

Convolutional equivalent layer for magnetic data processing

Diego Takahashi^{†*}, Vanderlei C. Oliveira Jr.[†] and Valéria C. F. Barbosa[†]

[†]*Observatório Nacional, Department of Geophysics, Rio de Janeiro, Brazil*

* Corresponding author: diego.takahashi@gmail.com

(September 8, 2021)

GEO-XXXX

Running head: **Magnetic convolutional equivalent layer**

ABSTRACT

We present a fast equivalent layer method for processing large-scale magnetic data. We demonstrate that the sensitivity matrix associated with an equivalent layer of dipoles assumes a Block-Toeplitz Toeplitz-Block (BTTB) structure for the particular case in which observations and equivalent sources are aligned on a horizontal and regularly-spaced grid. The product of a BTTB matrix and an arbitrary vector represents a 2D discrete convolution which, in turn, can be efficiently computed via 2D Fast Fourier Transform (2D FFT). In this case, the matrix-vector product uses only the elements forming the first column of the BTTB matrix, saving computational time and system memory. Our convolutional equivalent layer method uses this fast convolution approach to compute the matrix-vector products in the iterative conjugate gradient algorithm with the purpose of estimating the physical-property distribution over the equivalent layer and then processing large data sets.

Synthetic tests with a mid-size 100×50 grid of total-field anomaly data show a decrease of $\approx 10^4$ in floating-point operations and $\approx 25\times$ in computation runtime of our method compared to the classical approach of solving the least-squares normal equations via Cholesky

decomposition. Better results are obtained for millions of data, showing drastic decreases in RAM usage and runtime, allowing to perform magnetic data processing of large data sets on regular desktop computers.

PAREI AQUI

Our results also show that, compared to the standard Fourier approach Synthetic tests simulating data on irregular grids or over undulating observation surfaces show the robustness of the convolutional equivalent layer in processing magnetic surveys that violate the requirement that the data be measured on a regular grid and the observation surface be planar. Test on real magnetic data from Carajás Province, Brazil, with 1,310,000 observations on an irregular grid, confirms the success of our method, taking 385.56 seconds to estimate the physical-property distribution over the equivalent layer and 2.64 seconds for upward-continuing the data.

INTRODUCTION

Large-scale data processing with tens of thousands of data, is a reality in all areas of geophysics including the geophysical potential fields. The potential-field data processing includes convolution integrals which can be solved either in the space or Fourier domains. The earliest techniques of potential-field data processing were developed in the space domain. For example, Peters (1949) accomplished, in the space domain, the second and fourth derivatives of magnetic data and the upward- and downward-continuations of magnetic data by deriving coefficients that are used in a graphical convolution with the magnetic data. However, the techniques for processing potential-field data in space-domain were soon substituted by the Fourier-domain techniques. Dean (1958) pointed out that the operations of second derivative, analytic continuation, smoothing, the removing of residuals or regionals, and others for processing potential-field data are similar to the electric filter circuits in Fourier domain. Dean (1958) was the first to develop the theory of linear filter in Fourier domain for gravity and magnetic processing and to present filters in Fourier domain (Dean, 1958, see Table I, p 113) for some theoretical geophysical operations (e.g., derivatives and upward and downward continuations). Gunn (1975) presented a comprehensive analysis of processing potential-field data in Fourier domain.

An approach for processing potential-field data in space domain is the equivalent-layer technique. The equations deductions of the equivalent layer as a solution of the Laplace's equation in the region above the source was first presented by Kellogg (1929) and detailed explanations can also be found in Blakely (1996). Although the equivalent-layer technique has been known since the 1960s in geophysical literature (Danes, 1961; Bott, 1967; Dampney, 1969), its use has become feasible only recently because the advances in computational

power. In magnetic data processing, some authors explored this technique for calculating the first and second vertical derivatives fields (Emilia, 1973), reduction to the pole (Silva, 1986; Oliveira Jr. et al., 2013; Li et al., 2014), upward/downward continuations (Hansen and Miyazaki, 1984; Li and Oldenburg, 2010) and total magnetic induction vector components calculation (Sun et al., 2019).

Together with the rise in computational processing power, some works tried new implementations to increase the efficiency of the equivalent layer. In Leão and Silva (1989) the authors used a shifting window over the layer, increasing the number of linear systems to be solved, but the size of each linear system is reduced. Another approach for a fast equivalent layer was proposed by Li and Oldenburg (2010) who transformed the full sensitivity matrix into a sparse one by using the compression of the coefficient matrix using fast wavelet transforms based on orthonormal, compactly supported wavelets. Oliveira Jr. et al. (2013) divided the equivalent layer into a grid of fixed source windows. Instead of directly calculating the physical-property distribution of a finite set of equivalent sources (e.g., dipoles, point of masses) arranged in the entire equivalent layer, Oliveira Jr. et al. (2013) estimated the coefficients of a bivariate polynomial function describing the physical-property distribution within each equivalent-source window. The estimated polynomial-coefficients are transformed into the physical-property distribution and thus any standard linear transformation of the data can be performed. Grounded on excess mass constraint, Siqueira et al. (2017) proposed an iterative method for processing large gravimetric data using the equivalent layer without requiring the solution of a linear system. In Siqueira et al. (2017), the initial mass distribution over the equivalent layer is proportional to observed gravity data and it is updated at each iteration by adding mass corrections that are proportional to the residuals of observed and estimated data.

One of the greatest obstacles to the use of the equivalent-layer technique for processing potential-field data is the solution of the associated linear system. A wide variety of applications in mathematics and engineering that fall into Toeplitz systems propelled the development of a large variety of methods for solving them. Direct methods were conceived by Levinson (1946) and by Trench (1964). Currently, the iterative method of conjugate gradient is used in most cases, in Chan and Jin (2007) the authors presented an introduction on the topic for 1D data structures of Toeplitz matrices and also for 2D data structures, which they called block-Toeplitz Toeplitz-block matrices. In both cases, the solving strategy is to embed the Toeplitz/BTTB matrix into a Circulant/Block-Circulant Circulant-Block matrix, calculate its eigenvalues by a 1D or 2D fast Fourier transform of its first column, respectively and carry the matrix-vector product between kernel and parameters at each iteration of the conjugate gradient method in a very fast manner.

In potential field methods, the properties of Toeplitz system have been used for downward continuation (Zhang et al., 2016) and for 3D gravity-data inversion using a 2D multilayer model (Zhang and Wong, 2015). More recently, Hogue et al. (2020) provided an overview on modeling the gravity and magnetic kernels using the BTTB structures and Renaut et al. (2020) used BTTB the structures for inversion of both gravity and magnetic data to recover sparse subsurface structures. Takahashi et al. (2020) combined the fast equivalent source technique presented by Siqueira et al. (2017) with the concept of symmetric block-Toeplitz Toeplitz-block (BTTB) matrices to introduce the convolutional equivalent layer for gravimetric data technique. Takahashi et al. (2020) showed that the BTTB structure appears when the sensitivity matrix of the linear system, required to solve the gravimetric equivalent layer, is calculated on a regular spaced grid of dataset with constant height and each equivalent source is exactly beneath each observed data point. This work showed an

decrease in the order of 10^4 in floating-point operations needed to estimate the equivalent sources; thus, the Takahashi et al. (2020) method was able to efficiently process very large gravity data sets. Moreover, Takahashi et al. (2020) method yielded neither significant boundary effects nor noise amplification.

In this work, the convolutional equivalent layer using the block-Toeplitz Toeplitz-block idea, presented in Takahashi et al. (2020), will be used to solve the linear system required to estimate the physical property that produces a magnetic field on regular grids. Here, we achieve very fast solutions using a conjugate gradient algorithm combined with the fast Fourier transform. We present a novel method of exploring the symmetric structures of the second derivatives of the inverse of the distance contained in the magnetic kernel, to keep the memory RAM usage to the minimal by using only one equivalent source to carry the calculations of the forward problem. We also show tests of the magnetic convolutional equivalent layer when irregular grids are used. The convergence of the conjugate gradient maintains in an acceptable level even using irregular grids. Our results show the good performance of our method in producing fast and robust solutions for processing large amounts of magnetic data using the equivalent layer technique.

METHODOLOGY

Classical equivalent layer for magnetic data

Let \mathbf{d}^o be the $N \times 1$ observed data vector, whose i th element is the total-field anomaly d_i^o produced by arbitrarily magnetized sources at the position (x_i, y_i, z_i) , $i = 1, \dots, N$, of a right-handed Cartesian coordinate system with x -, y - and z -axis pointing to north, east and down, respectively. We consider that the total-field anomaly data d_i^o represent the discrete values of a harmonic function. Besides, we consider that the main geomagnetic field direction at the study area can be defined by the unit vector

$$\hat{\mathbf{F}} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \cos(I_0) \cos(D_0) \\ \cos(I_0) \sin(D_0) \\ \sin(I_0) \end{bmatrix}, \quad (1)$$

with constant inclination I_0 and declination D_0 . In this case, d_i^o can be approximated by the predicted total-field anomaly

$$\Delta T_i = \sum_{j=1}^M p_j a_{ij}, \quad (2)$$

which describes the magnetic induction exerted, at the observation point (x_i, y_i, z_i) , by a discrete layer of M dipoles (equivalent sources) defined on the horizontal plane $z = z_c$, where p_j is the magnetic moment intensity (in A m^2) of the j th dipole, that has unit volume and is located at the point (x_j, y_j, z_c) . In equation 2, a_{ij} is the harmonic function

$$a_{ij} = c_m \frac{\mu_0}{4\pi} \hat{\mathbf{F}}^\top \mathbf{H}_{ij} \hat{\mathbf{u}}, \quad (3)$$

the unit vector

$$\hat{\mathbf{u}} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \cos(I) \cos(D) \\ \cos(I) \sin(D) \\ \sin(I) \end{bmatrix}, \quad (4)$$

defines the magnetization direction of all dipoles, with constant inclination I and declination D , $\mu_0 = 4\pi 10^{-7}$ H/m is the magnetic constant, $c_m = 10^9$ is a factor that transforms the magnetic induction from Tesla (T) to nanotesla (nT) and \mathbf{H}_{ij} is a 3×3 matrix

$$\mathbf{H}_{ij} = \begin{bmatrix} h_{ij}^{xx} & h_{ij}^{xy} & h_{ij}^{xz} \\ h_{ij}^{xy} & h_{ij}^{yy} & h_{ij}^{yz} \\ h_{ij}^{xz} & h_{ij}^{yz} & h_{ij}^{zz} \end{bmatrix}, \quad (5)$$

where

$$h_{ij}^{\alpha\beta} = \begin{cases} \frac{3(\alpha_i - \alpha_j)^2}{r_{ij}^5} - \frac{1}{r_{ij}^3}, & \alpha = \beta \\ \frac{3(\alpha_i - \alpha_j)(\beta_i - \beta_j)}{r_{ij}^5}, & \alpha \neq \beta \end{cases}, \quad \alpha, \beta = x, y, z, \quad (6)$$

are the second derivatives of the inverse distance function

$$\frac{1}{r_{ij}} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}} \quad (7)$$

with respect to the coordinates of the observation point (x_i, y_i, z_i) .

Equation 2 can be rewritten in matrix notation as follows:

$$\mathbf{d}(\mathbf{p}) = \mathbf{A}\mathbf{p}, \quad (8)$$

where $\mathbf{d}(\mathbf{p})$ is the $N \times 1$ predicted data vector with i th element defined by the predicted total-field anomaly ΔT_i (equation 2), \mathbf{p} is the $M \times 1$ parameter vector whose j th element is the magnetic moment intensity p_j of the j th dipole and \mathbf{A} is the $N \times M$ sensitivity matrix with element ij defined by the harmonic function a_{ij} (equation 3). In the classical equivalent-layer technique, the common approach for estimating the parameter vector \mathbf{p} from the observed total-field anomaly data \mathbf{d}^o is solving the least-squares normal equations

$$\mathbf{A}^\top \mathbf{A} \mathbf{p} = \mathbf{A}^\top \mathbf{d}^o. \quad (9)$$

Equation 9 is usually solved by first computing the Cholesky factor \mathbf{G} of matrix $\mathbf{A}^\top \mathbf{A}$ and then using it to solve the linear systems (Golub and Loan, 2013, p. 262):

$$\begin{aligned}\mathbf{G}\mathbf{w} &= \mathbf{A}^\top \mathbf{d}^o \\ \mathbf{G}^\top \tilde{\mathbf{p}} &= \mathbf{w}\end{aligned}, \quad (10)$$

where \mathbf{w} is a dummy variable. This approach to estimate the parameter vector will be referenced throughout this work as the *classical method*. The computational cost associated with the classical method can be very high when dealing with large datasets. In the following subsections, we will show how to explore the structure of the sensitivity matrix \mathbf{A} and efficiently solve the least-squares normal equations (equation 9).

Matrix \mathbf{A} in terms of matrix components $\mathbf{A}_{\alpha\beta}$

To access the structure of the sensitivity matrix \mathbf{A} (equation 8), let us first rewrite its elements a_{ij} (equation 3) in the following way:

$$a_{ij} = a_{ij}^{xx} + a_{ij}^{xy} + a_{ij}^{xz} + a_{ij}^{yy} + a_{ij}^{yz} + a_{ij}^{zz}, \quad (11)$$

where

$$a_{ij}^{\alpha\beta} = \begin{cases} c_m \frac{\mu_0}{4\pi} (F_\alpha u_\beta) h_{ij}^{\alpha\beta} & , \quad \alpha = \beta \\ c_m \frac{\mu_0}{4\pi} (F_\alpha u_\beta + F_\beta u_\alpha) h_{ij}^{\alpha\beta} & , \quad \alpha, \beta = x, y, z, \end{cases} \quad (12)$$

are defined by the elements of $\hat{\mathbf{F}}$ (equation 1), $\hat{\mathbf{u}}$ (equation 4) and \mathbf{H}_{ij} (equations 5 and 6).

Then, we can rewrite the sensitivity matrix \mathbf{A} (equation 8) according to:

$$\mathbf{A} = \mathbf{A}_{xx} + \mathbf{A}_{xy} + \mathbf{A}_{xz} + \mathbf{A}_{yy} + \mathbf{A}_{yz} + \mathbf{A}_{zz}, \quad (13)$$

where $\mathbf{A}_{\alpha\beta}$ are $N \times M$ matrices with elements ij defined by $a_{ij}^{\alpha\beta}$ (equation 12).

Now we can define the structure of \mathbf{A} in terms of its components $\mathbf{A}_{\alpha\beta}$ (equation 13). To do this, we consider the particular case in which the observed total-field anomaly is located on an $N_x \times N_y$ regular grid of points spaced by Δ_x and Δ_y along the x - and y -directions, respectively, on a constant vertical coordinate z_0 . We also consider that the equivalent layer is formed by one dipole right below each observation point, at the constant coordinate z_c . In this case, the number of equivalent sources M is equal to the number of data N and, consequently, matrices \mathbf{A} and $\mathbf{A}_{\alpha\beta}$ become square ($N \times N$). Besides, the horizontal coordinates x_i and y_i of the observation points can be defined by

$$x_i = x_1 + [k(i) - 1] \Delta_x \quad (14)$$

and

$$y_i = y_1 + [l(i) - 1] \Delta_y , \quad (15)$$

where x_1 and y_1 are the lower limits for x_i and y_i , respectively, and $k(i)$ and $l(i)$ are integer functions defined according to the orientation of the data grid (Figure 1). For *x-oriented grids*, the integer functions are given by

$$k(i) = i - \left\lceil \frac{i}{N_x} \right\rceil N_x + N_x \quad (16)$$

and

$$l(i) = \left\lceil \frac{i}{N_x} \right\rceil . \quad (17)$$

For *y-oriented grids*, the integer functions are given by

$$k(i) = \left\lceil \frac{i}{N_y} \right\rceil \quad (18)$$

and

$$l(i) = i - \left\lceil \frac{i}{N_y} \right\rceil N_y + N_y . \quad (19)$$

In equations 16–19, $\lceil \cdot \rceil$ denotes the ceiling function (e.g., Graham et al., 1994, p. 67-68).

Equations 14–19 can also be used to define the coordinates x_j and y_j of the equivalent sources, but with index j instead of i .

By using equations 14–19 to define the coordinates x_i and y_i of the observation points and x_j and y_j of the equivalent sources, we can rewrite the elements $h_{ij}^{\alpha\beta}$ (equation 6) of matrix \mathbf{H}_{ij} (equation 5) as follows:

$$h_{ij}^{xx} = \frac{3(\Delta k_{ij} \Delta_x)^2}{r_{ij}^5} - \frac{1}{r_{ij}^3}, \quad (20)$$

$$h_{ij}^{yy} = \frac{3(\Delta l_{ij} \Delta_y)^2}{r_{ij}^5} - \frac{1}{r_{ij}^3}, \quad (21)$$

$$h_{ij}^{zz} = \frac{3\Delta_z^2}{r_{ij}^5} - \frac{1}{r_{ij}^3}, \quad (22)$$

$$h_{ij}^{xy} = \frac{3(\Delta k_{ij} \Delta_x)(\Delta l_{ij} \Delta_y)}{r_{ij}^5}, \quad (23)$$

$$h_{ij}^{xz} = \frac{3(\Delta k_{ij} \Delta_x) \Delta_z}{r_{ij}^5} \quad (24)$$

and

$$h_{ij}^{yz} = \frac{3(\Delta l_{ij} \Delta_y) \Delta_z}{r_{ij}^5}, \quad (25)$$

where $\Delta_z = z_c - z_0$,

$$\Delta k_{ij} = \frac{x_i - x_j}{\Delta_x} = k(i) - k(j), \quad (26)$$

$$\Delta l_{ij} = \frac{y_i - y_j}{\Delta_y} = l(i) - l(j) \quad (27)$$

and

$$\frac{1}{r_{ij}} = \frac{1}{\sqrt{(\Delta k_{ij} \Delta_x)^2 + (\Delta l_{ij} \Delta_y)^2 + \Delta_z^2}}. \quad (28)$$

Note that the integer functions $k(i)$, $k(j)$, $l(i)$ and $l(j)$ (equations 16–19) defining Δk_{ij} (equation 26), Δl_{ij} (equation 27) and $\frac{1}{r_{ij}}$ (equation 28) assume different forms depending on the grid orientation. Despite of that, it can be shown that

$$\Delta k_{ij} = -\Delta k_{ji}, \quad (29)$$

$$\Delta l_{ij} = -\Delta l_{ji} \quad (30)$$

and

$$\frac{1}{r_{ij}} = \frac{1}{r_{ji}} \quad (31)$$

for any grid orientation.

General structure of matrices $\mathbf{A}_{\alpha\beta}$

By using equations 20–28 to compute $a_{ij}^{\alpha\beta}$ (equation 12), we can show that matrices $\mathbf{A}_{\alpha\beta}$ (equation 13) assume well-defined structures that can be conveniently represented with *block indices* q and p (Takahashi et al., 2020). These indices are defined by the integer functions Δk_{ij} and Δl_{ij} (equations 26 and 27), in terms of the indices i of the observation points (x_i, y_i, z_0) and j of the equivalent sources (x_j, y_j, z_c) . For *x-oriented grids* (Figure 1), $Q = N_y$, $P = N_x$ and the block indices q and p are given by:

$$q \equiv q(i, j) = \Delta l_{ij} \quad (32)$$

and

$$p \equiv p(i, j) = \Delta k_{ij}, \quad (33)$$

where Δk_{ij} and Δl_{ij} (equations 26 and 27) are defined by integer functions $k(i)$, $k(j)$, $l(i)$ and $l(j)$ given by equations 16 and 17. For *y-oriented grids* (Figure 1), $Q = N_x$, $P = N_y$ and the block indices q and p are given by:

$$q \equiv q(i, j) = \Delta k_{ij} \quad (34)$$

and

$$p \equiv p(i, j) = \Delta l_{ij}, \quad (35)$$

where Δk_{ij} and Δl_{ij} (equations 26 and 27) are defined by integer functions $k(i)$, $k(j)$, $l(i)$ and $l(j)$ given by equations 18 and 19. Equations 32–35 show that q varies from $-Q + 1$ to $Q - 1$ and p from $-P + 1$ to $P - 1$, regardless of the grid orientation. They differ from those presented by Takahashi et al. (2020) due to the absence of the module.

Let us consider the small regular grid of $N_x = 3$ and $N_y = 2$ points shown by Figure 1. This grid may represent observation points (x_i, y_i, z_0) with constant vertical coordinate z_0 or equivalent sources (x_j, y_j, z_c) with constant vertical coordinate $z_c > z_0$. In both cases, the horizontal coordinates are defined by equations 14 and 15. Given an index i , associated with an observation point, and an index j , associated with an equivalent source, we can compute Δk_{ij} (equation 26), Δl_{ij} (equation 27) and $\frac{1}{r_{ij}}$ (equation 28). The matrices $\Delta\mathbf{K}$ and $\Delta\mathbf{L}$ having elements ij defined by Δk_{ij} and Δl_{ij} , respectively, assume different forms, depending on the grid orientation. For x -oriented grids (Figure 1), they are given by:

$$\Delta\mathbf{K} = \begin{bmatrix} 0 & -1 & -2 & 0 & -1 & -2 \\ 1 & 0 & -1 & 1 & 0 & -1 \\ 2 & 1 & 0 & 2 & 1 & 0 \\ 0 & -1 & -2 & 0 & -1 & -2 \\ 1 & 0 & -1 & 1 & 0 & -1 \\ 2 & 1 & 0 & 2 & 1 & 0 \end{bmatrix} \quad (36)$$

and

$$\Delta \mathbf{L} = \begin{bmatrix} 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (37)$$

For y -oriented grids (Figure 1), they are given by:

$$\Delta \mathbf{K} = \begin{bmatrix} 0 & 0 & -1 & -1 & -2 & -2 \\ 0 & 0 & -1 & -1 & -2 & -2 \\ 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & -1 & -1 \\ 2 & 2 & 1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (38)$$

and

$$\Delta \mathbf{L} = \begin{bmatrix} 0 & -1 & 0 & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}. \quad (39)$$

These examples (equations 36–39) show that different combinations of indices i and j result in integer functions Δk_{ij} and Δl_{ij} (equations 26 and 27) having the same numerical value. In these cases, not only the numerical values of the corresponding elements $a_{ij}^{\alpha\beta}$ (equation 12), but also their associated block indices q and p (equations 32–35) are the same. The

contrary is also true: elements $a_{ij}^{\alpha\beta}$ having different associated block indices q and p also have different numerical values. Because of that, using the alternative notation $a_{qp}^{\alpha\beta}$ to define the elements $a_{ij}^{\alpha\beta}$ in terms of its associated block indices q and p is a good approach to investigate the structure of a given matrix component $\mathbf{A}_{\alpha\beta}$ (equation 13). This approach allows identifying elements $a_{ij}^{\alpha\beta}$ having the same numerical value only by inspecting their associated block indices.

Note that, for x -oriented grids, matrices $\Delta\mathbf{K}$ (equation 36) and $\Delta\mathbf{L}$ (equation 37) define the block indices p (equation 33) and q (equation 32), respectively. In this case, they are composed of $Q \times Q$ blocks with $P \times P$ elements each, where $Q = N_y$ and $P = N_x$. For y -oriented grids, matrices $\Delta\mathbf{K}$ (equation 38) and $\Delta\mathbf{L}$ (equation 39) define the block indices q (equation 34) and p (equation 35), respectively. In this case, they are also composed of $Q \times Q$ blocks with $P \times P$ elements each, but now $Q = N_x$ and $P = N_y$. The examples shown by equations 36–39 also illustrate that, regardless of grid orientation, (i) the block index q is constant inside each block; (ii) blocks disposed along the same block diagonal are equal to each other; (iii) the block index p is constant on each diagonal of a given block; (iv) elements of a given block located on the same diagonal are also equal do each other. The results obtained with the small grid shown in Figure 1 can be easily generalized for larger grids. Based on the well-defined structure of block indices, we can define matrices $\mathbf{A}_{\alpha\beta}$ in a general form

$$\mathbf{A}_{\alpha\beta} = \begin{bmatrix} \mathbf{A}_{\alpha\beta}^0 & \mathbf{A}_{\alpha\beta}^{-1} & \cdots & \mathbf{A}_{\alpha\beta}^{-Q+1} \\ \mathbf{A}_{\alpha\beta}^1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{A}_{\alpha\beta}^{-1} \\ \mathbf{A}_{\alpha\beta}^{Q-1} & \cdots & \mathbf{A}_{\alpha\beta}^1 & \mathbf{A}_{\alpha\beta}^0 \end{bmatrix}_{N \times N}, \quad (40)$$

with blocks $\mathbf{A}_{\alpha\beta}^q$, $q = -Q + 1, \dots, Q - 1$, given by

$$\mathbf{A}_{\alpha\beta}^q = \begin{bmatrix} a_{q0}^{\alpha\beta} & a_{q(-1)}^{\alpha\beta} & \cdots & a_{q(-P+1)}^{\alpha\beta} \\ a_{q1}^{\alpha\beta} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{q(-1)}^{\alpha\beta} \\ a_{q(P-1)}^{\alpha\beta} & \cdots & a_{q1}^{\alpha\beta} & a_{q0}^{\alpha\beta} \end{bmatrix}_{P \times P}, \quad (41)$$

formed by elements $a_{qp}^{\alpha\beta}$, $p = -P + 1, \dots, P - 1$. This well-defined structure (equations 40 and 41) of matrix components $\mathbf{A}_{\alpha\beta}$ (equation 13) is called Block-Toeplitz Toeplitz-Block (BTTB) (e.g., Chan and Jin, 2007, p. 67).

Detailed structure of matrices \mathbf{A}_{xx} , \mathbf{A}_{yy} and \mathbf{A}_{zz}

Equations 40 and 41 define the general BTTB structure of all matrix components $\mathbf{A}_{\alpha\beta}$, but there are some differences between them. Let us consider the matrix component \mathbf{A}_{xx} , with elements a_{ij}^{xx} (equation 12) defined by the second derivative h_{ij}^{xx} (equation 20). It can be easily verified from equations 29 and 31 that $h_{ij}^{xx} = h_{ji}^{xx}$. As a consequence, $a_{ij}^{xx} = a_{ji}^{xx}$, which means that

$$\mathbf{A}_{xx} = (\mathbf{A}_{xx})^\top \quad (42)$$

for any grid orientation. Now, let us investigate the elements a_{qp}^{xx} forming the blocks \mathbf{A}_{xx}^q . For x -oriented grids (Figure 1), the block indices q and p are defined by equations 32 and 33 and a_{qp}^{xx} can be rewritten as follows:

$$a_{qp}^{xx} = c_m \frac{\mu_0}{4\pi} (F_x u_x) \frac{3(p\Delta_x)^2}{r_{qp}^5} - \frac{1}{r_{qp}^3}, \quad (43)$$

where

$$\frac{1}{r_{qp}} = \frac{1}{\sqrt{(p\Delta_x)^2 + (q\Delta_y)^2 + \Delta_z^2}}. \quad (44)$$

For y -oriented grids (Figure 1), the block indices q and p are defined by equations 34 and 35 and a_{qp}^{xx} can be rewritten as follows:

$$a_{qp}^{xx} = c_m \frac{\mu_0}{4\pi} (F_x u_x) \frac{3(q \Delta_x)^2}{r_{qp}^5} - \frac{1}{r_{qp}^3}, \quad (45)$$

where

$$\frac{1}{r_{qp}} = \frac{1}{\sqrt{(q \Delta_x)^2 + (p \Delta_y)^2 + \Delta_z^2}}. \quad (46)$$

From equations 43–46, we can easily verify that

$$\mathbf{A}_{xx}^q = \mathbf{A}_{xx}^{(-q)} \quad (47)$$

and

$$\mathbf{A}_{xx}^q = (\mathbf{A}_{xx}^q)^\top. \quad (48)$$

Note that these symmetries are valid for any grid orientation. From this results we conclude the matrix component \mathbf{A}_{xx} is *symmetric-Block-Toeplitz symmetric-Toeplitz-Block* for any grid orientation. The same reasoning can be used to show that matrices \mathbf{A}_{yy} and \mathbf{A}_{zz} also have this symmetric structure.

Detailed structure of matrix \mathbf{A}_{xy}

Let \mathbf{A}_{xy} be a matrix component with elements a_{ij}^{xy} (equation 12) defined by the second derivative h_{ij}^{xy} (equation 23). It can be easily verified from equations 29–31 that $h_{ij}^{xy} = h_{ji}^{xy}$. As a consequence, $a_{ij}^{xy} = a_{ji}^{xy}$, which means that

$$\mathbf{A}_{xy} = (\mathbf{A}_{xy})^\top \quad (49)$$

for any grid orientation. For x -oriented grids (Figure 1), the block indices q and p are defined by equations 32 and 33 and a_{qp}^{xy} can be rewritten as follows:

$$a_{qp}^{xy} = c_m \frac{\mu_0}{4\pi} (F_x u_y + F_y u_x) \frac{3(p \Delta_x)(q \Delta_y)}{r_{qp}^5}, \quad (50)$$

with $\frac{1}{r_{qp}}$ defined by equation 44. For y -oriented grids (Figure 1), the block indices q and p are defined by equations 34 and 35 and a_{qp}^{xy} can be rewritten as follows:

$$a_{qp}^{xy} = c_m \frac{\mu_0}{4\pi} (F_x u_y + F_y u_x) \frac{3(q\Delta_x)(p\Delta_y)}{r_{qp}^5}, \quad (51)$$

with $\frac{1}{r_{qp}}$ defined by equation 46. From equations 44, 46, 50 and 51, we can show that

$$\mathbf{A}_{xy}^q = -\mathbf{A}_{xy}^{(-q)} \quad (52)$$

and

$$\mathbf{A}_{xy}^q = -(\mathbf{A}_{xy}^q)^\top. \quad (53)$$

Note that these symmetries are valid for any grid orientation. From this results we conclude the matrix component \mathbf{A}_{xy} is *skew symmetric-Block-Toeplitz skew symmetric-Toeplitz-Block* for any grid orientation.

Detailed structure of matrices \mathbf{A}_{xz} and \mathbf{A}_{yz}

Let \mathbf{A}_{xz} be a matrix component with elements a_{ij}^{xz} (equation 12) defined by the second derivative h_{ij}^{xz} (equation 24). It can be easily verified from equations 29–31 that $h_{ij}^{xz} = -h_{ji}^{xz}$. As a consequence, $a_{ij}^{xz} = -a_{ji}^{xz}$, which means that

$$\mathbf{A}_{xz} = -(\mathbf{A}_{xz})^\top \quad (54)$$

for any grid orientation. For x -oriented grids (Figure 1), the block indices q and p are defined by equations 32 and 33 and a_{qp}^{xz} can be rewritten as follows:

$$a_{qp}^{xz} = c_m \frac{\mu_0}{4\pi} (F_x u_z + F_z u_x) \frac{3(p\Delta_x)\Delta_z}{r_{qp}^5}, \quad (55)$$

with $\frac{1}{r_{qp}}$ defined by equation 44. In this case, we can see that

$$\mathbf{A}_{xz}^q = \mathbf{A}_{xz}^{(-q)} \quad (56)$$

and

$$\mathbf{A}_{xz}^q = -(\mathbf{A}_{xz}^q)^\top . \quad (57)$$

This structure is called *symmetric-Block-Toeplitz skew symmetric-Toeplitz-Block* and is valid only for x -oriented grids. For y -oriented grids (Figure 1), the block indices q and p are defined by equations 34 and 35 and a_{qp}^{xz} can be rewritten as follows:

$$a_{qp}^{xz} = c_m \frac{\mu_0}{4\pi} (F_x u_z + F_z u_x) \frac{3(q\Delta_x)\Delta_z}{r_{qp}^5} , \quad (58)$$

with $\frac{1}{r_{qp}}$ defined by equation 46. Now, we conclude that

$$\mathbf{A}_{xz}^q = -\mathbf{A}_{xz}^{(-q)} \quad (59)$$

and

$$\mathbf{A}_{xz}^q = (\mathbf{A}_{xz}^q)^\top . \quad (60)$$

This structure is called *skew symmetric-Block-Toeplitz symmetric-Toeplitz-Block* and is valid only for y -oriented grids.

The same reasoning can be followed to show that

$$\mathbf{A}_{yz} = -(\mathbf{A}_{yz})^\top \quad (61)$$

for any grid orientation. Besides, we can also show that

$$\mathbf{A}_{yz}^q = -\mathbf{A}_{yz}^{(-q)} \quad (62)$$

and

$$\mathbf{A}_{yz}^q = (\mathbf{A}_{yz}^q)^\top \quad (63)$$

for x -oriented grids (*skew symmetric-Block-Toeplitz symmetric-Toeplitz-Block*), while

$$\mathbf{A}_{yz}^q = \mathbf{A}_{yz}^{(-q)} \quad (64)$$

and

$$\mathbf{A}_{yz}^q = -(\mathbf{A}_{yz}^q)^\top \quad (65)$$

for y -oriented grids (*symmetric-Block-Toeplitz skew symmetric-Toeplitz-Block*).

Convolutional equivalent layer

The computational cost associated with the classical method to estimate the parameter vector \mathbf{p} by solving the linear system 9 can be very high or even prohibitive when dealing with large data sets. In these cases, a well-known alternative is solving the normal equations (equation 9) iteratively by using the *standard Conjugate Gradient Least Squares (CGLS) method*:

Algorithm 1 Standard CGLS pseudocode (Aster et al., 2019, p. 166).

Input: \mathbf{A} and \mathbf{d}^o .

Output: Estimated parameter vector $\tilde{\mathbf{p}}$.

Set $it = 0$, $\tilde{\mathbf{p}}_{(it)} = \mathbf{0}$, $\mathbf{c}_{(it-1)} = \mathbf{0}$, $\beta_{(it)} = 0$, $\mathbf{s}_{(it)} = \mathbf{d}^o$ and $\mathbf{r}_{(it)} = \mathbf{A}^\top \mathbf{s}_{(it)}$.

$$1 - \text{If } it > 0, \beta_{(it)} = \frac{\|\mathbf{r}_{(it)}\|_2^2}{\|\mathbf{r}_{(it-1)}\|_2^2}$$

$$2 - \mathbf{c}_{(it)} = \mathbf{r}_{(it)} + \beta_{(it)} \mathbf{c}_{(it-1)}$$

$$3 - \alpha_{(it)} = \frac{\|\mathbf{r}_{(it)}\|_2^2}{\|\mathbf{A} \mathbf{c}_{(it)}\|_2^2}$$

$$4 - \tilde{\mathbf{p}}_{(it+1)} = \tilde{\mathbf{p}}_{(it)} + \alpha_{(it)} \mathbf{c}_{(it)}$$

$$5 - \mathbf{s}_{(it+1)} = \mathbf{s}_{(it)} - \alpha_{(it)} \mathbf{A} \mathbf{c}_{(it)}$$

$$6 - \mathbf{r}_{(it+1)} = \mathbf{A}^\top \mathbf{s}_{(it+1)}$$

$$7 - it = it + 1$$

$$8 - \text{Repeat previous steps until convergence.}$$

Setting a convergence criteria based on the minimum tolerance of the residuals is a good

option to carry out this algorithm efficiently and still obtaining very good results. Another possibility is to set an invariance to the Euclidean norm of residuals between iterations, which would increase algorithm runtime, but with smaller residuals. We chose the first option, as we achieve satisfactory results.

Note that the standard CGLS solution (Algorithm 1) requires neither inverse matrix nor matrix-matrix product. Instead, it only requires: one matrix-vector product out of the loop and two matrix-vector products per iteration (in steps 3 and 6). These products can be efficiently computed by using the 2D FFT, as a discrete convolution (see Appendix A). Takahashi et al. (2020) used this approach to develop an efficient algorithm for gravity data processing. This modified approach in which the standard CGLS method is modified to efficiently compute the matrix-vector products will be referenced throughout this work as the *convolutional equivalent layer method*.

Computational performance

In this sections we compare the efficiency of the classical (equation 10), standard CGLS (Algorithm 1) and the convolutional equivalent layer method (Algorithm 1 with matrix-vector products computed according to Appendix A). To do this, we compute the total number of *flops* associated to them (Golub and Loan, 2013, p. 12).

For the classical method, we have $\frac{1}{2}N^3$ flops to compute the lower triangle of $\mathbf{A}^\top \mathbf{A}$; $\frac{1}{3}N^3$ flops to compute the Cholesky factor \mathbf{G} of $\mathbf{A}^\top \mathbf{A}$ (Golub and Loan, 2013, p. 164); $2N^2$ flops to compute the matrix-vector product $\mathbf{A}^\top \mathbf{d}^o$; and $2N^2$ flops to solve the triangular systems given by equation 10 (Golub and Loan, 2013, p. 106). The resultant flop count for

the classical method is

$$f_{classical} = \frac{5}{6}N^3 + 4N^2 . \quad (66)$$

For the standard CGLS method (Algorithm 1) we have $2N^2$ to compute the matrix-vector product $\mathbf{A}^\top \mathbf{s}_{(it)}$ out of the loop; $4N$ in step 1; $2N$ in step 2; $2N^2 + 2N$ in step 3; $2N$ in step 4; $2N$ in step 5; and $2N^2$ in step 6. The resultant flop count is given by:

$$f_{cglsls} = 2N^2 + it(4N^2 + 12N) . \quad (67)$$

To compute the flops count of our method, we need only to replace the flops associated with matrix-vector products in the standard CGLS method by those associated with 2D convolution defined in Appendix A, which consists of $\kappa 4N \log_2(4N)$ flops to compute the 2D FFT for each matrix $\mathbf{L}_{\alpha\beta}$ (equation A-14); $\kappa 4N \log_2(4N)$ flops to compute $\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P}$ via 2D FFT; $24N$ flops to compute the Hadamard product $\mathbf{L} \circ (\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P})$; and $\kappa 4N \log_2(4N)$ flops to compute the IDFT in equation A-16. We use $\kappa = 5$ for the *radix-2* algorithm (Van Loan, 1992, p. 15). By replacing these flops into Algorithm 1, we obtain the complete number of flops

$$f_{conv} = \kappa 16N \log_2(4N) + 24N + it(\kappa 16N \log_2(4N) + 60N) . \quad (68)$$

Figure 2 shows a comparison between $f_{classical}$ (equation 66), f_{cglsls} (equation 67) and f_{conv} (equation 68) for different numbers of observation points up to 1,000,000. As we can see, the total flops count associated with our method is 10^7 orders of magnitude smaller than that associated with the classical method and 10^3 orders of magnitude smaller than that associated with the standard CGLS method by using a maximum number of iterations $N^{it} = 50$.

Figure 3 shows the time necessary to build matrix \mathbf{A} (equation 13) and solve the linear

system for N varying up to 10,000. With $N = 10,000$, the classical method takes more than sixty-three seconds, the standard CGLS more than twelve seconds, while our method takes only half a second. The CPU used for this test was a Intel Core i7-7700HQ@2.8GHz.

Table 1 shows a comparison between the RAM memory storage associated with each method. The classical and standard CGLS methods have to store the whole matrix \mathbf{A} (equation 13). For example, a dataset with $N = 10,000$ observation points has an associated sensitivity matrix \mathbf{A} formed by $N^2 = 100,000,000$ elements and takes approximately 763 Megabytes of memory (8 bytes per element). Using the same number of observation points $N = 10,000$, our method requires only 1.831 Megabytes to store the first columns of the BCCB matrices $\mathbf{C}_{\alpha\beta}$ (equation A-6) and 0.6104 Megabytes to store the complex matrix \mathbf{L} (equation A-18) (16 bytes per element). For a bigger dataset with $N = 1,000,000$, the amount of necessary RAM goes to 7,629,395, 183.096 and 61.035 Megabytes, respectively.

APPLICATION TO SYNTHETIC DATA

Our convolutional equivalent layer method requires a regular data grid located on a horizontal and flat observation surface. Here, we evaluate the performance of our method by applying it to simulated airborne magnetic surveys formed by i) a regular data grid on a flat surface; ii) irregular data grids on a flat surface; and iii) regular data grid on undulating surfaces. Note that the simulated surveys in (ii) and (iii) violate the premises of our method.

Simulated airborne surveys

The upper and middle rows in Figure ?? show, respectively, the simulated flight patterns and noise-corrupted total-field anomalies of the airborne magnetic surveys used in our tests. The lower row in Figure ?? shows the true upward-continued total-field anomalies at $z = -1, 300$ m. All magnetic data (middle and lower rows in Figure ??) are produced by the same three synthetic bodies: two prisms and one sphere with constant total-magnetization vector having inclination, declination and intensity of 0° , 45° , and 2.8284 A/m, respectively. The simulated main geomagnetic field has inclination and declination of 10° and 37° , respectively.

Figure ??a shows the simulated airborne survey on a regular grid of 100×50 observation points (totaling $N = 5, 000$ observation points), with a grid spacing of $\Delta x = 101.01$ m and $\Delta y = 163.265$ m along the x - and y -axis, respectively. The noise-corrupted total-field anomaly (middle panel) is calculated at $z = -900$ m, with a pseudorandom Gaussian noise having null mean and standard deviation of $XXXX$ nT.

Figures ??b and ??c show the simulated surveys on irregular grids obtained by per-

turbing the horizontal coordinates of the regular grid (upper panel in Figure ??a). For the survey shown in Figure ??b, the x and y coordinates are perturbed with sequences of pseudorandom Gaussian noises having null mean and standard deviations equal to 20% of the corresponding grid spacing, which results in absolute values of 20.2 m and 32.6 m, along the x - and y -directions, respectively. For the survey shown in Figure ??c, the standard deviations are equal to 30% of the corresponding grid spacing, which results in absolute values of 30.3 m and 49.0 m along the x - and y -directions, respectively. Their noise-corrupted total-field anomalies (middle panels in Figures ??b and ??c) are calculated on their corresponding irregular grids, on a flat observation surface at $z = -900$ m, with a pseudorandom Gaussian noise having null mean and standard deviation of $XXXX$ nT.

Figures ??d and ??e show the simulated surveys on the same regular grid shown in Figure ??a (upper panel). The difference is that they are no longer on a flat, but on undulating surfaces. For the survey shown in Figure ??d, the z coordinates of the undulating surface are defined by a sequence of pseudorandom Gaussian noise having mean -900 m and standard deviation equal to 5% of 900 m, which corresponds to 45 m. For the survey shown in Figure ??e, the standard deviation is equal to 10% of 900 m, which corresponds to 90 m. The noise-corrupted total-field anomalies of these simulated surveys (middle panels in Figures ??d and ??e) are calculated on their corresponding undulating surfaces (upper panels in Figures ??d and ??e), on the same regular grid shown in Figure ??a, with a pseudorandom Gaussian noise having null mean and standard deviation of $XXXX$ nT.

Tests with a regular data grid on a flat surface

Figure ?? show the difference between the simulated (middle row in Figure ??) and predicted data (not shown) obtained by using the classical (the upper row) and our method (the middle row). From now on, we designate this difference as data residuals. The lower row in Figure ?? shows the convergence curve of our method.

The data residuals using the classical method (equation 10) are shown in the upper panel of Figure ??a, with mean 0.3627 nT and standard deviation 0.2724 nT. This process took 17.10 seconds. Using our method, the data residuals (the middle panel in Figure ??a) have mean 0.5223 nT and standard deviation 0.4323 nT. In this case, however, the processing time was only 0.18 seconds. As expected, the Euclidean norm of the data residuals produced by our method (lower panel in Figure ??a) decreases and the convergence criterion is satisfied close to iteration 50. This result shows that, in practice, our method converges way before N iterations, where N is the number of observations. Setting the convergence to N iterations, besides being unnecessary, it also demands a larger computer processing time, even in this synthetic test with a small number of $N = 5,000$ observations.

Tests with irregular data grids on a flat surface

Figure ??b shows the results obtained with the irregular data grid perturbed by using 20% of the regular grid spacing. In this Figure we can see that the data residuals using the classical method (upper panel) yield a good data fit with mean 0.3630 nT and standard deviation 0.2731 nT. Using our method, the data residuals (middle panel in Figure ??b) also produced an acceptable data fitting with mean of 0.7147 nT and standart deviation of 0.5622 nT. The Euclidean norm of the data residuals obtained by our method (lower panel

in Figure ??b) decreases, as expected, and converges to a constant value close to iteration 50.

Figure ??b shows the results obtained with the irregular data grid perturbed by using 30% of the regular grid spacing. This figure shows that the data residuals obtained by the classical method (upper panel) produced an acceptable data fitting, having mean 0.3634 nT and standard deviation 0.2735 nT. Using our method, the data residuals (middle panel in Figure ??c) with mean 0.9788 nT and standard deviation 0.7462 nT produced a worse data fitting. The convergence of our method (lower panel in Figure ??c) shows that, similarly to the previous results, the Euclidean norm of the residuals decreases; however it starts increasing without achieving an invariance. Hence, the convergence is not achieved.

Alternative text →

Using our method, the data residuals (middle panel in Figure ??c) with mean (usar valor com a menor norma euclidiana) nT and standard deviation (usar valor com a menor norma euclidiana) nT also produced a good data fitting. The convergence of our method (lower panel in Figure ??c) shows that, similarly to the previous results, the Euclidean norm of the residuals decreases; however it starts increasing without achieving an invariance. In this case, we have to stop the algorithm before the iteration N . Note that this good result was obtained by using a very perturbed data grid (upper panel in Figure ??c).

← **Alternative text**

Tests with regular data grid an undulating surfaces

Figure ??d shows the results obtained with data on the undulating surface varying 5% of $z = 900$ m. In this case, the data residuals either using the classical method (upper

panel in Figure ??d) or our method (middle panel in Figure ??d) reveal acceptable data fittings. Using the classical method, data residuals have mean 0.3712 nT and standard deviation 0.2870 nT. Using our method, they have mean 0.9542 nT and standard deviation 0.8943 nT. Likewise, the Euclidean norm of the data residuals produced by our method (lower panel in Figure ??d) decreases up to iteration 50 and reaches an invariance in the subsequent iterations.

Figure ??d shows the results obtained with data on the undulating surface varying 10% of $z = 900$ m. By using the classical approach, the data residuals (upper panel in Figure ??e) yielded a good data fitting, with mean 0.3865 nT and standard deviation 0.3216 nT. By using our method, the data residuals (middle panel in Figure ??e) yielded a worse data fitting with mean 1.6109 nT and standard deviation 1.6231 nT. The convergence curve (lower panel in Figure ??e) reveals the inadequacy of our method in dealing with observations on rugged surfaces, as the Euclidean norm of the data residuals decreases slower than in previous tests. We stress that, in this test, the undulating surface (upper panel in Figure ??e) varies in a broad range from $z = -570$ m to about $z = -1, 230$ m. Thus, this simulated airborne magnetic survey greatly violates the requirement of a flat observation surface demanded by our method.

Although our method is formulated to deal with magnetic observations measured on a horizontally regular grid, on a flat surface, the results obtained with synthetic data show that our method is robust in dealing either with irregular grids in the horizontal directions or with uneven surfaces. However, the robustness of our method has limitations. The performance limitation of our method depends on the degree of departure of the (i) x - and y -coordinates from those of the closest regular grid and (ii) the z coordinates from a constant value. High departures in the x -, y , and z -coordinates lead to unacceptable data

fittings (large data residuals), as shown the middle panels in Figures ??c and ??e.

Magnetic data processing

We performed the upward continuations of the synthetic total-field anomalies (middle row in Figure ??) by using the classical method, our convolutional equivalent layer method, and the classical approach in the Fourier domain, which consists in computing the Fourier transform of the total-field anomaly (e.g., Blakely, 1996, p. 317).

Figure ?? shows the differences between the true upward-continued total-field anomalies (lower row in Figure ??) and the predicted upward-continued total-field anomalies (not shown). We conveniently denote these differences as continuation residuals.

Figure ?? shows that the continuation residuals obtained by using the classical method (upper row) and our method (middle row) are similar to each other in most of the tests. One exception is the synthetic test with data over an undulating surface (Figures ??e and ??e), which greatly violates the requirement of a flat observation surface demanded by our method. Note that the maximum absolute value of the continuation residuals produced by using our method (middle panel in Figure ??e) are ≈ 2.5 times greater than those produced by the classical method (upper panel in Figure ??e). Besides, they are generally concentrated at the boundaries of the study area.

In contrast, the continuation residuals obtained by using the classical Fourier approach (lower row in Figure ??) are, in most of the tests, approximately 6 times greater than those produced by the classical method (upper row in Figure ??) and 4 times greater than those produced by our method (middle row in Figure ??). Note that, similarly to our method, the maximum absolute values of the continuation residuals obtained by using the classical

Fourier approach are located at the boundaries of the simulated area. However, the values are significantly higher.

We call attention to the following aspects: In applying the classical method, our method, or the classical Fourier approach, we do not expand the data by using a padding scheme. The data residuals (upper and middle rows in Figure ??) and the continuation residuals (Figure ??) are shown without removing the edge effects. The computational time required by our method is much lower than that required by the classical method and has the same order of magnitude of that required by the classical Fourier approach. However, the classical Fourier approach shows upward-continued data with strong border effects if no padding scheme is applied to expand the data.

APPLICATION TO FIELD DATA

We applied the convolutional equivalent layer method to the aeromagnetic data of Carajás, northern Brazil. The survey is composed of 131 flight lines along north-south direction with line spacing of $\Delta y = 3,000$ m. Data were measured with spacing $\Delta x = 7.65$ m along lines, with an average distance to the ground of 900 m. (TEM QUE CHECAR) The total number of observation points is $N = 6,081,345$. Figure 4 shows the observed total-field anomaly data over the study area.

We compare the results obtained with an interpolated regular grid of $10,000 \times 131$ points, by using the nearest neighbor algorithm, and a decimated irregular grid, also with $10,000 \times 131$ points, totaling $N = 1,310,000$ observation points in both cases. Both application were made with an Intel core i7 7700HQ@2.8GHz processor in single-processing and single-threading modes. Figures 5a and 5b show, respectively, the data obtained by interpolation and decimation. With 1,310,000 observation points, it would be necessary 12.49 Terabytes of RAM to store the full sensitivity matrix with the classical method. In this case, our method uses only 59.97 Megabytes, allowing regular desktop computers to be able to process this amount of data.

As the study area is very large, the main magnetic field varies with position. For this application, we set the main field direction as that of a mid location (latitude -6.5° and longitude -50.75°) where the declination is -19.86° for the IGRF model at 1st January, 2014 (epoch of the survey). The inclination was calculated using the magnetic field calculator from NOAA and is equal to -7.4391° .

Alternative text →

As the study area is very large, the main magnetic field varies with position. For

this application, we set the main field direction as that of a mid location (latitude -6.5° and longitude -50.75°) where the declination is -19.86° and inclination is $XXXXXX^{\circ}$ according to IGRF model at 1st January, 2014 (epoch of the survey). (É ESTRANHO CALCULAR A DECLINAÇÃO COM UM MODELO E A INCLINAÇÃO COM OUTRO)

← Alternative text

We set the equivalent layer at 300 meters above the ground (600 m below the data TEM QUE CHECAR). By applying our method to the interpolated regular grid (Figure 5a), we obtain the predicted data shown in Figure 6a and data residuals (Figure 6b) with mean 0.0762 nT and the standard deviation 0.4886 nT, revealing an acceptable data fitting. Our method took ≈ 390.80 seconds to converge at about 200 iterations. (TEM QUE CHECAR)

POR QUE A CURVA DE CONVERGÊNCIA NÃO FOI MOSTRADA?

By applying our method to the decimated irregular grid (Figure 5b), we obtain the predicted data shown in Figure 7a and data residuals (Figure 7b) with mean 0.0717 nT and standard deviation 0.3144 nT. In this case, our method took ≈ 385.56 seconds to converge at about 2,000 iterations (Figure 8) (TEM ALGO ESTRANHO AQUI. COMO QUE 2000 ITERAÇÕES FOI MAIS MAIS RÁPIDO DO QUE AS 200 ITERAÇÕES DO GRID INTERPOLADO?). The convergence curve reveals a good convergence rate obtained with the decimated irregular grid. This result shows the robustness of our method in processing irregular grid.

Finally, Figure 9 shows the upward-continued magnetic data to a horizontal plane located at 5, 000 m using the estimated equivalent layer obtained by applying our method to the decimated irregular grid (Figure 5b). This process took ≈ 2.64 seconds, showing good results without visible errors or border effects.

CONCLUSIONS

We have proposed a fast convolutional equivalent-layer technique for processing magnetic data whose computation time is more than four orders of magnitude less than the classical equivalent layer. Mathematically, we have demonstrated that the sensitivity matrix associated with the linear system of the magnetic equivalent layer carries the structure of BTTB matrices, which means not only a very low computational cost to calculate a matrix-vector product, but also the possibility to store only the first column of the matrix BCCB. In this work, our novel fast convolutional equivalent-layer technique uses only one equivalent source to calculate the first six columns of the inverse of distance second derivatives matrices and to set up the first column of the BCCB matrix embedded from the original magnetic kernel sensitivity matrix. We solve the linear system by adapting the method of Conjugate Gradient Least Square to compute fastly the BTTB matrix-vector product of the magnetic forward modeling in the equivalent-layer technique.

The comparisons between the performances of classical equivalent-layer technique and fast convolutional equivalent-layer technique using synthetic magnetic data show similar estimates of the physical-property distribution over the equivalent layer. The difference in time, however, is noticeable: 2.04 seconds using the classical approach and 0.083 seconds using our approach. This difference was obtained with a mid-size mesh of 100×50 points, greater results can be obtained if more observation points are used.

The comparisons between the performances of classical approach in the Fourier domain and fast convolutional equivalent-layer technique for processing synthetic magnetic data show that the computational time required by the classical Fourier approach is the lowest one. However, the classical Fourier approach requires not only expanding the data by using

a padding function to avoid the border effects but also it requires the measurement of the data on a regular grid and on a planar observation surface. Although the fast convolutional equivalent-layer technique also requires that the data be measured on a regular grid and the observation surface be planar, the synthetic tests show the robustness of our method to deal either with irregular grids or with uneven observation surface. This robustness of our method may fail if the horizontal coordinates of the observations are greatly scattered or if the observation surface is rugged. However, a poor performance of our method can be easily detected because it leads to poor data fitting and the decay of the data-misfit function along the iterations (convergence curve) does not exhibit an invariance along successive iterations.

On an irregular grid totaling 1,310,000 observation points, the field data over the Carajás Province, northern Brazil, would require 12.49 Terabytes of RAM to store the full sensitivity matrix to run the classical equivalent-layer technique. However, the fast convolutional equivalent-layer technique neither requires the full computation nor the storage of a sensitivity matrix. Taking advantage of the symmetric or skew-symmetric matrices structures, it is possible to reconstruct the whole sensitivity matrix using only 59.97 Megabytes. When performed on a standard laptop computer with an Intel Core i7 7700HQ@2.8GHz processor in single-processing and single-threading modes, the total time spent by our method for estimating the physical-property distribution over the equivalent layer was approximately 385.56 seconds and for upward-continuing the total of 1,310,000 magnetic observations was 2.64 seconds.

ACKNOWLEDGEMENTS

Diego Takahashi was supported by a Phd scholarship from CAPES. Valéria C.F. Barbosa was supported by fellowships from CNPq (grant 307135/2014-4) 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). The authors thank the Geological Survey of Brazil (CPRM) for providing the field data.

APPENDIX A

BTTB MATRIX-VECTOR PRODUCT

This appendix follows a similar approach to that presented by Takahashi et al. (2020) to efficiently compute the product of the sensitivity matrix \mathbf{A} (equations 13) and a generic vector \mathbf{b} . Let this product be represented by

$$\mathbf{t} = \mathbf{A} \mathbf{b} , \quad (\text{A-1})$$

where

$$\mathbf{t} = \mathbf{t}_{xx} + \mathbf{t}_{xy} + \mathbf{t}_{xz} + \mathbf{t}_{yy} + \mathbf{t}_{yz} + \mathbf{t}_{zz} \quad (\text{A-2})$$

and

$$\mathbf{t}_{\alpha\beta} = \mathbf{A}_{\alpha\beta} \mathbf{b} . \quad (\text{A-3})$$

Let us also consider that vectors

$$\mathbf{t}_{\alpha\beta} = \begin{bmatrix} \mathbf{t}_{\alpha\beta}^0 \\ \vdots \\ \mathbf{t}_{\alpha\beta}^{Q-1} \end{bmatrix}_{N \times 1} \quad (\text{A-4})$$

and

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}^0 \\ \vdots \\ \mathbf{b}^{Q-1} \end{bmatrix}_{N \times 1} \quad (\text{A-5})$$

are composed of $P \times 1$ vectors $\mathbf{t}_{\alpha\beta}^q$ and \mathbf{b}^q , respectively, where q is the block index (equations 32 and 34). From equation A-3, we obtain an auxiliary matrix-vector product given by

$$\mathbf{w}_{\alpha\beta} = \mathbf{C}_{\alpha\beta} \mathbf{v} , \quad (\text{A-6})$$

where $\mathbf{C}_{\alpha\beta}$ is a $4N \times 4N$ block circulant matrix with circulant blocks (BCCB) (e.g., Davis, 1979, p. 184),

$$\mathbf{w}_{\alpha\beta} = \begin{bmatrix} \mathbf{w}_{\alpha\beta}^0 \\ \vdots \\ \mathbf{w}_{\alpha\beta}^{Q-1} \\ \mathbf{0}_{2N \times 1} \end{bmatrix}_{4N \times 1}, \quad (\text{A-7})$$

$$\mathbf{w}_{\alpha\beta}^q = \begin{bmatrix} \mathbf{t}_{\alpha\beta}^q \\ \mathbf{0}_{P \times 1} \end{bmatrix}_{2P \times 1}, \quad (\text{A-8})$$

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}^0 \\ \vdots \\ \mathbf{v}^{Q-1} \\ \mathbf{0}_{2N \times 1} \end{bmatrix}_{4N \times 1} \quad (\text{A-9})$$

and

$$\mathbf{v}^q = \begin{bmatrix} \mathbf{b}^q \\ \mathbf{0}_{P \times 1} \end{bmatrix}_{2P \times 1}, \quad (\text{A-10})$$

with $\mathbf{0}_{2N \times 1}$ and $\mathbf{0}_{P \times 1}$ being vectors of zeros. The key point here is that the auxiliary matrix-vector product (equation A-6) represents a 2D discrete convolution and can be efficiently computed by using the 2D Fast Fourier Transform (2D FFT).

The BCCB matrix $\mathbf{C}_{\alpha\beta}$ (equation A-6) is formed by $2Q \times 2Q$ blocks, where each block $\mathbf{C}_{\alpha\beta}^q$ is a $2P \times 2P$ circulant matrix. The entire BCCB matrix $\mathbf{C}_{\alpha\beta}$ is defined by properly

downshifting its first block column

$$[\mathbf{C}_{\alpha\beta}]_{(0)} = \begin{bmatrix} \mathbf{C}_{\alpha\beta}^0 \\ \vdots \\ \mathbf{C}_{\alpha\beta}^{Q-1} \\ \mathbf{0}_{2P \times 2P} \\ \mathbf{C}_{\alpha\beta}^{-Q+1} \\ \vdots \\ \mathbf{C}_{\alpha\beta}^{-1} \end{bmatrix}_{4N \times 2P}, \quad (\text{A-11})$$

where $\mathbf{0}_{2P \times 2P}$ is a matrix of zeros. Similarly, each block $\mathbf{C}_{\alpha\beta}^q$, $q = -Q + 1, \dots, Q - 1$, is obtained by properly downshifting its first column

$$\mathbf{c}_{\alpha\beta}^q = \begin{bmatrix} a_{q0}^{\alpha\beta} \\ \vdots \\ a_{q(P-1)}^{\alpha\beta} \\ 0 \\ a_{q(-P+1)}^{\alpha\beta} \\ \vdots \\ a_{q(-1)}^{\alpha\beta} \end{bmatrix}_{2P \times 1}, \quad (\text{A-12})$$

where $a_{qp}^{\alpha\beta}$, $p = -P + 1, \dots, P - 1$, are the elements of matrix component $\mathbf{A}_{\alpha\beta}$ described in terms of block indices q and p (equations 32–35). The BCCB matrix $\mathbf{C}_{\alpha\beta}$ is diagonalized by $\mathbf{F}_{2Q} \otimes \mathbf{F}_{2P}$, where “ \otimes ” denotes the Kronecker product (e.g., Horn and Johnson, 1991, p. 242) and \mathbf{F}_{2Q} and \mathbf{F}_{2P} are the $2Q \times 2Q$ and $2P \times 2P$ unitary DFT matrices (Davis, 1979, p. 31). Due to this property, the auxiliary matrix-vector product (equation A-6) can be computed as follows (Takahashi et al., 2020):

$$\mathbf{F}_{2Q}^* [\mathbf{L}_{\alpha\beta} \circ (\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P})] \mathbf{F}_{2P}^* = \mathbf{W}_{\alpha\beta}, \quad (\text{A-13})$$

where “ \circ ” denotes the Hadamard (element-wise) product (e.g., Horn and Johnson, 1991, p. 298), “ $*$ ” denotes the complex conjugate, $\mathbf{W}_{\alpha\beta}$ and \mathbf{V} are $2Q \times 2P$ matrices obtained by rearranging, respectively, vectors $\mathbf{w}_{\alpha\beta}$ (equation A-7) and \mathbf{v} (equation A-9) along their rows and $\mathbf{L}_{\alpha\beta}$ is a $2Q \times 2P$ matrix given by

$$\mathbf{L}_{\alpha\beta} = \sqrt{4QP} \mathbf{F}_{2Q} \mathbf{G}_{\alpha\beta} \mathbf{F}_{2P}, \quad (\text{A-14})$$

with

$$\mathbf{G}_{\alpha\beta} = \begin{bmatrix} (\mathbf{c}_{\alpha\beta}^0)^{\top} \\ \vdots \\ (\mathbf{c}_{\alpha\beta}^{Q-1})^{\top} \\ \mathbf{0}_{1 \times 2P} \\ (\mathbf{c}_{\alpha\beta}^{-Q+1})^{\top} \\ \vdots \\ (\mathbf{c}_{\alpha\beta}^{-1})^{\top} \end{bmatrix}_{2Q \times 2P}, \quad (\text{A-15})$$

defined by the first columns $\mathbf{c}_{\alpha\beta}^q$ (equation A-12), $q = -Q + 1, \dots, Q - 1$, of all circulant blocks $\mathbf{C}_{\alpha\beta}^q$ (equation A-11). Hence, the whole BCCB matrix $\mathbf{C}_{\alpha\beta}$ does not have to be formed, but only its first column. Besides, the symmetries defined by equations 42–65 imply that all elements of $\mathbf{G}_{\alpha\beta}$ can be obtained by using only the first column of $\mathbf{A}_{\alpha\beta}$. Consequently, the whole matrices $\mathbf{A}_{\alpha\beta}$ do not have to be formed as well, but only their first columns.

It is important noting that the left side of equation A-13 represents the 2D Inverse Discrete Fourier Transform (2D IDFT) of the term in brackets. This term, in turn, represents the Hadamard product of $\mathbf{L}_{\alpha\beta}$ (equation A-14) and the 2D Discrete Fourier Transform (2D DFT) of \mathbf{V} . Similarly, equation A-14 shows that $\mathbf{L}_{\alpha\beta}$ is obtained by computing the 2D DFT of matrix $\mathbf{G}_{\alpha\beta}$ (equation A-15). Hence, equations A-13 and A-14 can be efficiently

computed by using the 2D FFT. After that, the elements of vector $\mathbf{t}_{\alpha\beta}$ (equation A-3) can be retrieved from the first quadrant of matrix $\mathbf{W}_{\alpha\beta}$ (equation A-13). By combining the results obtained for all components $\alpha\beta$, $\alpha, \beta = x, y, z$, we can show that

$$\mathbf{F}_{2Q}^* [\mathbf{L} \circ (\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P})] \mathbf{F}_{2P}^* = \mathbf{W}, \quad (\text{A-16})$$

where

$$\mathbf{W} = \mathbf{W}_{xx} + \mathbf{W}_{xy} + \mathbf{W}_{xz} + \mathbf{W}_{yy} + \mathbf{W}_{yz} + \mathbf{W}_{zz} \quad (\text{A-17})$$

and

$$\mathbf{L} = \mathbf{L}_{xx} + \mathbf{L}_{xy} + \mathbf{L}_{xz} + \mathbf{L}_{yy} + \mathbf{L}_{yz} + \mathbf{L}_{zz}, \quad (\text{A-18})$$

with $\mathbf{L}_{\alpha\beta}$ defined by equation A-14. Then, the elements of \mathbf{t} (equation A-1) are obtained from the first quadrant of \mathbf{W} (equations A-16 and A-17).

Finally, it can be shown that the product

$$\mathbf{t} = \mathbf{A}^\top \mathbf{b} \quad (\text{A-19})$$

can be computed by using equation A-16. The difference is that, in this case, matrices $\mathbf{G}_{\alpha\beta}$ (equation A-15) are defined by using the new vectors

$$\mathbf{c}_{\alpha\beta}^q = \begin{bmatrix} a_{q0}^{\alpha\beta} \\ \vdots \\ a_{q(-P+1)}^{\alpha\beta} \\ 0 \\ a_{q(P-1)}^{\alpha\beta} \\ \vdots \\ a_{q1}^{\alpha\beta} \end{bmatrix}_{2P \times 1}. \quad (\text{A-20})$$

REFERENCES

- Aster, R. C., B. Borchers, and C. H. Thurber, 2019, Parameter estimation and inverse problems, 3 ed.: Elsevier.
- Blakely, R. J., 1996, Potential theory in gravity and magnetic applications, 1 ed.: Cambridge University Press.
- Bott, M., 1967, Solution of the linear inverse problem in magnetic interpretation with application to oceanic magnetic anomalies: *Geophysical Journal International*, **13**, 313–323.
- Chan, R. H.-F., and X.-Q. Jin, 2007, An introduction to iterative Toeplitz solvers: SIAM, 5.
- Dampney, C. N. G., 1969, The equivalent source technique: *Geophysics*, **34**, no. 1, 39–53.
- Danes, Z., 1961, Structure calculations from gravity data and density logs: *Mining Trans.*, 223 C.
- Davis, P. J., 1979, Circulant matrices: John Wiley & Sons, Inc.
- Dean, W. C., 1958, Frequency analysis for gravity and magnetic interpretation: *Geophysics*, **23**, 97–127.
- Emilia, D. A., 1973, Equivalent sources used as an analytic base for processing total magnetic field profiles: *Geophysics*, **38**, no. 2, 339–348.
- Golub, G. H., and C. F. V. Loan, 2013, Matrix computations (Johns Hopkins studies in the mathematical sciences), 4 ed.: Johns Hopkins University Press.
- Graham, L., D. E. Knuth, and O. Patashnik, 1994, Concrete mathematics: a foundation for computer science, 2 ed.: Addison-Wesley Publishing Company.
- Gunn, P. J., 1975, Linear transformations of gravity and magnetic fields: *Geophysical Prospecting*, **23**, 300–312.

- Hansen, R. O., and Y. Miyazaki, 1984, Continuation of potential fields between arbitrary surfaces: *Geophysics*, **49**, no. 6, 787–795.
- Hogue, J. D., R. A. Renaut, and S. Vatankhah, 2020, A tutorial and open source software for the efficient evaluation of gravity and magnetic kernels: *Computers & Geosciences*, **144**, 104575.
- Horn, R. A., and C. R. Johnson, 1991, Topics in matrix analysis, 1 ed.: Cambridge University Press.
- Kellogg, O. D., 1929, Foundations of potential theory: Frederick Ungar Publishing Company.
- Leão, J. W. D., and J. B. C. Silva, 1989, Discrete linear transformations of potential field data: *Geophysics*, **54**, no. 4, 497–507.
- Levinson, N., 1946, The wiener (root mean square) error criterion in filter design and prediction: *Journal of Mathematics and Physics*, **25**, no. 1-4, 261–278.
- Li, Y., M. Nabighian, and D. W. Oldenburg, 2014, Using an equivalent source with positivity for low-latitude reduction to the pole without striation: *Geophysics*, **79**, J81–J90.
- Li, Y., and D. W. Oldenburg, 2010, Rapid construction of equivalent sources using wavelets: *Geophysics*, **75**, no. 3, L51–L59.
- Oliveira Jr., V. C., V. C. F. Barbosa, and L. Uieda, 2013, Polynomial equivalent layer: *Geophysics*, **78**, no. 1, G1–G13.
- Peters, L. J., 1949, The direct approach to magnetic interpretation and its practical application: *Geophysics*, **14**, 290–320.
- Renaut, R. A., J. D. Hogue, S. Vatankhah, and S. Liu, 2020, A fast methodology for large-scale focusing inversion of gravity and magnetic data using the structured model matrix and the 2-d fast fourier transform: *Geophysical Journal International*, **223**, 1378–1397.

- Silva, J. B. C., 1986, Reduction to the pole as an inverse problem and its application to low-latitude anomalies: *Geophysics*, **51**, no. 2, 369–382.
- Siqueira, F. C., V. C. Oliveira Jr, and V. C. Barbosa, 2017, Fast iterative equivalent-layer technique for gravity data processing: A method grounded on excess mass constraint: *Geophysics*, **82**, no. 4, G57–G69.
- Sun, S., C. Chen, and Y. Liu, 2019, Constrained 3d inversion of magnetic data with structural orientation and borehole lithology: A case study in the macheng iron deposit, hebei, china: *Geophysics*, **84**, B121–B133.
- Takahashi, D., V. C. Oliveira Jr, and V. C. Barbosa, 2020, Convolutional equivalent layer for gravity data processing: *Geophysics*, **85**, G129–G141.
- Trench, W. F., 1964, An algorithm for the inversion of finite Toeplitz matrices: *Journal of the Society for Industrial and Applied Mathematics*, **12**, no. 3, 515–522.
- Van Loan, C., 1992, Computational frameworks for the Fast Fourier Transform: SIAM.
- Zhang, Y., and Y. S. Wong, 2015, BTTB-based numerical schemes for three-dimensional gravity field inversion: *Geophysical Journal International*, **203**, no. 1, 243–256.
- Zhang, Y., Y. S. Wong, and Y. Lin, 2016, BTTB–RRCG method for downward continuation of potential field data: *Journal of Applied Geophysics*, **126**, 74–86.

LIST OF TABLES

1 This table shows the RAM memory usage (in Megabytes) for storing the whole $N \times N$ matrix \mathbf{A} (equation 13), the first columns of the BCCB matrices $\mathbf{C}_{\alpha\beta}$ (equation A-6) (both need 8 bytes per element) and the matrix \mathbf{L} (equation A-18) (16 bytes per element).

LIST OF FIGURES

1 Schematic representation of an $N_x \times N_y$ regular grid of points (black dots) with $N_x = 3$ and $N_y = 2$, where each point has an associated index. This index may represent i or j , that are associated with observation points (x_i, y_i, z_0) and equivalent sources (x_j, y_j, z_c) . Left panel shows an example of x -oriented grid, with indices varying along x -axis, while right panel shows an example of y -oriented grid, with indices varying along y -axis.

2 Number of flops associated with classical method (equation 66), the standard CGLS method (equation 67) and our method (equation 68, all of them with $N^{it} = 50$). The number of observation points N varies from 5,000 to 1,000,000.

3 Comparison between the runtime of the equivalent-layer technique using the classical method, standard CGLS method and our method. The values for the standard CGLS and our method use $N^{it} = 50$ iterations.

4 Observed total-field anomaly over the Carajás Province, northen Brazil. The aeromagnetic survey was flown in 131 north-south flight lines at an average altitude of 900 m, totaling $N = 6,081,345$ observation points.

5 Observed total-field anomalies over the Carajás Province, northen Brazil, considering: (a) a regular grid ($10,000 \times 131$, totaling $N = 1,310,000$ observation points) of interpolated values from the original irregularly sampled data (Figure 4); and (b) an irregular grid ($10,000 \times 131$, totaling $N = 1,310,000$ observation points) of decimated values from the the original irregularly sampled data (Figure 4).

6 Results using our method applied to the interpolated $10,000 \times 131$ regular grid (Figure 5a) - (a) Predicted data and (b) Data residuals, defined as the difference between the observed (Figure 5a) and the predicted data (panel a), with mean of 0.0762 nT and standart deviation of 0.4886 nT.

7 Results using our method applied to the decimated $10,000 \times 131$ irregular grid
(Figure 5b) - (a) Predicted data and (b) Data residuals, defined as the difference between
the observed (Figure 5b) and the predicted data (panel a), with mean of 0.0717 nT and
standard deviation of 0.3144 nT.

8 Convergence curve using our method to the decimated irregular grid of the real
data of Carajás Province, Brazil.

9 Upward continuation transformation of real data of Carajás Province, Brazil at
5, 000 m after 2.64 seconds.

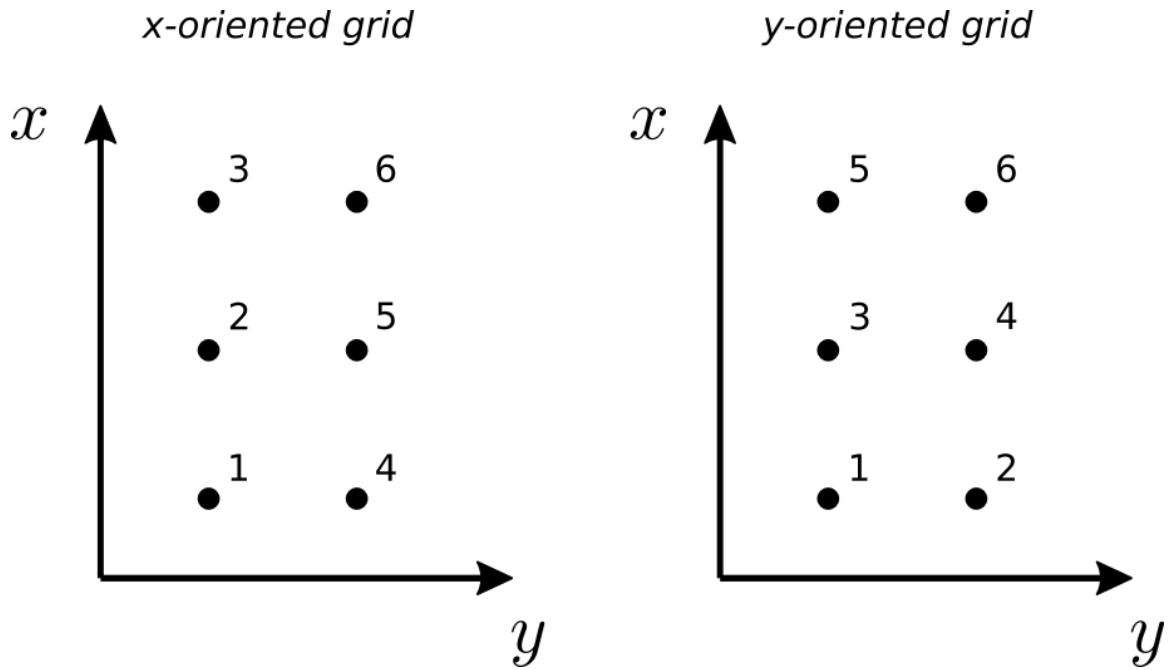


Figure 1: Schematic representation of an $N_x \times N_y$ regular grid of points (black dots) with $N_x = 3$ and $N_y = 2$, where each point has an associated index. This index may represent i or j , that are associated with observation points (x_i, y_i, z_0) and equivalent sources (x_j, y_j, z_c) . Left panel shows an example of x -oriented grid, with indices varying along x -axis, while right panel shows an example of y -oriented grid, with indices varying along y -axis.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

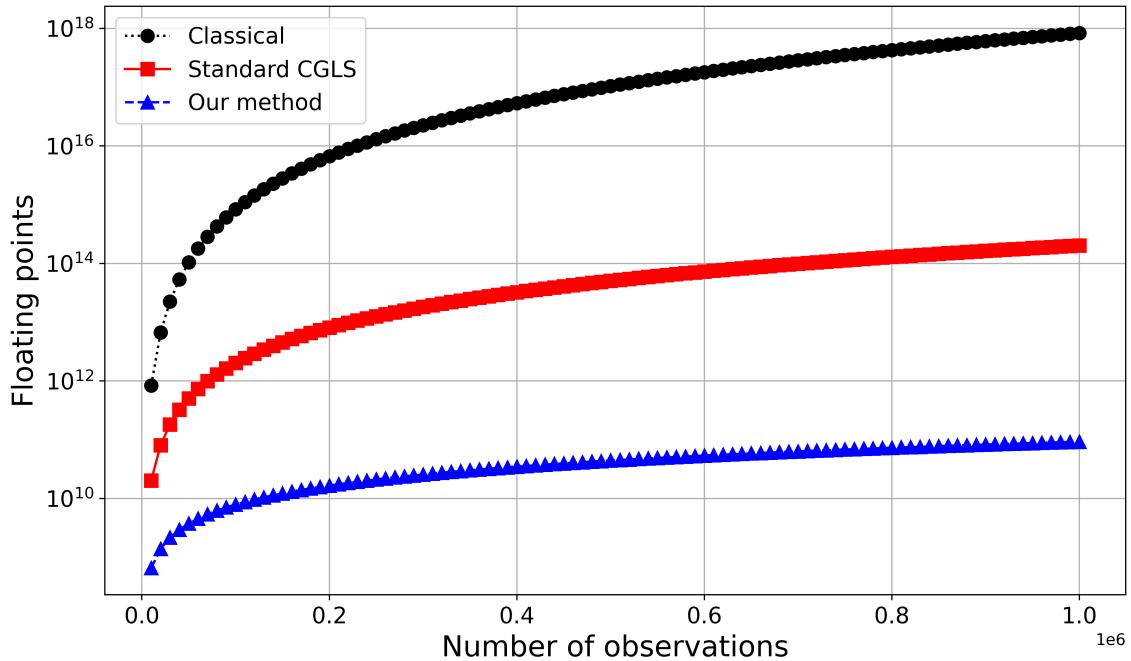


Figure 2: Number of flops associated with classical method (equation 66), the standard CGLS method (equation 67) and our method (equation 68, all of them with $N^{it} = 50$). The number of observation points N varies from 5,000 to 1,000,000.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

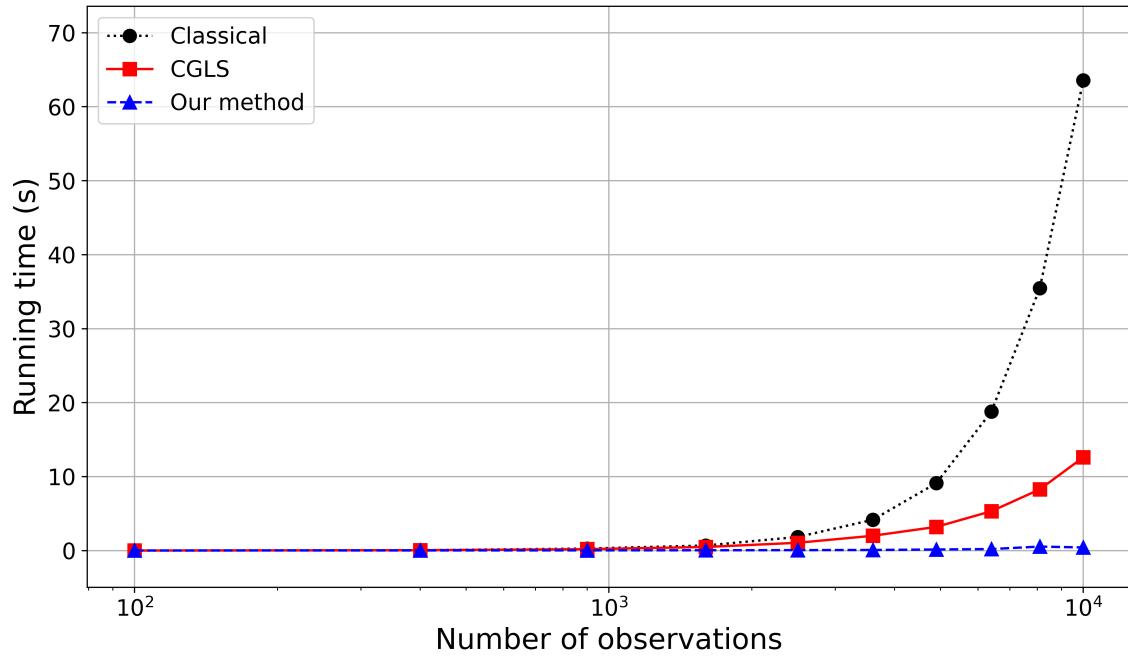


Figure 3: Comparison between the runtime of the equivalent-layer technique using the classical method, standard CGLS method and our method. The values for the standard CGLS and our method use $N^{it} = 50$ iterations.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

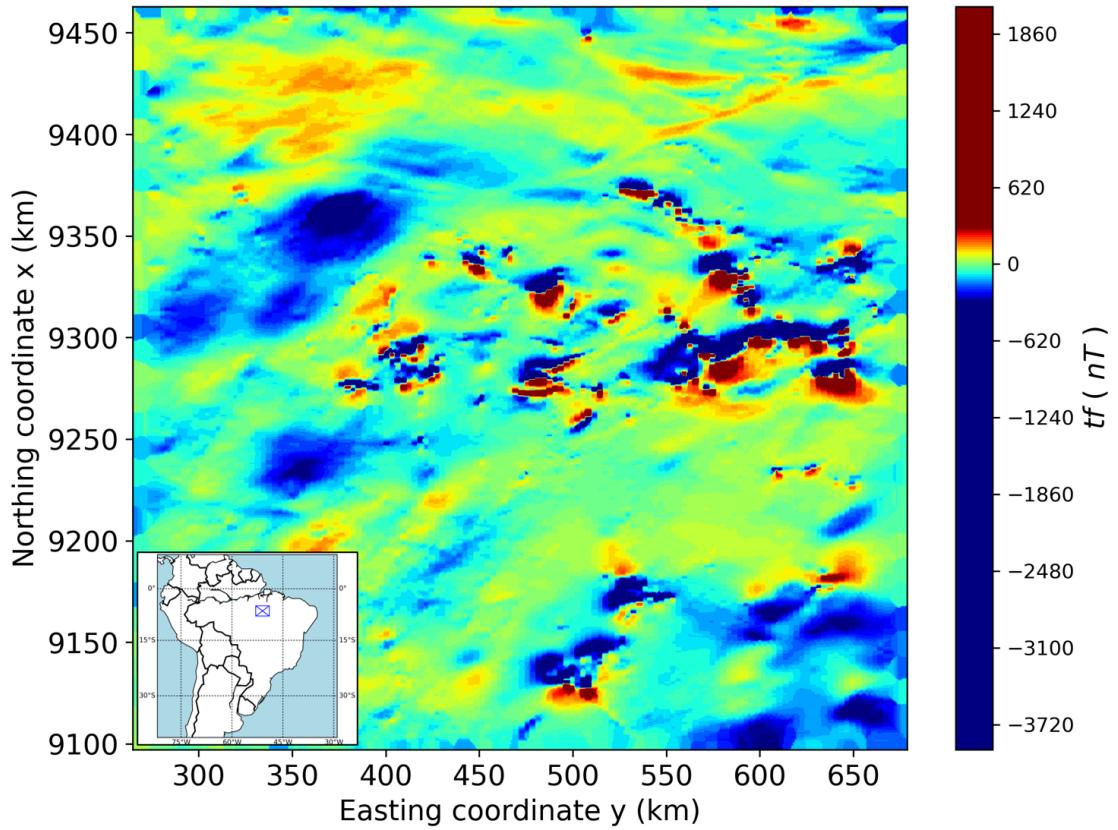


Figure 4: Observed total-field anomaly over the Carajás Province, northern Brazil. The aeromagnetic survey was flown in 131 north-south flight lines at an average altitude of 900 m, totaling $N = 6,081,345$ observation points.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

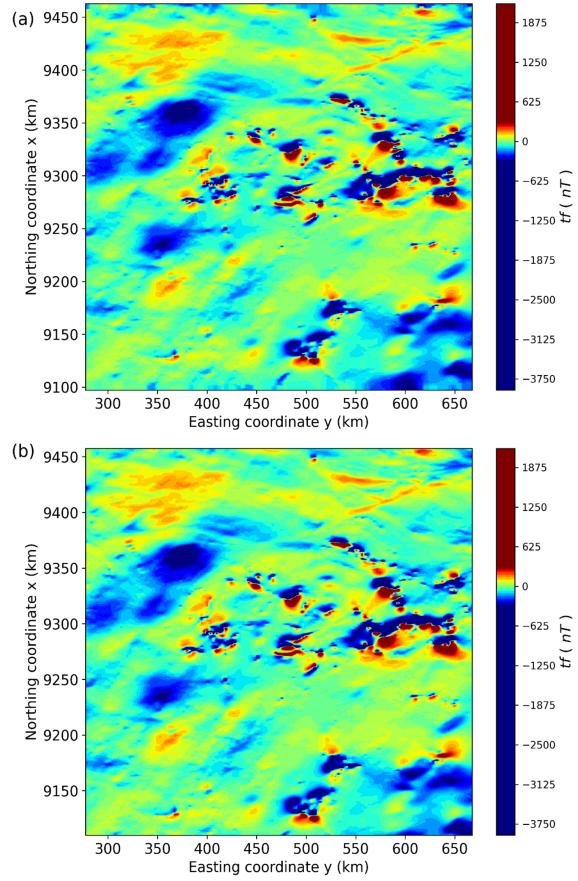


Figure 5: Observed total-field anomalies over the Carajás Province, northern Brazil, considering: (a) a regular grid ($10,000 \times 131$, totaling $N = 1,310,000$ observation points) of interpolated values from the original irregularly sampled data (Figure 4); and (b) an irregular grid ($10,000 \times 131$, totaling $N = 1,310,000$ observation points) of decimated values from the original irregularly sampled data (Figure 4).

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

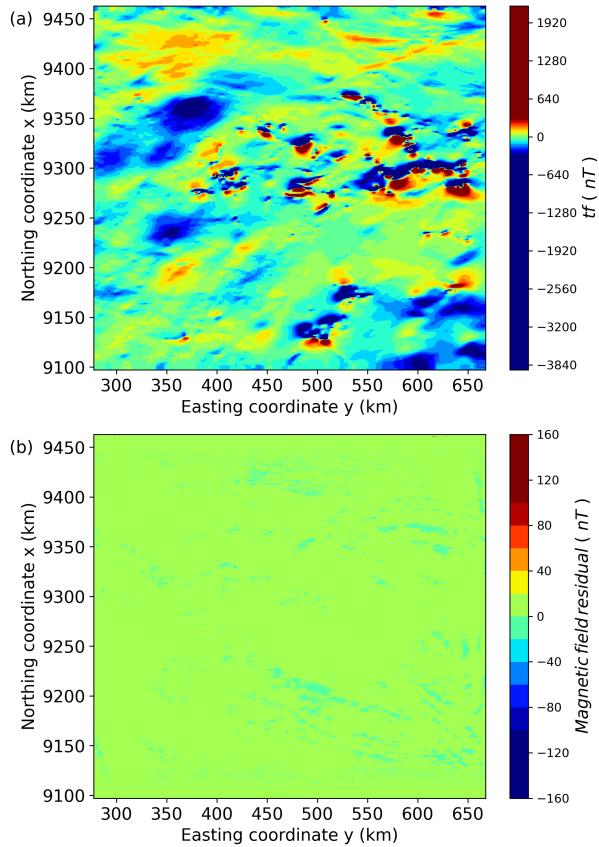


Figure 6: Results using our method applied to the interpolated $10,000 \times 131$ regular grid (Figure 5a) - (a) Predicted data and (b) Data residuals, defined as the difference between the observed (Figure 5a) and the predicted data (panel a), with mean of 0.0762 nT and standard deviation of 0.4886 nT.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

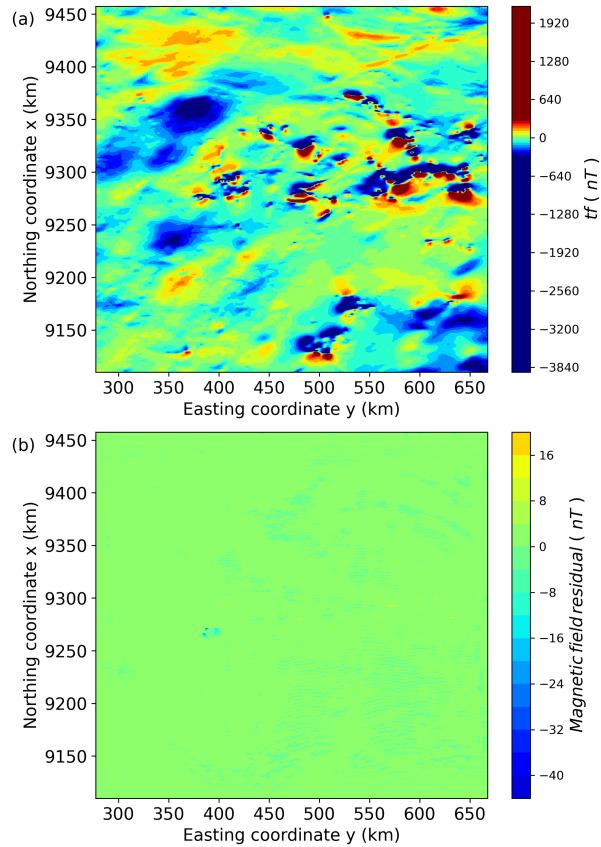


Figure 7: Results using our method applied to the decimated $10,000 \times 131$ irregular grid (Figure 5b) - (a) Predicted data and (b) Data residuals, defined as the difference between the observed (Figure 5b) and the predicted data (panel a), with mean of 0.0717 nT and standard deviation of 0.3144 nT.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

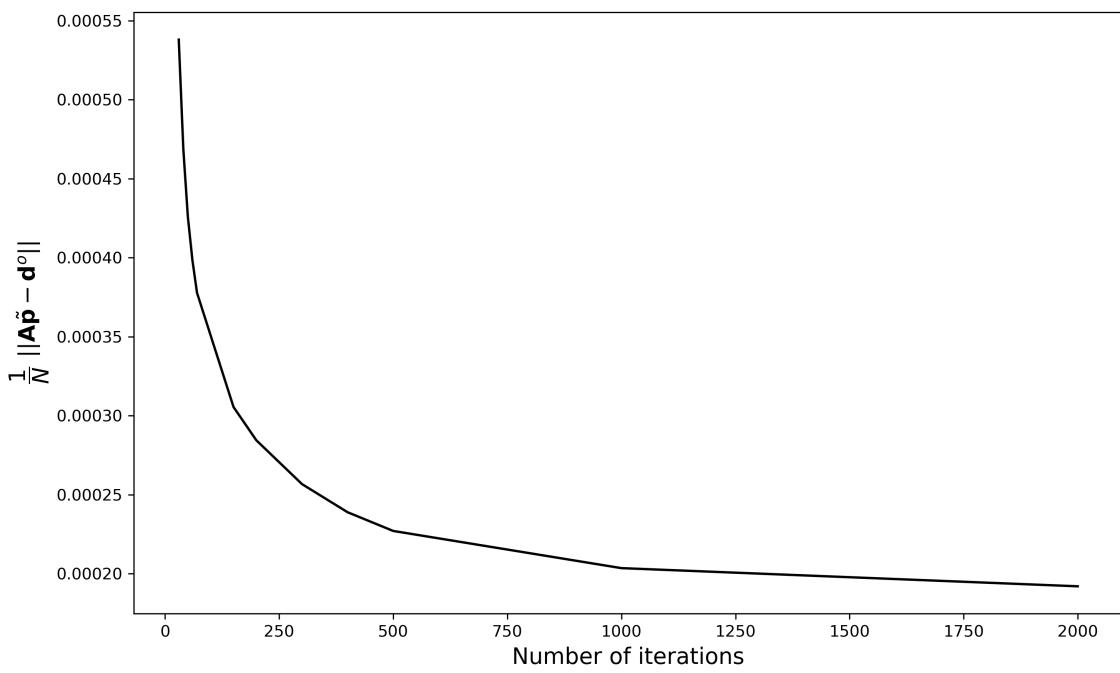


Figure 8: Convergence curve using our method to the decimated irregular grid of the real data of Carajás Province, Brazil.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

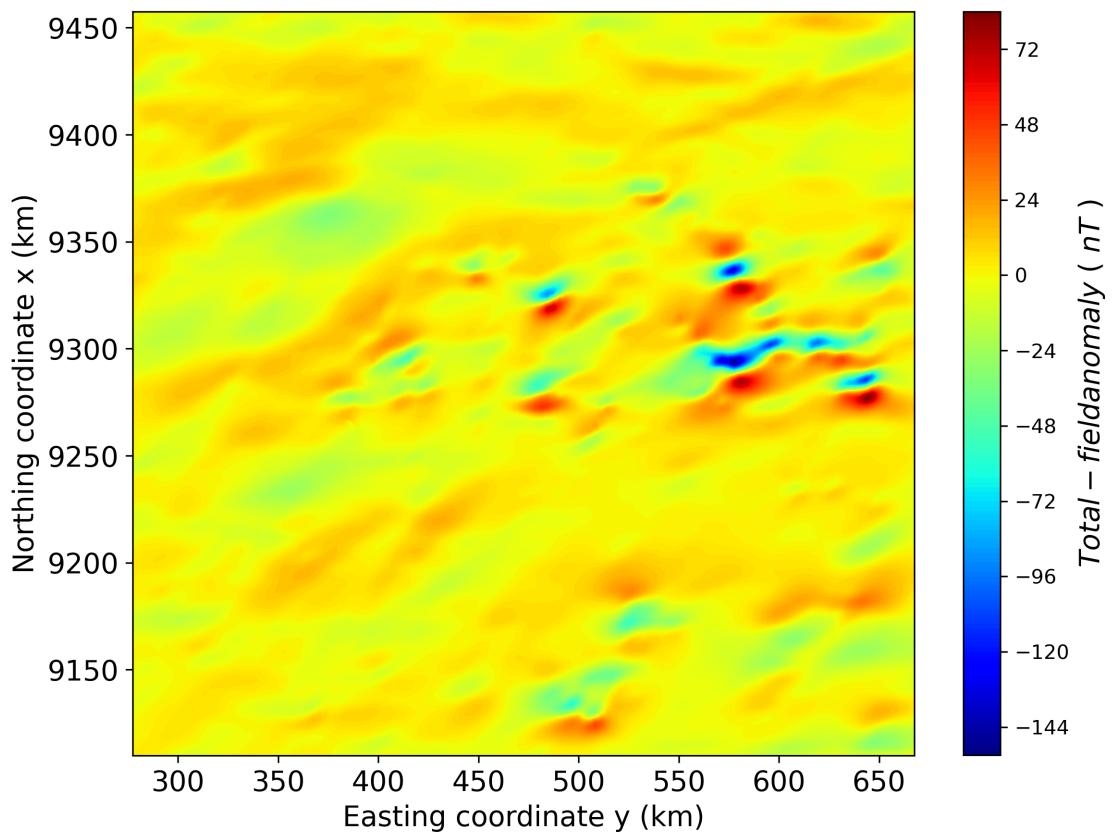


Figure 9: Upward continuation transformation of real data of Carajás Province, Brazil at 5,000 m after 2.64 seconds.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

N	A	First columns of matrices $\mathbf{C}_{\alpha\beta}$	L
100	0.0763	0.0183	0.00610
400	1.22	0.0744	0.0248
2,500	48	0.458	0.1528
10,000	763	1.831	0.6104
40,000	12,207	7.32	2.4416
250,000	476,837	45.768	15.3
500,000	1,907,349	91.56	30.518
1,000,000	7,629,395	183.096	61.035

Table 1: This table shows the RAM memory usage (in Megabytes) for storing the whole $N \times N$ matrix **A** (equation 13), the first columns of the BCCB matrices $\mathbf{C}_{\alpha\beta}$ (equation A-6) (both need 8 bytes per element) and the matrix **L** (equation A-18) (16 bytes per element).