Implementation of the Marchenko multiple elimination algorithm

Jan Thorbecke¹, Lele Zhang¹, Kees Wapenaar¹, and Evert Slob¹

ABSTRACT

The Marchenko multiple elimination (MME) and transmission compensation schemes retrieve primary reflections in the two-way traveltime domain without model information or using adaptive subtraction. Both schemes are derived from projected Marchenko equations and are similar to each other, but they use different time-domain truncation operators. The MME scheme retrieves a new data set without internal multiple reflections. The transmission-compensated Marchenko multiple elimination scheme does the same and additionally compensates for transmission losses in the primary reflections. Both schemes can be solved with an iterative algorithm based on a Neumann series. At each iteration, a convolution or correlation between the projected focusing function and the measured reflection response is performed, and, after each convolution or correlation, a truncation in the time domain is applied. After convergence, the resulting project focusing function is used for retrieving the transmission-compensated primary reflections and the projected Green’s function is used for the physical primary reflections. We have determined that internal multiples are removed by using time-windowed input data that only contain primary reflections. We evaluate both schemes in detail and develop an iterative implementation that reproduces the presented numerical examples. The software is part of our open-source suite of programs and fits into the Seismic Unix software suite of the Colorado School of Mines.

INTRODUCTION

The Marchenko algorithm can eliminate internal multiple reflections in reflection data (Behura et al., 2014; Slob et al., 2014; Wapenaar et al., 2014a). For these schemes, up- and downgoing focusing functions, with a focal point in the subsurface, are retrieved by solving the coupled Marchenko equations. Green’s functions can be computed once these focusing functions are known. A virtual reflection response, with an acquisition surface placed in the subsurface, can be obtained by deconvolving the solved up- and downgoing Green’s functions (Broggini et al., 2014; Slob et al., 2014; Wapenaar et al., 2014b; Van der Neut et al., 2015b; Matias et al., 2018). Based on the constructed virtual reflection response, an artifact-free image at the focal point can be created. The measured single-sided reflection response and a smooth velocity model are required for the implementation of these Marchenko redatuming schemes. The iterative implementation of the Marchenko redatuming scheme has been discussed in detail, and the software has been published by Thorbecke et al. (2017). Lomas and Curtis (2019) illustrate the concepts of the Marchenko method for redatuming and imaging with reproducible scripts in MATLAB. A Marchenko implementation is also part of PyLops (Ravasi and Vasconcelos, 2020).

A wide range of applications has been developed that are based on solutions of the Marchenko equations. Singh et al. (2015) modify the Marchenko scheme to account for free-surface-related multiple reflections. Dukalski and de Vos (2017), Ravasi (2017), and Singh et al. (2017) extend the Marchenko redatuming scheme for marine seismic data and show its performance in numerical and field examples. Meles et al. (2018) propose a different time-focusing condition of the Marchenko redatuming scheme for retrieval of virtual plane-wave responses. The plane-wave scheme allows multiple-free imaging at a fraction of the computational cost of the regular Marchenko scheme. Wapenaar et al. (2017) derive the homogeneous Green’s function retrieval scheme from the Marchenko equations, in which the homogeneous Green’s function between any two points inside a medium can be retrieved from the Marchenko equations, from which one can then determine the Green’s function in the region.

Peer-reviewed code related to this article can be found at http://software.seg.org/2021/0002. Manuscript received by the Editor 1 April 2020; revised manuscript received 20 August 2020; published ahead of production 24 November 2020; published online 19 February 2021.

¹Delft University of Technology, Department of Geoscience and Engineering, P.O. Box 5048, Delft 2600 GA, The Netherlands. E-mail: J.W.Thorbecke@tudelft.nl (corresponding author); L.Zhang-1@tudelft.nl; c.p.a.wapenaar@tudelft.nl; e.c.slob@tudelft.nl.

© 2021 Society of Exploration Geophysicists. All rights reserved.

DOI:10.1190/geo2020-0196.1
timate the initial focusing functions from data that does not rely on a velocity model for mildly varying media. Mildner et al. (2019) develop a method to estimate a source wavelet from the Marchenko focusing functions that enables more precise Marchenko redatuming. The Marchenko redatuming scheme also has been extended from acoustic media to elastic (da Costa Filho et al., 2014; Wapenaar, 2014) and dissipative (Slob, 2016) media. Lomas et al. (2020) develop a vertical seismic profile (VSP) Marchenko imaging methodology that enables imaging of horizontal and vertical structures.

Marchenko-based methods also have been developed for dealing with internal multiple reflections in the two-way traveltime domain. Meles et al. (2015) combine convolutional interferometry with the Marchenko redatuming scheme to approximately attenuate internal multiple reflections. Meles et al. (2017) and da Costa Filho et al. (2017) introduce a method that can directly construct primaries without the need for adaptive subtraction. Van der Neut and Wapenaar (2016) project the coupled Marchenko equations to the surface by convolving both sides of the equations with the initial Green’s function to reduce the requirement of model information. The schemes for Marchenko multiple elimination (MME) (Zhang and Staring, 2018) and transmission-compensated Marchenko multiple elimination (T-MME) (Zhang et al., 2019) are derived from the projected equations to eliminate all orders of internal multiple reflections without model information or adaptive subtraction. In this case, the projected focusing functions are regarded as regular filters that are defined within an offset independent time window; hence, they are truly model independent. The MME scheme retrieves the primaries from the reflection data, whereas T-MME retrieves the transmission-compensated primaries from the upgoing filter function. The MME scheme also has been tested on numerical and field data (Zhang and Slob, 2020a). The examples in Zhang and Slob (2019) show that all orders of internal multiple reflections are successfully eliminated by both schemes. Zhang and Slob (2019) extend the MME scheme to account for free-surface-related multiple reflections. Thus, free-surface and internal multiple reflections are removed in one step without adaptive subtraction or model information.

In this paper, we describe the implementation of the MME and T-MME schemes in detail. Both schemes eliminate internal multiple reflections without the need for model information or adaptive subtraction. Only a reflection response without a source wavelet and free-surface-related multiple reflections is required as input. This paper is organized as follows. In the “Theory” section, we briefly review the equations of the MME and T-MME schemes. In the implementation section, the processing details are explained step by step, and this section provides a user’s first step with the MME and T-MME schemes. The mechanism of the algorithm is illustrated with a simple three-reflector 1.5D horizontally layered model. This simple model is chosen to keep the number of events limited and to allow for an explanation that can be followed more easily. The method is not limited to simple models and can be applied successfully to complicated 3D media as well (Zhang and Slob, 2020c).

The software accompanying this paper contains scripts and source code to reproduce all of the numerical examples presented in this paper. The code also can be found at its GitHub repository (Thorbecke et al., 2017; Thorbecke and Brackenhoff, 2020), where the most recent version and latest developments are available. The commands to reproduce all figures in this paper can be found in the directory marchenko/demo/mme. The README_PRIMARIES in that directory explains in detail how to run the scripts. A more complicated (lateral varying) model can be found in the directory marchenko/demo/twoD. This example will take several hours to compute the reflection data and is not discussed here. To reproduce the figures and to carry out a few pre- and postprocessing steps, Seismic Unix (Cohen and Stockwell, 2016) is required.

**THEORY**

In this section, we give a brief overview of the theory of the MME and T-MME schemes. The acquisition surface is located at the surface boundary \( d \). The reflection response \( R(x_0, x_p, t) \) is measured with the source and receiver positioned at \( x_p \) and \( x_0 \), which is free from free-surface-related multiple reflections and source wavelet. The time is denoted \( t \).

**MME**

As presented by Zhang et al. (2019), we give the equations of the MME scheme as

\[
R_t(x_0', x_0'', t = t_2) = R(x_0', x_0'', t = t_2) + \sum_{m=1}^{\infty} M_{2m}(x_0', x_0'', t = t_2, t_2),
\]

with

\[
M_{2m}(x_0', x_0'', t, t_2) = \int_0^{\infty} \int_{D_0} R(x_0'', x_0', t') H(t-t' - \epsilon) dx_0'' dt' \times \int_0^{\infty} \int_{D_0} R(x_0, x_0'', t'') H(t' - t_2 - t'' - \epsilon) x_0 dx_0 d', \]

and initialization

\[
M_0(x_0', x_0'', t, t_2) = -(H(t + t_2 - \epsilon) - H(t + \epsilon)) \times R(x_0', x_0'', -t),
\]

where \( R \) denotes the retrieved data set without internal multiple reflections at time \( t_2 \) and \( H \) indicates the Heaviside function, which is used to apply the offset independent truncation window \( (\epsilon, t_2 - \epsilon) \) in the equations. The constant \( \epsilon \) indicates a small positive value, which can be taken as the half source time duration in practice. The initialization of the scheme (with \( M_0 \)) is the time-reversed shot record at shot position \( x_0'' \) for times between \( (\epsilon, t_2 - \epsilon) \). The \( R \) in the left side of equation 1 is the same shot record, but without internal multiples. We follow Zhang and Staring (2018) and make time \( t_2 \) constant and independent of the source and receiver positions in the reflection response. Note that the integration is carried out over the receiver coordinate for both integrals, the same as implemented in the source code. The second term in the right side of equation 1 predicts all of the internal multiple reflections correctly. Equation 3 indicates that the measured reflection response is the only input of the MME scheme given in equation 1. To retrieve a data set without internal multiple reflections for all times \( t \), this process must be repeated for all times \( t_2 \).

Equation 2 contains the terms that correct for the internal multiples that are present in \( R(x_0', x_0'', t) \). To better explain the right side of equation 2, we divide the expression into two parts:
Marchenko multiple elimination

\[ M_{2m}(x_0', x_0'', t, t_2) = \int_{t'=0}^{t_2} \int_{x_0'0}^{\infty} R(x_0', x_0'', t, t') H(t-t'-\varepsilon) \times M_{2m-1}(x_0', x_0'', t-t', t_2) dx_0'' dt'. \]

Equation 4 is a time-domain convolution of \( R \) with \( M \) integrated over the spatial coordinate \( x_0'' \), which is the receiver position of the shot at \( x_0' \). Equation 5 is a time-domain convolution of \( R \) with \( M \) integrated over the spatial coordinate \( x_0' \), which is the receiver position of the shot at \( x_0'' \). The Heaviside function in equation 4 is to exclude negative times for \( t-t'<\varepsilon \) and only use the causal time in \( M_{2m-1} \). In equation 5, the function \( M_{2(m-1)} \) contains nonzero values for times larger than \( t_2-\varepsilon \). These times should not contribute to the integral, and the Heaviside function guarantees that \( M_{2(m-1)} \) does not have a contribution to the integration result for values of \( t'' + t - t' > t_2 - \varepsilon \). Note that equations 4 and 5 perform very similar operations (they differ by a sign change) and are implemented by a single function. To evaluate equation 2, this function is applied two times. The convolution terms are the even-numbered operations with this function, and the correlation terms are the odd-numbered operations.

To better explain the method and for illustration purposes, the summation of the even \( M \) terms in equation 4 is defined as the field:

\[ k_{1,i}(x_0', x_0'', t, t_2) = R(x_0', x_0'', t, t_2) \]

\[ -\sum_{m=1}^{i} \int_{t'=0}^{t_2} \int_{x_0'0}^{\infty} R(x_0', x_0'', t', t'') H(t-t'-\varepsilon) \times M_{2m-1}(x_0', x_0'', t-t', t_2) dx_0'' dt'. \]

We can evaluate equation 6 also for \( t \geq t_2 - \varepsilon \) and the equation can be further split in the time domain as follows:

\[ k_{1,i}^{-}(x_0', x_0'', t, t_2) = \begin{cases} v_{1,i}^+(x_0', x_0'', t, t_2) & t < t_2 - \varepsilon \\ u_{1,i}^+(x_0', x_0'', t, t_2) & t \geq t_2 - \varepsilon \end{cases} \]

where \( u_{1}^+ \) and \( v_{1}^+ \) are similar to the projected Green’s function and focusing function in the regular Marchenko scheme as defined in Van der Neut and Wapenaar (2016). The minus superscript in equations 6 and 7 refers to upgoing wavefields at the receiver location \( x_0' \). To solve for \( M \) in equations 1 to 3, \( k_{1}^- \) is not needed. The time values in \( M \) between \( \varepsilon \) and \( t_2 - \varepsilon \) are used to solve the Marchenko equations and compute a value at \( t_2 \) in \( u_{1}^- \). The solution for time \( t_2 \) in the MME scheme is collected from \( u_{1}^- \) at \( t = t_2 \).

Similarly for the summation of the terms in equation 5, a down-going function is defined as

\[ v_{1,i}^+(x_0', x_0'', t, t_2) = \sum_{m=1}^{i} \int_{t'=0}^{t_2} \int_{x_0'0}^{\infty} R(x_0', x_0'', t', t'') H(t-t'+\varepsilon) \times M_{2(m-1)}(x_0', x_0'', t+t', t_2) dx_0 dt''. \]

where \( v_{1}^+ \) is similar to the projected focusing function in the regular Marchenko scheme as defined in Van der Neut and Wapenaar (2016). In \( v_{1}^+ \), the multiple annihilator is created, and this is demonstrated in the “Numerical examples” section. Equation 8 only holds for \( \varepsilon < t < t_2 - \varepsilon \). The plus superscript in equation 8 refers to downgoing wavefields. To solve the MME Marchenko equations, \( v_{1}^+ \) is not needed and is only defined for illustration purposes to explain the mechanism of the method.

Time \( t_2 \) is the instant two-way traveltime in which the solution of the Marchenko equation is computed. The primary reflection is collected from \( u_{1}^+ \) for every time instant \( t_2 \). This is a computationally expensive way because only one sample is collected in the output. Nevertheless, this process is fully automated and implemented without any human interaction or model information. It is possible to collect more than one sample around the instant time \( t_2 \) and to take bigger time steps, but the number of samples to use around \( t_2 \) must take into consideration the frequency bandwidth of the data. This statement is supported by examples in the detailed discussion of the implementation and allows the implementation of a faster algorithm.

In this MME scheme, the primary is collected from the original reflection data. The Marchenko scheme removes all overlapping internal multiples from earlier reflections, and the primary is untouched and keeps the physical reflection amplitude as present in the data.

T-MME

Internal multiple reflections and transmission losses in primary reflections can be accounted for by the T-MME scheme (Zhang et al., 2019). The equation is given by

\[ R_r(x_0', x_0'', t = t_2) = R(x_0', x_0'', t = t_2) \]

\[ + \sum_{m=1}^{\infty} \widetilde{M}_{2m}(x_0', x_0'', t = t_2, t_2), \]

with

\[ \widetilde{M}_{2m}(x_0', x_0'', t, t_2) = \int_{t'=0}^{t_2} \int_{x_0'0}^{\infty} R(x_0', x_0'', t', t'') H(t-t'+\varepsilon) dx_0'' dt' \]

\[ \times \int_{t'=0}^{t_2} \int_{x_0'0}^{\infty} R(x_0', x_0'', t', t'') H(t-t'+\varepsilon) dx_0 dt''. \]

and

\[ \tilde{M}_0(x_0', x_0'', t, t_2) = -(H(t+t_2+\varepsilon) - H(t+\varepsilon)) R(x_0', x_0'', -t), \]

where \( R_r \) denotes the retrieved data set without internal multiple reflections and transmission losses in primary reflections. The trun-
The primary reflection is, different than in the MME scheme, collected from \( \bar{v}_T \), which achieves the transmission compensation. There is no need to define \( k_1 \) because \( \bar{v}_T \) is already part of \( M_{2n} \). The scheme is applied for every time instant \( t_2 \) and has the same advantages and disadvantages as the MME scheme. In the T-MME scheme, the amplitude of the primary is automatically compensation. In equation 10, the Heaviside function guarantees that \( \bar{v}_T \) does not have a contribution for values of \( t'' + t - t' > t_2 + \epsilon \). In contrast to equation 5, time \( t_2 \) is now part of the integration and is included in the sum of \( M_{2n} \). Still, as given in equation 11, the measured reflection response is the only input to solve the T-MME scheme given in equation 9.

The primary reflection is, different than in the MME scheme, collected from \( \bar{v}_T \), which achieves the transmission compensation. There is no need to define \( k_1 \) because \( \bar{v}_T \) is already part of \( M_{2n} \). The scheme is applied for every time instant \( t_2 \) and has the same advantages and disadvantages as the MME scheme. In the T-MME scheme, the amplitude of the primary is automatically compensated because it is the only way to predict and attenuate internal multiples from earlier primary reflections. We will return to this remark in the explanation of Figure 1.

The MME and T-MME schemes require only the measured reflection response \( R \) as input. The reflection response \( R \) needs to be deconvolved for the source-wavelet, and the free-surface-related multiple reflections must be removed. The output of a surface-related multiple elimination (Verschuur et al., 1992) scheme can meet these requirements. Diffracted and refracted waves are beyond the capability of both schemes, and a detailed analysis about these limitations can be found in Zhang et al. (2019).

MARCHENKO ALGORITHM

The basic Marchenko algorithm (MME) is explained in Algorithm 1. The arrays in this algorithm are stored in C-order; the last (right-most) addressed dimension is contiguous in memory. The discrete dimensions of these arrays are within square brackets \([\ldots]\), and the arguments of function calls are within regular brackets \( [\ldots] \), and the arguments of function calls are within regular brackets \( (\ldots) \). The only data input of the algorithm is the measured reflection data \( R \). These reflection data must be properly preprocessed as explained in Brackenhoff et al. (2019). The preprocessing must take care of the following:

1) Elimination of free-surface multiples. Note that there is also a very similar Marchenko algorithm that takes into account free-surface multiples as well. Ravasi (2017) discusses a redatuming algorithm similar to Singh et al. (2015) and requires a smooth model of the medium, whereas Zhang and Slob (2019) remove all multiples and do not need any model information.
2) Sufficient (i.e., alias-free) sampling in the spatial receiver and shot direction. Note that there are Marchenko-based methods that can fill in missing shotpoints or receiver locations, under the assumption that the available data are unaliased (Wapenaar and van IJsseldijk, 2020).
3) Compensation for dissipation.
4) Shot amplitude regularization.
5) Deconvolution for the source wavelet.

Following Algorithm 1, the preprocessed reflection data are read from the disk, transformed (by fast Fourier transform [FFT] operator \( F[\ldots] \)) to the frequency (\( \omega \)) domain, and all shots and receivers are stored into the memory. This is the first step in the algorithm and the only significant data read. One single shot record (with shot number \( j \), from which we want to suppress the internal multiples, is selected from this reflection data in the next step. This shot record is transformed back to time, time reversed \((R^*)\), and stored in array \( DD \). The first loop in the algorithm loops over the selected number of time samples that are processed to attenuate internal multiples. Typically, this represents all samples in the shot record, with a possible exclusion of the number of samples to the first reflection event in the selected shot record. For each time sample \( i_i \), the iterative Marchenko algorithm is executed. The largest difference from the algorithm described in Thorbecke et al. (2017) is that time-truncation along the first arrival time (from a focal point in the subsurface) is replaced by a constant time-truncation and the computation of a first arrival time is not needed anymore. The initialization of the algorithm by \( M_0 \) is from the same shot record \( j \) from which we would like to attenuate the internal multiples (\( DD \)). The term \( M_0 \) is a copy of the time-reversed shot record (see equation 3) and is set to zero from the first sample 0 to sample \( n_i - ii + n_r \), where \( n_r \) is the total number of samples in the shot record. The extra samples of \( n_i \) take into account the time duration of the wavelet to exclude a possible reflection event at time \( ii \). The initialization of \( k_{1,0} \) is a complete (no time muting is carried out) copy of the shot record that still contains all of the internal multiples that we would like to remove.

With these two initializations, the iterations of the Marchenko algorithm can start. In each iteration, an updated field is computed by the integration of \( M_i \) with \( R \). This integration process is called synthesis, produces the output \( RM_i \), and is explained in more detail below. Depending on the iteration number \( i \), being odd or even, different time muting windows are in use to mute

**Figure 1.** Comparison of the MME and T-MME schemes. (a and b) A selected time \( t_2 \) equal to the two-way traveltime of the third reflector. The time-truncation window is indicated with a red dotted line. The dotted lines are events that are excluded in \( M_i \), and the solid lines are events included in \( M_i \), after application of the time window. (c and d) A time between two reflectors.

DOI:10.1190/geo2020-0196.1
Marchenko multiple elimination

Algo 1. The basic Marchenko algorithm, without transmission loss compensation, as implemented in the provided source code. Integer time sample number $i$, which runs from $i_{\text{start}}$ to $i_{\text{end}}$, represents time $t = it \cdot \Delta t$. The number of recorded time samples is $n_t$, the time duration of the source signature $\varepsilon = n_t \cdot \Delta t$, and time sample $ii$ represents instant time $t_{ii} = ii \cdot \Delta t$. The number of receivers in $R$ is $N_{\text{recv}}$, and the number of shots is $N_{\text{shots}}$. The $i$ loop represents the number of Marchenko iterations $n_t$. Note that the sample expression $n_t - ii$ it stands for negative time $-t$.

Main begin
Read SU-style input parameters
Initialization, reading of input parameters, and allocate arrays
READ(\$N_{\text{shots}}, i_{\text{os}}, N_{\text{recv}}\$
\$DD[N_{\text{recv}}, ii] = \mathcal{F}^{-1}\{R^*([j, i_{\text{os}}, N_{\text{recv}}]\$
for $ii\leftarrow i_{\text{start}}$ to $i_{\text{end}}$
do
\[$M_0[N_{\text{recv}}, ii] = \begin{cases} 0 & 0 < it < n_t - ii + n_e \\
-DD[N_{\text{recv}}, ii] & n_t - ii + n_e \leq it < n_t \end{cases}\$
\[$k_{1,0}[N_{\text{shots}}, ii] = DD[N_{\text{recv}}, n_t - it]\$
\[$v_{1,0}[N_{\text{shots}}, ii] = 0\$
for $i\leftarrow 0$ to $n_t$
do
synthesis(R, M, RM)
$M_{i+1}[N_{\text{shots}}, ii] = RM \cdot [N_{\text{shots}}, n_t - it]$
if $(i \% 2 == 0)$ then
$M_{i+1}[N_{\text{shots}}, ii] = 0; \quad ii - n_e < it < n_t$
$v_{1,i+1}[N_{\text{shots}}, ii] = v_{1,i}[N_{\text{shots}}, ii] + M_{i+1}[N_{\text{shots}}, ii]$
else
$k_{1,i+1}[N_{\text{shots}}, ii] = k_{1,i}[N_{\text{shots}}, ii] - M_{i+1}[N_{\text{shots}}, n_t - it]$
$M_{i+1}[N_{\text{shots}}, ii] = 0; \quad 0 < it < n_t - ii + n_e$
end
end
$R_i[j, N_{\text{shots}}, ii] = k_{1,0}[N_{\text{shots}}, ii]$ end

Main begin
Read SU-style input parameters
Initialization, reading of input parameters, and allocate arrays
READ($R(N_{\text{shots}}, i_{\text{os}}, N_{\text{recv}})$\n$DD[N_{\text{recv}}, ii] = \mathcal{F}^{-1}\{R^*([j, i_{\text{os}}, N_{\text{recv}}]\$
for $ii\leftarrow i_{\text{start}}$ to $i_{\text{end}}$
do
$k_{1,0}[N_{\text{shots}}, ii] = k_{1,0}^{(n-1)}[N_{\text{shots}}, ii]$
$M_0[N_{\text{shots}}, ii]$
= $\begin{cases} 0 & 0 < it < n_t - ii + n_e \\
DD[N_{\text{shots}}, n_t - it] - k_{1,0}[N_{\text{shots}}, n_t - it] & n_t - ii + n_e \leq it < n_t \end{cases}$
for $i\leftarrow 0$ to $n_t$
do
synthesis(R, M, RM)
$M_{i+1}[N_{\text{shots}}, ii] = RM[M_{i+1}[N_{\text{shots}}, n_t - it]]$
if $(i \% 2 == 0)$ then
$M_{i+1}[N_{\text{shots}}, ii] = 0; \quad ii - n_e < it < n_t$
else
$k_{1,i+1}[N_{\text{shots}}, ii] = k_{1,i+1}[N_{\text{shots}}, ii] - M_{i+1}[N_{\text{shots}}, n_t - it]$
$M_{i+1}[N_{\text{shots}}, ii] = 0; \quad 0 < it < n_t - ii + n_e$
end
end
$R_i[j, N_{\text{shots}}, ii] = k_{1,0}[N_{\text{shots}}, ii]$ end

events in $RM_i$ and to compute an updated $M_{i+1}$. For even iterations, the times between $ii - n_e$ and $n_t$ are set to zero, and, for odd iterations, the times between $0$ and $ii + n_e$ are set to zero. Only in the odd iterations is $k_{1,i}^O$ updated with the unmuted $M_{i+1}$. In this update of $k_{1,i}^O$ internal multiples around time $ii$ are attenuated. This is the update represented in equation 1, where the update $M_{2im}$ is in fact one even and one odd iteration in the implemented Marchenko algorithm and, hence, the notation with $2m - 1$ in equation 6.

In the regular redatuming Marchenko algorithm (Thorbecke et al., 2017), the truncation windows follow the first arrival time of a focal point in the subsurface. In the MME algorithm, the focal point is projected on the surface and the time-truncation is conveniently chosen at a constant time. The flat time window has the big advantage that it requires no additional model or data information. Meles et al. (2020) demonstrate that, in the application of the multiple elimination algorithm to dipping plane waves, a time truncation consistent with the dip angle must be used.

Depending on the position of strong reflectors, typically, 10–50 Marchenko iterations are needed for each time sample $ii$ in the selected shot record. The presence of strong reflectors in the shallow part makes the convergence slow at large time instances; see also Figure 8. The reason is that higher order multiples are attenuated with events that are created in $v_{ii}$ and that are removed again later when the first-order multiple is finally removed by a converged multiple attenuator. When the first-order multiple is removed, all multiples are removed; hence, all earlier higher order multiple attenuator artifacts will also vanish. Once the iterations are finished, the output of sample $ii$ of the updated Marchenko result $u_{ii}^r$ is stored in sample $ii$ in the multiple free shot record $R$, and the final output of the program that represents the selected shot record with attenuated internal multiples. It is a computationally intensive task to solve the Marchenko equations for each sample $ii$ in the shot record. Algorithm 2 is a faster (10–20x) implementation of Algorithm 1.

In Algorithm 2, after the Marchenko equations are solved at time sample $ii - 1$, the next time sample $ii$ is initialized with the result of time sample $ii - 1$ (Zhang and Slob, 2020b). The idea is that to remove the internal multiples at the next time sample there is no need to start from scratch and remove the multiples that were already removed in the previous time sample. For every next time sample, all earlier attenuated multiples need to be attenuated plus one (or a few) more. In the Marchenko update for the new time sample, only the multiples that were not removed previously need to be removed. This is a small deviation from the previous results, and usually two iterations are sufficient to accomplish the update for
the next time sample. The initial $M_0$ in the fast algorithm is the difference between the original data ($DD$) and $k_{i-1}^{r,-(+)}$; the already estimated internal multiples from time sample $ii - 1$. The initial $k_{i-1}^{r,0}$ is the previous result $k_{i-1}^{r,(-)}$. With these initializations, the update term $RM_i$ contains only a small correction because it is based on a converged previous result that is very close to the actual solution. To get the complete internal multiple in $M_{i+1}$, $DD$ is added to $RM_i$ (Zhang and Slob, 2020b).

In this fast algorithm, only one pair of even-odd iterations is needed to reach convergence. In principle, we could solve the equations only one time and use that result to update all other time samples. On simple models of numerically modeled data, this works fine indeed. However, on geologically complicated models of numerically modeled data and on field data, we have to do a full update every 10–20 recursive updates and the speed-up of the faster algorithm is limited to one order of magnitude. On complex data sets, we would advise beginning with the basic algorithm and then verifying if the fast algorithm can be used to speed up the computations. The reason for this limited use of the fast algorithm is that, for complex data sets and a large number of iterations, artifacts, for example, introduced by a limited aperture, can get amplified. The primary reflections will still converge, but numerical artifacts are not accounted for in the algorithm and can diverge. In the iterative scheme, each update adds two iterations to the already computed result based on, for example, 30 iterations. With 10 iterative updates, 20 iterations are added and can cause artifacts being amplified to signal level.

In the algorithm, we solve the Marchenko equations for each sample $ii$. From the theory, we know that the first event after sample $ii$ is a primary reflector (all multiple reflections generated by the reflectors before sample $ii$ are removed by the scheme). Hence, a few samples after sample $ii$ will still be free of internal multiples. We could make larger steps with $ii$ and use the Marchenko result for several samples (at least $n_i$ samples because that is the time resolution with which we already are working) after sample $ii$. This can speed up the code by $n_i$ (typically 20) times. This is similar to the fast algorithm, but without making any iterations and directly using the previous computed result.

The synthesis process shown in Algorithm 3 computes the second integrant in the right side of equation 2. The synthesis function is a straightforward matrix-vector multiplication. The reflection data are stored in such a way that the innermost loop, which sums over the receiver positions within a shot, is contiguous in memory.

To speed up the computation, a parallel OpenMP region is carried out over the outer $N_{shot}$ loop. An alternative implementation of the synthesis process is to make the frequency loop the outer loop and use a BLAS dgemv function to compute the matrix-vector multiplication. This implementation also will be efficient when all shots are computed at the same time and the BLAS matrix-matrix dgemm function becomes the kernel of the synthesis process. Note that in the synthesis process, the integration is carried out over the number of receivers per shot and each integration result is stored at the shot position. Thus, after the synthesis process $N_{shots}$ output traces are computed.

From a computational point of view, the transmission-compensated algorithm (T-MME) is the same as the MME algorithm, except for the application of the time-truncation window. The sample length of the wavelet ($n_s$) is applied in the opposite time direction for the T-MME algorithm. The extra samples of $n_s$ in the MME algorithm take into account the length of the wavelet to exclude a possible event at instant $ii$ in the initialization and update of $M_i$. Suppose that time $ii$ is the two-way traveltime of a reflector (see Figure 1a). The reflection of

![Figure 2](image_url)
the reflector is excluded in \( M_i \) in the MME algorithm (Figure 1a),
but it is included in \( M_i \) in the T-MME algorithm (Figure 1b).
In the T-MME algorithm, the reflected event at instant time \( ii \) ends up in
the updated \( \hat{r}_n \), whereas in the MME algorithm the reflected event
from the original shot record ends up in \( r_n \). When the instant time \( ii \)
is chosen between two reflectors, there is no difference between the
updates made in the MME (Figure 1c) or T-MME (Figure 1d)
scheme.

To get to the T-MME scheme from Algorithm 1, one has to re-
place in the defined time windows \( +n_e \) with \( -n_e \). Then, \( R \) becomes
\( \hat{R} \) that contains the transmission-compensated primary reflections.

**NUMERICAL EXAMPLES**

The Marchenko algorithm is illustrated with a 1.5D horizontally
layered model shown in Figure 2. The numerical modeling is
conducted with a finite-difference modeling program (Thorbecke
and Draganov, 2011) that is also included in the software package.
The input source signature, to model the reflection response
\( R(x_0, x', t) \), is approximately a sinc-function with a flat spectrum
of amplitude one between \( f_{\min} \) and \( f_{\max} \) (5–90 Hz) to represent a deconvolved source wave-
let. In the finite-difference program for modeling
\( R(x_0, x', t) \), a source of vertical force is
chosen. The receivers are placed at \( z = 0 \) and measure the pressure field.
A fixed spread acquisition is chosen between \(-2250 \) and \( 2250 \) m,
and the distance between the 901 source/receiver positions is \( 5 \) m. The receiver traces have a time sampling interval of \( 4 \) ms and 1024 recorded
time samples.

**The first iterations**

Figure 3 illustrates the first iteration of equation 5 with \( m = 1 \) to compute \( M_1 \) for time
sample number \( 276 \) \((t = 1.100 \) s\) from \( M_0 \). Time
sample 276 corresponds to the zero-offset arrival
time of the third reflector. In this first step, all
shots in the reflection data \( R \) are correlated with a
time-windowed shot record. In our example, we use the middle shot record:
\( R_0(x_R, x = (0, 0), t) \) (shot number \( j = 451 \)). Before the cor-
relation is carried out, the selected shot record is
first set to zero beyond time sample \( 276 - n_e \),
multiplied by \(-1 \) and time reversed, at which moment
we have \( M_0(t) \) in equation 3. In Figure 3b,
the shot record is convolved with a Ricker wave-
let to reduce the ringing of the flat spectrum of the (deconvolved) wavelet present in \( R \) (Figure 3a). The number of \( n_e \) (in this example
\( n_e = 20 \)) samples excludes the reflection from
the third reflector in \( M_0 \). In Figure 3, the middle
shot record of \( R \) (Figure 3a, where we used
source receiver reciprocity) is correlated with
the time windowed \( M_0(-t) \) (Figure 3b) to give
the result in Figure 3c. The events in Figure 3b
include the first and second reflections and the
first internal multiple between the first and second
reflectors. In the correlation result (Fig-
ure 3c), we see the autocorrelation of the three reflection events
around \( t = 0 \) (with events at negative times appearing at the bottom
of the panel). Note that the long train of events starting at the positive
time axis in Figure 3c can interfere with events at the end of the
time axis. To overcome this time interference, we usually pad the
time axis with zeros before the transformation to the frequency do-
main where the correlation is computed.

The correlation result is time reversed and is shown in Figure 3d
for the first 400 samples. There are only three events in Figure 3d,
and these originate from correlation with the three events in Figure
3b with the first three events \((r_1, r_2, m_i) \) in the shot record.
According to the integral in equation 5, to obtain an output trace of \( M_1 \)
the traces in Figure 3d are summed together. The stationary points
of the events in Figure 3d give a contribution in the result of the
summation. Besides the stationary points, truncated events (in time
and space) give unwanted contributions that show up as artifacts in
the final result.

The integration result is set to zero for samples larger than
\( 276 - n_e \) and ends up as a trace at position zero (the middle trace)
of \( M_1 \) shown in Figure 3e. In Figure 3e, the truncation appears to be

\[
M_1(x, x, t) = R(x, x_R, t) \times M_0(x_R, x, -t)
\]

Figure 3. Computational steps to compute \( M_1 \) from \( M_0 \) at time sample number 276. The middle shot record from \( R \) is shown in (a); time truncated after sample \( 276 - n_e \),
and convolved with a Ricker wavelet, it gives \( M_0 \) in (b). Time correlation of (a) with (b) gives
correlation (c) and applying the time window again gives (d). The traces in
d(d) are summed together, and only the stationary point of events above sample
\( 276 - n_e \) will end up in the middle trace of \( M_1 \) (e). The mute window \( n_e \) samples later than \( t = 0 \) are
needed to mute the autocorrelation of the first event. The labeled events \( r_i \) indicate
the \( i \)th reflector and \( m_i \) the \( i \)th multiple. (c) The labeled correlated events are
\( c_0 = r_1^* r_1 + r_2^* r_2 + m_1^* m_1 \), \( c_1 = r_2^* r_3 + m_1^* r_2 \), and \( c_2 = m_1^* r_1 \).
at sample 160, but that is the truncation in Figure 3b shifted upward in time with the arrival time of the first reflector. The truncation at sample 276 – \( n_z \) is indicated with a dotted line. There are two hyperbolic events visible and a few linear artifacts. The first hyperbolic event originates from correlation of events \( r_1, r_1 \) and events \( m_{1}, r_2 \), and the second hyperbolic event from the first internal multiple and first reflector \( c_2 = m_{1}, r_1 \). The linear events are unwanted artifacts due to truncation and can be suppressed by applying a smooth taper at the truncation boundaries in time and space.

Figure 4 demonstrates the computation of the second iteration to compute \( M_2 \) from \( M_1 \) (Figure 3e) according to equation 4 with \( m = 1 \). The reflection data (Figure 4a) are convolved with \( M_1 \) (Figure 4b) that contains three main events: a linear artifact \( (a_1) \) and two correlation results \( (c_1 \) and \( c_2 \). Convolving \( M_1 \) with the middle shot record of the data \( R \) gives Figure 4c. The hyperbolic events in \( M \) are now back at the same times as the reflection events in the shot record. The linear artifacts in \( M_1 \) also convolve with all events in \( R \) and introduce many (mostly linear) artifacts. The convolution result is reversed in time (Figure 4d), and, after the integration in equation 4 over the lateral coordinate \( x_k \), it becomes the middle trace in \( M_2 \) (Figure 4e). Most of the linear artifacts are reduced in amplitude due to the destructive interference in the integration, only the “\( a_2 = r_1, a_1 \)” artifact is still present in \( M_2 \). The first term in the sum in equation 1 is now computed: \( M_2 \). The last events in the time reverse of \( M_2 \) presented in Figure 4e, will already attenuate the multiple event \( m_1 \) in the shot record.

To compute \( M_1 \) (in general, odd-numbered updates to \( M_1 \)), events are shifted backward in time (correlation) with the times of the events in \( M_0 \). To compute \( M_2 \) (even-numbered updates to \( M_1 \)) from \( M_1 \), events are shifted forward (convolution) in time. The even and odd iterations are treated differently in the scheme. Each even iteration updates \( M_1 \) and \( v_i \), and each odd iteration updates \( M_1 \) and \( k_i \). The scheme reverts the time axis for each iteration, hence, the time windows, which set time samples to zero, also switch. These time windows, for sample 276, are shown in Figure 5. In these time windows, a smooth cosine-shaped transition zone is used to reduce the time-truncation artifacts.

### Multiple removal in action

The results in Figure 6 are partial solutions of the Marchenko equations computed for time sample \( i = 200 \). After applying the time window, which sets all samples in \( M_0 \) to zero beyond \( 200 - n_z \), there are no internal multiple reflections present anymore in \( M_0 \). The times between 0 and sample 200 include \( r_1 \) and \( r_2 \), but not \( m_1 \); see Figure 3b. In the first iteration to compute \( v_1 \), according to equation 8, one extra event in \( v_1 \) (Figure 6a event \( c_1 \)) is created to correct for the amplitude of the second reflector in \( v_1 \). Note that the time windows in Figure 6a appear to be around sample 80, but this is the time window applied in \( M_0 \) shifted by the correlation to negative times and time reversed. The amplitude of this event \( c_1 \) converges to the amplitude that can
annihilate the amplitude of the first multiple. Applying the converged \( r_1^+ \) on the reflection data through equation 4 causes all multiples arising from bounces between the first and second reflector to vanish from the data in equation 1. The scheme finishes without ever having “seen” the multiple; from \( r_1 \) and \( r_2 \) alone, it created an event that can attenuate all of the internal multiples between these reflectors. The arrows in Figure 6c, which shows \( k_{30}^+ (t) \) and is computed according to equation 6, point at the multiples that are already partly gone. The multiples are only partly removed because only a small offset range of \( r_2 \) is used at sample 200. Repeating the scheme for samples larger than 200 will include larger offsets of \( r_1 \) and \( r_2 \) and also attenuate the higher offsets for all internal multiples between \( r_1 \) and \( r_2 \).

For the investigation of the amplitudes of the event in \( r_1^+ \), we assume, for the sake of argument, that the reflection coefficient is a constant. The primary reflections in Figure 3b have local reflection \( a_1 \) and \( a_2 \) for, respectively, the events labeled \( r_1 \), \( r_2 \). We consider the situation in Figure 6, for time sample 200 that only creates one extra event in \( r_1^+ \). We demonstrate that, after sufficient iterations, the event in \( r_1^+ \) has converged to an amplitude that can cancel the first-order multiple and hence all higher order multiples related to that event. For sample 200, \( M_0 \) contains only the primary reflections \( r_1 \), \( r_2 \) that have amplitude:

\[
r_1^a = a_1,\quad r_2^a = (1 - a_1^2) a_2.
\]

Figure 7 is a sketch of the reflection paths and reflection and transmission coefficients for this two-reflector case. According to equation 5, for the first iteration \( M_0 \) is correlated with \( R \) and integrated over the receiver coordinate. After applying the time window on \( M_1 \) only one event remains: event \( c_1 = r_1^a, r_2 \) in Figure 6a with the amplitude

\[
c_{1,1}^a = a_1 (1 - a_1^2) a_2.
\]

The second subscript in \( c_{1,1}^a \) indicates the iteration number. This event is convolved with \( R \) in the next iteration (according to equation 4), and, after time windowing, only one event \( r_1, c_1 \) remains at the time of the reflection of the second reflector with the amplitude

\[
c_{1,2}^a = a_1^2 (1 - a_1^2) a_2.
\]

Figure 6. Creation of the event (labeled \( c_1 \)) that annihilates all of the internal multiples between the first and second reflectors; the artifact \( a_1 \) is ignored in the analysis. (a) The \( r_1^a, r_2 \) for the first Marchenko iteration at sample number \( i = 200 \). (b) The convergence of the maximum amplitude in \( c_1 \) is shown as a function of the iteration count. (c) The annihilated multiples in \( k_{30}^+ (t) \) after 30 iterations.

In each next iteration, alternating between equations 5 and 4, another multiplication with \( a_1 \) is added; in general, for iteration \( i \), we have

\[
c_{1,i}^a = (a_1)^i (1 - a_1^2) a_2.
\]

Summation of all odd \( c_{1,i}^a \) iterations \( i \) gives the final amplitude of the multiple annihilation event \( c_1 \) in \( r_1^+ \). The initialization of \( r_1^+ \) is zero, and the summation of the odd terms leads to

\[
\sum_{i=0}^{n_1} c_{1,1+2i}^a = \sum_{i=0}^{n_1} (a_1)^{1+2i} (1 - a_1^2) a_2 = a_1 a_2 - a_1^3 a_2 + a_1^5 a_2 + a_1^7 a_2 - a_1^9 a_2 + \ldots
\]

\[
\approx a_1 a_2.
\]

Application of \( r_1^+ \) to the data creates multiple-free data in the resulting \( u_1^+ \) and is shown in Figure 6c. The first-order internal multiple from the data and the multiple annihilation event \( c_1 \) in Figure 6a, with amplitude \( a_1 a_2 \), will meet each other in time just below the first reflector. At that point in time, the annihilator cancels the first-order downgoing internal multiple and, with that, all other related multiples. To be able to cancel the first downgoing internal multiple, the annihilator must have the same amplitude as that event. The first-order multiple event \( m_1^a \) in Figure 3b has the amplitude

\[
m_1^a = -(1 - a_1^2) a_2^2 a_1.
\]

After convergence of the scheme, the multiple annihilator event \( c_1 \) is convolved with the second reflector \( r_2 \) of \( R \) in the next iteration and arrives at the same time as \( m_1 \) and has the same amplitude as \( m_1^a \):

\[
\sum_{i=0}^{n_1} c_{1,1+2i}^a = \sum_{i=0}^{n_1} (a_1)^{1+2i} (1 - a_1^2) a_2
\]

\[
= a_1 a_2 - a_1^3 a_2 + a_1^5 a_2 + a_1^7 a_2 - a_1^9 a_2 + \ldots
\]

\[
\approx a_1 a_2.
\]
\[ c_1^2 = a_1a_2, r_2^2 \]
\[ = (1 - a_1^2)a_2^2a_1. \]  

(19)

This result is added to the data to cancel the internal multiple at \( m_1 \) as shown in Figure 6c and equation 1. Furthermore, convolution of \( c_1 \) with \( m_1 \) will create the annihilator of the second-order multiple; hence, \( c_1 \) will automatically annihilate all higher order multiples as well.

To complete the amplitude analysis, the amplitude of the second reflector in \( \bar{v}_1 \) (from the equivalent of equation 7 for the T-MME scheme) can be computed according to equation 10 and is constructed from the even amplitude terms in equation 16. The initialization of \( \bar{v}_1 \) is the time reversed shot record (DD in Algorithm 1). Summation of all even \( c_i^2a \), iterations \( i \) at the time of the second reflector creates the final amplitude for the second reflector in \( \bar{v}_1 \):

\[ a_2 = (1 - a_1^2)a_2 + \sum_{i=1}^{n} a_2^i(1 - a_1^2)a_2 \]
\[ = a_2 - a_2^2a_2 - a_2^4a_2 - a_2^4a_2 + a_2^6a_2 + \ldots \approx a_2. \]  

(20)

This shows that the transmission-compensated local reflectivity can be collected from \( v_1 \) as implemented in the T-MME scheme. The approximation sign is due to a limited number of iterations in the numerical implementation.

Figure 8 is obtained in the same way as Figure 6b, but with high-contrast layers. The velocity of the layers is the same as used in Figure 6, but the density contrast between the layers has been in-

Figure 8. Convergence of the maximum amplitude of the event (labeled \( c_1 \) in Figure 6a) that annihilates all the internal multiples between the first and second reflector in a high-contrast medium. 

Figure 9. The term \( M_i \) fields for a focal time at sample \( ii = 276 \); the zero-offset arrival of the third reflector. All figures are plotted with the same clipping factor.
creased from a factor 3 (1000–3000) to 10 (500–5000). Compared to Figure 6b, the convergence is much slower in this high-contrast medium. In equation 17, the higher order terms will have larger values in high-contrast media and require more iterations for convergence.

**Higher iteration counts**

The first few iterations for the update terms \( M_i \) are shown in Figure 9. The truncation time is chosen at sample 276, and a first-order multiple of the second layer is present in the initialization shot-record \( M_0 \) after time truncation. For higher numbers of iterations, the update terms become smaller in amplitude, indicating that the scheme converges. All of the updates show the same number of events, and only the amplitude of the events changes during the iterations.

In the odd iterations, the function \( k_{-1;i}(t) \) (see equation 6) is updated with the odd \( M_i(-t) \) terms and four selected iterations are shown in Figure 10. After two iterations, all order multiples are predicted, but with incorrect amplitudes. In the following iterations, the removal of higher order multiples is improved because the removal of the first-order multiple improves. After 20 iterations, the internal multiple events (indicated with arrows) have further attenuated and are not visible anymore; compare Figure 10b with 10h. The higher order multiples do not have to be removed by extra events in \( v_i^+ \), but they are removed automatically by removing the first-order multiple.

In Figure 10f, one can observe that the first internal multiple (pointed at by the top arrow) is already attenuated beyond sample 276 – \( n_e + 1 \), but it is not yet completely attenuated before sample 276 – \( n_e \). The first 276 – \( n_e \) samples belong to \( v_i^+ \), where the information on attenuation of the internal multiple is constructed, whereas samples from 276 – \( n_e + 1 \) onward belong to \( u_i^- \), where the multiple is already attenuated. The constant-time cross section (for all lateral positions) in \( k_{-1;i} \) at sample 276 is stored in the final output \( R_t \) at sample 276. In the \( v_i^+ \) part (between samples 1 and 276) of \( k_{-1;i} \), the second reflector has its local reflection coefficient as amplitude, whereas in the \( u_i^- \) part (from sample 276 onward) it has its physical amplitude with two-way transmission effects.

**Different time instances**

In Figure 11, the Marchenko equations are solved for different time samples \( i \) and it is possible to investigate how \( k_{-1;i} \) changes for larger sample numbers. It is observed that, not only at sample \( i \) but also before and beyond \( i \), the events related to internal multiples are attenuated. Sample point 276 corresponds to the arrival time

![Figure 10. Updates for \( k_{-1;i} \) for a focal time at sample \( t_2 = 276 \) after \( i \) iterations. The arrow indicates the first- and second-order internal multiple between the first and second reflector.](image-url)
of the third reflector. The times in Figure 11a–11d are all before sample 276, and we do not observe a change in the number of events. However, going from sample 246 (Figure 11a) to 276 (Figure 11d), one can see that the multiple, arriving in time between the second and third reflectors, gets more and more attenuated at larger and larger offsets. This also explains the success of the fast algorithm: to compute the solution in the next time sample, there is only a small change needed and a few iterations are sufficient to solve for the multiple attenuation at higher offsets. When the sample time $i_2$ passes the arrival time of the third reflector, a nonphysical event (pointed out by an arrow in Figure 11e–11h) appears just below the arrival time of the second reflector. This nonphysical reflector is the annihilator event in $v_{ij}$ that compensates all internal multiples created between the second and third reflector. The cancelation of the internal multiples is observed at larger time samples (Figure 11e–11h); all internal multiples related to the third reflector are canceled out.

Figure 12a and 12b shows sketches of the situation in which the time instant $i_2$ corresponds to a depth above or below the third reflector, respectively. The event that compensates all internal multiples related to the second reflector (the green arrow in Figure 12a) coincides in traveltime with the reflection of the second reflector and also compensates for the transmission loss of the reflection from the second reflector. The internal multiples related to the third reflector are compensated by the red-arrow event (Figure 12b) which coincides with the reflection time of the third re-

---

**Figure 11.** The term $k_{i_2}^{-1}$ after 32 iterations with different time instants $i_2 = 246$ to $i_2 = 316$ with steps of 10 samples. From each panel, a constant-time cross section is selected at $i_2$ and all of these cross sections make up the multiple-free data. The arrows point to an event that compensates all internal multiples created between the second and third reflectors.

**Figure 12.** Compensation of internal multiples by events (colored lines) that are created by the Marchenko method, applied for a point (a) above and (b) below the third reflector. The three reflectors are numbered from top to bottom.
of a reflector, there is a decision to be made where to put the reflection of that reflector. Setting the truncation time to \( t_2 - \epsilon \), the time instant \( t_2 \) is correctly obtained in \( u^i \). Changing the truncation time from \( t_2 - \epsilon \) to \( t_2 + \epsilon \), the time instant \( t_2 \) is correctly obtained in \( v^i \) instead of in \( u^i \). It is the time duration of the source wavelet that allows us to make this choice. By taking \( t_2 - \epsilon \), an error is introduced in \( v^i \) and hence \( u^i \) is correct at \( t_2 \), whereas by taking \( t_2 + \epsilon \) the error is in \( u^i \) and \( v^i \) will be correct at \( t_2 \).

The transmission-compensated (T-MME) scheme retrieves primary reflections with local reflection coefficients, whereas, in the regular (MME) scheme, the primary reflections keep their two-way reflection coefficients that include transmission losses. The local reflection retrieval of the T-MME scheme is exact for a horizontally layered medium, but, in laterally varying media, it is approximately true (Zhang et al., 2019). The only computational difference between the T-MME and MME schemes is the position of the time-truncation window.

CONCLUSION

In this paper, we have demonstrated step by step the correct and effective elimination of internal multiple reflections from the acoustic reflection response using the truncation time instant as a free...
parameter. With a simple four-layer model, we showed that in a 1.5D model MME retrieves the primary reflections with their physical amplitude whereas T-MME retrieves the primary reflections with their local reflection coefficient. The examples also show that, when a first-order internal multiple is eliminated, the whole train of multiples associated with that first-order multiple is eliminated as well. This is achieved by a single multiple annihilator that is computed from only primary reflections in the data. The compute kernels of the two schemes are identical and consist of a matrix vector product that alternates between a time correlation and a time convolution. Exact and efficient computation of the discrete correlations and convolutions is done with an FFT routine. The only difference between MME and T-MME is the implementation of the time truncation. MME excludes a chosen time window before the truncation time instant and obtains the result after the scheme has converged by evaluating the equation at the truncation time instant and obtains the result directly from the value at the truncation time instant when the scheme has converged. For each truncation time instant, the iterative scheme computes the multiple annihilators and the corresponding reflection output. The current standard implementation takes every data time sample as a truncation time; the faster implementation can take larger time steps for the truncation time by exploiting the finite frequency bandwidth of the data.

ACKNOWLEDGMENTS

The research of J. Thorbecke and K. Wapenaar has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant no. 742703). We would like to thank the associate editor and the three anonymous reviewers for their constructive comments and efforts to compile the code and reproduce the figures.

DATA AND MATERIALS AVAILABILITY

Data associated with this research are available and can be accessed via the following URL: 10.5281/zenodo.4271598.

REFERENCES


Marchenko multiple elimination

Lele Zhang received a doctorate degree (2019) in applied sciences from the Delft University of Technology (TU Delft), the Netherlands. Since the beginning of 2020, he has worked as a postdoctoral research fellow in the Department of Geosciences and Engineering at TU Delft. He has organized several international workshops and conference sessions and was guest editor for one special journal issue on multiple reflections in geophysics. His research interests include signal processing, wavefield propagation, and migration.

Jan Thorbecke received an M.S. (1991) and a Ph.D. (1997) in applied physics (geophysics) from the Delft University of Technology. From 1996 to 2001, he worked at Cray/SGI as a geophysical application engineer. From 2001 to 2002, he worked at Jason Geosystems, and since 2002, he has worked at HPE-Cray as an application analyst (0.8) and senior researcher at the Delft University of Technology (0.2). He has organized several conference workshops related to high-performance computing. In 2012, he has been awarded as Geophysics reviewer of the year. In 2020, he became distinguished technologist at HPE-Cray. His research interests include wavefield extrapolation, wavefield modeling, multidimensional convolution, and compute efficient algorithm design and implementation on supercomputers.

Evert Slob received a doctorate degree (1994, cum laude) in applied sciences from the Delft University of Technology (TU Delft), the Netherlands. From 1995 to 2011, he was research fellow, assistant, and associate professor, and since 2011, he has been a professor in the Department of Geosciences and Engineering at TU Delft. He was the general chair of the 10th International Conference on GPR in 2004. He served as a director of education of Civil Engineering and Geosciences from 2014 to 2018. He organized more than a dozen international workshops and conference sessions and was guest editor for 11 special journal issues on GPR and hydrogeophysics. He is a coauthor of Introduction to controlled-source electromagnetic methods published in 2019. He is a coeditor of Seismolectric exploration published in 2020. He was the SEG Editor from 2013 to 2015, and he received the Reginald Fessenden Award from SEG in 2020.