Fast equivalent layer technique for gravity data processing with Block-Toeplitz Toeplitz-Block matrix systems

Diego Takahashi[*], *Vanderlei C. Oliveira Jr.*[*] *and Valéria C. F. Barbosa*[*]

ABSTRACT

We present a new approach of the fast equivalent layer technique for gravity data processing, modifying Siqueira et al. (2017)'s work, with the potential to use very large datasets at low computational cost. Taking advantage of the properties related to the symmetric Block-Toeplitz Toeplitz-block (BTTB) and Block-Circulant Circulant-Block (BCCB) matrices, that raises when regular grids of observation points and equivalent sources are used with the equivalent layer, we developed an algorithm which greatly reduces the number of flops and memory RAM necessary to complete the process of parameter estimative using this technique. The algorithm is based on the struture of symmetric BTTB matrices, where all its elements are comprised by the first row and can be embedded into a symmetric BCCB matrix, that also only needs its first row to be completed reconstructed. From the first column, the eigenvalues of BCCB matrices can be calculated using the Fast Fourier Transform, which can be used to readily compute matrix-vector products. Using examples, we demonstrate that even small and medium sized grids benefits from this approach and larger the dataset, faster and more efficient this method becomes compared to the usual. Synthetic tests using the equivalent layer with the method presented in this work evaluate this approach, and demonstrate satisfatory results for different gravity processing, as upward and downward continuation. Tests with real data from Carajás, Brazil, shows its applicabillity and potential for field processing.

INTRODUCTION

The equivalent layer is a well-known technique for processing potential-field data in applied geophysics since the 60's. It comes from potential theory as a mathematical solution of the Laplace's equation, in the region above the sources, by using the Dirichlet boundary condition (Kellogg, 1929). This theory states that any potential field produced by an arbitrary 3D physical property distribution can be exactly reproduced by a fictitious layer located at any depth and having a continuos 2D physical property distribution. In pratical situations, the layer is approximated by a finite set of sources (e.g., point masses or dipoles) and their physical properties are estimated by solving a linear system to fit the observed potential field. These fictitious sources are called equivalent sources.

Many previous works have used the equivalent layer as a processing technique in potential methods. Dampney (1969) used for gridding and vertical continuation. Cordell (1992); Mendonça and Silva (1994) used for data interpolation and gridding. Emilia (1973); Hansen and Miyazaki (1984); Li and Oldenburg (2010) for upward continuation. Silva (1986); Leão and Silva (1989); Guspí and Novara (2009); Oliveira Jr. et al. (2013) to reduction to the pole of magnetic data. Boggs and Dransfield (2004) for combination of multiple data sets. Barnes and Lumley (2011) for gradiente data processing.

The classic equivalent layer formulation consists on the multiplication between the property distribution and the kernel of the function. For gravity data, this kernel is a harmonic funtion related to the inverse of the distance between the observation point and the equivalent source. When these observation points and equivalent sources are regularly spaced, a Toeplitz system arises. Toeplitz systems are well-known in many fields of science as mathematics - numerical partial and ordinary diferential equations (Lin et al., 2003) -, image processing (Chan et al., 1999) and neural networks (Wray and Green, 1994). Chan and Jin (2007) and Jin (2003) give many examples of applications for Toeplitz systems.

In potential methods the Toeplitz system properties was used for downward continuation (Zhang et al., 2016) and for 3-D gravity field inversion using a 2-D multilayer model (Zhang and Wong, 2015). In the particular case of gravity data, the kernel generates a linear system with a matrix known as symmetric Block-Toeplitz Toeplitz-Block (BTTB).

Because of the importance and vast occurency of Toeplitz systems, many authors studied methods for solving them. Direct methods were conceived by Levinson (1946) and by Trench (1964). Currently the conjugate gradient is used in most cases. In Grenander (1984), Szegö noticed that a circulant matrix can be diagonalized by taking the fast Fourier trasform of its first column, making it possible to calculate the matrix-vector product and solve the system with low computational cost (Strang and Aarikka, 1986; Olkin, 1986). Chan and Jin (2007) shows some preconditioners to embedd the Toeplitz and BTTB matrices into circulant matrices and Block-Circulant Circulant-Block (BCCB), respectively, solving the system applying the conjugate gradient method.

Although the use of the equivalent layer technique increased over the last decades, one of the its biggest problem stills its high computational cost for processing large data sets. Siqueira et al. (2017) developed a computationally efficient scheme for processing gravity data. This scheme does not solve a linear system, instead uses an iterative process that corrects the property distribution of the layer by proportionally adding mass to the gravity residual. Although of its efficiency, at each iteration the forward problem must be solved to evaluate the convergence of the estimative. This processes accounts for most of the computational cost of this method.

We propose the use of BTTB and BCCB matrices properties to solve the forward problem of Siqueira et al. (2017)'s method in a more efficient way, resulting in faster parameter estimation and the possibility to use very large datasets. We show how the system memory RAM usage can be drastically decreased by calculating only the first row of the BTTB matrix and embedding into a BCCB matrix. Using the Szegö theorema combined with Strang and Aarikka (1986) the matrix-vector product can be accomplished with very low cost, reducing in some orders of magnitude the number of operations required to complete the process. We present synthetic tests to validate our proposal and real field data from Carajás, Brazil to demonstrate its applicability.

METHODOLOGY

Equivalent layer theory for gravity data

In applied geophysics, the observed gravity disturbance $\delta g(x, y, z)$ (Heiskanen and Moritz, 1967) is commonly approximated as the vertical component z of the gravitational attraction produced by gravity sources, as follows:

$$\delta g(x, y, z) = c_g\, G \int \int_v \int \rho(x', y', z') \frac{(z' - z) \quad dx'dy'dz'}{[(x - x')^2 + (y - y')^2 + (z - z')^2]^{\frac{3}{2}}} \,, \tag{1}$$

where $c_g = 10^5$ is a constant transforming from m/s^2 to $mGal$, G is the Newton's gravitational constant $(m^3/kg\, s^2)$ and $\rho(x', y', z')$ is the density at the point (x', y', z') inside the volume V of the source. This integral is defined in a Cartesian coordinate system with axis x pointing to north, y pointing to east and z pointing downward.

According to the classic upward continuation integral (Henderson, 1960, 1970), it is possible to compute the gravity disturbance $\delta g(x_i, y_i, z_i)$, at a point (x_i, y_i, z_i), from the gravity disturbance $\delta g(x, y, z_0)$ at the constant plane z_0:

$$\delta g(x_i, y_i, z_i) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\delta g(x, y, z_0)(z_0 - z_i)}{[(x_i - x)^2 + (y_i - y)^2 + (z_i - z_0)^2]^{\frac{3}{2}}}\, dx\, dy, \tag{2}$$

where, $z_0 > z_i$. Equation 2 shows that the gravity disturbance $\delta g(x_i, y_i, z_i)$ is a convolution between $\delta g(x, y, z_0)$ and another harmonic function on the horizontal plane $z = z_0$. Multiplying and dividing equation 2 by G and discretizing the integral, we obtain:

$$\delta g(x_i, y_i, z_i) = \sum_{j=1}^{N} p_j a_{ij} \,, \tag{3}$$

where a_{ij} is given by:

$$a_{ij} = c_g\, G \frac{(z_j - z_i)}{[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2]^{\frac{3}{2}}} \tag{4}$$

and p_j is the coefficient representing the physical property of the j-th equivalent source. In this case, the harmonic function a_{ij} represents the vertical component of the gravitational attraction exerted, at the point (x_i, y_i, z_i), by a point mass located at the point (x_j, y_j, z_j), with mass p_j. Equation 3 can be expressed in matrix form as:

$$\mathbf{d}(\mathbf{p}) = \mathbf{A}\mathbf{p} \,, \tag{5}$$

where $\mathbf{d}(\mathbf{p})$ is an $N \times 1$ vector, whose i-th element is the predicted gravity disturbance $\delta g(x_i, y_i, z_i)$, $\mathbf{p}$ is an $N \times 1$ parameter vector, whose j-th element is the coefficient p_j representing the physical property of the j-th equivalent source and $\mathbf{A}$ is an $N \times N$ sensibility matrix, where each element a_{ij} is given by equation 4 and defines the field produced by the j-th equivalent source at the point (x_i, y_i, z_i). The solution of the parameters $\mathbf{p}$ can be solved by a linear inversion minimizing the function:

$$\Psi(\mathbf{p}) = \theta_g(\mathbf{p}) + \mu\, \theta_m(\mathbf{p}) \,, \tag{6}$$

where $\theta_g(\mathbf{p})$ is the misfit function given by:

$$\theta_g(\mathbf{p}) = ||\mathbf{d}^o - \mathbf{d}(\mathbf{p})||_2^2 \,, \tag{7}$$

which is the euclidian norm of the residual between the observed data $\mathbf{d}^o$ and the predicted data $\mathbf{d}(\mathbf{p})$.

The function $\theta_m(\mathbf{p})$ is the regularization, for example, the zeroth-order Tikhonov:

$$\theta_m(\mathbf{p}) = ||\mathbf{p}||_2^2 \,, \tag{8}$$

which is the euclidian norm of the parameters $\mathbf{p}$. The variable μ is a real positive number regularizing the parameter.

Derivating equation 6 in relation to $\mathbf{p}$ and making equal to zero, it is possible to estimate the parameters:

$$\mathbf{p}^* = \left(\mathbf{A}^\top \mathbf{A} + \mu\, \mathbf{I}\right)^{-1} \mathbf{A}^\top \mathbf{d}^o \,. \tag{9}$$

Fast equivalent layer technique

In Siqueira et al. (2017) the authors developed an iterative least-squares method to estimate the mass distribution of the equivalent layer based on the excess of mass and the positive correlation between the observed gravity data and the equivalent sources. This scheme is proven to have a better computational efficiency than the classical equivalent layer approach with data sets of at least 500 observation points, even using a large number of iterations. This work also proves that the excess of mass of a body is proportional to the surface integration of the gravity data. Considering each equivalent source directly beneath the observation points, a initial approximation of mass distibution is made:

$$\mathbf{p}^0 = \tilde{\mathbf{A}}^{-1} \mathbf{d}^o \,, \tag{10}$$

where $\mathbf{d}^o$ is the gravity data and $\tilde{\mathbf{A}}^{-1} = \Delta s / (2\pi\, G\, c_g)$ with Δs being an element of area located at the vertical coordinate z_i and centered at the horizontal coordinates (x_i, y_i), $i = 1, ..., N$. At each k-th iteration a mass correction vector $\mathbf{\Delta m^k}$ for every source is calculated by minimizing the function:

$$\phi(\mathbf{\Delta p^k}) = ||\mathbf{d}^o - \mathbf{A}\hat{\mathbf{p}}^k - \tilde{\mathbf{A}}\mathbf{\Delta p^k}||_2^2 \tag{11}$$

and the mass distribution of the equivalent sources is updated as:

$$\mathbf{\Delta p^{k+1}} = \mathbf{p^k} + \mathbf{\Delta p^k} \,. \tag{12}$$

Listing 1: A *Python* algorithm for Siqueira et al. (2017)'s fast equivalent layer.

```
diagonal_A = G*(10**5)*2*pi/(dx*dy)
for i in range (itmax):
        res = (data - A.dot(p))
        delta_p = res/diagonal_A
        p += delta_p
```

Each iteration of this algorithm, the matrix-vector product $\mathbf{Ap}$ must be calculated to get a new residual $\mathbf{d^0} - \mathbf{Ap}$. While it is true that for a small number of observation points this represents no computational effort, for very large data sets it is costful and can be overwhelming in terms of memory RAM to maintain such operation.

Structure of sensibility matrix A

If a gravity disturbance data $\delta g(x_i, y_i, z_i)$ (equation 3) is placed on a regular grid at the constant vertical coordinate $z_i = z_1$, $i = 1, \ldots, N$, and there is an equivalent source located directly below each point of this grid at a constant depth z_0, the elements a_{ij} (equation 4) can be rewritten as follows:

$$a_{ij} = c_g G \frac{\Delta z}{\left[(r_{ij})^2 + (\Delta z)^2\right]^{\frac{3}{2}}} , \tag{13}$$

where $\Delta z = z_0 - z_1$ and $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ represents the relative horizontal distance between the i-th observation point and the j-th equivalent source. It is worth noting that: (i) $r_{ij} = r_{ji}$ for any pair ij and (ii) r_{ij} computed for different pairs ij may assume the same value. As a consequence of these two properties, the matrix $\mathbf{A}$ (equation 5) has the structure of a symmetric Toeplitz by blocks, where each block is also a symmetric Toeplitz matrix (BTTB) (Chan and Jin, 2007; Golub and Loan, 2013). For example, consider a regular grid of $N_x \times N_y$ points, where $N_x = 4$ and $N_y = 3$, comprising a total number of $N = N_x \times N_y = 12$ points. In this case, matrix $\mathbf{A}$ is $N_x \times N_x$ grid of matrices given by:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_1} & \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} \\ \mathbf{A_2} & \mathbf{A_1} & \mathbf{A_0} & \mathbf{A_1} \\ \mathbf{A_3} & \mathbf{A_2} & \mathbf{A_1} & \mathbf{A_0} \end{pmatrix} , \tag{14}$$

where each block is a $N_y \times N_y$ matrix given by:

$$\mathbf{A}_l = \begin{pmatrix} a_0^\ell & a_1^\ell & a_2^\ell \\ a_1^\ell & a_0^\ell & a_1^\ell \\ a_2^\ell & a_1^\ell & a_0^\ell \end{pmatrix} , \tag{15}$$

where the elements a_k^l, $l = 0, \ldots, N_x - 1$, $k = 0, \ldots, N_y - 1$ are computed with equation 13. In this case, the complete matrix $\mathbf{A}$ can be obtained by computing only its first row or column.

BCCB matrix-vector product

As previous discussed in the Fast equivalent layer technique, the matrix-vector product accounts for most of the computational cost (Listing 1). When large data sets are used, this operation can take some time and even be prohibited by memory RAM shortage. In order to lessen this problem we transform the BTTB matrix $\mathbf{A}$ into a Block-Circulating Circulating-Block (BCCB) matrix $\mathbf{C}$ and use its eigenvalues to carry the product of $\mathbf{A}$ and an arbitrary vector $\mathbf{p}$. This strategy has been successfully applied by Zhang and Wong (2015) and Zhang et al. (2016) for optmizing the computational cost of 3D gravity inversion and downward continuation of potential field, respectively. Here, we use this strategy for improving the computational efficiency of the Fast equivalent layer method proposed by Siqueira et al. (2017).

Following the example of our BTTB matrix $\mathbf{A}$ (equation 14), its transformation into a BCCB matrix $\mathbf{C}$ is given by:

$$\mathbf{C} = \begin{pmatrix} \mathbf{C_0} & \mathbf{C_1} & \mathbf{C_2} & \mathbf{C_3} & \mathbf{0} & \mathbf{C_3} & \mathbf{C_2} & \mathbf{C_1} \\ \mathbf{C_1} & \mathbf{C_0} & \mathbf{C_1} & \mathbf{C_2} & \mathbf{C_3} & \mathbf{0} & \mathbf{C_3} & \mathbf{C_2} \\ \mathbf{C_2} & \mathbf{C_1} & \mathbf{C_0} & \mathbf{C_1} & \mathbf{C_2} & \mathbf{C_3} & \mathbf{0} & \mathbf{C_3} \\ \mathbf{C_3} & \mathbf{C_2} & \mathbf{C_1} & \mathbf{C_0} & \mathbf{C_1} & \mathbf{C_2} & \mathbf{C_3} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_3} & \mathbf{C_2} & \mathbf{C_1} & \mathbf{C_0} & \mathbf{C_1} & \mathbf{C_2} & \mathbf{C_3} \\ \mathbf{C_3} & \mathbf{0} & \mathbf{C_3} & \mathbf{C_2} & \mathbf{C_1} & \mathbf{C_0} & \mathbf{C_1} & \mathbf{C_2} \\ \mathbf{C_2} & \mathbf{C_3} & \mathbf{0} & \mathbf{C_3} & \mathbf{C_2} & \mathbf{C_1} & \mathbf{C_0} & \mathbf{C_1} \\ \mathbf{C_1} & \mathbf{C_2} & \mathbf{C_3} & \mathbf{0} & \mathbf{C_3} & \mathbf{C_2} & \mathbf{C_1} & \mathbf{C_0} \end{pmatrix}, \tag{16}$$

where each block $\mathbf{C}_l$ is:

$$\mathbf{C}_l = \begin{pmatrix} \mathbf{A}_l & \times \\ \times & \mathbf{A}_l \end{pmatrix} = \begin{pmatrix} a_0^\ell & a_1^\ell & a_2^\ell & 0^\ell & a_2^\ell & a_1^\ell \\ a_1^\ell & a_0^\ell & a_1^\ell & a_2^\ell & 0^\ell & a_2^\ell \\ a_2^\ell & a_1^\ell & a_0^\ell & a_1^\ell & a_2^\ell & 0^\ell \\ 0^\ell & a_2^\ell & a_1^\ell & a_0^\ell & a_1^\ell & a_2^\ell \\ a_2^\ell & 0^\ell & a_2^\ell & a_0^\ell & a_0^\ell & a_1^\ell \\ a_1^\ell & a_2^\ell & 0^\ell & a_2^\ell & a_1^\ell & a_0^\ell \end{pmatrix}. \tag{17}$$

Matrix $\mathbf{C}$ is a $4N_xN_y \times 4N_xN_y$ Block-Circulant formed by Circulant-Blocks matrix. This matrix is formed by a grid of $2N_x \times 2N_x$ blocks, where each block is a $2N_y \times 2N_y$ matrix.

The product $\mathbf{d} = \mathbf{Ap}$ (equation 5), then becomes:

$$\mathbf{Cv} = \mathbf{q}\,, \tag{18}$$

where $\mathbf{v}$ and $\mathbf{q}$ are $4N_xN_y \times 1$ vectors given by:

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{N_x-1} \\ \mathbf{0}_{(2N_xN_y)} \end{pmatrix} \tag{19}$$

and

$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_{N_x-1} \\ \mathbf{0}_{(2N_xN_y)} \end{pmatrix}, \tag{20}$$

where $\mathbf{0}_{(2N_xN_y)}$ is a $2N_xN_y \times 1$ vetor of zeros, and $\mathbf{v}_\ell$ and $\mathbf{q}_\ell$, $\ell = 0, \ldots, N_x - 1$ are:

$$\mathbf{v}_\ell = \begin{pmatrix} \mathbf{p}_\ell \\ \mathbf{0}_{(N_y)} \end{pmatrix} \tag{21}$$

and

$$\mathbf{q}_\ell = \begin{pmatrix} \mathbf{d}_\ell \\ \mathbf{0}_{(N_y)} \end{pmatrix}, \tag{22}$$

where $\mathbf{0}_{(N_y)}$ is a $N_y \times 1$ vetor of zeros.

By using the Kronecker product properties, the auxiliary matrix-vetor product can be rewritten as follows:

$$\mathbf{F}^*_{(2N_x)} \left[\mathbf{L} \circ \left(\mathbf{F}_{(2N_x)} \mathbf{V}\, \mathbf{F}_{(2N_y)}\right)\right] \mathbf{F}^*_{(2N_y)} = \mathbf{Q}\,, \tag{23}$$

where "$\circ$" denotes the Hadamard product, $\mathbf{L}$ is a $2N_x \times 2N_y$ row-oriented matrix containing the eigenvalues of $\mathbf{C}_{(BCCB)}$, and $\mathbf{V}$ and $\mathbf{Q}$ are $2N_x \times 2N_y$ row-oriented matrices obtained from the vectors $\mathbf{v}$ and $\mathbf{q}$.

One of the properties of BCCB matrices is that its eigenvalues can be calculated by a 2D Discrete Fourier Transform (Chan and Jin, 2007). This lead to a fast calculation of $\mathbf{L}$ using a 2D Fast Fourier Transform, making the matrix-vector product a low computational cost process.

Note that in general, the first column of blocks forming a BCCB matrix $\mathbf{C}_\ell$ (equation 16) is given by:

$$[\mathbf{C}]_{(0)} = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{C}_1 \\ \vdots \\ \mathbf{C}_{N_x-2} \\ \mathbf{C}_{N_x-1} \\ \mathbf{0} \\ \mathbf{C}_{N_x-1} \\ \mathbf{C}_{N_x-2} \\ \vdots \\ \mathbf{C}_1 \end{pmatrix}, \tag{24}$$

where each block $\mathbf{C}_\ell$, $\ell = 0, \ldots, N_x - 1$, is a $2N_y \times 2N_y$ circulant matrix and $\mathbf{0}$ is a $2N_y \times 2N_y$ matrix with all elements equal to zero. Thus, the first column of a circulant matrix $\mathbf{C}_\ell$ is given by:

$$
[\mathbf{C}_\ell]_{(0)} = \begin{pmatrix} a^\ell_{00} \\ a^\ell_{10} \\ \vdots \\ a^\ell_{(N_y-2)0} \\ a^\ell_{N_y-1)0} \\ 0 \\ a^\ell_{N_y-1)0} \\ a^\ell_{(N_y-2)0} \\ \vdots \\ a^\ell_{10} \end{pmatrix} . \tag{25}
$$

To complete the process, after calculating the inverse of $\mathbf{Q}$ it is necessary to rearrange its rows to obtain the vector $\mathbf{q}$ and also rearrange the elements of $\mathbf{q}$ to obtain the wanted vector $\mathbf{d}(\mathbf{p})$.

Computational performance

Equation 11 now can be calculated with the approach presented in the previous section. In a normal procedure of the fast equivalent layer, at each iteration a full matrix $\mathbf{A}$ (equation 5) is multiplied by the estimated mass distribution parameter vector $\hat{\mathbf{p}}^k$. As pointed in Siqueira et al. (2017) the number of flops (floating-point operations) necessary to estimate the N-dimensional parameter vector inside the iteration loop is:

$$
f_0 = N^{it}(3N + 2N^2). \tag{26}
$$

From equation 26 it is clear that the matrix-vector product ($2N^2$) accounts for most of the computational complexity in this method.

It is well known that FFT takes $N \log_2(N)$ flops (Brigham and Brigham, 1988). Computing the eigenvalues of the BCCB matrix ($4N \times 4N$) and applying 2D-FFT on the parameter vector (equation 23), takes $4N \log(4N)$ each. The point-multiplication takes $4N$. As it is necessary to compute the inverse FFT another two $4N \log(4N)$ must be taken in account. However, the sensibility matrix does not change during the process, thus, the eigenvalues of BCCB must be calculated only once, outside of the iteration. This lead us to:

$$
f_1 = N^{it}(7N + 12N \log(4N)). \tag{27}
$$

Another major improvement of this methodology is the exoneration of calculating the full sensibility matrix $\mathbf{A}$ (equation 5). Each element needs 12 flops (equation 4), totalizing $12N^2$ flops for the full matrix. Calculating only the first row of the BTTB matrix, $12N$ flops is required and the computation of the eigenvalues is $4N \log(4N)$ as mentioned above. The full flops count of Siqueira et al. (2017)'s method:

$$
f_s = 12N^2 + N^{it}(3N + 2N^2), \tag{28}
$$

is decreased in our method to:

$$f_s = 12N + 4N\log(4N) + N^{it}(7N + 12N\log(4N)). \tag{29}$$

Figure 1 shows the floating points to estimate the parameter vector using the fast equivalent layer with Siqueira et al. (2017)'s method (equation 26) and our approach (equation 27) versus the number of observation points varyig from $N = 5000$ to $N = 1000000$ with 50 iterations. The number of operations is drastically decreased.

Table 1 shows the system memory RAM usage needed to store the full matrix, the BTTB first row and the BCCB eigenvalues (8 times the BTTB first row). The quantities were computed for different numbers of data (N) with the same corresponding number of equivalent sources (N). This table considers that each element of the matrix is a double-precision number, which requires 8 bytes of storage, except for the BCCB complex eigenvalues, which requires 16 bytes per element. Notice that 1000000 observation points requires nearly 7.6 Terabytes of memory ram to store the whole sensibility matrix of the equivalent layer.

Using a PC with a Intel Core i7 4790@3.6GHz processor and 16 Gb of memory RAM, figure 2 shows the time necessary to run 50 iterations of the Siqueira et al. (2017)'s method and the one presented in this work. After 10000 observations points is clear how this method benefits from the new approach in calculating the forward problem. Because of the memory RAM available in this system, we could not test the comparison with more observations, limited to 22500. In figure 3 we show the time necessary to run the equivalent layer technique with 50 iterations using only the new approach, where the RAM is not a limitation factor. We could run up to 25 million observation points. In comparison, one million observation points took 26.8 seconds to run, where the maximum 22500 observation points, with Siqueira et al. (2017)'s method, took 48.3 seconds.

SYNTHETIC TESTS

The synthetic data presented in this section has the objective to validate this new approach when used jointly with the fast equivalent layer presented in Siqueira et al. (2017). We constructed a model with two polygonal prisms, with density contrast of 0.35 (upper-left body) and $0.4g/cm^3$ (upper-right body), and a sphere with radius of $1000m$ with density contrast of $-0.5g/cm^3$. The vertical component of gravity generated by this bodies were calculated and are shown in figure 4 together with their horizontal projections. A gaussian noise was added to the data, with mean of zero and a maximum of 0.5% of the maximum value of the original data. As previous said, only in regular grids the BTTB matrix structures appears. We created 10000 observation points regularly spaced in a grid of 100×100, with an uniform 100 m of height for all the observations.

In figure 5 we show the fitted data with the fast equivalent layer using Siqueira et al. (2017)'s work. In comparison we have the figure 6 showing the fitted data with the new approach presented in this work by calculating the forward problem using equation 18. In both figures A) is the original contaminated synthetic data, B) is the predict data by the equivalent layer and C) is the residual between the synthetic data and fitted data, with mean of $-8.264e^{-7}$ and standard deviation of 0.0144. As we can see in the figure 7, there is virtually no difference in the fitted data presented in figures 5b and 6c, showing that the result associated by calculating a matrix-vector product of a embedded BTTB into a BCCB matrix using equation 18 is the same as a normal matrix-vecotr product. In figure 8 we have the difference between the mass distribution of the equivalent sources estimated by using the two methods.

In figure 9 and figure 10 we show two forms of processing a gravity data using the equivalent layer, the upward and the downward continuation, respectively. The upward height is $300m$ and the downward is at $50m$. Both figures show: A) the upward or downward in the traditional Siqueira et al. (2017)'s work B) the upward or downward using this new approach and C) the residuals between the two forms of processing. For the upward continuation the mean of the residuals is $-5.938e^{-18}$ and the standard deviation is $8.701e^{-18}$. For the downward continuation the mean of the residuals is $5.914e^{-18}$ and the standard deviation is $9.014e^{-16}$. With Siqueira et al. (2017)'s method the upward processing took 7.62026 seconds and 0.00834 seconds with the new approach. For the downward processing the times necessary were 7.59654 seconds and 0.00547 seconds, respectively.

REAL DATA TESTS

Tests with real data are conducted with the gravity data of Carajás provided by CPRM (Companhia de Pesquisa de Recursos Minerais). This area covers the southeast part of the state of Pará, Brazil. Aeromagnetic and aerogravimetric data were collected in 113 flight lines with 3 km apart from each other with N-S orientation. There was two separated teams for data collection, each responsible for a determined area. For gravity data the sample spacing were $7.65m$ and $15.21m$ for each team, totalizing more 4353428 observation points. The height of the flights were fixed at $900m$. All 4353428 million gravity data were gridded into a regularly spaced dataset of 250000 observation points (500×500) for processing (figure 11).

Figure 12 shows the gridded gravity data (A), the fitted data with 50 iterations of the fast equivalent layer at 300 m depth using this new approach (B) and the residual (C). The mean of the residual was 0.000292 and standard deviation of 0.105 which demonstrates a good fit for the predicted data, evaluating this technique to be applied in real field data.

An upward continuation processing were made (Figure 13) at 5000 m over the real data. It shows a reasonable processing, atenuating the short-wave lenghts. The processing of the 250000 observations points took 0.216 seconds.

CONCLUSIONS

By exploring the properties related to Block-Toeplitz Toeplitz-block (BTTB) and Block-Circulant Circulant -Block (BCCB) matrices, we show a new approach for calculating the matrix-vector product of the fast equivalent layer technique from Siqueira et al. (2017)'s work when regular grids of observation points and equivalent sources are employed. This alghorithm greatly reduces the number of flops necessary to complete the estimative of the equivalent layer. For example, when processing one million observation points, the number of flops is reduced in 10^4 times. When processing such amount of data, the full sensibility matrix takes 7.6 Terabytes of memory RAM storage, which is impractical, while takes only 61.035 Megabytes with the method presented in this work.

This approach takes advantage of the symmetric BTTB system that arises when processing a harmonic function as the vertical component of gravity, that depends on the inverse of distance. Symmetric BTTB matrices can be stored by its only first row and can be embedded into a symmetric BCCB matrix that also only needs its first row. Using the Fast Fourier Transform it is possible to calculate the eigenvalues of BCCB matrices which can be used to compute a matrix-vector product in a very low computational cost. The time necessary to process medium sized grids of observational data, for example 22500 points, is cutted in 10^2 times.

Synthetic and real data tests were conducted with the equivalent layer using the method presented in this work, showing satisfatory results that evaluate it to process gravity data, as upward and downward continuations, even using very large datasets. In the future, applications of the equivalent layer using great amount of data, as in continental or global scale can be researched.

Figures

Figure 1

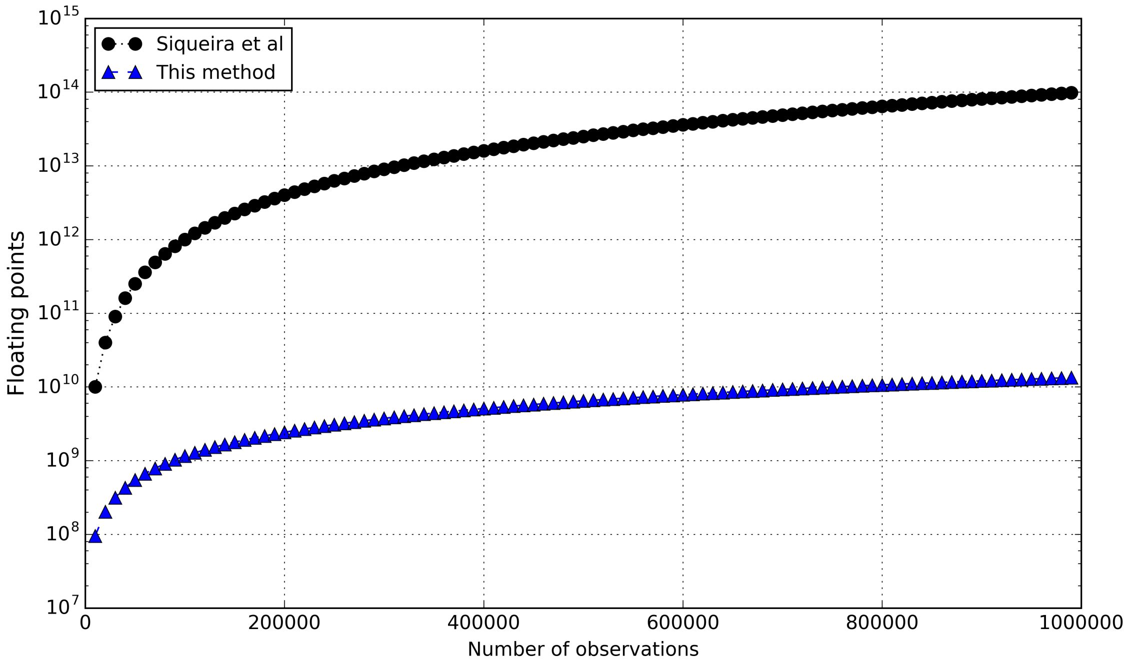

Figure 1: floating points to estimate the parameter vector using the fast equivalent layer with Siqueira et al. (2017)'s method (equation 26) and our approach (equation 27) versus the numbers of observation points varyig from $N = 5000$ to $N = 1000000$ with 50 iterations. The number of operations is drastically decreased.

Figure 2

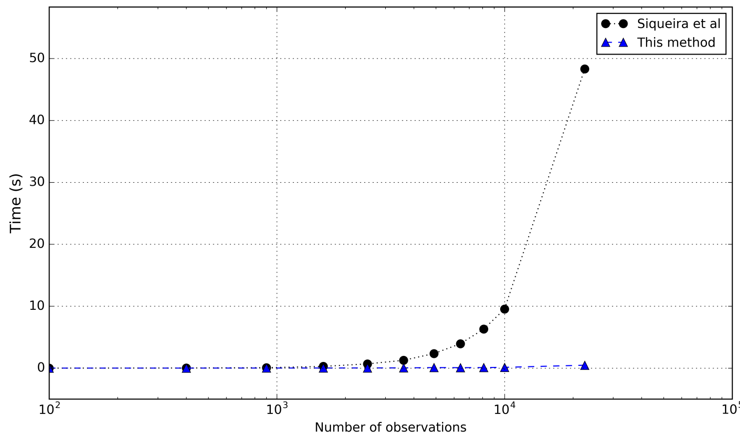

Figure 2: time necessary to run 50 iterations of the Siqueira et al. (2017)'s method and the one presented in this work. With the limitation of 16 Gb of memory RAM in our system, we could test only up to 22500 obervation points.

Figure 3

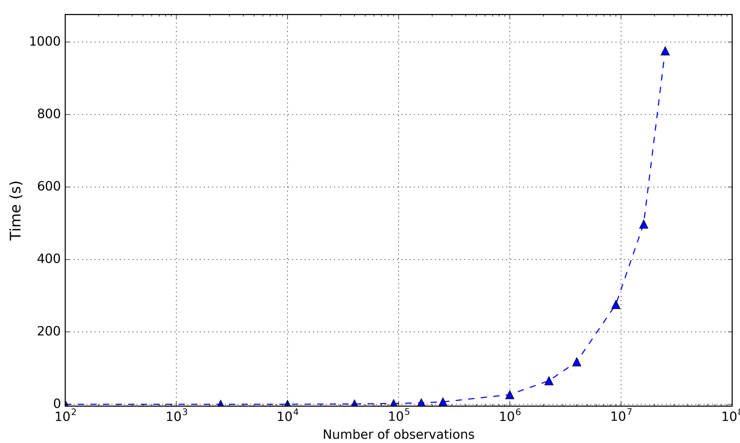

Figure 3: time necessary to run the equivalent layer technique with 50 iterations using only this new approach, where the RAM is not a limitation factor. We could run up to 25 million observation points. In comparison, 1 million observation points took 26.8 seconds to run, where the maximun 22500 observation points in figure 2, with Siqueira's method, took 48.3 seconds.

Figure 4

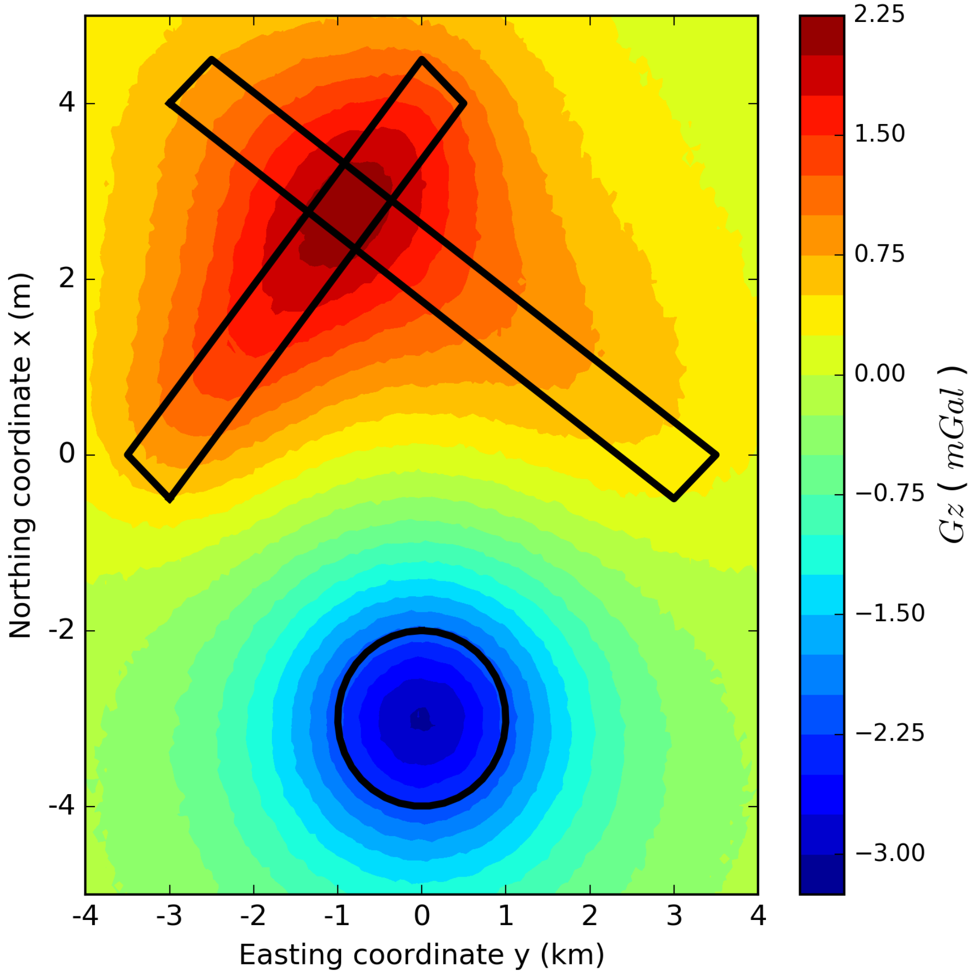

Figure 4: model with two polygonal prisms, with density contrast of 0.35 (upper-left body) and $0.4g/cm^3$ (upper-right body), and a sphere with radius of $1000m$ with density contrast of $-0.5g/cm^3$. The vertical component of gravity generated by this bodies were calculated and are shown together with their horizontal projections. A gaussian noise was added to the data with mean of zero and maximum value of 0.5% of the maximum of the original data. As previous said only in regular grids the BTTB matrix structures appears. We created 10000 observation points regularly spaced in a grid of 100×100, with a uniform $100m$ of height for all the observations.

Figure 5

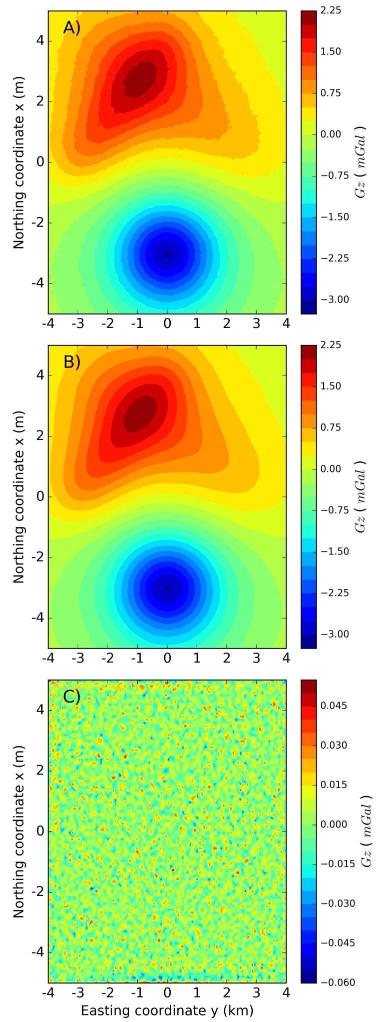

Figure 5: A) original contaminated synthetic data, B) fitted data using the fast equivalent layer with the Siqueira et al. (2017)'s work. C) residual between the synthetic data and fitted data, with mean of $-8.264e^{-7}$ and standard deviation of 0.0144.

Figure 6

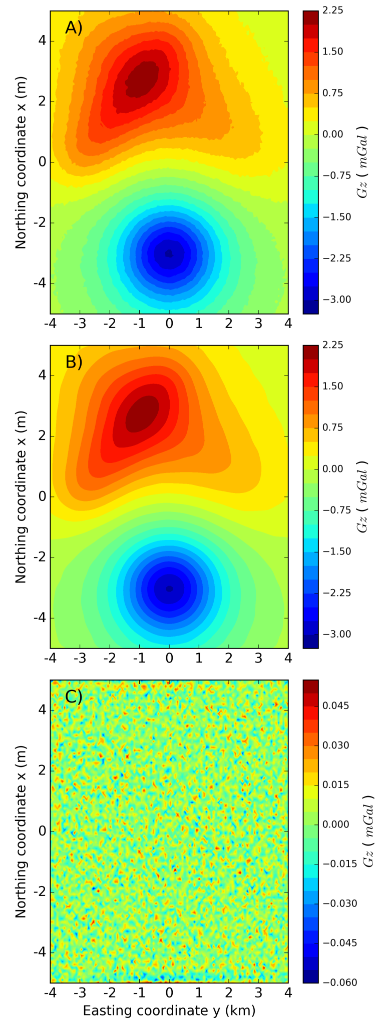

Figure 6: A) original contaminated synthetic data, B) fitted data with the new approach presented in this work of calculating the forward problem using equation 18 for the equivalent layer. C) residual between the synthetic data and fitted data, with mean of $-8.264e^{-7}$ and standard deviation of 0.0144.

Figure 7

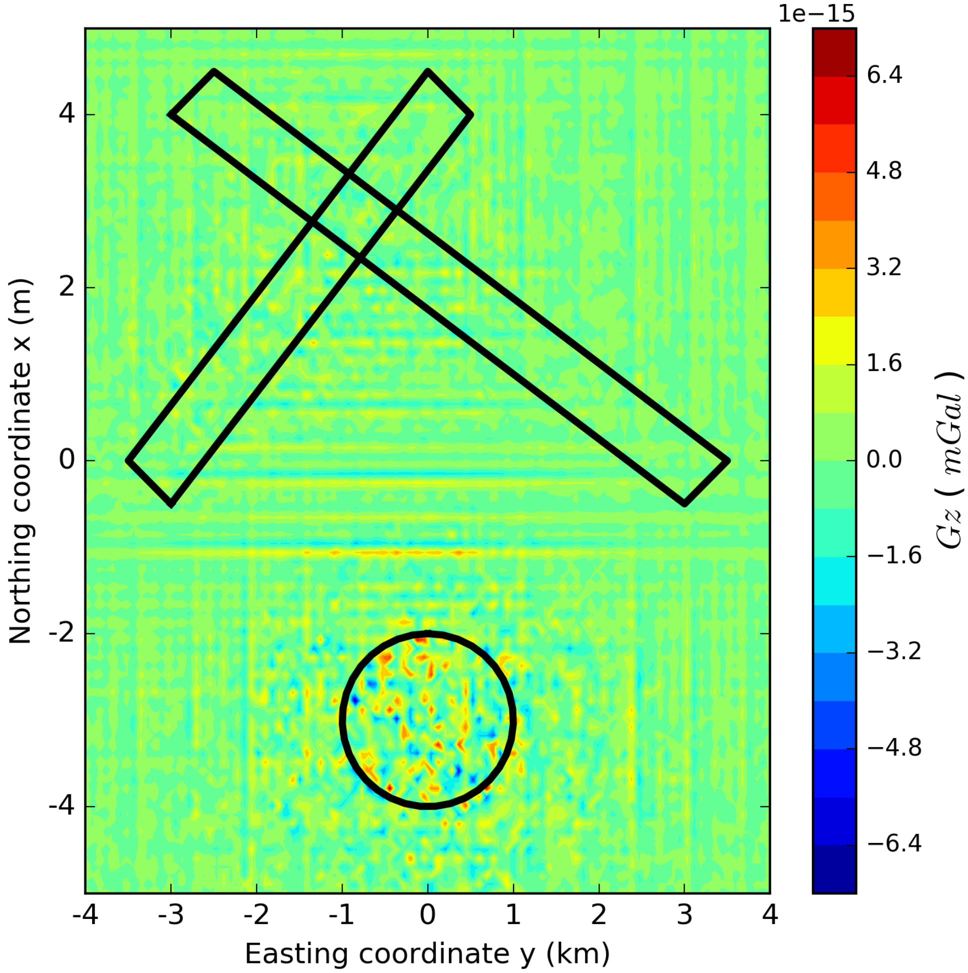

Figure 7: virtually zero difference in the fitted data presented in figures 5b and 6c, showing that matrix-vector product of a embedded BTTB into a BCCB matrix using equation 18 is pratically equal to a normal matrix-vector product.

Figure 8

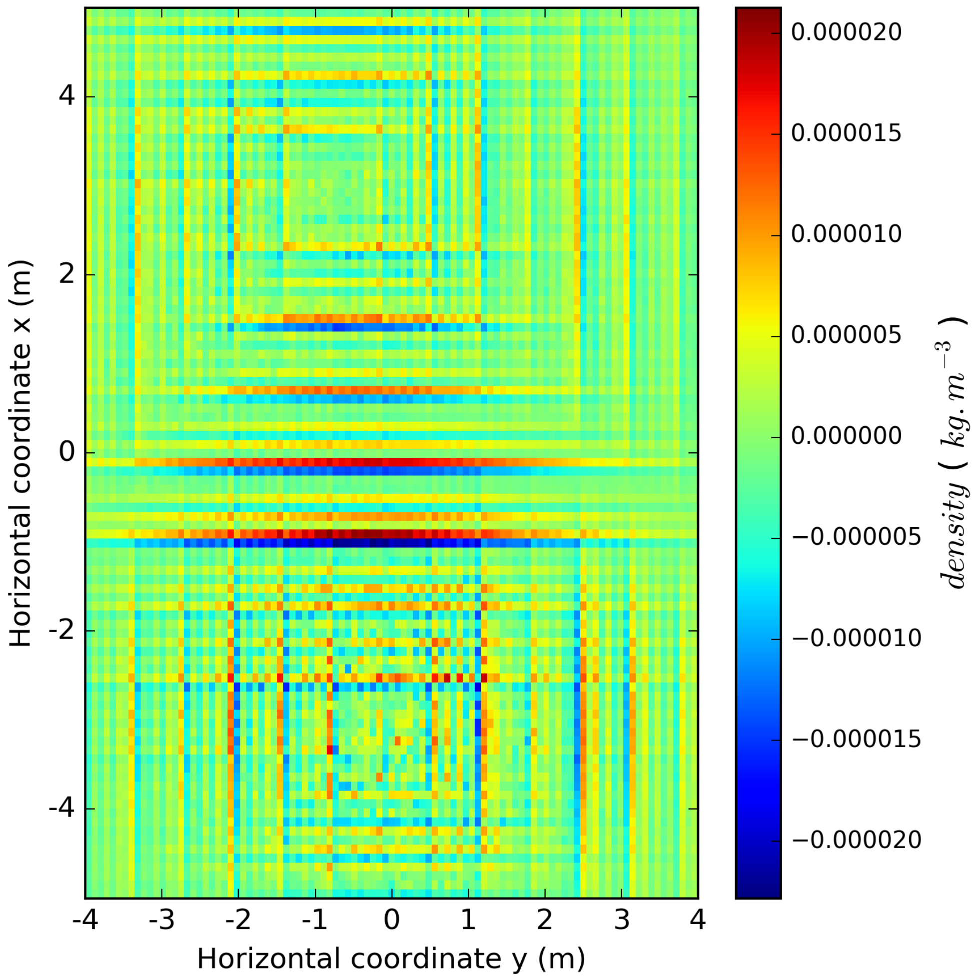

Figure 8: small difference in the estimated density between the classic matrix-vector product and the matrix-vector product of a embedded BTTB into a BCCB matrix using equation 18, showing that the estimative using this work's method is pratically the same.

Figure 9

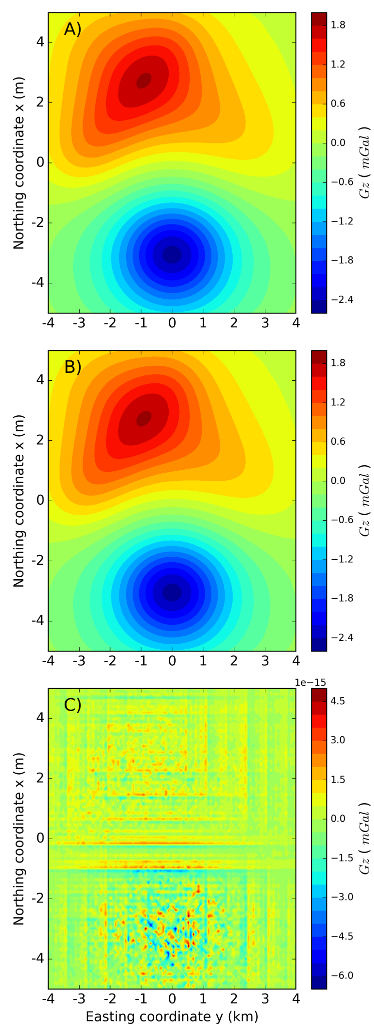

Figure 9: A) the upward continuation conducted at the height of $300m$ using the traditional Siqueira et al. (2017)'s work B) the upward continuation conducted at the same height of $300m$ using this new approach and C) residuals between the two forms of processing with mean of $-5.938e^{-18}$ and standard deviation of $8.701e^{-18}$. With Siqueira et al. (2017)'s method this processing took 7.62026 seconds and 0.00834 seconds with the new approach.

Figure 10

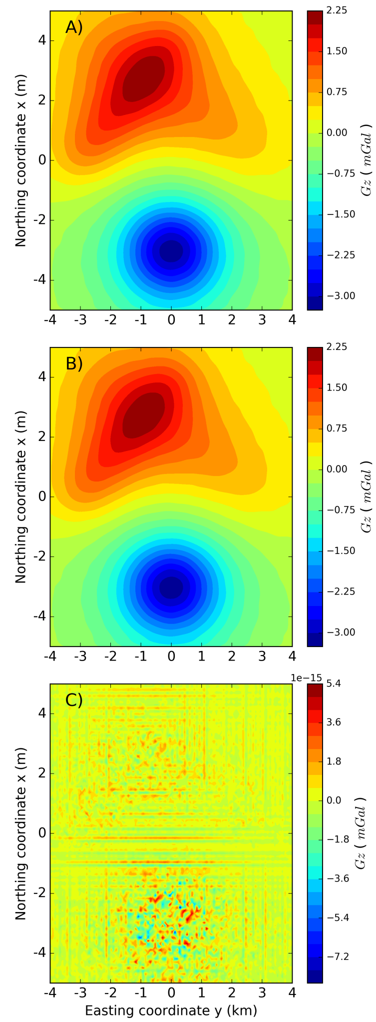

Figure 10: A) the downward continuation conducted at the height of $50m$ using the traditional Siqueira et al. (2017)'s work B) the upward continuation conducted at the same height of $50m$ using this new approach and C) residuals between the two forms of processing with mean of $5.914e^{-18}$ and standard deviation of $9.014e^{-16}$. With Siqueira et al. (2017)'s method this processing took 7.59654 seconds and 0.00547 seconds with the new approach.

Figure 11

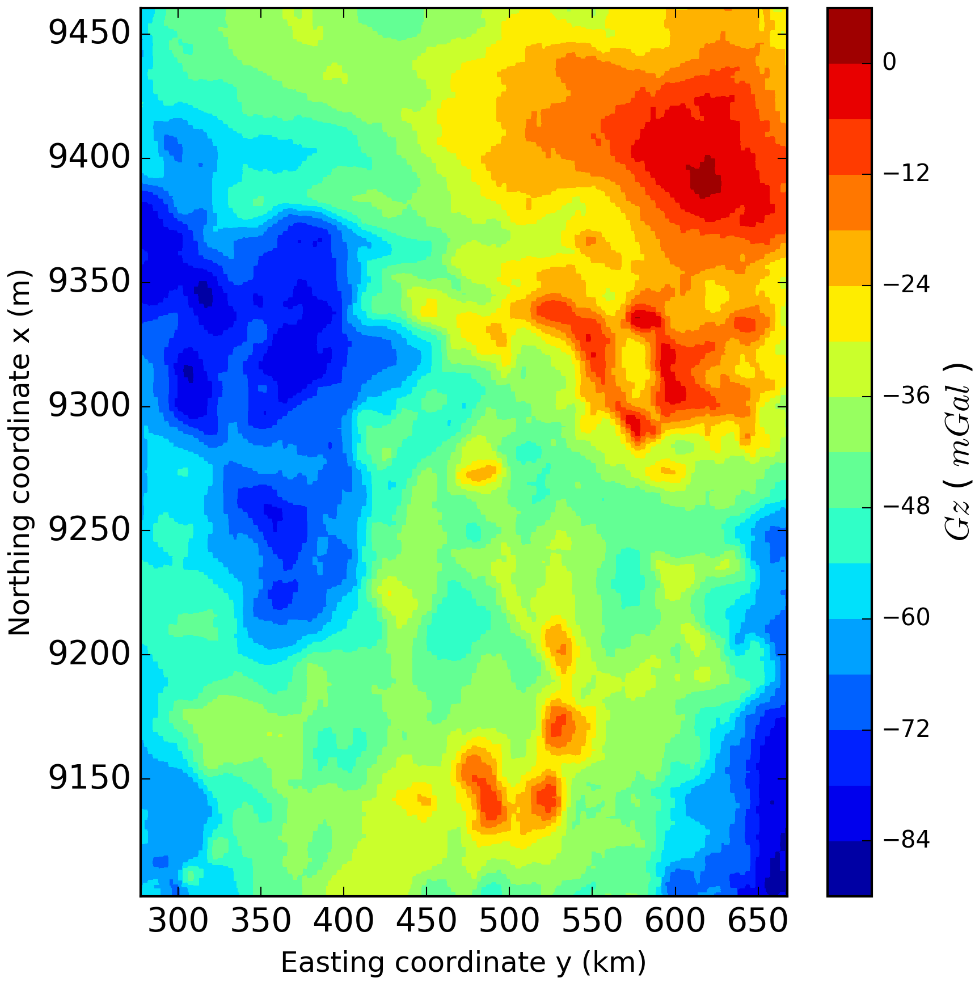

Figure 11: real data gravity data of Carajás gridded into a regularly spaced dataset of 250000 observation points (500×500). This area covers the southeast part of the state of Pará, Brazil. Aerogravimetric data was collected in 113 flight lines with $3km$ apart from each other and N-S orientation, totalizing more than 4 million observation points. The height of the flights were fixed at $900m$. All $4million$ gravity data were gridded for processing.

Figure 12

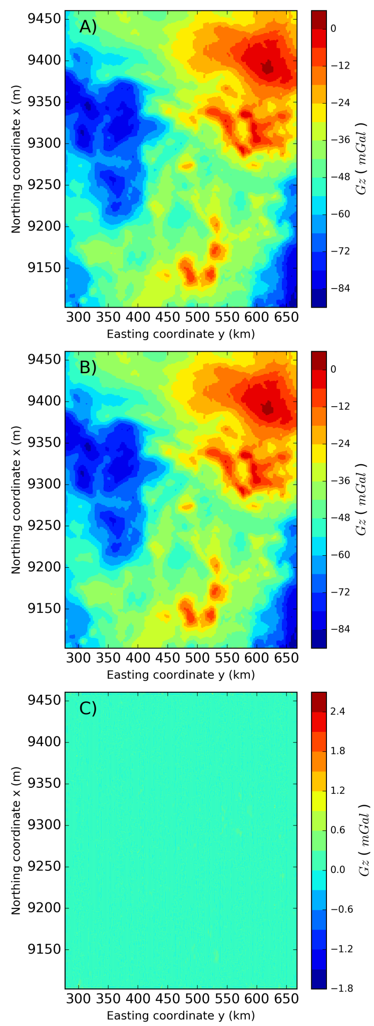

Figure 12: A) gridded gravity data. B) fitted data with 50 iterations of the fast equivalent layer at 400 m depth using the new approach of this work. C) residual, defined as the difference between A) and B). The mean of the residual was 0.000292 and standard deviation of 0.105 which demonstrates a good fit for the predicted data, evaluating this technique to be applied in real field data.

Figure 13

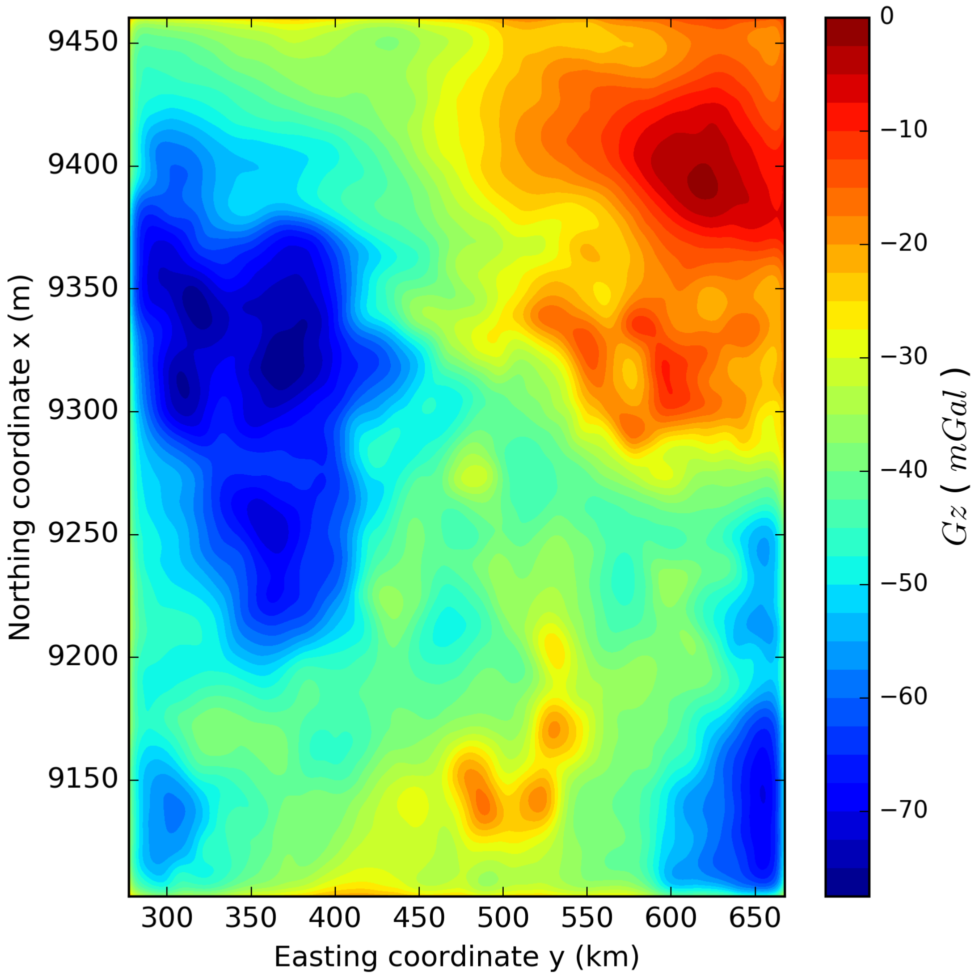

Figure 13: upward continuation processing at $2000m$ over the real data of Carajás. It shows a reasonable processing, atenuating the short-wave lenghts. The processing of the 250000 observations points took 0.216 seconds.

Tables

$N \times N$	**Full RAM (Mb)**	**BTTB RAM (Mb)**	**BCCB RAM (Mb)**
100×100	0.0763	0.0000763	0.0006104
400×400	1.22	0.0031	0.0248
2500×2500	48	0.0191	0.1528
10000×10000	763	0.00763	0.6104
40000×40000	12207	0.305	2.4416
250000×250000	476837	1.907	15.3
500000×500000	1907349	3.815	30.518
1000000×1000000	7629395	7.629	61.035

Table 1: Comparison between the system memory RAM usage needed to store the full matrix, the BTTB first row and the BCCB eigenvalues (eight times the BTTB). The quantities were computed for different numbers of data (N) with the same corresponding number of equivalent sources (N). This table considers that each element of the matrix is a double-precision number, which requires 8 bytes of storage, except for the BCCB complex eigenvalues, which requires 16 bytes per element.

ACKNOWLEDGMENTS

This study was financed by the brazilian agencies CAPES (in the form of a scholarship), FAPERJ (grant n.° E-26 202.729/2018) and CNPq (grant n.° 308945/2017-4).

REFERENCES

Barnes, G., and J. Lumley, 2011, Processing gravity gradient data: GEOPHYSICS, **76**, I33–I47.

Boggs, D., and M. Dransfield, 2004, Analysis of errors in gravity derived from the falcon airborne gravity gradiometer: ASEG-PESA Airborne Gravity 2004 Workshop, Geoscience Australia Record, 135–141.

Brigham, E. O., and E. O. Brigham, 1988, The fast fourier transform and its applications: prentice Hall Englewood Cliffs, NJ, **448**.

Chan, R. H., T. F. Chan, and C.-K. Wong, 1999, Cosine transform based preconditioners for total variation deblurring: IEEE transactions on Image Processing, **8**, 1472–1478.

Chan, R. H.-F., and X.-Q. Jin, 2007, An introduction to iterative toeplitz solvers: SIAM, **5**.

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

Dampney, C., 1969, The equivalent source technique: Geophysics, **34**, 39–53.

Emilia, D. A., 1973, Equivalent sources used as an analytic base for processing total magnetic field profiles: GEOPHYSICS, **38**, 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.

Grenander, U., 1984, Szeg o g.(1958), toeplitz forms and their applications: California Monographs in Mathematical Sciences. University of California Press, Berkeley, CA.

Guspí, F., and I. Novara, 2009, Reduction to the pole and transformations of scattered magnetic data using newtonian equivalent sources: GEOPHYSICS, **74**, L67–L73.

Hansen, R. O., and Y. Miyazaki, 1984, Continuation of potential fields between arbitrary surfaces: GEOPHYSICS, **49**, 787–795.

Heiskanen, W. A., and H. Moritz, 1967, Physical geodesy: Bulletin Géodésique (1946-1975), **86**, 491–492.

Henderson, R. G., 1960, A comprehensive system of automatic computation in magnetic and gravity interpretation: Geophysics, **25**, 569–585.

———, 1970, On the validity of the use of the upward continuation integral for total magnetic intensity data: Geophysics, **35**, 916–919.

Jin, X.-Q., 2003, Developments and applications of block toeplitz iterative solvers: Springer Science & Business Media, **2**.

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**, 497–507.

Levinson, N., 1946, The wiener (root mean square) error criterion in filter design and prediction: Journal of Mathematics and Physics, **25**, 261–278.

Li, Y., and D. W. Oldenburg, 2010, Rapid construction of equivalent sources using wavelets: GEOPHYSICS, **75**, L51–L59.

Lin, F., X. Jin, and S. Lei, 2003, Strang-type preconditioners for solving linear systems from delay differential equations: BIT Numerical Mathematics, **43**, 139–152.

Mendonça, C. A., and J. B. C. Silva, 1994, The equivalent data concept applied to the interpolation of potential field data: GEOPHYSICS, **59**, 722–732.

Oliveira Jr., V. C., V. C. F. Barbosa, and L. Uieda, 2013, Polynomial equivalent layer: GEOPHYSICS, **78**, G1–G13.

Olkin, J. A., 1986, Linear and nonlinear deconvolution problems: PhD thesis, Rice University.

Silva, J. B. C., 1986, Reduction to the pole as an inverse problem and its application to low-latitude anomalies: GEOPHYSICS, **51**, 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**, G57–G69.

Strang, G., and K. Aarikka, 1986, Introduction to applied mathematics: Wellesley-Cambridge Press Wellesley, MA, **16**.

Trench, W. F., 1964, An algorithm for the inversion of finite toeplitz matrices: Journal of the Society for Industrial and Applied Mathematics, **12**, 515–522.

Wray, J., and G. G. Green, 1994, Calculation of the volterra kernels of non-linear dynamic systems using an artificial neural network: Biological Cybernetics, **71**, 187–195.

Zhang, Y., and Y. S. Wong, 2015, Bttb-based numerical schemes for three-dimensional gravity field inversion: Geophysical Journal International, **203**, 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.