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

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

[1] *Observatório Nacional, Department of Geophysics, Rio de Janeiro, Brasil*
[2] *Universidade do Estado do Rio de Janeiro, Department of Applied Geology, Rio de Janeiro, Brasil*

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

1 FUNDAMENTALS

Let $\mathbf{d}$ be a $D \times 1$ vector, whose i-th element d_i is the observed potential field at the position (x_i, y_i, z_i), $i \in \{1 : D\}$. Consider that d_i can be satisfactorily approximated by a harmonic function

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

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

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

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

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

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

The equivalent-layer technique consists in solving a linear inverse problem to determine a parameter vector $\mathbf{p}$ leading to a predicted data vector $\mathbf{f}$ (equation 3) *sufficiently close to* the observed data vector $\mathbf{d}$, whose i-th element d_i is the observed potential field at (x_i, y_i, z_i). The notion of *closeness* is intrinsically related to the concept of *vector norm* (e.g., Golub and Van Loan, 2013, p. 68) or *measure of length* (e.g., Menke, 2018, p. 41). Because of that, almost all methods for determining $\mathbf{p}$ actually estimate a parameter vector $\tilde{\mathbf{p}}$ minimizing a length measure of the difference between $\mathbf{f}$ and $\mathbf{d}$ (see subsection 1.3). Given an estimate $\tilde{\mathbf{p}}$, it is then possible to compute a potential field transformation

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

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

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

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

1.1 Spatial distribution and total number of equivalent sources

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

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

CRITÉRIOS PARA DEFINIR A PROFUNDIDADE DA CAMADA: DAMPNEY (ESPAÇAMENTO DO GRID) E REIS (ESPAÇAMENTO DAS LINHAS)

1.2 Matrix G

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

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

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

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

and

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

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

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

For all harmonic functions discussed above, the sensitivity matrix $\mathbf{G}$ (equation 3) is always dense. For scattered potential-field data, $\mathbf{G}$ does not have a well-defined structure, regardless of whether the spatial distribution of the equivalent sources is set. Nevertheless, for the particular case in which (i) there is a single equivalent source right below each potential-field datum and (ii) both data and sources rely on planar and regularly spaced grids, Takahashi et al. (2020, 2022) show that $\mathbf{G}$ assumes a block-Toeplitz Toeplitz-block (BTTB) structure. In this case, the product of $\mathbf{G}$ and an arbitrary vector can be efficiently computed via 2D fast Fourier transform as a discrete convolution.

1.3 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 3) can be reparameterized into a $Q \times 1$ vector $\mathbf{q}$ according to:

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

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

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

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

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

which is a combination of particular measures of length given by

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

and

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

where μ is a positive scalar controlling the trade-off between $\Phi(\mathbf{q})$ and $\Theta(\mathbf{q})$; $\mathbf{W}_q$ is a $Q \times Q$ symmetric matrix imposing prior information on $\mathbf{q}$ given by

$$\mathbf{W}_q = \mathbf{H}^\top\mathbf{W}_p\,\mathbf{H}\,, \tag{14}$$

with $\mathbf{W}_p$ being a $P \times P$ symmetric matrix imposing prior information on $\mathbf{p}$; $\bar{\mathbf{q}}$ is a $Q \times 1$ vector of reference values for $\mathbf{q}$ satisfying

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

with $\bar{\mathbf{p}}$ being a $P \times 1$ vector containing reference values for the original parameter vector $\mathbf{p}$; and $\mathbf{W}_d$ is a $D \times D$ symmetric matrix defining the relative importance of each observed datum d_i. After obtaining an estimate $\tilde{\mathbf{q}}$ for the reparameterized parameter vector $\mathbf{q}$ (equation 9) minimizing $\Gamma(\mathbf{q})$ (equation 11), the estimate $\tilde{\mathbf{p}}$ for the original parameter vector (equation 3) is computed by

$$\tilde{\mathbf{p}} = \mathbf{H}\,\tilde{\mathbf{q}}\,. \tag{16}$$

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

$$\boldsymbol{\nabla}\Gamma(\mathbf{q}) = -2\,\mathbf{H}^{\top}\mathbf{G}^{\top}\mathbf{W}_d\,(\mathbf{d} - \mathbf{f}) + 2\,\mu\,\mathbf{W}_q\,(\mathbf{q} - \bar{\mathbf{q}}) \ . \tag{17}$$

Then, by considering that $\boldsymbol{\nabla}\Gamma(\tilde{\mathbf{q}}) = \mathbf{0}$ (equation 17), where $\mathbf{0}$ is a vector of zeros, as well as adding and subtracting the term $\left(\mathbf{H}^{\top}\mathbf{G}^{\top}\mathbf{W}_d\mathbf{G}\,\mathbf{H}\right)\bar{\mathbf{q}}$, we obtain

$$\tilde{\boldsymbol{\delta}}_q = \mathbf{B}\,\tilde{\boldsymbol{\delta}}_d \ , \tag{18}$$

where

$$\tilde{\boldsymbol{\delta}}_q = \tilde{\mathbf{q}} - \bar{\mathbf{q}} \ , \tag{19}$$

$$\tilde{\boldsymbol{\delta}}_d = \mathbf{d} - \mathbf{G}\,\mathbf{H}\,\bar{\mathbf{q}} \ , \tag{20}$$

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

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

$$\mathbf{B} = \mathbf{W}_q^{-1}\,\mathbf{H}^{\top}\mathbf{G}^{\top}\left(\mathbf{G}\,\mathbf{H}\,\mathbf{W}_q^{-1}\,\mathbf{H}^{\top}\mathbf{G}^{\top} + \mu\mathbf{W}_d^{-1}\right)^{-1} \ . \tag{22}$$

Evidently, we have considered that all inverses exist in equations 21 and 22.

Matrix $\mathbf{B}$ defined by equation 21 is commonly used for the cases in which $D > P$, i.e., when there are more data than parameters (overdetermined problems). In this case, we consider that the estimate $\tilde{\mathbf{q}}$ is obtained by solving the following linear system for $\tilde{\boldsymbol{\delta}}_q$ (equation 19):

$$\left(\mathbf{H}^{\top}\mathbf{G}^{\top}\mathbf{W}_d\,\mathbf{G}\,\mathbf{H} + \mu\,\mathbf{W}_q\right)\tilde{\boldsymbol{\delta}}_q = \mathbf{H}^{\top}\mathbf{G}^{\top}\mathbf{W}_d\,\tilde{\boldsymbol{\delta}}_d \ . \tag{23}$$

On the other hand, for the cases in which $D < P$ (underdetermined problems), matrix $\mathbf{B}$ is usually defined according to equation 22. In this case, we consider that the the estimate $\tilde{\mathbf{q}}$ is obtained in two steps, which consists in first solving a linear system for a dummy vector $\mathbf{u}$ and then computing a matrix-vector product as follows:

$$\begin{aligned}\left(\mathbf{G}\,\mathbf{H}\,\mathbf{W}_q^{-1}\,\mathbf{H}^{\top}\mathbf{G}^{\top} + \mu\,\mathbf{W}_d^{-1}\right)\mathbf{u} &= \tilde{\boldsymbol{\delta}}_d \\ \tilde{\boldsymbol{\delta}}_q &= \mathbf{W}_q^{-1}\mathbf{H}^{\top}\mathbf{G}^{\top}\mathbf{u}\end{aligned} \ . \tag{24}$$

After obtaining $\tilde{\boldsymbol{\delta}}_q$ (equations 23 and 24), the estimate $\tilde{\mathbf{q}}$ is computed with equation 19.

2 COMPUTATIONAL STRATEGIES

COMEÇAR EXPLICANDO AS LIMITAÇÕES DA SOLUÇÃO OBTIDA PELA FORMULAÇÃO GERAL

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. A flop is a floating point addition, subtraction, multiplication or division (Golub and Van Loan, 2013, p. 12–14).

To investigate the efficiency of equivalent-layer methods, we consider how they:

(i) set up the linear system (equations 23 and 24);
(ii) solve the linear system (equations 23 and 24);
(iii) perform potential-field transformations (equation 4).

We focus on the overall strategies used by the selected methods.

2.1 Notation for subvectors and submatrices

Here, we use a notation inspired on that presented by (Van Loan, 1992, p. 4) to represent subvectors and submatrices. Subvectors of $\mathbf{d}$, for example, are specified by $\mathbf{d}[\mathbf{i}]$, where $\mathbf{i}$ is a list of integer numbers that "pick out" the elements of $\mathbf{d}$ forming the subvector $\mathbf{d}[\mathbf{i}]$. For example, $\mathbf{i} = (1, 6, 4, 6)$ gives the subvector $\mathbf{d}[\mathbf{i}] = [d_1 \;\, d_6 \;\, d_4 \;\, d_6]^\top$. Note that the list $\mathbf{i}$ of indices may be sorted or not and it may also have repeated indices. The list may also have a single element $\mathbf{i} = (i)$, which results in the i-th element $d_i \equiv \mathbf{d}[i]$ of $\mathbf{d}$. We may also define regular lists of indices by using the colon notation. For example,

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

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

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

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

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

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

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

2.2 Moving window

The initial approach to enhance the computational efficiency of the equivalent-layer technique is commonly denoted *moving window* and involves first splitting the observed data d_i, $i \in \{1 : D\}$, into M overlapping subsets (or data windows) formed by D^m data each, $m \in \{1 : M\}$. The data inside the 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. 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 a set of P equivalent sources with scalar physical property p_j, $j \in \{1 : P\}$, and also split them into M overlapping subsets (or source windows) formed by P^m data each, $m \in \{1 : M\}$. The sources inside the m-th window have indices defined by an integer list $\mathbf{j}^m$ having P^m elements. Each source window has a $P^m \times 1$ parameter vector $\mathbf{p}^m$ and is located right below the corresponding m-th data window. Then, each $\mathbf{d}^m \equiv \mathbf{d}[\mathbf{i}^m]$ is approximated by

$$\mathbf{f}^m = \mathbf{G}^m \mathbf{p}^m \ , \tag{25}$$

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

Leão and Silva (1989) presented a pioneer work using the moving-window approach. Their method requires a regularly-spaced grid of observed data on a horizontal plane z_0. The data windows are defined by square local grids of $\sqrt{D'} \times \sqrt{D'}$ adjacent points, all of them having the same number of points D'. The equivalent sources in the m-th data window are located below the observation plane, at a constant vertical distance Δz_0. They are arranged on a regular grid of $\sqrt{P'} \times \sqrt{P'}$ adjacent points following the same grid pattern of the observed data. The local grid of sources for all data windows have the same number of elements P'. Besides, they are vertically aligned, but expands the limits of their corresponding data windows, so that $D' < P'$. Because of this spatial configuration of observed data and equivalent sources, we have that $\mathbf{G}^m = \mathbf{G}'$ (equation 25) for all data windows (i.e., $\forall\, m \in \{1 : M\}$), where $\mathbf{G}'$ is a $D' \times P'$ constant matrix.

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

$$t_c^m = \left(\mathbf{G}'\mathbf{a}'\right)^\top \left[\mathbf{G}' \left(\mathbf{G}'\right)^\top + \mu\, \mathbf{I}_{D'}\right]^{-1} \mathbf{d}^m \ , \quad m \in \{1 : M\} \ , \tag{26}$$

where $\mathbf{I}_{D'}$ is the identity matrix of order D' and $\mathbf{a}'$ is a $P' \times 1$ vector with elements computed by equation 5 by using all equivalent sources in the m-th subset and only the coordinate of the central point in the m-th data window. Due to the presumed spatial configuration of the observed data and equivalent sources, $\mathbf{a}'$ is the same for all data windows. Note that equation 26 combines the potential-field transformation (equation 4) with the solution of the undetermined problem (equation 24) for the particular case in which $\mathbf{H} = \mathbf{W}_p = \mathbf{I}_{P'}$ (equations 9 and 14), $\mathbf{W}_d = \mathbf{I}_{D'}$ (equation 12), $\bar{p} = \mathbf{0}$ (equation 15), where $\mathbf{I}_{P'}$ and $\mathbf{I}_{D'}$ are identity matrices of order P' and D', respectively, and $\mathbf{0}$ is a vector of zeros.

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

Soler and Uieda (2021) generalized the method proposed by Leão and Silva (1989) for irregularly spaced data on an undulating surface. A direct consequence of this generalization is that a different submatrix $\mathbf{G}^m \equiv \mathbf{G}[\mathbf{i}^m, \mathbf{j}^m]$ (equation 25) must be computed for each window. Differently from Leão and Silva

Algorithm 1: Generic pseudo-code for the method proposed by Leão and Silva (1989).

Initialization :
Set the indices $\mathbf{i}^m$ for each data window, $m \in \{1 : M\}$;
Set the indices $\mathbf{j}^m$ for each source window, $m \in \{1 : M\}$;
Set the constant depth $z_0 + \Delta z_0$ for all equivalent sources ;
Compute the vector $\mathbf{a}'$ associated with the desired potential-field transformation ;
Compute the matrix $\mathbf{G}'$;
Compute $(\mathbf{G}'\mathbf{a}')^\top \left[\mathbf{G}' \, (\mathbf{G}')^\top + \mu \, \mathbf{I}_{D'}\right]^{-1}$;
$m = 1$;
while $m < M$ **do**
  Compute t_c^m (equation 26) ;
  $m \leftarrow m + 1$;
end

(1989), Soler and Uieda (2021) store the computed $\tilde{\mathbf{p}}^m$ for all windows and subsequently use them to obtain a desired potential-field transformation (equation 4) as the superposed effect of all windows. The estimated $\tilde{\mathbf{p}}^m$ for all windows are combined to form a single $P \times 1$ vector $\tilde{\mathbf{p}}$, which is an estimate for original parameter vector $\mathbf{p}$ (equation 3). For each data window, Soler and Uieda (2021) solve an overdetermined problem (equation 23) for $\tilde{\mathbf{p}}^m$ by using $\mathbf{H} = \mathbf{W}_p = \mathbf{I}_{P^m}$ (equations 9 and 14), $\mathbf{W}_d^m$ (equation 12) equal to a diagonal matrix of weights for the data inside the m-th window and $\bar{p} = \mathbf{0}$ (equation 15), so that

$$\left[(\mathbf{G}^m)^\top \, \mathbf{W}_d^m \, \mathbf{G}^m + \mu \, \mathbf{I}_{P'}\right] \tilde{\mathbf{p}}^m = (\mathbf{G}^m)^\top \, \mathbf{W}_d^m \, \mathbf{d}^m \, . \tag{27}$$

The overall steps of their method are defined by the Algorithm 2. Note that Algorithm 2 starts with a residuals vector $\mathbf{r}$ that is iteratively updated. At each iteration, the potential field predicted a source window is computed at all observation points and removed from the residuals vector $\mathbf{r}$.

Algorithm 2: Generic pseudo-code for the method proposed by Soler and Uieda (2021).

Initialization :
Set the indices $\mathbf{i}^m$ for each data window, $m \in \{1 : M\}$;
Set the indices $\mathbf{j}^m$ for each source window, $m \in \{1 : M\}$;
Set the depth of all equivalent sources ;
Set a $D \times 1$ residuals vector $\mathbf{r} = \mathbf{d}$;
Set a $P \times 1$ vector $\tilde{\mathbf{p}} = \mathbf{0}$;
$m = 1$;
while $m < M$ **do**
  Set the matrix $\mathbf{W}_d^m$;
  Compute the matrix $\mathbf{G}^m$;
  Compute $\tilde{\mathbf{p}}^m$ (equation 27) ;
  $\tilde{\mathbf{p}}[\mathbf{j}^m] \leftarrow \tilde{\mathbf{p}}[\mathbf{j}^m] + \tilde{\mathbf{p}}^m$;
  $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{G}[:, \mathbf{j}^m] \, \tilde{\mathbf{p}}^m$;
  $m \leftarrow m + 1$;
end

2.3 Column update

Cordell (1992) proposed a computational strategy that was later used by Guspí and Novara (2009) and relies on first defining one equivalent source located right below each observed data d_i, $i \in \{1 : D\}$, at a vertical coordinate $z_i + \Delta z_i$, where Δz_i is proportional to the distance from the i-th observation point (x_i, y_i, z_i) to its closest neighbor. The second step consists in updating the physical property p_j of a given equivalent source, $j \in \{1 : D\}$ and remove its predicted potential field from the observed data vector $\mathbf{d}$, producing a residuals vector $\mathbf{r}$. Then, the same procedure is repeated for other sources with the purpose of iteratively updating $\mathbf{r}$ and the $D \times 1$ parameter vector $\mathbf{p}$ containing the physical property of all equivalent sources. At the end, the algorithm produces an estimate $\tilde{\mathbf{p}}$ for the parameter vector yielding a predicted potential field $\mathbf{f}$ (equation 3) satisfactorily fitting the observed data $\mathbf{d}$ according to a given criterion. Note that the method proposed by Cordell (1992) iteratively solves the linear $\mathbf{G}\tilde{\mathbf{p}} \approx \mathbf{d}$ with a $D \times D$ matrix $\mathbf{G}$. At each iteration, only a single column of $\mathbf{G}$ (equation 3) is used. An advantage of this *column-update approach* is that the full matrix $\mathbf{G}$ is never stored.

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

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

Initialization:
Compute a $D \times 1$ vector $\mathbf{\Delta z}$ whose i-th element Δz_i is a vertical distance controlling the depth of the i-th equivalent source, $i \in \{1 : D\}$;
Set a tolerance ϵ ;
Set a maximum number of iteration `ITMAX` ;
Set a $D \times 1$ residuals vector $\mathbf{r} = \mathbf{d}$;
Set a $D \times 1$ vector $\tilde{\mathbf{p}} = \mathbf{0}$;
Define the maximum absolute value $r_{\mathtt{max}}$ in $\mathbf{r}$;
$m = 1$;
while $(r_{\mathtt{max}} > \epsilon)$ **and** $(m < \mathtt{ITMAX})$ **do**
    Define the coordinates $(x_{\mathtt{max}}, y_{\mathtt{max}}, z_{\mathtt{max}})$ and index $i_{\mathtt{max}}$ of the observation point associated with $r_{\mathtt{max}}$;
    $\tilde{\mathbf{p}}[i_{\mathtt{max}}] \leftarrow \tilde{\mathbf{p}}[i_{\mathtt{max}}] + (a_{\mathtt{max}}\, \mathbf{\Delta z}[i_{\mathtt{max}}])$;
    $\mathbf{r} \leftarrow \mathbf{r} - (\mathbf{G}[:, i_{\mathtt{max}}]\, \tilde{\mathbf{p}}[i_{\mathtt{max}}])$;
    Define the new $r_{\mathtt{max}}$ in $\mathbf{r}$;
    $m \leftarrow m + 1$;
end

2.4 Row update

Mendonça and Silva (1994) proposes an algebraic reconstruction technique (ART) (e.g., van der Sluis and van der Vorst, 1987, p. 58) to estimate a parameter vector $\tilde{\mathbf{p}}$ for a regular grid of P equivalent sources on a horizontal plane z_0. Such methods iterate on the linear system rows to estimate corrections for the parameter vector, which may substantially save computer time and memory required to compute and store the full linear system matrix along the iterations. The convergence of such *row-update methods* depends on the linear system condition. The main advantage of such methods is not computing and storing the full linear system matrix, but iteratively using its rows. The particular ART method proposed by Mendonça and Silva (1994) considers that

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_e \\ \mathbf{d}_r \end{bmatrix} , \quad \mathbf{G} = \begin{bmatrix} \mathbf{G}_e \\ \mathbf{R}_r \end{bmatrix} , \tag{28}$$

where $\mathbf{d}_e$ and $\mathbf{d}_r$ are $D_e \times 1$ and $D_r \times 1$ vectors and $\mathbf{G}_e$ and $\mathbf{G}_r$ are $D_e \times P$ and $D_r \times P$ matrices, respectively. Mendonça and Silva (1994) designate $\mathbf{d}_e$ and $\mathbf{d}_r$ as, respectively, *equivalent* and *redundant* data. With the exception of a normalization strategy, Mendonça and Silva (1994) calculate a $P \times 1$ estimated parameter vector $\tilde{\mathbf{p}}$ by solving an underdetermined problem (equation 24) involving only the equivalent data $\mathbf{d}_e$ (equation 28) for the particular case in which $\mathbf{H} = \mathbf{W}_p = \mathbf{I}_P$ (equations 9 and 14), $\mathbf{W}_d = \mathbf{I}_{D_e}$ (equation 12) and $\bar{p} = \mathbf{0}$ (equation 15), which results in

$$\begin{aligned} \left(\mathbf{F} + \mu \, \mathbf{I}_{D_e}\right) \mathbf{u} &= \mathbf{d}_e \\ \tilde{\mathbf{p}} &= \mathbf{G}_e^{\top} \mathbf{u} \end{aligned} , \tag{29}$$

where $\mathbf{F}$ is a $P \times P$ matrix that replaces $\mathbf{G}_e \, \mathbf{G}_e^{\top}$. Mendonça and Silva (1994) presume that the estimated parameter vector $\tilde{\mathbf{p}}$ obtained from equation 29 leads to a $D_r \times 1$ residuals vector

$$\mathbf{r} = \mathbf{d}_r - \mathbf{G}_r \tilde{\mathbf{p}} \tag{30}$$

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

The overall method of Mendonça and Silva (1994) is defined by Algorithm 4. It is important noting that the number D_e of equivalent data in $\mathbf{d}_e$ increases by one per iteration, which means that the order of the linear system in equation 29 also increases by one at each iteration. Those authors also propose a computational strategy based on Cholesky decomposition for efficiently updating $\left(\mathbf{F} + \mu \, \mathbf{I}_{D_e}\right)$ at a given iteration (line 16 in Algorithm 4) by computing only its new elements with respect to those computed in the previous iteration.

2.5 Reparameterization

Estimate $\tilde{\mathbf{p}}$ by solving an overdetermined problem (equation 23) with $\mathbf{W}_p = \mathbf{I}_P$ (equation 14), $\mathbf{W}_d = \mathbf{I}_D$ (equation 12).

The reference vector $\bar{q}$ (equation 15) is set equal to $\mathbf{0}$ for direct solvers or computed iteratively for iterative solvers.

The difference between methods relies on how the reparameterization matrix $\mathbf{H}$ is defined.

PAREI AQUI - vou tentar usar o q barra para descrever os metodos iterativos

Barnes and Lumley (2011)

Oliveira Jr. et al. (2013)

Mendonça (2020)

2.6 Wavelet compression

Li and Oldenburg (2010)

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

Initialization :

Set a regular grid of P equivalent sources at a horizontal plane z_0 ;
Set a tolerance ϵ ;
Set a $D \times 1$ residuals vector $\mathbf{r} = \mathbf{d}$;
Define the maximum absolute value $r_{\mathtt{max}}$ in $\mathbf{r}$;
Define the index $i_{\mathtt{max}}$ of $r_{\mathtt{max}}$;
Define the list of indices $\mathbf{i}_r$ of the remaining data in $\mathbf{r}$;
Define $\mathbf{d}_e = \mathbf{d}[i_{\mathtt{max}}]$;
Compute $(\mathbf{F} + \mu \, \mathbf{I}_{D_e})$ and $\mathbf{G}_e$;
Compute $\tilde{\mathbf{p}}$ (equation 29) ;
Compute $\mathbf{r} = \mathbf{d}[\mathbf{i}_r] - \mathbf{G}[\mathbf{i}_r, :] \, \tilde{\mathbf{p}}$;
Define the maximum absolute value $r_{\mathtt{max}}$ in $\mathbf{r}$;
while $(r_{\mathtt{max}} > \epsilon)$ **do**
    Define the index $i_{\mathtt{max}}$ of $r_{\mathtt{max}}$;
    Define the list of indices $\mathbf{i}_r$ of the remaining elements in $\mathbf{r}$;
    $\mathbf{d}_e \leftarrow \begin{bmatrix} \mathbf{d}_e \\ \mathbf{d}[i_{\mathtt{max}}] \end{bmatrix}$;
    Update $(\mathbf{F} + \mu \, \mathbf{I}_{D_e})$ and $\mathbf{G}_e$;
    Compute $\tilde{\mathbf{p}}$ (equation 29) ;
    Compute $\mathbf{r} = \mathbf{d}[\mathbf{i}_r] - \mathbf{G}[\mathbf{i}_r, :] \, \tilde{\mathbf{p}}$;
    Define the maximum absolute value $r_{\mathtt{max}}$ in $\mathbf{r}$;
end

2.7 Iterative methods using the original G

Ideia: descrever o conjugate gradient e depois as diferenças dos outros métodos iterativos

Xia and Sprowl (1991)

Xia et al. (1993)

Siqueira et al. (2017)

Jirigalatu and Ebbing (2019)

2.8 Discrete convolution

Takahashi et al. (2020)

Takahashi et al. (2022)

3 TEXTO ANTIGO

Leão and Silva (1989) reduced the total processing time and memory usage of equivalent-layer technique by means of a moving data-window scheme. A small moving data window with N_w observations and a small equivalent layer with M_w equivalent sources ($M_w > N_w$) located below the observations are established. For each position of a moving-data window, Leão and Silva (1989) estimate a stable solution

$\mathbf{p_w}^*$ by using a data-space approach with the zeroth-order Tikhonov regularization (**?**), i.e.,

$$\left(\mathbf{A_w}\mathbf{A_w}^\top + \mu\mathbf{I}\right)\mathbf{w} \;=\; \mathbf{d_w}^o\,, \tag{31a}$$

$$\mathbf{A_w}^\top\mathbf{w} \;=\; \mathbf{p_w}^*\,, \tag{31b}$$

where $\mathbf{w}$ is a dummy vector, μ is a regularizing parameter, $\mathbf{d_w}^o$ is an N_w-dimensional vector containing the observed potential-field data, $\mathbf{A_w}$ is an $N_w \times M_w$ sensitivity matrix related to a moving-data window, $\mathbf{I}$ is an identity matrix of order N_w and the superscript $\top$ stands for a transpose. After estimating an $M_w \times 1$ parameter vector $\mathbf{p_w}^*$ (equation 31b) the desired transformation of the data is only calculated at the central point of each moving-data window, i.e.:

$$\hat{t}_k = \mathbf{t}_k^\top\,\mathbf{p_w}^*\,, \tag{32}$$

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

By shifting the moving-data window with a shift size of one data spacing, a new position of a data window is set up. Next, the aforementioned process (equations 31b and 32) is repeated for each position of a moving-data window, until the entire data have been processed. Hence, instead of solving a large inverse problem, Leão and Silva (1989) solve several much smaller ones.

To reduce the size of the linear system to be solved, Soler and Uieda (2021) adopted the same strategy proposed, originally, by Leão and Silva (1989) of using a small moving-data window sweeping the whole data. In Leão and Silva (1989), a moving-data window slides to the next adjacent data window following a sequential movement, the predicted data is calculated inside the data window and the desired transformation are only calculated at the center of the moving-data window. Unlike Leão and Silva (1989), Soler and Uieda (2021) do not adopt a sequential order of the data windows; rather, they adopt a randomized order of windows in the iterations of the gradient-boosting algorithm (**?**, **?** and **?**). The gradient-boosting algorithm in Soler and Uieda (2021) estimates a stable solution using the data and the equivalent sources that fall within a moving-data window; however, it calculates the predicted data and the residual data in the whole survey data. Next, the residual data that fall within a new position of the data window is used as input data to estimate a new stable solution within the data window which in turn is used to calculated a new predicted data and a new residual data in the whole survey data. Finally, unlike Leão and Silva (1989), in Soler and Uieda (2021) neither the data nor the equivalent sources need to be 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.

3.0.1 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). Iteractively, 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.

3.0.2 The wavelet compression and lower-dimensional subspace

For large data sets, the sensitivity matrix $\mathbf{A}$ (equation 3) is a drawback in applying the equivalent-layer technique because it is a large and dense matrix.

? transformed a large and full sensitivity matrix into a sparse one by using fast wavelet transforms. In the wavelet domain, **?** 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., $\mathbf{W_2}^\top$).

$$\tilde{\mathbf{A}} = \mathbf{W_2}\,\mathbf{A}\,\mathbf{W_2}^\top\,, \tag{33}$$

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 $\tilde{\mathbf{A}}_\mathbf{s}$ 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 **?**, the original sensitivity matrix $\mathbf{A}$ is high compressed resulting in a sparce matrix $\tilde{\mathbf{A}}_\mathbf{s}$ with a few percent of nonzero elements and the the inverse problem is solved in the wavelet domain by using $\tilde{\mathbf{A}}_\mathbf{s}$ 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

$$\tilde{\mathbf{A}}_\mathbf{L}^\top\,\tilde{\mathbf{A}}_\mathbf{L}\,\tilde{\mathbf{p}}_\mathbf{L}^*\, = \tilde{\mathbf{A}}_\mathbf{L}^\top\,\tilde{\mathbf{d}}^o\,, \tag{34}$$

where $\tilde{\mathbf{p}}_\mathbf{L}^*$ is obtained by solving the linear system given by equation 34,

$$\tilde{\mathbf{A}}_\mathbf{L} \;=\; \tilde{\mathbf{A}}_\mathbf{s}\,\tilde{\mathbf{L}}^{-\mathbf{1}}, \tag{35a}$$

$$\tilde{\mathbf{p}}_\mathbf{L} \;=\; \tilde{\mathbf{L}}\tilde{\mathbf{p}}, \tag{35b}$$

$$\tilde{\mathbf{d}}^o \;=\; \mathbf{W_2}\,\mathbf{d}^o, \tag{35c}$$

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

$$\tilde{\mathbf{p}} = \tilde{\mathbf{L}}^{-1}\,\tilde{\mathbf{p}}_\mathbf{L}^*\,, \tag{36}$$

and

$$\mathbf{p} = \mathbf{W_2}\,\tilde{\mathbf{p}}\,. \tag{37}$$

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

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

? overcame the solution of intractable large-scale equivalent-layer problem by using the subspace method (e.g., **?**, **?**; **?**, **?**; **?**, **?**; **?**, **?**). The subspace method reduces the dimension of the linear system of equations to be solved. Given a higher-dimensional space (e.g., M-dimensional model space, $\mathbb{R}^M$), there exists many lower-dimensional subspaces (e.g., Q-dimensional subspace) of $\mathbb{R}^M$. The linear inverse problem related to the equivalent-layer technique consists in finding an M-dimension parameter vector $\mathbf{p} \in \mathbb{R}^M$ which adequately fits the potential-field data. The subspace method looks for a parameter vector who lies in a Q-dimensional subspace of $\mathbb{R}^M$ which, in turn, is spanned by a set of Q vectors $\mathbf{v}_i = 1, ..., Q$, where $\mathbf{v}_i \in \mathbb{R}^M$ In matrix notation, the parameter vector in the subspace method can be written as

$$\mathbf{p} = \mathbf{V}\,\boldsymbol{\alpha}\,, \tag{38}$$

where $\mathbf{V}$ is an $M \times Q$ matrix whose columns $\mathbf{v}_i = 1, ..., Q$ form a basis vectors for a subspace Q of $\mathbb{R}^M$. In equation 38, the parameter vector $\mathbf{p}$ is defined as a linear combination in the space spanned by Q basis vectors $\mathbf{v}_i = 1, ..., Q$ and $\boldsymbol{\alpha}$ is a Q-dimensional unknown vector to be determined. The main advantage of the subspace method is that the linear system of M equations in M unknowns to be originally solved is reduced to a new linear system of Q equations in Q unknowns which requires much less computational effort since $Q << M$, i.e.:

$$\mathbf{V}^\top\,\mathbf{A}^\top\,\mathbf{A}\,\mathbf{V}\,\boldsymbol{\alpha}^* \;=\; \mathbf{V}^\top\,\mathbf{d}^{\boldsymbol{o}}\,. \tag{39}$$

To avoid the storage of matrices $\mathbf{A}$ and $\mathbf{V}$, **?** evaluates an element of the matrix $\mathbf{AV}$ by calculating the dot product between the row of matrix $\mathbf{A}$ and the column of the matrix $\mathbf{B}$. After estimating $\boldsymbol{\alpha}^*$ (equation 39) belonging to a Q-dimensional subspace of $\mathbb{R}^M$, the distribution over the equivalent layer $\mathbf{p}$ in the $\mathbb{R}^M$ is obtained by applying equation 38. The choice of the Q basis vectors $\mathbf{v}_i = 1, ..., Q$ (equation 38) in the subspace method is not strict. **?**, for example, chose the eigenvectors yielded by applying the singular value decomposition of the matrix containing the gridded data set. The number of eigenvectors used to form basis vectors will depend on the singular values.

The proposed subspace method for solving large-scale equivalent-layer problem by **?** was applied to estimate the mass excess or deficiency caused by causative gravity sources.

3.0.3 The quadtree discretization

To make the equivalent-layer technique tractable, **?** also transformed the dense sensitivity matrix $\mathbf{A}$ (equation 3) into a sparse matrix. In **?**, a sparce version of the sensitivity matrix is achived by grouping equivalent sources (e.g., they used prisms) distant from an observation point together to form a larger prism or larger block. Each larger block has averaged physical properties and averaged top- and bottom-surfaces of the grouped smaller prisms (equivalent sources) that are encompassed by the larger block. The authors called it the 'larger averaged block' and the essence of their method is the reduction in the number of equivalent sources, which means a reduction in the number of parameters to be estimated implying in model dimension reduction.

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

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

3.0.4 The reparametrization of the equivalent layer

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

$$\mathbf{p}^k = \mathbf{B}^k \, \mathbf{c}^k \, , \qquad k \, = \, 1 \ldots Q \, , \tag{40}$$

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

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

$$\left(\mathbf{B}^\top \, \mathbf{A}^\top \, \mathbf{A} \, \mathbf{B} \, + \, \mu \mathbf{I} \right) \, \mathbf{c}^* \, = \, \mathbf{B}^\top \, \mathbf{A}^\top \, \mathbf{d}^o \, , \tag{41}$$

where μ is a regularizing parameter, $\mathbf{c}^*$ is an estimated H-dimensional vector containing all coefficients describing all polynomial functions within all equivalent-source windows which compose the entire 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 such that the main-diagonal blocks are $\mathbf{B}^k$ matrices (equation 40) and all off-diagonal blocks are zero matrices. For ease of the explanation of equation 41, we keep only the zeroth-order Tikhonov regularization and omitting the first-order Tikhonov regularization (**?**) which was also used by Oliveira Jr. et al. (2013).

The main advantage of the PEL is solve H-dimensional system of equations (equation 41), where H totalizes the number of polynomial coefficients composing all equivalent-source windows, requiring a lower computational effort since $H <<< N$. To avoid the storage of matrices $\mathbf{A}$ and $\mathbf{B}$, Oliveira Jr. et al. (2013) evaluate an element of the matrix $\mathbf{AB}$ by calculating the dot product between the row of matrix $\mathbf{A}$ and the column of the matrix $\mathbf{B}$. After estimating all polynomial coefficients of all windows, the estimated coefficients ($\mathbf{c}^*$ in equation 41) are transformed into a single physical-property distribution encompassing the entire equivalent layer.

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

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

3.0.5 The iterative scheme without solving a linear system

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

Following this class of methods of iterative equivalent-layer technique that does not solve linear systems, Siqueira et al. (2017) developed a fast iterative equivalent-layer technique for processing gravity data in which the sensitivity matrix $\mathbf{A}$ (equation 3) is replaced by a diagonal matrix $N \times N$, i.e.:

$$\tilde{\tilde{\mathbf{A}}} = 2\,\pi\,\gamma\,\mathbf{\Delta S^{-1}}\,, \tag{42}$$

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

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

$$\tilde{\tilde{\mathbf{A}}}^{\top}\,\tilde{\tilde{\mathbf{A}}}\mathbf{\Delta}\,\hat{\mathbf{p}}^{k} \;=\; \tilde{\tilde{\mathbf{A}}}^{\top}\,\mathbf{r}^{k}\,, \tag{43}$$

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

$$r_i^k = d_i^o \;-\; d_i^k\,. \tag{44}$$

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

Because $\tilde{\tilde{\mathbf{A}}}$, in equation 43, is a diagonal matrix (equation 42), the parameter correction estimate is directly calculated without solving system of linear equations, and thus, an ith element of $\mathbf{\Delta}\,\hat{\mathbf{p}}^k$ is directly calculated by

$$\Delta\hat{p}^k{}_i = \frac{\Delta s_i \, r_i^k}{2\,\pi\,\gamma} \,. \tag{45}$$

The mass distribution over the equivalent layer is updated by:

$$\hat{p}_i^{k+1} = \hat{p}_i^k \;+\; \Delta\hat{p}^k{}_i \,. \tag{46}$$

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

$$p_i^o = \frac{\Delta s_i \, d_i^o}{2\,\pi\,\gamma} \,. \tag{47}$$

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

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

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \omega \left[\mathbf{A_1}^\top(\mathbf{d_1} - \mathbf{M}\mathbf{A_1}\mathbf{p}_k) + \mathbf{A_2}^\top(\mathbf{d_2} - \mathbf{M}\mathbf{A_2}\mathbf{p}_k)\right] \,, \tag{48}$$

where ω 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 corresponding gravity gradient kernels. Jirigalatu and Ebbing (2019) applied their method for processing two horizontal curvature components of Falcon airborne gravity gradient.

3.0.6 The convolutional equivalent layer with BTTB matrices

? (**?**, **?**) introduced the convolutional equivalent layer for gravimetric and magnetic data processing, respectively.

? demonstrated that the sensitivity matrix $\mathbf{A}$ (equation 3) associated with a planar equivalent layer formed by a set of point masses, each one directly beneath each observation point and considering a regular grid of observation points at a constant height has a symmetric block-Toeplitz Toeplitz-block (BTTB) structure. A symmetric BTTB matrix has, at least, two attractive properties. The first one is that it can be defined by using only the elements forming its first column (or row). The second attractive property is that any BTTB matrix can be embedded into a symmetric Block-Circulat Circulant-Block (BCCB) matrix. This means that the full sensitivity matrix $\mathbf{A}$ (equation 3) can be completely reconstruct by using the first column of the BCCB matrix only. In what follows, **?** computed the forward modeling by using only a single equivalent source. Specifically, it is done by calculating the eigenvalues of the BCCB matrix that can be efficiently computed by using only the first column of the BCCB matrix via 2D fast Fourier transform (2D FFT). By comparing with the classic approach in the Fourier domain, the convolutional equivalent layer for gravimetric data processing proposed by **?** performed upward- and downward-continue gravity data with a very small border effects and noise amplification.

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

The matrix-vector product in **?** (**?**, **?**) (e.g., $\mathbf{d} = \mathbf{A}\mathbf{p}$, such as in equation 3) is the main issue to be solved. To solve it efficiently, these authors involked the auxiliary linear system

$$\mathbf{w} = \mathbf{C}\mathbf{v} \, , \tag{49}$$

where $\mathbf{w}$ and $\mathbf{v}$ are, respectively, vectors of data and parameters completed by zeros and $\mathbf{C}$ is a BCCB matrix formed by $2Q \times 2Q$ blocks, where each block $\mathbf{C}_q$, $q = 0, \ldots, Q-1$, is a $2P \times 2P$ circulant matrix. The first column of $\mathbf{C}$ is obtained by rearranging the first column of the sensitivity matrix $\mathbf{A}$ (equation 3). Because a BCCB matrix is diagonalized by the 2D unitary discrete Fourier transform (DFT), $\mathbf{C}$ can be written as

$$\mathbf{C} = \left(\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}\right)^{*} \boldsymbol{\Lambda} \left(\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}\right) \, , \tag{50}$$

where the symbol "$\otimes$" denotes the Kronecker product (**?**), $\mathbf{F}_{2Q}$ and $\mathbf{F}_{2P}$ are the $2Q \times 2Q$ and $2P \times 2P$ unitary DFT matrices (**?**, p. 31), respectively, the superscritpt "$*$" denotes the complex conjugate and $\boldsymbol{\Lambda}$ is a $4QP \times 4QP$ diagonal matrix containing the eigenvalues of $\mathbf{C}$. Due to the diagonalization of the matrix $\mathbf{C}$, the auxiliary system (equation 49) can be rewritten by using equation 50 and premultiplying both sides of the result by $\left(\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}\right)$, i.e.,

$$\boldsymbol{\Lambda} \left(\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}\right) \mathbf{v} = \left(\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}\right) \mathbf{w} \, . \tag{51}$$

By applying the vec-operator (**?**) to both sides of equation 51, by premultiplying both sides of the result by $\mathbf{F}^{*}_{2Q}$ and then postmultiplying both sides of the result by $\mathbf{F}^{*}_{2P}$

$$\mathbf{F}^{*}_{2Q} \left[\mathbf{L} \circ \left(\mathbf{F}_{2Q} \, \mathbf{V} \, \mathbf{F}_{2P}\right)\right] \mathbf{F}^{*}_{2P} = \mathbf{W} \, , \tag{52}$$

where "$\circ$" denotes the Hadamard product (**?**, p. 298) and $\mathbf{L}$, $\mathbf{V}$ and $\mathbf{W}$ are $2Q \times 2P$ matrices obtained by rearranging, along their rows, the elements forming the diagonal of matrix $\boldsymbol{\Lambda}$, vector $\mathbf{v}$ and vector $\mathbf{w}$, respectively. The left side of equation 52 contains the 2D Inverse Discrete Fourier Transform (IDFT) of the term in brackets, which in turn represents the Hadamard product of matrix $\mathbf{L}$ and the 2D DFT of matrix $\mathbf{V}$. Matrix $\mathbf{L}$ contains the eigenvalues of $\boldsymbol{\Lambda}$ (equation 50) and can be efficiently computed by using only the first column of the BCCB matrix $\mathbf{C}$ (equation 49).

Actually, in **?** (**?**, **?**) a fast 2D discrete circular convolution (Van Loan, 1992) is used to process very large gravity and magnetic datasets efficiently. The convolutional equivalent layer was applied to perform upward continuation of large magnetic datasets. Compared to the classical Fourier approach, **?**'s (**?**) method produces smaller border effects without using any padding scheme.

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

3.0.7 The deconvolutional equivalent layer with BTTB matrices

To avoid the iterations of the conjugate gradient method in **?**, we can employ the deconvolution process. Equation 52 shows that estimate the matrix $\mathbf{V}$, containing the elements of parameter vector $\mathbf{p}$, is a inverse problem that could be solved by deconvolution. From equation 52, the matrix $\mathbf{V}$ can be obtain by deconvolution, i.e.

$$\mathbf{V} \;=\; \mathbf{F}_{2Q}^{*} \left[\frac{\left(\mathbf{F}_{2Q} \mathbf{W}\, \mathbf{F}_{2P}\right)}{\mathbf{L}} \right] \mathbf{F}_{2P}^{*} \,. \tag{53}$$

Equation 53 shows that the parameter vector (in matrix $\mathbf{V}$) can be theoretically obtain by dividing each potential-field observations (in matrix $\mathbf{W}$) by each eigenvalues (in matrix $\mathbf{L}$). Hence, the parameter vector is constructed by element-by-element division of data by eigenvalues.

However, the deconvolution often is extremely unstable. This means that a small change in data can lead to an enormous change in the estimated parameter. Hence, equation 53 requires regularization to be useful. We usede wiener deconvolution to obtain a stable solution, i.e.,

$$\mathbf{V} \;=\; \mathbf{F}_{2Q}^{*} \left[\left(\mathbf{F}_{2Q} \mathbf{W}\, \mathbf{F}_{2P}\right) \frac{\mathbf{L}^{*}}{\left(\mathbf{L}\, \mathbf{L}^{*} + \mu\right)} \right] \mathbf{F}_{2P}^{*} \,, \tag{54}$$

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

3.1 Solution stability

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

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

Next, for each noise-corrupted data $\mathbf{d}_{\ell}^{\mathbf{o}}$ and estimated solution $\hat{\mathbf{p}}_{\ell}$, the ℓth model perturbation δp_{ℓ} and the ℓth data perturbation δd_{ℓ} are, respectively, evaluated by

$$\delta p_{\ell} = \frac{\parallel \hat{\mathbf{p}}_{\ell} - \mathbf{p} \parallel_2}{\parallel \mathbf{p} \parallel_2}, \quad \ell = 1, ..., D, \tag{55}$$

and

$$\delta d_{\ell} = \frac{\parallel \mathbf{d}_{\ell}^{\mathbf{o}} - \mathbf{d} \parallel_2}{\parallel \mathbf{d} \parallel_2}, \quad \ell = 1, ..., D. \tag{56}$$

Regardless of the particular method used, the following inequality (?, p. 66) is applicable:

$$\delta p_\ell \leq \kappa \, \delta d_\ell, \quad \ell = 1, ..., D, \tag{57}$$

where κ is the constant of proportionality between the model perturbation δp_ℓ (equation 55) and the data perturbation δd_ℓ (equation 56). The constant κ acts as the condition number of an invertible matrix in a given inversion, and thus measures the instability of the solution. The larger (smaller) the value of κ the more unstable (stable) is the estimated solution.

Equation 57 shows a linear relationship between the model perturbation and the data perturbation. By plotting δp_ℓ (equation 55) against δd_ℓ (equation 56) produced by a set of D estimated solution obtained by applying a given equivalent-layer method, we obtain a straight line behaviour described by equation 57. By applying a linear regression, we obtain a fitted straight line whose estimated slope (κ in equation 57) quantifies the solution stability.

Here, the analysis of solution stability is numerically conducted by applying the classical equivalent-layer technique with zeroth-order Tikhonov regularization, the convolutional method for gravimetric and magnetic data, the deconvolutional method (equation 53) and the deconvolutional method with different values for the Wiener stabilization (equation 54).

4 NUMERICAL SIMULATIONS

We investigated different computational algorithms for inverting gravity disturbances and total-field anomalies. To test the capability of the fast equivalent-layer technique for processing that potential field data we measure of the computational effort by counting the number of floating-point operations (*flops*), such as additions, subtractions, multiplications, and divisions (Golub and Van Loan, 2013) for different number of observation points, ranging from $10,000$ up to $1,000,000$. The results generated when using iterative methods are set to $it = 50$ for the number of iterations.

4.1 Floating-point operations calculation

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

4.1.1 Normal equations using Cholesky decomposition

The equivalent sources can be estimated directly from solving the normal equations 3. In this work we will use the Cholesky decompositions method to calculate the necessary *flops*. In this method it is calculated the lower triangule of $\mathbf{A}^T\mathbf{A}$ $(1/2N^3)$, the Cholesky factor $(1/3N^3)$, a matrix-vector multiplication $(2N^2)$ and finally solving the triangular system $(2N^2)$, totalizing

$$f_{classical} = \frac{5}{6}N^3 + 4N^2 \tag{58}$$

4.1.2 Window method (Leão and Silva, 1989)

The moving data-window scheme (Leão and Silva, 1989) solve N linear systems with much smaller sizes (equation 31b). For our results we are considering a data-window of the same size of wich the authors presented in theirs work ($N_w = 49$) and the same number of equivalent sources ($M_w = 225$). We are doing this process for all the other techniques to standardize the resolution of our problem. Using the Cholesky decomposition with this method the *flops* are

$$f_{window} = N\frac{5}{6}M_w N_w^2 + 4N_w M_w \tag{59}$$

4.1.3 PEL method (Oliveira Jr. et al., 2013)

The polynomial equivalent layer uses a simliar approach od moving windows from Leão and Silva (1989). For this operations calculation (equation 41) we used a first degree polynomial (two variables) and each window contains $N_s = 1,000$ observed data and $M_s = 1,000$ equivalent sources. Following the steps given in (Oliveira Jr. et al., 2013) the total *flops* becomes

$$f_{pel} = \frac{1}{3}H^3 + 2H^2 + 2NM_sH + H^2N + 2HN + 2NP \tag{60}$$

where H is the number of constant coefficients for the first degree polynomial ($P = 3$) times the number of windows ($P \times N/N_s$).

4.1.4 Conjugate gradient least square (CGLS)

The CGLS method is a very stable and fast algorithm for solving linear systems iteratively. Its computational complexity envolves a matrix-vector product outside the loop ($2N^2$), two matrix-vector products inside the loop ($4N^2$) and six vector products inside the loop ($12N$) (**?**)

$$f_{cgls} = 2N^2 + it(4N^2 + 12N) \tag{61}$$

4.1.5 Wavelet compression method with CGLS (?)

For the wavelet method (equation 34) we have calculated a coompression rate of 98% ($C_r = 0.02$) for the threshold as the authors used in **?** and the wavelet transformation requiring $\log_2(N)$ *flops* each (equations 33 and 35c), with its inverse also using the same number of operations (equation 37). Combined with the conjugate gradient least square necessary steps and iterations, the number of *flops* are

$$f_{wavelet} = 2NC_r + 4N\log_2(N) + it(4N\log_2(N) + 4NC_r + 12C_r) \tag{62}$$

4.1.6 Fast equivalent layer for gravity data (Siqueira et al., 2017)

The fast equivalent layer from Siqueira et al. (2017) solves the linear system in *it* iterations. The main cost of this method (equations 43,44, 45 and 46)is the matrix-vector multiplication to asses the predicted data ($2N^2$) and three simply element by element vector sum, subtraction and division ($3N$ total)

$$f_{siqueira} = it(3N + 2N^2) \tag{63}$$

4.1.7 Convolutional equivalent layer for gravity data (?)

This methods replaces the matrix-vector multiplication of the iterative fast-equivalent technique (Siqueira et al., 2017) by three steps, involving a Fourier transform, an inverse Fourier transform, and a Hadamard product of matrices (equation 52). Considering that the first column of our BCCB matrix has $4N$ elements, the flops count of this method is

$$f_{convgrav} = \kappa 4N\log_2(4N) + it(27N + \kappa 8N\log_2(4N)) \tag{64}$$

In the resultant count we considered a *radix-2* algorithm for the fast Fourier transform and its inverse, which has a κ equals to 5 and requires $\kappa 4N\log_2(4N)$ flops each. The Hadarmard product of two matrices of $4N$ elements with complex numbers takes $24N$ flops. Note that equation 64 is different from the one presented in **?** because we also added the flops necessary to calculate the eigenvalues in this form. It does not differentiate much in order of magnitude because the iterative part is the most costful.

4.1.8 Convolutional equivalent layer for magnetic data (?)

The convolutional equivalent layer for magnetic data uses the same flops count of the main operations as in the gravimetric case (equation 52), the difference is the use of the conjugate gradient algorithm to solve the inverse problem. It requires a Hadamard product outside of the iterative loop and the matrix-vector and vector-vector multiplications inside the loop as seem in equation 61.

$$f_{convmag} = \kappa 16N \log_2(4N) + 24N + it(\kappa 16N \log_2(4N) + 60N) \tag{65}$$

4.1.9 Deconvolutional method

The deconvolution method does not require an iterative algorithm, rather it solves the estimative of the physical properties in a single step using the $4N$ eigenvalues of the BCCB matrix as in the convolutional method. From equation 53 it is possible to deduce this method requires two fast Fourier transform ($\kappa 4N \log_2(4N)$), one for the eigenvalues and another for the data transformation, a element by element division ($24N$) and finally, a fast inverse Fourier transform for the final estimative ($\kappa 4N \log_2(4N)$).

$$f_{deconv} = \kappa 12N \log_2(4N) + 24N \tag{66}$$

Using the deconvolutional method with a Wiener stabilization adds two multiplications of complex elements of the conjugates eigenvalues ($24N$ each) and the sum of $4N$ elements with the stabilization parameter μ as shown in equation 54

$$f_{deconvwiener} = \kappa 12N \log_2(4N) + 76N \tag{67}$$

CONFLICT OF INTEREST STATEMENT

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

AUTHOR CONTRIBUTIONS

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

FUNDING

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

ACKNOWLEDGMENTS

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

DATA AVAILABILITY STATEMENT

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

REFERENCES

Barnes, G. and Lumley, J. (2011). Processing gravity gradient data. *GEOPHYSICS* 76, I33–I47. doi:10.1190/1.3548548

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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