

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: *vanderlei@on.br*

(September 3, 2021)

GEO-XXXX

Running head: Magnetic convolutional equivalent layer

ABSTRACT

A fast equivalent layer for magnetic data processing is presented. Taking advantage of the sensitivity matrix structure of the magnetic kernel, when observation and equivalent sources are aligned on a regular spaced grid with constant height, it is possible to calculate the matrix-vector product in a very fast manner. The structure is called block-Toeplitz Toeplitz-block (BTTB) and this type of matrix is well known in literature to be embedded in a block-Circulant Circulant-block (BCCB), which in turn can have its eigenvalues calculated using only the first column and a 2D fast Fourier transform. We show that, despite this BTTB matrix is not symmetric, by using only the first equivalent source it is possible to calculate all the first columns of the inverse of distance second derivatives that composes the magnetic kernel and reconstruct the first column of the BCCB matrix, saving computational time and system memory. The conjugate gradient iterative method is used to solve the linear system and estimate the physical properties distributed over the equivalent layer. Synthetic tests show a decrease in the order of 10^4 in floating-point operations, 25 times in computation runtime with a mid-size 80×80 grid, and exponential decrease in memory RAM

usage, allowing to perform this operation with millions of observation points on desktop computers. Synthetic tests with irregular grids also show that this method can work with directional disturbances under certain limits. A real magnetic data of Carajás Province, Brazil, with 1,310,000 observation points in an irregular grid was used to successfully perform a data processing with this method, taking 385.56 seconds to estimate the physical property and 2.64 seconds for the upward-continuation.

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, ? 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. ? 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. ? 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 (?; see Table I, p 113) for some theoretical geophysical operations (e.g., derivatives and upward and downward continuations). ? 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 ? and detailed explanations can also be found in ?. Although the equivalent-layer technique has been known since the 1960s in geophysical literature (???), 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 (?), reduction to

the pole (??), upward/downward continuations (??) and total magnetic induction vector components calculation (?).

Together with the rise in computational processing power, some works tried new implementations to increase the efficiency of the equivalent layer. In ? 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 ? 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. ? divided the equivalent layer into a grid of fixed source windows. Instead of directly calculating the the physical-property distribution of a finite set of equivalent sources (e.g., dipoles, point of masses) arranged in the entire equivalent layer, ? 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, ? proposed an iterative method for processing large gravimetric data using the equivalent layer without requiring the solution of a linear system. In ?, 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 ? and by ?. Currently, the iterative method of conjugate gradient is used in most cases, in ? 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 (?) and for 3D gravity-data inversion using a 2D multilayer model (?). More recently, ? provided an overview on modeling the gravity and magnetic kernels using the BTTB structures and ? used BTTB the structures for inversion of both gravity and magnetic data to recover sparse subsurface structures. ? combined the fast equivalent source technique presented by ? with the concept of symmetric block-Toeplitz Toeplitz-block (BTTB) matrices to introduce the convolutional equivalent layer for gravimetric data technique. ? 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 ? method was able to efficiently process very large gravity data sets. Moreover, ? 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 ?, 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

The total-field anomaly

The Earth's magnetic field is commonly divided in three parts: main field, crustal field and external field. The main field is generated in the outer core in a process of self-sustaining dynamo, the crustal field is generated by magnetic bodies in the lithosphere and the external field is generated by electrical currents in the ionosphere and magnetosphere. For exploration geophysics, the crustal field is the object of study, which makes the separation of this data from the acquisition dataset a very important step.

The combination of the main field and crustal field is known as internal field or total-field. Taking the difference between the main field given by a model (e.g. IGRF) and this internal field, at the same location, we have the total-field anomaly.

Let $\Delta T(x_i, y_i, z_i), i = 1, \dots, N$, be a observed dataset in a region considering a right-handed Cartesian coordinate system with the x -axis pointing north, y -axis pointing east and z -axis pointing downward. The total-field anomaly at the i th observation can be approximated to:

$$\Delta T(x_i, y_i, z_i) = \hat{\mathbf{F}}^\top \mathbf{B}(x_i, y_i, z_i), \quad (1)$$

where, $\mathbf{B}(x_i, y_i, z_i)$ is the crustal field, $\hat{\mathbf{F}}^\top$ is the transposed unitary vector with the main field directions, with $\hat{\mathbf{F}}$ described as:

$$\hat{\mathbf{F}} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \cos(I_0) \cos(D_0) \\ \cos(I_0) \sin(D_0) \\ \sin(I_0) \end{bmatrix}, \quad (2)$$

where I_0 is the inclination and D_0 the declination of the main field, respectively.

Considering a uniform magnetized body with volume v and a total magnetization vector \mathbf{m} , the induced magnetic field at the i th observation is:

$$\mathbf{B}(x_i, y_i, z_i) = c_m \frac{\mu_0}{4\pi} \mathbf{M}(x_i, y_i, z_i) \mathbf{m}, \quad (3)$$

where, $\mu_0 = 4\pi 10^{-7}$ H/m is the magnetic constant, $c_m = 10^9$ is a constant that transforms the induced magnetic field from Tesla (T) to nanotesla (nT) and $\mathbf{M}(x_i, y_i, z_i)$ is a 3×3 matrix given by:

$$\mathbf{M}(x_i, y_i, z_i) = \begin{bmatrix} \partial_{xx}\phi(x_i, y_i, z_i) & \partial_{xy}\phi(x_i, y_i, z_i) & \partial_{xz}\phi(x_i, y_i, z_i) \\ \partial_{xy}\phi(x_i, y_i, z_i) & \partial_{yy}\phi(x_i, y_i, z_i) & \partial_{yz}\phi(x_i, y_i, z_i) \\ \partial_{xz}\phi(x_i, y_i, z_i) & \partial_{yz}\phi(x_i, y_i, z_i) & \partial_{zz}\phi(x_i, y_i, z_i) \end{bmatrix}, \quad (4)$$

where, $\partial_{\alpha\beta}\phi(x_i, y_i, z_i)$ with $\alpha = x, y, z$ and $\beta = x, y, z$, are the second derivatives of the function $\phi(x_i, y_i, z_i)$ with respect to x, y and z :

$$\phi(x_i, y_i, z_i) = \int_v \int_v \int_v \frac{1}{r} dv, \quad (5)$$

where

$$\frac{1}{r} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}} \quad (6)$$

and x_j, y_j, z_j are the j th Cartesian coordinates within the volume element of the magnetized body with volume v , where the integral (equation 5) is conducted.

Rewriting the equation 1 using the magnetic induction given by equation 3 the total-field anomaly $\Delta T(x, y, z)$ is given by:

$$\Delta T(x_i, y_i, z_i) = c_m \frac{\mu_0}{4\pi} m \hat{\mathbf{F}}^\top \mathbf{M}(x_i, y_i, z_i) \hat{\mathbf{m}}, \quad (7)$$

where m is the magnetization intensity and $\hat{\mathbf{m}}$ is the directional unitary vector.

Equivalent layer for magnetic data

Grounded on the equivalent layer principle it is possible to calculate the total-field anomaly $\Delta T(x_i, y_i, z_i)$ (equation 7) with the convolution between the harmonic function and the physical property:

$$\Delta T(x_i, y_i, z_i) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x_j, y_j, z_c) \left[c_m \frac{\mu_0}{4\pi} \hat{\mathbf{F}}^\top \mathbf{H} \hat{\mathbf{h}} \right] dx dy, \quad (8)$$

where z_c is a constant representing the depth of the equivalent layer with $z_i < z_c$. The unitary vector $\hat{\mathbf{h}}$ is the magnetization directions of the equivalent sources over the layer:

$$\hat{\mathbf{h}} = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \begin{bmatrix} \cos(I) \cos(D) \\ \cos(I) \sin(D) \\ \sin(I) \end{bmatrix}, \quad (9)$$

where I and D are, respectively, the inclination and declination of the equivalent sources, \mathbf{H} is a 3×3 matrix that contains the second derivatives in relation to the observed Cartesian coordinates x, y, z as presented in equation 6:

$$\mathbf{H} = \begin{bmatrix} \partial_{xx} \frac{1}{r} & \partial_{xy} \frac{1}{r} & \partial_{xz} \frac{1}{r} \\ \partial_{xy} \frac{1}{r} & \partial_{yy} \frac{1}{r} & \partial_{yz} \frac{1}{r} \\ \partial_{xz} \frac{1}{r} & \partial_{yz} \frac{1}{r} & \partial_{zz} \frac{1}{r} \end{bmatrix} = \begin{bmatrix} H_{xx} & H_{xy} & H_{xz} \\ H_{xy} & H_{yy} & H_{yz} \\ H_{xz} & H_{yz} & H_{zz} \end{bmatrix}, \quad (10)$$

and the physical property $p(x_j, y_j, z_c)$ represents the j th magnetic dipole moment (in Am^2) produced by the j th dipole locate over the equivalent layer at the j th Cartesian coordinates (x_j, y_j, z_c) .

Discretizing equation 8 we get:

$$\Delta T(x_i, y_i, z_i) = \sum_{j=1}^M p_j a_{ij}, \quad (11)$$

where the subscript j denotes a discrete equivalent source, totaling M equivalent sources that are distributed over the layer and a_{ij} is given by:

$$a_{ij} = c_m \frac{\mu_0}{4\pi} \hat{\mathbf{F}}^\top \mathbf{H}_{ij} \hat{\mathbf{h}}. \quad (12)$$

Equation 11 can be written in matrix notation as:

$$\mathbf{d}(\mathbf{p}) = \mathbf{A}\mathbf{p}, \quad (13)$$

where $\mathbf{d}(\mathbf{p})$ is the N -dimensional vector of total-field anomaly ($\Delta T(x_i, y_i, z_i), i = 1, \dots, N$), \mathbf{A} is a matrix containing the elements given by equation 12, also known as the sensitivity matrix and \mathbf{p} is the vector containing the dipole moments of each equivalent source.

Let \mathbf{d}^o be an N -dimensional vector that contains the observed total-field anomaly. By solving the least-squares normal equation associated with equation 13, we get

$$\mathbf{A}^\top \mathbf{A}\mathbf{p} = \mathbf{A}^\top \mathbf{d}^o, \quad (14)$$

and we estimate the parameter vector that contains the unknown dipole moments over the equivalent layer, i.e.,

$$\hat{\mathbf{p}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{d}^o \quad (15)$$

Equation 15 will be referenced throughout this work as the classical method for solving the equivalent layer.

Conjugate Gradient Least Square method (CGLS)

As an alternative from the clasical method of parameter estimative, the conjugate gradient (CG) is a well-known iterative method for solving linear systems with symmetric positive definite matrices. By minimizing the quadratic form:

$$\Phi(\mathbf{p}) = \frac{1}{2} \mathbf{p}^\top \mathbf{A} \mathbf{p} - \mathbf{d}^o \mathbf{p}, \quad (16)$$

it is possible to solve the system by constructing a basis of conjugate directions $c \in R^N$ (?). As we are solving a general least square problem and matrix \mathbf{A} (equation 12) is not symmetric, instead we minimize:

$$\|\mathbf{A} \mathbf{p} - \mathbf{d}^o\|_2, \quad (17)$$

by applying the conjugate gradient to the normal equations (equation 14).

In theory, this method is bound to converge after at most N iterations, but in a later part of this work we show with numerical results that the convergence is much faster for the linear system we are solving.

A pseudocode for the CGLS method follows:

Algorithm 1 Conjugate Gradient Least Square pseudocode (? , p. 155).

Input: $\mathbf{A} \in R^{N \times M}$ and $\mathbf{d}^o \in R^N$.

Output: Estimative of parameter vector $\hat{\mathbf{p}}$.

Let $it = 0$, $\hat{\mathbf{p}}^{(0)} = \mathbf{0}$, $\mathbf{c}^{(-1)} = \mathbf{0}$, $\beta_0 = 0$, $\mathbf{s}^{(0)} = \mathbf{d}^o - \mathbf{A}\hat{\mathbf{p}}^{(0)}$ and $\mathbf{r}^{(0)} = \mathbf{A}^\top \mathbf{s}^{(0)}$.

$$1 - \text{If } it > 0, \text{ let } \beta_{(it)} = \frac{\mathbf{r}^{(it)\top} \mathbf{r}^{(it)}}{\mathbf{r}^{(it-1)\top} \mathbf{r}^{(it-1)}}$$

$$2 - \mathbf{c}^{(it)} = \mathbf{r}^{(it)} - \alpha_{(it)} \beta_{(it)} \mathbf{c}^{(it-1)}.$$

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

$$4 - \hat{\mathbf{p}}^{(it+1)} = \hat{\mathbf{p}}^{(it)} - \alpha_{(it)} \mathbf{c}^{(it)}.$$

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

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

$$7 - it = it + 1.$$

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

Different from the classical least-square solution (equation 15), the CGLS solution (Algorithm 1) requires neither inverse matrix nor matrix-matrix product. The CGLS method

only requires: one matrix-vector multiplication out of the loop and two matrix-vector multiplications, in steps 3 and 6, at each iteration.

In this work, we will reduce the computational cost of the equivalent layer by substituting exactly these two matrix-vector products with a much more efficient algorithm.

Conjugate Gradient Least Square method convergence criteria

Setting a 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 interactions, which would increase algorithm runtime, but with smaller residuals. We chose the first option, as we achieve satisfactory results.

Non-symmetric Block-Toeplitz Toeplitz-Block structure of matrix A

Let us consider that 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. The notation used in this work will be the same as the one presented in ?, where the authors described the structure of the symmetric Block-Toeplitz Toeplitz-Block matrix of the gravimetric equivalent layer. Here, we also establish a relation between the pair of *matrix coordinates* (x_i, y_i) , $i = 1, \dots, N$ or (x_j, y_j) , $j = 1, \dots, M = N$ and a pair of *grid coordinates* (x_k, y_l) given as:

$$x_i \equiv x_k = x_1 + [k(i) - 1] \Delta x , \quad (18)$$

and

$$y_i \equiv y_l = y_1 + [l(i) - 1] \Delta y . \quad (19)$$

In a *x-oriented grid* the indices i (or j) relate as integer functions of k and l by:

$$i(k, l) = (l - 1) N_x + k , \quad (20)$$

$$l(i) = \left\lceil \frac{i}{N_x} \right\rceil \quad (21)$$

and

$$k(i) = i - \left\lceil \frac{i}{N_x} \right\rceil N_x + N_x . \quad (22)$$

For *y-oriented grid* they are given by:

$$i(k, l) = (k - 1) N_y + l , \quad (23)$$

$$k(i) = \left\lceil \frac{i}{N_y} \right\rceil \quad (24)$$

and

$$l(i) = i - \left\lceil \frac{i}{N_y} \right\rceil N_y + N_y , \quad (25)$$

where in equations 21 to 25, $\lceil \cdot \rceil$ is the ceiling function. Figure 1 shows an example of a grid $N_x \times N_y$, where $N_x = 4$ and $N_y = 3$ demonstrating the relation between the *matrix coordinates* with $k(i)$ and $l(i)$ depending on the orientation of the grid.

The $N \times M$ sensitivity matrix \mathbf{A} (equation 12) can be represented as a grid of $Q \times Q$ blocks \mathbf{A}^q , $q = -Q + 1, \dots, 0, \dots, Q - 1$. Each block \mathbf{A}^q has $P \times P$ elements a_p^q where $p = -P + 1, \dots, 0, \dots, P - 1$

In a *x-oriented grid* q and p give the number of blocks ($Q = N_y$) and the number of elements of each block ($P = N_x$). They can be defined by the integer functions:

$$q(i, j) = l(i) - l(j) \quad (26)$$

and

$$p(i, j) = k(i) - k(j) , \quad (27)$$

where equations 21 and 22 describe $l(i)$ and $l(j)$ and $k(i)$ and $k(j)$, respectively. When using y -oriented grids, q and p still define the number of block and block elements, but $Q = N_x$ and $P = N_y$. Moreover, the integer functions changes to:

$$q(i, j) = k(i) - k(j) \quad (28)$$

and

$$p(i, j) = l(i) - l(j), \quad (29)$$

where equation 24 now defines $k(i)$ and $k(j)$ and equation 25 defines $l(i)$ and $l(j)$. Note that equations 26, 27, 28 and 29 differs from the ones presented in ? by the absence of the module.

In both x - or y -orientation, matrix \mathbf{A} (equation 12) can be rewritten by the indices $q = -Q + 1, \dots, 0, \dots, Q - 1$ defining the number of its blocks:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^0 & \mathbf{A}^{-1} & \dots & \mathbf{A}^{-Q+1} \\ \mathbf{A}^1 & \mathbf{A}^0 & \mathbf{A}^{-1} & \vdots \\ \vdots & \mathbf{A}^1 & \ddots & \mathbf{A}^{-1} \\ \mathbf{A}^{Q-1} & \dots & \mathbf{A}^1 & \mathbf{A}^0 \end{bmatrix}_{N \times N}, \quad (30)$$

and by indice p , where each block has $P \times P$ elements a_p^q , $p = -P + 1, \dots, 0, \dots, P - 1$:

$$\mathbf{A}^q = \begin{bmatrix} a_0^q & a_{-1}^q & \dots & a_{-P+1}^q \\ a_1^q & a_0^q & a_{-1}^q & \vdots \\ \vdots & a_1^q & \ddots & a_{-1}^q \\ a_{P-1}^q & \dots & a_1^q & a_0^q \end{bmatrix}_{P \times P}, \quad (31)$$

In general, matrix \mathbf{A} (equation 12) is a non-symmetric BTTB, i.e., its blocks are non-symmetric ($\mathbf{A}^{-Q+1} \neq \mathbf{A}^{Q-1}$) and its elements also are non-symmetric ($a_{-1}^q \neq a_1^q$). Depending on specific values of the main field direction and the equivalent sources magnetization

directions, matrix \mathbf{A} can assume other structures, for example, when $\hat{\mathbf{F}} = [0, 0, 1]$ and $\hat{\mathbf{h}} = [0, 0, 1]$ it becomes symmetric. In this work, we are considering the more common situation for the matrix \mathbf{A} .

Also differently for the symmetric sensitivity matrix described by ?, the non-symmetric BTTB matrix cannot be reconstructed only by its first column. The construction of the matrix \mathbf{A} (equation 12) needs four columns: the first and last columns of the first column of blocks and the first and last columns of the last column of blocks. This has a physical implication in the equivalent layer which is not possible to use only one equivalent source to reproduce all elements of matrix \mathbf{A} , such as in the gravity case as demonstrated by ?. Rather, in the magnetic case it takes four equivalent sources positioned at each corner of the equivalent layer. Figure 2 shows the positioning of the equivalent sources in a regular grid $N_x = 4$ and $N_y = 3$ necessary to calculate the four columns capable of recovering the matrix \mathbf{A} .

In this work, we propose a different approach, by calculating the first column of all six different components of second derivatives matrices from \mathbf{H}_{ij} (equation 10). These matrices are, in fact, symmetric or skew-symmetric BTTBs, meaning that the first column has all elements of each matrix.

By substituting equations 2, 9 and 10 into equation 12, it is possible to describe each element of the sensitivity matrix by the second derivative components of \mathbf{H}_{ij} :

$$\begin{aligned} a_{ij} = c_m \frac{\mu_0}{4\pi} & (F_x H_{xx} + F_y H_{xy} + F_z H_{xz}) h_x + \\ & (F_x H_{xy} + F_y H_{yy} + F_z H_{yz}) h_y + \\ & (F_x H_{xz} + F_y H_{yz} + F_z H_{zz}) h_z . \end{aligned} \quad (32)$$

In equation 32, $c_m \frac{\mu_0}{4\pi}$ is a constant. If we consider that the components of the vectors $\hat{\mathbf{F}}$

(equation 2) and $\hat{\mathbf{h}}$ (equation 9) are constants multiplying the second derivatives \mathbf{H} (equation 10); then, the sensitivity matrix \mathbf{A} (equation 12) is purely the sum of the components $H_{xx} + H_{xy} + H_{xz} + H_{yx} + H_{yy} + H_{yz} + H_{xz} + H_{yz} + H_{zz}$ multiplied by the respective constants of each component. Thus, despite \mathbf{A} not being a symmetric BTTB matrix, it can be in fact, written by calculating only the first column of these components. In the next few sections we will describe each component \mathbf{H} as its own matrix.

Structure of matrices components \mathbf{H}_{xx} , \mathbf{H}_{yy} and \mathbf{H}_{zz}

We can describe the elements of \mathbf{H}_{xx} , \mathbf{H}_{yy} and \mathbf{H}_{zz} by substituting equations 18 and 19 in equation 10 as:

$$h_{ij}^{xx} = \frac{-1}{[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2]^{\frac{3}{2}}} + \frac{3(\Delta k_{ij} \Delta x)^2}{[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2]^{\frac{5}{2}}}, \quad (33)$$

$$h_{ij}^{yy} = \frac{-1}{[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2]^{\frac{3}{2}}} + \frac{3(\Delta l_{ij} \Delta y)^2}{[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2]^{\frac{5}{2}}} \quad (34)$$

and

$$h_{ij}^{zz} = \frac{-1}{[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2]^{\frac{3}{2}}} + \frac{3(\Delta z)^2}{[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2]^{\frac{5}{2}}}, \quad (35)$$

where $\Delta z = z_j - z_i$, $\Delta k_{ij} = k(i) - k(j)$ (equations 22 or 24) and $\Delta l_{ij} = l(i) - l(j)$ (equations 21 or 25). The principal components \mathbf{H}_{xx} , \mathbf{H}_{yy} and \mathbf{H}_{zz} (equations 33, 34 and 35, respectively) of matrix \mathbf{H} (equation 10) are symmetric-Block-Toeplitz symmetric-Toeplitz-Block matrices. This means that \mathbf{H}_{xx} , \mathbf{H}_{yy} and \mathbf{H}_{zz} are Toeplitz and symmetric by its blocks and each of the blocks are symmetric Toeplitz matrices. For example, \mathbf{H}_{xx} can be described by the *block index* q that represents the block diagonals of this matrix as a grid of $Q \times Q$

blocks $\mathbf{H}_{\mathbf{xx}}^q$, $q = 0, \dots, Q - 1$:

$$\mathbf{H}_{\mathbf{xx}} = \begin{bmatrix} \mathbf{H}_{\mathbf{xx}}^0 & \mathbf{H}_{\mathbf{xx}}^1 & \dots & \mathbf{H}_{\mathbf{xx}}^{Q-1} \\ \mathbf{H}_{\mathbf{xx}}^1 & \mathbf{H}_{\mathbf{xx}}^0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{H}_{\mathbf{xx}}^1 \\ \mathbf{H}_{\mathbf{xx}}^{Q-1} & \dots & \mathbf{H}_{\mathbf{xx}}^1 & \mathbf{H}_{\mathbf{xx}}^0 \end{bmatrix}_{N \times N}. \quad (36)$$

And each diagonal of the blocks are represented by $P \times P$ elements h_p^q , $p = 0, \dots, P - 1$:

$$\mathbf{H}_{\mathbf{xx}}^q = \{h_p^q\} = \begin{bmatrix} h_0^q & h_1^q & \dots & h_{P-1}^q \\ h_1^q & h_0^q & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_1^q \\ h_{P-1}^q & \dots & h_1^q & h_0^q \end{bmatrix}_{P \times P}. \quad (37)$$

In a *x-oriented grid* $Q = N_y$, $P = N_x$ and q and p can be defined by the functions:

$$q(i, j) = |l(i) - l(j)| \quad (38)$$

and

$$p(i, j) = |k(i) - k(j)|, \quad (39)$$

where $l(i)$ and $l(j)$ are defined by equation 21 and $k(i)$ and $k(j)$ are defined by equation 22. For *y-oriented grids*, $Q = N_x$, $P = N_y$ and the block indices q and p are defined, respectively, by the following integer functions of the matrix indices i and j :

$$q(i, j) = |k(i) - k(j)| \quad (40)$$

and

$$p(i, j) = |l(i) - l(j)|, \quad (41)$$

This struture can also describe matrices $\mathbf{H}_{\mathbf{yy}}$ and $\mathbf{H}_{\mathbf{zz}}$ in the same manner and they are identical to the structure of the gravity sensitiviy matrix from ?.

Structure of the components matrices $\mathbf{H}_{\mathbf{xy}}$

By substituting equations 18 and 19 in equation 10, we can also describe the elements of $\mathbf{H}_{\mathbf{xy}}$, as:

$$h_{ij}^{xy} = \frac{3(\Delta k_{ij} \Delta x)(\Delta l_{ij} \Delta y)}{\left[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2\right]^{\frac{5}{2}}}, \quad (42)$$

The component $\mathbf{H}_{\mathbf{xy}}$ (equation 42) of matrix \mathbf{H} (equation 10) are skew symmetric-Block-Toeplitz skew symmetric-Toeplitz-Block matrices. This means that $\mathbf{H}_{\mathbf{xy}}$ is Toeplitz and skew symmetric by its blocks and each of the blocks are skew symmetric Toeplitz matrices. This way, matrix $\mathbf{H}_{\mathbf{xy}}$ can be described by the *block index* q that represents the block diagonals of this matrix as a grid of $Q \times Q$ blocks $\mathbf{H}_{\mathbf{xy}}^q$, $q = -Q + 1, \dots, 0, \dots, Q - 1$:

$$\mathbf{H}_{\mathbf{xy}} = \begin{bmatrix} \mathbf{H}_{\mathbf{xy}}^0 & \mathbf{H}_{\mathbf{xy}}^{-1} & \cdots & \mathbf{H}_{\mathbf{xy}}^{-Q+1} \\ \mathbf{H}_{\mathbf{xy}}^1 & \mathbf{H}_{\mathbf{xy}}^0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{H}_{\mathbf{xy}}^{-1} \\ \mathbf{H}_{\mathbf{xy}}^{Q-1} & \cdots & \mathbf{H}_{\mathbf{xy}}^1 & \mathbf{H}_{\mathbf{xy}}^0 \end{bmatrix}_{N \times N}. \quad (43)$$

And each diagonal of the blocks are represented by $P \times P$ elements h_p^q , $p = -P + 1, \dots, 0, \dots, P - 1$:

$$\mathbf{H}_{\mathbf{xy}}^q = \{h_p^q\} = \begin{bmatrix} h_0^q & h_{-1}^q & \cdots & h_{-P+1}^q \\ h_1^q & h_0^q & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_{-1}^q \\ h_{P-1}^q & \cdots & h_1^q & h_0^q \end{bmatrix}_{P \times P}. \quad (44)$$

In a *x-oriented grid* $Q = N_y$, $P = N_x$ and q and p can be defined by the functions:

$$q(i, j) = l(i) - l(j) \quad (45)$$

and

$$p(i, j) = k(i) - k(j), \quad (46)$$

where $l(i)$ and $l(j)$ are defined by equation 21 and $k(i)$ and $k(j)$ are defined by equation 22. For y -oriented grids, $Q = N_x$, $P = N_y$ and the block indices q and p are defined, respectively, by the following integer functions of the matrix indices i and j :

$$q(i, j) = k(i) - k(j) \quad (47)$$

and

$$p(i, j) = l(i) - l(j) , \quad (48)$$

Important to clarify that in this case, as a skew symmetric matrix, the values of oposing diagonals have oposing signals, e.g., $\mathbf{H}_{\mathbf{xy}}^{-1} = -\mathbf{H}_{\mathbf{xy}}^1$ and $h_{-1}^q = -h_1^q$.

Structure of the components matrices \mathbf{H}_{xz}

By substituting equations 18 and 19 in equation 10, the elements of \mathbf{H}_{xz} , are given by:

$$h_{ij}^{xz} = \frac{3(\Delta k_{ij} \Delta x)(\Delta z)}{\left[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2\right]^{\frac{5}{2}}} , \quad (49)$$

The component \mathbf{H}_{xz} (equation 49) of matrix \mathbf{H} (equation 10) are skew symmetric-Block-Toeplitz symmetric-Toeplitz-Block matrices. This means that \mathbf{H}_{xz} is Toeplitz and skew symmetric by its blocks and each of the blocks are symmetric Toeplitz matrices. Thus, matrix \mathbf{H}_{xz} can be described by the *block index* q that represents the block diagonals of this matrix as a grid of $Q \times Q$ blocks \mathbf{H}_{xz}^q , $q = -Q + 1, \dots, 0, \dots, Q - 1$:

$$\mathbf{H}_{xz} = \begin{bmatrix} \mathbf{H}_{xz}^0 & \mathbf{H}_{xz}^{-1} & \cdots & \mathbf{H}_{xz}^{-Q+1} \\ \mathbf{H}_{xz}^1 & \mathbf{H}_{xz}^0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{H}_{xz}^{-1} \\ \mathbf{H}_{xz}^{Q-1} & \cdots & \mathbf{H}_{xz}^1 & \mathbf{H}_{xz}^0 \end{bmatrix}_{N \times N} . \quad (50)$$

And each diagonal of the blocks are represented by $P \times P$ elements h_p^q , $p = 0, \dots, P - 1$:

$$\mathbf{H}_{xz}^q = \{h_p^q\} = \begin{bmatrix} h_0^q & h_1^q & \cdots & h_{P-1}^q \\ h_1^q & h_0^q & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_1^q \\ h_{P-1}^q & \cdots & h_1^q & h_0^q \end{bmatrix}_{P \times P}. \quad (51)$$

In a *x-oriented grid* $Q = N_y$, $P = N_x$ and q and p can be defined by the functions:

$$q(i, j) = l(i) - l(j) \quad (52)$$

and

$$p(i, j) = |k(i) - k(j)|, \quad (53)$$

where $l(i)$ and $l(j)$ are defined by equation 21 and $k(i)$ and $k(j)$ are defined by equation 22. For *y-oriented grids*, $Q = N_x$, $P = N_y$ and the block indices q and p are defined, respectively, by the following integer functions of the matrix indices i and j :

$$q(i, j) = k(i) - k(j) \quad (54)$$

and

$$p(i, j) = |l(i) - l(j)|, \quad (55)$$

In this case as a skew symmetric matrix by blocks, the values of oposing diagonals blocks have oposing signals, e.g., $\mathbf{H}_{xz}^{-1} = -\mathbf{H}_{xz}^1$ but each block is a symmetric matrix.

Structure of the components matrices \mathbf{H}_{yz}

Finally, by substituting equations 18 and 19 in equation 10, we can also describe the elements of \mathbf{H}_{yz} , as:

$$h_{ij}^{yz} = \frac{3(\Delta l_{ij} \Delta y)(\Delta z)}{\left[(\Delta k_{ij} \Delta x)^2 + (\Delta l_{ij} \Delta y)^2 + (\Delta z)^2\right]^{\frac{5}{2}}}, \quad (56)$$

The component \mathbf{H}_{yz} (equation 56) of matrix \mathbf{H} (equation 10) are symmetric-Block-Toeplitz skew symmetric-Toeplitz-Block matrices. This means that \mathbf{H}_{yz} is Toeplitz and symmetric by its blocks and each of the blocks are skew symmetric Toeplitz matrices. Thus, matrix \mathbf{H}_{yz} can be described by the *block index* q that represents the block diagonals of this matrix as a grid of $Q \times Q$ blocks \mathbf{H}_{yz}^q , $q = 0, \dots, Q - 1$:

$$\mathbf{H}_{yz} = \begin{bmatrix} \mathbf{H}_{yz}^0 & \mathbf{H}_{yz}^1 & \cdots & \mathbf{H}_{yz}^{Q-1} \\ \mathbf{H}_{yz}^1 & \mathbf{H}_{yz}^0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{H}_{yz}^1 \\ \mathbf{H}_{yz}^{Q-1} & \cdots & \mathbf{H}_{yz}^1 & \mathbf{H}_{yz}^0 \end{bmatrix}_{N \times N}. \quad (57)$$

And each diagonal of the blocks are represented by $P \times P$ elements h_p^q , $p = -P + 1, \dots, 0, \dots, P - 1$:

$$\mathbf{H}_{yz}^q = \{h_p^q\} = \begin{bmatrix} h_0^q & h_{-1}^q & \cdots & h_{-P+1}^q \\ h_1^q & h_0^q & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_{-1}^q \\ h_{P-1}^q & \cdots & h_1^q & h_0^q \end{bmatrix}_{P \times P}. \quad (58)$$

In a *x-oriented grid* $Q = N_y$, $P = N_x$ and q and p can be defined by the functions:

$$q(i, j) = |l(i) - l(j)| \quad (59)$$

and

$$p(i, j) = k(i) - k(j), \quad (60)$$

where $l(i)$ and $l(j)$ are defined by equation 21 and $k(i)$ and $k(j)$ are defined by equation 22. For *y-oriented grids*, $Q = N_x$, $P = N_y$ and the block indices q and p are defined, respectively, by the following integer functions of the matrix indices i and j :

$$q(i, j) = |k(i) - k(j)| \quad (61)$$

and

$$p(i, j) = l(i) - l(j) , \quad (62)$$

Being a symmetric matrix by blocks, the values of \mathbf{H}_{yz} from oposing diagonals blocks are equal, but each block have skew symmetric oposing diagonals, i.e., $h_{-1}^q = -h_1^q$.

CGLS matrix-vector substitution

As pointed earlier in this work, the main improvement inside the CGLS method (Algorithm 1) for estimating the parameter vector $\hat{\mathbf{p}}$ (equation 15) is to substitute the matrix-vector multiplication $\mathbf{A}^\top \mathbf{s}^{(0)}$ out of the loop and the two matrix-vector multiplications inside the loop at steps 3 an 6, $\mathbf{A} \mathbf{c}^{(it)}$ and $\mathbf{A}^\top \mathbf{s}^{(it+1)}$. The latter two matrix-vector multiplications are accomplished at each iteration, thus taking most of CGLS method runtime.

Our method consists in calculating the six first columns of the second derivatives of \mathbf{H} (equation 10) and embedding them into the first six columns of the block-circulant circulant-block (BCCB) matrices related to the \mathbf{H} components. Thus, it is possible to calculate the first column of the BCCB matrix embedded from matrix \mathbf{A} (equation 12) by multiplying each component with its respective constants and summing as shown in equation 32. In ?, Appendix A, the authors demonstrated in details how to transform a symmetric BTB matrix into a BCCB matrix \mathbf{C} . The process here is the same and that work can be referenced to achieve the same results.

A new auxilary linear system is constructed to carry the matrix-vector product:

$$\mathbf{w} = \mathbf{Cv} , \quad (63)$$

where

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_0 \\ \vdots \\ \mathbf{w}_{Q-1} \\ \mathbf{0}_{2N \times 1} \end{bmatrix}_{4N \times 1}, \quad (64)$$

$$\mathbf{w}_q = \begin{bmatrix} \mathbf{d}_q(\mathbf{p}) \\ \mathbf{0}_{P \times 1} \end{bmatrix}_{2P \times 1}, \quad (65)$$

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_0 \\ \vdots \\ \mathbf{v}_{Q-1} \\ \mathbf{0}_{2N \times 1} \end{bmatrix}_{4N \times 1}, \quad (66)$$

and

$$\mathbf{v}_q = \begin{bmatrix} \mathbf{p}_q \\ \mathbf{0}_{P \times 1} \end{bmatrix}_{2P \times 1}, \quad (67)$$

where \mathbf{C} (equation 63) is a $4N \times 4N$ non-symmetric (BCCB) resulted from transforming \mathbf{A} (equation 12). Without having to calculate the whole BCCB matrix, its first column can be used to carry the multiplication of this new system (equation 63). Appendix A and C in ? shows how to use the 2D-FFT to compute the eigenvalues of matrix \mathbf{C} , store in a $2Q \times 2P$ matrix using the *vec*-operator and to carry the matrix-vector product. Denoting the matrix \mathbf{L} as the eigenvalues matrix follows:

$$\mathbf{F}_{2Q}^* [\mathbf{L} \circ (\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P})] \mathbf{F}_{2P}^* = \mathbf{W}, \quad (68)$$

where the symbol “ \circ ” references the Hadamard product, i.e., a element-wise complex multiplication between the eingenvectors and the 2D-FFT of the matrix rearranged along the rows of the parameters \mathbf{V} (equation 66) using the same *vec*-operator. The resulting inverse

2D-FFT denoted by $\mathbf{F}_{2Q}^* \otimes \mathbf{F}_{2P}^*$ is also a $2Q \times 2P$ matrix (\mathbf{W}) that can be rearranged to the predicted data vector $\mathbf{d}(\hat{\mathbf{p}})$ size N .

Computational performance

To compare the efficiency of our algorithm we will use a numerical approach and calculate the floating-point operations (*flops*), i.e., count the number of mathematical operations necessary to complete the estimative of parameter vector $\hat{\mathbf{p}}$ of the normal equations (equation 15) and both the CGLS methods (algorithm 1) for calculating the matrix-vector product by its standart way and our approach.

The *flops* needed to solve the linear system in equation 15 using the Cholesky factorization is:

$$f_{classical} = \frac{7}{3}N^3 + 6N^2, \quad (69)$$

where N is the total number of observation points and also the size of estimated parameter vector $\hat{\mathbf{p}}$.

For the more efficient CGLS algorithm the estimative can be done in:

$$f_{cglsl} = 2N^2 + it(4N^2 + 12N). \quad (70)$$

However, our approach reduces further to:

$$f_{ours} = \kappa 16N \log_2(4N) + 24N + it(\kappa 16N \log_2(4N) + 60N), \quad (71)$$

where κ depends on the FFT algorithm. By default, in this work we will use $\kappa = 5$ for the *radix-2* algorithm (?).

Figure 3 shows a comparative between the methods varying the number of observation points up to 1,000,000, where it is possible to observe a reduction of 10^7 orders of magnitude

to estimate parameter vector $\hat{\mathbf{p}}$ in relation to the non-iterative classical method and 10^3 orders of magnitude in relation to the standart CGLS algorithm using 50 iterations. A more detailed, step by step, flops count of the classical and CGLS algorithm can be found in Appendix A.

In Figure 4 we show the time necessary to construct matrix \mathbf{A} (equation 12) and solve the linear system up to 10,000 points of observation. With this dataset the classical method takes more than sixty-three seconds, the 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.

In Figure 5 a comparison between the time to complete the task to calculate the first column of the BCCB matrix embedded from the from \mathbf{A} (equation 12) by using only one equivalent source, i.e., calculating all six first column of the second derivatives matrices from \mathbf{H} (equation 10) and using four equivalent sources to calculate the four necessary columns from the non-symmetric matrix \mathbf{A} (equation 12). Although, very similar in time, with one source a small advantage can be observed as the number of data N increases and goes beyond $N = 200,000$. This test was done from $N = 10,000$ to $N = 700,000$ with increases of 5,625 observation points.

In Table 1 there is comparison between how much RAM memory is adressed to store the sensitivity matrix for each of the methods. The classical approach and the CGLS have to store the whole matrix \mathbf{A} (equation 12), this means that a dataset with for example $N = 10,000$ observation points, the sensitivity matrix has $N^2 = 100,000,000$ elements and takes approximately 763 Megabytes of memory (8 bytes per element). For our method, it is necessary to store the first six columns of each of the components from matrix \mathbf{H} (equation 10) embedded into the BCCB matrices. With the same dataset $N = 10,000$

it needs 1.831 Megabytes. After completing the steps to store the eigenvalues of matrix \mathbf{C} (equation 63) it takes only 0.6104 Megabytes. Here, we are considering 16 bytes per element as the eigenvalues are complex numbers resulting from the 2D FFT. For a bigger dataset as $N = 1,000,000$ the amount of RAM necessary goes to 7,629,395, 183.096 and 61.035 Megabytes, respectively, showing the necessity to find improved and efficient methods for the equivalent layer technique as the one presented in this work. We remember that throughout our work we are always considering $N = M$.

APPLICATION TO SYNTHETIC DATA

The convolutional equivalent layer requires that the data be measure on a regular grid and the observation surface be planar. Here, we evaluate the performance of the proposed method by applying it to simulate airborne magnetic surveys that i) perfectly obey these conditions to the flight pattern; ii) violate the requirement of a regular grid of the observations; and iii) violate the requirement of a flat observation surface. These simulated airborne magnetic surveys are illustrated in Figure 6 and Figures 7 and 8 show the corresponding results.

Figure 6a shows a regular grid of 100×50 observation points in the x - and y -directions, totaling $N = 5,000$ observation points. The noise-corrupted total-field anomaly (Figure 6b) is calculated at 900 m height. The data were produced by three geologic bodies modeled by: two prisms and a sphere with inclination, declination and intensity of 0° and 45° and 2.8284 A/m, respectively. The main geomagnetic field has inclination and declination of 10° and 37° , respectively.

Using the classical method (equation 15) a predicted data (not shown) was estimated and the data residuals are shown in Figure 7a, with mean of 0.3712 nT and standart deviation of 0.2798 nT. This process took 17.10 seconds.

From now on, we use the phrase "data residuals" to define the difference between the observed (second column in Figure 6) and the predicted data (not shown) obtained by the classical method or by the convolutional equivalent layer (our method).

Using our method, the running time to estimate the data residuals (Figure 7b), with mean of 0.5150 nT and standart deviation of 0.4363 nT, was 0.18 seconds. The data residuals have mean of 0.5150 nT and standart deviation of 0.4363 nT

Figure 7c shows the convergence of our method. The Euclidean norm of the data residuals decreases as expected when the convergence criterion was satisfied, close to iteration 50.

This result shows that, in practice, it is not necessary to run the conjugate gradient least square method at N iterations to get an exactly solution. Actually, the exactly solution would never occur due to roundoff errors. Hence, by setting the convergence to N iterations besides being unnecessary it also demands large computer processing time, even in this synthetic test with a small layer ($N = 5,000$ equivalent sources).

Tests with data on irregular grids

As shown in the methodology, a regular grid of observation points is needed to arise the BTTB matrix. In this section, we show the results when our method is applied directly to irregular grids of $N = 5\,000$ observation points. First, we start from the regular grid of 100×50 observation points, shown in Figure 6a, with a grid spacing of Δx of 101.01 m along the x -axis and Δy of 163.265 m along the y -axis. Next, the x - and y -coordinates of the observations were also contaminated with additive pseudorandom Gaussian noise with zero mean and standard deviations of 20% and 30% of the Δx and Δy spacing.

Figure 6d shows an irregular grid with uncertainty of 20% along both x - and y -directions of the observation points. Hence, the x - and y -coordinates of the observations shown in the regular grid (Figure 6a) were corrupted with sequences of pseudorandom Gaussian noise having zero means and standard deviations of 20.2 m and 32.65 m, respectively.

The noise-corrupted total-field anomaly (Figure 6e) is calculated on this irregular grid and on a flat observation surface at 900 m height.

Figure 7d shows that the data residuals using the classical approach show a good data fit with mean of 0.3630 nT and standart deviation of 0.2731 nT.

Using our method, the data residuals (Figure 7e) 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 (Figure 7f) decreases as expected and close to iteration 50 converges to a constant value.

Figure 6g shows an irregular grid with uncertainty of 30% along both x - and y -directions of the observation points. Hence, Gaussian pseudorandom noise sequences with zero means and standard deviations of 30.3 m and 48.98 m were added, respectively, to the x - and y -coordinates of the observations shown in the regular grid (Figure 6a), producing the simulated irregular grid shown in Figure 6g.

Figure 6h shows the noise-corrupted total-field anomaly calculated on this irregular grid and on a flat observation surface at 900 m height.

Figure 7g shows that the data residuals obtained by the classical approach produced an acceptable data fitting, having mean of 0.3634 nT and standart deviation of 0.2735 nT.

Using our method, the data residuals (Figure 7h) with mean of 0.9788 nT and standart deviation of 0.7462 nT produced a poor data fitting.

Figure 7i shows the convergence analysis of our method. Similarly to the previous results, in the begining of the iterations, the Euclidean norm of the data residuals obtained by our method decreases; however it starts increasing without achieving an invariance. Under this condition, the convergence is not achieved.

Tests with data over an undulating observation surface.

Here, we simulate an airborne magnetic survey considering the same regular grid of 100×50 observation points shown in Figure 6a. However, the observation points were no longer in a plane at 900 m height, but they are over an undulating observation surface. The next tests, the z -coordinate of the observations were contaminated with pseudorandom Gaussian noise mean of -900 m and standard deviations of 5% and 10% of the 900 m height.

Figure 6j shows an uneven surface of observations where the z -coordinates of the observations were corrupted with additive pseudorandom Gaussian noise having mean of -900 m and a standard deviation of 45 m.

Figure 6k shows the noise-corrupted total-field anomaly calculated on a regular grid of 100×50 observation points in the x - and y -coordinates (Figure 6a) and over the undulating observation surface shown in Figure 6j.

The data residuals either using classical approach (Figure 7j) or using our method (Figure 7k) reveal acceptable data fitting. Using the classical approach, the data residuals have mean of 0.3865 nT and standart deviation of 0.3216 nT. Using our method, the data residuals have mean of 1.6105 nT and standart deviation of 1.6231 nT.

Likewise, Figure 7l shows that the Euclidean norm of the data residuals, which were obtained by using our method, decreases up to the iteration 50 and reaches an invariance in the subsequent iterations.

Figure 6m shows an uneven surface of observations where the z -coordinates of the observations were corrupted with additive pseudorandom Gaussian noise having mean of -900 m and a standard deviation of 90 m.

The noise-corrupted total-field anomaly calculated on a regular grid of 100×50 observation points in the x - and y -coordinates (Figure 6a) and over the undulating observation surface (Figure 6m) is shown in Figure 6n.

Figure 7m shows, using the classical approach, the data residuals yielded an good data fitting, with mean of 0.4155 nT and standart deviation of 0.4005 nT. By using our method, the data residuals Figure 7n yielded a poor data fitting with mean of 6.622 nT and standart deviation of 5.901 nT. The convergence analysis (Figure7o) reveals the inadequacy of our method in dealing with rugged surface of observations, as the Euclidean norm of the data residuals decreases slower than previous tests

Although our method is formulated to deal with magnetic observations measured on a regular grid, in the x - and y -directions, and on a planar surface, the synthetic results show that our method is robust in dealing either with irregular grids in the horizontal directions or with uneven surface. However, the robustness of our method has limitations. The performance limitation of our method depends on the degree of the departure of the x - and y -coordinates of the data from there corresponding coordinates on a regular grid and from the amplitude of the undulating observation surface. High departures of the x - and y -coordinates from a regular grid and large variations in the data elevation (z -coordinates of the data) are associated with unacceptable data fittings (large data residuals) as shown in Figures 7h and 7n, respectively. However, the poor performance of our method in cases of irregular grid and uneven observation surface can be detected easily because, besides it leads to poor data fitting, it does not show an expected convergence as shown in Figures 7i and 7o.

Magnetic data processing

We performed the upward continuations of the noise-corrupted total-field anomalies (second column in Figure 6) by using the classical method (equation 15), the convolutional equivalent layer (our method), and the classical approach in the Fourier domain. The noise-free total-field anomalies produced by the synthetic sources at $z = 1,00$ m (third column in Figure 6) are called the true upward-continued total-field anomalies at $z = 1,300$ m.

Figure 8 shows the data residuals of the upward-continued total-field anomalies obtained by the classical method (first column), our method (second column) and the classical approach in the Fourier domain (third column). The upward continuation by using the classical approach in Fourier domain consists in computing the Fourier transform of the total-field anomaly (? , p. 317).

From now on, we use the phrase "data residuals of the upward-continued total-field anomalies" to define the difference between the true upward-continued total-field anomalies (third column in Figure 6) and the predicted upward-continued total-field anomalies (not shown).

Our results (Figure 8) show that the data residuals of the upward-continued total-field anomalies obtained by using the classical method (first column) and our method (second column) are, in most of the tests, similar to each other. One exception is the synthetic test with data over an undulating observation surface with uncertainty of 10% shown in Figures 6m-o and 7m-o. We can note that the absolute value of the data residuals of the upward-continued total-field anomalies produced by using our method (Figure 8n) are ≈ 2.5 times greater than those produced by the classical method (Figure 8m). However, we stress that the simulated undulating observation surface in this test (Figure 6m) 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.

In contrast, the data residuals of the upward-continued total-field anomalies obtained by using the the classic Fourier approach (third column in Figure 8) are, in most of the tests, approximately 6 times greater than those produced by the classical method (first column in Figure 8) and 4 times greater than those produced by our method (second column in Figure 8). We can note that the maximum and minimum values of the the data residuals of the upward-continued total-field anomalies obtained by using the the classic Fourier approach are located at the boundaries of the simulated area.

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 function. The data residuals (first and second columns in Figure 7) and the data residuals of the upward-continued total-field anomalies (Figure 8) are fully shown without removing the edge effects at the borders of the simulated area. As pointed out in the methodology, the computational time required by our method is much lower than the one required by the classical method. However, the computational time required by the classical Fourier approach is the lowest one. Although, the classical Fourier approach is faster than our method, the upward-continued data exhibit strong border effects if no one padding function to expand the data was applied.

APPLICATION TO FIELD DATA

The field data application was performed with the aeromagnetic data of Carajás, Pará, Brazil, provided by CPRM. The survey is composed of 131 flight lines N-S oriented, spacing $\Delta y = 3,000$ m. The magnetometer (Scintrex CS-3) was set to a interval between measurements of 0.1 s giving a spacing $\Delta x = 7.65$ m. The average flight heighth is $\Delta z = -900$ m. The total number of observation points is $N = 6,081,345$. Figure 9 shows the observed magnetic field data of the area.

For the actual data processing, we have made a comparison between an interpolated regular grid of $10,000 \times 131$ using a nearest neighbour algorithm and a decimated irregular grid, also of $10,000 \times 131$, totaling $N = 1,310,000$ observation points in both cases. The decimated grid was performed by using the regular grid created in the first case as a guide and by finding the nearest real observation point to this regular grid thus, ensuring that the irregular grid is the lesser deviant possible to conduct the BTB scheme. In figure 10a we show the result of the interpolation and in 10b the result of the 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 approach. However, taking advantage that the second derivatives of equation 6 are symmetric or skew-symmetric matrices, it is possible to reconstruct the whole sensitivity matrix storing only the first column of each component of equation 10, thus, using only 59.97 Megabytes, allowing desktop computers being able to process this amount of data.

As this area is very large, different values of the magnetic main field can be considered. For this processing, it was considered an approximated mid location of the area (latitude -6.5° and longitude -50.75°) where the declination is -19.86° for the IGRF model in

1st january, 2014. The inclination was calculated considering the Geocentric axial dipole model ($\tan I = 2 \times \tan \lambda$) and is equal to 12.84° . As the source magnetization is unknown, inclination and declination equals to the main field is being used.

To achieve high efficiency in property estimative of the equivalent sources, the method CGLS for inversion was used, combined with a fast matrix-vector product, only possible because of the BTTB structure of the sensitivity matrix. This fast matrix-vector product was also used for data processing (upward-continuation) in a very efficient way.

Using a equivalent layer at 300 meters above the ground the predicted data and its residual of the interpolated regular grid are shown in figure 11. The mean of 0.07979 nT and the standart deviation of 0.5060 nT of the residual shows the good result of physical property estimative. It was used 200 iterations of the CGLS method taking 390.80 seconds with a Intel core i7 7700HQ@2.8GHz processor in single-processing and single-threading modes. Using the same equivalent layer and CGLS configuration the predicted data and its residual of the decimated irregular grid are shown in figure 12. With a mean of 0.07348 nT, standart deviation of 0.3172 nT and lower residual amplitude compared to the regular grid, we show that the process of decimating the original data, without creating new observation points with interpolation, can be benefical to the method, even if an irregular grid is taken place. It took 385.56 seconds to complete the estimative.

The convergence analysis for the decimated irregular grid up to 2,000 CGLS iterations is in figure 13, showing good convergence rate and guaranteeing that the irregular grid is not disturbing the method.

In figure 14 the upward-continuation transformation using the estimated equivalent layer with the decimated grid was made in a horizontal plane at 5,000 meters and took

2.64 seconds, showing good results without visible errors or border effects problems and accentuating the long wavelenghts.

CONCLUSIONS

In this work, we were able to develop a fast equivalent layer technique for processing magnetic data with the method of Conjugate Gradient Least Square using the convolutional equivalent layer theory to obtain results of performance more than four orders of magnitude less than the classical equivalent layer. The sensitivity matrix of the magnetic equivalent layer carries the structure of BTTB matrices, which means a very low computational cost matrix-vector product and also the possibility to store only the first column of the matrix BCCB. In this work we propose a novel method to use only one equivalent source and calculating the first six columns of the inverse of distance second derivatives matrices to arrive in the first column of the BCCB matrix embedded from the original magnetic kernel sensitivity matrix.

Synthetic tests showed similar results estimating the physical property using a classical approach to solve a linear system and our method using the CGLS combined with the BTTB matrix-vector product. 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 80×80 points, greater results can be obtained if more observation points are used.

Real data test were also conducted in the region of Carajás, Pará, Brazil. With an irregular grid of 1,310,000 observation points, store the full sensitivity matrix it would be necessary 12.49 Terabytes of RAM. However, taking advantage of the symmetric or skew-symmetric matrices structures, it is possible to reconstruct the whole sensitivity matrix using only 59.97 Megabytes. Using 200 iterations of the CGLS method took 385.56 seconds and very good results of property estimative were obtained. Also the upward-continuation

transformation showed good results and took only 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 308945/2017-4) and FAPERJ (grant E-26/202.729/2018). The authors thank the Geological Survey of Brazil (CPRM) for providing the field data.

APPENDIX A

FLOPS COMPUTATIONS

Classical flops count

The flops count of the classical approach to solve the linear system (equation 15) using the Cholesky factorization is given by equation 69. The step-by-step count follows:

- (1) $\mathbf{A}^\top \mathbf{A}$: $2N^3$ (one matrix-matrix product).
- (2) $\mathbf{A}^\top \mathbf{A}$: $\frac{1}{3}N^3$ (one Cholesky factorization \mathbf{C}_f).
- (3) $\mathbf{A}^\top \mathbf{d}^o$: $2N^2$ (one matrix-vector product).
- (4) $\mathbf{C}_f(\mathbf{A}^\top \mathbf{d}^o)$: $2N^2$ (one matrix-vector product).
- (5) $\mathbf{C}_f^\top (\mathbf{C}_f \mathbf{A}^\top \mathbf{d}^o)$: $2N^2$ (one matrix-vector product).

Summing all calculations:

$$f_{classical} = \frac{7}{3}N^3 + 6N^2, \quad (A-1)$$

CGLS flops count

The flops count of CGLS algorithm 1 can be summarized as:

Out of the loop:

- (1) $\mathbf{A}^\top \mathbf{s}$: $2N^2$ (one matrix-vector product).

Inside the loop:

- (1) $\frac{\mathbf{r}^{(it)\top} \mathbf{r}^{(it)}}{\mathbf{r}^{(it-1)\top} \mathbf{r}^{(it-1)}}$: $4N$ (two vector-vector products).
- (2) $\mathbf{r}^{it} - \alpha_{it} \beta_{it} \mathbf{c}^{(it-1)}$: $2N$ (one scalar-vector product and one vector subtraction).
- (3) $\frac{\|\mathbf{r}^{(it)}\|_2^2}{(\mathbf{c}^{(it)\top} \mathbf{A}^\top)(\mathbf{A} \mathbf{c}^{(it)})}$: $2N^2 + 2N$ (one matrix-vector and one vector-vector product).
- (4) $\hat{\mathbf{p}}^{it} - \alpha_{it} \mathbf{c}^{(it)}$: $2N$ (one vector subtraction).
- (5) $\mathbf{s}^{it} - \alpha_{it} \mathbf{A} \mathbf{c}^{(it)}$: $2N$ (one vector subtraction, the matrix-vector product was calculated in step 3).
- (6) $\mathbf{A}^\top \mathbf{s}^{(it+1)}$: $2N^2$ (one matrix-vector product).

The result of all flops count leads to:

$$f_{cgls} = 2N^2 + it(4N^2 + 12N). \quad (\text{A-2})$$

Our modified CGLS flops count

All the flops count presented in previous section for the CGLS remains, only substituting the out of the loop matrix-vector product in step 1 and the two matrix-vector products inside the loop in steps 3 and 6. The computations necessary to carry the matrix-vector used in this work are given by:

- (1) \mathbf{L} : $\kappa 4N \log_2(4N)$ (one 2D FFT for the eigenvalues calculation of the sensitivity matrix \mathbf{A} or the transposed sensitivity matrix \mathbf{A}^\top).
- (2) $\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P}$: $\kappa 4N \log_2(4N)$ (one 2D FFT).
- (3) $\mathbf{L} \circ (\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P})$: $24N$ (one complex Hadamard matrix multiplication).

(4) $\mathbf{F}_{2Q}^* [\mathbf{L} \circ (\mathbf{F}_{2Q} \mathbf{V} \mathbf{F}_{2P})] \mathbf{F}_{2P}^*$: $\kappa 4N \log_2(4N)$ (one inverse 2D FFT).

Matrix-vector product total: $\kappa 12N \log_2(4N) + 24N$.

As matrix \mathbf{A} (equation 12) and its transposed never changes, it is not necessary to calculate the eigenvalues inside the loop at each iteration, we are considering that both are calculated out of the loop. Inside the loop, steps 2 to 4 are repeated two times per iteration. Substituting this result into the CGLS flops count (equation A-2) leads to:

$$f_{ours} = \kappa 16N \log_2(4N) + 24N + it(\kappa 16N \log_2(4N) + 60N). \quad (\text{A-3})$$

LIST OF TABLES

1 This table shows the RAM memory usage (in Megabytes) for storing the whole matrix **A** (equation 12), the sum of all six first columns of the BCCB matrices embedded from the components of the matrix **H** from equation 10 (both need 8 bytes per element) and the matrix **L** containing the eigenvalues complex numbers (16 bytes per element) resulting from the diagonalization of matrix **C** (equation 63). Here we must consider that N observation points forms a $N \times N$ matrix.

LIST OF FIGURES

1 Schematic representation of an $N_x \times N_y$ regular grid of points (black dots) defined by $N_x = 4$ and $N_y = 3$. The grids are oriented along the (a) x -axis and (b) y -axis. The grid coordinates are x_k and y_l , where the $k = 1, \dots, N_x$ and $l = 1, \dots, N_y$ are called the grid indices. The insets show the grid indices k and l .

2 Representation of the four equivalent sources (black dots) needed to reconstruct the non-symmetric matrix \mathbf{A} (equation 12). Each of the equivalent sources are located in the corner of the simulated regular grid of $M_x = 4$ and $M_y = 3$ sources. The influence of these sources on each of the observation points (blue dots) in the regular grid of $N_x = 4$ and $N_y = 3$ will give the four columns necessary of matrix \mathbf{A} .

3 Number of flops necessary to estimate the parameter vector $\hat{\mathbf{p}}$ using the non-iterative classical method (equation 69) the CGLS (equation 70) and our modified CGLS method (equation 71) with $N^{it} = 50$. The observation point N varied from 5,000 to 1,000,000. The *radix-2* 2D FFT algorithm was considered for our method, with $\kappa = 5$.

4 Comparison between the runtime of the equivalent-layer technique using the classical, the CGLS algorithm and our method. The values for the CGLS and our methods were obtained for $N^{it} = 50$ iterations.

5 Comparison between the runtime to calculate the first column of the BCCB matrix embedded from \mathbf{A} (equation 12) using only one and using four equivalent sources. Although the time is very similar, with one source a small advantage can be observed as the number of data N increases. This test was done from $N = 10,000$ to $N = 700,000$ with increases of 5,625 observation points.

6 Synthetic tests: the simulated airborne magnetic surveys - The first column shows the grids of observation points or the undulating observation surfaces that simulate air-

borne magnetic surveys. The second column shows the noise-corrupted total-field anomaly produced by the synthetic sources and calculated on the simulate airborne magnetic survey shown in the first column. The third column shows the noise-free total-field anomaly produced by the synthetic sources at $z = 1,300$ m. The results shown in these three columns were obtained by using the simulated airborne magnetic surveys as follows: (a)-(c) A regular grid of 100×50 observation points in the $x-$ and $y-$ directions and a flat observation surface at 900 m height. (d)-(f) A irregular grid with uncertainty of 20% in the $x-$ and $y-$ coordinates and a flat observation surface at 900 m height. (g)-(i) A irregular grid with uncertainty of 30% in the $x-$ and $y-$ coordinates and a flat observation surface at 900 m height. (j)-(l) A regular grid of 100×50 observation points in the $x-$ and $y-$ coordinates and an undulating observation surface with uncertainty of 5%. (m)-(o) A regular grid of 100×50 observation points in the $x-$ and $y-$ coordinates and an undulating observation surface with uncertainty of 10%. The black lines represent the horizontal projection of the sources .

7 Synthetic tests: the data residuals and convergence - The first column shows the data residuals using the classical method. The second and third columns show, respectively, the data residuals and the convergence curves using the convolutional equivalent layer (our method). The results shown in these three columns were obtained by using the simulated airborne magnetic surveys shown in Figure 6, i.e.: (a)-(c) A regular grid of 100×50 observation points in the $x-$ and $y-$ directions and a flat observation surface at 900 m height. (d)-(f) A irregular grid with uncertainty of 20% in the $x-$ and $y-$ coordinates and a flat observation surface at 900 m height. (g)-(i) A irregular grid with uncertainty of 30% in the $x-$ and $y-$ coordinates and a flat observation surface at 900 m height. (j)-(l) A regular grid of 100×50 observation points in the $x-$ and $y-$ coordinates and an undulating observation

surface with uncertainty of 5%. (m)-(o) A regular grid of 100×50 observation points in the $x-$ and $y-$ coordinates and an undulating observation surface with uncertainty of 10%. The black lines represent the horizontal projection of the sources .

8 Synthetic tests: the data residuals of the upward-continued total-field anomalies (second column of the Figure 6). The data residuals of the upward-continued total-field anomalies are defined as the difference between the noise-free total-field anomaly produced by the synthetic sources at $z = 1,300$ m (third column of the Figure 6) and the predicted total-field anomaly at $z = 1,300$ m obtained by using three methods: the classical method (first column); the convolutional equivalent layer (second column); and the classic approach in the Fourier domain (third column). The results shown in these three columns were obtained by using the simulated airborne magnetic surveys shown in Figure 6, i.e.: (a)-(c) A regular grid of 100×50 observation points in the $x-$ and $y-$ directions and a flat observation surface at 900 m height. (d)-(f) A irregular grid with uncertainty of 20% in the $x-$ and $y-$ coordinates and a flat observation surface at 900 m height. (g)-(i) A irregular grid with uncertainty of 30% in the $x-$ and $y-$ coordinates and a flat observation surface at 900 m height. (j)-(l) A regular grid of 100×50 observation points in the $x-$ and $y-$ coordinates and an undulating observation surface with uncertainty of 5%. (m)-(o) A regular grid of 100×50 observation points in the $x-$ and $y-$ coordinates and an undulating observation surface with uncertainty of 10%. The black lines represent the horizontal projection of the sources .

9 Observed magnetic field data of the Carajás, Brazil area. The aeromagnetic survey was done with 131 N-S lines at approximately $-900m$ height, totaling $N = 6,081,345$ observation points.

10 (a) Observed magnetic field data of the Carajás, Brazil area, interpolated for a

regular grid of $10,000 \times 131$, totaling $N = 1,310,000$ observation points. (b) Observed magnetic field data of the Carajás, Brazil area, decimated from the flight lines resulting in an irregular grid of $10,000 \times 131$, also totaling $N = 1,310,000$ observation points.

11 (a) Predicted data using our method for the interpolated $10,000 \times 131$ regular grid.
(b) Residuals between the observed (10) and the predicted data (panel a), with a mean of 0.07979 nT and standart deviation of 0.5060 nT.

12 (a) Predicted data using our method for the decimated $10,000 \times 131$ irregular grid.
(b) Residuals between the observed (10) and the predicted data (panel b), with a mean of 0.07348 nT and standart deviation of 0.3172 nT.

13 Convergence analysis of the CGLS method for the field data of Carajás, Brazil using the magnetic equivalent layer with a decimated irregular grid of $10,000 \times 131$ obser-
vation points up to 2,000 iterations.

14 Upward continuation transformation of real data of Carajás, Brazil at 5,000 meter.
It was necessary 2.64 seconds to complete the process.

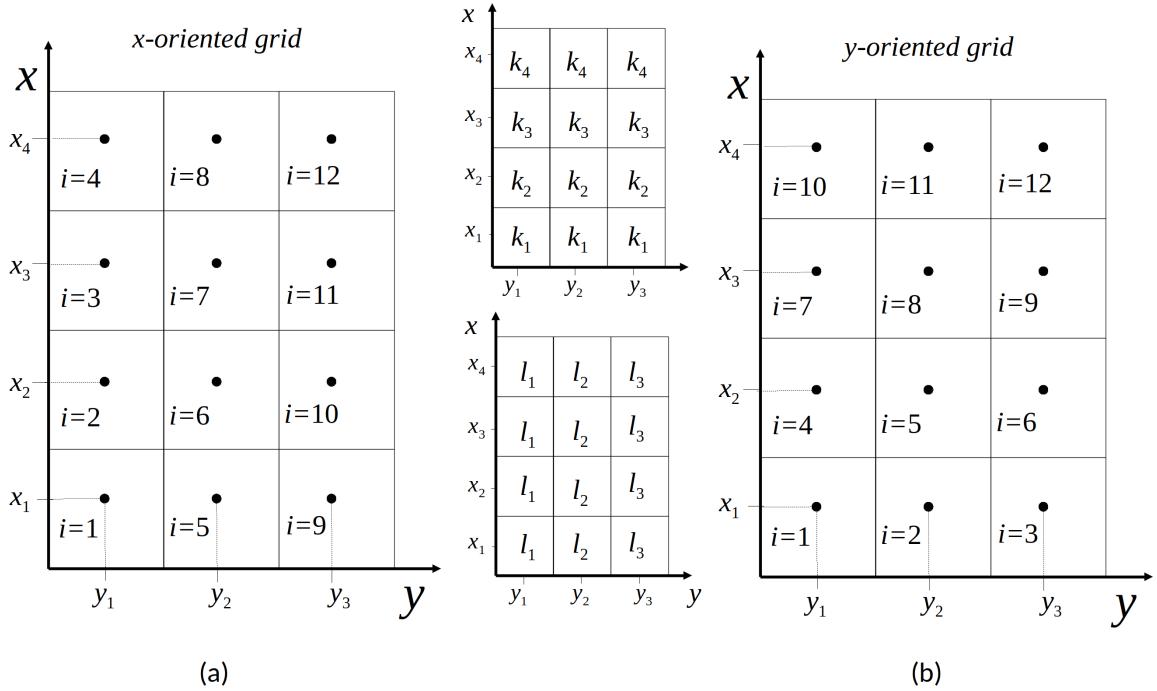


Figure 1: Schematic representation of an $N_x \times N_y$ regular grid of points (black dots) defined by $N_x = 4$ and $N_y = 3$. The grids are oriented along the (a) x -axis and (b) y -axis. The grid coordinates are x_k and y_l , where the $k = 1, \dots, N_x$ and $l = 1, \dots, N_y$ are called the grid indices. The insets show the grid indices k and l .

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

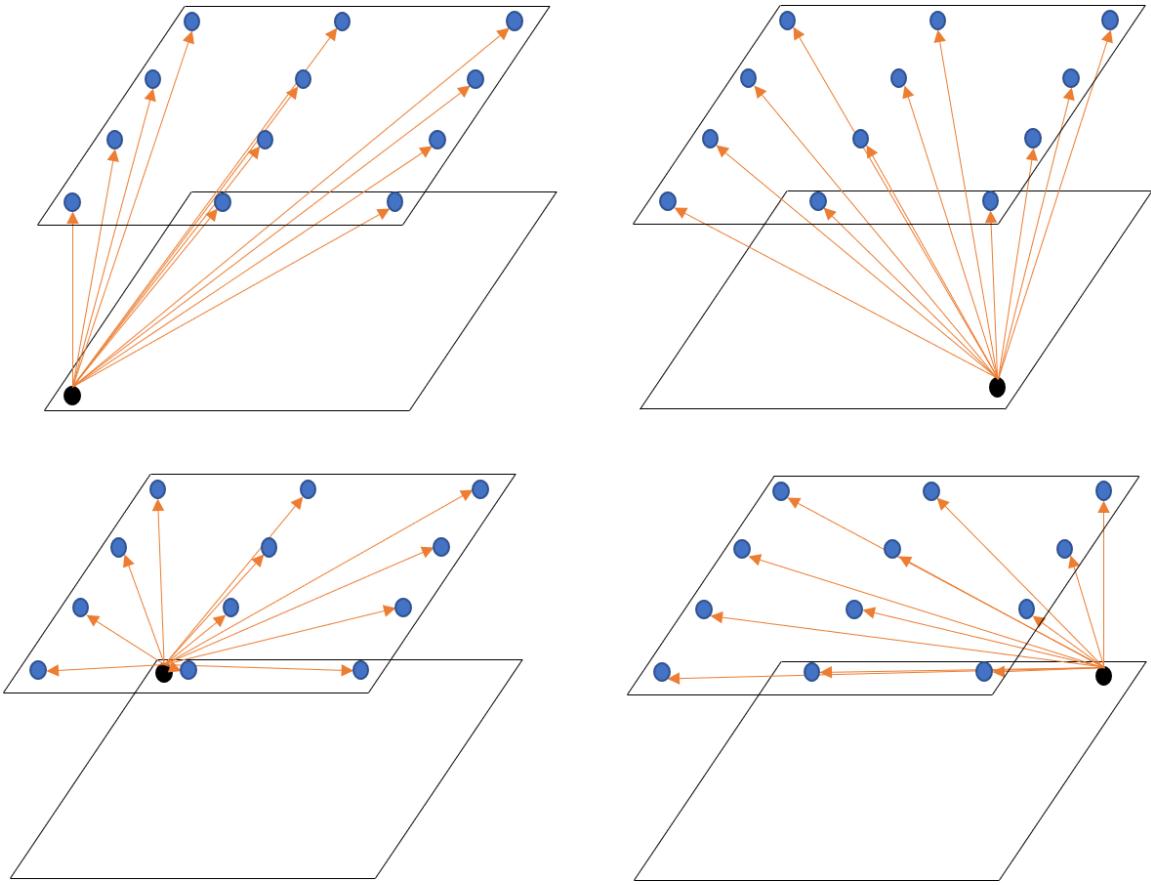


Figure 2: Representation of the four equivalent sources (black dots) needed to reconstruct the non-symmetric matrix \mathbf{A} (equation 12). Each of the equivalent sources are located in the corner of the simulated regular grid of $M_x = 4$ and $M_y = 3$ sources. The influence of these sources on each of the observation points (blue dots) in the regular grid of $N_x = 4$ and $N_y = 3$ will give the four columns necessary of matrix \mathbf{A} .

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

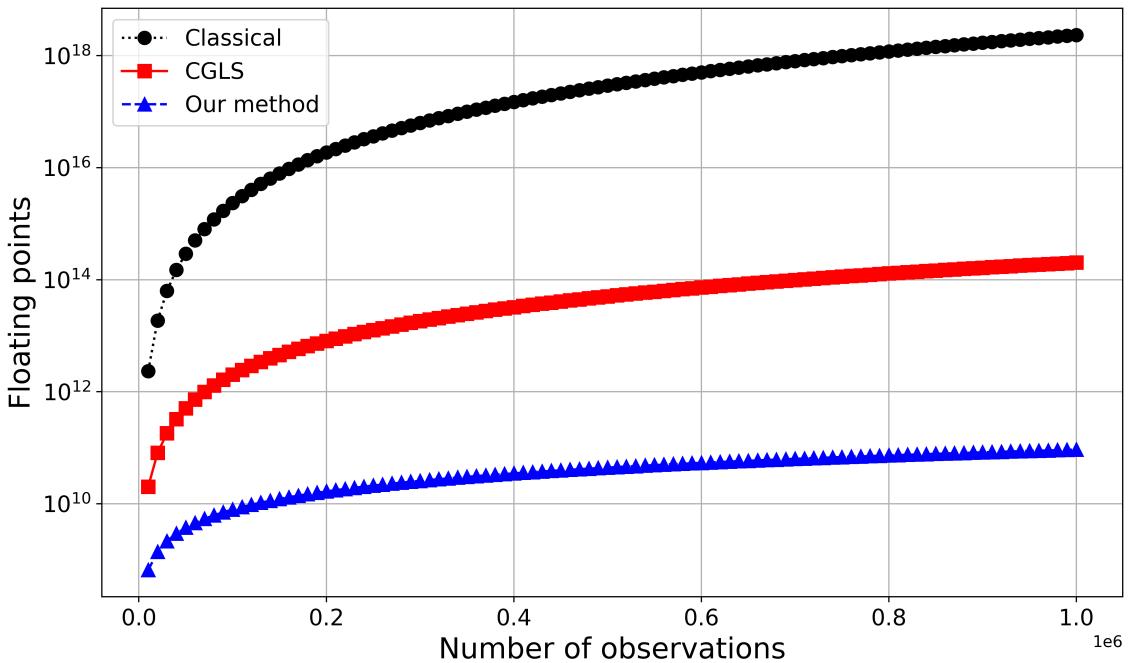


Figure 3: Number of flops necessary to estimate the parameter vector $\hat{\mathbf{p}}$ using the non-iterative classical method (equation 69) the CGLS (equation 70) and our modified CGLS method (equation 71) with $N^{it} = 50$. The observation point N varied from 5,000 to 1,000,000. The radix-2 2D FFT algorithm was considered for our method, with $\kappa = 5$.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

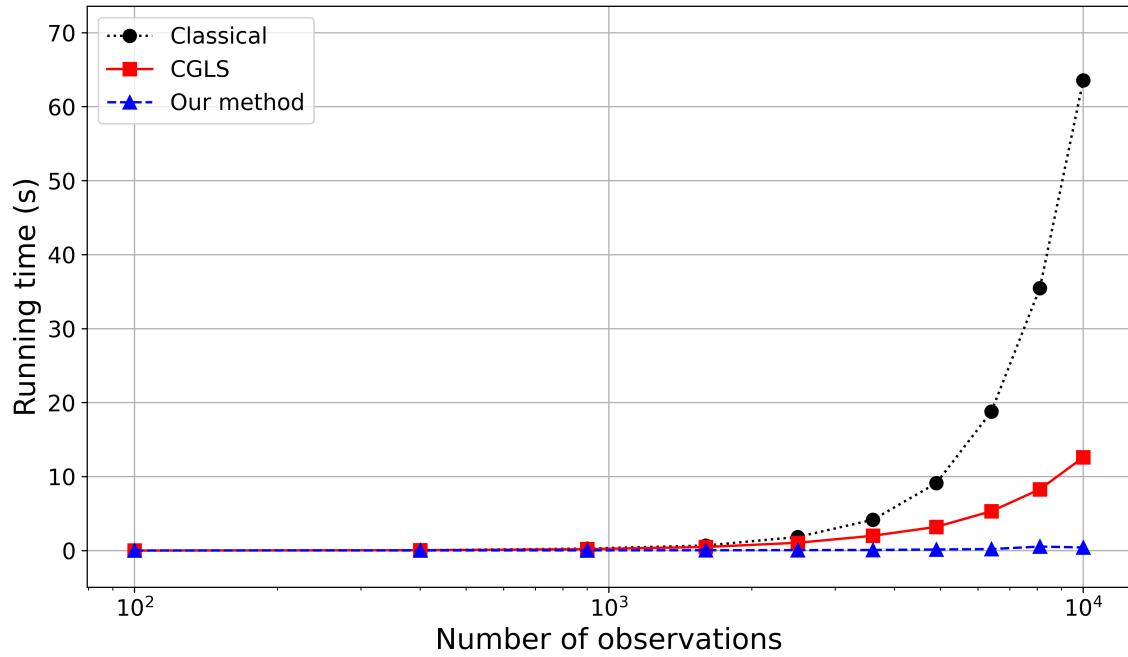


Figure 4: Comparison between the runtime of the equivalent-layer technique using the classical, the CGLS algorithm and our method. The values for the CGLS and our methods were obtained for $N^{it} = 50$ iterations.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

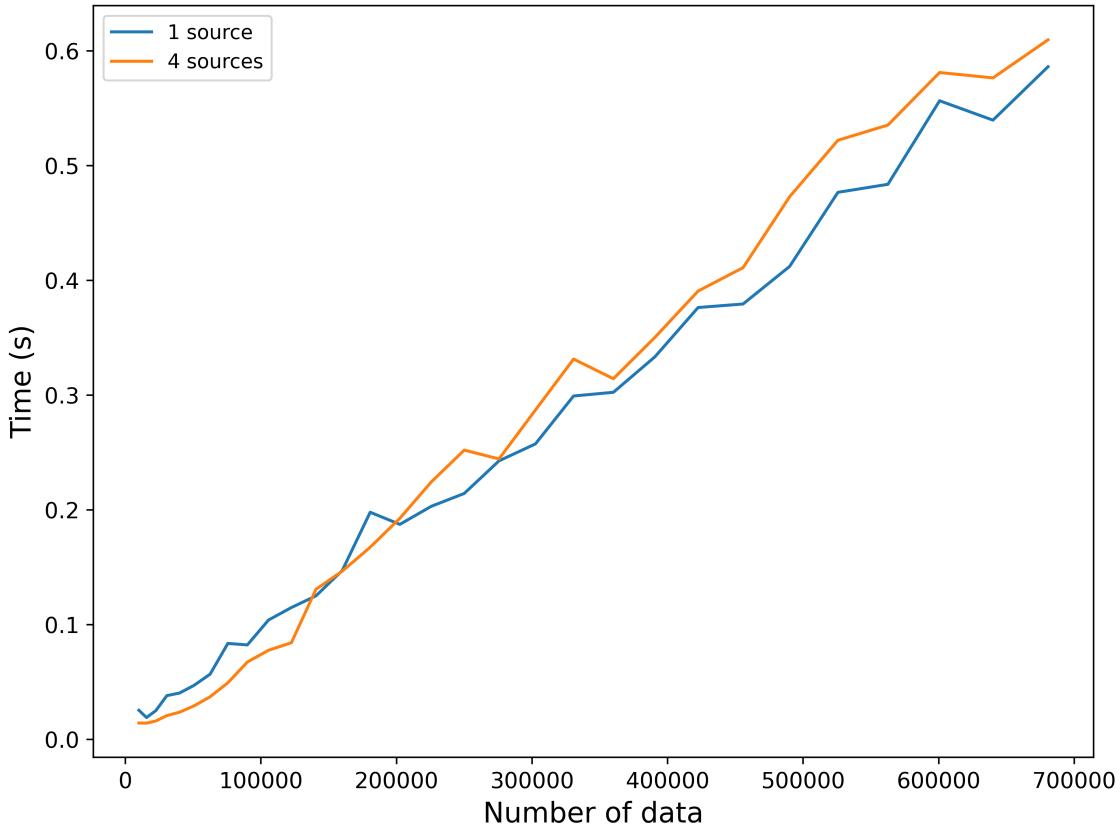


Figure 5: Comparison between the runtime to calculate the first column of the BCCB matrix embedded from \mathbf{A} (equation 12) using only one and using four equivalent sources. Although the time is very similar, with one source a small advantage can be observed as the number of data N increases. This test was done from $N = 10,000$ to $N = 700,000$ with increases of 5,625 observation points.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

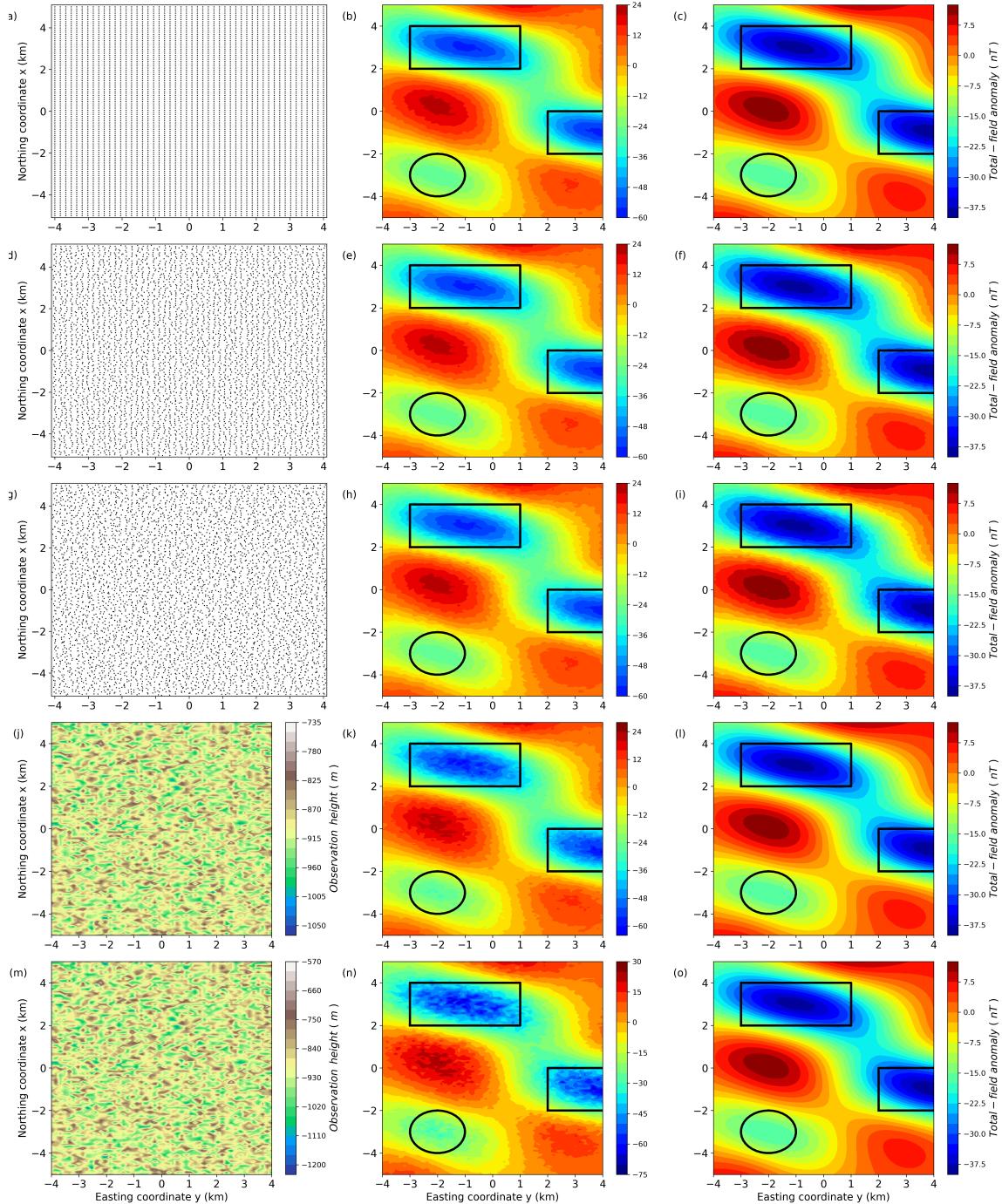


Figure 6: Synthetic tests: the simulated airborne magnetic surveys - The first column shows the grids of observation points or the undulating observation surfaces that simulate airborne magnetic surveys. The second column shows the noise-corrupted total-field anomaly produced by the synthetic sources and calculated on the simulate airborne magnetic survey shown in the first column. The third column shows the noise-free total-field anomaly pro-

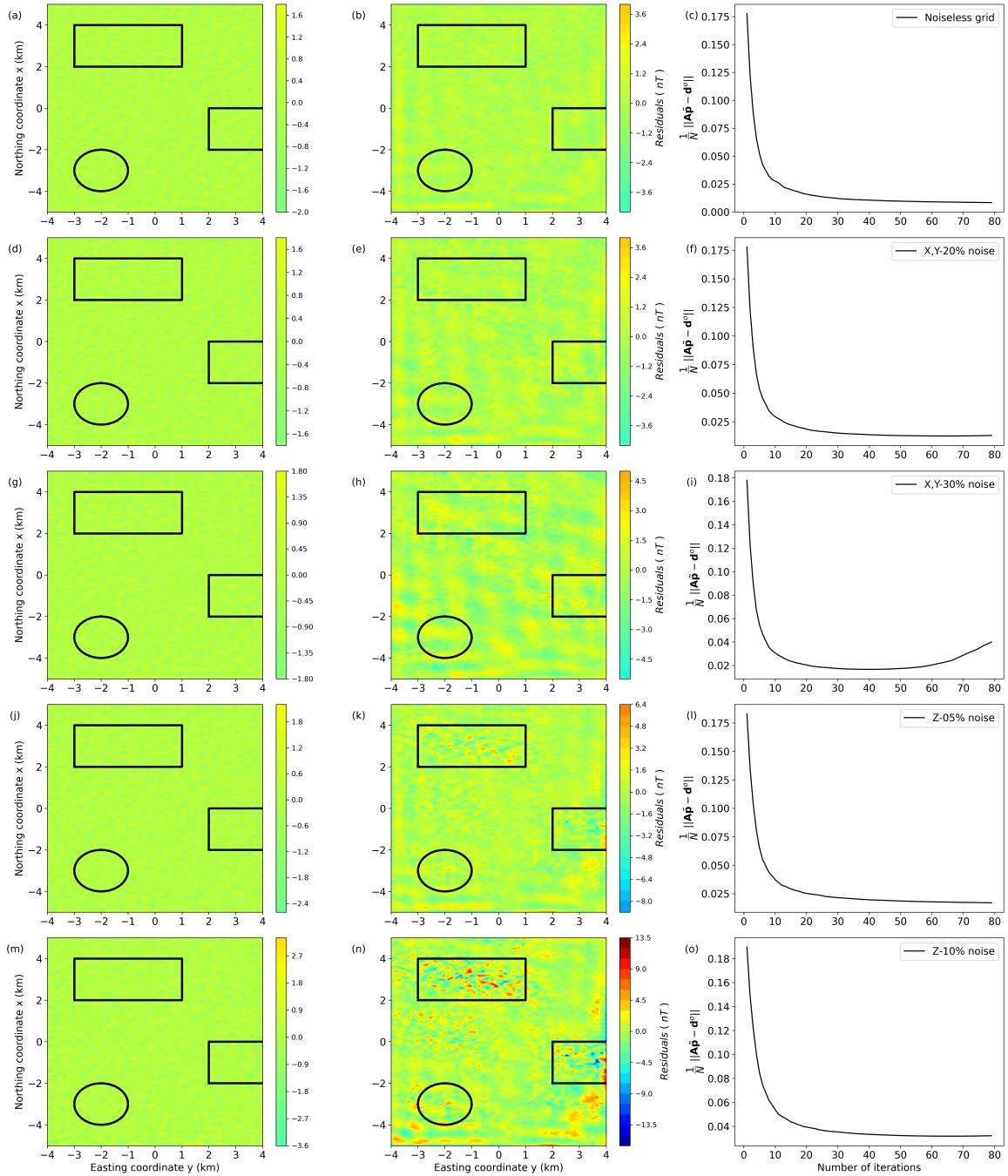


Figure 7: Synthetic tests: the data residuals and convergence - The first column shows the data residuals using the classical method. The second and third columns show, respectively, the data residuals and the convergence curves using the convolutional equivalent layer (our method). The results shown in these three columns were obtained by using the simulated airborne magnetic surveys shown in Figure 6, i.e.: (a)-(c) A regular grid of 100×50 obser-

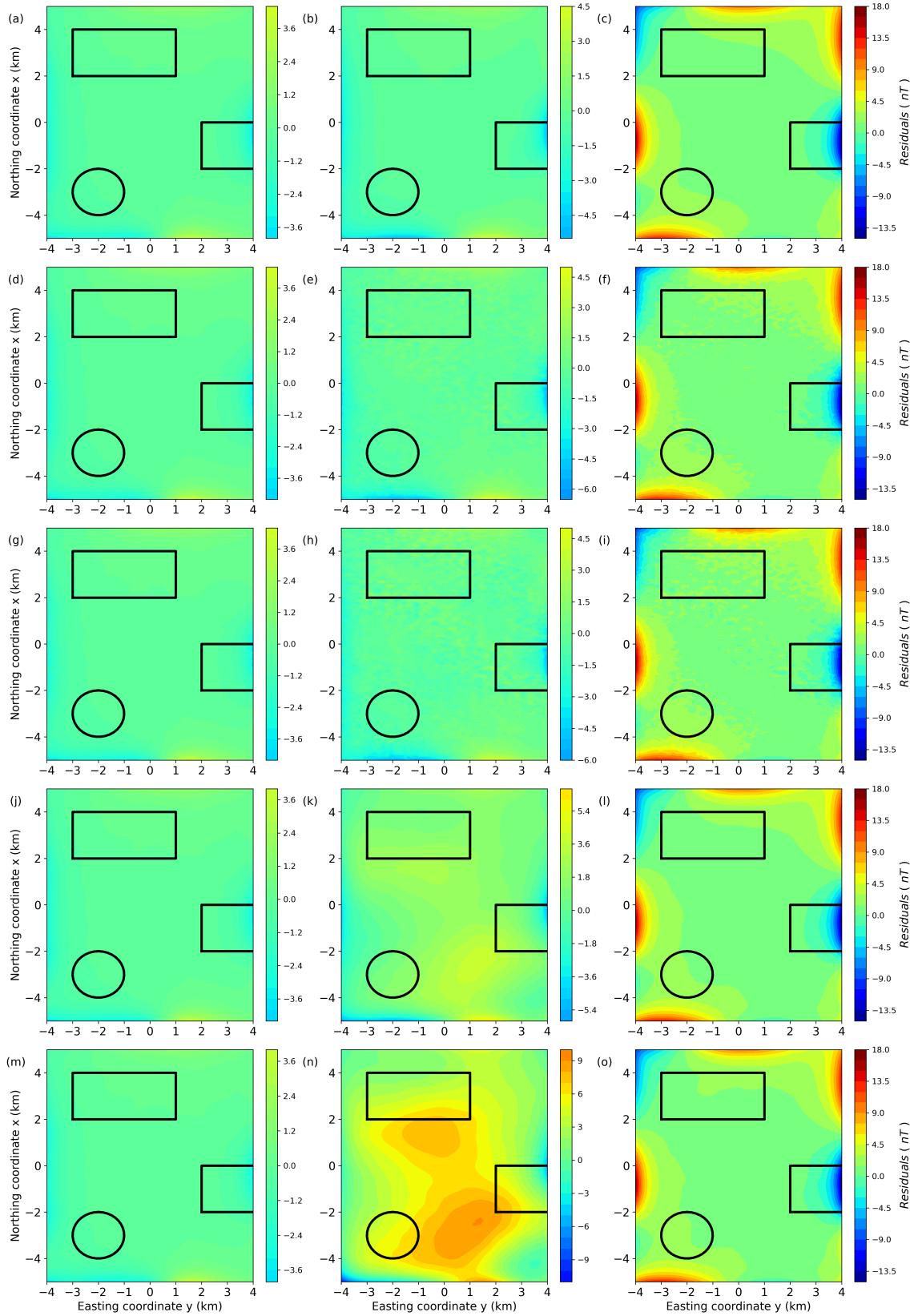


Figure 8: Synthetic tests: the data residuals of the upward-continued total-field anomalies

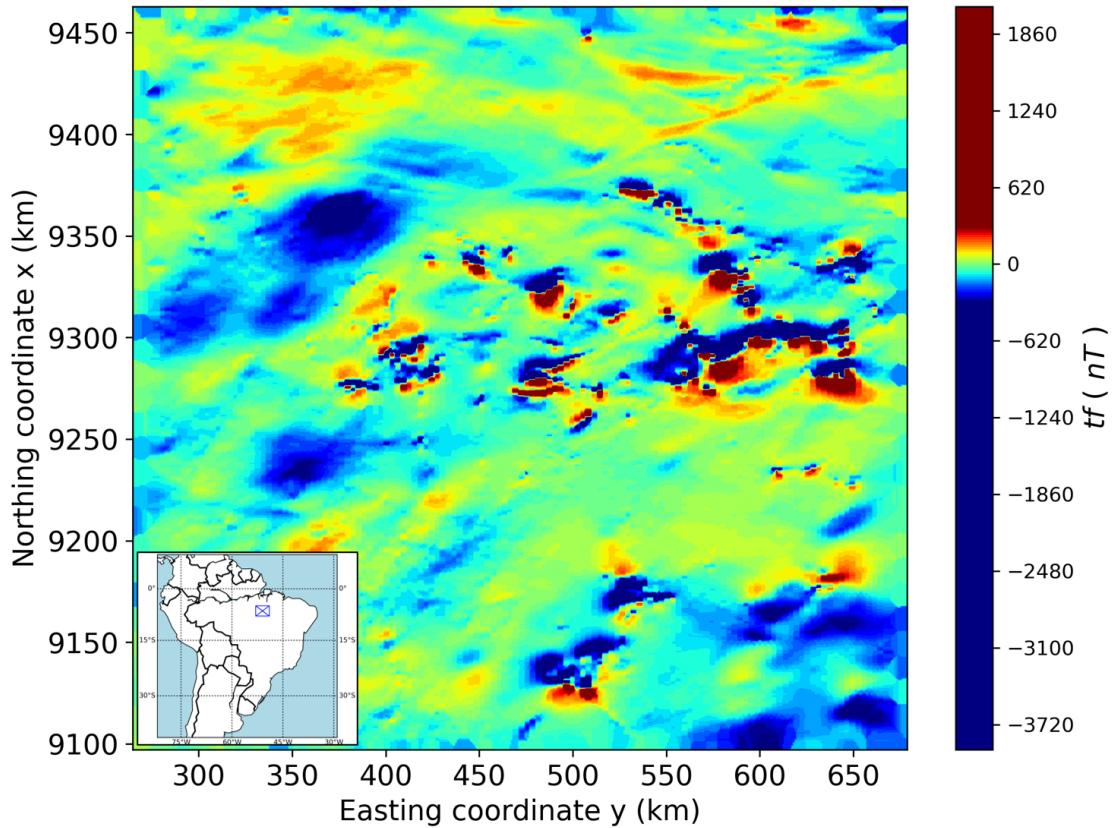


Figure 9: Observed magnetic field data of the Carajás, Brazil area. The aeromagnetic survey was done with 131 N-S lines at approximately $-900m$ height, totaling $N = 6,081,345$ observation points.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

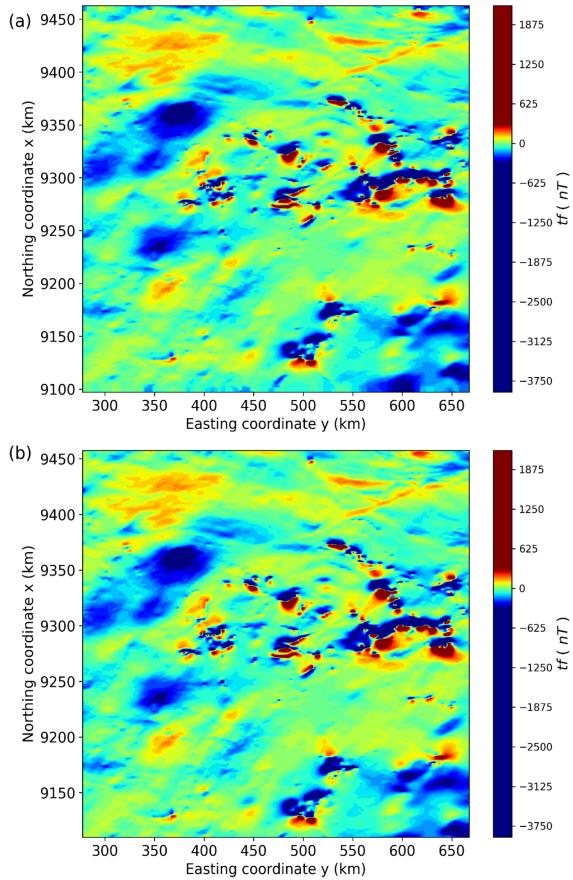


Figure 10: (a) Observed magnetic field data of the Carajás, Brazil area, interpolated for a regular grid of $10,000 \times 131$, totaling $N = 1,310,000$ observation points. (b) Observed magnetic field data of the Carajás, Brazil area, decimated from the flight lines resulting in an irregular grid of $10,000 \times 131$, also totaling $N = 1,310,000$ observation points.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

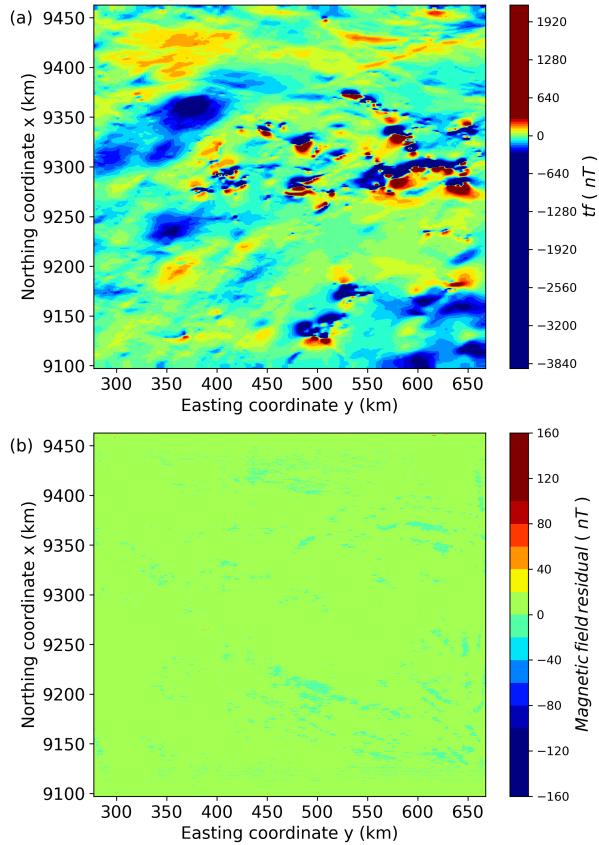


Figure 11: (a) Predicted data using our method for the interpolated $10,000 \times 131$ regular grid. (b) Residuals between the observed (10) and the predicted data (panel a), with a mean of 0.07979 nT and standart deviation of 0.5060 nT.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

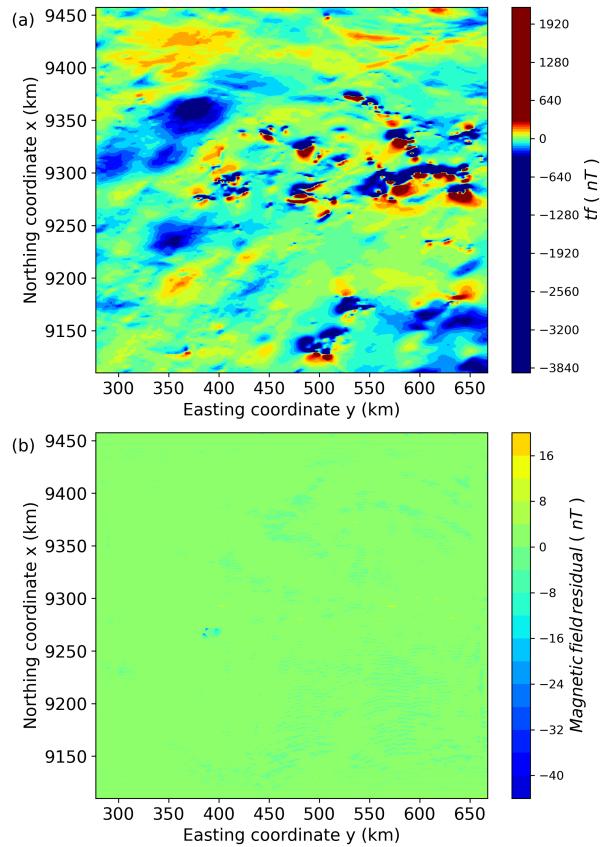


Figure 12: (a) Predicted data using our method for the decimated $10,000 \times 131$ irregular grid. (b) Residuals between the observed (10) and the predicted data (panel b), with a mean of 0.07348 nT and standart deviation of 0.3172 nT.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

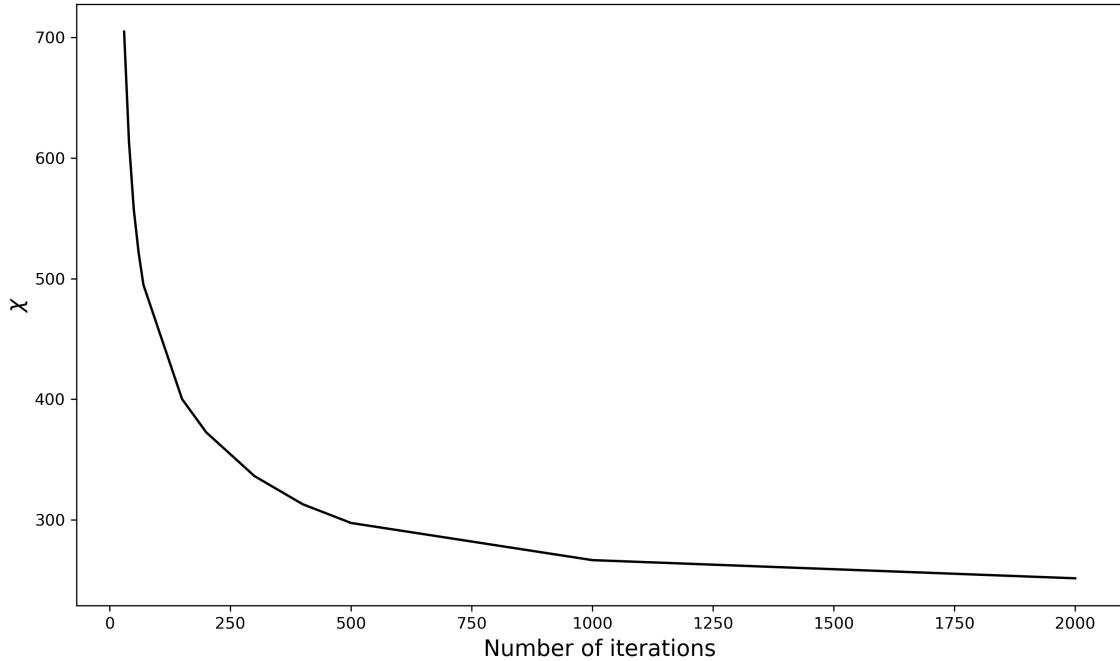


Figure 13: Convergence analysis of the CGLS method for the field data of Carajás, Brazil using the magnetic equivalent layer with a decimated irregular grid of $10,000 \times 131$ observation points up to 2,000 iterations.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

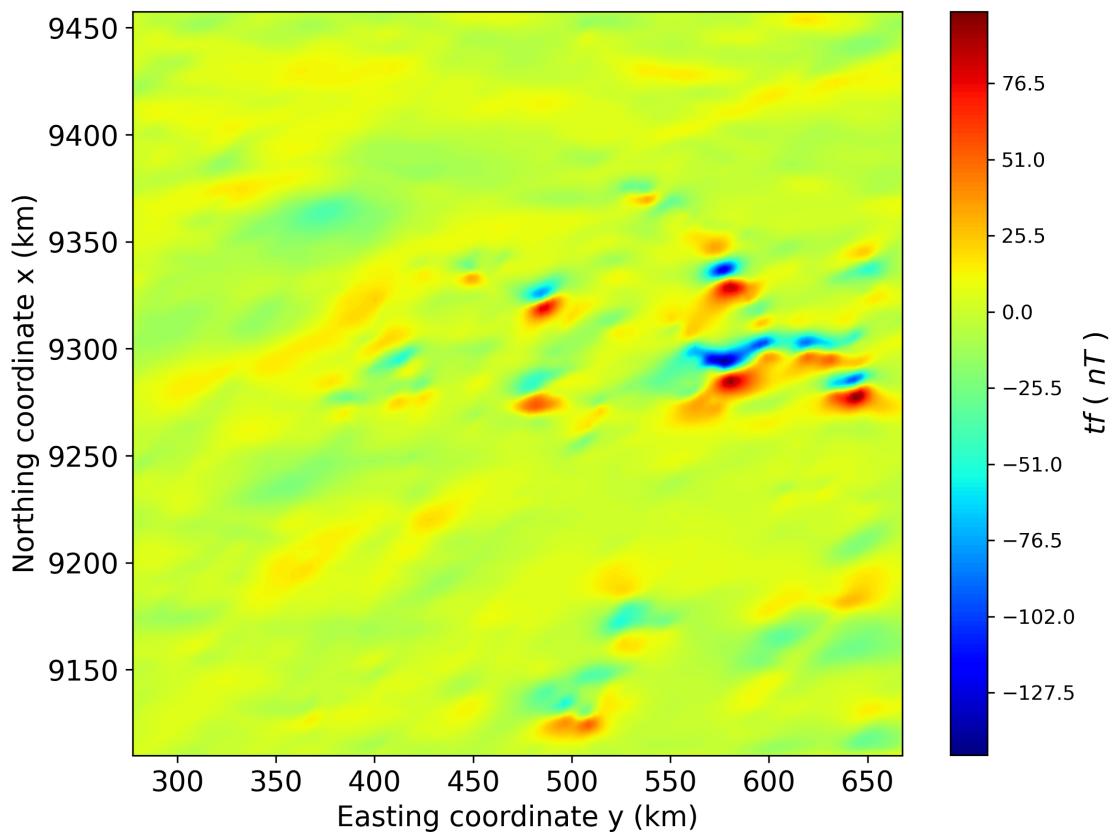


Figure 14: Upward continuation transformation of real data of Carajás, Brazil at 5,000 meter. It was necessary 2.64 seconds to complete the process.

Takahashi, Oliveira Jr. & Barbosa – GEO-XXXX

N	Matrix A	All six first columns of BCCB matrices	Matrix 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 matrix **A** (equation 12), the sum of all six first columns of the BCCB matrices embedded from the components of the matrix **H** from equation 10 (both need 8 bytes per element) and the matrix **L** containing the eigenvalues complex numbers (16 bytes per element) resulting from the diagonalization of matrix **C** (equation 63). Here we must consider that N observation points forms a $N \times N$ matrix.