On analysis-based two-step interpolation methods for randomly sampled seismic data

Pengliang Yang*, Jinghuai Gao, Wenchao Chen

Xi’an Jiaotong University, Xi’an 710049, China

ABSTRACT

Interpolating the missing traces of regularly or irregularly sampled seismic record is an exceedingly important issue in the geophysical community. Many modern acquisition and reconstruction methods are designed to exploit the transform domain sparsity of the few randomly recorded but informative seismic data using thresholding techniques. In this paper, to regularize randomly sampled seismic data, we introduce two accelerated, analysis-based two-step interpolation algorithms, the analysis-based FISTA (fast iterative shrinkage-thresholding algorithm) and the FPOCS (fast projection onto convex sets) algorithm from the IST (iterative shrinkage-thresholding) algorithm and the POCS (projection onto convex sets) algorithm. A MATLAB package is developed for the implementation of these thresholding-related interpolation methods. Based on this package, we compare the reconstruction performance of these algorithms, using synthetic and real seismic data. Combined with several thresholding strategies, the accelerated convergence of the proposed methods is also highlighted.

1. Introduction

Acquisition of reflection seismic data is a multidimensional resource-intensive process. However, fully recording the seismic data is unrealistic for many reasons: a finite number of active recording channels, surface obstacles, as well as some other physical and financial constraints (Curry, 2008). Therefore, the prestack trace interpolation of these uniformly or nonuniformly sampled seismic records prior to migration is seriously needed (Trickett et al., 2010).

There exist many effective methods to reconstruct the complete seismic wavefield depending on different acquisition geometries (Trickett et al., 2010; Naghizadeh and Sacchi, 2010). These methods can be roughly classified into two categories: (1) Interpolation using the linear, spatial prediction filters (see Spitz, 1991; Porsani, 1999; Wang, 2002, just name a few), which solve the problems of regular sampling pattern successfully; (2) Reconstruction with the aid of certain type of transforms such as the Fourier transform (Sacchi et al., 1998; Liu and Sacchi, 2004; Xu et al., 2005; Abma and Kabir, 2006; Naghizadeh, 2007), the Radon transform (Trad and Ulrych, 2002; Trad et al., 2003), the curvelet transform (Hennenfent, 2008; Hennenfent et al., 2010; Naghizadeh and Sacchi, 2010; Yang et al., 2011), etc., The latter is intended to exploit the transform domain sparsity and works well for irregularly sampled seismic records.

Seismic data interpolation in the sparse domain is an ongoing research area in the geophysical community. It is supported by the new proposed theory of compressive sensing (CS) which proves that perfect reconstruction can be achieved for the few randomly sampled data, using the sparsifying transformation. As Hennenfent (2008) said: ‘random undersampling renders coherent aliases into harmless incoherent random noise, effectively turning the interpolation problem into a much simpler denoising problem’. See the illustration in Fig. 1. Today thresholding denoising-related methods are naturally introduced into the geophysical world to handle the few sampled, aliased but still informative seismic data. Based on the basis pursuit denoising (BPDN) model (Chen et al., 2001), Herrmann and Hennenfent (2008) and Hennenfent et al. (2010) applied the iterative shrinkage-thresholding (IST) algorithm to interpolation of nonuniformly recorded seismic traces. Another existing thresholding algorithm, namely the projection onto convex sets (POCS) method introduced by Abma and Kabir (2006), has been proved to solve this seismic interpolation problem equivalently (Yang et al., 2011).

Due to the slow convergence rate of these existing algorithms, researchers have studied the techniques to improve the convergence performance: Beck and Teboulle (2009) proposed a fast iterative shrinkage-thresholding algorithm (FISTA) from a more general mathematical background; Gao et al. (2010) and Yang et al. (2011) utilized the exponential decreasing thresholding strategy as the substitute of the linear one employed by
Abma and Kabir; Gao et al. (2011) even proposed a data-driven schedule. In this paper, we aim to recast the problem of the sparsity-promoting seismic data interpolation as an analysis formulation, using the tight frame-based transform; present the IST algorithm in the analysis fashion, different from the synthesis formula in Hennenfent and Herrmann (2006); reestablish the close connection between the IST algorithm and the POCS algorithm; present two two-step interpolation algorithms with accelerated convergence rate, namely the analysis-based FISTA method and the fast POCS (FPOCS) method; develop a self-contained MATLAB implementation package to compare the interpolation performance of these methods, using the typical synthetic and real seismic data.

2. The sparsity-promoting seismic data interpolation

2.1. Problem statement

Now let us formulate the problem of seismic trace interpolation. The observed seismic record \(d_{\text{obs}}\), including many missing samples, is connected with the complete seismic data \(d\) to be recovered via the relation:

\[
d_{\text{obs}} = Md
\]

in which \(M\) denotes a diagonal matrix with diagonal entries 1 for the observed samples and 0 otherwise. For the convenience of mathematical expression, a 2D discrete seismic data set \(d = [d_{t_1,n_1}, \ldots, d_{t_{n_1}n_2}] \in \mathbb{R}^{n_1 \times n_2}\) can be reordered as a vector \(d \in \mathbb{R}^n\), \(n = n_1 \times n_2\) through lexicographic ordering \(d_{t_1,n_1} \rightarrow d_{t_2-1,n_1+1}\). This kind of expression can be easily generalized to higher dimensional cases. To recover the missing traces in the seismic data, one needs to project the seismic data into certain transform domain, that is,

\[
d_{\text{obs}} = Ax,
\]

where \(A \in \mathbb{R}^{m \times n}\) corresponds to a synthetic operator of an orthogonal basis or a frame. This leads to

\[
d_{\text{obs}} = Md = Max,
\]

where \(x\) indicates the representation coefficients in the transform domain. Such a problem is always underdetermined, and some priori knowledge is needed. The sparsity in certain transform domain always holds well from a large number of informed studies, and can be taken as a reasonable constraint (Lustig et al., 2008), which corresponds to the \(\ell^0\)-norm of the unknown \(x\). Due to the nonconvex complexity of the \(\ell^0\)-norm minimization, \(\ell^1\)-norm is always taken as an alternative to \(\ell^0\)-norm, and has proved to be able to achieve the same sparseness as the \(\ell^0\)-norm (Donoho, 2006). The regularization problem becomes

\[
\min_x f(x) = \frac{1}{2}\|d_{\text{obs}} - Kx\|^2_2 + \tau \|x\|_1,
\]

where \(K = MA\). This is a synthesis formulation since the complete seismic data \(d\) can be synthesized from its representation coefficients, \(d = Ax\), where \(\hat{x}\) is the minimizer of problem (4). Obviously,
the problem of seismic data interpolation is well accommodated within the context of compressive sensing (CS) (Candes, 2006; Donoho, 2006).

Consider a tight frame such that $A^*A = I$ and $x = A^*d = A^*d$, where the symbol * indicates the adjoint. Then, the problem can be rewritten as

$$\min j(d) = \frac{1}{2}d_{\text{obs}} - Md_{\text{data}}^2 + \nu(A^*d)_{\ell_1}.$$  \hfill (5)

It is very important to point out that (5) is referred to as the analysis formulation since it directly analyzes the complete seismic record $d$ as the unknown (Elad et al., 2007).

2.2. IST algorithm and its analysis version

There exist a number of algorithms to solve the problem in (4). Figueiredo and Nowak (2003) and Daubechies et al. (2004) proposed the well-known iterative shrinkage-thresholding (IST) algorithm, which can be generally expressed as

$$x(k+1) = T_{\lambda}(x(k) + K^\#(d_{\text{obs}} - Kx(k))), \quad k = 1, 2, \ldots, N$$  \hfill (6)

in which $\lambda$ is a positive threshold parameter. $N$ is the total number of iterations of this algorithm. $T$ corresponds to the soft thresholding operator $S$ with the threshold value $\lambda = \tau$ (Daubechies et al., 2004). Even though the $\ell^2$ penalty is formulated in (4) and (5) due to its strict convex property, it is also very important to point out that $T$ is related to the hard thresholding operator $H$ with the threshold $\lambda = \sqrt{2}\tau$ when the $\ell^2$-norm is taken as the penalty term instead of $\ell^1$-norm (Wright et al., 2009). Here, the soft and hard thresholding operators performed elementwise are defined as

$$S_{\lambda}(u) = \begin{cases} u - \lambda \frac{u}{|u|} & |u| > \lambda \\ 0 & |u| \leq \lambda \end{cases} \quad \hfill (7)$$

and

$$H_{\lambda}(u) = \begin{cases} u, & |u| > \lambda \\ 0, & |u| \leq \lambda \end{cases} \quad \hfill (8)$$

respectively. Recall that $A$ is a tight frame, $x = A^*d$, $K = MA$. $M$ is a diagonal matrix such that $M^* = M = M^2$, $M^*d_{\text{obs}} = M^2d = d_{\text{obs}}$. From (6) we obtain the analysis-based IST algorithm

$$d^{k+1} = Ax^{k+1} = AT_j[x^k + K^\#(d_{\text{obs}} - Kx^k)]$$

$$= AT_j[A^*d^k + (MA)^\#(d_{\text{obs}} - MAx^k)]$$

$$= AT_j[A^*d^k + M^\#(d_{\text{obs}} - Md^k)]$$

$$= AT_j[A^*d_{\text{obs}} + (I - M)d^k].$$  \hfill (9)

2.3. Another analysis-based algorithm: the POCS method

Now we define

$$u(k) = d_{\text{obs}} + (I - M)d^k, \quad k = 1, 2, \ldots, N.$$  \hfill (10)

Clearly, $u(k)$ denotes the interpolated data including the original seismic traces at the $k$th iteration. In terms of (9), we have

$$d^{k+1} = AT_j[A^*u^k], \quad k = 1, 2, \ldots, N.$$  \hfill (11)

Using the relation (10) at the $k+1$th iteration again, it holds

$$u^{k+1} = d_{\text{obs}} + (I - M)AT_j[A^*u^k].$$  \hfill (12)

Note that the convergence of the IST algorithm guarantees

$$\lim_{k \to \infty} d^k = d$$  \hfill (Daubechies et al., 2004), thus we obtain

$$\lim_{k \to \infty} u^k = \lim_{k \to \infty} d_{\text{obs}} + (I - M)d^k = d.$$  \hfill (13)

On the other hand, $M(I - M) = 0$, we have

$$Mu^k = d_{\text{obs}}, \quad k = 1, 2, \ldots, N$$  \hfill (14)

according to (11). In light of this interpretation, we use $d^k$ to replace $u^k$ in (11), obtaining another updating rule

$$d^{k+1} = d_{\text{obs}} + (I - M)AT_j[A^*d^k].$$  \hfill (15)

Immediately, the well-known POCS algorithm is derived from the analysis-based IST algorithm. This algorithm has been widely used in many applications, such as image reconstruction and inpainting (Gunturk et al., 2002; Gurleyuz, 2006a,b; Cai and Chan, 2008), and seismic data interpolation (Abma and Kabir, 2006).

2.4. Two-step variants: analysis-based FISTA and FPOCS

It is a well-known fact that the convergence rate of the IST algorithm is very slow ($\mathcal{O}(1/k^2)$). Beck and Teboulle (2009) proposed a fast iterative shrinkage-thresholding algorithm (FISTA) with improved convergence rate ($\mathcal{O}(1/k^4)$), which can be specified as

$$\begin{cases} x(k+1) = x(k) + \frac{1 - \alpha}{\alpha}d(k) - \frac{\alpha - 1}{\alpha}d(k-1), \\ x^{k+1} = T_{\lambda}(x(k) + K^\#(d_{\text{obs}} - Kx^{k+1})), \end{cases} \quad \hfill (16)$$

with $x(0) = 0$, $t(0) = 1$ and $t(k+1) = (1 + \sqrt{1 + 4(t(k))^2})/2$. This algorithm can be considered as a variant of the two-step iterative shrinkage-thresholding (TwIST) algorithm (Bioucas-Dias and Figueiredo, 2007). Even though the FISTA algorithm has been widely used in signal and image processing as well as many other related fields, to the best of our knowledge, most of the applications of this method focus on its synthesis version (15).

In the same vein as the derivation of (9), by applying $A$ to both sides of (15), we obtain the analysis-based FISTA algorithm

$$\begin{cases} d(k) = d(k) + \frac{1 - \alpha}{\alpha}d(k) - d(k-1), \\ d^{k+1} = AT_j[A^*d_{\text{obs}} + (I - M)d^k]. \end{cases} \quad \hfill (17)$$

Motivated by this analysis-based FISTA algorithm and the same idea of the derivation in Section 2.3, a fast POCS (FPOCS) algorithm can be developed with ease:

$$\begin{cases} d(k) = d(k) + \frac{1 - \alpha}{\alpha}d(k) - d(k-1), \\ d^{k+1} = d_{\text{obs}} + (I - M)AT_j[A^*d^k]. \end{cases} \quad \hfill (18)$$

It is also important to point out that this may be the first time that the analysis-based FISTA and the FPOCS algorithms are introduced into the world of seismic data interpolation. As will be demonstrated by our numerical results, the analysis-based FISTA algorithm and the FPOCS algorithm indeed directly solve the minimization problem (5) with improved convergence speed.

We remark that all the above discussions focus on interpolation in noise free cases. To alleviate the fictitious features caused by POCS reconstruction, Gao et al. (2011) suggested replacing $d_{\text{obs}}$ with $\varepsilon d_{\text{obs}} + (1 - 2\varepsilon)MAAT_j[A^*d^k]$ in Eq. (14), leading to

$$d^{k+1} = \varepsilon d_{\text{obs}} + (1 - 2\varepsilon)MAAT_j[A^*d^k].$$  \hfill (19)

in which $\varepsilon \in [0, 1]$. Obviously, we obtain the standard POCS solver when $\varepsilon = 1$. Naturally, we can include noise attenuation by adding a reinsertion step in our FPOCS algorithm, that is,
3. MATLAB implementation

3.1. Thresholding strategies

The regularization parameter $\tau$ plays an important role in these thresholding-related interpolation algorithms, and is closely related to the threshold $\lambda$ at each iteration:

(Soft thresholding) : $\lambda = \tau$

(20)

(Hard thresholding) : $\lambda = \sqrt{2\tau}$

(21)

In our MATLAB implementation, the following four thresholding schemes are available:

(constant) : $\tau_k = \text{const.}, \ k = 1, 2, \ldots, N$

(22)

(linear) : $\tau_k = \tau_{\text{max}} - \frac{k-1}{N-1}(\tau_{\text{max}} - \tau_{\text{min}}), \ k = 1, 2, \ldots, N$

(23)

(exponential) : $\tau_k = \left(\frac{\tau_{\text{max}}}{\tau_{\text{min}}}\right)^{(\frac{k-1}{N-1})} \cdot \tau_{\text{max}}, \ k = 1, 2, \ldots, N$

(24)

(data_driven) : $\tau_1 = \nu_1, \ \tau_k = \nu_j, \ \tau_N = \nu_N$, $j = \left\lceil (k-1)N_v/(N-1) \right\rceil, \ k = 2, \ldots, N-1$

(25)

with

\[
\begin{align*}
\tau_{\text{max}} &= \max_i \left\{ \| A^* d_{\text{obs}} \| \right\}, \\
\tau_{\text{min}} &= \min_i \left\{ \| A^* d_{\text{obs}} \| \right\},
\end{align*}
\]

(26)

where $\tau_{\text{max}}$ and $\tau_{\text{min}}$ stand for the chosen maximum and minimum regularization parameter, respectively. $p_{\text{max}}$ and $p_{\text{min}}$ are user defined percentages. $\nu$ denotes the vector by reordering the amplitudes of $\| A^* d_{\text{obs}} \| \in [\tau_{\text{min}}, \tau_{\text{max}}]$ in descending order, and $N_v$ its length. The symbol $\lceil \rceil$ indicates the smallest integer not less than $x$. It is also interesting to note that Abma and Kabir (2006) employed a linear decreasing scheme to improve the convergence of POCs implementation.\(^2\) The convergence rate of these algorithms proves to be remarkably improved by the exponential and the data-driven decreasing thresholding schemes in Eqs. (24) and (25) (Gao et al., 2010, 2011; Yang et al., 2011). The linear and exponential decreasing thresholding strategies are graphically plotted in Fig. 2. More details of the later three schemes can be found in Gao et al. (2011). As will be demonstrated in Section 4, combined with these thresholding schedules, the accelerated convergence of these two-step methods (FISTA and FPOCS) is clearly observed after many iterations.

3.2. Stopping criterion and performance evaluation

Note that the above thresholding schemes imply that the algorithms will not be stopped until a user defined maximum number of iterations $N$ is reached. To terminate the algorithms in time, another dynamic stopping criterion is designed:

\[
\| d^{k+1} - d^k \|^2 < \text{tolerance}
\]

(27)

\(^2\) Abma and Kabir use hard thresholding and $\lambda_k = \tau_k = (N-k)/N \max_i \| (A^t d^k)^i \|$ in Eq. (14). See http://www.freeusp.org/synthetics/POCS_example/try3.m for more details.

\(^1\) In our implementation, we use $p_{\text{max}} = 99\%$, $p_{\text{min}} = 1/\text{nt}$.

In which tolerance should be set prior to the execution of the algorithms. Of course, the other stopping criterion using cross correlation can also be added if no improvement of the interpolated results can be achieved after many iterations, see Herrmann et al. (2007) for an example.

To evaluate the quality of seismic data reconstruction, we define signal-to-noise-ratio (SNR) as noted in Hennenfent and Herrmann (2006):

\[
\text{SNR} = 10 \log_{10} \left( \frac{\| d^f \|^2}{\| d - d^f \|^2} \right) \text{ (dB)},
\]

(28)

where $d^f$ denotes currently reconstructed seismic data. The subroutine compute_snrm.m is devoted to fulfill this evaluation.

3.3. Implementation illustration

3.3.1. Sampling and algorithms coding

Here we list the structure of our MATLAB implementation to interpolate seismic data, as shown in Fig. 3. Random sampling is
performed by `random_sampling.m`. Note that the regular sampling pattern and sampling including large gaps (many empty traces) are out of the scope of this paper even though we have coded the subroutines (coined `regular_sampling.m` and `gap_sampling.m`), because they violate the random sampling condition required by the theory of CS. The interested readers are encouraged to carry out some additional tests.

The analysis-based thresholding interpolation algorithms are coded as `ist_interp.m`, `fista_interp.m`, `pocs_interp.m` and `fpocs_interp.m`. `next_t.m` will be repeatedly called by `fista_interp.m` and `fpocs_interp.m`.

### 3.3.2. Fields of the input and the output variables

The input variable `pars` of the four aforementioned methods is made up of five fields:

- `pars.niter` indicates the maximum number of iterations $N$;
- `pars.decrease_mode` specifies the decreasing mode for the regularization parameter $\tau$, and may be ‘constant’, ‘linear’, ‘exponential’ or ‘data_driven’;
- `pars.thresholding` may be ‘hard’ or ‘soft’ thresholding;
- `pars.tolerance` provides the tolerance so that these algorithms can be stopped in advance until the maximum number of iterations is reached.
- and `pars.alpha` denotes the parameter $\alpha$ in Eq. (19) to include noise attenuation.

The output includes the reconstructed seismic data and a report, whose fields consist of:

- `report.snr`, evaluated by `compute_snr.m`;
- and `report.iters`, returning the number of iterations actually used by adding the dynamic stopping rule to monitor the execution process of these algorithms.

### 3.3.3. Tight frame-based transform invoked: FFT

The methods in this paper are applicable to all the tight frame-based transforms. There are many such nice transforms available, such as the Fourier transform, various of orthogonal wavelet transforms, as well as the redundant transforms. Curvelet transform has been successfully applied by Hennenfent et al. (2010), and also tested with the IST algorithm and POCS algorithm by us in Yang et al. (2011). Even so, we mainly consider the fast Fourier transform (FFT) to analyze the seismic data. The main reason is that FFT has been well developed with most efficient implementation available on all kinds of computational platforms. Some other tight frame-based transforms such as the curvelet transform still rely on FFT with a little more computational complexity and memory requirement. FFT is an excellent candidate for the

---

According to our experience, the implementation of the curvelet transform on 2D seismic data with size $2048 \times 2048$ seems prohibitive in MATLAB with 2 GB RAM PC. Let alone the 3D curvelet transform.

---

Fig. 4. Experimental seismic data. (a) Synthetic data 1 (produced using 35 Hz Ricker wavelet): 512 traces, 512 samples in each trace, sampling interval $= 0.002$ s.
(b) Synthetic data 2 (one shot produced using Marmousi model, first arrival muted): 201 traces, 600 samples in each trace, sampling interval $= 0.003$ s. (c) Field data (one shot from marine seismic data): 430 traces, 512 samples in each trace, spatial interval $= 12.5$ m, merely $0.124-2.124$ s. (a’–c’) The irregularly sampled data of panels a–c of this figure, random decimating rate $= 30\%$, SNR $= 5.24579$ dB, $5.29779$ dB, $4.80488$ dB.
Fig. 5. The SNR curves with different thresholding strategies (test data: synthetic data 1). (a) Constant thresholding, $\tau = 0.005$, $N = 200$. (b) Constant thresholding, $\tau = 0.01$, $N = 100$. (c–d) Linear decreasing thresholding, $N = 200, 100$. (e–f) Exponential decreasing thresholding, $N = 200, 100$. (g–h) Data-driven thresholding, $N = 200, 100$. 

P. Yang et al. / Computers & Geosciences 51 (2013) 449–461
practical multidimensional applications, even though multidimensional implementation is not the goal of this paper and we do not provide such examples here.

4. Numerical results and discussions

In this section, we will compare the IST algorithm, the POCS algorithm, the FISTA algorithm and the FPOCS algorithm with two synthetic seismic data sets and one field marine data, as shown in Fig. 4. These data sets are randomly decimated with a 30% decimating rate.

4.1. Synthetic example 1

The first numerical test is based on the synthetic data 1 (Fig. 4a). Different thresholding strategies are tested with

![Interpolated results using the exponential decreasing thresholding for the synthetic data 1, N = 100.](image)

(a) IST-Soft thresholding interpolated, SNR = 39.0042 dB. (b) FISTA-Soft thresholding interpolated, SNR = 40.9801 dB. (c) POCS-Soft thresholding interpolated, SNR = 39.0044 dB. (d) FPOCS-Soft thresholding interpolated, SNR = 40.9803 dB. (e) IST-Hard thresholding interpolated, SNR = 41.4019 dB. (f) FISTA-Hard thresholding interpolated, SNR = 37.8882 dB. (g) POCS-Hard thresholding interpolated, SNR = 44.8484 dB. (h) FPOCS-Hard thresholding interpolated, SNR = 39.1458 dB. (a’–h’) The difference between the complete synthetic data 1 in Fig. 4a and panels a–h of this figure.
Fig. 7. The SNR curves with different maximum number of iterations $N$ (test data: synthetic data 2). (a–d) Exponential decreasing thresholding, $N = 10, 35, 50, 100$. (a’–d’) Data-driven thresholding, $N = 10, 35, 50, 100$. 

P. Yang et al. / Computers & Geosciences 51 (2013) 449–461
hundreds of iterations. The two regularization parameters are chosen in the constant thresholding scheme: \( \tau = 0.005, 0.01 \). As shown in Fig. 5a and b, for the same method, the smaller regularization parameter (\( \tau = 0.005 \)) needs more iterations (about 160 iterations) to converge but with higher interpolation SNR, while the larger regularization parameter (\( \tau = 0.01 \)) needs less iterations (about 80 iterations) to converge, corresponding to lower SNR. At the beginning of the interpolation, the linear decreasing thresholding scheme seems very inefficient; after many iterations, the regularization parameter becomes rather small and the reconstruction comes into play more and more, see Fig. 5c and d. Among all the four thresholding strategies, the exponential decreasing thresholding schedule and the data-driven one exhibit excellent convergent rate, according to the SNR measure (Fig. 5e–h). A satisfactory

Fig. 8. The satisfying interpolation results obtained with 35 iterations for the synthetic data 2 using exponential decreasing schedule. (a) IST-Soft thresholding interpolated, SNR = 13.1419 dB. (b) FISTA-Soft thresholding interpolated, SNR = 13.6316 dB. (c) POCS-Soft thresholding interpolated, SNR = 13.1419 dB. (d) FPOCS-Soft thresholding interpolated, SNR = 13.6342 dB. (e) IST-Hard thresholding interpolated, SNR = 13.5518 dB. (f) FISTA-Hard thresholding interpolated, SNR = 9.34111 dB. (g) POCS-Hard thresholding interpolated, SNR = 8.7847 dB. (h) FPOCS-Hard thresholding interpolated, SNR = 9.6547 dB. (a’–h’) The difference between the complete synthetic data 2 in Fig. 4b and panels a–h of this figure.
interpolated result using the exponential decreasing schedule is plotted in Fig. 6.

It is noteworthy that the proposed two-step methods (the analysis-based FISTA method and the FPOCS method) remarkably improve the convergence speed of the IST algorithm and the POCS algorithm, particularly for the constant thresholding scheme. Such kind of convergence acceleration of the proposed methods is still observable with the exponential decreasing thresholding strategy and the data-driven method in all the SNR curves of the following interpolation results. The latter two methods will therefore be extensively utilized in the remainder.

4.2. Synthetic example 2

Obviously, in the constant thresholding method the const $\lambda$ is sharply varied in terms of the input data, and hard to be chosen, even though a good convergence rate may be obtainable with appropriately chosen const. Due to the low convergence behavior of the linear decreasing thresholding method, we turn to the exponential and the data-driven schedule with another example using the synthetic data 2 (Fig. 4b, a more complicated shot produced using the Marmousi model, first arrival muted). The purpose of designing this example is twofold: (1) By showing another more complicated data set, we further confirm that the

Fig. 9. Interpolated results for the field data, $N = 50$, tolerance $= 10^{-6}$ using data-driven thresholding. (a) IST-Soft thresholding interpolated, SNR = 11.6762 dB, 50 iterations actually used. (b) FISTA-Soft thresholding interpolated, SNR = 11.6141 dB, 50 iterations actually used. (c) POCS-Soft thresholding interpolated, SNR = 11.6776 dB, 45 iterations actually used. (d) FPOCS-Soft thresholding interpolated, SNR = 11.6141 dB, 50 iterations actually used. (e) IST-Hard thresholding interpolated, SNR = 4.8032 dB, 29 iterations actually used. (f) FISTA-Hard thresholding interpolated, SNR = 4.79925 dB, 40 iterations actually used. (g) POCS-Hard thresholding interpolated, SNR = 4.80406 dB, 17 iterations actually used. (h) FPOCS-Hard thresholding interpolated, SNR = 4.80149 dB, 36 iterations actually used.
proposed two-step methods indeed exhibit better convergence performance. (2) Noting that too many iterations are not necessary and impractical in real applications, a reasonable total number of iterations $N$ should be given.

This synthetic data is tested with the iterations, $N = 10, 35, 50, 100$. The corresponding SNR curves are depicted in Fig. 7, using the exponential method (Fig. 7a–d) and the data-driven method (Fig. 7a’–d’). Again, we see that at the early iterations, with soft thresholding the proposed two-step methods slightly outperform the original one-step methods and improve the convergence rate to some extent. It is shown that the reconstruction extremely deteriorates due to the complexity of the observed data, compared with the satisfying recovering performance in the first example. It seems that a well-converged solution can be obtained within 35 iterations in this example (Fig. 8), and more iterations improve the interpolation very little. Empirically, $N \in [30, 50]$ is enough to achieve a satisfactory interpolation.

4.3. Real data example

In the previous two examples, it is very important to note the following. (1) Even though hard thresholding is possible to be superior to soft thresholding in terms of the final interpolated seismic record according to the SNR measure (see the SNR curves in synthetic example 1, Fig. 5e–h), its interpolation performance is not robust enough (see Fig. 7). In the exponential decreasing scheme, some SNR curves of hard thresholding first reach a peak and then decline. In the data-driven method, the interpolation SNR is poor. (2) Some algorithms can obtain better reconstructed seismic data before the maximum iteration number $N$ is reached. This indicates that it is very necessary to add the dynamic stopping criterion in the implementation of these algorithms.

Here we provide the interpolated result (Fig. 9) of a real marine seismic record (Fig. 4c). We choose $N=50$ and add the dynamic stopping rule in Eq. (27) by setting $tolerence = 10^{-6}$ such that these algorithms can be terminated in time. Note that hard thresholding fails in this example, see Figs. 9 and 10. To show true cause of the failure of hard thresholding, we have plotted the associated SNR curves with and without the dynamic stopping criterion, using the exponential decreasing thresholding and the data-driven scheme, see Fig. 10. Soft thresholding exhibits reliable and robust interpolation performance in our test; hard thresholding fails even though we do not use the dynamic stopping criterion. We believe this may arise from the nonconvex complexity of the $\ell^0$ constraint, which is closely related to hard thresholding (Loris et al., 2010).

4.4. Discussions

It is interesting to note that in the field data example the convergence rate of the data-driven schedule is slightly poorer than that of the exponential decreasing thresholding method, in contrast to the previous two synthetic examples. It is also worth noting that the SNRs of POCS and FPOCS algorithms at the early iterations are always higher than that of IST and FISTA algorithms.
In fact, it is a natural result because we use zeros to initialize \( d^{(k)} \) for all algorithms. In POCS-type methods it always holds that \( Md^{(k+1)} = d_{arb} = Md, \; \nu \) (cf. Eq. (13)), thus \( \|d^{(k+1)} - d\| = \|d-Md^{(k+1)} - d\| \). That is why the starting points of the SNR curves of POCS-type methods are always not 0 dB (about 5 dB due to random decimating). But in IST-type methods, a large value is taken as the threshold at the starting point \( t_k \approx t_{\max} \) when \( k \) is very small, cf. (26), almost all of the Fourier coefficients are thrown away, leading to too many negligible values in \( d^{(k+1)} \) such that \( \|d^{(k+1)}\|^2 = \|d\|^2 \) and SNR \( \approx 0 \) (cf. Eqs. (9) and (28)). It seems that at the early iterations, to obtain comparable performance of interpolation as POCS-type methods, a better thresholding strategy is required in the IST-type methods. For moderate \( N \), when \( k \rightarrow N \), the IST method can achieve the same performance as the POCS algorithm, as shown in the previous SNR curves. This is easy to understand in consideration of Eq. (12).

From the numerical results above, we remark the following.

- The proposed two-step methods (the analysis-based FISTA algorithm and the FPOCS algorithm) indeed exhibit convergence acceleration for seismic data interpolation, compared with the one-step methods (the IST algorithm and the POCS algorithm). Combined with good thresholding schedules, the acceleration is still observable even though not significant sometimes.

- The required iterations can be dramatically reduced by combing a good thresholding schedule and the dynamic stopping criterion. Empirically, a reasonable number of total iterations \( N \) can be controlled within 30–50. The dynamic stopping criterion is expected to obtain a satisfactory result with fewer iterations.

- Soft thresholding deserves to be adopted in the proposed two-step methods due to its reliable and robust reconstruction performance. Even though hard thresholding is possible to obtain good interpolated result, it corresponds to the \( \ell^0 \) constraint. This nonconvex penalty is closely related to a complicated combinatorial problem, perhaps leading to the failure of interpolation.

Additionally, we comment that it is valuable to observe the interpolated result with other different random decimating rate, not merely 30%. We do not show these results any more for several reasons: First, they can be obtained with ease using our MATLAB package, and the interested readers are encouraged to do more tests, such as adding reinsertion step to include noise attenuation with different \( \xi \) (in our numerical results \( \xi = 1 \)). More importantly, the main contribution of this paper is that we proposed two accelerated analysis-based two-step interpolation algorithms. More attention should be paid to the improved convergence between the two-step methods and the standard one-step method (in all the SNR curves we should care about the differences of FPOCS vs. POCS, FISTA vs. IST). Combined with good thresholding schedules and the dynamic stopping rule, the superiority of the proposed methods can be clearly observed.

5. Conclusions

In this paper we derived two accelerated, analysis-based two-step interpolation algorithms, the analysis-based FISTA algorithm and the FPOCS algorithm, from the IST algorithm and the POCS algorithm. We developed a MATLAB implementation package to demonstrate the superior interpolation performance of the proposed methods on synthetic and real seismic data. Compared with the one-step methods, the proposed two-step algorithms enjoy faster convergence. Even though this paper only presents some results from the 2D seismic data, multidimensional implementation of the proposed methods will definitely enhance the interpolated results, and is a subject of further research.

Acknowledgments

This work is supported by National Natural Science Foundation of China (40730424) and National Science & Technology Major Project (2011ZX05023-005). We wish to thank the authors of SeismicLab (http://www.geo.phys.ualberta.ca/saig/SeismicLab) for their publicly available codes, which make the comparison of the seismic interpolation methods of this paper convenient. We would like to thank Ignace Loris for his valuable discussions and enthusiastic encouragement about sparsity constrained inversion. We also thank two anonymous reviewers for their valuable comments, which stimulate us to include the data-driven schedule and to rephrase the exhibition of our methods.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cageo.2012.07.023.

References


