

A fast methodology for large-scale focusing inversion of gravity and magnetic data using the structured model matrix and the 2-D fast Fourier transform

Rosemary A. Renaut¹, Jarom D. Hogue¹, Saeed Vatanikhah^{2,3} and Shuang Liu³

¹*School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85281, USA. E-mail: renaut@asu.edu*

²*Institute of Geophysics, University of Tehran, Tehran, Iran*

³*Hubei Subsurface Multi-scale Imaging Key Laboratory, Institute of Geophysics and Geomatics, China University of Geosciences, Wuhan, China*

Accepted 2020 August 4. Received 2020 August 4; in original form 2020 April 30

SUMMARY

We discuss the focusing inversion of potential field data for the recovery of sparse subsurface structures from surface measurement data on a uniform grid. For the uniform grid, the model sensitivity matrices have a block Toeplitz Toeplitz block structure for each block of columns related to a fixed depth layer of the subsurface. Then, all forward operations with the sensitivity matrix, or its transpose, are performed using the 2-D fast Fourier transform. Simulations are provided to show that the implementation of the focusing inversion algorithm using the fast Fourier transform is efficient, and that the algorithm can be realized on standard desktop computers with sufficient memory for storage of volumes up to size $n \approx 10^6$. The linear systems of equations arising in the focusing inversion algorithm are solved using either Golub–Kahan bidiagonalization or randomized singular value decomposition algorithms. These two algorithms are contrasted for their efficiency when used to solve large-scale problems with respect to the sizes of the projected subspaces adopted for the solutions of the linear systems. The results confirm earlier studies that the randomized algorithms are to be preferred for the inversion of gravity data, and for data sets of size m it is sufficient to use projected spaces of size approximately $m/8$. For the inversion of magnetic data sets, we show that it is more efficient to use the Golub–Kahan bidiagonalization, and that it is again sufficient to use projected spaces of size approximately $m/8$. Simulations support the presented conclusions and are verified for the inversion of a magnetic data set obtained over the Wuskwatim Lake region in Manitoba, Canada.

Key words: Gravity anomalies and Earth structure; Inverse theory; Numerical approximations and analysis.

1 INTRODUCTION

The determination of subsurface structures from the inversion of measured potential field data is important for many practical applications concerned with oil and gas exploration, mining and regional investigations (Blakely 1995; Nabighian *et al.* 2005). For gravity and magnetic potential field data, there are many techniques for the reconstruction of the subsurface structures from the data. These include the direct inversion of a forward model described by a sensitivity matrix for gravity and magnetic potential field data, (Li & Oldenburg 1996; Pilkington 1997; Li & Oldenburg 1998; Portniaguine & Zhdanov 1999; Boulanger & Chouteau 2001; Silva & Barbosa 2006; Lelièvre & Oldenburg 2006; Farquharson 2008). When the problem is large-scale, it is important to consider alternative algorithms that avoid the generation and storage of the sensitivity matrix (Cox *et al.* 2010; Uieda & Barbosa 2012; Vatanikhah

et al. 2019). Other directions include application of wavelet and compression techniques (Portniaguine & Zhdanov 2002; Li & Oldenburg 2003; Voronin *et al.* 2015), or the use of the structure of the sensitivity matrix for the design of efficient algorithms based on the 2-D fast Fourier transform (2DFFT, Pilkington 1997; Bruun & Nielsen 2007; Zhang & Wong 2015; Chen & Liu 2018).

While the use of the fast Fourier transform (FFT) has been applied in a number of contexts in relation to forward modeling for geophysics kernels (Li *et al.* 2018; Zhao *et al.* 2018; Hogue *et al.* 2019), it appears that Bruun & Nielsen (2007) provided the first discussion of the use of the structure of the sensitivity matrix in relation to the use of the 2DFFT for the inversion of 2-D potential field data. They observed that the sensitivity matrix exhibits a block Toeplitz Toeplitz block (BTTB) structure provided that the data are measured on uniform grid. It is the BTTB structure that facilitates the use of the 2DFFT, and the associated reduction in memory requirements, via

an embedding in a block Circulant Circulant block (BCCB) matrix (Li & Chouteau 1998; Vogel 2002; Chan & Jin 2007).

It is well-known, that an inversion algorithm for the reconstruction of subsurface structures will not provide suitable compact and reliable estimates of the structures using a smoothing regularization. Instead, state-of-the-art approaches for resolving complex structures require stabilization with a general L_p norm regularizer ($0 \leq p < 2$) (Last & Kubik 1983; Portniaguine & Zhdanov 1999, 2002; Vatanikhah *et al.* 2020b). Although the BTTB structure has been applied in inversion algorithms for gravity data using standard Tikhonov smoothers (Bruun & Nielsen 2007; Zhang & Wong 2015), it has not been applied for focusing inversion. Moreover, while the singular value decomposition (SVD) is useful for the solution of linear systems for small scale problems (Vatanikhah *et al.* 2014, 2015), focusing algorithms for large-scale problems rely on alternatives to the SVD for improved efficiency. On the other hand, a reasonable approximation of the dominant terms of the SVD can be useful in enabling automatic determination of regularization parameters (Renaut *et al.* 2017). Two algorithms that are both more efficient than the SVD and provide useful approximations to the dominant terms of the SVD, are the randomized singular value decomposition (RSVD) (Halko *et al.* 2011) and the iterative Krylov method based on the Golub-Kahan bidiagonalization (GKB) algorithm (Paige & Saunders 1982; Renaut *et al.* 2017). Both of these algorithms provide an approximation to the SVD dependent on the number of terms required to provide a good approximation of the dominant space, which is model and size dependent (Vatanikhah *et al.* 2017, 2018, 2020a). Recommendations for the application of the RSVD with power iteration, and the sizes of the projected spaces to be used for both GKB and RSVD algorithms, are available for problems of moderate scale that can be solved without the use of the 2DFFT (Vatanikhah *et al.* 2017, 2018, 2020a; Luiken & van Leeuwen 2020). These recommendations need to be extended for the solution of large-scale problems.

The consideration of memory demand, computing efficiency and effective determination of rank for the focusing inversion of potential field data is the focus of this paper. Of interest, is the development of an approach that allows domain padding and takes advantage of the structure of the sensitivity matrix, for both gravity and magnetic problems, to enable the use of the 2DFFT for matrix operations with the sensitivity matrix. The use of the structure also significantly reduces the storage requirements by eliminating the need to store the sensitivity matrix. An efficient implementation provides the opportunity to solve large-scale problems and make recommendations for the sizes (ranks) of the projected spaces for the solutions. Thus, a complete validation and study of the RSVD and GKB algorithms for the solution of the large-scale focusing inversion problem is important.

1.1 Overview of main scientific contributions

In this paper, we examine the impact of the application of the 2DFFT, and the sizes of projected spaces for solutions, in focusing inversion algorithms for the large-scale inversion of gravity and magnetic potential field data sets. There are three main contributions of this paper that expand on related work in the literature. First, we analyse the efficiency, in terms of memory and computational cost, of the use of the 2DFFT for all operations with the sensitivity matrix, or its transpose, within the iteratively regularized least squares algorithm for focusing inversion, where the systems of equations are solved using GKB or RSVD algorithms. Secondly, we examine and contrast

the sizes of the projected spaces that are required by the GKB or RSVD algorithms for gravity and magnetic data sets. Finally, we conclude that the GKB algorithm is to be preferred for the solution of the magnetic inversion problem, but that the RSVD algorithm is suggested for the inversion of the gravity data sets, and that the recommended size of the projected space is approximately $m/8$ for data sets with m measurements. We demonstrate that the methodology is suitable for focusing inversion of large-scale data sets and can provide volumetric parameter reconstructions with more than 1 million variables using a laptop computer.

The paper is organized as follows. In Section 2, we present a general methodology for the inversion of gravity and magnetic potential field data, including the derivation of the forward model with uniform placement of measurement stations (Section 2.1), overviews of the focusing inversion methodology (Section 2.2), numerical solvers (Section 2.3), algorithms (Section 2.4), and computational costs (Section 2.5). Numerical results applying the presented algorithms to synthetic and practical data are described in Section 3. We discuss the parameter choices for the algorithms that apply to all computational implementations in Section 3.1, and the generation of the simulation data in Section 3.2. The computational costs for one iteration of the iterative algorithm are contrasted for implementations with, and without, the 2DFFT (Section 3.3.1). The comparative costs of using either the GKB, or the RSVD, solver within the iterative focusing algorithm are examined in Section 3.3.2. We provide validating results for the inversion of magnetic data obtained over a portion of the Wuskwatim Lake region in Manitoba, Canada in Section 3.4 and conclusions in Section 4. Appendix A gives brief details on the use of the 2DFFT for matrices with BTTB structure. The supporting numerical evidence for the figures illustrating the results is provided in a number of tables in Appendix B.

2 METHODOLOGY

2.1 Forward model and BTTB structure

We consider the inversion of measured potential field data $\mathbf{d}_{\text{obs}}$ that describes the response at the surface due to unknown subsurface model parameters $\mathbf{m}$. The data and model parameters are connected via the forward model

$$\mathbf{d}_{\text{obs}} = \mathbf{G}\mathbf{m}, \quad (1)$$

where $\mathbf{G}$ is the sensitivity, or model, matrix. This linear relationship is obtained via the discretization of a Fredholm integral equation of the first kind,

$$d(a, b, c) = \iiint h(a, b, c, x, y, z) \zeta(x, y, z) dx dy dz, \quad (2)$$

where exact values $\mathbf{d}$ and $\mathbf{m}$ are the discretizations of continuous functions d and ζ , respectively, and $\mathbf{G}$ in (1) provides the discrete approximation of the integrals of the kernel function h over the volume cells. For the specific kernels associated with gravity and magnetic data, assuming for magnetic data that there is no remanent magnetization or self-demagnetization, h is spatially invariant in all dimensions, $h(a, b, c, x, y, z) = h(x - a, y - b, z - c)$ and (2) describes a convolution operation.

Using the formulation of the integral of the kernel as derived by Haáz (1953) and Li & Chouteau (1998) for the gravity kernel, and by Rao & Babu (1991) for the magnetic kernel, sensitivity matrix $\mathbf{G}$ decomposes by column blocks as

$$\mathbf{G} = [\mathbf{G}^{(1)}, \dots, \mathbf{G}^{(n_z)}], \quad (3)$$

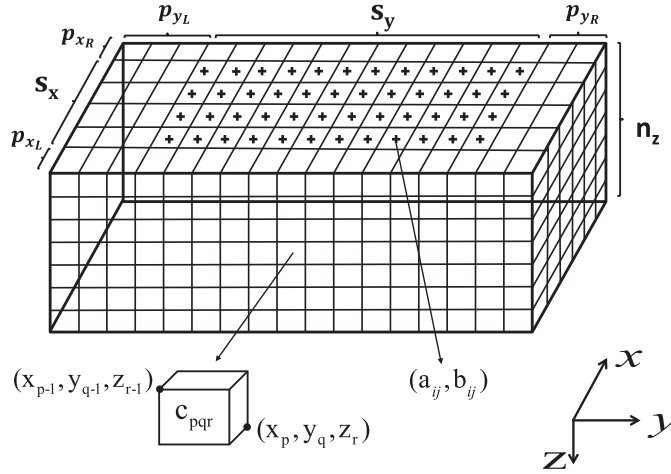


Figure 1. The configuration of prism c_{pqr} , $1 \leq p \leq s_x + p_{xL} + p_{xR} = n_x$, $1 \leq q \leq s_y + p_{yL} + p_{yR} = n_y$, $1 \leq r \leq n_z$, in the volume relative to a station on the surface at location $s_{ij} = (a_{ij}, b_{ij})$, $1 \leq i \leq s_x$, $1 \leq j \leq s_y$. Here the stations are shown as located at the centres of the cells on the surface of the domain and that there are no measurements taken in the padded portion of the domain.

where block $\mathbf{G}^{(r)}$ is for the r th depth layer, and there are n_z depth layers. The individual entries in $\mathbf{G}$ correspond to the projections of the contributions from prisms c_{pqr} in the volume to measurement stations, denoted by s_{ij} , at or near the surface. The configurations of the volume and measurement domains are illustrated in Fig. 1. We assume that the measurement stations are all on the surface with coordinates $(a_i, b_j, 0)$ in (x, y, z) . Prism c_{pqr} of the domain has dimensions Δ_x , Δ_y and Δ_z in x , y and z directions with coordinates that are integer multiples of Δ_x , Δ_y and Δ_z , and is indexed by $1 \leq p \leq s_x + p_{xL} + p_{xR} = n_x$, $1 \leq q \leq s_y + p_{yL} + p_{yR} = n_y$, and $1 \leq r \leq n_z$. This indexing assumes that there is padding around the domain in x and y directions by additional borders of p_{xL} , p_{xR} , p_{yL} and p_{yR} cells. The distinction between the padded and unpadded portions of the domain is that there are no measurement stations in the padded regions. This yields $\mathbf{G} \in \mathbb{R}^{m \times n}$ where $m = s_x s_y$, and $n = n_x n_y n_z$, and each $\mathbf{G}^{(r)} \in \mathbb{R}^{m \times n_r}$, where $n_r = n_x n_y$.

In (3), $m \leq n_r \ll n$ and the system is drastically underdetermined for any reasonable discretization of the depth (z) dimension of the volume. Moreover, when n is large the use of the matrix $\mathbf{G}$ requires both significant computational cost for evaluation of matrix-matrix operations and significant storage. Without taking account of structure in $\mathbf{G}$, and assuming that a dot product of real vectors of length n requires $2n$ floating point operations (flops), calculating $\mathbf{G}\mathbf{H}$, for $\mathbf{H} \in \mathbb{R}^{n \times p}$, takes $\mathcal{O}(2nmp)$ flops and storage of matrix $\mathbf{G}$ uses approximately $8mn \times 10^{-9}$ GB.¹ For example, suppose $p = m = n/8$ and $n = 10^6$, then storage of $\mathbf{G}$ requires approximately 1000 GB, and the single matrix multiplication uses $\approx 10^{18}/32$ flops or 10^7 Gflops, without any consideration of additional software and system overheads. These observations limit the ability to do large-scale stabilized inversion of potential field data in real time using current desktop computers, or laptops, without taking into account any further information on the structure of $\mathbf{G}$.

Boulanger & Chouteau (2001) observed that the configuration of the locations of the stations in relation to the domain discretization is significant in generating $\mathbf{G}^{(r)}$ with a symmetric structure and reduces the storage requirements. Then, Bruun & Nielsen (2007) recognized that the structure can be effectively utilized to improve

the efficiency of operations with $\mathbf{G}$ and to further reduce the memory demands. Assuming that the stations are always placed uniformly with respect to the domain prisms, and provided that the distances between stations are fixed in x and y , then matrix $\mathbf{G}^{(r)}$ for the gravity kernel has symmetric BTTB structure. Then, it is possible to embed $\mathbf{G}^{(r)}$ in a BCCB matrix and matrix operations can be efficiently performed using the 2DFFT, as explained by Vogel (2002). This structure was also discussed and then utilized for efficient forward operations with $\mathbf{G}$ by Chen & Liu (2018). They assumed that the stations are placed symmetrically with respect to the domain coordinates, as illustrated for the staggered configuration in Fig. 1 with the stations at the centre of the cells on the surface. Bruun & Nielsen (2007) demonstrated that $\mathbf{G}^{(r)}$ for the magnetic kernel can also exhibit BTTB structure, but they did not use the standard computation of the magnetic kernel integral (Rao & Babu 1991). Hogue *et al.* (2019) provided a thorough derivation of the BTTB structure for $\mathbf{G}^{(r)}$ using the approach of Rao & Babu (1991), for which the sensitivity matrix has blocks that are unsymmetric. The use of padding for the domain, and the modifications required in the generation of the required entries in the matrix $\mathbf{G}^{(r)}$, was discussed. Regardless of whether operations with $\mathbf{G}$ are implemented using the 2DFFT or by direct multiplication, it is far faster to generate $\mathbf{G}$ taking advantage of the BTTB structure (Hogue *et al.* 2019). Here, we are concerned with efficient stabilized inversion of potential field data using this BTTB structure. We refer to Appendix A for a brief discussion of the implementation of the needed operations using $\mathbf{G}$ when implemented using the 2DFFT (Hogue *et al.* 2019).

2.2 Stabilized inversion

The determination of the solution of (1) is an ill-posed problem; even if $\mathbf{G}$ is well-conditioned the problem is underdetermined because $m \ll n$. There is a considerable literature on the solution of this ill-posed problem and we refer in particular to Vatankehah *et al.* (2020b) for a relevant overview, and specifically the use of the unifying framework for determining an acceptable solution of (1) by stabilization. Solution $\mathbf{m}^*$ is estimated as the minimizer of the nonlinear objective function $\Phi_\alpha(\mathbf{m})$ subject to bound constraints

¹ We assume one double floating point number requires 8 bytes and note 1 byte is 10^{-9} GB.

$$\begin{aligned}
\mathbf{m}_{\min} &\leq \mathbf{m} \leq \mathbf{m}_{\max} \\
\mathbf{m}^* &= \arg \min_{\mathbf{m}_{\min} \leq \mathbf{m} \leq \mathbf{m}_{\max}} \{\Phi_{\alpha}(\mathbf{m})\} \\
&= \arg \min_{\mathbf{m}_{\min} \leq \mathbf{m} \leq \mathbf{m}_{\max}} \{\Phi_d(\mathbf{m}) + \alpha^2 \Phi_S(\mathbf{m})\}.
\end{aligned} \quad (4)$$

Regularization parameter α trades off the relative weighting of the weighted data misfit $\Phi_d(\mathbf{m})$, and stabilizer $\Phi_S(\mathbf{m})$, which are given by

$$\begin{aligned}
\Phi_d(\mathbf{m}) &= \|\mathbf{W}_d(\mathbf{G}\mathbf{m} - \mathbf{d}_{\text{obs}})\|_2^2, \text{ and} \\
\Phi_S(\mathbf{m}) &= \|\mathbf{W}_h \mathbf{W}_z \mathbf{W}_L \mathbf{D}(\mathbf{m} - \mathbf{m}_{\text{apr}})\|_2^2.
\end{aligned} \quad (5)$$

The weighting matrices $\mathbf{W}_h$, $\mathbf{W}_z$ and $\mathbf{W}_L$ are all diagonal, with dimensions that depend on the size of $\mathbf{D}$. We assume throughout that $\mathbf{D} = \mathbf{I}_{n \times n}$ and refer to (Vatankhah *et al.* 2020b, eq. 5) for the modification in the weighting matrices that is required for derivative approximations using $\mathbf{D}$. We also use $\mathbf{m}_{\text{apr}} = \mathbf{0}$. When initial estimates for the parameter are available, perhaps from physical measurements, these can be incorporated into $\mathbf{m}_{\text{apr}}$ as an initial estimate for $\mathbf{m}$. The diagonal weighting matrix $\mathbf{W}_d$ has entries $(\mathbf{W}_d)_{ii} = 1/\sigma_i$ where we suppose that the measured data can be given by $\mathbf{d}_{\text{obs}} = \mathbf{d}_{\text{exact}} + \boldsymbol{\eta}$, where $\mathbf{d}_{\text{exact}}$ is the exact but unknown data, and $\boldsymbol{\eta}$ is a noise vector drawn from uncorrelated Gaussian data with variance components σ_i^2 .

In $\mathbf{W} = \mathbf{W}_h \mathbf{W}_z \mathbf{W}_L$, $\mathbf{W}_h$ and $\mathbf{W}_z$ are constant hard constraint and constant depth weighting matrices. The hard constraint matrix can be used to impose specific known values for entries of $\mathbf{m}$ (Boulanger & Chouteau 2001; Vatankhah *et al.* 2018). Here we do not impose any known values and use $\mathbf{W}_h = \mathbf{I}_{n \times n}$. Depth weighting $\mathbf{W}_z$ is routinely used in the context of potential field inversion and is imposed to counteract the natural decay of the kernel with depth. With the same column structure as for $\mathbf{G}$, $\mathbf{W}_z = \text{blockdiag}(\mathbf{W}_z^{(1)}, \dots, \mathbf{W}_z^{(n_z)})$ where $\mathbf{W}_z^{(r)} = (0.5(z_r + z_{r-1}))^{-\beta} \mathbf{I}_{n_r \times n_r}$, $0.5(z_r + z_{r-1})$ is the average depth for depth level r , and β is a parameter that depends on the data set, Li & Oldenburg (1996). The focusing matrix $\mathbf{W}_L$ depends on the parameter vector $\mathbf{m}$ via

$$(\mathbf{W}_L)_{ii} = ((\mathbf{m}_i - (\mathbf{m}_{\text{apr}})_i)^2 + \epsilon^2)^{\frac{\lambda-2}{4}}, \quad i = 1 \dots n, \quad (6)$$

for the i th entry. Parameter λ determines the form of the stabilization, and focusing parameter $0 < \epsilon \ll 1$ is chosen to avoid division by zero. Our choice $\lambda = 1$ yields an approximation to the L_1 norm (Wohlberg & Rodríguez 2007; Vatankhah *et al.* 2017) and is preferred for inversion of potential field data. The implementation makes it easy to switch to $\lambda = 0$, yielding a solution which is compact, or $\lambda = 2$ for a smooth solution. Based on prior studies we use $\epsilon^2 = 1e - 9$, Vatankhah *et al.* (2017).

2.3 Numerical solution

We first reiterate that (4) is only nonlinear in $\mathbf{m}$ through the definition of $\mathbf{W}_L$. Supposing that $\mathbf{W}_L$ is constant and that $\text{null}(\mathbf{W}_d \mathbf{G}) \cap \text{null}(\mathbf{W}) = \mathbf{0}$, then the solution $\mathbf{m}^*$ of (4) without the bound constraints is given analytically by

$$\begin{aligned}
\mathbf{m} &= \mathbf{m}_{\text{apr}} + (\mathbf{G}^T \mathbf{W}_d^T \mathbf{W}_d \mathbf{G} + \alpha^2 \mathbf{W}^T \mathbf{W})^{-1} \\
&\quad \times \mathbf{G}^T \mathbf{W}_d^T \mathbf{W}_d (\mathbf{d}_{\text{obs}} - \mathbf{G} \mathbf{m}_{\text{apr}}).
\end{aligned} \quad (7)$$

Equivalently, assuming that $\mathbf{W}$ is invertible, and defining $\tilde{\mathbf{G}} = \mathbf{W}_d \mathbf{G} \mathbf{W}^{-1}$, $\tilde{\mathbf{r}} = \mathbf{W}_d (\mathbf{d}_{\text{obs}} - \mathbf{G} \mathbf{m}_{\text{apr}})$ and $\mathbf{y} = \mathbf{m} - \mathbf{m}_{\text{apr}}$, then $\mathbf{y}$ solves

the normal equations

$$\mathbf{y} = \mathbf{W}^{-1} (\tilde{\mathbf{G}}^T \tilde{\mathbf{G}} + \alpha^2 \mathbf{I})^{-1} \tilde{\mathbf{G}}^T \tilde{\mathbf{r}}, \quad (8)$$

and $\mathbf{m}^*$ can be found by restricting $\mathbf{y} + \mathbf{m}_{\text{apr}}$ to lie within the bound constraints.

The solution $\mathbf{y}$ of (8) can be used to obtain the iterative solution for (4) using the iteratively reweighted least squares algorithm (IRLS, Vatankhah *et al.* 2020b). Superscript k is used to indicate a variable at an iteration k . Then, α is replaced by $\alpha^{(k)}$, $\mathbf{W}_L$ by matrix $\mathbf{W}_L^{(k)}$ with entries $(\mathbf{W}_L^{(k)})_{ii} = ((\mathbf{m}_i^{(k-1)} - \mathbf{m}_i^{(k-2)})^2 + \epsilon^2)^{\frac{\lambda-2}{4}}$ and $\mathbf{m} - \mathbf{m}_{\text{apr}}$ by $\mathbf{m} - \mathbf{m}^{(k-1)}$, initialized with $\mathbf{W}_L^{(1)} = \mathbf{I}$, and $\mathbf{m}^{(0)} = \mathbf{m}_{\text{apr}}$ respectively. Update $\mathbf{y}^{(k)}$ is the solution of the normal eq. (8), and $\mathbf{m}^{(k)}$ is the restriction of $\mathbf{y}^{(k)} + \mathbf{m}^{(k-1)}$ to the bound constraints.

2.4 Algorithmic details

The IRLS algorithm relies on the use of an appropriate solver for finding $\mathbf{y}^{(k)}$ as the solution of the normal equations (8) for each update k , and a method for estimating the regularization parameter $\alpha^{(k)}$. We consider two solvers for the solution of the normal equations at each iteration; one based on the RSVD, and the second using the GKB. With an appropriate implementation, both solvers can also provide an approximate truncated SVD of $\tilde{\mathbf{G}}$.

The GKB Algorithm 1 uses the factorization $\tilde{\mathbf{G}} \mathbf{A}_{t_p} = \mathbf{H}_{t_p+1} \mathbf{B}_{t_p}$, where $\mathbf{A}_{t_p} \in \mathcal{R}^{n \times t_p}$ and $\mathbf{H}_{t_p+1} \in \mathcal{R}^{m \times t_p+1}$. Steps 6 and 11 of Algorithm 1 apply the modified Gram-Schmidt re-orthogonalization to the columns of $\mathbf{A}_{t_p}$ and $\mathbf{H}_{t_p+1}$, as is required to avoid the loss of column orthogonality. Step 15 provides the terms that give the rank t_p approximate SVD, $(\mathbf{H}_{t_p+1} \mathbf{U}_{t_p}) \boldsymbol{\Sigma}_{t_p} (\mathbf{A}_{t_p} \mathbf{V}_{t_p})^T$, of $\tilde{\mathbf{G}}$ (Renaut *et al.* 2017). The quality of this approximation depends on the conditioning of $\tilde{\mathbf{G}}$, Paige & Saunders (1982). In particular, the projected system of the GKB algorithm inherits the ill-conditioning of the original system, rather than just the dominant terms of the full SVD expansion. Thus, the approximate singular values include dominant terms that are good approximations to the dominant singular values of the original system, as well as very small singular values that approximate the tail of the singular spectrum of the original system. The accuracy of the dominant terms increases quickly with increasing t_p , Paige & Saunders (1982) and the effectiveness of the algorithm depends on both t and t_p .

The RSVD Algorithm 2 includes a single power iteration in Steps 3–6. Without the use of the power iteration in the RSVD it is necessary to use larger projected systems in order to obtain a good approximation of the singular space of the original system, Halko *et al.* (2011). When using the RSVD for potential field inversion it is better to apply a power iteration but the gain from taking more than one power iteration is insignificant as compared to the increased computational time required (Vatankhah *et al.* 2020a). The RSVD, with and without power iteration, also depends on two parameters t and t_p , where here t is the target rank and t_p is size of the oversampled system, $t_p > t$. For given t and t_p the algorithm uses an eigendecomposition with t_p terms to find the SVD approximation of $\tilde{\mathbf{G}}$ with t_p terms. Hence, the total projected space is of size t_p which is then restricted to size t for estimating the approximation of $\tilde{\mathbf{G}}$.³

³We note that using $(\mathbf{Y} + \mathbf{Y}^T)/2$ in Step 10 of Algorithm 2, rather than $\mathbf{Y}$, assures that the matrix is symmetric which is important for the efficiency of eig.

²We use $\mathbf{I}_{n \times n}$ to denote the identity matrix of size $n \times n$.

Given an approximate SVD, there are many efficient techniques that can be used to automatically estimate $\alpha^{(k)}$, and there is extensive literature on the subject, for example, Hansen (2010). Consistent with earlier studies on stabilized inversion, we find $\alpha^{(k)}$, for $k > 1$, using the unbiased predictive risk estimator (UPRE) for the RSVD but the truncated version TUPRE for the GKB, Renaut *et al.* (2017); Vatankhah *et al.* (2018, 2017, 2020a). For UPRE and TUPRE, $\alpha^{(k)}$ is found using the approximate SVD of size $t < t_p$. Whereas the solution $\mathbf{y}^{(k)}(\alpha^{(k)})$ is then calculated using all t_p terms of the approximate SVD when using TUPRE, $\mathbf{y}^{(k)}(\alpha^{(k)})$ is obtained using just the first t terms when applying the UPRE algorithm. The matrix $\mathbf{\Gamma}(\alpha, \mathbf{\Sigma})$ in each algorithm is the diagonal matrix with entries $\sigma_i/(\sigma_i^2 + \alpha^2)$.

Algorithm 1: The GKB algorithm to obtain solution $\mathbf{y}$ of (8).

Input: $\tilde{\mathbf{r}} \in \mathcal{R}^m$, $\tilde{\mathbf{G}} \in \mathcal{R}^{m \times n}$, a target rank t and size of oversampled projected problem t_p , $t < t_p \ll m$.

Output: $\mathbf{y}$.

- 1 Set $\mathbf{a} = \text{zeros}(n, 1)$, $\mathbf{B} = \text{sparse}(\text{zeros}(t_p + 1, t_p))$,
 $\mathbf{H} = \text{zeros}(m, t_p + 1)$, $\mathbf{A} = \text{zeros}(n, t_p)$;
- 2 Set $\beta = \|\tilde{\mathbf{r}}\|_2$, $\mathbf{h} = \tilde{\mathbf{r}}/\beta$, $\mathbf{H}(:, 1) = \mathbf{h}$;
- 3 **for** $i = 1 : t_p$ **do**
- 4 $\mathbf{b} = \tilde{\mathbf{G}}^T \mathbf{h} - \beta \mathbf{a}$;
- 5 **for** $j = 1 : i - 1$ **do**
- 6 $\mathbf{b} = \mathbf{b} - (\mathbf{A}(:, j)^T \mathbf{b}) \mathbf{A}(:, j)$ (modified Gram-Schmidt (MGS))
- 7 **end**
- 8 $\gamma = \|\mathbf{b}\|_2$, $\mathbf{a} = \mathbf{b}/\gamma$, $\mathbf{B}(i, i) = \gamma$, $\mathbf{A}(:, i) = \mathbf{a}$;
- 9 $\mathbf{c} = \tilde{\mathbf{G}} \mathbf{a} - \gamma \mathbf{h}$;
- 10 **for** $j = 1 : i$ **do**
- 11 $\mathbf{c} = \mathbf{c} - (\mathbf{H}(:, j)^T \mathbf{c}) \mathbf{H}(:, j)$ (MGS)
- 12 **end**
- 13 $\beta = \|\mathbf{c}\|_2$, $\mathbf{h} = \mathbf{c}/\beta$, $\mathbf{B}(i + 1, i) = \beta$, $\mathbf{H}(:, i + 1) = \mathbf{h}$;
- 14 **end**
- 15 SVD for sparse matrix: $\mathbf{U}_{t_p} \mathbf{\Sigma}_{t_p} \mathbf{V}_{t_p}^T = \text{svds}(\mathbf{B}, t_p)$;
- 16 Apply UPRE to find α using $\mathbf{U}_{t_p}(:, 1 : t)$ and $\mathbf{\Sigma}_{t_p}(1 : t, 1 : t)$;
- 17 Solution $\mathbf{y} = \|\tilde{\mathbf{r}}\|_2 \mathbf{A}_{t_p} \mathbf{V}_{t_p} \mathbf{\Gamma}(\alpha, \mathbf{\Sigma}_{t_p}) \mathbf{U}_{t_p}(1, :)^T$;

2.5 Computational costs

The computational cost of practical implementations of the GKB or RSVD algorithms for finding the parameter vector $\mathbf{y}^{(k)}$ is of interest when operations with matrix $\mathbf{G}$ are implemented using the 2DFFT. The associated impact of t_p on the comparative costs of these algorithms with increasing m and n is also relevant. In the estimates, we focus on the dominant costs in terms of **flops**, recalling that the underlying cost of a dot product of two vectors of length m is assumed to be $2m$. Further, the costs ignore any overheads of data movement and data access.

First, we address the evaluation of matrix products with $\tilde{\mathbf{G}}$ or $\tilde{\mathbf{G}}^T$ required at Steps 4 and 9 of Algorithm 1 and Steps 2, 4, 6 and 8 of Algorithm 2. Matrix operations with $\mathbf{G}$, rather than $\tilde{\mathbf{G}}$, use the 2DFFT, as described in Appendix A for $\mathbf{G}\mathbf{x}$, $\mathbf{G}^T \mathbf{y}$ and $\mathbf{y}^T \mathbf{G}$, based on the discussion in Vogel (2002). The dominant cost of a single matrix vector operation in each case is $4n_x n_y n_z \log_2(4n_x n_y) = 4n \log_2(4n_r)$. Thus, multiplication with a matrix of size $n \times t_p$ has dominant cost

$$4nt_p \log_2(4n_r), \quad (9)$$

Algorithm 2: The RSVD algorithm with one power iteration to obtain solution $\mathbf{y}$ of (8)

Input: $\tilde{\mathbf{r}} \in \mathcal{R}^m$, $\tilde{\mathbf{G}} \in \mathcal{R}^{m \times n}$, a target matrix rank t and size of oversampled projected problem t_p , $t < t_p \ll m$.

Output: $\mathbf{y}$.

- 1 Generate a Gaussian random matrix $\mathbf{\Omega} \in \mathcal{R}^{t_p \times n}$;
 - 2 $\mathbf{Y} = \mathbf{\Omega} \tilde{\mathbf{G}} \in \mathcal{R}^{t_p \times n}$;
 - 3 $[\mathbf{Q}, \sim] = \text{qr}(\mathbf{Y}^T, 0)$, $\mathbf{Q} \in \mathcal{R}^{n \times t_p}$. (economic QR decomposition);
 - 4 $\mathbf{Y} = \tilde{\mathbf{G}} \mathbf{Q} \in \mathcal{R}^{m \times t_p}$;
 - 5 $[\mathbf{Q}, \sim] = \text{qr}(\mathbf{Y}, 0)$, $\mathbf{Q} \in \mathcal{R}^{m \times t_p}$;
 - 6 $\mathbf{Y} = \mathbf{Q}^T \tilde{\mathbf{G}}$, $\mathbf{Y} \in \mathcal{R}^{t_p \times n}$;
 - 7 $[\mathbf{Q}, \sim] = \text{qr}(\mathbf{Y}^T, 0)$, $\mathbf{Q} \in \mathcal{R}^{n \times t_p}$;
 - 8 $\mathbf{B} = \tilde{\mathbf{G}} \mathbf{Q} \in \mathcal{R}^{m \times t_p}$;
 - 9 Compute $\mathbf{Y} = \mathbf{B}^T \mathbf{B} \in \mathcal{R}^{t_p \times t_p}$;
 - 10 Eigen-decomposition of $\mathbf{B}^T \mathbf{B}$: $[\tilde{\mathbf{V}}, \mathbf{D}] = \text{eig}((\mathbf{Y} + \mathbf{Y}^T)/2)$;
 - 11 $\mathbf{S} = \text{diag}(\sqrt{|\text{real}(\mathbf{D})|})$, $[\mathbf{S}, \text{indsort}] = \text{sort}(\mathbf{S}, 'descend')$;
 - 12 $\tilde{\mathbf{S}}_t = \text{diag}(\mathbf{S}(1 : t))$, $\tilde{\mathbf{V}} = \tilde{\mathbf{V}}(:, \text{indsort}(1 : t))$,
 $\tilde{\mathbf{U}} = \tilde{\mathbf{V}}/(\mathbf{S}(1 : t)^T)$;
 - 13 Apply UPRE to find α using $\tilde{\mathbf{U}}$, $\tilde{\mathbf{S}}_t$, and $\mathbf{B}^T \tilde{\mathbf{r}}$;
 - 14 Solution $\mathbf{y} = \mathbf{Q} \tilde{\mathbf{V}} \mathbf{\Gamma}(\alpha, \tilde{\mathbf{S}}_t) \tilde{\mathbf{U}}^T (\mathbf{B}^T \tilde{\mathbf{r}})$;
 - 15 Note if we form $\tilde{\mathbf{V}}_t = \mathbf{Q} \tilde{\mathbf{V}}$; and $\tilde{\mathbf{U}}_t = \mathbf{B} \tilde{\mathbf{U}} \tilde{\mathbf{S}}_t^{-1}$, then $\tilde{\mathbf{U}}_t \tilde{\mathbf{S}}_t \tilde{\mathbf{V}}_t^T$ is a t -rank approximation of matrix $\tilde{\mathbf{G}}$;
-

Table 1. Computational costs for standard operations. Matrix $\mathbf{G} \in \mathcal{R}^{m \times n}$, $\mathbf{X} \in \mathcal{R}^{n \times t}$, $\mathbf{Y} \in \mathcal{R}^{m \times t}$, sparse bidiagonal $\mathbf{B} \in \mathcal{R}^{t+1 \times t}$, $\mathbf{A}^T \mathbf{A} \in \mathcal{R}^{t \times t}$, and $\mathbf{Z} \in \mathcal{R}^{m \times t}$. The modified Gram-Schmidt for $\mathbf{C} \in \mathcal{R}^{m \times i}$ is repeated for $i = 1 : t$, yielding the given estimate. These costs use the basic unit that the inner product $\mathbf{x}^T \mathbf{x}$ for $\mathbf{x}$ of length n requires $2n$ operations.

GX	$\mathbf{G}^T \mathbf{Y}$	svds(B)	MGS(C)	$\text{eig}(\mathbf{A}^T \mathbf{A})$	$[\mathbf{Q}, \sim] = \text{qr}(\mathbf{Z})$
$2mnt$	$2mnt$	$6t(m + t)$	$2mt^2$	$9t^3$	$4t^2(m - t/3)$

in place of $2mnt_p$. In the IRLS algorithm we need to use operations with $\tilde{\mathbf{G}} = \mathbf{W}_d \mathbf{G} \mathbf{W}^{-1}$ rather than $\mathbf{G}$. Specifically, we need

$$\tilde{\mathbf{G}} \mathbf{x} = \mathbf{W}_d (\mathbf{G} (\mathbf{W}^{-1} \mathbf{x})). \quad (10)$$

where multiplications with diagonal matrices are insignificant and the 2DFFT is applied for the evaluation of $\mathbf{G} \mathbf{w}$ where $\mathbf{w} = \mathbf{W}^{-1} \mathbf{x}$. Matrix-matrix operations are also required but, clearly, operations $\tilde{\mathbf{G}}^{(k)} \mathbf{X}$, $(\tilde{\mathbf{G}}^{(k)})^T \mathbf{Z}$, $\mathbf{Z}^T \tilde{\mathbf{G}}^{(k)}$ are just loops over the relevant columns (or rows) of the matrices $\mathbf{X}$ and $\mathbf{Z}$, with the appropriate weighting matrices provided before and after application of the 2DFFT. The details are provided in Appendix A.

To determine the impact of t (and t_p) we estimate the major cost of the IRLS algorithm by assessing the dominant costs of using the GKB and RSVD algorithms for finding the solution of (8). The assumptions for the dominant costs of standard algorithms, given in Table 2, are quoted from Golub & Van Loan (2013). The cost for eig depends significantly on problem size and symmetry. Here t can be quite large, when m is large, but the matrix is symmetric, hence we use the estimate $9t^3$ (Golub & Van Loan 2013, Algorithm 8.3.3). For a sparse bidiagonal matrix $\mathbf{B}$ the cost of using function svds is at most quadratic in the variables. A comment on the cost of the qr operation is also required. Generally, in forming the QR factorization of a matrix we would maintain the information on the Householder reflectors that are used in the reduction of the matrix to upper triangular form, rather than accumulating the matrix $\mathbf{Q}$. The cost is reduced significantly if $\mathbf{Q}$ is not accumulated. In Steps 2, 4, 6

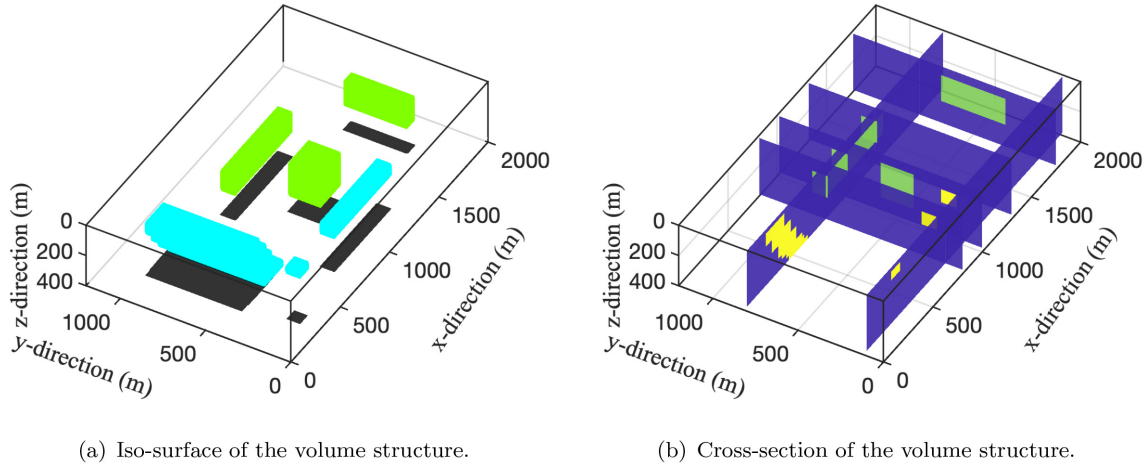


Figure 2. The basic volume structure within the domain of size $2000 \times 1200 \times 400$. The extent of each structure is shown by the shadow on the base of the volume. The same structure is used for the results using the padded domain.

Table 2. Dimensions of the volume used in the experiments with scaling of the small problem size (25,15,2) by scale factor ℓ in each dimension. m and n are the dimensions of the measurement vector and the model parameters, respectively, $\mathbf{G} \in \mathcal{R}^{m \times n}$. Without padding $m = s_x s_y = 375\ell^2$ and $n = mn_z$, where $n_x = s_x$ and $n_y = s_y$. $n_{\text{pad}} = n_x n_y n_z$ is the volume dimension with padding, using $n_x = s_x + 2 \text{round}(\text{pad } s_x)$ and $n_y = s_y + 2 \text{round}(\text{pad } s_y)$ where $\text{pad} = 5$ per cent on each side of the domain so that $p_{xL} = p_{xR} = \text{round}(\text{pad } s_x)$, and similarly for s_y .

ℓ	(s_x, s_y, n_z)	m	n	n_{pad}	τ_2^g	τ_2^m	SNR ^g	SNR ^m
4	(100, 60, 8)	6000	48000	58080	.0138	.0081	24.0	24.0
5	(125, 75, 10)	9375	93750	113710	.0147	.0083	24.0	24.0
6	(150, 90, 12)	13500	162000	199200	.0133	.0074	24.0	24.0
7	(175, 105, 14)	18375	257250	310730	.0133	.0070	24.0	24.0
8	(200, 120, 16)	24000	384000	464640	.0133	.0071	24.0	24.1
9	(225, 135, 18)	30375	546750	662450	.0133	.0069	24.0	24.0
10	(250, 150, 20)	37500	750000	916320	.0132	.0070	24.0	24.0
11	(275, 165, 22)	45375	998250	1206500	.0135	.0075	24.0	24.0
12	(300, 180, 24)	54000	1296000	1568160	.0135	.0075	24.0	24.0

and 8 of Algorithm 2, however, we will need to evaluate products of $\mathbf{Q}$ with $\tilde{\mathbf{G}}$ or its transpose. To take advantage of the 2DFFT we then need to first evaluate a product of $\mathbf{Q}$ with a diagonal scaling matrix, which amounts to accumulation of matrix $\mathbf{Q}$. Experiments, that are not reported here, show that it is more efficient to accumulate $\mathbf{Q}$ as given in Algorithm 2, rather than to first evaluate the product of $\mathbf{Q}$ with a diagonal scaling matrix without pre accumulation. The cost for accumulating $\mathbf{Q}$ is $2t^2(m - t/3)$ for a matrix of size $m \times t$, (Golub & Van Loan 2013, p. 255) yielding a total cost for the qr step of $4t^2(m - t/3)$, as also reported by Xiang & Zou (2013).

Using the results in Table 1 we can estimate the dominant costs of Algorithms 1 and 2. In the estimates we do not distinguish between costs based on t_p or t . It is reasonable to assume $t < m$ and $t_p \approx t$, when m is large. We also ignore the distinction between m and n_r , where $n_r > m$ for padded domains. The cost of finding $\alpha^{(k)}$ and then evaluating $\mathbf{y}^{(k)}$ is of lower order than the dominant costs involved with finding the needed factorizations. Using LOT to indicate the lower order terms that are ignored, and assuming the calculation without the use of the 2DFFT, the most significant terms yield

$$\text{Cost}_{\text{GKB}} = 4nmt + 2t^2(n + m) + LOT \quad (11)$$

$$\begin{aligned} \text{Cost}_{\text{RSVD}} &= 8nmt + 4t^2(2n + m - t) + 2mt^2 + 9t^3 + LOT \\ &= 8nmt + 4t^2(2n + 3/2m) + 5t^3 + LOT. \end{aligned} \quad (12)$$

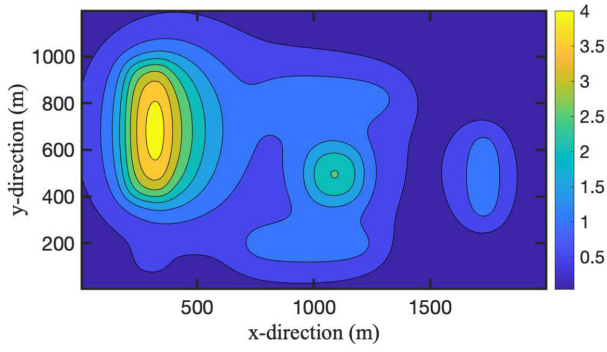
When using the 2DFFT, the first two entries $2mnt$ in Table 1 are replaced by $4nt \log_2(4n_r)$. Then, using $m \approx n_r$, it is just the first term in each estimate that is replaced leading to the costs with the 2DFFT as

$$\text{Cost}_{2\text{DFFT}_{\text{GKB}}} = 8nt \log_2(4m) + 2t^2(n + m) + LOT \quad (13)$$

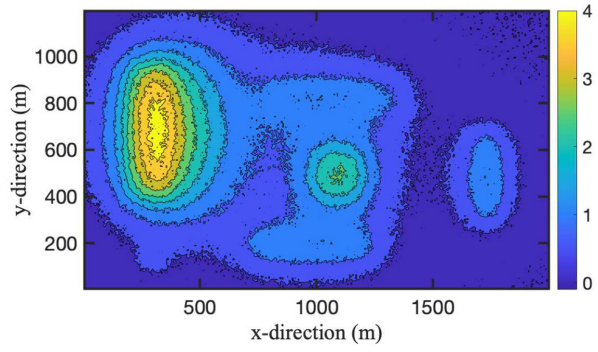
$$\begin{aligned} \text{Cost}_{2\text{DFFT}_{\text{RSVD}}} &= 16nt \log_2(4m) + 4t^2(2n + 3/2m) \\ &\quad + 5t^3 + LOT. \end{aligned} \quad (14)$$

Eqs (11) and (12) and (13) and (14) suggest, just in terms of flop count, that it is generally more than two times as expensive to use the RSVD in place of the GKB. Hence, to obtain a solution at a comparable cost we would hope to use a smaller t for the RSVD than for the GKB. This expectation contradicts earlier experiments contrasting these algorithms for the inversion of gravity data, using the RSVD without power iteration (Vatankhah *et al.* 2018). Alternatively, it would be desired that the RSVD should converge in the IRLS far faster than the GKB. Further, theoretically, the gain of using the 2DFFT is that the major terms are $8t^2n$ and $2t^2n$ for the RSVD and GKB, respectively, as compared to $8nmt > 8t^2n$ and $4mnt > 2t^2n$, noting $t < m$. Even though the costs should go up with order nt^2 eventually with the 2DFFT, this is still far slower than the increase mnt that arises without taking advantage of the structure.

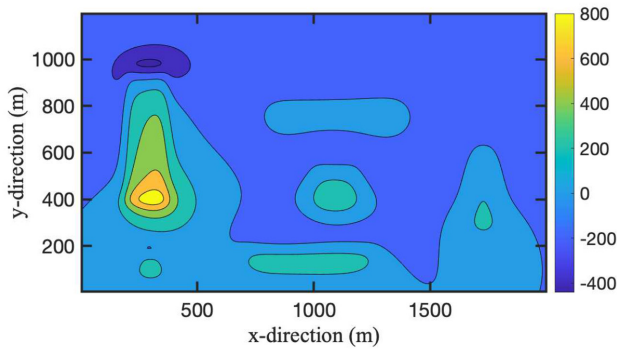
Measuring the computational cost just in terms of the flop count can be misleading (Xiang & Zou 2013). It was noted by Xiang &



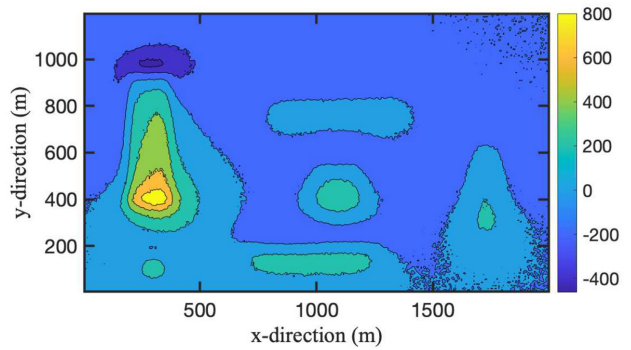
(a) True gravity anomaly



(b) Noisy gravity anomaly



(c) True magnetic anomaly



(d) Noisy magnetic anomaly

Figure 3. The calculated true and noisy anomalies for the volume structure given in Fig. 2(a), where the units are mGal and nT for gravity and magnetic data, respectively. The anomalies used for the inversion using the padded domain are exactly the same as given here.

Zou (2013) that a distinction between the GKB and RSVD algorithms, where the latter is without the power iteration, is that the operations required in the GKB involve many BLAS2 (matrix–vector) operations, requiring repeated access to the matrix or its transpose, as compared to BLAS3 (matrix–matrix) operations for RSVD implementations. The Householder operations within the *qr* algorithm also involve BLAS2 operations. Hence, when using MATLAB, the major distinction should be between the use of functions that are *builtin* and compiled, or are not compiled. The functions *qr* and *eig* are *builtin* and hence optimized, but all other operations that are used in the two algorithms do not use any compiled code. There is no compiled option for the MGS used in steps 6 and 11 of Algorithm 1. Almost all operations in Algorithm 2 use *builtin* functions or BLAS3 operations for matrix products that do not involve the matrices with BTTB structure. Thus, in the evaluation of the two algorithms in the MATLAB environment, we will consider computational costs directly, rather than just the estimates given by (13) and (14). The estimates of the flop counts are more broadly-relevant for higher-level programming environments. In all the implementations none of the results quoted will use multiple cores or GPUs.

3 NUMERICAL EXPERIMENTS

We now validate the fast and efficient methods for inversion of potential field data using the BTTB structure of the gravity and magnetic kernel matrices.

3.1 Implementation parameter choices

Diagonal depth weighting matrix $\mathbf{W}_z$ uses $\beta = 0.8$ for the gravity problem, and $\beta = 1.4$ for the magnetic problem, consistent with recommendations by Li & Oldenburg (1998) and Pilkington (1997), respectively. Diagonal $\mathbf{W}_d$ is determined by the noise in the data, and hard constraint matrix $\mathbf{W}_h$ is taken to be the identity. No prior information is imposed on the parameters and we use $\mathbf{m}_{\text{apr}} = 0$. Regularization parameter $\alpha^{(k)}$ is found using the UPRE method for $k > 1$, but initialized with appropriately large $\alpha^{(1)}$ given by

$$\alpha^{(1)} = \left(\frac{n}{m}\right)^{3.5} \frac{\sigma_1}{\text{mean}(\sigma_i)}. \quad (15)$$

Here σ_i are the estimates of the ordered singular values for $\mathbf{W}_d \mathbf{G} \mathbf{W}^{-1}$ given by the use of the RSVD or GKB algorithm, and the mean value is taken only over $\sigma_i > 0$. This follows the practice

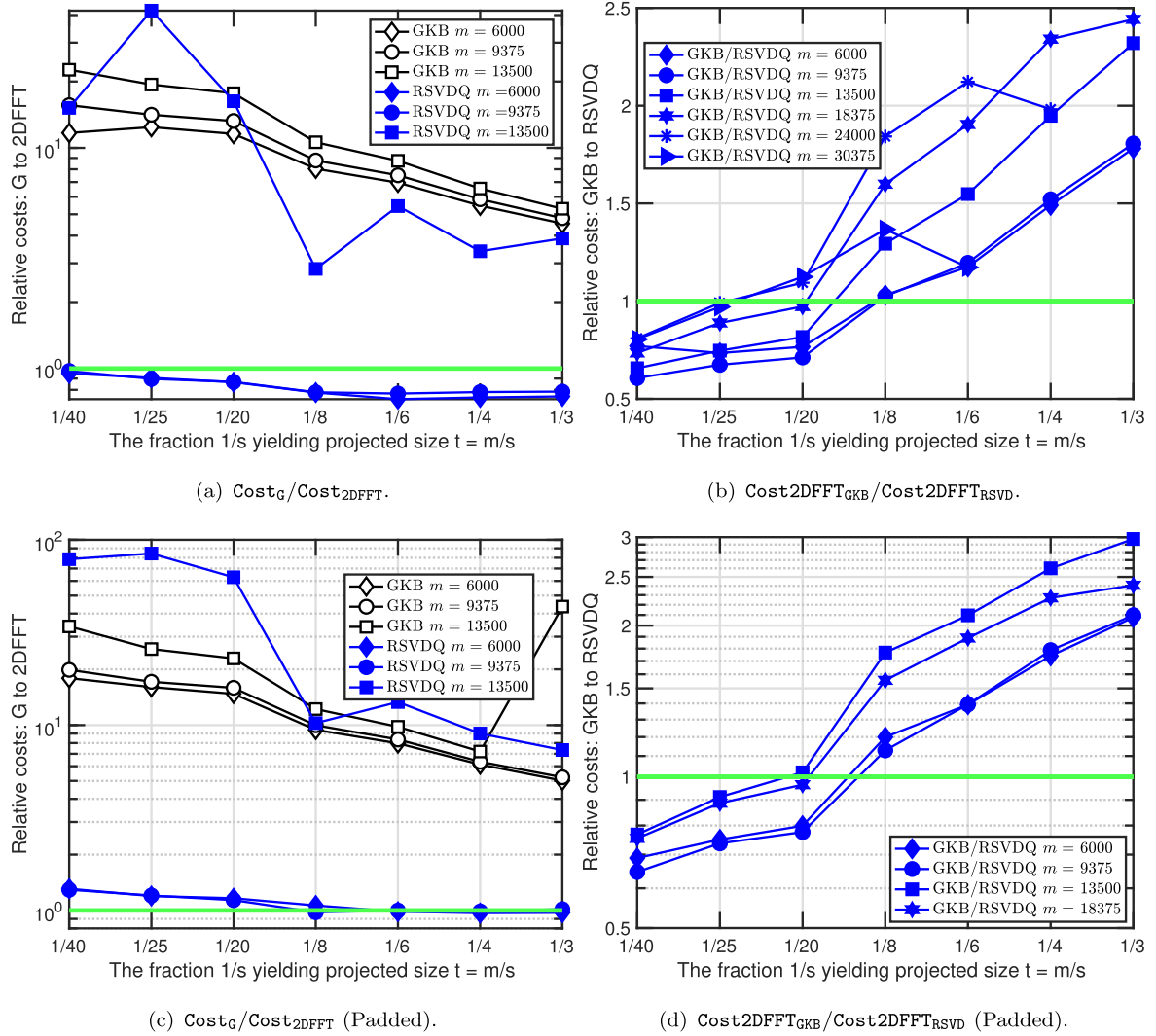


Figure 4. The x-axis in these plots is the size t used for the projected problem in terms of a fraction $1/s$ of m with rounding $t = \text{floor}(m/s)$. It is not on a linear or log scale. The size of m is given in the legends and corresponds to the problem sizes indicated in Table 2. Panel (a) shows the relative computational costs for one iteration of the IRLS algorithm using $\mathbf{G}$ directly as compared to the 2DFFT, as indicated by $\text{Cost}_G / \text{Cost}_{2\text{DFFT}}$. No result is given for $m = 13\,500$ because the direct implementation using matrix $\mathbf{G}$ requires too much memory for the specific computing environment. The lines with solid blue symbols are for results using the RSVD algorithm, and the open black symbols are for the GKB algorithm. Panel (b) shows the relative computational cost for one iteration of the IRLS algorithm for inversion using the GKB as compared to the RSVD algorithm ($\text{Cost}_{2\text{DFFT}_{\text{GKB}}} / \text{Cost}_{2\text{DFFT}_{\text{RSVD}}}$). Here, the results include problem sizes $m = 24\,000$ and $m = 30\,375$, the asterisk and the right pointing triangle, respectively. Because m is larger the relative costs are only calculated up to $m/4$ and $m/8$ for $m = 24\,000$ and $m = 30\,375$, respectively. Panels (c)–(d) provide the equivalent data as panels (a) and (b) with padding, and without the larger problems for m . In all cases the green horizontal line is at $y = 1$ for the measured quantity. Values below the horizontal line indicate that the numerator is less than the denominator, corresponding to the greater efficiency of the quantity in the numerator.

implemented in Vatankeh et al. (2018) and Renaut et al. (2017) for studies using the RSVD and GKB, and which was based on the recommendation to use a large value for $\alpha^{(1)}$, Farquharson & Oldenburg (2004).

For all simulations, the IRLS algorithm is iterated to convergence as determined by the χ^2 test for the predicted data,

$$\|\mathbf{W}_d(\mathbf{G}\mathbf{m}^{(k)} - \mathbf{d}_{\text{obs}})\|_2^2 \leq m + \sqrt{2m}, \quad (16)$$

or

$$\frac{\|\mathbf{W}_d(\mathbf{G}\mathbf{m}^{(k)} - \mathbf{d}_{\text{obs}})\|_2^2}{m + \sqrt{2m}} \leq 1. \quad (17)$$

If this is not attained for $k \leq K_{\text{max}}$, the iteration is terminated. Noisy data are generated for observed data $\mathbf{d}_{\text{obs}} = \mathbf{d}_{\text{exact}} + \boldsymbol{\eta}$ using

$$\boldsymbol{\eta}_i = (\tau_1 |(\mathbf{d}_{\text{exact}})_i| + \tau_2 \|\mathbf{d}_{\text{exact}}\|_\infty) \mathbf{e}_i, \quad (18)$$

where $\mathbf{e}$ is drawn from a Gaussian normal distribution with mean 0 and variance 1. The pairs (τ_1, τ_2) are chosen to provide a signal-to-noise ratio (SNR), as calculated by

$$\text{SNR} = 20 \log_{10} \frac{\|\mathbf{d}_{\text{exact}}\|_2}{\|\mathbf{d}_{\text{obs}} - \mathbf{d}_{\text{exact}}\|_2}, \quad (19)$$

that is approximately constant across the increasing resolutions of the problem. Recorded for all simulations are (i) the values of the

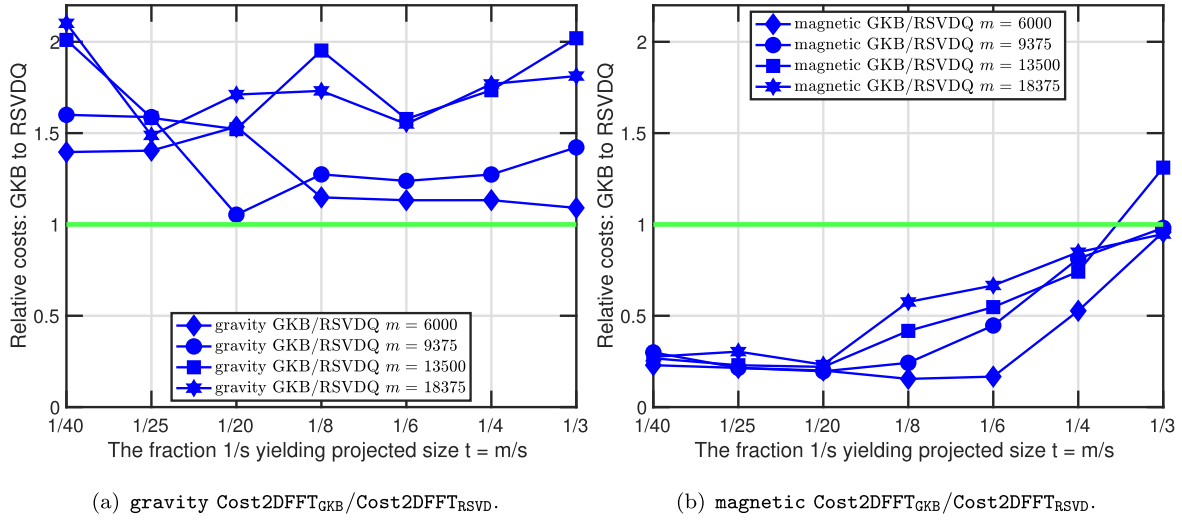


Figure 5. Computational cost to convergence of the IRLS algorithm for inversion using the GKB as compared to the RSVD algorithm, $\text{Cost2DFFT}_{\text{GKB}}/\text{Cost2DFFT}_{\text{RSVD}}$, for the gravity and magnetic problems respectively, in panels (a) and (b). The x-axis in these plots is the size t used for the projected problem in terms of a fraction $1/s$ of m with rounding $t = \text{floor}(m/s)$. The legends indicate the problem size m , as in Fig. 4. In all cases the green horizontal line is at $y = 1$ for the measured quantity. Values below the horizontal line indicate that the numerator is less than the denominator, corresponding to the greater efficiency of the quantity in the numerator.

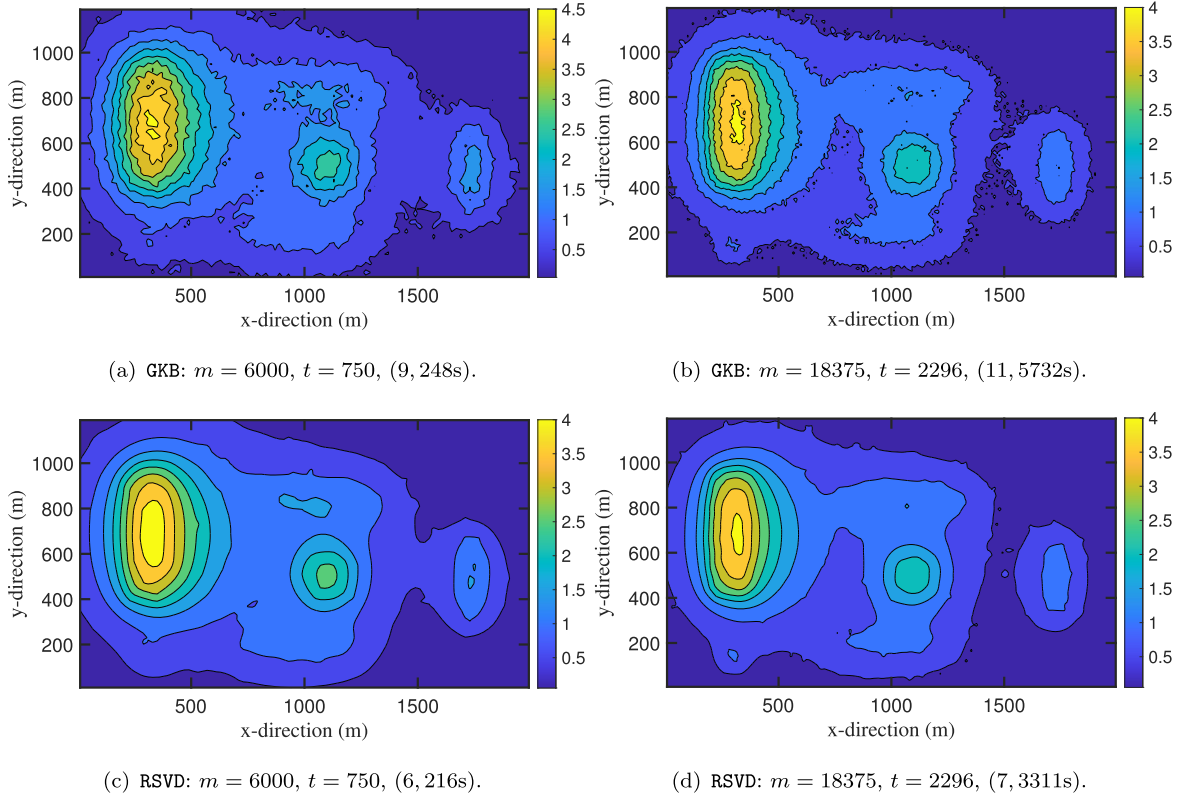


Figure 6. The predicted anomalies obtained using GKB in Figs 6(a)–(b) and RSVD in Figs 6(c)–(d) for gravity data. In the captions are m , $t = \text{floor}(m/8)$, and the pairs (K, Costs) . The units for the anomalies are mGal.

relative error $\text{RE}^{(k)}$, as defined by

$$\text{RE}^{(k)} = \frac{\|\mathbf{m}_{\text{exact}} - \mathbf{m}^{(k)}\|_2}{\|\mathbf{m}_{\text{exact}}\|_2}, \quad (20)$$

(ii) the number of iterations to convergence K which is limited to 25 in all cases, (iii) the scaled χ^2 estimate given by (17) at the final iteration and (iv) the time to convergence measured in seconds, or to iteration $K = 25$ when convergence is not achieved.

3.1.1 The sizes of the projected and enlarged spaces: t and t_p

The RSVD and GKB algorithms provide approximations for the spectral expansion of $\hat{\mathbf{G}}$, with the quality of these approximations dependent on both t and t_p . The GKB algorithm inherits the ill-conditioning of $\hat{\mathbf{G}}$ but the RSVD approach provides the dominant terms, and is not impacted by the tail of the spectrum. A typical recommendation for the RSVD algorithm is to enlarge t by a fixed amount, $t_p = t + p$

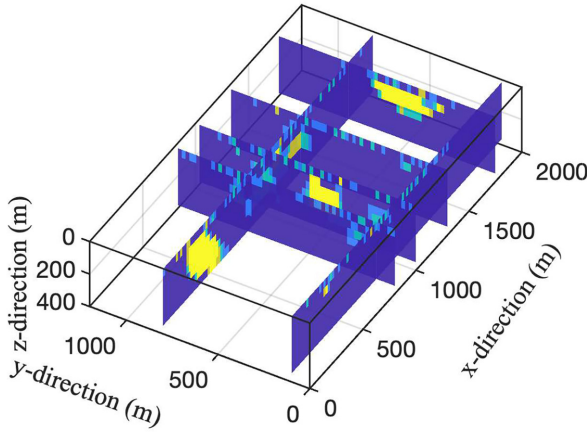
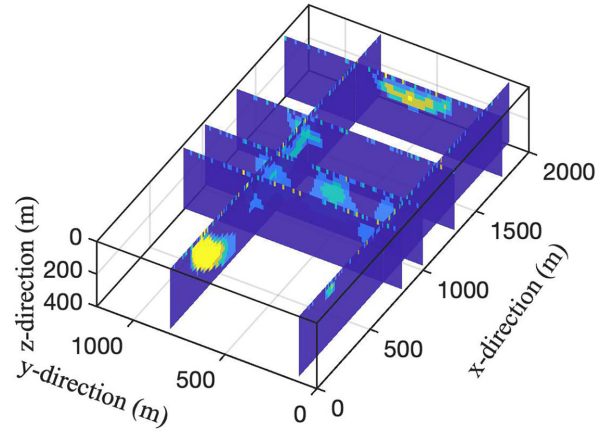
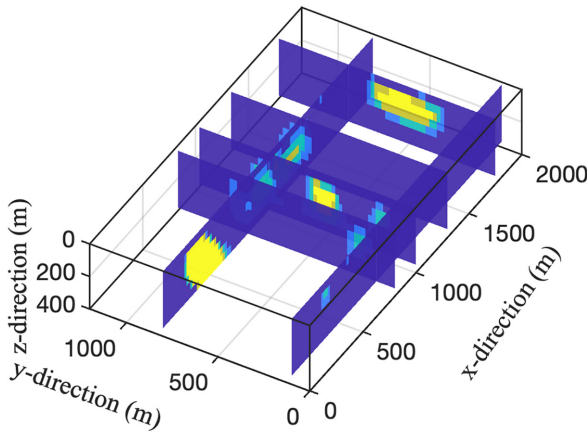
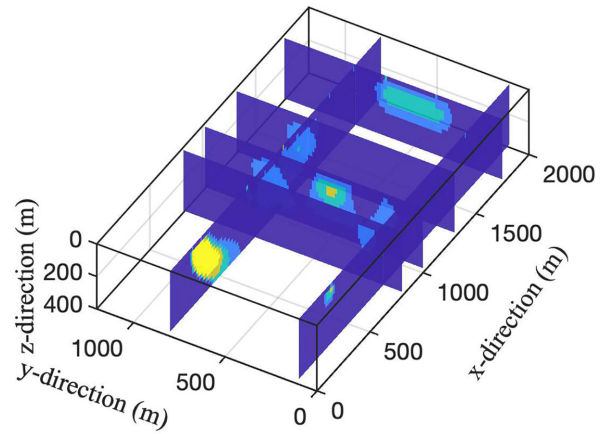
(a) GKB: $m = 6000$, $t = 750$, $(.76, .90)$.(b) GKB: $m = 18375$, $(.75, .81)$.(c) RSVD: $m = 6000$, $t = 750$, $(.57, .96)$.(d) RSVD: $m = 18375$, $t = 2296$, $(.60, .95)$.

Figure 7. The reconstructed volumes corresponding to the gravity anomalies in Figs 6(a)–(d). Parameters $t = 750$ and $t = 2296$ are for $t = \text{floor}(m/8)$. In the captions m , $t = \text{floor}(m/8)$, and the pairs $[\text{RE}, \chi^2/(m + \sqrt{2}m)]$.

for some p generally smaller than t , for example $p = 10$ (Halko *et al.* 2011). This will not be robust for finding a good approximation of the dominant space of size t for the GKB algorithm. In our simulations we enlarge the space by 5 per cent using $t_p = \text{floor}(1.05t)$. This will include terms from the tail of the spectrum for the GKB but the top t terms from the projected space of size $t_p > t$ will be more accurate estimates of the true dominant t terms than if obtained with $t_p = t$ (Paige & Saunders 1982). This enlargement of the space also provides a suitable oversampling for the RSVD algorithm. Effectively, by using a 5 per cent enlargement of t , we assume that the first t terms from the t_p approximation provide good approximations of the dominant t spectral components of the original matrix $\tilde{\mathbf{G}}$ for both algorithms. For the GKB this uniform choice for t_p is a suitable compromise between taking t_p too small and contaminating the solutions by components from the less accurate approximations of the small components, and a reliable, but larger, choice for t_p that provides a good approximation of the dominant terms without major increase in computational costs.

The size of t in relation to m is also crucial. For the RSVD algorithm with a single power iteration a suitable choice for t , when $t_p = t + 10$, is $t \approx m/s$, where $s \approx 8$ for the gravity problem and $s \approx 4$ for magnetic data inversion (Vatankhah *et al.* 2020a). These problem

sizes are smaller, s is larger, from the requirements that $s \approx 6$ and $s \approx 2$, respectively, when no power iteration is included. For the inversion of gravity data using the GKB algorithm $t_p \approx m/20$ has been suggested (Vatankhah *et al.* 2017). To provide a viable comparison of the costs and accuracies using the GKB and RSVD algorithms, we pick a range of t as a fraction of m given by $t = \text{floor}(m/s)$, where $s = 40, 25, 20, 8, 6, 4$ and 3 .

3.2 Synthetic data

For the validation of the algorithms, we pick a volume structure with a number of boxes of different dimensions, and a six-layer dipping dike. The same structure is used for generation of the gravity and magnetic potential field data. For gravity data the densities of all aspects of the structure are set to 1 g cm^{-3} , with the homogeneous background set to 0 g cm^{-3} . For the magnetic data, the susceptibilities in SI units are 0.06 for the dipping dike, one extended well and the very small well. The three other structures have susceptibilities set to 0.04. The distinction between these structures with different susceptibilities is illustrated in the illustration of the iso-structure in Fig. 2(a) and the cross-section in Fig. 2(b).

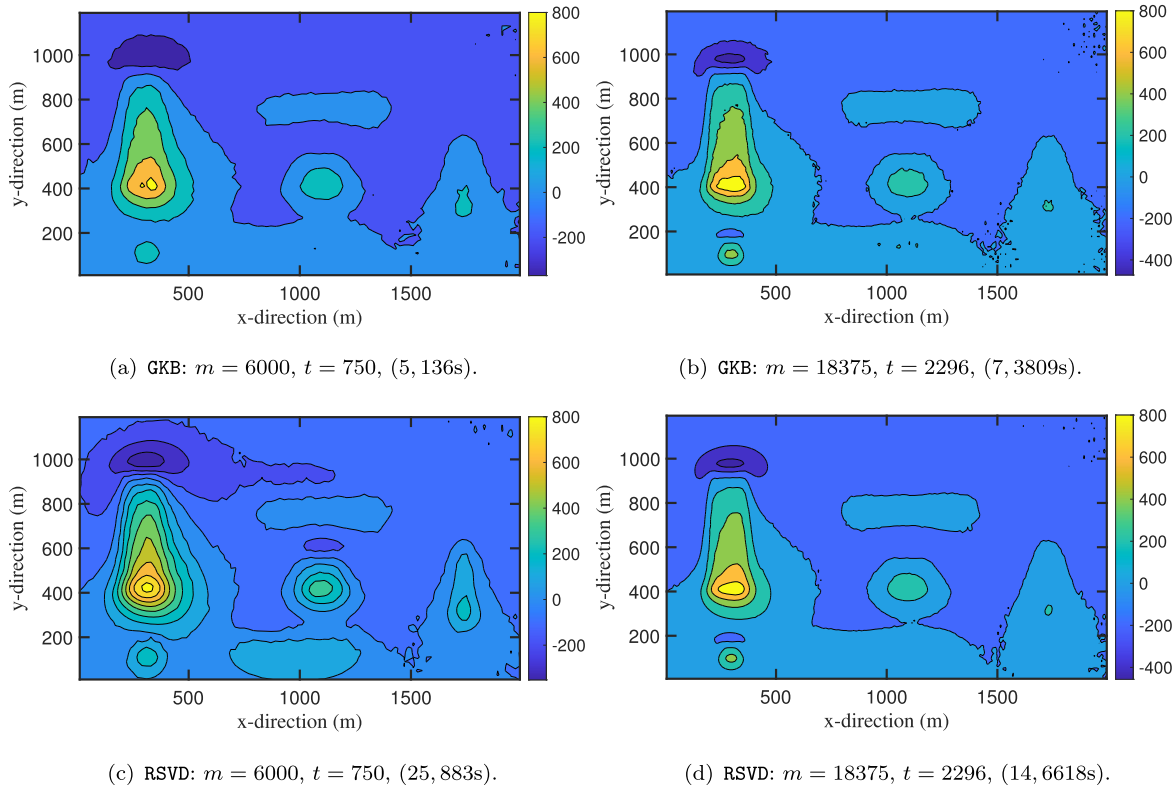


Figure 8. For magnetic data the predicted anomalies obtained using GKB in Figs 8(a)–(b) and RSVD in Figs 8(c)–(d). In the captions are m , $t = \text{floor}(m/8)$, and the pairs (K, Costs) . The units for the anomalies are nT.

The domain volume is discretized in x , y and z into the number of blocks as indicated by triples (s_x, s_y, n_z) with increasing resolution for increasing values of these triples. They are generated by taking $(s_x, s_y, n_z) = (25, 15, 2)$, and then scaling each dimension by scaling factor $\ell \geq 4$ for the test cases, correspondingly, $s_x s_y = 375$ is scaled by ℓ^2 with increasing ℓ , yielding a minimum problem size with $m = 6000$ and $n = 48\,000$. The grid sizes are given by the triples $(\Delta_x, \Delta_y, \Delta_z) = (2000/s_x, 1200/s_y, 400/n_z)$. The problem sizes considered for each simulation are detailed in Table 2. For padding we compare the case with $\text{pad} = 0$ and 5 per cent across x and y dimensions. These are rounded to the nearest integer yielding $p_{xL} = p_{xR} = \text{round}(\text{pad} s_x)$, and $n_x = s_x + 2 \text{round}(\text{pad} s_x)$. n_y is calculated in the same way, yielding $n = (s_x + 2 \text{round}(\text{pad} s_x))(s_y + 2 \text{round}(\text{pad} s_y))n_z$. Certainly, the decision to use $\text{pad} = 5$ per cent is quite large, but is chosen to demonstrate that the solutions obtained using the 2DFFT are robust to boundary conditions, and thus not impacted by the restriction due to lack of padding or very small padding.

For these structures and resolutions, noisy data are generated as given in (18) to yield an SNR of approximately 24 across all scales as calculated using (19). This results in different choices of τ_1 and τ_2 for each problem size and dependent on the gravity or magnetic data case, denoted by (τ_1^g, τ_2^g) and (τ_1^m, τ_2^m) , respectively. In all cases we use $\tau_1^g = \tau_1^m = 0.02$ and adjust τ_2 . The parameter choices of τ_2^g and τ_2^m used for the simulations are detailed in Table 1. As an example, we illustrate the true and noisy data for gravity and magnetic data, when $m = 54\,000$, in Fig. 3.

3.3 Numerical results

The validation and analysis of the algorithms for the inversion of the potential field data is presented in terms of (i) the cost per iteration

of the algorithm (Section 3.3.1), (ii) the total cost to convergence of the algorithm (Section 3.3.2) and (iii) the quality of the obtained solutions, (Section 3.3.3). Supporting quantitative data that summarize the illustrated results are presented as Tables in Appendix B.

3.3.1 Comparative cost of RSVD and GKB algorithms per IRLS iteration

We investigate the computational cost, as measured in seconds, for one iteration of the inversion algorithm using both the direct multiplications using matrix $\mathbf{G}$ (and $\mathbf{G}^T$), and the circulant embedding, for the problem sizes m as indicated in Table 2, using both the RSVD and GKB algorithms for the inversion of magnetic data. For fair comparison, the timing results that are reported use MATLAB release (2019b) implemented on the same iMac 4.2 GHz Quad-Core Intel Core i7 with 32 GB RAM, unless otherwise stated. In this environment, the size of the matrix $\mathbf{G}$ is too large for effective memory usage when $\ell > 6$. The comparative timing results for one step of the IRLS algorithm are illustrated in Fig. 4. The specific values for problem sizes $m = 6000$ – $18\,375$ are given in Table B1. Results for the inversion of gravity data are qualitatively the same and are reported in Renaut *et al.* (2020).

It is not beneficial to use the 2DFFT in place of the matrix $\mathbf{G}$ for the smaller scale implementation of the RSVD algorithm, when $m = 6000$ or 9375 , all the ratios of the relative costs are below the horizontal line (Fig. 4a). For $m \geq 13\,500$ the situation is reversed and the use of the 2DFFT with the RSVD is more efficient. All of the simulations using the GKB with the 2DFFT in place of the matrix $\mathbf{G}$ are more efficient. The relative gain in reduced computational cost by using the 2DFFT depends on the algorithm used within the IRLS inversion algorithm. The decrease in efficiency for a given problem size,

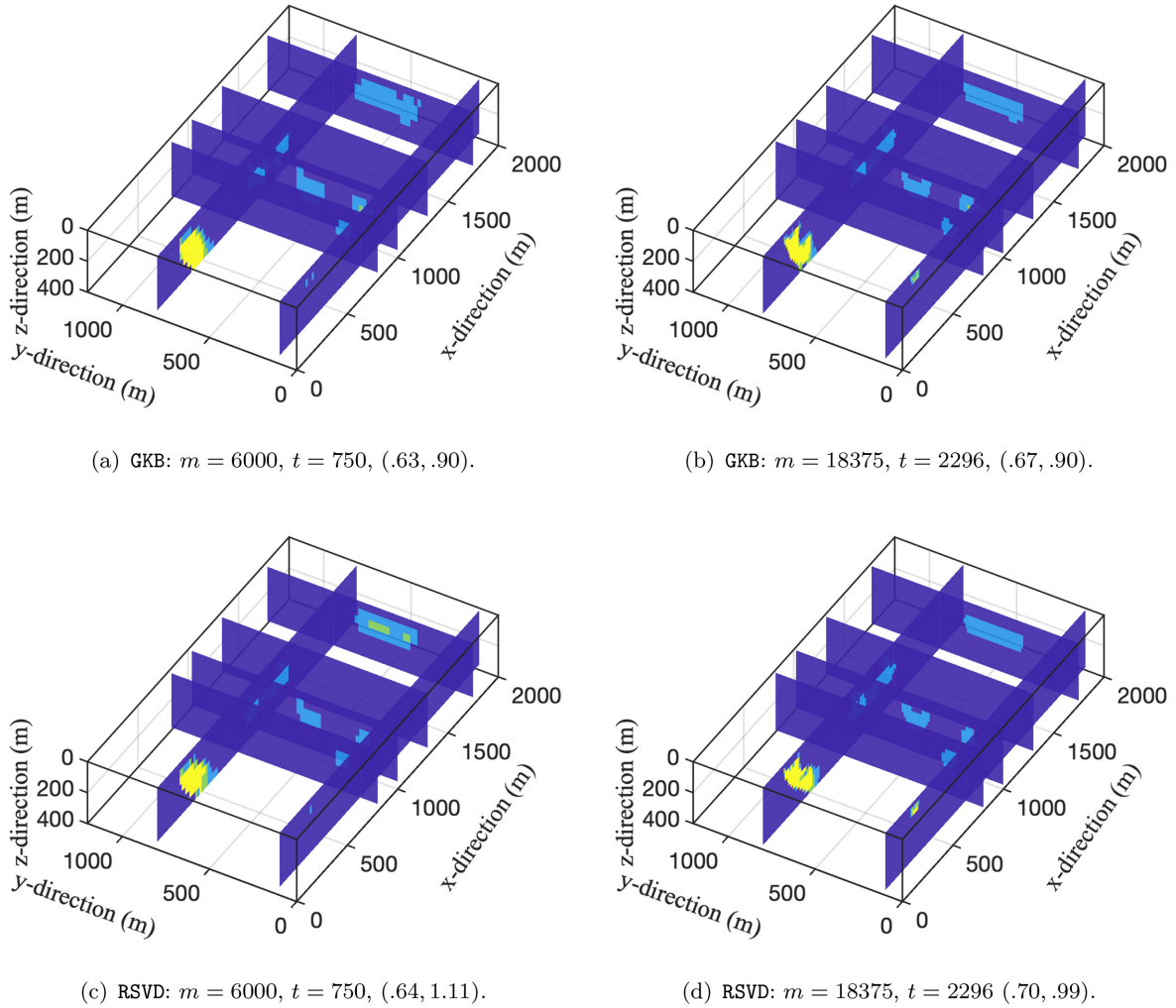


Figure 9. The reconstructed volumes corresponding to the magnetic anomalies in Figs 8(a)–(d). Parameters $t = 750$ and $t = 2296$ are for $t = \text{floor}(m/8)$. In the captions are m , $t = \text{floor}(m/8)$, and the pairs $[\text{RE}, \chi^2/(m + \sqrt{2}m)]$.

fixed m but increasing t , is explained by the theoretical discussion relating to equations (13) and (14). As t increases the impact of the efficient matrix multiplication using the 2DFFT is reduced. Fig. 4(b) demonstrates that the relative costs comparing GKB to RSVD are not constant across all t . The GKB is generally cheaper for smaller t , and the RSVD cheaper for larger t . These results confirm the analysis of the computational cost in terms of flops provided in (13) and (14) for small t . The relative computational costs increase from roughly 0.6–2.5, increasing with both m and t (Fig. 4b). This improved relative performance of RSVD with increasing m and t appears to violate the flop count analysis in (13) and (14). As discussed in Section 2.5, this is a feature of the implementation. While RSVD is implemented using the MATLAB builtin function `qr` which uses compiled code for faster implementation, GKB only uses builtin operations for performing the MGS re-orthogonalization of the basis matrices $\mathbf{A}_{t_p}$ and $\mathbf{H}_{t_p}$. The conclusions from Figs 4(a) and (b) are confirmed for padded domains (Figs 4c and d). Here the relative efficiency of using the 2DFFT for the RSVD algorithm approximately holds for all m , with the relative values approaching $y = 1$ as t increases (Fig. 4c). Moreover, the higher efficiency of the GKB breaks down at a smaller t than occurs without padding in place (Fig. 4d). In general, as m and t increase these results demonstrate the benefit of using the 2DFFT and the RSVD solver.

3.3.2 Comparative cost of RSVD and GKB algorithms to convergence

The computational cost of the IRLS algorithm for solving the inversion problem to convergence depends on t , the use of GKB or RSVD algorithms, and is problem specific. The relative total computational costs to convergence, $\text{Cost2DFFT}_{\text{GKB}}/\text{Cost2DFFT}_{\text{RSVD}}$, are illustrated via Figs 5(a) and (b), for the gravity and magnetic results, respectively. The specific timing results are provided in Table B2. The number of iterations to convergence, and the achieved RE, for the volumes with and without padding, are given in Tables B3 and B4 for the gravity and magnetic problems, respectively.

The RSVD algorithm is always most efficient for the solution of the gravity problem (Fig. 5b), which is consistent with the conclusion presented by Vatankehah *et al.* (2018) for the RSVD without power iteration. The RSVD algorithm generally converges more quickly and yields a smaller relative error (Table B3). Good results can be achieved for relatively small t as compared to m , and $t \approx m/8$ leads to acceptable errors. For comparable choices of t the errors using the GKB are generally larger (Table B3).

There is a distinct benefit to the use of the GKB algorithm, except for large t , for the magnetic problem (Fig. 5a). The RSVD algorithm generally requires more iterations than the GKB algorithm, and the

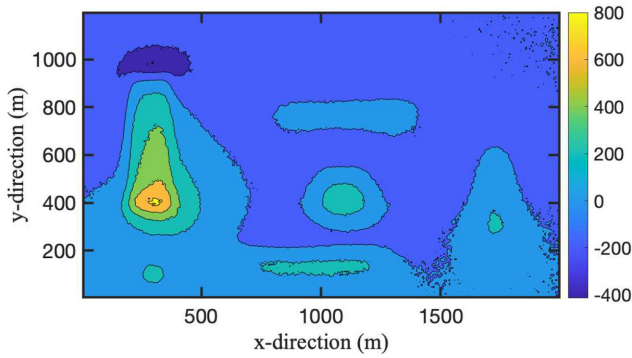
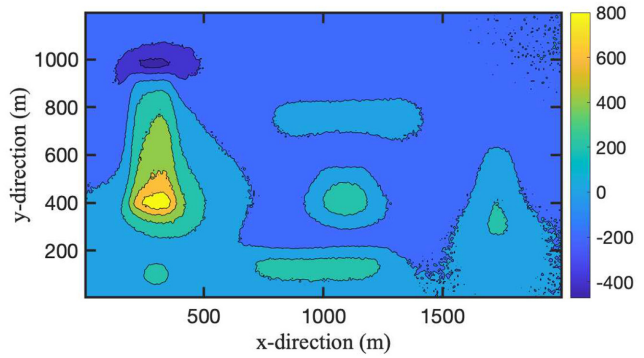
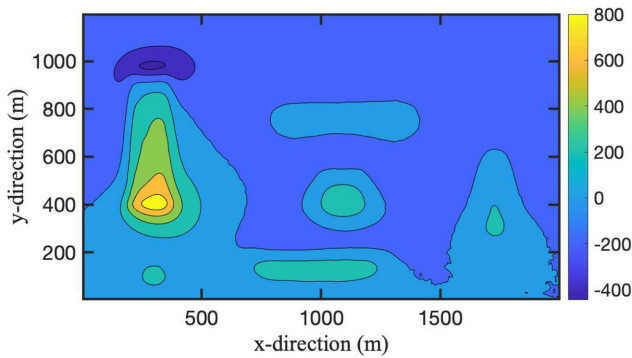
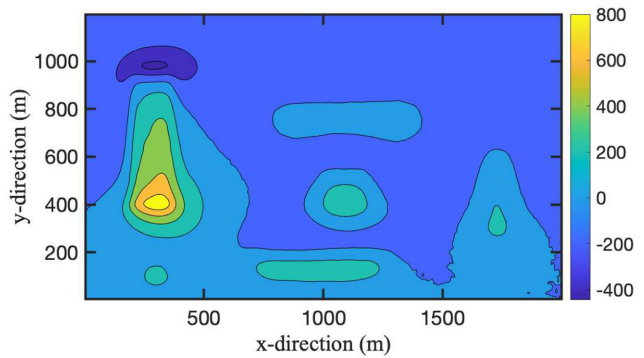
(a) GKB: $m = 45375$, $t = 2268$, (9, 13595s).(b) GKB: $m = 54000$, $t = 2700$, (8, 21649s).(c) RSVD: $m = 45375$, $t = 2268$, (15, 21266s).(d) RSVD: $m = 54000$, $t = 2700$, (16, 41981s).

Figure 10. The magnetic anomalies using the GKB and RSVD algorithms in Figs 10(a)–(b) and (c)–(d), respectively. In the captions are m , $t = \text{floor}(m/8)$ and the pairs (K, Costs) . The units for the anomalies are nT.

obtained relative errors are then comparable, or slightly larger (Table B4). For the smaller problems the RSVD algorithm often reaches $K_{\max}$ without convergence. In these cases it does not mean that RSVD blows up, but that the noise level is not achieved within $K_{\max}$ iterations due to the limited size of t . On the other hand, for these same cases GKB converges in less than half the number of iterations. The slow convergence of RSVD occurs when t is small relative to m , $t = \text{floor}(m/s)$ with $s = 40, 25$ and 20 . This verifies that the RSVD needs to take a large projected subspace t for smaller m in order to capture the required dominant spectral space, and confirms the conclusions presented in Vatankeh et al. (2020a).

Our results lead to a new conclusion concerning these two algorithms for solving the magnetic data inversion problem. In particular, the GKB algorithm should be adopted for inversion of magnetic data. Further, the relative error obtained using the GKB generally decreases with increasing t , and it is sufficient to use subspaces with $t \approx \text{floor}(m/8)$. It remains to verify these assertions by illustrating the results of the inversions and the predicted anomalies for a selection of cases.

3.3.3 Illustrating solutions with increasing m and t

We first examine a set of solutions for which the timing results were compared in Section 3.3.2. The predicted gravity anomalies with $m = 6000$ are generally less accurate than with $m = 18375$ (Fig. 6). The reconstructed volumes show that the RSVD algorithm provides better results in all cases, and the high resolution, $m = 18375$, results are very good, (Figs 7a–d). For the high resolution solution obtained for $m = 18375$ the reduced cost of the RSVD algorithm as compared to the GKB algorithm is significant. Good results are also obtained at the coarser resolution, $m = 6000$, with the RSVD algorithm. There is a significant reduction in computational cost from about 55 min to less than 5 min.

The GKB algorithm provides predicted magnetic anomalies that are in good agreement with the true data (Figs 8a–b). For $m = 6000$ the RSVD algorithm has not converged to the tolerance at $K_{\max}$ iterations; the scaled χ^2 estimate is 1.11. The predicted magnetic anomaly obtained in this case is less acceptable and the structures of the reconstructed volume are not as well-resolved as with $m = 18375$ (Figs 9c–d). The small well near the surface is almost not identified in Fig. 9(c) but is clear at the higher resolution. The other

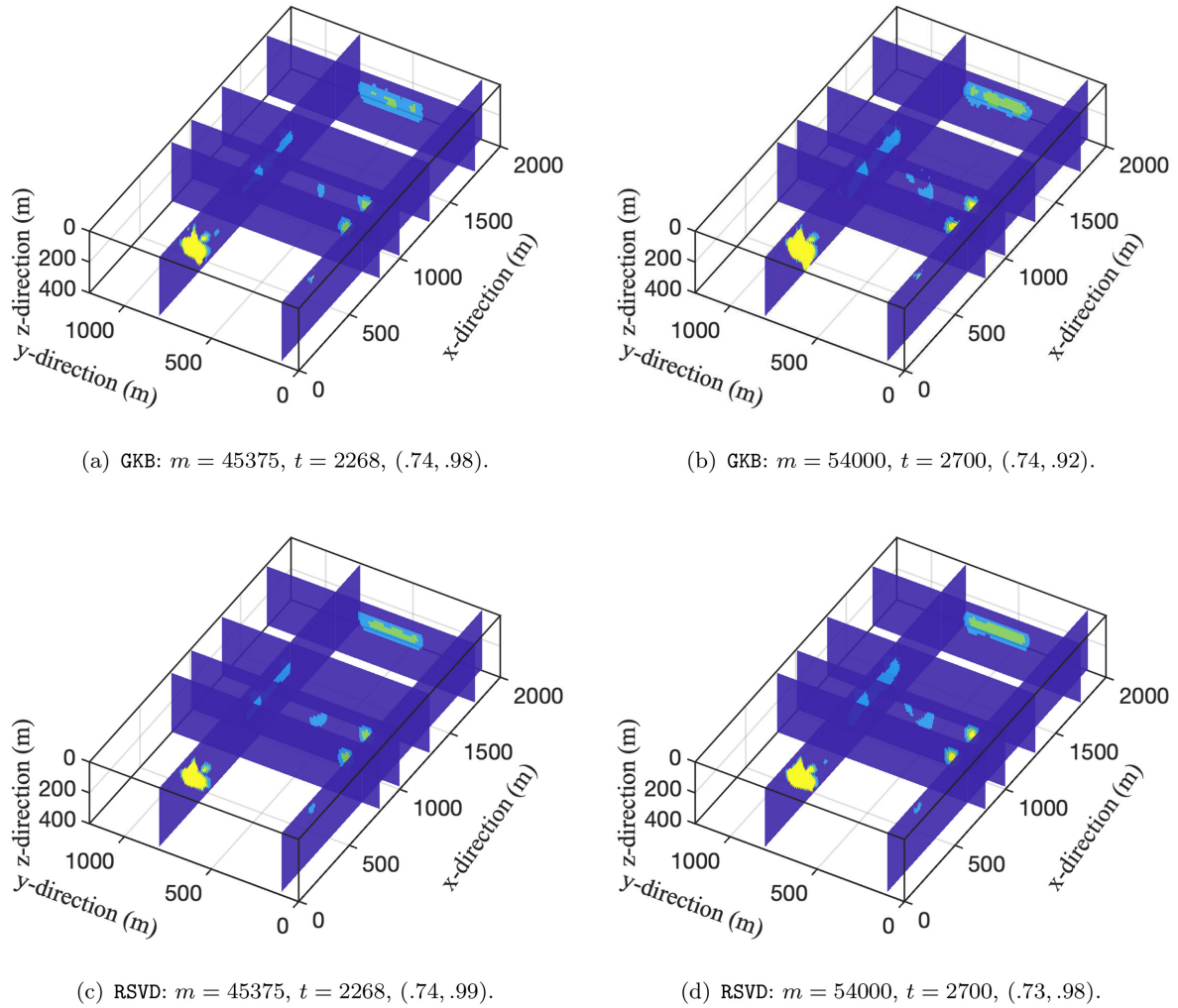


Figure 11. The reconstructed volumes corresponding to the magnetic anomalies in Figs 10(a)–(d). Parameters $t = 2268$ and $t = 2700$ are for $t = \text{floor}(m/20)$. In the captions are m , $t = \text{floor}(m/8)$ and the pairs $[\text{RE}, \chi^2/(m + \sqrt{2}m)]$.

Table 3. Inversion of magnetic data as illustrated in Fig. 12 for $m = 3844$ on a grid of 62×62 stations, with $\Delta x = \Delta y = 100$ m and padding of five cells in both x and y -directions, yielding blocks of size $n_r = 5184$. The inversion uses the GKB algorithm with $t = 480 [\text{floor}(m/8)]$ and $t_p = 504$. The noise in the algorithm uses (18) as given for the simulations with $\tau_1 = 0.02$ and $\tau_2 = 0.018$. These results are obtained using a MacBook Pro laptop with 2.5 GHz Dual-Core Intel Core i7 chip and 16GB memory.

n	n_z	Δz	K	$\alpha^{(1)}$	$\alpha^{(K)}$	$\chi^2/(m + \sqrt{2}m)$	Cost(s)
103680	20	100	17	$4.60e + 05$	8558	0.87	334
207360	40	50	18	$5.36e + 06$	5887	0.90	754
305856	59	33	19	$2.07e + 07$	4930	0.70	1126
414720	80	25	18	$6.09e + 07$	4116	0.95	1513
518400	100	20	18	$1.33e + 08$	3701	0.94	2018
616896	119	16	18	$2.43e + 08$	3386	0.90	2095
829440	160	12	18	$6.90e + 08$	2933	0.95	3091
1036800	200	10	18	$1.51e + 09$	2627	0.95	3690
1238976	239	8	18	$2.80e + 09$	2396	0.96	4389

structures in the domain are also resolved better with $m = 18375$ (Fig. 9d). If there is a large amount of data and a high resolution volume is required, then it is important to use the GKB algorithm in order to limit the computational cost, with the cost decreasing from about 110 min to about 63 min. For $m = 6000$ good solutions are obtained using the GKB algorithm in just over 2 min.

We now investigate the quality of solutions obtained for magnetic data using higher resolution data sets to verify the conclusion based on small data sets that GKB is preferred over RSVD. Problem sizes $(m, n) = (45\,375, 998\,250)$ and $(54\,000, 1\,296\,000)$ are solved using $t = \text{floor}(m/20)$, yielding $t = 2268$ and $t = 2700$, respectively. The largest matrix required by both algorithms is of size $n \times t_p$ and

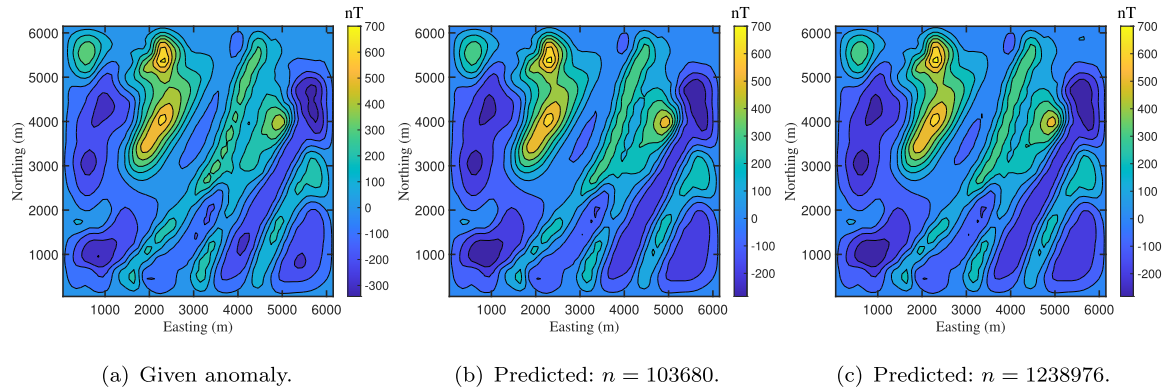


Figure 12. The given magnetic anomaly in Fig. 12(a) and the obtained predicted anomalies for the inversion using the parameters for the first and last lines of data in Table 3 in Figs 12(b)–(c), respectively.

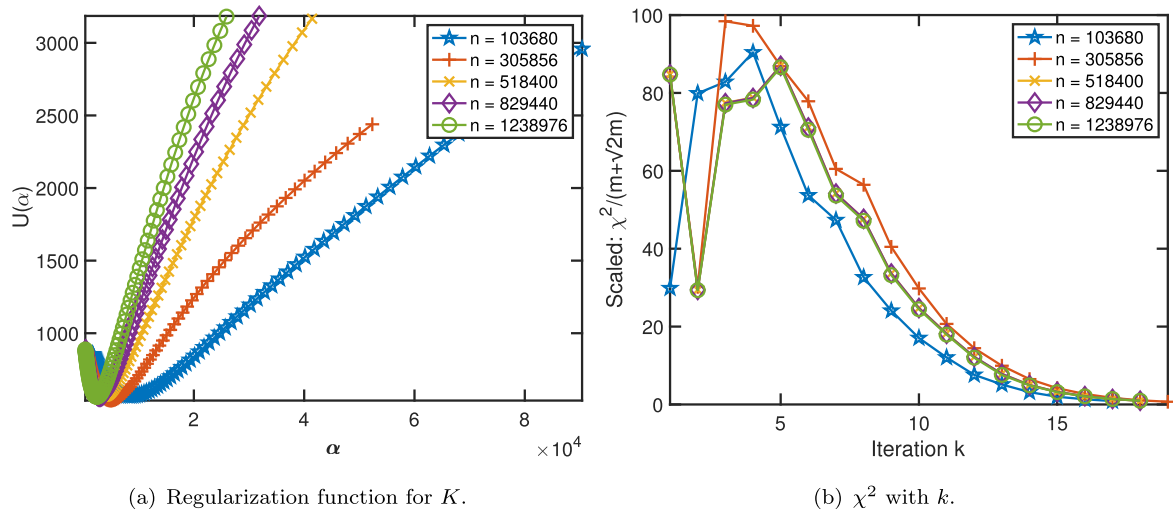


Figure 13. The plot of the regularization function $U(\alpha)$ for the UPRE algorithm, at the final iteration K for increasing values of n as indicated in Table 3 in Fig. 13(a) and the progression of the scaled χ^2 estimate as a function of iteration k and for increasing n in Fig. 13(b).

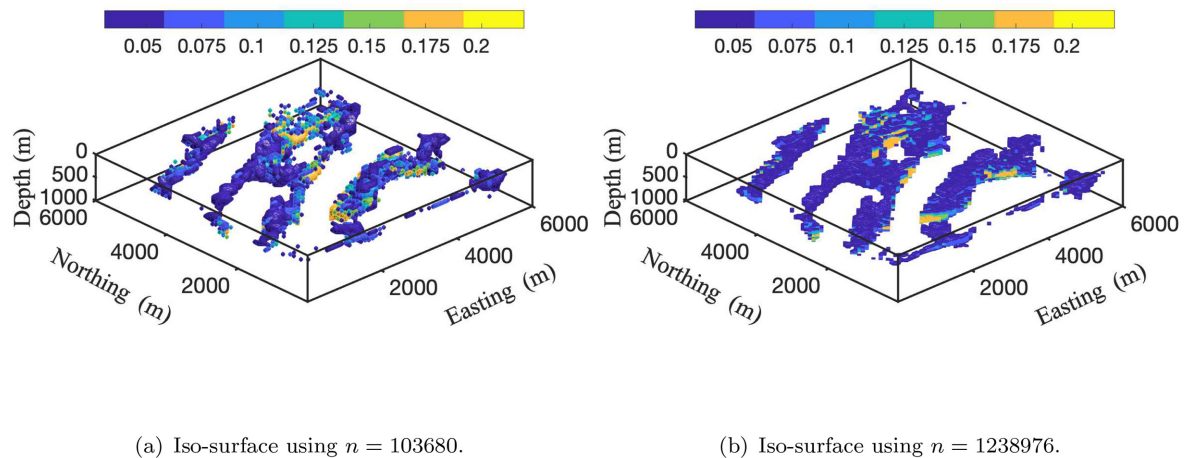


Figure 14. The reconstructed volumes showing parameters $\kappa > 0.05$ and depth from 0 to 1000, corresponding to the predicted anomalies in Fig. 13.

requires 17.7 and 27.4 GB, in each case. This memory demand is too large for implementation on the environment with just 32 GB RAM. A desktop computer with the Intel(R) Xeon (R) Gold 6138 CPU 2.00 GHz chip and with MATLAB release 2019b is used for these experiments.

The predicted magnetic anomalies are always better for the larger problem (Fig. 10). There are significant artefacts at the coarser resolution when using the RSVD (Fig. 10c) but the reconstructed volume is acceptable (Fig. 11c). The cost of achieving this solution is about 5.9 hr, close to the 6 hr required for solution of the $m = 54\,000$

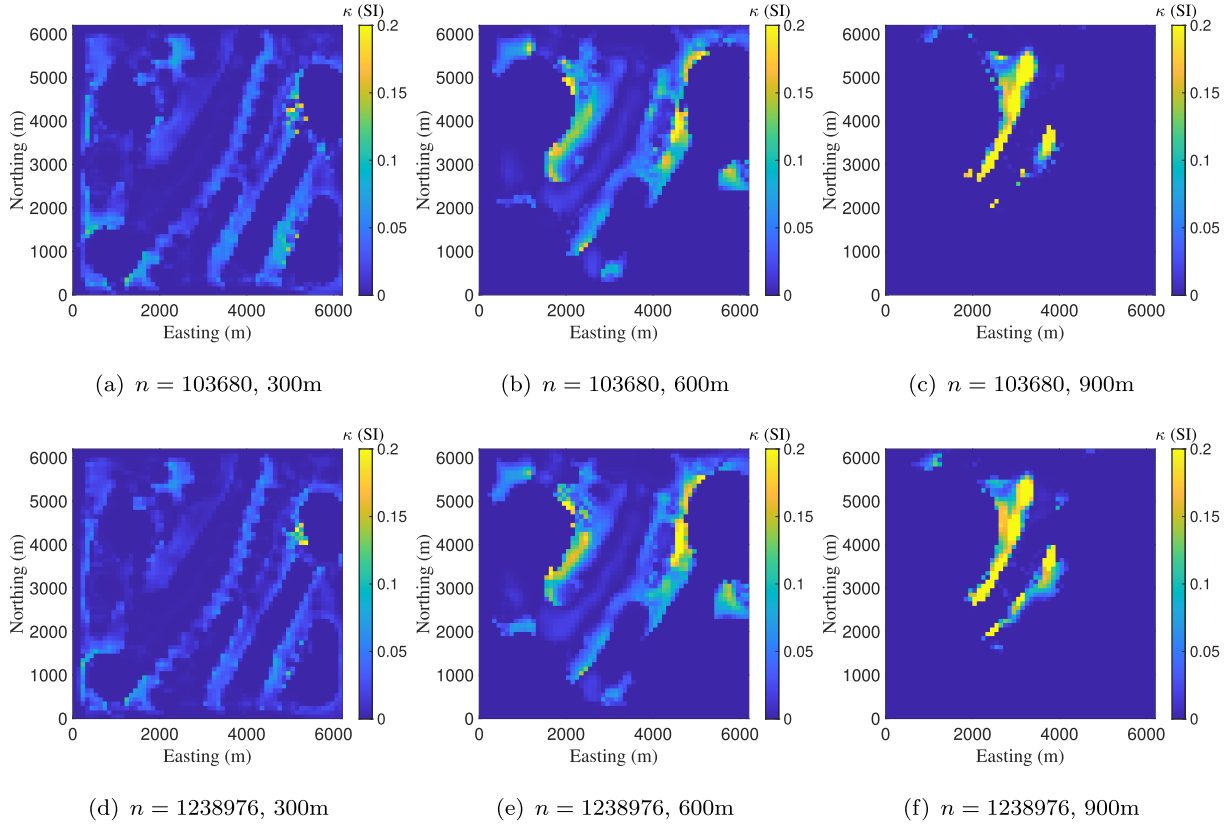


Figure 15. Slices through the volumes illustrated in Fig. 14 for depths 300, 600 and 900.

problem using GKB. For GKB there seems little gain in using $m = 54\,000$. Acceptable results are achieved in 227 min (approximately 3.7 hr) $m = 45\,375$ (Fig. 10a). These results demonstrate that the requirement $t \approx \text{floor}(m/8)$ can be relaxed for larger problems that are too large for the computing environment when using this choice of t . Here it is sufficient to use a relatively smaller projected space, $t \approx \text{floor}(m/20)$.

Numerical experiments for the inversion of gravity data that are reported in Renaut *et al.* (2020), similar to the testing for the magnetic data, confirm that the RSVD algorithm with power iteration is to be preferred for the inversion of gravity data, yielding acceptable solutions at lower cost than when using the GKB algorithm.

3.4 Real data

For validation of the simulated results on a practical data set we apply the GKB algorithm for the inversion of a magnetic field anomaly that was collected over a portion of the Wuskwatim Lake region in Manitoba, Canada. This data set was discussed by Pilkington (2009) and also used by Vatankhah *et al.* (2020a) for inversion using the RSVD algorithm with a single power iteration. Further details of the geological relevance of this data set is given in these references. The use of this data set allows direct comparison with the existing results. Here we use a grid of $62 \times 62 = 3184$ measurements at 100 m intervals in the east-north direction with padding of five cells yielding a horizontal cross section of size 72×72 in the east-north directions. The depth dimension is discretized with $\Delta z = 100$ m, yielding a regular cube, to $\Delta z = 8$ m for rectangular prisms with a smaller edge length in the depth dimension for a total depth of 2000

m, and providing increasing values of n from 103 680 to 1 238 976 (Table 3). The given magnetic anomaly is illustrated in Fig. 13(a).

In each inversion the GKB algorithm is run with $\mathbf{m}_{\text{apr}} = 0$, $t = 480 = \text{floor}(m/8)$, and $t_p = 504$. A noise distribution based on (18) is used using $\tau_1 = 0.02$ and $\tau_2 = 0.018$. The computations are performed on a MacBook Pro laptop with 2.5 GHz Dual-Core Intel Core i7 chip and 16GB memory.

All inversions converge to the tolerance $\chi^2/(m + \sqrt{2m}) < 1$ in no more than 19 iterations for all problem sizes (Table 3). The measured computational costs show that it is feasible to invert for large parameter volumes, in times ranging from just under 5 min for the coarsest resolution, to just over 73 min for the volume with the highest resolution (Table 3). The UPRE function, $U(\alpha)$, has a well-defined minimum at the final iteration for all resolutions, including those not shown (Fig. 13a). Thus, $\alpha^{(K)}$ minimizes the unbiased predictive risk of the solution and indicates that the solution is neither over nor under regularized. Qualitatively, the scaled χ^2 convergence curve is independent of n (Fig. 13b), suggesting that the noise level in the measured data has been met independent of the depth resolution of the reconstruction.

The predicted anomalies provide better agreement to the measured anomaly, as compared to (Vatankhah *et al.* 2020a, fig. 15b), with respect to structure and the given values (Figs 12b–c). More structure is seen in the volumes presented in Figs 14(a) and (b) as compared to (Vatankhah *et al.* 2020a, fig. 19), and the increased resolution provides greater detail in Fig. 14(b) as compared to Fig. 14(a). The volume reconstructions are presented for the depth from 0 to 1000 m only, but there is little structure evident at greater depth, not shown here. Slices at 300, 600 and 900 m contrast the detail at these depths for the lower and higher resolutions (Figs 15a–f). Inversion for higher resolution leads to more structure at

increased depth (Fig. 14). The results using the GKB with $t = 480$ are consistent with those using the RSVD with $t = 1100$ (Vatankhah et al. 2020a). The RSVD algorithm using one power iteration did not converge within 50 steps, under the same configurations for m , n and t .

4 CONCLUSIONS AND FUTURE WORK

Two algorithms for the focused inversion of potential field data measured on a uniform grid have been examined and validated for the independent inversion of gravity and magnetic data sets. Due to the uniform grid, the sensitivity matrix $\mathbf{G}$ has a BTB structure for each depth level of the model space, and all operations with $\mathbf{G}$ can be accomplished efficiently using 2-D fast Fourier transforms. The storage requirements for $\mathbf{G}$ are significantly reduced. With the decreases in computational cost and memory, it becomes feasible to solve large-scale focusing inversion problems on a standard desktop computer.

The two solvers, GKB and RSVD, that are used within the focusing inversion algorithm for the solution of linear systems are parameter dependent. For a system matrix of size m by n , the algorithms proceed by finding approximate matrices of size m by t , where $t < m$. The size of t in relation to m impacts the efficiency and accuracy of the focusing inversion. The presented results demonstrate that it is sufficient to use $t \approx \text{floor}(m/8)$ for the focusing inversion of large-scale problems. It is more efficient, however, to use the GKB to invert the magnetic data sets and the RSVD for the gravity data sets.

ACKNOWLEDGEMENTS

The authors would like to thank Dr Mark Pilkington for providing data from the Wuskwatim Lake area. Rosemary Renaut acknowledges the support of NSF grant DMS 1913136: ‘Approximate Singular Value Expansions and Solutions of Ill-Posed Problems’. The authors also appreciate the comments of one anonymous reviewer and the editor. Their comments contributed to a significant improvement in the presentation of the results. This paper was written and prepared by first author Rosemary A. Renaut. Second author, Jarom D. Hogue, wrote most of the code for the implementation of the fast Fourier transform and processed the data for the large-scale simulations. The third and fourth authors were involved with the analysis of the results and geophysics interpretation.

REFERENCES

- Blakely, R.J., 1995. *Potential Theory in Gravity and Magnetic Applications*, Cambridge Univ. Press.
- Boulanger, O. & Chouteau, M., 2001. Constraints in 3D gravity inversion, *Geophys. Prospect.*, **49**(2), 265–280.
- Bruun, C.E. & Nielsen, T.B., 2007. Algorithms and software for large-scale geophysical reconstructions, *Master’s thesis*, DTU, DK-2800 Kgs, Technical University of Denmark, Lyngby, Denmark.
- Chan, R.H.-F. & Jin, X.-Q., 2007. *An Introduction to Iterative Toeplitz Solvers*, Society for Industrial and Applied Mathematics, doi:10.1137/1.9780898718850.
- Chen, L. & Liu, L., 2018. Fast and accurate forward modeling of gravity field using prismatic grids, *Geophys. J. Int.*, **216**(2), 1062–1071.
- Cox, L.H., Wilson, G.A. & Zhdanov, M.S., 2010. 3D inversion of airborne electromagnetic data using a moving footprint, *Explor. Geophys.*, **41**(4), 250–259.
- Farquharson, C.G., 2008. Constructing piecewise-constant models in multi-dimensional minimum-structure inversions, *Geophysics*, **73**(1), K1–K9.
- Farquharson, C.G. & Oldenburg, D.W., 2004. A comparison of automatic techniques for estimating the regularization parameter in non-linear inverse problems, *Geophys. J. Int.*, **156**(3), 411–425.
- Golub, G.H. & Van Loan, C.F., 2013. *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins Univ. Press. ISBN 9781421407944.
- Haáz, I.B., 1953. Relations between the potential of the attraction of the mass contained in a finite rectangular prism and its first and second derivatives, *Geophys. Trans. II*, **7**, 57–66.
- Halko, N., Martinsson, P.G. & Tropp, J.A., 2011. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Rev.*, **53**(2), 217–288.
- Hansen, P.C., 2010. *Discrete Inverse Problems*, Society for Industrial and Applied Mathematics, doi:10.1137/1.9780898718836.
- Hogue, J.D., Renaut, R.A. & Vatankhah, S., 2020. A tutorial and open source software for the efficient evaluation of gravity and magnetic kernels, *Comput. Geosci.*, **14**, 104575.
- Last, B.J. & Kubik, K., 1983. Compact gravity inversion, *Geophysics*, **6**, 713–721.
- Lelièvre, P.G. & Oldenburg, D.W., 2006. Magnetic forward modelling and inversion for high susceptibility, *Geophys. J. Int.*, **166**(1), 76–90.
- Li, K., Chen, L.-W., Chen, Q.-R., Dai, S.-K., Zhang, Q.-J., Zhao, D.-D. & Ling, J.-X., 2018. Fast 3D forward modeling of the magnetic field and gradient tensor on an undulated surface, *Appl. Geophys.*, **15**(3), 500–512.
- Li, X. & Chouteau, M., 1998. Three-dimensional gravity modeling in all space, *Surv. Geophys.*, **19**(4), 339–368.
- Li, Y. & Oldenburg, D.W., 1996. 3-D inversion of magnetic data, *Geophysics*, **61**(2), 394–408.
- Li, Y. & Oldenburg, D.W., 1998. 3-D inversion of gravity data, *Geophysics*, **63**(1), 109–119.
- Li, Y. & Oldenburg, D.W., 2003. Fast inversion of large-scale magnetic data using wavelet transforms and a logarithmic barrier method, *Geophys. J. Int.*, **152**(2), 251–265.
- Luiken, N. & van Leeuwen, T., 2020. Comparing RSVD and Krylov methods for linear inverse problems, *Comput. Geosci.*, **137**, 104427.
- Nabighian, M.N. et al., 2005. Historical development of the gravity method in exploration, *Geophysics*, **70**(6), 63ND–89ND.
- Paige, C.C. & Saunders, M.A., 1982. LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Software*, **8**(1), 43–71.
- Pilkington, M., 1997. 3-D magnetic imaging using conjugate gradients, *Geophysics*, **62**(4), 1132–1142.
- Pilkington, M., 2009. 3D magnetic data-space inversion with sparseness constraints, *Geophysics*, **74**(1), L7–L15.
- Portniaguine, O. & Zhdanov, M.S., 1999. Focusing geophysical inversion images, *Geophysics*, **64**(3), 874–887.
- Portniaguine, O. & Zhdanov, M.S., 2002. 3D magnetic inversion with data compression and image focusing, *Geophysics*, **67**(5), 1532–1541.
- Rao, D.B. & Babu, N.R., 1991. A rapid method for three-dimensional modeling of magnetic anomalies, *Geophysics*, **56**(11), 1729–1737.
- Renaut, R.A., Vatankhah, S. & Ardestani, V.E., 2017. Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems, *SIAM J. Sci. Comput.*, **39**, B221–B243.
- Renaut, R.A., Hogue, J.D. & Vatankhah, S., 2020. A fast methodology for large-scale focusing inversion of gravity and magnetic data using the structured model matrix and the 2D fast Fourier transform, available at: <https://arxiv.org/abs/2004.13904>.
- Silva, J.B.C. & Barbosa, V.C.F., 2006. Interactive gravity inversion, *Geophysics*, **71**(1), J1–J9.
- Uieda, L. & Barbosa, V.C.F., 2012. Robust 3D gravity gradient inversion by planting anomalous densities, *Geophysics*, **77**(4), G55–G66.
- Vatankhah, S., Ardestani, V.E. & Renaut, R.A., 2014. Automatic estimation of the regularization parameter in 2D focusing gravity inversion: application of the method to the Safo manganese mine in the northwest of Iran, *J. Geophys. Eng.*, **11**(4), 045001.
- Vatankhah, S., Ardestani, V.E. & Renaut, R.A., 2015. Application of the χ^2 principle and unbiased predictive risk estimator for determining the

- regularization parameter in 3-D focusing gravity inversion, *Geophys. J. Int.*, **200**(1), 265–277.
- Vatankhah, S., Renaut, R.A. & Ardestani, V.E., 2017. 3-D projected ℓ_1 inversion of gravity data using truncated unbiased predictive risk estimator for regularization parameter estimation, *Geophys. J. Int.*, **210**(3), 1872–1887.
- Vatankhah, S., Renaut, R.A. & Ardestani, V.E., 2018. A fast algorithm for regularized focused 3-D inversion of gravity data using the randomized SVD, *Geophysics*, **83**, G25–G34.
- Vatankhah, S., Ardestani, V.E., Niri, S.S., Renaut, R.A. & Kabirzadeh, H., 2019. IGUG: A MATLAB package for 3D inversion of gravity data using graph theory, *Comput. Geosci.*, **128**, 19–29.
- Vatankhah, S., Liu, S., Renaut, R.A., Hu, X. & Baniamerian, J., 2020a. Improving the use of the randomized singular value decomposition for the inversion of gravity and magnetic data, *Geophysics*, G93–G107.
- Vatankhah, S., Renaut, R.A. & Liu, S., 2020b. Research note: A unifying framework for the widely used stabilization of potential field inverse problems, *Geophys. Prospect.*, **68**, 1416–1421.
- Vogel, C., 2002. *Computational Methods for Inverse Problems*, Society for Industrial and Applied Mathematics, doi:10.1137/1.9780898717570.
- Voronin, S., Mikesell, D. & Nolet, G., 2015. Compression approaches for the regularized solutions of linear systems from large-scale inverse problems, *GEM - Int. J. Geomath.*, **6**(2), 251–294.
- Wohlberg, B. & Rodríguez, P., 2007. An iteratively reweighted norm algorithm for minimization of total variation functionals, *IEEE Signal Process. Lett.*, **14**(12), 948–951.
- Xiang, H. & Zou, J., 2013. Regularization with randomized SVD for large-scale discrete inverse problems, *Inverse Prob.*, **29**(8), 085008.
- Zhang, Y. & Wong, Y. S., 2015. BTTB-based numerical schemes for three-dimensional gravity field inversion, *Geophys. J. Int.*, **203**(1), 243–256.
- Zhao, G., Chen, B., Chen, L., Liu, J. & Ren, Z., 2018. High-accuracy 3D Fourier forward modeling of gravity field based on the Gauss-FFT technique, *J. Appl. Geophys.*, **150**, 294–303.

APPENDIX A: MULTIPLICATION USING BTTB STRUCTURE

The multiplication $\mathbf{G}\mathbf{x}$, where $\mathbf{x} \in \mathcal{R}^n$, uses the column block structure of $\mathbf{G}$ which was given in (3). Then, $\mathbf{G}\mathbf{x} = \sum_{r=1}^{n_z} \mathbf{G}^{(r)}\mathbf{x}^{(r)}$ where $\mathbf{x}$ is blocked consistently with $\mathbf{G}$. Each $\mathbf{G}^{(r)}$ has BTTB structure and can be embedded in a circulant matrix in order to evaluate $\mathbf{G}^{(r)}\mathbf{x}^{(r)}$ using the 2DFFT (Vogel 2002). The first column of the circulant extension is reshaped into $\mathbf{T} \in \mathcal{R}^{(s_x+n_x-1) \times (s_y+n_y-1)}$, and $\mathbf{x}^{(r)}$ is reshaped and embedded into $\mathbf{W} \in \mathcal{R}^{(s_x+n_x-1) \times (s_y+n_y-1)}$ (Hogue *et al.* 2019). Assume that the 2DFFT of $\mathbf{T}$ is precomputed and that $\cdot *$ represents element-wise multiplication. Then, $\mathbf{G}^{(r)}\mathbf{x}^{(r)}$ is extracted from $\text{ifft2}(\text{fft2}(\mathbf{T}) \cdot \text{fft2}(\mathbf{W}))$, with cost

$$\text{Cost}_{\mathbf{G}^{(r)}\mathbf{x}^{(r)}} = \text{Cost}_{\text{fft2}(\mathbf{W})} + \text{Cost}_{\cdot *} + \text{Cost}_{\text{ifft2}(\cdot)}. \quad (\text{A1})$$

The 2DFFT of $\mathbf{W}$ is computed as $\text{1DFFT}((\text{1DFFT}(\mathbf{W}))^T)^T$, where the 1DFFT is applied to each column of the array independently. Using

the cost of a 1DFFT as $(n/2)\log_2(n)$ for an n -length vector, Vogel (2002), this gives, using $n_r \approx m$ except when the padding is large,

$$\begin{aligned} \text{Cost}_{\text{fft2}(\mathbf{W})} &\approx 2n_y(n_x \log_2(2n_x)) + 2n_x(n_y \log_2(2n_y)) \\ &= 2m(\log_2(2n_x)) + \log_2(2n_y) = 2m \log_2(4m). \end{aligned}$$

The element-wise complex multiplication in (A1) is for a reshaped vector of size $(s_x + n_x - 1)(s_y + n_y - 1) \approx 4m$, and each complex multiplication requires 6 flops. The inverse 2DFFT requires approximately the same number of operations as the forward 2DFFT. Hence

$$\text{Cost}_{\mathbf{G}^{(r)}\mathbf{x}^{(r)}} \approx 4m \log_2(4m) + 24m, \quad (\text{A2})$$

and

$$\begin{aligned} \text{Cost}_{\mathbf{G}\mathbf{x}} &\approx 4mn_z \log_2(4m) + 24mn_z + (m - 1)n_z \\ &\approx 4n \log_2(4m) + 25n + LOT, \end{aligned} \quad (\text{A3})$$

where the first term is for the multiplication and the second for the summation over the n_z vectors of length m . Immediately, the dominant cost for obtaining $\mathbf{G}\mathbf{x}$, for $\mathbf{x} \in \mathcal{R}^{n \times t_p}$, ignoring all but third order terms is

$$\text{Cost}_{\mathbf{G}\mathbf{x}} \approx 4t_p n \log_2(4m) + LOT. \quad (\text{A4})$$

The derivation of the computation, and the cost, for obtaining $\mathbf{G}^T \mathbf{y}$ for $\mathbf{y} \in \mathcal{R}^m$ follows similarly, noting that $\mathbf{G}^T \mathbf{y} = [\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \dots, \mathbf{G}^{(n_z)}]^T \mathbf{y}$ requires the computation of $(\mathbf{G}^{(r)})^T \mathbf{y}$ for each r and that no summation is required as in (A3). Hence $\text{Cost}_{\mathbf{G}^T \mathbf{y}} \approx 4n \log_2(4m)$ and $\text{Cost}_{\mathbf{G}\mathbf{y}} \approx 4t_p n \log_2(4m)$. Noting that $\mathbf{X}^T \mathbf{G}^T = (\mathbf{G}\mathbf{X})^T$ and $\mathbf{Y}^T \mathbf{G} = (\mathbf{G}^T \mathbf{Y})^T$, the computations and computational costs are immediately obtained from those of $\mathbf{G}\mathbf{x}$ and $\mathbf{G}^T \mathbf{y}$, respectively.

APPENDIX B: SUPPORTING NUMERICAL RESULTS OF SIMULATIONS

Supporting results illustrated as figures in Sections 3.3.1–3.3.3 are reported in a set of tables, with captions describing the details. Table B1 reports the timing for one iteration of the inversion algorithm using both GKB and RSVD algorithms for magnetic data inversion, comparing timings using matrix $\mathbf{G}$ directly and the 2DFFT. The time to convergence for the algorithms is given in Table B1 for both gravity and magnetic data sets for domains without padding. Tables B3 and B4 give the details of the number of iteration steps to convergence K and the resulting relative errors, RE, for the timing results of Table B2. Results for $m = 6000$ are provided in Renaut *et al.* (2020).

Table B1. Timing results in seconds for one step of the inversion algorithm applied to magnetic potential field data for the simulations described in Table 2 without padding and with padding (indicated by P), and for problem sizes up to $m = 18\,375$, excluding $m = 6000$. $t_p = \text{floor}(1.05t)$ is the size of the oversampled projected space for GKB and RSVD implementations. The columns under Direct use of **G** do not use the 2DFFT. These results are illustrated in Fig. 4.

magnetic			WITH 2DFFT				Direct use of G			
m	t	t_p	GKB	RSVD	PGKB	PRSVD	GKB	RSVD	PGKB	PRSVD
9375	234	245	8	13	7	11	120	12	143	14
9375	375	393	14	20	14	18	193	18	232	22
9375	468	491	18	26	18	24	244	22	294	27
9375	1171	1229	72	70	77	68	633	55	765	66
9375	1562	1640	115	96	125	90	862	74	1044	89
9375	2343	2460	230	151	257	144	1347	118	1633	142
9375	3125	3281	389	215	435	208	1869	169	2278	211
13500	337	353	19	29	16	20	430	440	532	1597
13500	540	567	36	48	32	35	689	1996	831	2985
13500	675	708	49	60	46	45	867	977	1050	2821
13500	1687	1771	213	164	224	127	2255	465	2739	1301
13500	2250	2362	351	227	382	182	3068	1235	3738	2425
13500	3375	3543	733	376	818	315	4798	1279	5890	2834
13500	4500	4725	1259	542	1413	475	6666	2108	61661	3487
18375	459	481	41	56	54	72	NA	NA	NA	NA
18375	735	771	84	94	104	117	NA	NA	NA	NA
18375	918	963	117	121	145	150	NA	NA	NA	NA
18375	2296	2410	554	346	674	433	NA	NA	NA	NA
18375	3062	3215	944	496	1136	601	NA	NA	NA	NA
18375	4593	4822	1999	854	2409	1061	NA	NA	NA	NA
18375	5000	5250	2317	949	2868	1192	NA	NA	NA	NA

Table B2. Timing results to convergence for inversion of gravity and magnetic potential field data for the simulations described in Table 2 without padding, for problem sizes up to $m = 18\,375$, excluding $m = 6000$. Entries with * indicate that the algorithm did not converge to the required tolerance. In the last two columns the relative costs of GKB as compared to RSVD. Relative errors and number of iterations are presented in Tables B3 and B4, for gravity and magnetic data, respectively.

			gravity		magnetic		Cost2DFFT _{GKB} /Cost2DFFT _{RSVD}	
m	t	t_p	GKB	RSVD	GKB	RSVD	gravity	magnetic
9375	234	245	265	166	152	509*	1.60	0.30
9375	375	393	411	259	174	811*	1.59	0.21
9375	468	491	342	325	199	1014*	1.05	0.20
9375	1171	1229	1064	835	626	2582*	1.27	0.24
9375	1562	1640	1235	997	948	2121	1.24	0.45
9375	2343	2460	1899	1492	1728	2126	1.27	0.81
9375	3125	3281	2918	2052	2915	2971	1.42	0.98
13500	337	353	595	296	246	923*	2.01	0.27
13500	540	567	671	424	347	1514*	1.58	0.23
13500	675	708	802	527	413	1877*	1.52	0.22
13500	1687	1771	2704	1385	1077	2581	1.95	0.42
13500	2250	2362	2518	1597	1608	2937	1.58	0.55
13500	3375	3543	4308	2483	3071	4142	1.73	0.74
13500	4500	4725	6925	3429	6699	5109	2.02	1.31
18375	459	481	1427	679	594	2157*	2.10	0.28
18375	735	771	1642	1104	1070	3524*	1.49	0.30
18375	918	963	2084	1218	1026	4413*	1.71	0.23
18375	2296	2410	5732	3311	3809	6618	1.73	0.58
18375	3062	3215	6959	4490	5639	8469	1.55	0.67
18375	4593	4822	12347	6979	10979	12949	1.77	0.85
18375	5000	5250	13975	7711	12544	13239	1.81	0.95

Table B3. Results for inversion of gravity potential field data for the simulations described in Table 2 without padding and with padding and for problem sizes up to $m = 18\,375$, with $K_{\max} = 25$, excluding $m = 6000$. t_p is the size of the projected space for GKB and RSVD implementations. Reported are the number of iterations to convergence, K , for convergence as defined by (17) and the calculated relative errors $RE^{(K)}$. Entries with * indicate that the algorithm did not converge to the required tolerance.

gravity			GKB		RSVD		PGKB		PRSDV	
m	t	t_p	K	RE	K	RE	K	RE	K	RE
9375	234	245	21	1.05	8	0.49	21	1.05	9	0.51
9375	375	393	19	1.01	8	0.50	21	1.03	9	0.53
9375	468	491	12	0.82	8	0.50	13	0.84	8	0.53
9375	1171	1229	12	0.78	8	0.53	11	0.79	8	0.57
9375	1562	1640	9	0.66	7	0.53	9	0.68	8	0.58
9375	2343	2460	8	0.65	7	0.55	8	0.67	8	0.59
9375	3125	3281	8	0.64	7	0.57	8	0.66	7	0.60
13500	337	353	24	1.03	8	0.56	23	1.03	8	0.60
13500	540	567	15	0.90	7	0.58	15	0.93	7	0.61
13500	675	708	14	0.89	7	0.58	14	0.91	7	0.62
13500	1687	1771	13	0.85	7	0.61	12	0.85	6	0.64
13500	2250	2362	8	0.70	6	0.62	8	0.71	6	0.64
13500	3375	3543	7	0.69	6	0.63	8	0.71	6	0.64
13500	4500	4725	7	0.69	6	0.63	8	0.70	6	0.64
18375	459	481	24	1.07	8	0.56	*	1.08	8	0.59
18375	735	771	16	0.95	8	0.57	16	0.95	8	0.59
18375	918	963	15	0.93	7	0.58	15	0.94	7	0.60
18375	2296	2410	11	0.75	7	0.60	11	0.75	7	0.60
18375	3062	3215	9	0.72	7	0.60	10	0.73	7	0.61
18375	4593	4822	8	0.70	7	0.61	9	0.71	7	0.61
18375	5000	5250	8	0.70	7	0.61	9	0.71	7	0.61

Table B4. Results for inversion of magnetic potential field data for the simulations described in Table 2 without padding and with padding and for problem sizes up to $m = 18\,375$, excluding $m = 6000$, with $K_{\max} = 25$. t_p is the size of the projected space for GKB and RSVD implementations. Reported are the number of iterations to convergence, K , for convergence as defined by (17) and the calculated relative errors $RE^{(K)}$. Entries with * indicate that the algorithm did not converge to the required tolerance.

magnetic			GKB		RSVD		PGKB		PRSDV	
m	t	t_p	K	RE	K	RE	K	RE	K	RE
9375	234	245	12	0.81	*	0.82	6	0.71	*	0.89
9375	375	393	8	0.72	*	0.78	6	0.69	*	0.82
9375	468	491	7	0.70	*	0.77	6	0.69	*	0.80
9375	1171	1229	7	0.66	*	0.70	6	0.66	*	0.71
9375	1562	1640	7	0.66	15	0.70	6	0.66	12	0.71
9375	2343	2460	7	0.65	10	0.67	6	0.66	9	0.68
9375	3125	3281	8	0.65	10	0.67	8	0.66	9	0.68
13500	337	353	10	0.74	*	0.73	5	0.67	*	0.77
13500	540	567	8	0.69	*	0.69	5	0.67	*	0.73
13500	675	708	7	0.67	*	0.68	5	0.67	*	0.71
13500	1687	1771	5	0.64	13	0.66	5	0.65	10	0.68
13500	2250	2362	5	0.64	11	0.65	5	0.66	10	0.68
13500	3375	3543	5	0.64	10	0.64	5	0.67	9	0.66
13500	4500	4725	7	0.62	9	0.63	5	0.67	9	0.66
18375	459	481	11	0.78	*	0.80	6	0.69	*	0.80
18375	735	771	10	0.74	*	0.75	6	0.69	*	0.76
18375	918	963	7	0.69	*	0.73	6	0.69	*	0.75
18375	2296	2410	7	0.67	14	0.70	5	0.72	12	0.72
18375	3062	3215	7	0.68	13	0.70	6	0.69	11	0.71
18375	4593	4822	7	0.68	13	0.69	6	0.70	11	0.72
18375	5000	5250	7	0.68	12	0.68	6	0.70	11	0.72