Total variation regularization of the 3-D gravity inverse problem using a randomized generalized singular value decomposition

Saeed Vatankhah,1 Rosemary A. Renaut2 and Vahid E. Ardestani1

1Institute of Geophysics, University of Tehran, Tehran 1417466191, Iran. E-mail: svatan@ut.ac.ir
2School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85281, USA

Accepted 2018 January 16. Received 2017 December 4; in original form 2017 September 23

SUMMARY
We present a fast algorithm for the total variation regularization of the 3-D gravity inverse problem. Through imposition of the total variation regularization, subsurface structures presenting with sharp discontinuities are preserved better than when using a conventional minimum-structure inversion. The associated problem formulation for the regularization is nonlinear but can be solved using an iteratively reweighted least-squares algorithm. For small-scale problems the regularized least-squares problem at each iteration can be solved using the generalized singular value decomposition. This is not feasible for large-scale, or even moderate-scale, problems. Instead we introduce the use of a randomized generalized singular value decomposition in order to reduce the dimensions of the problem and provide an effective and efficient solution technique. For further efficiency an alternating direction algorithm is used to implement the total variation weighting operator within the iteratively reweighted least-squares algorithm. Presented results for synthetic examples demonstrate that the novel randomized decomposition provides good accuracy for reduced computational and memory demands as compared to use of classical approaches.

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

1 INTRODUCTION
Regularization is imposed in order to find acceptable solutions to the ill-posed and non-unique problem of gravity inversion. Most current regularization techniques minimize a global objective function that consists of a data misfit term and a stabilization term; (Li & Oldenburg 1998; Portniaguine & Zhdanov 1999; Boulanger & Chouteau 2001; Vatankhah et al. 2017a). Generally for potential field inversion the data misfit is measured as a weighted \(L_2\)-norm of the difference between the observed and predicted data of the reconstructed model (Pilkington 2009). Stabilization aims to both counteract the ill-posedness of the problem so that changes in the model parameters due to small changes in the observed data are controlled, (Fedi et al. 2005; Paoletti et al. 2014), and to impose realistic characteristics on the reconstructed model. Many forms of robust and reliable stabilizations have been used by the geophysical community. The minimum-structure inversion based on a \(L_2\)-norm measure of the gradient of the model parameters was commonly used in potential field inversion, (Li & Oldenburg 1996, 1998). Although the methodology was used successfully in many research papers, it is not able to recover a non-smooth density distribution such as expected by presence of mineral bodies, dikes, etc. (Farquharson 2008). Sparse inversion based on a minimum volume constraint, \(L_0\)-norm of the model parameters, was introduced by Last & Kubik (1983) and then was developed and used commonly for reconstructing blocky images of the subsurface (Portniaguine & Zhdanov 1999; Boulanger & Chouteau 2001; Ajo-Franklin et al. 2007; Vatankhah et al. 2015). On the other hand, the \(L_1\) and Cauchy norms for the model parameters can be used to reconstruct sparse and compact solutions (Pilkington 2009; Vatankhah et al. 2017a). Alternatively, total variation (TV) regularization based on a \(L_1\) norm of the gradient of the model parameters preserves edges in the model, and provides a reconstruction of the model that is blocky and non-smooth (Bertete-Aguirre et al. 2002; Farquharson 2008). Therefore, the selection of the stabilization to be used for gravity inversion depends on an anticipated true representation of the subsurface model and is problem specific. The most important point here is that when the dimension of the inverse problem increases, efficient and practical algorithms for all aforementioned regularization formulations are desired. For example the following methods were all used for solution of the large-scale problem; (1) the symmetry of the gravity kernel, (Boulanger & Chouteau 2001); (2) data space inversion, (Pilkington 2009); and (3) the projection of the problem onto a smaller subspace, (Oldenburg et al. 1993; Vatankhah et al. 2017a). Here the focus is on the development of a new randomized algorithm for the 3-D linear inversion of gravity data with TV stabilization. The approach is very effective when the underlying inverse problem is of a moderate scale.
Regularization using TV stabilization yields an objective function which is nonlinear in the solution. We therefore use the iteratively reweighted generalized Tikhonov least-squares (IRLS) formulation to solve the nonlinear problem, (Portnaguine & Zhdanov 1999). This presents no difficulty for small-scale problems; the generalized singular value decomposition (GSVD) can be used to compute the mutual decomposition of the model and stabilization matrices from which the solution is immediate at each iteration. Furthermore, the use of the GSVD provides a convenient form for the regularization parameter-choice rules, (Xiang & Zou 2013; Chung & Palmer 2015). The approach using the GSVD, however, is not practical when the dimension of the problem increases, neither in terms of computational costs nor memory demands. Randomized algorithms compute low-rank matrix approximations with reduced memory and computational costs (Halko et al. 2011; Xiang & Zou 2013; Voronin et al. 2015; Wei et al. 2016). Random sampling is used to construct a low-dimensional subspace that captures the dominant spectral properties of the original matrix and then, as developed by Wei et al. (2016), a GSVD can be applied to regularize for this dominant spectral space. We demonstrate that applying this randomized GSVD (RGSVD) methodology at each step of the iterative TV regularization provides a fast and efficient algorithm for 3-D gravity inversion that inherits all the properties of TV inversion for small-scale problems. We also show how the weighting matrix at each iteration can be determined and how an alternating direction algorithm can also reduce the memory overhead associated with calculating the GSVD of the projected subproblem at each step of the IRLS.

2 INVERSION METHODOLOGY

Suppose the subsurface is divided into a large number of cells of fixed size but unknown density (Li & Oldenburg 1998; Boulanger & Chouteau 2001). Unknown densities of the cells are stacked in vector \( \mathbf{m} \in \mathbb{R}^m \) and measured data on the surface are stacked in \( \mathbf{d}_{\text{obs}} \in \mathbb{R}^n \), and are related to the densities via the linear relationship

\[
\mathbf{d}_{\text{obs}} = \mathbf{Gm}.
\]

(1)

for forward model matrix \( \mathbf{G} \in \mathbb{R}^{n \times m} \) (\( m \ll n \)). The aim is to find an acceptable model for the densities that predicts the observed data at the noise level. For gravity inversion, eq. (1) is modified through inclusion of an estimated prior model, \( \mathbf{m}_{\text{apr}} \), yielding (Li & Oldenburg 1998)

\[
\mathbf{d}_{\text{obs}} - \mathbf{Gm}_{\text{apr}} = \mathbf{Gm} - \mathbf{Gm}_{\text{apr}}.
\]

(2)

First, eq. (1) is replaced by \( G\mathbf{y} = \mathbf{r} \) using \( \mathbf{r} = \mathbf{d}_{\text{obs}} - \mathbf{Gm}_{\text{apr}} \) and \( \mathbf{y} = \mathbf{m} - \mathbf{m}_{\text{apr}} \). Then a diagonal depth weighting matrix \( W_{\text{depth}} = 1/(z_j + \xi)^\beta \) (Boulanger & Chouteau 2001) is incorporated in the model via \( \tilde{G} = GW_{\text{depth}}^{1/2} \) yielding

\[
\mathbf{r} = \tilde{G}\mathbf{h}.
\]

(3)

for \( \mathbf{h} = W_{\text{depth}}^\beta \mathbf{y} \). In \( W_{\text{depth}} \), \( z_j \) is the mean depth of cell \( j \), \( \beta \) is a weighting parameter usually chosen based on the data type and dimension of the problem and \( \xi \) is very small positive number (Boulanger & Chouteau 2001). Due to the ill-posedness of the problem, eq. (3) cannot be solved directly but must be stabilized in order to provide a practically acceptable solution. Using the L2-TV regularization methodology \( h \) is obtained by minimizing the objective function (Wohlberg & Rodrigue 2007),

\[
P^\nu(h) = \| W_d(\tilde{G}\mathbf{h} - \mathbf{r}) \|^2_2 + \alpha^2 \| \nabla \mathbf{h} \|^2_2,
\]

(4)
in which diagonal weighting matrix \( W_d \) approximates the square root of the inverse covariance matrix for the independent noise in the data. Specifically \( (W_d)_{ii} = 1/\eta_i \) in which \( \eta_i \) is the standard deviation of the noise for the \( i \)th datum. Regularization parameter \( \alpha \) provides a trade-off between the weighted data misfit and the stabilization. The absolute value of the gradient vector, \( \| \nabla \mathbf{h} \| \), is equal to \( \sqrt{(D_x\mathbf{h})^2 + (D_y\mathbf{h})^2 + (D_z\mathbf{h})^2} \) in which \( D_x, D_y \), and \( D_z \) are the discrete derivative operators in \( x, y \), and \( z \)-directions, respectively. The derivatives at the centres of the cells are approximated to low order using forward differences with backward differencing at the boundary points, Li & Oldenburg (2000), yielding matrices \( D_x, D_y \) and \( D_z \) which are square of size \( n \times n \). Although (4) is a convex optimization problem with a unique solution, the TV term is not differentiable everywhere with respect to \( \mathbf{h} \) and to find a solution it is helpful to rewrite the stabilization term using a weighted \( L_2 \)-norm, following Wohlberg & Rodriguez (2007). Given vectors \( \mathbf{x}, \mathbf{y} \) and \( \mathbf{z} \) and a diagonal matrix \( W \) with entries \( w_i \) we have

\[
\left\| \begin{pmatrix} W_{RR} & 0 & 0 \\ 0 & W_{RR} & 0 \\ 0 & 0 & W_{RR} \end{pmatrix} \begin{pmatrix} \xi \\ \chi \\ \psi \end{pmatrix} \right\|_2 = \sum_r \left( w_r \xi_r^2 + w_r \chi_r^2 + w_r \psi_r^2 \right)
\]

\[
= \sum_r w_r^2 \left( \sqrt{\xi_r^2 + \chi_r^2 + \psi_r^2} \right)^2.
\]

(5)

Setting \( w_i = (\xi_i^2 + \chi_i^2 + \psi_i^2)^{-1/4} \) we have

\[
\left\| \begin{pmatrix} W_{RR} & 0 & 0 \\ 0 & W_{RR} & 0 \\ 0 & 0 & W_{RR} \end{pmatrix} \begin{pmatrix} \xi \\ \chi \\ \psi \end{pmatrix} \right\|_2 = \sum_r \left( \sqrt{\xi_r^2 + \chi_r^2 + \psi_r^2} \right)^2.
\]

(6)

Now with \( \mathbf{h} = D_x \mathbf{h}, \chi = D_y \mathbf{h}, \psi = D_z \mathbf{h} \) and \( W^{(k)} \) defined to have entries \( w^{(k)} \) calculated for \( \nabla \mathbf{h} \) at iteration \( k \) given by \( w^{(k)} = ((D_x\mathbf{h}^{(k-1)})^2 + (D_y\mathbf{h}^{(k-1)})^2 + (D_z\mathbf{h}^{(k-1)})^2 + \epsilon^2)^{-1/4} \), the TV stabilizer is approximated via

\[
\| \nabla \mathbf{h} \|^2_1 = \left\| (D_x\mathbf{h})^2 + (D_y\mathbf{h})^2 + (D_z\mathbf{h})^2 \right\|_1 \approx \| W^{(k)} D\mathbf{h} \|^2_2.
\]

(7)

for derivative operator \( D = [D_x; D_y; D_z] \). Here, \( 0 < \epsilon \ll 1 \) is added to avoid the possibility of division by zero, and superscript \( k \) indicates that matrix \( W_k \) is updated using the model parameters of the previous iteration. Hence (4) is rewritten as a general Tikhonov functional

\[
P^\nu(h) = \| W_d(\tilde{G}\mathbf{h} - \mathbf{r}) \|^2_2 + \alpha^2 \| \tilde{D}\mathbf{h} \|^2_2,
\]

(8)

for which the minimum is explicitly expressible as

\[
\mathbf{h} = (\tilde{G}^T \tilde{G} + \alpha^2 \tilde{D}^T \tilde{D})^{-1} \tilde{G}^T \tilde{r}, \tilde{G} = W_d \tilde{G} \text{ and } \tilde{r} = W_d \mathbf{r}.
\]

(9)

The model update is then

\[
\mathbf{m}(\alpha) = \mathbf{m}_{\text{apr}} + W_{\text{depth}}^{-1}\mathbf{h}(\alpha).
\]

(10)

While the Tikhonov function can be replaced by a standard form Tikhonov function, that is, with regularization term \( \| \mathbf{h} \|^2_2 \), when \( \tilde{D} \) is easily invertible, for example, when diagonal, the form of \( \tilde{D} \) in this case makes that transformation prohibitive for cost and we must solve using the general form.

We show the iterative process for the solution in Algorithm 2. It should be noted that the iteration process terminates when solution satisfies the noise level or a predefined maximum number of iterations is reached (Boulanger & Chouteau 2001). Furthermore, the density limits \( \rho_{\text{min}}, \rho_{\text{max}} \) are imposed at each iteration. If at any iteration a density value falls outside these predefined density bounds,
the value is projected back to the nearest bound value (Boulanger & Chouteau 2001).

For small-scale problems in which the dimensions of \( \hat{G} \), and consequently \( \hat{D} \), are small, the solution \( \hat{h}(\alpha) \) is found at minimal cost using the GSVD of the matrix pair \([ \hat{G}, \hat{D} ]\) as it is shown in Appendix A (Aster et al. 2013; Vatankhah et al. 2014). Furthermore, given the GSVD the regularization parameter may be estimated cheaply using standard parameter-choice techniques (Xiang & Zou 2013; Chung & Palmer 2015). But for large-scale problems it is not practical to calculate the GSVD at each iteration, both with respect to computational cost and memory demands. Instead the size of the original large problem can be reduced greatly using a randomization technique which provides the GSVD in a more feasible and efficient manner. The solution of reduced system still is a good approximation of the original system (Halko et al. 2011; Xiang & Zou 2013; Voronin et al. 2015; Xiang & Zou 2015; Wei et al. 2016). Here, we use the Randomized GSVD (RGSVD) algorithm developed by Wei et al. (2016) for underdetermined problems in which the mutual decomposition of the matrix pair \([ \hat{G}, \hat{D} ]\) is approximated by

\[
\hat{G} \approx U \Lambda Z, \quad \hat{D} = VMZ, \quad U \in \mathbb{R}^{m \times q}, \quad V \in \mathbb{R}^{n \times q}, \quad \Lambda \in \mathbb{R}^{q \times q}, \quad M \in \mathbb{R}^{q \times q}, \quad Z \in \mathbb{R}^{q \times n},
\]

Algorithm 1 RGSVD algorithm. Given matrices \( \hat{G} \in \mathbb{R}^{m \times n} (m \leq n) \) and \( \hat{D} \in \mathbb{R}^{n \times n} \), a target matrix rank \( q \) and a small constant oversampling parameter \( p \) satisfying \( q + p = l \leq m \), compute an approximate GSVD of \([ \hat{G}, \hat{D} ]\): \( \hat{G} \approx U \Lambda Z, \hat{D} = VMZ \) with \( U \in \mathbb{R}^{m \times q}, V \in \mathbb{R}^{n \times q}, \Lambda \in \mathbb{R}^{q \times q}, M \in \mathbb{R}^{q \times q} \) and \( Z \in \mathbb{R}^{q \times n} \).

1: Generate a Gaussian random matrix \( \Omega \in \mathbb{R}^{l \times m} \).
2: Form the matrix \( Y = \Omega \hat{G} \in \mathbb{R}^{l \times n} \).
3: Compute orthonormal matrix \( Q \in \mathbb{R}^{n \times l} \) via QR factorization \( Y^T Q = QR \).
4: Set \( Q = Q(:,1 : q) \) and form the matrices \( B_1 = \hat{G} Q \in \mathbb{R}^{m \times q} \) and \( B_2 = \hat{D} Q \in \mathbb{R}^{n \times q} \).
5: Compute the GSVD of \([ B_1, B_2 ]\): \( [ B_1 \quad B_2 ] = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} \Lambda & M \end{bmatrix} X^T \),
using \([ U, V, X, \Lambda, M ] = \text{gsvd}(B_1, B_2, 0) \).
6: Form the matrix \( Z = X^T Q \in \mathbb{R}^{q \times n} \).

The steps of the algorithm are given in Algorithm 1. Steps 1 to 3 are used to form matrix \( Q \) which approximates the range of \( \hat{G} \). At step 4, \( \hat{G} \) is projected into a lower dimensional matrix \( B_1 \), for which \( B_1 \) provides information on the range of \( \hat{G} \). The same projection is applied to matrix \( \hat{D} \). In step 5, an economical size GSVD is computed for the matrix pair \([ B_1, B_2 ]\). Parameter \( q \) balances the accuracy and efficiency of the Algorithm 1 and determines the dimension of the subspace for the projected problem. When \( q \) is small, this methodology is very effective and leads to a fast GSVD computation. Simultaneously, the parameter \( q \) has to be selected large enough to capture the dominant spectral properties of the original problem with the aim that the solution obtained using the RGSVD is close to the solution that would be obtained using the GSVD.

For any GSVD decomposition (11), including that obtained via Algorithm 1, \( h(\alpha) \) of (9) is given by

\[
h(\alpha) = (Z^T \Lambda^T U^T U \Lambda Z + \alpha^2 Z^T M^T V^T VMZ)^{-1} Z^T \Lambda^T U^T \tilde{F},
\]

which simplifies to (Aster et al. 2013; Vatankhah et al. 2014)

\[
h(\alpha) = \sum_{i=1}^{q} \frac{\gamma_i^2 - \alpha^2}{\gamma_i^2} \tilde{F}_i.
\]

Here \( \gamma_i \) is the \( i \)-th generalized singular value, see Appendix A, and \( Z \) is the Moore-Penrose inverse of \( \hat{Z} \) (Wei et al. 2016). Incorporating this single step of the RGSVD within the TV algorithm yields the iteratively reweighted TVRGSVD algorithm given in Algorithm 2. Note here that steps 1–3 and the calculation of \( B_1 \) in step 4 in Algorithm 1 are the same for all TV iterations and thus outside the loop in Algorithm 2. At each iteration \( k \), the matrices \( W, \hat{D} \) and \( \hat{B}_2 \) are updated, and the GSVD is determined for the matrix pair \([ B_1, B_2 \]) of (9). We should note here that it is immediate to use the minimum gradient support (MGS) stabilizer introduced by Portniaguine & Zhdanov (1999) in Algorithm 2 via replacing \((-1/4)W_R \) with \((-1/2)\) and keeping all other parameters fixed.

Algorithm 2 Iterative TV inversion algorithm using randomized GSVD

**Input:** \( d_{obs}, m_{qpr}, G, W_d, D_x, D_y, D_z, q, \epsilon > 0, \rho_{\text{min}}, \rho_{\text{max}}, K_{\text{max}}, \beta = 1 \)
1: Initialize \( m^{(0)} = m_{qpr}, W^{(1)} = I, D = [D_x \quad D_y \quad D_z], \hat{D}^{(1)} = D, k = 1 \)
2: Calculate \( \hat{G} = W_d G_{\text{depth}}, \tilde{F}^{(1)} = W_d (d_{obs} - G m_{qpr}) \)
3: Generate a Gaussian random matrix \( \Omega \in \mathbb{R}^{l \times m} \).
4: Compute matrix \( Y = \Omega \hat{G} \in \mathbb{R}^{l \times n} \).
5: Compute orthonormal matrix \( Q \in \mathbb{R}^{n \times l} \) via QR factorization \( Y^T Q = QR \).
6: Set \( Q = Q(:, 1 : q) \) and form the matrix \( B_1 = \hat{G} Q \in \mathbb{R}^{m \times q} \)
7: while Not converged, noise level not satisfied, and \( k < K_{\text{max}} \)
8: Form the matrix \( B_2 = \hat{D}^{(k)} Q \in \mathbb{R}^{n \times q} \).
9: Compute the GSVD of \([ B_1, B_2 ]\): \( [ U, V, X, \Lambda, M ] = \text{gsvd}(B_1, B_2, 0) \).
10: Form the matrix \( Z = X^T Q \in \mathbb{R}^{q \times n} \).
11: Estimate \( \alpha^{(k)} \) using (15)
12: Set \( h^{(k)} = \sum_{i=1}^{q} \frac{\gamma_i^2}{\gamma_i^2 - \epsilon} \tilde{F}_i \),
13: Set \( m^{(k)} = m^{(k-1)} - (W_{\text{gpr}})^{-1} h^{(k)} \),
14: Impose constraint conditions on \( m^{(k)} \) to force \( \rho_{\text{min}} \leq m^{(k)} \leq \rho_{\text{max}} \).
15: Test convergence and exit loop if converged.
16: Calculate \( W^{(k+1)}_R = \text{diag}((D, h^{(k)})^2 + (D, h^{(k)})^2 + (D, h^{(k)})^2 + \epsilon)^{-1/2} \) and set \( W^{(k+1)}_{R_{\text{gpr}}} = \text{diag}(W^{(k+1)}_R, W^{(k+1)}_{\hat{D}^{(k+1)}}) \).
17: Set \( \tilde{F}^{(k+1)} = W_d (d_{obs} - G m^{(k)}), \hat{D}^{(k+1)} = W^{(k+1)}_R D \).
18: \( k = k + 1 \)
end while

**Output:** Solution \( \rho = m^{(k)} \), \( K = k \).

The dominant costs of finding solution (13) for each step of Algorithm 2 depend on the specifics of the algorithm used to estimate the GSVD. Using the RGSVD the cost is determined by the use of Algorithm 1 and the cost for finding \( Z \) needed in (13). These costs are summarized in terms of estimates of the number of floating point operations (flops) in Table 1.
Table 1. Computational cost of each step of Algorithm 1 based on approximate count of flops.

<table>
<thead>
<tr>
<th>Step</th>
<th>Computational Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2^{2n}$</td>
</tr>
<tr>
<td>2</td>
<td>$2^{2n}$ and $6qn^2$</td>
</tr>
<tr>
<td>3</td>
<td>$8q^2(p + m) + (118/3)q^3$</td>
</tr>
<tr>
<td>4</td>
<td>$2q^2n$</td>
</tr>
<tr>
<td>5</td>
<td>$6q^2 + 20q^3$</td>
</tr>
</tbody>
</table>

Figure 1. Estimates of total number of flops using RGSVD and GSVD algorithms for two choices for $(m, n)$ with $n \gg m$.

(a) $(m, n) = (900, 9000)$  
(b) $(m, n) = (6000, 60000)$

Figure 2. A model that consists of two dikes dipping in opposite directions. (a) Cross-section at northing = 725 m; (b) Plane-section at depth = 100 m.

Figure 3. The gravity anomaly produced by the model shown in Fig. 2 and contaminated by Gaussian noise.

seen by using the flop estimates to compare RGSVD and GSVD for examples of $m, n$ and choices for $q$ for RGSVD. The RGSVD offers a significant reduction in the computational cost, as illustrated in Figs 1(a) and (b). In these plots we give the estimates in each case for using either $D$ with $p = 3n$ or $p = n$, and for the major terms $6qn^2$ and $(208/3)q^3$, in each case. To compare a moderate- and a large-scale problem we illustrate for pairs $(m, n) = (900, 9000)$ and $(6000, 60000)$, respectively. These results show that the major terms are sufficient to determine the relative impact of using RGSVD in place of GSVD; RGSVD will be more efficient for $q/(12n) \ll 1$. We will show for simulated data that this comes without loss of accuracy.

2.0.1 An Alternating Direction algorithm

Step 9 of Algorithm 2 requires the economy GSVD decomposition in which matrix $B_k^{(3)}$ is of size $3n \times q$. For a large-scale problem the computational cost and memory demands with the calculation of the GSVD limit the size of the problem that can be solved. We therefore turn to an alternative approach for large-scale 3-D problems and adopt the use of an Alternating Direction (AD) strategy in which to handle the large-scale problem requiring derivatives in greater than two dimensions we split the problem into pieces handling each direction one after the other. This is a technique that has been in the literature for some time for handling the solution of large-scale partial differential equations, most notably through the alternating direction implicit method (Peaceman & Rachford 1955). Matrix $D_k^{(4)}$ generated via steps 16 and 17 of Algorithm 2 can be changed without changing the other steps of the algorithm. For the AD algorithm we may therefore alternate over...
Table 2. Parameters and results for Algorithm 2 applied to the small-scale dipping dikes model. The number of iterations is $K$, $\alpha^{(K)}$ is the final regularization parameter, $\text{RE}^{(K)} = \frac{\|m_{\text{exact}} - m^{(K)}\|_2}{\|m_{\text{exact}}\|_2}$ is the relative error of the reconstructed model at iteration $K$. The final $\chi^2_{\text{computed}}$ is also reported.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Results</th>
<th>$\text{Input Parameters}$</th>
<th>$\text{Results}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_{\text{ap}}$</td>
<td>$\rho_{\text{min}} (\text{g cm}^{-3})$</td>
<td>$\rho_{\text{max}} (\text{g cm}^{-3})$</td>
<td>$K_{\text{max}}$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>300</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>$\neq$ 0</td>
<td>0</td>
<td>1</td>
<td>200</td>
</tr>
</tbody>
</table>

Figure 4. The reconstructed model for data in Fig. 3 using Algorithm 2 with $q = 100$. (a) Cross-section at northing $= 725$ m; (b) Plane-section at depth $= 100$ m.

Figure 5. The reconstructed model for data in Fig. 3 using Algorithm 2 with $q = 300$. (a) Cross-section at northing $= 725$ m; (b) Plane-section at depth $= 100$ m.

Figure 6. The reconstructed model for data in Fig. 3 using Algorithm 2 with $q = 500$. (a) Cross-section at northing $= 725$ m; (b) Plane-section at depth $= 100$ m.

$D^{(k)} = W_{R_x}^{(k)} D_x$, $W_{R_y}^{(k)} D_y$ or $W_{R_z}^{(k)} D_z$, dependent on $(k \mod 3) = 0, 1$, or 2, respectively. Then $B^{(k)}$ is only of size $n \times n$, yielding reduced memory demands for calculating the GSVD. We note that $D$ is also initialized consistently. The AD approach amounts to apply the edge preserving algorithm in each direction independently and cycling over all directions. Practically, we also find that there is nothing to be gained by requiring that the derivative matrices are square, and following (Hansen 2007) ignore the derivatives at the boundaries. For a 1-D problem this amounts to taking matrix $D_x$ of size $(n - 1) \times n$ for a line with $n$ points. Thus dependent on $k \mod 3$ we have a weighting matrix of size $p_x \times p_x$, $p_y \times p_y$ or $p_z \times p_z$ for matrices $D_x$, $D_y$ and $D_z$ of sizes $p_x \times n$, $p_y \times n$ and $p_z \times n$, respectively, and where $p_x$, $p_y$ and $p_z$ (all less than $n$) depend on the number of boundary points in each direction. Matrix $B^{(k)}$ is thus reduced in size and the GSVD calculation is more efficient at each step. Furthermore, we find that rather than calculating the relevant weight matrix $W_{R_x}$ for the given dimension we actually form the weighted entry that leads to approximation of the...
Figure 7. The reconstructed model for data in Fig. 3 using Algorithm 2 with $q = 300$ at $K = 50$. (a) Cross-section at northing = 725 m; (b) Plane-section at depth = 100 m.

Figure 8. The reconstructed model for data in Fig. 3 using Algorithm 2 with $q = 500$ when uncorrected upper density bound $\rho_{\text{max}} = 2 \text{g cm}^{-3}$ is selected. (a) Cross-section at northing = 725 m; (b) Plane-section at depth = 100 m.

Figure 9. (a) The initial model which generated by adding Gaussian noise with standard deviation of $(0.05 |\mathbf{m}_{\text{true}}| + 0.02 |\mathbf{m}_{\text{true}}|)$ to the true model. (b) The reconstructed model for data in Fig. 3 using Algorithm 2 with $q = 500$ when model shown in Fig. 9(a) is used as $\mathbf{m}_{\text{apr}}$.

Figure 10. Model consists of six bodies with various geometries and sizes embedded in a homogeneous background. Bodies have the densities $1 \text{ g cm}^{-3}$ and $0.8 \text{ g cm}^{-3}$. (a) Plane-section at depth = 100 m; (b) Plane-section at depth = 300 m; (c) Plane-section at depth = 500 m; (d) Plane-section at depth = 700 m.
relevant component of the gradient vector for all three dimensions, thus realistically approximating the gradient as given in (5). For the presented results we will use the AD version of Algorithm 2, noting that this is not necessary for the smaller problems, but is generally more efficient and reliable for the larger 3-D problem formulations.

2.1 Estimation of the regularization parameter \( \alpha \)

As presented to this point we have assumed a known value for the regularization parameter \( \alpha \). Practically we wish to find \( \alpha \) dynamically to appropriately regularize at each step of the iteration so as to recognize that the conditioning of the problem changes with the iteration. Here we use the method of unbiased predictive risk estimation which we have found to be robust in earlier work (Renault et al. 2017; Vatankhah et al. 2015; Vatankhah et al. 2017a). The method, which goes back to Mallows (1973), requires some knowledge of the noise level in the data and was carefully developed in Vogel (2002) for the standard Tikhonov functional. The method was further extended for use with TV regularization by Lin et al. (2010). Defining the residual \( R(h(\alpha)) = \tilde{G}h(\alpha) - \tilde{r} \) and influence matrix \( H_{TV,\alpha} = \tilde{G}(G^T \tilde{G} + \alpha^2 D^T D)^{-1} \tilde{G}^T \) the optimal parameter \( \alpha \) is the minimizer of

\[
U(\alpha) = \|R(h(\alpha))\|^2_2 + 2 \text{trace}(H_{TV,\alpha}) - m,
\]

which is given in terms of the GSVD by

\[
U(\alpha) = \sum_{i=1}^q \left( \frac{1}{\gamma_i^2 \alpha^2 + 1} \right)^2 (u_i^T r)^2 + 2 \sum_{i=1}^q \frac{\gamma_i^2}{\gamma_i^2 + \alpha^2} - q.
\]

Typically \( \alpha_{\text{opt}} \) is found by evaluating (15) on a range of \( \alpha \), between minimum and maximum \( \gamma_i \), and then that \( \alpha \) which minimizes the function is selected as \( \alpha_{\text{opt}} \).

### 3 SYNTHETIC EXAMPLES

#### 3.1 Model consisting of two dipping dikes

As a first example, we use a complex model that consists of two embedded dipping dikes of different sizes and dipping in opposite directions but with the same density contrast 1 g cm\(^{-3}\), Fig. 2. Gravity data, \( d_{\text{exact}} \), are generated on the surface for a grid of 30 \( \times 30 = 900 \) points with grid spacing 50 m. Gaussian noise with standard deviation (0.02 \( d_{\text{exact}} \)) + 0.002 \( ||d_{\text{exact}}|| \) is added to each datum. Example noisy data, \( d_{\text{obs}} \), are illustrated in Fig. 3. Inversion is performed for the subsurface volume of 9000 cubes of size 50 m in each dimension using the matrix \( \tilde{G} \) of size 900 \( \times \) 900. Use of this relatively small model permits examination of the inversion methodology with respect to different parameter choices and provides the framework to be used for more realistic larger models. All computations are performed on a desktop computer with Intel Core i7-4790 CPU 3.6 GHz processor and 16 GB RAM.

The parameters of the inversion needed for Algorithm 2, and the results, are detailed for each example in Table 2. These are the maximum number of iterations for the inversion \( K_{\text{max}} \), the bound constraints for the model \( \rho_{\text{min}} \) and \( \rho_{\text{max}} \), the initial data \( m_{\text{opt}} \) and the choice for \( q \). Further, the inversion terminates if \( \chi_{\text{computed}}^2 = ||W(d_{\text{obs}} - d_{\text{exact}})||^2_2 \leq m + \sqrt{2m} = 942.4 \) is satisfied for iteration \( K < K_{\text{max}} \). The results for different choices of \( q \), but all other parameters the same, are illustrated in Figs 4–6, respectively. With \( q = 100 \) two dipping structures are recovered but the iteration has not converged by \( K = 200 \), and both the relative error and \( \chi_{\text{computed}}^2 \) are large. With \( q = 300 \) and 500 better reconstructions of the dikes are achieved although the extension of the left dike is overestimated. While the errors are nearly the same, the inversions terminate at \( K = 174 \) and \( K = 49 \) for \( q = 300 \) and \( q = 500 \), respectively. This leads to low computational time when using \( q = 500 \) as compared with the other two cases, see Table 2. We should note here that for all three cases the relative error decreases rapidly for the early iterations, after which there is little change in the model between iterations. For example, the result for \( q = 300 \) at iteration \( K = 50 \), as illustrated in Fig. 7 and detailed in Table 2, is acceptable and is achieved with a substantial reduction in the computational time.

As compared to inversion using the \( L_2 \)-norm of the gradient of the model parameters, see, for example, Li & Oldenburg (1996), or conventional minimum structure inversion, the results obtained using Algorithm 2 provide a subsurface model that is not smooth. Further, as compared with \( L_0 \) and \( L_1 \) norms applied for the model parameters, see, for example, Last & Kubik (1983), Portniaguine & Zhdanov (1999) and Vatankhah et al. (2017a), the TV inversion does not yield a model that is sparse or compact. On the other hand, the TV inversion is far less dependent on correct specification of the model constraints. This is illustrated in Fig. 8, which is the same case as Fig. 6, but with \( \rho_{\text{max}} = 2 \) g cm\(^{-3}\), and demonstrates that the approach is generally robust. Finally, to contrast with the algorithm presented in Bertete-Aguirre et al. (2002) the result in

### Table 3. The results of the inversion for the multiple bodies example using Algorithm 2.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{\text{opt}} )</td>
<td>( \rho_{\text{max}} ) (g cm(^{-3}))</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 12. The reconstructed model for data in Fig. 11 using Algorithm 2 with $q = 1000$. (a) Plane-section at depth $= 100$ m; (b) Plane-section at depth $= 300$ m; (c) Plane-section at depth $= 500$ m; (d) Plane-section at depth $= 700$ m.

Figure 13. The gravity anomaly produced by the reconstructed model shown in Fig. 12.

Fig. 9(b) is for an alternative choice for $m_{apr}$, illustrated in Fig. 9(a), and $q = 500$ which is consistent with the approach in Bertete-Aguirre et al. (2002). Here $m_{apr}$ is obtained by taking the true model with the addition of Gaussian noise with a standard deviation of $(0.05m_{true} + 0.02\|m_{true}\|)$. Not surprisingly the reconstructed density model is more focused and is closer to the true model. The computed value for $\chi^2_{computed}$ is, however, larger than the specified value and the algorithm terminates at $K_{max}$. This occurs due to the appearance of an incorrect density distribution in the first layer of the subsurface. Together, these results demonstrate that the TVRGSVD technique is successful and offers a good option for the solution of larger scale models.

3.2 Model of multiple bodies

We now consider a more complex and larger model that consists of six bodies with various geometries, sizes, depths and densities, as illustrated in Fig. 10, and is further detailed in Vatankhah et al. (2017a). The surface gravity data are calculated on a $100 \times 60$ grid with $100$ m spacing. Gaussian noise with standard deviation of $(0.02(d_{exact}), + 0.001 \|d_{exact}\|)$ is added to each datum, giving the noisy data set as shown in Fig. 11. The subsurface is divided into $100 \times 60 \times 10 = 60000$ cubes with sizes $100$ m in each dimension. For the inversion we use Algorithm 2 with $m_{apr} = 0$, $\rho_{min} = 0$ g cm$^{-3}$, $\rho_{max} = 1$ g cm$^{-3}$ and $K_{max} = 50$. We perform the inversion for three values of $q$, 500, 1000 and 2000, and report the results in Table 3. The reconstruction with $q = 500$ is less satisfactory than that achieved with larger $q$ and the iterations terminate at $K_{max}$ with a large value of $\chi^2_{computed}$. The results with $q = 1000$ and $q = 2000$ have similar relative errors, but the computational cost is much reduced using $q = 1000$; although the desired $\chi^2_{computed}$ is not achieved the result is close and acceptable. The reconstructed model and associated gravity response, using $q = 1000$, are illustrated in Figs 12 and 13, respectively. While the maximum depths of the anomalies are overestimated, the horizontal borders are reconstructed accurately. These two examples suggest that $q > m/6$ is suitable for Algorithm 2, which confirms our previous conclusions when using the randomized SVD, see Vatankhah et al. (2017b).

We should note that as compared to the case in which the standard Tikhonov functional can be used, see such results with the focusing inversion in Vatankhah et al. (2017a,b), the computational cost here is much higher.

4 REAL DATA

We use the gravity data from the Goiás Alkaline Province (GAP) of the central region of Brazil. The GAP is a result of mafic-alkaline magmatism that occurred in the Late Cretaceous and includes mafic–ultramafic alkaline complexes in the northern portion, subvolcanic alkaline intrusions in the central region and volcanic products to the south with several dikes throughout the area (Dutra et al. 2012). We select a region from the northern part of GAP in which Morro do Engenho Complex (ME) outcrops and another intrusive body, A2, is completely covered by Quaternary sediments (Dutra & Marangoni 2009). The data were digitized carefully from fig. 3 in Dutra & Marangoni (2009) and re-gridded into $45 \times 53 = 2385$ data points with spacing 1 km, Fig. 14(a). For these data, the result of the smooth inversion using Li & Oldenburg (1998) algorithm was presented in Dutra & Marangoni (2009) and re-gridded into $45 \times 53 = 2385$ data points with spacing 1 km, Fig. 14(a). For these data, the result of the smooth inversion using Li & Oldenburg (1998) algorithm was presented in Dutra & Marangoni (2009) and Dutra et al. (2012). Furthermore, the result of focusing inversion based on $L_1$-norm stabilizer is available in Vatankhah et al. (2017b). The results using…
Table 4. The results of the inversion for the data presented in Fig. 14 using Algorithm 2.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_{\text{ap}}$</td>
<td>$\rho_{\text{min}} (\text{g cm}^{-3})$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 14. (a) Residual gravity data over the Morro do Engenho complex digitized from Dutra & Marangoni (2009); (b) the response of the reconstructed model shown in Fig. 15.

Figure 15. The plane-sections of the reconstructed model for the data in Fig. 14(a) using Algorithm 2 with $q = 400$. The sections are at the depths specified in the figures.

The TV inversion presented here can therefore be compared with the inversions using both aforementioned algorithms.

For the inversion we divide the subsurface into $45 \times 53 \times 14 = 33,390$ rectangular 1 km prisms. The density bounds $\rho_{\text{min}} = 0 \text{ g cm}^{-3}$ and $\rho_{\text{max}} = 0.3 \text{ g cm}^{-3}$ are selected based on geological information from Dutra & Marangoni (2009). Algorithm 2 was implemented with $q = 500$ and $K_{\text{max}} = 100$. The results of the inversion are presented in Table 4, with the illustration of the predicted data due to the reconstructed model in Fig. 14(b) and the reconstructed model in Fig. 15. As compared to the smooth estimate of the subsurface shown in Dutra & Marangoni (2009), the obtained subsurface is blocky and non-smooth. The subsurface is also not as sparse as that obtained by the focusing inversion in Vatankhah et al. (2017b). The ME and A2 bodies extend up to maximum 12 and 8 km, respectively. Unlike the result obtained using the focusing inversion, for the TV inversion the connection between ME and A2 at depths 4 to 7 km is not strong. We should note that the computational time for the focusing inversion presented in Vatankhah et al. (2017b) is much smaller than for the TV algorithm presented here.

5 CONCLUSIONS

We developed an algorithm for total variation regularization applied to the 3-D gravity inverse problem. The presented algorithm provides a non-smooth and blocky image of the subsurface which may be useful when discontinuity of the subsurface is anticipated. Using the randomized generalized singular value decomposition, we have demonstrated that moderate-scale problems can be solved in a reasonable computational time. This seems to be the first use...
of the TV regularization using the RGSVD for gravity inversion. Our presented results use an Alternating Direction implementation to further improve the efficiency and reduce the memory demands of the problem. The results show that there is less sensitivity to the provision of good bounds for the density values, and thus the TV may have a role to play for moderate-scale problems where limited information on subsurface model parameters is available. It was of interest to investigate the 3-D algorithm developed here in view of the results of Bertete-Aguirre et al. (2002) which advocated TV regularization in the context of 2-D gravity inversion. We obtained results that are comparable with the simulations presented in Bertete-Aguirre et al. (2002) but here for much larger problems. In our simulations we have found that the computational time is much larger than that required for the focusing algorithm presented in Vatankhah et al. (2017a,b). Because it is not possible to transform the TV regularization to standard Tikhonov form, the much more expensive GSVD algorithm has to be used in place of the SVD that can be utilized for the focusing inversion presented in Vatankhah et al. (2017b). On the other hand, the advantage of the TV inversion as compared to the focusing inversion is the lesser dependence on the density constraints and the generation of a subsurface which is not as smooth and admits discontinuities. The impact of the algorithm was illustrated for the inversion of real gravity data from the Morro do Engenho complex in central Brazil.

ACKNOWLEDGEMENTS

We would like to thank two reviewers, Dr. Valeria Paoletti and one anonymous reviewer. Their helpful comments helped us to improve the manuscript.

REFERENCES


Li, Y. & Oldenburg, D.W., 1996. 3-D inversion of magnetic data, Geophysics, 61, 394–408.


APPENDIX A: THE GENERALIZED SINGULAR VALUE DECOMPOSITION

Suppose $\hat{G} \in \mathbb{R}^{m \times n}$, $\hat{D} \in \mathbb{R}^{p \times n}$ and $\mathcal{N}(\hat{G}) \cap \mathcal{N}(\hat{D}) = \emptyset$, where $\mathcal{N}(\hat{G})$ is the null space of matrix $\hat{G}$. Then there exist orthogonal matrices $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{p \times p}$ and a non-singular matrix $W \in \mathbb{R}^{n \times n}$ such that $U^* \hat{G} V^* = \begin{bmatrix} \hat{U} & 0 \\ 0 & \hat{V} \end{bmatrix}$, $U^* \hat{D} V^* = \begin{bmatrix} \hat{U} & 0 \\ 0 & \hat{V} \end{bmatrix}$, and $W = \begin{bmatrix} \hat{U} & 0 \\ 0 & \hat{V} \end{bmatrix}$. The SVD of $\hat{G}$ is then given by $\hat{G} = U \Sigma V^*$, where $\Sigma = \begin{bmatrix} \hat{U} & 0 \\ 0 & \hat{V} \end{bmatrix}$.
$X \in \mathbb{R}^{n \times n}$ such that \( \tilde{G} = U \Lambda X^T \) and \( \tilde{D} = V XM^T \) (Paige & Saunders 1981; Aster et al. 2013). Here, \( \Lambda \in \mathbb{R}^{n \times n} \) is zero except for entries \( 0 < \Lambda_1, \cdots, \Lambda_{m+n} < 1 \), and \( M \) is diagonal of size \( p \times n \) with entries \( M_{1,1} > M_{2,2} \geq \cdots \geq M_{p,p} > 0 \), where \( p := \min(p, n) \). The generalized singular values of the matrix pair \( [\tilde{G}, \tilde{D}] \) are \( \gamma_i = \lambda_i / \mu_i \), where \( \gamma_1 = \cdots = \gamma_{n-m} = 0 < \gamma_{n-m+1} \leq \cdots \leq \gamma_n \), and \( \Lambda^T \Lambda = \text{diag}(0, \cdots, 0, \lambda_1^2, \cdots, \lambda_n^2) \). The generalized singular values of the matrix pair \( [A_1, B_1] \) are \( \alpha_i \), \( \beta_i \), \( \gamma_i \), \( \delta_i \), where \( \gamma_i = \delta_i = \beta_i = \alpha_i = 0 \). The total cost of GSVD is \( 8 \text{mn}^2 + 20n^3 + 6m^2 + 20m^3 \). Next the algorithm requires the calculation of two singular value decompositions, for the top and lower blocks of the matrix in the QR factorization. This costs \( 6n^2m^2 \) and \( 6n^2m^2 + 20m^3 \), respectively, Golub & van Loan (2013, page 493). The total cost for our application is thus \( 58n^3 + (8p + 2m)n^2 + 2n^3(3n + 10m) \). Now for \( p = n \) and \( p = 3n \) the total cost of GSVD are \( 82/3m^3 + 2mn^2 + +2mn^2(3n + 10m) \) and \( 130/3m^3 + 2mn^2 + +2mn^2(3n + 10m) \), respectively. Now also including the cost \( 26n^2 \), found using the SVD, for finding the pseudo-inverse of \( X \), noting that for these large-scale problems while \( X \) is theoretically non-singular in exact arithmetic it can be very ill-conditioned and the solution is more stably obtained using \( X^+ \). In contrast for the RGSVD we have projected matrices \( B_1 \in \mathbb{R}^{n \times q} \) and \( B_2 \in \mathbb{R}^{n \times q} \), and the costs for QR factorization and two SVDS are \( 2q^2(m + p - q/3) \), \( 6mq^2 + 20q^2 \) and \( 6pq^2 + 20q^2 \), respectively. Then, the total cost is equal to \( 2q^2(8p + 8m) + (118/3)q^2 \) which for the cases \( p = n \) and \( p = 3n \) gives totals \( 8q^2(n + m) + 118/3q^2 \) and \( 8q^2(3n + m) + 118/3q^2 \), respectively. Then we need also \( Z \) costing approximately, \( 6n^2q + 20q^2 \).