

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

Diego Takahashi <sup>1,\*</sup>, André L. A. Reis <sup>2</sup>, Vanderlei C. Oliveira Jr. <sup>1</sup> and Valéria C. F. Barbosa <sup>1</sup>

<sup>1</sup>*Observatório Nacional, Department of Geophysics, Rio de Janeiro, Brasil*

<sup>2</sup>*Universidade do Estado do Rio de Janeiro, Department of Applied Geology, Rio de Janeiro, Brasil*

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

## 2 ABSTRACT

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

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

## 1 INTRODUCTION

In accordance with potential theory, a continuous potential-field data (gravity and magnetic data) produced by any source can be exactly reproduced by a continuous and infinite 2D physical-property surface distribution called the equivalent layer. The equivalent layer is a mathematical solution of Laplace's equation in the source-free region with the observed potential-field data as the Dirichlet boundary condition (?). Grounded on well-established potential theory, the equivalent-layer technique has been used by exploration geophysicists for processing potential-field data since the late 1960s (Dampney, 1969).

Although there has always been a high demand for gravity and magnetic data processing, the equivalent-layer technique has not been massively used. This occurs because its high computational cost makes it inefficient for processing massive data sets. In the classic equivalent-layer technique, the continuous problem of the equivalent layer involving integrals is approximated by a discrete form. First, a discrete and finite set of equivalent sources (such as point masses, prisms, magnetic dipoles, doublets) is arranged in a layer with finite horizontal dimensions beneath the observation surface. Next, a linear system of equations is set up with a large and full sensitivity matrix. Then, a regularized linear inverse problem is solved to estimate the physical property of each equivalent source within the discrete equivalent layer subject to fitting a discrete set of potential-field observations. Finally, the estimated physical-property distribution within the equivalent layer is used to accomplish the desired processing of the potential-field data (e.g., interpolation, upward/downward continuation, reduction to the pole). The latter step involves multiplying the matrix of Green's functions associated with the desired transformation by the estimated physical-property distribution.

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

As another strategy to reduce the computational workload of the equivalent-layer technique, some authors have employed column- and row-action updates. These methods involve iterative calculations of a single

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

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

96 Following the strategy of the wavelet compression, ? transformed the full sensitivity matrix into a sparse  
97 one using the compression of the coefficient matrix via wavelet transforms based on the orthonormal  
98 compactly supported wavelets.

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

114 (2019) efficiently compute the predicted data and residual data for the entire survey data per iteration in the  
115 wavenumber domain.

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

123 The method introduced by Takahashi et al. (2020, 2022) can be reformulated to eliminate the need for  
124 conjugate gradient iterations. This reformulation involves employing a *direct deconvolution* approach,  
125 similar to the concept described by (e.g., ?, p. 220), utilizing a *Wiener filter* (e.g., ?, p. 263).

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

## 2 FUNDAMENTALS

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

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

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

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

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

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

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

144 The equivalent-layer technique consists in solving a linear inverse problem to determine a parameter  
 145 vector  $\mathbf{p}$  leading to a predicted data vector  $\mathbf{f}$  (equation 58) *sufficiently close to* the observed data vector  $\mathbf{d}$ ,  
 146 whose  $i$ -th element  $d_i$  is the observed potential field at  $(x_i, y_i, z_i)$ . The notion of *closeness* is intrinsically  
 147 related to the concept of *vector norm* (e.g., Golub and Loan, 2013, p. 68) or *measure of length* (e.g., ?,  
 148 p. 41). Because of that, almost all methods for determining  $\mathbf{p}$  actually estimate a parameter vector  $\tilde{\mathbf{p}}$   
 149 minimizing a length measure of the difference between  $\mathbf{f}$  and  $\mathbf{d}$  (see subsection 2.3). Given an estimate  $\tilde{\mathbf{p}}$ ,  
 150 it is then possible to compute a potential field transformation

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

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

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

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

### 154 2.1 Spatial distribution and total number of equivalent sources

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

161 From a theoretical point of view, the equivalent layer reproducing a given potential field data set cannot  
 162 cross the true gravity or magnetic sources. This condition is a consequence of recognizing that the equivalent  
 163 layer is essentially an indirect solution of a boundary value problem of potential theory (e.g., ??Dampney,  
 164 1969; ?; ?). In practical applications, however, there is no guarantee that this condition is satisfied. Actually,  
 165 its is widely known from practical experience (e.g., ?) that the equivalent-layer technique works even for  
 166 the case in which the layer cross 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  
 169 grid spacing, considering evenly spaced data. However, when dealing with a survey pattern that has  
 170 unevenly spaced data, ? adopted an alternative empirical criterion. According to their proposal, the depth  
 171 of the equivalent layer should range from two to three times the spacing between adjacent flight lines. The  
 172 criteria of Dampney (1969) and ? are valid for planar equivalent layers. Cordell (1992) have proposed  
 173 and an alternative criterion for scattered data that leads to an undulating equivalent layer. This criterion  
 174 have been slightly modified by ?, Guspí and Novara (2009) and Soler and Uieda (2021), for example,  
 175 and consists in setting one equivalent source below each datum at a depth proportional to the horizontal  
 176 distance to the nearest neighboring data points. Soler and Uieda (2021) have compared different strategies  
 177 for defining the equivalent sources depth for the specific problem of interpolating gravity data, but they  
 178 have not found significant differences between them.

## 179 2.2 Matrix G

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

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

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

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

183 and

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

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

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

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

The main challenge in the equivalent-layer technique is the computational complexity associated with handling large datasets. This complexity arises because the sensitivity matrix  $\mathbf{G}$  (equation 58) is dense regardless of the harmonic function  $g_{ij}$  (equation 2) employed. In the case of scattered potential-field data, the structure of  $\mathbf{G}$  is not well-defined, regardless of the spatial distribution of the equivalent sources. However, in a specific scenario where (i) each potential-field datum is directly associated with a single equivalent source located directly below it, and (ii) both the data and sources are based on planar and regularly spaced grids, ?? demonstrate that  $\mathbf{G}$  exhibits a block-Toeplitz Toeplitz-block (BTTB) structure. In such cases, the product of  $\mathbf{G}$  and an arbitrary vector can be efficiently computed using a 2D fast Fourier transform as a discrete convolution.

## 2.3 General formulation

A general formulation for almost all equivalent-layer methods can be achieved by first considering that the  $P \times 1$  parameter vector  $\mathbf{p}$  (equation 58) can be reparameterized into a  $Q \times 1$  vector  $\mathbf{q}$  according to:

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

where  $\mathbf{H}$  is a  $P \times Q$  matrix. The predicted data vector  $\mathbf{f}$  (equation 58) can then be rewritten as follows:

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

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

In this case, the problem of estimating a parameter vector  $\tilde{\mathbf{p}}$  minimizing a length measure of the difference between  $\mathbf{f}$  (equation 58) and  $\mathbf{d}$  is replaced by that of estimating an auxiliary vector  $\tilde{\mathbf{q}}$  minimizing the goal function

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

which is a combination of particular measures of length given by

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

and

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

where the regularization parameter  $\mu$  is a positive scalar controlling the trade-off between the data-misfit function  $\Phi(\mathbf{q})$  and the regularization function  $\Theta(\mathbf{q})$ ;  $\mathbf{W}_d$  is a  $D \times D$  symmetric matrix defining the relative

226 importance of each observed datum  $d_i$ ;  $\mathbf{W}_q$  is a  $Q \times Q$  symmetric matrix imposing prior information on  $\mathbf{q}$ ;  
 227 and  $\bar{\mathbf{q}}$  is a  $Q \times 1$  vector of reference values for  $\mathbf{q}$  that satisfies

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

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

229 After obtaining an estimate  $\tilde{\mathbf{q}}$  for the reparameterized parameter vector  $\mathbf{q}$  (equation 9), the estimate  $\tilde{\mathbf{p}}$  for  
 230 the original parameter vector (equation 58) is computed by

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

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

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

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

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

234 where

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

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

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

237 or, equivalently (? , p. 62),

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

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

239 The  $Q \times D$  matrix  $\mathbf{B}$  defined by equation 20 is commonly used for the case in which  $D > Q$ , i.e., when  
 240 there are more data than parameters (overdetermined problems). In this case, we consider that the estimate  
 241  $\tilde{\mathbf{q}}$  is obtained by solving the following linear system for  $\tilde{\boldsymbol{\delta}}_q$  (equation 18):

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

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

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

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

### 248 2.3.1 Classical approach

249 The classical approach in the equivalent-layer technique consists in using  $\mathbf{H} = \mathbf{I}_P$ , so that  $P = Q$ ,  $\mathbf{p} = \mathbf{q}$   
250 (equation 9),  $\bar{\mathbf{p}} = \bar{\mathbf{q}}$  (equation 14) and  $\tilde{\mathbf{p}} = \tilde{\mathbf{q}}$  (equation 15). In this case, the linear system (equations 22  
251 and 23) is directly solved for

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

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

## 3 NOTATION FOR SUBVECTORS AND SUBMATRICES

253 Here, we use a notation inspired on that presented by (Van Loan, 1992, p. 4) to represent subvectors and  
254 submatrices. Subvectors of  $\mathbf{d}$ , for example, are specified by  $\mathbf{d}[\mathbf{i}]$ , where  $\mathbf{i}$  is a list of integer numbers that  
255 “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  
256  $\mathbf{d}[\mathbf{i}] = [d_1 \ d_6 \ d_4 \ d_6]^\top$ . Note that the list  $\mathbf{i}$  of indices may be sorted or not and it may also have repeated  
257 indices. For the particular case in which the list has a single element  $\mathbf{i} = (i)$ , then it can be used to extract  
258 the  $i$ -th element  $d_i \equiv \mathbf{d}[i]$  of  $\mathbf{d}$ . Sequential lists with increment of 1, if the starting index is smaller than the  
259 final index, or  $-1$ , if the starting index is greater than the final index, can be represented by using the colon  
260 notation. For example,

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

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

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

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

264 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  
265 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  $\mathbf{G}[:, j]$   
266 represents the  $j$ -th column. Finally, we may use the colon notation to define the following submatrix:

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

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

## 4 COMPUTATIONAL STRATEGIES

268 Here, we review some strategies for reducing the computational cost of equivalent-layer technique. Typically, estimating a parameter vector  $\tilde{\mathbf{p}}$  or  $\tilde{\mathbf{q}}$  requires to solve a large-scale linear inversion (equations 22 and 23). This, in turn, means to deal with some obstacles concerning large computational cost:

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

274 Two important factors affecting the efficiency of a given matrix algorithm are the storage and amount of required arithmetic. Here, we quantify this last factor by counting *flops*, which are floating point additions, subtractions, multiplications or divisions (Golub and Loan, 2013, p. 12–14). We focus on the overall strategies used by the selected methods.

### 278 4.1 Moving window

279 The initial approach to enhance the computational efficiency of the equivalent-layer technique is commonly denoted *moving window* and involves first splitting the observed data  $d_i$ ,  $i \in \{1 : D\}$ , into  $M$  overlapping subsets (or data windows) formed by  $D^m$  data each,  $m \in \{1 : M\}$ . The data inside the  $m$ -th window are usually adjacent to each other and have indices defined by an integer list  $\mathbf{i}^m$  having  $D^m$  elements. The number of data  $D^m$  forming the data windows are not necessarily equal to each other. 284 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 285 a set of  $P$  equivalent sources with scalar physical property  $p_j$ ,  $j \in \{1 : P\}$ , and also split them into  $M$  286 overlapping subsets (or source windows) formed by  $P^m$  data each,  $m \in \{1 : M\}$ . The sources inside the 287  $m$ -th window have indices defined by an integer list  $\mathbf{j}^m$  having  $P^m$  elements. Each source window has a 288  $P^m \times 1$  parameter vector  $\mathbf{p}^m$  and is located right below the corresponding  $m$ -th data window. Then, each 289  $\mathbf{d}^m \equiv \mathbf{d}[\mathbf{i}^m]$  is approximated by

$$\mathbf{f}^m = \mathbf{G}^m \mathbf{p}^m, \quad (25)$$

290 where  $\mathbf{G}^m \equiv \mathbf{G}[\mathbf{i}^m, \mathbf{j}^m]$  is a submatrix of  $\mathbf{G}$  (equation 58) formed by the elements computed with equation 291 2 using only the data and equivalent sources located inside the window  $m$ -th. The main idea of the moving- 292 window approach is using the  $\tilde{\mathbf{p}}^m$  estimated for each window to obtain (i) an estimate  $\tilde{\mathbf{p}}$  of the parameter 293 vector for the entire equivalent layer or (ii) a given potential-field transformation  $\mathbf{t}$  (equation 4). The main 294 advantages of this approach is that (i) the estimated parameter vector  $\tilde{\mathbf{p}}$  or transformed potential field are 295 not obtained by solving the full, but smaller linear systems and (ii) the full matrix  $\mathbf{G}$  (equation 58) is never 296 stored.

297 Leão and Silva (1989) presented a pioneer work using the moving-window approach. Their method 298 requires a regularly-spaced grid of observed data on a horizontal plane  $z_0$ . The data windows are defined by 299 square local grids of  $\sqrt{D'} \times \sqrt{D'}$  adjacent points, all of them having the same number of points  $D'$ . The 300 equivalent sources in the  $m$ -th data window are located below the observation plane, at a constant vertical 301 distance  $\Delta z_0$ . They are arranged on a regular grid of  $\sqrt{P'} \times \sqrt{P'}$  adjacent points following the same 302 grid pattern of the observed data. The local grid of sources for all data windows have the same number 303 of elements  $P'$ . Besides, they are vertically aligned, but expands the limits of their corresponding data 304 windows, so that  $D' < P'$ . Because of this spatial configuration of observed data and equivalent sources, 305 we have that  $\mathbf{G}^m = \mathbf{G}'$  (equation 25) for all data windows (i.e.,  $\forall m \in \{1 : M\}$ ), where  $\mathbf{G}'$  is a  $D' \times P'$  306 constant matrix.

307 By omitting the normalization strategy used by Leão and Silva (1989), their method consists in directly  
 308 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  
 309 as follows:

$$t_c^m = (\mathbf{a}')^\top \mathbf{B}' \mathbf{d}^m, \quad m \in \{1 : M\}, \quad (26)$$

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

$$\mathbf{B}' = (\mathbf{G}')^\top \left[ \mathbf{G}' (\mathbf{G}')^\top + \mu \mathbf{I}_{D'} \right]^{-1} \quad (27)$$

312 is a particular case of matrix  $\mathbf{B}$  associated with underdetermined problems (equation 21) for the particular  
 313 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  
 314  $\mathbf{I}_{P'}$  and  $\mathbf{I}_{D'}$  are identity matrices of order  $P'$  and  $D'$ , respectively, and  $\mathbf{0}$  is a vector of zeros. Due to the  
 315 presumed spatial configuration of the observed data and equivalent sources,  $\mathbf{a}'$  and  $\mathbf{G}'$  are the same for all  
 316 data windows. Hence, only the data vector  $\mathbf{d}^m$  is modified according to the position of the data window.  
 317 Note that equation 26 combines the potential-field transformation (equation 4) with the solution of the  
 318 undetermined problem (equation 23).

319 The method proposed by Leão and Silva (1989) can be outlined by the Algorithm 1. Note that Leão and  
 320 Silva (1989) directly compute the transformed potential  $t_c^m$  at the central point of each data window without  
 321 explicitly computing and storing an estimated for  $\mathbf{p}^m$  (equation 25). It means that their method allows  
 322 computing a single potential-field transformation. A different transformation or the same one evaluated at  
 323 different points require running their moving-data window method again.

---

**Algorithm 1:** 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 27) ;
  - 7 Compute the vector  $(\mathbf{a}')^\top \mathbf{B}'$  ;
  - 8  $m = 1$  ;
  - 9 **while**  $m < M$  **do**
  - 10   | Compute  $t_c^m$  (equation 26) ;
  - 11   |  $m \leftarrow m + 1$  ;
  - 12 **end**
- 

324 Soler and Uieda (2021) generalized the method proposed by Leão and Silva (1989) for irregularly spaced  
 325 data on an undulating surface. A direct consequence of this generalization is that a different submatrix  
 326  $\mathbf{G}^m \equiv \mathbf{G}[\mathbf{i}^m, \mathbf{j}^m]$  (equation 25) must be computed for each window. Differently from Leão and Silva  
 327 (1989), Soler and Uieda (2021) store the computed  $\tilde{\mathbf{p}}^m$  for all windows and subsequently use them to obtain  
 328 a desired potential-field transformation (equation 4) as the superposed effect of all windows. The estimated  
 329  $\tilde{\mathbf{p}}^m$  for all windows are combined to form a single  $P \times 1$  vector  $\tilde{\mathbf{p}}$ , which is an estimate for original  
 330 parameter vector  $\mathbf{p}$  (equation 58). For each data window, Soler and Uieda (2021) solve an overdetermined  
 331 problem (equation 22) for  $\tilde{\mathbf{p}}^m$  by using  $\mathbf{H} = \mathbf{W}_q = \mathbf{I}_{P^m}$  (equations 9 and 13),  $\mathbf{W}_d^m$  (equation 12) equal to

332 a diagonal matrix of weights for the data inside the  $m$ -th window and  $\bar{\mathbf{p}} = \mathbf{0}$  (equation 14), so that

$$\left[ (\mathbf{G}^m)^\top \mathbf{W}_d^m \mathbf{G}^m + \mu \mathbf{I}_{P'} \right] \tilde{\mathbf{p}}^m = (\mathbf{G}^m)^\top \mathbf{W}_d^m \mathbf{d}^m. \quad (28)$$

333 Unlike Leão and Silva (1989), Soler and Uieda (2021) do not adopt a sequential order of the data  
 334 windows; rather, they adopt a randomized order of windows in their iterations. The overall steps of the  
 335 method proposed by Soler and Uieda (2021) are defined by the Algorithm 2. For convenience, we have  
 336 omitted the details about the randomized window order and the normalization strategy employed by Soler  
 337 and Uieda (2021). Note that this algorithm starts with a residuals vector  $\mathbf{r}$  that is iteratively updated. The  
 338 iterative algorithm in Soler and Uieda (2021) estimates a solution ( $\tilde{\mathbf{p}}^m$  in equation 28) using the data and  
 339 the equivalent sources that fall within a moving-data window; however, it calculates the predicted data  
 340 and the residual data in the whole survey data. Next, the residual data that fall within a new position of  
 341 the data window is used as input data to estimate a new solution within the data window which, in turn, is  
 342 used to calculate a new predicted data and a new residual data in the whole survey data. Regarding the  
 343 equivalent-source layout, Soler and Uieda (2021) proposed the block-averaged sources locations in which  
 344 the survey area is divided into horizontal blocks and one single equivalent source is assigned to each block.  
 345 Each single source per block is placed over the layer with its horizontal coordinates given by the average  
 346 horizontal positions of observation points. According to Soler and Uieda (2021), the block-averaged  
 347 sources layout may prevent aliasing of the interpolated values, specially when the observations are unevenly  
 348 sampled. This strategy also reduces the number of equivalent sources without affecting the accuracy of  
 349 the potential-field interpolation. This reduction reduces the computational load for estimating the physical  
 350 property on the equivalent layer. This block-averaged sources layout is not included in the Algorithm 2.

---

**Algorithm 2:** 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 28) ;
  - 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**
- 

351 **4.2 Column-action update**

352 We call the computational strategy *column-action update* because a single source is used to calculate the  
 353 predicted data and the residual data in the whole survey data. Hence, a single column of the sensitivity  
 354 matrix  $\mathbf{G}$  (equation 58) is calculated iteratively.

355 Cordell (1992) proposed a computational strategy that was later used by Guspí and Novara (2009) and  
 356 relies on first defining one equivalent source located right below each observed data  $d_i$ ,  $i \in \{1 : D\}$ , at  
 357 a vertical coordinate  $z_i + \Delta z_i$ , where  $\Delta z_i$  is proportional to the distance from the  $i$ -th observation point  
 358 ( $x_i, y_i, z_i$ ) to its closest neighbor. The second step consists in updating the physical property  $p_j$  of a single  
 359 equivalent source,  $j \in \{1 : D\}$  and remove its predicted potential field from the observed data vector  $\mathbf{d}$ ,  
 360 producing a residuals vector  $\mathbf{r}$ . At each iteration, the single equivalent source is the one located vertically  
 361 beneath the observation station of the maximum data residual. Next, the predicted data produced by this  
 362 single source is calculated over all of the observation points and a new data residual  $\mathbf{r}$  and the  $D \times 1$   
 363 parameter vector  $\mathbf{p}$  containing the physical property of all equivalent sources are updated iteratively. During  
 364 each subsequent iteration, Cordell's method either incorporates a single equivalent source or adjusts an  
 365 existing equivalent source to match the maximum amplitude of the current residual field. The convergence  
 366 occurs when all of the residuals are bounded by an envelope of prespecified expected error. At the end, the  
 367 algorithm produces an estimate  $\tilde{\mathbf{p}}$  for the parameter vector yielding a predicted potential field  $\mathbf{f}$  (equation  
 368 58) satisfactorily fitting the observed data  $\mathbf{d}$  according to a given criterion. Note that the method proposed  
 369 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  
 370 single column of  $\mathbf{G}$  (equation 58) is used. An advantage of this *column-action update approach* is that the  
 371 full matrix  $\mathbf{G}$  is never stored.

372 Algorithm 3 delineates the Cordell's method. Note that a single column  $\mathbf{G}[:, i_{\max}]$  of the  $D \times D$  matrix  $\mathbf{G}$   
 373 (equation 58) is used per iteration, where  $i_{\max}$  is the index of the maximum absolute value in  $\mathbf{r}$ . As pointed  
 374 out by Cordell (1992), the method does not necessarily decrease monotonically along the iterations. Besides,  
 375 the method may not converge depending on how the vertical distances  $\Delta z_i$ ,  $i \in \{1 : D\}$ , controlling the  
 376 depths of the equivalent sources are set. According to Cordell (1992), the maximum absolute value  $r_{\max}$   
 377 in  $\mathbf{r}$  decreases robustly at the beginning and oscillates within a narrowing envelope for the subsequent  
 378 iterations.

379 Guspí and Novara (2009) generalized Cordell's method to perform reduction to the pole and other  
 380 transformations on scattered magnetic observations by using two steps. The first step involves computing  
 381 the vertical component of the observed field using equivalent sources while preserving the magnetization  
 382 direction. In the second step, the vertical observation direction is maintained, but the magnetization  
 383 direction is shifted to the vertical. The main idea employed by both Cordell (1992) and Guspí and Novara  
 384 (2009) is an iterative scheme that uses a single equivalent source positioned below a measurement station  
 385 to compute both the predicted data and residual data for all stations. This approach entails a computational  
 386 strategy where a single column of the sensitivity matrix  $\mathbf{G}$  (equation 58) is calculated per iteration.

### 387 4.3 Row-action update

388 We call the computational strategy *row-action update* because a single row of the sensitivity matrix  $\mathbf{G}$   
 389 (equation 58) is calculated iteratively. Hence, the equivalent-layer solution is updated by processing a  
 390 new datum (one matrix row) at each iteration. To reduce the total processing time and memory usage of  
 391 equivalent-layer technique, Mendonça and Silva (1994) proposed a strategy called *equivalent data concept*.  
 392 The equivalent data concept is grounded on the principle that there is a subset of redundant data that does  
 393 not contribute to the final solution and thus can be dispensed. Conversely, there is a subset of observations,  
 394 called equivalent data, that contributes effectively to the final solution and fits the remaining observations  
 395 (redundant data). Iteratively, Mendonça and Silva (1994) selected the subset of equivalent data that is  
 396 substantially smaller than the original dataset. This selection is carried out by incorporating one data point  
 397 at a time.

**Algorithm 3:** Generic pseudo-code for the method proposed by Cordell (1992).**Initialization :**


---

```

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

```

---

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

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_e \\ \mathbf{d}_r \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{G}_e \\ \mathbf{R}_r \end{bmatrix}, \quad (29)$$

409 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,  
 410 respectively. Mendonça and Silva (1994) designate  $\mathbf{d}_e$  and  $\mathbf{d}_r$  as, respectively, *equivalent* and *redundant*  
 411 data. With the exception of a normalization strategy, Mendonça and Silva (1994) calculate a  $P \times 1$  estimated  
 412 parameter vector  $\tilde{p}$  by solving an underdetermined problem (equation 23) involving only the equivalent  
 413 data  $\mathbf{d}_e$  (equation 29) for the particular case in which  $\mathbf{H} = \mathbf{W}_p = \mathbf{I}_P$  (equations 9 and 13),  $\mathbf{W}_d = \mathbf{I}_{D_e}$   
 414 (equation 12) and  $\bar{p} = 0$  (equation 14), which results in

$$\begin{aligned} (\mathbf{F} + \mu \mathbf{I}_{D_e}) \mathbf{u} &= \mathbf{d}_e \\ \tilde{p} &= \mathbf{G}_e^\top \mathbf{u}, \end{aligned} \quad (30)$$

415 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  
 416 (1994) presume that the estimated parameter vector  $\tilde{p}$  obtained from equation 30 leads to a  $D_r \times 1$  residuals  
 417 vector

$$\mathbf{r} = \mathbf{d}_r - \mathbf{G}_r \tilde{p} \quad (31)$$

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

419 The overall method of Mendonça and Silva (1994) is defined by Algorithm 4. It is important noting  
 420 that the number  $D_e$  of equivalent data in  $\mathbf{d}_e$  increases by one per iteration, which means that the order  
 421 of the linear system in equation 30 also increases by one at each iteration. Those authors also propose a  
 422 computational strategy based on Cholesky factorization (e.g., Golub and Loan, 2013, p. 163) for efficiently  
 423 updating  $(\mathbf{F} + \mu \mathbf{I}_{D_e})$  at a given iteration (line 16 in Algorithm 4) by computing only its new elements with  
 424 respect to those computed in the previous iteration.

---

**Algorithm 4:** Generic pseudo-code for the method proposed by Mendonça and Silva (1994).
 

---

**Initialization :**

```

1 Set a regular grid of  $P$  equivalent sources at a horizontal plane  $z_0$  ;
2 Set a tolerance  $\epsilon$  ;
3 Set a  $D \times 1$  residuals vector  $\mathbf{r} = \mathbf{d}$  ;
4 Define the maximum absolute value  $r_{\max}$  in  $\mathbf{r}$  ;
5 Define the index  $i_{\max}$  of  $r_{\max}$  ;
6 Define the list of indices  $\mathbf{i}_r$  of the remaining data in  $\mathbf{r}$  ;
7 Define  $\mathbf{d}_e = \mathbf{d}[i_{\max}]$  ;
8 Compute  $(\mathbf{F} + \mu \mathbf{I}_{D_e})$  and  $\mathbf{G}_e$  ;
9 Compute  $\tilde{\mathbf{p}}$  (equation 30) ;
10 Compute  $\mathbf{r} = \mathbf{d}[\mathbf{i}_r] - \mathbf{G}[\mathbf{i}_r, :] \tilde{\mathbf{p}}$  ;
11 Define the maximum absolute value  $r_{\max}$  in  $\mathbf{r}$  ;
12 while ( $r_{\max} > \epsilon$ ) do
13   Define the index  $i_{\max}$  of  $r_{\max}$  ;
14   Define the list of indices  $\mathbf{i}_r$  of the remaining elements in  $\mathbf{r}$  ;
15    $\mathbf{d}_e \leftarrow \begin{bmatrix} \mathbf{d}_e \\ \mathbf{d}[i_{\max}] \end{bmatrix}$  ;
16   Update  $(\mathbf{F} + \mu \mathbf{I}_{D_e})$  and  $\mathbf{G}_e$  ;
17   Update  $\tilde{\mathbf{p}}$  (equation 30) ;
18   Update  $\mathbf{r} = \mathbf{d}[\mathbf{i}_r] - \mathbf{G}[\mathbf{i}_r, :] \tilde{\mathbf{p}}$  ;
19   Define the maximum absolute value  $r_{\max}$  in  $\mathbf{r}$  ;
20 end
```

---

#### 425 4.4 Reparameterization

426 Another approach for improving the computational performance of equivalent-layer technique consists  
 427 in setting a  $P \times Q$  reparameterization matrix  $\mathbf{H}$  (equation 9) with  $Q \ll P$ . This strategy has been used  
 428 in applied geophysics for decades (e.g., Skilling and Bryan, 1984; Kennett et al., 1988; Oldenburg et al.,  
 429 1993; Barbosa et al., 1997) and is known as *subspace method*. The main idea relies in reducing the linear  
 430 system dimension from the original  $P$ -space to a lower-dimensional subspace (the  $Q$ -space). An estimate  
 431  $\tilde{\mathbf{q}}$  for the reparameterized parameter vector  $\mathbf{q}$  is obtained in the  $Q$ -space and subsequently used to obtain  
 432 an estimate  $\tilde{\mathbf{p}}$  for the parameter vector  $\mathbf{p}$  (equation 58) in the  $P$ -space by using equation 9. Hence, the key  
 433 aspect of this *reparameterization approach* is solving an appreciably smaller linear inverse problem for  $\tilde{\mathbf{q}}$   
 434 than that for the original parameter vector  $\tilde{\mathbf{p}}$  (equation 58).

435 Oliveira Jr. et al. (2013) have used this approach to describe the physical property distribution on the  
 436 equivalent layer in terms of piecewise bivariate polynomials. Specifically, their method consists in splitting  
 437 a regular grid of equivalent sources into source windows inside which the physical-property distribution  
 438 is described by bivariate polynomial functions. The key aspect of their method relies on the fact that the

439 total number of coefficients required to define the bivariate polynomials is considerably smaller than the  
 440 original number of equivalent sources. Hence, they formulate a linear inverse problem for estimating the  
 441 polynomial coefficients and use them later to compute the physical property distribution on the equivalent  
 442 layer.

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

$$\left( \mathbf{H}^\top \mathbf{G}^\top \mathbf{G} \mathbf{H} + \mu \mathbf{W}_q \right) \tilde{\mathbf{q}} = \mathbf{H}^\top \mathbf{G}^\top \mathbf{d}, \quad (32)$$

446 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  
 447 regularization (e.g., ?, p. 103). Note that, in this case, the prior information is defined in the  $P$ -space for the  
 448 original parameter vector  $\mathbf{p}$  and then transformed to the  $Q$ -space. Another characteristic of their method is  
 449 that it is valid for processing irregularly-spaced data on an undulating surface.

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

455 Before Oliveira Jr. et al. (2013) and ?, ? also proposed a computationally efficient method for equivalent-  
 456 layer technique based on reparameterization. A key difference, however, is that ? did not set a  $P \times Q$   
 457 reparameterization matrix  $\mathbf{H}$  (equation 9) with  $Q \ll P$ . Instead, they used a matrix  $\mathbf{H}$  with  $Q \approx 1.7P$ .  
 458 Their central idea is setting a reparameterization scheme that groups distant equivalent sources into blocks  
 459 by using a bisection process. This scheme leads to a quadtree representation of the physical-property  
 460 distribution on the equivalent layer, so that matrix  $\mathbf{GH}$  (equation 10) is notably sparse. ? explore this  
 461 sparsity in solving the overdetermined problem for  $\tilde{\mathbf{q}}$  (equation 32) via conjugate-gradient method (e.g.,  
 462 Golub and Loan, 2013, sec. 11.3).

#### 463 4.5 Wavelet compression

464 Previously to ?, the idea of transforming the dense matrix  $\mathbf{G}$  (equation 58) into a sparse one has already  
 465 been used in the context of equivalent-layer technique. ? proposed a method that applies the discrete  
 466 wavelet transform to introduce sparsity into the original dense matrix  $\mathbf{G}$ . Those authors approximate a  
 467 planar grid of potential-field data by a regularly-spaced grid of equivalent sources, so that the number of  
 468 data  $D$  and sources  $P$  is the same, i.e.,  $D = P$ . Specifically, ? proposed a method that applies the wavelet  
 469 transform to the original dense matrix  $\mathbf{G}$  and sets to zero the small coefficients that are below a given  
 470 threshold, which results in an approximating sparse representation of  $\mathbf{G}$  in the wavelet domain. They first  
 471 consider the following approximation

$$\mathbf{d}_w \approx \mathbf{G}_s \mathbf{p}_w, \quad (33)$$

472 where

$$\mathbf{d}_w = \mathcal{W} \mathbf{d}, \quad \mathbf{p}_w = \mathcal{W} \mathbf{p}, \quad (34)$$

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

$$\mathbf{G}_w = \mathcal{W} \mathbf{G} \mathcal{W}^\top \quad (35)$$

475 with absolute value smaller than a given threshold.

476 ? solve a normalized inverse problem in the wavelet domain. Specifically, they first define a matrix

$$\mathbf{G}_L = \mathbf{G}_s \mathbf{L}^{-1} \quad (36)$$

477 and a normalized parameter vector

$$\mathbf{p}_L = \mathbf{L} \mathbf{p}_w, \quad (37)$$

478 where  $\mathbf{L}$  is a diagonal and invertible matrix representing an approximation of the first-order Tikhonov  
479 regularization in the wavelet domain. Then they solve an overdetermined problem (equation 22) to obtain  
480 an estimate  $\tilde{\mathbf{p}}_L$  for  $\mathbf{p}_L$  (equation 37), with  $\mathbf{G}_L$  (equation 36),  $\mathbf{H} = \mathbf{I}_P$  (equations 9),  $\mu = 0$  (equation 11),  
481  $\mathbf{W}_d = \mathbf{I}_D$  (equation 12) and  $\bar{p} = 0$  (equation 14) via conjugate-gradient method (e.g., Golub and Loan,  
482 2013, sec. 11.3). Finally, ? compute an estimate  $\tilde{\mathbf{p}}$  for the original parameter vector given by

$$\tilde{\mathbf{p}} = \mathcal{W}^\top (\mathbf{L}^{-1} \tilde{\mathbf{p}}_L), \quad (38)$$

483 where the term within parenthesis is an estimate  $\tilde{\mathbf{p}}_w$  of the parameter vector  $\mathbf{p}_w$  (equation 34) in the wavelet  
484 domain and matrix  $\mathcal{W}^\top$  represents an inverse wavelet transform.

#### 485 4.6 Iterative methods using the full matrix $\mathbf{G}$

486 Xia and Sprowl (1991) introduced an iterative method for estimating the parameter vector  $\tilde{\mathbf{p}}$  (equation  
487 58), which was subsequently adapted to the Fourier domain by Xia et al. (1993). Their method uses the full  
488 and dense sensitivity matrix  $\mathbf{G}$  (equation 58) (without applying any compression or reparameterization, for  
489 example) to compute the predicted data at all observation points per iteration. More than two decades later,  
490 Siqueira et al. (2017) have proposed an iterative method similar to that presented by Xia and Sprowl (1991).  
491 The difference is that Siqueira et al.'s algorithm was deduced from the *Gauss' theorem* (e.g., ?, p. 43) and  
492 the *total excess of mass* (e.g., ?, p. 60). Besides, Siqueira et al. (2017) have included a numerical analysis  
493 showing that their method produces very stable solutions, even for noise-corrupted potential-field data.

494 The iterative method proposed by Siqueira et al. (2017) is outlined in Algorithm 5, presumes an equivalent  
495 layer formed by monopoles (point masses) and can be applied to irregularly-spaced data on an undulating  
496 surface. Note that the residuals  $\mathbf{r}$  are used to compute a correction  $\Delta\mathbf{p}$  for the parameter vector at each  
497 iteration (line 11), which requires a matrix-vector product involving the full matrix  $\mathbf{G}$ . Interestingly, this  
498 approach for estimating the physical property distribution on an equivalent layer is the same originally  
499 proposed by ? for estimating the basement relief under sedimentary basins. The methods of Xia and  
500 Sprowl (1991) and Siqueira et al. (2017) were originally proposed for processing gravity data, but can  
501 be potentially applied to any harmonic function because they actually represent iterative solutions of the  
502 classical *Dirichlet's problem* or the *first boundary value problem of potential theory* (? , p. 236) on a plane.

503 Recently, Jirigalatu and Ebbing (2019) presented another iterative method for estimating a parameter  
504 vector  $\tilde{\mathbf{p}}$  (equation 58). With the purpose of combining different potential-field data, their method basically  
505 modifies that shown in Algorithm 5 by changing the initial approximation and the iterative correction for  
506 the parameter vector. Specifically, Jirigalatu and Ebbing (2019) replace line 4 by  $\tilde{\mathbf{p}} = 0$ , where  $0$  is a vector  
507 of zeros, and line 9 by  $\Delta\mathbf{p} = \omega \mathbf{G}^\top \mathbf{r}$ , where  $\omega$  is a positive scalar defined by trial and error. Note that  
508 this modified approach requires two matrix-vector products involving the full matrix  $\mathbf{G}$  per iteration. To  
509 overcome the high computational cost of these two products, Jirigalatu and Ebbing (2019) set an equivalent

510 layer formed by prisms and compute their predicted potential field in the wavenumber domain by using the  
 511 Gauss-FFT technique ?.

---

**Algorithm 5:** Generic pseudo-code for the iterative method proposed by Siqueira et al. (2017). The symbol “ $\circ$ ” denotes the entrywise or Hadamard product (e.g., ?, p. 298) and  $\sigma$  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  $\gamma$  is the gravitational constant.

**Initialization :**

```

1 Set  $P$  equivalent sources on a horizontal plane  $z_0$  ;
2 Set a tolerance  $\epsilon$  ;
3 Set an auxiliary vector  $\sigma$  ;
4 Compute  $\tilde{\mathbf{p}} = \sigma \circ \mathbf{d}$  ;
5 Compute  $\mathbf{G}$  (equation 58) ;
6 Compute  $\mathbf{r} = \mathbf{d} - \mathbf{G} \tilde{\mathbf{p}}$  ;
7 Compute  $\delta = \|\mathbf{r}\|/D$  ;
8 while ( $\delta > \epsilon$ ) do
9   Compute  $\Delta\mathbf{p} = \sigma \circ \mathbf{r}$  ;
10  Update  $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + \Delta\mathbf{p}$  ;
11  Compute  $\nu = \mathbf{G} \Delta\mathbf{p}$  ;
12  Update  $\mathbf{r} \leftarrow \mathbf{r} - \nu$  ;
13  Compute  $\delta = \|\nu\|/D$  ;
14 end
```

---

512 **4.7 Discrete convolution**

513 Recently, ?? proposed the *convolutional equivalent-layer method*, which explores the structure of the  
 514 sensitivity matrix  $\mathbf{G}$  (equation 58) for the particular case in which (i) there is a single equivalent source  
 515 right below each potential-field datum and (ii) both data and sources rely on planar and regularly spaced  
 516 grids. Specifically, they consider a regular grid of  $D$  potential-field data at points  $(x_i, y_i, z_0)$ ,  $i \in \{1 : D\}$ ,  
 517 on a horizontal plane  $z_0$ . The data indices  $i$  may be ordered along the  $x$ - or  $y$ -direction, which results  
 518 in an  $x$ - or  $y$ -oriented grid, respectively. They also consider a single equivalent source located right  
 519 below each datum, at a constant vertical coordinate  $z_0 + \Delta z$ ,  $\Delta z > 0$ . In this case, the number of data  
 520 and equivalent sources are equal to each other (i.e.,  $D = P$ ) and  $\mathbf{G}$  (equation 58) assumes a *doubly block*  
 521 *Toeplitz* (? , p. 28) or *block-Toeplitz-Block* (BTB) (? , p. 67) structure formed by  $D_B \times D_B$  blocks,  
 522 where each block has  $D_b \times D_b$  elements, with  $D = D_B D_b$ . This particular structure allows formulating  
 523 the product of  $\mathbf{G}$  and an arbitrary vector as a *fast 2D discrete convolution* via *Fast Fourier Transform* (FFT)  
 524 (Van Loan, 1992, section 4.2).

525 Consider, for example, the particular case in which  $D_B = 4$ ,  $D_b = 3$  and  $D = 12$ . In this case,  $\mathbf{G}$   
 526 (equation 58) is a  $12 \times 12$  block matrix given by

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}^0 & \mathbf{G}^1 & \mathbf{G}^2 & \mathbf{G}^3 \\ \mathbf{G}^{-1} & \mathbf{G}^0 & \mathbf{G}^1 & \mathbf{G}^2 \\ \mathbf{G}^{-2} & \mathbf{G}^{-1} & \mathbf{G}^0 & \mathbf{G}^1 \\ \mathbf{G}^{-3} & \mathbf{G}^{-2} & \mathbf{G}^{-1} & \mathbf{G}^0 \end{bmatrix}_{D \times D}, \quad (39)$$

527 where each block  $\mathbf{G}^\ell$ ,  $\ell \in \{(1 - D_B) : (D_B - 1)\}$ , is a  $3 \times 3$  Toeplitz matrix. ?? have deduced the specific  
 528 relationship between blocks  $\mathbf{G}^\ell$  and  $\mathbf{G}^{-\ell}$  and also between a given block  $\mathbf{G}^\ell$  and its transposed  $(\mathbf{G}^\ell)^\top$

529 according to the harmonic function  $g_{ij}$  (equation 2) defining the element  $ij$  of the sensitivity matrix  $\mathbf{G}$   
 530 (equation 58) and the orientation of the data grid.

531 Consider the matrix-vector products

$$\mathbf{G} \mathbf{v} = \mathbf{w} \quad (40)$$

532 and

$$\mathbf{G}^\top \mathbf{v} = \mathbf{w}, \quad (41)$$

533 involving a  $D \times D$  sensitivity matrix  $\mathbf{G}$  (equation 58) defined in terms of a given harmonic function  $g_{ij}$   
 534 (equation 2), where

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}^0 \\ \vdots \\ \mathbf{v}^{D_B-1} \end{bmatrix}_{D \times 1}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}^0 \\ \vdots \\ \mathbf{w}^{D_B-1} \end{bmatrix}_{D \times 1}, \quad (42)$$

535 are arbitrary partitioned vectors formed by  $D_B$  sub-vectors  $\mathbf{v}^\ell$  and  $\mathbf{w}^\ell$ ,  $\ell \in \{0 : (D_B - 1)\}$ , all of them  
 536 having  $D_b$  elements. Equations 40 and 41 can be computed in terms of an auxiliary matrix-vector product

$$\mathbf{G}_c \mathbf{v}_c = \mathbf{w}_c, \quad (43)$$

537 where

$$\mathbf{v}_c = \begin{bmatrix} \mathbf{v}_c^0 \\ \vdots \\ \mathbf{v}_c^{D_B-1} \\ \mathbf{0}_{2D \times 1} \end{bmatrix}_{4D \times 1}, \quad \mathbf{w}_c = \begin{bmatrix} \mathbf{w}_c^0 \\ \vdots \\ \mathbf{w}_c^{D_B-1} \\ \mathbf{0}_{2D \times 1} \end{bmatrix}_{4D \times 1}, \quad (44)$$

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

$$\mathbf{v}_c^\ell = \begin{bmatrix} \mathbf{v}^\ell \\ \mathbf{0}_{D_b \times 1} \end{bmatrix}_{2D_b \times 1}, \quad \mathbf{w}_c^\ell = \begin{bmatrix} \mathbf{w}^\ell \\ \mathbf{0}_{D_b \times 1} \end{bmatrix}_{2D_b \times 1}, \quad (45)$$

539 and  $\mathbf{G}_c$  is a  $4D \times 4D$  *doubly block circulant* (?, p. 28) or *block-circulant circulant-block* (BCCB) (?, p.  
 540 76) matrix. What follows aims at explaining how the original matrix-vector products defined by equations  
 541 40 and 41, involving a  $D \times D$  BTTB matrix  $\mathbf{G}$  exemplified by equation 39, can be efficiently computed in  
 542 terms of the auxiliary matrix-vector product given by equation 43, which has a  $4D \times 4D$  BCCB matrix  $\mathbf{G}_c$ .

543 Matrix  $\mathbf{G}_c$  (equation 43) is formed by  $2D_B \times 2D_B$  blocks, where each block  $\mathbf{G}_c^\ell$ ,  $\ell \in \{(1 - D_B) : (D_B - 1)\}$  is a  $2D_b \times 2D_b$  circulant matrix. For the case in which the original matrix-vector product is that  
 544 defined by equation 40, the first column of blocks forming the BCCB matrix  $\mathbf{G}_c$  is given by  
 545

$$\mathbf{G}_c[:, : 2D_b] = \begin{bmatrix} \mathbf{G}_c^0 \\ \mathbf{G}_c^{-1} \\ \vdots \\ \mathbf{G}_c^{1-D_B} \\ \mathbf{0}_{2D_b \times 2D_b} \\ \mathbf{G}_c^{D_B-1} \\ \vdots \\ \mathbf{G}_c^1 \end{bmatrix}_{4D \times 2D_b}, \quad (46)$$

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

$$\mathbf{G}_c^\ell[:, 1] = \begin{bmatrix} \mathbf{G}^\ell[:, 1] \\ 0 \\ (\mathbf{G}^\ell[1, D_b : 2])^\top \end{bmatrix}_{2D_b \times 2D_b}, \quad \ell \in \{(1 - D_B) : (D_B - 1)\}, \quad (47)$$

547 where  $\mathbf{G}^\ell$  are the blocks forming the BTTB matrix  $\mathbf{G}$  (equation 39). For the case in which the original  
548 matrix-vector product is that defined by equation 41, the first column of blocks forming the BCCB matrix  
549  $\mathbf{G}_c$  is given by

$$\mathbf{G}_c[:, : 2D_b] = \begin{bmatrix} \mathbf{G}_c^0 \\ \mathbf{G}_c^1 \\ \vdots \\ \mathbf{G}_c^{D_B-1} \\ \mathbf{0}_{2D_b \times 2D_b} \\ \mathbf{G}_c^{1-D_B} \\ \vdots \\ \mathbf{G}_c^{-1} \end{bmatrix}_{4D \times 2D_b}, \quad (48)$$

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

$$\mathbf{G}_c^\ell[:, 1] = \begin{bmatrix} (\mathbf{G}^\ell[1, :])^\top \\ 0 \\ \mathbf{G}^\ell[D_b : 2, 1] \end{bmatrix}_{2D_b \times 2D_b}, \quad \ell \in \{(1 - D_B) : (D_B - 1)\}. \quad (49)$$

551 The complete matrix  $\mathbf{G}_c$  (equation 43) is obtained by properly downshifting the block columns  $\mathbf{G}_c[:, : 2D_b]$  defined by equations 46 or 48. Similarly, the  $\ell$ -th block  $\mathbf{G}_c^\ell$  of  $\mathbf{G}_c$  is obtained by properly downshifting  
552 the first columns  $\mathbf{G}_c^\ell[:, 1]$  defined by equations 47 or 49.

554 Note that  $\mathbf{G}_c$  (equation 43) is a  $4D \times 4D$  matrix and  $\mathbf{G}$  (equation 39) is a  $D \times D$  matrix. It seems weird  
555 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  
556 to use the fact that BCCB matrices are diagonalized by the 2D unitary discrete Fourier transform (DFT)  
557 (e.g., Davis, 1979, p. 31). Because of that,  $\mathbf{G}_c$  can be written as

$$\mathbf{G}_c = (\mathcal{F}_{2D_B} \otimes \mathcal{F}_{2D_b})^* \Lambda (\mathcal{F}_{2D_B} \otimes \mathcal{F}_{2D_b}), \quad (50)$$

558 where the symbol “ $\otimes$ ” denotes the Kronecker product (e.g., ?, p. 243),  $\mathcal{F}_{2D_B}$  and  $\mathcal{F}_{2D_b}$  are the  $2D_B \times 2D_B$   
559 and  $2D_b \times 2D_b$  unitary DFT matrices (e.g., Davis, 1979, p. 31), respectively, the superscript “\*” denotes  
560 the complex conjugate and  $\Lambda$  is a  $4D \times 4D$  diagonal matrix containing the eigenvalues of  $\mathbf{G}_c$ . Due to the  
561 diagonalization of the matrix  $\mathbf{G}_c$ , equation 43 can be rewritten by using equation 50 and premultiplying  
562 both sides of the result by  $(\mathcal{F}_{2D_B} \otimes \mathcal{F}_{2D_b})$ , i.e.,

$$\Lambda (\mathcal{F}_{2D_B} \otimes \mathcal{F}_{2D_b}) \mathbf{v}_c = (\mathcal{F}_{2D_B} \otimes \mathcal{F}_{2D_b}) \mathbf{w}_c. \quad (51)$$

563 By following ?, we rearrange equation 51 as follows

$$\mathcal{L} \circ (\mathcal{F}_{2D_B} \mathcal{V}_c \mathcal{F}_{2D_b}) = \mathcal{F}_{2D_B} \mathcal{W}_c \mathcal{F}_{2D_b} \quad (52)$$

564 where “ $\circ$ ” denotes the Hadamard product (e.g., ?, p. 298) and  $\mathcal{L}$ ,  $\mathcal{V}_c$  and  $\mathcal{W}_c$  are  $2D_B \times 2D_b$  matrices  
 565 obtained by rearranging, along their rows, the elements forming the diagonal of  $\Lambda$  (equation 50), vector  $\mathbf{v}_c$   
 566 and vector  $\mathbf{w}_c$  (equation 44), respectively. Then, by premultiplying both sides of equation 52 by  $\mathcal{F}_{2D_B}^*$  and  
 567 then postmultiplying both sides by  $\mathcal{F}_{2D_b}^*$ , we obtain

$$\mathcal{F}_{2D_B}^* [\mathcal{L} \circ (\mathcal{F}_{2D_B} \mathcal{V}_c \mathcal{F}_{2D_b})] \mathcal{F}_{2D_b}^* = \mathcal{W}_c. \quad (53)$$

568 Finally, we get from equation 50 that matrix  $\mathcal{L}$  can be computed by using only the first column  $\mathbf{G}_c[:, 1]$  of  
 569 the BCCB matrix  $\mathbf{G}_c$  (equation 43) according to (?)

$$\mathcal{L} = \sqrt{4D} \mathcal{F}_{2D_B} \mathcal{C} \mathcal{F}_{2D_b}, \quad (54)$$

570 where  $\mathcal{C}$  is a  $2D_B \times 2D_b$  matrix obtained by rearranging, along its rows, the elements of  $\mathbf{G}_c[:, 1]$  (equation  
 571 43).

572 The whole procedure to compute the original matrix-vector products  $\mathbf{G}\mathbf{v}$  (equation 40) and  $\mathbf{G}^\top\mathbf{v}$   
 573 (equation 41) consists in (i) rearranging the elements of the vector  $\mathbf{v}$  and the first column  $\mathbf{G}[:, 1]$  of matrix  
 574  $\mathbf{G}$  into the matrices  $\mathcal{V}_c$  and  $\mathcal{C}$  (equations 53 and 54), respectively; (ii) computing terms  $\mathcal{F}_{2D_B} \mathcal{A} \mathcal{F}_{2D_b}$  and  
 575  $\mathcal{F}_{2D_B}^* \mathcal{A} \mathcal{F}_{2D_b}^*$ , where  $\mathcal{A}$  is a given matrix, and a Hadamard product to obtain  $\mathcal{W}_c$  (equation 53); and (iii)  
 576 retrieve the elements of vector  $\mathbf{w}$  (equation 40) from  $\mathcal{W}_c$  (equation 53). It is important noting that the steps  
 577 (i) and (iii) do not have any computational cost because they involve only reorganizing elements of vectors  
 578 and matrices. Besides, the terms  $\mathcal{F}_{2D_B} \mathcal{A} \mathcal{F}_{2D_b}$  and  $\mathcal{F}_{2D_B}^* \mathcal{A} \mathcal{F}_{2D_b}^*$  in step (ii) represent, respectively, the  
 579 2D Discrete Fourier Transform (2D-DFT) and the 2D Inverse Discrete Fourier Transform (2D-IDFT) of  $\mathcal{A}$ .  
 580 These transforms can be efficiently computed by using the 2D Fast Fourier Transform (2D-FFT). Hence,  
 581 the original matrix-vector products  $\mathbf{G}\mathbf{v}$  (equation 40) and  $\mathbf{G}^\top\mathbf{v}$  (equation 41) can be efficiently computed  
 582 by using the 2D-FFT.

583 PAREI AQUI: É NECESSÁRIO FALAR QUE A PARTIR DE  $\mathbf{G}[:, 1]$  É POSSÍVEL OBTER  $\mathbf{G}^T[:, 1]$   
 584 SABENDO-SE O TIPO DE SIMETRIA DA ESTRUTURA BTTB. ISSO POSSIBILITA USAR  $\mathbf{G}[:, 1]$   
 585 DEFINIR A  $\mathbf{G}_c[:, 1]$  ASSOCIADA AO PRODUTO  $\mathbf{G}\mathbf{v} = \mathbf{w}$  OU AO PRODUTO  $\mathbf{G}^\top\mathbf{v} = \mathbf{w}$

## 5 SOLUTION STABILITY

586 The solution stability of the equivalent-layer methods is rarely addressed. Here, we follow the numerical  
 587 stability analysis presented in Siqueira et al. (2017).

588 Let us assume noise-free potential-field data  $\mathbf{d}$ , we estimate a physical-property distribution  $\mathbf{p}$  (estimated  
 589 solution) within the equivalent layer. Then, the noise-free data  $\mathbf{d}$  are contaminated with additive  $D$  different  
 590 sequences of pseudorandom Gaussian noise, creating different noise-corrupted potential-field data  $\mathbf{d}_\ell^o$ ,  
 591  $\ell = 1, \dots, D$ . From each  $\mathbf{d}_\ell^o$ , we estimate a physical-property distribution  $\hat{\mathbf{p}}_\ell$  within the equivalent layer.

592 Next, for each noise-corrupted data  $\mathbf{d}_\ell^o$  and estimated solution  $\hat{\mathbf{p}}_\ell$ , the  $\ell$ th model perturbation  $\delta p_\ell$  and the  
 593  $\ell$ th data perturbation  $\delta d_\ell$  are, respectively, evaluated by

$$\delta p_\ell = \frac{\|\hat{\mathbf{p}}_\ell - \mathbf{p}\|_2}{\|\mathbf{p}\|_2}, \quad \ell = 1, \dots, D, \quad (55)$$

594 and

$$\delta d_\ell = \frac{\|\mathbf{d}_\ell^o - \mathbf{d}\|_2}{\|\mathbf{d}\|_2}, \quad \ell = 1, \dots, D. \quad (56)$$

**Algorithm 6:** Generic pseudo-code for the convolutional equivalent-layer method proposed by ??.**Initialization :**


---

```

1 Set the regular grid of  $P$  equivalent sources on a horizontal plane  $z_0$  ;
2 Set a tolerance  $\epsilon$  and a maximum number of iterations ITMAX ;
3 Compute the first column  $\mathbf{G}[:, 1]$  of the sensitivity matrix  $\mathbf{G}$  (equation 58) for the particular case in
   which it has a BTTB structure (equation 39);
4 Rearrange the elements of  $\mathbf{G}[:, 1]$  into the first column  $\mathbf{G}_c[:, 1]$  of the BCCB matrix  $\mathbf{G}_c$  (equation 43);
5 Rearrange  $\mathbf{G}_c[:, 1]$  to obtain  $\mathcal{C}$  (equation 54);
6 Compute  $\mathcal{F}_{2D_B} \mathcal{C} \mathcal{F}_{2D_b}$  by using a 2D Fast Fourier Transform (2D-FFT) and multiply by  $\sqrt{4D}$  to
   obtain  $\mathcal{L}$  (equation 54);
7 Set  $\tilde{\mathbf{p}} = \mathbf{0}$  ;
8 Set  $\mathbf{r} = \mathbf{d}$  and compute  $\delta = \|\mathbf{r}\|/D$  ;
9 Compute  $\boldsymbol{\vartheta} = \mathbf{G}^\top \mathbf{r}$  (Algorithm 7) and  $\rho_0 = \boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$  ;
10 Set  $\tau = 0$  and  $\boldsymbol{\eta} = \mathbf{0}$  ;
11  $m = 1$  ;
12 while ( $\delta > \epsilon$ ) and ( $m < \text{ITMAX}$ ) do
13   Update  $\boldsymbol{\eta} \leftarrow \boldsymbol{\vartheta} + \tau \boldsymbol{\eta}$  ;
14   Compute  $\boldsymbol{\nu} = \mathbf{G} \boldsymbol{\eta}$  (Algorithm 7);
15   Compute  $v = \rho_0 / (\boldsymbol{\nu}^\top \boldsymbol{\nu})$  ;
16   Update  $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + v \boldsymbol{\eta}$  ;
17   Update  $\mathbf{r} \leftarrow \mathbf{r} - v \boldsymbol{\nu}$  and compute  $\delta \leftarrow \|v \boldsymbol{\nu}\|/D$  ;
18   Compute  $\boldsymbol{\vartheta} = \mathbf{G}^\top \mathbf{r}$  (Algorithm 7) and  $\rho = \boldsymbol{\vartheta}^\top \boldsymbol{\vartheta}$  ;
19   Compute  $\tau = \rho / \rho_0$  ;
20   Update  $\rho_0 \leftarrow \rho$  ;
21    $m \leftarrow m + 1$  ;
22 end

```

---

**Algorithm 7:** Pseudo-code for computing the generic matrix-vector product given by equation 40 via fast 2D discrete convolution.

---

```

1 Rearrange  $\mathbf{v}$  (equations 40 and 42) into  $\mathbf{v}_c$  (equations 43–45);
2 Rearrange  $\mathbf{v}_c$  into  $\mathcal{V}_c$ , compute  $\mathcal{F}_{2D_B} \mathcal{V}_c \mathcal{F}_{2D_b}$  via 2D-FFT and evaluate its Hadamard product with
   matrix  $\mathcal{L}$  to define an auxiliary matrix  $\mathcal{A} = \mathcal{L} \circ (\mathcal{F}_{2D_B} \mathcal{V}_c \mathcal{F}_{2D_b})$ ;
3 Compute  $\mathcal{F}_{2D_B}^* \mathcal{A} \mathcal{F}_{2D_b}^*$  via 2D-FFT inverse to evaluate  $\mathcal{W}_c$  (equation 53);
4 Retrieve  $\mathbf{w}_c$  (equations 43–45) from  $\mathcal{W}_c$  (equation 53);
5 Retrieve  $\mathbf{w}$  (equations 40 and 42) from  $\mathbf{w}_c$  (equations 43–45);

```

---

595    Regardless of the particular method used, the following inequality (Aster et al., 2018, p. 66) is applicable:

$$\delta p_\ell \leq \kappa \delta d_\ell, \quad \ell = 1, \dots, D, \quad (57)$$

596    where  $\kappa$  is the constant of proportionality between the model perturbation  $\delta p_\ell$  (equation 84) and the data  
 597    perturbation  $\delta d_\ell$  (equation 85). The constant  $\kappa$  acts as the condition number of an invertible matrix in a  
 598    given inversion, and thus measures the instability of the solution. The larger (smaller) the value of  $\kappa$  the  
 599    more unstable (stable) is the estimated solution.

600    Equation 86 shows a linear relationship between the model perturbation and the data perturbation. By  
 601    plotting  $\delta p_\ell$  (equation 84) against  $\delta d_\ell$  (equation 85) produced by a set of  $D$  estimated solution obtained by  
 602    applying a given equivalent-layer method, we obtain a straight line behaviour described by equation 86.

603 By applying a linear regression, we obtain a fitted straight line whose estimated slope ( $\kappa$  in equation 86)  
 604 quantifies the solution stability.

605 Here, the analysis of solution stability is numerically conducted by applying the classical equivalent-  
 606 layer technique with zeroth-order Tikhonov regularization, the convolutional method for gravimetric and  
 607 magnetic data, the deconvolutional method (equation 82) and the deconvolutional method with different  
 608 values for the Wiener stabilization (equation 83).

## 6 THE EQUIVALENT-LAYER TECHNIQUE

### 609 6.1 Fundamentals

610 Consider a set of  $N$  potential-field observations (gravity or magnetic data)  $d_i^o (x_i, y_i, z_i)$ ,  $i = 1, \dots, N$ ,  
 611 at the  $i$ th observation point  $(x_i, y_i, z_i)$  of a Cartesian coordinate system with  $x$ -,  $y$ - and  $z$ -axis pointing to  
 612 north, east and down, respectively. Physically, the discrete set of potential-field observations is produced by  
 613 an unknown source distribution in the subsurface. Mathematically, it represents a discrete set of a harmonic  
 614 function.

615 A standard way to deal with the classical equivalent-layer technique is approximate the observed potential-  
 616 field data by the predicted data, which in turn are produced by a fictitious layer of sources, called equivalent  
 617 layer. The equivalent layer is located below the observation surface, at depth  $z_0$  ( $z_0 > z_i$ ), and with finite  
 618 horizontal dimensions being composed by a finite discrete set of equivalent sources (e.g., point masses,  
 619 dipoles, or prisms). Mathematically, this approximation can be written in matrix notation as

$$\mathbf{d} = \mathbf{A}\mathbf{p}, \quad (58)$$

620 where  $\mathbf{d}$  is an  $N$ -dimensional predicted data vector whose  $i$ th element,  $d_i (x_i, y_i, z_i)$ ,  $i = 1, \dots, N$ , is the  
 621 predicted potential-field observation,  $\mathbf{p}$  is an  $M$ -dimensional parameter vector whose  $j$ th element  $p_j$  can be  
 622 a physical property of the  $j$ th equivalent source and  $\mathbf{A}$  is the  $N \times M$  sensitivity matrix whose  $ij$ th element  
 623  $a_{ij}$  is a harmonic function.

### 624 6.2 Computational strategies

625 The classical equivalent-layer technique consists of estimating the parameter vector  $\mathbf{p}$  from the  $N$ -  
 626 dimensional observed data vector  $\mathbf{d}^o$  whose  $i$ th element is defined as the  $d_i^o (x_i, y_i, z_i)$ ,  $i = 1, \dots, N$ .  
 627 Usually, this estimate can be obtained by a regularized least-squares solution. The estimated parameter  
 628 is stable, fits the observed data and can be used to yield a desired linear transformation of the data, such  
 629 as interpolation, upward (or downward) continuation, reduction to the pole, joint processing of gravity  
 630 gradient data and more. Mathematically, the desired linear transformation of the data can be obtained by

$$\hat{\mathbf{t}} = \mathbf{T}\mathbf{p}^*, \quad (59)$$

631 where  $\hat{\mathbf{t}}$  is an  $N$ -dimensional transformed data vector,  $\mathbf{p}^*$  is an  $M$ -dimensional estimated parameter vector  
 632 and  $\mathbf{T}$  is the  $N \times M$  matrix of Green's functions whose  $ij$ th element is the transformed field at the  $i$ th  
 633 observation point produced by the  $j$ th equivalent source.

634 The biggest hurdle to use the classical equivalent-layer technique is the computational complexity to  
 635 handle large datasets because the sensitivity matrix  $\mathbf{A}$  (equation 58) is dense. Usually, the estimated  
 636 parameter vector  $\mathbf{p}^*$  requires to solve a large-scale linear inversion which in turn means to deal with

637 some obstacles concerning large computational cost: i) the large computer memory to store large and full  
 638 matrices; ii) the long computation time to multiply a matrix by a vector; and iii) the long computation time  
 639 to solve a large linear system of equations.

640 Here, we review some strategies for reducing the computational cost of equivalent-layer technique. These  
 641 strategies are the following:

#### 642 6.2.1 The moving data-window scheme

643 Leão and Silva (1989) reduced the total processing time and memory usage of equivalent-layer technique  
 644 by means of a moving data-window scheme. A small moving data window with  $N_w$  observations and  
 645 a small equivalent layer with  $M_w$  equivalent sources ( $M_w > N_w$ ) located below the observations are  
 646 established. For each position of a moving-data window, Leão and Silva (1989) estimate a stable solution  
 647  $\mathbf{p}_w^*$  by using a data-space approach with the zeroth-order Tikhonov regularization (Aster et al., 2018), i.e.,

$$(A_w A_w^\top + \mu I) w = d_w^o, \quad (60a)$$

$$A_w^\top w = p_w^*, \quad (60b)$$

648 where  $w$  is a dummy vector,  $\mu$  is a regularizing parameter,  $d_w^o$  is an  $N_w$ -dimensional vector containing  
 649 the observed potential-field data,  $A_w$  is an  $N_w \times M_w$  sensitivity matrix related to a moving-data window,  $I$   
 650 is an identity matrix of order  $N_w$  and the superscript  $\top$  stands for a transpose. After estimating an  $M_w \times 1$   
 651 parameter vector  $p_w^*$  (equation 60b) the desired transformation of the data is only calculated at the central  
 652 point of each moving-data window, i.e.:

$$\hat{t}_k = t_k^\top p_w^*, \quad (61)$$

653 where  $\hat{t}_k$  is the transformed data calculated at the central point  $k$  of the data window and  $t_k$  is an  $M_w \times 1$   
 654 vector whose elements form the  $k$ th row of the  $N_w \times N_w$  matrix of Green's functions  $T$  (equation 59) of  
 655 the desired linear transformation of the data.

656 By shifting the moving-data window with a shift size of one data spacing, a new position of a data  
 657 window is set up. Next, the aforementioned process (equations 60b and 61) is repeated for each position of  
 658 a moving-data window, until the entire data have been processed. Hence, instead of solving a large inverse  
 659 problem, Leão and Silva (1989) solve several much smaller ones.

660 To reduce the size of the linear system to be solved, Soler and Uieda (2021) adopted the same strategy  
 661 proposed, originally, by Leão and Silva (1989) of using a small moving-data window sweeping the whole  
 662 data. In Leão and Silva (1989), a moving-data window slides to the next adjacent data window following a  
 663 sequential movement, the predicted data is calculated inside the data window and the desired transformation  
 664 are only calculated at the center of the moving-data window. Unlike Leão and Silva (1989), Soler and  
 665 Uieda (2021) do not adopt a sequential order of the data windows; rather, they adopt a randomized  
 666 order of windows in the iterations of the gradient-boosting algorithm (Friedman, 2001 and 2002). The  
 667 gradient-boosting algorithm in Soler and Uieda (2021) estimates a stable solution using the data and the  
 668 equivalent sources that fall within a moving-data window; however, it calculates the predicted data and the  
 669 residual data in the whole survey data. Next, the residual data that fall within a new position of the data  
 670 window is used as input data to estimate a new stable solution within the data window which in turn is  
 671 used to calculate a new predicted data and a new residual data in the whole survey data. Finally, unlike  
 672 Leão and Silva (1989), in Soler and Uieda (2021) neither the data nor the equivalent sources need to be  
 673 distributed in regular grids. Indeed, Leão and Silva (1989) built their method using regular grids, but in fact

regular grids are not necessary. Regarding the equivalent-source layout, Soler and Uieda (2021) proposed the block-averaged sources locations in which the survey area is divided into horizontal blocks and one single equivalent source is assigned to each block. Each single source per block is placed over the layer with its horizontal coordinates given by the average horizontal positions of observation points. According to Soler and Uieda (2021), the block-averaged sources layout reduces the number of equivalent sources significantly and the gradient-boosting algorithm provides even greater efficiency in terms of data fitting.

## 680 6.2.2 The equivalent-data concept

To reduced the total processing time and memory usage of equivalent-layer technique, Mendonça and Silva (1994) proposed a strategy called 'equivalent data concept'. The equivalent data concept is grounded on the principle that there is a subset of redundant data that does not contribute to the final solution and thus can be dispensed. Conversely, there is a subset of observations, called equivalent data, that contributes effectively to the final solution and fits the remaining observations (redundant data). Iteratively, Mendonça and Silva (1994) selected the subset of equivalent data that is substantially smaller than the original dataset. This selection is carried out by incorporating one data point at a time.

According to Mendonça and Silva (1994), the number of equivalent data is about one-tenth of the total number of observations. These authors used the equivalent data concept to carry out an interpolation of gravity data. They showed a reduction of the total processing time and memory usage by, at least, two orders of magnitude as opposed to using all observations in the interpolation process via the classical equivalent-layer technique.

## 693 6.2.3 The wavelet compression and lower-dimensional subspace

For large data sets, the sensitivity matrix  $\mathbf{A}$  (equation 58) is a drawback in applying the equivalent-layer technique because it is a large and dense matrix.

Li and Oldenburg (2010) transformed a large and full sensitivity matrix into a sparse one by using fast wavelet transforms. In the wavelet domain, Li and Oldenburg (2010) applyied a 2D wavelet transform to each row and column of the original sensitivity matrix  $\mathbf{A}$  to expand it in the wavelet bases. This operation can be done by premultiplying the original sensitivity matrix  $\mathbf{A}$  by a matrix representing the 2D wavelet transform  $\mathbf{W}_2$  and then the resulting is postmultiplied by the transpose of  $\mathbf{W}_2$  (i.e.,  $\mathbf{W}_2^\top$ ).

$$\tilde{\mathbf{A}} = \mathbf{W}_2 \mathbf{A} \mathbf{W}_2^\top, \quad (62)$$

where  $\tilde{\mathbf{A}}$  is the expanded original sensitivity matrix in the wavelet bases with many elements zero or close to zero. Next, the matrix  $\tilde{\mathbf{A}}$  is replaced by its sparse version  $\tilde{\mathbf{A}}_s$  in the wavelet domain which in turn is obtained by retaining only the large elements of the  $\tilde{\mathbf{A}}$ . Thus, the elements of  $\tilde{\mathbf{A}}$  whose amplitudes fall below a relative threshold are discarded. In Li and Oldenburg (2010), the original sensitivity matrix  $\mathbf{A}$  is high compressed resulting in a sparce matrix  $\tilde{\mathbf{A}}_s$  with a few percent of nonzero elements and the the inverse problem is solved in the wavelet domain by using  $\tilde{\mathbf{A}}_s$  and a incomplete conjugate gradient least squares, without an explicit regularization parameter and a limited number of iterations. The solution is obtained by solving the following linear system

$$\tilde{\mathbf{A}}_L^\top \tilde{\mathbf{A}}_L \tilde{\mathbf{p}}_L^* = \tilde{\mathbf{A}}_L^\top \tilde{\mathbf{d}}^o, \quad (63)$$

709 where  $\tilde{\mathbf{p}}_L^*$  is obtained by solving the linear system given by equation 63,

$$\tilde{\mathbf{A}}_L = \tilde{\mathbf{A}}_s \tilde{\mathbf{L}}^{-1}, \quad (64a)$$

$$\tilde{\mathbf{p}}_L = \tilde{\mathbf{L}} \tilde{\mathbf{p}}, \quad (64b)$$

$$\tilde{\mathbf{d}}^o = \mathbf{W}_2 \mathbf{d}^o, \quad (64c)$$

710 where  $\tilde{\mathbf{L}}$  is a diagonal and invertible weighting matrix representing the finite-difference approximation in  
711 the wavelet domain. Finally, the distribution over the equivalent layer in the space domain  $\mathbf{p}$  is obtained by  
712 applying an inverse wavelet transform in two steps, i.e.:

$$\tilde{\mathbf{p}} = \tilde{\mathbf{L}}^{-1} \tilde{\mathbf{p}}_L^*, \quad (65)$$

713 and

$$\mathbf{p} = \mathbf{W}_2 \tilde{\mathbf{p}}. \quad (66)$$

714 Although the data misfit quantifying the difference between the observed and predicted data by the  
715 equivalent source is calculated in the wavelet domain, we understand that the desired transformation is  
716 calculated via equation 59 which uses a full matrix of Green's functions  $\mathbf{T}$ .

717 Li and Oldenburg (2010) used the equivalent-layer technique with a wavelet compression to perform an  
718 upward continuation of total-field anomaly between uneven surfaces. For regularly spaced grid of data, Li  
719 and Oldenburg (2010) reported that high compression ratios are achieved with insignificant loss of accuracy.  
720 As compared to the upward-continued total-field anomaly by equivalent layer using the dense matrix, Li  
721 and Oldenburg's (2010) approach, using the Daubechies wavelet, decreased CPU (central processing unit)  
722 time by up to two orders of magnitude.

723 Mendonça (2020) overcame the solution of intractable large-scale equivalent-layer problem by using the  
724 subspace method (e.g., Skilling and Bryan, 1984; Kennett et al., 1988; Oldenburg et al., 1993; Barbosa  
725 et al., 1997). The subspace method reduces the dimension of the linear system of equations to be solved.  
726 Given a higher-dimensional space (e.g.,  $M$ -dimensional model space,  $\mathbb{R}^M$ ), there exists many lower-  
727 dimensional subspaces (e.g.,  $Q$ -dimensional subspace) of  $\mathbb{R}^M$ . The linear inverse problem related to the  
728 equivalent-layer technique consists in finding an  $M$ -dimension parameter vector  $\mathbf{p} \in \mathbb{R}^M$  which adequately  
729 fits the potential-field data. The subspace method looks for a parameter vector who lies in a  $Q$ -dimensional  
730 subspace of  $\mathbb{R}^M$  which, in turn, is spanned by a set of  $Q$  vectors  $\mathbf{v}_i = 1, \dots, Q$ , where  $\mathbf{v}_i \in \mathbb{R}^M$ . In matrix  
731 notation, the parameter vector in the subspace method can be written as

$$\mathbf{p} = \mathbf{V} \boldsymbol{\alpha}, \quad (67)$$

732 where  $\mathbf{V}$  is an  $M \times Q$  matrix whose columns  $\mathbf{v}_i = 1, \dots, Q$  form a basis vectors for a subspace  $Q$  of  $\mathbb{R}^M$ .  
733 In equation 67, the parameter vector  $\mathbf{p}$  is defined as a linear combination in the space spanned by  $Q$  basis  
734 vectors  $\mathbf{v}_i = 1, \dots, Q$  and  $\boldsymbol{\alpha}$  is a  $Q$ -dimensional unknown vector to be determined. The main advantage of  
735 the subspace method is that the linear system of  $M$  equations in  $M$  unknowns to be originally solved is  
736 reduced to a new linear system of  $Q$  equations in  $Q$  unknowns which requires much less computational  
737 effort since  $Q \ll M$ , i.e.:

$$\mathbf{V}^\top \mathbf{A}^\top \mathbf{A} \mathbf{V} \boldsymbol{\alpha}^* = \mathbf{V}^\top \mathbf{d}^o. \quad (68)$$

738 To avoid the storage of matrices  $\mathbf{A}$  and  $\mathbf{V}$ , Mendonça (2020) evaluates an element of the matrix  $\mathbf{AV}$  by  
739 calculating the dot product between the row of matrix  $\mathbf{A}$  and the column of the matrix  $\mathbf{B}$ . After estimating

740  $\alpha^*$  (equation 68) belonging to a  $Q$ -dimensional subspace of  $\mathbb{R}^M$ , the distribution over the equivalent layer  
 741  $\mathbf{p}$  in the  $\mathbb{R}^M$  is obtained by applying equation 67. The choice of the  $Q$  basis vectors  $\mathbf{v}_i = 1, \dots, Q$  (equation  
 742 67) in the subspace method is not strict. Mendonça (2020), for example, chose the eigenvectors yielded by  
 743 applying the singular value decomposition of the matrix containing the gridded data set. The number of  
 744 eigenvectors used to form basis vectors will depend on the singular values.

745 The proposed subspace method for solving large-scale equivalent-layer problem by Mendonça (2020)  
 746 was applied to estimate the mass excess or deficiency caused by causative gravity sources.

#### 747 6.2.4 The quadtree discretization

748 To make the equivalent-layer technique tractable, Barnes and Lumley (2011) also transformed the dense  
 749 sensitivity matrix  $\mathbf{A}$  (equation 58) into a sparse matrix. In Barnes and Lumley (2011), a sparse version of  
 750 the sensitivity matrix is achieved by grouping equivalent sources (e.g., they used prisms) distant from an  
 751 observation point together to form a larger prism or larger block. Each larger block has averaged physical  
 752 properties and averaged top- and bottom-surfaces of the grouped smaller prisms (equivalent sources) that  
 753 are encompassed by the larger block. The authors called it the 'larger averaged block' and the essence of  
 754 their method is the reduction in the number of equivalent sources, which means a reduction in the number  
 755 of parameters to be estimated implying in model dimension reduction.

756 The key of the Barnes and Lumley's (2011) method is the algorithm for deciding how to group the smaller  
 757 prisms. In practice, these authors used a recursive bisection process that results in a quadtree discretization  
 758 of the equivalent-layer model.

759 By using the quadtree discretization, Barnes and Lumley (2011) were able to jointly process multiple  
 760 components of airborne gravity-gradient data using a single layer of equivalent sources. To our knowledge,  
 761 Barnes and Lumley (2011) are the pioneers on processing full-tensor gravity-gradient data jointly. In  
 762 addition to computational feasibility, Barnes and Lumley's (2011) method reduces low-frequency noise  
 763 and can also remove the drift in time-domain from the survey data. Those authors stressed that the  
 764  $G_{zz}$ -component calculated through the single estimated equivalent-layer model projected on a grid at a  
 765 constant elevation by inverting full gravity-gradient data has the low-frequency error reduced by a factor of  
 766 2.4 as compared to the inversion of an individual component of the gravity-gradient data.

#### 767 6.2.5 The reparametrization of the equivalent layer

768 Oliveira Jr. et al. (2013) reparametrized the whole equivalent-layer model by a piecewise bivariate-  
 769 polynomial function defined on a set of  $Q$  equivalent-source windows. In Oliveira Jr. et al.'s (2013)  
 770 approach, named polynomial equivalent layer (PEL), the parameter vector within the  $k$ th equivalent-source  
 771 window  $\mathbf{p}^k$  can be written in matrix notation as

$$\mathbf{p}^k = \mathbf{B}^k \mathbf{c}^k, \quad k = 1 \dots Q, \quad (69)$$

772 where  $\mathbf{p}^k$  is an  $M_w$ -dimensional vector containing the physical-property distribution within the  $k$ th  
 773 equivalent-source window,  $\mathbf{c}^k$  is a  $P$ -dimensional vector whose  $l$ th element is the  $l$ th coefficient of the  
 774  $\alpha$ th-order polynomial function and  $\mathbf{B}^k$  is an  $M_w \times P$  matrix containing the first-order derivative of the  
 775  $\alpha$ th-order polynomial function with respect to one of the  $P$  coefficients.

776 By using a regularized potential-field inversion, Oliveira Jr. et al. (2013) estimates the polynomial  
 777 coefficients for each equivalent-source window by solving the following linear system

$$\left( \mathbf{B}^\top \mathbf{A}^\top \mathbf{A} \mathbf{B} + \mu \mathbf{I} \right) \mathbf{c}^* = \mathbf{B}^\top \mathbf{A}^\top \mathbf{d}^o, \quad (70)$$

778 where  $\mu$  is a regularizing parameter,  $\mathbf{c}^*$  is an estimated  $H$ -dimensional vector containing all coefficients  
 779 describing all polynomial functions within all equivalent-source windows which compose the entire  
 780 equivalent layer,  $\mathbf{I}$  is an identity matrix of order  $H$  ( $H = P\dot{Q}$ ) and  $\mathbf{B}$  is an  $M \times H$  block diagonal matrix  
 781 such that the main-diagonal blocks are  $\mathbf{B}^k$  matrices (equation 69) and all off-diagonal blocks are zero  
 782 matrices. For ease of the explanation of equation 70, we keep only the zeroth-order Tikhonov regularization  
 783 and omitting the first-order Tikhonov regularization (Aster et al., 2018) which was also used by Oliveira Jr.  
 784 et al. (2013).

785 The main advantage of the PEL is solve  $H$ -dimensional system of equations (equation 70), where  $H$   
 786 totalizes the number of polynomial coefficients composing all equivalent-source windows, requiring a  
 787 lower computational effort since  $H \ll N$ . To avoid the storage of matrices  $\mathbf{A}$  and  $\mathbf{B}$ , Oliveira Jr. et al.  
 788 (2013) evaluate an element of the matrix  $\mathbf{AB}$  by calculating the dot product between the row of matrix  $\mathbf{A}$   
 789 and the column of the matrix  $\mathbf{B}$ . After estimating all polynomial coefficients of all windows, the estimated  
 790 coefficients ( $\mathbf{c}^*$  in equation 70) are transformed into a single physical-property distribution encompassing  
 791 the entire equivalent layer.

792 As stated by Oliveira Jr. et al. (2013), the computational efficiency of PEL approach stems from the fact  
 793 that the total number of polynomial coefficients  $H$  required to depict the physical-property distribution  
 794 within the equivalent layer is generally much smaller than the number of equivalent sources. Consequently,  
 795 this leads to a considerably smaller linear system that needs to be solved. Hence, the main strategy of  
 796 polynomial equivalent layer is the model dimension reduction.

797 The polynomial equivalent layer was applied to perform upward continuations of gravity and magnetic  
 798 data and reduction to the pole of magnetic data.

#### 799 6.2.6 The iterative scheme without solving a linear system

800 There exists a class of methods that iteratively estimate the distribution of physical properties within an  
 801 equivalent layer without the need to solve linear systems. The method initially introduced by Cordell (1992)  
 802 and later expanded upon by Guspi and Novara (2009) updates the physical property of sources, located  
 803 beneath each potential-field data, by removing the maximum residual between the observed and fitted data.  
 804 In addition, Xia and Sprowl (1991) and Xia et al. (1993) have developed efficient iterative algorithms for  
 805 updating the distribution of physical properties within the equivalent layer in the wavenumber and space  
 806 domains, respectively. Specifically, in Xia and Sprowl's (1991) method the physical-property distribution is  
 807 updated by using the ratio between the squared depth to the equivalent source and the gravitational constant  
 808 multiplied by the residual between the observed and predicted observation at the measurement station.  
 809 Neither of these methods solve linear systems.

810 Following this class of methods of iterative equivalent-layer technique that does not solve linear systems,  
 811 Siqueira et al. (2017) developed a fast iterative equivalent-layer technique for processing gravity data in  
 812 which the sensitivity matrix  $\mathbf{A}$  (equation 58) is replaced by a diagonal matrix  $N \times N$ , i.e.:

$$\tilde{\mathbf{A}} = 2\pi\gamma\Delta\mathbf{S}^{-1}, \quad (71)$$

813 where  $\gamma$  is Newton's gravitational constant and  $\Delta\mathbf{S}^{-1}$  is a diagonal matrix of order  $N$  whose diagonal  
 814 elements  $\Delta s_i$ ,  $i = 1, \dots, N$  are the element of area centered at the  $i$ th horizontal coordinates of the  $i$ th  
 815 observation point. The physical foundations of Siqueira et al.'s (2017) method rely on two constraints: i) the  
 816 excess of mass; and ii) the positive correlation between the gravity observations and the mass distribution  
 817 over the equivalent layer.

818 Although Siqueira et al.'s (2017) method does not solve any linear system of equations, it can be  
 819 theoretically explained by solving the following linear system at the  $k$ th iteration:

$$\tilde{\tilde{\mathbf{A}}}^\top \tilde{\tilde{\mathbf{A}}} \Delta \hat{\mathbf{p}}^k = \tilde{\tilde{\mathbf{A}}}^\top \mathbf{r}^k, \quad (72)$$

820 where  $\mathbf{r}^k$  is an  $N$ -dimensional residual vector whose  $i$ th element is calculated by subtracting the  $i$ th  
 821 observed data  $d_i^o$  from the  $i$ th fitted data  $d_i^k$  at the  $k$ th iteration, i.e.,

$$r_i^k = d_i^o - d_i^k. \quad (73)$$

822 and  $\Delta \hat{\mathbf{p}}^k$  is an estimated  $N$ -dimensional vector of parameter correction.

823 Because  $\tilde{\tilde{\mathbf{A}}}$ , in equation 72, is a diagonal matrix (equation 71), the parameter correction estimate is  
 824 directly calculated without solving system of linear equations, and thus, an  $i$ th element of  $\Delta \hat{\mathbf{p}}^k$  is directly  
 825 calculated by

$$\Delta \hat{p}_i^k = \frac{\Delta s_i r_i^k}{2 \pi \gamma}. \quad (74)$$

826 The mass distribution over the equivalent layer is updated by:

$$\hat{p}_i^{k+1} = \hat{p}_i^k + \Delta \hat{p}_i^k. \quad (75)$$

827 Siqueira et al.'s (2017) method starts from a mass distribution on the equivalent layer, whose  $i$ th mass  $p_i^o$  is  
 828 proportional to the  $i$ th observed data  $d_i^o$ , i.e.,

$$p_i^o = \frac{\Delta s_i d_i^o}{2 \pi \gamma}. \quad (76)$$

829 Siqueira et al. (2017) applied their fast iterative equivalent-layer technique to interpolate, calculate the  
 830 horizontal components, and continue upward (or downward) gravity data.

831 For jointly process two gravity gradient components, Jirigalatu and Ebbing (2019) used the Gauss-FFT  
 832 for forward calculation of potential fields in the wavenumber domain combined with Landweber's iteration  
 833 coupled with a mask matrix  $\mathbf{M}$  to reduce the edge effects without increasing the computation cost. The  
 834 mask matrix  $\mathbf{M}$  is defined in the following way: if the corresponding pixel does not contain the original  
 835 data, the element of  $\mathbf{M}$  is set to zero; otherwise, it is set to one. The  $k$ th Landweber iteration is given by

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \omega \left[ \mathbf{A}_1^\top (\mathbf{d}_1 - \mathbf{M} \mathbf{A}_1 \mathbf{p}_k) + \mathbf{A}_2^\top (\mathbf{d}_2 - \mathbf{M} \mathbf{A}_2 \mathbf{p}_k) \right], \quad (77)$$

836 where  $\omega$  is a relaxation factor,  $\mathbf{d}_1$  and  $\mathbf{d}_2$  are the two gravity gradient components and  $\mathbf{A}_1$  and  $\mathbf{A}_2$  are the  
 837 corresponding gravity gradient kernels. Jirigalatu and Ebbing (2019) applied their method for processing  
 838 two horizontal curvature components of Falcon airborne gravity gradient.

## 839 6.2.7 The convolutional equivalent layer with BTTB matrices

840 Takahashi et al. (2020, 2022) introduced the convolutional equivalent layer for gravimetric and magnetic  
 841 data processing, respectively.

842 Takahashi et al. (2020) demonstrated that the sensitivity matrix  $\mathbf{A}$  (equation 58) associated with a planar  
 843 equivalent layer formed by a set of point masses, each one directly beneath each observation point and  
 844 considering a regular grid of observation points at a constant height has a symmetric block-Toeplitz Toeplitz-  
 845 block (BTTB) structure. A symmetric BTTB matrix has, at least, two attractive properties. The first one is  
 846 that it can be defined by using only the elements forming its first column (or row). The second attractive  
 847 property is that any BTTB matrix can be embedded into a symmetric Block-Circulant Circulant-Block  
 848 (BCCB) matrix. This means that the full sensitivity matrix  $\mathbf{A}$  (equation 58) can be completely reconstruct  
 849 by using the first column of the BCCB matrix only. In what follows, Takahashi et al. (2020) computed  
 850 the forward modeling by using only a single equivalent source. Specifically, it is done by calculating the  
 851 eigenvalues of the BCCB matrix that can be efficiently computed by using only the first column of the  
 852 BCCB matrix via 2D fast Fourier transform (2D FFT). By comparing with the classic approach in the  
 853 Fourier domain, the convolutional equivalent layer for gravimetric data processing proposed by Takahashi  
 854 et al. (2020) performed upward- and downward-continue gravity data with a very small border effects and  
 855 noise amplification.

856 By using the original idea of the convolutional equivalent layer proposed by Takahashi et al. (2020)  
 857 for gravimetric data processing, Takahashi et al. (2022) developed the convolutional equivalent layer for  
 858 magnetic data processing. By assuming a regularly spaced grid of magnetic data at a constant height and a  
 859 planar equivalent layer of dipoles, Takahashi et al. (2022) proved that the sensitivity matrix linked with  
 860 this layer possess a BTTB structure in the specific scenario where each dipole is exactly beneath each  
 861 observed magnetic data point. Takahashi et al. (2022) used a conjugate gradient least-squares (CGLS)  
 862 algorithm which does not require an inverse matrix or matrix-matrix multiplication. Rather, it only requires  
 863 matrix-vector multiplications per iteration, which can be effectively computed using the 2D FFT as a  
 864 discrete convolution. The matrix-vector product only uses the elements that constitute the first column of  
 865 the associated BTTB matrix, resulting in computational time and memory savings. Takahashi et al. (2022)  
 866 showed the robustness of the convolutional equivalent layer in processing magnetic survey that violates the  
 867 requirement of regular grids in the horizontal directions and flat observation surfaces.

868 The matrix-vector product in Takahashi et al. (2020, 2022) (e.g.,  $\mathbf{d} = \mathbf{Ap}$ , such as in equation 58) is the  
 869 main issue to be solved. To solve it efficiently, these authors involved the auxiliary linear system

$$\mathbf{w} = \mathbf{Cv}, \quad (78)$$

870 where  $\mathbf{w}$  and  $\mathbf{v}$  are, respectively, vectors of data and parameters completed by zeros and  $\mathbf{C}$  is a BCCB  
 871 matrix formed by  $2Q \times 2Q$  blocks, where each block  $\mathbf{C}_q$ ,  $q = 0, \dots, Q - 1$ , is a  $2P \times 2P$  circulant matrix.  
 872 The first column of  $\mathbf{C}$  is obtained by rearranging the first column of the sensitivity matrix  $\mathbf{A}$  (equation 58).  
 873 Because a BCCB matrix is diagonalized by the 2D unitary discrete Fourier transform (DFT),  $\mathbf{C}$  can be  
 874 written as

$$\mathbf{C} = (\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P})^* \boldsymbol{\Lambda} (\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}), \quad (79)$$

875 where the symbol “ $\otimes$ ” denotes the Kronecker product (Neudecker, 1969),  $\mathbf{F}_{2Q}$  and  $\mathbf{F}_{2P}$  are the  $2Q \times 2Q$   
 876 and  $2P \times 2P$  unitary DFT matrices (Davis, 1979, p. 31), respectively, the superscript “ $*$ ” denotes the  
 877 complex conjugate and  $\boldsymbol{\Lambda}$  is a  $4QP \times 4QP$  diagonal matrix containing the eigenvalues of  $\mathbf{C}$ . Due to the

878 diagonalization of the matrix  $\mathbf{C}$ , the auxiliary system (equation 78) can be rewritten by using equation 79  
 879 and premultiplying both sides of the result by  $(\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P})$ , i.e.,

$$\Lambda (\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}) \mathbf{v} = (\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}) \mathbf{w}. \quad (80)$$

880 By applying the vec-operator (Takahashi et al., 2020) to both sides of equation 80, by premultiplying both  
 881 sides of the result by  $\mathbf{F}_{2Q}^*$  and then postmultiplying both sides of the result by  $\mathbf{F}_{2P}^*$

$$\mathbf{F}_{2Q}^* [\mathbf{L} \circ (\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P})] \mathbf{F}_{2P}^* = \mathbf{W}, \quad (81)$$

882 where “ $\circ$ ” denotes the Hadamard product (Horn and Johnson, 1991, p. 298) and  $\mathbf{L}$ ,  $\mathbf{V}$  and  $\mathbf{W}$  are  $2Q \times 2P$   
 883 matrices obtained by rearranging, along their rows, the elements forming the diagonal of matrix  $\Lambda$ , vector  $\mathbf{v}$   
 884 and vector  $\mathbf{w}$ , respectively. The left side of equation 81 contains the 2D Inverse Discrete Fourier Transform  
 885 (IDFT) of the term in brackets, which in turn represents the Hadamard product of matrix  $\mathbf{L}$  and the 2D  
 886 DFT of matrix  $\mathbf{V}$ . Matrix  $\mathbf{L}$  contains the eigenvalues of  $\Lambda$  (equation 79) and can be efficiently computed  
 887 by using only the first column of the BCCB matrix  $\mathbf{C}$  (equation 78).

888 Actually, in Takahashi et al. (2020, 2022) a fast 2D discrete circular convolution (Van Loan, 1992) is  
 889 used to process very large gravity and magnetic datasets efficiently. The convolutional equivalent layer  
 890 was applied to perform upward continuation of large magnetic datasets. Compared to the classical Fourier  
 891 approach, Takahashi et al.’s (2022) method produces smaller border effects without using any padding  
 892 scheme.

893 Without taking advantage of the symmetric BTTB structure of the sensitivity matrix (Takahashi et al.,  
 894 2020) that arises when gravimetric observations are measured on a horizontally regular grid, on a flat  
 895 surface and considering a regular grid of equivalent sources whithin a horizontal layer, Mendonça (2020)  
 896 explored the symmetry of the gravity kernel to reduce the number of forward model evaluations. By  
 897 exploting the symmetries of the gravity kernels and redundancies in the forward model evaluations on a  
 898 regular grid and combining the subspace solution based on eigenvectors of the gridded dataset, Mendonça  
 899 (2020) estimated the mass excess or deficiency produced by anomalous sources with positive or negative  
 900 density contrast.

### 901 6.2.8 The deconvolutional equivalent layer with BTTB matrices

902 To avoid the iterations of the conjugate gradient method in Takahashi et al. (2022), we can employ the  
 903 deconvolution process. Equation 81 shows that estimate the matrix  $\mathbf{V}$ , containing the elements of parameter  
 904 vector  $\mathbf{p}$ , is a inverse problem that could be solved by deconvolution. From equation 81, the matrix  $\mathbf{V}$  can  
 905 be obtain by deconvolution, i.e.

$$\mathbf{V} = \mathbf{F}_{2Q}^* \left[ \frac{(\mathbf{F}_{2Q} \mathbf{W} \mathbf{F}_{2P})}{\mathbf{L}} \right] \mathbf{F}_{2P}^*. \quad (82)$$

906 Equation 82 shows that the parameter vector (in matrix  $\mathbf{V}$ ) can be theoretically obtain by dividing each  
 907 potential-field observations (in matrix  $\mathbf{W}$ ) by each eigenvalues (in matrix  $\mathbf{L}$ ). Hence, the parameter vector  
 908 is constructed by element-by-element division of data by eigenvalues.

909 However, the deconvolution often is extremely unstable. This means that a small change in data can lead  
 910 to an enormous change in the estimated parameter. Hence, equation 82 requires regularization to be useful.

911 We used wiener deconvolution to obtain a stable solution, i.e.,

$$\mathbf{V} = \mathbf{F}_{2Q}^* \left[ (\mathbf{F}_{2Q} \mathbf{W} \mathbf{F}_{2P}) \frac{\mathbf{L}^*}{(\mathbf{L} \mathbf{L}^* + \mu)} \right] \mathbf{F}_{2P}^*, \quad (83)$$

912 where the matrix  $\mathbf{L}^*$  contains the complex conjugate eigenvalues and  $\mu$  is a parameter that controls the  
913 degree of stabilization.

### 914 6.3 Solution stability

915 The solution stability of the equivalent-layer methods is rarely addressed. Here, we follow the numerical  
916 stability analysis presented in Siqueira et al. (2017).

917 Let us assume noise-free potential-field data  $\mathbf{d}$ , we estimate a physical-property distribution  $\mathbf{p}$  (estimated  
918 solution) within the equivalent layer. Then, the noise-free data  $\mathbf{d}$  are contaminated with additive  $D$  different  
919 sequences of pseudorandom Gaussian noise, creating different noise-corrupted potential-field data  $\mathbf{d}_\ell^o$ ,  
920  $\ell = 1, \dots, D$ . From each  $\mathbf{d}_\ell^o$ , we estimate a physical-property distribution  $\hat{\mathbf{p}}_\ell$  within the equivalent layer.

921 Next, for each noise-corrupted data  $\mathbf{d}_\ell^o$  and estimated solution  $\hat{\mathbf{p}}_\ell$ , the  $\ell$ th model perturbation  $\delta p_\ell$  and the  
922  $\ell$ th data perturbation  $\delta d_\ell$  are, respectively, evaluated by

$$\delta p_\ell = \frac{\|\hat{\mathbf{p}}_\ell - \mathbf{p}\|_2}{\|\mathbf{p}\|_2}, \quad \ell = 1, \dots, D, \quad (84)$$

923 and

$$\delta d_\ell = \frac{\|\mathbf{d}_\ell^o - \mathbf{d}\|_2}{\|\mathbf{d}\|_2}, \quad \ell = 1, \dots, D. \quad (85)$$

924 Regardless of the particular method used, the following inequality (Aster et al., 2018, p. 66) is applicable:

$$\delta p_\ell \leq \kappa \delta d_\ell, \quad \ell = 1, \dots, D, \quad (86)$$

925 where  $\kappa$  is the constant of proportionality between the model perturbation  $\delta p_\ell$  (equation 84) and the data  
926 perturbation  $\delta d_\ell$  (equation 85). The constant  $\kappa$  acts as the condition number of an invertible matrix in a  
927 given inversion, and thus measures the instability of the solution. The larger (smaller) the value of  $\kappa$  the  
928 more unstable (stable) is the estimated solution.

929 Equation 86 shows a linear relationship between the model perturbation and the data perturbation. By  
930 plotting  $\delta p_\ell$  (equation 84) against  $\delta d_\ell$  (equation 85) produced by a set of  $D$  estimated solution obtained by  
931 applying a given equivalent-layer method, we obtain a straight line behaviour described by equation 86.  
932 By applying a linear regression, we obtain a fitted straight line whose estimated slope ( $\kappa$  in equation 86)  
933 quantifies the solution stability.

934 Here, the analysis of solution stability is numerically conducted by applying the classical equivalent-  
935 layer technique with zeroth-order Tikhonov regularization, the convolutional method for gravimetric and  
936 magnetic data, the deconvolutional method (equation 82) and the deconvolutional method with different  
937 values for the Wiener stabilization (equation 83).

## 7 NUMERICAL SIMULATIONS

### 938 7.1 Floating-point operations calculation

939 To measure the computational effort of the different algorithms to solve the equivalent layer linear system,  
 940 a non-hardware dependent method can be useful because allow us to do direct comparison between them.  
 941 Counting the floating-point operations (*flops*), i.e., additions, subtractions, multiplications and divisions is  
 942 a good way to quantify the amount of work of a given algorithm (Golub and Loan, 2013). For example, the  
 943 number of *flops* necessary to multiply two vectors  $\mathbb{R}^D$  is  $2D$ . A common matrix-vector multiplication with  
 944 dimension  $\mathbb{R}^{D \times D}$  and  $\mathbb{R}^D$ , respectively, is  $2D^2$  and a multiplication of two matrices  $\mathbb{R}^{D \times D}$  is  $2D^3$ . Figure  
 945 1 shows the total flops count for the different methods presented in this review with a crescent number of  
 946 data, ranging from 10,000 to 1,000,000.

#### 947 7.1.1 Normal equations using Cholesky decomposition

948 The equivalent sources can be estimated directly from solving the normal equations 18. In this work  
 949 we will use the Cholesky decompositions method to calculate the necessary *flops* for a overdetermined  
 950 problem (equation 20). In this method it is necessary to calculate the lower triangule matrix of the left side  
 951 equation ( $1/2D^3$ ), the Cholesky factor ( $1/3D^3$ ), a matrix-vector multiplication ( $2D^2$ ) and finally solving  
 952 the triangular system ( $2D^2$ ), totalizing

$$f_{classical} = \frac{5}{6}D^3 + 4D^2 \quad (87)$$

#### 953 7.1.2 Window method (Leão and Silva, 1989)

954 The moving data-window scheme (Leão and Silva, 1989) solve  $M$  linear systems with much smaller  
 955 sizes (equation 60b) in comparison to the original  $D \times D$  system. For our results we are considering a  
 956 data-window of the same size of wich the authors presented in theirs work ( $D' = 49$ ) and the same number  
 957 of equivalent sources ( $P' = 225$ ). Using the algorithm 1 as a guide, we have a matrix-matrix multiplication  
 958 ( $2D'^2P'$ ), a scalar multiplication and a sum with diagonal matrices ( $D'$  each), a matrix inverse ( $D'^3$ ),  
 959 another matrix-matrix product ( $2D'P'^2$ ), a matrix-vetor product ( $2P'D'$ ) and finally a iteration with a  
 960 vector-vector multiplication ( $2D'$ ). The *flops* are

$$f_{window} = M2D' + 2P'D' + \frac{2D'^3}{3} + 2D' + 2D'P'(D' + P') \quad (88)$$

961 Here we are considering a  $2D'^3/3$  *flops* count for the Gauss-Jordan inverse matrix algorithm and  $M = D$   
 962 as we want to calculate the same number of transformations as observation points. Notice that this algorithm  
 963 takes advantage of a regular grid to calculate only once the inverse matrix and the harmonic functions of  
 964  $a'$ , with a irregular grid these calculations would be necessary at each iteration. Also this method does  
 965 not store the equivalent sources estimatives saving computer memory, however, any other potential field  
 966 transformation would require to run the algorithm again with the compatible harmonic function  $a'$  (equation  
 967 5).

#### 968 7.1.3 PEL method (Oliveira Jr. et al., 2013)

969 The polynomial equivalent layer uses a similiar approach of moving windows from Leão and Silva (1989).  
 970 For this operations calculation (equation 32) we used a first degree polynomial (two variables) and each

971 window contains  $D' = 1,000$  observed data and  $P' = 1,000$  equivalent sources. Following the steps given  
 972 in (Oliveira Jr. et al., 2013) the total *flops* becomes

$$f_{pel} = \frac{1}{3}N_c^3 + 2N_c^2 + 2DP'N_c + N_c^2D + 2N_cD + 2DC \quad (89)$$

973 where  $N_c$  is the number of constant coefficients for the first degree polynomial ( $C = 3$ ) times the number  
 974 of windows ( $C \times M$ ).

#### 975 7.1.4 Conjugate gradient least square (CGLS)

976 The CGLS method is a very stable and fast algorithm for solving linear systems iteratively. Its computa-  
 977 tional complexity envolves a matrix-vector product outside the loop ( $2D^2$ ), two matrix-vector products  
 978 inside the loop ( $4D^2$ ) and six vector products inside the loop ( $12D$ ) (Aster et al., 2018)

$$f_{ccls} = 2D^2 + it(4D^2 + 12D) \quad (90)$$

#### 979 7.1.5 Wavelet compression method with CGLS (Li and Oldenburg, 2010)

980 For the wavelet method (equation 63) we have calculated a compression rate of 98% ( $C_r = 0.02$ )  
 981 as the authors used in Li and Oldenburg (2010) and the wavelet transformation requiring  $\log_2(D)$  *flops*  
 982 each (equations 34 and 35), with its inverse also using the same number of operations (equation 38). The  
 983 normalization using diagonal matrix  $L$  in equations 36, 37 and 38 can be simplified to a matrix-vector  
 984 product ( $2DP$ ) and two vector-vector products( $2P_{each}$ ). Combined with the conjugate gradient least  
 985 square necessary steps and iterations, the number of *flops* are

$$f_{wavelet} = 2DP + 4P + 2DC_r + 4D \log_2(D) + it(4D \log_2(D) + 4DC_r + 12C_r) \quad (91)$$

#### 986 7.1.6 Fast equivalent layer for gravity data (Siqueira et al., 2017)

987 The fast equivalent layer from Siqueira et al. (2017) solves the linear system in *it* iterations. The main  
 988 cost of this method (algorithm 5) is the matrix-vector multiplication to assess the predicted data ( $2D^2$ ) and  
 989 three simply element by element vector sum, subtraction and division ( $3D$  total)

$$f_{siqueira} = it(3D + 2D^2) \quad (92)$$

#### 990 7.1.7 Convolutional equivalent layer for gravity data (Takahashi et al., 2020)

991 This methods replaces the matrix-vector multiplication of the iterative fast-equivalent technique (Siqueira  
 992 et al., 2017) by three steps, involving a Fourier transform, an inverse Fourier transform, and a Hadamard  
 993 product of matrices (equation 53). Considering that the first column of our BCCB matrix has  $4D$  elements,  
 994 the flops count of this method is a combination of algorithms 5 and 7

$$f_{convgrav} = \kappa 4D \log_2(4D) + it(27D + \kappa 8D \log_2(4D)) \quad (93)$$

995 In the resultant count we considered a radix-2 algorithm for the fast Fourier transform and its inverse,  
 996 which has a  $\kappa$  equals to 5 and requires  $\kappa 4D \log_2(4D)$  flops each. The Hadarmard product of two matrices  
 997 of  $4D$  elements with complex numbers takes  $24D$  flops. Note that equation 93 is different from the one

998 presented in Takahashi et al. (2020) because we also added the flops necessary to calculate the eigenvalues  
999 in this form. It does not differentiate much in order of magnitude because the iterative part is the most  
1000 costful.

1001 7.1.8 Convolutional equivalent layer for magnetic data (Takahashi et al., 2022)

1002 The convolutional equivalent layer for magnetic (algorithm 6) data uses the same flops count of the main  
1003 operations as in the gravimetric case (equation 53), the difference is the use of the conjugate gradient  
1004 algorithm to solve the inverse problem. It requires a Hadamard product outside of the iterative loop and the  
1005 matrix-vector and vector-vector multiplications inside the loop as seem in equation 90.

$$f_{convmag} = \kappa 16D \log_2(4D) + 24D + it(\kappa 16D \log_2(4D) + 60D) \quad (94)$$

1006 7.1.9 Deconvolutional method

1007 The deconvolution method does not require an iterative algorithm, rather it solves the estimative of the  
1008 physical properties in a single step using the  $4D$  eigenvalues of the BCCB matrix as in the convolutional  
1009 method. From equation 82 it is possible to deduce this method requires two fast Fourier transform  
1010 ( $\kappa 4D \log_2(4D)$ ), one for the eigenvalues and another for the data transformation, an element by element  
1011 division ( $24D$ ) and finally, a fast inverse Fourier transform for the final estimative ( $\kappa 4D \log_2(4D)$ ).

$$f_{deconv} = \kappa 12D \log_2(4D) + 24D \quad (95)$$

1012 Using the deconvolutional method with a Wiener stabilization adds two multiplications of complex  
1013 elements of the conjugates eigenvalues ( $24D$  each) and the sum of  $4D$  elements with the stabilization  
1014 parameter  $\mu$  as shown in equation 83

$$f_{deconvwiener} = \kappa 12D \log_2(4D) + 76D \quad (96)$$

## 8 SYNTHETIC DATA SIMULATIONS

1015 For all applications, we generate a model composed by two spheres and a polygonal prism in a regular  
 1016 spaced grid of  $50 \times 50$ . The upper left sphere has a density contrast of  $600 \text{ kg/m}^3$ , the right upper sphere a  
 1017 negative contrast of  $-500 \text{ kg/m}^3$  and the bottom prism is equal to  $550 \text{ kg/m}^3$ . To generate the magnetic  
 1018 data, the bodies are in the same position and all of them have the same magnetization intensity and  
 1019 direction ( $3.46 \text{ A/m}$  intensity,  $35.26^\circ$  inclination and  $45.0^\circ$  declination) within a simulated geomagnetic  
 1020 field direction of  $20.0^\circ$  inclination and  $35.0^\circ$  declination. These synthetic data are shown in figures 3 and  
 1021 7, respectively.

### 1022 8.1 Stability analysis

1023 For the stability analysis we show the comparison of the normal equations solution (equation 59) with  
 1024 zeroth-order Tikhonov regularization (Aster et al., 2018), the convolutional method (equation 81), the  
 1025 deconvolutional method (equation 82) and the deconvolutional method with different values for the Wiener  
 1026 stabilization (equation 83). We create 21 data sets, for both gravity and magnetic data, adding a crescent  
 1027 pseudo-random noise to the original data, which varies from 0% to 10% of the maximum anomaly value in  
 1028 intervals of 0.5%. These noises has mean equal to zero and a Gaussian distribution. These synthetic data  
 1029 are shown in figures 3 and 7, where panel (A) of each figure represents the noise free data and panel (B) is  
 1030 the maximum noised data for gravity and magnetic, respectively.

1031 Figure 2 shows how the euclidian norm of the equivalent sources residuals varies as the level of the noise  
 1032 is increased for the gravimetric data. We can see that for all methods, a linear tendency can be observed as  
 1033 it is expected. The inclination of the straight line is a indicative of the stability of each method. As show  
 1034 in the graph the deconvolutional method is very unstable and it is really necessary to use a stabilization  
 1035 method to have a good parameter estimative. In contrast, a correct value of the stabilization parameter is  
 1036 necessary to not overshoot the smoothness of the solution as it is the case for the zeroth-order Tikhonov  
 1037 regularization as well. Using this gravimetric data, the optimal value for the Wiener stabilization parameter  
 1038 is  $\mu = 10^{-20}$ .

1039 Figure 4 shows the comparison of the predicted data for each method with the original data (figure 3)  
 1040 using the most noised-corrupted data from the set of the stability analysis. The classical with zeroth-order  
 1041 Tikhonov regularization and the convolutional methods (figures 4(A) and 4(B)) yield very similar results for  
 1042 the predicted data confirming its similarities with the stabilization despite the bid difference in floating-point  
 1043 operations. Figure 4(C) shows the deconvolutional method without a stabilization and demonstrates the  
 1044 necessity to use it for this method. Figure 4(D) shows the deconvolutional method with Wiener stabilization  
 1045  $\mu = 10^{-15}$  which is too high, demonstrating the over smoothness of the predicted data. Figures 4(E) and  
 1046 4(F) shows the predicted data for an optimal value of the Wiener parameter  $\mu = 10^{-20}$  and a low value  
 1047  $\mu = 10^{-25}$ , respectively.

1048 The upward continuation is a processing technique to visualize the data in a higher altitude. In practice is  
 1049 expected a lower amplitude signal and a smoother data as the high frequency anomalies tends to disappear.  
 1050 Figure 5(A) shows the true modeled upward data at an height of  $-500 \text{ m}$ . Figures 5(B), (C), (D) and  
 1051 (E) show the result of the upward processing for the classical, convolutional, deconvolutional and the  
 1052 deconvolutional with Wiener parameter  $\mu = 10^{-20}$ , respectively. It is clear that all methods seems to  
 1053 predict the upward data very reasonable, except the deconvolutional method without stabilization.

1054 For the magnetic data, figure 6 shows a very similar behavior of the stability as the previous case. The  
 1055 Wiener parameter seems to have the best solution for  $\mu = 10^{-13}$ . For both types of data the best Wiener

1056 parameter seems to be one that produces a low slope for the straight line in the stability analysis, discordant  
1057 from the classical and convolutional methods.

1058 Figure 8 shows the comparison of the predicted data for each method with the original magnetic data in  
1059 figure 7 using the most noised-corrupted data modeled from the stability analysis. As the previous case the  
1060 classical (figure 8(A)) and the convolutional (figure 8(B)) methods have very similar predicted data but  
1061 estimated with less orders of magnitude in floating-point operations. The deconvolutional (figure 8(C)) have  
1062 have a strong disagreement with the observed data showing the need for a stabilization method. Figure  
1063 8(D) has a value of  $\mu = 10^{-10}$  and the predicted data became to smooth by it. The optimal value of the  
1064 Wiener parameter is shown in figure 8(E) with  $\mu = 10^{-13}$  and figure 8(F) shows a predicted data with a  
1065 low stabilization value with  $\mu = 10^{-16}$ .

1066 Figure 9(A) shows the true modeled upward data at an height of  $-1400$  m. Figures 9(B), (C), (D) and  
1067 (E) show the result of the upward processing for the classical, convolutional, deconvolutional and the  
1068 deconvolutional with Wiener parameter  $\mu = 10^{-13}$ , respectively. As in the gravimetric case, all methods  
1069 seems to predict the upward data, except the deconvolutional method without stabilization.

## 9 REAL DATA RESULTS

1070 In this section, we show the applications of the convolutional and the deconvolutional strategies in a real  
1071 data set from the North of Brazil. The region is located in the Carajás Mineral Province (CMP) in the  
1072 Amazon craton (Moroni et al., 2001; Villas and Santos, 2001). This area is known for its intensive mineral  
1073 exploration such as iron, copper, gold, manganese, and, recently, bauxite.

1074 **9.1 Geological setting**

1075 The Amazon craton is one of the largest and least-known Archean-Proterozoic areas in the world,  
1076 comprehending a region with a thousand square kilometers. It is one of the main tectonic units in South  
1077 America, which is covered by five Phanerozoic basins: Maranhão (Northeast), Amazon (Central), Xingu-  
1078 Alto Tapajós (South), Parecis (Southwest), and Solimões (West). The craton is limited by the Andean  
1079 Orogenic Belt to the west and the by Araguaia Fold Belt to the east and southeast. The Amazon craton has  
1080 been subdivided into provinces according to two models, one geochronological and the other geophysical-  
1081 structural (Amaral, 1974; Teixeira et al., 1989; Tassinari and Macambira, 1999). Thus, seven geological  
1082 provinces with distinctive ages, evolution, and structural patterns can be observed, namely : (i) Carajás with  
1083 two domains - the Mesoarchean Rio Maria and Neoarchean Carajás; (ii) Archean-Paleoproterozoic Central  
1084 Amazon, with Iriri-Xingu and Curuá-Mapuera domains; (ii) Trans-Amazonian (Ryacian), with the Amapá  
1085 and Bacajá domains; (iv) the Orosinian Tapajós-Parima, with Peixoto de Azevedo, Tapajós, Uaimiri, and  
1086 Parima domains; (v) Rondônia-Juruena (Statherian), with Jamari, Juruena, and Jauru domains; (vi) The  
1087 Statherian Rio Negro, with Rio Negro and Imeri domains; and (vii) Sunsás (Meso-Neoproterozoic), with  
1088 Santa Helena and Nova Brasilândia domains (Santos et al., 2000). Nevertheless, we focus this work only  
1089 on the Carajás Province.

1090 The Carajás Mineral Province (CMP) is located in the east-southeast region of the craton, within an old  
1091 tectonically stable nucleus in the South American Plate that became tectonically stable at the beginning of  
1092 Neoproterozoic (Salomao et al., 2019). This area has been the target of intensive exploration at least since  
1093 the final of the '60s, after the discovery of large iron ore deposits. There are several greenstone belts in the  
1094 region, among them are the Andorinhas, Inajá, Cumaru, Carajás, Serra Leste, Serra Pelada, and Sapucaia  
1095 (Santos et al., 2000). The mineralogic and petrologic studies in granite stocks show a variety of minerals  
1096 found in the province, such as amphibole, plagioclase, biotite, ilmenite, and magnetite (Cunha et al., 2016).  
1097 These two latter minerals contribute to the high magnetic response in the CMP area. This fact opens the  
1098 opportunity for potential field applications for the geophysical description of the area.

1099 **9.2 Potential field data applications**

1100 Here we compare the performance of the convolutional and deconvolutional algorithms in a real potential  
1101 field data set. We focus the application on a region in the Southeast of the State of Pará. The aeromagnetic  
1102 data were acquired by the Geological Survey of Brazil-CPRM. The survey area covers  $\approx 58000 \text{ km}^2$  with  
1103 high-resolution gravity and magnetic data. The flight and the tie lines were acquired and spaced at  $3 \text{ km}$   
1104 and  $12 \text{ km}$  oriented in the directions  $N - S$  and  $E - W$ , respectively, with a mean flight height of  $900 \text{ m}$   
1105 above the ground. For both applications, we interpolated gravity and magnetic anomalies data, calculating  
1106 the data set in a grid of  $1000 \times 500$  ( $N = 500000$  observation points) at the same mean flight height. About  
1107 the computational resources, we processed both data on an Intel Core i7 7700HQ@2.8 GHz processor and  
1108 16GB RAM. We show in Figure 10 and Figure 12 the interpolated aerogravimetric and aeromagnetic data,  
1109 respectively. We also use the same equivalent layer grid configuration in gravity and magnetic applications.

1110 This setup is composed by a grid of  $1000 \times 500$  equivalent sources (a total number of  $M = 500000$  points)  
1111 positioned below the observation plane, but a different depth in each application.

1112 We apply both strategies to the gravimetric case. We set a depth for the equivalent layer equal to  $1200\text{ m}$   
1113 below the observation plane. Figure 11A and Figure 11C show the predicted data for convolutional and  
1114 deconvolutional strategies. The residual maps (the difference between the observed and predicted data)  
1115 are show in figures 11B and 11D for the convolutional and deconvolutional equivalent-layer technique,  
1116 respectively. For the convolutional case, the mean residual and standard deviation values are  $\approx 0.00\text{ mGal}$   
1117 and  $\approx 0.15\text{ mGal}$ , respectively. For the deconvolutional case, the mean residual and standard deviation  
1118 values are  $\approx 0.46\text{ mGal}$  and  $\approx 1.23\text{ mGal}$ , respectively. These last results show that the estimated density  
1119 distributions (not shown) fit the observed data for both applications. To show the performance of the  
1120 algorithms, we performed an upward continuation by using the estimated density distributions (figures 11E  
1121 and 11F). There is a little difference on the processing time between both strategies. The convolutional  
1122 algorithm took  $\approx 9.18\text{ s}$  and the deconvolutional algorithm took  $\approx 0.53\text{ s}$ . We conclude that both strategies  
1123 are capable of processing gravimetric observations from large areas with dense coverage data. Despite a  
1124 little difference in processing time, the deconvolutional equivalent-layer technique proved to be faster than  
1125 the convolutional strategy.

1126 Finally, we test the convolutional and deconvolutional algorithms for processing total-field anomalies.  
1127 We stress that the Carajás area is very large and the main field direction varies significantly. For this reason,  
1128 we consider a mean direction for the main field equal to  $-19.865^\circ$  and  $-7.43915^\circ$  (the same as the mid  
1129 location of the area) for the inclination and declination, respectively. Furthermore, we are not considering  
1130 the knowledge about the magnetization direction of the sources, and choose a magnetization direction for  
1131 the equivalent layer equal to the main field direction. For this application, we set a depth of  $900\text{ m}$  (below  
1132 the observation plane) for the equivalent layer. Figure 12A and Figure 12C show the predicted data for  
1133 convolutional and deconvolutional algorithms. The residual maps (the difference between the observed and  
1134 predicted data) are show in figures 12B and 12D for the convolutional and deconvolutional techniques,  
1135 respectively. The convolutional equivalent layer produced a mean residual and standard deviation values  
1136 of  $\approx 0.06\text{ nT}$  and  $\approx 1.97\text{ nT}$ , respectively. The deconvolutional algorithm produced a mean residual  
1137 and standard deviation values of  $\approx 18.99\text{ nT}$  and  $\approx 33.64\text{ nT}$ , respectively. To show the performance  
1138 of the algorithms, we performed an upward continuation (figures 12E and 12F) by using the estimated  
1139 magnetic-moment distributions (not shown). Similarly to the gravity application, the deconvolutional  
1140 equivalent layer presents faster results than the convolutional algorithm. The deconvolutional and the  
1141 convolutional approaches took  $\approx 0.89\text{ s}$  and  $\approx 82.08\text{ s}$ , respectively. Despite the difference between the  
1142 processing time of both strategies and considering the mean value of residuals and standard deviations, we  
1143 conclude that the convolutional strategy fits the observation data better than the deconvolutional approach.

## 10 DISCUSSION AND CONCLUSION

1144 We show in this work that the computational cost of the equivalent layer can vary from up to  $10^9$  flops  
1145 depending on the method without compromising the linear system stability. The convolutional method has  
1146 the necessity of using a regular grid, a problem that can be easily overcome with a gridding or interpolation  
1147 method. The deconvolution is one of the fastest method but, has two drawbacks besides the regular grid,  
1148 the necessity of choosing the correct parameter of stabilization and also a higher residual and border effects  
1149 after processing, that can be seen from the upward continuation of the real data from Carajás. The authors  
1150 advise cautionary use of the latter method despite the computational cost reduction.

## CONFLICT OF INTEREST STATEMENT

1151 The authors declare that the research was conducted in the absence of any commercial or financial  
1152 relationships that could be construed as a potential conflict of interest.

## AUTHOR CONTRIBUTIONS

1153 The Author Contributions section is mandatory for all articles, including articles by sole authors. If an  
1154 appropriate statement is not provided on submission, a standard one will be inserted during the production  
1155 process. The Author Contributions statement must describe the contributions of individual authors referred  
1156 to by their initials and, in doing so, all authors agree to be accountable for the content of the work. Please  
1157 see here for full authorship criteria.

## FUNDING

1158 Diego Takahashi was supported by a Post-doctoral scholarship from CNPq (grant 300809/2022-0) Valéria  
1159 C.F. Barbosa was supported by fellowships from CNPq (grant 309624/2021-5) and FAPERJ (grant  
1160 26/202.582/2019). Vanderlei C. Oliveira Jr. was supported by fellowships from CNPq (grant 315768/2020-  
1161 7) and FAPERJ (grant E-26/202.729/2018).

## ACKNOWLEDGMENTS

1162 We thank the Brazilian federal agencies CAPES, CNPq, state agency FAPERJ and Observatório Nacional  
1163 research institute and Universidade do Estado do Rio de Janeiro.

## DATA AVAILABILITY STATEMENT

1164 The datasets generated for this study can be found in the frontiers-paper Github repository link:  
1165 <https://github.com/DiegoTaka/frontiers-paper>.

## REFERENCES

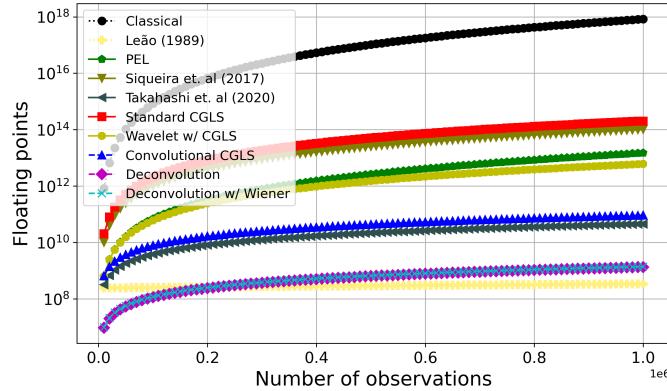
- 1166 Amaral, G. (1974). *Geologia Pré-Cambriana da Região Amazônica*. Ph.D. thesis, Universidade de São  
1167 Paulo
- 1168 Aster, R. C., Borchers, B., and Thurber, C. H. (2018). *Parameter estimation and inverse problems*  
1169 (Elsevier)
- 1170 Barbosa, V. C. F., Silva, J. B., and Medeiros, W. E. (1997). Gravity inversion of basement relief using  
1171 approximate equality constraints on depths. *Geophysics* 62, 1745–1757
- 1172 Barnes, G. and Lumley, J. (2011). Processing gravity gradient data. *Geophysics* 76, I33–I47. doi:10.1190/  
1173 1.3548548
- 1174 Cordell, L. (1992). A scattered equivalent-source method for interpolation and gridding of potential-field  
1175 data in three dimensions. *Geophysics* 57, 629–636
- 1176 Cunha, I. R., Dall’Agnol, R., and Feio, G. R. L. (2016). Mineral chemistry and magnetic petrology of the  
1177 archean planalto suite, carajás province – amazonian craton: Implications for the evolution of ferroan  
1178 archean granites. *Journal of South American Earth Sciences* 67, 100–121
- 1179 Dampney, C. N. G. (1969). The equivalent source technique. *Geophysics* 34, 39–53. doi:10.1190/1.  
1180 143996

- 1181 Davis, K. and Li, Y. (2011). Fast solution of geophysical inversion using adaptive mesh, space-filling  
1182 curves and wavelet compression. *Geophysical Journal International* 185, 157–166. doi:10.1111/j.  
1183 1365-246X.2011.04929.x
- 1184 Davis, P. J. (1979). *Circulant matrices* (John Wiley & Sons, Inc.)
- 1185 Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics* ,  
1186 1189–1232
- 1187 Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics and data analysis* 38,  
1188 367–378
- 1189 Golub, G. H. and Loan, C. F. V. (2013). *Matrix Computations (Johns Hopkins Studies in the Mathematical  
1190 Sciences)* (Johns Hopkins University Press), 4 edn.
- 1191 Guspí, F. and Novara, I. (2009). Reduction to the pole and transformations of scattered magnetic data using  
1192 newtonian equivalent sources. *Geophysics* 74, L67–L73
- 1193 Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis* (Cambridge University Press), 1 edn.
- 1194 Jirigalatu, J. and Ebbing (2019). A fast equivalent source method for airborne gravity gradient data.  
1195 *Geophysics* 84, G75–G82. doi:10.1190/GEO2018-0366.1
- 1196 Kellogg, O. D. (1929). *Foundations of Potential Theory* (Frederick Ungar Publishing Company)
- 1197 Kennett, B., Sambridge, M., and Williamson, P. (1988). Subspace methods for large inverse problems with  
1198 multiple parameter classes. *Geophysical Journal International* 94, 237–247
- 1199 Leão, J. W. D. and Silva, J. B. C. (1989). Discrete linear transformations of potential field data. *Geophysics*  
1200 54, 497–507. doi:10.1190/1.1442676
- 1201 Li, Y. and Oldenburg, D. W. (2010). Rapid construction of equivalent sources using wavelets. *Geophysics*  
1202 75, L51–L59. doi:10.1190/1.3378764
- 1203 Mendonça, C. A. (2020). Subspace method for solving large-scale equivalent layer and density mapping  
1204 problems. *Geophysics* 85, G57–G68. doi:10.1190/geo2019-0302.1
- 1205 Mendonça, C. A. and Silva, J. B. C. (1994). The equivalent data concept applied to the interpolation of  
1206 potential field data. *Geophysics* 59, 722–732. doi:10.1190/1.1443630
- 1207 Moroni, M., Girardi, V., and Ferrario, A. (2001). The serra pelada au-pge deposit, serra dos carajás (pará  
1208 state, brazil): geological and geochemical indications for a composite mineralising process. *Mineralium  
1209 Deposita* 36, 768–785
- 1210 Neudecker, H. (1969). Some theorems on matrix differentiation with special reference to kronecker matrix  
1211 products. *Journal of the American Statistical Association* 64, 953–963. doi:10.1080/01621459.1969.  
1212 10501027
- 1213 Oldenburg, D., McGillivray, P., and Ellis, R. (1993). Generalized subspace methods for large-scale inverse  
1214 problems. *Geophysical Journal International* 114, 12–20
- 1215 Oliveira Jr., V. C., Barbosa, V. C. F., and Uieda, L. (2013). Polynomial equivalent layer. *Geophysics* 78,  
1216 G1–G13. doi:10.1190/geo2012-0196.1
- 1217 Salomao, G. N., Dall’Agnol, R., Angelica, R. S., Figueiredo, M. A., Sahoo, P. K., Medeiros-filho, C. A.,  
1218 et al. (2019). Geochemical mapping and estimation of background concentrations in soils of carajás  
1219 mineral province, eastern amazonian craton, brazil. *Geochemistry: Exploration, Environment, Analysis*  
1220 19, 431–447
- 1221 Santos, J. O. S., Hartmann, L. A., Gaudette, H. E., Groves, D. I., Mcnaughton, M. J., and Fletcher, I. R.  
1222 (2000). A new understanding of the provinces of the amazon craton based on integration of field mapping  
1223 and u-pb and sm-nd geochronology. *Gondwana Research* 3, 453–488
- 1224 Siqueira, F. C., Oliveira Jr, V. C., and Barbosa, V. C. (2017). Fast iterative equivalent-layer technique for  
1225 gravity data processing: A method grounded on excess mass constraint. *Geophysics* 82, G57–G69

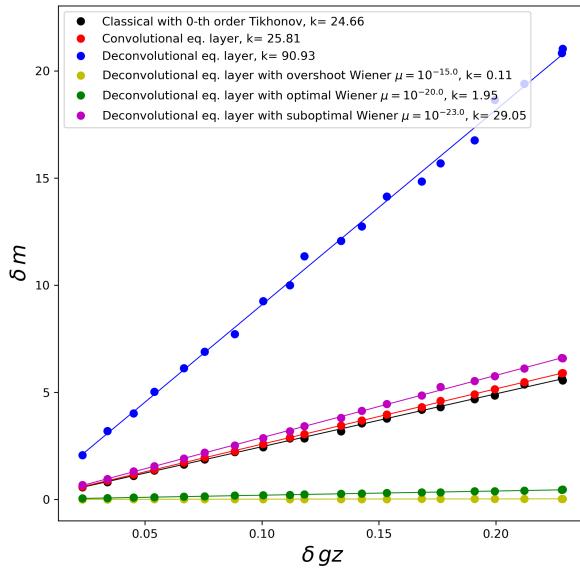
- 1226 Skilling, J. and Bryan, R. (1984). Maximum entropy image reconstruction-general algorithm. *Monthly*  
1227 *Notices of the Royal Astronomical Society*, Vol. 211, NO. 1, P. 111, 1984 211, 111
- 1228 Soler, S. R. and Uieda, L. (2021). Gradient-boosted equivalent sources. *Geophysical Journal International*  
1229 227, 1768–1783. doi:10.1093/gji/ggab297
- 1230 Takahashi, D., Oliveira, V. C., and Barbosa, V. C. (2022). Convolutional equivalent layer for magnetic data  
1231 processing. *Geophysics* 87, 1–59
- 1232 Takahashi, D., Oliveira Jr, V. C., and Barbosa, V. C. (2020). Convolutional equivalent layer for gravity data  
1233 processing. *Geophysics* 85, G129–G141
- 1234 Tassinari, C. C. and Macambira, M. J. (1999). Geochronological provinces of the amazonian craton.  
1235 *Episodes* 22, 174–182
- 1236 Teixeira, W., Tassinari, C., Cordani, U. G., and Kawashita, K. (1989). A review of the geochronology of  
1237 the amazonian craton: Tectonic implications. *Precambrian Research* 42, 213–227
- 1238 Van Loan, C. (1992). *Computational Frameworks for the Fast Fourier Transform* (SIAM)
- 1239 Villas, R. N. and Santos, M. (2001). Gold deposits of the carajás mineral province: deposit types and  
1240 metallogenesis. *Mineralium Deposita* 36, 300–331
- 1241 Xia, J. and Sprowl, D. R. (1991). Correction of topographic distortion in gravity data. *Geophysics* 56,  
1242 537–541
- 1243 Xia, J., Sprowl, D. R., and Adkins-Helgeson, D. (1993). Correction of topographic distortions in potential-  
1244 field data; a fast and accurate approach. *Geophysics* 58, 515–523. doi:10.1190/1.1443434

## 11 SUPPLEMENTARY TABLES AND FIGURES

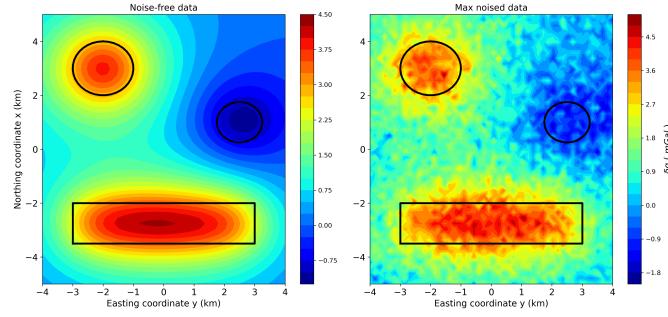
### 1245 11.1 Figures



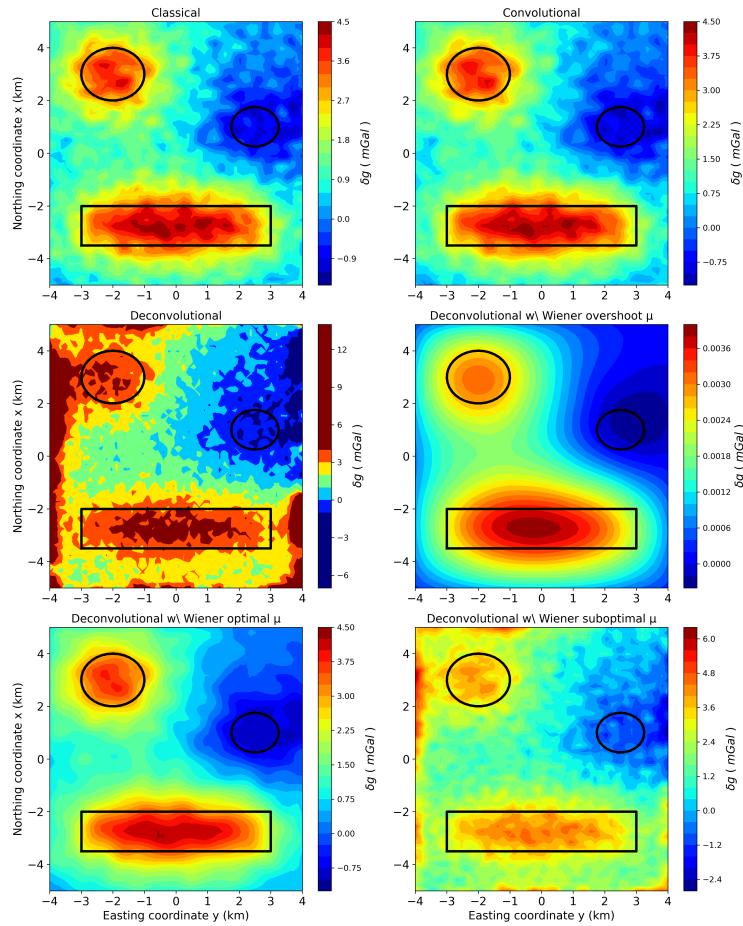
**Figure 1.** Number of *flops* for many of the methods described in this work to estimate the equivalent sources using gravity data. The range of observations varies from 10,000 to 1,000,000.



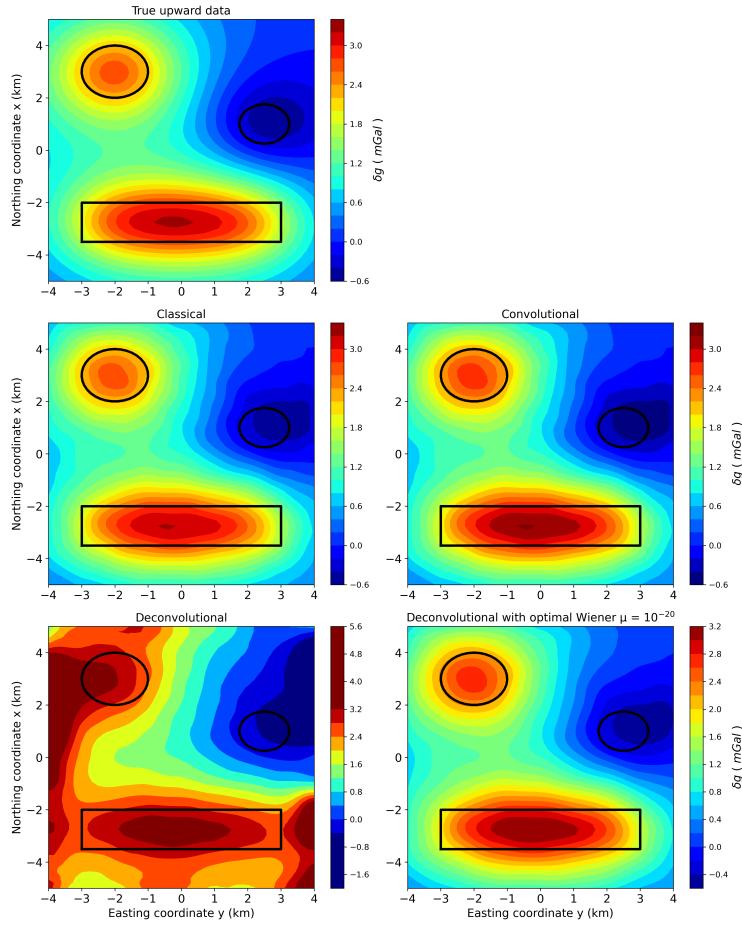
**Figure 2.** Stability analysis of some of the equivalent layer methods of the gravimetric case.



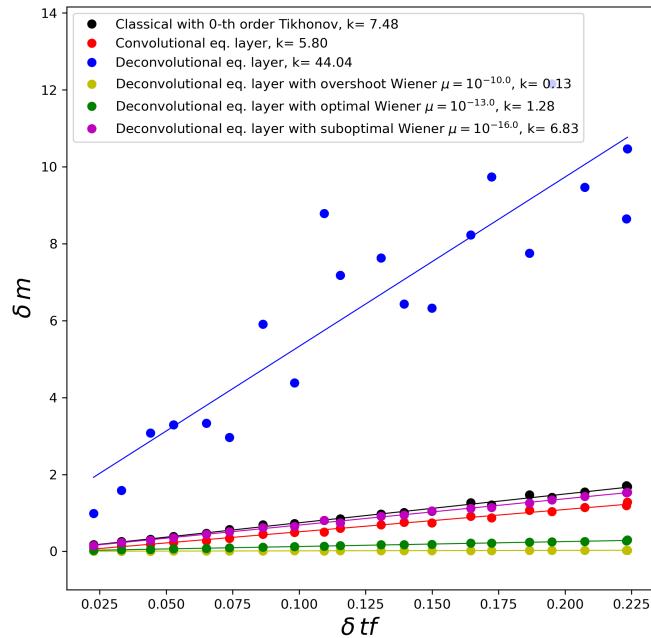
**Figure 3.** Synthetic data of the gravimetric case. The observations points are placed in a regular grid of  $50 \times 50$ . Panel (A) shows the noise-free data and panel (B) shows the maximum noised data (10%).



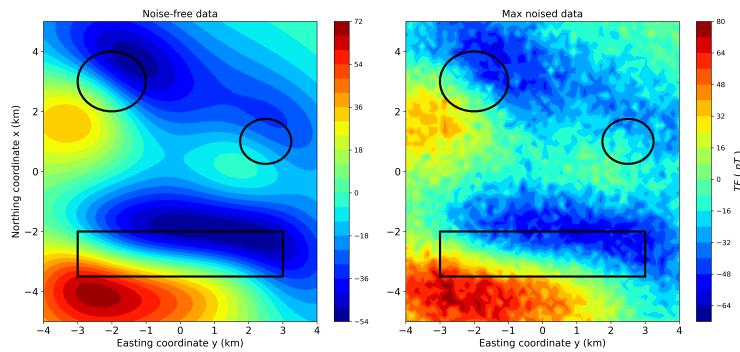
**Figure 4.** Predicted gravity data for different methods of the equivalent layer with maximum level of noise. Panel (A) is the classical method, (B) is the convolutional, (C) is the deconvolutional, (D) is the deconvolutional method using Wiener stabilization with a too high value for  $\mu$ , (E)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



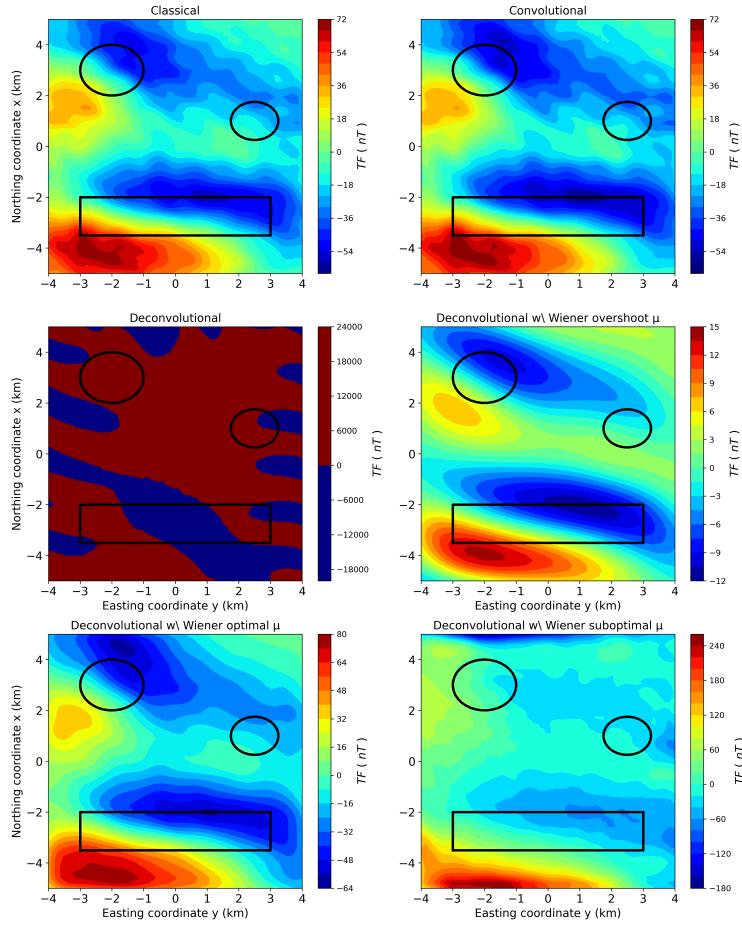
**Figure 5.** True noiseless upward gravimetric data at  $z_i = -500$  m height and predicted data for different methods of the equivalent layer with maximum level of noise. Panel (A) is the true upward gravity data, Panel (B) is the classical method, (C) is the convolutional, (D) is the deconvolutional,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



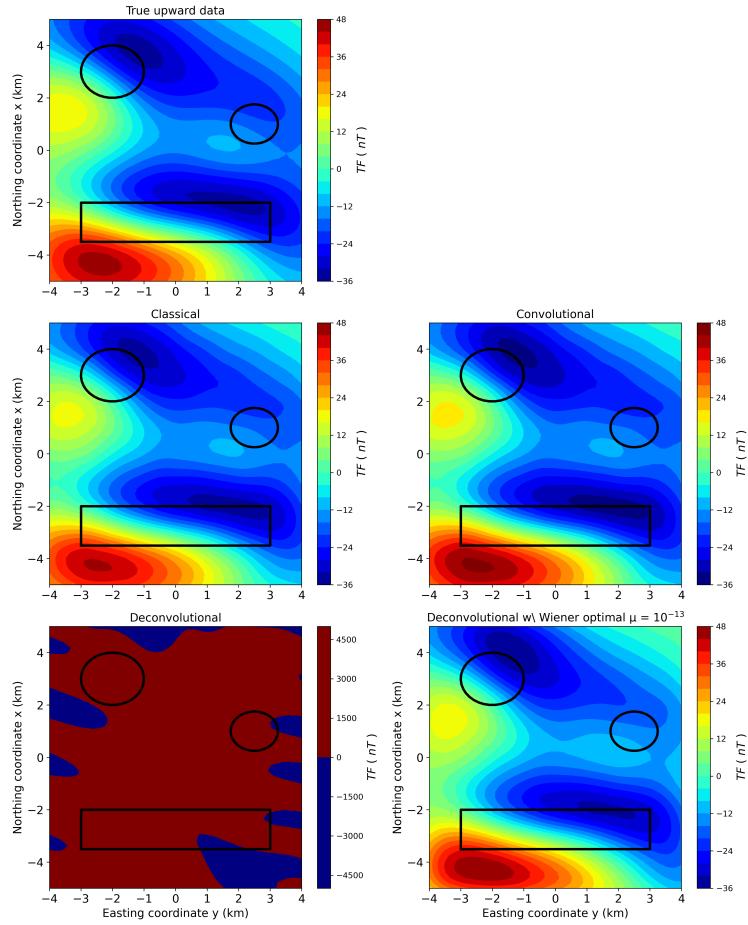
**Figure 6.** Stability analysis of some of the equivalent layer methods of the magnetic case.



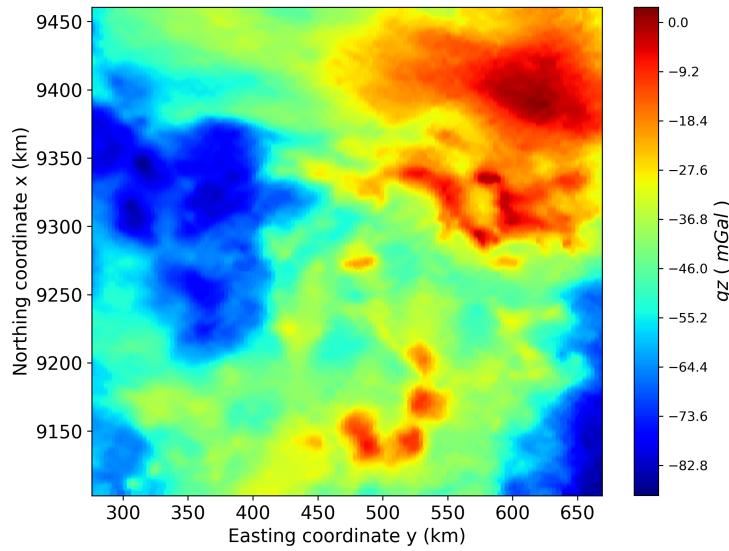
**Figure 7.** Synthetic data of the magnetic case. The observations points are placed in a regular grid of  $50 \times 50$ . Panel (A) shows the noise-free data and panel (B) shows the maximum noised data (10%).



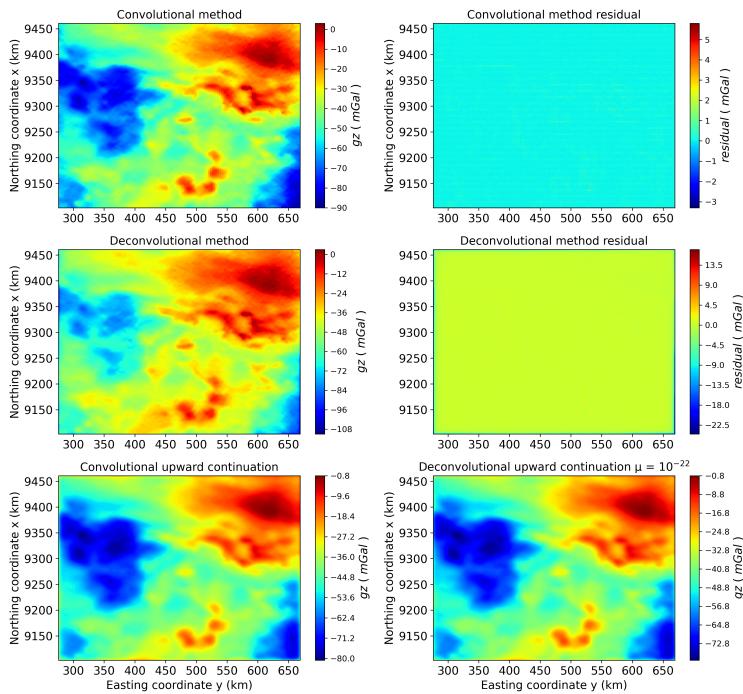
**Figure 8.** Predicted magnetic data for different methods of the equivalent layer with maximum level of noise. Panel **(A)** is the classical method, **(B)** is the convolutional, **(C)** is the deconvolutional, **(D)** is the deconvolutional method using Wiener stabilization with a too high value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



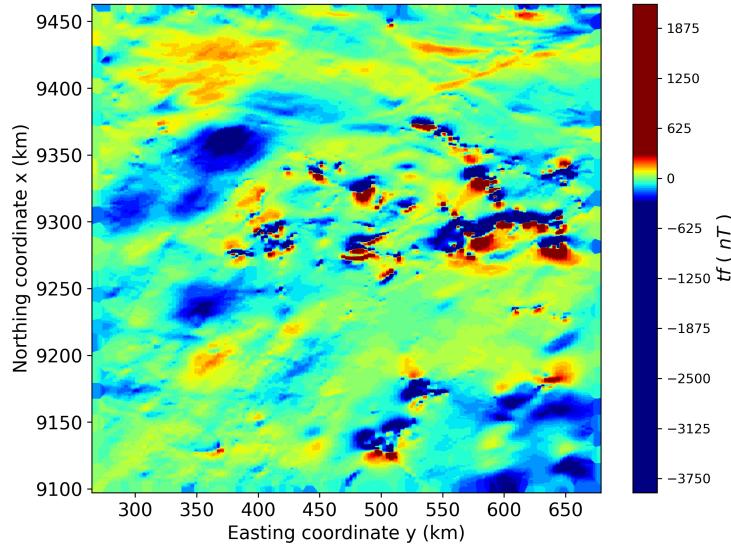
**Figure 9.** True noiseless upward magnetic data at  $z_i = -1400$  m height and predicted data for different methods of the equivalent layer with maximum level of noise. Panel **(A)** is the true upward magnetic data, Panel **(B)** is the classical method, **(C)** is the convolutional, **(D)** is the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



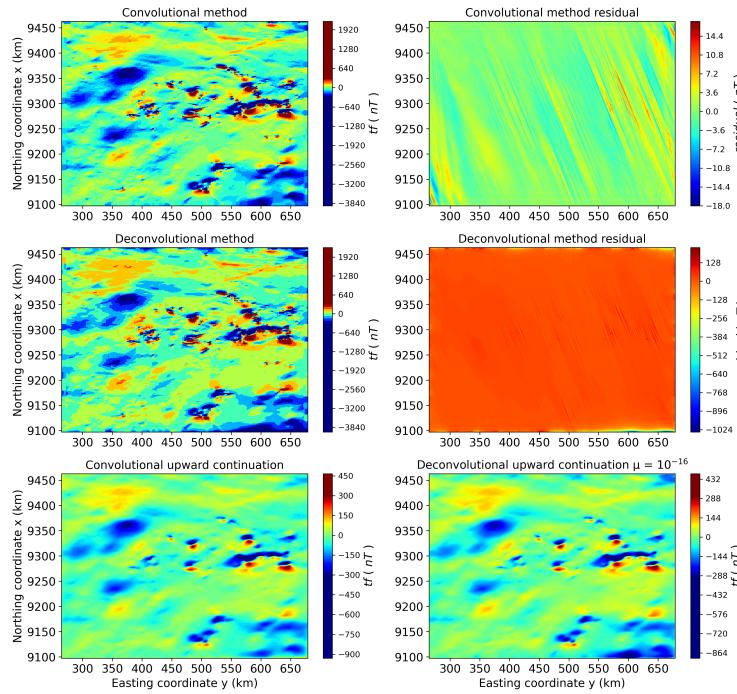
**Figure 10.** Gridded real aerogravimetric data from Carajás, Brazil. A regular grid of  $1,000 \times 500$  is being used, totalizing  $N, M = 500,000$  obsevation points and equivalent sources.



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



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



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