Non-conventional seismic differencing algorithms in time-lapse II

Vanja Vracar and Robert J. Ferguson

ABSTRACT

Conventional seismic differencing in time-lapse studies proves to capture large errors in seismic amplitude and smaller errors in seismic phase through numerical experiments. Keeping this observation in mind we implement four non-conventional seismic differencing algorithms: 1) Cross-correlation differencing (CCD), 2) Pseudo cross-correlation differencing (PCCD), 3) Conventional imaging condition differencing (CICD) and 4) Imaging condition differencing (ICD). The CCD algorithm uses cross-correlation and Gaussian function to create filtering operator that is further multiplied by conventional difference and later migrated. The algorithm proves to be computationally costly as it is performed in the time domain. Hence the same concept is implemented in the frequency domain, named PCCD algorithm. The PCCD algorithm improves computational time and resolution significantly. As both, CCD and PCCD algorithms are dependent on user to move from filtering operator creation to non-conventional differencing to migration, we develop CICD algorithm as a pilot algorithm to ICD. CICD is based on the pre-stack depth migration and conventional differencing. It performs wavefield extrapolation and conventional differencing at the imaging condition. As CICD algorithm proves to be robust, we develop ICD algorithm that is based on pre-stack depth migration and non-conventional differencing, namely PCCD filtering operator. ICD algorithm fully eliminates dependence on the user, improves resolution and computational cost. These four algorithms are tested on two data sets and prove to be a valuable alternative tool in seismic differencing and time-lapse studies.

INTRODUCTION

Conventional seismic differencing relies on a number of assumptions that may not always represent reality. Systematic error, error associated with the use of conventional (imperfect) imaging algorithms, and error due to source/receiver coupling variations are assumed to be small relative to the seismic response of fluid transport. Source / receiver positioning must be the same between surveys in time-lapse. The result is that conventional differencing involves simple match filtering followed by subtraction where the interpretable product is an image of the change in fluid location superimposed upon some background noise level. In reality, errors are often very large.

We observe that though errors might be large, and with the exception of source / receiver location repeatability, coupling variation and system errors result in differences in seismic amplitude and not necessarily seismic phase so that any methodology beyond simple match-filtering and differencing might incorporate this observation.

In order to implement this concept cross-correlation is employed. Cross-correlation groups similarities and dissimilarities around zero lag and elsewhere, respectively. The cross-correlation models can be further used in filtering of data, where in this case similari-

ties are eliminated. Now, seismic difference models will highlight primarily dissimilarities, that is major reservoir property changes of two time-lapse steps, hence, significantly clean ambiguity.

THEORY

The cross-correlation imaging condition of seismic depth migration essentially uses a model of the source wavefield to identify reflection amplitudes on an input data set. This identification occurs most simply through cross-correlation, and this is a process that relies on good phase fidelity in both the model and the data; reflection amplitude in the data whose corresponding phase matches that of source amplitude is mapped to zero lag. Data at zero lag are then mapped to the image space and, when this is done for all depth grid levels, the migrated seismic image is produced.

Central to the imaging condition, then, is that reflection data are identified, captured, and used in the image, while all non matching data are discarded. In application to time-lapse analysis, the imaging idea seems well suited as it is sensitive to phase, and phase is recorded most reliably in the seismic method. Rather than forward model a reflection, and then look for a similar reflection shape in the data, it seems reasonable to use a reference data set (with all reflections, multiples, and so on) - the complete seismic wavefield instead of a model. Used instead of a source model in the imaging condition, the reference wavefield will act to find all similar energy in the monitor survey and then map that energy to zero lag. Any differences, i.e. variation due to fluid flow, is discarded. A simple modification to this approach causes all similar events to be discarded and the difference - the fluid flow - is captured and imaged.

In this section conventional differencing is reviewed and four new differencing methods based on ideas from seismic imaging are outlined. These are:

1. cross-correlation differencing (CCD),
2. pseudo cross-correlation differencing (PCCD),
3. conventional imaging condition differencing (CICD) and
4. imaging condition differencing (ICD).

The CCD is a method implemented in time domain. It calculates the cross-correlation of two time-lapse steps, that is further multiplied by a Gaussian filter to notch out data at zero lag in cross-correlation. The result is further transformed to time domain and multiplied by the conventional difference. As the algorithm is computationally costly the same method is implemented in frequency domain. We name cross-correlation in frequency domain pseudo cross-correlation and implement the PCCD. It takes the pseudo cross-correlation of two time-lapse steps. Then, Gaussian filter is created and convolved with pseudo cross-correlation result to notch out data at zero lag in frequency domain. The result is further inverse Fourier transformed and multiplied by the conventional difference. As both algorithms are dependent on the user to move from filtering operator creation to non-conventional differencing to migration, we develop CICD algorithm as a pilot algorithm to

ICD. CICD is based on the pre-stack depth migration and conventional differencing. It performs wavefield extrapolation and conventional differencing at the imaging condition. As CICD proves to be robust, we develop ICD that is also based on pre-stack depth migration but performs non-conventional differencing (PCCD) at imaging condition. ICD algorithm fully eliminates dependence on the user, improves resolution and computational cost.

Conventional Differencing

We define baseline and monitor surveys to image original data and data after time passed, respectively. Conventional differencing is obtained employing conventional matrix subtraction (Vracar and Ferguson, 2010) as:

$$D = M - B, \tag{1}$$

where B and M define baseline and monitor surveys, respectively. Baseline and monitor surveys are recorded in time.

Step I: Velocity Models	→	**Step II:** Seismic Models	→	**Step III:** Differencing Models	→	**Step IV:** Migration Models

Table 1. Conventional differencing workflow. The workflow consists of four steps: velocity models are used to generate synthetic seismic models. Synthetics are then differenced and migrated to produce conventional difference models.

In table 1 steps to obtain conventional difference models are presented. First, velocity models are taken to generate synthetic seismic models. In Vracar and Ferguson (2010) paper we describe forward modelling flows and follow similar modelling here. Then, synthetics are differenced and migrated employing pre-stack depth migration to obtain conventional difference models. Conventional difference seismic models are ambiguous and hard to interpret for differences only, hence, we intend to clean imaging by filtering and eliminating similarities of two time-lapse steps (Vracar and Ferguson, 2010).

Cross-correlation differencing in the time domain (CCD)

Cross-correlation differencing (CCD) is implemented entirely in the time domain and as such, it is the most natural of the methods that we will present to those familiar with conventional differencing.

Step I: Velocity Models	→	**Step II:** Seismic Models	→	**Step III:** Cross-Correlation Filtering Models	→	**Step IV:** Differencing Models	→	**Step V:** Migration Models

Table 2. CCD workflow. This workflow is similar to workflow 1 with the additional step of cross-correlation filtering. This step is to locate and eliminate similarities to highlight dissimilarities.

The workflow 2 is similar to workflow 1, but it employs an additional step of cross-correlation filtering.

Step I: CALCULATION OF CROSS-CORRELATION MATRIX

calculate the cross-correlation matrix of baseline and monitor surveys

↓

Step II: CALCULATION OF CROSS-CORRELATION

multiply cross-correlation matrix by monitor survey

↓

Step III: GAUSSIAN FILTER

create Gaussian filter

↓

Step IV: CROSS-CORRELATION FILTERING

multiply cross-correlation of baseline and monitor surveys by Gaussian filter

↓

Step V: INVERSION OF CROSS-CORRELATION MATRIX

build an inverse matrix of cross-correlation matrix

↓

Step VI: FILTERING BY INVERSE CROSS-CORRELATION MATRIX

multiply Gaussian filtered cross-correlation matrix
by inverse matrix of cross-correlation matrix

↓

Step VII: CONVENTIONAL DIFFERENCE FILTERING

multiply conventional difference of monitor and baseline by the result of step VI

Table 3. CCD in the time domain. In Step we to VII outline actions executed in order to filter conventional difference employing cross-correlation, Gaussian filter and matrix inversion.

In Step I (Table 3), the cross-correlation matrix is built. Cross-correlation is closely related to convolution (Margrave, 2008): where in convolution of two signals we reverses one, shift, multiply, add, and repeat, in cross-correlation neither signal is reversed and the process is otherwise the same (Margrave, 2008). Since convolution and cross-correlation share similar characteristics we refer to convolution theory and modify it to accommodate for cross-correlation operations.

Let's step back and firstly look at the definition of convolution (Lines and Newrick, 2008):

$$p(t) \equiv a(t) \bullet h(t), \tag{2}$$

where $\bullet$ stands for the convolution operation and $p(t)$, $a(t)$ and $h(t)$ denote convolution of

two signals, signal one and signal two, respectively. In mathematical terms, convolution is an operation on two functions, $a(t)$ and $h(t)$, producing the third function, $p(t)$, that can be described as manipulated version of one of the original functions. Convolution honors commutativity, associativity and distributivity (Margrave, 2008). In geophysical terms, convolution is an operation acting on two signals, such that one is viewed as the filter to the other. In practice, convolution models filtering of seismic energy by various rock layers in the Earth (Schlumberger, 2011).

Now, the convolution stated by equation (2) can be further examined mathematically through a definition of convolutional integral (Margrave, 1998):

$$p(t) = \int_{-\infty}^{\infty} a(t-\tau)h(\tau)d\tau, \tag{3}$$

where $p(t)$, $a(t-\tau)$ and $h(\tau)$ denote filtered output, filter impulse response and input signal, respectively. Equation (3) can be rewritten in terms of matrix operations according to

$$\mathbf{p} = \mathbf{A}\,\mathbf{h} \tag{4}$$

(Margrave, 1998), where $\mathbf{A}$ is a convolution matrix and $\mathbf{p}$ and $\mathbf{h}$ are column vectors. If expanded equation (4) yields (Margrave, 1998):

$$\begin{bmatrix} \vdots \\ p_0 \\ p_1 \\ p_2 \\ p_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & a_0 & a_{-1} & a_{-2} & \vdots \\ \vdots & a_1 & a_0 & a_{-1} & \vdots \\ \vdots & a_2 & a_1 & a_0 & \vdots \\ \vdots & a_3 & a_2 & a_1 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ h_0 \\ h_1 \\ h_2 \\ h_3 \\ \vdots \end{bmatrix}. \tag{5}$$

Equation (5) highlights matrix A to have constant entries at each descending diagonal from left to right. In mathematical terms, this matrix is known as Toeplitz matrix, named by Otto Toeplitz, a German mathematician working on functional analysis (Bini, 1995). Common applications of Toeplitz matrices include the numerical solution of some differential and integral equations, the computation of splines, time series analysis, Markov chains, signal and image processing (Bini, 1995). In geophysical terms, this matrix structure is known as the convolution matrix (Inannen, 2010). It is created by populating each row by a filter with zero time shifted to the diagonal. Since convolution is briefly reviewed, we look at cross-correlation.

Let's consider the definition of the cross-correlation (Lines and Newrick, 2008):

$$x(t) \equiv m(t) \otimes b(t), \tag{6}$$

where $\otimes$ stands for cross-correlation operation and $x(t)$, $m(t)$ and $b(t)$ denote cross-correlation of two signals, signal one and signal two, respectively. Recall the basic definition of cross-correlation to be a measure of similarity of two waveforms.

Its definition in equation (6) can be expressed as the cross-correlation integral given by:

$$x(t) = \int_{-\infty}^{\infty} a(t+\tau)b(\tau)d\tau, \tag{7}$$

where $x(t)$, $a(t+\tau)$, $b(\tau)$ denote filtered output, reverse filter impulse response and impulse signal, respectively. Cross-correlation in equation (7) can now be related to convolution in equation (3) by time reversing filter impulse and populating matrices as:

$$\mathbf{x} = \mathbf{A}\,\mathbf{b}, \tag{8}$$

that is

$$\begin{bmatrix} \vdots \\ x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & a_{-2} & a_{-1} & a_0 & \vdots \\ \vdots & a_{-1} & a_0 & a_1 & \vdots \\ \vdots & a_0 & a_1 & a_2 & \vdots \\ \vdots & a_1 & a_2 & a_3 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ b_0 \\ b_1 \\ b_2 \\ b_3 \\ \vdots \end{bmatrix}, \tag{9}$$

where $\mathbf{x}$ and $\mathbf{b}$ denote column vectors and $\mathbf{A}$ denotes the cross-correlation matrix. Generation of matrix A is the goal of Step I in the CCD approach (Table 3).

In Step II the cross-correlation matrix is multiplied by the monitor survey to calculate the cross-correlation of baseline and monitor surveys. The above statement can be described in mathematical terms through matrix operations as:

$$\mathbf{x}_{corr} = \mathbf{A}\,\mathbf{m}, \tag{10}$$

where **A** and **m** are the cross-correlation matrix for one baseline trace, one trace from the monitor survey.

Equation (10) groups all similarities around zero lag and all dissimilarities elsewhere.

In Step III, a Gaussian filter is created. We employ the time domain Gaussian function defined as (Margrave, 2008):

$$g(t) = e^{-\alpha^2 t^2}, \tag{11}$$

where $g(t)$, α and t denote Gaussian filter, Gaussian width and time, respectively.

In Step IV the cross-correlation matrix in time is scalar multiplied by the Gaussian filter defined in equation (11) as

$$\mathbf{f} = \mathbf{g} \odot \mathbf{x}_{corr}, \tag{12}$$

where $\odot$ indicates scalar multiplication between the elements of vectors. Equation (12) deletes zero lag, hence leaves only dissimilarities of the two surveys.

Now, that the data is cross-correlated and filtered, the cross-correlation matrix $\mathbf{A}$ is employed to provide inverse cross-correlation operator, $\mathbf{A^{-1}}$, as Step V. This step is to reverse the cross-correlation operation.

In Step VI filtered cross-correlation is then multiplied by the inverse cross-correlation matrix:

$$\mathbf{r} = \mathbf{A^{-1}} \odot \mathbf{f}. \tag{13}$$

In Step VII we multiply the conventional difference by filtered data:

$$\mathbf{d^{\star}} = \mathbf{r}[\mathbf{m} - \mathbf{b}]. \tag{14}$$

Now all differences are highlighted and almost all similarities are eliminated.

As a single analytic process, CCD is

$$\mathbf{d^{\star}} = \mathbf{A}^{-1}\left[\mathbf{g} \odot \left[\mathbf{A}\ \left[\mathbf{m} \odot \left[\mathbf{m} - \mathbf{b}\right]\right]\right]\right]. \tag{15}$$

Note, though much of the computational effort in CCD is restricted to scalar multiplication, convolution and then de-convolution by $\mathbf{A}$ and $\mathbf{A}^{-1}$ can be quite expensive.

Computational cost of CCD

Let's analyze the computational cost of CCD. We assume the baseline and monitor surveys to be $M \times N$ matrices.

operation	cost
build cross-correlation matrix	$O(MN)$
multiplication	$O(M^2N^2)$
build Gaussian filter	$O(MN)$
build cross-correlation operator	$O(M^3N^3)$
multiplication	$O(MN)$
migration	$O(MNlog(MN))$

Table 4. Computational cost of the CCD workflow in the time domain. It takes $O(M^5N^5 + 3MN + MNlog(MN))$ operations to complete the workflow. The process is successful, but costly.

Its computational cost is outlined in Table 4 (Knuth, 1997), where "O" defines number of operations. Although, the algorithm is successful, it is computationally costly as it takes $O(M^5N^5 + 3MN + MNlog(MN))$ operations to complete. The most expensive operation seems to be the creation of the inverse cross-correlation operator. This is a step is considered for optimization to reduce cost. We look into developing the same workflow in the frequency domain.

Pseudo cross-correlation differencing in the frequency domain (PCCD)

Due to a large computational cost in the time domain the algorithm is adapted to work in the frequency domain. We expect to save computational cost in frequency domain because

the matrix inversion will be substituted by the Fourier transformation. The algorithm is named Pseudo cross-correlation differencing (PCCD).

Step I: Velocity Models	→	**Step II:** Seismic Models	→	**Step III:** Pseudo Cross-Correlation Filtering Models	→	**Step IV:** Differencing Models	→	**Step V:** Migration Models

Table 5. PCCD workflow. This workflow is similar to workflow 3, however, the additional step that takes place is pseudo cross-correlation filtering. This step is to locate and eliminate similarities to highlight dissimilarities.

Table 5 is similar to Table 1, however, cross-correlation filtering is computed in the frequency domain.

Step I: DATA TRANSFORMATION

transform of baseline and monitor surveys from time to the frequency domain

↓

Step II: PSEUDO CROSS-CORRELATION

calculate the pseudo cross-correlation of baseline and monitor surveys

↓

Step III: GAUSSIAN FILTERING

create Gaussian filter in the frequency domain

↓

Step IV: CONVOLUTION

convolution of pseudo cross-correlation of baseline and monitor surveys with Gaussian filter

↓

Step V: INVERSION OF PSEUDO CROSS-CORRELATION

inversion of pseudo cross-correlation

↓

Step VI: INVERSE FOURIER TRANSFORMATION

inverse Fourier transform the result of step V

↓

Step VII: CONVENTIONAL DIFFERENCE FILTERING

multiply conventional difference of monitor and baseline by the result of step VII

Table 6. PCCD in the frequency domain. Step I to VII outline actions executed in order to filter conventional difference employing pseudo cross-correlation, Gaussian filtering and pseudo cross-correlation inversion.

Table 6 summarizes the PCCD algorithm. In Step I, the baseline survey $b(t)$ and monitor survey $m(t)$ are Fourier transformed to the frequency domain according to

$$B(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} b(t) e^{iwt} dt, \tag{16}$$

$$M(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} m(t) e^{iwt} dt, \tag{17}$$

where $B(t)$, $M(t)$ and w denote baseline and monitor surveys and frequency in the frequency domain, respectively. $B(t)$ and $M(t)$ are further decomposed into their phase and

amplitude components according to

$$B(w) = A_b(w)e^{i\phi_b(w)}, \tag{18}$$

$$M(w) = A_m(w)e^{i\phi_m(w)}, \tag{19}$$

where $A_b(w)$, $A_m(w)$, are the amplitude spectra of the baseline and monitor surveys respectively, and $\phi_b(w)$ and $\phi_m(w)$ are the phase spectra.

In Step II, $B(t)$ and $M(t)$ are cross-correlated according to

$$P_{cc} = A_m(w)e^{i(\phi_m(w)-\phi_b(w))}. \tag{20}$$

Note that the amplitude spectrum of baseline survey is omitted in equation (20)as our approach to differencing, in this case deletion of all data that are similar between baseline and monitor, requires only phase information from both spectra and amplitude information only from the monitor spectrum.

In Step III the Gaussian filter in the frequency domain is created as:

$$G(w) = \sqrt{\frac{\pi}{\alpha}}e^{(-\frac{\pi w}{\alpha})}, \tag{21}$$

where $G(w)$ and α denote Gaussian filter in the frequency domain and Gaussian width, respectively. In Step IV pseudo cross-correlation is convolved with Gaussian filter such as:

$$S(w) = \int_{-\infty}^{\infty} G(w^\star)P_{cc}(w - w^\star)dw^\star. \tag{22}$$

where w^* denote a dummy frequency variable. In Step V pseudo cross-correlation inverse operator (PCCIO) is computed from the following equation:

$$R(w) = e^{i\phi_m(w)}S(w). \tag{23}$$

In order to compute PCCIO we only use the definition of the phase of the monitor survey to restore the phase of filtered data. This is because the phase spectrum only depends on seismic traveltime, and such provides confidence in results. The amplitude spectrum, not used in this calculation, can be ambiguous. Any variations in shot coupling, geophone coupling or shot strength seem to reflect most strongly in the amplitude rather than in the phase spectrum. Seismic noise sources along with model estimation and imaging errors appear to distort amplitudes much more than phases.

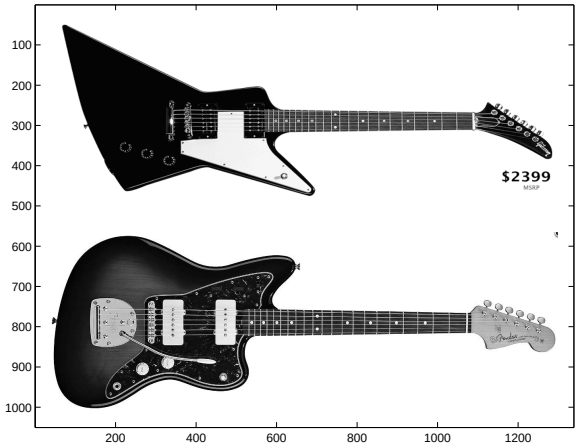


FIG. 1. Two guitars manufactured by different companies are presented: Gibson Explorer at the top and Fender Jazzmaster at the bottom.

Two guitars are presented in Figure 1: Gibson Explorer guitar at the top and Fender Jazzmaster guitar at the bottom. They are manufactured by different companies and will serve for the experiment. First, employing MATLAB functions amplitude and phase spectra are extracted to yield:

$$B(w) = A_1(w)e^{i\phi_1(w)}, \tag{24}$$

$$M(w) = A_2(w)e^{i\phi_2(w)}, \tag{25}$$

where A_1, A_2, ϕ_1 and ϕ_2 stand for amplitude spectrum of Gibson Explorer, amplitude spectrum of Fender Jazzmaster, phase spectrum of Gibson Explorer and phase spectrum of Fender Jazzmaster, respectively. Now, a third image is created employing the amplitude spectrum of Fender and the phase spectrum of Gibson as:

$$B(w) = A_2(w)e^{i\phi_1(w)}. \tag{26}$$

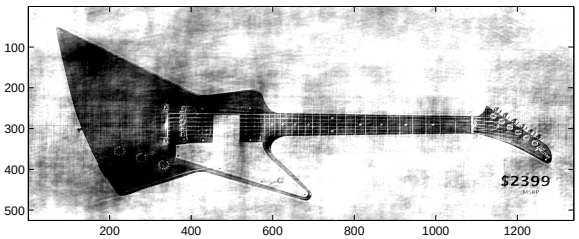


FIG. 2. Restored image using amplitude spectrum of Fender Jazzmaster image and the phase spectrum of Gibson Explorer image.

The resulting image in Figure 2 resembles much like the Gibson, and the effect of the Fender amplitude shows as blurring of the image. This experiment proves that amplitude spectrum is less reliable when compared to phase spectrum, hence proves the choice of using just monitor survey's phase spectrum in computing PCCIO sufficient in Step V.

In Step VI the computed result in equation (23) is inverse Fourier transformed from frequency to time domain:

$$r(t^{\star}) = \int_{-\infty}^{\infty} R(w) e^{iwt^{\star}} dw. \tag{27}$$

In Step VII the computed result multiplies the conventional difference as:

$$\mathbf{d}^{\star\star} = \mathbf{r} \odot [\mathbf{m} - \mathbf{b}]. \tag{28}$$

Computational Cost of PCCD

Let's analyze the computational cost of PCCD. We assume the baseline and monitor surveys to be $M \times N$ matrices.

operation	cost
fast Fourier transform (FFT) X 2	$O(2MNlog(MN))$
pseudo cross-correlation	$O(M^2N^2)$
build Gaussian filter	$O(MN)$
convolution	$O(MNlogMN)$
build inverse pseudo cross-correlation operator	$O(MN)$
inverse fast Fourier transform (IFFT)	$O(MNlogMN)$
migration	$O(MNlog(MN))$

Table 7. Computational cost of the PCCD workflow in the frequency domain. It takes $O(M^2N^2 + 2MN + 5MNlogMN)$ operations to complete the workflow. The process is successful and much cheaper if executed in the frequency domain.

PCCD's computational cost is outlined in Table 4 (Knuth, 1997). The most expensive calculation is convolution with $O(M^2N^2)$ operations, and the least expensive calculation is fast forward/inverse Fourier transformation with $O\left(N \log_2 N\right)$ operations. PCCD algorithm is successful and computationally much cheaper, then CCD algorithm:

$$O(M^2N^2 + 2MN + 5MNlogMN) << O(M^5N^5 + 3MN + MNlog(MN)). \tag{29}$$

Simplifying equation 29 to:

$$O(4MNlogMN) << O(M^3N^3 + MN) \tag{30}$$

significant decrease in computational cost from CCD to PCCD is noted.

Conventional imaging condition difference (CICD)

The workflow 1 summarizes steps taking place sequentially under user's supervision. In order to improve its computational efficiency and eliminate dependance on the user to move from step to step, pre-stack depth migration algorithm (PSDM) implemented by CREWES is modified. The migration algorithm of choice is the Split step Fourier (SSF) migration (Stoffa et al., 1990) as it is computationally efficient.

As a pilot experiment to implement PCCD with migration algorithm, PSDM is manipulated to take two time-lapse steps and calculate their conventional difference.

RECURSIVE CICD ALGORITHM				
Step I: Velocity Models	→	**Step II:** Seismic Models	→	**Step III:** Differencing and Migration (at every depth level)

Table 8. Recursive conventional difference imaging condition algorithm. The workflow starts with velocity and seismic models, then at every depth level performs conventional differencing until it reaches the maximum depth level.

Table 8 illustrates the general steps taking place within CICD algorithm. The workflow begins with velocity and seismic models. It performs conventional differencing at every depth level until it reaches the maximum depth level.

RECURSIVE ALGORITHM
Step I: DATA TRANSFORMATION

transformation of baseline and monitor surveys from space to $f - kx$ domain

↓

Step II: DATA EXTRAPOLATION

wavefield extrapolation of baseline and monitor surveys at every depth step

↓

Step III: CONVENTIONAL DIFFERENCING

CD method at every depth step

Table 9. Imaging condition differencing at every depth step. Step I to VII outline actions executed in order to filter conventional difference employing pseudo cross-correlation, Gaussian filter and pseudo cross-correlation inversion.

Table 9 describes steps taken in modifying pre-stack depth migration algorithm to produce conventional difference of two time-lapse steps. In step I, baseline and monitor surveys are transformed from space to f-kx domain. PSDM is implemented to execute one depth level at the time starting at the surface and ending at maximum depth level. In step II, the isotropic split step extrapolation of baseline and monitor surveys at every depth step occurs. Now, the conventional difference is embedded before the depth level increases. The process recursively repeats until it reaches maximum depth. The output is, therefore, conventional difference of two time-lapse steps.

Computational Cost of CICD

Let's analyze the computational cost of CICD. We assume the baseline and monitor surveys to be mxn matrices.

operation	cost
PSDM	$O(MNlog(MN))$
CD	$O(MN)$

Table 10. Computational cost of the PCCD workflow in the frequency domain. It takes $O(MN + MNNlog(MN))$ operations to complete the workflow. The process is successful and much cheaper if executed in the frequency domain.

Table 10 is a summary of computational cost for CICD. It takes the same number of operations as execution of the conventional differencing and migration, however, eliminates user dependance.

Imaging condition differencing (ICD)

Having successfully tested manipulation of PSDM for CD producing the CICD algorithm, we proceed to accommodate PSDM for PCCD.

RECURSIVE ICD ALGORITHM

Step I: Velocity Models	→	**Step II:** Seismic Models	→	**Step III:** Filtering, Differencing and Migration (at every depth level)

Table 11. Recursive imaging condition differencing algorithm. The workflow starts with velocity and seismic models, then at every depth step performs PCCD algorithm until it reaches the maximum depth level.

Table 11 outlines steps taking place in the Imaging condition differencing (ICD) algorithm. It is similar to workflow in Table 8, however, it performs PCCD as oppose to CD algorithm at every depth level.

RECURSIVE ALGORITHM
Step I: DATA TRANSFORMATION

transformation of baseline and monitor surveys from space to f-kx domain

↓

Step II: DATA EXTRAPOLATION

wavefield extrapolation of baseline and monitor surveys at every depth step

↓

Step III: DATA FILTERING AND DIFFERENCING

PCCD method at every depth step

Table 12. Imaging condition differencing algorithm. Pre-stack depth migration algorithm is modified to migrate, filter and difference two time-lapse steps at every depth level producing only difference of two time-lapse steps.

Table 12 outlines the recursive ICD algorithm. In step I the data is transformed from space to f-kx domain. Now, a series of operations take place at one depth level, starting at the surface. In step II the isotropic split step extrapolation takes place at one depth level of baseline and monitor surveys. In step III the PCCD algorithm takes place at the same depth level. Then, the depth level increases and the process repeats until it reaches maximum depth level.

Computational Cost of ICD

Let's analyze the computational cost of ICD.

operation	cost
PSDM	$O(MNlog(MN))$
PCCD	$O(M^2N^2 + 2MN + 5MNlogMN)$

Table 13. Computational cost of the PCCD workflow in the frequency domain. It takes $O(M^2N^2 + 2MN + 6MNlogMN)$ operations to complete the workflow. The process is successful and much cheaper if executed in the frequency domain.

Table 13 shows the computational cost of ICD algorithm. The algorithm is optimal and user independent.

EXAMPLE I

Velocity models and Synthetics

The data set used is the EAGE/SEG salt velocity model (Aminzadeh et al., 1996). The model consists of complex salt structures with large velocity contrasts across the salt/sediment interface (Aminzadeh et al., 1996). This is a well known pre-stack depth

migration (PSDM) free testing data set publicly available, hence convenient to use in this work.

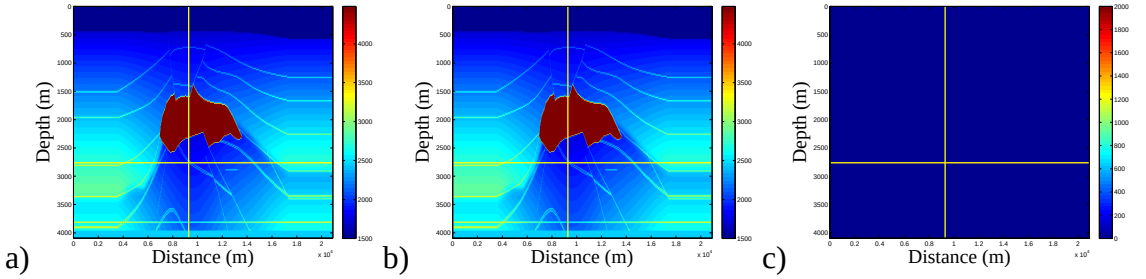


FIG. 3. EAGE/SEG salt velocity models: a) original model, b) author manipulated original model by inserting small box in the sub-salt region to accomodate time-laspe analysis and c) conventional difference of a) and b) models.

In Figure 3 velocity models are captured. Figure 3(a) show original EAGE/SEG model. Figure 3(b) shows manipulated model. To accommodate time-lapse study and analysis the author inserts a small reflector in the sub-salt region of the EAGE/SEG salt velocity model and assumes it to mimic changes due to production. The box is of constant velocity and it location is indicated by the yellow cross arrow. I denote original model baseline survey and manipulated model monitor survey. Assume that baseline and monitor survey are two time-lapse steps. Figure 3(c) shows conventional difference of Figures 3(a) and 3(b).

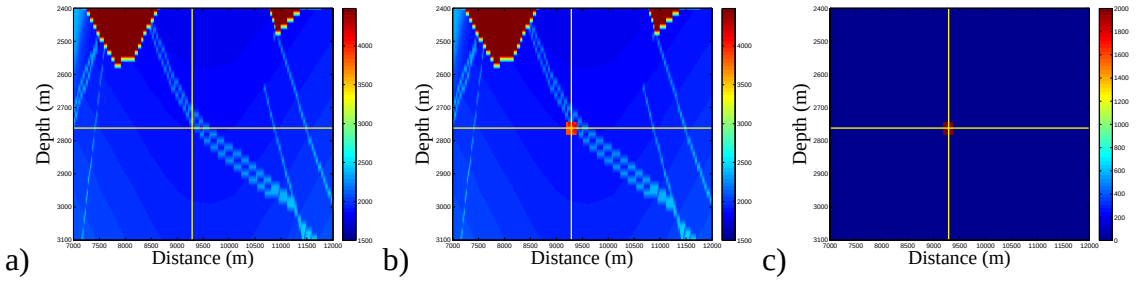


FIG. 4. EAGE/SEG salt zoomed in velocity models: a) baseline survey, b) monitor survey and c) conventional difference of a) and b) models.

Figure 4 is a zoomed in version of Figure 3. It focuses on the inserted reflector, that is baseline to monitor surveys small change in the data is captured.

Now, taking velocity models synthetics are created employing $afd\_shotrec$, MATLAB CREWES toolbox function created by Dr Gary Margrave.

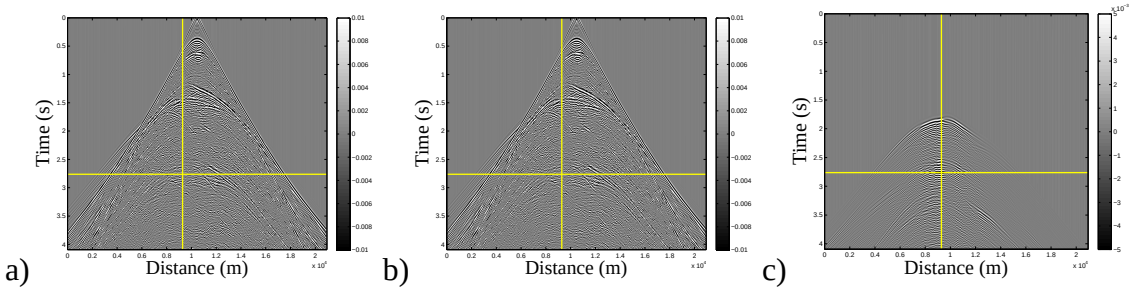


FIG. 5. EAGE/SEG salt shot gathers: a) shot gather of baseline survey, b) shot gather of monitor surveyl and c) difference of models a) and b).

Figure 5 shows shot gather of baseline survey, shot gather monitor survey and difference of the two gathers.

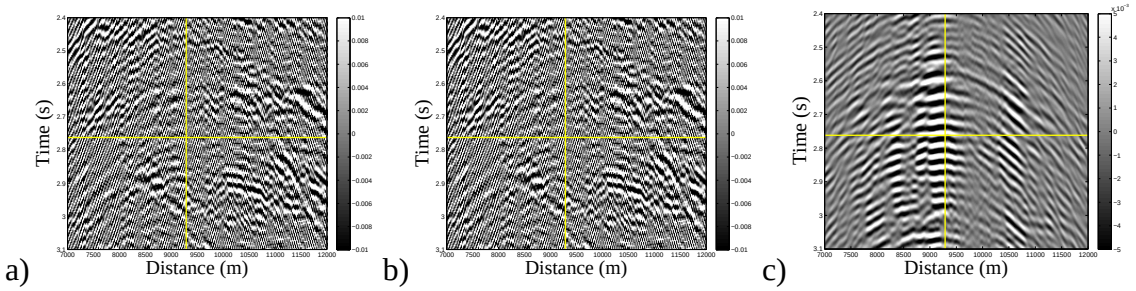


FIG. 6. Zoomed in EAGE/SEG salt shot gathers: a) shot gather of baseline survey, b) shot gather of monitor survey and c) difference of models a) and b).

Figure 6 shows zoomed in models, focusing on the area around inserted box. Observing closely Figure 6(c), it is hard to determine the location of the box.

Conventional Difference

Shot gathers are migrated employing a MATLAB function from the CREWES toolbox, $ss\_salt\_psdm\_diff\_script$. This function is based on Split step Fourier migration (Stoffa et al., 1990) and implemented by CREWES.

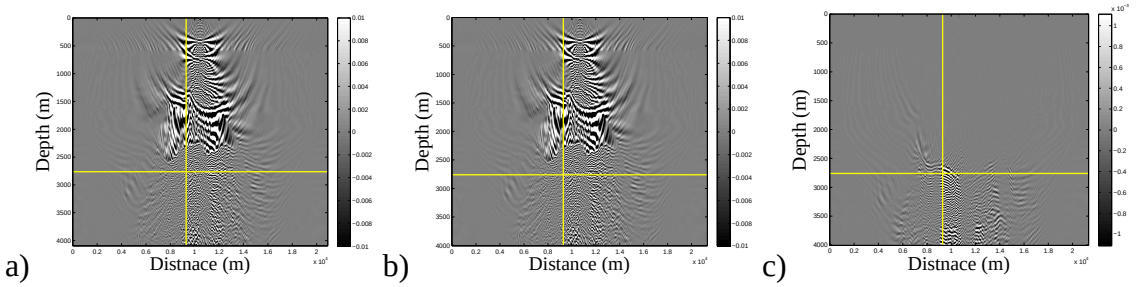


FIG. 7. EAGE/SEG salt shot gathers migrated: a) migrated baseline shot gather, b) migrated monitor shot gather and c) difference of a) and b).

Figure 7 shows baseline survey migrated, monitor survey migrated and difference of the two.

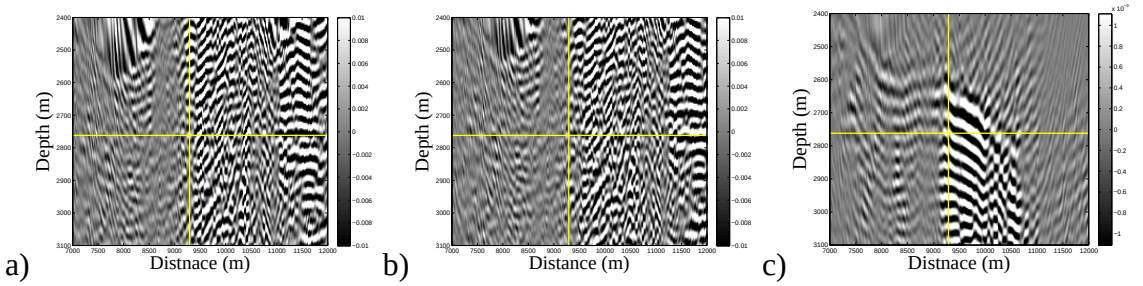


FIG. 8. Zoomed in EAGE/SEG salt shot gathers migrated: a) migrated baseline shot gather, b) migrated monitor shot gather and c) difference of a) and b).

Figure 8 is a zoomed in version of Figure 7 around the area of interest. It is very difficult to determine the location of the box if there were no cross arrows.

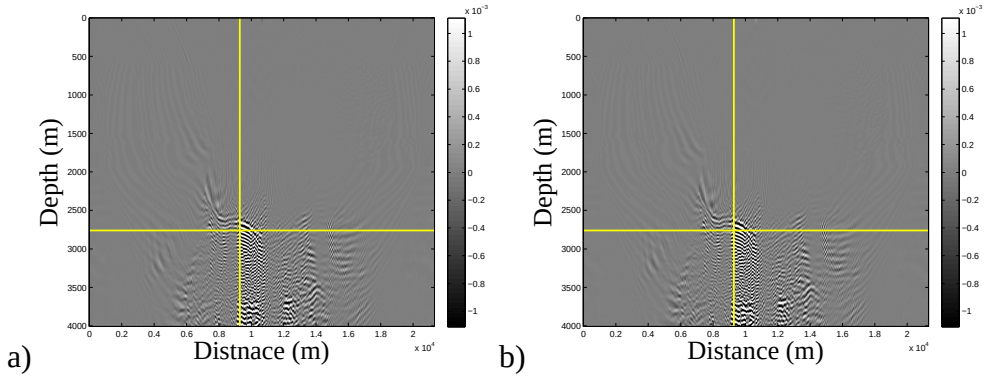


FIG. 9. EAGE/SEG conventional difference models: a) conventional difference of migarated baseline and monitor surveys and b) conventional difference of baseline and monitor shot gathers migrated.

Figure 9(a) captures the difference of migrated baseline and monitor surveys. Figure 9(b) captures migration of difference of baseline and monitor surveys.

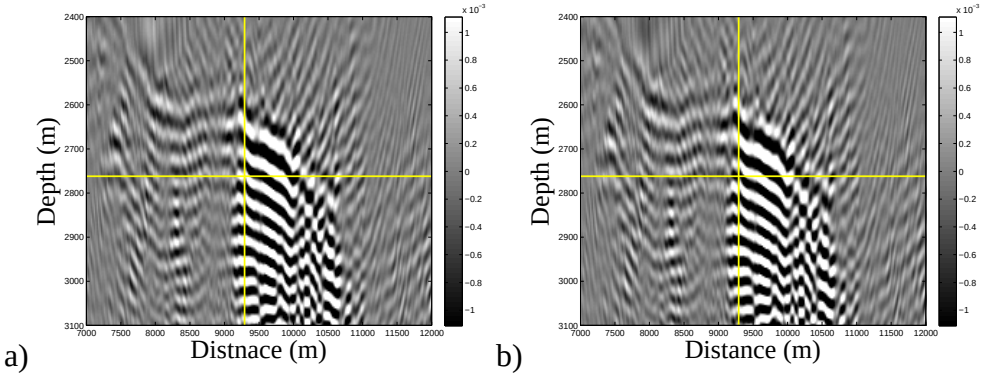


FIG. 10. Zoomed in EAGE/SEG conventional difference models: a) conventional difference of migarated baseline and monitor surveys and b) conventional difference of baseline and monitor shot gathers migrated.

Figure 10 captures zoomed in models of Figure 9. There is no significant difference, apart from machine precision, as expected.

CCD

The baseline and monitor survey shot gathers are taken and filtered employing cross-correlation method. The filtered data is migrated using split step Fourier migration by the user.

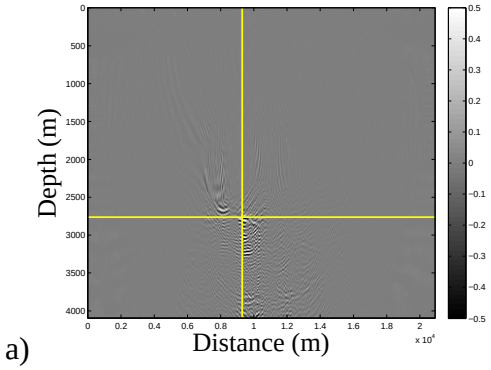
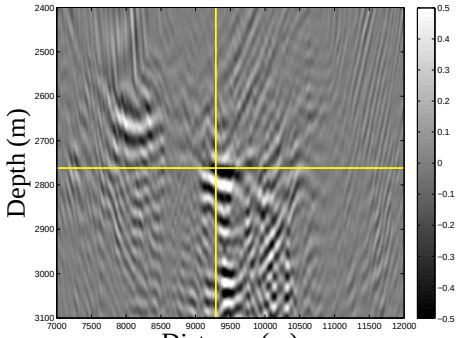


FIG. 11. Cross-correlation differencing result.

Figure 11 captures the result of CCD and PSDM.



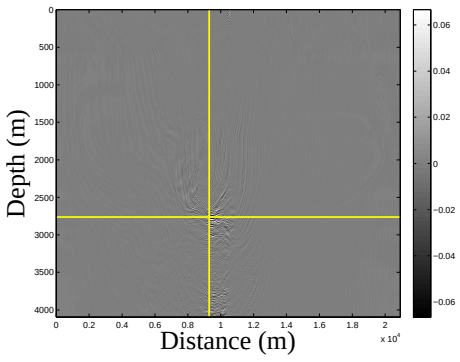
a)

FIG. 12. Cross-correlation differencing result.

Figure 12 is the zoomed in Figure 11. Although the yellow cross-arrows point to the location of inserted box, its location is clear to spot without them. The algorithm achieves considerable improvement when compared to conventional differencing.

PCCD

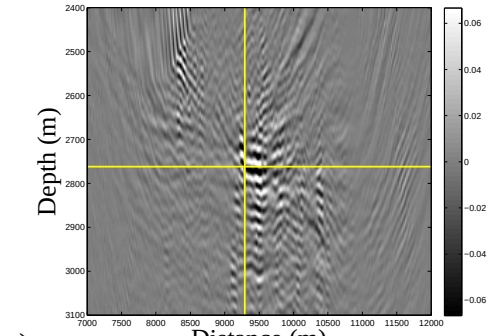
The baseline and monitor survey shot gathers are filtered employing pseudo cross-correlation method. Then, the user migrates output data invoking split step Fourier migration.



a)

FIG. 13. Pseudo cross-correlation differencing result.

Figure 13 captures the result of PCCD and PSDM.



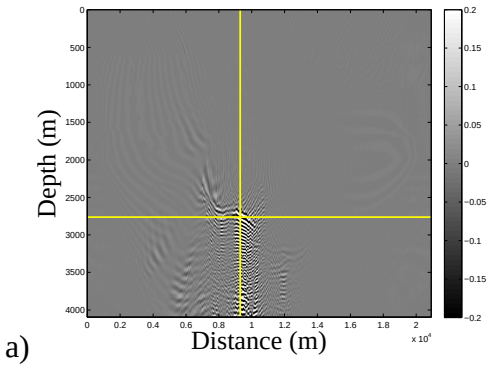
a)

FIG. 14. Zoomed in pseudo cross-correlation differencing result.

Figure 14 is the zoomed in Figure 13. PCCD is also considerable improvement to result from conventional differencing. The location of the box and its resolution are better when compared to the CCD result.

CICD

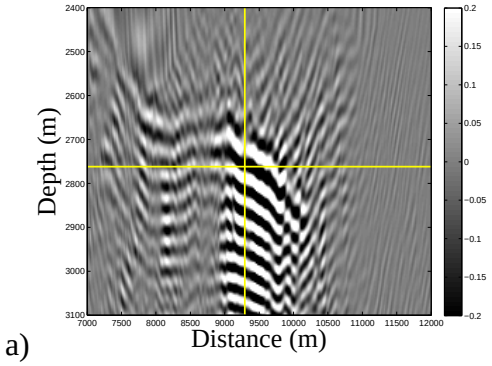
CICD is implemented as a pilot algorithm to combine PSDM and CD, which would otherwise involve user to execute each step.



a)

FIG. 15. Conventional imaging condition differencing result.

Figure 15 captures the result of CICD.



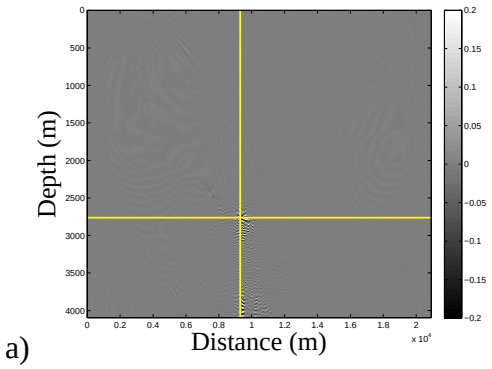
a)

FIG. 16. Zoomed in conventional imaging condition differencing result.

Figure 16 is the zoomed in Figure 15. CICD shows no improvements when compared to conventional differencing and migration result as expected. It proves the workflow for combining PCCD and PSDM possible.

ICD

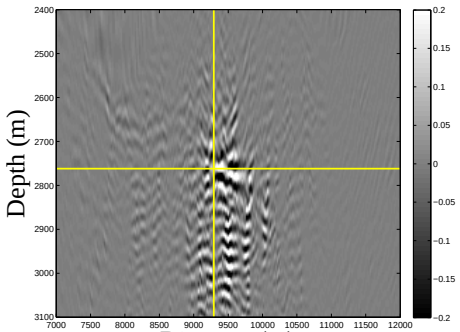
ICD is implemented as an algorithm to combine PSDM in PCCD.



a)

FIG. 17. Imaging condition differencing result.

Figure 17 captures the result of ICD.



a)

FIG. 18. Zoomed in imaging condition differencing result.

Figure 18 is the zoomed in Figure 17. The result shows improvements to the PCCD and the PSDM method as it eliminates dependence on the user to execute each step. The ICD result captures the box location and cleans data around the box when compared to the PCCD and the PSDM method.

EXAMPLE II

Velocity models and Synthetics

The second data set used to test the CCD and PCCD filtering is the 10^{th} SPE Comparative solution project. Here, I only employ filtering, there is no migration. Recall velocity and synthetic seismic models generated from Gassmann equations and finite difference algorithm in elastic medium presented in Vracar and Ferguson (2010) paper. Velocity models are shown in Vracar and Ferguson (2010) paper. Seismic synthetic models are shown in Milicevic and Ferguson (2009) paper.

CCD

Seismic synthetic models are passed to the CCD algorithm.

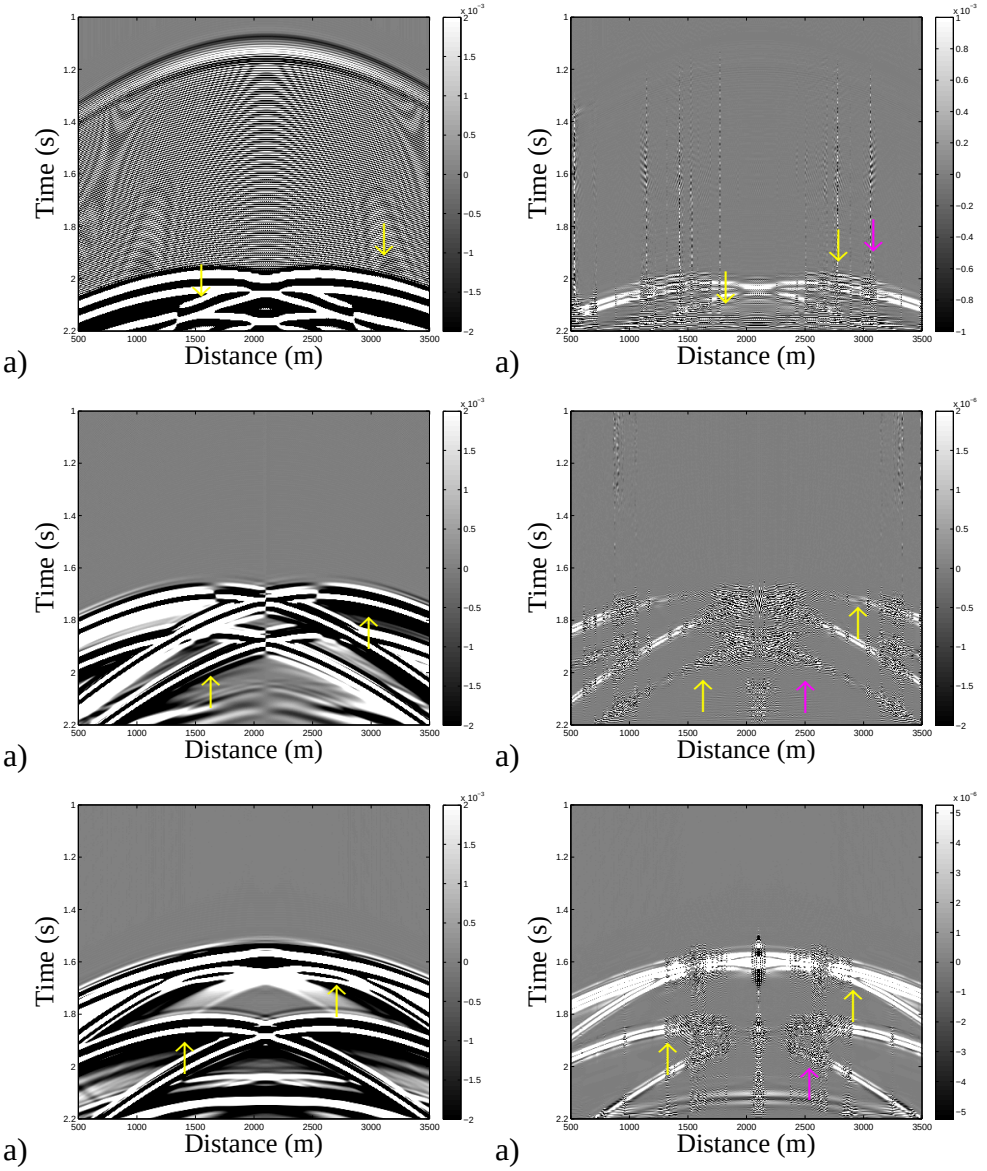


FIG. 19. CCD results on 3C-3D shot gather models: a) x-component conventional differencing of day 1 and 14, b) x-component CCD filtering of day 1 and 14, c) y-component conventional differencing of day 1 and 14, d) y-component CCD filtering of day 1 and 14, e) z-component conventional differencing of day 1 and 14, f) z-component CCD filtering of day 1 and 14. The yellow and magenta arrows denote waterfronts and numerical artifacts, respectively.

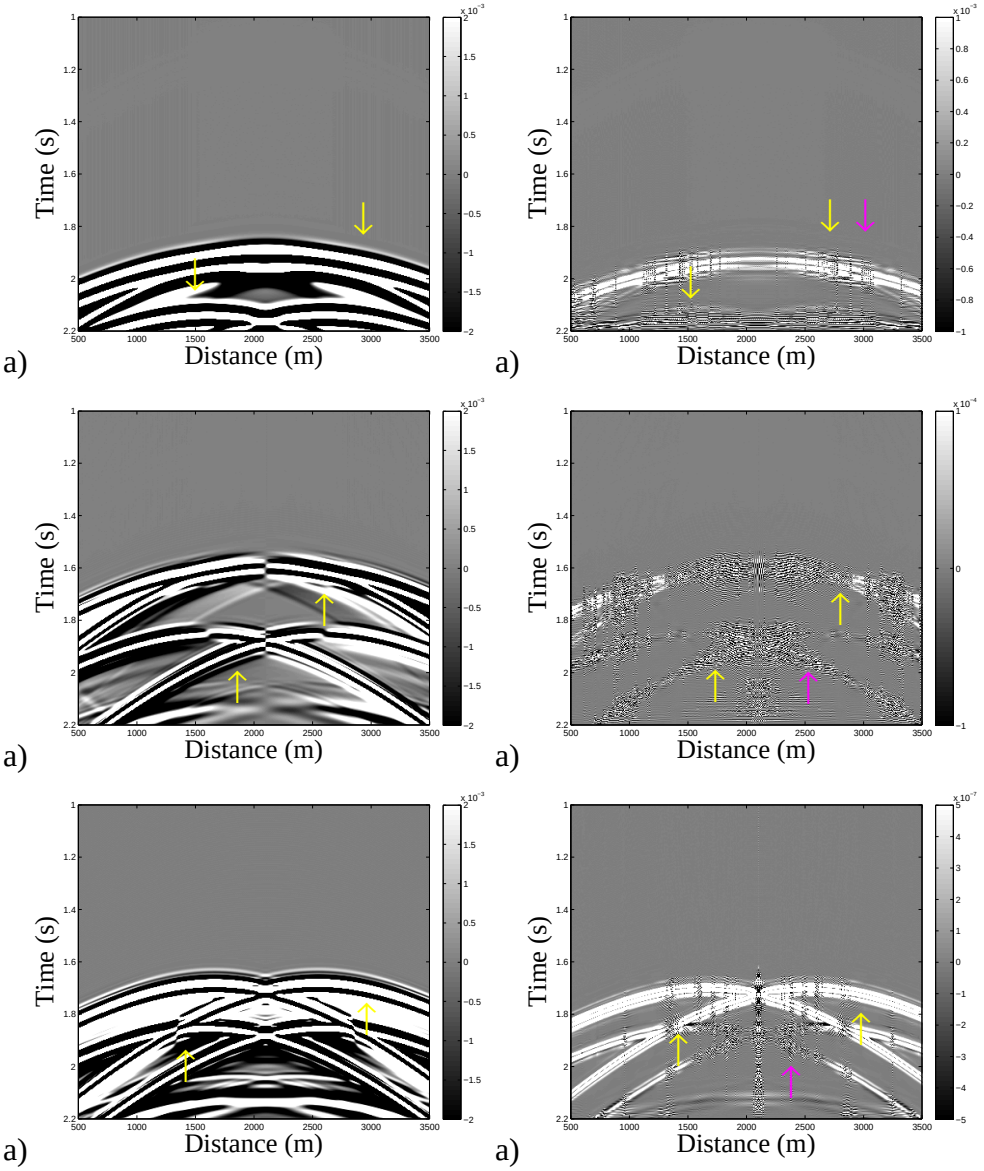


FIG. 20. CCD results on 3C-3D shot gather models: a) x-component conventional differencing of day 1 and 28, b) x-component CCD filtering of day 1 and 28, c) y-component conventional differencing of day 1 and 28, d) y-component CCD filtering of day 1 and 28, e) z-component conventional differencing of day 1 and 28, f) z-component CCD filtering of day 1 and 28. The yellow and magenta arrows denote waterfronts and numerical artifacts, respectively.

Figure 19 captures day 1 and day 14 differenced models. Models 19(a), 19(c) and 19(e) capture the CCD filtering of days 1 and 14 of x, y and z components, respectively. Models 19(b), 19(d) and 19(f) capture the CCD filtering of days 1 and 14 of x, y and z components, respectively. The yellow arrows point to the location of waterfronts after days 1 and 14. Figure 20 captures day 1 and day 28 differenced models. Models 20(a), 20(c) and 20(e) capture the CCD filtering of days 1 and 28 of x, y and z components, respectively. Models 20(b), 20(d) and 20(f) capture the CCD filtering of days 1 and 28 of x, y and z components, respectively. The yellow arrows point to the location of waterfronts after days 1 and 28. The magenta arrows point to the numerical artifacts produced by the

algorithm. When compared the CCD filtered data is much easier to interpret as it eliminates amplitude ambiguities around events.

PCCD

The same data set is used for testing the PCCD algorithm.

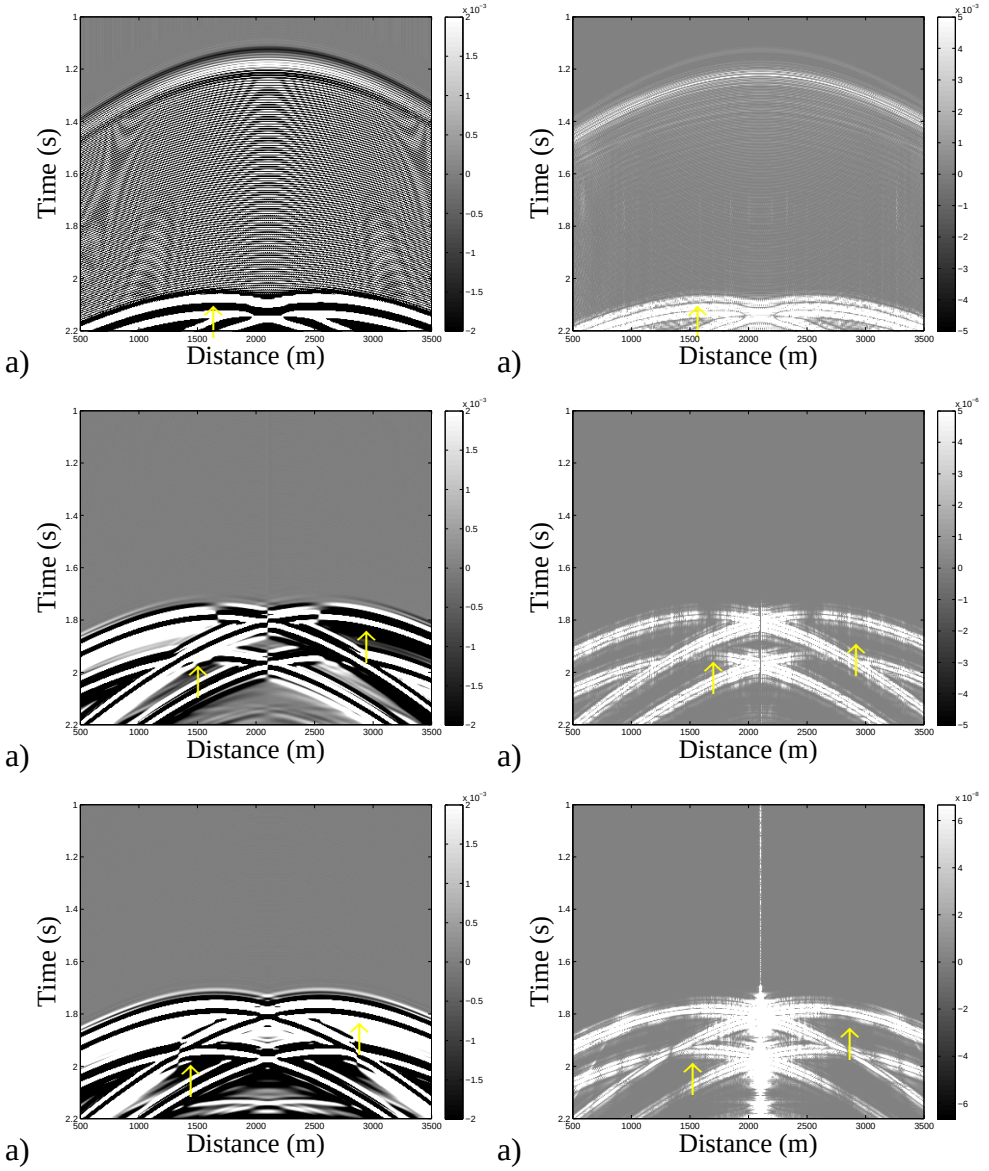


FIG. 21. PCCD results on 3C-3D shot gather models: a) x-component conventional differencing of day 1 and 14, b) x-component PCCD filtering of day 1 and 14, c) y-component conventional differencing of day 1 and 14, d) y-component PCCD filtering of day 1 and 14, e) z-component conventional differencing of day 1 and 14, f) z-component PCCD filtering of day 1 and 14. The yellow arrows denote waterfronts.

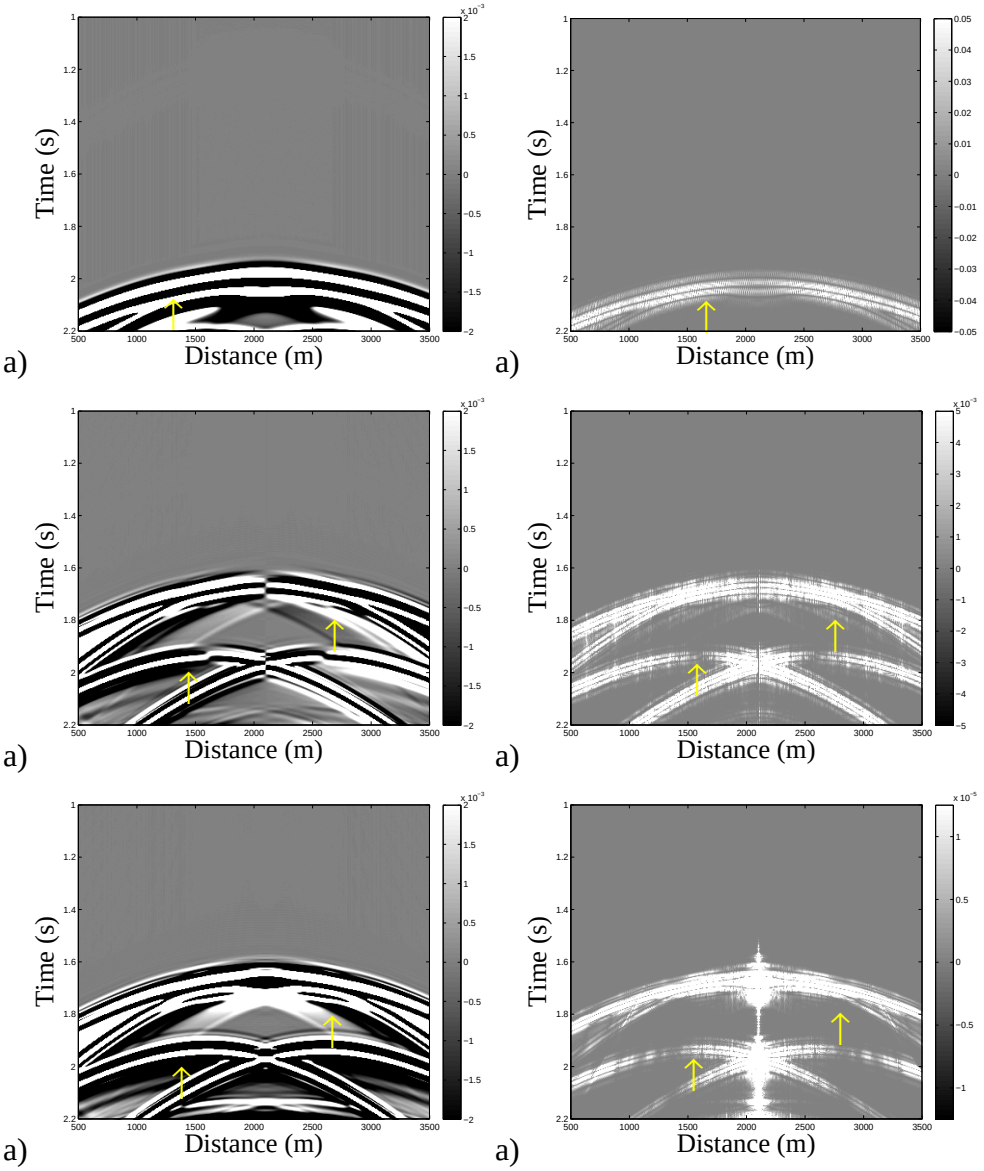


FIG. 22. PCCD results on 3C-3D shot gather models: a) x-component conventional differencing of day 1 and 28, b) x-component PCCD filtering of day 1 and 28, c) y-component conventional differencing of day 1 and 28, d) y-component PCCD filtering of day 1 and 28, e) z-component conventional differencing of day 1 and 28, f) z-component PCCD filtering of day 1 and 28. The yellow arrows denote waterfronts.

Figure 21 captures day 1 and day 14 differenced models. Models 21(a), 21(c) and 21(e) capture the CCD filtering of days 1 and 14 of x, y and z components, respectively. Models 21(b), 21(d) and 21(f) capture the CCD filtering of days 1 and 14 of x, y and z components, respectively. The yellow arrows point to the location of waterfronts after days 1 and 14. Figure 22 captures day 1 and day 28 differenced models. Models 22(a), 22(c) and 22(e) capture the CCD filtering of days 1 and 28 of x, y and z components, respectively. Models 22(b), 22(d) and 22(f) capture the CCD filtering of days 1 and 28 of x, y and z components, respectively. The yellow arrows point to the location of waterfronts after days 1 and 28. Note that numerical artifacts do not show on these plots. When

compared the PCCD filtered data is much easier to interpret as it eliminates amplitude ambiguities around events. When CCD and PCCD are compared, PCCD shows favorable due to imaging no numerical artifacts and it is computationally economic.

FUTURE WORK

As in examples there are some numerical artifacts on seismic models, hence algorithm optimization should be attempted. Computation time of CCD non-conventional algorithm still has potential for improvement. CCD employs matrix inversion, therefore, if its computation cost improves it will optimize the algorithm. Gaussian filter is used to eliminate zero-lag. It is worth considering other filtering methods to improve resolution.

CONCLUSIONS

We implement four seismic differencing methods: 1) cross-correlation differencing (CCD), 2) pseudo cross-correlation differencing (PCCD), 3) conventional imaging condition differencing (CICD) and 4) imaging condition differencing (ICD).

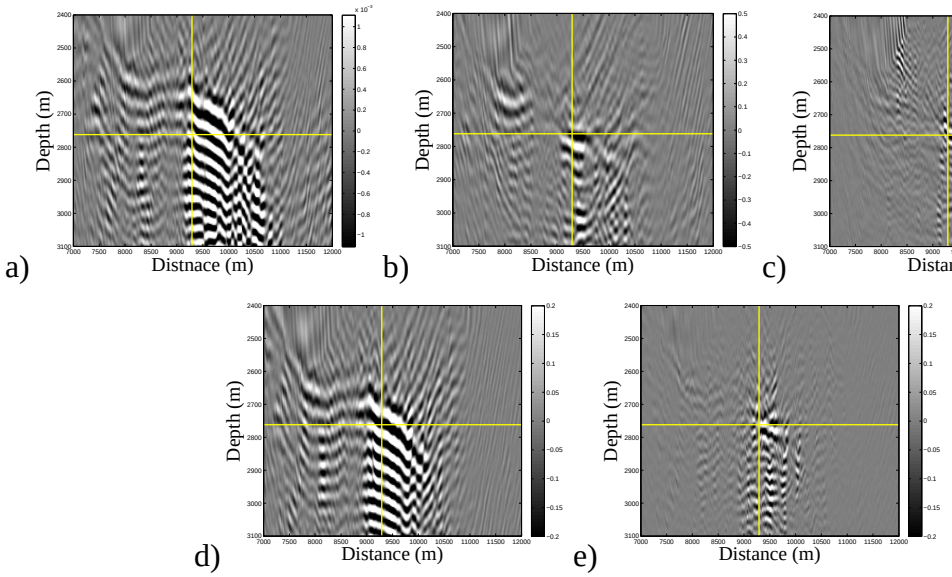


FIG. 23. Summary of differencing methods: a) CD, b) CCD, c) PCCD, d) CICD and e) ICD.

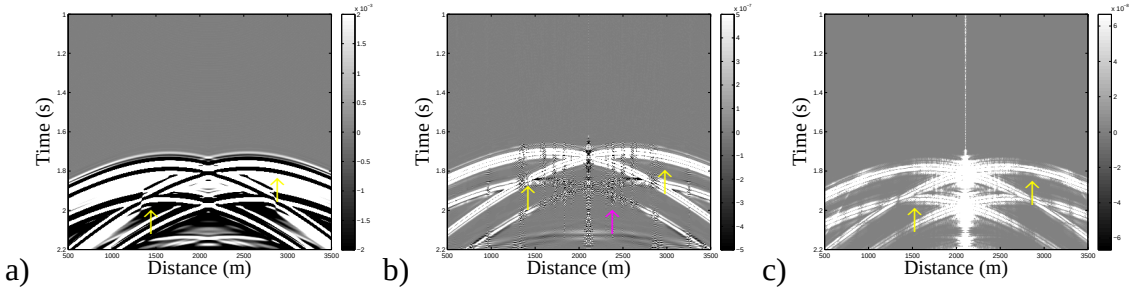


FIG. 24. Comparison of 3C-3D shot gather models z-component: a) CD, b) CCD and c) PCCD.

Figure 23 captures conventional differencing and four new seismic differencing algorithms for EGAE/SEG salt data. Figure 24 captures conventional differencing and four new seismic differencing algorithms for generated synthetic data. The CCD and PCCD algorithms are performed in time and frequency domains, respectively. They both depend on the user to execute each step. They are further followed by multiplication by conventional differencing and pre-stack depth migration (PSDM). The CCD and PCCD are considerable improvements to conventional differencing as they clearly show box location. CICD is a pilot algorithm to combine PSDM with PCCD. It proves to be efficient and robust when compared to conventional differencing, however, no imaging improvements are noted. The ICD method combines PCCD, differencing and migration in one algorithm, hence minimizes user's dependence and improves computational time and imaging. In Figures 23 and 24, CCD, PCCD and ICD highlight differences in time-lapse steps eliminating similarities by cross-correlation, filtering and inversion. Also, significant imaging improvements are noted, hence interpretation becomes easier.

ACKNOWLEDGMENT

We thank CREWES Project directors, staff, students and sponsors for support of this research.

REFERENCES

Aminzadeh, F., N. Burkhard, J. Long, T. Kunz, and P. Duclos, 1996, Three dimensional seg/eage models - an update: The Leading Edge, **15**, 131–134.

Bini, D., 1995, Toeplitz matrices, algorithms and applications: ECRIM News Online Edition, **22**.

Inannen, K., 2010, Signal analysis: Course notes, University of Calgary.

Knuth, D., 1997, The art of computer programming: Addison-Wesley.

Lines, L. R., and R. Newrick, 2008, Geophysical interpretation: SEG.

Margrave, G. F., 1998, Theory of nonstationary linear filtering in the fourier domain with application to time-variant filtering: Geophysics, **63**, 244 – 259.

———, 2008, Methods of seismic data processing: Course notes, University of Calgary and CREWES Project.

Milicevic, V., and R. J. Ferguson, 2009, Numerical fluid flow modelling and its seismic response in time-lapse: CREWES Research Report, **21**, 1–13.

Schlumberger, 2011, Oilfield glossary: online resourse: http://www.glossary.oilfield.slb.com/.

Stoffa, P. L., J. T. Fokkema, R. M. de Luna Freire, and W. P. Kessinger, 1990, Split-step Fourier migration: Geophysics, **55**, 410–421.

Vracar, V., and R. J. Ferguson, 2010, Conventional and non-conventional differencing in time-lapse: CREWES Research Report, **22**, 1–12.