Computational aspects of the equivalent-layer technique: review

Valéria C. F. Barbosa [1,*]**, Vanderlei C. Oliveira Jr.** [1]**, Diego Takahashi** [1] **and André L. A. Reis** [2]

[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, the
13 high computational cost of the equivalent-layer technique has been overcome by using a variety
14 of strategies such as: moving data-window scheme, column- and row-action updates of the
15 sensitivity matrix, reparametrization, sparsity induction of the sensitivity matrix, iterative methods
16 using the full sensitivity matrix, iterative deconvolution by using the concept of block-Toeplitz
17 Toeplitz-block (BTTB) matrices and direct deconvolution. We compute the number of floating-point
18 operations of some of these strategies adopted in the equivalent-layer technique to show their
19 effectiveness in reducing the computational demand. Numerically, we also address the stability of
20 some of these strategies used in the equivalent-layer technique by comparing with the stability
21 via the classic equivalent-layer technique with the zeroth-order Tikhonov regularization. We
22 show that even for the most computationally efficient methods, which can save up to 10^9 flops,
23 the stability of the linear system is maintained. The two most efficient strategies, iterative and
24 direct deconvolutions, can process large datasets quickly and yield good results. However, direct
25 deconvolution has some drawbacks. Real data from Carajás Mineral Province, Brazil, is also
26 used to validate the results showing a potential field transformation.

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

1 INTRODUCTION

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

42   Because the linear inverse problem to be solved in the equivalent-layer technique is set up with a full
43 sensitivity matrix, its computational cost strongly depends on the number of potential-field observations
44 and can be very inefficient for dealing with massive data sets. To overcome this problem, computationally
45 efficient methods based on equivalent-layer technique have arose in the late 1980s. To our knowledge,
46 the first method towards improving the efficiency was proposed by Leão and Silva (1989), who used
47 an overlapping moving-window scheme spanning the data set. The strategy adopted in Leão and Silva
48 (1989) involves solving several smaller, regularized linear inverse problems instead of one large problem.
49 This strategy uses a small data window and distributes equivalent sources on a small regular grid at a
50 constant depth below the data surface, with the sources' window extending beyond the boundaries of the
51 data window. Because of the spatial layouts of observed data and equivalent sources in Leão and Silva
52 (1989), the small sensitivity submatrix containing the coordinates of the data and equivalent sources within
53 a window remains constant for all data windows. This holds true regardless of the specific locations of
54 the data and equivalent sources within each window. For each position of the data window, this scheme
55 consists in computing the processed field at the center of the data window only, and the next estimates of the
56 processed field are obtained by shifting the data window across the entire dataset. More recently, Soler and
57 Uieda (2021) extended the method introduced by Leão and Silva (1989) to accommodate irregularly spaced
58 data collected on a non-flat surface. Unlike Leão and Silva (1989), in the generalization proposed by Soler
59 and Uieda (2021), the sensitivity submatrix that includes the coordinates of the data and equivalent sources
60 needs to be computed for each window. Soler and Uieda (2021) developed a computational approach to
61 further enhance the efficiency of the equivalent-layer technique by combining two strategies. The first one —
62 the block-averaging source locations — reduces the number of model parameters and the second strategy —
63 the gradient-boosted algorithm — reduces the size of the linear system to be solved by iteratively fitting the
64 equivalent source model along overlapping windows. It is worth noting that the equivalent-layer strategy of
65 using a moving-window scheme either in Leão and Silva (1989) or in Soler and Uieda (2021) is similar to
66 discrete convolution.

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

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

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

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

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

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

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

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

2 FUNDAMENTALS

132 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),
133 $i \in \{1 : D\}$, of a topocentric Cartesian system with x, y and z axes pointing to north, east and down,
134 respectively. Consider that d_i can be satisfactorily approximated by a harmonic function

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

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

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

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

$$\mathbf{f} = \mathbf{G}\mathbf{p}\,, \tag{3}$$

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

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

$$\mathbf{t} = \mathbf{A}\tilde{\mathbf{p}}\,, \tag{4}$$

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

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

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

2.1 Spatial distribution and total number of equivalent sources

154 There is no well-established criteria to define the optimum number P or the spatial distribution of the
155 equivalent sources. We know that setting an equivalent layer with more (less) sources than potential-field
156 data usually leads to an underdetermined (overdetermined) inverse problem (e.g., Menke, 2018, p. 52–53).
157 Concerning the spatial distribution of the equivalent sources, the only condition is that they must rely on a
158 surface that is located below and does not cross that containing the potential field data. Soler and Ueida
159 (2021) present a practical discussion about this topic.

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

167 Regarding the depth of the equivalent layer, Dampney (1969) proposed a criterion based on horizontal data
168 sampling, suggesting that the equivalent-layer depth should be between two and six times the horizontal grid
169 spacing, considering evenly spaced data. However, when dealing with a survey pattern that has unevenly
170 spaced data, Reis et al. (2020) adopted an alternative empirical criterion. According to their proposal,
171 the depth of the equivalent layer should range from two to three times the spacing between adjacent
172 flight lines. The criteria of Dampney (1969) and Reis et al. (2020) are valid for planar equivalent layers.
173 Cordell (1992) have proposed and an alternative criterion for scattered data that leads to an undulating
174 equivalent layer. This criterion have been slightly modified by Guspí et al. (2004), Guspí and Novara
175 (2009) and Soler and Uieda (2021), for example, and consists in setting one equivalent source below
176 each datum at a depth proportional to the horizontal distance to the nearest neighboring data points. Soler
177 and Uieda (2021) have compared different strategies for defining the equivalent sources depth for the
178 specific problem of interpolating gravity data, but they have not found significant differences between them.
179 Regarding the horizontal layout, Soler and Uieda (2021) proposed the block-averaged sources locations
180 in which the survey area is divided into horizontal blocks and one single equivalent source is assigned
181 to each block. The horizontal coordinates of the single source in a given block is defined by the average
182 horizontal coordinates of the observation points at the block. According to Soler and Uieda (2021), this
183 block-averaged layout may prevent aliasing of the interpolated values, specially when the observations
184 are unevenly sampled. This strategy also reduces the number of equivalent sources without affecting the
185 accuracy of the potential-field interpolation. Besides, it reduces the computational load for estimating the
186 physical property on the equivalent layer.

187 ## 2.2 Matrix G

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

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

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

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

191 and

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

192 In this case, the equivalent layer is formed by punctual sources representing monopoles or dipoles (e.g.,
193 Dampney, 1969; Emilia, 1973; Leão and Silva, 1989; Cordell, 1992; Oliveira Jr. et al., 2013; Siqueira et al.,
194 2017; Reis et al., 2020; Takahashi et al., 2020; Soler and Uieda, 2021; Takahashi et al., 2022). Another
195 common approach consists in not defining g_{ij} by using equations 6–8, but other harmonic functions
196 obtained by integrating them over the volume of regular prisms (e.g., Li and Oldenburg, 2010; Barnes and
197 Lumley, 2011; Li et al., 2014; Jirigalatu and Ebbing, 2019). There are also some less common approaches
198 defining the harmonic function g_{ij} (equation 2) as the potential field due to plane faces with constant
199 physical property (Hansen and Miyazaki, 1984), doublets (Silva, 1986) or by computing the double
200 integration of the inverse distance function with respect to z (Guspí and Novara, 2009).

201   A common assumption for most of the equivalent-layer methods is that the harmonic function g_{ij}
202 (equation 2) is independent on the actual physical relationship between the observed potential field and
203 their true sources (e.g., Cordell, 1992; Guspí and Novara, 2009; Li et al., 2014). Hence, g_{ij} can be
204 defined according to the problem. The only condition imposed to this function is that it decays to zero
205 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.
206 However, several methods use a function g_{ij} that preserves the physical relationship between the observed
207 potential field and their true sources. For the case in which the observed potential field is gravity data, g_{ij}
208 is commonly defined as a component of the gravitational field produced at (x_i, y_i, z_i) by a point mass or
209 prism located at (x_j, y_j, z_j), with unit density. On the other hand, g_{ij} is commonly defined as a component
210 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
211 magnetization intensity, when the observed potential field is magnetic data.

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

3   LINEAR INVERSE PROBLEM OF EQUIVALENT-LAYER TECHNIQUE

3.1   General formulation

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

$$\mathbf{p} = \mathbf{H}\,\mathbf{q}\,, \tag{9}$$

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

$$\mathbf{f} = \mathbf{G}\,\mathbf{H}\,\mathbf{q}\,. \tag{10}$$

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

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

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

231 which is a combination of particular measures of length given by

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

232 and

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

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

$$\bar{\mathbf{p}} = \mathbf{H}\,\bar{\mathbf{q}}\,, \tag{14}$$

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

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

$$\tilde{\mathbf{p}} = \mathbf{H}\,\tilde{\mathbf{q}}\,. \tag{15}$$

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

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

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

$$\tilde{\boldsymbol{\delta}}_q = \mathbf{B}\,\boldsymbol{\delta}_d\,, \tag{17}$$

243 where

$$\tilde{\mathbf{q}} = \tilde{\boldsymbol{\delta}}_q + \bar{\mathbf{q}}\,, \tag{18}$$

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

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

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

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

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

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

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

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

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

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

255 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
256 equation 18.

3.2 Formulation without reparameterization

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

$$\tilde{\boldsymbol{\delta}}_p = \tilde{\mathbf{p}} - \bar{\mathbf{p}} \,, \tag{24}$$

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

3.3 Linear system solvers

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

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

$$\mathbf{M} = \mathcal{G}\mathcal{G}^\top \,, \tag{25}$$

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

$$\begin{aligned}\mathcal{G}\,\mathbf{s} &= \mathbf{y} \\ \mathcal{G}^\top \mathbf{x} &= \mathbf{s}\end{aligned} \tag{26}$$

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

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

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

4 FLOATING-POINT OPERATIONS

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

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

$$f_{\texttt{Cholesky}} = \tfrac{1}{3}D^3 + 2D^2 + 2\left(P^2 + P\right)D \,. \tag{27}$$

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

$$f_{\texttt{CGLS}} = 2P(D + 1) + \texttt{ITMAX}\left[2P\left(2D + 3\right) + 4D\right] \,. \tag{28}$$

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

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

5   NUMERICAL STABILITY

317 All equivalent-layer methods aim at obtaining an estimate $\tilde{\mathbf{p}}$ for the parameter vector $\mathbf{p}$ (equation 3), which
318 contains the physical property of the equivalent sources. Some methods do it by first obtaining an estimate
319 $\tilde{\mathbf{q}}$ for the reparameterized parameter vector $\mathbf{q}$ (equation 9) and then using it to obtain $\tilde{\mathbf{p}}$ (equation 15).
320 The stability of a solution $\tilde{\mathbf{p}}$ against noise in the observed data is rarely addressed. Here, we follow the
321 numerical stability analysis presented in Siqueira et al. (2017).

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

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

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

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

328 and the data perturbation

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

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

6   NOTATION FOR SUBVECTORS AND SUBMATRICES

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

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

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

,

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

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

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

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

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

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

7 COMPUTATIONAL STRATEGIES

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

339   (i) the large computer memory to store large and full matrices;

340   (ii) the long computation time to multiply a matrix by a vector; and

341   (iii) the long computation time to solve a large linear system of equations.

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

7.1 Moving window

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

350 m-th window are usually adjacent to each other and have indices defined by an integer list $\mathbf{i}^m$ having

351 D^m elements. The number of data D^m forming the data windows are not necessarily equal to each other.

352 Each data window has a $D^m \times 1$ observed data vector $\mathbf{d}^m \equiv \mathbf{d}[\mathbf{i}^m]$. The second step consists in defining

353 a set of P equivalent sources with scalar physical property p_j, $j \in \{1 : P\}$, and also split them into M

354 overlapping subsets (or source windows) formed by P^m data each, $m \in \{1 : M\}$. The sources inside the

355 m-th window have indices defined by an integer list $\mathbf{j}^m$ having P^m elements. Each source window has a

356 $P^m \times 1$ parameter vector $\mathbf{p}^m$ and is located right below the corresponding m-th data window. Then, each

357 $\mathbf{d}^m \equiv \mathbf{d}[\mathbf{i}^m]$ is approximated by

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

358 where $\mathbf{G}^m \equiv \mathbf{G}[\mathbf{i}^m, \mathbf{j}^m]$ is a submatrix of $\mathbf{G}$ (equation 3) formed by the elements computed with equation

359 2 using only the data and equivalent sources located inside the window m-th. The main idea of the moving-

360 window approach is using the $\tilde{\mathbf{p}}^m$ estimated for each window to obtain (i) an estimate $\tilde{\mathbf{p}}$ of the parameter

361 vector for the entire equivalent layer or (ii) a given potential-field transformation $\mathbf{t}$ (equation 4). The main

362 advantages of this approach is that (i) the estimated parameter vector $\tilde{\mathbf{p}}$ or transformed potential field are

363 not obtained by solving the full, but smaller linear systems and (ii) the full matrix $\mathbf{G}$ (equation 3) is never

364 stored.

365   Leão and Silva (1989) presented a pioneer work using the moving-window approach. Their method

366 requires a regularly-spaced grid of observed data on a horizontal plane z_0. The data windows are defined by

367 square local grids of $\sqrt{D'} \times \sqrt{D'}$ adjacent points, all of them having the same number of points D'. The

368 equivalent sources in the m-th data window are located below the observation plane, at a constant vertical

369 distance Δz_0. They are arranged on a regular grid of $\sqrt{P'} \times \sqrt{P'}$ adjacent points following the same

370 grid pattern of the observed data. The local grid of sources for all data windows have the same number

371 of elements P'. Besides, they are vertically aligned, but expands the limits of their corresponding data

372 windows, so that $D' < P'$. Because of this spatial configuration of observed data and equivalent sources,

373 we have that $\mathbf{G}^m = \mathbf{G}'$ (equation 32) for all data windows (i.e., $\forall\, m \in \{1 : M\}$), where $\mathbf{G}'$ is a $D' \times P'$

374 constant matrix.

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

376 computing the transformed potential field t_c^m at the central point $(x_c^m, y_c^m, z_0 + \Delta z_0)$ of each data window

377 as follows:

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

378 where $\mathbf{a}'$ is a $P' \times 1$ vector with elements computed by equation 5 by using all equivalent sources in the

379 m-th window and only the coordinate of the central point in the m-th data window and

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

380 is a particular case of matrix $\mathbf{B}$ associated with underdetermined problems (equation 21) for the particular

381 case in which $\mathbf{H} = \mathbf{W}_q = \mathbf{I}_{P'}$ (equations 9 and 13), $\mathbf{W}_d = \mathbf{I}_{D'}$ (equation 12), $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14), where

382 $\mathbf{I}_{P'}$ and $\mathbf{I}_{D'}$ are identity matrices of order P' and D', respectively, and $\mathbf{0}$ is a vector of zeros. Due to the

383 presumed spatial configuration of the observed data and equivalent sources, $\mathbf{a}'$ and $\mathbf{G}'$ are the same for all

384 data windows. Hence, only the data vector $\mathbf{d}^m$ is modified according to the position of the data window.

385 Note that equation 33 combines the potential-field transformation (equation 4) with the solution of the

386 undetermined problem (equation 23).

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

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

$$f_{\mathsf{LS89}} = {}^7\!/6(D')^3 + 4P'(D')^2 + M\,2P' \,. \tag{35}$$

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

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

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

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

425 by:

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

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

7.2 Column-action update

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

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

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

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

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

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

$$f_{\texttt{C92}} = D \log(D) + \texttt{ITMAX} \left[2D + D \log_2(D) \right] . \tag{38}$$

7.3 Row-action update

469 We call the computational strategy *row-action update* because a single row of the sensitivity matrix
470 $\mathbf{G}$ (equation 3) is calculated iteratively. Hence, the equivalent-layer solution is updated by processing a
471 new datum (one matrix row) at each iteration. To reduce the total processing time and memory usage of
472 equivalent-layer technique, Mendonça and Silva (1994) proposed a strategy called *equivalent data concept*.
473 The equivalent data concept is grounded on the principle that there is a subset of redundant data that does
474 not contribute to the final solution and thus can be dispensed. Conversely, there is a subset of observations,
475 called equivalent data, that contributes effectively to the final solution and fits the remaining observations
476 (redundant data). Iteratively, Mendonça and Silva (1994) selected the subset of equivalent data that is
477 substantially smaller than the original dataset. This selection is carried out by incorporating one data point
478 at a time.

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

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

490 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,
491 respectively. Mendonça and Silva (1994) designate $\mathbf{d}_e$ and $\mathbf{d}_r$ as, respectively, *equivalent* and *redundant*
492 data. With the exception of a normalization strategy, Mendonça and Silva (1994) calculate a $P \times 1$ estimated
493 parameter vector $\tilde{\mathbf{p}}$ by solving an underdetermined problem (equation 23) involving only the equivalent
494 data $\mathbf{d}_e$ (equation 39) for the particular case in which $\mathbf{H} = \mathbf{W}_p = \mathbf{I}_P$ (equations 9 and 13), $\mathbf{W}_d = \mathbf{I}_{D_e}$
495 (equation 12) and $\bar{p} = \mathbf{0}$ (equation 14), which results in

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

496 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
497 (1994) presume that the estimated parameter vector $\tilde{\mathbf{p}}$ obtained from equation 40 leads to a $D_r \times 1$ residuals
498 vector

$$\mathbf{r} = \mathbf{d}_r - \mathbf{G}_r \tilde{\mathbf{p}} \tag{41}$$

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

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

7.4 Reparameterization

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

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

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

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

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

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

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

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

$$f_{\texttt{reparam.}} = 2Q(DP + D + 1) + 2PQ + \texttt{ITMAX}\left[2Q\left(2D + 3\right) + 4D\right] . \tag{43}$$

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

7.5 Sparsity induction

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

$$\mathbf{d}_w \approx \mathbf{G}_s\,\mathbf{p}_w , \tag{44}$$

where

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

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

$$\mathbf{G}_w = \mathcal{W}\,\mathbf{G}\,\mathcal{W}^\top \tag{46}$$

with absolute value smaller than a given threshold.

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

$$\mathbf{G}_L = \mathbf{G}_s\,\mathbf{L}^{-1} \tag{47}$$

and a normalized parameter vector

$$\mathbf{p}_L = \mathbf{L}\,\mathbf{p}_w , \tag{48}$$

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

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

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

572   Barnes and Lumley (2011) also proposed a computationally efficient method for equivalent-layer
573   technique by inducing sparsity into the original sensitivity matrix $\mathbf{G}$ (equation 3). Their approach consists
574   in setting a $P \times Q$ reparameterization matrix $\mathbf{H}$ (equation 9) with $Q \approx 1.7\,P$. Note that, differently from
575   Oliveira Jr. et al. (2013) and Mendonça (2020), Barnes and Lumley (2011) do not use the reparameterization
576   with the purpose of reducing the number of the parameters. Instead, they use a reparameterization scheme
577   that groups distant equivalent sources into blocks by using a bisection process. This scheme leads to
578   a quadtree representation of the physical-property distribution on the equivalent layer, so that matrix
579   $\mathbf{G}\mathbf{H}$ (equation 10) is notably sparse. Barnes and Lumley (2011) explore this sparsity in solving the
580   overdetermined problem for $\tilde{\mathbf{q}}$ (equation 42) via conjugate-gradient method (e.g., Golub and Van Loan,
581   2013, sec. 11.3).

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

7.6   Iterative methods using the full matrix G

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

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

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

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

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

$$f_{\texttt{SOB17}} = 2D^2 + 2D + \texttt{ITMAX} \left(2D^2 + 3D\right) . \tag{50}$$

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

7.7 Iterative deconvolution

625 Recently, Takahashi et al. (2020, 2022) proposed the *convolutional equivalent-layer method*, which
626 explores the structure of the sensitivity matrix $\mathbf{G}$ (equation 3) for the particular case in which (i) there
627 is a single equivalent source right below each potential-field datum and (ii) both data and sources rely
628 on planar and regularly spaced grids. Specifically, they consider a regular grid of D potential-field data
629 at points (x_i, y_i, z_0), $i \in \{1 : D\}$, on a horizontal plane z_0. The data indices i may be ordered along the
630 $x-$ or $y-$direction, which results in an $x-$ or $y-$oriented grid, respectively. They also consider a single
631 equivalent source located right below each datum, at a constant vertical coordinate $z_0 + \Delta z$, $\Delta z > 0$. In
632 this case, the number of data and equivalent sources are equal to each other (i.e., $D = P$) and $\mathbf{G}$ (equation
633 3) assumes a *doubly block Toeplitz* (Jain, 1989, p. 28) or *block-Toeplitz Toeplitz-block* (BTTB) (Chan and
634 Jin, 2007, p. 67) structure formed by $N_B \times N_B$ blocks, where each block has $N_b \times N_b$ elements, with
635 $D = N_B \, N_b$. This particular structure allows formulating the product of $\mathbf{G}$ and an arbitrary vector as a *fast*
636 *discrete convolution* via *Fast Fourier Transform* (FFT) (Van Loan, 1992, section 4.2).

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

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

639 where each block $\mathbf{G}^n$, $n \in \{(1 - N_B) : (N_B - 1)\}$, is a 3×3 Toeplitz matrix. Takahashi et al. (2020,
640 2022) have deduced the specific relationship between blocks $\mathbf{G}^n$ and $\mathbf{G}^{-n}$ and also between a given block
641 $\mathbf{G}^n$ and its transposed $(\mathbf{G}^n)^\top$ according to the harmonic function g_{ij} (equation 2) defining the element ij
642 of the sensitivity matrix $\mathbf{G}$ (equation 3) and the orientation of the data grid.

643 Consider the matrix-vector products

$$\mathbf{G} \, \mathbf{v} = \mathbf{w} \tag{52}$$

644 and

$$\mathbf{G}^\top \mathbf{v} = \mathbf{w} , \tag{53}$$

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

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

647 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
648 having N_b elements. Equations 52 and 53 can be computed in terms of an auxiliary matrix-vector product

$$
\mathbf{G}_c \, \mathbf{v}_c = \mathbf{w}_c \, , \tag{55}
$$

649 where

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

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

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

651 and $\mathbf{G}_c$ is a $4D \times 4D$ *doubly block circulant* (Jain, 1989, p. 28) or *block-circulant circulant-block* (BCCB)
652 (Chan and Jin, 2007, p. 76) matrix. What follows aims at explaining how the original matrix-vector products
653 defined by equations 52 and 53, involving a $D \times D$ BTTB matrix $\mathbf{G}$ exemplified by equation 51, can be
654 efficiently computed in terms of the auxiliary matrix-vector product given by equation 55, which has a
655 $4D \times 4D$ BCCB matrix $\mathbf{G}_c$.

656   Matrix $\mathbf{G}_c$ (equation 55) is formed by $2N_B \times 2N_B$ blocks, where each block $\mathbf{G}_c^n$, $n \in \{(1 - N_B) :$
657 $(N_B - 1)\}$ is a $2N_b \times 2N_b$ circulant matrix. For the case in which the original matrix-vector product is that
658 defined by equation 52, the first column of blocks forming the BCCB matrix $\mathbf{G}_c$ is given by

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

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

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

660 where $\mathbf{G}^n$ are the blocks forming the BTTB matrix $\mathbf{G}$ (equation 51). For the case in which the original
661 matrix-vector product is that defined by equation 53, the first column of blocks forming the BCCB matrix
662 $\mathbf{G}_c$ is given by

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

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

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

664 The complete matrix $\mathbf{G}_c$ (equation 55) is obtained by properly downshifting the block columns $\mathbf{G}_c[:,:$
665 $2N_b]$ defined by equation 58 or 60. Similarly, the n-th block $\mathbf{G}_c^n$ of $\mathbf{G}_c$ is obtained by properly downshifting
666 the first columns $\mathbf{G}_c^\ell[:,1]$ defined by equation 59 or 61.

667   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
668 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
669 to use the fact that BCCB matrices are diagonalized by the 2D unitary discrete Fourier transform (DFT)
670 (e.g., Davis, 1979, p. 31). Because of that, $\mathbf{G}_c$ can be written as

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

671 where the symbol "$\otimes$" denotes the Kronecker product (e.g., Horn and Johnson, 1991, p. 243), $\boldsymbol{\mathcal{F}}_{2N_B}$ and
672 $\boldsymbol{\mathcal{F}}_{2N_b}$ are the $2N_B \times 2N_B$ and $2N_b \times 2N_b$ unitary DFT matrices (e.g., Davis, 1979, p. 31), respectively,
673 the superscritpt "$*$" denotes the complex conjugate and $\boldsymbol{\Lambda}$ is a $4D \times 4D$ diagonal matrix containing the
674 eigenvalues of $\mathbf{G}_c$. Due to the diagonalization of the matrix $\mathbf{G}_c$, equation 55 can be rewritten by using
675 equation 62 and premultiplying both sides of the result by $\left(\boldsymbol{\mathcal{F}}_{2N_B} \otimes \boldsymbol{\mathcal{F}}_{2N_b}\right)$, i.e.,

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

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

$$
\boldsymbol{\mathcal{L}} \circ \left(\boldsymbol{\mathcal{F}}_{2N_B} \boldsymbol{\mathcal{V}}_c \boldsymbol{\mathcal{F}}_{2N_b}\right) = \boldsymbol{\mathcal{F}}_{2N_B} \boldsymbol{\mathcal{W}}_c \boldsymbol{\mathcal{F}}_{2N_b} \tag{64}
$$

677 where "$\circ$" denotes the Hadamard product (e.g., Horn and Johnson, 1991, p. 298) and $\boldsymbol{\mathcal{L}}$, $\boldsymbol{\mathcal{V}}_c$ and $\boldsymbol{\mathcal{W}}_c$ are
678 $2N_B \times 2N_b$ matrices obtained by rearranging, along their rows, the elements forming the diagonal of $\boldsymbol{\Lambda}$
679 (equation 62), vector $\mathbf{v}_c$ and vector $\mathbf{w}_c$ (equation 56), respectively. Then, by premultiplying both sides of
680 equation 64 by $\boldsymbol{\mathcal{F}}_{2N_B}^*$ and then postmultiplying both sides by $\boldsymbol{\mathcal{F}}_{2N_b}^*$, we obtain

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

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

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

683 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
684 55). It is important noting that the matrices $\mathcal{C}$ and $\mathcal{L}$ (equation 66) associated with the BTTB matrix $\mathbf{G}$
685 (equation 51) are different from those associated with $\mathbf{G}^\top$.

686   The whole procedure to compute the original matrix-vector products $\mathbf{G}\mathbf{v}$ (equation 52) and $\mathbf{G}^\top\mathbf{v}$
687 (equation 53) consists in (i) rearranging the elements of the vector $\mathbf{v}$ and the first column $\mathbf{G}[:, 1]$ of matrix
688 $\mathbf{G}$ 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
689 $\mathcal{F}^*_{2N_B}\, \mathcal{A}\, \mathcal{F}^*_{2N_b}$, where $\mathcal{A}$ is a given matrix, and a Hadamard product to obtain $\mathcal{W}_c$ (equation 65); and (iii)
690 retrieve the elements of vector $\mathbf{w}$ (equation 52) from $\mathcal{W}_c$ (equation 65). It is important noting that the steps
691 (i) and (iii) do not have any computational cost because they involve only reorganizing elements of vectors
692 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
693 2D Discrete Fourier Transform (2D-DFT) and the 2D Inverse Discrete Fourier Transform (2D-IDFT) of $\mathcal{A}$.
694 These transforms can be efficiently computed by using the 2D Fast Fourier Transform (2D-FFT). Hence,
695 the original matrix-vector products $\mathbf{G}\mathbf{v}$ (equation 52) and $\mathbf{G}^\top\mathbf{v}$ (equation 53) can be efficiently computed
696 by using the 2D-FFT.

697   Algorithms 7 and 8 show pseudo-codes for the convolutional equivalent-layer method proposed by
698 Takahashi et al. (2020, 2022). Note that those authors formulate the overdetermined problem (equation
699 22) of obtaining an estimate $\tilde{\mathbf{p}}$ for the parameter vector $\mathbf{p}$ (equation 3) as an *iterative deconvolution* via
700 *conjugate gradient normal equation residual* (CGNR) Golub and Van Loan (2013, sec. 11.3) or *conjugate*
701 *gradient least squares* (CGLS) (Aster et al., 2019, p. 165) method. They consider $\mathbf{H} = \mathbf{I}_P$ (equation 9),
702 $\mu = 0$ (equation 11), $\mathbf{W}_d = \mathbf{W}_q = \mathbf{I}_P$ (equations 12 and 13) and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14). As shown by
703 Takahashi et al. (2020, 2022), the CGLS produces stable estimates $\tilde{\mathbf{p}}$ for the parameter vector $\mathbf{p}$ (equation
704 3) in the presence of noisy potential-field data $\mathbf{d}$. This is a well-known property of the CGLS method (e.g.,
705 Aster et al., 2019, p. 166).

706   The key aspect of Algorithm 7 is replacing the matrix-vector products of CGLS (Algorithm 1) by fast
707 convolutions (Algorithm 8). A fast convolution requires one 2D-DFT, one 2D-IDFT and an entrywise
708 product of matrices. We consider that the 2D-DFT/IDFT are computed with 2D-FFT and requires
709 $\lambda\,(4D)\log_2(4D)$ flops, where $\lambda = 5$ is compatible with a radix-2 FFT (Van Loan, 1992, p. 16), and
710 the entrywise product $24D$ flops because it involves two complex matrices having $4D$ elements (Golub
711 and Van Loan, 2013, p. 36). Hence, Algorithm 8 requires $\lambda\,(16D)\log_2(4D) + 26D$ flops, whereas a
712 conventional matrix-vector multiplication involving a $D \times D$ matrix requires $2D^2$ (table 1). Finally,
713 Algorithm 7 requires two 2D-FFTs (lines 4 and 5), one fast convolution and an inner product (line 8)
714 previously to the while loop. Per iteration, there are three saxpys (lines 12, 15 and 16), two inner products
715 (lines 14 and 17) and two fast convolutions (lines 13 and 17), so that:

$$f_{\texttt{TOB20}} = \lambda\,(16D)\log_2(4D) + 26D + \texttt{ITMAX}\left[\lambda\,(16D)\log_2(4D) + 58D\right]\, . \tag{67}$$

7.8 Direct deconvolution

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

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

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

721 where

$$\breve{\boldsymbol{\mathcal{L}}} = \boldsymbol{\mathcal{L}}^* \oslash \left(\boldsymbol{\mathcal{L}} \circ \boldsymbol{\mathcal{L}}^* + \zeta \mathbf{1} \right) \,, \tag{69}$$

722 $\mathbf{1}$ is a $4D \times 4D$ matrix of ones, "$\oslash$" denotes entrywise division and ζ is a positive scalar. Note that $\zeta = 0$
723 leads to $\mathbf{1} \oslash \boldsymbol{\mathcal{L}}$. In this case, the entrywise division may be problematic due to the elements of $\boldsymbol{\mathcal{L}}$ having
724 absolute value equal or close to zero. So, a small ζ is set to avoid this problem in equation 69. Next, we use
725 $\breve{\boldsymbol{\mathcal{L}}}$ to obtain a matrix $\boldsymbol{\mathcal{V}}_c$ from equation 68. Finally, the elements of the estimated parameter vector $\tilde{\mathbf{p}}$ are
726 retrieved from the first quadrant of $\boldsymbol{\mathcal{V}}_c$. This procedure represents a *direct deconvolution* (e.g., Aster et al.,
727 2019, p. 220) using a *Wiener filter* (e.g., Gonzalez and Woods, 2002, p. 263).

728 The required total number of flops associated with the direct deconvolution aggregates one 2D-FFT
729 to compute matrix $\boldsymbol{\mathcal{L}}$ (equation 66), one entrywise product $\boldsymbol{\mathcal{L}} \circ \boldsymbol{\mathcal{L}}^*$ involving complex matrices and one
730 entrywise division to compute $\breve{\boldsymbol{\mathcal{L}}}$ (equation 69) and a fast convolution (Algorithm 8) to evaluate equation
731 68, which results in:

$$f_{\texttt{deconv.}} = \lambda \left(12D \right) \log_2(4D) + 72D \,. \tag{70}$$

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

8 NUMERICAL SIMULATIONS

8.1 Flops count

741

742   Figure 1 shows the total number of flops for solving the overdetermined problem (equation 22) with
743 different equivalent-layer methods (equations 27, 28, 35, 37, 38, 43, 50, 67, and 70), by considering
744 the particular case in which $\mathbf{H} = \mathbf{I}_P$ (equation 9 and subsection 3.2), $\mu = 0$ (equation 11), $\mathbf{W}_d = \mathbf{I}_D$
745 (equation 12) and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14), where $\mathbf{I}_P$ and $\mathbf{I}_D$ are the identities of order P and D, respectively.
746 The flops are computed for different number of potential-field data ranging from $10,000$ to $1,000,000$.
747 Figure 1 shows that the moving data-window strategy by using Leão and Silva's 1989 method and direct
748 deconvolution are the fastest methods.

749   The control parameters to run the equivalent-layer methods shown in Figure 1 are the following: i) in
750 CGLS, reparameterization approaches (e.g., Oliveira Jr. et al., 2013; Mendonça, 2020), Siqueira et al.
751 (2017), and Takahashi et al. (2020) (equations 28 38), 43, 50 and 67) we set ITMAX $= 50$; ii) Cordell
752 (1992) we set ITMAX $= 4\,D$; iii) in Leão and Silva (1989) (equation 35) we set $D' = 49\ (7 \times 7)$ and
753 $P' = 225\ (15 \times 15)$; and iv) in Soler and Uieda (2021) (equation 37) we set $D' = P' = 900\ (30 \times 30)$.

8.2 Synthetic potential-field data

754

755   We create a model composed of three synthetic bodies: a sphere centered at $(x, y, z) = (3, -2, 2)$ km,
756 with radius of 1 km; a sphere centered at $(x, y, z) = (1, 2.5, 1.8)$ km, with radius of 750 m; and a right prism
757 having polygonal horizontal cross-section, top at $z = 900$ m and thickness 600 m. The density contrasts
758 of the upper left sphere, upper right sphere and prism are, respectively, $600\,\text{kg/m}^3$, $-500\,\text{kg/m}^3$ and to
759 $550\,\text{kg/m}^3$. All synthetic bodies have a total-magnetization vector with intensity $3.46\,\text{A/m}$, inclination
760 $35.26°$ and declination $45.0°$. We consider a main geomagnetic field with constant inclination $20.0°$ and
761 declination $35.0°$.

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

8.3 Stability analysis and upward-continued data

771

772   We set two planar equivalent layers having one source below each datum at a constant vertical coordinate
773 $z = 300$ m. Note that, in this case, both layers have a number of sources P equal to the number of data D.
774 One layer is formed by point masses and is applied to the synthetic gravity data. The other is applied to the
775 synthetic magnetic data and is composed of dipoles.

776   We have applied the Cholesky factorization (equations 25 and 26), the iterative deconvolution (Algorithms
777 7 and 8) proposed by Takahashi et al. (2020, 2022) and the direct deconvolution (equations 68 and 69) with
778 four different values for the parameter ζ to the 21 gravity and 21 magnetic data sets.

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

786 All these 42 estimated parameters vectors (21 for gravity and 21 for magnetic data) were obtained by
787 solving the overdetermined problem (equation 22) with the same method for the particular case in which
788 $\mathbf{H} = \mathbf{I}$ (equation 9 and subsection 3.2), $\mathbf{W}_d = \mathbf{W}_q = \mathbf{I}$ (equations 12 and 13) and $\bar{\mathbf{p}} = \mathbf{0}$ (equation 14),
789 where $\mathbf{I}$ is the identity of order D.

790 Figure **??** shows how the numerical stability curves vary as the level of the noise is increased for
791 the gravimetric data. We can see that for all methods, a linear tendency is observed as it is expected.
792 The inclination of the straight line indicates the stability of each method. As shown in Figure **??**, the
793 deconvolutional method without stabilization parameter (blue dots with $\zeta = 0$) exhibits high instability,
794 emphasizing the necessity of using the Wiener filter ($\zeta > 0$ in equation 69) to achieve accurate parameter
795 estimation. Using these gravimetric data, an optimal stabilization parameter $\zeta = 10^{-22}$ (green dots) is
796 necessary to yield a stable solution and prevent overshooting the smootheness of the solution. For the
797 magnetic data, Figure **??** shows a very similar behavior of the stability as the previous case. In this case,
798 the Wiener parameter seems to have the best solution for $\zeta = 10^{-14}$ (green dots).

799 For both potential-field data, the estimated slopes (Figures **??** and **??**) obtained by using direct
800 deconvolution with an optimum ζ (green dots), Cholesky factorization (black dots) and iterative
801 deconvolution (red dots) proposed by Takahashi et al. (2020) are close to each other. This result suggest
802 that the direct deconvolution with an optimum Wiener parameter seems to be one that produces a low slope
803 for the straight line in the stability analysis.

804 Figures **??** and **??** present the comparison between the residuals due the equivalent layers obtained
805 with the Cholesky factorization, iterative deconvolution and direct deconvolution with optimum ζ using,
806 respectively, gravity and magnetic data (Figures **??** and **??**). The inverted data are those corrupted with the
807 highest noise level (Figures **??**B and **??**B).

808 Using the gravimetric data, the Cholesky factorization (Figure **??**A), the iterative deconvolution (Figure
809 **??**B) proposed by Takahashi et al. (2020), and the direct deconvolution with an optimum $\zeta = 10^{-22}$
810 (Figure **??**E) yield acceptable data fitting despite their significant difference in floating-point operations. In
811 contrast, the residuals obtained using direct deconvolutional method with a Wiener stabilization ζ equal to
812 zero (Figure **??**C), a high value of $\zeta = 10^{-20}$ (Figure **??**D), and a low value of $\zeta = 10^{-24}$ (Figure **??**F)
813 result in unacceptable data fitting.

814 Using the magnetic data, Figure **??** shows a very similar behavior as in the previous case, indicating a
815 similar pattern. Hence, the residuals obtained using the Cholesky factorization (Figure **??**A), the iterative
816 deconvolution (Figure **??**B) proposed by Takahashi et al. (2020), and the direct deconvolution with
817 an optimum $\zeta = 10^{-14}$ (Figure **??**E) show acceptable data fittings. The remainder residuals (direct
818 deconvolutional method with $\zeta = 0$ in Figure **??**C, $\zeta = 10^{-12}$ in Figure **??**D, and $\zeta = 10^{-16}$ in Figure
819 **??**F) show that the estimated equivalent layers do not fit the noisy magnetic data.

820 The upward continuation of the potential-field data is a processing technique to predict the data in a
821 higher altitude. In practice, the interpreter expects a lower amplitude signal and smoother data as the high

822 frequency anomalies tends to disapear. Figure **??**A shows the true modeled upward gravity data at a height
823 of -500 m. Figures **??**B–E show the results of the upward processing obtained by using the Cholesky
824 factorization, the iterative deconvolution proposed by Takahashi et al. (2020) and the direct deconvolution
825 with $\zeta = 0$ and $\zeta = 10^{-22}$ (optimum value), respectively. Except the direct deconvolutional method
826 without stabilization ($\zeta = 0$, Figure **??**D), all the three most stable equivalent-layer methods predict the
827 upward-continued gravity data very reasonable.

828   Figure **??** shows the comparison between the true (Figure **??**A) and the predicted (Figures **??**B–E)
829 upward-continued magnetic data at a height of -1400 m. As in the gravimetric case, the predicted upward-
830 continued magnetic data via Cholesky factorization (Figure **??**C), the iterative deconvolution proposed
831 by Takahashi et al. (2020) (Figure **??**D) and the direct deconvolution with optimum value of $\zeta = 10^{-14}$
832 (Figure **??**E) are acceptable, except the direct deconvolutional method without stabilization ($\zeta = 0$, Figure
833 Figure **??**D).

9 APPLICATIONS TO FIELD DATA

834 In this section, we show the applications of the iterative (Algorithm 7) and direct (equations 68 and 69)
835 deconvolutions to field data sets over the Carajás Mineral Province (CMP) in the Amazon craton (Moroni
836 et al., 2001; Villas and Santos, 2001). This area is known for its intensive mineral exploration such as iron,
837 copper, gold, manganese, and, recently, bauxite.

9.1 Geological setting

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

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

9.2 Potential-field data

864 The field data used here were acquired obtained from an airborne survey conducted by Lasa Prospecções
865 S/A. and Microsurvey Aerogeofísica Consultoria Científica Ltda. between April/2013 and October/2014.
866 The aeromagnetic data were acquired by the Geological Survey of Brazil-CPRM (ACHO QUE A CPRM
867 NÃO ADQUIRI DADOS). The survey area covers $\approx 58000 \, \mathrm{km}^2$ with high-resolution gravity and magnetic
868 data. The flight and the tie lines were acquired and spaced at $3 \, \mathrm{km}$ and $12 \, \mathrm{km}$ oriented in the directions
869 $N - S$ and $E - W$, respectively, with a mean flight height of $900 \, \mathrm{m}$ above the ground. For both applications,
870 we have interpolated gravity and magnetic anomalies data, calculating the data set in a grid of 1000×500
871 ($D = 500,000$ observation points) at the same mean flight height. The interpolated aerogravimetric and
872 aeromagnetic data are shwon, respectively, in Figures **??** and **??**.

9.3 Applications

In all applications, we set the same equivalent sources layout with one source located below each datum (so $P = D$) on a horizontal plane distant 1200 m from the observations. This setup is composed by a grid of 1000×500 equivalent sources ($M = 500,000$).

For performing the upward continuation of the gravimetric data (Figure **??**) over CMP in the Amazon craton, we apply the two equivalent-layer methods (iterative and direct deconvolutions). By applying the iterative and direct deconvolutions, we estimated density distributions (not shown) that produced the predicted gravimetric data shown, respectively, in Figures **??**A and C. The gravimetric residuals obtained by both the iterative (Figure **??**B) and direct (Figure **??**D) deconvolutions are close to zero for most of the survey area indicating that both methods achieve acceptable data fittings. For the iterative deconvolution, the residuals have a mean of ≈ 0.0005 mGal and a standard deviation of ≈ 0.16 mGal. For the direct deconvolution, the mean and standard deviation of the residuals are ≈ 0.46 mGal and ≈ 1.23 mGal, respectively. We stress that the residuals from the direct deconvolution (Figure **??**D) are greater than the iterative deconvolution (Figure **??**B). A possible reason for this difference may be the border effects. Figures **??**E and F show the upward-continued gravimetric data at a height of -3500 m by using the iterative and direct deconvolutions, respectively. Considering the processing time, the iterative and direct deconvolutions took ≈ 9.18 s and ≈ 0.53 s, respectively. These results demonstrate the efficiency of both the iterative and direct deconvolution methods in processing large gravimetric datasets from extensive areas with dense data coverage. It is worth noting that the direct deconvolution method exhibits faster processing times compared to the iterative deconvolution, which aligns with expectations

We also apply the iterative and direct deconvolutions to compute the upward continuation of the total-field anomalies (Figure **??**) over CMP in the Amazon craton, Brazil. Due to the extensive survey area, the main magnetic field direction varies significantly across the area and thus we consider a mean direction for the main magnetic field with declination $-19.865°$ and inclination $-7.43915°$. We set a magnetization direction for the equivalent sources equal to the main magnetic field direction. Besides, we set the equivalent layer at constant depth 900 m. After estimating the density distributions (not shown), we obtain the predicted data (Figure **??**A) and residuals (Figure **??**B) using the iterative deconvolution with a mean residual of $\approx 0.06\,nT$ and a standard deviation of ≈ 1.97 nT. Similarly, we generate the predicted data (Figure **??**C) and residuals (Figure **??**D) using the direct deconvolution method with a mean residual of $\approx 18.99\,nT$ and a standard deviation of ≈ 33.64 nT. The observed total-field anomaly upward-continued at a height of -3500 m by using the iterative and direct deconvolutions are shown in Figures **??**E and F, respectively. Considering the processing time to upward-continued the magnetic data, the iterative and direct deconvolutions took, respectively, ≈ 82.08 s and ≈ 0.89 s, highlighting the faster processing time of the direct deconvolution method compared to the iterative deconvolution. Despite the difference in processing time between the two deconvolutions and taking into account the mean and standard deviations of the residuals, our conclusion is that the iterative deconvolution provides a better fit to the total-field anomalies compared to the direct deconvolution.

10   DISCUSSION AND CONCLUSION

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

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

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

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

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

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

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

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

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

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

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

CONFLICT OF INTEREST STATEMENT

1007 The authors declare that the research was conducted in the absence of any commercial or financial
1008 relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

1009 The Author Contributions section is mandatory for all articles, including articles by sole authors. If an
1010 appropriate statement is not provided on submission, a standard one will be inserted during the production
1011 process. The Author Contributions statement must describe the contributions of individual authors referred
1012 to by their initials and, in doing so, all authors agree to be accountable for the content of the work. Please
1013 see here for full authorship criteria.

FUNDING

ACKNOWLEDGMENTS

DATA AVAILABILITY STATEMENT

1020 The datasets generated for this study can be found in the frontiers-paper Github repository link:
1021 https://github.com/XXXXXX/eqlayer-review-computational.

REFERENCES

1022 Amaral, G. (1974). *Geologia Pré-Cambriana da Região Amazônica*. Ph.D. thesis, Universidade de ãão
1023   Paulo
1024 Aster, R. C., Borchers, B., and Thurber, C. H. (2019). *Parameter Estimation and Inverse Problems*
1025   (Elsevier), 3 edn.
1026 Barbosa, V. C. F., Silva, J. B., and Medeiros, W. E. (1997). Gravity inversion of basement relief using
1027   approximate equality constraints on depths. *Geophysics* 62, 1745–1757
1028 Barnes, G. and Lumley, J. (2011). Processing gravity gradient data. *GEOPHYSICS* 76, I33–I47. doi:10.
1029   1190/1.3548548
1030 Blakely, R. J. (1996). *Potential Theory in Gravity and Magnetic Applications* (Cambridge University
1031   press)
1032 Bott, M. H. P. (1960). The use of Rapid Digital Computing Methods for Direct Gravity Interpretation
1033   of Sedimentary Basins. *Geophysical Journal International* 3, 63–67. doi:10.1111/j.1365-246X.1960.
1034   tb00065.x
1035 Chan, R. H.-F. and Jin, X.-Q. (2007). *An introduction to iterative Toeplitz solvers*, vol. 5 (SIAM)

Cordell, L. (1992). A scattered equivalent-source method for interpolation and gridding of potential-field data in three dimensions. *Geophysics* 57, 629–636

Cunha, I. R., Dall'Agnol, R., and Feio, G. R. L. (2016). Mineral chemistry and magnetic petrology of the archean Planalto Suite, Carajás Province – Amazonian Craton: Implications for the evolution of ferroan archean granites. *Journal of South American Earth Sciences* 67, 100–121

Dampney, C. N. G. (1969). The equivalent source technique. *GEOPHYSICS* 34, 39–53. doi:10.1190/1.1439996

Davis, P. J. (1979). *Circulant matrices* (John Wiley & Sons, Inc.)

Elfving, T., Hansen, P. C., and Nikazad, T. (2017). Convergence analysis for column-action methods in image reconstruction. *Numerical Algorithms* 74, 905–924. doi:10.1007/s11075-016-0176-x

Emilia, D. A. (1973). Equivalent sources used as an analytic base for processing total magnetic field profiles. *GEOPHYSICS* 38, 339–348. doi:10.1190/1.1440344

Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences (Johns Hopkins University Press), 4 edn.

Gonzalez, R. C. and Woods, R. E. (2002). *Digital Image Processing* (Prentice Hall), 2 edn.

Gonzalez, S. P., Barbosa, V. C. F., and Oliveira Jr., V. C. (2022). Analyzing the ambiguity of the remanent-magnetization direction separated into induced and remanent magnetic sources. *Journal of Geophysical Research: Solid Earth* 127, 1–24. doi:10.1029/2022JB024151

Guspí, F., Introcaso, A., and Introcaso, B. (2004). Gravity-enhanced representation of measured geoid undulations using equivalent sources. *Geophysical Journal International* 159, 1–8. doi:10.1111/j.1365-246X.2004.02364.x

Guspí, F. and Novara, I. (2009). Reduction to the pole and transformations of scattered magnetic data using Newtonian equivalent sources. *GEOPHYSICS* 74, L67–L73. doi:10.1190/1.3170690

Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Review* 34, 561–580. doi:10.1137/1034115

Hansen, R. O. and Miyazaki, Y. (1984). Continuation of potential fields between arbitrary surfaces. *GEOPHYSICS* 49, 787–795. doi:10.1190/1.1441707

Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis* (Cambridge University Press), 1 edn.

Jain, A. K. (1989). *Fundamentals of Digital Image Processing* (Pearson), 1 edn.

Jirigalatu, J. and Ebbing (2019). A fast equivalent source method for airborne gravity gradient data. *Geophysics* 84, G75–G82. doi:10.1190/GEO2018-0366.1

Kellogg, O. D. (1967). *Foundations of Potential Theory* (Springer-Verlag), reprint from the first edition of 1929 edn.

Kennett, B., Sambridge, M., and Williamson, P. (1988). Subspace methods for large inverse problems with multiple parameter classes. *Geophysical Journal International* 94, 237–247

Leão, J. W. D. and Silva, J. B. C. (1989). Discrete linear transformations of potential field data. *Geophysics* 54, 497–507. doi:10.1190/1.1442676

Li, Y., Nabighian, M., and Oldenburg, D. W. (2014). Using an equivalent source with positivity for low-latitude reduction to the pole without striation. *GEOPHYSICS* 79, J81–J90. doi:10.1190/geo2014-0134.1

Li, Y. and Oldenburg, D. W. (2010). Rapid construction of equivalent sources using wavelets. *GEOPHYSICS* 75, L51–L59. doi:10.1190/1.3378764

Mendonça, C. A. (2020). Subspace method for solving large-scale equivalent layer and density mapping problems. *GEOPHYSICS* 85, G57–G68. doi:10.1190/geo2019-0302.1

Mendonça, C. A. and Silva, J. B. C. (1994). The equivalent data concept applied to the interpolation of potential field data. *Geophysics* 59, 722–732. doi:10.1190/1.1443630

Menke, W. (2018). *Geophysical data analysis: Discrete inverse theory* (Elsevier), 4 edn.

Moroni, M., Girardi, V., and Ferrario, A. (2001). The Serra Pelada Au-PGE deposit, Serra dos Carajás (Pará state, Brazil): geological and geochemical indications for a composite mineralising process. *Mineralium Deposita* 36, 768–785

Oldenburg, D., McGillivray, P., and Ellis, R. (1993). Generalized subspace methods for large-scale inverse problems. *Geophysical Journal International* 114, 12–20

Oliveira Jr., V. C., Barbosa, V. C. F., and Uieda, L. (2013). Polynomial equivalent layer. *GEOPHYSICS* 78, G1–G13. doi:10.1190/geo2012-0196.1

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical recipes: the art of scientific computing* (Cambridge University Press), 3 edn.

Reis, A. L. A., Oliveira Jr., V. C., and Barbosa, V. C. F. (2020). Generalized positivity constraint on magnetic equivalent layers. *Geophysics* 85, 1–45. doi:10.1190/geo2019-0706.1

Roy, A. (1962). Ambiguity in geophysical interpretation. *GEOPHYSICS* 27, 90–99. doi:10.1190/1.1438985

Salomao, G. N., Dall'Agnol, R., Angelica, R. S., Figueiredo, M. A., Sahoo, P. K., Medeiros-Filho, C. A., et al. (2019). Geochemical mapping and estimation of background concentrations in soils of Carajás mineral province, eastern Amazonian Craton, Brazil. *Geochemistry: Exploration, Environment, Analysis* 19, 431–447

Santos, J. O. S., Hartmann, L. A., Gaudette, H. E., Groves, D. I., Mcnaughton, M. J., and Fletcher, I. R. (2000). A new understanding of the provinces of the Amazon Craton based on integration of field mapping and U-Pb and Sm-Nd geochronology. *Gondwana Research* 3, 453–488

Silva, J. B. C. (1986). Reduction to the pole as an inverse problem and its application to low-latitude anomalies. *GEOPHYSICS* 51, 369–382. doi:10.1190/1.1442096

Siqueira, F., Oliveira Jr., V. C., and Barbosa, V. C. F. (2017). Fast iterative equivalent-layer technique for gravity data processing: A method grounded on excess mass constraint. *GEOPHYSICS* 82, G57–G69. doi:10.1190/GEO2016-0332.1

Skilling, J. and Bryan, R. (1984). Maximum entropy image reconstruction-general algorithm. *Monthly Notices of the Royal Astronomical Society, Vol. 211, NO. 1, P. 111, 1984* 211, 111

Soler, S. R. and Uieda, L. (2021). Gradient-boosted equivalent sources. *Geophysical Journal International* 227, 1768–1783. doi:10.1093/gji/ggab297

Takahashi, D., Oliveira Jr., V. C., and Barbosa, V. C. (2022). Convolutional equivalent layer for magnetic data processing. *Geophysics* 87, 1–59

Takahashi, D., Oliveira Jr., V. C., and Barbosa, V. C. F. (2020). Convolutional equivalent layer for gravity data processing. *GEOPHYSICS* 85, G129–G141. doi:10.1190/geo2019-0826.1

Tassinari, C. C. and Macambira, M. J. (1999). Geochronological provinces of the Amazonian Craton. *Episodes* 22, 174–182

Teixeira, W., Tassinari, C., Cordani, U. G., and Kawashita, K. (1989). A review of the geochronology of the Amazonian Craton: Tectonic implications. *Precambrian Research* 42, 213–227

van der Sluis, A. and van der Vorst, H. A. (1987). Numerical solution of large, sparse linear algebraic systems arising from tomographic problems. In *Seismic tomography with applications in global seismology and exploration geophysics*, ed. G. Nolet (D. Reidel Publishing Company), chap. 3. 49–83

Van Loan, C. F. (1992). *Computational Frameworks for the fast Fourier transform.* Frontiers in Applied Mathematics (SIAM)

Villas, R. N. and Santos, M. (2001). Gold deposits of the Carajás mineral province: deposit types and metallogenesis. *Mineralium Deposita* 36, 300–331

Xia, J. and Sprowl, D. R. (1991). Correction of topographic distortion in gravity data. *Geophysics* 56, 537–541

Xia, J., Sprowl, D. R., and Adkins-Heljeson, D. (1993). Correction of topographic distortions in potential-field data; a fast and accurate approach. *Geophysics* 58, 515–523. doi:10.1190/1.1443434

Zhao, G., Chen, B., Chen, L., Liu, J., and Ren, Z. (2018). High-accuracy 3D Fourier forward modeling of gravity field based on the Gauss-FFT technique. *Journal of Applied Geophysics* 150, 294–303. doi:10.1016/j.jappgeo.2018.01.002

Zidarov, D. (1965). Solution of some inverse problems of applied geophysics. *Geophysical Prospecting* 13, 240–246. doi:10.1111/j.1365-2478.1965.tb01932.x

11 ALGORITHMS

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

Initialization :
1 Compute $\mathbf{G}$;
2 Set $\mathbf{r} = \mathbf{d}$ and compute $\delta = \|\mathbf{r}\|/D$;
3 Compute $\boldsymbol{\vartheta} = \mathbf{G}^\top \mathbf{r}$ and $\rho_0 = \boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$;
4 Set $\tilde{\mathbf{p}} = \mathbf{0}$, $\tau = 0$ and $\boldsymbol{\eta} = \mathbf{0}$;
5 $m = 1$;
6 **while** $(\delta > \epsilon)$ **and** $(m < \texttt{ITMAX})$ **do**
7 $\quad$ Update $\boldsymbol{\eta} \leftarrow \boldsymbol{\vartheta} + \tau \, \boldsymbol{\eta}$;
8 $\quad$ Compute $\boldsymbol{\nu} = \mathbf{G} \, \boldsymbol{\eta}$;
9 $\quad$ Compute $\upsilon = \rho_0/(\boldsymbol{\nu}^\top \boldsymbol{\nu})$;
10 $\quad$ Update $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + \upsilon \, \boldsymbol{\eta}$;
11 $\quad$ Update $\mathbf{r} \leftarrow \mathbf{r} - \upsilon \, \boldsymbol{\nu}$ and $\delta \leftarrow \|\upsilon \boldsymbol{\nu}\|/D$;
12 $\quad$ Compute $\boldsymbol{\vartheta} = \mathbf{G}^\top \mathbf{r}$ and $\rho = \boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$;
13 $\quad$ Compute $\tau = \rho/\rho_0$;
14 $\quad$ Update $\rho_0 \leftarrow \rho$;
15 $\quad$ $m \leftarrow m + 1$;
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 $\quad$ Compute t_c^m (equation 33) ;
11 $\quad$ $m \leftarrow m + 1$;
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 $\mathbf{\Delta z}$ whose i-th element Δz_i is a vertical distance controlling the depth of the i-th equivalent source, $i \in \{1 : D\}$;
2 Set a tolerance ϵ ;
3 Set a maximum number of iterations ITMAX ;
4 Set a $D \times 1$ residuals vector $\mathbf{r} = \mathbf{d}$;
5 Set a $D \times 1$ vector $\tilde{\mathbf{p}} = \mathbf{0}$;
6 Define the maximum absolute value $r_{\mathtt{max}}$ in $\mathbf{r}$;
7 $m = 1$;
8 **while** $(r_{\mathtt{max}} > \epsilon)$ **and** $(m < \mathtt{ITMAX})$ **do**
9   Define the coordinates $(x_{\mathtt{max}}, y_{\mathtt{max}}, z_{\mathtt{max}})$ and index $i_{\mathtt{max}}$ of the observation point associated with
      $r_{\mathtt{max}}$;
10    $\tilde{\mathbf{p}}[i_{\mathtt{max}}] \leftarrow \tilde{\mathbf{p}}[i_{\mathtt{max}}] + (r_{\mathtt{max}}\, \mathbf{\Delta z}[i_{\mathtt{max}}])$;
11    $\mathbf{r} \leftarrow \mathbf{r} - (\mathbf{G}[:, i_{\mathtt{max}}]\, \tilde{\mathbf{p}}[i_{\mathtt{max}}])$;
12   Define the new $r_{\mathtt{max}}$ 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 $r_{\mathtt{max}}$ in $\mathbf{r}$;
5 Define the index $i_{\mathtt{max}}$ of $r_{\mathtt{max}}$;
6 Define the list of indices $\mathbf{i}_r$ of the remaining data in $\mathbf{r}$;
7 Define $\mathbf{d}_e = \mathbf{d}[i_{\mathtt{max}}]$;
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 $r_{\mathtt{max}}$ in $\mathbf{r}$;
12 **while** $(r_{\mathtt{max}} > \epsilon)$ **do**
13 $\quad$ Define the index $i_{\mathtt{max}}$ of $r_{\mathtt{max}}$;
14 $\quad$ Define the list of indices $\mathbf{i}_r$ of the remaining elements in $\mathbf{r}$;
15 $\quad \mathbf{d}_e \leftarrow \begin{bmatrix} \mathbf{d}_e \\ \mathbf{d}[i_{\mathtt{max}}] \end{bmatrix}$;
16 $\quad$ Update $(\mathbf{F} + \mu\,\mathbf{I}_{D_e})$ and $\mathbf{G}_e$;
17 $\quad$ Update $\tilde{\mathbf{p}}$ (equation 40) ;
18 $\quad$ Update $\mathbf{r} = \mathbf{d}[\mathbf{i}_r] - \mathbf{G}[\mathbf{i}_r, :]\,\tilde{\mathbf{p}}$;
19 $\quad$ Define the maximum absolute value $r_{\mathtt{max}}$ in $\mathbf{r}$;
20 **end**

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

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

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

Initialization :
1 Set the regular grid of P equivalent sources on a horizontal plane z_0 ;
2 Set a tolerance ϵ and a maximum number of iterations ITMAX ;
3 Compute the first column $\mathbf{G}[:, 1]$ and row $\mathbf{G}[1, :]$ of the sensitivity matrix $\mathbf{G}$ (equation 3) for the particular case in which it has a BTTB structure (equation 51);
4 Rearrange the elements of $\mathbf{G}[:, 1]$ into matrix $\mathcal{C}$, 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 $\boldsymbol{\vartheta} = \mathbf{G}^{\top} \mathbf{r}$ (Algorithm 8) and $\rho_0 = \boldsymbol{\vartheta}^{\top} \boldsymbol{\vartheta}$;
9 Set $\tau = 0$ and $\boldsymbol{\eta} = \mathbf{0}$;
10 $m = 1$;
11 **while** $(\delta > \epsilon)$ **and** $(m < \text{ITMAX})$ **do**
12 $\quad$ Update $\boldsymbol{\eta} \leftarrow \boldsymbol{\vartheta} + \tau \boldsymbol{\eta}$;
13 $\quad$ Compute $\boldsymbol{\nu} = \mathbf{G} \boldsymbol{\eta}$ (Algorithm 8);
14 $\quad$ Compute $\upsilon = \rho_0/(\boldsymbol{\nu}^{\top} \boldsymbol{\nu})$;
15 $\quad$ Update $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + \upsilon \boldsymbol{\eta}$;
16 $\quad$ Update $\mathbf{r} \leftarrow \mathbf{r} - \upsilon \boldsymbol{\nu}$ and $\delta \leftarrow \|\upsilon \boldsymbol{\nu}\|/D$;
17 $\quad$ Compute $\boldsymbol{\vartheta} = \mathbf{G}^{\top} \mathbf{r}$ (Algorithm 8) and $\rho = \boldsymbol{\vartheta}^{\top} \boldsymbol{\vartheta}$;
18 $\quad$ Compute $\tau = \rho/\rho_0$;
19 $\quad$ Update $\rho_0 \leftarrow \rho$;
20 $\quad$ $m \leftarrow m + 1$;
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	$\mathbf{G\,H}$	$2DQP$
eq. 15	$\mathbf{H\,\tilde{q}}$	$2PQ$
eq. 22	$(\mathbf{G\,H})^{\top}(\mathbf{G\,H})$	$2Q^2D$
eq. 22	$(\mathbf{G\,H})^{\top}\boldsymbol{\delta}_d$	$2QD$
eq. 23	$(\mathbf{G\,H})(\mathbf{G\,H})^{\top}$	$2D^2Q$
eq. 23	$(\mathbf{G\,H})^{\top}\mathbf{u}$	$2QD$
eq. 25	lower triangle of $\mathcal{G}$	$D^3/3$ or $Q^3/3$
eq. 26	solve triangular systems	$2D^2$ or $2Q^2$
Alg. 1	$\boldsymbol{\eta} \leftarrow \boldsymbol{\vartheta} + \tau\,\boldsymbol{\eta}$	$2Q$
Alg. 1	$\boldsymbol{\vartheta}^{\top}\boldsymbol{\vartheta}$	$2Q$
Alg. 6	$\boldsymbol{\sigma} \circ \mathbf{d}$	D

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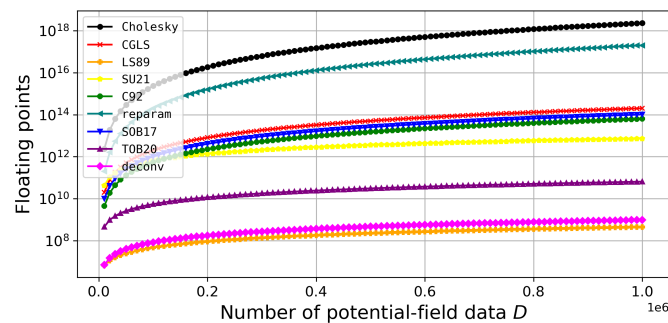
13 FIGURES

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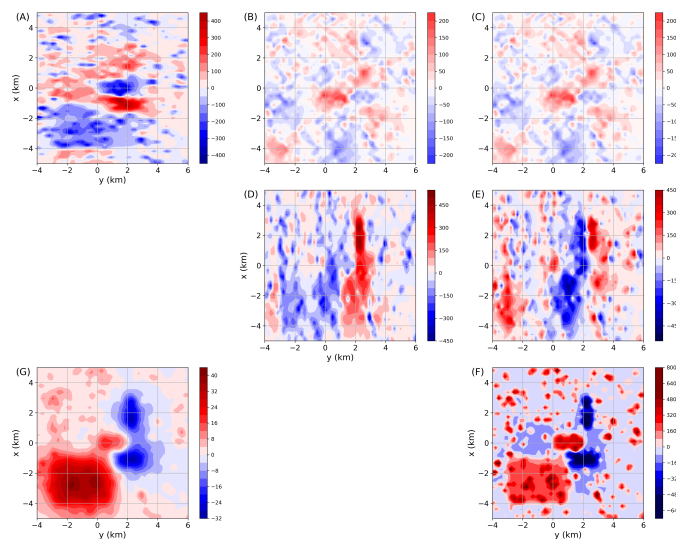

Figure 2. Synthetic gravity disturbance data (in mGal). The data are located on a regular grid of 50×50 points. Panel (A) shows the noise-free data. Panel (B) shows the synthetic data corrupted with a pseudorandom Gaussian noise having zero mean and standard deviation equal to 10% of the maximum absolute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

Figure 3. Numerical stability curves obtained for the 21 synthetic gravity data sets by using the Cholesky factorization with $\mu \approx 10^{-22}$, the iterative deconvolution (TOB20) proposed by Takahashi et al. (2020) with 40 iterations (Algorithm 7) and the direct deconvolution (deconv.) computed with four different values for ζ (equation 69): 0, 10^{-20} (overshoot), 10^{-22} (optimal) and 10^{-24} (suboptimal). The stability parameter κ (equation 29) obtained for the six curves described above are 24.66 (Cholesky), 25.81 (TOB20), 90.93, 1.95, 17.47 and 41.24 (deconv. with null, overshoot, optimal and suboptimal ζ).

Figure 4. Residuals between the predicted data (equation 3) due to the estimated parameter vector $\tilde{\mathbf{p}}$ obtained with the gravity data having maximum noise level (Figure **??**B) by using **(A)** the Cholesky factorization with $\mu \approx 10^{-22}$, **(B)** the iterative deconvolution (TOB20) proposed by Takahashi et al. (2020) with 40 iterations (Algorithm 7) and **(C)**–**(F)** the direct deconvolution (deconv.) computed, respectively, with four different values for ζ (equation 69): 0, 10^{-20} (overshoot), 10^{-22} (optimal) and 10^{-24} (suboptimal). All values are in mGal.

Figure 5. Upward continuation of the synthetic gravity data with maximum noise level (Figure **??**B). **(A)** Noise-free gravity data produced by the model at $z = -500$ m. **(B)**–**(E)** Continued data due to the estimated parameter vector $\tilde{\mathbf{p}}$ obtained, respectively, by using the Cholesky factorization with $\mu \approx 10^{-22}$, the iterative deconvolution (TOB20) proposed by Takahashi et al. (2020) with 40 iterations (Algorithm 7) and the direct deconvolution (deconv.) computed, respectively, with two different values for ζ (equation 69): 0 and 10^{-22} (optimal). All values are in mGal.
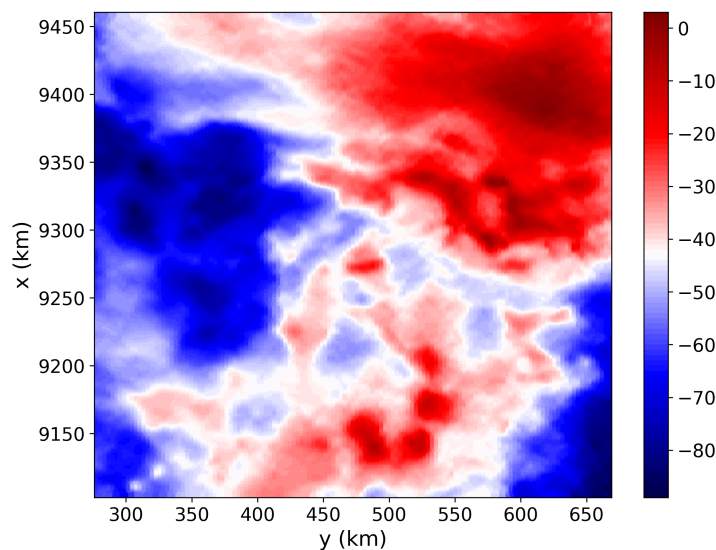

Figure 6. Field aerogravimetric data over Carajás, Brazil. There are $D = 500,000$ observations located on regular grid of $1,000 \times 500$ poins.
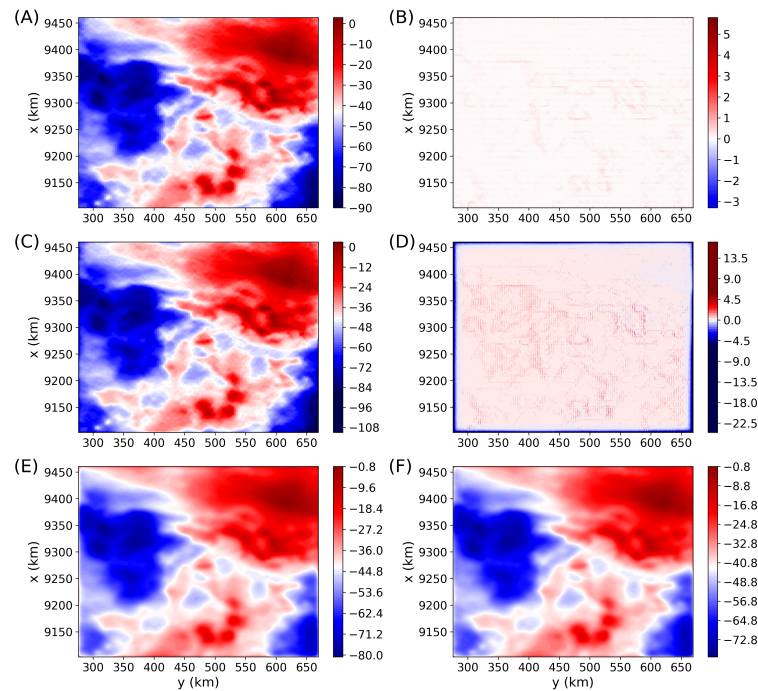
Figure 7. Upward continuation of the aerogravimetric data over Carajás, Brazil. **(A)** Predicted data due to the equivalent layer obtained from the iterative deconvolution (Algorithm 7) with 50 iterations. **(B)** Residuals between the predicted data shown in **(A)** and the observed data (Figure **??**). **(C)** Predicted data due to the equivalent layer obtained from the direct deconvolution computed with $\zeta = 10^{-22}$ (equation 69). **(D)** Residuals between the predicted data shown in **(C)** and the observed data (Figure **??**). **(E)** and **(F)** Continued field at $z = -3500\text{m}$ obtained from iterative and direct deconvolutions.