

Time domain internal multiple prediction

Kris Innanen

ABSTRACT

The standard 1.5D internal multiple prediction algorithm generates output in the (k_g, ω) domain. Through some manipulation versions in the (k_g, t) and (x_g, t) domains are derived, along with the 1D time domain formula. Algorithms for calculating the predictions in these domains centre on partial convolutions in vertical time or pseudodepth. This is implemented by forming masking matrices to overlie standard dispersion or convolution matrices. The result are some intuitive codes which produce very clean predictions, even in the presence of gathers prior to deconvolution.

INTRODUCTION

The CREWES project is working towards successful land application of inverse scattering series based internal multiple prediction and removal technology (for a complete review of the nature and classification of multiples, and a review of methods for their suppression, see Weglein and Dragoset, 2005). This paper, which is one of several on the subject this year (see also Sun and Innanen, 2015b; Keating et al., 2015), concerns the derivation and testing of the algorithm in several new domains. In a companion paper (Innanen, 2015), these domains are analyzed for their practical ability to increase the precision with which multiples are predicted in event-rich environments, which is a key issue in land applications.

Several classes of wave equation-based removal of internal multiples exist (Weglein et al., 1997; Jakubowicz, 1998; Berkhout, 1999), however the inverse scattering series internal multiple suppression algorithm (Araújo, 1994; Weglein et al., 1997, 2003; Otnes et al., 2004; Ramírez and Weglein, 2005; Pan and Innanen, 2014; Zou and Weglein, 2015) remains the high bar for predicting the arrival time and approximate amplitude of internal multiples in the absence of subsurface velocity or structural information.

Land application remains challenging, for reasons outlined by Luo et al. (2011). Noisy traces with proximal and/or interfering primaries and multiples are common; on occasion the pre-subtraction prediction sections themselves are informative, but too noisy for subtraction to be advisable (Reshef et al., 2003; Hernandez and Innanen, 2014). However, the possible impact on the interpretation of land sections of even a small up-tick in the precision of multiple removal continually fosters investigation of promising workflows (Fu et al., 2010; Wu et al., 2011; Sonika et al., 2012; Ras et al., 2012; de Melo et al., 2014, 2015).

A promising line of research is to seek optimum domains in which the basic prediction mechanisms are carried out. The automated search for, and combination of, sub-events in a data record is fixed to occur in the pseudo-depth or vertical travel time domains (Weglein et al., 2003), but the output domains, i.e., the experimental variables on the left-hand side of the formula, can be varied quite widely. The standard form of the algorithm has the

prediction emerging in the wavenumber/frequency domain, but formulations in the τ - p domain (Coates and Weglein, 1996) may have advantages in terms of reduction of artifacts (Sun and Innanen, 2015a).

The output domain is critical also because it restricts and defines the variability we may assign to the search limiting parameter ϵ (whose importance was first discussed by Coates and Weglein, 1996). For instance, high angle noise in 1.5D multiple prediction has been shown to be suppressed by setting the parameter $\epsilon \propto k_g$, rather than giving it a fixed value (Innanen and Pan, 2015). This was possible with the standard form of the prediction algorithm because k_g is one of the output variables of the formula.

In this paper we will derive forms for 1.5D internal multiple prediction in several output domains, using the standard (k_g, ω) domain as a starting point, and provide some numerical examples of some of them in action. All of the formulas are completely equivalent, in the sense that the output of the (k_g, t) algorithm could be produced by calculating the (k_g, ω) domain output and inverse Fourier transforming over ω . We will leave for the companion paper the problem of justifying the effort of deriving such forms.

SPACE-TIME AND WAVENUMBER-TIME DOMAIN PREDICTION FORMULAS

Let $s(x_g, t)$ represent a single shot record of data acquired over a laterally invariant medium, with t being the time and x_g being the lateral geophone position. Assume that s has been preprocessed to remove all downgoing and/or surface-propagating wave components, but that it has not been deconvolved, i.e., it has an intact causal wavelet. Let s now to be used as input to the inverse scattering internal multiple prediction algorithm. The original formula (as presented by Weglein et al., 1997, 2003), when reduced to the 1.5D case, can be written:

$$\text{IM}_{k\omega}(k_g, \omega) = \int_{-\infty}^{\infty} dx e^{ik_z x} b_1(k_g, x) \int_{-\infty}^{x-\epsilon_1} dy e^{-ik_z y} b_1(k_g, y) \int_{y+\epsilon_2}^{\infty} dz e^{ik_z z} b_1(k_g, z), \quad (1)$$

where k_g is the Fourier conjugate to the lateral geophone position variable x_g , and z is pseudo-depth, ϵ is the “search limiting” parameter (selected in order to enforce a minimum separation between any two events to be combined), and where the input $b_1(k_g, z)$ is a function of s and a single homogeneous wave velocity. The output prediction is the in the (k_g, ω) domain, where ω is the temporal angular frequency. Our first step is to replace b_1 with s and z with t , which has no effect on the output of equation (1). Letting $S(k_g, t)$ be the Fourier transform of $s(x_g, t)$ over x_g , we have

$$\text{IM}_{k\omega}(k_g, \omega) = \int_{-\infty}^{\infty} dt e^{i\omega t} S(k_g, t) \int_{-\infty}^{t-\epsilon_1} dt' e^{-i\omega t'} S(k_g, t') \int_{t'+\epsilon_2}^{\infty} dt'' e^{i\omega t''} S(k_g, t'') \quad (2)$$

This is the (k_g, ω) version of the algorithm. Recognizing the products of the integrals over time as representing modified convolutions and correlations (see Appendix A), we re-write equation (2) as

$$\text{IM}_{kt}(k_g, t) = \int_{-\infty}^{\infty} dt' S(k_g, t' - t) \int_{\alpha(t, t')}^{\beta(t)} dt'' S(k_g, t' - t'') S(k_g, t''), \quad (3)$$

where

$$\begin{aligned}\alpha(t, t') &= t' - (t - \epsilon_2) \\ \beta(t) &= t - \epsilon_1.\end{aligned}\quad (4)$$

This is the (k_g, t) version of the algorithm. Further, recognizing that the remaining products of $S(k_g, t)$ represent convolutions in space, we write

$$\mathbf{IM}_{xt}(x, t) = \int dx' \int dt' s(x - x', t' - t) \int dx'' \int_{\alpha(t, t')}^{\beta(t)} dt'' s(x' - x'', t' - t'') s(x'', t''), \quad (5)$$

which is the (x_g, t) version of the algorithm. The 1D normal incidence version of the algorithm, which accepts a single trace $s(t)$ as input, is produced by setting $k_g = 0$ in equation (3):

$$\mathbf{IM}_t(t) = \int_{-\infty}^{\infty} dt' s(t' - t) \int_{\alpha(t, t')}^{\beta(t)} dt'' s(t' - t'') s(t''). \quad (6)$$

Equations (2), (3), and (5) are equivalent formulas and all three in principle produce the same output. However, the fact that the output domains differ will introduce some important practical differences between the three, in particular in the flexibility with which the parameter ϵ can be selected and tuned.

ALGORITHMS

We next take up efficient approaches to evaluating the prediction formulas in equations (3)–(6). The algorithm used to evaluate the 1D formula in equation (6) is the core of each 1.5D algorithm also, so we will consider it in detail.

1D time domain algorithm

Equation (6) can be rewritten in terms of a mask operator O :

$$\mathbf{IM}_t(t) = \int_{-\infty}^{\infty} dt' s(t' - t) \int_{-\infty}^{\infty} dt'' [O(t, t', t'') s(t' - t'')] s(t''), \quad (7)$$

which consists of two Heaviside functions H in a product:

$$O(t, t', t'') = H[t'' - \alpha(t, t')] H[\beta(t) - t'']. \quad (8)$$

The mask suppresses contributions from t'' values below $\alpha(t, t')$ and t'' values above $\beta(t)$, regions which represent violations of the lower-higher-lower rule and which if included would introduce artifacts. Momentarily setting $O = 1$, the formula reads

$$\mathbf{IM}_t(t) + \text{artifacts} = \int_{-\infty}^{\infty} dt' s(t' - t) \int_{-\infty}^{\infty} dt'' s(t' - t'') s(t''). \quad (9)$$

The right hand side is now precisely a convolution followed by a correlation of the trace with itself. Let us use this form to build up an algorithm for formula (6). Equation (9) can be expressed in discrete form in terms of convolution and correlation matrices:

$$\mathbf{im} + \text{artifacts} = \mathbf{M}_R \mathbf{M}_C \mathbf{s}, \quad (10)$$

where

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix}, \quad (11)$$

is the $(N \times 1)$ discretized input trace, and where

$$\mathbf{M}_C = \begin{bmatrix} s_1 & 0 & 0 & \dots & 0 & 0 \\ s_2 & s_1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & 0 & 0 \\ s_N & s_{N-1} & s_{N-2} & \dots & 0 & 0 \\ 0 & s_N & s_{N-1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & s_N & s_{N-1} \\ 0 & 0 & 0 & \dots & 0 & s_N \end{bmatrix} \quad (12)$$

is the $(2N-1 \times N)$ matrix enacting convolution of \mathbf{s} with itself, and

$$\mathbf{M}_R = \begin{bmatrix} s_N & 0 & 0 & \dots & 0 & 0 \\ s_{N-1} & s_N & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & 0 & 0 \\ s_1 & s_2 & s_3 & \dots & 0 & 0 \\ 0 & s_1 & s_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & s_1 & s_2 \\ 0 & 0 & 0 & \dots & 0 & s_1 \end{bmatrix} \quad (13)$$

is the $(3N-2 \times 2N-1)$ matrix enacting the correlation step. Figure 1 is a schematic diagram of this equation for the prediction + artifacts. White arrows are reminders that correlation matrices are convolution matrices containing time-reversed versions of the input trace.

By altering this structure we obtain an implementation of the 1D prediction algorithm. The alteration involves the convolution matrix \mathbf{M}_C being overlain by the matrix $\mathbf{O}(t(j), \epsilon)$, a discretized version of the mask operator O in equation (8). This occurs through the element-wise product \odot :

$$\mathbf{O}(t(j), \epsilon) \odot \mathbf{M}_C. \quad (14)$$

Figure 2 is an illustration of the masking of the convolution matrix. In the diagram the light-coloured regions are pass regions, where the matrix elements are equal to 1, and the dark regions are cut regions, where the matrix elements are equal to 0. \mathbf{O} is designed by calculating the two boundaries between light and dark regions, which are the lines

$$\begin{aligned} t'' &= \beta(t), & \text{which is vertical, on the right hand side, and} \\ t'' &= \alpha(t, t'), & \text{which is diagonal, at the bottom.} \end{aligned} \quad (15)$$

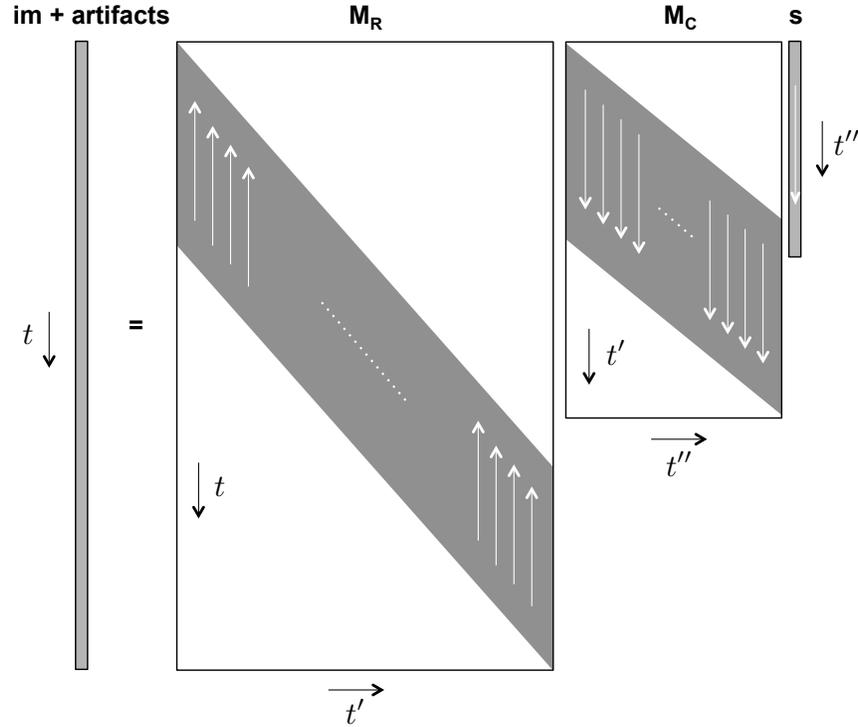


FIG. 1. Schematic illustration of convolution/correlation in matrix form.

The matrix \mathbf{O} cuts out the dark coloured portion of the convolution matrix prior to its application to the vector \mathbf{s} . The passing of convolution contributions from the upper-left chisel-shaped region is the way the lower-higher-lower rule (Weglein et al., 2003; Weglein and Matson, 1998) is imposed in time-domain prediction.

The masked convolution matrix (equation 14) then replaces the bare convolution matrix in a calculation like that of the right hand side of equation (10); the mask suppresses the *artifacts* term on the left, and a clean prediction is obtained. However, the replacement cannot occur in an otherwise unaltered version of equation (10). This is because $\mathbf{O}(t(j), \epsilon)$ is a function of the output time, $t(j)$, which means a different masking matrix is needed for each element of the output vector \mathbf{im} .

Equation (10) can be extended to accommodate this requirement by invoking matrices of higher dimension, but for simplicity we will instead break the single set of matrix products into a loop over elements of the output vector, $\mathbf{im}(j)$, and calculate each one using a slightly different masked convolution operator. The procedure is illustrated in Figure 3, in which calculations of two elements of the prediction, at $t(a)$ and $t(b)$, are examined. As we fill in the prediction vector from top to bottom, the two boundary lines as defined in equation (15) move away from the top left corner. At time $t(a)$, the pass region is the white area, and the cut region begins at the light grey boundary. As we move from $t(a)$ to $t(b)$, the pass region grows to include both the white and light grey areas, and the cut region shrinks, now consisting only of the dark grey areas.

At the j th element of the output prediction vector \mathbf{im} , we need only calculate using the

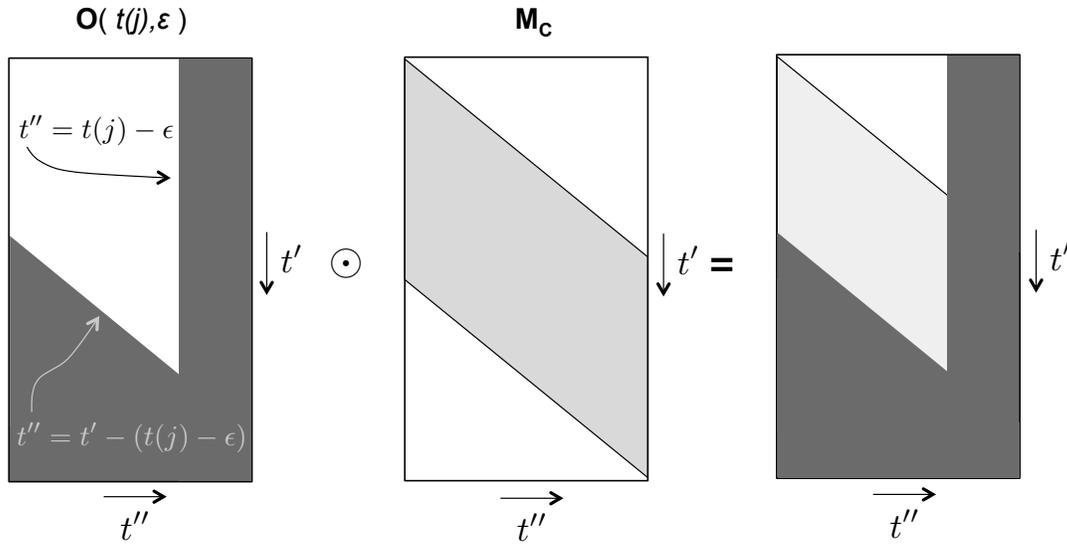


FIG. 2. A matrix $O(t(j), \epsilon)$ of ones and zeros (light and dark regions respectively) is applied in a Hadamard product \odot to the convolution matrix M_C . The result is an operator which can be applied to the input trace in accordance with the lower-higher-lower rule, appropriate for one output time $t(j)$. A slightly different composition must be calculated for each output point in the prediction.

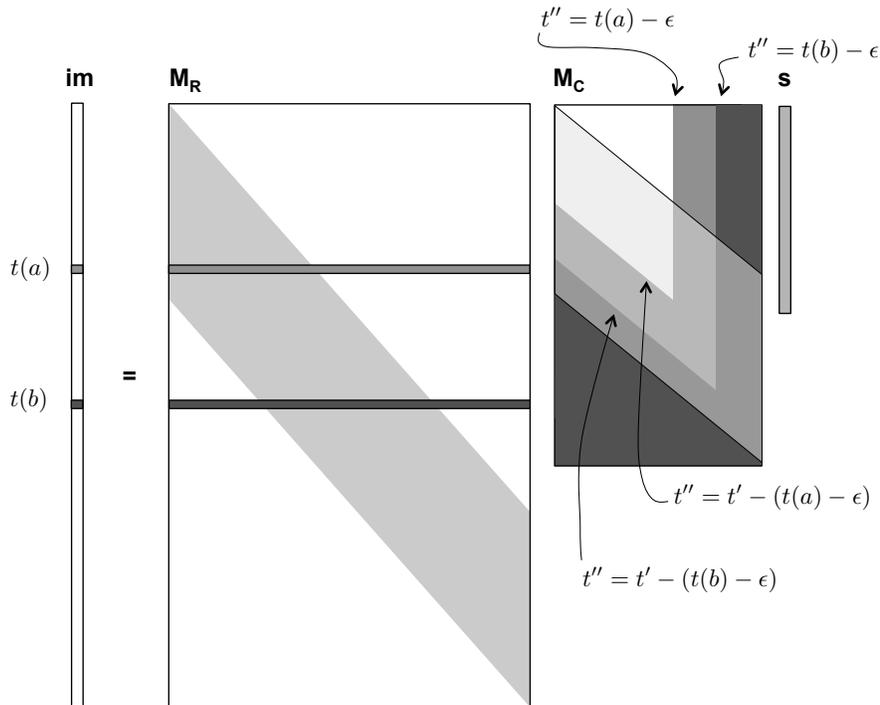


FIG. 3. Illustration of the calculation of two output points in the vector \mathbf{im} , at time indices a and b . Each output time requires a different masking operator to be overlain on M_C , with a pass region above and to the left of the lines defined by $t'' = t(a) - \epsilon$ and $t'' = t' - (t(a) - \epsilon)$ for the output at index a , and $t'' = t(b) - \epsilon$ and $t'' = t' - (t(b) - \epsilon)$ for the output at index b . The former requires a smaller pass region (white in the figure), the latter a larger (white + light grey). The algorithm loops over the output time, requiring a single row of M_R for each step. The n th masking operator can be constructed by small alterations to the $n - 1$ th operator, so computationally the loop is not expensive.

j th row of \mathbf{M}_R , since we are only computing a single output element $\text{im}(j)$. Algebraically:

$$\text{im}(j) = \mathbf{M}_R(j, \cdot) [\mathbf{O}(t(j), \epsilon) \odot \mathbf{M}_C] \mathbf{s}. \quad (16)$$

The role of the search limiting parameter ϵ is to disallow contributions from nearby points in the trace. In the time domain formula and algorithm, this takes the form of slightly shrinking the pass region from what it would be if ϵ were set to zero. This is illustrated in Figure 4, wherein the pass region with a finite ϵ is illustrated with the boundaries associated with $\epsilon = 0$ sketch in in dashed lines.

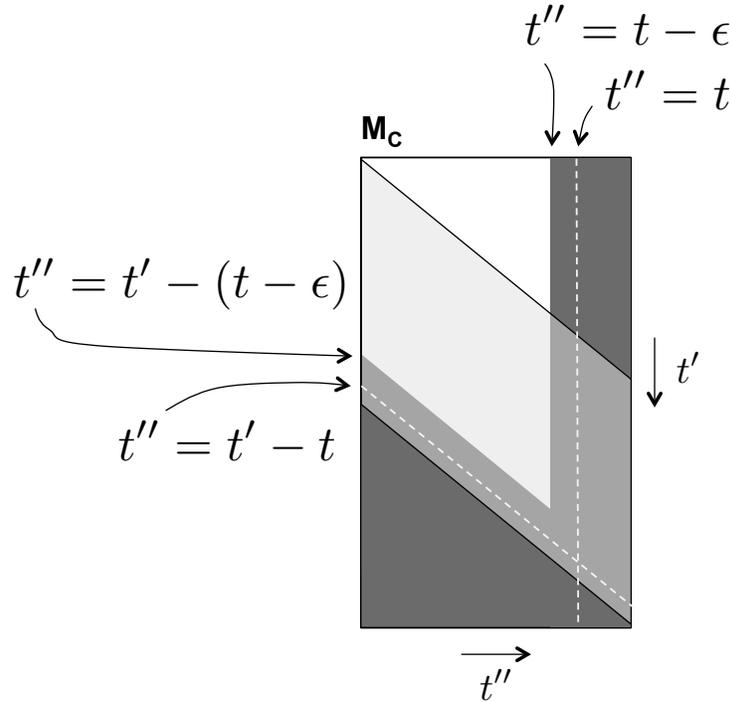


FIG. 4. An example masking operator $\mathbf{O}(t, \epsilon)$, with the boundaries of the pass region with ϵ set to zero overlain as dashed lines.

1.5D offset-time domain algorithm

The standalone 1D algorithm discussed above is also the core of the algorithms for the 1.5D wavenumber/time and space/time formulas, equations (3) and (5) respectively. The wavenumber-time algorithm makes trivial use of the 1D algorithm, repeating it once for each output k_g value. The space-time prediction is also straightforward, but it is not quite so trivial and so we will briefly develop it here.

By inspection of equation (5), again momentarily neglecting the integration limits α and β , the first step in the 1.5D space-time prediction is to autoconvolve the data, but this time across both x_g and t . Let the shot record $s(x_g, t)$ be reordered into a single column vector of concatenated traces called $\mathbf{s}^{(2)}$. This vector is the right hand element of the system illustrated in Figure 5; each vertical white arrow represents a trace, with the arrowhead being a reminder of the direction of increasing time. The matrix $\mathbf{M}_c^{(2)}$ (also containing arrows indicating increasing time) which premultiplies $\mathbf{s}^{(2)}$ in order to carry

out the autoconvolution has a block structure. This element of the system is one to the left of $s^{(2)}$ in Figure 5. Each block corresponds to the 1D convolution matrix needed to autoconvolve one of the traces within $s^{(2)}$.

The algorithm involves an outer loop over the output time, $t(j)$, with j ranging from 1 to the maximum output time sample. For a particular output time point $t(j)$, the integration limits α and β are constant, assuming that at most $\epsilon = \epsilon(t)$. Thus, at each $t(j)$, the integration limits can be reintroduced by multiplying each block of $M_c^{(2)}$ by a single masking operator $O(t(j), \epsilon)$. This is illustrated on the top right block of $M_c^{(2)}$ in Figure 5. With the masking in place, the composition of the two rightmost elements in Figure 5 takes place. The result is then in turn multiplied with a correctly-sized correlation matrix $M_R^{(2)}$, the third element from the right in Figure 5. The arrows are a reminder that the time reverse of the traces within columns of each block is needed. Finally, since the result is correct only for the j th output time $t(j)$, the appropriate elements of the result are slotted into the left hand side vector. This is repeated for all output times $t(j)$.

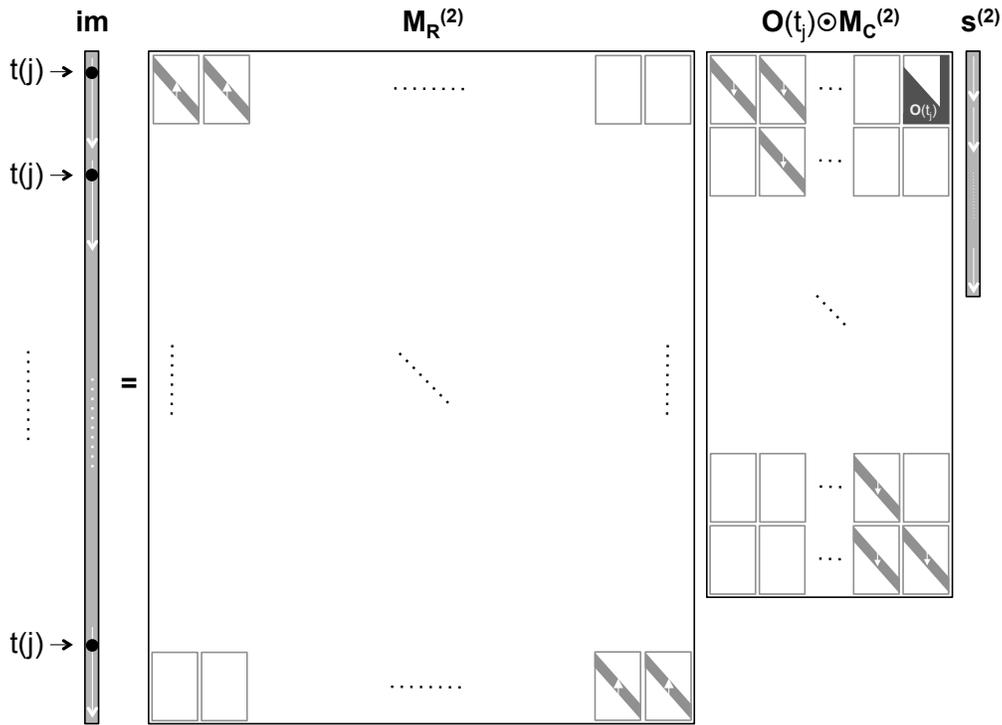


FIG. 5. The shot record $s(x_g, t)$ is reordered to form the vector $s^{(2)}$. This is multiplied by the block convolution matrix $M_c^{(2)}$ overlain by the j th masking matrix, and the result is further multiplied by the correlation matrix $M_R^{(2)}$. One of the masking matrices is illustrated on the top right block - one of these overlying each of the blocks should be visualized. The result is slotted into the output prediction vector, at elements corresponding to the j th output time. The process is repeated for all output times $t(j)$.

SYNTHETIC EXAMPLES

Here we will present some simple proof-of-concept examples for the 1D and 1.5D versions of the algorithms as developed previously.

1D examples

To test the 1D prediction we will use the implementation of Ganley's forward modelling method (Ganley, 1981) as implemented by Margrave (2015). A test velocity model, illustrated in Figure 6, is input into the forward modelling code, with source and receiver embedded at depth zero. The full wavefield is illustrated at the top of column (a) of Figure 7, and the trace extracted at the receiver location is illustrated at the bottom. In (b) and (c) the upgoing and downgoing components of the fields are illustrated. The trace at the bottom of column (b) is taken as input to the internal multiple algorithm.

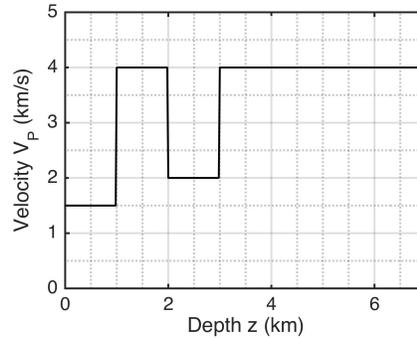


FIG. 6. Velocity model used for testing 1D time-domain internal multiple prediction.

The 1D algorithm is enacted on the extracted trace from the upgoing wave field; results are plotted in Figure 8. In black is the input trace, and in red is the prediction, shifted up for illustrative purposes. Of note: although the wavelet was left intact in the input, the prediction matches the actual multiples very well in phase and amplitude.

1.5D examples

The 1.5D space-time domain prediction algorithm is tested as follows. A two interface velocity model separating piecewise linear intervals is input into a ray tracing code, producing travel time curves for the two reflected primaries and the first order internal multiple (Figure 9). This allows a small split spread shot record to be quickly modelled, with amplitudes neglected (Figure 10a). The results of the 1.5D space/time prediction are illustrated in Figure 10b and c, the latter being a single trace extracted at zero offset.

CONCLUSIONS

On the assumption that the domain of output of the inverse scattering series internal multiple prediction technology could have important practical consequences for (especially) land applications, we have shown how to derive 1.5D formulas for prediction in the wavenumber/time, offset/time, and (in 1D) time domains. Further, we have shown how to assemble efficient algorithms which make use of commonly available signal processing functions (e.g., the construction of convolution or dispersion matrices, lexicographical reordering of matrices, etc.). Illustrative numerical examples set the stage for the efforts discussed in the companion paper, in which the practical benefits of the time domain are further developed.

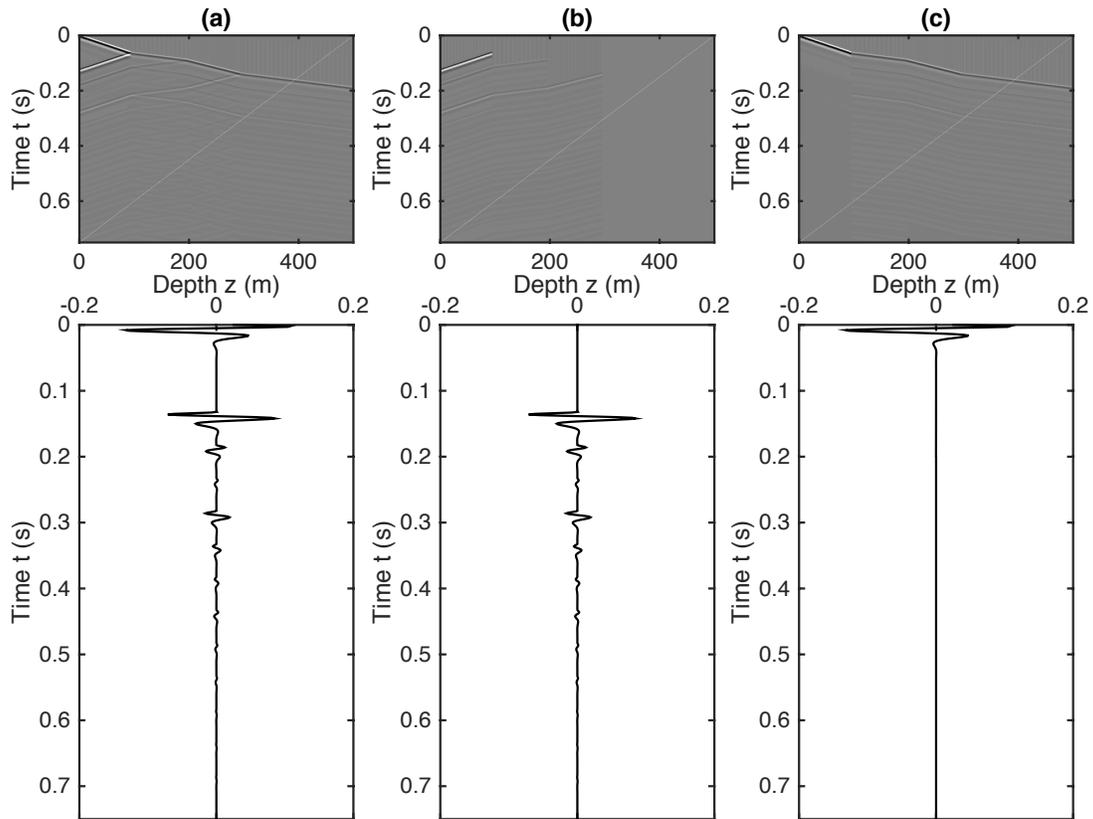


FIG. 7. 1D VSP modeling output. (a) VSP data set and extracted trace, full wavefield; (b) VSP data and extracted trace, upgoing wavefield; (c) VSP data and downgoing wavefield. The extracted trace in column (b) is input to the internal multiple prediction algorithm.

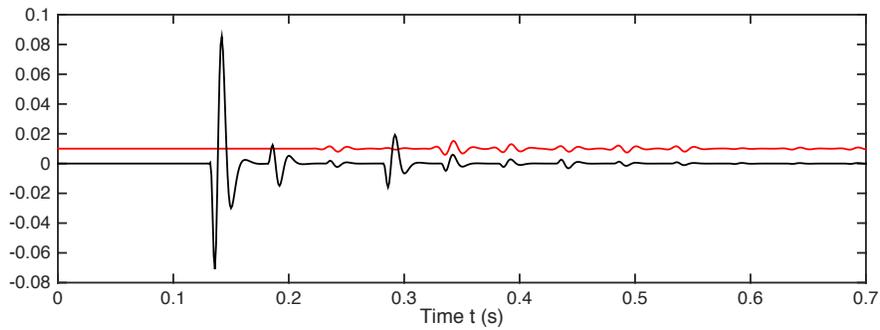


FIG. 8. Input trace (black) vs predicted multiple (red).

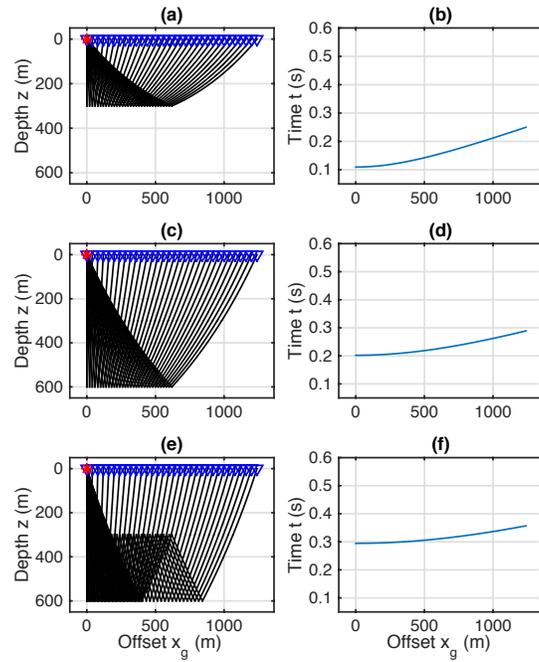


FIG. 9. Ray paths and travel times through layered $v(z)$ model. (a)-(b) Shallow primary; (c)-(d) deep primary; (e)-(f) 1st order internal multiple.

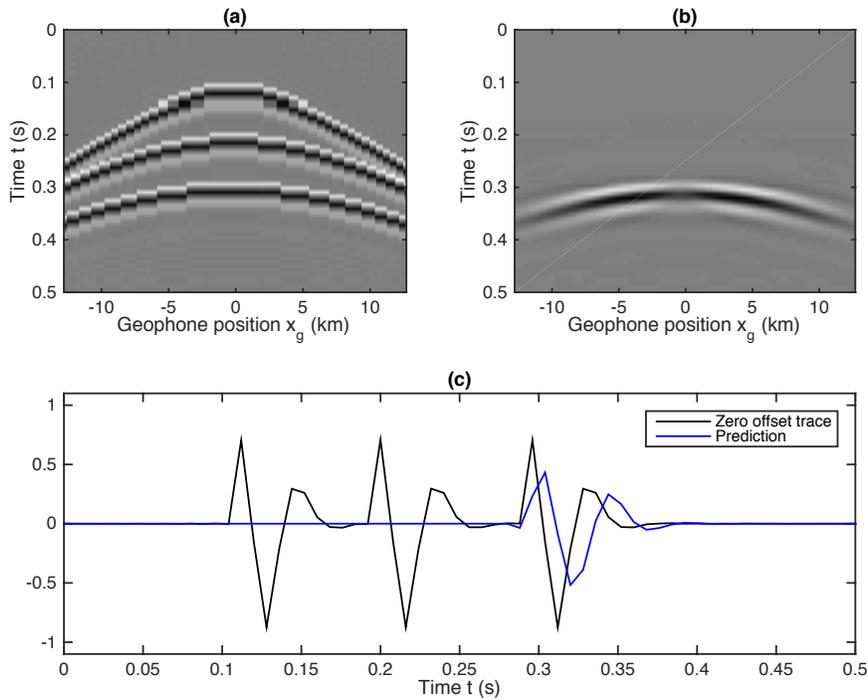


FIG. 10. (a) Input shot record containing two primaries and one multiple; (b) output of 1.5D space/time prediction, full record; (c) output of 1.5D space/time prediction, zero offset trace extracted (black original, blue prediction).

ACKNOWLEDGEMENTS

We thank the sponsors of CREWES for continued support. This work was funded by CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 461179-13.

APPENDIX A: TIME DOMAIN PREDICTION FORMULA

We begin with the 1D form for the internal multiple prediction formula, as per Weglein and Matson (1998) but in the time domain:

$$\mathbf{IM}_\omega(\omega) = \int dt' e^{i\omega t'} s(t') \int_{-\infty}^{t'-\epsilon_1} dt'' e^{-i\omega t''} s(t'') \int_{t''+\epsilon_2}^{\infty} dt''' e^{i\omega t'''} s(t'''). \quad (17)$$

Using the identity

$$\int dt' f(t') \int_{-\infty}^{t'-\epsilon_1} dt'' g(t'') = \int_{-\infty}^{\infty} dt' g(t') \int_{t'+\epsilon_1}^{\infty} dt'' f(t''), \quad (18)$$

this may be re-written

$$\begin{aligned} \mathbf{IM}_\omega(\omega) &= \int dt' e^{-i\omega t'} s(t') \left[\int_{t'+\epsilon_2}^{\infty} dt'' e^{i\omega t''} s(t'') \right] \left[\int_{t'+\epsilon_1}^{\infty} dt''' e^{i\omega t'''} s(t''') \right] \\ &= \int dt' e^{-i\omega t'} s(t') \left\{ \left[\int dt'' e^{i\omega t''} s'(t'', t', \epsilon_2) \right] \left[\int dt''' e^{i\omega t'''} s'(t''', t', \epsilon_1) \right] \right\}, \end{aligned} \quad (19)$$

where

$$s'(a, b, \epsilon) = H[a - (b + \epsilon)]s(a), \quad (20)$$

and where H is the Heaviside function

$$H(x) = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases}. \quad (21)$$

We will next inverse Fourier transform $\mathbf{IM}_\omega(\omega)$ to create the required time-domain expression. We will use the fact that the quantity in brackets $\{\cdot\}$ in equation (19), being the product of two Fourier transforms, can be interpreted as the Fourier transform of a convolution:

$$\{\cdot\} = \int dt'' e^{i\omega t''} \left[\int dt''' H(t'' - t''' - (t' + \epsilon_2))s(t'' - t''')H(t''' - (t' + \epsilon_1))s(t''') \right].$$

To begin, we have that

$$\mathbf{IM}_t(t) = \frac{1}{2\pi} \int d\omega e^{-i\omega t} \mathbf{IM}_\omega(\omega). \quad (22)$$

Substituting equation (19) into (22) and making use of the convolution form above, we obtain

$$\mathbf{IM}_t(t) = \int \int \int dt' dt'' dt''' s(t')s(t'' - t''')s(t''') \Omega \left[\frac{1}{2\pi} \int d\omega e^{-i\omega[t-(t''-t')]} \right], \quad (23)$$

where $\Omega = H[t'' - t''' - (t' + \epsilon_2)]H[t''' - (t' + \epsilon_1)]$. Recognizing the integral over ω as a delta function, we evaluate the t' integral to obtain:

$$\text{IM}_t(t) = \int dt' s(t' - t) \int_{t'-(t-\epsilon_2)}^{t-\epsilon_1} dt'' s(t' - t'') s(t'') \quad (24)$$

as desired.

REFERENCES

- Araújo, F. V., 1994, Linear and non-linear methods derived from scattering theory: backscattered tomography and internal multiple attenuation: Ph.D. thesis, Universidade Federal da Bahia.
- Berkhout, A. J., 1999, Multiple removal based on the feedback model: *The Leading Edge*, **18**, No. 1, 127–131.
- Coates, R. T., and Weglein, A. B., 1996, Internal multiple attenuation using inverse scattering: results from prestack 1 & 2d acoustic and elastic synthetics: SEG Technical Program Expanded Abstracts, 1522–1525.
- de Melo, F., Idris, M., Wu, J., and Kostov, C., 2014, Cascaded internal multiple attenuation with inverse scattering series: Western Canada case study: SEG Expanded Abstracts.
- de Melo, F., Kostov, C., and Wu, J., 2015, Internal multiple attenuation on radial gathers with inverse scattering series prediction: EAGE Expanded Abstracts.
- Fu, Q., Luo, Y., Kelamis, P. G., Huo, S., Sindi, G., and Weglein, A. B., 2010, The inverse scattering series approach towards the elimination of land internal multiples: SEG Expanded Abstracts.
- Ganley, D. C., 1981, A method for calculating synthetic seismograms which includes the effects of absorption and dispersion: *Geophysics*, **46**, No. 8, 1100–1107.
- Hernandez, M. J., and Innanen, K. A., 2014, Identifying internal multiples using 1d prediction: physical modelling and land data examples: *Canadian Journal of Geophysical Exploration*, **39**, No. 1, 37–47.
- Innanen, K. A., 2015, A nonstationary search parameter for internal multiple prediction: CREWES Annual Report, **27**.
- Innanen, K. A., and Pan, P., 2015, Large dip artifacts in 1.5D internal multiple prediction and their mitigation: CSEG Expanded Abstracts.
- Jakubowicz, H., 1998, Wave equation prediction and removal of interbed multiples, *in* SEG Expanded Abstracts, Society of Exploration Geophysicists, 1527–1530.
- Keating, S., Sun, J., Pan, P., and Innanen, K. A., 2015, Nonstationary l_1 adaptive subtraction with application to internal multiple attenuation: CREWES Annual Report, **27**.
- Luo, Y., Kelamis, P. G., Fu, Q., Huo, S., Sindi, G., Hsu, S., and Weglein, A. B., 2011, Elimination of land internal multiples based on the inverse scattering series: *The Leading Edge*, **30**, No. 8, 884–889.
- Margrave, G. F., 2015, Stratigraphic filter and Q estimation: CSEG Expanded Abstracts.
- Otnes, E., Hokstad, K., and Sollie, R., 2004, Attenuation of internal multiples for multicomponent and towed streamer data, *in* SEG Expanded Abstracts, Society of Exploration Geophysicists, 1297–1300.
- Pan, P., and Innanen, K. A., 2014, Numerical analysis of 1.5D internal multiple prediction: SEG Expanded Abstracts.
- Ramírez, A. C., and Weglein, A., 2005, Progressing the analysis of the phase and amplitude prediction properties of the inverse scattering internal multiple attenuation algorithm.: *J. of Seismic Expl.*, **13**, 283–301.

- Ras, P., Volterrani, S., Hannan, A., Rao, N., Al-Ashwak, S., Kidambi, V., and Al-Qadeeri, B., 2012, A well-driven interbed demultiple workflow for deep carbonate reservoirs of kuwait: EAGE Expanded Abstracts.
- Reshef, M., Keydar, S., and Landa, E., 2003, Multiple prediction without pre stack data - an efficient tool for interpretive processing: *First Break*, **21**, No. 3.
- Sonika, S., Zarkhidze, A., Heim, J., and Dragoset, W., 2012, Complementary data-driven methods for interbed demultiple of land data: EAGE Expanded Abstracts.
- Sun, J., and Innanen, K. A., 2015a, 1.5D internal multiple prediction in the plane wave domain: CSEG Expanded Abstracts.
- Sun, S., and Innanen, K. A., 2015b, 2d internal multiple prediction in the double planewave domain: CREWES Annual Report, **27**.
- Weglein, A., and Dragoset, W., 2005, Multiple Attenuation., *Geophysics reprint series: Soc. Expl. Geophys.*
- Weglein, A. B., Araújo, F. V., Carvalho, P. M., Stolt, R. H., Matson, K. H., Coates, R. T., Corrigan, D., Foster, D. J., Shaw, S. A., and Zhang, H., 2003, Inverse scattering series and seismic exploration: *Inverse Problems*, , No. 19, R27–R83.
- 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**, No. 6, 1975–1989.
- Weglein, A. B., and Matson, K. H., 1998, Inverse scattering internal multiple attenuation: An analytic example and subevent interpretation, *in SPIE Conference on Mathematical Methods in Geophysical Imaging*, 108–117.
- Wu, Z. J., Sonika, S., and Dragoset, W., 2011, Robust internal multiple prediction: SEG Expanded Abstracts.
- Zou, Y., and Weglein, A. B., 2015, An internal-multiple elimination algorithm for all first-order internal multiples for a 1d earth: SEG Expanded Abstracts.