Modified Componentwise Gradient Method for Solving Structural Magnetic Inverse Problem

Elena N. Akimova[1,2(✉)], Vladimir E. Misilov[1,2], and Andrey I. Tretyakov[1,2]

[1] Krasovskii Institute of Mathematics and Mechanics, Ural Branch of RAS, Yekaterinburg, Russia
aen15@yandex.ru, out.mrscreg@gmail.com, fr1z2rt@gmail.com
[2] Yeltsin Ural Federal University, Yekaterinburg, Russia

Abstract. An original variant of the componentwise gradient method is constructed to solve a nonlinear magnetic inverse problem: using magnetic data, find a boundary surface between two layers with constant arbitrarily directed magnetizations. An efficient parallel algorithm is created and implemented on a multicore CPU and multiple GPUs to solve the problem. We study the efficiency and speedup of the parallel algorithm. We solve various model problems with synthetic magnetic data on a fine grid. A comparison of the proposed method with the conjugate gradient method shows that the new one allows for a significant reduction of computation time.

Keywords: Componentwise gradient method · Parallel algorithms
Magnetic inverse problem · Multicore CPU and multiple GPUs

1 Introduction

The solution of structural gravity problems and magnetic inverse problems has an extraordinary importance in the study of the Earth's crust structure [1–3].

This paper deals with the problem of finding an interface between layers with different magnetizations using known magnetization contrast, interface depth, and magnetic field [4,5].

The problem is described by a nonlinear integral equation of the first kind and thus is ill-posed. It is therefore necessary to use iterative regularization methods [6].

Real observations are performed on large areas. To increase the accuracy and the level of detail, it is essential to use finer grids, which leads to big data sets. The application of modern computing technologies and parallel computations makes it possible to significantly reduce computation time.

This work was partly supported by the Ural Branch of the Russian Academy of Sciences (project no. 18-1-1-8).

L. Sokolinsky and M. Zymbler (Eds.): PCT 2018, CCIS 910, pp. 162–173, 2018.
https://doi.org/10.1007/978-3-319-99673-8_12

An effective method to determine the structural boundary in the case of arbitrarily directed magnetization was constructed in [7,8] on the basis of the linearized conjugate gradient method.

A time-efficient componentwise gradient method for solving gravity inverse problems was constructed in [9]. In the present paper, we use this method to solve the magnetic inverse problem of finding a magnetization interface in the case of an arbitrarily directed magnetization. Here, we modify the method for better performance. The modification consists in offsetting the indices of the components with respect to the angle of the magnetization vector.

Moreover, we construct a parallel algorithm based on the modified componentwise method and implement this parallel algorithm using the Intel CPUs and NVIDIA Tesla GPUs of the Uran supercomputer, which is installed at the Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences. We also investigate the efficiency and speedup of the parallel algorithm and compare it with a conjugate gradient-based algorithm in terms of iteration number and computation time.

2 Problem Statement

Let us introduce a cartesian coordinate system in which the $x0y$ plane coincides with the Earth's surface and the z axis is directed downwards, as shown in Fig. 1. Assume that the lower half-space consists of two layers with constant magnetizations J_1 and J_2, divided by the surface sought, which is described by a bounded function $\zeta = \zeta(x, y)$, and $\lim\limits_{|x|+|y|\to\infty} (h - \zeta(x, y)) = 0$ for some h.

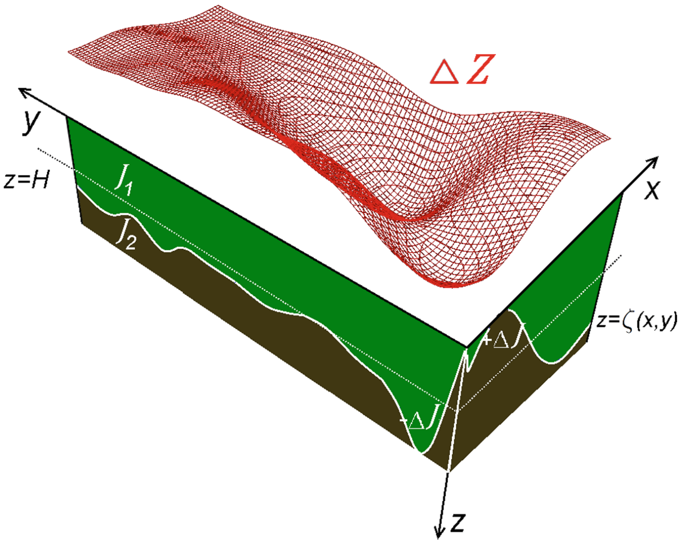

Fig. 1. Two-layer medium for the magnetic problem

The function ζ must satisfy the following equation:

$$\Delta Z(x',y',0) = \frac{1}{4\pi}\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty}\left[\frac{\Delta J_x(x-x') + \Delta J_y(y-y') - \Delta J_z h}{\big((x'-x)^2 + (y'-y)^2 + h^2\big)^{3/2}}\right.$$
$$\left. - \frac{\Delta J_x(x-x') + \Delta J_y(y-y') - \Delta J_z \zeta(x,y)}{\big((x'-x)^2 + (y'-y)^2 + \zeta^2(x,y)\big)^{3/2}}\right] dx\, dy, \qquad (1)$$

where $\Delta J_x, \Delta J_y, \Delta J_z$ are the components of the magnetization contrast $\Delta J = J_2 - J_1$, and $\Delta Z(x,y,0)$ is the vertical component of the anomalous magnetic field measured at the Earth's surface.

A preliminary processing of data with the aim of extracting the anomalous field from the measured magnetic data is performed using a technique described and implemented in [10].

Equation (1) is a nonlinear two-dimensional integral equation of the first kind.

After discretization of the region $\Pi = \{(x,y) : a \leqslant x \leqslant b, c \leqslant y \leqslant d\}$ by means of an $n = M \times N$ grid and approximation of the integral operator using quadrature rules, we obtain a vector F on the right-hand side and an approximation of the solution vector z of dimension n. Equation (1) can be thus written as

$$\Delta F_i = \frac{\Delta x \Delta y}{4\pi}\sum_{j=1..n}\left[\frac{\Delta J_x(x_i - x_j) + \Delta J_y(y_i - y_j) - \Delta J_z h}{\big((x_i - x_j)^2 + (y_i - y_j)^2 + h^2\big)^{3/2}}\right.$$
$$\left. - \frac{\Delta J_x(x_i - x_j) + \Delta J_y(y_i - y_j) - \Delta J_z z_j}{\big((x_i - x_j)^2 + (y_i - y_j)^2 + z_j^2\big)^{3/2}}\right], \qquad (2)$$

We can rewrite the equation as

$$A(z) = F. \qquad (2a)$$

3 Numerical Methods for the Solution of the Problem

3.1 Linearized Conjugate Gradient Method

The *linearized conjugate gradient method* (LCGM) has the following form [11]:

$$\begin{aligned}
z^{k+1} &= z^k - \psi \frac{\langle p^k, S(z^k)\rangle}{\|A'(z^k)p^k\|^2} p^k,\\
p^k &= S(z^k) + \beta^k p^{k-1},\\
p^0 &= S(z^0),\\
\beta^k &= \max\left\{\frac{\langle S(z^k), \big(S(z^k) - S(z^{k-1})\big)\rangle}{\|S(z^{k-1})\|^2}, 0\right\},\\
S(z) &= A'(z)^T\big(A(z) - F\big),
\end{aligned} \qquad (3)$$

where z^k is the approximation of the solution in the kth iteration, $k \in \mathbb{N}$, and ψ is a damping factor.

A parallel algorithm based on this method was developed and implemented in [8] for NVIDIA GPUs using CUDA technology.

3.2 Componentwise Gradient Method

The *componentwise gradient method* (CWM) has the following form [9]:

$$z_i^{k+1} = z_i^k - \psi \frac{A_i(z^k) - F_i}{\|\nabla A_i(z^k)\|^2} \left(\frac{\partial A_i(z^k)}{\partial z_i} \right), \tag{4}$$

where z_i is the ith component of the solution approximation, $i = 1, \ldots, n$, $k \in \mathbb{N}$, and ψ is a damping factor.

The main idea of this method is to minimize the residual $A_i(z) - F_i$ at one grid node i by changing the value z_i at this node. The idea is based on the fact that the value of a gravity or magnetic (in the case of vertically directed magnetization) field depends on $1/r^2$. Thus, the value of z_i exerts the greatest influence on the field value F_i at the node directly situated above it. In the case of an arbitrarily directed magnetization, the correlation between z_i and F_i is weaker, so this method will not be as effective as it is for vertical magnetization.

3.3 Modified Componentwise Gradient Method

Let us find the approximation of a new point j at which F_j is mostly influenced by z_i in the case of an arbitrarily directed magnetization. This point is displaced from the point i by the biases $\bar{x}$ and $\bar{y}$. To find $\bar{x}$, we need to solve the following problem:

$$\bar{x} = \arg\max_x \left[- \frac{\varDelta J_x(x) - \varDelta J_z h}{\left(x^2 + h^2\right)^{3/2}} \right].$$

The necessary condition for maximum is

$$\frac{d}{dx}\left[- \frac{\varDelta J_x(x) - \varDelta J_z h}{\left(x^2 + h^2\right)^{3/2}} \right] = 0.$$

Write the derivative:

$$-\frac{\varDelta J_x(2x^2 - h^2) + 3\varDelta J_z h x}{\left(x^2 + h^2\right)^{5/2}} = 0.$$

Evidently, $x \neq 0$ for the case of nonvertical magnetization and the surface lies below the Earth's level, *i.e.* $h > 0$, so that

$$\varDelta J_x(2x^2 - h^2) + 3\varDelta J_z h x = 0.$$

Write the roots of this equation:

$$\bar{x}_{1,2} = \frac{\big(-3\varDelta J_z \pm \sqrt{8\varDelta J_x^2 + 9\varDelta J_z^2}\big)h}{4\varDelta J_x}.$$

Assume that $\varDelta J_z > 0$. Then, obviously, the relation $\operatorname{sgn}(\varDelta J_x) = \operatorname{sgn}(\bar{x})$ must hold. Only the first root (the one with the plus sign) satisfies this condition. For $\varDelta J_z < 0$, we have the second root (the one with the minus sign).

The $\bar{y}$ bias can be found in the same way. We can now write the *modified componentwise gradient method* (MCWM) as follows:

$$\begin{gathered} z_i^{k+1} = z_i^k - \psi \frac{A_j(z^k) - F_j}{\|\nabla A_j(z^k)\|^2} \left(\frac{\partial A_j(z^k)}{\partial z_i} \right), \\ j = i + M \frac{\big(-3\varDelta J_z + \operatorname{sgn}(\varDelta J_z)\sqrt{8\varDelta J_y^2 + 9\varDelta J_z^2}\big)h}{4\varDelta J_y \varDelta y} \\ + \frac{\big(-3\varDelta J_z + \operatorname{sgn}(\varDelta J_z)\sqrt{8\varDelta J_x^2 + 9\varDelta J_z^2}\big)h}{4\varDelta J_x \varDelta x}, \end{gathered} \tag{5}$$

where $\varDelta x$ and $\varDelta y$ are the grid element sizes.

We should also check whether the offsetted indices are out of the grid. If so, we should use the boundary values.

4 Parallel Implementation

The parallel algorithms based on the componentwise methods were implemented on a multicore CPU, using OpenMP technology, and NVIDIA M2090 GPUs, using CUDA technology.

Note that storing a Jacobian matrix for a $2^9 \times 2^9$ grid takes more than 512 GB.

The elements of the Jacobian matrix in the constructed algorithms are calculated on-the-fly, which means that the value of an element is computed when calling this element, without storing it previously in memory.

The most expensive operation is to compute the values of the integral operator and its Jacobian matrix. This operation consists of four nested loops. In the OpenMP implementation, the outer loops are parallelized using '#pragma omp parallel', whereas the inner loops are vectorized using '#pragma simd' directives. When using multiple GPUs, two outer loops are distributed to the GPUs, and two inner loops are executed on each GPU. The CPU transfers the data between the host memory and GPUs, and then calls the kernel functions.

The adjustment of the kernel execution parameters for the grid size is an important problem. In [12], we proposed an original method for automatic adjustment of parameters. This method is based on rescaling the optimal parameters found for a reference grid size.

This imposes some constraints on the input data and GPUs configuration:

- the grid size must be divisible by 128 $(128, 256, 512, 1024, \ldots)$;
- the number of GPUs must be a power of 2 $(1, 2, 4, 8, \ldots)$.

5 Numerical Experiments

The model problems consisted in finding the interface between two layers. Figure 2 shows the model surface z^* considered in all model problems.

Figures 3, 4, 5, 6 and 7 show the model magnetic fields $\Delta Z_i(x, y, 0)$. These fields were obtained by solving the direct problem for the surface with the asymptotic plane $H = 10$ km and various magnetization contrasts:

$$\begin{aligned}\Delta J_1 &= (0, 0, 1)\ \mathrm{A/m},\\ \Delta J_2 &= (0.19, 0.19, 1)\ \mathrm{A/m},\\ \Delta J_3 &= (0.41, 0.41, 1)\ \mathrm{A/m},\\ \Delta J_4 &= (0.71, 0.71, 1)\ \mathrm{A/m},\\ \Delta J_5 &= (1.23, 1.23, 1)\ \mathrm{A/m}.\end{aligned}$$

These contrasts correspond to magnetization direction angles of 0°, 15°, 30°, 45°, and 60°.

The problems were solved on the Uran supercomputer nodes (two eight-core Intel E5-2660 CPUs and eight NVIDIA Tesla M2090 GPUs) by the following three methods:

- linearized conjugate gradient method (LCGM) (3);
- componentwise gradient method (CWM) (4);
- modified componentwise gradient method MCWM (5).

The reconstructed interfaces are shown in Fig. 8.

The condition $\|A(z) - F\|/\|F\| < \varepsilon$, $\varepsilon = 0.011$, was taken as termination criterion for all methods. The parameter ψ was set at 0.85 in the CGM for 60°, as well as in the CWM and MCWM for 45°. In the CWM and MCWM for 60°, it was set at 0.75. Everywhere else, it was set at 1.

The relative error of all solutions is $\delta = \|z - z^*\|/\|z^*\| < 0.01$.

Table 1 summarizes the numbers of iterations N and average execution times T for 10 runs on two eight-core Intel E5-2660 CPUs (16 cores) with a 512×512 grid.

Speedup and efficiency coefficients are used to analyse the scaling of parallel algorithms. The speedup is expressed as $S_m = T_1/T_m$, where T_1 is the execution time of a program running on one GPU, and T_m is the execution time for m GPUs. The efficiency is defined as $E_m = S_m/m$. The ideal values are $S_m = m$ and $E_m = 1$, but real values are lower because of the overhead.

Table 2 summarizes the average execution times for the CWM method on a 512×512 grid for various numbers of GPUs.

The experiments show that the constructed modified algorithms are very effective. New algorithms are more economical in terms of operations and time at each iteration step. For the model problems, the componentwise method has a better performance in terms of number of iterations and computation time than the conjugate gradient methods. The parallel algorithms demonstrate an excellent scaling; the efficiency is more than 100% for eight GPUs. Probably, this is due to a non-optimal automatic adjustment of the kernel execution parameters for some configurations of GPUs.

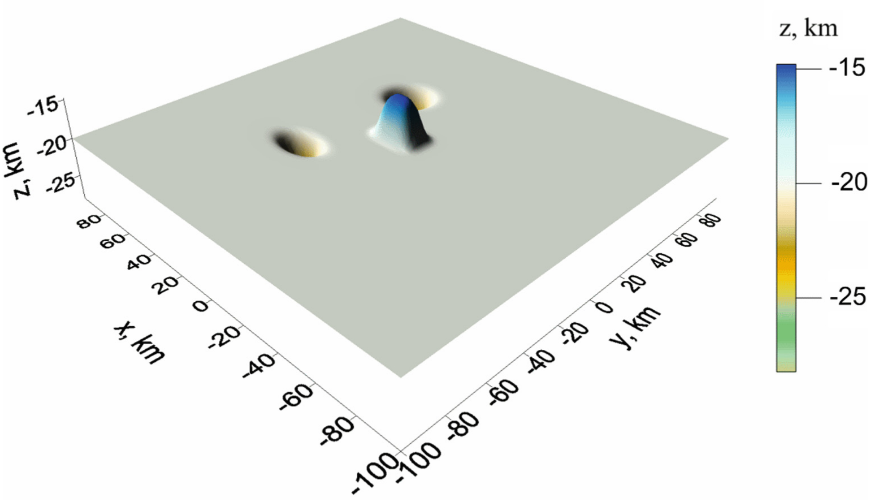

Fig. 2. The original surface z^*

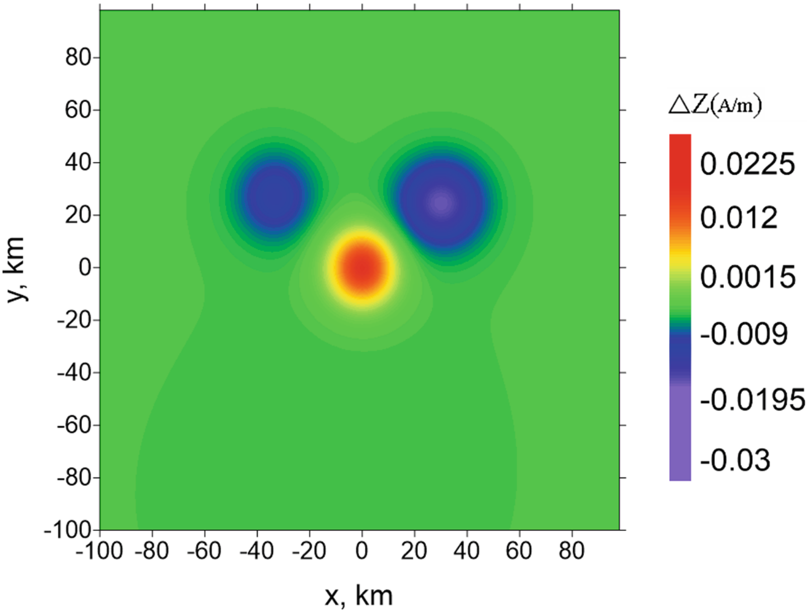

Fig. 3. Model gravitational field for an angle of 0°

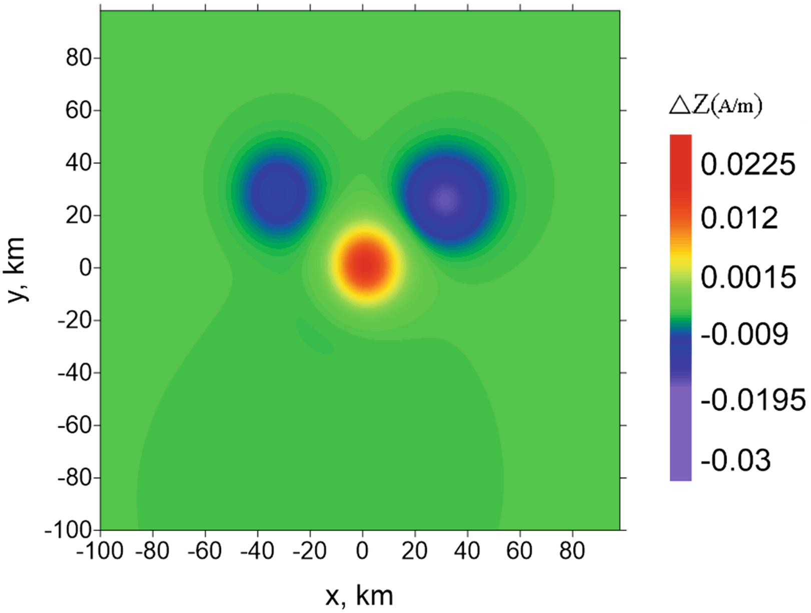

Fig. 4. Model gravitational field for an angle of 15°

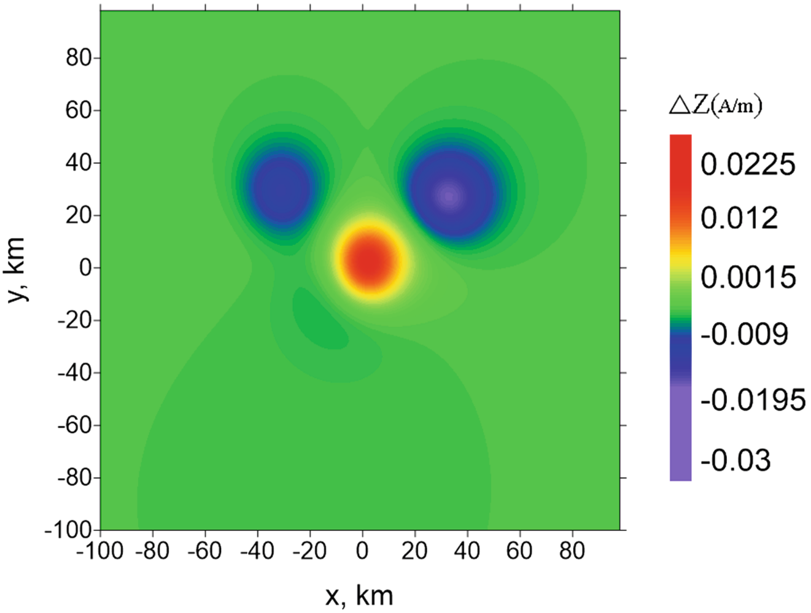

Fig. 5. Model gravitational field for an angle of 30°

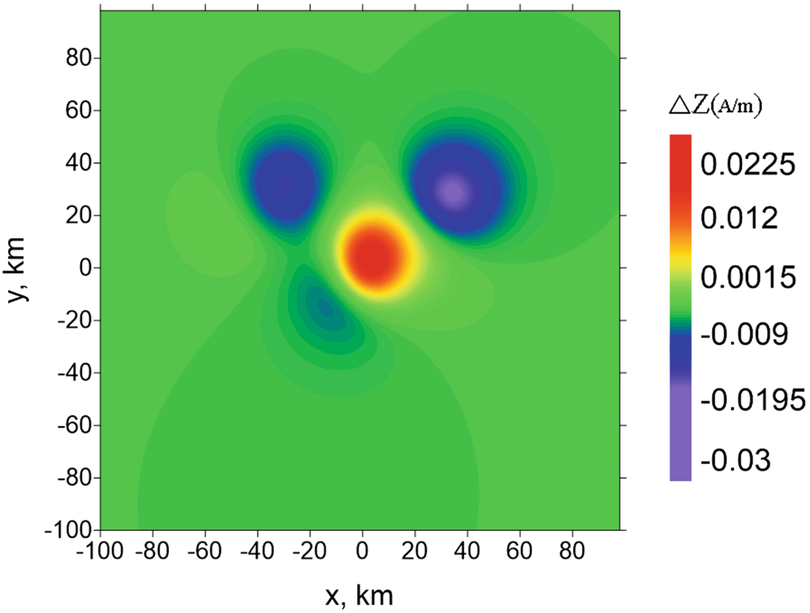

Fig. 6. Model gravitational field for an angle of 45°

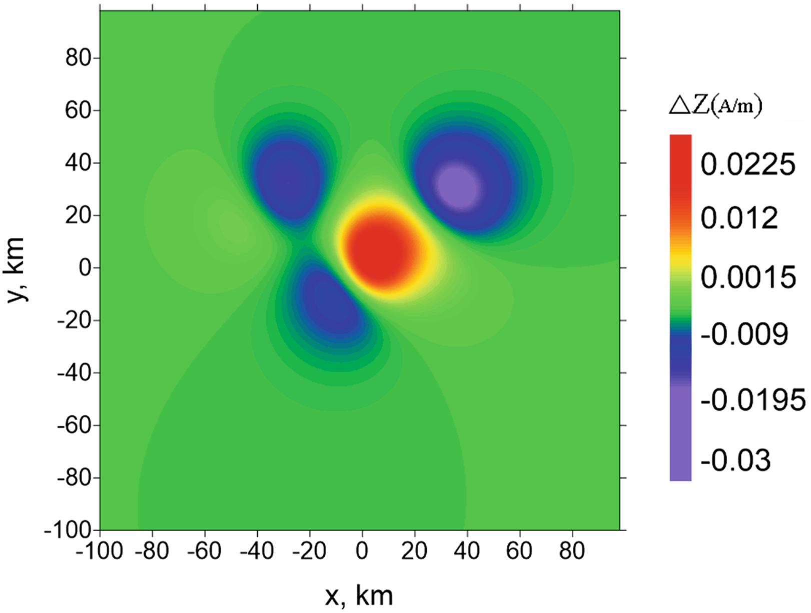

Fig. 7. Model gravitational field for an angle of 60°

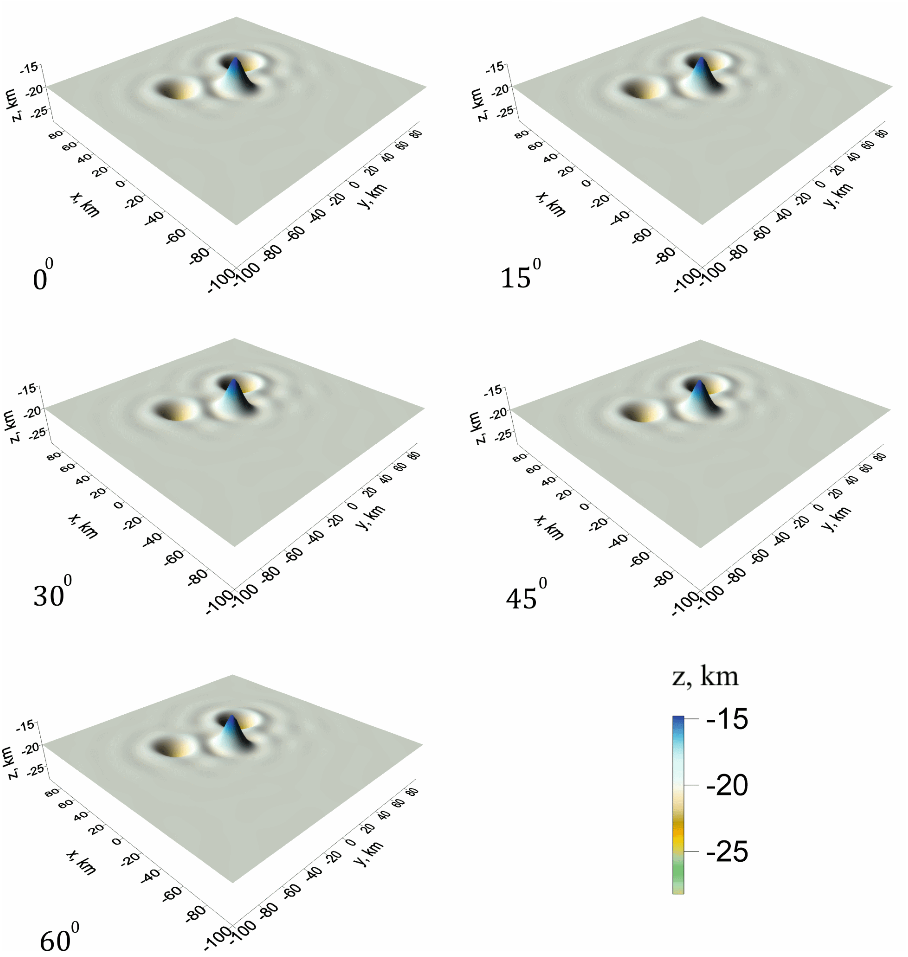

Fig. 8. Reconstructed surfaces for various magnetization angles

Table 1. Comparison of methods

Magnetization angle	CGM		CWM		MCWM	
	N	T (min)	N	T (min)	N	T (min)
0°	20	120	6	36	6	36
15°	20	120	6	36	6	36
30°	20	120	8	48	7	42
45°	25	150	10	60	9	54
60°	26	156	16	96	14	84

Table 2. Execution times (in minutes) of the parallel CWM algorithm on multiple GPUs

Magnetization angle	Number of GPUs			
	1	2	4	8
	Execution time T, minutes			
0°	7.6	2.8	1.3	0.7
15°	7.6	2.8	1.3	0.7
30°	8.3	3.3	1.5	0.9
45°	10.8	4.5	2.3	1.1
60°	17.7	7.9	4.2	2.1

6 Conclusions

We constructed an original variant of a componentwise gradient method for a structural magnetic inverse problem consisting in finding a contact surface in the case of an arbitrarily directed magnetization.

We developed parallel algorithms based on the componentwise gradient method and its modified variant. The parallel algorithms were implemented on a multicore CPU, using OpenMP technology, and on multiple GPUs, using CUDA technology. Model problems with fine grids were solved. The parallel algorithms demonstrated an excellent scaling and nearly 100% efficiency.

The componentwise gradient methods (CWM and MCWM) are very effective for solving problems with a nearly vertical magnetization direction; in this case, computation times are reduced by a factor of 2 to 4. For greater magnetization angles, the modified componentwise gradient method (MCWM) show better computation times compared to the unmodified componentwise method.

References

1. Martyshko, P.S., Byzov, D.D., Martyshko, M.P.: Solving the structural inverse problem of magnetic prospecting with respect to demagnetization for a two-layer medium model. Dokl. Earth Sci. **453**(2), 1264–1267 (2013). https://doi.org/10.1134/S1028334X1312012X
2. Akimova, E.N., Martyshko, P.S., Misilov, V.E.: Algorithms for solving the structural gravity problem in a multilayer medium. Dokl. Earth Sci. **453**(2), 1278–1281 (2013). https://doi.org/10.1134/S1028334X13120180
3. Martyshko, P.S., Pyankov, V.A., Akimova, E.N., Vasin, V.V., Misilov, V.E.: On solving a structural gravimetry problem on supercomputer "Uran" for the Bashkir Predural's area. In: GeoInformatics 2013 – 12th International Conference on Geoinformatics: Theoretical and Applied Aspects (2013)
4. Malkin, N.R.: On solution of inverse magnetic problem for one contact surface (the case of layered masses). DAN SSSR, Ser. A (9), 232–235 (1931)

5. Akimova, E.N., Martyshko, P.S., Misilov, V.E.: Parallel algorithms for solving structural inverse magnetometry problem on multicore and graphics processors. In: Proceedings of 14th International Multidisciplinary Scientific GeoConference SGEM 2014, vol. 1, no. 2, pp. 713–720 (2014)
6. Bakushinskiy, A., Goncharsky, A.: Ill-Posed Problems: Theory and Applications. Mathematics and Its Applications, 258 p., vol. 301. Springer Science & Business Media, Heidelberg (1994). https://doi.org/10.1007/978-94-011-1026-6
7. Misilov, V.E.: On solving the structural inverse magnetic problem of finding a contact surface in the case of arbitrary directed magnetization. In: 15th EAGE International Conference on Geoinformatics: Theoretical and Applied Aspects (2016)
8. Akimova, E.N., Martyshko, P.S., Misilov, V.E., Tretyakov, A.I.: On solving the inverse structural magnetic problem for large grids on GPUs. In: AIP Conference Proceedings, vol. 1863, p. 050010 (2017). https://doi.org/10.1063/1.4992207
9. Akimova, E.N., Misilov, V.E.: A fast componentwise gradient method for solving structural inverse gravity problem. In: Proceedings of 15th International Multidisciplinary Scientific GeoConference SGEM 2015, vol. 3, no. 1, pp. 775–782 (2015)
10. Martyshko, P.S., Fedorova, N.V., Akimova, E.N., Gemaidinov, D.V.: Studying the structural features of the lithospheric magnetic and gravity fields with the use of parallel algorithms. Izv. Phys. Solid Earth **50**(4), 508–513 (2014). https://doi.org/10.1134/S1069351314040090
11. Akimova, E.N., Martyshko, P.S. and Misilov, V.E.: A fast parallel gradient algorithm for solving structural inverse gravity problem. In: AIP Conference Proceedings, vol. 1648, p. 850063 (2015). https://doi.org/10.1063/1.4913118
12. Akimova, E.N., Misilov, V.E., Tretyakov, A.I.: Optimized algorithms for solving structural inverse gravimetry and magnetometry problems on GPUs. In: Sokolinsky, L., Zymbler, M. (eds.) PCT 2017. CCIS, vol. 753, pp. 144–155. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67035-5_11