k_1^-$) … reflector has its local reflection coefficient … $u_1^-$ … physical amplitude with two-way transmission effects.

### 4.3.4 Different time instances

In Figure 23 the Marchenko equations are solved for different time … investigate how $g_1^-$ changes for larger sample numbers. It is observed … also before $i$ and beyond $i$ … related to internal multiples are a… corresponds to the arrival time of the 23 … 23 … reflector. The … 276 and we do not observe a change in the number of events. Ho… 23a) to 276 (Figure 23f) one can see that the multiple, arriving in tim… reflector, gets more and more attenuated at larger and larger … fast algorithm, to compute the solution in the next time sam… a few iterations are sufficient to solve for the multiple atte…

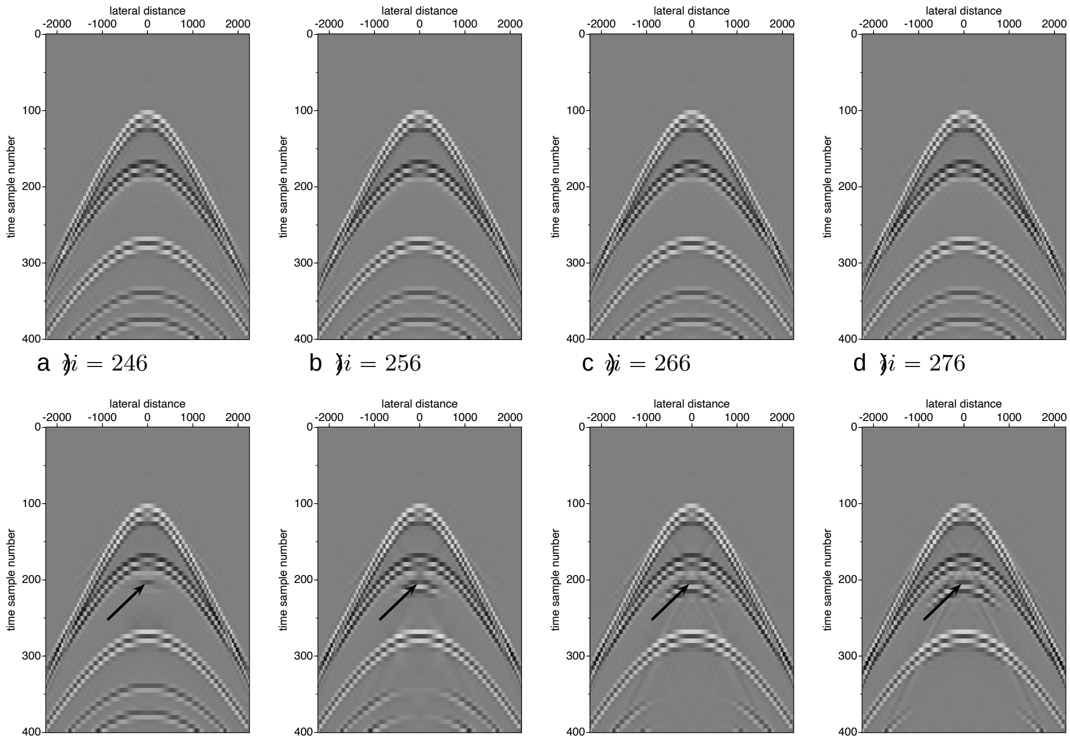



a) $M_2(-t)$ b) $k^-_{1,2}$ c) $M_4(-t)$ d) $k^-_{1,4}$

e) $M_{10}(-t)$ f) $k^-_{1,10}$ g) $M_{20}(-t)$ h) $k^-_{1,20}$

Figure 2.2: Updates $k^-_{1,i}$ for a focal time $t_2 = 276$ at sample iterations. The arrow ind
first and second-order internal multiple between the first an

time $i$ passes the arrival time of the third reflector, a non-phy
Figure 2.3a 2.3b) appears just below the arrival time of the second re
the annihilator $a_1^+$ that compensates all internal multiples creat
reflector. The cancelation of the internal multiples (2.3a-2.3b), is obse
all internal multiples related to the third reflector are can
Figure 2.4a and 2.4b sketches of the situation where the time $i$ corresponds to a depth ta
or below the third reflector, respectively. The event that c
the second reflector (2.4a) coincides in travel time with the re
reflector and also compensates for the transmission loss of
internal multiples related to the third reflector (2.4a) are compe
which coincides with the reflection time of the third reflecto
reflector (upward red arrow) and creates the non-physical re 2.3-2.3b
pointed with an arrow.




a) $i = 246$ b) $i = 256$ c) $i = 266$ d) $i = 276$
e) $i = 286$ f) $i = 296$ g) $i = 306$ h) $i = 316$

Figure 23: $k_1^-$ after 32 iterations with $i = 246$ to $i = 316$ in time steps of 10 sampl
From each panel a constant-time cross section is selected as
multiple-free data. The arrows point to an event that compen
the second and third refector.

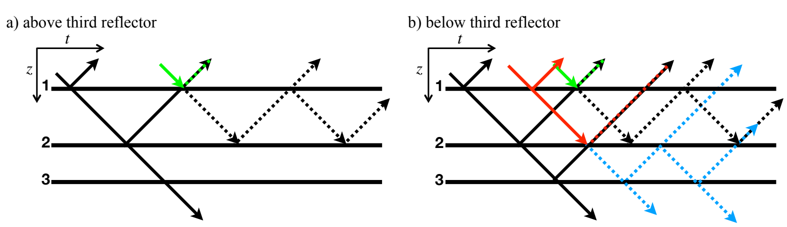
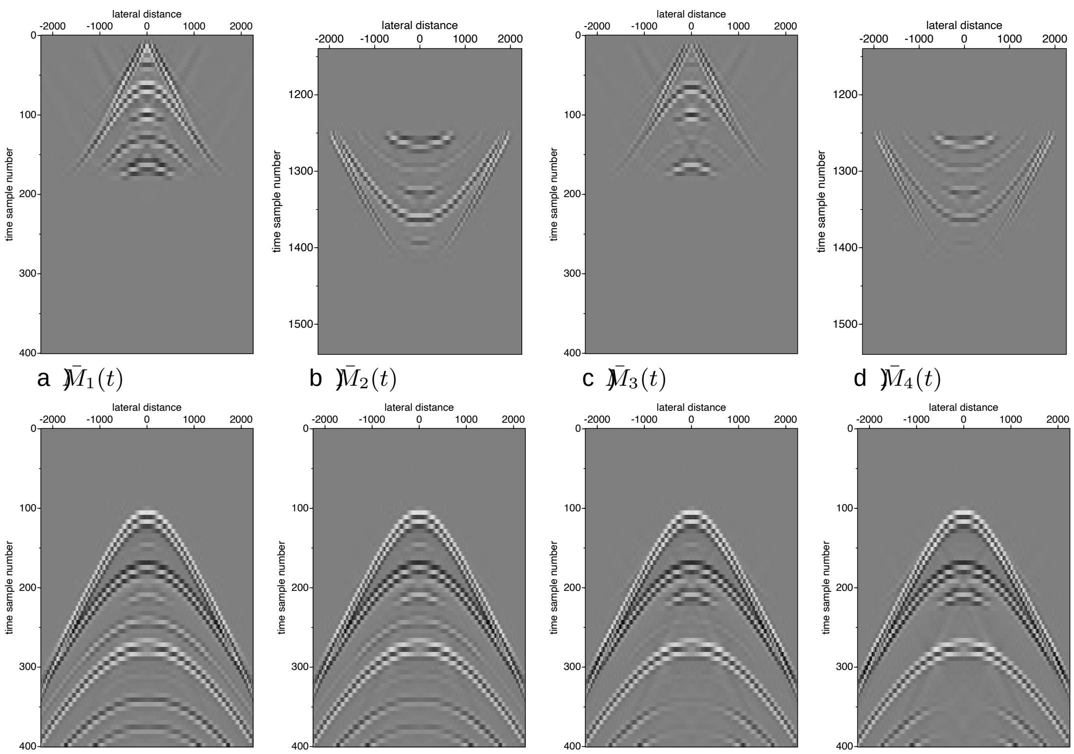



Figure 24: Compensation of internal multiples by events (colou
Marchenko method, applied for a point above (a) and below (b
are numbered from top to bottom.

Figure 25 show the same pictures as Figure 21 - Figure 25 show the same as Fig
22, but now with the T-MME scheme (Zhang et al., 2018). The T-MME scheme for t
sample 276 includes the reflection of the third reflector, since $276+n_\varepsilon$ ... The extra reflector introduced in $\bar{M}_i$ and $\bar{v}^-_{1,i}$, the non-physical primary,
second reflector, is clearly visible. After 20 iterations (Figure 25) ... the time instant 296). The difference is that in the T-MME scheme ($t_2+\varepsilon$ = 276+8) the value at $t_2$ (time sample 276) in $\bar{v}^-$ is exactly the right reflection coefficient in the final data output $R_r[276]$, while in the MME scheme (Figure 23) the truncation starts at $t_2-\varepsilon$ (sample 296-8) and the value at time sample 296 is the correct value for the ph... and is stored in the $R_t[296]$ data output (
In the example for the MME scheme (58) we have shown (equation ...) that the second reflector in $\bar{v}^-_1$ was modified from its physical amplitude to its local amplitude. It is exactly this feature that T-MME exploits. ... arrival time of a reflector there is a decision to be made whe...
Setting the truncation time at $t_2-\varepsilon$ the amplitude at time $t_2$ is correctly obtained in $\bar{u}^-_1$. Changing the truncation time to $t_2+\varepsilon$, the amplitude at time $t_2$ is correctly obtained in $\bar{v}^-_1$ ... $\varepsilon$ is the time duration of the source wavelet that allows to ... $t_2-\varepsilon$ ... is introduced in $\bar{v}^-_1$ and in $\bar{u}^-_1$ is correct at $t_2$, whereas by taking $t_2+\varepsilon$ ... $\bar{v}^-_1$ will be correct at $t_2$.
The transmission compensated (T-MME) scheme retrieves primary ... coefficients, while in the regular (MME) scheme the primary ref... coefficients that include transmission losses. The local refle... for a horizontally layered medium, but in laterally varying ... (Zhang et al., 2019). The only computational difference between the T-MME and ... time-truncation window.



a) $\bar{M}_1(t)$ b) $\bar{M}_2(t)$ c) $\bar{M}_3(t)$ d) $\bar{M}_4(t)$
e) $\bar{v}^-_{1,2}(t)$ f) $\bar{v}^-_{1,4}(t)$ g) $\bar{v}^-_{1,10}(t)$ h) $\bar{v}^-_{1,20}(t)$

Figure 25: Panels (a)-(d) show $\bar{M}_i$ for a focal time $t_2=276$ with the transmission compensated scheme T-MME after $i=1,2,3,4$ iterations. Panels (e)-(h) show $\bar{k}^-_i$ for a focal time $t_2=276$ with the transmission compensated scheme T-MME after $i=2,4,10,20$ iterations. All figures are plotted with the same clipping fa...

# marchenko_primaries

The marchenko_primaries program has the following parameters and opti

```
MARCHENKO_primaries - Iterative primary reflections retrieval

marchenko_primaries file_tinv= file_shot= [optional parameters]

Required parameters:

  file_shot= .............. Reflection response: R

Optional parameters:

 INTEGRATION
  ishot=nshots/2 ........... shot number(s) to remove internal multiple
  file_tinv= .............. shot-record to remove internal multiples
  file_src= ............... optional source wavelet to convolve select
 COMPUTATION
  tap=0 ................... lateral taper R_ishot(1), file_shot(2), o
  ntap=0 .................. number of taper points at boundaries
  fmin=0 .................. minimum frequency in the Fourier transform
  fmax=70 ................. maximum frequency in the Fourier transform
  plane_wave=0 ............ model plane wave
  src_angle=0 ............. angle with horizontal of plane source arra
  src_velo=1500 ........... velocity to use in src_angle definition
  t0=0.1 .................. time shift in plane-wave source wavelet fo
 MARCHENKO ITERATIONS
  niter=22 ................ number of iterations to initialize and res
  niterec=2 ............... number of iterations in recursive part of t
  niterskip=50 ............ restart scheme each niterskip samples with
  istart=20 ............... start sample of iterations for primaries
  iend=nt ................. end sample of iterations for primaries
 MUTE-WINDOW
  shift=20 ................ number of points to account for wavelet (ep
  smooth=shift/2 .......... number of points to smooth mute with cosine
 REFLECTION RESPONSE CORRECTION
  tsq=0.0 ................. scale factor n for t^n for true amplitude r
  Q=0.0 ......,............ Q correction factor
  f0=0.0 .................. ... for Q correction factor
  scale=2 ................. scale factor of R for summation of Mi with M
  pad=0 ................... amount of samples to pad the reflection ser
 OUTPUT DEFINITION
  file_rr= ................ output file with primary only shot record
  file_dd= ................ output file with input of the algorithm
  file_iter= .............. output file with -Mi(-t) for each iteratio
          ............... M0.su=M0 : initialisation of algorithm
          ............... RMi: iterative terms
          ............... k1min.su: k1min terms
  file_vplus= ............. output file with v+
  file_vmin= .............. output file with v-
  file_uplus= ............. output file with u+
  file_umin= .............. output file with u-
  file_update= ............ output file with updates only => removed ir
  T=0 ..................... :1 compute transmission-losses compensate
  verbose=0 ............... silent option; >0 displays info


 author  : Lele Zhang & Jan Thorbecke : 2020
```

Defining file_iter writes for each iteration $-M_i(-t) = RM_i(t)$ in the algorithm ... 1 before applying the mute window. $k_{1,i}^-$ ... $k_{1,i}^+$ ... the update ... setting verbose=2 the energy of the focusing update term and can be used to monitor the convergence ... file_update= ... the program writes the updates (= estimated ... useful when the (modeled) data does not have the correct amp The parameter niterskip enables the fast algorithm when it is set large value istart=... run with iterations ... niterskip set to >1 the fast algorithm i effect and the next iterations use ... niterec=2 iterations ... skip iterations

the scheme uses the niter full iterations to avoid possible cumulative num
amplifed artefacts. The istart scheme continues with iterations and t
itself. By setting istart=0 the scheme does not do any new iterations in t
uses the result of the previous iterations=0 ... It is a small shift to
≈ shift samples and is possible due to limited bandwidth of the d
The T= parameter is a switch to enable the T-MME algorithm. The plane_wave
src_velo, src_angle uses plane-waves as input shot, Meles et al. (2016, 2020) as explained i
The commands to reproduce all fgures in this paper can be found in marchenko/demo/mdd
The README_PRIMARIES in that directory explains in detail h
plicated (lateral varying) model ... The Marchenko ... demo/ ... complex
take several hours to compute the refection data and is not d
Besides the new Marchenko primaries removal program the pac
fnite diference modeling code, that is used to model all da
(Thorbecke and Draganov, 2011) and the standard Marchenko (Thorbecke et al., 2017). The
directory contains programs to calculate source ... wavelets (makewave)
and programs for basic processing steps.

Description of files:

1) model.scr computes the model and the 'basis' shot of R =
   - runtime on 4 cores is 4-5 minutes and produces a 3.3 GB
2) itertions.scr computes the intermediate results of the
nuscript.
   - runtime on 4 cores is
3) epsPrimaries.scr selected output from step 2) are conve
   To reproduce the postscript files of the manuscript SU p
-3) epsModel.scr to generate the postscript files for the

optional scripts not needed to reproduce the figures:
+) primaries.scr computes the internal multiple attenuat
   - runtime on 4 cores is ~500 s.
+) primariesPlane.scr: computes the internal moval schem
+) clean: remove all produced files and start with a clean

---------------------------
To reproduce the Figures in the Manuscript:
---------------------------
* Figure 2: Model + Initial wavefield

==> run model.scr to generate the data .su files: this wil
- hom_cp.su, hom_ro.su
- model10_cp.su, model10_ro.su
- shot5_fd_rp.su
- shot5_hom_fd_rp.su
- shot5_rp.su
- wavefw.su

==> run './epsPrimaries.scr Figure2' to generate the post

model_cp_line.eps => Figure 2a
model_ro_line.eps => Figure 2b
shotx0_rp.eps => Figure 2c

It also produces two extra pictures of the wavelet used in
wavefw_freq.eps

```
wavefw.eps

-------------------------
* Figure 3: First Iteration

==> run './iterations.scr Figure34910' to compute the int
This will take 15 seconds. The generated files are:

- M0_276000.su
- Mi_2760##.su
- k1min_2760##.su
- v1plus_2760##.su
- iter_2760##.su (not used)
- pred_rr_276.su (not used)
- DDshot_450.su (not used): selected shot record convolve

where ## ranges from 01 to 34

To generate the postscript files for Figure 3:

==> run './epsPrimaries.scr Figure3'

This will produce the following files:

shotx0_rp.eps  => Figure 2c == Figure 3a
M0_276000_flip.eps => Figure 3b
fconvN0fulltime.eps => Figure 3c
fconvN0flip.eps  => Figure 3d
Mi_276001.eps => Figure 3e


-------------------------
* Figure 4 second iteration
To generate the postscript files for Figure 4:
==> run './epsPrimaries.scr Figure4'

This will produce the following files:

fconvN1fulltime.eps => Figure 4c
fconvN1flip.eps => Figure 4d
Mi_276002.eps => Figure 4e

The window time function in Figure 5 is not reproduced.

-------------------------
* Figure 6 v1plus and convergence
==> run './iterations.scr Figure6' to compute the marchen

To generate the postscript files for Figure 6:
==> run './epsPrimaries.scr Figure6'

This will produce the following files:
v1plus_200001.eps => Figure 6a
v1plus_max.eps => Figure 6b
k1min_200030.eps => Figure 6b

-------------------------
```

* Figure 8 To compute the convergence for a strong contras
cd strongContrast
==> run ./model.scr
==> run ./iterations.scr Figure8

To generate the postscript files for Figure 8:
==> run './epsPrimaries.scr Figure8'

This will produce the following files:
v1plusStrong_max.eps => Figure 8

Don't forget to go back to the main directory with the regu
cd ../

---------------------------
* Figure 9 iterations M_i

To generate the postscript files for Figure 9:
==> run './epsPrimaries.scr Figure9'

This will produce the following files:
Mi_276002.eps => Figure 9b
Mi_276004.eps => Figure 9d
Mi_276012.eps => Figure 9f
Mi_276020.eps => Figure 9h
Mi_276001.eps => Figure 9a
Mi_276003.eps => Figure 9c
Mi_276011.eps => Figure 9e
Mi_276019.eps => Figure 9g

---------------------------
* Figure 10 iterations M_i and k_1^-

To generate the postscript files for Figure 10:
==> run './epsPrimaries.scr Figure10'

This will produce the following files:
Mi_276002flip.eps => Figure 10a
k1min_276002.eps => Figure 10b
Mi_276004flip.eps => Figure 10c
k1min_276004.eps => Figure 10d
Mi_276010flip.eps => Figure 10e
k1min_276010.eps => Figure 10f
Mi_276020flip.eps => Figure 10g
k1min_276020.eps => Figure 10h

---------------------------
* Figure 11 iterations k_1^- for different ii 246:316:10

To generate the data
==> run ./iterations.scr Figure11
this will take ~2 minutes and generate a lot of files

To generate the postscript files for Figure 11:

```
==> run './epsPrimaries.scr Figure11'

This will produce the following files:
k1min_246032.eps => Figure 11a
k1min_256032.eps => Figure 11b
k1min_266032.eps => Figure 11c
k1min_276032.eps => Figure 11d
k1min_286032.eps => Figure 11e
k1min_296032.eps => Figure 11f
k1min_306032.eps => Figure 11g
k1min_316032.eps => Figure 11h


-------------------------
* Figure 13 iterations M_i and k_1^- for ii-276 T-MME sche

To generate the data
==> run ./iterations.scr Figure13
this will take ~15 seconds
****NOTE this will overwrite the results of the MME-schem

To generate the postscript files for Figure 13:
==> run './epsPrimaries.scr Figure13'

This will produce the following files:
Mi_276002T.eps => Figure 13a
Mi_276004T.eps => Figure 13b
Mi_276001T.eps => Figure 13c
Mi_276003T.eps => Figure 13d
k1min_276002T.eps => Figure 13e
k1min_276004T.eps => Figure 13f
k1min_276010T.eps => Figure 13g
k1min_276020T.eps => Figure 13h
```

This is mainly a copy of Computers & Geosciences

Seismic imaging is a technique to image geological structur
wavefields measured at the surface of the earth. The measured
activated and controlled sources such as air-guns or vibra
source of the wavefield can originate from earthquakes, ocea
as traffic. The primary reflection of a geological structure,
propagating wavefield, is of main interest and is used to com
geological structure, wavefields are partly reflected upwar
Between two strong reflecting structures, the wavefield can
generate so called internal multiples. These multiple refle
surface and difficult to distinguish from primary reflection
migrated from time to depth and construct an image of the su
recognized as such, they will get imaged being primary reflec
multiples distort the actual image of the subsurface; the d
structures that are positioned along with the primary refle
important to recognize these multiple reflections, and if po
This removal can be performed at different stages of the proc
subsurface. The internal multiples can be directly remove
redatuming step or after the imaging step. For removal after
imaged multiples is subtracted from the image to obtain an i
discuss a method for removing internal multiples during the
Besides internal multiples that are reflected between bound
free-surface-related multiples. These multiples are gene
bounce back into the subsurface by the surface of the earth.
this paper. They are assumed to be removed prior to the remov
The Marchenko algorithm can eliminate internal multiples f Slob et al. 2014;
Behura et al., 2014). In this algorithm the up- and down-going focus
in the subsurface, are key to the method. The goal of the Ma
and down-going parts of the focusing functions from the ref
so-called Marchenko equations. This set of equations can b Wapenaar
et al., 2014; Thorbecke et al., 2017), or a direct method (van der Neut et al., 2014; Ravasi, 2017)
The Marchenko method has found many different applications,
monitoring (van IJsseldijk et al., 2023), adaptive subtraction of Marchenko est Staring
et al., 2018), homogeneous Green's function Brackenhoff et al., 2019), and direct multip
elimination on Zhang and Slob, 2020b). In this paper, a particularly e
of the Marchenko method for imaging by plane-waves is highl
this method are discussed in more detail.
Meles et al. (2018) show that besides focal-points, focal-planes ca
equations. Meles et al. (2020) build on the Marchenko Multiple Elimination
Zhang and Slob (2019) and introduce the plane-wave MME method. The ma
wave-based Marchenko method is that with a minimal effort of
wave), for each depth level (or time instant for MME), a multi
3D applications, the plane-wave Marchenko method is comput
multiple- Brackenhoff et al., 2022). A single plane-wave can be sufficient
image if the subsurface interfaces are near-flat. Multiple p
are needed for subsurface interfaces with varying dips, or g
the subsurface Almobarak, 2021). The Marchenko algorithm for the fo
similar to the focal-point algorithm. The initial point-fo
replaced by a time-reversed direct plane-wave response. T
time windows to separate the Green's functions from the foc
to hold for a plane wave are the same as for a point source; a
the direct and later arrivals. In this paper, we discuss in d

the Marchenko plane-wave method, discuss the implementatio
applications on numerically modeled and field data.
The software accompanied by this paper contains scripts and
examples presented in this paper. The code can also be foun Thorbecke et al. (2017; Thorbecke and Brookes, 2019) where the most recent updated vers
developments are available. To reproduce the figures and per
Seismic Unix (Cohen and Stockwell, 2016) is required.
Most pictures in this section of the manual can be reproduced OpenSource

The Marchenko method is introduced by two coupled equations
and downgoing focusing functions and Green's functions) we
us to suppress internal multiples by using the up- and downg
reflection data from the acquisition surface $\mathbb{D}_0$ to a focal level in the subsurfa
step, the internal multiples of the overburden are suppress
acquisition boundary $\mathbb{D}_0$. The reflection response $R(\mathbf{x}_R, \mathbf{x}_S, t)$ is a scaled version of the Gree
without the direct wave (Wapenaar et al., 2012) measured with sources and rece
$\mathbf{x}_S$ and $\mathbf{x}_R$ on this boundary. The recorded time $t$. This reflection response
contain free-surface related multiple reflections neither a
required to remove the free-surface multiples and the direc
the measured reflection data.
The up- and downgoing parts $f_1^-$ and $f_1^+$ are focusing functions related
the decomposed Green's functions $G^{-,+}$ and $G^{-,-}$ in the actual medium and with the re
at the surface (Wapenaar et al., 2014). The focusing functions have a focal point $\mathbf{x}_A$
This focal point serves as a virtual source for the Green's f
of the decomposed Green's function refer to the direction of
the virtual source $\mathbf{x}_A$. The left most superscript ($-$) indicates upgoing field
at the receiver locations. The relation between the two unkn
Green's functions is given by Wapenaar et al. (2014) in the two equations (

$$G^{-,+}(\mathbf{x}_R, \mathbf{x}_A, t) + f_1^-(\mathbf{x}_R, \mathbf{x}_A, t) = \int_{\mathbb{D}_0} \int_{t'=0}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t') f_1^+(\mathbf{x}_S, \mathbf{x}_A, t - t') \mathrm{d}t' \mathrm{d}\mathbf{x}_S, \quad (59)$$

$$G^{-,-}(\mathbf{x}_R, \mathbf{x}_A, t) + f_1^+(\mathbf{x}_R, \mathbf{x}_A, -t) = \int_{\mathbb{D}_0} \int_{t'=0}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t') f_1^-(\mathbf{x}_S, \mathbf{x}_A, t' - t) \mathrm{d}t' \mathrm{d}\mathbf{x}_S. \quad (60)$$

In the compact operator notation of Van der Neut et al. (2016) equations 59 and 60 are written as

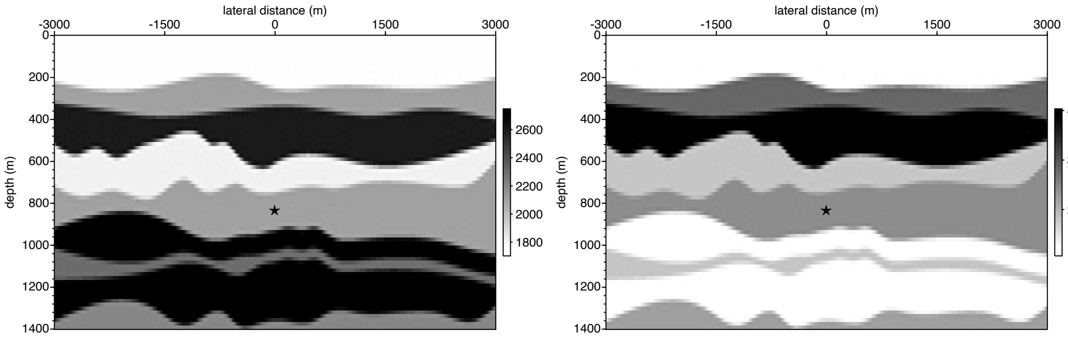
$$G^{-,+} + f_1^- = R f_1^+, \quad (3)$$

$$G^{-,-} + f_1^{+\star} = R f_1^{-\star}, \quad (4)$$

where $\star$ denotes the time-reverse. These two equations contain
and two focusing functions. The only known in these equations is $R$.
Wapenaar et al. (2013) use the reasoning that the Green's function an
certain circumstances, be separated in time (Wapenaar et al., 2014). Therefore, a t
$\Theta(t)$ is defined that passes the focusing function and removes t
side of equations 3 and 4. For point-sources that radiate in all directi
for both the up- and downgoing traveling waves. Up- and down
propagate at opposite dipping angles; hence, two time windo
These time windows remove all events that arrive at later ti
virtual source $\mathbf{x}_A$ to the receiver positions $\mathbf{x}_R$ at $\mathbb{D}_0$, including the direct wave
results in the following two equations that only have two un
known)

$$f_1^- = \Theta_b R f_1^+, \quad (5)$$

$$f_1^{+\star} - f_{1,d}^{+\star} = \Theta_a R f_1^{-\star}, \quad (6)$$




a) velocity model in $m/s$ b) density model in $kg/m^3$

Figure 2.26: Multi-layer model with velocity (a) and density (b) pa
point-source is marked with a $\star$.

where $f_1^+ = f_{1,m}^+ + f_{1,d}^+$, with $f_{1,d}^+$ the direct arrival and $f_{1,m}^+$ all events that arrive before the direct arrival $t_d$. The separation of $f_{1,d}^+$ and $f_{1,m}^+$ can be successfully applied whe overlapping reflection events with the direct response. In o requires additional steps (Zhang et al., 2019). The time windows (Wapenaar et al., 2021) are defined as

$$\Theta_b(t) = \theta(t_b - t), \quad (7)$$

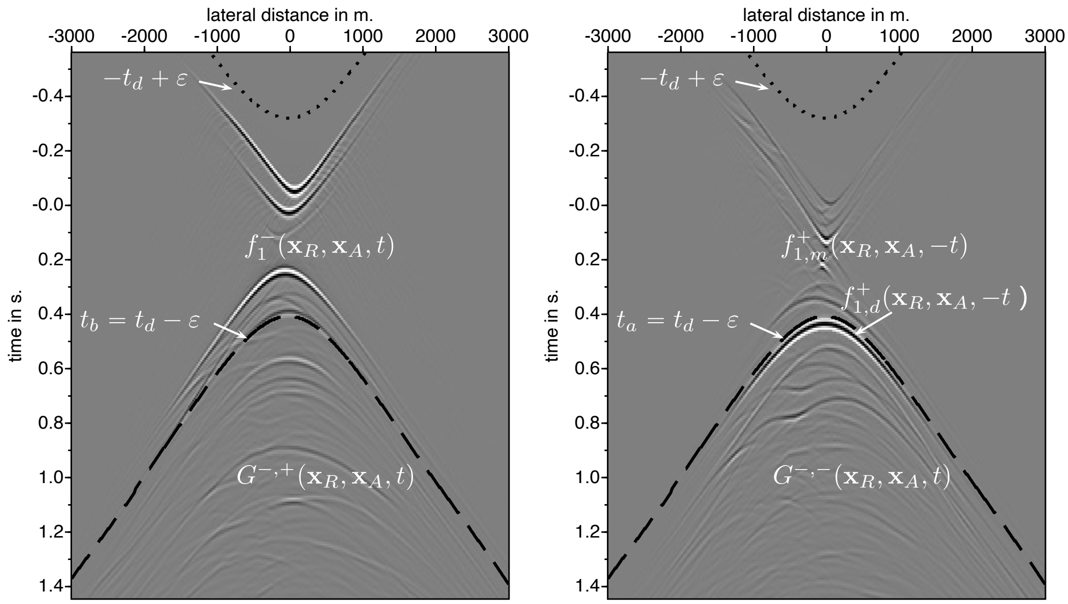
$$\Theta_a(t) = \theta(t_a - t), \quad (8)$$

where $\theta(t)$ denotes a tapered Heaviside step function. Note that t time $t_a$ and onset $t_b$. In the point-source case $t_b = t_d - \varepsilon = t_a$, which makes equation 8 equal to equation 7, with the window function $\theta(t_d - \varepsilon - t)$. The $\varepsilon$ takes into account the finite l band-limited wavelet and ensures that the direct wave is rem (Broggini et al., 2014). Epsilon is typically chosen as half the dominar To illustrate the application of the time windows, a virtua in the laterally varying model of Figure 2.26 (Meles et al., 2018). Figure 2.27 shows the focusing functions and Green's functions for functions (indicated with a dashed line). Figure 2.27a represents the left-hand side of equation 3 and the time window that separates $f_1^-$ from $G^{-,+}$. In Figure 2.27b, the time window separates $f_{1,m}^+$ from $G^{-,-}$ and represents the left-hand side of equation 4. The convolution/correlation in the right-hand sides of equations 3 and 4 frequency domain, and a discrete Fourier transform in time data into the frequency domain. The discrete Fourier transf a periodicity equal to the number of time samples ($n_t$). Given this periodicity in time occurring in time wrap-around into negative times. The time windows in equations 7 and 8 pass all events earlier than $t = t_a$ and will also pass these time wrap-arou these wrap-around events a time window is also implemented. In Figure 2.27, we see that the focusing functions also include events at negat earlier than $-t_a + \varepsilon$. Hence, the cutoff point of the time window at n $-t_a$. The implemented time windows become

$$\Theta'_b(t) = \theta(t_b - t) - \theta(-t_b - t), \quad (9)$$

$$\Theta'_a(t) = \theta(t_a - t) - \theta(-t_a - t), \quad (10)$$

and the time windows at negative times, to suppress time wra lines in Figure 2.27. There is no guarantee that this time window suppr windows are not sufficient to suppress the wrap-around, zeros To solve the unknown focusing functions in equations 5 and 6 the method described by The iterative method described by Behura et al. (2014) and Wapenaar et al. (2014) starts with $f_{1,d}^+$ as the initial $f_1^+$ and solves equation 5 for $f_1^-$ and substitute the output $f_1^-$ in equation 6 This process is repeated until the updates to the focusing f




a) Separation of $G^{-,+}$ and $f_1^-$ b) Separation of $G^{-,-}$ and $f_1^{+*}$

Figure 27: Illustration of the time window function to separate t
function. The dashed black lines indicate the separation li
white arrows. The dotted line indicates the time window tha

algorithm to solve the Marchenko equations is shown in Figure 28. The
iteration count at 0 for the initial solution 5 and in the scheme
equation 6. Thorbecke et al. (2017) explain in more detail the implementation
Depending on the application it is not always needed to sol
Staring et al. (2018) the results of the first iteration are used to predi
subtraction method is used to suppress the predicted multip
to solve the coupled equations Van der Neut et al. (2015) and Ravasi (2017)
Meles et al. (2018) show that plane-wave focusing functions $\tilde{f}_1^{\pm}$ and associated plane-wave
functions $\tilde{G}^{-,+}$ and $\tilde{G}^{-,-}$ can be obtained by integrating an appropriate se
tions $f_1^+$ and $f_1^-$, each involving the solution of a Marchenko equatio
quantities for focusing functions and Green's functions. Wapenaar et al. T
al., 2021; Brackenhoff et al., 2022) are defined by the following integration

$$\tilde{f}_1^{\pm}(\mathbf{x},\mathbf{p}_A,t)=\int_{\mathbb{D}_A} f_1^{\pm}(\mathbf{x},\mathbf{x}_A,t-\mathbf{p}\cdot\mathbf{x}_{H,A})d\mathbf{x}_A, \qquad (11)$$

with $\mathbf{p}=(p_1,p_2)$ and $p_1$ and $p_2$ horizontal ray parameters and $\mathbf{p}_A=(\mathbf{p},x_{3,A})$ the ray parameter o
the plane-wave. The surface $\mathbb{D}_A$ is the depth level at which focusing
plane-wave Green's functions are defined by a similar integr $\mathbf{x}_{H,A}=(x_{1,A}-x_{1,c},x_{2,A}-x_{2,c})$
and $(x_{1,c},x_{2,c})$ is the rotation point of the plane-wave. The rotatio
lateral extent of the plane-wave. This $t=0$ of the plane-wave
By making this choice, the $t=0$ time-axis in the computed plane-wave Gr
is the same as in the point-source Marchenko solutions with
plane-wave.
Note that the plane-wave integration 11 for a time-reversed wavefield $P(\mathbf{x},\mathbf{x}_A,-t)$ gives

$$\tilde{P}(\mathbf{x},\mathbf{p}'_A,-t)=\int_{\mathbb{D}_A} P(\mathbf{x},\mathbf{x}_A,-(t-\mathbf{p}\cdot\mathbf{x}_{H,A}))d\mathbf{x}_A, \qquad (12)$$

with $\mathbf{p}'_A=(-\mathbf{p},x_{3,A})$, a plane-wave dipping opposite to that of the plane-wave of the surface

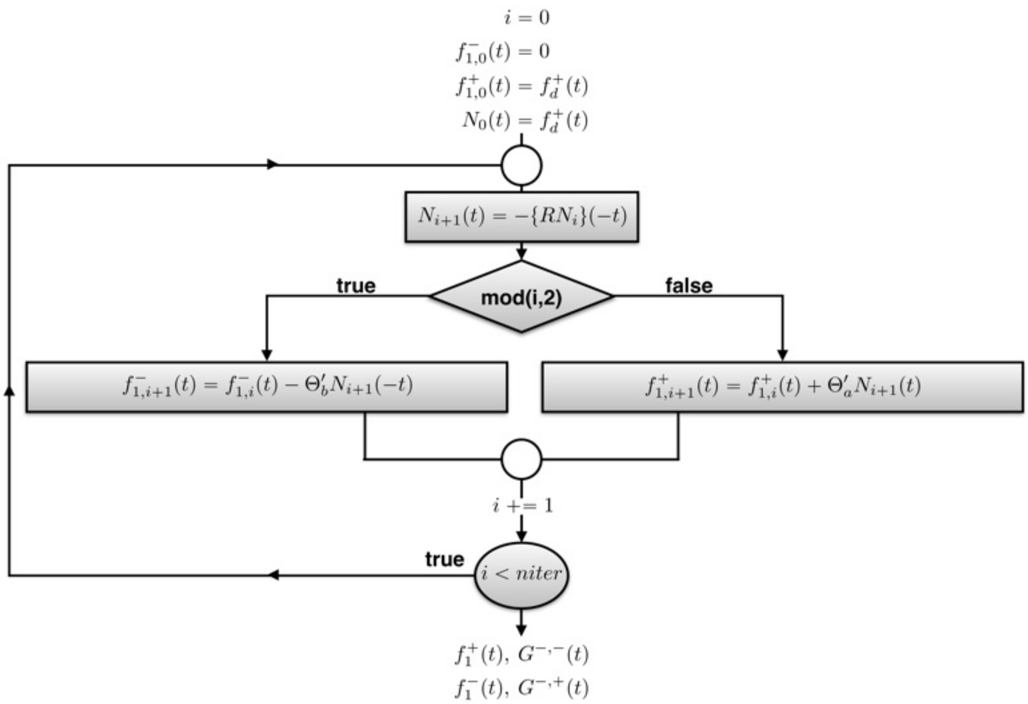



Figure 1.28: Flow chart of the Marchenko algorithm. The up- and downgoing focusing functions ($f_1^-$ and $f_1^+$) are alternately updated. The scheme is finished after a pre- chosen between 10-20 iterations.

$\mathbb{D}_A$ with a homogeneous velocity $c_{H,A}$ at the surface, the time shifts are linear to the distance from the rotation point.
Applying the same integral relations as in the plane- of equations 13 and 14 (Meles et al., 2018).

$$\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t) + \tilde{f}_1^-(\mathbf{x}_R, \mathbf{p}_A, t) = \{R\tilde{f}_1^+\}(\mathbf{x}_R, \mathbf{p}_A, t), \tag{13}$$

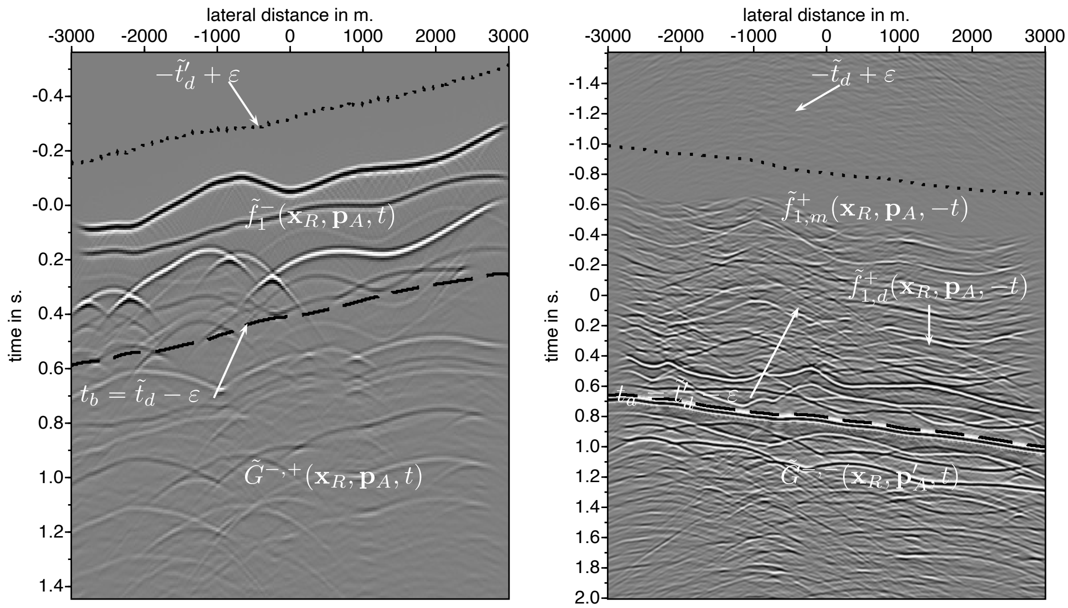
$$\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}'_A, t) + \tilde{f}_1^{+\star}(\mathbf{x}_R, \mathbf{p}_A, t) = \{R\tilde{f}_1^{-\star}\}(\mathbf{x}_R, \mathbf{p}_A, t). \tag{14}$$

Applying a time window that separates the Green function fr equations with two unknowns $\tilde{f}_{1,d}^{+\star}(\mathbf{x}_R, \mathbf{p}_A, t)$ assuming

$$\tilde{f}_1^-(\mathbf{x}_R, \mathbf{p}_A, t) = \tilde{\Theta}_b\{R\tilde{f}_1^+\}(\mathbf{x}_R, \mathbf{p}_A, t), \tag{15}$$

$$\tilde{f}_1^{+\star}(\mathbf{x}_R, \mathbf{p}_A, t) - \tilde{f}_{1,d}^{+\star}(\mathbf{x}_R, \mathbf{p}_A, t) = \tilde{\Theta}_a\{R\tilde{f}_1^{-\star}\}(\mathbf{x}_R, \mathbf{p}_A, t) \tag{16}$$

with $\tilde{f}_1^{+\star} = \tilde{f}_{1,m}^{+\star} + \tilde{f}_{1,d}^{+\star}$, where $\tilde{f}_{1,d}^{+\star}$ is the direct arrival of the plane-wave with by $(\mathbf{p}, x_{3,A})$, and $\tilde{f}_{1,m}^{+\star}$ contains the events that arrive after $\tilde{t}_d$, where $\tilde{t}_d$ is the direc first arrival time of a plane-wave with $(\mathbf{p}, x_{3,A})$. Note that the Green functions in 13 and 14 contain plane-waves with opposite dipping $\mathbf{p}_A$ and $\mathbf{p}'_A$. To separate $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ from $\tilde{f}_1^-(\mathbf{x}_R, \mathbf{p}_A, t)$ the direct arrival $\tilde{t}_d$, as the first arrival of $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$, is required. For $\tilde{t}_d'$ in this case, our choice is the sa Wapenaar et al. (2021). This choice uses the same $\tilde{t}_d$ as the $\tilde{t}_d'$ on the propagation angle $\mathbf{p}_A$ the upgoing Green's functions $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ and $\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}'_A, t)$. Note that the defined arri $\tilde{t}_d'$ has the same arrival time of an upward propagating $(\mathbf{p}, x_{3,A})$ with mirrored angle $\mathbf{p}'_A$.
The time window $\tilde{\Theta}_b(t)$ removes $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ from equation 13. The first non-zero contri $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ is at time $t_b = \tilde{t}_d - \epsilon$. The time window $\tilde{\Theta}_a(t)$ removes $\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}'_A, t)$ and $\tilde{f}_{1,d}^{+\star}(\mathbf{x}_R, \mathbf{p}_A, t)$ from equation 14, hence all events at times $t_a = \tilde{t}_d' - \epsilon$ are set to zero. Similar to the




a) Separation of $\tilde{G}^{-,+}(\mathbf{p}_A)$ and $\tilde{f}_1^-(\mathbf{p}_A)$.

b) Separation of $\tilde{G}^{-,-}(\mathbf{p}'_A)$ and $\tilde{f}_1^{+*}(\mathbf{p}_A)$.

Figure 29: Illustration of the time window function to separate t
focusing function. The dashed black lines indicate the separ
with white arrows. The dotted line indicates the time window

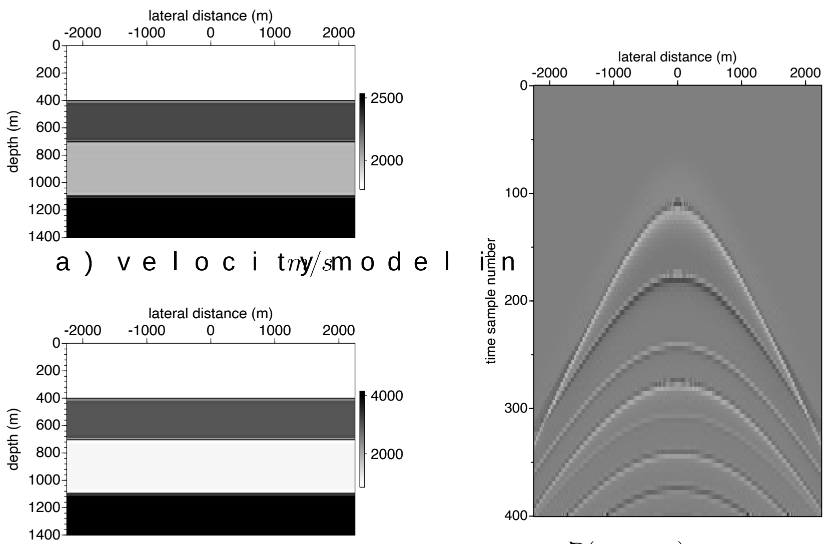
scheme these time windows are implemented with an additiona
time wrap-around and are given by

$$\tilde{\Theta}'_b(t) = \theta(\tilde{t}_d - \varepsilon - t) - \theta(-\tilde{t}'_d + \varepsilon - t), \quad (17)$$

$$\tilde{\Theta}'_a(t) = \theta(\tilde{t}'_d - \varepsilon - t) - \theta(-\tilde{t}_d + \varepsilon - t). \quad (18)$$

Similar to Figure 27, Figure 29 shows the plane-wave focusing functions and
left-hand side of equations 13 and 14, and the window functions separating the
depth of the plane-waves is chosen at 800 m. In the following
these two time window functions are discussed in more detai
implementation of the Marchenko algorithm are explained.

The plane-wave method is illustrated with two numerical ex
variant medium. In two dimensions the downward propagating $\tilde{f}_1^+(\mathbf{x}, \mathbf{p}_A, t)$
in equation 11 defines a plane-wave with a dip angle $\alpha$ and $p_1 = \frac{\sin(\alpha)}{c}$. In the first
numerical example we assume a medium with a laterally invariant ... as shown in
Figure 30a and b. Figure 31 shows $\tilde{f}_1^+(\mathbf{x}_R, \mathbf{p}_A, t)$, for a focal plane at a depth of 900 m
angles: Figure 31a with an angle of 0 degrees and Figure 31c with an angle of 3 degrees.
31b and 31d the focus function is shown for receivers at the foca
focal-point Marchenko method, the medium for the plane-wave
below the focal level. The focus function $\tilde{f}_1^+(\mathbf{x}, \mathbf{p}_A, t)$ has a focus at time $t = \mathbf{p} \cdot \mathbf{x}_{H,A}$. For a
dipping plane-wave this time focus occurs ... Figure 31a and c show ...
the computed $\tilde{f}_1^+(\mathbf{x}_R, \mathbf{p}_A, t)$ at the surface (0 m depth) and includes an extra
that compensates for the multiples generated between the fr
respectively. The compensation of multiples ... Zhang and Slob
(2019). The focus functions in Figure 31a and 31c also show ... event, the do
present at the surface is compensated by the refected event

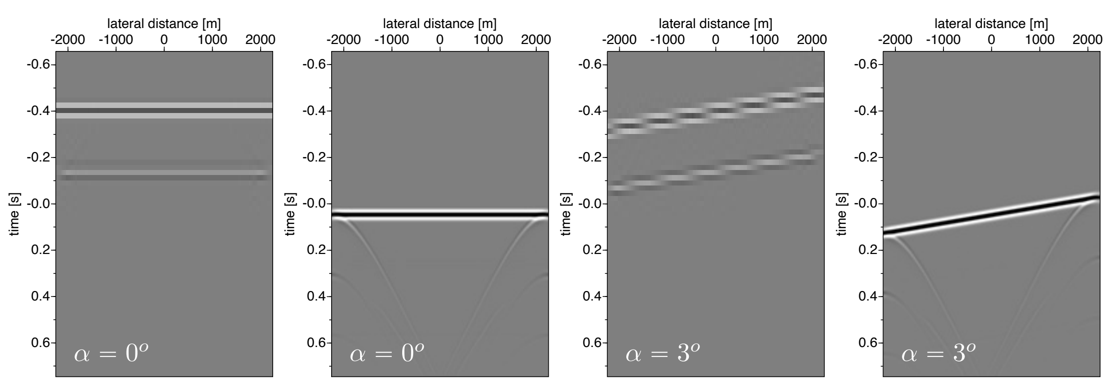



a) velocity model in $m/s$

b) density model in $kg/m^3$

c) $R(\mathbf{x}_R, \mathbf{x}, t)$

Figure 30: Two dimensional four layer model with velocity (a) and
source record, with source position $\mathbf{x} = (x_1 = 0, x_3 = 0)$ and receivers at $\mathbf{x}_R = (x_1 = x_R, x_3 = 0)$ (c).
The source wavelet $R$ has a flat frequency spectrum from 5 to 90 Hz.



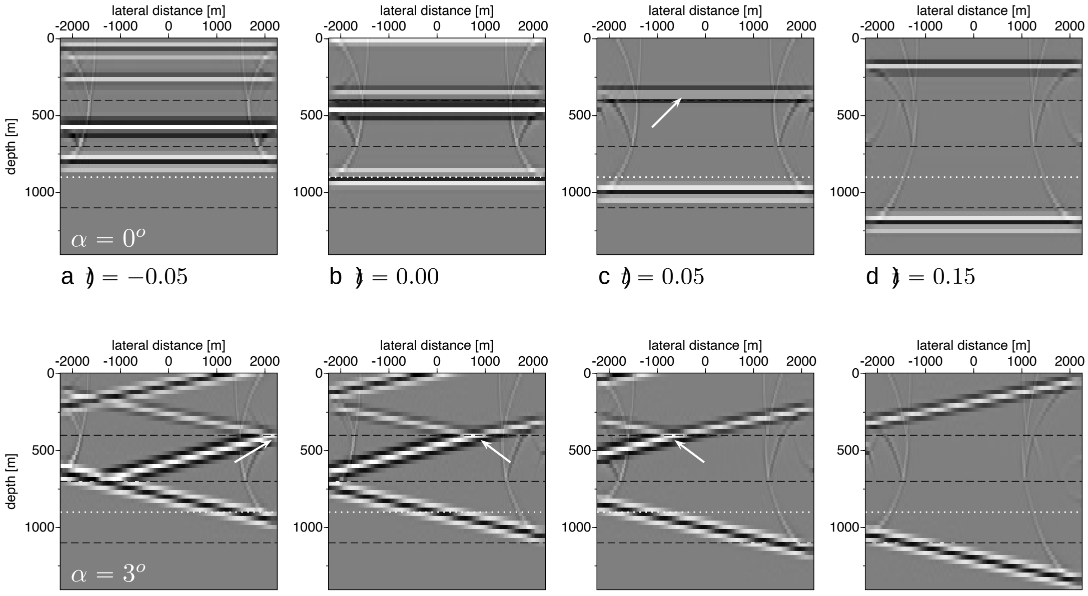

a) depth $x_3 = 0$ m

b) depth $x_3 = 900$ m

c) depth $x_3 = 0$ m

d) depth $x_3 = 900$ m

Figure 31: Time recordings of the plane wave $\tilde{f}_1^+(\mathbf{x}_R, p_A, t)$ focusing function of
meter measured with $x_3 = 0$ and $x_3 = 900$ meter at in the truncated medium for
plane-wave propagation angles (0 and 3 degrees). At the end-
are present due to the limited lateral extent of the constru




a) $t = -0.05$ b) $t = 0.00$ c) $t = 0.05$ d) $t = 0.15$

e) $t = -0.05$ f) $t = 0.00$ g) $t = 0.05$ h) $t = 0.15$

Figure 3.2: The snapshots of $\tilde{f}_1^+(\mathbf{x}_R, \mathbf{p}_A, t) + \tilde{f}_1^-(\mathbf{x}_R, \mathbf{p}_A, t)$ at two different plane-propagation angles. Note the diffraction effects at the edge indicates the focal depth of the plane-wave.

To illustrate this compensation effect, snapshots of $\tilde{f}_1^+(\mathbf{x}, \mathbf{p}_A, t)$ in a Figures 3.1a to 3.1f the focu propagating into the truncated medium (that is homogeneous are shown in Figure 3.2 for the same angles of 0 and 3 degrees. Figures 3.2a to 3.2d show four differen snapshots of the superposition of the downgoing $\tilde{f}_1^+(\mathbf{x}_R, \mathbf{p}_A, t)$ and upgoing plane-wave $\tilde{f}_1^-(\mathbf{x}_R, \mathbf{p}_A, t)$. The snapshots of a plane-wave with an angle of 3 d to 3.2h. At 0.05 seconds before $t = 0$ (Figure 3.2a and 3.2e) the two upward travelin from the interfaces at 400 and 700 meter depth in $\tilde{f}_1^-(\mathbf{x}_R, \mathbf{p}_A, t)$. The two downg snapshots in Figure 3.2 ($t = 0.00$ s) and 3.2 ($t = 0.05$ s) show that the second dow traveling event coincides at the first interface (at 400 m dep interface (at 700 m depth) and these events compensate each the pictures. The fourth snapshot show that after this comp the reflectors at 400 and 700 m depth have vanished and only on reflected wavefield (from the reflector at 700 m depth) are re upward traveling multiples in $\tilde{f}_1^+(\mathbf{x}_R, \mathbf{p}_A, t)$ is a direct illustration of the Marchenko eq illustration in Figure 3.2 demonstrates that the internal multiple compen the plane-wave Marchenko method.

In Figure 3.3 the experiment is repeated in the 2.6 The focal-plane chosen at 800 m depth, just below the fourth reflector. The s events in $\tilde{f}_1^+(\mathbf{x}_R, \mathbf{p}_A, t)$ compensate the upgoing events at interfaces an internal multiples.

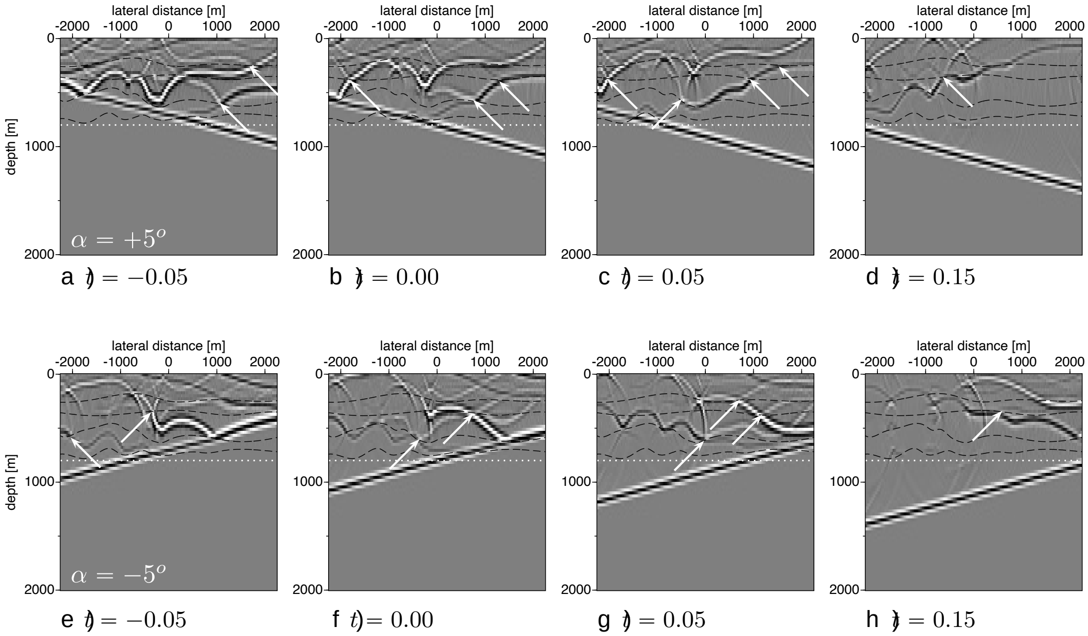
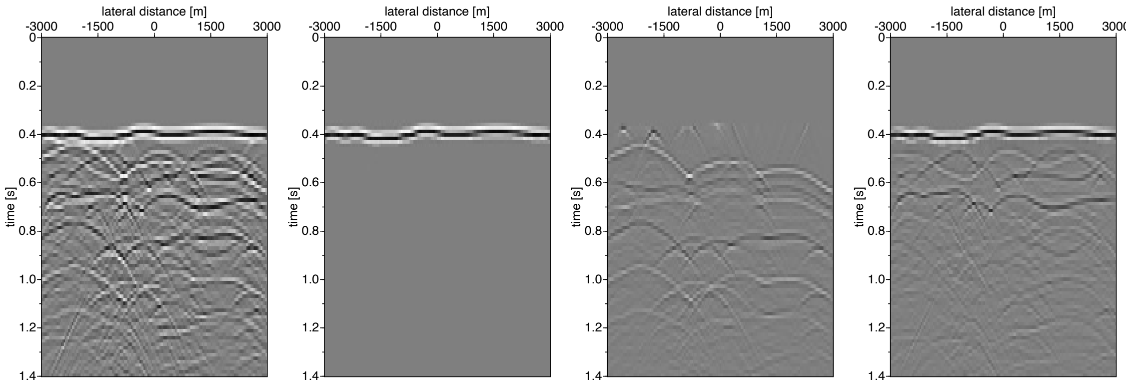



Figure 3.3: The snapshots of $\tilde{f}_1^+(x_R, p_A, t) + \tilde{f}_1^-(x_R, p_A, t)$ in model of Figure 2.6 for two different plane-wave propagation angles. Note that there ... from edges on the interfaces. The white dotted-line indica... arrows indicate positions at a reflector where an up-going re... is compensated by a down-going event from the focusing funct...

### 5.3.1 Horizontal plane-waves

To start the iterative Marchenko algorithm $\tilde{f}_{1,d}^+(x_R, p_A, t)$ is required; the first ar... a plane-wave response from a focal-plane in the subsurface. ... in a macro model estimated from the reflection data. The compu... the first arrival times to get the time-reverse ... $\tilde{f}_{1,d}^+(x_R, p_A, t)$. As a first example it i... is made for a horizontal (zero-degree) plane-wave defined a... model of Figure 2.6 (same model as in Meles et al., 2018).

Figure 3.4a shows the forward modeled plane-wave response of a hor... receivers at the surface. The first arrival of Figure 3.4a, is the time-reversal ... of the input $\tilde{f}_{1,d}^+(x_R, p_A, t)$ of the Marchenko scheme. The computed down-... functions, after 16 iterations of equations 15 and 16 ... the results in equations 13 and 14, are shown in Figure 3.4b and 3.4c respectively and give the expected response... Marchenko mute-line $\tilde{\Theta}_{a,b}$ (equations 17 and 18) is used to separate the Green's fu... the focusing functions. ... $t = 0$ ... is symmetric $\tilde{\Theta}_a = \tilde{\Theta}_a$. This is the same time symme... use for the Marchenko point-source algorithm.

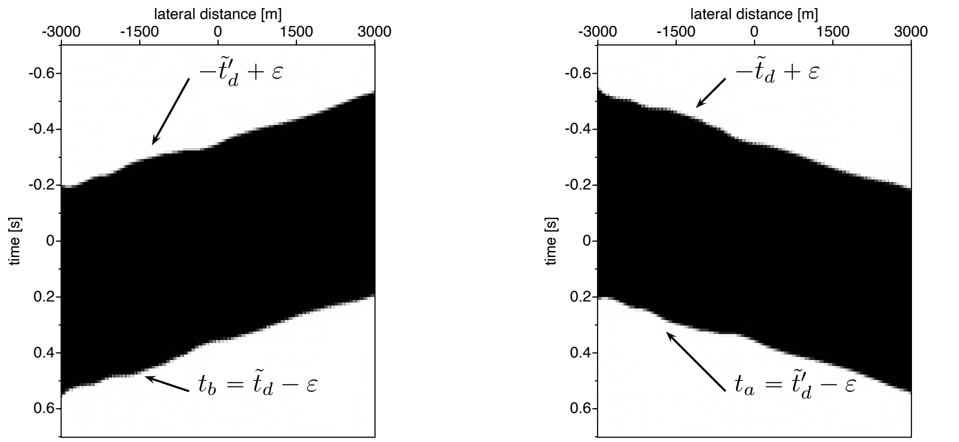



a) $\tilde{G}(\mathbf{x}_R, \mathbf{p}_A, t)$ b) $\tilde{f}^{+\star}_{1,d}(\mathbf{x}_R, \mathbf{p}_A, t)$ c) $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ d) $\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}'_A, t)$

Figure 34: Results of the plane-wave Marchenko scheme for a hori $\mathbf{p}_A = (0, x_{3,A})$. Adding the up- and downgoing Green's functions of c and d, t algorithm, gives the same wavefeld as the directly forward with the same clipping factor.

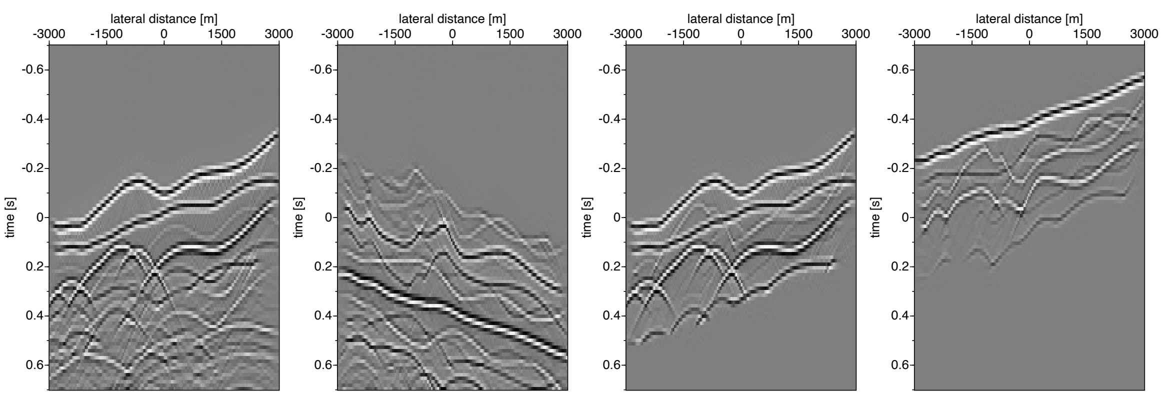
### 5.3.2 Dipping plane-waves

The Marchenko algorithm for dipping plane-waves follows th waves. As indicated by equations 17 and 18 the Marchenko time windows have to be d for dipping plane-waves. For horizontal plane-waves the im $t = 0$ and $\tilde{\Theta}'_a = \tilde{\Theta}'_b$. This does not hold anymore for dipping plane-wa Figure 35 is an example of time windows that are designed for a dipping plane-wave of +5 one for the even iterations (Figure 35a), and one for the odd iteration reverse angle (Figure 35b). An update of these windows for a pos first arrival time for a plane-wave at the same depth level wi In the following we explain in more detail what is expressed by equations 15 and 16. The plane-wave Marchenko scheme starts with forward m plane-wave at depth. This modeled wavefield is depicted in Figure 37a, where the depth of the plane-wave is chosen at 800 m. The direct arrival is selected 37b. The time reverse of this field, $\tilde{f}^{+}_{1,d}(\mathbf{x}_R, \mathbf{p}_A, t)$. The field $\tilde{f}^{+}_{1,d}(\mathbf{x}_R, \mathbf{p}_A, -t)$ is, together with the refection data $R$, input of the plane-wave Marchenko scheme.




a) even iterations b) odd iterations

Figure 3.5: The time windows for dipping plane-waves for even (a) a
+5 degrees. The wavefelds in the black area of the windows pa
zero.




a) frst iteration b) second iteration c) $\tilde{f}_1^-(\mathbf{x}_R, \mathbf{p}_A, t)$ d) $\tilde{f}_1^+(\mathbf{x}_R, \mathbf{p}_A, t)$

Figure 3.6: Basic plane-wave Marchenko results for a plane-wave wi
Note, that the results of the frst iteration (a) is dipping i
(b) and the algorithm uses the time windows, designed for di 17
and 18, to take this into account.

In the Marchenko algorithm the iterations are alternating be

with $R$, or a correlation with $R$ (Thorbecke et al., 2017). In the first step, correlation (16), the correlation b

wavefield is shifted backward in time, $\tilde{t}_d$, and in the second step, convolution

equation 15, the wavefield is shifted forward in time $\tilde{t}'_d$. In Figure 36 the result of the

the first iteration (correlation) is shown and in Figure 36b the result of the second iteration

is shown. In Figure 36a we can see that the first event, that starts at ne

the undulation of the first reflector and has an opposite dip c

The result of the first iteration in Figure 36a is muted in time (with the window 35) the window

mute $\tilde{G}^{-,+}(\mathbf{x}_R,\mathbf{p}_A,t)$, followed by time-reversal and convolution with $R$ (equation 16). The result of this is

shown in Figure 36c. In this second iteration the convolution step b

times corresponding to reflection times in the Figure 37a. Note that the model

that the arrival in Figure 36c, just inset from the left at time 0.2 s dipp

is the same as the first arrival $\tilde{t}_d$ in Figure 37a at Figure 37b. The result of this second

muted in time (with the window 35) into $\tilde{G}^{-}(\mathbf{x}_R,\mathbf{p}'_A,t)$ and the first arrival e

a $\tilde{f}^+_{1,d}(\mathbf{x}_R,\mathbf{p}_A,t)$, as indicated in 20. The odd iteration $\tilde{f}^-_1(\mathbf{x}_R,\mathbf{p}_A,t)$ are building u

$\tilde{G}^{-,+}(\mathbf{x}_R,\mathbf{p}_A,t)$ and the even iterations $\tilde{f}^+_1(\mathbf{x}_R,\mathbf{p}_A,t)$ are building up $\tilde{G}^{-,-}(\mathbf{x}_R,\mathbf{p}'_A,t)$. Figure 36e

and 36f show $\tilde{f}^+_1(\mathbf{x}_R,\mathbf{p}_A,t)$ and $\tilde{f}^-_1(\mathbf{x}_R,\mathbf{p}_A,t)$ respectively after 16 iterations.

In Figure 37 three plane-wave responses are shown with angles of -

Comparing these three plane-wave responses $\tilde{G}^{-,+}$ shows that each angle illum

ferent parts of the medium. This is clearly seen in the even

combining different plane-wave responses into one image a fu

using only a few plane-waves (Meles et al., 2018). In plane-wave imaging, that suppres

can use the same strategies as point-source Marchenko, for e

der Neut (2015), Staring et al. (2018), or Multi Dimensional Deconvolution (Ravasi

et al., 2016). Almobarak (2021) discusses different plane-wave imaging me

Marchenko Green's function plane-wave response is a comput

Figure 38 shows horizontal plane-wave images from the Troll field

was kindly provided by Equinor. This dataset is part of a ti

a small part of this data-set, with source and receiver posi

source spacing is 12.5 meter and traces are recorded with a t

pre-processed to remove free-surface multiples (Qu and Verschuur, 2020) and deconvo

The imaging is carried out according to the method described in Meles et al. (2018). In the standard

imaging method, a forward modeled plane-wave response is co

smooth macro model of the data. This plane-wave depth respo

responses of the recorded data and integrated over the recei

This creates the plane-wave response of the data (Rietveld et al., 1992). The plane-wave res

of the data is correlated with the same modeled plane-wave re

condition $t=0$ is used to construct the image for all depth levels.

the left side picture of Figure 38 labelled 'standard'.

To compute the Marchenko based image, the first arrivals of t

at each depth level is input to the plane-wave Marchenko algo. The computed $\tilde{G}^{-,+}$

is, similar to the standard imaging method, correlated with

for each depth level, and the imaging condition $t=0$ is used to construct the image for all dep

horizontal plane-wave ($\mathbf{p}_A = (p=0, x_{3,A})$) at one depth level $x_{3,A}$ the imaging condition is r

by

$$\tilde{I}(\mathbf{x}_R,\mathbf{p}_A) = \int_t \tilde{f}^+_{1,d}(\mathbf{x}_R,\mathbf{p}_A,t)\tilde{G}^{-,+}(\mathbf{x}_R,\mathbf{p}_A,t)dt. \quad (19)$$

The advantage of using $\tilde{G}^{-,+}$ is that this field does not contain downgoing

layers above the focal/imaging depth. The Marchenko method

and downgoing parts in the Greens functions by applying the

data. Alternative strategies to compute an image without i

Thorbecke et al. (2021) illustrate the working of the internal multip

on reflection data in Figure 12 (source-r... Staring et al. (2018)

Marchenko redatuming and imaging on field data using an adap

of that paper shows internal multiples that are first predict

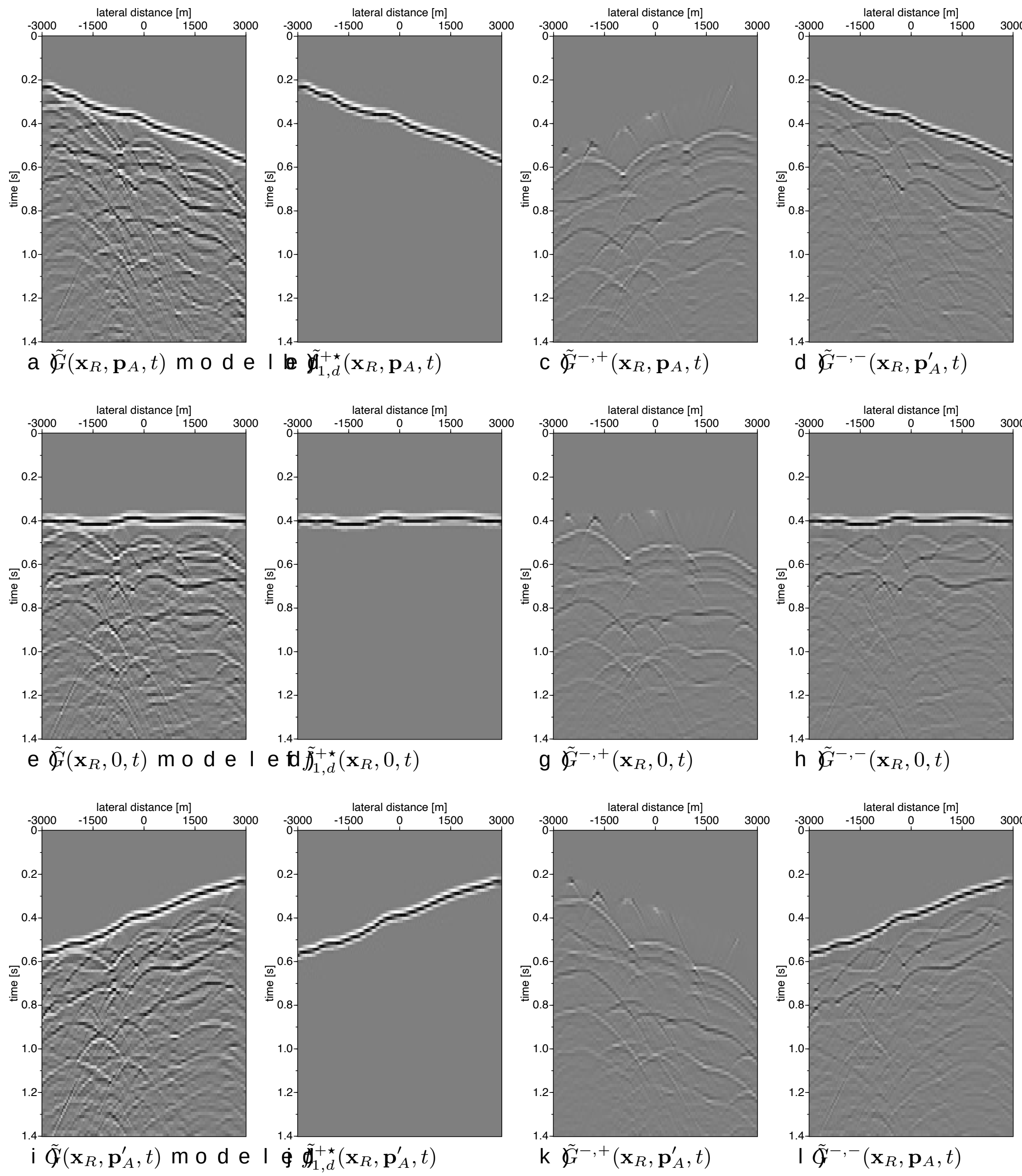
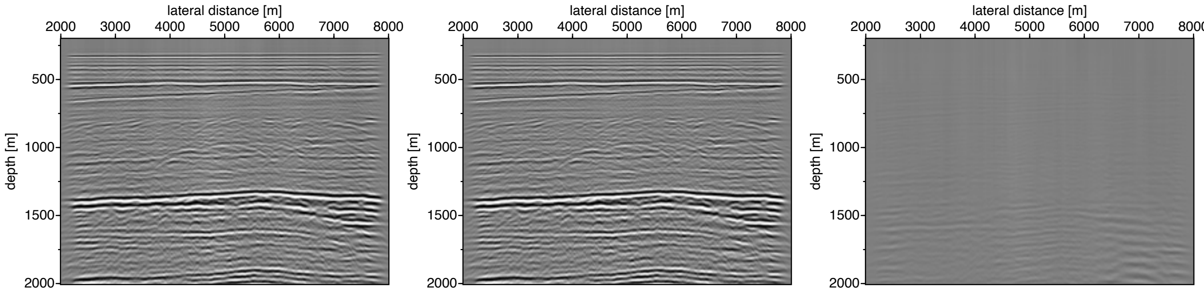



Figure 3.7: Marchenko computed plane-wave responses for angles at
Note the difference in illumination in the decomposed Green's
plane-wave. Analogous to Figure 3.4 the addition of the Marchenko computed up-
Green's function gives the forward modeled response in a.




a) Standard at 0 degrees   b) Marchenko at 0 degrees   c) difference

Figure 38: Plane-wave images of the Troll field data-set for a ho
imaging (left) and Marchenko based imaging (middle). All im
factor.

The middle picture of Figure 38 shows the Marchenko-created image. The differ
image and the Marchenko-based image is shown in Figure 38c. From this difference plot i
that the Marchenko method predicts and removes internal mul
multiple removal on the images van IJsseldijk et al. (2023) showed that Marchenko multip
on this dataset improves the confidence of the effects of smal

The plane-wave Marchenko method is a straightforward extens
A counter-intuitive aspect of the plane-wave method is that
Green's functions have opposite dipping angles. This is ta
separate the Green's function from the focusing function. I
function for a specific dip angle, one would have to run the
angles. In this paper, the use of these time windows is illus
The plane-wave Marchenko method can give a computational ad
Specially for imaging applications with 3-dimensional dat
that case only a few plane-wave migrations are needed to com

The authors thank Equinor (formerly Statoil A.S.) for provi
This research was funded by the European Research Council (E
2020 research and innovation program (grant agreement no. 7

- Name of the code/library: OpenSource code for Finite Dif
  cessing utilities
- Contact: j.w.thorbecke@tudelft.nl
- Hardware requirements: tested on x86_64 and aarch64 proc
- Program language: C and Fortran
- Software required: C compiler, Fortran compiler, GNU Mak
  display and generation of the figures is done with Seismic Un
- Program size: 147 MB

The source codes are available for downloading at https://github.com/
The scripts to reproduce the results are in OpenSource/marchenko
The README in that directory explains all the steps to repro
the reproduction of the measured data example please contac
data if we can share the data.

## A.1 Plane-waves

To model a plane-wave with a tilt Thorbecke and Draganov (2011) finite-difference parameter value p [s/m] is defined by a chosen velocity and a p depth-level. The plane-wave is triggered at all grid-poin depth-level in the finite-difference grid. To simulate a dip defines the dipping plane-wave $t_p$ (eq. 20-22) time-delay depends li distance from the rotation point. The rotation point of the $t=0$, is chosen at the center of the plane-wave $\mathbf{x}_c$.

$$\mathbf{p} = sin(\alpha)/c_p, \tag{20}$$

$$\mathbf{x}_p = (\mathbf{x} - \mathbf{x}_c) * dx, \tag{21}$$

$$t_p(\mathbf{x}) = \mathbf{x}_p * \mathbf{p}. \tag{22}$$

where $\alpha$ is the dip angle, $c_p$ the propagation velocity of the medium, and $\mathbf{x_c} = (x_{1,c}, x_{2,c})$ are the horizontal coordinates of the central location of the plane in the definition of $\mathbf{x}_p$, we ensure that the center of the plane-wave sour of $t=0$ Brackenhoff et al. (2022).

In a homogenous model a plane-wave, modeled with these time a plane-wave on a slant line (src_angle=$\alpha$) in a medium with velocity (src_velo=$c_p$). In Figure 30 to 30 four snapshots are shown of a propagating plane-w and a propagation velocity of 1500 m/s. The snapshots of the x-position -3000 and 0, have a negative time-delay. The $x=0$ is at $t=0$, the shot shot that is initiated first (at negative time) is positioned -3000 m in Figure 30 (a) and has a time-delay of -0.1742 s. To account for the modeling of dipping plane-waves, the Thorbecke and Draganov (2011) finite-difference mo is adapted. To model a negative angle the shot that is initi sample in Figure (most right position) 30 with +3000 m in Figure 30 de -0.1742 s.

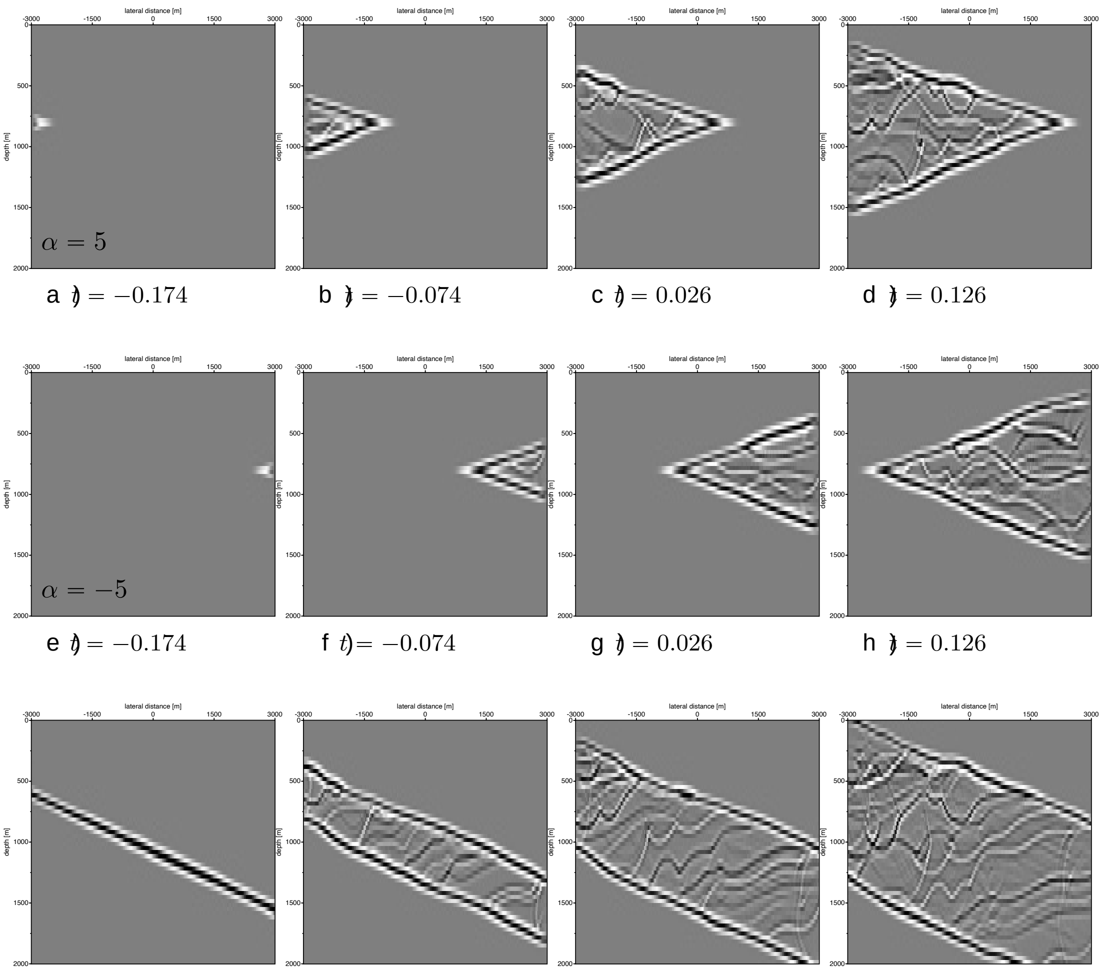
An alternative implementation of a dipping plane-wave is ac grid-point that also varies in depth. All sources are active at $t=0$ any time-delays between sources. Figures 30 to 30 show the snapshots of this impl disadvantage of this implementation is that in solving the a defined angle $\alpha$ and depth, the ray-parameter of $\tilde{G}^{-}(\mathbf{p}, A)$ is not related to $\tilde{G}^{-,+}(-\mathbf{p}, A)$, more to since $\mathbf{p}$ is position dependent.

## A.2 Time-shifts in Marchenko equations

In the use of tilted plane-waves time-shifted sources play understand the effects of a time-shift in the regular Marche 40 to 40 the standard Marchenko results are shown for a point in Figure 30 at 900 meter depth. In Figure 40 the forward modeled operator is shifted +0 hence the time-reverse of that $f^+_{1,d}$ is shifted -0.3 seconds backward. The Marchenko results in Figure 40 to 40 show that the solution of the Marchen change; the same fields are computed, $f_1^-$, $f_1^+$ and $G^-$ are shifted back in time and $G^+$ is shifted forward in time.

To compute the time-shifted Marchenko results the time-windows in the Marchenko equations, to separate the focal- from the Green's functio that for even and odd iterations different time-windows hav constant-time shift of 0.3 seconds.

The results of the time-shifted Marchenko can of course also to be careful. For example $G^-$ can be back-propagated into the medium to b depth, but before back-propagating $G^{-,+}$ in the medium the time-shifted respo back to the origin, $t=0$ of $f^+_{1,d}$, otherwise $G^{-,+}$ will not focus at the focal depth of $f^+_{1,d}$.




a) $t = -0.174$ b) $t = -0.074$ c) $t = 0.026$ d) $t = 0.126$

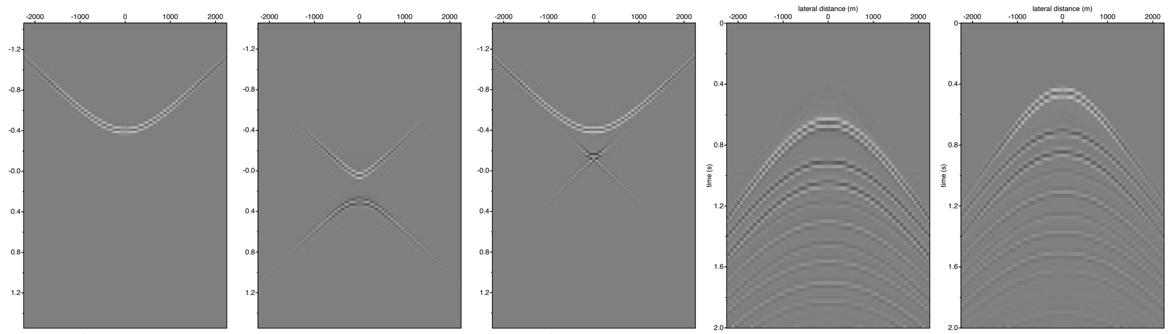
e) $t = -0.174$ f) $t = -0.074$ g) $t = 0.026$ h) $t = 0.126$

i) $t = 0.1$ j) $t = 0.2$ k) $t = 0.3$ l) $t = 0.4$

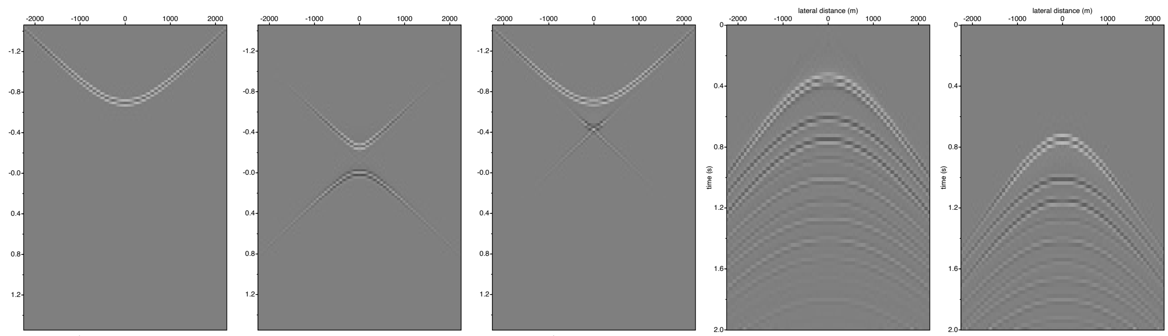
Figure 39: Time snapshots for diferent implementations to model a the time-delayed implementation at time instances -0.174, +5 degrees. Pictures e-h show the time-delayed implementat of -5 degrees. Pictures i to l show a titled grid position im

## A.3 Time wrap-around

The time wrap-around events only occur for deep focusing lev 29 in the paper that are chosen not that deep. To illustrate the wr 29 is shown in 41 with a focal level at 1900 m. depth. 41(a) shows the time wrap-around for times $-\tilde{t}_d + \varepsilon$. After padding with 2x the number of t 1024 to 2048 time samples) the time wrap-around is not visib

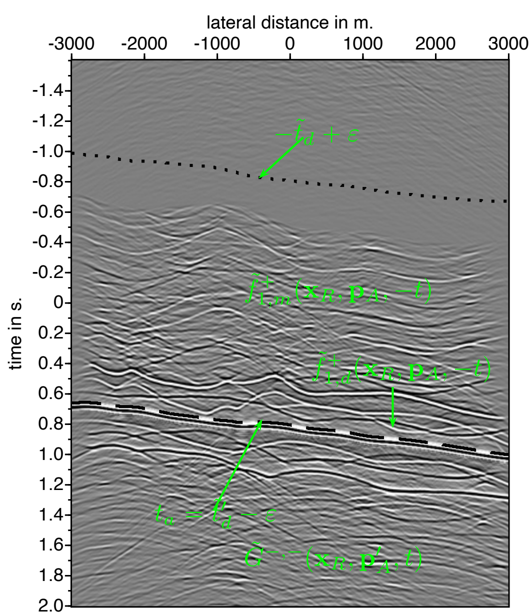


a) $f_{1,d}^+(t)$ b) $f_1^-(t)$ c) $f_1^+(t)$ d) $G^{-,+}(t)$ e) $G^{-,-}(t)$

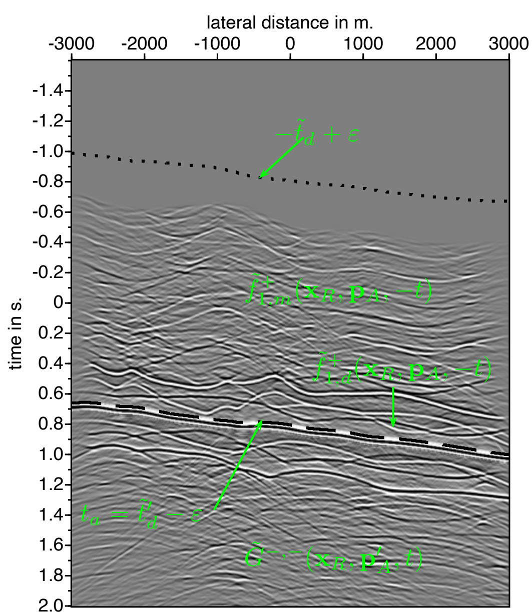


f) $f_{1,d}^+(t)$ g) $f_1^-(t)$ h) $f_1^+(t)$ i) $G^{-,+}$ j) $G^{-,-}(t)$

Figure 40: Standard and time-shifted Marchenko results for a foc
model of Figure 30. The applied shift is +0.3 s. forward $(f_{1,d}^+)^*$ in time on the




a) Separation without time-padding.




b) Separation with time-padding.

Figure 41: Illustration of the time-window function for the plane $\tilde G^{-,-}(\mathbf{p}'_A)$ to
the focusing function $\tilde f_1^{+*}(\mathbf{p}_A)$. The dashed black lines indicate the separa
and are indicated with white arrows. The dotted black line i
time wrap-around.

A.4 Scripts to reproduce the fgures in this chapter.

```
-------------------------
The instructions below explain how to reproduce the resul

"Design, implementation and application of the Marchenko

by Jan Thorbecke, Mohammed Almobarak, Johno van IJsseldij
-------------------------

#Figure 1: model lateral varying created by Giovanni Mele
cd marchenko/demo/planewave/twoD
# this creates the gridded models
./model.scr
# to create the eps figures
./epsModel.scr model

#Figure 2: time-window seperation point-source algorithm
cd marchenko/demo/planewave/twoD
#First generate all 601 shots to model the reflection data
./shots.scr #this script is based on slurm
#model direct field
./direct.scr
#subtract direct field from modeled shots to create Refle
./remove_direct.scr

#Figure 2
# initial modeling for first arrival
./IniPoint.scr
#run the marchenko algorithm for a point in the middle at 8
./sumGandFpoint.scr

#Figure 4 same as figure 2 but now for plane-waves
#first model the first arrivals of the plane-wave respons
./IniFocus.scr
#run plane-wave marchenko with plane-wave of 5 degrees an
sumGandF.scr

#Figure 5 (1D model + middle shot)
cd marchenko/demo/planewave/oneD
./model.scr
./epsModel.scr

#Figure 6 & 7 : plane wave snapshots angle 0 and 3 in 1D med
cd marchenko/demo/planewave/oneD
./initialFocusPlane.scr
./marchenkoPlane.scr
./backpropf1plusPlane.scr
./epsPlane.scr plane
./epsPlane.scr snapshots

#Figure 8 (see figure 1 for generating data)
cd marchenko/demo/planewave/twoD
./model.scr
./IniFocus.scr (for 0 5 and -5)
./marchenkoPlaneIter.scr (for 0 5 and -5)
#back propagating snapshots through Giovanni's Model
```

```
./backpropf1plusPlane.scr
./epsPlane.scr snapshots

#Figure 9
./epsPlane.scr shots0

#Figure 10: use fmute with returnmask=1
cd marchenko/demo/planewave/twoD
./muteW.scr
./epsModel.scr mute

#Figure 11
cd marchenko/demo/planewave/twoD
./epsPlane.scr shots5
./epsPlane.scr iter

#Figure 12
cd marchenko/demo/planewave/twoD
./epsPlane.scr shots5
./epsPlane.scr shots0
./epsPlane.scr shots-5


#Figure 13 Troll data field
Not provided have to ask permission for sharing data to Eq
Data stored at /palmyra/data/jthorbecke/Johno/data

#Imaging bash script
iangle=0
nangle=1
startangle=0
while (( iangle < nangle ))
do
  (( angle = ${startangle} + ${iangle}*${dangle} ))
  echo angle=$angle
  filegm=gmplanes$angle.su
  filefd=fdplanes$angle.su
  invangle=$((angle * -1))

  # data with internal multiples
  syn2d file_syn=fdplanes$angle.su file_shot=$Rdata nsho
  fconv file_in1=fdplanes$invangle.su file_in2=Rfocus$a
  # scale with 5e-5 * dt (0.004) = 2e-7
  sugain < Rimage$angle.su scale=2e-7 > nep.su
  mv nep.su Rimage$angle.su
  suwind < Rimage$angle.su itmin=0 itmax=0 | sustrip > Rim

  # data after Marchenko
  fconv file_in1=fdplanes$invangle.su file_in2=gmplanes$
  suwind < Gimage$angle.su itmin=0 itmax=0 | sustrip > Gim
  (( iangle = $iangle + 1))
done

#generate eps files
for angle in -3 0 3
do
for file in Rimage366x481_$angle.su Gimage366x481_$angl
```

```
do
   file_base=${file%.su}
   supsimage < $file hbox=3 wbox=4 labelsize=12 linewidth
       n1tic=2 x1beg=0 x1end=2005 f1=200 f1num=0 d1=5 d1nu
       label1="depth [m]" label2="lateral distance [m]" \
       f2=2000 f2num=2000 d2num=1000 d2=12.5 clip=1e8 > $f
done
done
```

--------------------------
--------------------------

Description of files:

1) ./model.scr computes the gridded velocity/density mode

2) ./shots.scr computes 601 shots using slurm arrays: ~10

3) ./direct.scr compute 1 shot that contains direct wave o

4) ./remove_direct.scr removes the direct wave from the d

   Figures of the model and the middle shot are generated by
   ./epsModel.scr model

5) ./IniFocus.scr compute plane wave responses at 800 m de

   To compute the plane wave respone with the sources all st

   ./IniFocus.scr

m a r c h e n k o 3 D

MDD

```
.
├── Common_Public_License.txt
├── SU_LEGAL_STATEMENT.txt
├── CODE_OF_CONDUCT.md
├── README.md
├── INSTALL ................ Short instructions to install t
├── REPRODUCE .... Summary how to reproduce the examples in t
├── Make_include_template .. Template to create your own Mak
├── Make_include File with system specifc setting and can be adap
├── Makefile .. Controls the compilation and linking of the p
├── FFTlib ........... Library for FFT transformation routine
├── doc ........................ where you can fnd this ma
├── include ............... Directory for the include fle fro
├── lib ........................ Directory where the FFT libra
├── bin ........... Directory for the binaries compiled and l
├── fdelmodc ....... This directory contains all source code
│   ├── FiguresPaper The bash-script to generate the Figures from i
│   │   and Drag (2011)
│   └── demo ............ Bash-script which demonstrate the pos
├── fdemmodc
├── extrap
├── extrap3d
├── marchenko
├── marchenko3D
├── raytime3d
├── fdelmodc3D
├── fdacrtmc
├── zfp
├── utils Source code for programs to generate models, wavelet
├── raytime
├── marchenko_applications
├── corrvir
├── MDD
├── MatInv
├── 3DFD
├── movies
└── scripts

marchenko
├── writeDataIter.c
├── par.h
├── segy.h
├── fmute.c
├── readTinvData.c
├── marchenko_primaries.c
├── readShotData.c
└── applyMute.c
```

```
│
├── marchenko.c
├── applyMute_tshift.c
├── marchenko
├── synthesis.c
├── marchenko_tshift
├── marchenkojan.c
├── findFirstBreak.c
├── writeData.c
├── verbosepkg.c
├── docpkge.c
├── synthesis_cgemm.c
├── getpars.c
├── getFileInfo.c
├── wallclock_time.c
├── readData.c
├── atopkge.c
├── Makefile
├── marchenko_primaries
├── fmute
├── marchenko_tshift.c
└── name_ext.c
demo
   ├── README
   └── oneD
      ├── epsMarchenkoIter.scr
      ├── figAppendix.scr
      ├── conv.gnp
      ├── p5all.scr
      ├── referenceShot.scr
      ├── model.scr
      ├── primariesFrame.scr
      ├── epsPrimaries.scr
      ├── README
      ├── primariesPlane.scr
      ├── primaries.scr
      ├── marchenkoPlaneReg.scr
      ├── initialFocus1300.scr
      ├── iterations.scr
      ├── marchenkodt.scr
      ├── epsModel.scr
      ├── backProp_f2sum_movie.scr
      ├── line3
      ├── initialFocus.scr
      ├── epsBackprop.scr
      ├── epsIterwithLabels.scr
      ├── clean
      ├── marchenkoIter.scr
      ├── marchenkoPlane.scr
      ├── primariesFocus.scr
      ├── initialFocusPlane.scr
      ├── test.scr
      ├── migr.scr
      ├── epsCompare.scr
      ├── marchenko.scr
      ├── f2Plreg.su
      ├── model2.scr
      ├── backpropf2.scr
```

```
│  ├─ windowA60W10.txt
│  ├─ epsWindows.scr
│  ├─ primaries_skiptest.scr
│  └─ 3D
│     ├─ marchenko.scr
│     └─ marchenkoIter.scr
├─ twoD
│  ├─ check.scr
│  ├─ clean
│  ├─ direct.scr
│  ├─ initialFocus_slurm.scr
│  ├─ model.scr
│  ├─ remove_direct.scr
│  ├─ shots_pbs.scr
│  ├─ shots_slurm.scr
│  ├─ backpropf2.scr
│  ├─ backProp_makemovie.scr
│  ├─ eps.scr
│  ├─ marchenko.scr
│  ├─ referenceShot.scr
│  ├─ homgview.scr
│  ├─ marchenko_ray.pbs
│  ├─ marchenko_ray.scr
│  ├─ initialFocus_pbs.scr
│  ├─ epsPrimaries.scr
│  ├─ homg_reference.scr
│  ├─ initialPlane.scr
│  ├─ primaries.scr
│  ├─ README
│  ├─ initialFocus.scr
│  ├─ epsPlane.scr
│  ├─ homgpng.scr
│  ├─ marchenkoPlane.scr
│  └─ rayvsp.scr
├─ ScientificReports
│  ├─ README
│  ├─ back_injrate_planes.scr
│  ├─ backpropf2.scr
│  ├─ check.scr
│  ├─ clean
│  ├─ direct.scr
│  ├─ epsBack.scr
│  ├─ initialFocus.scr
│  ├─ marchenko.scr
│  ├─ model.scr
│  ├─ remove_direct.scr
│  ├─ shotslurm.scr
│  └─ NatureSnapshots.tex
├─ primaries
│  ├─ marchenkojan.scr
│  ├─ marchenko.scr
│  ├─ marchenko_invisible.scr
│  └─ marchenkoGiovanni.scr
└─ invisible
   ├─ marchenko.scr
   ├─ clean
   └─ p4all.scr
```

```
│   │   │── README
│   │   │── model.scr
│   │   │── eps.scr
│   │   │── primaries.scr
│   │── WS15
│   │   │── job.pbs
│   │   │── README.1
│   │   │── README.2
│   │   │── README.3
│   │   │── README.4
│   │   │── README.5
│   │   │── setup.sh
│   │   │── MarchenkoWorkshop.pdf
│   │── mme
│   │   │── epsPrimaries.scr
│   │   │── iterations.scr
│   │   │── model.scr
│   │   │── primariesPlane.scr
│   │   │── primaries.scr
│   │   │── README_PRIMARIES
│   │   │── epsModel.scr
│   │   │── clean
│   │   │── primariesTestuv.scr
│   │   │── strongContrast
│   │   │   │── epsPrimaries.scr
│   │   │   │── model.scr
│   │   │   │── iterations.scr
│── Papers
│   │── MelesGJI2018.pdf
│   │── ThorbeckeGPY2017.pdf
│   │── WapenaarSR2018.pdf
│   │── ZhangGPY2019.pdf

│ /
│── bin/
│   │── basop ....... Executable for basic operations (shift, e
│   │── extendModel Executable to extends the edges of a file with firs
│   │── fconv Executable for auto-, cross-correlation, deconv
│   │── fdelmodc .... Executable for elastic acoustic finite-di
│   │── green ... Executable for the calculation of 2D Greens fu
│   │── makemod ........... Executable for building gridded s
│   │── makewave .................... Executable to generate w
│── /include
│   │── genfft.h ................... Include file for the FFT l
│── /lib
│   │── libgenfft.a ..... Library which contains the objects o
│ /
│── fdelmodc/
│   │── Makefile ...... controls the compilation and linking o
│   │── fdelmodc.h ......... header file which defines structure
│   │── par.h ............ header file from SU for reading in pro
│   │── SUsegy.h ......... adjusted segy header file, which defin
│   │── segy.h ...................... original segy header fr
│   │── acoustic2.c ........ Kernel of acoustic FD using 2'nd o
│   │── acoustic4.c ........ Kernel of acoustic FD using 4'th o
│   │── acoustic6.c ........ Kernel of acoustic FD using 6'th o
│   │── applySource.c .Routine which adds source amplitude(s)
```

```
├── atopkge.c ................. converts ascii to arithmet
├── CMWC4096.c .................. random number generat
├── defineSource.c ...... computes, or read from fle, the s
├── docpkge.c .............. function for self-documenta
├── elastic4.c ......... Kernel of elastic FD using 4'th o
├── fdelmodc.c ........... main FD modeling program, conta
├── fileOpen.c .............. fle handling routines to op
├── gaussGen.c ........ generate a Gaussian distribution o
├── getBeamTimes.c . stores energy felds (beams) in arrays
├── getModelInfo.c reads gridded model fle to compute min/max
├── getParameters.c ..... reads in all parameters to set up
├── getRecTimes.c ......... stores the wavefeld at the rece
├── getWavelets.c reads source wavelet fle and computes maximum
├── getpars.c ..... functions to get parameters from the co
├── name_ext.c ... inserts a character string after the fle
├── readModel.c reads gridded model fles and computes medium p
├── recvPar.c ..... calculates the receiver positions base
├── spline3.c .......... computes interpolation based on t
├── taperEdges.c tapers the wavefeld to suppress unwanted r
├── verbosepkg.c functions to print out verbose, error and w
├── viscoacoustic4.c Kernel of visco-acoustic FD using 4'
├── viscoelastic4.c .. Kernel of visco-elastic FD using 4'
├── wallclock_time.c ........ function used to calculate wa
├── writeRec.c ............ writes the receiver array(s) t
├── writeSnapTimes.c writes gridded wavefeld(s) at a desire
├── writeSrcRecPos.c writes the source and receiver positi
└── writesufile.c ............... writes an 2D array to a SU
```


Almobarak, Mohammed, 2021, Plane-wave Marchenko imaging me
University of Technology.

Behura, J., K. Wapenaar and R. Snieder, 2014, Autofocus imag
scattering theory: Geophysics, 79(3), A19–A26.

Brackenhof, J., 2016, Rescaling of incorrect source strengt
TU Delft Repository, Delft University of Technology.

Brackenhof, J., Thorbecke, J., and Wapenaar, K., 2019, Virt
- considerations for practical applications: Journal of G 124(11),
802–821.

Brackenhof, J., Thorbecke, J., Meles, G., Koehne, V., Bar
Marchenko applications: Implementation and examples: Geo 70, 35–56.

Broggini, F., K. Wapenaar, J. van der Neut and R. Snieder, 20
and application to imaging with multidimensional deconvo
Solid Earth, 119, 425–441.

Broggini, F., R. Snieder, and K. Wapenaar, 2014, Data-driv
multidimensional deconvolution: Numerical examples fro
Geophysics, 79(3), WA107–WA115.

Cohen, J. K. and J. W. Stockwell, 2016, CWP/SU: Seismic Un*
software package for seismic research and processing: Cen
of Mines.

Costa Filho, C. A. da, M. Ravasi, A. Curtis, and G. A. Meles,
retrieval through single-sided Marchenko inverse scatter 90, 063201.

Costa Filho, C. A. da, G. A. Meles, A. Curtis, M. Ravasi and A
Marchenko focusing functions: 79th Annual International M
tists and Engineers, Expanded Abstracts, Tu P9 15.

Dukalski, M. S., and K. de Vos, 2017, Marchenko inversion in a
related multiples: Geophysical Journal International, 212(1), 760–776.

Jia, X., Guitton, A. and Snieder, R., 2018, A practical imp
with a Gulf of Mexico dataset: Geophysics, 83(5), S409–S419.

Lomas, A., and A. Curtis, 2019, An introduction to Marchenko 84(2),
F35–F45.

Lomas, A., S. Singh, and A. Curtis, 2020, Imaging vertical s
vertical seismic-profile data: Geophysics, 85(2), S103–S113.

Matias, M. A., R. Pestana, and J. van der Neut, 2018 Marchenk
tion: Geophysical Prospecting, 66(9), 1653–1666.

Meles, G. A., K. Löer, M. Ravasi, A. Curtis and C. A. da Costa F
and removal using Marchenko autofocusing and seismic inte 80(1), A7–A11.

Meles, G. A., C. A. da Costa Filho and A. Curtis, 2017, Synthe
from multiply-scattered data: 79th Annual International M
tists and Engineers, Expanded Abstracts, Tu P4 11.

Meles, G. A., K. Wapenaar, and J. Thorbecke, 2018, Virtual p
tuming: Geophysical Journal International, 214(1), 508–519.

Meles, G. A., L. Zhang, J. Thorbecke, K. Wapenaar and E. Slob
plane-wave responses: Geophysical Prospecting, 68(6), 1834–1846.

Mildner, C., F. Broggini, K. de Vos and J. O. A. Robertsson,
trum Estimation Using Marchenko Focusing Functions: 79th
Expanded Abstracts, We B2 01.

Mildner C., F. Broggini, K de Vos, and J. O. A. Robertsson, 2
using Marchenko focusing functions: Geophysics, 84(6), Q73–Q83.

Pereira, R., Ramzy, M., Griscenco, P., Huard, B., Huang, H.,
multiple attenuation for OBN data with overburden/target
Meeting, Society of Exploration Geophysicists, Expanded A

Qu, S. and Verschuur, D. J. 2020, Simultaneous joint migrati
version of time-lapse seismic data sets: Geophysical Prosp 68, 1167–1188.

Ravasi, M., I Vasconcelos, A. Kritski, A. Curtis, C. A. da C
oriented Marchenko imaging of a North Sea field: Geophysica 205(1), 99–104.

Ravasi, M., 2017, Rayleigh-Marchenko redatuming for targe
physics, 82(6), S439–S452.

Ravasi, M., and I. Vasconcelos, 2020, PyLops - A linear-oper
optimization: SoftwareX, 11, 100361.

Rietveld, W., G. Berkhout, and K. Wapenaar, C., 1992, Optim
reservoirs: Geophysics, 57(10), 1334–1345.

Singh, S., R. Snieder, J. Behura, J. van der Neut, K. Wapenaa
Imaging with primaries, internal multiples, and free-surf 80(5), S165–S174.

Singh, S., J. van der Neut, K. Wapenaar and R. Snieder, 2016
receivers and virtual sources in the subsurface: 86th Ann
Abstracts, p. 5166–5171.

Singh, S., R. Snieder, J. van der Neut, J. Thorbecke, E. Slo
free-surface multiples in Marchenko imaging: Geophysics, 82(1), R19–R30.

Slob, E., K. Wapenaar, F. Broggini and R. Snieder, 2014, Seis
with Marchenko-type equations: Geophysics, 79(2), S63–S76.

Slob, E., 2016, Green's function retrieval and Marchenko i
Physical Review Letters, 116(16), 164301.

Sripanich, Y., Vasconcelos, I., and Wapenaar, K., 2019, Velo
and depth-imaging domains for media with lateral hete 84(6), Q57–Q72.

Staring, M., J. van der Neut and K. Wapenaar, 2016, An inter
redatuming including free-surface multiples: 86th Annual
stracts p. 5172–5176.

Staring, M., R. Pereira, H. Douma, J. van der Neut, and K. Wap
method for source-receiver Marchenko redatuming on feld d
SEG, Expanded Abstracts, p. 4808–4812.

Staring, M., R. Pereira, H. Douma, J. van der Neut, and K. Wap
redatuming on feld data using an adaptive double-focusing 83(6), S570–S590.

Thomsen, H., 2016, Investigating the robustness of Green's
and Seismic Interferometry: MSc Thesis, ETH Zürich.

Thorbecke, J. and D. Draganov, 2011, Finite-diference mode
Geophysics, 76(6), H1–H18.

Thorbecke, J., J. van der Neut and K. Wapenaar, 2013, Green
equations: A sensitivity analysis: 83th Annual Internat
SPMI 6.3.

Thorbecke, J., E. Slob, J. Brackenhof, J. van der Neut, and K
Marchenko method: Geophysics, 82(6), WB29–WB45.

Thorbecke, J., and J. Brackenhof, 2020, OpenSource code for
https://github.com/JanThorbecke/OpenSource, doi 10.528

Thorbecke, J. and L. Zhang and K. Wapenaar and E. Slob, 2021
multiple elimination a forming: Geophysics, 86(2), F9–F23.

van IJsseldijk, J., J. van der Neut, J. Thorbecke, and K. Wap
traveltime changes in a reservoir using primaries and inte
zone isolation: Geophysics, accepted, 2023

van IJsseldijk, J., J. Brackenhof, J. Thorbecke, and K. Wape
Marchenko method on the Troll feld: under review for Geophy

Van der Neut, J., J. Thorbecke, K. Wapenaar and E. Slob, 201
Marchenko equation: 77th Annual International Meeting, EA

Van der Neut, J., I. Vasconcelos and K. Wapenaar, 2015b, On
substitution of the coupled Marchenko equations: Geophys 203(2), 792–
813.

Van der Neut, J., K. Wapenaar, J. Thorbecke, E. Slob and I V
adaptive Marchenko imaging: The Leading Edge, Vol. 34, p. 818–822.

Van der Neut, J., and K. Wapenaar, 2016, Adaptive overburden
Marchenko equations: Geophysics, 81(5), T265–T284.

van der Neut, J., Johnson, J.L., van Wijk, K., Singh, S., Slo
equation for acoustic inverse source problems: J. Acoust.

Verschuur, E., A. Berkhout, and K. Wapenaar, 1992, Adaptiv
Geophysics, 57(9), 1166–1177.

Wapenaar, K., F. Broggini and R. Snieder, 2012, Creating a vi
data: heuristic derivation and stationary-phase analysis 190 (2), p. 1020–1024.

Wapenaar, K., F. Broggini, E. Slob and R. Snieder, 2013, Th
inverse scattering, data-driven focusing, Green's functi
Review Letters, Vol. 110 (8), 084301.

Wapenaar, K., 2014, Single-sided Marchenko focusing of comp
E, 90(6), 063202.

Wapenaar, K., J. Thorbecke, J. van der Neut, F. Broggini, E. S
retrieval from reflection data, in absence of a receiver at
the Acoustical Society of America, 135(5), 2847–2861.

Wapenaar, K., J. Thorbecke, J. van der Neut, F. Broggini, E.
imaging: Geophysics, 79(3), WA39–WA57.

Wapenaar, K., J. Thorbecke, and J. van der Neut, 2016a, A sin
representation for holographic imaging, inverse scatteri
Green's function retrieval: Geophysical Journal Internat 205, p. 531–535.

Wapenaar, K., J. Thorbecke, J. van der Neut, E. Slob and R. S
responses, Part II: Data-driven single-sided 65(6), 1430–1451: Ge

Wapenaar, K., and van IJsseldijk, J., 2020, Discrete represe
sampled data: Geophysics, 85(2), A1–A5.

Wapenaar, K., Brackenhof, J., Dukalski, M., Meles, G., Rein
J., van der Neut, J., and Zhang, L., 2021, Marchenko redatu
and their mutual relations: Geophysics, 86(5), WC117–WC140.

Zhang, L., and M. Staring, 2018, Marchenko scheme based int
acoustic wavefield: Journal of Applied Geophysics, 159, 429–433.

Zhang, L., and E. Slob, 2019, Free-surface and internal mult
subtraction: Geophysics, 84(1), A7–A11.

Zhang, L., J. Thorbecke, K. Wapenaar, and E. Slob, 2019, Tran
retrieval in data domain and consequences for imaging: Geo 84(4), Q27–Q36.

Zhang, L., and E. Slob, 2020a, A field data example of Marchen 85 (2), S65–S70.

Zhang, L., and E. Slob, 2020b, A fast algorithm for multiple
in primary reflections: Geophysical Journal International 221(1), 371–377.

Zhang, L., and E. Slob, 2020c, Marchenko multiple eliminat
Journal International, 222(2), 1138–1144.