Internal multiple prediction in the time and offset domains

Andrew Iverson, Kris Innanen and Daniel Trad

ABSTRACT

Varying strategies have been implemented to predict and attenuate internal multiples (Xio et al. 2003). One data driven method to predict internal multiples uses the inverse scattering series (Weglein et al., 1997). The method predicts internal multiples where the only required inputs for the algorithm are the data itself and a search limiting parameter epsilon. It has been displayed that reformulating the calculation domain such that the input and output domains are equivalent allows for the use of a nonstationary epsilon (Innanen, 2015). The version of the algorithm in offset-time is outlined where epsilon can vary in both spatial and temporal dimensions. With the domain of calculation in offset-time this allows for the testing and determining of an epsilon schedule based on the input seismic dataset. For some geologic models in this domain a single epsilon value is insufficient for the prediction of multiples while also minimizing artifacts. Displayed is a nonstationary epsilon implementation that assists in artifact minimization.

INTRODUCTION

As a seismic wave propagates through a medium the source wavelet will be altered due to mechanisms such as attenuation, short path multiples and various other inelastic effects that will change the wavelet shape. Some of the energy that is traveling though the medium will also be scattered or reflected when there is a significant change in medium properties. If the wave is reflected off a single boundary and recorded at the surface this is termed a primary reflection. The seismic events that reflect off multiple interfaces prior to reaching the surface are termed multiples (Figure 1). It has been shown that internal multiples can negatively impact the interpretation of seismic data (Iverson, 2014). An issue in processing seismic data is removing these unwanted multiple reflections while leaving the primaries unaltered in the data.

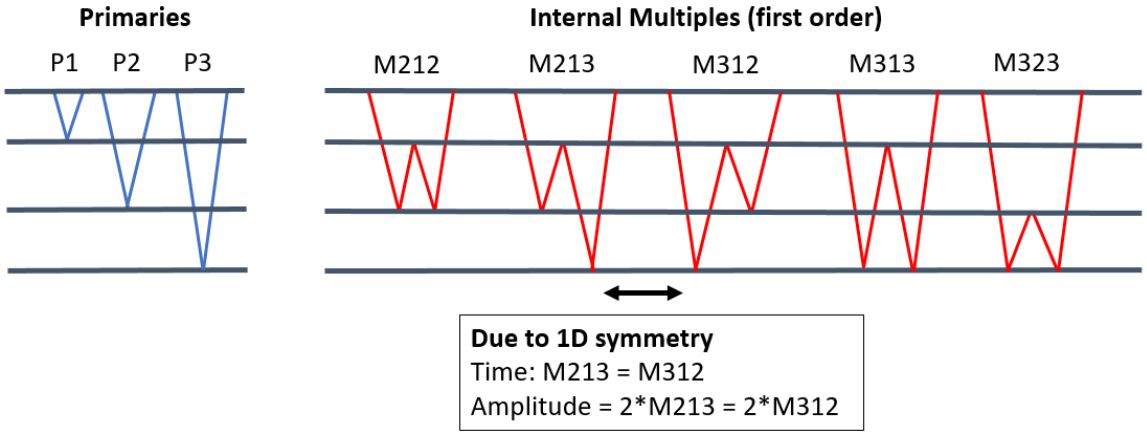

FIG. 1. (Left) Primary events for a three-layer model plus half-space (Right) First order internal multiples for the three-layer model plus half-space

Using the Inverse Scattering Series (ISS), the location in time of multiples can be predicted solely with the seismic data and a search limiting parameter epsilon. Implementing the method on land seismic data continues to be challenging in part due to the selection of the epsilon parameter. Recently the algorithm has been applied in various domains with increased success (Sun and Innanen, 2016). As the algorithm was originally written epsilon must be a single value for all space dimensions and time. There are cases though where a stationary epsilon will be insufficient (Innanen, 2017; Innanen and Pan, 2014). This could be due to steeply dipping reflections, changes in frequency content of the data or any other time variant changes to the wavelet. It has been shown how in certain 1.5D domains that epsilon can vary in the transformed spatial dimension such as wavenumber (Innanen and Pan, 2014). This limitation to the spatial domain is due to the input data to the algorithm not sharing the same space as the output. These 1.5D transform domains allow for a variation of epsilon in the spatial direction (k_g) due to the method utilizing the 1D version of the algorithm over every spatial step. Thus, for every wavenumber epsilon can change as there is no communication between offsets. The inverse scattering series equation for predicting multiples has been recast in the offset-time domain (Innanen, 2015). It was shown in 1D how improvements in the prediction can be made through a nonstationary epsilon (Innanen, 2017). With the equation in offset-time there is now flexibility to vary epsilon in any both spatial and temporal dimensions.

INVERSE SCATTERING SERIES THEORY IN OFFSET-TIME

The inverse scattering series takes the resulting wavefield and a term epsilon to predict internal multiples (Weglein et al., 1997). This was initially written in the wavenumber pseudodepth domain. Giving equation (1) below to predict interbed multiples from the seismic data alone. The term epsilon is used to account for the bandwidth of the data through implementation in the integration bounds. This sets a limit on the distance the multiple must have traveled to prevent the algorithm from predicting an event within the wavelength of a single wavelet. This method will only predict long-path multiples assuming epsilon is chosen correctly. The combination of the pseudo-depth terms in the integration limits ensures that the lower-higher-lower criteria is met and that no unwanted artifacts are in the prediction.

$$
\begin{aligned}
&b_3(k_g, k_s, \omega) \\
&= \frac{1}{(2\pi)^2} \iint_{-\infty}^{\infty} dk_1 e^{-iq_1(\epsilon_g - \epsilon_s)}\, dk_2 e^{iq_2(\epsilon_g - \epsilon_s)} \int_{-\infty}^{\infty} dz_1 e^{i(q_g + q_1)z_1} b_1(k_g, -k_1, z_1) \\
&\quad \times \int_{-\infty}^{z_1 - \varepsilon} dz_2 e^{-i(q_1 + q_2)z_2} b_1(k_1, -k_2, z_2) \int_{z_2 + \varepsilon}^{\infty} dz_3 e^{i(q_2 + q_s)z_3} b_1(k_2, -k_s, z_3), \qquad (1)
\end{aligned}
$$

Where in equation (1)

$$
q_x = \frac{\omega}{c_0} \sqrt{1 - \frac{k_x^2 c_0^2}{\omega^2}}, \qquad (2)
$$

b_3 is the interbed multiple prediction, b_1 is the prepared input data, q_x is the vertical wavenumber and ϵ is the depth below free surface of the source (s) and receiver (g), k is the Fourier conjugate variable, z_1, z_2 and z_3 are the depths chosen to satisfy the lower-higher-lower relationship and ε is the search limiting parameter (Sun and Innanen, 2014).

Multiples are predicted in the Fourier domain through the specific combinations of events which obey the lower-higher-lower relationship in the data. This is done through multiplication in the Fourier domain. In time this is equivalent to a combination of convolutions and correlations of the events that obey the selection criteria. It is shown schematically how two deeper events can be convolved relative to a shallower event, which can be correlated to mimic the equivalent multiple (Figure 2). The algorithm in the offset-time domain (Innanen, 2015) will explicitly use these convolutions and correlations to compute the multiple prediction.

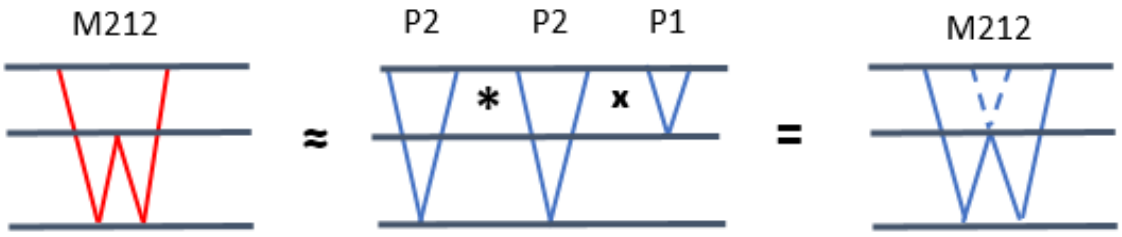

FIG. 2. Schematic displaying how a multiple can be replicated with a combination of primaries through a convolution (*) and correlation (x)

1.5D offset-time domain

Equation (1) can be simplified and reduced to a 1.5D domain by assuming a v(z) medium. This is done by assuming that the source and receiver wavenumbers are equivalent.

$$k_g = k_s, \tag{3}$$

Using this assumption alters the vertical wavenumber from equation (2) to give the following

$$q_g + q_s = 2q_g = k_z, \tag{4}$$

Giving the 1.5D Version of the algorithm

$$b_3(k_g,\omega) = \int_{-\infty}^{\infty} dz_1 e^{ik_z z_1} b_1(k_g,z_1) \int_{-\infty}^{z_1-\varepsilon} dz_2 e^{-ik_z z_2} b_1(k_g,z_2) \times \int_{z_2+\varepsilon}^{\infty} dz_3 e^{ik_z z_3} b_1(k_g,z_3), \tag{5}$$

The input data and method can be altered so that the procedure is carried out in other domains which has shown increased multiple prediction accuracy (Sun & Innanen, 2016); Innanen, 2017). The first step is to replace b_1 in terms of pseudo depth (z) with S_1 in terms of time (t) and letting $S_1(k_g,t)$ be the Fourier transform of $s_1(x,t)$ over the spatial dimension (Innanen, 2015) gives equation (6)

$$b_3(k_g,\omega) = \int_{-\infty}^{\infty} dt e^{i\omega t} S_1(k_g,t) \int_{-\infty}^{t-\varepsilon} dt_2 e^{i\omega t_1} S_1(k_g,t') \int_{t'+\varepsilon}^{\infty} dt'' e^{i\omega t''} S_1(k_g,t''), \tag{6}$$

For this the output domain is (k_g,ω) as the convolutions and correlations are applied through multiplication in the frequency domain. This can be equivalently rewritten so that

the convolutions and correlations are performed in the time domain. Giving the (k_g, t) version of the algorithm equation (7).

$$b_3(k_g, t) = \int_{-\infty}^{\infty} dt' S_1(k_g, t' - t) \int_{t'-(t-\varepsilon)}^{t-\varepsilon} dt'' S_1(k_g, t' - t'') S_1(k_g, t''), \quad (7)$$

Then by noting the remanding spatial convolutions applied in frequency this can be written fully in the time and space domain. Giving the (x, t) version of the algorithm (Innanen, 2015).

$$B_3(x, t) = \int_{-\infty}^{\infty} dx' \int_{-\infty}^{\infty} dt' s_1(x - x', t' - t) \int_{-\infty}^{\infty} dx'' \int_{t'-(t-\varepsilon)}^{t-\varepsilon} dt'' s_1(\mathrm{x}' - \mathrm{x}'', t' - t'') s_1(x'', t''), \quad (8)$$

Then by assuming there is no spatial component in equation (7)

$$k_g = 0, \quad (9)$$

The algorithm can be written and reduced to a 1D time version.

$$B_3(t) = \int_{-\infty}^{\infty} dt' s_1(t' - t) \int_{t'-(t-\varepsilon)}^{t-\varepsilon} dt'' s_1(t' - t'') s_1(t''), \quad (10)$$

The inverse scattering series can now be implemented in either time or time and space depending on the required number of dimensions. With the benefit of this being the data input and prediction output domains are identical. Next the process of applying the algorithm in practice on a discrete dataset will be discussed.

Implementing the time domain algorithm

The first step is to rewrite equation (10) so that the integration limits are applied directly to the data with the use of two Heaviside step functions (Innanen, 2015). This will be the equivalent of applying the lower higher lower constraint directly to the data. This is shown in equation (11)

$$B_3(t) = \int_{-\infty}^{\infty} dt' s_1(t' - t) \int_{-\infty}^{\infty} dt'' [O(t, t', t'') s_1(t' - t'')] s_1(t''), \quad (11)$$

Where the mask $O(t, t', t'')$,

$$O(t, t', t'') = H[t'' - (t' - (t - \varepsilon))] H[t - \varepsilon - t''], \quad (12)$$

When preforming a convolution on a discrete dataset there are multiple was to implement the operation. It can be performed by tracking indices for the set of multiplications and summations. It can also be completed through the construction of a convolution matrix where the convolution is then carried out through matrix multiplication. Displayed in Figure 3 is the matrix multiplication approach of equation (11) (Innanen, 2015). The correlation matrix (M_R), convolution matrix (M_C) and the input data (s) when multiplied through will give the prediction (im). This also has the masking matrix applied to the convolution matrix (M_C). The masking matrix is noted by the shaded region of the matrix being zeroed from equation (12).

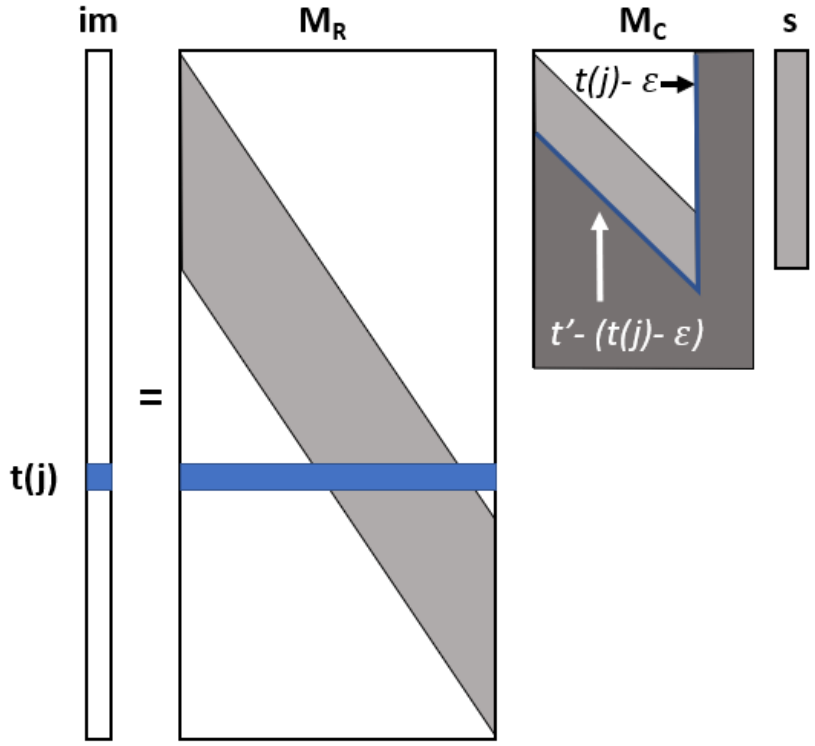

FIG. 3. Adapted from (Innanen, 2015) Visually displays the calculation of the convolutions and correlations for a given time t(j) through matrix multiplication, with the mask matrix applied to the convolution matrix M_C.

The mask noted in equation (11) is created through applying a binary matrix to the convolution matrix where the bounds on the mask will ensure that the lower-higher-lower criteria is met (Figure 4). Note that the internal multiple prediction is competed for each time step as the mask changes relative to the given time to ensuring the lower-higher-lower criteria is obeyed for all times.

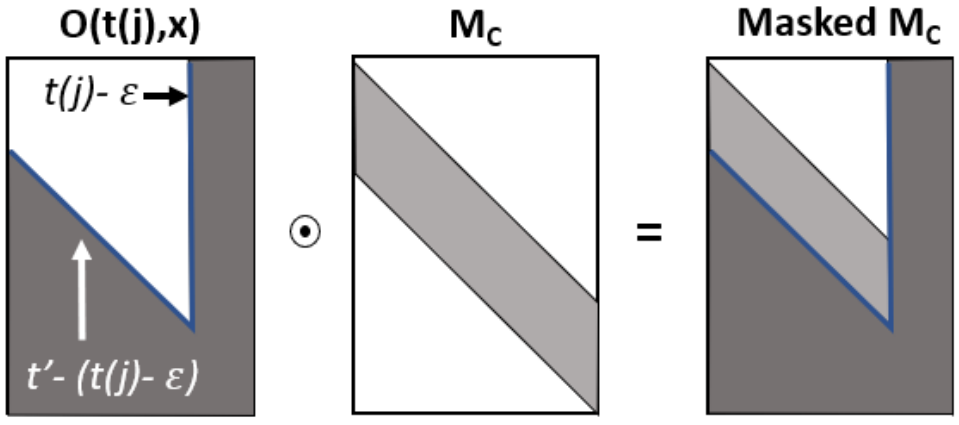

FIG. 4. Adapted from (Innanen, 2015) displays the mask matrix applied to the convolution matrix where the shaded region is set to zero and the bounds are determined such that the lower-higher-lower criteria is met for the given *t(j)*

For the (x,t) case, with the addition of the spatial dimension the process of applying the matrix multiplication is largely the same. Now the equivalent of a 2D convolution in both offset and time must be performed. The correlation and convolution matrixes are now block matrices to compute both the spatial and temporal convolutions (Innanen, 2015) (Figure 5). The input trace ($s_1(x,t)$) is also altered so that it is now a stacked 1D vector for all offsets, similarly for the prediction output vector (im). The masking matrix is applied in a similar manner to the 1D case but for every block in the convolution matrix. For the multidimensional case the prediction is also calculated for a single time as the mask matrix will be change for every time step (Innanen, 2015).

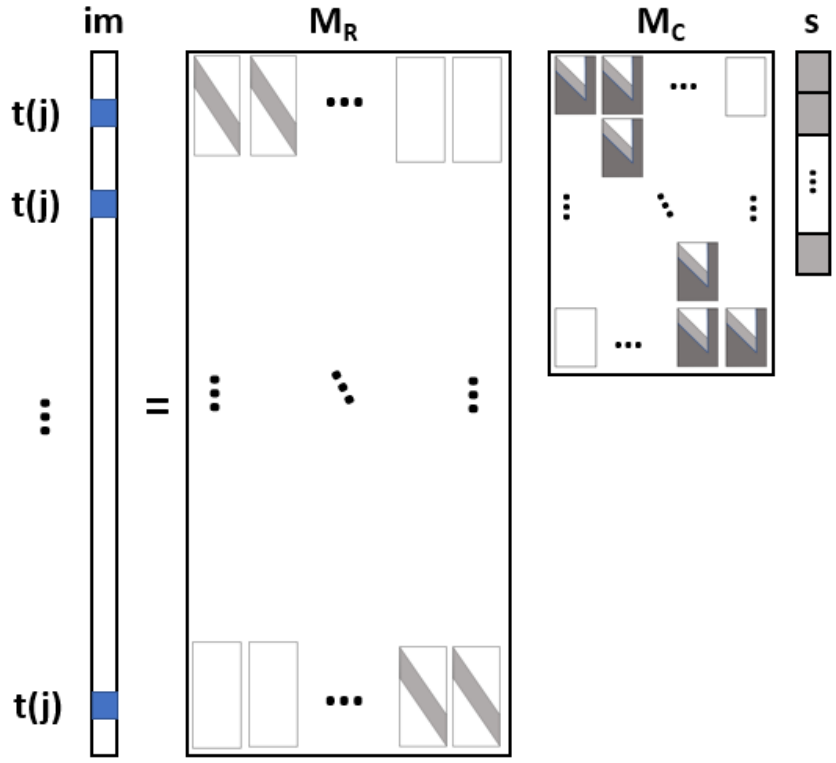

FIG. 5. Adapted from (Innanen, 2015) Displays the prediction algorithm for the (x,t) case for a given time t(j) calculated for all offset with the convolution and correlation matrix, where the mask matrix is applied to each convolution matrix in the block matrix M_C. this is applied to the (x,t) trace (s) which has been stacked into a single column vector

In practice with the addition of the spatial dimension relative to the 1D case these matrices become large, to the point where it becomes difficult to implement due to computer memory limitations. In practice these ideas of masking matrixes and convolution matrices can be implemented with the use of 2D convolution functions (e.g. conv2 in MATLAB). This will allow the calculation of the 2D convolution without the requirements of storing the entire matrix in memory. Pseudo code of how this can be implemented is displayed in Figure 6.

```
for it = tstart:tend {
   tindex = it-epsilon
   mask= 1
   mask(tindex:tend,x) = 0

   ms1= s1.*mask % apply mask to data b1
   cms1 = conv2(ms1,ms1) % 2D convolution with masked data
   b3(it,x) = conv2(cms1,s1R)  % 2D convolution with cmb1 and time reversed data (s1R)
}
```

FIG. 6. Pseudo code displaying the implementation of the (x, t) case with the use of 2D convolution functions for a stationary epsilon. The mask is applied in a similar manner where all values are set to zero given that they are either below the calculation time (it) and epsilon number of samples above (it).

Since the mask matrix is calculated for every time slice and as stated previously the input and output domains are both (x, t) epsilon can be nonstationary. If epsilon is varied strictly in the time dimension this is completed by having epsilon as a function of time for all offsets. If epsilon is to vary in both time and space, then a 2D epsilon schedule is built and the mask which was previously displayed will now vary with respect to offset. With the epsilon model matching the size of the input data. This is applied directly to the input data for each time step shown in Figure 7.

```
for it = tstart:tend {
   mask= 0
   for ix = xstart:xend {
      for iit = tstart:it-epsilon(ix,it) {
         mask(itt,ix) = 1
      }
   }

   ms1= s1.*mask % apply mask to data b1
   cms1 = conv2(ms1,ms1) % 2D convolution with masked data
   b3(it,x) = conv2(cms1,s1R)  % 2D convolution with cmb1 and time reversed data (s1R)
}
```

FIG. 7. Pseudo code displaying the implementation of the (x, t) case with the use of 2D convolution functions for a nonstationary epsilon in both time and space dimensions

This is also displayed schematically (Figure 8) where for a given calculation time there is a single epsilon for all offsets or one that varies with offset.

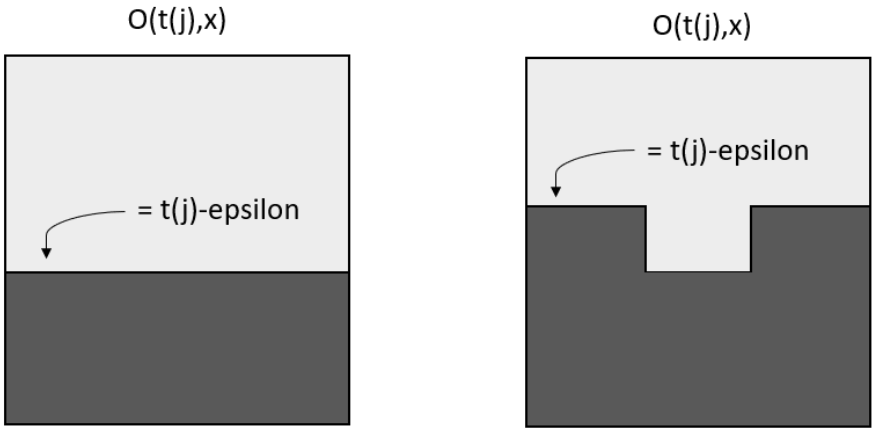

FIG. 8. Visually displays mask matrix applied to the data (s) prior to the 2D convolution function where the bounds are determined such that the lower-higher-lower criteria is met for the given *t(j)* for (Left) epsilon constant with offset (Right) An example of epsilon varying with offset with two epsilon values with a harsh cutoff

OFFSET-TIME INTERNAL MULTIPLE PREDICTION

To evaluate the (x, t) domain version of the algorithm a simple geologic model is used. Using finite difference modeling the shot record was created in MATLAB using afd_shotrec from the CREWES toolbox. (Figure 9).

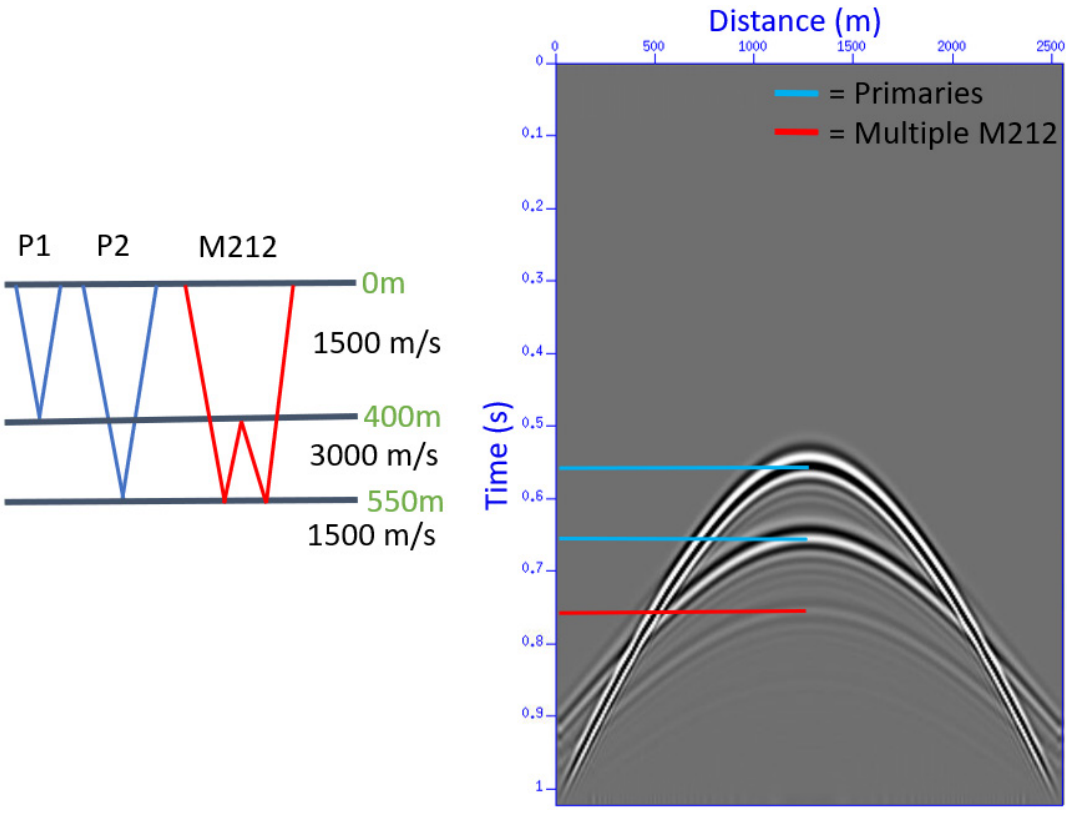

FIG. 9. (Left) Geologic model displaying velocities and depths used (Right) Shot record with two primaries and first order internal multiple defined

Using finite difference, the primaries and all orders multiples are modeled within the recorded window. The 2D model was spatially sampled every 10m and a temporal sample rate of 0.002s, this created a grid that is 512x256 samples. The seismic shot record was created by convolving the result with a 30Hz Ricker wavelet. This geologic model has created a shot record which has a strong first order internal multiple. To use any version of the algorithm at a minimum a stationary epsilon must be selected. Initially two values of epsilon are evaluated, a value of 30 and 70 (Figure 10).

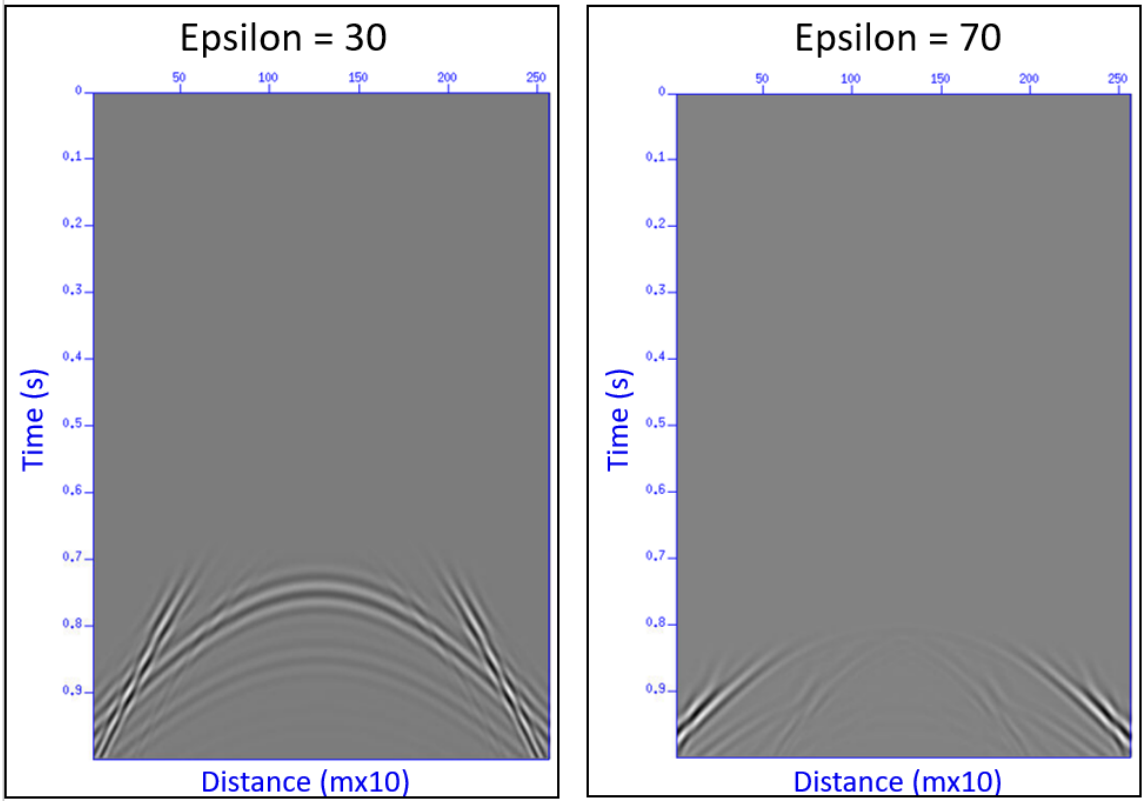

FIG. 10. (Left) offset-time multiple prediction with epsilon=30 (Right) offset-time multiple prediction with epsilon=30

For this geologic model with an epsilon of 30 the first order internal multiple has been predicted along with higher order multiples. There is also a steeply dipping event predicted which is an unwanted artifact from the prediction. This issue has been previously noted when predicted in the (k_g, z) domain (Innanen and Pan, 2014). The artifact in that domain was due to the steeply dipping event becoming broad in the transformed domain at larger values of k_g. Using an epsilon value of 70 has diminished this artifact but epsilon has also become sufficiently large that it now impacts the algorithms ability to predict the internal multiples. It was previously shown in the (k_g, z) domain how a k_g varying epsilon could mitigate this issue leading to the next epsilon schedule test (Figure 11).

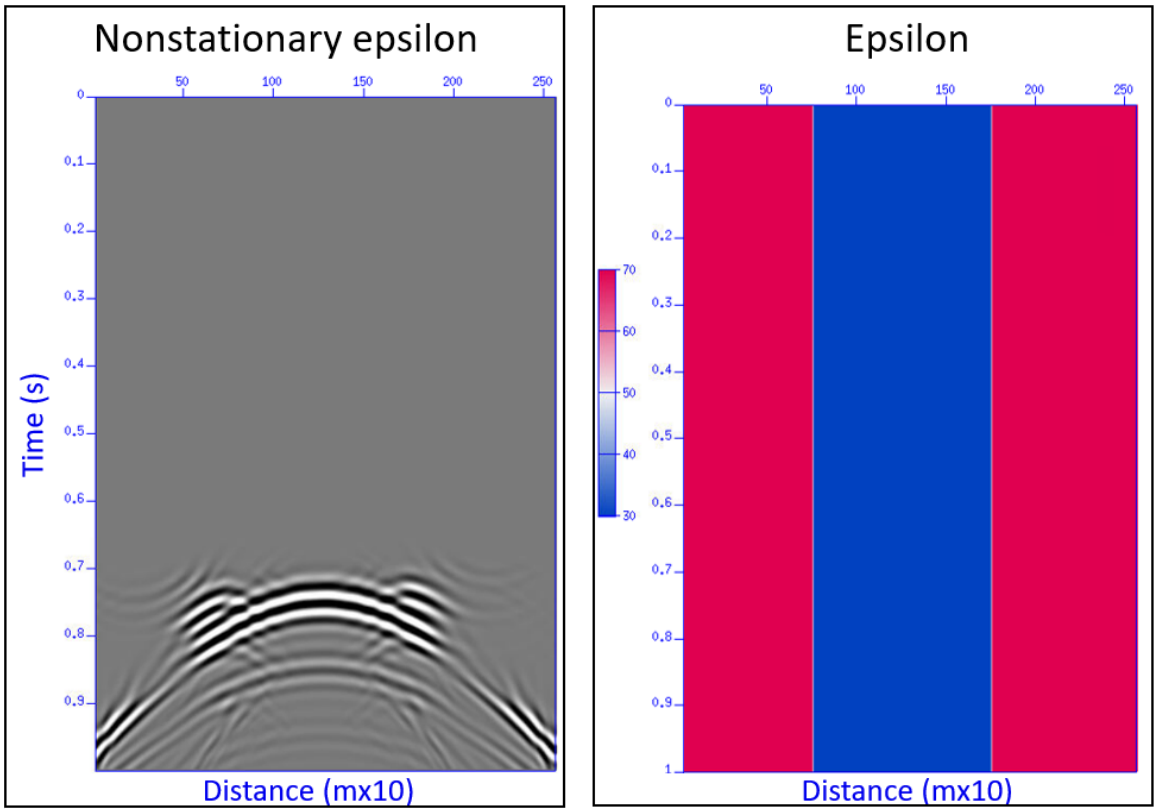

FIG. 11. (Left) offset-time multiple prediction with spatially varying epsilon (Right) epsilon schedule used for prediction with harsh cutoff

The two epsilon values of 30 and 70 are combined to produce a spatially variant epsilon. The internal multiples are visible in the prediction but a new artifact has now also been introduced. This epsilon schedule was created with a sharp cutoff from the value of 30 to 70 over a single sample point. This has assisted in removing the steeply dipping artifact previously noted but has also introduced this new issue. With the sharp contrast in epsilon there appear to be two additional internal multiples that are now also being predicted on both sides of the first order multiple, which are not present in the input dataset. Next to attempt to mitigate this new issue an epsilon schedule with a smooth linear trend from the 30 to 70 is utilized (Figure 12).

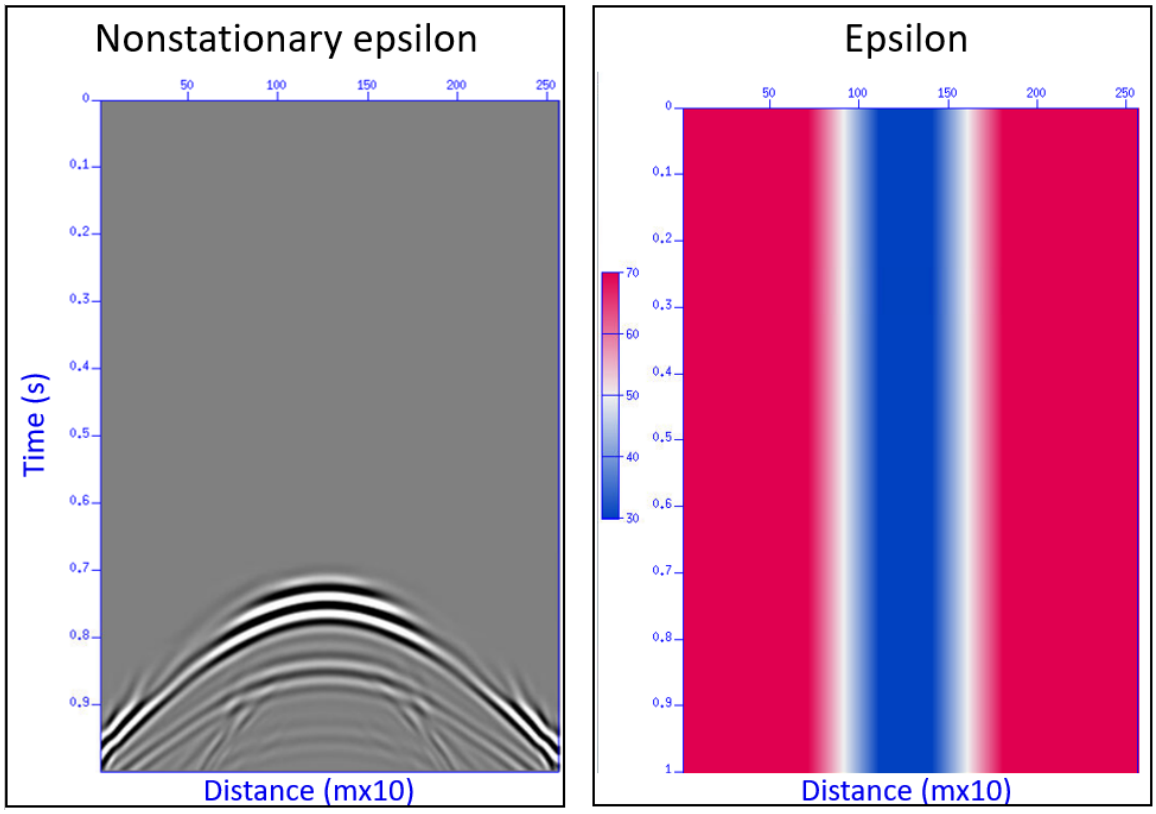

FIG. 12. (Left) offset-time multiple prediction with spatially varying epsilon (Right) epsilon schedule used for prediction with linear taper

This has improved the prediction of the internal multiples relative to the previous harsh cutoff model. The strong first order multiple along with the higher order multiples are predicted but with artifacts remaining. The issues with the prediction near the location of the original steeply dipping artifact remain with this model that only varies in space. The last epsilon model to be displayed varies both in space and time. This was designed to overlap the steeply dipping event in the input data and have the liner taper to minimize the artifacts that appeared to arise from a harsh cutoff (Figure 13).

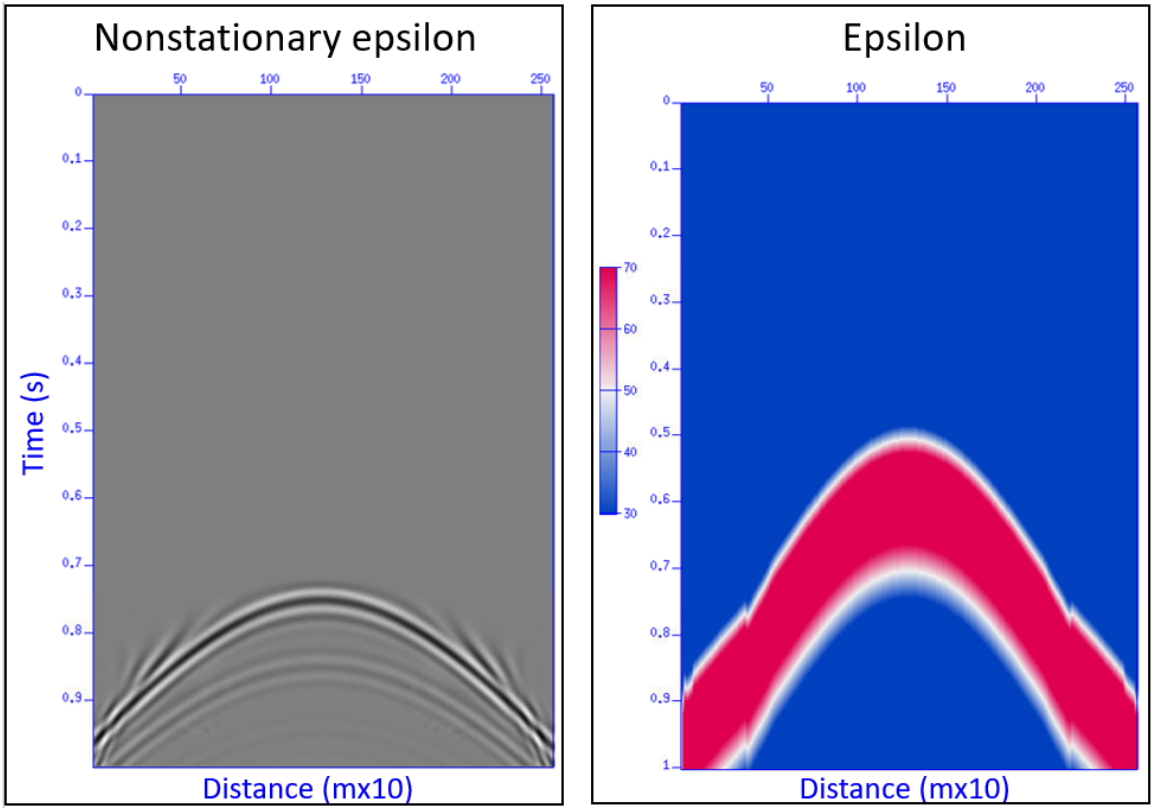

FIG. 13. (Left) offset-time multiple prediction with spatially varying epsilon (Right) epsilon schedule used for prediction varying in both offset and time

This final schedule which varies in both offset and time is capable of both predicting the multiples in the data and minimizing the artifacts. Though there is still some residual artifact present from the steeply dipping event, improvements could possibly be made to adjust epsilon to optimize for this model. With this full flexibility to vary epsilon in any dimension. This now introduces future study of how to derive the optimum epsilon schedule for a given input model.

CONCLUSIONS

The method using the inverse scattering series for internal multiple prediction has been adapted to be compute in offset-time (Innanen, 2015). With this change in computational domain comes the ability to utilize a non-stationary epsilon. Displayed was an example of how varying epsilon can improve the prediction. This is completed on a model where a stationary epsilon will be insufficient. Also displayed is the how a sharp epsilon boundary can cause new artifacts to be present in the prediction. Now geologic models can be tested with the offset-time algorithm with the flexibility in epsilon schedule creation.

ACKNOWLEDGEMENTS

The authors would like to thank the sponsors of CREWES for the support of this work. This work was also funded NSERC through the grant CRDPJ 461179-13. Andrew Iverson would also like to thank QEII, NSERC and SEG for their support.

REFERENCES

Innanen, K. A., and Pan, P., 2014, Large dip artifacts in 1.5D internal multiple prediction and their mitigation, CREWES Annual Report, 26.

Innanen, K. A., 2015, Time domain internal multiple prediction, CREWES Annual Report, 27.

Innanen, K. A., 2015, A nonstationary search parameter for internal multiple prediction, CREWES Annual Report, 27.

Innanen, K. A., 2017, Time- and offset-domain internal multiple prediction with nonstationary parameters, Geophysics, 82(2), V105-V116.

Iverson, A., 2014, The impact of interbed multiples on the inversion and interpretation of prestack data, CSEG Recorder, 39(1).

Sun, J., and Innanen, K. A., 2014, 1.5D internal multiple prediction in the plane wave domain, CREWES Annual Report, 26.

Sun, J., and Innanen, K. A., 2016, Literature review and discussions of inverse scattering series on internal multiple prediction, CREWES Annual Report, 28.

Weglein, A. B., Gasparotto, F. A., Carvalho, P. M., and Stolt, R. H., 1997, An inverse-scattering series method for attenuating multiples in seismic reflection data, Geophysics, 62(6), 1975-1989.

Xiao, C., Bancroft, J. C., Brown, J. R., and Cao, Z., 2003, Multiple suppression: A literature review, CREWES Annual Report, 15.