

A MATLAB-Based Numerical and GUI Implementation of Cross-Gradients Joint Inversion of Gravity and Magnetic Data

Junjie Zhou^{1,2*}, Xingdong Zhang^{1,2}, Chunxiao Xiu^{1,2}

¹Key Laboratory of Geo-Detection (China University of Geosciences, Beijing), Ministry of Education, Beijing, China

²School of Geophysics and Information Technology, China University of Geosciences, Beijing, China
Email: *zjjs_195@163.com

Received 25 January 2015; accepted 13 February 2015; published 15 February 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The cross-gradients joint inversion technique has been applied to multiple geophysical data with a significant improvement on compatibility, but its numerical implementation for practical use is rarely discussed in the literature. We present a MATLAB-based three-dimensional cross-gradients joint inversion program with application to gravity and magnetic data. The input and output information was examined with care to create a rational, independent design of a graphical user interface (GUI) and computing kernel. For 3D visualization and data file operations, UBC-GIF tools are invoked using a series of I/O functions. Some key issues regarding the iterative joint inversion algorithm are also discussed: for instance, the forward difference of cross gradients, and matrix pseudo inverse computation. A synthetic example is employed to illustrate the whole process. Joint and separate inversions can be performed flexibly by switching the inversion mode. The resulting density model and susceptibility model demonstrate the correctness of the proposed program.

Keywords

Joint Inversion, Cross Gradients, Gravity and Magnetic Data, Numerical Implementation, Graphic User Interface

1. Introduction

The 3D joint inversion (*i.e.*, imaging) technique of multiple geophysical data often reveals a subsurface structure

*Corresponding author.

How to cite this paper: Zhou, J.J., Zhang, X.D. and Xiu, C.X. (2015) A MATLAB-Based Numerical and GUI Implementation of Cross-Gradients Joint Inversion of Gravity and Magnetic Data. *Journal of Software Engineering and Applications*, 8, 93-101. <http://dx.doi.org/10.4236/jsea.2015.82010>

more accurately than an individual inversion of single dataset. The joint inversion process relies on a certain linkage among various models, for which several methodologies have been proposed [1]-[7]. One conventional scheme takes sample property measurements when available, and determines their statistical interrelationships to restrict the inversion process. Unlike this, structural link methods are widely used to perform joint inversion without requiring rock property data. The cross-gradients technique is one such method, and its structural similarity criterion has been shown to give good results [8]. First proposed by Gallardo and Meju [9], the method was based on the assumption that, although petrophysical correlation varies, implicitly common boundaries exist. The method has been successfully applied to both synthetic and real data, but the relevant numerical implementation for practical use has rarely been discussed in previous work.

We now present a simple and general numerical implementation of 3D cross-gradients joint inversion framework for gravity and magnetic data in the MATLAB programming environment, noted for its powerful scientific computing capability (especially for matrix solution) and its graphical utilities. First, the joint inversion theory is briefly reviewed, and the separate design of a graphical user interface (GUI) and computing kernel is analyzed. Second, some key issues concerning numerical implementation are discussed (e.g., forward difference of cross-gradients, and matrix inverse computation). A synthetic joint inversion experiment was carried out to demonstrate the basic use of the proposed program. A comparative example is presented to indicate the practicality of the program.

2. General Framework of Cross Gradients Joint Inversion

2.1. Inputs and Outputs

Generally, an inversion converts acquired geophysical data into a physical property model, which is then assigned geological information to explain the nature of the subsurface material or its geological structure. Cross-gradients joint inversion uses two or more types of data to reproduce physical models. This is often found to be more accurate than dealing with each data type separately because of the constraints offered by the additional structural information. **Figure 1** shows the basic input and output of a cross-gradients joint inversion of gravity and magnetic data. This data is obviously necessary for the inversion to be carried out, but other information—reference models, boundary constraints and inversion parameters needed to control the procedure—is optional. The standard output comprises 3D density and susceptibility distribution and their corresponding predictions.

The input and output (I/O) dataset is usually stored as observational data, configurations and 3D model files. A commonly adopted approach to organizing such datasets is to use the University of British Columbia Geophysical Inversion Facility (UBC-GIF) format (ASCII encoding) [10]. In this format, observational data is arranged in columns along with observation locations and estimated errors. The models are ordered and sorted into columns with the associated mesh configuration file. To handle the frequent use of the I/O operations of these file formats, several MATLAB functions are written (listed in **Table 1**). These import the relevant datasets that are then processed utilizing a computing kernel module (described below). UBC-GIF tools are again invoked by these functions to export files and 3D graphics of the resultant models and predictions.

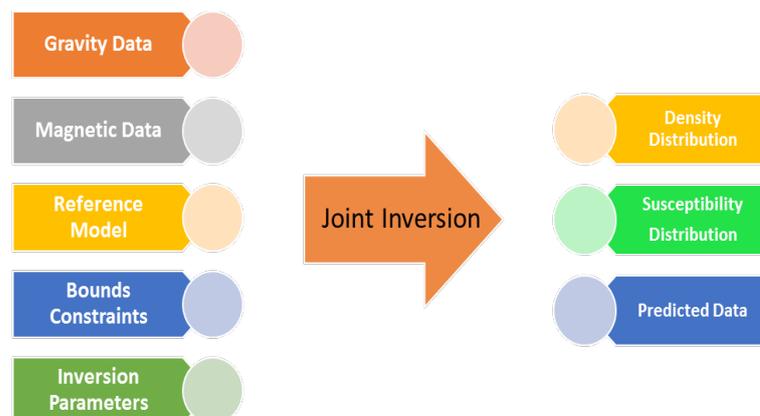


Figure 1. Schematic diagram of the input and output items associated with a joint inversion.

Table 1. I/O functions list for UBC-GIF format file.

Function name	Instruction
ubc_read_grv	read *.grv data file
ubc_save_grv	save *.grv data file
ubc_read_mag	read *.mag data file
ubc_save_mag	save *.mag data file
ubc_read_modl	read *.den or *.sus model file
ubc_save_modl	save *.den or *.sus model file
ubc_read_msh	read *.msh mesh file
ubc_save_msh	save *.msh mesh file
ubc_view_modl	use UBC-GIF tools to visualize 3D model
ubc_view_data	use UBC-GIF tools to visualize 2D data

a. For more information about UBC-GIF tools, please refer to [10].

2.2. Graphical User Interface Design

Cross-gradients joint inversion involves different kinds of observed data, different physical models and various choices of inversion parameters, all of which may cause confusion to users and make the program difficult for them. Some inversion parameters, such as regularization and the structural coupled factor, are usually chosen by trial and error; however, the results may be distorted if incorrect parameters are used, necessitating a data experiment before the inversion is performed. A GUI has been designed to encourage users to perform easy inversions and data experiments without confusion [11].

Figure 2 shows the main interface of this program. The files and their corresponding parameters need to be completed in the editing box controls on the panel. Four graphical box controls display inversion data during processing (blank while initializing). All the available operations are arranged as menu bars above the panel.

The inversion parameters and input file directions are recorded by a customized project file that can be either loaded or saved using the first menu item *Project*. The *View* menu is used to visualize the observational data or the 3D models using UBC-GIF tools, which produce 3D visuals of higher quality than the built-in MATLAB functions. The *Inversion* menu contains two modes—joint inversion, and separate inversion without structural constraints. The *Tests* menu helps the user to carry out data experiments and assists in the selection of correct inversion parameters such as depth weighting and regularization. Finally, the *Help* menu is linked to a user's manual and an *About* dialogue box. There is also a status bar displaying the local time and current state of the program.

2.3. Compute-Intensive Module Design

The computation kernel module has been designed to be separate from the GUI for its strict requirement of computing resources, including the CPU loadings and memory allocations. Following Fregoso and Gallardo [8], the inversion equation may be expressed with some slight modification as:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{N}_k^{-1} \mathbf{n}_k + \mathbf{N}_k^{-1} \mathbf{B}_k^T \mathbf{t}_k, \quad (1)$$

where

$$\begin{aligned} (\mathbf{B}_k \mathbf{N}_k^{-1} \mathbf{B}_k^T) \mathbf{t}_k &= \mathbf{B}_k \mathbf{N}_k^{-1} \mathbf{n}_k - \boldsymbol{\tau}_k \\ \mathbf{N}_k &= \mathbf{P}_k^T \mathbf{G}^T \mathbf{C}_d^{-1} \mathbf{G} \mathbf{P}_k + \mathbf{L}^T \mathbf{C}_L^{-1} \mathbf{L} + \mathbf{Z} \mathbf{C}_p^{-1} \mathbf{Z} \\ \mathbf{n}_k &= \mathbf{P}_k^T \mathbf{G}^T \mathbf{C}_d^{-1} [\tilde{\mathbf{g}}(\mathbf{p}_k) - \mathbf{d}] + \mathbf{L}^T \mathbf{C}_L^{-1} \mathbf{L} \mathbf{p}_k + \mathbf{C}_p^{-1} \mathbf{Z} (\mathbf{p}_k - \mathbf{p}_0) \end{aligned} \quad (2)$$

Here subscript k denotes the iteration number; 0 is *a priori* information; \mathbf{d} is a data vector; \mathbf{p} is a model vector transformed from property model m (in combined manner); matrix \mathbf{Z} is a weighting matrix; \mathbf{L} is a Laplacian matrix; \mathbf{C} is a covariance matrix with respect to subscript d , L and p , which indicate data, smoothness and smallness terms; $\boldsymbol{\tau}$ is a column vector of assembled cross-gradients components in 3D space; \mathbf{B} is a Jacobian matrix of cross-gradients; and \mathbf{P} is a Jacobian matrix of transform function. The final models \mathbf{m}_1 and \mathbf{m}_2 are extracted from \mathbf{p} by the inverse transform function when the iteration procedure satisfies the given data and struc-

tural misfit thresholds. For more details of Equation (1), please refer to [8]. The parameters (Table 2) to be assigned in the GUI partly constitute the arrays described above. The cross-gradients joint inversion procedure for this iterative scheme is illustrated in Figure 3.

There are several key considerations when numerically implementing the iterative Equation (1). First, the gradient operator involved in $\boldsymbol{\tau}$, \mathbf{L} and \mathbf{B} must be calculated in a 3D difference scheme, preferably by using forward differences to eliminate the chessboard pattern [12]. The discretization of τ_x , τ_y and τ_z are, from [8].

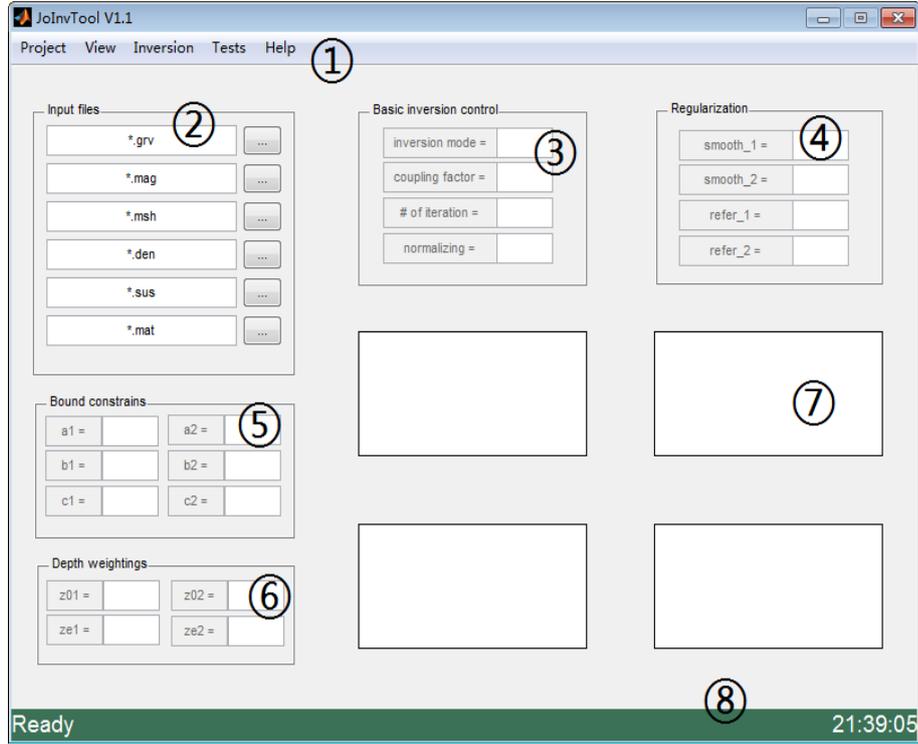


Figure 2. The main window of the cross gradients joint inversion program. 1) Menu bar; 2) Input files textboxes; 3) Basic control parameter textboxes; 4) Regularization textboxes; 5) Bound constrains textboxes; 6) Depth (Distance) weightings parameter textboxes; 7) Data display area; 8) Status bar.

Table 2. Control parameters of the inversion.

Types	Parameter	Instruction
string	gravity data file	(required) *.grv file
string	magnetic data file	(required) *.mag file
string	mesh file	(required) *.msh file
string	density reference model	(optional) *.den file
string	magnetization reference model	(optional) *.sus file
string	sensitivity file	(optional) *.mat file
real	physical property bounds	(optional) lower and upper bounds
real	depth weighting parameters	(required) used to compute \mathbf{Z}
string	inversion mode	(required) joint or separate mode
real	structural coupling factor	(optional) a positive number
integer	number of iteration	(required) the maximum # of iteration
real	smoothness factors	(required) regularization parameters
real	smallness factors	(required) regularization parameters

$$\begin{aligned} \tau_x &= \left[(m_{1y} - m_{1c})m_{2z} + (m_{1z} - m_{1y})m_{2c} + (m_{1c} - m_{1z})m_{2y} \right] / [dydz] \\ \tau_y &= \left[(m_{1z} - m_{1c})m_{2x} + (m_{1x} - m_{1z})m_{2c} + (m_{1c} - m_{1x})m_{2z} \right] / [dxdz], \\ \tau_z &= \left[(m_{1x} - m_{1c})m_{2y} + (m_{1y} - m_{1x})m_{2c} + (m_{1c} - m_{1y})m_{2x} \right] / [dxdy] \end{aligned} \quad (3)$$

where subscripts x, y, z denote the cell adjacent to the center in each direction, and c is the center cell (Figure 4). The exact value of \mathbf{B} is difficult to compute directly, but by discretizing τ , an approximate value is obtained by a Taylor’s expansion of the cross gradients at the zero point and neglecting the higher orders. For most cases, τ takes a value close to zero, so the approximate solution is always sufficiently precise (the reader is referred to [8] [9] for more details). Note that, since the computation for each element is independent, it is suggested that the code be vectorized in order to utilize the full potential of MATLAB’s matrix computing capability [13] [14]. A built-in function “sparse” is employed to perform efficient assembling of these matrices [15].

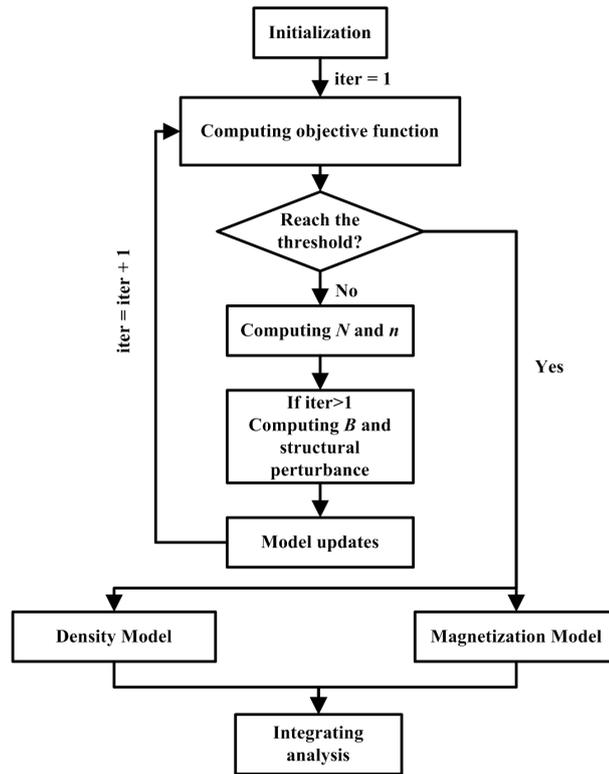


Figure 3. Flowchart of the joint inversion process.

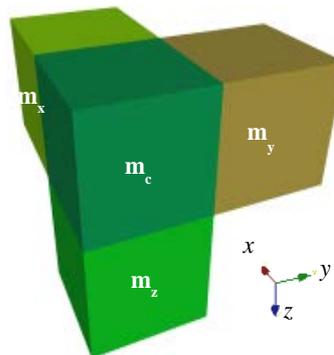


Figure 4. Schematic of forward difference for gradient computations involved in cross-gradients and smoothness operators.

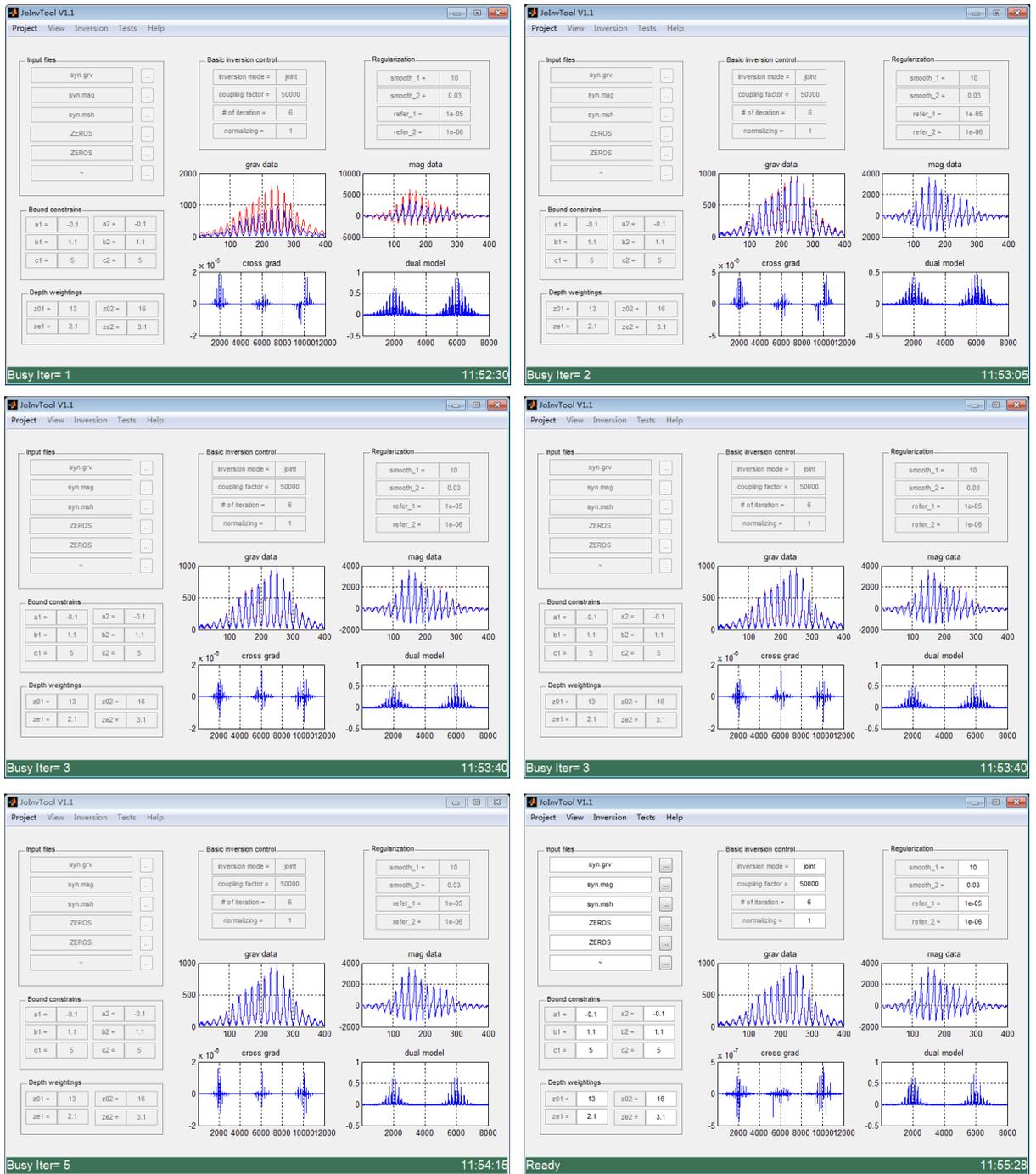


Figure 5. Iterative implementation of cross-gradients joint inversion. Observed data (red lines), current predicted data (blue lines), model updates and cross-gradients array (also blue lines in the following graphs) are displayed at each iteration. The textboxes are locked (grey) during the computation process.

Second, matrix N is inverted by a standard inverse algorithm when its condition number is not too large; fortunately, matrix C_p plays a regulator role to eliminate the ill-conditioned problem [16]. It can generally be solved by the standard inverse solver “inv” with high precision, but the inverse of the composed matrix $[BN^{-1}B^T]$ is usually difficult to compute because the production of several of the involved matrices makes it singular. In previous work, truncated singular values decomposition (SVD) has been used to calculate the pseudo inverse [8]; however, here we prefer to resolve the problem using a damped least-squares technique in which a damping

factor is introduced:

$$\mathbf{B}_k \mathbf{N}_k^{-1} \mathbf{B}_k^T \approx \mathbf{B}_k \mathbf{N}_k^{-1} \mathbf{B}_k^T + \frac{\max(\mathbf{B}_k \mathbf{N}_k^{-1} \mathbf{B}_k^T)}{\beta} \mathbf{I}. \quad (4)$$

where \mathbf{I} is the identity matrix, and β is a structural coupling factor. The second term on the right-hand side of Equation (4) in effect makes the total matrix well-posed. Taken together with the composed array $[\mathbf{B} \mathbf{N}^{-1} \mathbf{n} - \mathbf{r}]$ on the right-hand side, a linear system emerges which may be solved using the built-in CG solver “bicg” [15].

3. Synthetic Example

A synthetic experiment was carried out to illustrate the basic use of this program. The whole standard iterative process is shown in Figure 5. The textboxes are locked until the process is complete. The gravity and magnetic data (Figure 6) are generated on a synthetic geological model (Figure 7), and with 2.5% Gaussian noise added. The model consisted of two anomalies, A and B, with density contrast 0.5 g/cm^3 , 1 g/cm^3 and susceptibility 1 (SI unit, similarly hereinafter) and 0.5, respectively. The mesh comprised $20 \times 20 \times 10$ regular prisms of 50 m side length. All of the observed and geometry data were collected into the relevant files. The depth weighting parameters were chosen by UBC-GIF tools [10], and the regularization parameters were selected by trial and error aided by L-curve optimization [16].

When the process was complete, the View menu invoked UBC-GIF tools to visualize the resulting models and data shown in Figure 6 and Figure 7. It is seen that the inverted density model and susceptibility model reproduced the data very well, indicating the accuracy of the fitting feature of this process. When compared to the true density and susceptibility models, the recovered values reflected the relative amplitude and locations correctly.

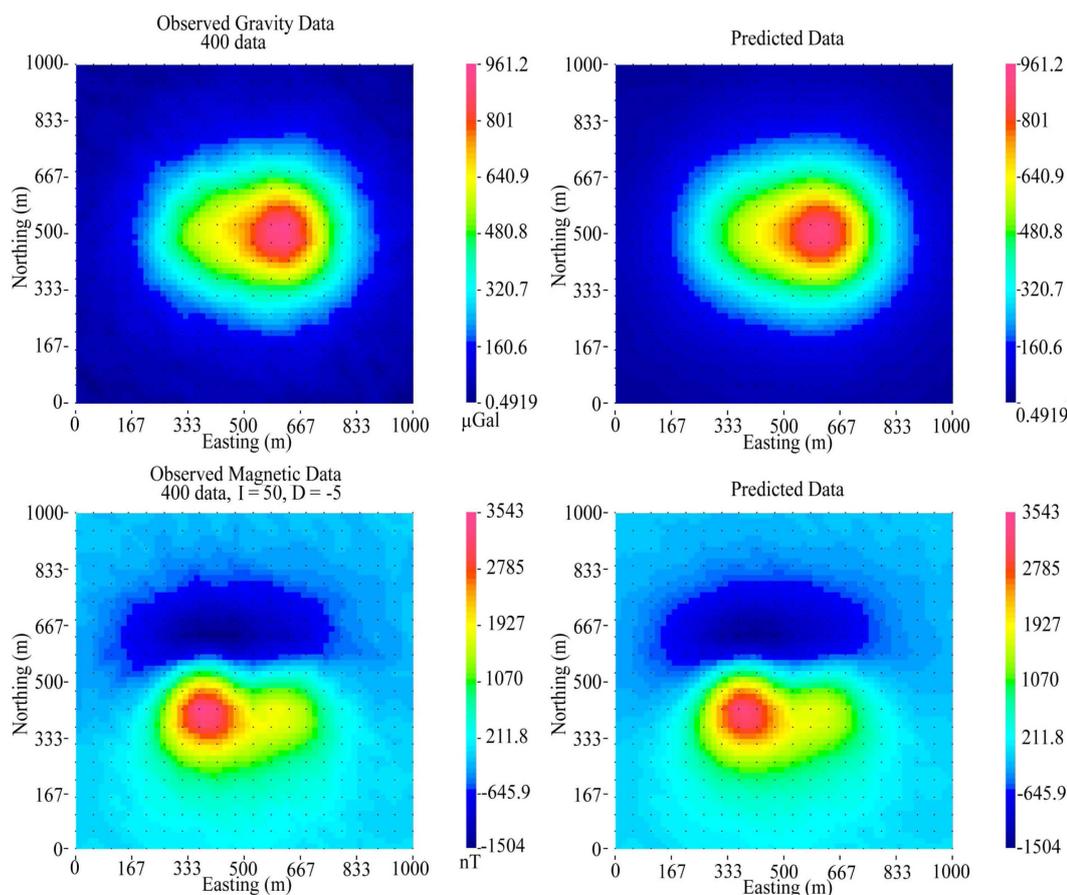


Figure 6. Synthetic gravity (μGal) and magnetic data (nT) and predicted joint inversion using standard mode.

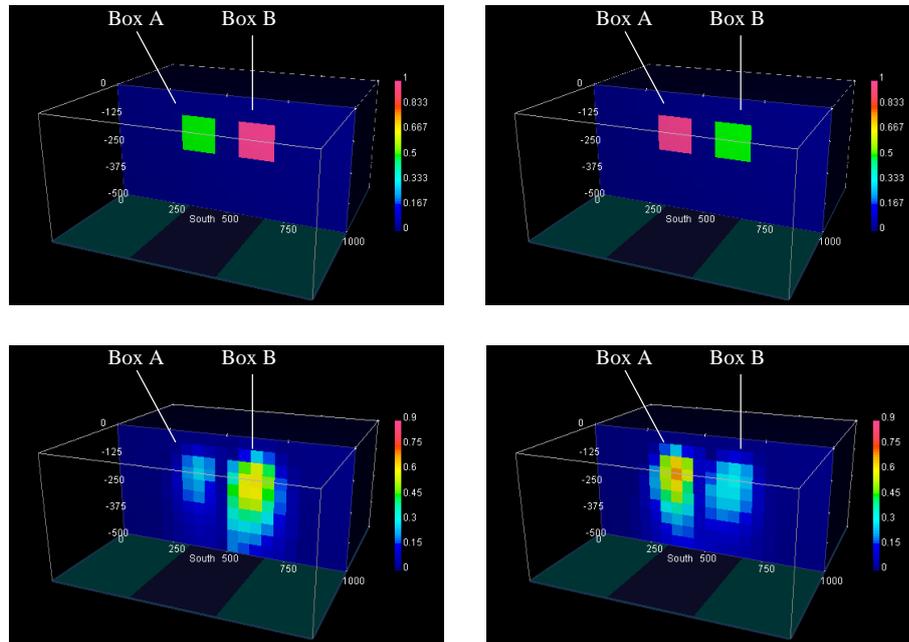


Figure 7. Synthetic density model (left) and susceptibility model (right) in a 3D subsurface and their corresponding joint inversion results using standard mode. The displayed profile is located at $x = 500$ m.

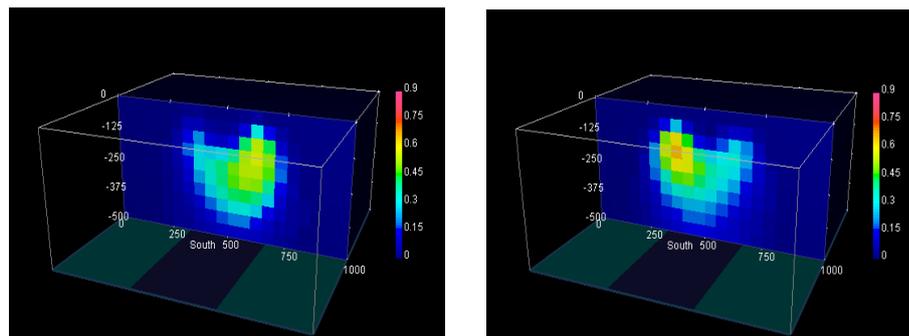


Figure 8. Separate inversion results of density model (left) and susceptibility model (right) in 3D subsurface.

To demonstrate the significant improvement produced by joint inversion, we also conducted a comparative experiment by changing the inversion mode parameter from *joint* to *separate*. The recovered models are shown in **Figure 8**, which show that the anomalies were recovered in the correct location but with low resolution. Note that the boundary between the two anomalies is blurred, making it hard to identify the two independent targets in the separate images.

4. Conclusion

A MATLAB-based numerical implementation of cross-gradients joint inversion of gravity and magnetic data is presented. A user-friendly GUI has also been designed for flexible handling of multiple observations, models and inversion control parameters. The inversion process is carried out using an independent module. Some important issues are discussed, such as cross-gradients computing and matrix inverting. The use of this program has been illustrated by a simple example using synthetic data. The correctness and accessibility of this program have been demonstrated. Although this program is available for both research and practical use, we suggest porting this MATLAB-based program to other professional software platforms (e.g., ArcGIS) for more integrating and convenient applications in the future work.

Acknowledgements

This work was funded by National Natural Science Foundation of China, Beijing Higher Education Young Elite Teacher Project (YETP0650), The National High Technology Research and Development Program (“863” Program) of China (No. 2013AA063901-4 and 2013AA063905-4), R & D of Key Instruments and Technologies for Deep Resources Prospecting (The National R&D Projects for Key Scientific Instruments) (No. ZDYZ2012-1-02-04), Constrained multi-parameter inversion of geophysical technology and software systems (No. 2014AA06 A613), and The Fundamental Research Funds for the Central Universities. We thank the UBC-GIF and Roman Shekhtman, the developer of Meshtools3D and gm-data-viewer, for making the programs available to the scientific community.

References

- [1] Zeyen, H. and Pous, J. (1993) 3-D Joint Inversion of Magnetic and Gravimetric Data with *a Priori* Information. *Geophysical Journal International*, **112**, 244-256. <http://dx.doi.org/10.1111/j.1365-246X.1993.tb01452.x>
- [2] Gallardo, L.A., Perez-Flores, M.A. and Gomez-Trevino, E. (2005) Refinement of Three-Dimensional Multilayer Models of Basins and Crustal Environments by Inversion of Gravity and Magnetic Data. *Tectonophysics*, **397**, 37-54. <http://dx.doi.org/10.1016/j.tecto.2004.10.010>
- [3] Bosch, M., Meza, R. and Jimenez, R. (2006) Joint Gravity and Magnetic Inversion in 3D Using Monte Carlo Methods. *Geophysics*, **71**, G163-G156. <http://dx.doi.org/10.1190/1.2209952>
- [4] Pilkington, M. (2006) Joint Inversion of Gravity and Magnetic Data for Two-Layer Models. *Geophysics*, **71**, L35-L42. <http://dx.doi.org/10.1190/1.2194514>
- [5] Shamsipour, P., Marcotte, D. and Chouteau, M. (2012) 3D Stochastic Joint Inversion of Gravity and Magnetic Data. *Journal of Applied Geophysics*, **79**, 27-37. <http://dx.doi.org/10.1016/j.jappgeo.2011.12.012>
- [6] Zhdanov, M.S., Gribenko, A. and Wilson, G. (2012) Generalized Joint Inversion of Multimodal Geophysical Data Using Gramian Constraints. *Geophysical Research Letters*, **39**, L09301. <http://dx.doi.org/10.1029/2012GL051233>
- [7] Williams, N.C. (2008) Geologically-Constrained UBC-GIF Gravity and Magnetic Inversions with Examples from the Agnew-Wiluna Greenstone Belt, Western Australia. Ph. D. Dissertation. The University of British Columbia, Vancouver.
- [8] Frogoso, E. and Gallardo, L.A. (2009) Cross-Gradients Joint 3D Inversion with Applications to Gravity and Magnetic Data. *Geophysics*, **74**, L31-L42. <http://dx.doi.org/10.1190/1.3119263>
- [9] Gallardo, L.A. and Meju, M.A. (2004) Joint Two-Dimensional DC Resistivity and Seismic Travel Time Inversion with Cross-Gradients Constraints. *Journal of Geophysical Research: Solid Earth* (1978-2012), **109**, B03311. <http://dx.doi.org/10.1029/2003JB002716>
- [10] Oldenburg, D.W. and Jones, F.H.M. (2007) Inversion for Applied Geophysics: Learning Resources about Geophysical Inversion. The University of British Columbia, Vancouver. <http://www.eos.ubc.ca/ubcgif/iag/index.htm>
- [11] Özgü Arısoy, M. and Dikmen, Ü. (2011) Potensoft: MATLAB-Based Software for Potential Field Data Processing, Modeling and Mapping. *Computers & Geosciences*, **37**, 935-942. <http://dx.doi.org/10.1016/j.cageo.2011.02.008>
- [12] Lelièvre, P.G. and Oldenburg, D.W. (2009) A Comprehensive Study of Including Structural Orientation Information in Geophysical Inversions. *Geophysical Journal International*, **178**, 623-637. <http://dx.doi.org/10.1111/j.1365-246X.2009.04188.x>
- [13] Hahn, B.H. and Valentine, D.T. (2013) Essential Matlab for Engineers and Scientists. 5th Edition, Academic Press, Boston, 129-160. <http://dx.doi.org/10.1016/B978-0-12-394398-9.00006-X>
- [14] Eshagh, M. and Abdollahzadeh, M. (2012) Software for Generating Gravity Gradients Using a Geopotential Model Based on an Irregular Semivectorization Algorithm. *Computers & Geosciences*, **39**, 152-160. <http://dx.doi.org/10.1016/j.cageo.2011.06.003>
- [15] Pidlisecky, A., Haber, E. and Knight R. (2007) RESINVM3D: A 3D Resistivity Inversion Package. *Geophysics*, **72**, H1-H10. <http://dx.doi.org/10.1190/1.2402499>
- [16] Hansen, P.C. (1994) Regularization Tools: A Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems. *Numerical Algorithms*, **6**, 1-35. <http://dx.doi.org/10.1007/BF02149761>

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or **Online Submission Portal**.

