Submission code: RZM0LGPM4A

See this and other abstracts on our website: https://home.sbgf.org.br/Pages/resumos.php

Mesh refinement in the 3D inversion of potential-field data through the planting method

Gabriel Silva (University of São Paulo), Leonardo Uieda (IAG | USP)

Mesh refinement in the 3D inversion of potential-field data through the planting method

Abstract

Geophysical inversions are essential for interpreting geological structures through indirect observations, like potential field data. The planting method is a computationally efficient inversion algorithm that grows a solution iteratively from “seed” elements within a discretized 3D mesh of rectangular prisms. The original software implementation has been updated and the method has been expanded to include a new mesh refinement strategy. Current results demonstrate that the mesh refinement is successful at recovering complex geometries from a single starting seed. The influence of mesh size on the inversion accuracy and the calibration of critical parameters (e.g., regularization and normalization factors) require further investigation.

Introduction

The potential-field methods in geophysics are widely used for imaging the Earth’s crustal structure at different scales. To image the data sources, it’s needed to infer the physical properties (density or magnetization) that explain the data, and this is a inverse problem in geophysics. The planting method, introduced by Uieda and Barbosa (2012a,b), is an inversion method that consists on an algorithm that grows a solution around “seeds”, defined as retangular prisms with known properties, within a 3D discretized subsurface mesh of rectangular prisms. The growth process is controlled by a data misfit function, a compactness regularization function (Uieda and Barbosa, 2012b), and the shape-of-anomaly function of René (1986). The method stands out because of its low computational cost since it only requires a relatively small number of forward modeling operations and avoids large linear algebra systems.

The method was introduced 12 years ago therefore its original code required updates from Python 2 to Python 3. Additionally, some key questions remained:

1. Can coarse-mesh inversion results, which can be produced quickly and are more stable, help guide inversions on a finer mesh?
2. How does the mesh geometry (e.g., flattened or elongated prisms) influence the solution?

We have updated the code to Python 3 and using modern scientific Python libraries. We have also added a mesh refinement process and investigated the use of a skeletonization method to extract seeds from an existing solution produced on a coarser mesh.

Method

We define the vector $\mathbf{g}$ containing L observations of gravity or magnetic anomalies. The subsurface is discretized into a mesh of M juxtaposed rectangular prisms, each with a homogeneous physical property distribution. Thus, $\mathbf{g}$ can be approximated by the sum of the effects from each prism. In matrix notation:

$$\mathbf{d} = \mathbf{A}\mathbf{p} \tag{1}$$

where $\mathbf{d}$ is the L-dimensional vector of predicted data, $\mathbf{p}$ is the M-dimensional parameters vector whose j-th element contains the properties of the j-th prism, and $\mathbf{A}$ is the $L \times M$ sensibility matrix whose elements represent the gravitational or magnetic effects of each prism with unit physical property, calculated using the equations from Nagy et al. (2000).

The misfit between $\mathbf{g}$ and $\mathbf{d}$ can be expressed through the normalized Euclidean norm function

$$\phi(\mathbf{p}) = \sqrt{\frac{\sum_{i=1}^{L}(g_i - d_i)^2}{\sum_{i=1}^{L}(g_i)^2}}. \tag{2}$$

Another data misfit function is the "shape of anomaly" from René (1986). It measures the shape difference of the predicted and observed data, not considering amplitude differences

$$\psi(\mathbf{p}) = \sqrt{\sum_{i=1}^{L}(\alpha g_i - d_i)^2}, \tag{3}$$

in which α is the scale difference between the two vector. If both data have the same shape, their difference will be α. Therefore, a great value for α is the one which minimizes $\psi(\mathbf{p})$, and is obtained by differentiating ψ with respect to α and setting the derivative equal to zero. Doing so and isolating α, we have

$$\alpha = \frac{\sum_{i=1}^{L} g_i d_i}{\sum_{i=1}^{L}(g_i)^2}. \tag{4}$$

The solution to the inverse problem is constructed by progressively adding prisms from the discretized 3D model around a "seed" a prism with known physical properties. Initially, all elements of the parameter vector $\mathbf{p}$ are set to zero. During the growing process, the $\mathbf{p}$ elements corresponding to the seeds are assigned their respective physical property values. Each iteration consists of testing the addition of a prism in the neighborhood of every seed within the current solution, where the prism's $\mathbf{p}$ element will be assigned a non-zero physical property value. For a prism accretion, the neighbor prism must satisfy the following criteria:

1. The prism accretion must satisfy the following condition:

$$\frac{|\phi_n - \phi_a|}{\phi_a} \geq \delta, \tag{5}$$

   in which ϕ_n is the value of ϕ (Eq. 2) with the neighbor accretion, ϕ_a is the current value of ϕ and δ a small positive value that controls how much the solution is allowed to grow.

2. Among all the prisms that satisfy the previous condition, the best one for accretion will be the prism which produces the smallest value of the goal function (Uieda and Barbosa, 2012b)

$$\Gamma = \psi(\mathbf{p}) + \mu\theta, \tag{6}$$

   where ψ is the shape of anomaly function (Eq. 3), θ is a regularization parameter and θ is the compactness regularization function. From (Uieda and Barbosa, 2012b),

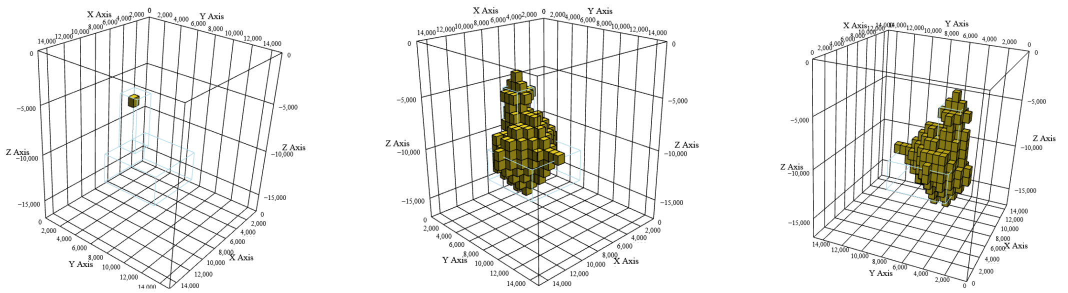
$$\theta = \frac{1}{f}\sum_{j=1}^{M}\frac{p_j}{p_j + \epsilon} l_j, \tag{7}$$

   where p_j ($j = 1, \ldots, M$) is the j-th parameter, $\epsilon > 0$ prevents division by zero, l_j is the prism-seed distance, and f is the mesh's average dimension.

If no neighboring prism satisfies the criteria above, the seed does not grow in that iteration. The process continues until no seed can grow further. The method's computational efficiency stems from the fact that the columns of the sensitivity matrix $\mathbf{A}$ only include contributions from neighboring prisms rather than all prisms in the mesh. Once a prism is added to the solution, its corresponding column can be discarded.

Results

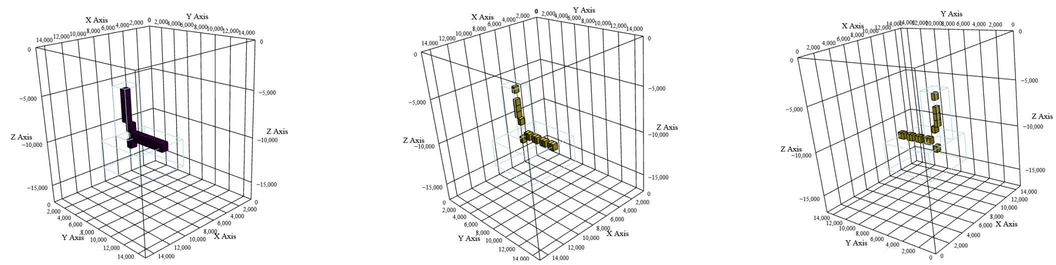
Figure 1 shows a single seed within a coarse mesh and the resulting inversion solution, viewed from two perspectives. The blue wireframe represents the synthetic geological model, interpreted as the intersection of a dike with a sill.



(a) Seed. (b) Solution. (c) Solution.

Figure 1: Single seed and its solution from two viewing angles, within a mesh comprising 20 × 26 × 20 rectangular prisms (each 800 × 600 × 900 m in dimension) spanning the domain x = [0, 15,000 m], y = [0, 15,000 m], and z = [-17,000, 0 m]. The synthetic model contains two primary geological bodies defined by their vertex coordinates: the first body occupies x = [5,000, 11,000 m], y = [4,000, 10,000 m], z = [-13,000, -10,000 m], while the second, shallower body is located at x = [7,000, 9,000 m], y = [4,000, 6,000 m], z = [-10,000, -4,500 m].

For mesh refinement, we applied the method of Lee et al. (1994) to extract a seed "skeleton" from the initial solution (Figure 1). This skeleton was converted into new seeds within a higher-resolution mesh (containing more prisms in each dimension than the original mesh), and the inversion process was repeated. Figure 2 displays both the skeleton from the coarse mesh and the seed prisms in the refined mesh, while Figure 3 presents the inversion solution obtained using this refined mesh.

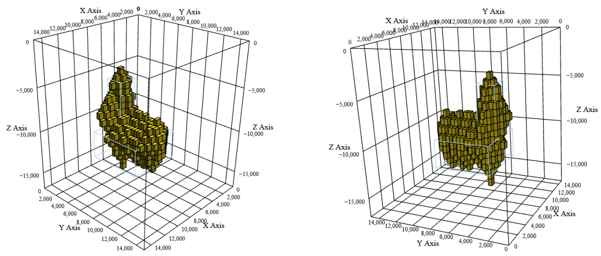


(a) Skeleton. (b) Seeds. (c) Seeds.

Figure 2: Skeleton of coarse mesh solution and the seeds converted in a new mesh with 26 x 39 x 26 rectangular prisms (each 600 × 400 × 700 m in dimension).

Conclusions

The original Python 2 code was updated with new functionalities, including skeletonization (Lee et al., 1994), mesh refinement, and variable prism dimensions, alongside modern Python libraries like Xarray (Hoyer and Hamman, 2017) and PyVista (Sullivan and Kaszynski, 2021). While the method has



(a) Solution. (b) Solution.

Figure 3: Solution in a new mesh with 26 $\times$ 39 $\times$ 26 rectangular prisms (each 600 $\times$ 400 $\times$ 700 m in dimension).

proven efficient for inversion, limitations remain: optimal values for parameters such as regularization and threshold are unclear, and the mesh behavior with varying prism resolutions per dimension requires further investigation. These questions will be addressed through analysis of results from diverse inversions and geological models.

References

Hoyer, S., and J. J. Hamman, 2017, Xarray: N-D labeled arrays and datasets in python: Journal of Open Research Software, **5**, 10.

Lee, T., R. Kashyap, and C. Chu, 1994, Building skeleton models via 3-d medial surface axis thinning algorithms: CVGIP: Graphical Models and Image Processing, **56**, 462–478.

Nagy, D., G. Papp, and J. Benedek, 2000, The gravitational potential and its derivatives for the prism: Journal of Geodesy, **74**, 552–560.

René, R. M., 1986, Gravity inversion using open, reject, and ”shape-of-anomaly” fill criteria: Geophysics, **51**, 988–994.

Sullivan, C. B., and A. Kaszynski, 2021, PyVista: 3d visualization and mesh analysis in python: https://www.pyvista.org. (Version 0.33.3).

Uieda, L., and V. C. F. Barbosa, 2012a, Robust 3d gravity gradient inversion by planting anomalous densities: GEOPHYSICS, **77**, G55–G66.

———, 2012b, Use of the ”shape-of-anomaly” data misfit in 3d inversion by planting anomalous densities: Presented at the SEG Technical Program Expanded Abstracts 2012, Society of Exploration Geophysicists.