

The computation aspects of the equivalent-layer technique: review and perspective

Diego Takahashi^{1,*}, André L. A. Reis², Vanderlei C. Oliveira Jr.¹ and Valéria C. F. Barbosa¹

¹*Observatório Nacional, Department of Geophysics, Rio de Janeiro, Brasil*

²*Universidade do Estado do Rio de Janeiro, Department of Applied Geology, Rio de Janeiro, Brasil*

Correspondence*:
Valéria C.F. Barbosa
valcris@on.br

ABSTRACT

Equivalent-layer technique is a powerful tool for processing potential-field data in the space domain. However, the greatest hindrance for using the equivalent-layer technique is its high computational cost for processing massive data sets. The large amount of computer memory usage to store the full sensitivity matrix combined with the computational time required for matrix-vector multiplications and to solve the resulting linear system, are the main drawbacks that made unfeasible the use of the equivalent-layer technique for a long time. More recently, the advances in computational power propelled the development of methods to overcome the heavy computational cost associated with the equivalent-layer technique. We present a comprehensive review of the computation aspects concerning the equivalent-layer technique addressing how previous works have been dealt with the computational cost of this technique. Historically, the high computational cost of the equivalent-layer technique has been overcome by using a variety of strategies such as: moving data-window scheme, column- and row-action updates of the sensitivity matrix, reparametrization, sparsity induction of the sensitivity matrix, iterative methods using the full sensitivity matrix, iterative deconvolution by using the concept of block-Toeplitz Toeplitz-block (BTTB) matrices and direct deconvolution. We compute the number of floating-point operations of some of these strategies adopted in the equivalent-layer technique to show their effectiveness in reducing the computational demand. Numerically, we also address the stability of some of these strategies used in the equivalent-layer technique by comparing with the stability via the classic equivalent-layer technique with the zeroth-order Tikhonov regularization. We show that even for the most computationally efficient methods, which can save up to 10^9 flops, the stability of the linear system is maintained. The two most efficient strategies, iterative and direct deconvolutions, can process large datasets quickly and yield good results. However, direct deconvolution has some drawbacks. Real data from Carajás Mineral Province, Brazil, is also used to validate the results showing a potential field transformation.

Keywords: equivalent layer, gravimetry, fast algorithms, computational cost, stability analysis

1 INTRODUCTION

The equivalent-layer technique has been used by exploration geophysicists for processing potential-field data since the late 1960s (Dampney, 1969). This technique is based on a widely accepted principle, which states that a discrete set of observed potential-field data due to 3D sources can be approximated by that due to a discrete set of virtual sources (such as point masses, dipoles, prisms, doublets). From a theoretical point of view, the equivalent-layer technique is grounded on potential theory (Kellogg, 1967) and consists in considering that the potential field data can be approximated by a linear combination of harmonic functions describing the potential field due to the virtual sources. These sources, commonly called equivalent sources, are arranged on a layer with finite horizontal dimensions and located below the observations. In the classical approach, a linear inverse problem is solved to estimate the physical property of each equivalent source subject to fit the observations. Then, the estimated physical-property distribution on the equivalent layer is used to accomplish the desired potential-field transformation (e.g., interpolation, upward/downward continuation, reduction to the pole). The later step is done by multiplying the estimated physical-property distribution by the matrix of Green's functions associated with the desired potential-field transformation.

Because the linear inverse problem to be solved in the equivalent-layer technique is set up with a full sensitivity matrix, its computational cost strongly depends on the number of potential-field observations and can be very inefficient for dealing with massive data sets. To overcome this problem, computationally efficient methods based on equivalent-layer technique have arose in the late 1980s. To our knowledge, the first method towards improving the efficiency was proposed by Leão and Silva (1989), who used an overlapping moving-window scheme spanning the data set. The strategy adopted in Leão and Silva (1989) involves solving several smaller, regularized linear inverse problems instead of one large problem. This strategy uses a small data window and distributes equivalent sources on a small regular grid at a constant depth below the data surface, with the sources' window extending beyond the boundaries of the data window. Because of the spatial layouts of observed data and equivalent sources in Leão and Silva (1989), the small sensitivity submatrix containing the coordinates of the data and equivalent sources within a window remains constant for all data windows. This holds true regardless of the specific locations of the data and equivalent sources within each window. For each position of the data window, this scheme consists in computing the processed field at the center of the data window only, and the next estimates of the processed field are obtained by shifting the data window across the entire dataset. More recently, Soler and Uieda (2021) extended the method introduced by Leão and Silva (1989) to accommodate irregularly spaced data collected on a non-flat surface. Unlike Leão and Silva (1989), in the generalization proposed by Soler and Uieda (2021), the sensitivity submatrix that includes the coordinates of the data and equivalent sources needs to be computed for each window. Soler and Uieda (2021) developed a computational approach to further enhance the efficiency of the equivalent-layer technique by combining two strategies. The first one — the block-averaging source locations — reduces the number of model parameters and the second strategy — the gradient-boosted algorithm — reduces the size of the linear system to be solved by iteratively fitting the equivalent source model along overlapping windows. It is worth noting that the equivalent-layer strategy of using a moving-window scheme either in Leão and Silva (1989) or in Soler and Uieda (2021) is similar to discrete convolution.

As another strategy to reduce the computational workload of the equivalent-layer technique, some authors have employed column- and row-action updates, which are commonly applied to image reconstruction methods (e.g., Elfving et al., 2017). These methods involve iterative calculations of a single column and a single row of the sensitivity matrix, respectively. Following the strategy column-action update,

Cordell (1992) proposed a computational method in which a single equivalent source positioned below a measurement station is iteratively used to compute both the predicted data and residual data for all stations. In Cordell's method, a single column of the sensitivity matrix is calculated per iteration, meaning that a single equivalent source contributes to data fitting in each iteration. Guspí and Novara (2009) further extended Cordell's method by applying it to scattered magnetic observations. Following the strategy of column-action update, Mendonça and Silva (1994) developed an iterative procedure where one data point is incorporated at a time, and a single row of the sensitivity matrix is calculated per iteration. This strategy adopted by Mendonça and Silva (1994) is known as *equivalent data concept*. This concept is based on the principle that certain data points within a dataset are redundant and, as a result, do not contribute to the final solution. On the other hand, there is a subset of observations known as equivalent data, which effectively contributes to the final solution and fits the remaining redundant data. In their work, Mendonça and Silva (1994) adopted an iterative approach to select a substantially smaller subset of equivalent data from the original dataset.

The next strategy involves reparametrizing the equivalent layer with the objective of solving a smaller linear inverse problem by reducing the dimension of the model space. Oliveira Jr. et al. (2013) reduced the model parameters by approximating the equivalent-source layer by a piecewise-polynomial function defined on a set of user-defined small equivalent-source windows. The estimated parameters are the polynomial coefficients for each window and they are much smaller than the original number of equivalent sources. By using the subspace method, Mendonça (2020) reparametrizes the equivalent layer, which involves reducing the dimension of the linear system from the original parameter-model space to a lower-dimensional subspace. The subspace bases span the parameter-model space and they are constructed by applying the singular value decomposition to the matrix containing the gridded data.

Following the strategy of sparsity induction, Li and Oldenburg (2010) transformed the full sensitivity matrix into a sparse one using orthonormal compactly supported wavelets. Barnes and Lumley (2011) proposed an alternative approach to introduce sparsity based on the use of quadtree discretization to group equivalent sources far from the computation points. Those authors explore the induced sparsity by using specific iterative methods to solve the linear system.

The strategy named iterative methods estimates iteratively the parameter vector that represents a distribution over an equivalent layer. Xia and Sprowl (1991) and Xia et al. (1993) have developed efficient iterative algorithms for updating the distribution of physical properties within the equivalent layer in the wavenumber and space domains, respectively. Specifically, in Xia and Sprowl's (1991) method the physical-property distribution is updated by using the ratio between the squared depth to the equivalent source and the gravitational constant multiplied by the residual between the observed and predicted observation at the measurement station. Siqueira et al. (2017) developed an iterative solution where the sensitivity matrix is transformed into a diagonal matrix with constant terms through the use of the *excess mass criterion* and of the positive correlation between the observed gravity data and the masses on the equivalent layer. The fundamentals of the Siqueira et al.'s method is based on the Gauss' theorem (e.g., Kellogg, 1967, p. 43) and the total excess of mass (e.g., Blakely, 1996, p. 60). All these iterative methods use the full and dense sensitivity matrix to calculate the predicted data and residual data in the whole survey data per iteration. Hence, the iterative methods proposed by Xia and Sprowl (1991), Xia et al. (1993) and Siqueira et al. (2017) neither compress nor reparametrize the sensitivity matrix. Jirigalatu and Ebbing (2019) also proposed an iterative equivalent layer that uses the full and dense sensitivity matrix. However, in their approach, Jirigalatu and Ebbing (2019) efficiently compute the predicted data and residual data for the entire survey per iteration in the wavenumber domain.

115 Following the strategy of the iterative deconvolution, Takahashi et al. (2020, 2022) developed fast and
116 effective equivalent-layer techniques for processing, respectively, gravity and magnetic data by modifying
117 the forward modeling to estimate the physical-property distribution over the equivalent layer through a
118 2D discrete fast convolution. These methods took advantage of the Block-Toeplitz Toeplitz-block (BTTB)
119 structure of the sensitivity matrices, allowing them to be calculated by using only their first column. In
120 practice, the forward modeling uses a single equivalent source, which significantly reduces the required
121 RAM memory.

122 The method introduced by Takahashi et al. (2020, 2022) can be reformulated to eliminate the need for
123 conjugate gradient iterations. This reformulation involves employing a *direct deconvolution* approach (e.g.,
124 Aster et al., 2019, p. 220) with *Wiener filter* (e.g., Gonzalez and Woods, 2002, p. 263).

125 Here, we present a comprehensive review of diverse strategies to solve the linear system of the equivalent
126 layer alongside an analysis of the computational cost and stability of these strategies. To do this analysis,
127 we are using the floating-point operations count to evaluate the performance of a selected set of methods.
128 To test the stability, we are using the linear system sensitivity to noise as a comparison parameter for the
129 fastest of these methods alongside the classical normal equations. A potential-field transformation will also
130 be used to evaluate the quality of the equivalent sources estimation results using both synthetic and real
131 data from Carajás Mineral Province, Brazil.

2 FUNDAMENTALS

Let \mathbf{d} be a $D \times 1$ vector, whose i -th element d_i is the observed potential field at the position (x_i, y_i, z_i) , $i \in \{1 : D\}$, of a topocentric Cartesian system with x , y and z axes pointing to north, east and down, respectively. Consider that d_i can be satisfactorily approximated by a harmonic function

$$f_i = \sum_{j=1}^P g_{ij} p_j, \quad i \in \{1 : D\}, \quad (1)$$

where, p_j represents the scalar physical property of a virtual source (i.e., monopole, dipole, prism) located at (x_j, y_j, z_j) , $j \in \{1 : P\}$ and

$$g_{ij} \equiv g(x_i - x_j, y_i - y_j, z_i - z_j), \quad z_i < \min\{z_j\}, \quad \forall i \in \{1 : D\}, \quad (2)$$

is a harmonic function, where $\min\{z_j\}$ denotes the minimum z_j , or the vertical coordinate of the shallowest virtual source. These virtual sources are called *equivalent sources* and they form an *equivalent layer*. In matrix notation, the potential field produced by all equivalent sources at all points (x_i, y_i, z_i) , $i \in \{1 : D\}$, is given by:

$$\mathbf{f} = \mathbf{G}\mathbf{p}, \quad (3)$$

where \mathbf{p} is a $P \times 1$ vector with j -th element p_j representing the scalar physical property of the j -th equivalent source and \mathbf{G} is a $D \times P$ matrix with element g_{ij} given by equation 2.

The equivalent-layer technique consists in solving a linear inverse problem to determine a parameter vector \mathbf{p} leading to a predicted data vector \mathbf{f} (equation 3) *sufficiently close to* the observed data vector \mathbf{d} , whose i -th element d_i is the observed potential field at (x_i, y_i, z_i) . The notion of *closeness* is intrinsically related to the concept of *vector norm* (e.g., Golub and Van Loan, 2013, p. 68) or *measure of length* (e.g., Menke, 2018, p. 41). Because of that, almost all methods for determining \mathbf{p} actually estimate a parameter vector $\tilde{\mathbf{p}}$ minimizing a length measure of the difference between \mathbf{f} and \mathbf{d} (see subsection 3.1). Given an estimate $\tilde{\mathbf{p}}$, it is then possible to compute a potential field transformation

$$\mathbf{t} = \mathbf{A}\tilde{\mathbf{p}}, \quad (4)$$

where \mathbf{t} is a $T \times 1$ vector with k -th element t_k representing the transformed potential field at the position (x_k, y_k, z_k) , $k \in \{1 : T\}$, and

$$a_{kj} \equiv a(x_k - x_j, y_k - y_j, z_k - z_j), \quad z_k < \min\{z_j\}, \quad \forall k \in \{1 : T\}, \quad (5)$$

is a harmonic function representing the kj -th element of the $T \times P$ matrix \mathbf{A} .

2.1 Spatial distribution and total number of equivalent sources

There is no well-established criteria to define the optimum number P or the spatial distribution of the equivalent sources. We know that setting an equivalent layer with more (less) sources than potential-field data usually leads to an underdetermined (overdetermined) inverse problem (e.g., Menke, 2018, p. 52–53). Concerning the spatial distribution of the equivalent sources, the only condition is that they must rely on a surface that is located below and does not cross that containing the potential field data. Soler and Uieda (2021) present a practical discussion about this topic.

From a theoretical point of view, the equivalent layer reproducing a given potential field data set cannot cross the true gravity or magnetic sources. This condition is a consequence of recognizing that the equivalent layer is essentially an indirect solution of a boundary value problem of potential theory (e.g., Roy, 1962; Zidarov, 1965; Dampney, 1969; Li et al., 2014; Reis et al., 2020). In practical applications, however, there is no guarantee that this condition is satisfied. Actually, it is widely known from practical experience (e.g., Gonzalez et al., 2022) that the equivalent-layer technique works even for the case in which the layer cross the true sources.

Regarding the depth of the equivalent layer, Dampney (1969) proposed a criterion based on horizontal data sampling, suggesting that the equivalent-layer depth should be between two and six times the horizontal grid spacing, considering evenly spaced data. However, when dealing with a survey pattern that has unevenly spaced data, Reis et al. (2020) adopted an alternative empirical criterion. According to their proposal, the depth of the equivalent layer should range from two to three times the spacing between adjacent flight lines. The criteria of Dampney (1969) and Reis et al. (2020) are valid for planar equivalent layers. Cordell (1992) have proposed and an alternative criterion for scattered data that leads to an undulating equivalent layer. This criterion have been slightly modified by Guspí et al. (2004), Guspí and Novara (2009) and Soler and Uieda (2021), for example, and consists in setting one equivalent source below each datum at a depth proportional to the horizontal distance to the nearest neighboring data points. Soler and Uieda (2021) have compared different strategies for defining the equivalent sources depth for the specific problem of interpolating gravity data, but they have not found significant differences between them. Regarding the horizontal layout, Soler and Uieda (2021) proposed the block-averaged sources locations in which the survey area is divided into horizontal blocks and one single equivalent source is assigned to each block. The horizontal coordinates of the single source in a given block is defined by the average horizontal coordinates of the observation points at the block. According to Soler and Uieda (2021), this block-averaged layout may prevent aliasing of the interpolated values, specially when the observations are unevenly sampled. This strategy also reduces the number of equivalent sources without affecting the accuracy of the potential-field interpolation. Besides, it reduces the computational load for estimating the physical property on the equivalent layer.

2.2 Matrix G

Generally, the harmonic function g_{ij} (equation 2) is defined in terms of the inverse distance between the observation point (x_i, y_i, z_i) and the j -th equivalent source at (x_j, y_j, z_j) ,

$$\frac{1}{r_{ij}} \equiv \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}, \quad (6)$$

or by its partial derivatives of first and second orders, respectively given by

$$\partial_{\alpha} \frac{1}{r_{ij}} \equiv \frac{-(\alpha_i - \alpha_j)}{r_{ij}^3}, \quad \alpha \in \{x, y, z\}, \quad (7)$$

and

$$\partial_{\alpha\beta} \frac{1}{r_{ij}} \equiv \begin{cases} \frac{3(\alpha_i - \alpha_j)^2}{r_{ij}^5}, & \alpha = \beta, \\ \frac{3(\alpha_i - \alpha_j)(\beta_i - \beta_j)}{r_{ij}^5} - \frac{1}{r_{ij}^3}, & \alpha \neq \beta, \end{cases} \quad \alpha, \beta \in \{x, y, z\}. \quad (8)$$

In this case, the equivalent layer is formed by punctual sources representing monopoles or dipoles (e.g., Dampney, 1969; Emilia, 1973; Leão and Silva, 1989; Cordell, 1992; Oliveira Jr. et al., 2013; Siqueira et al., 2017; Reis et al., 2020; Takahashi et al., 2020; Soler and Uieda, 2021; Takahashi et al., 2022). Another common approach consists in not defining g_{ij} by using equations 6–8, but other harmonic functions obtained by integrating them over the volume of regular prisms (e.g., Li and Oldenburg, 2010; Barnes and Lumley, 2011; Li et al., 2014; Jirigalatu and Ebbing, 2019). There are also some less common approaches defining the harmonic function g_{ij} (equation 2) as the potential field due to plane faces with constant physical property (Hansen and Miyazaki, 1984), doublets (Silva, 1986) or by computing the double integration of the inverse distance function with respect to z (Guspi and Novara, 2009).

A common assumption for most of the equivalent-layer methods is that the harmonic function g_{ij} (equation 2) is independent on the actual physical relationship between the observed potential field and their true sources (e.g., Cordell, 1992; Guspi and Novara, 2009; Li et al., 2014). Hence, g_{ij} can be defined according to the problem. The only condition imposed to this function is that it decays to zero as the observation point (x_i, y_i, z_i) goes away from the position (x_j, y_j, z_j) of the j -th equivalent source. However, several methods use a function g_{ij} that preserves the physical relationship between the observed potential field and their true sources. For the case in which the observed potential field is gravity data, g_{ij} is commonly defined as a component of the gravitational field produced at (x_i, y_i, z_i) by a point mass or prism located at (x_j, y_j, z_j) , with unit density. On the other hand, g_{ij} is commonly defined as a component of the magnetic induction field produced at (x_i, y_i, z_i) by a dipole or prism located at (x_j, y_j, z_j) , with unit magnetization intensity, when the observed potential field is magnetic data.

The main challenge in the equivalent-layer technique is the computational complexity associated with handling large datasets. This complexity arises because the sensitivity matrix \mathbf{G} (equation 3) is dense regardless of the harmonic function g_{ij} (equation 2) employed. In the case of scattered potential-field data, the structure of \mathbf{G} is not well-defined, regardless of the spatial distribution of the equivalent sources. However, in a specific scenario where (i) each potential-field datum is directly associated with a single equivalent source located directly below it, and (ii) both the data and sources are based on planar and regularly spaced grids, Takahashi et al. (2020, 2022) demonstrate that \mathbf{G} exhibits a block-Toeplitz Toeplitz-block (BTTB) structure. In such cases, the product of \mathbf{G} and an arbitrary vector can be efficiently computed using a 2D fast Fourier transform as a discrete convolution.

3 LINEAR INVERSE PROBLEM OF EQUIVALENT-LAYER TECHNIQUE

3.1 General formulation

A general formulation for almost all equivalent-layer methods can be achieved by first considering that the $P \times 1$ parameter vector \mathbf{p} (equation 3) can be reparameterized into a $Q \times 1$ vector \mathbf{q} according to:

$$\mathbf{p} = \mathbf{H} \mathbf{q}, \quad (9)$$

where \mathbf{H} is a $P \times Q$ matrix. The predicted data vector \mathbf{f} (equation 3) can then be rewritten as follows:

$$\mathbf{f} = \mathbf{G} \mathbf{H} \mathbf{q}. \quad (10)$$

Note that the original parameter vector \mathbf{p} is defined in a P -dimensional space whereas the reparameterized parameter vector \mathbf{q} (equation 9) lies in a Q -dimensional space. For convenience, we use the terms P -space and Q -space to designate them.

In this case, the problem of estimating a parameter vector $\tilde{\mathbf{p}}$ minimizing a length measure of the difference between \mathbf{f} (equation 3) and \mathbf{d} is replaced by that of estimating an auxiliary vector $\tilde{\mathbf{q}}$ minimizing the goal function

$$\Gamma(\mathbf{q}) = \Phi(\mathbf{q}) + \mu \Theta(\mathbf{q}), \quad (11)$$

which is a combination of particular measures of length given by

$$\Phi(\mathbf{q}) = (\mathbf{d} - \mathbf{f})^\top \mathbf{W}_d (\mathbf{d} - \mathbf{f}), \quad (12)$$

and

$$\Theta(\mathbf{q}) = (\mathbf{q} - \bar{\mathbf{q}})^\top \mathbf{W}_q (\mathbf{q} - \bar{\mathbf{q}}), \quad (13)$$

where the regularization parameter μ is a positive scalar controlling the trade-off between the data-misfit function $\Phi(\mathbf{q})$ and the regularization function $\Theta(\mathbf{q})$; \mathbf{W}_d is a $D \times D$ symmetric matrix defining the relative importance of each observed datum d_i ; \mathbf{W}_q is a $Q \times Q$ symmetric matrix imposing prior information on \mathbf{q} ; and $\bar{\mathbf{q}}$ is a $Q \times 1$ vector of reference values for \mathbf{q} that satisfies

$$\bar{\mathbf{p}} = \mathbf{H} \bar{\mathbf{q}}, \quad (14)$$

where $\bar{\mathbf{p}}$ is a $P \times 1$ vector containing reference values for the original parameter vector \mathbf{p} .

After obtaining an estimate $\tilde{\mathbf{q}}$ for the reparameterized parameter vector \mathbf{q} (equation 9), the estimate $\tilde{\mathbf{p}}$ for the original parameter vector (equation 3) is computed by

$$\tilde{\mathbf{p}} = \mathbf{H} \tilde{\mathbf{q}}. \quad (15)$$

The reparameterized vector $\tilde{\mathbf{q}}$ is obtained by first computing the gradient of $\Gamma(\mathbf{q})$,

$$\nabla \Gamma(\mathbf{q}) = -2 \mathbf{H}^\top \mathbf{G}^\top \mathbf{W}_d (\mathbf{d} - \mathbf{f}) + 2 \mu \mathbf{W}_q (\mathbf{q} - \bar{\mathbf{q}}). \quad (16)$$

Then, by considering that $\nabla \Gamma(\tilde{\mathbf{q}}) = \mathbf{0}$ (equation 16), where $\mathbf{0}$ is a vector of zeros, as well as adding and subtracting the term $(\mathbf{H}^\top \mathbf{G}^\top \mathbf{W}_d \mathbf{G} \mathbf{H}) \bar{\mathbf{q}}$, we obtain

$$\tilde{\delta}_q = \mathbf{B} \delta_d, \quad (17)$$

where

$$\tilde{\mathbf{q}} = \tilde{\delta}_q + \bar{\mathbf{q}}, \quad (18)$$

244

$$\delta_d = \mathbf{d} - \mathbf{G} \mathbf{H} \bar{\mathbf{q}}, \quad (19)$$

245

$$\mathbf{B} = \left(\mathbf{H}^\top \mathbf{G}^\top \mathbf{W}_d \mathbf{G} \mathbf{H} + \mu \mathbf{W}_q \right)^{-1} \mathbf{H}^\top \mathbf{G}^\top \mathbf{W}_d, \quad (20)$$

or, equivalently (Menke, 2018, p. 62),

$$\mathbf{B} = \mathbf{W}_q^{-1} \mathbf{H}^\top \mathbf{G}^\top \left(\mathbf{G} \mathbf{H} \mathbf{W}_q^{-1} \mathbf{H}^\top \mathbf{G}^\top + \mu \mathbf{W}_d^{-1} \right)^{-1}. \quad (21)$$

Evidently, we have considered that all inverses exist in equations 20 and 21.

The $Q \times D$ matrix \mathbf{B} defined by equation 20 is commonly used for the case in which $D > Q$, i.e., when there are more data than parameters (overdetermined problems). In this case, we consider that the estimate

250 $\tilde{\mathbf{q}}$ is obtained by solving the following linear system for $\tilde{\delta}_q$ (equation 18):

$$\left(\mathbf{H}^\top \mathbf{G}^\top \mathbf{W}_d \mathbf{G} \mathbf{H} + \mu \mathbf{W}_q \right) \tilde{\delta}_q = \mathbf{H}^\top \mathbf{G}^\top \mathbf{W}_d \delta_d. \quad (22)$$

251 On the other hand, for the cases in which $D < Q$ (underdetermined problems), matrix \mathbf{B} is usually defined
 252 according to equation 21. In this case, the general approach involves estimating $\tilde{\mathbf{q}}$ in two steps. The first
 253 consists in solving a linear system for a dummy vector, which is subsequently used to compute $\tilde{\mathbf{q}}$ by a
 254 matrix-vector product as follows:

$$\begin{aligned} \left(\mathbf{G} \mathbf{H} \mathbf{W}_q^{-1} \mathbf{H}^\top \mathbf{G}^\top + \mu \mathbf{W}_d^{-1} \right) \mathbf{u} &= \delta_d \\ \tilde{\delta}_q &= \mathbf{W}_q^{-1} \mathbf{H}^\top \mathbf{G}^\top \mathbf{u} \end{aligned} \quad (23)$$

255 where \mathbf{u} is a dummy vector. After obtaining $\tilde{\delta}_q$ (equations 22 and 23), the estimate $\tilde{\mathbf{q}}$ is computed with
 256 equation 18.

257 3.2 Formulation without reparameterization

258 Note that, for the particular case in which $\mathbf{H} = \mathbf{I}_P$ (equation 9), where \mathbf{I}_P is the identity of order P ,
 259 $P = Q$, $\mathbf{p} = \mathbf{q}$, $\bar{\mathbf{p}} = \bar{\mathbf{q}}$ (equation 14) and $\tilde{\mathbf{p}} = \tilde{\mathbf{q}}$ (equation 15). In this case, the linear system (equations 22
 260 and 23) is directly solved for

$$\tilde{\delta}_p = \tilde{\mathbf{p}} - \bar{\mathbf{p}}, \quad (24)$$

261 instead of $\tilde{\delta}_q$ (equation 18).

262 3.3 Linear system solvers

263 According to their properties, the linear systems associated with over and underdetermined problems
 264 (equations 22 and 23) can be solved by using *direct methods* such as LU, Cholesky or QR factorization, for
 265 example (Golub and Van Loan, 2013, sections 3.2, 4.2 and 5.2). These methods involve factorizing the
 266 linear system matrix in a product of “simple” matrices (i.e., triangular, diagonal or orthogonal). Here, we
 267 consider the *Cholesky factorization*, (Golub and Van Loan, 2013, p. 163).

268 Let us consider a real linear system $\mathbf{M} \mathbf{x} = \mathbf{y}$, where \mathbf{M} is a symmetric and positive definite matrix
 269 (Golub and Van Loan, 2013, p. 159). In this case, the Cholesky factorization consists in computing

$$\mathbf{M} = \mathcal{G} \mathcal{G}^\top, \quad (25)$$

270 where \mathcal{G} is a lower triangular matrix called *Cholesky factor* and having positive diagonal entries. Given \mathcal{G} ,
 271 the original linear system is replaced by two triangular systems, as follows:

$$\begin{aligned} \mathcal{G} \mathbf{s} &= \mathbf{y} \\ \mathcal{G}^\top \mathbf{x} &= \mathbf{s} \end{aligned} \quad (26)$$

272 where \mathbf{s} is a dummy vector. For the overdetermined problem (equation 22), $\mathbf{M} =$
 273 $\left(\mathbf{H}^\top \mathbf{G}^\top \mathbf{W}_d \mathbf{G} \mathbf{H} + \mu \mathbf{W}_q \right)$, $\mathbf{x} = \tilde{\delta}_q$ and $\mathbf{y} = \left(\mathbf{H}^\top \mathbf{G}^\top \mathbf{W}_d \delta_d \right)$. For the underdetermined problem
 274 (equation 23), $\mathbf{M} = \left(\mathbf{G} \mathbf{H} \mathbf{W}_q^{-1} \mathbf{H}^\top \mathbf{G}^\top + \mu \mathbf{W}_d^{-1} \right)$, $\mathbf{x} = \mathbf{u}$ and $\mathbf{y} = \delta_d$.

275 The use of direct methods for solving large linear systems may be problematic due to computer (i) storage
 276 of large matrices and (ii) time to perform matrix operations. This problem may be specially complicated in
 277 equivalent-layer technique for the cases in which the sensitivity matrix \mathbf{G} does not have a well-defined
 278 structure (sec. 2.2)

279 These problems can be overcome by solving the linear system using an iterative method. These methods
 280 produce a sequence of vectors that typically converge to the solution at a reasonable rate. The main
 281 computational cost associated with these methods is usually some matrix-vector products per iteration. The
 282 *conjugate gradient* (CG) is a very popular iterative method for solving linear systems in equivalent-layer
 283 methods. This method was originally developed to solve systems having a square and positive definite
 284 matrix. There are two adapted versions of the CG method. The first is called *conjugate gradient normal*
 285 *equation residual* (CGNR) Golub and Van Loan (2013, sec. 11.3) or *conjugate gradient least squares*
 286 (CGLS) (Aster et al., 2019, p. 165) and is used to solve overdetermined problems (equation 22). The second
 287 is called *conjugate gradient normal equation error* (CGNE) method Golub and Van Loan (2013, sec. 11.3)
 288 and is used to solve the underdetermined problems (equation 23). Algorithm 1 outlines the CGLS method
 289 applied to the overdetermined problem (equation 22).

4 FLOATING-POINT OPERATIONS

290 Two important factors affecting the efficiency of a given matrix algorithm are the storage and amount of
 291 required arithmetic. Here, we quantify this last factor associated with different computational strategies to
 292 solve the linear system of the equivalent-layer technique (section 7). To do it, we opted by counting *flops*,
 293 which are floating point additions, subtractions, multiplications or divisions (Golub and Van Loan, 2013,
 294 p. 12–14). This is a non-hardware dependent approach that allows us to do direct comparison between
 295 different equivalent-layer methods. Most of the flops count used here can be found in Golub and Van Loan
 296 (2013, p. 12, 106, 107 and 164).

297 Let us consider the case in which the overdetermined problem (equation 22) is solved by Cholesky
 298 factorization (equations 25 and 26) directly for the parameter vector $\tilde{\mathbf{p}}$ by considering the particular case in
 299 which $\mathbf{H} = \mathbf{I}_P$ (equation 9 and subsection 3.2), $\mu = 0$ (equation 11), $\mathbf{W}_d = \mathbf{I}_D$ (equation 12) and $\tilde{\mathbf{p}} = \mathbf{0}$
 300 (equation 14), where \mathbf{I}_P and \mathbf{I}_D are the identities of order P and D , respectively. Based on the information
 301 provided in table 1, the total number of flops can be determined by aggregating the flops required for
 302 various computations. These computations include the matrix-matrix and matrix-vector products $\mathbf{G}^\top \mathbf{G}$
 303 and $\mathbf{G}^\top \mathbf{d}$, the Cholesky factor \mathcal{G} , and the solution of triangular systems. Thus, we can express the total
 304 number of flops as follows:

$$f_{\text{Cholesky}} = 1/3 D^3 + 2D^2 + 2(P^2 + P)D. \quad (27)$$

305 The same particular overdetermined problem can be solved by using the CGLS method (Algorithm 1).
 306 In this case, we use table 1 again to combine the total number of flops associated with the matrix-vector
 307 and inner products defined in line 3, before starting the iteration, and the 3 saxpys, 2 inner products and 2
 308 matrix-vector products per iteration (lines 7 – 12). By considering a maximum number of iterations ITMAX,
 309 we obtain

$$f_{\text{CGLS}} = 2P(D + 1) + \text{ITMAX} [2P(2D + 3) + 4D]. \quad (28)$$

310 The same approach used to deduce equations 27 and 28 is applied to compute the total number of flops for
 311 the selected equivalent-layer methods discussed in section 7.

To simplify our analysis, we do not consider the number of flops required to compute the sensitivity matrix \mathbf{G} (equation 3) or the matrix \mathbf{A} associated with a given potential-field transformation (equation 4) because they depend on the specific harmonic functions g_{ij} and a_{ij} (equations 2 and 5). We also neglect the required flops to compute \mathbf{H} , \mathbf{W}_d , \mathbf{W}_q (equations 9, 12 and 13), \bar{p} (equation 14), retrieve $\tilde{\mathbf{q}}$ from $\tilde{\delta}_q$ (equation 18) and computing δ_d (equation 19).

5 NUMERICAL STABILITY

All equivalent-layer methods aim at obtaining an estimate $\tilde{\mathbf{p}}$ for the parameter vector \mathbf{p} (equation 3), which contains the physical property of the equivalent sources. Some methods do it by first obtaining an estimate $\tilde{\mathbf{q}}$ for the reparameterized parameter vector \mathbf{q} (equation 9) and then using it to obtain $\tilde{\mathbf{p}}$ (equation 15). The stability of a solution $\tilde{\mathbf{p}}$ against noise in the observed data is rarely addressed. Here, we follow the numerical stability analysis presented in Siqueira et al. (2017).

For a given equivalent-layer method (section 7), we obtain an estimate $\tilde{\mathbf{p}}$ assuming noise-free potential-field data \mathbf{d} . Then, we create L different noise-corrupted data \mathbf{d}^ℓ , $\ell \in \{1 : L\}$, by adding L different sequences of pseudorandom Gaussian noise to \mathbf{d} , all of them having zero mean. From each \mathbf{d}^ℓ , we obtain an estimate $\tilde{\mathbf{p}}^\ell$. Regardless of the particular equivalent-layer method used, the following inequality (Aster et al., 2019, p. 66) holds true:

$$\Delta p^\ell \leq \kappa \Delta d^\ell, \quad \ell \in \{1 : L\}, \quad (29)$$

where κ is the constant of proportionality between the model perturbation

$$\Delta p^\ell = \frac{\|\tilde{\mathbf{p}}^\ell - \tilde{\mathbf{p}}\|}{\|\tilde{\mathbf{p}}\|}, \quad \ell \in \{1 : L\}, \quad (30)$$

and the data perturbation

$$\Delta d^\ell = \frac{\|\mathbf{d}^\ell - \mathbf{d}\|}{\|\mathbf{d}\|}, \quad \ell \in \{1 : L\}, \quad (31)$$

with $\|\cdot\|$ representing the Euclidean norm. The constant κ acts as the condition number associated with the pseudo-inverse in a given linear inversion. The larger (smaller) the value of κ , the more unstable (stable) is the estimated solution. Because of that, we designate κ as *stability parameter*. Equation 29 shows a linear relationship between the model perturbation Δp^ℓ and the data perturbation Δd^ℓ (equations 30 and 31). We estimate the κ (equation 29) associated with a given equivalent-layer method as the slope of the straight line fitted to the *numerical stability curve* formed by the L points $(\Delta p^\ell, \Delta d^\ell)$.

6 NOTATION FOR SUBVECTORS AND SUBMATRICES

Here, we use a notation inspired on that presented by Van Loan (1992, p. 4) to represent subvectors and submatrices. Subvectors of \mathbf{d} , for example, are specified by $\mathbf{d}[\mathbf{i}]$, where \mathbf{i} is a list of integer numbers that “pick out” the elements of \mathbf{d} forming the subvector $\mathbf{d}[\mathbf{i}]$. For example, $\mathbf{i} = (1, 6, 4, 6)$ gives the subvector $\mathbf{d}[\mathbf{i}] = [d_1 \ d_6 \ d_4 \ d_6]^\top$. Note that the list \mathbf{i} of indices may be sorted or not and it may also have repeated indices. For the particular case in which the list has a single element $\mathbf{i} = (i)$, then it can be used to extract the i -th element $d_i \equiv \mathbf{d}[\mathbf{i}]$ of \mathbf{d} . Sequential lists can be represented by using the colon notation. We consider two types of sequential lists. The first has starting index is smaller than the final index and increment of 1.

The second has starting index is greater than the final index and increment of -1 . For example,

$$\begin{aligned}\mathbf{i} &= (3 : 8) \Leftrightarrow \mathbf{d}[\mathbf{i}] = [d_3 \ d_4 \ \dots \ d_8]^\top \\ \mathbf{i} &= (8 : 3) \Leftrightarrow \mathbf{d}[\mathbf{i}] = [d_8 \ d_7 \ \dots \ d_3]^\top \\ \mathbf{i} &= (: 8) \Leftrightarrow \mathbf{d}[\mathbf{i}] = [d_1 \ d_2 \ \dots \ d_8]^\top \\ \mathbf{i} &= (3 :) \Leftrightarrow \mathbf{d}[\mathbf{i}] = [d_3 \ d_4 \ \dots \ d_D]^\top\end{aligned},$$

where D is the number of elements forming \mathbf{d} .

The notation above can also be used to define submatrices of a $D \times P$ matrix \mathbf{G} . For example, $\mathbf{i} = (2, 7, 4, 6)$ and $\mathbf{j} = (1, 3, 8)$ lead to the submatrix

$$\mathbf{G}[\mathbf{i}, \mathbf{j}] = \begin{bmatrix} g_{21} & g_{23} & g_{28} \\ g_{71} & g_{73} & g_{78} \\ g_{41} & g_{43} & g_{48} \\ g_{61} & g_{63} & g_{68} \end{bmatrix}.$$

Note that, in this case, the lists \mathbf{i} and \mathbf{j} “pick out”, respectively, the rows and columns of \mathbf{G} that form the submatrix $\mathbf{G}[\mathbf{i}, \mathbf{j}]$. The i -th row of \mathbf{G} is given by the $1 \times P$ vector $\mathbf{G}[i, :]$. Similarly, the $D \times 1$ vector $\mathbf{G}[:, j]$ represents the j -th column. Finally, we may use the colon notation to define the following submatrix:

$$\mathbf{i} = (2 : 5), \mathbf{j} = (3 : 7) \Leftrightarrow \mathbf{G}[\mathbf{i}, \mathbf{j}] = \begin{bmatrix} g_{23} & g_{24} & g_{25} & g_{26} & g_{27} \\ g_{33} & g_{34} & g_{35} & g_{36} & g_{37} \\ g_{43} & g_{44} & g_{45} & g_{46} & g_{47} \\ g_{53} & g_{54} & g_{55} & g_{56} & g_{57} \end{bmatrix},$$

which contains the contiguous elements of \mathbf{G} from rows 2 to 5 and from columns 3 to 7.

7 COMPUTATIONAL STRATEGIES

The linear inverse problem of the equivalent-layer technique (section 3) for the case in which there are large volumes of potential-field data requires dealing with:

- (i) the large computer memory to store large and full matrices;
- (ii) the long computation time to multiply a matrix by a vector; and
- (iii) the long computation time to solve a large linear system of equations.

Here, we review some strategies aiming at reducing the computational cost of the equivalent-layer technique. We quantify the computational cost by using flops (section 4) and compare the results with those obtained for Cholesky factorization and CGLS (equations 27 and 28). We focus on the overall strategies used by the selected methods.

7.1 Moving window

The initial approach to enhance the computational efficiency of the equivalent-layer technique is commonly denoted *moving window* and involves first splitting the observed data d_i , $i \in \{1 : D\}$, into M overlapping subsets (or data windows) formed by D^m data each, $m \in \{1 : M\}$. The data inside the

350 m -th window are usually adjacent to each other and have indices defined by an integer list \mathbf{i}^m having
 351 D^m elements. The number of data D^m forming the data windows are not necessarily equal to each other.
 352 Each data window has a $D^m \times 1$ observed data vector $\mathbf{d}^m \equiv \mathbf{d}[\mathbf{i}^m]$. The second step consists in defining
 353 a set of P equivalent sources with scalar physical property p_j , $j \in \{1 : P\}$, and also split them into M
 354 overlapping subsets (or source windows) formed by P^m data each, $m \in \{1 : M\}$. The sources inside the
 355 m -th window have indices defined by an integer list \mathbf{j}^m having P^m elements. Each source window has a
 356 $P^m \times 1$ parameter vector \mathbf{p}^m and is located right below the corresponding m -th data window. Then, each
 357 $\mathbf{d}^m \equiv \mathbf{d}[\mathbf{i}^m]$ is approximated by

$$\mathbf{f}^m = \mathbf{G}^m \mathbf{p}^m, \quad (32)$$

358 where $\mathbf{G}^m \equiv \mathbf{G}[\mathbf{i}^m, \mathbf{j}^m]$ is a submatrix of \mathbf{G} (equation 3) formed by the elements computed with equation
 359 2 using only the data and equivalent sources located inside the window m -th. The main idea of the moving-
 360 window approach is using the $\tilde{\mathbf{p}}^m$ estimated for each window to obtain (i) an estimate $\tilde{\mathbf{p}}$ of the parameter
 361 vector for the entire equivalent layer or (ii) a given potential-field transformation \mathbf{t} (equation 4). The main
 362 advantages of this approach is that (i) the estimated parameter vector $\tilde{\mathbf{p}}$ or transformed potential field are
 363 not obtained by solving the full, but smaller linear systems and (ii) the full matrix \mathbf{G} (equation 3) is never
 364 stored.

365 Leão and Silva (1989) presented a pioneer work using the moving-window approach. Their method
 366 requires a regularly-spaced grid of observed data on a horizontal plane z_0 . The data windows are defined by
 367 square local grids of $\sqrt{D'} \times \sqrt{D'}$ adjacent points, all of them having the same number of points D' . The
 368 equivalent sources in the m -th data window are located below the observation plane, at a constant vertical
 369 distance Δz_0 . They are arranged on a regular grid of $\sqrt{P'} \times \sqrt{P'}$ adjacent points following the same
 370 grid pattern of the observed data. The local grid of sources for all data windows have the same number
 371 of elements P' . Besides, they are vertically aligned, but expands the limits of their corresponding data
 372 windows, so that $D' < P'$. Because of this spatial configuration of observed data and equivalent sources,
 373 we have that $\mathbf{G}^m = \mathbf{G}'$ (equation 32) for all data windows (i.e., $\forall m \in \{1 : M\}$), where \mathbf{G}' is a $D' \times P'$
 374 constant matrix.

375 By omitting the normalization strategy used by Leão and Silva (1989), their method consists in directly
 376 computing the transformed potential field t_c^m at the central point $(x_c^m, y_c^m, z_0 + \Delta z_0)$ of each data window
 377 as follows:

$$t_c^m = (\mathbf{a}')^\top \mathbf{B}' \mathbf{d}^m, \quad m \in \{1 : M\}, \quad (33)$$

378 where \mathbf{a}' is a $P' \times 1$ vector with elements computed by equation 5 by using all equivalent sources in the
 379 m -th window and only the coordinate of the central point in the m -th data window and

$$\mathbf{B}' = (\mathbf{G}')^\top \left[\mathbf{G}' (\mathbf{G}')^\top + \mu \mathbf{I}_{D'} \right]^{-1} \quad (34)$$

380 is a particular case of matrix \mathbf{B} associated with underdetermined problems (equation 21) for the particular
 381 case in which $\mathbf{H} = \mathbf{W}_q = \mathbf{I}_{P'}$ (equations 9 and 13), $\mathbf{W}_d = \mathbf{I}_{D'}$ (equation 12), $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14), where
 382 $\mathbf{I}_{P'}$ and $\mathbf{I}_{D'}$ are identity matrices of order P' and D' , respectively, and $\mathbf{0}$ is a vector of zeros. Due to the
 383 presumed spatial configuration of the observed data and equivalent sources, \mathbf{a}' and \mathbf{G}' are the same for all
 384 data windows. Hence, only the data vector \mathbf{d}^m is modified according to the position of the data window.
 385 Note that equation 33 combines the potential-field transformation (equation 4) with the solution of the
 386 undetermined problem (equation 23).

387 The method proposed by Leão and Silva (1989) can be outlined by the Algorithm 2. Note that Leão and
 388 Silva (1989) directly compute the transformed potential t_c^m at the central point of each data window without
 389 explicitly computing and storing an estimated for \mathbf{p}^m (equation 32). It means that their method allows
 390 computing a single potential-field transformation. A different transformation or the same one evaluated at
 391 different points require running their moving-data window method again.

392 The total number of flops in Algorithm 2 depends on computing the $P' \times D'$ matrix \mathbf{B}' (equation 34) in
 393 line 6 and use it to define the $1 \times P'$ vector $(\mathbf{a}')^\top \mathbf{B}'$ (line 7) before starting the iterations and computing
 394 an inner product (equation 33) per iteration. We consider that the total number of flops associated with \mathbf{B}'
 395 is obtained by the matrix-matrix product $\mathbf{G}' (\mathbf{G}')^\top$, its inverse and then the premultiplication by $(\mathbf{G}')^\top$. By
 396 using table 1 and considering that inverse is computed via Cholesky factorization, we obtain that the total
 397 number of flops for lines 6 and 7 is $2(D')^2 P' + 7(D')^3/6 + 2(D')^2 P'$. Then, the total number of flops for
 398 Algorithm 2 is

$$f_{\text{LS89}} = 7/6(D')^3 + 4P'(D')^2 + M 2P'. \quad (35)$$

399 Soler and Uieda (2021) generalized the method proposed by Leão and Silva (1989) for irregularly spaced
 400 data on an undulating surface. A direct consequence of this generalization is that a different submatrix
 401 $\mathbf{G}^m \equiv \mathbf{G}[\mathbf{i}^m, \mathbf{j}^m]$ (equation 32) must be computed for each window. Differently from Leão and Silva
 402 (1989), Soler and Uieda (2021) store the computed $\tilde{\mathbf{p}}^m$ for all windows and subsequently use them to obtain
 403 a desired potential-field transformation (equation 4) as the superposed effect of all windows. The estimated
 404 $\tilde{\mathbf{p}}^m$ for all windows are combined to form a single $P \times 1$ vector $\tilde{\mathbf{p}}$, which is an estimate for original
 405 parameter vector \mathbf{p} (equation 3). For each data window, Soler and Uieda (2021) solve an overdetermined
 406 problem (equation 22) for $\tilde{\mathbf{p}}^m$ by using $\mathbf{H} = \mathbf{W}_q = \mathbf{I}_{P^m}$ (equations 9 and 13), \mathbf{W}_d^m (equation 12) equal to
 407 a diagonal matrix of weights for the data inside the m -th window and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14), so that

$$\left[(\mathbf{G}^m)^\top \mathbf{W}_d^m \mathbf{G}^m + \mu \mathbf{I}_{P'} \right] \tilde{\mathbf{p}}^m = (\mathbf{G}^m)^\top \mathbf{W}_d^m \mathbf{d}^m. \quad (36)$$

408 Unlike Leão and Silva (1989), Soler and Uieda (2021) do not adopt a sequential order of the data windows;
 409 rather, they adopt a randomized order of windows in their iterations. The overall steps of the method
 410 proposed by Soler and Uieda (2021) are defined by the Algorithm 3. For convenience, we have omitted the
 411 details about the randomized window order, normalization strategy employed and block-averaged sources
 412 layout proposed by those authors (see subsection 2.1). Note that this algorithm starts with a residuals vector
 413 \mathbf{r} that is iteratively updated. The iterative algorithm in Soler and Uieda (2021) estimates a solution ($\tilde{\mathbf{p}}^m$ in
 414 equation 36) using the data and the equivalent sources that fall within a moving-data window; however, it
 415 calculates the predicted data and the residual data in the whole survey data. Next, the residual data that fall
 416 within a new position of the data window is used as input data to estimate a new solution within the data
 417 window which, in turn, is used to calculate a new predicted data and a new residual data in the whole
 418 survey data.

419 The computational cost of Algorithm 3 can be defined in terms of the linear system (equation 36) to be
 420 solved for each window (line 10) and the subsequent updates in lines 11 and 12. We consider that the linear
 421 system cost can be quantified by the matrix-matrix and matrix-vector products $(\mathbf{G}^m)^\top \mathbf{G}^m$ and $(\mathbf{G}^m)^\top \mathbf{d}^m$,
 422 respectively, and solution of the linear system (line 10) via Cholesky factorization (equations 25 and 26).
 423 The following updates represent a saxpy without scalar-vector product (line 11) and a matrix-vector product
 424 (line 12). In this case, according to table 1, the total number of flops associated with Algorithm 3 is given

425 by:

$$f_{\text{sv21}} = M \left[\frac{1}{3}(P')^3 + 2(D' + 1)(P')^2 + (4D' + 1)P' \right], \quad (37)$$

426 where P' and D' represent, respectively, the average number of equivalent sources and data at each window.

427 7.2 Column-action update

428 We call the computational strategy *column-action update* because a single source is used to calculate the
429 predicted data and the residual data in the whole survey data. Hence, a single column of the sensitivity
430 matrix \mathbf{G} (equation 3) is calculated iteratively.

431 Cordell (1992) proposed a computational strategy that was later used by Guspí and Novara (2009) and
432 relies on first defining one equivalent source located right below each observed data d_i , $i \in \{1 : D\}$, at
433 a vertical coordinate $z_i + \Delta z_i$, where Δz_i is proportional to the distance from the i -th observation point
434 (x_i, y_i, z_i) to its closest neighbor. The second step consists in updating the physical property p_j of a single
435 equivalent source, $j \in \{1 : D\}$ and remove its predicted potential field from the observed data vector \mathbf{d} ,
436 producing a residuals vector \mathbf{r} . At each iteration, the single equivalent source is the one located vertically
437 beneath the observation station of the maximum data residual. Next, the predicted data produced by this
438 single source is calculated over all of the observation points and a new data residual \mathbf{r} and the $D \times 1$
439 parameter vector \mathbf{p} containing the physical property of all equivalent sources are updated iteratively. During
440 each subsequent iteration, Cordell's method either incorporates a single equivalent source or adjusts an
441 existing equivalent source to match the maximum amplitude of the current residual field. The convergence
442 occurs when all of the residuals are bounded by an envelope of prespecified expected error. At the end, the
443 algorithm produces an estimate $\tilde{\mathbf{p}}$ for the parameter vector yielding a predicted potential field \mathbf{f} (equation
444 3) satisfactorily fitting the observed data \mathbf{d} according to a given criterion. Note that the method proposed
445 by Cordell (1992) iteratively solves the linear $\mathbf{G}\tilde{\mathbf{p}} \approx \mathbf{d}$ with a $D \times D$ matrix \mathbf{G} . At each iteration, only a
446 single column of \mathbf{G} (equation 3) is used. An advantage of this *column-action update approach* is that the
447 full matrix \mathbf{G} is never stored.

448 Algorithm 4 delineates the Cordell's method. Note that a single column $\mathbf{G}[:, i_{\max}]$ of the $D \times D$ matrix \mathbf{G}
449 (equation 3) is used per iteration, where i_{\max} is the index of the maximum absolute value in \mathbf{r} . As pointed out
450 by Cordell (1992), the method does not necessarily decrease monotonically along the iterations. Besides,
451 the method may not converge depending on how the vertical distances Δz_i , $i \in \{1 : D\}$, controlling the
452 depths of the equivalent sources are set. According to Cordell (1992), the maximum absolute value r_{\max}
453 in \mathbf{r} decreases robustly at the beginning and oscillates within a narrowing envelope for the subsequent
454 iterations.

455 Guspí and Novara (2009) generalized Cordell's method to perform reduction to the pole and other
456 transformations on scattered magnetic observations by using two steps. The first step involves computing
457 the vertical component of the observed field using equivalent sources while preserving the magnetization
458 direction. In the second step, the vertical observation direction is maintained, but the magnetization
459 direction is shifted to the vertical. The main idea employed by both Cordell (1992) and Guspí and Novara
460 (2009) is an iterative scheme that uses a single equivalent source positioned below a measurement station
461 to compute both the predicted data and residual data for all stations. This approach entails a computational
462 strategy where a single column of the sensitivity matrix \mathbf{G} (equation 3) is calculated per iteration.

463 The total number of flops in Algorithm 4 consists in finding the maximum absolute value in vector \mathbf{r}
464 (line 6) before the while loop. Per iteration, there is a saxpy (line 11) and another search for the maximum
465 absolute value in vector \mathbf{r} (line 12). By considering that selecting the maximum absolute value in a $D \times 1$

vector is a $D \log_2(D)$ operation (e.g., Press et al., 2007, p. 420), the total number of flops in Algorithm 38 is given by:

$$f_{c92} = D \log(D) + \text{ITMAX} [2D + D \log_2(D)] . \quad (38)$$

7.3 Row-action update

We call the computational strategy *row-action update* because a single row of the sensitivity matrix \mathbf{G} (equation 3) is calculated iteratively. Hence, the equivalent-layer solution is updated by processing a new datum (one matrix row) at each iteration. To reduce the total processing time and memory usage of equivalent-layer technique, Mendonça and Silva (1994) proposed a strategy called *equivalent data concept*. The equivalent data concept is grounded on the principle that there is a subset of redundant data that does not contribute to the final solution and thus can be dispensed. Conversely, there is a subset of observations, called equivalent data, that contributes effectively to the final solution and fits the remaining observations (redundant data). Iteratively, Mendonça and Silva (1994) selected the subset of equivalent data that is substantially smaller than the original dataset. This selection is carried out by incorporating one data point at a time.

Mendonça and Silva (1994) proposes an algebraic reconstruction technique (ART) (e.g., van der Sluis and van der Vorst, 1987, p. 58) to estimate a parameter vector $\tilde{\mathbf{p}}$ for a regular grid of P equivalent sources on a horizontal plane z_0 . Such methods iterate on the linear system rows to estimate corrections for the parameter vector, which may substantially save computer time and memory required to compute and store the full linear system matrix along the iterations. The convergence of such *row-update methods* depends on the linear system condition. The main advantage of such methods is not computing and storing the full linear system matrix, but iteratively using its rows. In contrast to ART-type algorithms, the rows in Mendonça and Silva (1994) are not processed sequentially. Instead, in Mendonça and Silva (1994), the rows are introduced according to their residual magnitudes (maximum absolute value in \mathbf{r}), which are computed based on the estimate over the equivalent layer from the previous iteration. The particular ART method proposed by Mendonça and Silva (1994) considers that

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_e \\ \mathbf{d}_r \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{G}_e \\ \mathbf{G}_r \end{bmatrix}, \quad (39)$$

where \mathbf{d}_e and \mathbf{d}_r are $D_e \times 1$ and $D_r \times 1$ vectors and \mathbf{G}_e and \mathbf{G}_r are $D_e \times P$ and $D_r \times P$ matrices, respectively. Mendonça and Silva (1994) designate \mathbf{d}_e and \mathbf{d}_r as, respectively, *equivalent* and *redundant* data. With the exception of a normalization strategy, Mendonça and Silva (1994) calculate a $P \times 1$ estimated parameter vector $\tilde{\mathbf{p}}$ by solving an underdetermined problem (equation 23) involving only the equivalent data \mathbf{d}_e (equation 39) for the particular case in which $\mathbf{H} = \mathbf{W}_p = \mathbf{I}_P$ (equations 9 and 13), $\mathbf{W}_d = \mathbf{I}_{D_e}$ (equation 12) and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14), which results in

$$\begin{aligned} (\mathbf{F} + \mu \mathbf{I}_{D_e}) \mathbf{u} &= \mathbf{d}_e \\ \tilde{\mathbf{p}} &= \mathbf{G}_e^\top \mathbf{u} \end{aligned}, \quad (40)$$

where \mathbf{F} is a computationally-efficient $D_e \times D_e$ matrix that approximates $\mathbf{G}_e \mathbf{G}_e^\top$. Mendonça and Silva (1994) presume that the estimated parameter vector $\tilde{\mathbf{p}}$ obtained from equation 40 leads to a $D_r \times 1$ residuals vector

$$\mathbf{r} = \mathbf{d}_r - \mathbf{G}_r \tilde{\mathbf{p}} \quad (41)$$

499 having a maximum absolute value $r_{\max} \leq \epsilon$, where ϵ is a predefined tolerance.

500 The overall method of Mendonça and Silva (1994) is defined by Algorithm 5. It is important noting
 501 that the number D_e of equivalent data in \mathbf{d}_e increases by one per iteration, which means that the order
 502 of the linear system in equation 40 also increases by one at each iteration. Those authors also propose
 503 a computational strategy based on Cholesky factorization (e.g., Golub and Van Loan, 2013, p. 163) for
 504 efficiently updating $(\mathbf{F} + \mu \mathbf{I}_{D_e})$ at a given iteration (line 16 in Algorithm 5) by computing only its new
 505 elements with respect to those computed in the previous iteration.

506 7.4 Reparameterization

507 Another approach for improving the computational performance of equivalent-layer technique consists
 508 in setting a $P \times Q$ reparameterization matrix \mathbf{H} (equation 9) with $Q \ll P$. This strategy has been used
 509 in applied geophysics for decades (e.g., Skilling and Bryan, 1984; Kennett et al., 1988; Oldenburg et al.,
 510 1993; Barbosa et al., 1997) and is known as *subspace method*. The main idea relies in reducing the linear
 511 system dimension from the original P -space to a lower-dimensional subspace (the Q -space). An estimate
 512 $\tilde{\mathbf{q}}$ for the reparameterized parameter vector \mathbf{q} is obtained in the Q -space and subsequently used to obtain
 513 an estimate $\tilde{\mathbf{p}}$ for the parameter vector \mathbf{p} (equation 3) in the P -space by using equation 9. Hence, the key
 514 aspect of this *reparameterization approach* is solving an appreciably smaller linear inverse problem for $\tilde{\mathbf{q}}$
 515 than that for the original parameter vector \mathbf{p} (equation 3).

516 Oliveira Jr. et al. (2013) have used this approach to describe the physical property distribution on the
 517 equivalent layer in terms of piecewise bivariate polynomials. Specifically, their method consists in splitting
 518 a regular grid of equivalent sources into source windows inside which the physical-property distribution
 519 is described by bivariate polynomial functions. The key aspect of their method relies on the fact that the
 520 total number of coefficients required to define the bivariate polynomials is considerably smaller than the
 521 original number of equivalent sources. Hence, they formulate a linear inverse problem for estimating the
 522 polynomial coefficients and use them later to compute the physical property distribution on the equivalent
 523 layer.

524 The method proposed by Oliveira Jr. et al. (2013) consists in solving an overdetermined problem (equation
 525 22) for estimating the polynomial coefficients $\tilde{\mathbf{q}}$ with $\mathbf{W}_d = \mathbf{I}_D$ (equation 12) and $\bar{\mathbf{q}} = \mathbf{0}$ (equation 14), so
 526 that

$$\left(\mathbf{H}^\top \mathbf{G}^\top \mathbf{G} \mathbf{H} + \mu \mathbf{W}_q \right) \tilde{\mathbf{q}} = \mathbf{H}^\top \mathbf{G}^\top \mathbf{d}, \quad (42)$$

527 where $\mathbf{W}_q = \mathbf{H}^\top \mathbf{W}_p \mathbf{H}$ is defined by a matrix \mathbf{W}_p representing the zeroth- and first-order Tikhonov
 528 regularization (e.g., Aster et al., 2019, p. 103). Note that, in this case, the prior information is defined in the
 529 P -space for the original parameter vector \mathbf{p} and then transformed to the Q -space. Another characteristic of
 530 their method is that it is valid for processing irregularly-spaced data on an undulating surface.

531 Mendonça (2020) also proposed a reparameterization approach for the equivalent-layer technique. Their
 532 approach, however, consists in setting \mathbf{H} as a truncated singular value decomposition (SVD) (e.g., Aster
 533 et al., 2019, p. 55) of the observed potential field. Differently from Oliveira Jr. et al. (2013), however, the
 534 method of Mendonça (2020) requires a regular grid of potential-field data on horizontal plane. Another
 535 difference is that these authors uses $\mathbf{W}_q = \mathbf{I}_Q$ (equation 13), which means that the regularization is defined
 536 directly in the Q -space.

537 We consider an algorithm (not shown) that solves the overdetermined problem (equation 22) by combining
 538 the reparameterization with CGLS method (Algorithm 1). It starts with a reparameterization step defined

by defining a matrix $\mathbf{C} = \mathbf{G} \mathbf{H}$ (equation 10). Then, the CGLS (Algorithm 1) is applied by replacing \mathbf{G} with \mathbf{C} . In this case, the linear system is solved by the reparameterized parameter vector $\tilde{\mathbf{q}}$ instead of $\tilde{\mathbf{p}}$. At the end, the estimated $\tilde{\mathbf{q}}$ is transformed into $\tilde{\mathbf{p}}$ (equation 15). Compared to the original CGLS shown in Algorithm 1, the algorithm discussed here has the additional flops associated with the matrix-matrix product to compute \mathbf{C} and the matrix-vector product of equation 15 outside the while loop. Then, according to table 1, the total number of flops given by:

$$f_{\text{reparam.}} = 2Q(DP + D + 1) + 2PQ + \text{ITMAX} [2Q(2D + 3) + 4D] . \quad (43)$$

The important aspect of this approach is that, for the case in which $Q \ll P$ (equation 9), the number of flops per iteration can be substantially decreased with respect to those associated with Algorithm 1. In this case, the flops decrease per iteration compensates the additional flops required to compute \mathbf{C} and obtain $\tilde{\mathbf{p}}$ from $\tilde{\mathbf{q}}$ (equation 15).

7.5 Sparsity induction

Li and Oldenburg (2010) proposed a method that applies the discrete wavelet transform to introduce sparsity into the original dense matrix \mathbf{G} (equation 3). Those authors approximate a planar grid of potential-field data by a regularly-spaced grid of equivalent sources, so that the number of data D and sources P is the same, i.e., $D = P$. Specifically, Li and Oldenburg (2010) proposed a method that applies the wavelet transform to the original dense matrix \mathbf{G} and sets to zero the small coefficients that are below a given threshold, which results in an approximating sparse representation of \mathbf{G} in the wavelet domain. They first consider the following approximation

$$\mathbf{d}_w \approx \mathbf{G}_s \mathbf{p}_w , \quad (44)$$

where

$$\mathbf{d}_w = \mathbf{W} \mathbf{d} , \quad \mathbf{p}_w = \mathbf{W} \mathbf{p} , \quad (45)$$

are the observed data and parameter vector in the wavelet domain; \mathbf{W} is a $D \times D$ orthogonal matrix defining a discrete wavelet transform; and \mathbf{G}_s is a sparse matrix obtained by setting to zero the elements of

$$\mathbf{G}_w = \mathbf{W} \mathbf{G} \mathbf{W}^\top \quad (46)$$

with absolute value smaller than a given threshold.

Li and Oldenburg (2010) solve a normalized inverse problem in the wavelet domain. Specifically, they first define a matrix

$$\mathbf{G}_L = \mathbf{G}_s \mathbf{L}^{-1} \quad (47)$$

and a normalized parameter vector

$$\mathbf{p}_L = \mathbf{L} \mathbf{p}_w , \quad (48)$$

where \mathbf{L} is a diagonal and invertible matrix representing an approximation of the first-order Tikhonov regularization in the wavelet domain. Then they solve an overdetermined problem (equation 22) to obtain an estimate $\tilde{\mathbf{p}}_L$ for \mathbf{p}_L (equation 48), with \mathbf{G}_L (equation 47), $\mathbf{H} = \mathbf{I}_P$ (equations 9), $\mu = 0$ (equation 11), $\mathbf{W}_d = \mathbf{I}_D$ (equation 12) and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14) via conjugate-gradient method (e.g., Golub and Van Loan, 2013, sec. 11.3). Finally, Li and Oldenburg (2010) compute an estimate $\tilde{\mathbf{p}}$ for the original parameter vector given by

$$\tilde{\mathbf{p}} = \mathbf{W}^\top (\mathbf{L}^{-1} \tilde{\mathbf{p}}_L) , \quad (49)$$

where the term within parenthesis is an estimate $\tilde{\mathbf{p}}_w$ of the parameter vector \mathbf{p}_w (equation 45) in the wavelet domain and matrix \mathbf{W}^\top represents an inverse wavelet transform.

Barnes and Lumley (2011) also proposed a computationally efficient method for equivalent-layer technique by inducing sparsity into the original sensitivity matrix \mathbf{G} (equation 3). Their approach consists in setting a $P \times Q$ reparameterization matrix \mathbf{H} (equation 9) with $Q \approx 1.7 P$. Note that, differently from Oliveira Jr. et al. (2013) and Mendonça (2020), Barnes and Lumley (2011) do not use the reparameterization with the purpose of reducing the number of the parameters. Instead, they use a reparameterization scheme that groups distant equivalent sources into blocks by using a bisection process. This scheme leads to a quadtree representation of the physical-property distribution on the equivalent layer, so that matrix $\mathbf{G}\mathbf{H}$ (equation 10) is notably sparse. Barnes and Lumley (2011) explore this sparsity in solving the overdetermined problem for $\tilde{\mathbf{q}}$ (equation 42) via conjugate-gradient method (e.g., Golub and Van Loan, 2013, sec. 11.3).

It is difficult to predict the exact sparsity obtained from the methods proposed by Li and Oldenburg (2010) and Barnes and Lumley (2011) because it depends on several factors, including the observed potential-field data. According to Li and Oldenburg (2010), their wavelet approach results in a sparse matrix having $\approx 2\%$ of the elements in \mathbf{G}_w (equation 46). The reparameterization proposed by Barnes and Lumley (2011) leads to a sparse matrix $\mathbf{G}\mathbf{H}$ (equation 10) with only $\approx 1\%$ of non-zero elements. These sparsity patterns can be efficiently explored, for example, in computing the required matrix-vector products along the iterations of the CGLS method (Algorithm 1).

7.6 Iterative methods using the full matrix \mathbf{G}

Xia and Sprowl (1991) introduced an iterative method for estimating the parameter vector $\tilde{\mathbf{p}}$ (equation 3), which was subsequently adapted to the Fourier domain by Xia et al. (1993). Their method uses the full and dense sensitivity matrix \mathbf{G} (equation 3) (without applying any compression or reparameterization, for example) to compute the predicted data at all observation points per iteration. More than two decades later, Siqueira et al. (2017) have proposed an iterative method similar to that presented by Xia and Sprowl (1991). The difference is that Siqueira et al.'s algorithm was deduced from the *Gauss' theorem* (e.g., Kellogg, 1967, p. 43) and the *total excess of mass* (e.g., Blakely, 1996, p. 60). Besides, Siqueira et al. (2017) have included a numerical analysis showing that their method produces very stable solutions, even for noise-corrupted potential-field data.

The iterative method proposed by Siqueira et al. (2017) is outlined in Algorithm 6, presumes an equivalent layer formed by monopoles (point masses) and can be applied to irregularly-spaced data on an undulating surface. Note that the residuals \mathbf{r} are used to compute a correction $\Delta\mathbf{p}$ for the parameter vector at each iteration (line 11), which requires a matrix-vector product involving the full matrix \mathbf{G} . Interestingly, this approach for estimating the physical property distribution on an equivalent layer is the same originally proposed by Bott (1960) for estimating the basement relief under sedimentary basins. The methods of Xia and Sprowl (1991) and Siqueira et al. (2017) were originally proposed for processing gravity data, but can be potentially applied to any harmonic function because they actually represent iterative solutions of the classical *Dirichlet's problem* or the *first boundary value problem of potential theory* (Kellogg, 1967, p. 236) on a plane.

Recently, Jirigalatu and Ebbing (2019) presented another iterative method for estimating a parameter vector $\tilde{\mathbf{p}}$ (equation 3). With the purpose of combining different potential-field data, their method basically modifies that shown in Algorithm 6 by changing the initial approximation and the iterative correction for the parameter vector. Specifically, Jirigalatu and Ebbing (2019) replace line 5 by $\tilde{\mathbf{p}} = \mathbf{0}$, where $\mathbf{0}$ is a vector

of zeros, and line 11 by $\Delta \mathbf{p} = \omega \mathbf{G}^\top \mathbf{r}$, where ω is a positive scalar defined by trial and error. Note that this modified approach requires two matrix-vector products involving the full matrix \mathbf{G} per iteration. To overcome the high computational cost of these two products, Jirigalatu and Ebbing (2019) set an equivalent layer formed by prisms and compute their predicted potential field in the wavenumber domain by using the Gauss-FFT technique Zhao et al. (2018).

The iterative method proposed by Siqueira et al. (2017) (Algorithm 6) requires one entrywise product in line 5 and a matrix-vector followed by subtraction in line 7 before the while loop. At each iteration, there is another entrywise product (line 11), a half saxpy (line 12) and a saxpy (lines 11 and 12). Then, we get from table 1 that the total number of flops is given by:

$$f_{\text{SOB17}} = 2D^2 + 2D + \text{ITMAX} (2D^2 + 3D) . \quad (50)$$

Note that the number of flops per iteration in f_{SOB17} (equation 50) has the same order of magnitude, but is smaller than that in f_{CGLS} (equation 28).

7.7 Iterative deconvolution

Recently, Takahashi et al. (2020, 2022) proposed the *convolutional equivalent-layer method*, which explores the structure of the sensitivity matrix \mathbf{G} (equation 3) for the particular case in which (i) there is a single equivalent source right below each potential-field datum and (ii) both data and sources rely on planar and regularly spaced grids. Specifically, they consider a regular grid of D potential-field data at points (x_i, y_i, z_0) , $i \in \{1 : D\}$, on a horizontal plane z_0 . The data indices i may be ordered along the x - or y -direction, which results in an x - or y -oriented grid, respectively. They also consider a single equivalent source located right below each datum, at a constant vertical coordinate $z_0 + \Delta z$, $\Delta z > 0$. In this case, the number of data and equivalent sources are equal to each other (i.e., $D = P$) and \mathbf{G} (equation 3) assumes a *doubly block Toeplitz* (Jain, 1989, p. 28) or *block-Toeplitz Toeplitz-block* (BTTB) (Chan and Jin, 2007, p. 67) structure formed by $N_B \times N_B$ blocks, where each block has $N_b \times N_b$ elements, with $D = N_B N_b$. This particular structure allows formulating the product of \mathbf{G} and an arbitrary vector as a *fast discrete convolution* via *Fast Fourier Transform* (FFT) (Van Loan, 1992, section 4.2).

Consider, for example, the particular case in which $N_B = 4$, $N_b = 3$ and $D = 12$. In this case, \mathbf{G} (equation 3) is a 12×12 block matrix given by

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}^0 & \mathbf{G}^1 & \mathbf{G}^2 & \mathbf{G}^3 \\ \mathbf{G}^{-1} & \mathbf{G}^0 & \mathbf{G}^1 & \mathbf{G}^2 \\ \mathbf{G}^{-2} & \mathbf{G}^{-1} & \mathbf{G}^0 & \mathbf{G}^1 \\ \mathbf{G}^{-3} & \mathbf{G}^{-2} & \mathbf{G}^{-1} & \mathbf{G}^0 \end{bmatrix}_{D \times D}, \quad (51)$$

where each block \mathbf{G}^n , $n \in \{(1 - N_B) : (N_B - 1)\}$, is a 3×3 Toeplitz matrix. Takahashi et al. (2020, 2022) have deduced the specific relationship between blocks \mathbf{G}^n and \mathbf{G}^{-n} and also between a given block \mathbf{G}^n and its transposed $(\mathbf{G}^n)^\top$ according to the harmonic function g_{ij} (equation 2) defining the element ij of the sensitivity matrix \mathbf{G} (equation 3) and the orientation of the data grid.

Consider the matrix-vector products

$$\mathbf{G} \mathbf{v} = \mathbf{w} \quad (52)$$

and

$$\mathbf{G}^\top \mathbf{v} = \mathbf{w}, \quad (53)$$

involving a $D \times D$ sensitivity matrix \mathbf{G} (equation 3) defined in terms of a given harmonic function g_{ij} (equation 2), where

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}^0 \\ \vdots \\ \mathbf{v}^{N_B-1} \end{bmatrix}_{D \times 1}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}^0 \\ \vdots \\ \mathbf{w}^{N_B-1} \end{bmatrix}_{D \times 1}, \quad (54)$$

are arbitrary partitioned vectors formed by N_B sub-vectors \mathbf{v}^n and \mathbf{w}^n , $n \in \{0 : (N_B - 1)\}$, all of them having N_b elements. Equations 52 and 53 can be computed in terms of an auxiliary matrix-vector product

$$\mathbf{G}_c \mathbf{v}_c = \mathbf{w}_c, \quad (55)$$

where

$$\mathbf{v}_c = \begin{bmatrix} \mathbf{v}_c^0 \\ \vdots \\ \mathbf{v}_c^{N_B-1} \\ \mathbf{0} \end{bmatrix}_{4D \times 1}, \quad \mathbf{w}_c = \begin{bmatrix} \mathbf{w}_c^0 \\ \vdots \\ \mathbf{w}_c^{N_B-1} \\ \mathbf{0} \end{bmatrix}_{4D \times 1}, \quad (56)$$

are partitioned vectors formed by $2N_b \times 1$ sub-vectors

$$\mathbf{v}_c^n = \begin{bmatrix} \mathbf{v}^n \\ \mathbf{0} \end{bmatrix}_{2N_b \times 1}, \quad \mathbf{w}_c^n = \begin{bmatrix} \mathbf{w}^n \\ \mathbf{0} \end{bmatrix}_{2N_b \times 1}, \quad (57)$$

and \mathbf{G}_c is a $4D \times 4D$ doubly block circulant (Jain, 1989, p. 28) or block-circulant circulant-block (BCCB) (Chan and Jin, 2007, p. 76) matrix. What follows aims at explaining how the original matrix-vector products defined by equations 52 and 53, involving a $D \times D$ BTTB matrix \mathbf{G} exemplified by equation 51, can be efficiently computed in terms of the auxiliary matrix-vector product given by equation 55, which has a $4D \times 4D$ BCCB matrix \mathbf{G}_c .

Matrix \mathbf{G}_c (equation 55) is formed by $2N_B \times 2N_B$ blocks, where each block \mathbf{G}_c^n , $n \in \{(1 - N_B) : (N_B - 1)\}$ is a $2N_b \times 2N_b$ circulant matrix. For the case in which the original matrix-vector product is that defined by equation 52, the first column of blocks forming the BCCB matrix \mathbf{G}_c is given by

$$\mathbf{G}_c[:, : 2N_b] = \begin{bmatrix} \mathbf{G}_c^0 \\ \mathbf{G}_c^{-1} \\ \vdots \\ \mathbf{G}_c^{1-N_B} \\ \mathbf{0} \\ \mathbf{G}_c^{N_B-1} \\ \vdots \\ \mathbf{G}_c^1 \end{bmatrix}_{4D \times 2N_b}, \quad (58)$$

with blocks \mathbf{G}_c^n having the first column given by

$$\mathbf{G}_c^n[:, 1] = \begin{bmatrix} \mathbf{G}^n[:, 1] \\ 0 \\ (\mathbf{G}^n[1, N_b : 2])^\top \end{bmatrix}_{2N_b \times 2N_b}, \quad n \in \{(1 - N_B) : (N_B - 1)\}, \quad (59)$$

where \mathbf{G}^n are the blocks forming the BTTB matrix \mathbf{G} (equation 51). For the case in which the original matrix-vector product is that defined by equation 53, the first column of blocks forming the BCCB matrix \mathbf{G}_c is given by

$$\mathbf{G}_c[:, : 2N_b] = \begin{bmatrix} \mathbf{G}_c^0 \\ \mathbf{G}_c^1 \\ \vdots \\ \mathbf{G}_c^{N_B-1} \\ \mathbf{0} \\ \mathbf{G}_c^{1-N_B} \\ \vdots \\ \mathbf{G}_c^{-1} \end{bmatrix}_{4D \times 2N_b}, \quad (60)$$

with blocks \mathbf{G}_c^n having the first column given by

$$\mathbf{G}_c^n[:, 1] = \begin{bmatrix} (\mathbf{G}^n[1, :])^\top \\ 0 \\ \mathbf{G}^n[N_b : 2, 1] \end{bmatrix}_{2N_b \times 2N_b}, \quad n \in \{(1 - N_B) : (N_B - 1)\}. \quad (61)$$

The complete matrix \mathbf{G}_c (equation 55) is obtained by properly downshifting the block columns $\mathbf{G}_c[:, : 2N_b]$ defined by equation 58 or 60. Similarly, the n -th block \mathbf{G}_c^n of \mathbf{G}_c is obtained by properly downshifting the first columns $\mathbf{G}_c^\ell[:, 1]$ defined by equation 59 or 61.

Note that \mathbf{G}_c (equation 55) is a $4D \times 4D$ matrix and \mathbf{G} (equation 51) is a $D \times D$ matrix. It seems weird to say that computing $\mathbf{G}_c \mathbf{v}_c$ is more efficient than directly computing $\mathbf{G} \mathbf{v}$. To understand this, we need first to use the fact that BCCB matrices are diagonalized by the 2D unitary discrete Fourier transform (DFT) (e.g., Davis, 1979, p. 31). Because of that, \mathbf{G}_c can be written as

$$\mathbf{G}_c = (\mathcal{F}_{2N_B} \otimes \mathcal{F}_{2N_b})^* \Lambda (\mathcal{F}_{2N_B} \otimes \mathcal{F}_{2N_b}), \quad (62)$$

where the symbol “ \otimes ” denotes the Kronecker product (e.g., Horn and Johnson, 1991, p. 243), \mathcal{F}_{2N_B} and \mathcal{F}_{2N_b} are the $2N_B \times 2N_B$ and $2N_b \times 2N_b$ unitary DFT matrices (e.g., Davis, 1979, p. 31), respectively, the superscript “ $*$ ” denotes the complex conjugate and Λ is a $4D \times 4D$ diagonal matrix containing the eigenvalues of \mathbf{G}_c . Due to the diagonalization of the matrix \mathbf{G}_c , equation 55 can be rewritten by using equation 62 and premultiplying both sides of the result by $(\mathcal{F}_{2N_B} \otimes \mathcal{F}_{2N_b})$, i.e.,

$$\Lambda (\mathcal{F}_{2N_B} \otimes \mathcal{F}_{2N_b}) \mathbf{v}_c = (\mathcal{F}_{2N_B} \otimes \mathcal{F}_{2N_b}) \mathbf{w}_c. \quad (63)$$

By following Takahashi et al. (2020), we rearrange equation 63 as follows

$$\mathcal{L} \circ (\mathcal{F}_{2N_B} \mathbf{v}_c \mathcal{F}_{2N_b}) = \mathcal{F}_{2N_B} \mathbf{w}_c \mathcal{F}_{2N_b} \quad (64)$$

where “ \circ ” denotes the Hadamard product (e.g., Horn and Johnson, 1991, p. 298) and \mathcal{L} , \mathbf{v}_c and \mathbf{w}_c are $2N_B \times 2N_b$ matrices obtained by rearranging, along their rows, the elements forming the diagonal of Λ (equation 62), vector \mathbf{v}_c and vector \mathbf{w}_c (equation 56), respectively. Then, by premultiplying both sides of equation 64 by $\mathcal{F}_{2N_B}^*$ and then postmultiplying both sides by $\mathcal{F}_{2N_b}^*$, we obtain

$$\mathcal{F}_{2N_B}^* [\mathcal{L} \circ (\mathcal{F}_{2N_B} \mathbf{v}_c \mathcal{F}_{2N_b})] \mathcal{F}_{2N_b}^* = \mathbf{w}_c. \quad (65)$$

Finally, we get from equation 62 that matrix \mathcal{L} can be computed by using only the first column $\mathbf{G}_c[:, 1]$ of the BCCB matrix \mathbf{G}_c (equation 55) according to (Takahashi et al., 2020)

$$\mathcal{L} = \sqrt{4D} \mathcal{F}_{2N_B} \mathcal{C} \mathcal{F}_{2N_b}^*, \quad (66)$$

where \mathcal{C} is a $2N_B \times 2N_b$ matrix obtained by rearranging, along its rows, the elements of $\mathbf{G}_c[:, 1]$ (equation 55). It is important noting that the matrices \mathcal{C} and \mathcal{L} (equation 66) associated with the BTTB matrix \mathbf{G} (equation 51) are different from those associated with \mathbf{G}^\top .

The whole procedure to compute the original matrix-vector products $\mathbf{G}\mathbf{v}$ (equation 52) and $\mathbf{G}^\top \mathbf{v}$ (equation 53) consists in (i) rearranging the elements of the vector \mathbf{v} and the first column $\mathbf{G}[:, 1]$ of matrix \mathbf{G} into the matrices \mathcal{V}_c and \mathcal{C} (equations 65 and 66), respectively; (ii) computing terms $\mathcal{F}_{2N_B} \mathcal{A} \mathcal{F}_{2N_b}^*$ and $\mathcal{F}_{2N_B}^* \mathcal{A} \mathcal{F}_{2N_b}$, where \mathcal{A} is a given matrix, and a Hadamard product to obtain \mathcal{W}_c (equation 65); and (iii) retrieve the elements of vector \mathbf{w} (equation 52) from \mathcal{W}_c (equation 65). It is important noting that the steps (i) and (iii) do not have any computational cost because they involve only reorganizing elements of vectors and matrices. Besides, the terms $\mathcal{F}_{2N_B} \mathcal{A} \mathcal{F}_{2N_b}^*$ and $\mathcal{F}_{2N_B}^* \mathcal{A} \mathcal{F}_{2N_b}$ in step (ii) represent, respectively, the 2D Discrete Fourier Transform (2D-DFT) and the 2D Inverse Discrete Fourier Transform (2D-IDFT) of \mathcal{A} . These transforms can be efficiently computed by using the 2D Fast Fourier Transform (2D-FFT). Hence, the original matrix-vector products $\mathbf{G}\mathbf{v}$ (equation 52) and $\mathbf{G}^\top \mathbf{v}$ (equation 53) can be efficiently computed by using the 2D-FFT.

Algorithms 7 and 8 show pseudo-codes for the convolutional equivalent-layer method proposed by Takahashi et al. (2020, 2022). Note that those authors formulate the overdetermined problem (equation 22) of obtaining an estimate $\tilde{\mathbf{p}}$ for the parameter vector \mathbf{p} (equation 3) as an *iterative deconvolution* via *conjugate gradient normal equation residual* (CGNR) Golub and Van Loan (2013, sec. 11.3) or *conjugate gradient least squares* (CGLS) (Aster et al., 2019, p. 165) method. They consider $\mathbf{H} = \mathbf{I}_P$ (equation 9), $\mu = 0$ (equation 11), $\mathbf{W}_d = \mathbf{W}_q = \mathbf{I}_P$ (equations 12 and 13) and $\tilde{\mathbf{p}} = \mathbf{0}$ (equation 14). As shown by Takahashi et al. (2020, 2022), the CGLS produces stable estimates $\tilde{\mathbf{p}}$ for the parameter vector \mathbf{p} (equation 3) in the presence of noisy potential-field data \mathbf{d} . This is a well-known property of the CGLS method (e.g., Aster et al., 2019, p. 166).

The key aspect of Algorithm 7 is replacing the matrix-vector products of CGLS (Algorithm 1) by fast convolutions (Algorithm 8). A fast convolution requires one 2D-DFT, one 2D-IDFT and an entrywise product of matrices. We consider that the 2D-DFT/IDFT are computed with 2D-FFT and requires $\kappa(4D) \log_2(4D)$ flops, where $\kappa = 5$ is compatible with a radix-2 FFT (Van Loan, 1992, p. 16), and the entrywise product $24D$ flops because it involves two complex matrices having $4D$ elements (Golub and Van Loan, 2013, p. 36). Hence, Algorithm 8 requires $\kappa(16D) \log_2(4D) + 26D$ flops, whereas a conventional matrix-vector multiplication involving a $D \times D$ matrix requires $2D^2$ (table 1). Finally, Algorithm 7 requires two 2D-FFTs (lines 4 and 5), one fast convolution and an inner product (line 8) previously to the while loop. Per iteration, there are three saxpys (lines 12, 15 and 16), two inner products (lines 14 and 17) and two fast convolutions (lines 13 and 17), so that:

$$f_{\text{TOB20}} = \kappa(16D) \log_2(4D) + 26D + \text{ITMAX} [\kappa(16D) \log_2(4D) + 58D] . \quad (67)$$

7.8 Direct deconvolution

The method proposed by Takahashi et al. (2020, 2022) can be reformulated to avoid the iterations of the conjugate gradient method. This alternative formulation consists in considering that $\mathbf{v} = \mathbf{p}$ and $\mathbf{w} = \mathbf{d}$ in

equation 52, where \mathbf{p} is the parameter vector (equation 3) and \mathbf{d} the observed data vector. In this case, the equality “=” in equation 52 becomes an approximation “ \approx ”. Then, equation 64 is manipulated to obtain

$$\mathbf{V}_c \approx \mathcal{F}_{2N_B}^* \left[\left(\mathcal{F}_{2N_B} \mathcal{W}_c \mathcal{F}_{2N_b} \right) \circ \check{\mathcal{L}} \right] \mathcal{F}_{2N_b}^*, \quad (68)$$

where

$$\check{\mathcal{L}} = \mathcal{L}^* \oslash (\mathcal{L} \circ \mathcal{L}^* + \zeta \mathbf{1}), \quad (69)$$

$\mathbf{1}$ is a $4D \times 4D$ matrix of ones, “ \oslash ” denotes entrywise division and ζ is a positive scalar. Note that $\zeta = 0$ leads to $\mathbf{1} \oslash \mathcal{L}$. In this case, the entrywise division may be problematic due to the elements of \mathcal{L} having absolute value equal or close to zero. So, a small ζ is set to avoid this problem in equation 69. Next, we use $\check{\mathcal{L}}$ to obtain a matrix \mathbf{V}_c from equation 68. Finally, the elements of the estimated parameter vector $\tilde{\mathbf{p}}$ are retrieved from the first quadrant of \mathbf{V}_c . This procedure represents a *direct deconvolution* (e.g., Aster et al., 2019, p. 220) using a *Wiener filter* (e.g., Gonzalez and Woods, 2002, p. 263).

The required total number of flops associated with the direct deconvolution aggregates one 2D-FFT to compute matrix \mathcal{L} (equation 66), one entrywise product $\mathcal{L} \circ \mathcal{L}^*$ involving complex matrices and one entrywise division to compute $\check{\mathcal{L}}$ (equation 69) and a fast convolution (Algorithm 8) to evaluate equation 68, which results in:

$$f_{\text{deconv.}} = \kappa (12D) \log_2(4D) + 72D. \quad (70)$$

Differently from the convolutional equivalent-layer method proposed by Takahashi et al. (2020, 2022), the alternative direct deconvolution presented here produces an estimated parameter vector $\tilde{\mathbf{p}}$ directly from the observed data \mathbf{d} , in a single step, avoiding the conjugate gradient iterations. On the other hand, the alternative method presented here requires estimating a set of tentative parameter vectors $\tilde{\mathbf{p}}$ for different predefined ζ . Besides, there must be criterion to choose the best $\tilde{\mathbf{p}}$ from this tentative set. This can be made, for example, by using the well-known *L-curve* (Hansen, 1992). From a computational point of view, the number of CGLS iterations in the method proposed by Takahashi et al. (2020, 2022) is equivalent to the number of tentative estimated parameter vectors required to form the L-curve in the proposed direct deconvolution.

8 NUMERICAL SIMULATIONS

8.1 Flops count

Figure 1 shows the total number of flops for solving the overdetermined problem (equation 22) with different equivalent-layer methods (equations 27, 28, 35, 37, 38, 43, 50, 67, and 70), by considering the particular case in which $\mathbf{H} = \mathbf{I}_P$ (equation 9 and subsection 3.2), $\mu = 0$ (equation 11), $\mathbf{W}_d = \mathbf{I}_D$ (equation 12) and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14), where \mathbf{I}_P and \mathbf{I}_D are the identities of order P and D , respectively. The flops are computed for different number of potential-field data ranging from 10, 000 to 1, 000, 000.

FALTOU DEFINIR OS PARÂMETROS ESPECÍFICOS DE CADA MÉTODO

8.2 Synthetic potential-field data

We create a model composed of three synthetic bodies: a sphere centered at $(x, y, z) = (3, -2, 2)$ km, with radius of 1 km; a sphere centered at $(x, y, z) = (1, 2.5, 1.8)$ km, with radius of 750 m; and a right prism having polygonal horizontal cross-section, top at $z = 900$ m and thickness 600 m. The density contrasts of the upper left sphere, upper right sphere and prism are, respectively, 600 kg/m^3 , -500 kg/m^3 and to 550 kg/m^3 . All synthetic bodies have a total-magnetization vector with intensity 3.46 A/m , inclination 35.26° and declination 45.0° . We consider a main geomagnetic field with constant inclination 20.0° and declination 35.0° .

We have computed noise-free gravity disturbance and total-field anomaly data \mathbf{d} produced by the model on the same regularly spaced grid of 50×50 points at $z = 50$ m (Figures 2A and 3A). We have also simulated additional $L = 20$ gravity data sets \mathbf{d}^ℓ , $\ell \in \{1 : L\}$, by adding pseudo-random Gaussian noise with zero mean and crescent standard deviations to the noise-free data (Figure 1A). The standard deviations vary from 0.5% to 10% of the maximum absolute value in the noise-free data. We applied the same procedure to produce additional 20 noise-corrupted magnetic data sets from the noise-free data shown in Figure 3A. Figures 2B and 3B show, respectively, the gravity disturbance and total-field anomaly data corrupted with maximum noise level. The remaining noise-corrupted gravity and magnetic data are not shown.

8.3 Stability analysis

We set two planar equivalent layers having one source below each datum at a constant vertical coordinate $z = 300$ m. Note that, in this case, both layers have a number of sources P equal to the number of data D . One layer is formed by point masses and is applied to the synthetic gravity data. The other is applied to the synthetic magnetic data and is composed of dipoles.

We have applied the Cholesky factorization (equations 25 and 26), the iterative deconvolution (Algorithms 7 and 8) proposed by Takahashi et al. (2020, 2022) and the direct deconvolution (equations 68 and 69) with four different values for the parameter ζ to the 21 gravity and 21 magnetic data sets.

For each method, we have obtained one estimate $\tilde{\mathbf{p}}$ from the noise-free gravity data \mathbf{d} and $L = 20$ estimates $\tilde{\mathbf{p}}^\ell$ from the noise-corrupted gravity data \mathbf{d}^ℓ , $\ell \in \{1 : L\}$, for the planar equivalent layer of point masses, totaling 21 estimated parameter vectors and 20 pairs $(\Delta p^\ell, \Delta d^\ell)$ of model and data perturbations (equations 30 and 31). Other 21 estimates for the parameter vector and 20 pairs $(\Delta p^\ell, \Delta d^\ell)$ were obtained in the same way for the equivalent layer of dipoles by using the synthetic magnetic data. Figures 4 and 5 show the numerical stability curves computed with each method for synthetic gravity and magnetic data, respectively.

780 All these 42 estimated parameters vectors (21 for gravity and 21 for magnetic data) were obtained by
 781 solving the overdetermined problem (equation 22) with the same method for the particular case in which
 782 $\mathbf{H} = \mathbf{I}$ (equation 9 and subsection 3.2), $\mathbf{W}_d = \mathbf{W}_q = \mathbf{I}$ (equations 12 and 13) and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14),
 783 where \mathbf{I} is the identity of order D .

784 COMENTARIOS SOBRE AS CURVAS DE ESTABILIDADE (NECESSITA REVISÃO) \Rightarrow

785 Figure 4 shows how the euclidian norm of the equivalent sources residuals varies as the level of the noise
 786 is increased for the gravimetric data. We can see that for all methods, a linear tendency can be observed as
 787 it is expected. The inclination of the straight line is a indicative of the stability of each method. As show
 788 in the graph the deconvolutional method is very unstable and it is really necessary to use a stabilization
 789 method to have a good parameter estimative. In contrast, a correct value of the stabilization parameter is
 790 necessary to not overshoot the smootheness of the solution as it is the case for the zeroth-order Tikhonov
 791 regularization as well. Using this gravimetric data, the optimal value for the Wiener stabilization parameter
 792 is $\mu = 10^{-20}$.

793 For the magnetic data, figure 5 shows a very similar behavior of the stability as the previous case. The
 794 Wiener parameter seems to have the best solution for $\mu = 10^{-13}$. For both types of data the best Wiener
 795 parameter seems to be one that produces a low slope for the straight line in the stability analysis, discordant
 796 from the classical and convolutional methods.

797 \Leftarrow COMENTARIOS SOBRE AS CURVAS DE ESTABILIDADE (NECESSITA REVISÃO)

798 COMENTARIOS SOBRE OS DADOS PREDITOS (NECESSITA REVISÃO) \Rightarrow

799 Figure 6 shows the comparison of the predicted data for each method with the original data (figure 2)
 800 using the most noised-corrupted data from the set of the stability analysis. The classical with zeroth-order
 801 Tikhonov regularization and the convolutional methods (figures 6(A) and 6(B)) yield very similar results for
 802 the predicted data confirming its similarities with the stabilization despite the bid difference in floating-point
 803 operations. Figure 6(C) shows the deconvolutional method without a stabilization and demonstrates the
 804 necessity to use it for this method. Figure 6(D) shows the deconvolutional method with Wiener stabilization
 805 $\mu = 10^{-15}$ which is too high, demonstrating the over smoothness of the predicted data. Figures 6(E) and
 806 6(F) shows the predicted data for an optimal value of the Wiener parameter $\mu = 10^{-20}$ and a low value
 807 $\mu = 10^{-25}$, respectively.

808 Figure 7 shows the comparison of the predicted data for each method with the original magnetic data in
 809 figure 3 using the most noised-corrupted data modeled from the stability analysis. As the previous case the
 810 classical (figure 7(A)) and the convolutional (figure 7(B)) methods have very similar predicted data but
 811 estimated with less orders of magnitude in floating-point operations. The deconvolutional (figure 7(C)) have
 812 have a strong disagreement with the observed data showing the need for a stabilization method. Figure
 813 7(D) has a value of $\mu = 10^{-10}$ and the predicted data became to smooth by it. The optimal value of the
 814 Wiener parameter is shown in figure 7(E) with $\mu = 10^{-13}$ and figure 7(F) shows a predicted data with a
 815 low stabilization value with $\mu = 10^{-16}$.

816 \Leftarrow COMENTARIOS SOBRE OS DADOS PREDITOS (NECESSITA REVISÃO)

817 COMENTARIOS SOBRE AS TRANSFORMAÇÕES (NECESSITA REVISÃO) \Rightarrow

818 The upward continuation is a processing technique to visualize the data in a higher altitude. In practice is
 819 expected a lower amplitude signal and a smoother data as the high frequency anomalies tends to disappear.
 820 Figure 8(A) shows the true modeled upward data at an height of -500 m. Figures 8(B), (C), (D) and

821 (E) show the result of the upward processing for the classical, convolutional, deconvolutional and the
822 deconvolutional with Wiener parameter $\mu = 10^{-20}$, respectively. It is clear that all methods seems to
823 predict the upward data very reasonable, except the deconvolutional method without stabilization.

824 Figure 9(A) shows the true modeled upward data at an height of -1400 m. Figures 9(B), (C), (D) and
825 (E) show the result of the upward processing for the classical, convolutional, deconvolutional and the
826 deconvolutional with Wiener parameter $\mu = 10^{-13}$, respectively. As in the gravimetric case, all methods
827 seems to predict the upward data, except the deconvolutional method without stabilization.

828 \Leftarrow COMENTARIOS SOBRE AS TRANSFORMAÇÕES (NECESSITA REVISÃO)

9 REAL DATA RESULTS

In this section, we show the applications of the convolutional and the deconvolutional strategies in a real data set from the North of Brazil. The region is located in the Carajás Mineral Province (CMP) in the Amazon craton (Moroni et al., 2001; Villas and Santos, 2001). This area is known for its intensive mineral exploration such as iron, copper, gold, manganese, and, recently, bauxite.

9.1 Geological setting

The Amazon craton is one of the largest and least-known Archean-Proterozoic areas in the world, comprehending a region with a thousand square kilometers. It is one of the main tectonic units in South America, which is covered by five Phanerozoic basins: Maranhão (Northeast), Amazon (Central), Xingu-Alto Tapajós (South), Parecis (Southwest), and Solimões (West). The craton is limited by the Andean Orogenic Belt to the west and the by Araguaia Fold Belt to the east and southeast. The Amazon craton has been subdivided into provinces according to two models, one geochronological and the other geophysical-structural (Amaral, 1974; Teixeira et al., 1989; Tassinari and Macambira, 1999). Thus, seven geological provinces with distinctive ages, evolution, and structural patterns can be observed, namely : (i) Carajás with two domains - the Mesoarchean Rio Maria and Neoarchean Carajás; (ii) Archean-Paleoproterozoic Central Amazon, with Iriri-Xingu and Curuá-Mapuera domains; (iii) Trans-Amazonian (Ryacian), with the Amapá and Bacajá domains; (iv) the Orosinian Tapajós-Parima, with Peixoto de Azevedo, Tapajós, Uaimiri, and Parima domains; (v) Rondônia-Juruena (Statherian), with Jamari, Juruena, and Jauru domains; (vi) The Statherian Rio Negro, with Rio Negro and Imeri domains; and (vii) Sunsás (Meso-Neoproterozoic), with Santa Helena and Nova Brasilândia domains (Santos et al., 2000). Nevertheless, we focus this work only on the Carajás Province.

The Carajás Mineral Province (CMP) is located in the east-southeast region of the craton, within an old tectonically stable nucleus in the South American Plate that became tectonically stable at the beginning of Neoproterozoic (Salomao et al., 2019). This area has been the target of intensive exploration at least since the final of the '60s, after the discovery of large iron ore deposits. There are several greenstone belts in the region, among them are the Andorinhas, Inajá, Cumaru, Carajás, Serra Leste, Serra Pelada, and Sapucaia (Santos et al., 2000). The mineralogic and petrologic studies in granite stocks show a variety of minerals found in the province, such as amphibole, plagioclase, biotite, ilmenite, and magnetite (Cunha et al., 2016). These two latter minerals contribute to the high magnetic response in the CMP area. This fact opens the opportunity for potential field applications for the geophysical description of the area.

9.2 Potential field data applications

Here we compare the performance of the convolutional and deconvolutional algorithms in a real potential field data set. We focus the application on a region in the Southeast of the State of Pará. The aeromagnetic data were acquired by the Geological Survey of Brazil-CPRM. The survey area covers $\approx 58000 \text{ km}^2$ with high-resolution gravity and magnetic data. The flight and the tie lines were acquired and spaced at 3 km and 12 km oriented in the directions $N - S$ and $E - W$, respectively, with a mean flight height of 900 m above the ground. For both applications, we interpolated gravity and magnetic anomalies data, calculating the data set in a grid of 1000×500 ($N = 500000$ observation points) at the same mean flight height. About the computational resources, we processed both data on an Intel Core i7 7700HQ@2.8 GHz processor and 16GB RAM. We show in Figure ?? and Figure ?? the interpolated aerogravimetric and aeromagnetic data, respectively. We also use the same equivalent layer grid configuration in gravity and magnetic applications.

869 This setup is composed by a grid of 1000×500 equivalent sources (a total number of $M = 500000$ points)
870 positioned below the observation plane, but a different depth in each application.

871 We apply both strategies to the gravimetric case. We set a depth for the equivalent layer equal to 1200 m
872 below the observation plane. Figure ??A and Figure ??C show the predicted data for convolutional and
873 deconvolutional strategies. The residual maps (the difference between the observed and predicted data)
874 are show in figures ??B and ??D for the convolutional and deconvolutional equivalent-layer technique,
875 respectively. For the convolutional case, the mean residual and standard deviation values are $\approx 0.00\text{ mGal}$
876 and $\approx 0.15\text{ mGal}$, respectively. For the deconvolutional case, the mean residual and standard deviation
877 values are $\approx 0.46\text{ mGal}$ and $\approx 1.23\text{ mGal}$, respectively. These last results show that the estimated density
878 distributions (not shown) fit the observed data for both applications. To show the performance of the
879 algorithms, we performed an upward continuation by using the estimated density distributions (figures ??E
880 and ??F). There is a little difference on the processing time between both strategies. The convolutional
881 algorithm took $\approx 9.18\text{ s}$ and the deconvolutional algorithm took $\approx 0.53\text{ s}$. We conclude that both strategies
882 are capable of processing gravimetric observations from large areas with dense coverage data. Despite a
883 little difference in processing time, the deconvolutional equivalent-layer technique proved to be faster than
884 the convolutional strategy.

885 Finally, we test the convolutional and deconvolutional algorithms for processing total-field anomalies.
886 We stress that the Carajás area is very large and the main field direction varies significantly. For this reason,
887 we consider a mean direction for the main field equal to -19.865° and -7.43915° (the same as the mid
888 location of the area) for the inclination and declination, respectively. Furthermore, we are not considering
889 the knowledge about the magnetization direction of the sources, and choose a magnetization direction for
890 the equivalent layer equal to the main field direction. For this application, we set a depth of 900 m (below
891 the observation plane) for the equivalent layer. Figure ??A and Figure ??C show the predicted data for
892 convolutional and deconvolutional algorithms. The residual maps (the difference between the observed
893 and predicted data) are show in figures ??B and ??D for the convolutional and deconvolutional techniques,
894 respectively. The convolutional equivalent layer produced a mean residual and standard deviation values
895 of $\approx 0.06\text{ nT}$ and $\approx 1.97\text{ nT}$, respectively. The deconvolutional algorithm produced a mean residual
896 and standard deviation values of $\approx 18.99\text{ nT}$ and $\approx 33.64\text{ nT}$, respectively. To show the performance
897 of the algorithms, we performed an upward continuation (figures ??E and ??F) by using the estimated
898 magnetic-moment distributions (not shown). Similarly to the gravity application, the deconvolutional
899 equivalent layer presents faster results than the convolutional algorithm. The deconvolutional and the
900 convolutional approaches took $\approx 0.89\text{ s}$ and $\approx 82.08\text{ s}$, respectively. Despite the difference between the
901 processing time of both strategies and considering the mean value of residuals and standard deviations, we
902 conclude that the convolutional strategy fits the observation data better than the deconvolutional approach.

10 DISCUSSION AND CONCLUSION

903 We have presented a review of the strategies used to overcome the intensive computational cost of the
904 equivalent-layer technique for processing potential-field data. Each of these strategies is rarely used
905 individually; rather, some developed equivalent-layer methods combine more than one strategy to make
906 then computationally efficient in handling large-scale data sets. This comprehensive review addresses the
907 following specific strategies for reducing the computational cost of equivalent-layer technique.

908 The first one is the moving data-window scheme spanning the data set. This strategy solves several much
909 smaller, regularized linear inverse problems instead of a single large one. Each linear inversion is solved
910 using the potential-field observations and equivalent sources within a given moving window and can be
911 applied to both regularly or irregularly spaced data sets. If the data and the sources are distributed on planar
912 and regularly spaced grids, this strategy offers a significant advantage because the sensitivity submatrix of
913 a given moving window remains the same for all windows. Otherwise, the computational efficiency of the
914 equivalent-layer technique using the moving-window strategy decreases because the sensitivity submatrix
915 for each window must be computed.

916 The second and third strategies, referred to as the column-action and row-action updates, involve
917 iteratively calculating a single column and a single row of the sensitivity matrix, respectively. By following
918 the column-action update strategy, a single column of the sensitivity matrix is calculated during each
919 iteration. This implies that a single equivalent source contributes to the fitting of data in each iteration.
920 Conversely, in the row-action update strategy, a single row of the sensitivity matrix is calculated per
921 iteration, which means that one potential-field observation is incorporated in each iteration, forming a new
922 subset of equivalent data much smaller than the original data. Both strategies (column- and row-action
923 updates) have a great advantage because a single column or a single row of the sensitivity matrix is
924 calculated iteratively. However, to our knowledge, the strategy of the column-action update presents some
925 issues related to convergence, and the strategy of the row-action update can also have issues if the number
926 of equivalent data is not significantly smaller than the original number of data points.

927 The fourth strategy is the sparsity induction of the sensitivity matrix using wavelet compression, which
928 involves transforming a full sensitivity matrix into a sparse one with only a few nonzero elements. The
929 developed equivalent-layer methods using this strategy achieve sparsity by setting matrix elements to
930 zero if their values are smaller than a predefined threshold. We highlight two methods that employ the
931 sparsity induction strategy. The first method, known as wavelet-compression equivalent layer, compresses
932 the coefficients of the original sensitivity matrix using discrete wavelet transform, achieves sparsity in the
933 sensitivity matrix, and solves the inverse problem in the wavelet domain without an explicit regularization
934 parameter. The regularized solution in the wavelet domain is estimated using a conjugate gradient (CG)
935 least squares algorithm, where the number of iterations serves as a regularization factor. The second
936 equivalent-layer method that uses the sparsity induction strategy applies quadtree discretization of the
937 parameters over the equivalent layer, achieves sparsity in the sensitivity matrix, and solves the inverse
938 problem using CG algorithm. In quadtree discretization, equivalent sources located far from the observation
939 point are grouped together to form larger equivalent sources, reducing the number of parameters to be
940 estimated. Computationally, the significant advantage of the equivalent-layer methods employing wavelet
941 compression and quadtree discretization is the sparsity induction in the sensitivity matrix, which allows
942 for fast iteration of the CG algorithm. However, we acknowledge that this strategy requires computing
943 the full and dense sensitivity matrix, which can be considered a drawback when processing large-scale
944 potential-field data.

945 The fifth strategy is the reparametrization of the original parameters to be estimated in the equivalent-layer
946 technique. In this strategy, the developed equivalent-layer methods reduce the dimension of the linear
947 system of equations to be solved by estimating a lower-dimensional parameter vector. We highlight two
948 methods that used the reparametrization strategy: i) the polynomial equivalent layer (PEL) and; ii) the
949 lower-dimensional subspace of the equivalent layer. In the PEL, there is an explicit reparametrization
950 of the equivalent layer by representing the unknown distribution over the equivalent layer as a set of
951 piecewise-polynomial functions defined on a set of equivalent-source windows. The PEL method estimates
952 the polynomial coefficients of all equivalent-source windows. Hence, PEL reduces the dimension of the
953 linear system of equations to be solved because the polynomial coefficients within all equivalent-source
954 windows are much smaller than both the number of equivalent sources and the number of data points.
955 In the lower-dimensional subspace of the equivalent layer, there is an implicit reparametrization of the
956 equivalent layer by reducing the linear system dimension from the original and large-model space to a
957 lower-dimensional subspace. The lower-dimensional subspace is grounded on eigenvectors of the matrix
958 composed by the gridded data set. The main advantage of the reparametrization of the equivalent layer is to
959 deal with lower-dimensional linear system of equations. However, we acknowledge that this strategy may
960 impose an undesirable smoothing effect on both the estimated parameters over the equivalent layer and the
961 predicted data.

962 The sixth strategy involves an iterative scheme in which the estimated distribution over the equivalent
963 layer is updated iteratively. Following this strategy, the developed equivalent-layer methods differ either in
964 terms of the expression used for the estimated parameter correction or the domain utilized (wavenumber or
965 space domains). The iterative estimated correction may have a physical meaning, such as the excess mass
966 constraint. All the iterative methods are efficient as they can handle irregularly spaced data on an undulating
967 surface, and the updated corrections for the parameter vector at each iteration are straightforward, involving
968 the addition of a quantity proportional to the data residual. However, they have a disadvantage because the
969 iterative strategy requires computing the full and dense sensitivity matrix to compute the predicted and
970 residual data in all observation stations per iteration.

971 The seventh strategy is called iterative deconvolutional of the equivalent layer. This strategy deals with
972 regularly spaced grids of data stations and equivalent sources which are located at a constant height and
973 depth, respectively. Specifically, one source is placed directly below each observation station, which results
974 in sensitivity matrices with a BTTB (Block-Toeplitz Toeplitz-Block) structure. It is possible to embed the
975 BTTB matrix into a matrix of Block-Circulant Circulant-Block (BCCB) structure, which requires only
976 one equivalent source. This allows for fast matrix-vector product using a 2D fast Fourier transform (2D
977 FFT). As a result, the potential-field forward modeling can be calculated using a 2D FFT with only one
978 equivalent source required. The main advantages of this strategy are that the entire sensitivity matrices
979 do not need to be formed or stored; only their first columns are required. Additionally, it allows for a
980 highly efficient iteration of the CG algorithm. However, the iterative deconvolutional of the equivalent
981 layer requires observations and equivalent sources aligned on a horizontal and regularly-spaced grid.

982 The eighth strategy is a direct deconvolution method, which is a mathematical process very common in
983 geophysics. However, to our knowledge, direct deconvolution has never been used to solve the inverse
984 problem associated with the equivalent-layer technique. From the mathematical expressions in the iterative
985 deconvolutional equivalent layer with BTTB matrices, direct deconvolution arises naturally since it is an
986 operation inverse to convolution. The main advantage of applying the direct deconvolution strategy in
987 the equivalent layer is that it avoids, for example, the iterations of the CG algorithm. However, the direct

988 deconvolution is known to be an unstable operation. To mitigate this instability, the Wiener deconvolution
989 method can be adopted.

990 We show in this work that the computational cost of the equivalent layer can vary from up to 10^9 flops
991 depending on the method without compromising the linear system stability. The moving data-window
992 scheme and direct deconvolution are the fastest methods; however, they both have drawbacks. To be
993 computationally efficient, the moving data-window scheme and the direct deconvolution require data and
994 equivalent sources that are distributed on planar and regularly spaced grids. Moreover, they both requires
995 choosing an optimum parameter of stabilization. We stress that the direct deconvolution has an additional
996 disadvantage in terms of a higher data residual and border effects over the equivalent layer after processing.
997 These effects can be seen from the upward continuation of the real data from Carajás.

998 We draw the readers' attention to the possibility of combining more than one aforementioned strategies
999 for reducing the computational cost of the equivalent-layer technique.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

The Author Contributions section is mandatory for all articles, including articles by sole authors. If an appropriate statement is not provided on submission, a standard one will be inserted during the production process. The Author Contributions statement must describe the contributions of individual authors referred to by their initials and, in doing so, all authors agree to be accountable for the content of the work. Please see here for full authorship criteria.

FUNDING

Diego Takahashi was supported by a Post-doctoral scholarship from CNPq (grant 300809/2022-0) Valéria C.F. Barbosa was supported by fellowships from CNPq (grant 309624/2021-5) and FAPERJ (grant 26/202.582/2019). Vanderlei C. Oliveira Jr. was supported by fellowships from CNPq (grant 315768/2020-7) and FAPERJ (grant E-26/202.729/2018).

ACKNOWLEDGMENTS

We thank the brazilian federal agencies CAPES, CNPq, state agency FAPERJ and Observatório Nacional research institute and Universidade do Estado do Rio de Janeiro.

DATA AVAILABILITY STATEMENT

The datasets generated for this study can be found in the frontiers-paper Github repository link: <https://github.com/DiegoTaka/frontiers-paper>.

REFERENCES

- Amaral, G. (1974). *Geologia Pré-Cambriana da Região Amazônica*. Ph.D. thesis, Universidade de São Paulo
- Aster, R. C., Borchers, B., and Thurber, C. H. (2019). *Parameter Estimation and Inverse Problems* (Elsevier), 3 edn.
- Barbosa, V. C. F., Silva, J. B., and Medeiros, W. E. (1997). Gravity inversion of basement relief using approximate equality constraints on depths. *Geophysics* 62, 1745–1757
- Barnes, G. and Lumley, J. (2011). Processing gravity gradient data. *GEOPHYSICS* 76, I33–I47. doi:10.1190/1.3548548
- Blakely, R. J. (1996). *Potential Theory in Gravity and Magnetic Applications* (Cambridge University press)
- Bott, M. H. P. (1960). The use of Rapid Digital Computing Methods for Direct Gravity Interpretation of Sedimentary Basins. *Geophysical Journal International* 3, 63–67. doi:10.1111/j.1365-246X.1960.tb00065.x
- Chan, R. H.-F. and Jin, X.-Q. (2007). *An introduction to iterative Toeplitz solvers*, vol. 5 (SIAM)

- Cordell, L. (1992). A scattered equivalent-source method for interpolation and gridding of potential-field data in three dimensions. *Geophysics* 57, 629–636
- Cunha, I. R., Dall’Agnol, R., and Feio, G. R. L. (2016). Mineral chemistry and magnetic petrology of the archean Planalto Suite, Carajás Province – Amazonian Craton: Implications for the evolution of ferroan archean granites. *Journal of South American Earth Sciences* 67, 100–121
- Dampney, C. N. G. (1969). The equivalent source technique. *GEOPHYSICS* 34, 39–53. doi:10.1190/1.1439996
- Davis, P. J. (1979). *Circulant matrices* (John Wiley & Sons, Inc.)
- Elfving, T., Hansen, P. C., and Nikazad, T. (2017). Convergence analysis for column-action methods in image reconstruction. *Numerical Algorithms* 74, 905–924. doi:10.1007/s11075-016-0176-x
- Emilia, D. A. (1973). Equivalent sources used as an analytic base for processing total magnetic field profiles. *GEOPHYSICS* 38, 339–348. doi:10.1190/1.1440344
- Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences (Johns Hopkins University Press), 4 edn.
- Gonzalez, R. C. and Woods, R. E. (2002). *Digital Image Processing* (Prentice Hall), 2 edn.
- Gonzalez, S. P., Barbosa, V. C. F., and Oliveira Jr., V. C. (2022). Analyzing the ambiguity of the remanent-magnetization direction separated into induced and remanent magnetic sources. *Journal of Geophysical Research: Solid Earth* 127, 1–24. doi:10.1029/2022JB024151
- Guspi, F., Introcaso, A., and Introcaso, B. (2004). Gravity-enhanced representation of measured geoid undulations using equivalent sources. *Geophysical Journal International* 159, 1–8. doi:10.1111/j.1365-246X.2004.02364.x
- Guspi, F. and Novara, I. (2009). Reduction to the pole and transformations of scattered magnetic data using Newtonian equivalent sources. *GEOPHYSICS* 74, L67–L73. doi:10.1190/1.3170690
- Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Review* 34, 561–580. doi:10.1137/1034115
- Hansen, R. O. and Miyazaki, Y. (1984). Continuation of potential fields between arbitrary surfaces. *GEOPHYSICS* 49, 787–795. doi:10.1190/1.1441707
- Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis* (Cambridge University Press), 1 edn.
- Jain, A. K. (1989). *Fundamentals of Digital Image Processing* (Pearson), 1 edn.
- Jirigalatu, J. and Ebbing (2019). A fast equivalent source method for airborne gravity gradient data. *Geophysics* 84, G75–G82. doi:10.1190/GEO2018-0366.1
- Kellogg, O. D. (1967). *Foundations of Potential Theory* (Springer-Verlag), reprint from the first edition of 1929 edn.
- Kennett, B., Sambridge, M., and Williamson, P. (1988). Subspace methods for large inverse problems with multiple parameter classes. *Geophysical Journal International* 94, 237–247
- Leão, J. W. D. and Silva, J. B. C. (1989). Discrete linear transformations of potential field data. *Geophysics* 54, 497–507. doi:10.1190/1.1442676
- Li, Y., Nabighian, M., and Oldenburg, D. W. (2014). Using an equivalent source with positivity for low-latitude reduction to the pole without striation. *GEOPHYSICS* 79, J81–J90. doi:10.1190/geo2014-0134.1
- Li, Y. and Oldenburg, D. W. (2010). Rapid construction of equivalent sources using wavelets. *GEOPHYSICS* 75, L51–L59. doi:10.1190/1.3378764
- Mendonça, C. A. (2020). Subspace method for solving large-scale equivalent layer and density mapping problems. *GEOPHYSICS* 85, G57–G68. doi:10.1190/geo2019-0302.1

- Mendonça, C. A. and Silva, J. B. C. (1994). The equivalent data concept applied to the interpolation of potential field data. *Geophysics* 59, 722–732. doi:10.1190/1.1443630
- Menke, W. (2018). *Geophysical data analysis: Discrete inverse theory* (Elsevier), 4 edn.
- Moroni, M., Girardi, V., and Ferrario, A. (2001). The Serra Pelada Au-PGE deposit, Serra dos Carajás (Pará state, Brazil): geological and geochemical indications for a composite mineralising process. *Mineralium Deposita* 36, 768–785
- Oldenburg, D., McGillivray, P., and Ellis, R. (1993). Generalized subspace methods for large-scale inverse problems. *Geophysical Journal International* 114, 12–20
- Oliveira Jr., V. C., Barbosa, V. C. F., and Uieda, L. (2013). Polynomial equivalent layer. *GEOPHYSICS* 78, G1–G13. doi:10.1190/geo2012-0196.1
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical recipes: the art of scientific computing* (Cambridge University Press), 3 edn.
- Reis, A. L. A., Oliveira Jr., V. C., and Barbosa, V. C. F. (2020). Generalized positivity constraint on magnetic equivalent layers. *Geophysics* 85, 1–45. doi:10.1190/geo2019-0706.1
- Roy, A. (1962). Ambiguity in geophysical interpretation. *GEOPHYSICS* 27, 90–99. doi:10.1190/1.1438985
- Salomao, G. N., Dall’Agnol, R., Angelica, R. S., Figueiredo, M. A., Sahoo, P. K., Medeiros-Filho, C. A., et al. (2019). Geochemical mapping and estimation of background concentrations in soils of Carajás mineral province, eastern Amazonian Craton, Brazil. *Geochemistry: Exploration, Environment, Analysis* 19, 431–447
- Santos, J. O. S., Hartmann, L. A., Gaudette, H. E., Groves, D. I., McNaughton, M. J., and Fletcher, I. R. (2000). A new understanding of the provinces of the Amazon Craton based on integration of field mapping and U-Pb and Sm-Nd geochronology. *Gondwana Research* 3, 453–488
- Silva, J. B. C. (1986). Reduction to the pole as an inverse problem and its application to low-latitude anomalies. *GEOPHYSICS* 51, 369–382. doi:10.1190/1.1442096
- Siqueira, F., Oliveira Jr., V. C., and Barbosa, V. C. F. (2017). Fast iterative equivalent-layer technique for gravity data processing: A method grounded on excess mass constraint. *GEOPHYSICS* 82, G57–G69. doi:10.1190/GEO2016-0332.1
- Skilling, J. and Bryan, R. (1984). Maximum entropy image reconstruction-general algorithm. *Monthly Notices of the Royal Astronomical Society, Vol. 211, NO. 1, P. 111, 1984* 211, 111
- Soler, S. R. and Uieda, L. (2021). Gradient-boosted equivalent sources. *Geophysical Journal International* 227, 1768–1783. doi:10.1093/gji/ggab297
- Takahashi, D., Oliveira Jr., V. C., and Barbosa, V. C. (2022). Convolutional equivalent layer for magnetic data processing. *Geophysics* 87, 1–59
- Takahashi, D., Oliveira Jr., V. C., and Barbosa, V. C. F. (2020). Convolutional equivalent layer for gravity data processing. *GEOPHYSICS* 85, G129–G141. doi:10.1190/geo2019-0826.1
- Tassinari, C. C. and Macambira, M. J. (1999). Geochronological provinces of the Amazonian Craton. *Episodes* 22, 174–182
- Teixeira, W., Tassinari, C., Cordani, U. G., and Kawashita, K. (1989). A review of the geochronology of the Amazonian Craton: Tectonic implications. *Precambrian Research* 42, 213–227
- van der Sluis, A. and van der Vorst, H. A. (1987). Numerical solution of large, sparse linear algebraic systems arising from tomographic problems. In *Seismic tomography with applications in global seismology and exploration geophysics*, ed. G. Nolet (D. Reidel Publishing Company), chap. 3. 49–83
- Van Loan, C. F. (1992). *Computational Frameworks for the fast Fourier transform*. Frontiers in Applied Mathematics (SIAM)

- 1118 Villas, R. N. and Santos, M. (2001). Gold deposits of the Carajás mineral province: deposit types and
1119 metallogenesis. *Mineralium Deposita* 36, 300–331
- 1120 Xia, J. and Sprowl, D. R. (1991). Correction of topographic distortion in gravity data. *Geophysics* 56,
1121 537–541
- 1122 Xia, J., Sprowl, D. R., and Adkins-Heljeson, D. (1993). Correction of topographic distortions in potential-
1123 field data; a fast and accurate approach. *Geophysics* 58, 515–523. doi:10.1190/1.1443434
- 1124 Zhao, G., Chen, B., Chen, L., Liu, J., and Ren, Z. (2018). High-accuracy 3D Fourier forward modeling
1125 of gravity field based on the Gauss-FFT technique. *Journal of Applied Geophysics* 150, 294–303.
1126 doi:10.1016/j.jappgeo.2018.01.002
- 1127 Zidarov, D. (1965). Solution of some inverse problems of applied geophysics. *Geophysical Prospecting*
1128 13, 240–246. doi:10.1111/j.1365-2478.1965.tb01932.x

11 ALGORITHMS

Algorithm 1: Generic pseudo-code for the CGLS applied to the overdetermined problem (equation 22) for the particular case in which $\mathbf{H} = \mathbf{I}_P$ (equation 9 and subsection 3.2), $\mu = 0$ (equation 11), $\mathbf{W}_d = \mathbf{I}_D$ (equation 12) and $\tilde{\mathbf{p}} = \mathbf{0}$ (equation 14), where \mathbf{I}_P and \mathbf{I}_D are the identities of order P and D , respectively.

Initialization :

```

1 Compute  $\mathbf{G}$ ;
2 Set  $\mathbf{r} = \mathbf{d}$  and compute  $\delta = \|\mathbf{r}\|/D$ ;
3 Compute  $\boldsymbol{\vartheta} = \mathbf{G}^\top \mathbf{r}$  and  $\rho_0 = \boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$ ;
4 Set  $\tilde{\mathbf{p}} = \mathbf{0}$ ,  $\tau = 0$  and  $\boldsymbol{\eta} = \mathbf{0}$ ;
5  $m = 1$ ;
6 while ( $\delta > \epsilon$ ) and ( $m < \text{ITMAX}$ ) do
7   Update  $\boldsymbol{\eta} \leftarrow \boldsymbol{\vartheta} + \tau \boldsymbol{\eta}$ ;
8   Compute  $\boldsymbol{\nu} = \mathbf{G} \boldsymbol{\eta}$ ;
9   Compute  $v = \rho_0 / (\boldsymbol{\nu}^\top \boldsymbol{\nu})$ ;
10  Update  $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + v \boldsymbol{\eta}$ ;
11  Update  $\mathbf{r} \leftarrow \mathbf{r} - v \boldsymbol{\nu}$  and  $\delta \leftarrow \|\mathbf{r}\|/D$ ;
12  Compute  $\boldsymbol{\vartheta} = \mathbf{G}^\top \mathbf{r}$  and  $\rho = \boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$ ;
13  Compute  $\tau = \rho / \rho_0$ ;
14  Update  $\rho_0 \leftarrow \rho$ ;
15   $m \leftarrow m + 1$ ;
16 end

```

Algorithm 2: Generic pseudo-code for the method proposed by Leão and Silva (1989).

Initialization :

```

1 Set the indices  $\mathbf{i}^m$  for each data window,  $m \in \{1 : M\}$ ;
2 Set the indices  $\mathbf{j}^m$  for each source window,  $m \in \{1 : M\}$ ;
3 Set the constant depth  $z_0 + \Delta z_0$  for all equivalent sources;
4 Compute the vector  $\mathbf{a}'$  associated with the desired potential-field transformation;
5 Compute the matrix  $\mathbf{G}'$ ;
6 Compute the matrix  $\mathbf{B}'$  (equation 34);
7 Compute the vector  $(\mathbf{a}')^\top \mathbf{B}'$ ;
8  $m = 1$ ;
9 while  $m < M$  do
10   Compute  $t_c^m$  (equation 33);
11    $m \leftarrow m + 1$ ;
12 end

```

Algorithm 3: Generic pseudo-code for the method proposed by Soler and Uieda (2021).

Initialization :

- 1 Set the indices \mathbf{i}^m for each data window, $m \in \{1 : M\}$;
- 2 Set the indices \mathbf{j}^m for each source window, $m \in \{1 : M\}$;
- 3 Set the depth of all equivalent sources ;
- 4 Set a $D \times 1$ residuals vector $\mathbf{r} = \mathbf{d}$;
- 5 Set a $P \times 1$ vector $\tilde{\mathbf{p}} = \mathbf{0}$;
- 6 $m = 1$;
- 7 **while** $m < M$ **do**
- 8 Set the matrix \mathbf{W}_d^m ;
- 9 Compute the matrix \mathbf{G}^m ;
- 10 Compute $\tilde{\mathbf{p}}^m$ (equation 36) ;
- 11 $\tilde{\mathbf{p}}[\mathbf{j}^m] \leftarrow \tilde{\mathbf{p}}[\mathbf{j}^m] + \tilde{\mathbf{p}}^m$;
- 12 $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{G}[:, \mathbf{j}^m] \tilde{\mathbf{p}}^m$;
- 13 $m \leftarrow m + 1$;
- 14 **end**

Algorithm 4: Generic pseudo-code for the method proposed by Cordell (1992).

Initialization :

- 1 Compute a $D \times 1$ vector $\Delta \mathbf{z}$ whose i -th element Δz_i is a vertical distance controlling the depth of the i -th equivalent source, $i \in \{1 : D\}$;
- 2 Set a tolerance ϵ ;
- 3 Set a maximum number of iterations ITMAX ;
- 4 Set a $D \times 1$ residuals vector $\mathbf{r} = \mathbf{d}$;
- 5 Set a $D \times 1$ vector $\tilde{\mathbf{p}} = \mathbf{0}$;
- 6 Define the maximum absolute value r_{\max} in \mathbf{r} ;
- 7 $m = 1$;
- 8 **while** $(r_{\max} > \epsilon)$ **and** $(m < \text{ITMAX})$ **do**
- 9 Define the coordinates $(x_{\max}, y_{\max}, z_{\max})$ and index i_{\max} of the observation point associated with r_{\max} ;
- 10 $\tilde{\mathbf{p}}[i_{\max}] \leftarrow \tilde{\mathbf{p}}[i_{\max}] + (r_{\max} \Delta \mathbf{z}[i_{\max}])$;
- 11 $\mathbf{r} \leftarrow \mathbf{r} - (\mathbf{G}[:, i_{\max}] \tilde{\mathbf{p}}[i_{\max}])$;
- 12 Define the new r_{\max} in \mathbf{r} ;
- 13 $m \leftarrow m + 1$;
- 14 **end**

Algorithm 5: Generic pseudo-code for the method proposed by Mendonça and Silva (1994).**Initialization :**

```

1 Set a regular grid of  $P$  equivalent sources at a horizontal plane  $z_0$  ;
2 Set a tolerance  $\epsilon$  ;
3 Set a  $D \times 1$  residuals vector  $\mathbf{r} = \mathbf{d}$  ;
4 Define the maximum absolute value  $r_{\max}$  in  $\mathbf{r}$  ;
5 Define the index  $i_{\max}$  of  $r_{\max}$  ;
6 Define the list of indices  $\mathbf{i}_r$  of the remaining data in  $\mathbf{r}$  ;
7 Define  $\mathbf{d}_e = \mathbf{d}[i_{\max}]$  ;
8 Compute  $(\mathbf{F} + \mu \mathbf{I}_{D_e})$  and  $\mathbf{G}_e$  ;
9 Compute  $\tilde{\mathbf{p}}$  (equation 40) ;
10 Compute  $\mathbf{r} = \mathbf{d}[\mathbf{i}_r] - \mathbf{G}[\mathbf{i}_r, :] \tilde{\mathbf{p}}$  ;
11 Define the maximum absolute value  $r_{\max}$  in  $\mathbf{r}$  ;
12 while ( $r_{\max} > \epsilon$ ) do
13   Define the index  $i_{\max}$  of  $r_{\max}$  ;
14   Define the list of indices  $\mathbf{i}_r$  of the remaining elements in  $\mathbf{r}$  ;
15    $\mathbf{d}_e \leftarrow \begin{bmatrix} \mathbf{d}_e \\ \mathbf{d}[i_{\max}] \end{bmatrix}$  ;
16   Update  $(\mathbf{F} + \mu \mathbf{I}_{D_e})$  and  $\mathbf{G}_e$  ;
17   Update  $\tilde{\mathbf{p}}$  (equation 40) ;
18   Update  $\mathbf{r} = \mathbf{d}[\mathbf{i}_r] - \mathbf{G}[\mathbf{i}_r, :] \tilde{\mathbf{p}}$  ;
19   Define the maximum absolute value  $r_{\max}$  in  $\mathbf{r}$  ;
20 end

```

Algorithm 6: Generic pseudo-code for the iterative method proposed by Siqueira et al. (2017). The symbol “ \circ ” denotes the entrywise or Hadamard product (e.g., Horn and Johnson, 1991, p. 298) and $\boldsymbol{\sigma}$ is a $P \times 1$ vector whose j -th element is the ratio of a predefined element of area centered at the j -th equivalent source and the term $2\pi\gamma$, where γ is the gravitational constant.

Initialization :

```

1 Set  $P$  equivalent sources on a horizontal plane  $z_0$  ;
2 Set a tolerance  $\epsilon$  ;
3 Set a maximum number of iterations ITMAX ;
4 Set an auxiliary vector  $\boldsymbol{\sigma}$  ;
5 Compute  $\tilde{\mathbf{p}} = \boldsymbol{\sigma} \circ \mathbf{d}$  ;
6 Compute  $\mathbf{G}$  (equation 3) ;
7 Compute  $\mathbf{r} = \mathbf{d} - \mathbf{G} \tilde{\mathbf{p}}$  ;
8 Compute  $\delta = \|\mathbf{r}\|/D$  ;
9  $m = 1$  ;
10 while ( $\delta > \epsilon$ ) and ( $m < \text{ITMAX}$ ) do
11   Compute  $\Delta \mathbf{p} = \boldsymbol{\sigma} \circ \mathbf{r}$  ;
12   Update  $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + \Delta \mathbf{p}$  ;
13   Compute  $\boldsymbol{\nu} = \mathbf{G} \Delta \mathbf{p}$  ;
14   Update  $\mathbf{r} \leftarrow \mathbf{r} - \boldsymbol{\nu}$  ;
15   Compute  $\delta = \|\mathbf{r}\|/D$  ;
16    $m \leftarrow m + 1$  ;
17 end

```

Algorithm 7: Generic pseudo-code for the convolutional equivalent-layer method proposed by Takahashi et al. (2020, 2022).

Initialization :

- 1 Set the regular grid of P equivalent sources on a horizontal plane z_0 ;
 - 2 Set a tolerance ϵ and a maximum number of iterations ITMAX ;
 - 3 Compute the first column $\mathbf{G}[:, 1]$ and row $\mathbf{G}[1, :]$ of the sensitivity matrix \mathbf{G} (equation 3) for the particular case in which it has a BTTB structure (equation 51);
 - 4 Rearrange the elements of $\mathbf{G}[:, 1]$ into matrix \mathbf{C} , compute its 2D-DFT via 2D-FFT and multiply by $\sqrt{4D}$ to obtain a matrix \mathbf{L}' (equation 66);
 - 5 Rearrange the elements of $\mathbf{G}[1, :]$ into matrix \mathbf{C} , compute its 2D-DFT via 2D-FFT and multiply by $\sqrt{4D}$ to obtain a matrix \mathbf{L}'' (equation 66);
 - 6 Set $\tilde{\mathbf{p}} = \mathbf{0}$;
 - 7 Set $\mathbf{r} = \mathbf{d}$ and compute $\delta = \|\mathbf{r}\|/D$;
 - 8 Compute $\boldsymbol{\vartheta} = \mathbf{G}^\top \mathbf{r}$ (Algorithm 8) and $\rho_0 = \boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$;
 - 9 Set $\tau = 0$ and $\boldsymbol{\eta} = \mathbf{0}$;
 - 10 $m = 1$;
 - 11 **while** ($\delta > \epsilon$) **and** ($m < \text{ITMAX}$) **do**
 - 12 Update $\boldsymbol{\eta} \leftarrow \boldsymbol{\vartheta} + \tau \boldsymbol{\eta}$;
 - 13 Compute $\boldsymbol{\nu} = \mathbf{G} \boldsymbol{\eta}$ (Algorithm 8);
 - 14 Compute $v = \rho_0 / (\boldsymbol{\nu}^\top \boldsymbol{\nu})$;
 - 15 Update $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + v \boldsymbol{\eta}$;
 - 16 Update $\mathbf{r} \leftarrow \mathbf{r} - v \boldsymbol{\nu}$ and $\delta \leftarrow \|\mathbf{r}\|/D$;
 - 17 Compute $\boldsymbol{\vartheta} = \mathbf{G}^\top \mathbf{r}$ (Algorithm 8) and $\rho = \boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$;
 - 18 Compute $\tau = \rho / \rho_0$;
 - 19 Update $\rho_0 \leftarrow \rho$;
 - 20 $m \leftarrow m + 1$;
 - 21 **end**
-

Algorithm 8: Pseudo-code for computing the generic matrix-vector products given by equations 52 and 53 via fast 2D discrete convolution for a given vector \mathbf{v} (equation 54) and matrix \mathbf{L} (equation 66).

- 1 Rearrange the elements of \mathbf{v} (equations 52 and 54) into the matrix \mathbf{V}_c (equation 65);
 - 2 Compute $\mathcal{F}_{2N_B} \mathbf{V}_c \mathcal{F}_{2N_b}$ via 2D-FFT;
 - 3 Compute the Hadamard product with matrix \mathbf{L} (equation 66);
 - 4 Compute 2D-IDFT via 2D-FFT to obtain matrix \mathbf{W}_c (65);
 - 5 Retrieve \mathbf{w} (equations 52 and 54) from \mathbf{w}_c (equations 55–57);
-

12 TABLES

Reference	Term	flops
eq. 10	$\mathbf{G}\mathbf{H}$	$2DQP$
eq. 15	$\mathbf{H}\tilde{\mathbf{q}}$	$2PQ$
eq. 22	$(\mathbf{G}\mathbf{H})^\top (\mathbf{G}\mathbf{H})$	$2Q^2D$
eq. 22	$(\mathbf{G}\mathbf{H})^\top \boldsymbol{\delta}_d$	$2QD$
eq. 23	$(\mathbf{G}\mathbf{H}) (\mathbf{G}\mathbf{H})^\top$	$2D^2Q$
eq. 23	$(\mathbf{G}\mathbf{H})^\top \mathbf{u}$	$2QD$
eq. 25	lower triangle of \mathcal{G}	$D^3/3$ or $Q^3/3$
eq. 26	solve triangular systems	$2D^2$ or $2Q^2$
Alg. 1	$\boldsymbol{\eta} \leftarrow \boldsymbol{\vartheta} + \tau \boldsymbol{\eta}$	$2Q$
Alg. 1	$\boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$	$2Q$
Alg. 6	$\boldsymbol{\sigma} \circ \mathbf{d}$	D

Table 1. Total number of flops associated with some useful terms according to Golub and Van Loan (2013, p. 12). The number of flops associated with equations 25 and 26 depends if the problem is over or underdetermined. Note that $P = Q$ for the case in which $\mathbf{H} = \mathbf{I}_P$ (subsection 3.2). The term associated with Algorithm 1 is a vector update called *saxpy* (Golub and Van Loan, 2013, p. 4). The terms defined here are references to compute the total number of flops throughout the manuscript.

13 FIGURES

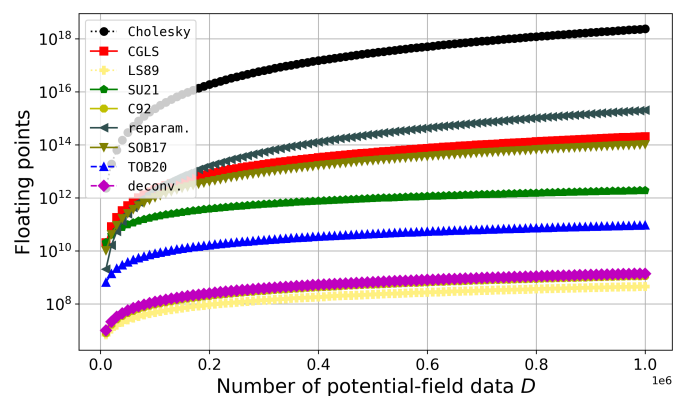


Figure 1. Total number of flops for different equivalent-layer methods (equations 27, 28, 35, 37, 38, 43, 50, 67, and 70). The number of potential-field data D varies from 10,000 to 1,000,000.

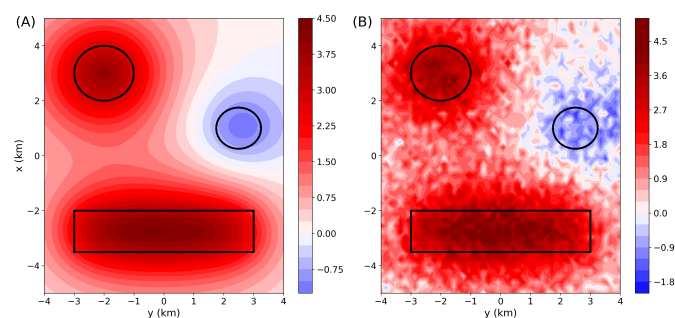


Figure 2. Synthetic gravity disturbance data (in mGal). The data are located on a regular grid of 50×50 points. Panel (A) shows the noise-free data. Panel (B) shows the synthetic data corrupted with a pseudorandom Gaussian noise having zero mean and standard deviation equal to 10% of the maximum absolute value in the noise-free data. The black lines represent the projection of the synthetic bodies on the xy plane.

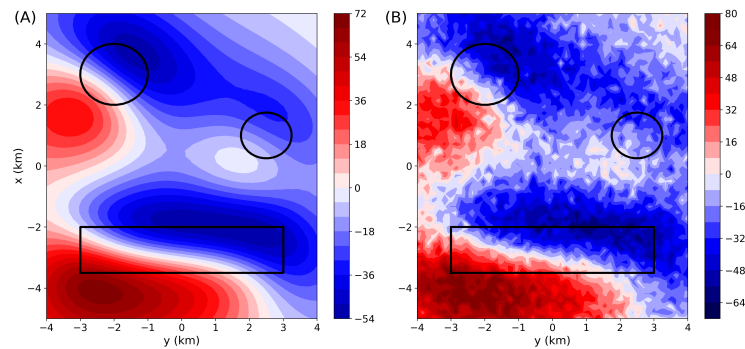


Figure 3. Synthetic total-field anomaly data (in nT). The data are located on a regular grid of 50×50 points. Panel (A) shows the noise-free data. Panel (B) shows the synthetic data corrupted with a pseudorandom Gaussian noise having zero mean and standard deviation equal to 10% of the maximum absolute value in the noise-free data. The black lines represent the projection of the synthetic bodies on the xy plane.

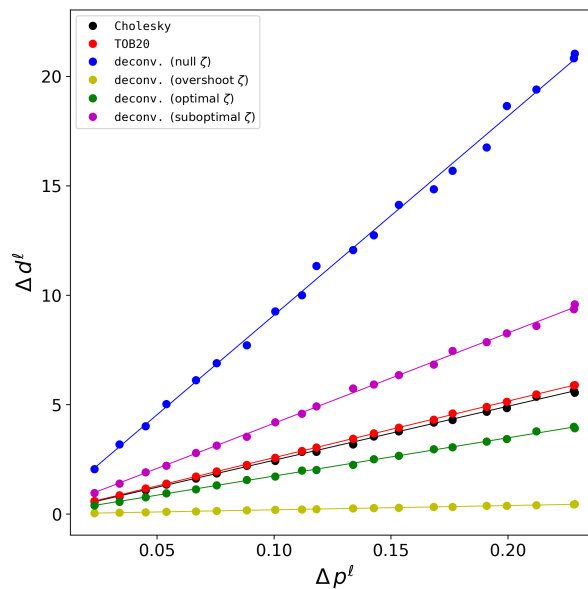


Figure 4. Numerical stability curves obtained for the 21 synthetic gravity data sets by using the Cholesky factorization with $\mu \approx 10^{-22}$, the iterative deconvolution (T0B20) proposed by Takahashi et al. (2020) with 40 iterations (Algorithm 7) and the direct deconvolution (deconv.) computed with four different values for ζ (equation 69): 0, 10^{-20} (overshoot), 10^{-22} (optimal) and 10^{-24} (suboptimal). The stability parameter κ (equation 29) obtained for the six curves described above are 24.66 (Cholesky), 25.81 (T0B20), 90.93, 1.95, 17.47 and 41.24 (deconv. with null, overshoot, optimal and suboptimal ζ).

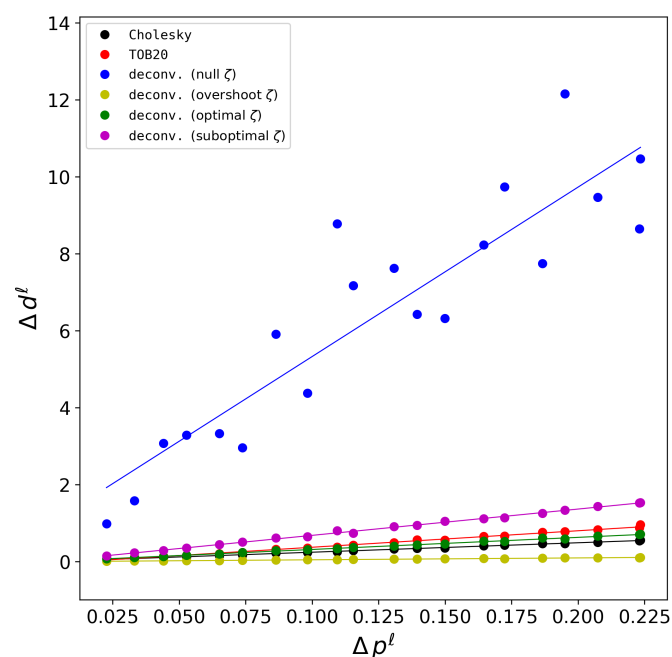


Figure 5. Numerical stability curves obtained for the 21 synthetic magnetic data sets by using the Cholesky factorization with $\mu \approx 10^{-14}$, the iterative deconvolution (TOB20) proposed by Takahashi et al. (2022) with 40 iterations (Algorithm 7) and the direct deconvolution (deconv.) computed with four different values for ζ (equation 69): 0, 10^{-12} (overshoot), 10^{-14} (optimal) and 10^{-16} (suboptimal). The stability parameter κ (equation 29) obtained for the six curves described above are 2.46 (Cholesky), 4.29 (TOB20), 44.04, 0.49, 3.15 and 6.83 (deconv. with null, overshoot, optimal and suboptimal ζ).

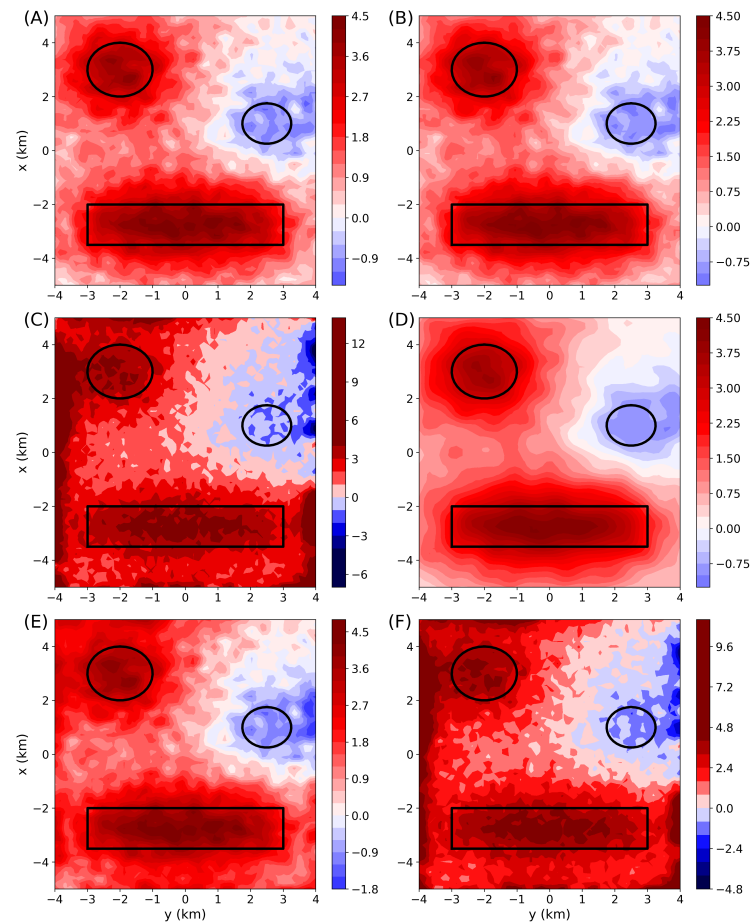


Figure 6. Predicted data f (equation 3) due to the estimated parameter vector $\tilde{\mathbf{p}}$ obtained with the gravity data having maximum noise level (Figure 2B) by using (A) the Cholesky factorization with $\mu \approx 10^{-22}$, (B) the iterative deconvolution (TOB20) proposed by Takahashi et al. (2020) with 40 iterations (Algorithm 7) and (C)–(F) the direct deconvolution (deconv.) computed, respectively, with four different values for ζ (equation 69): 0, 10^{-20} (overshoot), 10^{-22} (optimal) and 10^{-24} (suboptimal). All values are in mGal.

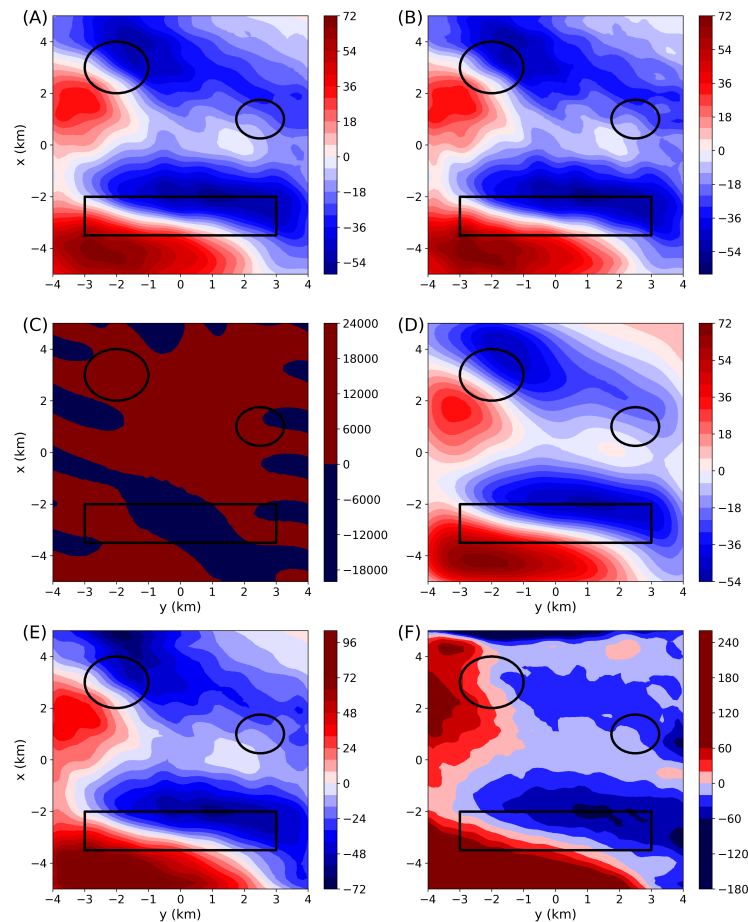


Figure 7. Predicted data f (equation 3) due to the estimated parameter vector $\tilde{\mathbf{p}}$ obtained with the magnetic data having maximum noise level (Figure 3B) by using (A) the Cholesky factorization with $\mu \approx 10^{-14}$, (B) the iterative deconvolution (TOB20) proposed by Takahashi et al. (2022) with 40 iterations (Algorithm 7) and (C)–(F) the direct deconvolution (deconv.) computed with four different values for ζ (equation 69): 0, 10^{-12} (overshoot), 10^{-14} (optimal) and 10^{-16} (suboptimal). All values are in nT.

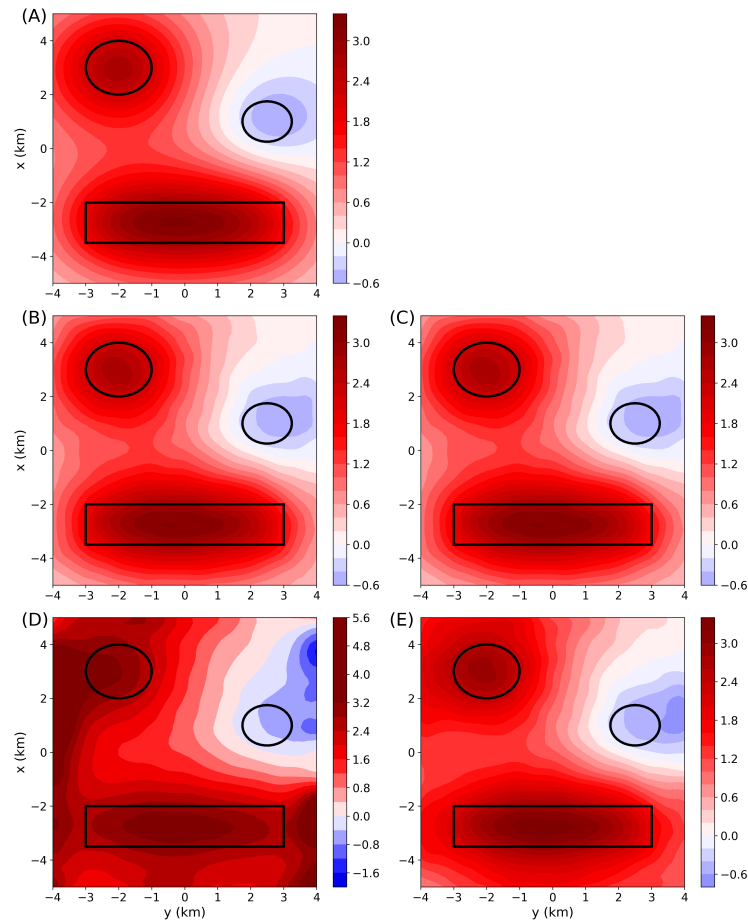


Figure 8. Upward continuation of the synthetic gravity data with maximum noise level (Figure 2B). **(A)** Noise-free gravity data produced by the model at $z = -500$ m. **(B)–(E)** Continued data due to the estimated parameter vector $\tilde{\mathbf{p}}$ obtained, respectively, by using the Cholesky factorization with $\mu \approx 10^{-22}$, the iterative deconvolution (T0B20) proposed by Takahashi et al. (2020) with 40 iterations (Algorithm 7) and the direct deconvolution (deconv.) computed, respectively, with two different values for ζ (equation 69): 0 and 10^{-22} (optimal). All values are in mGal.

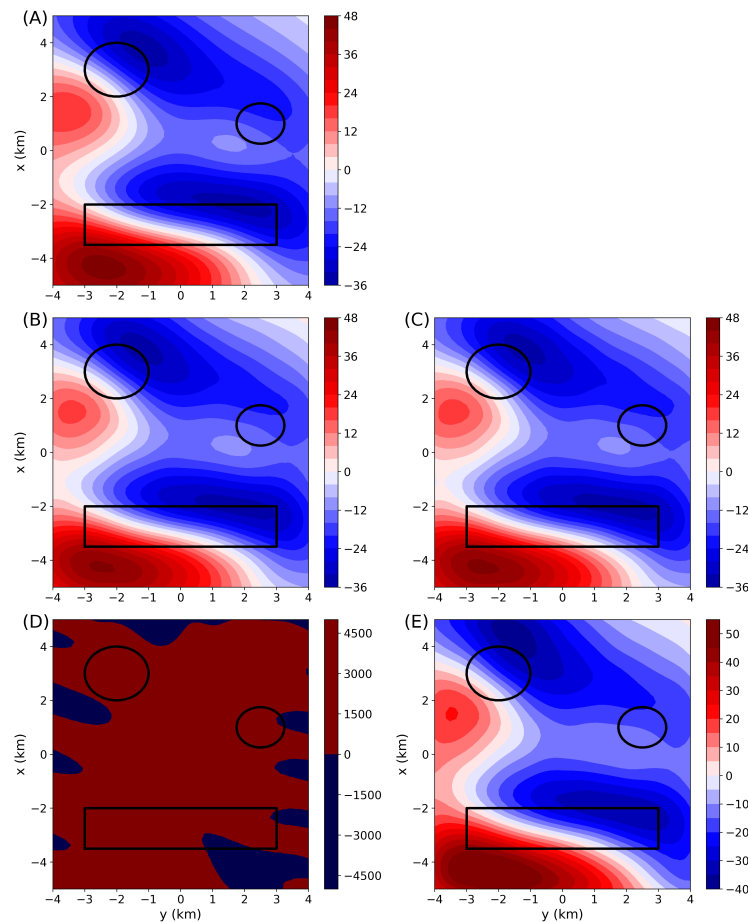


Figure 9. Upward continuation of the synthetic magnetic data with maximum noise level (Figure 3B). (A) Noise-free magnetic data produced by the model at $z = -1400$ m. (B)–(E) Continued data due to the estimated parameter vector $\tilde{\mathbf{p}}$ obtained, respectively, by using the Cholesky factorization with $\mu \approx 10^{-14}$, the iterative deconvolution (TOB20) proposed by Takahashi et al. (2022) with 40 iterations (Algorithm 7) and the direct deconvolution (deconv.) computed, respectively, with two different values for ζ (equation 69): 0 and 10^{-14} (optimal). All values are in nT.