

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

2 ABSTRACT

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

27 **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 (?) 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. 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

71 stations. In Cordell's 1992 method, a single column of the sensitivity matrix is calculated per iteration,
72 meaning that a single equivalent source contributes to data fitting in each iteration. Guspí and Novara (2009)
73 further extended Cordell's 1992 method by applying it to scattered magnetic observations. Following the
74 strategy of column-action update, Mendonça and Silva (1994) developed an iterative procedure where one
75 data point is incorporated at a time, and a single row of the sensitivity matrix is calculated per iteration.
76 This strategy adopted by Mendonça and Silva (1994) is known as 'equivalent data concept'. The concept of
77 equivalent data is based on the principle that certain data points within a dataset are redundant and, as a
78 result, do not contribute to the final solution. On the other hand, there is a subset of observations known as
79 equivalent data, which effectively contributes to the final solution and fits the remaining redundant data.
80 In their work, Mendonça and Silva (1994) adopted an iterative approach to select a substantially smaller
81 subset of equivalent data from the original dataset.

82 The next strategy involves reparametrizing the equivalent layer with the objective of solving a smaller
83 linear inverse problem by reducing the dimension of the model space. Following the strategy of the
84 reparametrization of the equivalent layer, Barnes and Lumley (2011) proposed a quadtree discretization
85 of the equivalent sources. Oliveira Jr. et al. (2013) reduced the model parameters by approximating
86 the equivalent-source layer by a piecewise-polynomial function defined on a set of user-defined small
87 equivalent-source windows. The estimated parameters are the polynomial coefficients for each window and
88 they are much smaller than the original number of equivalent sources. By using the subspace method, ?
89 reparametrizes the equivalent layer, which involves reducing the dimension of the linear system from the
90 original parameter-model space to a lower-dimensional subspace. The subspace bases span the parameter-
91 model space and they are constructed by applying the singular value decomposition to the matrix containing
92 the gridded data.

93 Following the strategy of the wavelet compression, Li and Oldenburg (2010) transformed the full
94 sensitivity matrix into a sparse one using the compression of the coefficient matrix via wavelet transforms
95 based on the orthonormal compactly supported wavelets.

96 The strategy named iterative methods estimates iteratively the parameter vector that represents a distri-
97 bution over an equivalent layer. Xia and Sprowl (1991) and Xia et al. (1993) have developed efficient
98 iterative algorithms for updating the distribution of physical properties within the equivalent layer in the
99 wavenumber and space domains, respectively. Specifically, in Xia and Sprowl's (1991) method the physical-
100 property distribution is updated by using the ratio between the squared depth to the equivalent source and
101 the gravitational constant multiplied by the residual between the observed and predicted observation at the
102 measurement station. Siqueira et al. (2017) developed an iterative solution where the sensitivity matrix is
103 transformed into a diagonal matrix with constant terms through the use of the *excess mass criterion* and
104 of the positive correlation between the observed gravity data and the masses on the equivalent layer. The
105 fundamentals of the Siqueira et al.'s (2017) method is based on the Gauss' theorem (e.g., ?, p. 43) and the
106 total excess of mass (e.g., ?, p. 60). All these iterative methods use the full and dense sensitivity matrix to
107 calculate the predicted data and residual data in the whole survey data per iteration. Hence, the iterative
108 methods proposed by Xia and Sprowl (1991), Xia et al. (1993) and Siqueira et al. (2017) neither compress
109 nor reparametrize the sensitivity matrix. Jirigalatu and Ebbing (2019) also proposed an iterative equivalent
110 layer that uses the full and dense sensitivity matrix. However, in their approach, Jirigalatu and Ebbing
111 (2019) efficiently compute the predicted data and residual data for the entire survey data per iteration in the
112 wavenumber domain.

113 Following the strategy of the iterative deconvolution ?, ?, developed fast and effective equivalent-layer
114 techniques for processing, respectively, gravity and magnetic data by modifying the forward modeling to

115 estimate the physical-property distribution over the equivalent layer through a 2D discrete convolution that
116 can be efficiently computed via 2D FFT. These methods took advantage of the Block-Toeplitz Toeplitz-
117 block (BTTB) structure of the sensitivity matrices, allowing them to be calculated by using only their first
118 column. In practice, the forward modeling uses a single equivalent source, which significantly reduces the
119 the required RAM memory.

120 The method introduced by ? (?, ?) can be reformulated to eliminate the need for conjugate gradient
121 iterations. This reformulation involves employing a *direct deconvolution* approach, similar to the concept
122 described by (e.g., ?, p. 220), utilizing a *Wiener filter* (e.g., ?, p. 263).

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

2 FUNDAMENTALS

130 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) ,
 131 $i \in \{1 : D\}$, of a topocentric Cartesian system with x , y and z axes pointing to north, east and down,
 132 respectively. Consider that d_i can be satisfactorily approximated by a harmonic function

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

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

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

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

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

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

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

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

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

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

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

151 2.1 Spatial distribution and total number of equivalent sources

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

158 From a theoretical point of view, the equivalent layer reproducing a given potential field data set cannot
 159 cross the true gravity or magnetic sources. This condition is a consequence of recognizing that the equivalent
 160 layer is essentially an indirect solution of a boundary value problem of potential theory (e.g., ??Dampney,
 161 1969; ?; ?). In practical applications, however, there is no guarantee that this condition is satisfied. Actually,
 162 its is widely known from practical experience (e.g., ?) that the equivalent-layer technique works even for
 163 the case in which the layer cross the true sources.

164 Regarding the depth of the equivalent layer, Dampney (1969) proposed a criterion based on horizontal data
 165 sampling, suggesting that the equivalent-layer depth should be between two and six times the horizontal
 166 grid spacing, considering evenly spaced data. However, when dealing with a survey pattern that has
 167 unevenly spaced data, ? adopted an alternative empirical criterion. According to their proposal, the depth
 168 of the equivalent layer should range from two to three times the spacing between adjacent flight lines. The
 169 criteria of Dampney (1969) and ? are valid for planar equivalent layers. Cordell (1992) have proposed
 170 and an alternative criterion for scattered data that leads to an undulating equivalent layer. This criterion
 171 have been slightly modified by ?, Guspí and Novara (2009) and Soler and Uieda (2021), for example,
 172 and consists in setting one equivalent source below each datum at a depth proportional to the horizontal
 173 distance to the nearest neighboring data points. Soler and Uieda (2021) have compared different strategies
 174 for defining the equivalent sources depth for the specific problem of interpolating gravity data, but they
 175 have not found significant differences between them.

176 2.2 Matrix G

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

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

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

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

180 and

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

181 In this case, the equivalent layer is formed by punctual sources representing monopoles or dipoles (e.g.,
 182 Dampney, 1969; ?; Leão and Silva, 1989; Cordell, 1992; Oliveira Jr. et al., 2013; Siqueira et al., 2017; ?;
 183 ?; Soler and Uieda, 2021; ?). Another common approach consists in not defining g_{ij} by using equations
 184 6–8, but other harmonic functions obtained by integrating them over the volume of regular prisms (e.g.,
 185 Li and Oldenburg, 2010; Barnes and Lumley, 2011; ?; Jirigalatu and Ebbing, 2019). There are also some
 186 less common approaches defining the harmonic function g_{ij} (equation 2) as the potential field due to plane
 187 faces with constant physical property (?), doublets (?) or by computing the double integration of the inverse
 188 distance function with respect to z (Guspí and Novara, 2009).

189 A common assumption for most of the equivalent-layer methods is that the harmonic function g_{ij}
 190 (equation 2) is independent on the actual physical relationship between the observed potential field and

their true sources (e.g., Cordell, 1992; Guspí and Novara, 2009; ?). 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, ?? 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

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

where

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

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

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

or, equivalently (? , 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 $\tilde{\mathbf{q}}$ is obtained by solving the following linear system for $\tilde{\boldsymbol{\delta}}_q$ (equation 18):

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

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

242 matrix-vector product as follows:

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

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

245 3.2 Formulation without reparameterization

246 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 ,
247 $P = Q$, $\mathbf{p} = \mathbf{q}$, $\bar{\mathbf{p}} = \bar{\mathbf{q}}$ (equation 14) and $\tilde{\mathbf{p}} = \tilde{\mathbf{q}}$ (equation 15). In this case, the linear system (equations 22
248 and 23) is directly solved for

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

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

250 3.3 Linear system solvers

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

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

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

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

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

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

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

267 These problems can be overcome by solving the linear system using an iterative method. These methods
268 produce a sequence of vectors that typically converge to the solution at a reasonable rate. The main
269 computational cost associated with these methods is usually some matrix-vector products per iteration. The
270 *conjugate gradient* (CG) is a very popular iterative method for solving linear systems in equivalent-layer

271 methods. This method was originally developed to solve systems having a square and positive definite
 272 matrix. There are two adapted versions of the CG method. The first is called *conjugate gradient normal*
 273 *equation residual* (CGNR) Golub and Loan (2013, sec. 11.3) or *conjugate gradient least squares* (CGLS)
 274 (?, p. 165) and is used to solve overdetermined problems (equation 22). The second is called *conjugate*
 275 *gradient normal equation error* (CGNE) method Golub and Loan (2013, sec. 11.3) and is used to solve
 276 the underdetermined problems (equation 23). Algorithm 1 outlines the CGLS method applied to the
 277 overdetermined problem (equation 22).

4 FLOATING-POINT OPERATIONS

278 Two important factors affecting the efficiency of a given matrix algorithm are the storage and amount of
 279 required arithmetic. Here, we quantify this last factor associated with different computational strategies
 280 to solve the linear system of the equivalent-layer technique (section 7). To do it, we opted by counting
 281 *flops*, which are floating point additions, subtractions, multiplications or divisions (Golub and Loan, 2013,
 282 p. 12–14). This is a non-hardware dependent approach that allows us to do direct comparison between
 283 different equivalent-layer methods. Most of the flops count used here can be found in Golub and Loan
 284 (2013, p. 12, 106, 107 and 164).

285 Let us consider the case in which the overdetermined problem (equation 22) is solved by Cholesky
 286 factorization (equations 25 and 26) directly for the parameter vector $\tilde{\mathbf{p}}$ by considering the particular case in
 287 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 $\tilde{\mathbf{p}} = \mathbf{0}$
 288 (equation 14), where \mathbf{I}_P and \mathbf{I}_D are the identities of order P and D , respectively. Based on the information
 289 provided in table 1, the total number of flops can be determined by aggregating the flops required for
 290 various computations. These computations include the matrix-matrix and matrix-vector products $\mathbf{G}^\top \mathbf{G}$
 291 and $\mathbf{G}^\top \mathbf{d}$, the Cholesky factor \mathcal{G} , and the solution of triangular systems. Thus, we can express the total
 292 number of flops as follows:

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

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

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

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

5 NUMERICAL STABILITY

300 All equivalent-layer methods aim at obtaining an estimate $\tilde{\mathbf{p}}$ for the parameter vector \mathbf{p} (equation 3), which
 301 contains the physical property of the equivalent sources. Some methods do it by first obtaining an estimate
 302 $\tilde{\mathbf{q}}$ for the reparameterized parameter vector \mathbf{q} (equation 9) and then using it to obtain $\tilde{\mathbf{p}}$ (equation 15).
 303 The stability of a solution $\tilde{\mathbf{p}}$ against noise in the observed data is rarely addressed. Here, we follow the
 304 numerical stability analysis presented in Siqueira et al. (2017).

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

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

310 where κ is the constant of proportionality between the model perturbation

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

311 and the data perturbation

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

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

6 NOTATION FOR SUBVECTORS AND SUBMATRICES

317 Here, we use a notation inspired on that presented by Van Loan (1992, p. 4) to represent subvectors and
 318 submatrices. Subvectors of \mathbf{d} , for example, are specified by $\mathbf{d}[\mathbf{i}]$, where \mathbf{i} is a list of integer numbers that
 319 “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
 320 $\mathbf{d}[\mathbf{i}] = [d_1 \ d_6 \ d_4 \ d_6]^\top$. Note that the list \mathbf{i} of indices may be sorted or not and it may also have repeated
 321 indices. For the particular case in which the list has a single element $\mathbf{i} = (i)$, then it can be used to extract
 322 the i -th element $d_i \equiv \mathbf{d}[i]$ of \mathbf{d} . Sequential lists can be represented by using the colon notation. We consider
 323 two types of sequential lists. The first has starting index is smaller than the final index and increment of 1.
 324 The second has starting index is greater than the final index and increment of -1 . For example,

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

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

326 The notation above can also be used to define submatrices of a $D \times P$ matrix \mathbf{G} . For example, $\mathbf{i} =$
 327 $(2, 7, 4, 6)$ and $\mathbf{j} = (1, 3, 8)$ lead to the submatrix

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

328 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
 329 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 $\mathbf{G}[:, j]$
 330 represents the j -th column. Finally, we may use the colon notation to define the following submatrix:

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

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

7 COMPUTATIONAL STRATEGIES

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

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

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

341 7.1 Moving window

342 The initial approach to enhance the computational efficiency of the equivalent-layer technique is com-
 343 monly denoted *moving window* and involves first splitting the observed data $d_i, i \in \{1 : D\}$, into M
 344 overlapping subsets (or data windows) formed by D^m data each, $m \in \{1 : M\}$. The data inside the
 345 m -th window are usually adjacent to each other and have indices defined by an integer list \mathbf{i}^m having
 346 D^m elements. The number of data D^m forming the data windows are not necessarily equal to each other.
 347 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
 348 a set of P equivalent sources with scalar physical property $p_j, j \in \{1 : P\}$, and also split them into M
 349 overlapping subsets (or source windows) formed by P^m data each, $m \in \{1 : M\}$. The sources inside the
 350 m -th window have indices defined by an integer list \mathbf{j}^m having P^m elements. Each source window has a
 351 $P^m \times 1$ parameter vector \mathbf{p}^m and is located right below the corresponding m -th data window. Then, each
 352 $\mathbf{d}^m \equiv \mathbf{d}[\mathbf{i}^m]$ is approximated by

$$353 \quad \mathbf{f}^m = \mathbf{G}^m \mathbf{p}^m, \tag{32}$$

353 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
 354 2 using only the data and equivalent sources located inside the window m -th. The main idea of the moving-
 355 window approach is using the $\tilde{\mathbf{p}}^m$ estimated for each window to obtain (i) an estimate $\tilde{\mathbf{p}}$ of the parameter
 356 vector for the entire equivalent layer or (ii) a given potential-field transformation \mathbf{t} (equation 4). The main
 357 advantages of this approach is that (i) the estimated parameter vector $\tilde{\mathbf{p}}$ or transformed potential field are
 358 not obtained by solving the full, but smaller linear systems and (ii) the full matrix \mathbf{G} (equation 3) is never
 359 stored.

Leão and Silva (1989) presented a pioneer work using the moving-window approach. Their method requires a regularly-spaced grid of observed data on a horizontal plane z_0 . The data windows are defined by square local grids of $\sqrt{D'} \times \sqrt{D'}$ adjacent points, all of them having the same number of points D' . The equivalent sources in the m -th data window are located below the observation plane, at a constant vertical distance Δz_0 . They are arranged on a regular grid of $\sqrt{P'} \times \sqrt{P'}$ adjacent points following the same grid pattern of the observed data. The local grid of sources for all data windows have the same number of elements P' . Besides, they are vertically aligned, but expands the limits of their corresponding data windows, so that $D' < P'$. Because of this spatial configuration of observed data and equivalent sources, 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 $D' \times P'$ constant matrix.

By omitting the normalization strategy used by Leão and Silva (1989), their method consists in directly 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 as follows:

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

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

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

is a particular case of matrix \mathbf{B} associated with underdetermined problems (equation 21) for the particular 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 $\mathbf{I}_{P'}$ and $\mathbf{I}_{D'}$ are identity matrices of order P' and D' , respectively, and $\mathbf{0}$ is a vector of zeros. Due to the presumed spatial configuration of the observed data and equivalent sources, \mathbf{a}' and \mathbf{G}' are the same for all data windows. Hence, only the data vector \mathbf{d}^m is modified according to the position of the data window. Note that equation 33 combines the potential-field transformation (equation 4) with the solution of the undetermined problem (equation 23).

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

The total number of flops in Algorithm 2 depends on computing the $P' \times D'$ matrix \mathbf{B}' (equation 34) in 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 an inner product (equation 33) per iteration. We consider that the total number of flops associated with \mathbf{B}' is obtained by the matrix-matrix product $\mathbf{G}' (\mathbf{G}')^\top$, its inverse and then the premultiplication by $(\mathbf{G}')^\top$. By using table 1 and considering that inverse is computed via Cholesky factorization, we obtain that the total 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 Algorithm 2 is

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

Soler and Uieda (2021) generalized the method proposed by Leão and Silva (1989) for irregularly spaced data on an undulating surface. A direct consequence of this generalization is that a different submatrix $\mathbf{G}^m \equiv \mathbf{G}[\mathbf{i}^m, \mathbf{j}^m]$ (equation 32) must be computed for each window. Differently from Leão and Silva

397 (1989), Soler and Uieda (2021) store the computed $\tilde{\mathbf{p}}^m$ for all windows and subsequently use them to obtain
 398 a desired potential-field transformation (equation 4) as the superposed effect of all windows. The estimated
 399 $\tilde{\mathbf{p}}^m$ for all windows are combined to form a single $P \times 1$ vector $\tilde{\mathbf{p}}$, which is an estimate for original
 400 parameter vector \mathbf{p} (equation 3). For each data window, Soler and Uieda (2021) solve an overdetermined
 401 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
 402 a diagonal matrix of weights for the data inside the m -th window and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14), so that

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

403 Unlike Leão and Silva (1989), Soler and Uieda (2021) do not adopt a sequential order of the data
 404 windows; rather, they adopt a randomized order of windows in their iterations. The overall steps of the
 405 method proposed by Soler and Uieda (2021) are defined by the Algorithm 3. For convenience, we have
 406 omitted the details about the randomized window order and the normalization strategy employed by Soler
 407 and Uieda (2021). Note that this algorithm starts with a residuals vector \mathbf{r} that is iteratively updated. The
 408 iterative algorithm in Soler and Uieda (2021) estimates a solution ($\tilde{\mathbf{p}}^m$ in equation 36) using the data and
 409 the equivalent sources that fall within a moving-data window; however, it calculates the predicted data
 410 and the residual data in the whole survey data. Next, the residual data that fall within a new position of
 411 the data window is used as input data to estimate a new solution within the data window which, in turn, is
 412 used to calculate a new predicted data and a new residual data in the whole survey data. Regarding the
 413 equivalent-source layout, Soler and Uieda (2021) proposed the block-averaged sources locations in which
 414 the survey area is divided into horizontal blocks and one single equivalent source is assigned to each block.
 415 Each single source per block is placed over the layer with its horizontal coordinates given by the average
 416 horizontal positions of observation points. According to Soler and Uieda (2021), the block-averaged
 417 sources layout may prevent aliasing of the interpolated values, specially when the observations are unevenly
 418 sampled. This strategy also reduces the number of equivalent sources without affecting the accuracy of the
 419 potential-field interpolation. Besides, it reduces the computational load for estimating the physical property
 420 on the equivalent layer. This block-averaged sources layout is not included in the Algorithm 3.

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

$$f_{SU21} = M \left[\frac{1}{3}(P')^3 + 2(D' + 1)(P')^2 + (4D' + 1)P' \right], \quad (37)$$

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

429 7.2 Column-action update

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

433 Cordell (1992) proposed a computational strategy that was later used by Guspí and Novara (2009) and
 434 relies on first defining one equivalent source located right below each observed data d_i , $i \in \{1 : D\}$, at

a vertical coordinate $z_i + \Delta z_i$, where Δz_i is proportional to the distance from the i -th observation point (x_i, y_i, z_i) to its closest neighbor. The second step consists in updating the physical property p_j of a single equivalent source, $j \in \{1 : D\}$ and remove its predicted potential field from the observed data vector \mathbf{d} , producing a residuals vector \mathbf{r} . At each iteration, the single equivalent source is the one located vertically beneath the observation station of the maximum data residual. Next, the predicted data produced by this single source is calculated over all of the observation points and a new data residual \mathbf{r} and the $D \times 1$ parameter vector \mathbf{p} containing the physical property of all equivalent sources are updated iteratively. During each subsequent iteration, Cordell's method either incorporates a single equivalent source or adjusts an existing equivalent source to match the maximum amplitude of the current residual field. The convergence occurs when all of the residuals are bounded by an envelope of prespecified expected error. At the end, the algorithm produces an estimate $\tilde{\mathbf{p}}$ for the parameter vector yielding a predicted potential field \mathbf{f} (equation 3) satisfactorily fitting the observed data \mathbf{d} according to a given criterion. Note that the method proposed 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 single column of \mathbf{G} (equation 3) is used. An advantage of this *column-action update approach* is that the full matrix \mathbf{G} is never stored.

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

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

The total number of flops in Algorithm 4 consists in finding the maximum absolute value in vector \mathbf{r} (line 6) before the while loop. Per iteration, there is a `saxpy` (line 11) and another search for the maximum 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., ?, 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,

476 called equivalent data, that contributes effectively to the final solution and fits the remaining observations
 477 (redundant data). Iteratively, Mendonça and Silva (1994) selected the subset of equivalent data that is
 478 substantially smaller than the original dataset. This selection is carried out by incorporating one data point
 479 at a time.

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

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

491 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,
 492 respectively. Mendonça and Silva (1994) designate \mathbf{d}_e and \mathbf{d}_r as, respectively, *equivalent* and *redundant*
 493 data. With the exception of a normalization strategy, Mendonça and Silva (1994) calculate a $P \times 1$ estimated
 494 parameter vector $\tilde{\mathbf{p}}$ by solving an underdetermined problem (equation 23) involving only the equivalent
 495 data \mathbf{d}_e (equation 39) for the particular case in which $\mathbf{H} = \mathbf{W}_p = \mathbf{I}_P$ (equations 9 and 13), $\mathbf{W}_d = \mathbf{I}_{D_e}$
 496 (equation 12) and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14), which results in

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

497 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
 498 (1994) presume that the estimated parameter vector $\tilde{\mathbf{p}}$ obtained from equation 40 leads to a $D_r \times 1$ residuals
 499 vector

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

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

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

507 7.4 Reparameterization

508 Another approach for improving the computational performance of equivalent-layer technique consists
 509 in setting a $P \times Q$ reparameterization matrix \mathbf{H} (equation 9) with $Q \ll P$. This strategy has been used
 510 in applied geophysics for decades (e.g., Skilling and Bryan, 1984; Kennett et al., 1988; Oldenburg et al.,

511 1993; Barbosa et al., 1997) and is known as *subspace method*. The main idea relies in reducing the linear
 512 system dimension from the original P -space to a lower-dimensional subspace (the Q -space). An estimate
 513 $\tilde{\mathbf{q}}$ for the reparameterized parameter vector \mathbf{q} is obtained in the Q -space and subsequently used to obtain
 514 an estimate $\tilde{\mathbf{p}}$ for the parameter vector \mathbf{p} (equation 3) in the P -space by using equation 9. Hence, the key
 515 aspect of this *reparameterization approach* is solving an appreciably smaller linear inverse problem for $\tilde{\mathbf{q}}$
 516 than that for the original parameter vector $\tilde{\mathbf{p}}$ (equation 3).

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

525 The method proposed by Oliveira Jr. et al. (2013) consists in solving an overdetermined problem (equation
 526 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
 527 that

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

528 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
 529 regularization (e.g., ?, p. 103). Note that, in this case, the prior information is defined in the P -space for the
 530 original parameter vector \mathbf{p} and then transformed to the Q -space. Another characteristic of their method is
 531 that it is valid for processing irregularly-spaced data on an undulating surface.

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

537 Before Oliveira Jr. et al. (2013) and ?, Barnes and Lumley (2011) also proposed a computationally
 538 efficient method for equivalent-layer technique based on reparameterization. A key difference, however,
 539 is that Barnes and Lumley (2011) did not set a $P \times Q$ reparameterization matrix \mathbf{H} (equation 9) with
 540 $Q \ll P$. Instead, they used a matrix \mathbf{H} with $Q \approx 1.7 P$. Their central idea is setting a reparameterization
 541 scheme that groups distant equivalent sources into blocks by using a bisection process. This scheme
 542 leads to a quadtree representation of the physical-property distribution on the equivalent layer, so that
 543 matrix \mathbf{GH} (equation 10) is notably sparse. Barnes and Lumley (2011) explore this sparsity in solving the
 544 overdetermined problem for $\tilde{\mathbf{q}}$ (equation 42) via conjugate-gradient method (e.g., Golub and Loan, 2013,
 545 sec. 11.3).

546 We consider an algorithm (not shown) that solves the overdetermined problem (equation 22) by combining
 547 the reparameterization with CGLS method (Algorithm 1). It starts with a reparameterization step defined
 548 by defining a matrix $\mathbf{C} = \mathbf{G} \mathbf{H}$ (equation 10). Then, the CGLS (Algorithm 1) is applied by replacing \mathbf{G}
 549 with \mathbf{C} . In this case, the linear system is solved by the reparameterized parameter vector $\tilde{\mathbf{q}}$ instead of $\tilde{\mathbf{p}}$.
 550 At the end, the estimated $\tilde{\mathbf{q}}$ is transformed into $\tilde{\mathbf{p}}$ (equation 15). Compared to the original CGLS shown
 551 in Algorithm 1, the algorithm discussed here has the additional flops associated with the matrix-matrix
 552 product to compute \mathbf{C} and the matrix-vector product of equation 15 outside the while loop. Then, according

553 to table 1, the total number of flops given by:

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

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

558 **7.5 Wavelet compression**

559 Previously to Barnes and Lumley (2011), the idea of transforming the dense matrix \mathbf{G} (equation 3) into a
 560 sparse one has already been used in the context of equivalent-layer technique. Li and Oldenburg (2010)
 561 proposed a method that applies the discrete wavelet transform to introduce sparsity into the original dense
 562 matrix \mathbf{G} . Those authors approximate a planar grid of potential-field data by a regularly-spaced grid of
 563 equivalent sources, so that the number of data D and sources P is the same, i.e., $D = P$. Specifically, Li
 564 and Oldenburg (2010) proposed a method that applies the wavelet transform to the original dense matrix \mathbf{G}
 565 and sets to zero the small coefficients that are below a given threshold, which results in an approximating
 566 sparse representation of \mathbf{G} in the wavelet domain. They first consider the following approximation

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

567 where

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

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

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

570 with absolute value smaller than a given threshold.

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

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

573 and a normalized parameter vector

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

574 where \mathbf{L} is a diagonal and invertible matrix representing an approximation of the first-order Tikhonov
 575 regularization in the wavelet domain. Then they solve an overdetermined problem (equation 22) to obtain
 576 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),
 577 $\mathbf{W}_d = \mathbf{I}_D$ (equation 12) and $\bar{p} = 0$ (equation 14) via conjugate-gradient method (e.g., Golub and Loan,
 578 2013, sec. 11.3). Finally, Li and Oldenburg (2010) compute an estimate $\tilde{\mathbf{p}}$ for the original parameter vector
 579 given by

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

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

582 7.6 Iterative methods using the full matrix G

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

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

600 Recently, Jirigalatu and Ebbing (2019) presented another iterative method for estimating a parameter
 601 vector $\tilde{\mathbf{p}}$ (equation 3). With the purpose of combining different potential-field data, their method basically
 602 modifies that shown in Algorithm 6 by changing the initial approximation and the iterative correction for
 603 the parameter vector. Specifically, Jirigalatu and Ebbing (2019) replace line 5 by $\tilde{\mathbf{p}} = \mathbf{0}$, where $\mathbf{0}$ is a vector
 604 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
 605 this modified approach requires two matrix-vector products involving the full matrix \mathbf{G} per iteration. To
 606 overcome the high computational cost of these two products, Jirigalatu and Ebbing (2019) set an equivalent
 607 layer formed by prisms and compute their predicted potential field in the wavenumber domain by using the
 608 Gauss-FFT technique ?.

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

$$f_{S0B17} = 2D^2 + 2D + \text{ITMAX} (2D^2 + 3D) . \quad (50)$$

613 Note that the number of flops per iteration in f_{S0B17} (equation 50) has the same order of magnitude, but is
 614 smaller than that in f_{CGLS} (equation 28).

615 7.7 Iterative deconvolution

616 Recently, ?? proposed the *convolutional equivalent-layer method*, which explores the structure of the
 617 sensitivity matrix \mathbf{G} (equation 3) for the particular case in which (i) there is a single equivalent source right
 618 below each potential-field datum and (ii) both data and sources rely on planar and regularly spaced grids.
 619 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
 620 horizontal plane z_0 . The data indices i may be ordered along the x - or y -direction, which results in an
 621 x - or y -oriented grid, respectively. They also consider a single equivalent source located right below each

622 datum, at a constant vertical coordinate $z_0 + \Delta z$, $\Delta z > 0$. In this case, the number of data and equivalent
 623 sources are equal to each other (i.e., $D = P$) and \mathbf{G} (equation 3) assumes a *doubly block Toeplitz* (?, p. 28)
 624 or *block-Toeplitz-block* (BTB) (?, p. 67) structure formed by $N_B \times N_B$ blocks, where each block
 625 has $N_b \times N_b$ elements, with $D = N_B N_b$. This particular structure allows formulating the product of \mathbf{G}
 626 and an arbitrary vector as a *fast discrete convolution* via *Fast Fourier Transform* (FFT) (Van Loan, 1992,
 627 section 4.2).

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

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

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

634 Consider the matrix-vector products

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

635 and

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

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

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

638 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
 639 having N_b elements. Equations 52 and 53 can be computed in terms of an auxiliary matrix-vector product

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

640 where

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

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

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

and \mathbf{G}_c is a $4D \times 4D$ *doubly block circulant* (? , p. 28) or *block-circulant circulant-block* (BCCB) (? , 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.

657 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
 658 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
 659 to use the fact that BCCB matrices are diagonalized by the 2D unitary discrete Fourier transform (DFT)
 660 (e.g., Davis, 1979, p. 31). Because of that, \mathbf{G}_c can be written as

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

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

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

666 By following ?, we rearrange equation 63 as follows

$$\mathcal{L} \circ (\mathcal{F}_{2N_B} \mathcal{V}_c \mathcal{F}_{2N_b}) = \mathcal{F}_{2N_B} \mathcal{W}_c \mathcal{F}_{2N_b} \quad (64)$$

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

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

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

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

673 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
 674 55). It is important noting that the matrices \mathcal{C} and \mathcal{L} (equation 66) associated with the BTTB matrix \mathbf{G}
 675 (equation 51) are different from those associated with \mathbf{G}^\top .

676 The whole procedure to compute the original matrix-vector products $\mathbf{G}\mathbf{v}$ (equation 52) and $\mathbf{G}^\top \mathbf{v}$
 677 (equation 53) consists in (i) rearranging the elements of the vector \mathbf{v} and the first column $\mathbf{G}[:, 1]$ of matrix
 678 \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
 679 $\mathcal{F}_{2N_B}^* \mathcal{A} \mathcal{F}_{2N_b}^*$, where \mathcal{A} is a given matrix, and a Hadamard product to obtain \mathcal{W}_c (equation 65); and (iii)
 680 retrieve the elements of vector \mathbf{w} (equation 52) from \mathcal{W}_c (equation 65). It is important noting that the steps
 681 (i) and (iii) do not have any computational cost because they involve only reorganizing elements of vectors
 682 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
 683 2D Discrete Fourier Transform (2D-DFT) and the 2D Inverse Discrete Fourier Transform (2D-IDFT) of \mathcal{A} .
 684 These transforms can be efficiently computed by using the 2D Fast Fourier Transform (2D-FFT). Hence,
 685 the original matrix-vector products $\mathbf{G}\mathbf{v}$ (equation 52) and $\mathbf{G}^\top \mathbf{v}$ (equation 53) can be efficiently computed
 686 by using the 2D-FFT.

687 Algorithms 7 and 8 show pseudo-codes for the convolutional equivalent-layer method proposed by ??.
 688 Note that those authors formulate the overdetermined problem (equation 22) of obtaining an estimate $\tilde{\mathbf{p}}$ for
 689 the parameter vector \mathbf{p} (equation 3) as an *iterative deconvolution* via *conjugate gradient normal equation*
 690 *residual* (CGNR) Golub and Loan (2013, sec. 11.3) or *conjugate gradient least squares* (CGLS) (?, p. 165)
 691 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)
 692 and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14). As shown by ??, the CGLS produces stable estimates $\tilde{\mathbf{p}}$ for the parameter vector
 693 \mathbf{p} (equation 3) in the presence of noisy potential-field data \mathbf{d} . This is a well-known property of the CGLS
 694 method (e.g., ?, p. 166).

695 The key aspect of Algorithm 7 is replacing the matrix-vector products of CGLS (Algorithm 1) by
 696 fast convolutions (Algorithm 8). A fast convolution requires one 2D-DFT, one 2D-IDFT and an entryw-
 697 ise product of matrices. We consider that the 2D-DFT/IDFT are computed with 2D-FFT and requires
 698 $\kappa(4D) \log_2(4D)$ flops, where $\kappa = 5$ is compatible with a radix-2 FFT (Van Loan, 1992, p. 16), and the
 699 entrywise product $24D$ flops because it involves two complex matrices having $4D$ elements (Golub and
 700 Loan, 2013, p. 36). Hence, Algorithm 8 requires $\kappa(16D) \log_2(4D) + 26D$ flops, whereas a conventional
 701 matrix-vector multiplication involving a $D \times D$ matrix requires $2D^2$ (table 1). Finally, Algorithm 7 requires
 702 two 2D-FFTs (lines 4 and 5), one fast convolution and an inner product (line 8) previously to the while
 703 loop. Per iteration, there are three saxpys (lines 12, 15 and 16), two inner products (lines 14 and 17) and
 704 two fast convolutions (lines 13 and 17), so that:

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

705 7.8 Direct deconvolution

706 The method proposed by ?? can be reformulated to avoid the iterations of the conjugate gradient method.
 707 This alternative formulation consists in considering that $\mathbf{v} = \mathbf{p}$ and $\mathbf{w} = \mathbf{d}$ in equation 52, where \mathbf{p} is the
 708 parameter vector (equation 3) and \mathbf{d} the observed data vector. In this case, the equality “=” in equation 52
 709 becomes an approximation “ \approx ”. Then, equation 64 is manipulated to obtain

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

710 where

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

711 $\mathbf{1}$ is a $4D \times 4D$ matrix of ones, “ \oslash ” denotes entrywise division and ζ is a positive scalar. Note that $\zeta = 0$
 712 leads to $\mathbf{1} \oslash \mathcal{L}$. In this case, the entrywise division may be problematic due to the elements of \mathcal{L} having
 713 absolute value equal or close to zero. So, a small ζ is set to avoid this problem in equation 69. Next, we use
 714 $\check{\mathcal{L}}$ to obtain a matrix \mathcal{V}_c from equation 68. Finally, the elements of the estimated parameter vector $\tilde{\mathbf{p}}$ are
 715 retrieved from the first quadrant of \mathcal{V}_c . This procedure represents a *direct deconvolution* (e.g., ?, p. 220)
 716 using a *Wiener filter* (e.g., ?, p. 263).

717 The required total number of flops associated with the direct deconvolution aggregates one 2D-FFT
 718 to compute matrix \mathcal{L} (equation 66), one entrywise product $\mathcal{L} \circ \mathcal{L}^*$ involving complex matrices and one
 719 entrywise division to compute $\check{\mathcal{L}}$ (equation 69) and a fast convolution (Algorithm 8) to evaluate equation
 720 68, which results in:

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

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

8 SYNTHETIC DATA SIMULATIONS

For all applications, we generate a model composed by two spheres and a polygonal prism in a regular spaced grid of 50×50 . The upper left sphere has a density contrast of 600 kg/m^3 , the right upper sphere a negative contrast of -500 kg/m^3 and the bottom prism is equal to 550 kg/m^3 . To generate the magnetic data, the bodies are in the same position and all of them have the same magnetization intensity and direction (3.46 A/m intensity, 35.26° inclination and 45.0° declination) within a simulated geomagnetic field direction of 20.0° inclination and 35.0° declination. These synthetic datas are shown in figures 3 and 7, respectively.

8.1 Stability analysis

For the stability analysis we show the comparison of the Cholesky factorization (equations 25 and 26) to solve overdetermined problems (equation 22), the iterative deconvolutional method (algorithms 7 and 8) and the deconvolutional method with different values for the Wiener stabilization (equation 69). We create 21 data sets, for both gravity and magnetic data, adding a crescent pseudo-random noise to the original data, which varies from 0% to 10% of the maximum anomaly value in intervals of 0.5%. These noises has mean equal to zero and a Gaussian distribution. These synthetic datas are shown in figures 3 and 7, where panel (A) of each figure represents the noise free data and panel (B) is the maximum noised data for gravity and magnetic, respectively.

Figure 2 shows how the euclidian norm of the equivalent sources residuals varies as the level of the noise is increased for the gravimetric data. We can see that for all methods, a linear tendency can be observed as it is expected. The inclination of the straight line is a indicative of the stability of each method. As show in the graph the deconvolutional method is very unstable and it is really necessary to use a stabilization method to have a good parameter estimative. In contrast, a correct value of the stabilization parameter is necessary to not overshoot the smoothness of the solution as it is the case for the zeroth-order Tikhonov regularization as well. Using this gravimetric data, the optimal value for the Wiener stabilization parameter is $\mu = 10^{-20}$.

Figure 4 shows the comparison of the predicted data for each method with the original data (figure 3) using the most noised-corrupted data from the set of the stability analysis. The classical with zeroth-order Tikhonov regularization and the convolutional methods (figures 4(A) and 4(B)) yield very similar results for the predicted data confirming its similarities with the stabilization despite the bid difference in floating-point operations. Figure 4(C) shows the deconvolutional method without a stabilization and demonstrates the necessity to use it for this method. Figure 4(D) shows the deconvolutional method with Wiener stabilization $\mu = 10^{-15}$ which is too high, demonstrating the over smoothness of the predicted data. Figures 4(E) and 4(F) shows the predicted data for an optimal value of the Wiener parameter $\mu = 10^{-20}$ and a low value $\mu = 10^{-25}$, respectively.

The upward continuation is a processing technique to visualize the data in a higher altitude. In practice is expected a lower amplitude signal and a smoother data as the high frequency anomalies tends to disappear. Figure 5(A) shows the true modeled upward data at an height of -500 m . Figures 5(B), (C), (D) and (E) show the result of the upward processing for the classical, convolutional, deconvolutional and the deconvolutional with Wiener parameter $\mu = 10^{-20}$, respectively. It is clear that all methods seems to predict the upward data very reasonable, except the deconvolutional method without stabilization.

For the magnetic data, figure 6 shows a very similar behavior of the stability as the previous case. The Wiener parameter seems to have the best solution for $\mu = 10^{-13}$. For both types of data the best Wiener

770 parameter seems to be one that produces a low slope for the straight line in the stability analysis, discordant
771 from the classical and convolutional methods.

772 Figure 8 shows the comparison of the predicted data for each method with the original magnetic data in
773 figure 7 using the most noised-corrupted data modeled from the stability analysis. As the previous case the
774 classical (figure 8(A)) and the convolutional (figure 8(B)) methods have very similar predicted data but
775 estimated with less orders of magnitude in floating-point operations. The deconvolutional (figure 8(C)) have
776 have a strong disagreement with the observed data showing the need for a stabilization method. Figure
777 8(D) has a value of $\mu = 10^{-10}$ and the predicted data became to smooth by it. The optimal value of the
778 Wiener parameter is shown in figure 8(E) with $\mu = 10^{-13}$ and figure 8(F) shows a predicted data with a
779 low stablization value with $\mu = 10^{-16}$.

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

9 REAL DATA RESULTS

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

788 9.1 Geological setting

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

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

813 9.2 Potential field data applications

814 Here we compare the performance of the convolutional and deconvolutional algorithms in a real potential
815 field data set. We focus the application on a region in the Southeast of the State of Pará. The aeromagnetic
816 data were acquired by the Geological Survey of Brazil-CPRM. The survey area covers $\approx 58000 \text{ km}^2$ with
817 high-resolution gravity and magnetic data. The flight and the tie lines were acquired and spaced at 3 km
818 and 12 km oriented in the directions $N - S$ and $E - W$, respectively, with a mean flight height of 900 m
819 above the ground. For both applications, we interpolated gravity and magnetic anomalies data, calculating
820 the data set in a grid of 1000×500 ($N = 500000$ observation points) at the same mean flight height. About
821 the computational resources, we processed both data on an Intel Core i7 7700HQ@2.8 GHz processor and
822 16GB RAM. We show in Figure 10 and Figure 12 the interpolated aerogravimetric and aeromagnetic data,
823 respectively. We also use the same equivalent layer grid configuration in gravity and magnetic applications.

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

826 We apply both strategies to the gravimetric case. We set a depth for the equivalent layer equal to 1200 m
827 below the observation plane. Figure 11A and Figure 11C show the predicted data for convolutional and
828 deconvolutional strategies. The residual maps (the difference between the observed and predicted data)
829 are show in figures 11B and 11D for the convolutional and deconvolutional equivalent-layer technique,
830 respectively. For the convolutional case, the mean residual and standard deviation values are $\approx 0.00\text{ mGal}$
831 and $\approx 0.15\text{ mGal}$, respectively. For the deconvolutional case, the mean residual and standard deviation
832 values are $\approx 0.46\text{ mGal}$ and $\approx 1.23\text{ mGal}$, respectively. These last results show that the estimated density
833 distributions (not shown) fit the observed data for both applications. To show the performance of the
834 algorithms, we performed an upward continuation by using the estimated density distributions (figures 11E
835 and 11F). There is a little difference on the processing time between both strategies. The convolutional
836 algorithm took $\approx 9.18\text{s}$ and the deconvolutional algorithm took $\approx 0.53\text{s}$. We conclude that both strategies
837 are capable of processing gravimetric observations from large areas with dense coverage data. Despite a
838 little difference in processing time, the deconvolutional equivalent-layer technique proved to be faster than
839 the convolutional strategy.

840 Finally, we test the convolutional and deconvolutional algorithms for processing total-field anomalies.
841 We stress that the Carajás area is very large and the main field direction varies significantly. For this reason,
842 we consider a mean direction for the main field equal to -19.865° and -7.43915° (the same as the mid
843 location of the area) for the inclination and declination, respectively. Furthermore, we are not considering
844 the knowledge about the magnetization direction of the sources, and choose a magnetization direction for
845 the equivalent layer equal to the main field direction. For this application, we set a depth of 900 m (below
846 the observation plane) for the equivalent layer. Figure 12A and Figure 12C show the predicted data for
847 convolutional and deconvolutional algorithms. The residual maps (the difference between the observed and
848 predicted data) are show in figures 12B and 12D for the convolutional and deconvolutional techniques,
849 respectively. The convolutional equivalent layer produced a mean residual and standard deviation values
850 of $\approx 0.06\text{ nT}$ and $\approx 1.97\text{ nT}$, respectively. The deconvolutional algorithm produced a mean residual
851 and standard deviation values of $\approx 18.99\text{ nT}$ and $\approx 33.64\text{ nT}$, respectively. To show the performance
852 of the algorithms, we performed an upward continuation (figures 12E and 12F) by using the estimated
853 magnetic-moment distributions (not shown). Similarly to the gravity application, the deconvolutional
854 equivalent layer presents faster results than the convolutional algorithm. The deconvolutional and the
855 convolutional approaches took $\approx 0.89\text{s}$ and $\approx 82.08\text{s}$, respectively. Despite the difference between the
856 processing time of both strategies and considering the mean value of residuals and standard deviations, we
857 conclude that the convolutional strategy fits the observation data better than the deconvolutional approach.

10 DISCUSSION AND CONCLUSION

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

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

871 The second and third strategies, referred to as the column-action and row-action updates, involve
872 iteratively calculating a single column and a single row of the sensitivity matrix, respectively. By following
873 the column-action update strategy, a single column of the sensitivity matrix is calculated during each
874 iteration. This implies that a single equivalent source contributes to the fitting of data in each iteration.
875 Conversely, in the row-action update strategy, a single row of the sensitivity matrix is calculated per
876 iteration, which means that one potential-field observation is incorporated in each iteration, forming a new
877 subset of equivalent data much smaller than the original data. Both strategies (column- and row-action
878 updates) have a great advantage because a single column or a single row of the sensitivity matrix is
879 calculated iteratively. However, to our knowledge, the strategy of the column-action update presents some
880 issues related to convergence, and the strategy of the row-action update can also have issues if the number
881 of equivalent data is not significantly smaller than the original number of data points.

882 The fourth strategy is the wavelet compression which consists in transforming a large and full sensitivity
883 matrix into a sparse one, with a few nonzero elements, by using a fast wavelet transform. By using the sparse
884 sensitivity matrix, the inverse problem is solved in the wavelet domain without an explicit regularization
885 parameter. A regularized solution is obtained by using a conjugate gradient least squares where the number
886 of iterations performed acts as a regularization. Computationally, the significant advantage of the wavelet-
887 compression strategy is its ability to handle a sparse sensitivity matrix, which enables a rapid iteration
888 of the conjugate gradient (CG) algorithm. However, we understand that this strategy requires computing
889 the full and dense sensitivity matrix, which can be considered a drawback when processing large-scale
890 potential-field data.

891 The fifth strategy is the reparametrization of the original parameters to be estimated in the equivalent-layer
892 technique. In this strategy, the developed equivalent-layer methods reduce the dimension of the linear
893 system of equations to be solved by estimating a lower-dimensional parameter vector. We highlight three
894 methods that used the reparametrization strategy: i) the quadtree discretization of the equivalent layer; ii)
895 the polynomial equivalent layer (PEL) and; ii) the lower-dimensional subspace of the equivalent layer.
896 In the quadtree discretization, the equivalent sources located far from the observation point are grouped
897 together to create a larger averaged source implying in a reduction in the number of parameters to be
898 estimated. In the PEL, there is an explicit reparametrization of the equivalent layer by representing the
899 unknown distribution over the equivalent layer as a set of piecewise-polynomial functions defined on a set of

900 equivalent-source windows. The PEL method estimates the polynomial coefficients of all equivalent-source
901 windows. Hence, PEL reduces the dimension of the linear system of equations to be solved because the
902 polynomial coefficients within all equivalent-source windows are much smaller than both the number of
903 equivalent sources and the number of data points. In the lower-dimensional subspace of the equivalent layer,
904 there is an implicit reparametrization of the equivalent layer by reducing the linear system dimension
905 from the original and large-model space to a lower-dimensional subspace. The lower-dimensional subspace
906 is grounded on eigenvectors of the matrix composed by the gridded data set. The main advantage of the
907 reparametrization of the equivalent layer is to deal with lower-dimensional linear system of equations.
908 However, we acknowledge that this strategy may impose an undesirable smoothing effect on both the
909 estimated parameters over the equivalent layer and the predicted data.

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

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

930 The eighth strategy is a direct deconvolution method, which is a mathematical process very common in
931 geophysics. However, to our knowledge, direct deconvolution has never been used to solve the inverse
932 problem associated with the equivalent-layer technique. From the mathematical expressions in the iterative
933 deconvolutional equivalent layer with BTTB matrices, direct deconvolution arises naturally since it is an
934 operation inverse to convolution. The main advantage of applying the direct deconvolution strategy in
935 the equivalent layer is that it avoids, for example, the iterations of the CG algorithm. However, the direct
936 deconvolution is known to be an unstable operation. To mitigate this instability, the Wiener deconvolution
937 method can be adopted.

938 ***** PAREI AQUI *****

939 ***** O TEXTO ABAIXO PERTENCE A VERSAO ANTERIOR DA CONCLUSAO *****

940 From the above-mentioned strategies, further classifications may be identified for reducing the com-
941 putational cost of equivalent-layer technique. For example, taking into account the mathematical bases,
942 we identify four groups: i) the reduction of the dimensionality of the linear system of equations to be

943 solved; ii) the generation of a sparse linear system of equations to be solved; iii) the explicit iterative
944 method without solving a linear system of equations and; iv) the convolution (deconvolution). The first
945 mathematical basis reduces the linear system of equations to be solved for estimating the distribution over
946 the equivalent layer. This is achieved by using different strategies: a) the moving data-window scheme
947 spanning the data set by setting a small moving-data window; b) the reparametrization of the dataset by
948 selecting a subset of observations much smaller than the original data; c) the explicit reparametrization of
949 the model (the equivalent layer) by using quadtree discretization or piecewise-polynomial functions defined
950 on a set of equivalent-source windows and; d) the implicit reparametrization of the model (the equivalent
951 layer) by using the subspace method. The second mathematical basis generates a sparse linear system of
952 equations by transforming the full sensitivity matrix into a sparse one. This is achieved by using different
953 strategies: a) the compression of the coefficient of the sensitivity matrix via wavelet transforms and; b)
954 the grouping of equivalent sources distant from an observation point to form a larger equivalent source
955 via quadtree discretization of the equivalent layer; The third mathematical basis does not solve a linear
956 system of equations for estimating the distribution over the equivalent layer. This is grounded on gradient
957 method as an optimization algorithm that iteratively updates the parameter without calculating a full
958 Hessian matrix and solving linear systems. This is achieved by using different strategies: a) the conjugate
959 gradient least-squares regularized by the number of iterations with the sparse wavelet compression of
960 the coefficient matrix; b) the iterative Landweber algorithm with forward modelling of potential-field
961 data with Gauss-FFT; c) the gradient-boosting algorithm operating on overlapping windows with the
962 block-averaged sources reducing the number of the sources; and d) The explicit iterative method with (or
963 not) physical reasoning without calculating a full Hessian matrix and solving linear systems The fourth
964 mathematical basis for estimating the distribution over the equivalent layer is based on FFT convolution
965 with the sensitivity matrices exhibiting BTTB structure when the observations and equivalent sources are
966 aligned on a horizontal and regularly-spaced grid.

967 We would like to draw the readers' attention to the possibility of grouping the aforementioned strategies
968 for reducing the computational cost of the equivalent-layer technique according to different mathematical
969 bases. An example of the mathematical basis might be the reduction of the processing time spent on the
970 forward modelling accounts.

CONFLICT OF INTEREST STATEMENT

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

AUTHOR CONTRIBUTIONS

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

FUNDING

978 Diego Takahashi was supported by a Post-doctoral scholarship from CNPq (grant 300809/2022-0) Valéria
979 C.F. Barbosa was supported by fellowships from CNPq (grant 309624/2021-5) and FAPERJ (grant
980 26/202.582/2019). Vanderlei C. Oliveira Jr. was supported by fellowships from CNPq (grant 315768/2020-
981 7) and FAPERJ (grant E-26/202.729/2018).

ACKNOWLEDGMENTS

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

DATA AVAILABILITY STATEMENT

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

REFERENCES

- 986 Amaral, G. (1974). *Geologia Pré-Cambriana da Região Amazônica*. Ph.D. thesis, Universidade de São
987 Paulo
- 988 Aster, R. C., Borchers, B., and Thurber, C. H. (2018). *Parameter estimation and inverse problems*
989 (Elsevier)
- 990 Barbosa, V. C. F., Silva, J. B., and Medeiros, W. E. (1997). Gravity inversion of basement relief using
991 approximate equality constraints on depths. *Geophysics* 62, 1745–1757
- 992 Barnes, G. and Lumley, J. (2011). Processing gravity gradient data. *Geophysics* 76, I33–I47. doi:10.1190/
993 1.3548548
- 994 Cordell, L. (1992). A scattered equivalent-source method for interpolation and gridding of potential-field
995 data in three dimensions. *Geophysics* 57, 629–636
- 996 Cunha, I. R., Dall’Agnol, R., and Feio, G. R. L. (2016). Mineral chemistry and magnetic petrology of the
997 archean planalto suite, carajás province – amazonian craton: Implications for the evolution of ferroan
998 archean granites. *Journal of South American Earth Sciences* 67, 100–121
- 999 Dampney, C. N. G. (1969). The equivalent source technique. *Geophysics* 34, 39–53. doi:10.1190/1.
1000 1439996

- 1001 Davis, K. and Li, Y. (2011). Fast solution of geophysical inversion using adaptive mesh, space-filling
1002 curves and wavelet compression. *Geophysical Journal International* 185, 157–166. doi:10.1111/j.
1003 1365-246X.2011.04929.x
- 1004 Davis, P. J. (1979). *Circulant matrices* (John Wiley & Sons, Inc.)
- 1005 Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics* ,
1006 1189–1232
- 1007 Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics and data analysis* 38,
1008 367–378
- 1009 Golub, G. H. and Loan, C. F. V. (2013). *Matrix Computations (Johns Hopkins Studies in the Mathematical
1010 Sciences)* (Johns Hopkins University Press), 4 edn.
- 1011 Guspí, F. and Novara, I. (2009). Reduction to the pole and transformations of scattered magnetic data using
1012 newtonian equivalent sources. *Geophysics* 74, L67–L73
- 1013 Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis* (Cambridge University Press), 1 edn.
- 1014 Jirigalatu, J. and Ebbing (2019). A fast equivalent source method for airborne gravity gradient data.
1015 *Geophysics* 84, G75–G82. doi:10.1190/GEO2018-0366.1
- 1016 Kellogg, O. D. (1929). *Foundations of Potential Theory* (Frederick Ungar Publishing Company)
- 1017 Kennett, B., Sambridge, M., and Williamson, P. (1988). Subspace methods for large inverse problems with
1018 multiple parameter classes. *Geophysical Journal International* 94, 237–247
- 1019 Leão, J. W. D. and Silva, J. B. C. (1989). Discrete linear transformations of potential field data. *Geophysics*
1020 54, 497–507. doi:10.1190/1.1442676
- 1021 Li, Y. and Oldenburg, D. W. (2010). Rapid construction of equivalent sources using wavelets. *Geophysics*
1022 75, L51–L59. doi:10.1190/1.3378764
- 1023 Mendonça, C. A. (2020). Subspace method for solving large-scale equivalent layer and density mapping
1024 problems. *Geophysics* 85, G57–G68. doi:10.1190/geo2019-0302.1
- 1025 Mendonça, C. A. and Silva, J. B. C. (1994). The equivalent data concept applied to the interpolation of
1026 potential field data. *Geophysics* 59, 722–732. doi:10.1190/1.1443630
- 1027 Moroni, M., Girardi, V., and Ferrario, A. (2001). The serra pelada au-pge deposit, serra dos carajás (pará
1028 state, brazil): geological and geochemical indications for a composite mineralising process. *Mineralium
1029 Deposita* 36, 768–785
- 1030 Neudecker, H. (1969). Some theorems on matrix differentiation with special reference to kronecker matrix
1031 products. *Journal of the American Statistical Association* 64, 953–963. doi:10.1080/01621459.1969.
1032 10501027
- 1033 Oldenburg, D., McGillivray, P., and Ellis, R. (1993). Generalized subspace methods for large-scale inverse
1034 problems. *Geophysical Journal International* 114, 12–20
- 1035 Oliveira Jr., V. C., Barbosa, V. C. F., and Uieda, L. (2013). Polynomial equivalent layer. *Geophysics* 78,
1036 G1–G13. doi:10.1190/geo2012-0196.1
- 1037 Salomao, G. N., Dall’Agnol, R., Angelica, R. S., Figueiredo, M. A., Sahoo, P. K., Medeiros-filho, C. A.,
1038 et al. (2019). Geochemical mapping and estimation of background concentrations in soils of carajás
1039 mineral province, eastern amazonian craton, brazil. *Geochemistry: Exploration, Environment, Analysis*
1040 19, 431–447
- 1041 Santos, J. O. S., Hartmann, L. A., Gaudette, H. E., Groves, D. I., Mcnaughton, M. J., and Fletcher, I. R.
1042 (2000). A new understanding of the provinces of the amazon craton based on integration of field mapping
1043 and u-pb and sm-nd geochronology. *Gondwana Research* 3, 453–488
- 1044 Siqueira, F. C., Oliveira Jr, V. C., and Barbosa, V. C. (2017). Fast iterative equivalent-layer technique for
1045 gravity data processing: A method grounded on excess mass constraint. *Geophysics* 82, G57–G69

- 1046 Skilling, J. and Bryan, R. (1984). Maximum entropy image reconstruction-general algorithm. *Monthly*
1047 *Notices of the Royal Astronomical Society*, Vol. 211, NO. 1, P. 111, 1984 211, 111
- 1048 Soler, S. R. and Uieda, L. (2021). Gradient-boosted equivalent sources. *Geophysical Journal International*
1049 227, 1768–1783. doi:10.1093/gji/ggab297
- 1050 Takahashi, D., Oliveira, V. C., and Barbosa, V. C. (2022). Convolutional equivalent layer for magnetic data
1051 processing. *Geophysics* 87, 1–59
- 1052 Takahashi, D., Oliveira Jr, V. C., and Barbosa, V. C. (2020). Convolutional equivalent layer for gravity data
1053 processing. *Geophysics* 85, G129–G141
- 1054 Tassinari, C. C. and Macambira, M. J. (1999). Geochronological provinces of the amazonian craton.
1055 *Episodes* 22, 174–182
- 1056 Teixeira, W., Tassinari, C., Cordani, U. G., and Kawashita, K. (1989). A review of the geochronology of
1057 the amazonian craton: Tectonic implications. *Precambrian Research* 42, 213–227
- 1058 Van Loan, C. (1992). *Computational Frameworks for the Fast Fourier Transform* (SIAM)
- 1059 Villas, R. N. and Santos, M. (2001). Gold deposits of the carajás mineral province: deposit types and
1060 metallogenesis. *Mineralium Deposita* 36, 300–331
- 1061 Xia, J. and Sprowl, D. R. (1991). Correction of topographic distortion in gravity data. *Geophysics* 56,
1062 537–541
- 1063 Xia, J., Sprowl, D. R., and Adkins-Heljeson, D. (1993). Correction of topographic distortions in potential-
1064 field data; a fast and accurate approach. *Geophysics* 58, 515–523. doi:10.1190/1.1443434

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  $\vartheta = \mathbf{G}^\top \mathbf{r}$  and  $\rho_0 = \vartheta^\top \vartheta$  ;
4 Set  $\tilde{\mathbf{p}} = \mathbf{0}$ ,  $\tau = 0$  and  $\eta = \mathbf{0}$  ;
5  $m = 1$  ;
6 while ( $\delta > \epsilon$ ) and ( $m < \text{ITMAX}$ ) do
7   Update  $\eta \leftarrow \vartheta + \tau \eta$  ;
8   Compute  $\nu = \mathbf{G} \eta$  ;
9   Compute  $v = \rho_0 / (\nu^\top \nu)$  ;
10  Update  $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + v \eta$  ;
11  Update  $\mathbf{r} \leftarrow \mathbf{r} - v \nu$  and  $\delta \leftarrow \|v \nu\|/D$  ;
12  Compute  $\vartheta = \mathbf{G}^\top \mathbf{r}$  and  $\rho = \vartheta^\top \vartheta$  ;
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  $\Delta z_i$  is a vertical distance controlling the depth of
  the  $i$ -th equivalent source,  $i \in \{1 : D\}$  ;
2 Set a tolerance  $\epsilon$  ;
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 ϵ ;
- 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., ?, p. 298) and σ 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 ϵ ;
- 3 Set a maximum number of iterations ITMAX ;
- 4 Set an auxiliary vector σ ;
- 5 Compute $\tilde{\mathbf{p}} = \sigma \circ \mathbf{d}$;
- 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 $\Delta\mathbf{p} = \sigma \circ \mathbf{r}$;
- 12 Update $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + \Delta\mathbf{p}$;
- 13 Compute $\nu = \mathbf{G} \Delta\mathbf{p}$;
- 14 Update $\mathbf{r} \leftarrow \mathbf{r} - \nu$;
- 15 Compute $\delta = \|\nu\|/D$;
- 16 $m \leftarrow m + 1$;
- 17 **end**

Algorithm 7: Generic pseudo-code for the convolutional equivalent-layer method proposed by ??.**Initialization :**

```

1 Set the regular grid of  $P$  equivalent sources on a horizontal plane  $z_0$  ;
2 Set a tolerance  $\epsilon$  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  $\mathcal{C}$ , compute its 2D-DFT via 2D-FFT and multiply by
    $\sqrt{4D}$  to obtain a matrix  $\mathcal{L}'$  (equation 66);
5 Rearrange the elements of  $\mathbf{G}[1, :]$  into matrix  $\mathcal{C}$ , compute its 2D-DFT via 2D-FFT and multiply by
    $\sqrt{4D}$  to obtain a matrix  $\mathcal{L}''$  (equation 66);
6 Set  $\tilde{\mathbf{p}} = \mathbf{0}$  ;
7 Set  $\mathbf{r} = \mathbf{d}$  and compute  $\delta = \|\mathbf{r}\|/D$  ;
8 Compute  $\vartheta = \mathbf{G}^\top \mathbf{r}$  (Algorithm 8) and  $\rho_0 = \vartheta^\top \vartheta$  ;
9 Set  $\tau = 0$  and  $\eta = \mathbf{0}$  ;
10  $m = 1$  ;
11 while ( $\delta > \epsilon$ ) and ( $m < \text{ITMAX}$ ) do
12   Update  $\eta \leftarrow \vartheta + \tau \eta$  ;
13   Compute  $\nu = \mathbf{G} \eta$  (Algorithm 8);
14   Compute  $v = \rho_0 / (\nu^\top \nu)$  ;
15   Update  $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + v \eta$  ;
16   Update  $\mathbf{r} \leftarrow \mathbf{r} - v \nu$  and  $\delta \leftarrow \|v \nu\|/D$  ;
17   Compute  $\vartheta = \mathbf{G}^\top \mathbf{r}$  (Algorithm 8) and  $\rho = \vartheta^\top \vartheta$  ;
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 \mathcal{L} (equation 66).

```

1 Rearrange the elements of  $\mathbf{v}$  (equations 52 and 54) into the matrix  $\mathcal{V}_c$  (equation 65);
2 Compute  $\mathcal{F}_{2N_B} \mathcal{V}_c \mathcal{F}_{2N_b}$  via 2D-FFT;
3 Compute the Hadamard product with matrix  $\mathcal{L}$  (equation 66);
4 Compute 2D-IDFT via 2D-FFT to obtain matrix  $\mathcal{W}_c$  (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 Loan (2013, p. 12). The 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 Loan, 2013, p. 4). The terms defined here are references to compute the total number of flops throughout the manuscript.

13 SUPPLEMENTARY TABLES AND FIGURES

1065 13.1 Figures

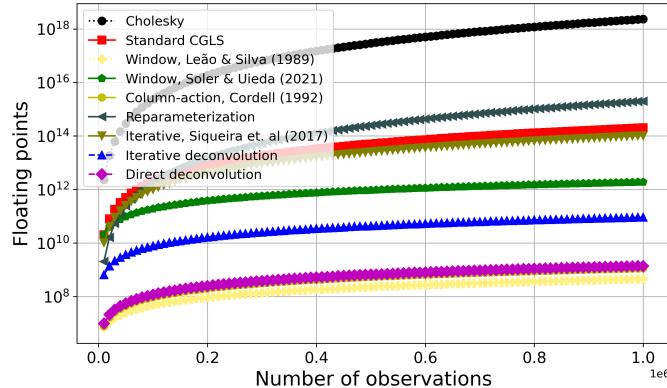


Figure 1. Number of *flops* for many of the methods described in this work to estimate the equivalent sources using gravity data. The range of observations varies from 10,000 to 1,000,000.

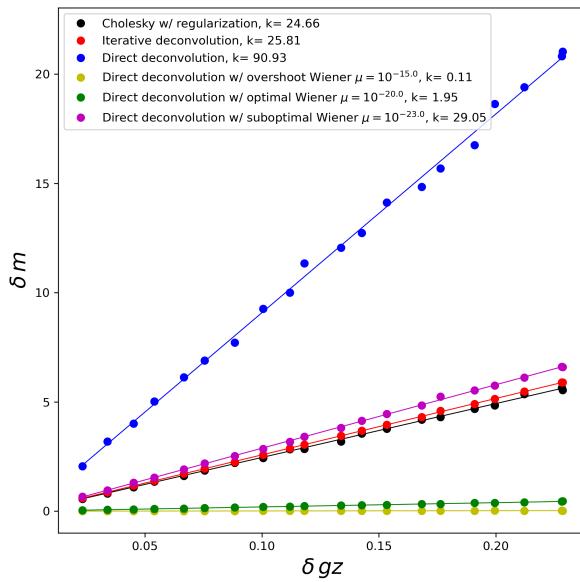


Figure 2. Stability analysis of some of the equivalent layer methods of the gravimetric case.

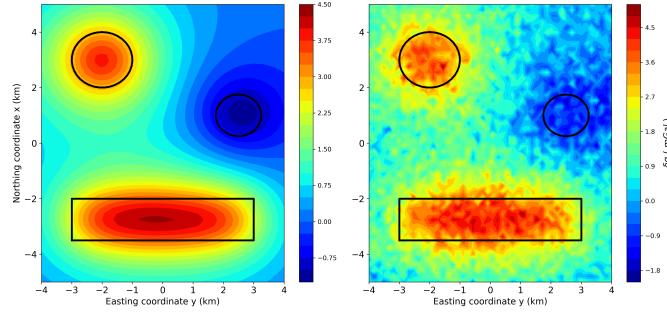


Figure 3. Synthetic data of the gravimetric case. The observations points are placed in a regular grid of 50×50 . Panel (A) shows the noise-free data and panel (B) shows the maximum noised data (10%).

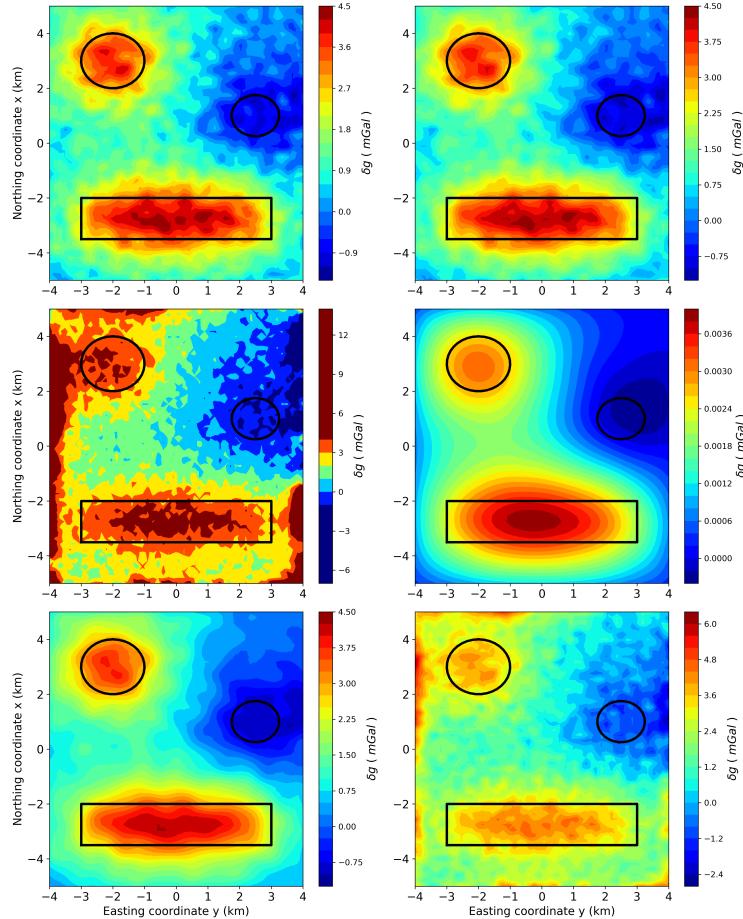


Figure 4. Predicted gravity data for different methods of the equivalent layer with maximum level of noise. Panel (A) is the classical method, (B) is the convolutional, (C) is the deconvolutional, (D) is the deconvolutional method using Wiener stabilization with a too high value for μ , (E) is the deconvolutional method using Wiener stabilization with a optimal value for μ and (F) is the deconvolutional method using Wiener stabilization with a too low value for μ .

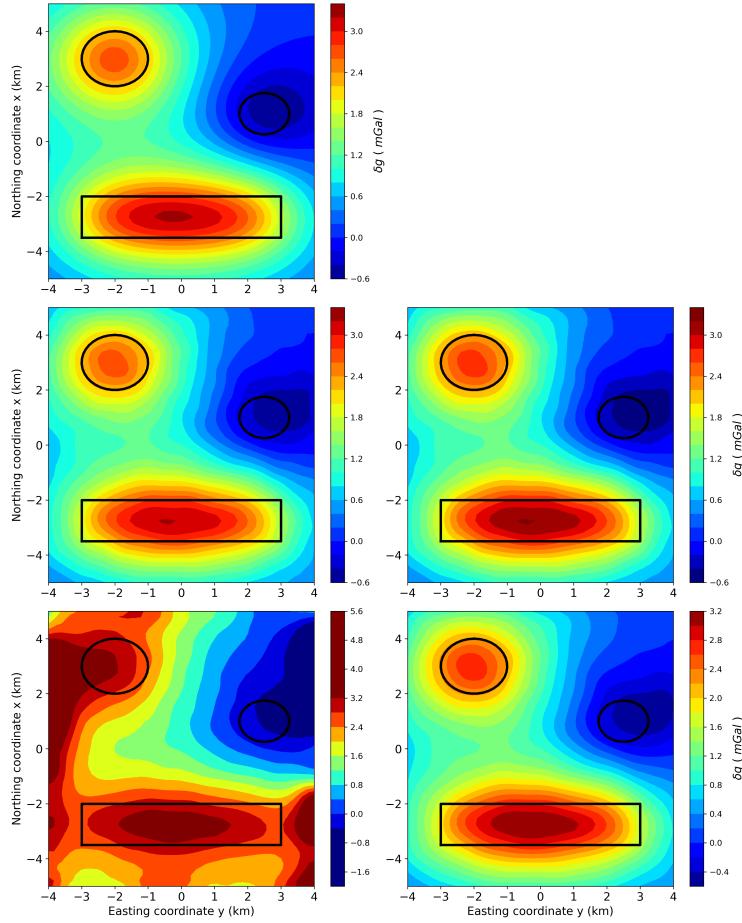


Figure 5. True noiseless upward gravimetric data at $z_i = -500$ m height and predicted data for different methods of the equivalent layer with maximum level of noise. Panel (A) is the true upward gravity data, Panel (B) is the classical method, (C) is the convolutional, (D) is the deconvolutional, (E) is the deconvolutional method using Wiener stabilization with a optimal value for $\mu = 10^{-20}$.

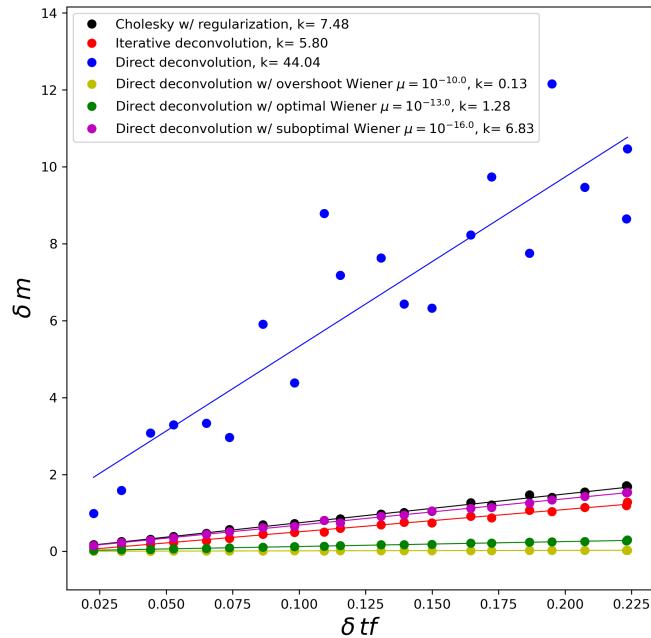


Figure 6. Stability analysis of some of the equivalent layer methods of the magnetic case.

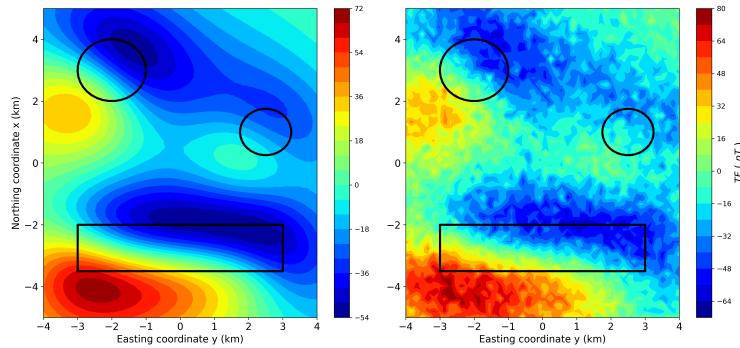


Figure 7. Synthetic data of the magnetic case. The observations points are placed in a regular grid of 50×50 . Panel (A) shows the noise-free data and panel (B) shows the maximum noised data (10%).

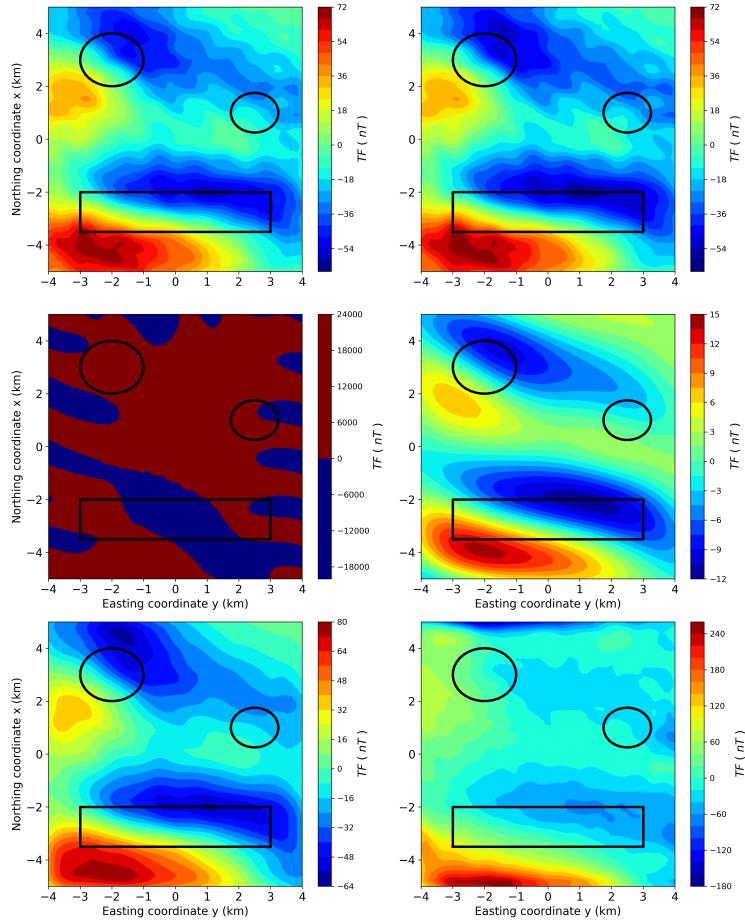


Figure 8. Predicted magnetic data for different methods of the equivalent layer with maximum level of noise. Panel (**A**) is the classical method, (**B**) is the convolutional, (**C**) is the deconvolutional, (**D**) is the deconvolutional method using Wiener stabilization with a too high value for μ , (**E**) is the deconvolutional method using Wiener stabilization with a optimal value for μ and (**F**) is the deconvolutional method using Wiener stabilization with a too low value for μ .

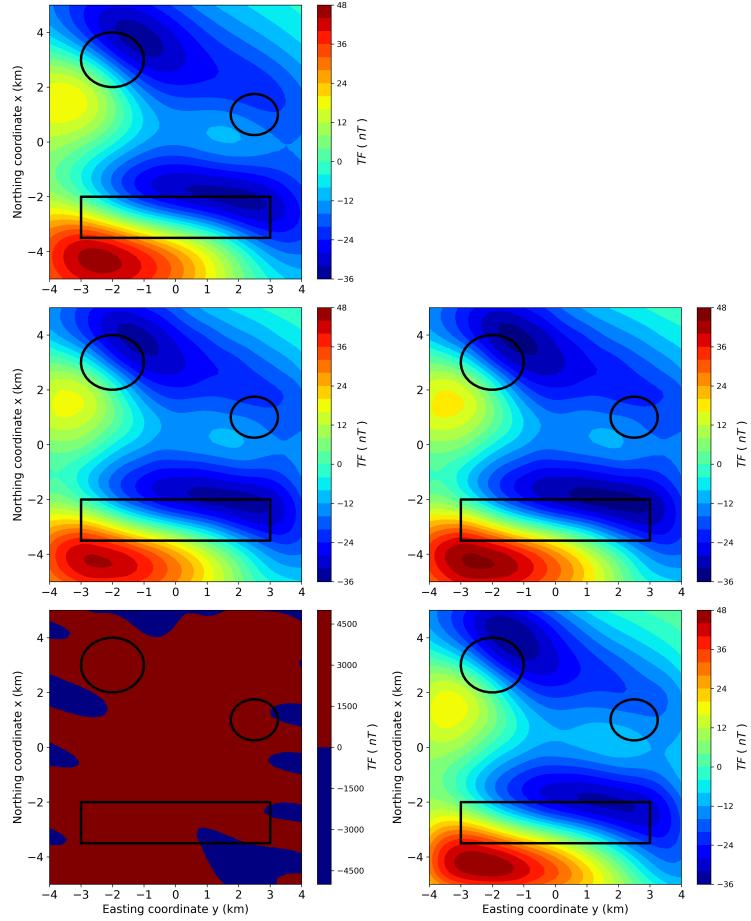


Figure 9. True noiseless upward magnetic data at $z_i = -1400$ m height and predicted data for different methods of the equivalent layer with maximum level of noise. Panel **(A)** is the true upward magnetic data, Panel **(B)** is the classical method, **(C)** is the convolutional, **(D)** is the deconvolutional, **(E)** is the deconvolutional method using Wiener stabilization with a optimal value for $\mu = 10^{-13}$.

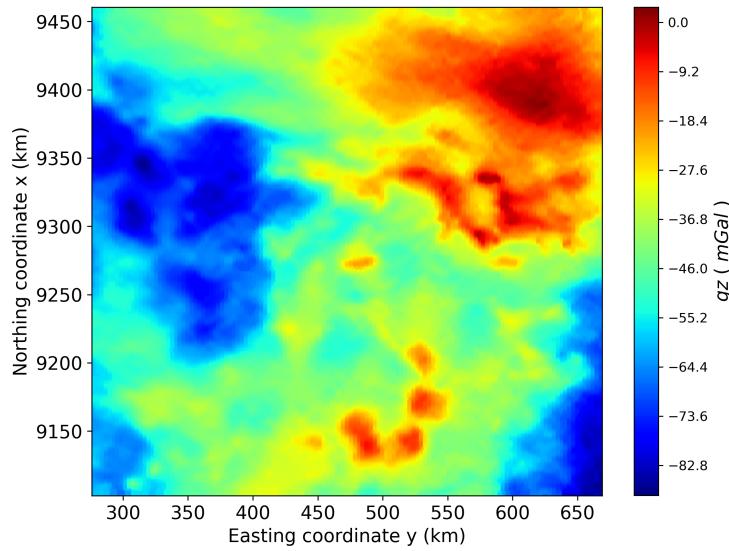


Figure 10. Gridded real aerogravimetric data from Carajás, Brazil. A regular grid of $1,000 \times 500$ is being used, totalizing $N, M = 500,000$ obsevation points and equivalent sources.

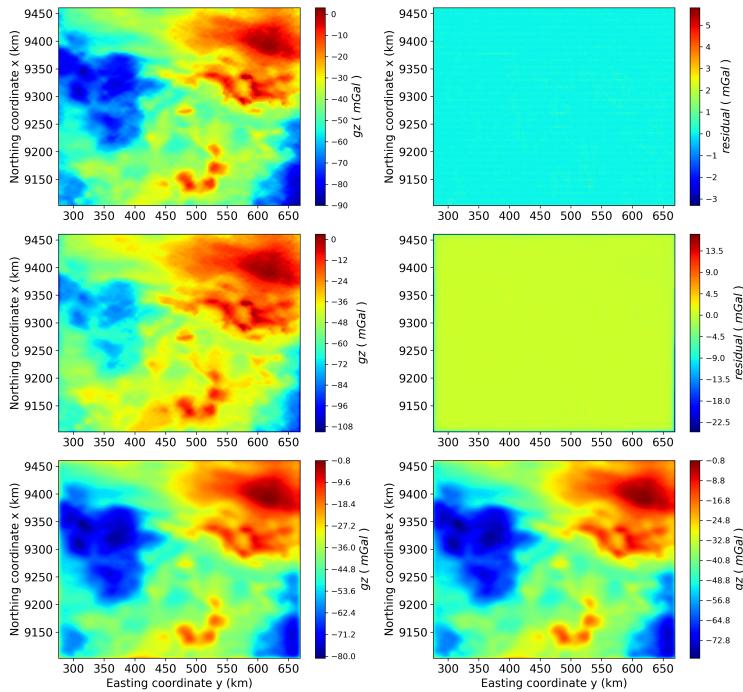


Figure 11. Panel (A) shows the Carajás predicted gravimetric data from convolutional equivalent layer method. Panel (B) shows the residual from the convolutional equivalent layer method. Panel (C) shows the predicted data from deconvolutional equivalent layer method. Panel (D) shows the residual from the deconvolutional equivalent layer method. Panel (E) shows the upward continuation at $z_i = -3500$ m for the convolutional method and Panel (F) shows the upward continuation at $z_i = -3500$ m for the deconvolutional method.

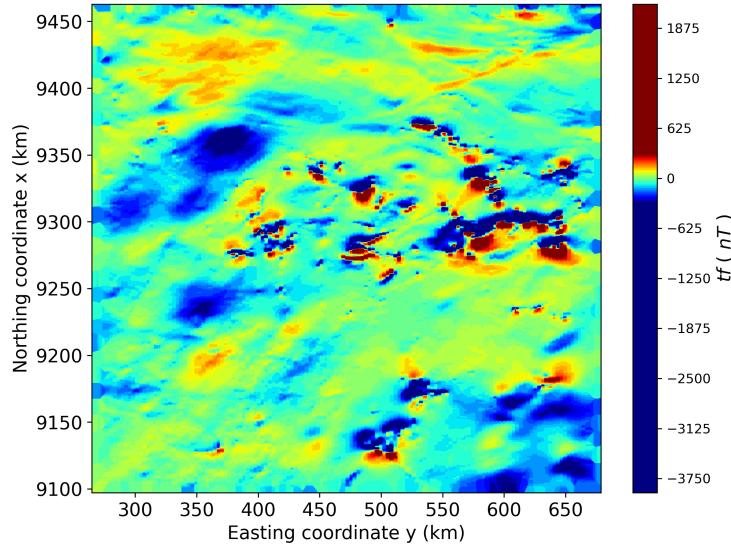


Figure 12. Gridded real aeromagnetic data from Carajás, Brazil. A regular grid of $1,000 \times 500$ is being used, totalizing $N, M = 500,000$ obsevation points and equivalent sources.

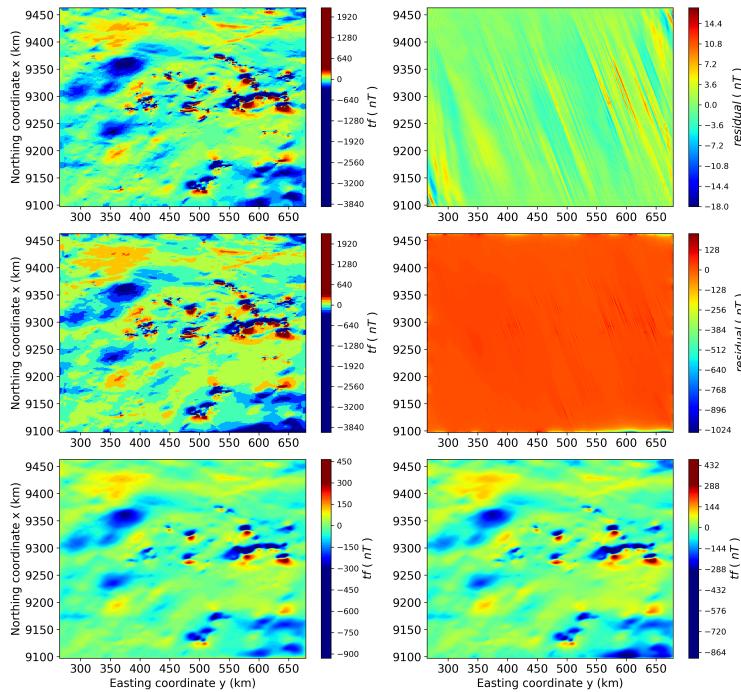


Figure 13. Panel (A) shows the Carajás predicted magnetic data from convolutional equivalent layer method. Panel (B) shows the residual from the convolutional equivalent layer method. Panel (C) shows the predicted data from deconvolutional equivalent layer method. Panel (D) shows the residual from the deconvolutional equivalent layer method. Panel (E) shows the upward continuation at $z_i = -3500$ m for the convolutional method and Panel (F) shows the upward continuation at $z_i = -3500$ m for the deconvolutional method.