Inverse Problems and Imaging
Volume 15, No. 1, 2021, 159–183

doi:10.3934/ipi.2020076

FAST NON-CONVEX LOW-RANK MATRIX DECOMPOSITION FOR SEPARATION OF POTENTIAL FIELD DATA USING MINIMAL MEMORY

Dan Zhu

Institute of Geophysics & Geomatics
China University of Geosciences (Wuhan)
Wuhan, MO 430074, China

Rosemary A. Renaut*

School of Mathematical and Statistical Sciences
Arizona State University
Tempe, AZ 85287-1804, USA

Hongwei Li

School of Mathematics & Physics
China University of Geosciences (Wuhan)
Wuhan, MO 430074, China

Tianyou Liu

Institute of Geophysics & Geomatics
China University of Geosciences (Wuhan)
Wuhan, MO 430074, China

(Communicated by Ming Yan)

2020 *Mathematics Subject Classification.* Primary: 65F22, 65F55; Secondary: 86A20, 86A22.

Key words and phrases. Gravity and magnetic data, Low-rank method, Potential field separation, Fast algorithm with minimal memory storage, Tongling, Anhui province, China.

The first author is supported by the National Key R & D Program of China 2018YFC1503705; The second author is supported by the NSF grant DMS 1913136; The third author is supported by the National Key R & D Program of China 2018YFC1503705 and Hubei Subsurface Multi-scale Imaging Key Laboratory (China University of Geosciences) SMIL-2018-06;

* Corresponding author: Rosemary A. Renaut.

Abstract. A fast non-convex low-rank matrix decomposition method for potential field data separation is presented. The singular value decomposition of the large size trajectory matrix, which is also a block Hankel matrix, is obtained using a fast randomized singular value decomposition algorithm in which fast block Hankel matrix-vector multiplications are implemented with minimal memory storage. This fast block Hankel matrix randomized singular value decomposition algorithm is integrated into the `Altproj` algorithm, which is a standard non-convex method for solving the robust principal component analysis optimization problem. The integration of this improved estimation for the partial singular value decomposition avoids the construction of the trajectory matrix in the robust principal component analysis optimization problem. Hence, gravity and magnetic data matrices of large size can be computed and potential field data separation is achieved with better computational efficiency. The presented algorithm is also robust and, hence, algorithm-dependent parameters are easily determined. The performance of the algorithm, with and without the efficient estimation of the low rank matrix, is contrasted for the separation of synthetic gravity and magnetic data matrices of different sizes. These results demonstrate that the presented algorithm is not only computationally more efficient but it is also more accurate. Moreover, it is possible to solve far larger problems. As an example, for the adopted computational environment, matrices of sizes larger than 205×205 generate “out of memory” exceptions without the improvement, whereas a matrix of size 2001×2001 can now be calculated in 1062.29s. Finally, the presented algorithm is applied to separate real gravity and magnetic data in the Tongling area, Anhui province, China. Areas which may exhibit mineralizations are inferred based on the separated anomalies.

1. **Introduction.**

To study target geological sources, the target gravity, or magnetic anomalies, that are caused by the target sources should be separated from the total fields which are the superposition of the gravity and magnetic fields caused by all underground sources. The separated anomalies are used for data inversion and interpretation of geological features. Therefore, the separation of potential field data is an important step for high-quality inversion and interpretation. Deep sources generate large scale smooth anomalies which are called regional anomalies. Residual anomalies, which are on a small scale, are caused by shallow sources. There are many methods for separating regional-residual anomalies. They can be classified into three types. The classical methods of the first group separate the data in the spatial domain. These include methods such as the moving average, polynomial fitting, minimum curvature, and empirical mode decomposition, [23, 1, 17, 16]. Methods of the second and third types separate the anomalies in the frequency or wavelet domains, respectively. These include methods such as matched filtering, Wiener filtering, continuation, and discrete wavelet analysis, [6, 19, 21, 20, 7]. While algorithms that separate the anomalies in the frequency or wavelet domains are easy to implement [10, 29], the spectral overlapping of the regional and residual anomalies makes it difficult to obtain satisfactory results [30].

It has been demonstrated in areas of image and signals processing that the use of a low-rank matrix decomposition for robust principal component analysis (`RPCA`) is very effective [5]. The fundamental observation is that practical data from applied science fields is usually distributed on low-dimensional manifolds in high-dimensional spaces [13]. Mathematically, for `RPCA`, it is assumed that the matrix is a linear combination of a matrix which is of low rank and one which is sparse. Because `RPCA` is robust and provides high accuracy separation, it has been applied in many fields, and there is much research on solving the optimization problem. Generally, the Lagrange function is used to transform the double-objective optimization problem into a single-objective optimization problem that is solved using convex optimization. Iterative thresholding, accelerated proximal gradient, exact augmented Lagrange multiplier (EALM), and inexact augmented Lagrange multiplier (IALM) algorithms have been proposed to solve the convex optimization problem [28, 2, 14]. Due to the high computational cost of convex RPCA, a more efficient non-convex RPCA algorithm, named Altproj, was proposed in [18]. The higher accuracy of Altproj has lead to its wide adoption.

A low-rank matrix decomposition algorithm for potential field separation (LRMD_PFS), based on RPCA and singular spectrum analysis, is discussed in [32]. Singular spectrum analysis is a classical method using the trajectory matrix and the singular value decomposition (SVD) [22, 3, 24]. An important step in LRMD_PFS is the construction of the trajectory matrix (which is a block Hankel matrix) of the total field. This is separated into a low-rank matrix and a sparse matrix using convex RPCA. The separated low-rank and sparse matrices are the approximations of the trajectory matrices for the regional and residual anomalies, respectively. The sparse features of the regional anomalies in the frequency domain, and the localization features of the residual anomalies in the spatial domain, are both considered in LRMD_PFS. Although LRMD_PFS separates the anomalies without the use of a Fourier transform to the frequency domain, it can also be seen as providing a new group of methods because it provides a combination of the features of the potential field data in both spatial and frequency domains. Hence, as compared to classical methods, LRMD_PFS is more robust and has higher accuracy. The computational cost of LRMD_PFS is, however, high. There is a large memory demand associated with generating and storing the large scale trajectory matrix, and a large number of operations are required to generate the SVD of a large matrix. For example, if the size of the matrix is 101×101, then the size of the constructed trajectory matrix is 2601×2601. The trajectory matrix then requires memory that is 663 times that of the original data. For a matrix of size 201×201, the size of the trajectory matrix is 10201×10201, and the memory demand increases by a factor of almost 2576. Therefore, the size of the trajectory matrix increases rapidly with the size of the original matrix.

In this paper, a fast block Hankel matrix randomized SVD (FBHMRSVD) algorithm that requires minimal storage is proposed. FBHMRSVD is based on fast block Hankel matrix-vector multiplications (FBHMVM) [27, 15] and the use of a randomized SVD (RSVD) [12, 11, 25]. A fast non-convex low-rank matrix decomposition for potential field separation (FNCLRMD_PFS) based on the FBHMRSVD is obtained. FBHMRSVD is used to achieve the SVD of the trajectory matrix without constructing the large trajectory matrix. The implementation of FBHMRSVD within Altproj yields the approximation of the trajectory matrices for the regional and residual anomalies without explicit construction of the trajectory matrix. Thus, the large scale potential field data matrix can be separated using FNCLRMD_PFS and with lower computational cost and higher accuracy than LRMD_PFS. The algorithm is developed and contrasted with the classical approach for separation of synthetic data sets in Sections 2-4. Results showing that the algorithm efficiently and effectively separates real gravity and magnetic data in the Tongling area, Anhui province, China are presented in Section 5.

2. The fast block Hankel matrix randomized singular value decomposition: FBHMRSVD.

2.1. Fast block Hankel matrix-vector multiplication: FBHMVM. Consider a 2D gridded potential field data matrix $\mathbf{X} = [x_{mn}] \in \mathcal{R}^{P\times Q}$, where x_{mn} denotes the element at the mth row and nth column of the matrix $\mathbf{X}$. Before constructing the trajectory matrix, the Hankel matrix $\mathbf{T}_j$ is constructed using the jth column of $\mathbf{X}$ as follows,

$$\mathbf{T}_j = \begin{bmatrix} x_{1j} & x_{2j} & \cdots & x_{(P-K+1)j} \\ x_{2j} & x_{3j} & \cdots & x_{(P-K+2)j} \\ \vdots & \vdots & & \vdots \\ x_{Kj} & x_{(K+1)j} & \cdots & x_{Pj} \end{bmatrix}.$$

Here, generally, $K = \lfloor (P+1)/2 \rfloor$, where $\lfloor\,\rfloor$ denotes the integer part of its argument. If $\mathbf{T}_j$ has size $K\times L$, where $L = P-K+1$, then trajectory matrix $\mathbf{T}$ of size $K\hat{K}\times L\hat{L}$ is constructed as the block Hankel matrix with $\hat{K}\times\hat{L}$ blocks, where $\hat{L} = (Q-\hat{K}+1)$, and is given by

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 & \cdots & \mathbf{T}_{Q-\hat{K}+1} \\ \mathbf{T}_2 & \mathbf{T}_3 & \cdots & \mathbf{T}_{Q-\hat{K}+2} \\ \vdots & \vdots & & \vdots \\ \mathbf{T}_{\hat{K}} & \mathbf{T}_{\hat{K}+1} & \cdots & \mathbf{T}_Q \end{bmatrix}.$$

Setting $\hat{K} = \lfloor (Q+1)/2 \rfloor$ makes $\mathbf{T}$ as near to square as possible. The construction of $\mathbf{T}$ from $\mathbf{X}$ is denoted by

$$\mathbf{T} = \mathcal{H}(\mathbf{X}). \tag{1}$$

Given a block Hankel matrix the efficient evaluation of matrix-vector products

$$\mathbf{y} = \mathbf{T}\mathbf{b}, \tag{2}$$

is required. Direct evaluation using first (1) to find $\mathbf{T}$ and then calculating (2), uses $8K\hat{K}L\hat{L} - K\hat{K} - L\hat{L}$ flops and requires storage of $K\hat{K}L\hat{L} + K\hat{K} + L\hat{L}$ floating point entries. But, using Algorithm 1, $\mathbf{y}$ can be calculated from $\mathbf{X}$ and $\mathbf{b}$ without constructing $\mathbf{T}$ with $\mathcal{O}(PQ\log_2 PQ)$ flops and a storage requirement of $3PQ+K\hat{K}+L\hat{L}$ entries. The fast operation that combines (1) and (2) is detailed in Algorithm 1, and is denoted by

$$\mathbf{y} = \texttt{FBHMVM}(\mathbf{X}, \mathbf{b}, K, \hat{K}).$$

Algorithm 1 Fast block Hankel matrix-vector multiplication: $\mathbf{y} = \texttt{FBHMVM}(\mathbf{X}, \mathbf{b}, K, \hat{K})$.

```
1: Input: potential field data matrix $\mathbf{X} \in \mathcal{R}^{P\times Q}$; vector $\mathbf{b}$.
2: $\hat{\mathbf{W}} = \texttt{ifft2}(\texttt{fft2}(\mathbf{T}^{circ}).*\texttt{fft2}(\mathbf{W}))$.
3: Output: $\mathbf{y} = \mathbf{J}\,\texttt{vec}(\texttt{extract}(\hat{\mathbf{W}}))$.
```

Algorithm 1 uses the exchange matrix $\mathbf{J}$. This is the permutation matrix which is 0 everywhere except for 1s on the counter diagonal. It is also referred to as the reversal matrix, backward identity, or standard involutory permutation matrix. $\mathbf{T}^{circ}$ is defined by

$$\mathbf{T}^{circ} = \begin{bmatrix} \hat{\mathbf{T}}_{\hat{K}} & \cdots & \hat{\mathbf{T}}_1 & \hat{\mathbf{T}}_{\hat{L}} & \cdots & \hat{\mathbf{T}}_2 \end{bmatrix},$$

where $\hat{\mathbf{T}}_j$ is embedded from the jth column of $\mathbf{X}$ as follows,

$$\hat{\mathbf{T}}_j = \begin{bmatrix} x_{Kj} & \cdots & x_{1j} & x_{Lj} & \cdots & x_{2j} \end{bmatrix}^T.$$

$\mathbf{W}$ is constructed from $\mathbf{b}$ as follows,

$$\mathbf{W} = \begin{bmatrix} \mathbf{B} & \mathbf{0}_{L(\hat{L}-1)} \\ \mathbf{0}_{(L-1)\hat{L}} & \mathbf{0}_{(L-1)(\hat{L}-1)} \end{bmatrix}^T.$$

where $\mathbf{b} = \mathtt{vec}(\mathbf{B})$, and the operation $\mathtt{vec}(\cdot)$ denotes the vectorization operation. Moreover, the extraction operation is defined by

$$\mathtt{extract}(\hat{\mathbf{W}}) = \hat{\mathbf{W}}(1:L, 1:\hat{L}).$$

2.2. Fast block Hankel matrix-matrix multiplication: FBHMMM.

Algorithm 1 applied for block Hankel matrix-matrix multiplication

$$\mathbf{Y} = \mathbf{TC},$$

yields Algorithm 2. Here, $\mathbf{C}$ is $P_C \times Q_C$ and the process of Algorithm 2 is denoted by

$$\mathbf{Y} = \mathtt{FBHMMM}(\mathbf{X}, \mathbf{C}, K, \hat{K}).$$

Algorithm 2 Fast block Hankel matrix-matrix multiplication: $\mathbf{Y} = \mathtt{FBHMMM}(\mathbf{X}, \mathbf{C}, K, \hat{K})$.

```
Input: potential field data matrix $\mathbf{X} \in \mathcal{R}^{P \times Q}$; matrix $\mathbf{C} \in \mathcal{R}^{P_C \times Q_C}$; parameters $K$ and $\hat{K}$.
for $j = 1 : Q_C$. do
    $\mathbf{Y}(:, j) = \mathtt{FBHMVM}(\mathbf{X}, \mathbf{C}(:, j), K, \hat{K})$.
end for
Output: $\mathbf{Y}$.
```

2.3. The fast block Hankel matrix randomized SVD: FBHMRSVD.

The SVD is the basis of matrix rank reduction, and it is an important step in RPCA. We use

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \mathtt{SVD}(\mathcal{H}(\mathbf{X})) = \mathtt{SVD}(\mathbf{T})$$

to denote the SVD $\mathbf{T} = \mathbf{U\Sigma V}^T$. Here $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots]$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots]$ are unitary matrices, $\mathbf{u}_1, \mathbf{u}_2, \cdots$ are left singular vectors, $\mathbf{v}_1, \mathbf{v}_2, \cdots$ are right singular vectors; and $\mathbf{\Sigma} = \mathtt{diag}(\sigma_1, \sigma_2, \cdots)$ is a diagonal matrix where $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant 0$ are the singular values of $\mathbf{T}$.

The cost of obtaining all terms of the SVD is $\mathcal{O}((L\hat{L})(K\hat{K})^2)$, [8], which can be prohibitive when P and Q are large. For the rank reduction problem not all terms are required and it can be sufficient to obtain a partial SVD with r terms,

$$\mathbf{T}_r = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T, \tag{3}$$

where $\mathtt{rank}(\mathbf{T}_r) = r$. Generally, low-rank features of $\mathbf{T}$ are required and so r is relatively small. Still, the cost of finding the exact dominant r terms in (3) is high. On the other hand, the randomized singular value decomposition (RSVD), [12, 11], has been proposed for efficient determination of a low rank matrix approximation $\mathbf{T}_r$ without the exact calculation of the components in (3). We implement the RSVD by taking advantage of all steps employing matrix-matrix multiplications with $\mathbf{T}$

using Algorithm 2, and without explicitly obtaining $\mathbf{T}$. This process, given in Algorithm 3, is denoted by

$$[\mathbf{U}_r, \mathbf{\Sigma}_r, \mathbf{V}_r] = \texttt{FBHMRSVD}(\mathbf{X}, K, \hat{K}, r, p, q).$$

The integer parameters p and q are integral to the implementation of an `RSVD` algorithm. They represent an oversampling and power iteration parameter, respectively. When the required rank r is relatively small with respect to the full rank of the matrix, it is sufficient to take $p = r$. While the accuracy of `RSVD` increases with increasing q, the cost also increases. But if the spectrum separates into a dominant larger set of values with $\sigma_\ell \gg \sigma_{\ell+1}$, it is sufficient to use a relatively small q, such as $q = 0$, 1 or 2, where $q > 0$ applies q steps of a power iteration to improve the approximation to the dominant singular values.

Algorithm 3 Fast block Hankel matrix `RSVD`: $[\mathbf{U}_r, \mathbf{\Sigma}_r, \mathbf{V}_r] = \texttt{FBHMRSVD}(\mathbf{X}, K, \hat{K}, r, p, q)$.

```
1:  Input: potential field data matrix $\mathbf{X} \in \mathcal{R}^{P \times Q}$; desired rank $r$; oversampling
    parameter $p$; power iteration parameter $q$; parameter $K$ and $\hat{K}$.
2:  $\ell = r + p$, $k = 0$.
3:  Generate a Gaussian random matrix $\mathbf{\Omega} \in \mathcal{R}^{\ell \times K\hat{K}}$.
4:  $\mathbf{A}^{(0)} = \texttt{FBHMMM}(\mathbf{X}, \mathbf{\Omega}^T, L, \hat{L})$.
5:  QR factorization: $[\mathbf{Q}^{(0)}, \sim] = \texttt{qr}(\mathbf{A}^{(0)})$, where $\mathbf{Q}^{(0)} \in \mathcal{R}^{L\hat{L} \times \ell}$ is an orthonormal
    matrix.
6:  while $q > k$ do
7:      $\mathbf{A}^{(1)} = \texttt{FBHMMM}(\mathbf{X}, \mathbf{Q}^{(0)}, K, \hat{K})$.
8:      $[\mathbf{Q}^{(1)}, \sim] = \texttt{qr}(\mathbf{A}^{(1)})$.
9:      $\mathbf{A}^{(2)} = \texttt{FBHMMM}(\mathbf{X}, \mathbf{Q}^{(1)}, L, \hat{L})$.
10:     $[\mathbf{Q}^{(2)}, \sim] = \texttt{qr}(\mathbf{A}^{(2)})$.
11:     $\mathbf{Q}^{(0)} = \mathbf{Q}^{(2)}, k = k + 1$.
12: end while
13: $\mathbf{B} = \texttt{FBHMMM}(\mathbf{X}, \mathbf{Q}^{(0)}, K, \hat{K})$
14: Compute the economy SVD of $\mathbf{B}$: $[\tilde{\mathbf{U}}_\ell, \tilde{\mathbf{\Sigma}}_\ell, \tilde{\mathbf{V}}_\ell] = \texttt{svd}(\mathbf{B})$.
15: $\mathbf{V}_r = \tilde{\mathbf{V}}_\ell(:, 1:r)$; $\mathbf{\Sigma}_r = \tilde{\mathbf{\Sigma}}_\ell(1:r, 1:r)$; $\mathbf{U}_r = \mathbf{Q}^{(0)}\tilde{\mathbf{U}}_\ell(:, 1:r)$
16: Output: $\mathbf{U}_r$, $\mathbf{\Sigma}_r$, $\mathbf{V}_r$.
```

In Algorithm 3, steps 4, 7, 9, and 13 involve trajectory matrix-matrix multiplications and are replaced by the use of Algorithm 2 in order to avoid calculation of $\mathbf{T}$. The original equations are

$$\mathbf{A}^{(0)} = \mathbf{T}^T\mathbf{\Omega}^T, \quad \mathbf{A}^{(1)} = \mathbf{T}\mathbf{Q}^{(0)}, \quad \mathbf{A}^{(2)} = \mathbf{T}^T\mathbf{Q}^{(1)} \text{ and } \mathbf{B} = \mathbf{T}\mathbf{Q}^{(0)},$$

where $\mathbf{A}^{(0)}$, $\mathbf{A}^{(2)} \in \mathcal{R}^{L\hat{L} \times \ell}$ and $\mathbf{A}^{(1)}$, $\mathbf{Q}^{(1)}$, $\mathbf{B} \in \mathcal{R}^{K\hat{K} \times \ell}$. Because Algorithm 3 can be recast without using Algorithm 2 for matrix-matrix multiplications, the accuracy of the two algorithms is the same, up to floating point arithmetic errors that may accrue. But the computational cost is much reduced. The computational cost in terms of flops and storage for each algorithm are detailed, for each step, in Table 1. The storage and flops required for steps 5, 8, 10, 14, and 15 are the same for both approaches. But noting that $K\hat{K}L\hat{L} \gg PQ$, the costs are far lower in steps 4, 7, 9, and 13 when implemented using Algorithm 2. When $\ell = 2r$ the dominant computational costs and storage requirements are then $\mathcal{O}(rPQ \log_2 PQ)$

Table 1. Computational cost measured in terms of floating point operations and storage of floating point entries at each step of Algorithm 3 implemented with (`FBHMRSVD`), and without (`RSVD`), the use of `FBHMMM` for multiplications with $\mathbf{T}$. Here $\ell = \mathcal{O}(r)$, and $\ell = 2r$ when $p = r$.

	`FBHMRSVD`		`RSVD`	
Step	Cost in flops	Cost in storage	Cost in flops	Cost in storage
4, 9	$\mathcal{O}(\ell PQ \log_2 PQ)$	$PQ + \ell(K\hat{K} + L\hat{L})$	$2\ell K\hat{K}L\hat{L}$	$K\hat{K}L\hat{L} + \ell(K\hat{K} + L\hat{L})$
5, 10	$2\ell^2(L\hat{L} - \ell/3)$	$2\ell L\hat{L}$	$2\ell^2(L\hat{L} - \ell/3)$	$2\ell L\hat{L}$
7, 13	$\mathcal{O}(\ell PQ \log_2 PQ)$	$PQ + \ell(K\hat{K} + L\hat{L})$	$4\ell K\hat{K}L\hat{L}$	$K\hat{K}L\hat{L} + \ell(K\hat{K} + L\hat{L})$
8	$2\ell^2(K\hat{K} - \ell/3)$	$2\ell K\hat{K}$	$2\ell^2(K\hat{K} - \ell/3)$	$2\ell K\hat{K}$
14	$\mathcal{O}(\ell^3)$	$2\ell^2 + \ell$	$\mathcal{O}(\ell^3)$	$2\ell^2 + \ell$
15	$\ell r(2\ell + 3K\hat{K})$	$r(K\hat{K} + L\hat{L}) + 2\ell^2$ $+\ell L\hat{L} + r + \ell$	$\ell r(2\ell + 3K\hat{K})$	$r(K\hat{K} + L\hat{L}) + 2\ell^2$ $+\ell L\hat{L} + r + \ell$

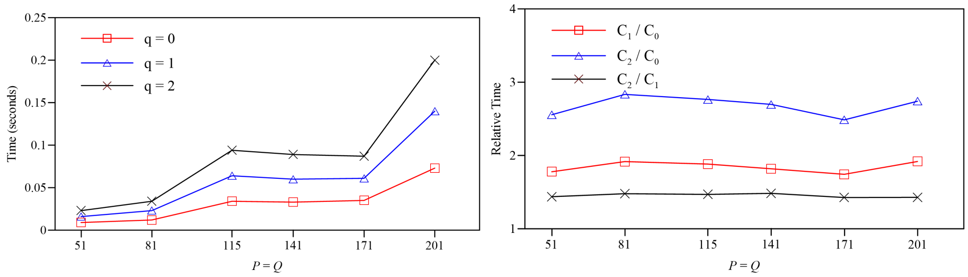
and $\mathcal{O}(rPQ)$, respectively, for Algorithm 3 as compared to terms that on the order of $\mathcal{O}(r(PQ)^2)$ and $\mathcal{O}((PQ)^2)$ without fast multiplication.

Step 14 of Algorithm 3 generates the required `SVD` of $\mathbf{B}$ using the Matlab function `svd`($\mathbf{B}$,′`econ`′). There are multiple other options that may be used, including `svds`($\mathbf{B}, r$) and `eig`($(\mathbf{Y} + \mathbf{Y}^T)/2$), where $\mathbf{Y} = \mathbf{B}^T\mathbf{B}$, as described in [26]. Generating $\mathbf{Y}$, however, introduces extra computational costs that increase the overall cost of obtaining the rank r terms of the `SVD`. For small r, as here, we have found that it is generally more efficient to use Step 14 as given, even if in a few cases the required terms may be achieved more efficiently using `eig`. Should a larger rank r approximation be desired for general signal separation, then using `eig` could be helpful, as in the `RSVD` algorithm considered in [26] for obtaining rank r approximations with $r \gg 10$.

While the `RSVD` has been introduced to improve the efficiency in general for finding a high accuracy partial SVD, [11], there are other options for efficiently calculating a high accuracy partial `SVD` of a block Hankel matrix, e.g. [15]. But, it is not hard to show that the `RSVD` is more efficient even for the block Hankel case. For example, if $\mathbf{X}$ is of size 51×51 and $\mathbf{T}$ of size 676×676, then for a test over 50 runs the average cost as compared to using `RSVD` is 4.65 times higher. This relative performance deteriorates with increasing sizes of $\mathbf{X}$ and $\mathbf{T}$. With the same experiment the relative cost increases to 6.74 when $\mathbf{X}$ and $\mathbf{T}$ increase to 201×201 and 10201×10201, respectively. Thus our interest is the use of the `RSVD` for this large scale application.

2.4. Experiments on `FBHMRSVD`.

We now discuss the influence of the parameters on the accuracy and computational costs of Algorithm 3. We compare the computational costs with, and without, the use of Algorithm 2 for matrix multiplication, and the accuracy as compared to the use of the partial `SVD`. Computations reported using `RSVD` and partial `SVD` are all based on the constructions of the trajectory matrices. These computations are performed on a desktop with 32GB memory and a 4.2GHz Quadcore processor using Matlab 2020a.



(a) Mean computational times.

(b) Mean relative computational times.

Figure 1. Experiments for Algorithm 3 for $r = 10$ with $q = 0$, 1, 2, and increasing $P = Q$. Each experiment is repeated 50 times for each parameter setting. Let C_q be the measured computational cost in terms of clock time measured in seconds of the algorithm in each case for given q. Figure 1(a) is the mean of each C_q over the 50 experiments and Figure 1(b) shows the ratios C_1/C_0, C_2/C_0 and C_2/C_1.

Figure 1 illustrates the impact of the power iteration, parameter, q, on different sizes of the matrix $\mathbf{X}$, for rank $r = 10$ and $q = 0$, 1 and 2. Figure 1(a) shows that the computational cost increases with q. Figure 1(b) demonstrates an approximate doubling in cost when going from $q = 0$ to $q = 1$, and tripling in going from $q = 0$ to $q = 2$. The improvement in reducing the relative error in the rank r approximation of $\mathbf{X}$, as measured by $\|\mathbf{X} - \hat{\mathbf{X}}\|/\sigma_{r+1}$[i], where $\hat{\mathbf{X}}$ is the rank r approximation of $\mathbf{X}$ calculated using Algorithm 3, is shown in Table 2. Clearly there is a trade off between cost and accuracy, thus we recommend $q = 1$ as a suitable compromise.

Table 2. Comparisons of the rank r relative errors using Algorithm 3 to estimate the rank r partial SVD of $\mathbf{X}$, with the mean reported over 50 runs in each case.

Matrix size	Rank r relative error ($r = 10$)		
$\mathbf{X}$	$q = 0$	$q = 1$	$q = 2$
51×51	1.6056	1.0022	1.0000
81×81	1.6977	1.0028	1.0002
115×115	1.4600	1.0150	1.0006
141×141	1.9577	1.0185	1.0006
171×171	1.5904	1.0038	1.0001
201×201	1.4598	1.0063	1.0003

[i] Recall that the minimum rank r error is σ_{r+1} and is achieved by the exact partial SVD with r terms.

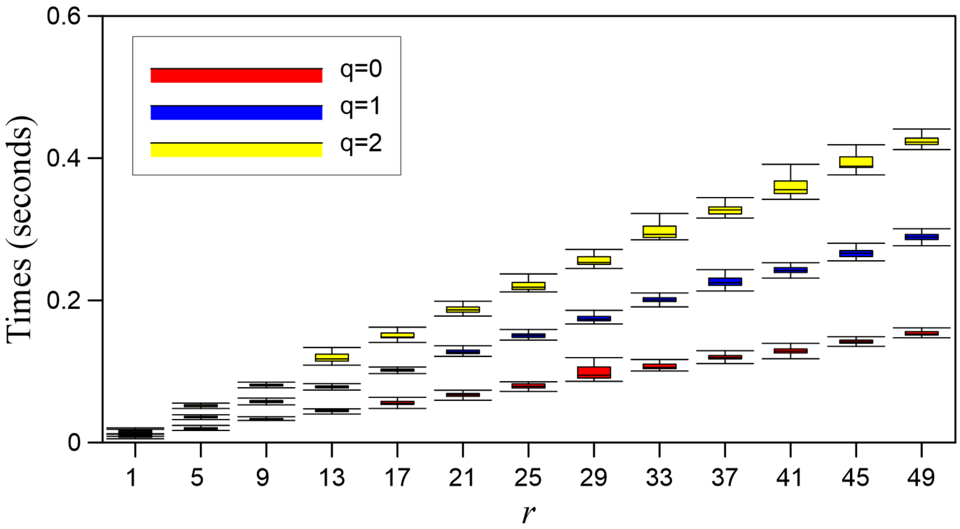


FIGURE 2. Experiments for Algorithm 3 with $q = 0, 1, 2$ and increasing r, $r = 1 : 4 : 49$, and $\mathbf{X}$ of size 141×141. Each experiment is repeated 50 times for each r and q. The box plots for the computational times are reported in seconds.

Figure 2 summarizes the influence of r, increasing from 1 to 49 in increments of 4, on the computational cost of Algorithm 3 with increasing q for an example matrix $\mathbf{X}$ of size of 141×141. The computational cost increases approximately linearly with r for each choice of q with doubling of costs increasing from $q = 0$ to $q = 1$. Because matrix $\mathbf{T}$ is assumed to contain significant low-rank features of the regional anomalies, it is appropriate to assume that r is small.

Table 3 provides a summary of experiments that contrast the computational costs of Algorithm 3, both with and without use of fast matrix-matrix multiplication, `FBHMRSVD`, and `RSVD`, and for direct calculation using the partial `SVD`. `FBHMVM` can also be realized using the `1DFFT` [15], and these experiments are also reported. They demonstrate that using the `1DFFT` generally increases the computational costs. Here in Algorithm 3 we use $r = 10$ and $q = 1$. Each experiment is performed 10 times and the mean result is reported in each case. It is immediate that `FBHMRSVD` is most efficient for all sizes of $\mathbf{X}$, and as the size increases the comparative reduction in cost is significant, already reaching almost a factor 100 for $\mathbf{X}$ of size 401×401.

A monotonic increase in computational cost with increased size of $\mathbf{X}$ is not always observed, as in the results for $\mathbf{X}$ of size 115×115 given in Table 3. The code is implemented in MATLAB and uses the builtin MATLAB functions for the `FFT` and inverse `FFT`. But MATLAB has a mechanism to chose an optimal `FFT` algorithm dependent on the size of the transform that is required. Then a non-monotonic increase in computational cost can occur. We demonstrate this feature of the MATLAB `FFT` implementation in Appendix B.

To investigate the performance for a different release of MATLAB and compute environment, we also ran a test on a laptop with Intel(R) Core (TM) i7-6500U CPU running at 2.50Hz to 2.60Hz with 8.00GB memory, and running MATLAB 2016a.

Table 3. Comparisons of the mean computational times over 10 runs, with rank $r = 10$ and $q = 1$, for Algorithm 3, both with and without use of fast matrix-matrix multiplication, `FBHMRSVD` and `RSVD`, respectively, as compared to direct calculation using the partial `SVD`. Algorithm 3 is evaluated using both the `1DFFT` and `2DFFT` for fast calculation. / denotes that an "out of memory" error is reported.

Matrix sizes		Time (seconds)			
$\mathbf{X}$	$\mathbf{T}$	`FBHMRSVD(2DFFT)`	`FBHMRSVD(1DFFT)`	`RSVD`	`SVD`
81×81	1681×1681	0.023	0.038	0.13	0.24
115×115	3364×3364	0.064	0.079	0.41	0.92
141×141	5041×5041	0.058	0.129	0.78	1.95
171×171	7396×7396	0.059	0.14	1.45	3.93
201×201	10201×10201	0.13	0.31	2.80	7.61
311×311	24336×24336	0.47	0.64	14.09	41.18
401×401	40401×40401	1.46	1.15	111.14	206.79
601×601	90601×90601	1.85	2.17	/	/
1001×1001	251001×251001	2.44	3.58	/	/
2001×2001	1002001×1002001	13.93	26.13	/	/

The `SVD` and `RSVD` calculations without the use of the `FFT` are out of memory at 206×206 and 241×241. Thus, an advantage of our method is that it can be widely implemented on general computers.

3. Fast non-convex low-rank matrix decomposition potential field separation: `FNCLRMD_PFS`.

3.1. Methodology.

We suppose that the total field data matrix of size $P \times Q$ is

$$\mathbf{X} = \mathbf{X}_D + \mathbf{X}_S,$$

where the gridded data matrices of the regional and residual anomalies are denoted by $\mathbf{X}_D$ and $\mathbf{X}_S$, respectively. Practically, $\mathbf{X}_D$ and $\mathbf{X}_S$ are unknown and the objective of potential field data separation is their estimation given $\mathbf{X}$. This means that the block Hankel matrix $\mathbf{T} = \mathcal{H}(\mathbf{X})$ represents $\mathbf{X}_D$ and $\mathbf{X}_S$, separately,

$$\mathbf{T} = \mathbf{T}_D + \mathbf{T}_S = \mathcal{H}(\mathbf{X}_D) + \mathcal{H}(\mathbf{X}_S).$$

Now, it has been shown in [32] that the elements of matrix $\mathbf{X}$ are given by,

$$x_{mn} = \sum_{j=1}^{J} r_j e^{i\phi_j} e^{iu_j m + iv_j n},$$

where (u_j, v_j), r_j, and ϕ_j denote the $2D$ wavenumber, amplitude, and phase of the jth $2D$ component, respectively. It is also proved in [32] that

$$\texttt{rank}(\mathbf{T}) = J,$$

and

$$\sigma_j \approx \sqrt{KL(p-K+1)(q-L+1)} r_j, j = 1, 2, \cdots, J,$$

when $r_1 \geqslant r_2 \geqslant \cdots \geqslant r_J$ and the nonzero singular values of $\mathbf{T}$ are $\sigma_1 \geqslant \sigma_2 \cdots \geqslant \sigma_J > 0$. Moreover, using J_D and J_S to represent the number of nonzero wavenumber components of the regional field data and residual field data, respectively, it is known that

$$J_D \ll J_S.$$

Hence,

$$\texttt{rank}(\mathbf{T}_D) \ll \texttt{rank}(\mathbf{T}_S),$$

and $\mathbf{T}_D$ has a much lower rank than $\mathbf{T}_S$. On the other hand, $\mathbf{T}_S$ is sparse because $\mathbf{X}_S$ is known to be sparse. Hence, the separation of the two signals can be achieved by solving the optimization problem

$$\min \{\texttt{rank}(\mathbf{T}_D), \|\mathbf{T}_S\|_0\} \quad \texttt{subject to}\, \mathbf{T} = \mathbf{T}_D + \mathbf{T}_S. \tag{4}$$

The algorithm `LRMD_PFS` introduced in [32] uses a convex method to solve the optimization problem in (4) by transforming to the optimization problem,

$$\min \|(\mathbf{T}_D)\|_* + \alpha\|\mathbf{T}_S\|_1, \quad \texttt{subject to}\, \mathbf{T} = \mathbf{T}_D + \mathbf{T}_S, \tag{5}$$

where $\|\cdot\|_*$ denotes the nuclear norm which is the sum of the singular values, and $\alpha > 0$ denotes a weighting parameter. Because $\mathbf{T}$ is generally large, the solution of (5) is computationally demanding. The `Altproj` Algorithm [18] to solve (4) is nonconvex and proceeds by alternately updating $\mathbf{T}_S$ by projecting $\mathbf{T} - \mathbf{T}_D$ onto the set of sparse matrices, and $\mathbf{T}_D$ by projecting $\mathbf{T} - \mathbf{T}_S$ onto the set of low-rank matrices. At each step the partial `SVD` of $\mathbf{T}$ is required. Thus, it is ideal to implement the `Altproj` Algorithm using the `FBHMRSVD` Algorithm 3 for all estimates of the partial `SVD`. The solution of (4) with the application of the `Altproj` Algorithm combined with Algorithm 3 is detailed in Algorithm 4, and is denoted by

$$[\mathbf{X}_D^*, \mathbf{X}_S^*] = \texttt{FNCLRMD\_PFS}(\mathbf{X}, K, \hat{K}, r^*, \beta, M, \epsilon).$$

Here r^*, β, M, and ϵ are desired rank, thresholding parameter, an iteration parameter, and a convergence tolerance respectively.

3.2. **Experiments for synthetic geologic models.** Algorithm 4 is applied for the separation of synthetic geologic models for which the total field, regional anomaly, and residual anomaly are shown in Figures 3(a), 3(b), 5(a), and 5(b), respectively. The parameters of these models, for which the matrices are of sizes 201×201, are detailed in Table 4. The impact of the choice of the parameters on the computational cost and the quality of the solutions is investigated.

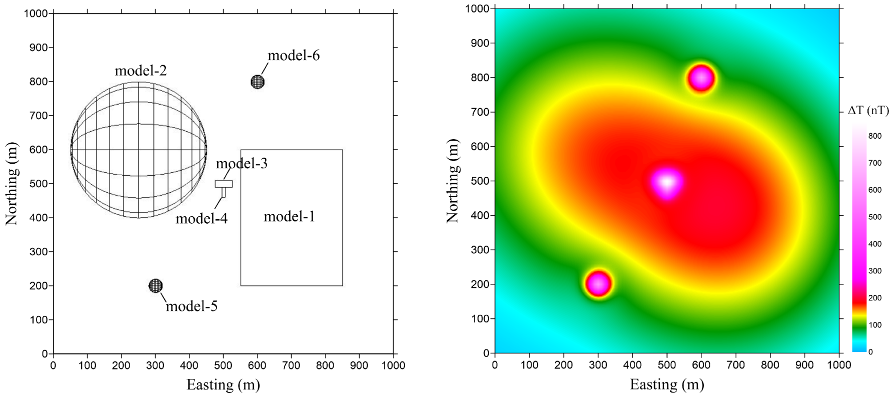
Table 4. The parameters that define the geologic models in Figures 3(a) and 6(a).

Geologic model	Shape	Central position	Model parameters (length, width, depth extent)/radius	Density (g/cm^3)	Magnetization (A/m)
model-1	Block	$(700, 400, 600)$	$(300, 400, 200)$	0.5	8000
model-2	sphere	$(250, 600, 700)$	200	0.4	7000
model-3	Block	$(500, 500, 40)$	$(50, 20, 40)$	0.5	5000
model-4	Block	$(500, 475, 40)$	$(10, 30, 40)$	0.5	5000
model-5	sphere	$(300, 200, 40)$	20		5000
model-6	sphere	$(600, 800, 40)$	20		5000
model-7	sphere	$(200, 200, 40)$	20	0.7	
model-8	Block	$(800, 800, 40)$	$(80, 80, 40)$	0.5	

3.2.1. *Parameter settings.* The quality of the solution of the separated regional and residual anomalies depends on the parameters r^* and β. A default interval for the adjustment of β, $0 < \beta < 1/\sqrt{\texttt{max}(KL, \hat{K}\hat{L})}$ was recommended in [32]. Experiments demonstrated that the results are consistent for a large subinterval. Figure 4(a) shows the `RMSE` as defined by $\|\mathbf{X}^* - \hat{\mathbf{X}}_D\|_F/PQ$ where $\mathbf{X}^*$ is the true regional data. when applying Algorithm 4 with $r^* = 6$ and 10 and different choices for β. The tests are repeated over 50 runs in each case. While the `RMSE`s are relatively insensitive to $\beta \in [0.0003, 0.007]$, it is evident from Figure 4(b), that the computational cost

Algorithm 4 Fast non-convex low-rank matrix decomposition potential field separation:
$[\mathbf{X}_D^*, \mathbf{X}_S^*] = \texttt{FNCLRMD\_PFS}(\mathbf{X}, K, \hat{K}, r^*, \beta, M, \epsilon)$.

```
Input: potential field data matrix $\mathbf{X} \in \mathcal{R}^{P\times Q}$; parameter $K$ and $\hat{K}$; desired rank $r^*$; thresholding parameter $\beta$; iteration parameter $M$; convergence tolerance $\epsilon$.
Definition: $\sigma_j(\mathbf{M})$ denotes the $j$th largest singular value of $\mathbf{M}$; $\texttt{P}_k(\mathbf{H})$ denotes the best rank $k$ approximation of $\mathbf{H}$; $\texttt{HT}_\zeta(\mathbf{H})$ denotes hard-thresholding applied to $\mathbf{H}$ such that entries with absolute values less than $\zeta$ are set to 0; $\texttt{IP}(\mathbf{H})$ denotes inverse projection for recovering the matrix from its trajectory matrix by averaging the counter diagonal of each one of its blocks [9].
$[\mathbf{U}^{(0)}, \mathbf{\Sigma}^{(0)}, \mathbf{V}^{(0)}] = \texttt{FBHMRSVD}(\mathbf{X}, K, \hat{K}, 1, 1, 1)$,
$\zeta_0 = \beta \mathbf{\Sigma}^{(0)}(1,1)$.
$\mathbf{X}_D^{(0)} = 0$; $\mathbf{X}_S^{(0)} = \texttt{HT}_{\zeta_0}(\mathbf{X} - \mathbf{X}_D)$.
for $k = 1$ to $r^*$ do
  for $t = 0$ to $M$ do
    $[\mathbf{U}^{(t+1)}, \mathbf{\Sigma}^{(t+1)}, \mathbf{V}^{(t+1)}] = \texttt{FBHMRSVD}(\mathbf{X} - \mathbf{X}_S^{(t)}, K, \hat{K}, k+1, k+1, 1)$.
    $\zeta = \beta(\mathbf{\Sigma}^{(t+1)}(k+1, k+1) + (\frac{1}{2})^t \mathbf{\Sigma}^{(t+1)}(k,k))$.
    $\mathbf{X}_D^{(t+1)} = \texttt{IP}(\texttt{P}_k(\mathbf{X} - \mathbf{X}_D^{(t)}))$, where $\texttt{P}_k(\mathbf{X} - \mathbf{X}_D^{(t)}) = \mathbf{U}_k^{(t+1)} \mathbf{\Sigma}_k^{(t+1)} (\mathbf{V}_k^{(t+1)})^T$.
    $\mathbf{X}_S^{(t+1)} = \texttt{HT}_\zeta(\mathbf{X} - \mathbf{X}_D^{(t+1)})$.
    if $\|\mathbf{X}_D^{(t+1)} - \mathbf{X}_D^{(t)}\|_F < \epsilon \|\mathbf{X}\|_F$ then
      break
    end if
  end for
end for
$\mathbf{X}_D^* = \mathbf{X}_D^{(t+1)}$, $\mathbf{X}_S^* = \mathbf{X} - \mathbf{X}_D^*$.
Output: $\mathbf{X}_D^*$, $\mathbf{X}_S^*$.
```



(a)

(b)

Figure 3. Figures 3(a) and 3(b) are the geologic models and the forward magnetic field, respectively.

depends dramatically on the choice of r^*. This means the computational time is much more affected by r^* than β, and Algorithm 4 is relatively robust to the choice of β.

Using $\beta \in [0.0003, 0.007]$ as indicated from the previous experiment, the total field is separated for r^* increasing from 1 to 20. The RMSEs of the results are shown in Figure 4(c) and it is immediate that the RMSE decreases rapidly for $r^* = 1 : 4$, but is relatively stable and independent of r^* for $r^* > 4$. But, as seen in Figure 4(d), the computational cost increases with r^*. Thus, there is a trade-off in terms of accuracy and computational cost in choosing r^*. Practically, the regional anomaly should be low rank and it is sufficient to take r^* small, and generally not significantly larger than 10.

The performance of Algorithm 4 depends on the number of inner iterations M and convergence tolerance ϵ. For efficiency M should be small, but for accuracy, and thus with increased cost, ϵ should also be small. Here we have used $M = 10$ and $\epsilon = 0.0001$ throughout. The results demonstrate that FNCLRMD_PFS is much more efficient than LRMD_PFS.

4. Synthetic field data experiments.

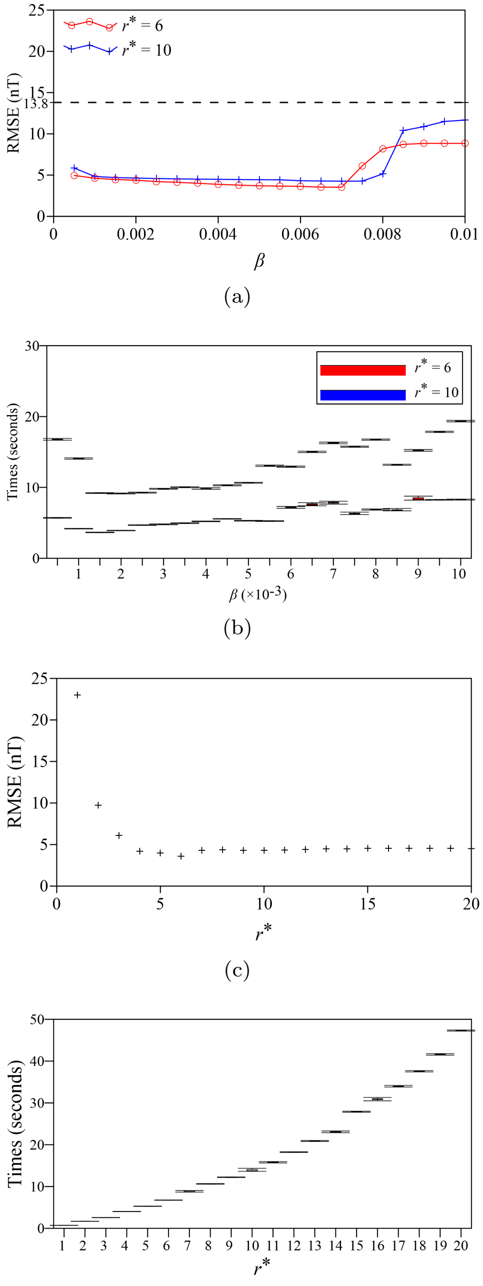
4.1. Experiment 1: Magnetic data.

The accuracy and the computational cost of Algorithm 4, dependent on β, for the gridded data matrices in Figure 3(b), for matrices of different sizes, is contrasted with the results obtained using LRMD_PFS dependent on α. The experiment is repeated for different choices of β and α in the recommended intervals, and the result with the smallest RMSE is selected as the final result and reported in Table 5. Here the results are over 10 runs for each parameter choice. The results are illustrated in Figures 5 for the problem of size 201 × 201 by each algorithm, respectively.

The RMSEs obtained using Algorithm 4 are between 3.52 and 4.19 nT, while the RMSEs for the same experiments using LRMD_PFS are between 13.80 and 16.39 nT, hence demonstrating the higher accuracy of the new algorithm. It is more significant, however, that Algorithm 4 performs better than LRMD_PFS with respect to reducing computational time and memory requirements. In the given computational environment, it is not possible to obtain the data matrices of sizes much greater than 201 × 201 using LRMD_PFS. In contrast, it is possible to solve the problem for matrices of sizes 2001 × 2001 using Algorithm 4. For the smaller problem of size 201 × 201, Figures 5(c) and 5(d) show the separated regional and residual anomalies using Algorithm 4, while Figures 5(e) and 5(f) show the separated anomalies obtained using LRMD_PFS. It can be seen from Figures 5(c) to 5(f), that Algorithm 4 performs well around the boundaries, but that the two methods are comparable in the central areas.

4.2. Experiment 2: Gravity data.

For this experiment, the synthetic geologic models, the total field, the regional anomaly and the residual gravity anomaly are shown in Figures 6(a), 6(b), 7(a), and 7(b), respectively. The parameters that define the models, all for data matrices of size 201 × 201, are detailed in Table 4, and the results are illustrated in Figure 7. In contrast to experiment 1, the residual anomaly is generated for geologic models with different scales. In the application of Algorithm 4 for the separation of the data we set $r^* = 6$ and $\beta = 0.0005$. This yields a RMSE of 0.0028 mGal. In contrast the smallest RMSE using LRMD_PFS is 0.017 mGal and is obtained with $\alpha = 0.0007$. Not only is Algorithm 4 more accurate,



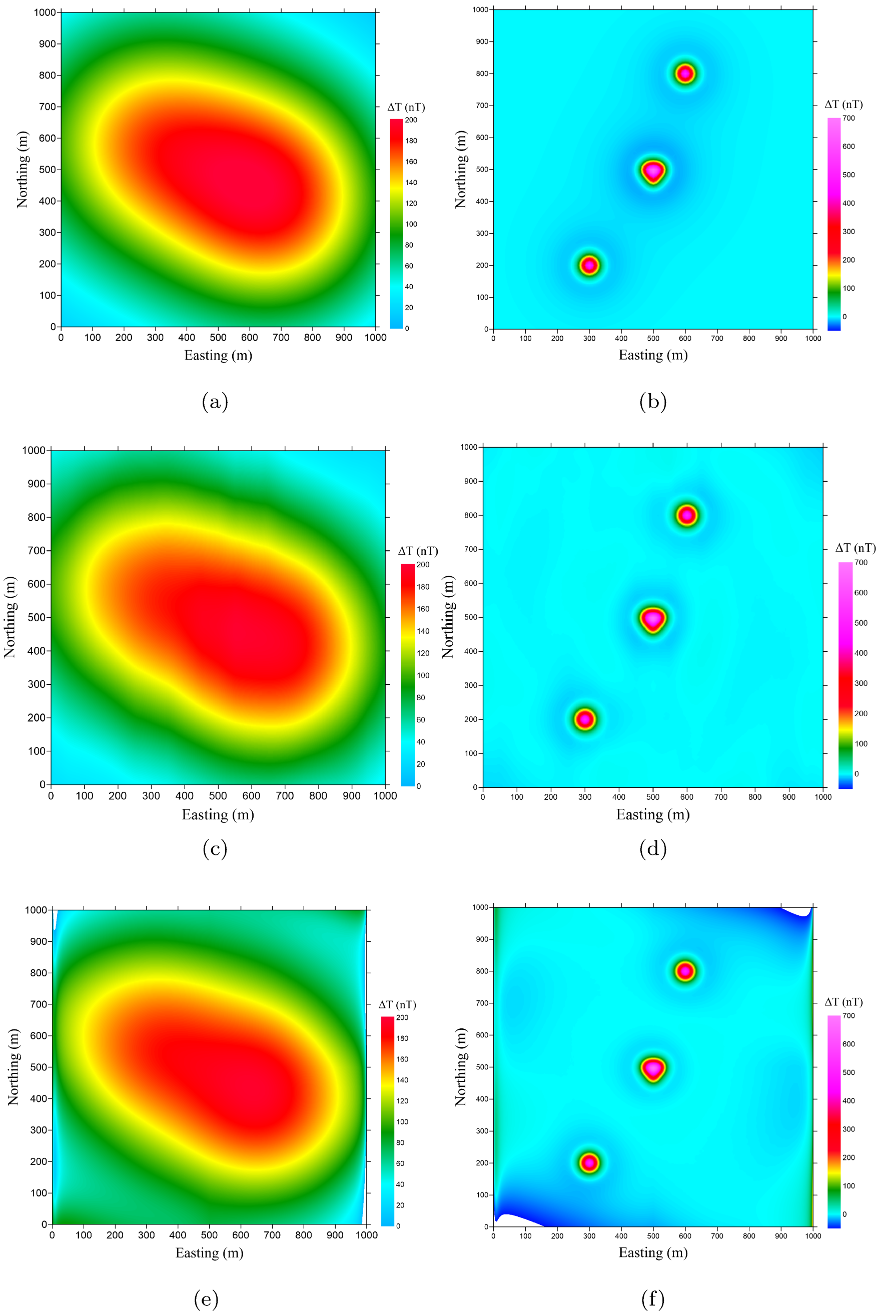
(a)

(b)

(c)

(d)

FIGURE 4. These results show tests of the parameters for Algorithm 4. Figures 4(a) and 4(b) are the RMSE and the computational times for the separations of the data in Figure 3 with different β; Figures 4(c) and 4(d) are the RMSE and the computational times of the separations of the data in Figure 3 with different r^*. These are results over 50 runs in each case.



(a)

(b)

(c)

(d)

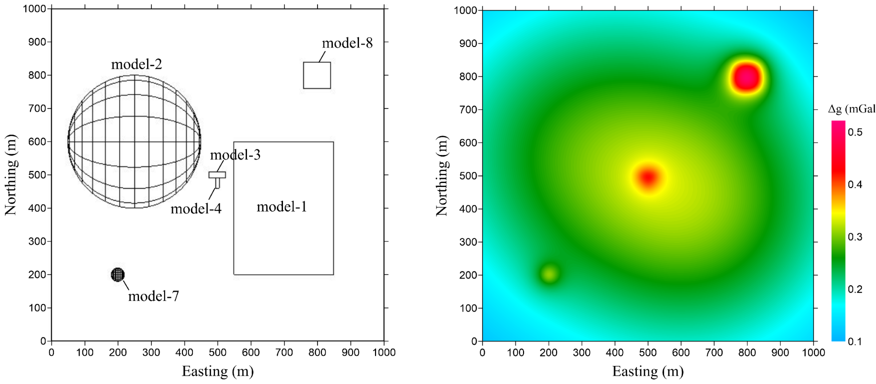
(e)

(f)

Figure 5. Figures 5(a) and 5(b) are the synthetic regional and residual anomalies for the models in Figure 3, respectively; Figures 5(c) and 5(d) are the separated regional and residual anomalies, respectively, for data of size 201 × 201 obtained using Algorithm 4 with $\beta = 0.0062$ and $r^* = 6$; Figures 5(e) and 5(f) are the separated regional and residual anomalies, respectively, obtained using `LRMD_PFS` with $\alpha = 0.0005$.

Table 5. Comparisons of the computational times of the FBHMRSVD, RSVD, and SVD. / denotes that either the computational time is too high to perform the experiment, or an "out of memory" error is reported. Matrices $\mathbf{X}$ and $\mathbf{T}$ are square with sizes $P = Q$ and $\hat{K} = \hat{L}$, respectively. Reported are the mean times and RMSE over 10 runs.

Matrix sizes		FNCLRMD_PFS				LRMD_PFS		
P	$\hat{K}$	r^*	β	RMSE (nT)	Time (s)	α	RMSE (nT)	Time (s)
101	2601	6	0.0156	3.78	4.11	0.002	16.39	17.03
141	5041	6	0.0130	3.52	3.00	0.001	15.10	109.85
171	7396	6	0.0088	3.63	2.87	0.0008	14.46	410.89
201	10201	6	0.0062	3.59	6.75	0.0005	13.80	862.96
241	14641	6	0.0044	3.56	15.06	/	/	/
311	24336	6	0.0026	3.61	27.00	/	/	/
401	40401	6	0.0014	3.65	37.81	/	/	/
601	90601	6	0.0007	3.64	107.43	/	/	/
1001	251001	6	0.0002	3.75	171.93	/	/	/
2001	1002001	6	0.00002	4.19	1305.42	/	/	/



(a)

(b)

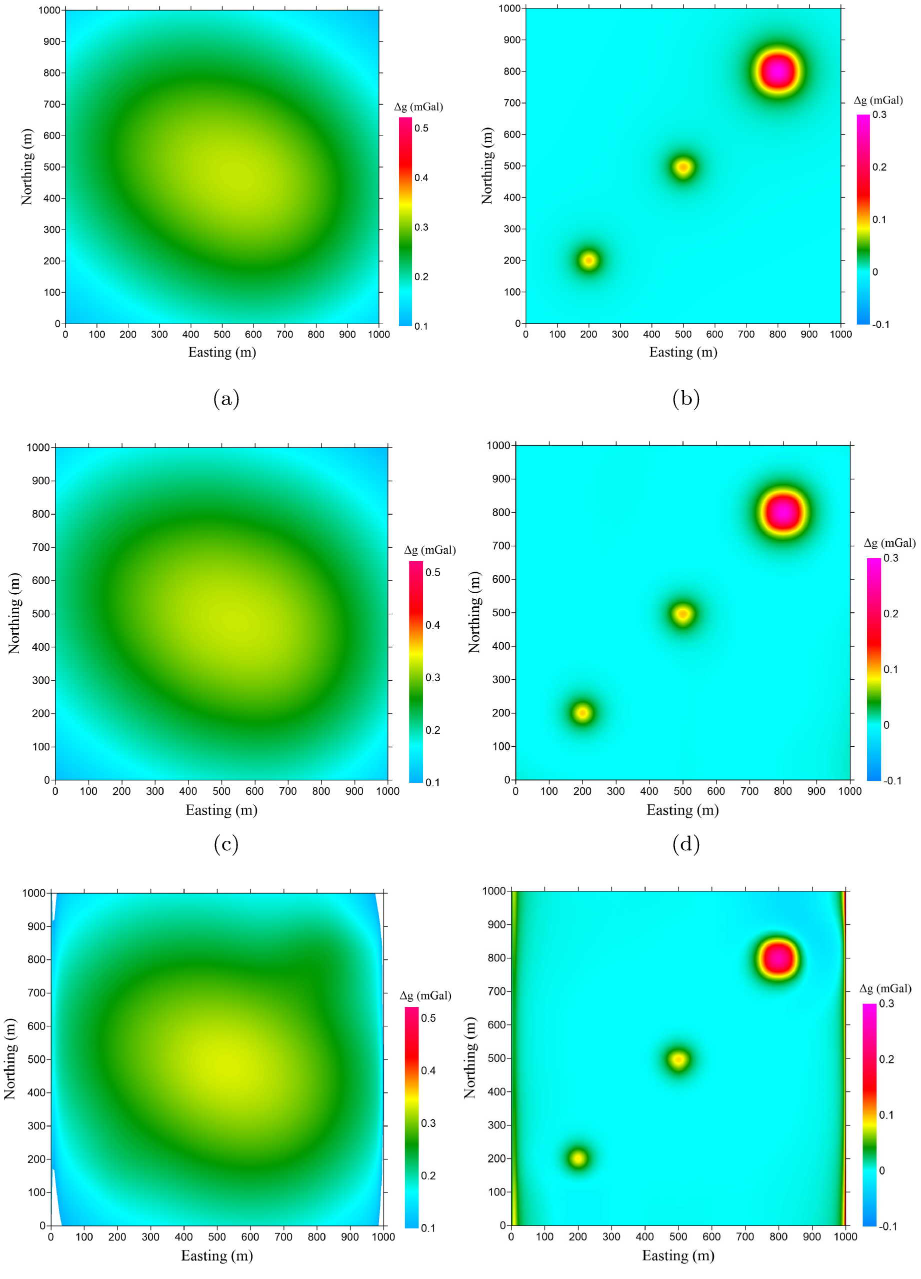
Figure 6. Figures 6(a) and 6(b) are the geologic models and the forward gravity field, respectively.

the computational clock time is reduced to 9.53s from 1277.94s. The results of the separation by the two methods are shown in Figures 7(c) to 7(f). It can be seen from Figures 7(c) to 7(f), that the obtained gravity values of the separated regional anomaly in the north-east region using LRMD_PFS are higher than the synthetic regional anomaly. The results using Algorithm 4 are also qualitatively better.

In summary, our experiments demonstrate that Algorithm 4 has higher accuracy and lower computational cost than the LRMD_PFS for the separation of both magnetic and gravity data.

5. Investigation of FNCLRMD_PFS for a practical data set.

The Tongling region is a good example of skarn deposits lying in Anhui province of China. The



(a) (b)

(c) (d)

(e) (f)

Figure 7. Figures 7(a) and 7(b) are the synthetic regional and residual anomalies for the models in Figure 6, respectively; Figures 7(c) and 7(d) are the separated regional and residual anomalies, respectively, for data of size 201 × 201 obtained using Algorithm 4 with $\beta = 0.0005$ and $r^* = 6$; Figures 7(e) and 7(f) are the separated regional and residual anomalies, respectively, obtained using `LRMD_PFS` with $\alpha = 0.0007$.

Fenghuangshan copper deposit, which is a famous area in the Tongling region, is situated in the east central of the Middle-lower Yangtze metallogenic belt. The mineral deposits are generally of hydrothermal metasomatic type. Thus, the ore bodies occur in the contact zones between igneous rocks and sedimentary rocks. Therefore, in order to predict the location of concealed ore bodies, the separation of anomalies produced by igneous rocks is required.

The study area has three types of rocks. These include sedimentary rocks, igneous rocks, and skarn (or ore body). The physical properties of the sedimentary rocks are medium densities and non-magnetizations, while the igneous rocks are low density (with residual density -0.1 g/cm^3) and medium magnetization (with magnetic susceptibility $0 \sim 3400\times10^{-6}\times 4\pi$ SI). In contrast, the skarn and ore bodies are of high density (with residual density 0.7 g/cm^3) and strong magnetization (with magnetic susceptibility larger than $10000\times10^{-6}\times4\pi$ SI). The difference in the density and magnetic properties of these different rocks makes it effective to study the igneous rocks and ore bodies through gravity and magnetic exploration. Our objective is to separate the combination of regional anomalies of low-gravity and high-magnetism that are produced by igneous rocks, and the combination of local anomalies of high-gravity and high-magnetism produced by skarn and ore bodies. Thus providing a basis for inversion and interpretation.

The algorithm is applied for the separation of the anomalies in Figure 8. Figure 8(a) is a Bouguer gravity anomaly map. The Bouguer gravity anomalies in the study area are high in the north (about 12mGal) and low in the south (about 3mGal). The data matrix has size 247×257. In separating the gravity field we use $r^* = 10$ and $\beta = 0.01$. The separated regional and residual gravity anomalies are shown in Figures 9(a) and 9(b), respectively.

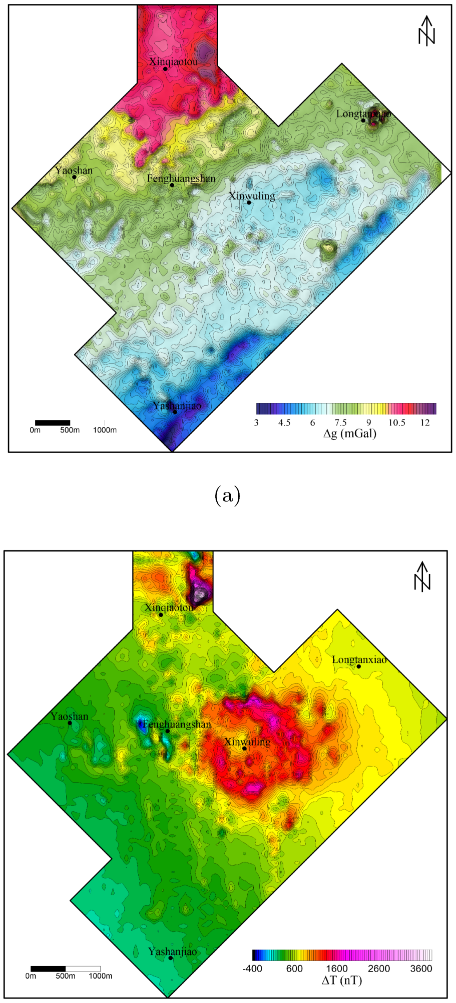
The reduce to the pole (RTP) magnetic anomaly is shown in Figure 8(b). There is a local high magnetic anomaly centered around Xinwuling. The size of the magnetic data matrix is 197×199. In separating the RTP magnetic field we use $r^* = 10$ and $\beta = 0.005$. The separated regional and residual magnetic anomalies are shown in Figures 9(c) and 9(d), respectively.

As we can see in Figure 9, the separated regional gravity anomaly reflects the structures of deep underground sources, and it also reflects the distribution of the igneous rocks in the deep for the corresponding local low gravity anomaly and known Fenghuangshan rocks. Due to the good correspondence of the high magnetic anomaly with Fenghuangshan rocks, the separated regional magnetic anomaly mainly reflects the distribution of the igneous rocks in the deep. The gravity anomaly low and magnetic anomaly highs extend to the north-east of the Fenghuangshan rocks. Thus, the Fenghuangshan rocks in the deep are deduced to extend to the north-east. The areas which correspond to local high gravity and magnetic anomalies in Figures 9(b) and 9(d) are inferred to be skarns or shallow ore bodies which is consistent with known ore and skarn located in these areas. Therefore, we infer the unknown areas which may exhibit mineralizations based on the relations of the gravity and magnetic anomalies in Figures 9(b) and 9(d), as shown in Figure 10.

The computations for the separation of the practical data were also performed on the laptop with Intel(R) Core (TM) i7-6500U CPU running at 2.50Hz to 2.60Hz with 8.00GB memory, and running MATLAB 2016b.

6. Conclusions and future work.

A fast non-convex low-rank matrix decomposition algorithm, FNCLRMD_PFS, for the separation of potential field data has been



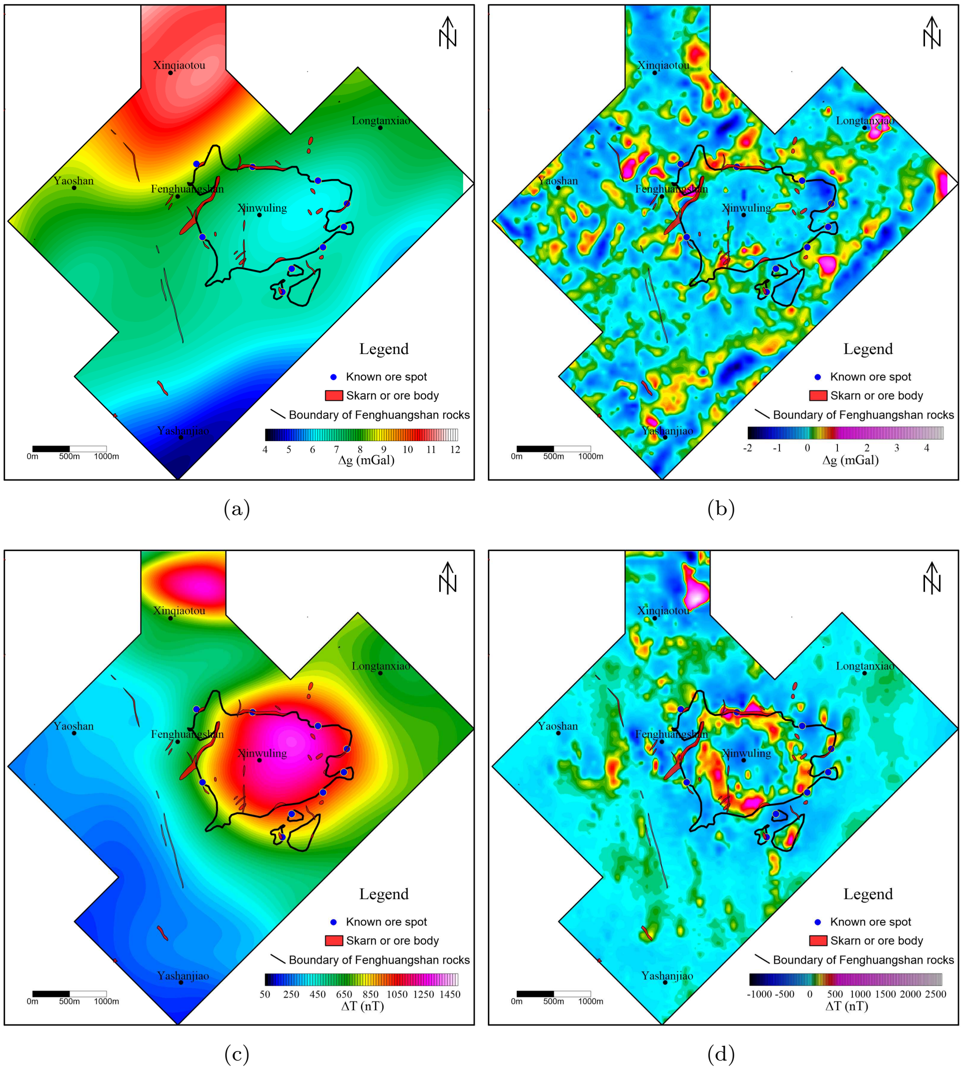
(a)

(b)

Figure 8. Figures 8(a) and 8(b) are the maps of the Bouguer gravity and the RTP magnetic anomalies of the study area in Tongling.

presented and validated. The core of FNCLRMD_PFS is the efficient computation of the partial SVD of the block Hankel trajectory matrix, $\mathbf{T}$, without direct construction of $\mathbf{T}$. Thus, the low-rank matrix decomposition non-convex algorithm for potential field data separation can be realized without requiring the storage or construction of $\mathbf{T}$, and the resulting storage and computational costs are lower than required when using LRMD_PFS. FNCLRMD_PFS depends on two parameters, these are the estimate r^* of the rank of the regional anomaly matrix, and a threshold parameter β.

Synthetic experiments were used to obtain recommendations for the settings of these parameters. These show that a suitable default interval for adjusting β is $0 < \beta < 1/\sqrt{\mathtt{max}(KL, \hat{K}\hat{L})}$. The parameter r^*, when it is not too small, mainly



(a)

(b)

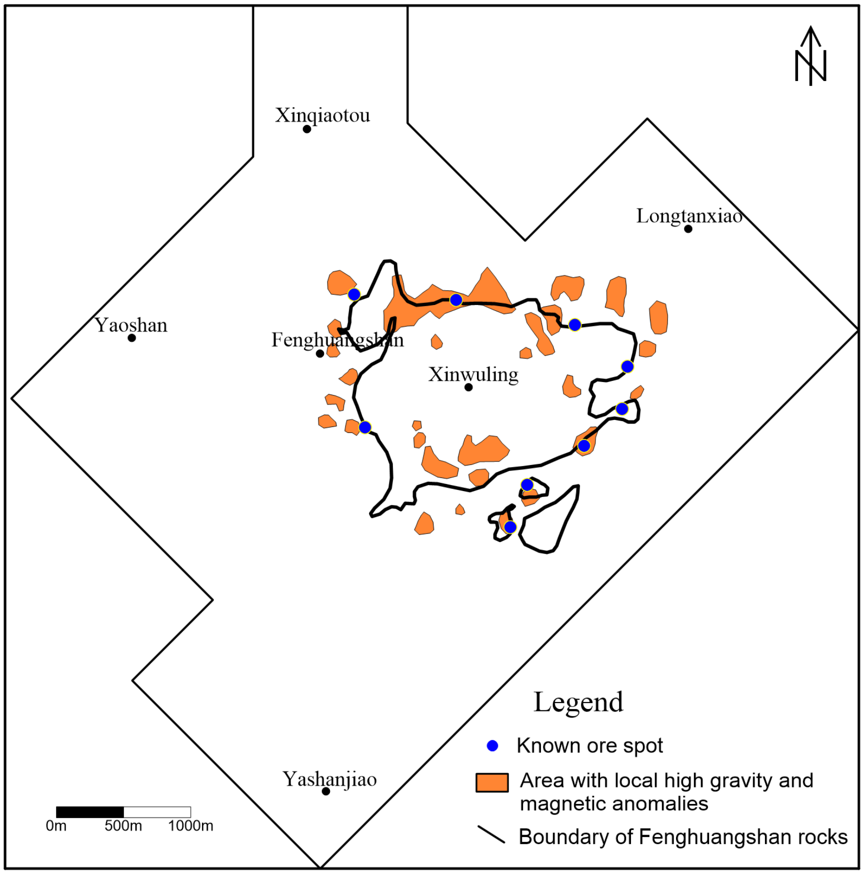
(c)

(d)

FIGURE 9. Figures 9(a) and 9(b) are the separated regional and residual gravity anomalies of the study area, respectively; Figures 9(c) and 9(d) are the separated regional and residual magnetic anomalies of the study area, respectively.

influences the computational time but not the accuracy. The experimental results demonstrate that the presented algorithm is robust and, thus, the choice of parameters, provided the interval for β and rank r^* are chosen as recommended, is straightforward.

Synthetic data sets were set up for gravity and magnetic data and used to contrast the accuracy and computational cost of FNCLRMD_PFS with LRMD_PFS. These results demonstrated that FNCLRMD_PFS has higher accuracy and is more computationally efficient than LRMD_PFS. Moreover, it is feasible to use FNCLRMD_PFS for matrices of much larger size than is possible with LRMD_PFS which exhibits either



(a)

FIGURE 10. Predictions of the distributions of areas that may have sharns or ore bodies based on the separated high-gravity and high-magnetic fields.

with an extreme requirement on computational time or the report of "out of memory" for matrices of large size. Specifically, `FNCLRMD_PFS` can be used to compute large size potential field data with high accuracy at acceptable computational cost. Finally, `FNCLRMD_PFS` was also used for the separation of real data in the Tongling area, Anhui province, China. The separated low-gravity and high-magnetic regional anomalies have good correspondence to the igneous rocks, and the separated high-gravity and high-magnetic residual anomalies exhibit good correspondence to the known ore spots. Consequently, unknown areas of mineralizations can be inferred from the separated anomalies.

We note that an acceleration of the `Altproj` algorithm for signal separation, named `AAP-Hankel` for accelerated alternating projections, is presented in the preprint [4]. Theoretical convergence results are presented for the separation of $1D$ signals and it is stated that the theoretical results extend for problems in signals in higher dimensions. The computational costs and storage requirements are on the same order as FNCLRMD_PFS, with the distinction being the unknown constants in the order terms. Computational experiments demonstrate that AAP-Hankel is faster than algorithm SAP which is an extension of Altproj for missing data, [31]. In the future it will be interesting to contrast the performance of FNCLRMD_PFS and AAP-Hankel for the separation of practical potential field data sets.

REFERENCES

[1] W. B. Agocs, Least squares residual anomaly determination, *Geophysics*, **16** (1951), 686–696.

[2] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences*, **2** (2009), 183–202.

[3] D. S. Broomhead and G. P. King, Extracting qualitative dynamics from experimental data, *Physica D: Nonlinear Phenomena*, **20** (1986), 217–236.

[4] H.Q. Cai, J.-F. Cai, T. Wang, and G. Yin, Accelerated Structured Alternating Projections for Robust Spectrally Sparse Signal Recovery, preprint, http://arxiv.org/abs/1910.05859, (2020).

[5] E. J. Candès, X. D. Li, Y. Ma and J. Wright, Robust principal component analysis?, *Journal of the ACM (JACM)*, **58** (2011), Art. 11, 37 pp.

[6] K. C. Clarke, Optimum second-derivative and downward-continuation filters, *Geophysics*, **34** (1969), 424–437.

[7] M. Fedi and T. Quarta, Wavelet analysis for the regional-residual and local separation of potential field anomalies, *Geophysical Prospecting*, **46** (1998), 507–525.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Press, Baltimore, 1996.

[9] N. Golyandina, I. Florinsky, and K. Usevich, Filtering of digital terrain models by 2D singular spectrum analysis, *International Journal of Ecology & Development*, **8** (2007), 81–94.

[10] Z. Z. Hou and W. C. Yang, Wavelet transform and multi-scale analysis on gravity anomalies of China, *Chinese Journal of Geophysics*, **40** (1997), 85–95.

[11] N. Halko, P. Martinsson and J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Review*, **53** (2011), 217–288.

[12] E. Liberty, F. Woolfe, P. Martinsson, V. Rokhlin and M. Tygert, Randomized algorithms for the low-rank approximation of matrices, *Proceedings of the National Academy of Sciences*, **104** (2007), 20167–20172.

[13] Z. C. Lin and H. Y. Zhang, *Low-rank Models in Visual Analysis,* Elsevier Science Publishing Co Inc, New York, 2017.

[14] Z. C. Lin, M. M. Chen, and Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, preprint, arXiv:1009.5055, (2013).

[15] L. Lu, W. Xu and S. Z. Qiao, A fast SVD for multilevel block Hankel matrices with minimal memory storage, *Numerical Algorithms*, **69** (2015), 875–891.

[16] A. Mandal, and S. Niyogi Filter assisted bi-dimensional empirical mode decomposition: a hybrid approach for regional-residual separation of gravity anomaly, *Journal of Applied Geophysics*, **159** (2018), 218–227.

[17] K. L. Mickus, C. L. V. Aiken and W. D. Kennedy Regional-residual gravity anomaly separation using the minimum-curvature technique, *Geophysics*, **56** (1991), 279–283.

[18] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, Non-convex robust PCA, *Advances in Neural Information Processing Systems*, (2014), 1107–1115.

[19] R. S. Pawlowski Preferential continuation for potential-field anomaly enhancement, *Geophysics*, **60** (1995), 390–398.

[20] R. S. Pawlowski and R. O. Hansen Gravity anomaly separation by Wiener filtering, *Geophysics*, **55** (1990), 539–548.

[21] A. Spector and F. S. Grant Statistical models for interpreting aeromagnetic data, *Geophysics*, **35** (1970), 293–302.

[22] F. Takens, Detecting strange attractors in turbulence, *Dynamical Systems and Turbulence, Warwick 1980*, (1981), 366–381.

[23] W. M. Telford, L. P. Geldart and R. E. Sheriff, *Applied Geophysics,* Cambridge University Press, Cambridge, 2003.

[24] A. A. Tsonis and J. B. Elsner, Mapping the channels of communication between the tropics and higher latitudes in the atmosphere, *Physica D: Nonlinear Phenomena*, **92** (1996), 237–244.

[25] S. Vatankhah, R. A. Renaut and V. E. Ardestani, A fast algorithm for regularized focused 3D inversion of gravity data using randomized singular-value decomposition, *Geophysics*, **83** (2018), G25–G34.

[26] S. Vatankhah, S. Liu, R.A. Renaut, X. Hu and J. Baniamerian, Improving the use of the randomized singular value decomposition for the inversion of gravity and magnetic data, *Geophysics*, **85** (2020), G93–G107.

[27] C. R. Vogel, *Computational Methods for Inverse Problems*, Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[28] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang and S. C. Yan, Sparse representation for computer vision and pattern recognition, *Proceedings of the IEEE*, **98** (2010), 1031–1044.

[29] W. C. Yang, Z. Q. Shi, Z. Z. Hou and Z. Y. Cheng Discrete wavelet transform for multiple decomposition of gravity anomalies, *Chinese Journal of Geophysics*, **44** (2001), 529–537.

[30] L. L. Zhang, T. Y. Hao and W. W. Jiang Separation of potential field data using 3-D principal component analysis and textural analysis, *Geophysical Journal International*, **179** (2009), 1397–1413.

[31] S. Zhang and M. Wang, Correction of simultaneous bad measurements by exploiting the low-rank Hankel structure, *2018 IEEE International Symposium on Information Theory* (ISIT), (2018), 646–650.

[32] D. Zhu, H. W. Li, T. Y. Liu, L. H. Fu and S. H. Zhang, Low-rank matrix decomposition method for potential field data separation, *Geophysics*, **85** (2020), G1–G16.

Appendix A. Notation. Acronyms and notation used throughout are provided in Tables 6 and 7.

Table 6. Acronyms used throughout

Acronym	Description
FBHMRSVD	fast block Hankel matrix randomized SVD algorithm
FBHMMM	fast block Hankel matrix-matrix multiplication algorithm
FBHMVM	fast block Hankel matrix-vector multiplication Algorithm
FNCLRMD_PFS	fast non-convex low-rank matrix decomposition algorithm for potential field separation
EALM	exact augmented Lagrange multiplier method
IALM	inexact augmented Lagrange multiplier method
LRMD_PFS	low-rank matrix decomposition for potential field separation
RPCA	robust principal component analysis
RSVD	randomized singular value decomposition
SVD	singular value decomposition
RMSE	root mean square error
RTP	reduce to the pole

Appendix B. The impact of the choice of the FFT used by Matlab on the computational cost. Our initial investigation of the computational cost of Algorithm 1 demonstrated a general tendency for the computational cost to increase monotonically with increasing size of the matrices. There were, however, outlier sizes which were significantly higher in cost and departed from the general monotonic increase in time. This is illustrated in Figure 11 for which we conducted an experiment to test the cost of step 1 in Algorithm 1 using the $\texttt{vec}(\mathbf{X})$ with its dimensions between 2^{15} and 80000. For each matrix dimension, the code is run 80 times, and the average time is calculated. But, because the MATLAB function determines an optimal transform to use for a given matrix size, at greater cost in the first run, this first run is excluded from the estimate of the average cost for each matrix size. A spike in cost is seen between 60000 and 70000, actually at 63001,

Table 7. Notation used throughout

Notation	Description
$\mathbf{J}$	exchange matrix
$\mathbf{X}$	$2D$ gridded potential field data matrix
$\mathbf{T}_j$	Hankel matrix constructed from the jth column of $\mathbf{X}$
$\mathbf{T}$	trajectory matrix of $\mathbf{X}$
$\mathbf{X}_1, \cdots, \mathbf{X}_Q$	first to Qth columns of $\mathbf{X}$, respectively
$\mathbf{U}, \mathbf{V}, \mathbf{\Sigma}$	`SVD` of $\mathbf{T}$, $\mathbf{T} = \mathbf{U}\mathbf{\Sigma}V^T$
$\mathbf{U}_r, \mathbf{V}_r, \mathbf{\Sigma}_r$	rank-r partial `SVD` of $\mathbf{T}$ using `FBHMRSVD`
$\mathbf{X}_D$, $\mathbf{X}_S$	data matrices of regional and residual anomalies, respectively
$\mathbf{T}_D$, $\mathbf{T}_S$	trajectory matrices of $\mathbf{X}_D$ and $\mathbf{X}_S$, respectively
$\mathbf{X}_D^*$, $\mathbf{X}_S^*$	approximations of $\mathbf{X}_D$ and $\mathbf{X}_S$ using `FNCLRMD_PFS`, respectively
$\mathbf{u}_1, \mathbf{u}_2, \cdots$	$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots]$, $\mathbf{u}_1, \mathbf{u}_2, \cdots$ are the left singular vectors of $\mathbf{T}$
$\mathbf{v}_1, \mathbf{v}_2, \cdots$	$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots]$, $\mathbf{v}_1, \mathbf{v}_2, \cdots$ are the right singular vectors of $\mathbf{T}$
x_{mn}	element at mth row and nth column of $\mathbf{X}$
P, Q	$\mathbf{X}$ is of size $P \times Q$
K, L	$\mathbf{T}_j$ is of size $K \times L$
$\hat{K}$, $\hat{L}$	$\mathbf{T}$ is a block Hankel matrix with $\hat{K} \times \hat{L}$ blocks
P_C, Q_C	$\mathbf{C}$ is of size $P_C \times Q_C$
σ_1, σ_2, $\cdots$	$\mathbf{\Sigma} = \mathtt{diag}(\sigma_1, \sigma_2, \cdots)$, where σ_1, σ_2, $\cdots$ are the singular values of $\mathbf{T}$
r	desired rank parameter in `FBHMRSVD`
p	oversampling parameter in `FBHMRSVD`
q	power iteration parameter in `FBHMRSVD`
r^*	desired rank parameter in `FNCLRMD_PFS`
β	thresholding parameter in `FNCLRMD_PFS`
α	weighting parameter in `LRMD_PFS`
$\|\cdot\|_p$, $\|\cdot\|_*$	ℓ_p and nuclear norms, respectively

but overall the tendency is a gradual increase in computational cost and outliers are not frequent. We note that $63001 = 251 \times 251$ is not prime but 251 is prime, and the determination of an optimal transform depends on the factorization of the transform size. We conclude that there may be cases where the computational cost of Algorithm 1 spikes because of this situation. On the other hand, for the problem of this size the calculation of the `RSVD` with, and without, the use of Algorithm 2 for the matrix multiplications has a computational cost in each case of 2.614s and 30.885s, respectively. Hence, even when the `FFT` transform is relatively slow, the use of a fast block Hankel matrix multiplication is still faster than the use of a direct matrix-multiplication without the use of the `FFT`.

Acknowledgments. The authors are appreciative of the reviewers' comments that assisted with improving the paper. Rosemary Renaut acknowledges the support of NSF grant DMS 1913136: "Approximate Singular Value Expansions and Solutions of Ill-Posed Problems". Dan Zhu and Hongwei Li acknowledge the support of the National Key R&D Program of China (2018YFC1503705). Hongwei Li acknowledges the support of Hubei Subsurface Multi-scale Imaging Key Laboratory (China University of Geosciences) (SMIL-2018-06). We also acknowledge Anhui Geology and Mineral Exploration Bureau 321 Geological Team for providing the data sets that were used for the real data experiments.

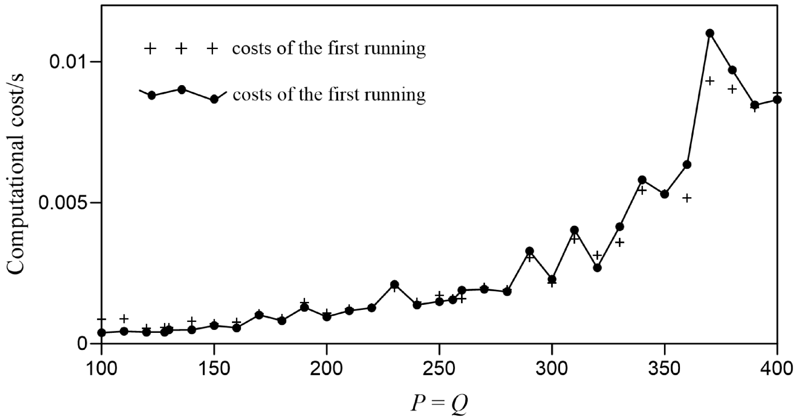


FIGURE 11. Demonstrating non-monotonic increase in computational time using Algorithm 1 for $P \times Q$ between 2^{15} and 80000.

Received March 2020; revised August 2020.

E-mail address: zhud_igg@cug.edu.cn
E-mail address: renaut@asu.edu
E-mail address: hwli@cug.edu.cn
E-mail address: liuty@cug.edu.cn