UDC 551.241:528.026:550.38:519.61
Original scientific paper
Received: 22.12.2003.

Stable parallel algorithms for solving the inverse gravimetry and magnetometry problems

Elena N. Akimova and **Vladimir V. Vasin**
Institute of Mathematics and Mechanics, Ural Branch of Russian Academy of Sciences,
S. Kovalevskaya str.,16, Ekaterinburg, 620219, RUSSIA
e-mail: aen@imm.uran.ru, vasin@imm.uran.ru

SUMMARY

The three-dimensional inverse problems of gravimetry and magnetometry for finding the interfaces between mediums from the gravitational and magnetic data are investigated. We assume that a model of the lower half-space consists of three mediums with constant densities which are separated by the surfaces S_1 and S_2 to be determined.

The inverse problems are reduced to nonlinear integral equations of the first kind, hence these problems are ill-posed. After discretization of the integral equation we obtain a system of nonlinear equations of large dimension. To solve this system, we use the iteratively regularized Gauss-Newton method. To realize one step of this method, we have to solve a system of linear algebraic equations with full matrix. For this aim, parallel variants of the Gauss, Gauss-Jordan and the conjugate gradient method are applied.

Their realization has been implemented on the Massively Parallel Computing System MVS-1000. The analysis of the efficiency of parallelization of the iterative algorithms with different numbers of processors is carried out. Parallelization of the algorithms decreases significantly the time of solving the problems. The interfaces S_1 and S_2 obtained by the Gauss-Newton method correspond to the real geological perceptions about the Ural region under investigation.

Key words: *Parallel algorithms, gravimetry, magnetometry, parallelization.*

1. BASIC EQUATIONS AND PRELIMINARY DATA PROCESSING

We assume that the gravitational anomaly is formed by the deviation of the desired surface S from the horizontal plane $z = H$ (S_i ($i = 1, 2$), $H_1 = 2$, $H_2 = 10$ in our case).

Then, in the Descartes coordinate system, the gravity equation with respect to the unknown function $z = z(x, y)$, which describes the interface, is reduced to the two-dimensional nonlinear integral equation:

$$A[z] \equiv f\Delta\sigma \int_a^b \int_c^d \left\{ \frac{1}{\left[(x-x')^2 + (y-y')^2 + z^2(x',y')\right]^{1/2}} - \frac{1}{\left[(x-x')^2 + (y-y')^2 + H^2\right]^{1/2}} \right\} dx'\,dy' = F(x,y) \quad (1)$$

where f is the gravitation constant, $\Delta\sigma$ is the density jump on the interface and $F(x,y)$ is the anomalous gravitational field.

The magnetometry equation has the following form:

$$B[z] \equiv \Delta J \int_a^b \int_c^d \left\{ \frac{z(x',y')}{\left[(x-x')^2+(y-y')^2+z^2(x',y')\right]^{3/2}} - \frac{H}{\left[(x-x')^2+(y-y')^2+H^2\right]^{3/2}} \right\} dx'\,dy' = G(x,y) \quad (2)$$

where G is the anomalous magnetic field and ΔJ is the averaged jump of the component z of the magnetization vector.

To select the anomalous gravity and magnetic field, which serve for the right-hand sides of Eqs. (1) and (2), we imply the following technique (see Ref. [1]).

It is commonly accepted that the field recalculation upward to a level $z=+H$ practically eliminates the effect of anomaly-forming objects, located up to the depth $z=-H$. By geological evidence, the field sources foreign to our analysis are located more deeply than $z=-H$ ($H=2$ *km* or $H=10$ *km*). Therefore, the measured field was continued upward to the level $z=H$.

The stronger distortions in this procedure occur near the boundary of the domain, hence the integration is done over a finite area. To diminish these distortions, the values of a function that is the solution of the plane Dirichlet problem were preliminary subtracted from the measured field. In the investigated domain this function satisfies the two-dimensional Laplace equation and coincides with the given field on the boundary of the domain.

In our opinion, this function can be used as the field of the lateral sources. For recalculation upward, the Poisson formula for a subspace was used:

$$U(x,y,H) \equiv \frac{1}{2\pi} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{HU(x',y',0)\,dx'\,dy'}{\left[(x-x')^2+(y-y')^2+H^2\right]^{3/2}}$$

To get rid of the sources in the horizontal layer from the ground surface to the level $z=-H$, the field recalculated upward was then continued downward to the depth $z=-H$. In this case we solve the integral equation:

$$K\omega \equiv \frac{1}{2\pi} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{2H\omega(x',y')\,dx'\,dy'}{\left[(x-x')^2+(y-y')^2+(2H)^2\right]^{3/2}} = U(x,y,H)$$

to find the unknown function $\omega(x,y)=U(x,y,-H)$. Here the function $U(x,y,H)$ is given. In order to stably solve this equation we used the Lavrentiev regularization method of the form:

$$(K+\alpha I)\omega = U$$

for a suitable regularization parameter $\alpha>0$.

Since the singularity of the function obtained lies below the plane $z=-H$, this function can be interpreted as the field of deep sources.

The sum of this field recalculated on the ground surface and the solution of the Dirichlet problem was used as a field of lateral and deep sources (extrinsic sources). The difference of the measured field and the field of the extrinsic sources was used as the gravity effect (the function $F(x,y)$ in Eq. (1)) of sources located in the horizontal layer from the ground surface to the depth $z=-H$.

The similar procedure was used for selecting the anomalous magnetic field (the function $G(x,y)$ in Eq. (2)).

2. ITERATIVE METHOD FOR SOLVING THE NONLINEAR SYSTEM

After discretizing the Eqs. (1) and (2) on the grid $n = M \times N$ and approximating the integral operators A and B by the quadrature formulas, we obtain a system of nonlinear equations of the form:

$$\boldsymbol{A}_n[\boldsymbol{z}] = \boldsymbol{F}_n \quad (3)$$

For solving this system the iteratively regularized Newton method [2] is used:

$$z^{k+1} = z^k - [A'_n(z^k) + \alpha_k I]^{-1}(A_n(z^k) + \alpha_k z^k - F_n) \quad (4)$$

where $\boldsymbol{A}'_n(z^k)$ is the Jacobi matrix calculated at the point z^k, $\boldsymbol{I}$ is the identity operator, α_k is a sequence of the positive parameters and $\boldsymbol{F}_n$ is a vector approximation of the function $F(x,y)$ (or $G(x,y)$).

The iterative method of the Eq. (3) can be written in the following form:

$$\boldsymbol{A}^k\boldsymbol{z}^{k+1} = \boldsymbol{F}^k \quad (5)$$

where $\boldsymbol{A}^k = \boldsymbol{A}'_n(z^k) + \alpha_k\boldsymbol{I}$ is an $n\times n$ matrix, $\boldsymbol{F}^k = \boldsymbol{A}^k z^k - (\boldsymbol{A}_n(z^k) + \alpha_k z^k - \boldsymbol{F})$ is an n-dimensional vector.

So, for finding the next approximation z^{k+1} in Newton method, Eq. (4), it is necessary to solve the system of linear algebraic equations, Eq. (5), with full $n \times n$ matrix.

The careful analysis showed that, for a acceptable starting point z^0 and parameters α_k, the matrix $\boldsymbol{A}^k$ has n different eigenvalues for all realized iterations ($k = 1,2,\dots,5$).

This implies that the corresponding eigenvectors are linearly independent and the matrix $\boldsymbol{S}^k$ whose columns are eigenvectors has the inverse $(\boldsymbol{S}^k)^{-1}$. Hence, the matrix $\boldsymbol{A}^k$ can be represented in the form [3]:

$$\boldsymbol{A}^k = \boldsymbol{S}^k\boldsymbol{\Lambda}^k(\boldsymbol{S}^k)^{-1}$$

where $\boldsymbol{\Lambda}^k$ is the diagonal matrix. From this formula it follows that the matrix $\boldsymbol{A}^k$ ($k = 1, 2,\dots, 5$) is invertible.

Moreover, for problem described by Eq. (1) the condition number $cond(A^k)$ of the matrix $\boldsymbol{A}^k$ varies within the intervals $2.8 \le cond(A^k) \le 727$ for the interface S_1 and $1.8 \le cond(A^k) \le 2642$ for the interface S_2. For problem described by Eq. (2) the condition number varies within the intervals $1.1 \le cond(A^k) \le 405$ for the interface S_1 and $1.2 \le cond(A^k) \le 328$ for the interface S_2.

This guarantees the stability of calculations under the realization of the iterative process given by Eq. (4) and the monotone decrease of the Newton method errors $\Delta^k = \| z^k - z \|$.

The further analysis showed that, in fact, for method given by Eq. (4) the conditions of the Newton-Kantorovich theorem [4] at the iteration points z^k were fulfilled, which implies practical convergence of this method.

It should be noted that if an operator A is monotone, then for any $\alpha_k > 0$ the operator $[A'(z) + \alpha_k I]^{-1}$ exists and is bounded, and under some conditions method given by Eq. (4) converges to a solution of the equation $A(z) = F$ for $\alpha_k \to 0$, $k \to \infty$ (see Ref. [2]). It is known [5] that the gravimetry operator is not monotone, so the convergence theorem has not been proved.

3. GAUSS, GAUSS-JORDAN AND CONJUGATE GRADIENT METHODS. PARALLEL REALIZATION

So, for finding the next approximation z^{k+1} for each iteration of Newton method given by Eq. (4) it is necessary to solve Eq. (5). The problem given by Eq. (5) can be solved by the Gauss or Gauss-Jordan elimination algorithms or by the conjugate gradient method.

The main idea of the Gaussian elimination method is reducing the full matrix $\boldsymbol{A}$ of system described by Eq. (5) to the upper triangular form, that is, to obtain the system of equations in the following form:

$$\begin{cases} x_1 + c_{12}x_2 + c_{13}x_3 + \ldots + c_{1n}x_n = c_{1n+1} \\ \qquad x_2 + c_{23}x_3 + \ldots + c_{2n}x_n = c_{2n+1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad\qquad\qquad\qquad x_n = c_{nn+1} \end{cases} \qquad (6)$$

From system given by Eq. (6) we find the unknowns by the formulas:

$$x_k = c_{kn+1} - c_{kk+1}x_{k+1} - \ldots - c_{kn}x_n, \; k = n, n-1, \ldots, 1 \qquad (7)$$

The Gauss-Jordan algorithm is one of the variants of the Gaussian elimination algorithm. In this case the matrix $\boldsymbol{A}$ of system given by Eq. (5) is reduced to the diagonal form, but not an upper triangular form. In the $(k+1)$-*th* step the current matrix $\boldsymbol{A}^k$ has the following form:

$$\boldsymbol{A}^k = \begin{bmatrix} 1 & 0 & \ldots & 0 & a^{(k)}_{1,k+1} & \ldots & a^{(k)}_{1n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & a^{(k)}_{k,k+1} & \ldots & a^{(k)}_{kn} \\ 0 & 0 & \ldots & 0 & a^{(k)}_{k+1,k+1} & \ldots & a^{(k)}_{k+1,n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & a^{(k)}_{n,k+1} & \ldots & a^{(k)}_{nn} \end{bmatrix} \qquad (8)$$

We divide the $(k+1)$-*th* row by the coefficient $a^{(k)}_{k+1,k+1}$ and eliminate all off-diagonal elements of the $(k+1)$-*th* column. We make this elimination by the subtraction of the obtained $(k+1)$-*th* row multiplied by $a^{(k)}_{j,k+1}$ from the j-*th* row ($j = 1, 2, \ldots, n$; $j \neq k+1$).

To guarantee the numerical stability of the Gauss and Gauss-Jordan algorithms in the general case, a partial choice of the pivot element is necessary. If we take the maximum (with respect to modulus) element in the k-*th* row as the pivot element at the k-*th* step of the elimination, then, before realization of the step, it is necessary to rearrange the k-*th* column and the column with the pivot element.

The parallel realization of the Gauss or Gauss-Jordan method for m processors is the following. Conditionally, we divide the vectors z and F into m parts so that $n = m \cdot L$. The matrix $\boldsymbol{A}$ is divided by the horizontal lines into m blocks, respectively (Figure 1).

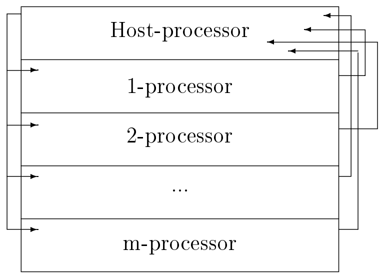

Fig. 1 The diagram of the data distribution over the processors

Assume that the rows of the matrix $\boldsymbol{A}$ with numbers $1, 2, \ldots, L$ are stored in the memory of the first processor (the Host), the rows with numbers $(L+1), (L+2), \ldots, 2L$ are stored in the memory of the second processor, and so on. The rows with numbers $(m-1)L+1, \ldots, Lm$ are stored in the memory of the m-*th* processor. The main idea of the parallel elimination is the following. At the every step each processor eliminates the unknowns from its own part of the L equations. The Host processor chooses the pivot element among the elements of the current row, modifies this row and sends it to each of the other processors [6].

During the Gauss elimination process, more and more processors become idle at every step, since the number of the equations is diminished by one. This influences the efficiency of the algorithm. In the Gauss-Jordan method all the processors make calculations with their own parts until the end. The waiting time decreases and the efficiency of the algorithm increases.

The conjugate gradient method is used for solving the problem with a symmetric matrix. For this we transform the system given by Eq. (5) to the following form:

$$(\boldsymbol{A}^k)^T \boldsymbol{A}^k z^{k+1} = (\boldsymbol{A}^k)^T \boldsymbol{F}^k \qquad (9)$$

where $(\boldsymbol{A}^k)^T$ is the conjugated matrix.

For an arbitrary initial approximation z^0, the first approximation z^1 is found from it by the steepest descent method:

$$z^1 = z^0 - \frac{\left\| r^0 \right\|^2 \cdot \left(Az^0 - F \right)}{\left(Ar^0, r^0 \right)}, \quad r^0 = Az^0 - F \qquad (10)$$

Further from z^0 and z^1 we find all successive approximations of the conjugate gradient method:

$$z^{k+1} = z^k - \alpha_k\left(Az^k - F\right) + \beta_k\left(z^k - z^{k-1}\right)$$
$$\alpha_k = \frac{\left\|r^k\right\|^2\left(Ap^k, p^k\right) - \left(r^k, p^k\right)\left(Ar^k, p^k\right)}{\left(Ar^k, r^k\right)\left(Ap^k, p^k\right) - \left(Ar^k, p^k\right)^2}, r^k = Az^k - F$$
$$\beta_k = \frac{\left\|r^k\right\|^2\left(Ar^k, p^k\right) - \left(r^k, p^k\right)\left(Ar^k, r^k\right)}{\left(Ar^k, r^k\right)\left(Ap^k, p^k\right) - \left(Ar^k, p^k\right)^2}, p^k = z^k - z^{k-1}$$
$$k = 0,1,2,... \tag{11}$$

The stopping condition of the iterative process given by Eqs. (10) and (11) is:

$$\frac{\left\|Az^k - F\right\|}{\left\|F\right\|} < \varepsilon$$

Parallelization of the iterative process given by Eqs. (10) and (11) for solving the problem given by Eq. (9) is based on the dividing of the matrices $(\boldsymbol{A}^k)^T$ and $\boldsymbol{A}^k$ by the horizontal lines into m blocks. The data distribution over the processors is similar to the data distribution in the Gauss method (Figure 1).

At every step of the conjugate gradient method, each processor calculates its own part of the solution vector z^k. In the case of the multiplication of the matrix by the vector, each processor multiplies its own part of rows of the matrix by the whole vector. In the case of the matrix product $(\boldsymbol{A}^k)^T\boldsymbol{A}^k$ each processor multiplies its own part of rows of the conjugated matrix $(\boldsymbol{A}^k)^T$ by the whole matrix $\boldsymbol{A}^k$.

4. EFFICIENCY OF THE METHODS

Parallelization of the basic algorithms and their realization on the Massively Parallel Computing System MVS-1000 [7] is implemented. The analysis of the efficiency of parallelization of the iterative algorithms with different numbers of processors is carried out.

MVS-1000/16 of the Research Institute KVANT production consists of 16 Intel Pentium III-800, 256 MByte, 10 GByte disk, two 100 Mbit network controllers (Digital DS21143 Tulip and Intel PRO/100). Educational computing cluster consists of 8 Intel Pentium III700, 128 MByte, 14 GByte disk, 100 Mbit network controller 3Com 3c905B Cyclone.

For comparison of the executing times of the sequential and parallel algorithms, we will consider the coefficients of the speed up and efficiency:

$$S_m = T_1 / T_m, \quad E_m = S_m / m$$

where T_m is the execution time of the parallel algorithm on MVS-1000 with m ($m>1$) processors, T_1 is the execution time of the sequential algorithm on one processor:

$$T_m = T_c + T_e + T_i$$

where T_c is the computing time, T_e is the exchange time and T_i is the idle time. The number m of processors corresponds to the mentioned division of the vectors z and F into m parts so that $n = m \cdot L$.

On the other hand, the efficiency can be calculated using only the parallel version of a program on a parallel computer without using the execution time of the sequential algorithm T_1. The efficiency can be defined as:

$$E = G/(G+1)$$

where G is the granularity of a parallel algorithm [8].

The granularity of a parallel algorithm is the ratio of the amount of computations to the amount of communications within a parallel algorithm implementation. Taking into account the possibility that the processors may be not equally balanced and the processor idle time can occur, then the granularity is calculated using the following expression:

$$G = T_c / (T_e + T_i)$$

The granularity may be estimated as:

$$G \le \frac{max\left(T_{comp}\right)\cdot m}{min\left(T_{comm}\right)\cdot m}$$

where $max(T_{comp})$ is the maximum computation time for one processor and $min(T_{comm})$ is the minimum communication time for one processor.

Table 1 and Table 2 show the execution times T_m and the coefficients of the speed up S_m and the efficiencies E_m and E obtained by using the granularity G of the iteratively regularized Newton method after 5 iterations using the parallel and sequential ($m=1$) Gauss and Gauss-Jordan algorithms for problems given by Eqs. (1) to (4) for $111{\times}35$ points of the grid domain.

Table 1 Gauss Method for the 111×35 grid

m	T_m, min.	S_m	E_m	E
1	57.48	-	-	-
2	46.85	1.23	0.61	0.66
3	36.18	1.59	0.53	0.59
4	29.38	1.96	0.49	0.56
5	25.78	2.23	0.45	0.53
6	21.83	2.63	0.44	0.52
8	17.25	3.33	0.42	0.49
10	14.17	4.06	0.41	0.48
12	12.35	4.65	0.39	0.44

Table 2 Gauss-Jordan Method for the 111×35 grid

m	T_m, min.	S_m	E_m	E
1	114.1	-	-	-
2	60.50	1.89	0.94	0.97
3	42.38	2.69	0.90	0.91
4	33.53	3.40	0.85	0.88
5	28.48	4.01	0.80	0.85
6	23.88	4.78	0.79	0.83
8	19.88	5.74	0.72	0.78
10	16.45	6.93	0.69	0.72
12	15.35	7.42	0.62	0.66

Table 3 Conjugate Gradient Method for the 111×35 grid

m	T_m, min.	S_m	E_m	E
1	84.38	-	-	-
2	43.20	1.95	0.98	0.99
4	22.75	3.71	0.93	0.95
5	18.63	4.52	0.90	0.93
10	10.37	8.14	0.81	0.84
11	9.67	8.79	0.80	0.82
17	7.03	12.0	0.71	0.75

Table 3 shows the execution times T_m and the coefficients of the speed up S_m and the efficiencies E_m and E obtained by using the granularity G of the iteratively regularized Newton method after 5 iterations using the parallel and sequential ($m=1$) conjugate gradient method for problems given by Eqs. (1) to (4) for 111×35 points of the grid domain.

The results of calculations show that the parallel Gauss and Gauss-Jordan algorithms have efficiency of parallelization high enough, and the Gauss-Jordan algorithm efficiency is higher. But the conjugate gradient method efficiency is higher than the efficiency of the Gauss and Gauss-Jordan algorithms. This fact can be explained by the small exchange time. The elements of the matrix $(\boldsymbol{A}^k)^T\boldsymbol{A}^k$ are formed independently in m processors. At every step of the conjugate gradient method, each processor calculates its own part of the solution vector $\boldsymbol{z}^k$.

In the case of the parallel Gauss algorithm with the number of processors $m<5$, the efficiency is $E_m \geq 0.45$. In the case of the parallel Gauss-Jordan algorithm with the number of processors $m \leq 5$, the efficiency is $E_m \geq 0.8$. In the case of the parallel conjugate gradient method with the number of processors $m \leq 5$, the efficiency is $E_m \geq 0.9$. When the number of processors m is small, then the speed up S_m increases almost linearly as the number m increases. On the other hand, when m is large, then the exchange time increases, so the efficiency E_m decreases.

In Figure 2, the graph 1 and the graph 2 or the graph 3 show the efficiencies E_m depending on the number m of processors for Gauss and Gauss-Jordan algorithms or for the conjugate gradient method, respectively. The graph 1G, the graph 2G and the graph 3G show the efficiencies E obtained by using the granularity G depending on the number m of processors.

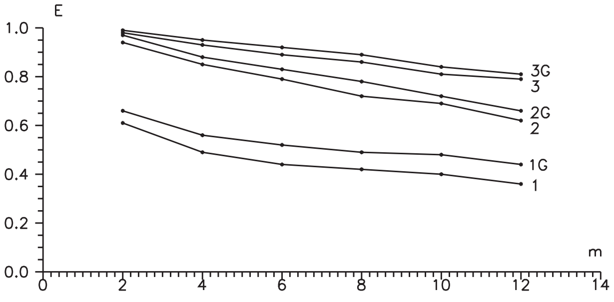

Fig. 2 Efficiencies for the Gauss, Gauss-Jordan and conjugate gradient method

The results of the experiments obtained using granularity were compared with the results obtained by standard methods of efficiency calculation. The efficiency calculated by using the granularity concept is higher than that using the classical method.

5. NUMERICAL RESULTS

In Figures 3 and 4 the profiles ($y=24$ km) of the interfaces S_1 and S_2 for the real gravity and magnetic fields of some area in the Urals for $H=2$ km and $H=10$ km are represented.

In each figure, Curve 1 (continuous lines) is the profile of the gravimetry solution given by Eq. (1) obtained by the iteratively regularized Newton method given by Eq. (4) using the parallel technique, and Curve 2 (dotted lines) is the profile of the magnetometry solution given by Eq. (2).

To approximate the integral operator in Eq. (4), we used the two-dimensional analogue of the rectangular quadrature formulas for 111×35 points of the grid domain with the mesh widths $h_x=0.5$ and $h_y=2$ km. The parameters α_k were chosen from numerical experiments.

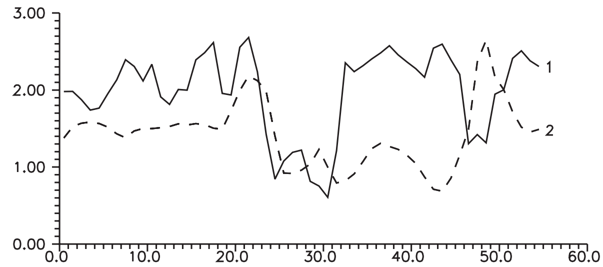

Fig. 3 Profiles (y=24 km) of the interface S_1

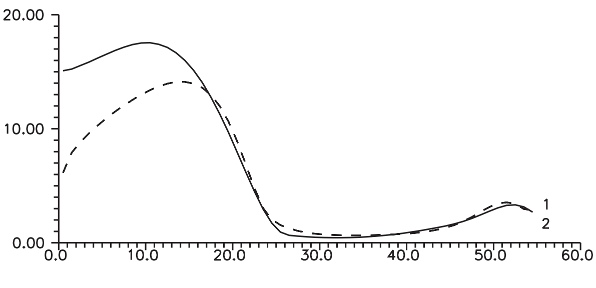

Fig. 4 Profiles (y=24 km) of the interface S_2

In the reconstruction of the interfaces S_1 ($H=2$), we used the following data: $f = 6.67\cdot10^{-5}$ is the gravitation constant, $\Delta\sigma=0.48$ g/cm^3 is the density jump on the interface, $\Delta J=6.2$ is the averaged jump of the z-th component of the magnetization vector, $z^0(x,y)=0.3$ km is the initial guess and $\alpha_k=2.5$.

The following data were used in the reconstruction of the interfaces S_2 ($H=10$): $\Delta J=4.39$, $f=6.67\cdot10^{-5}$, $\Delta\sigma=$ 0.23 g/cm^3, $z^0(x,y)=0.3$ km and $\alpha_k=1.1$.

In Figures 5 to 8 the reconstructed interfaces S_1 and S_2 for the real gravity and magnetic fields of some area in the Urals for $H=2$ km and $H=10$ km are represented.

They are reconstructed by the iteratively regularized Newton method given by Eq. (4) with the help of the parallel Gauss or Gauss-Jordan algorithms or the conjugate gradient method.

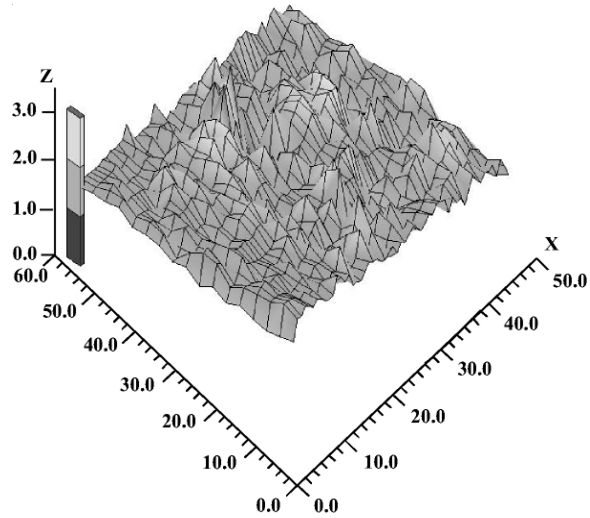

Fig. 5 The reconstructed interface S_1 for the gravimetry problem

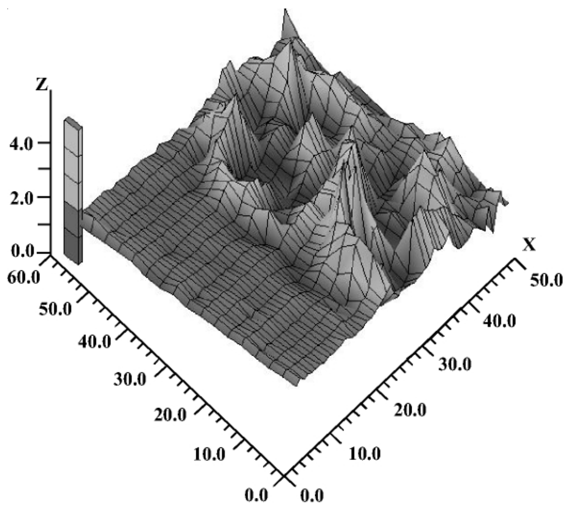

Fig. 6 The reconstructed interface S_1 for the magnetometry problem

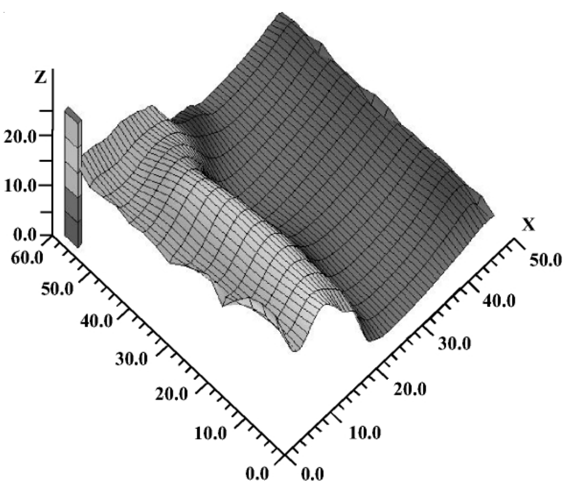

Fig. 7 The reconstructed interface S_2 for the gravimetry problem

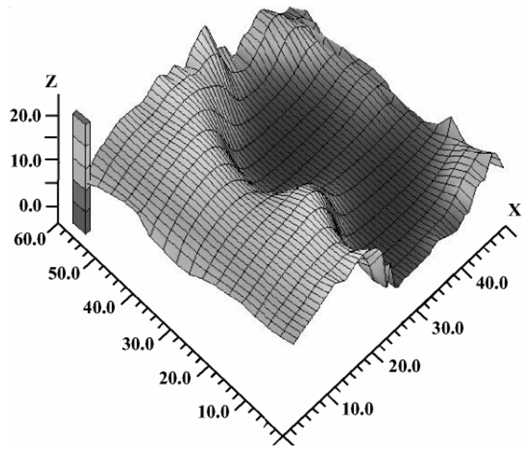

Fig. 8 The reconstructed interface S_2 for the magnetometry problem

6. CONCLUDING REMARKS

The main conclusion is the following. The interfaces S_1 and S_2 obtained as solutions of the gravimetry and magnetometry inverse problem, Eqs. (1) and (2), by the iteratively regularized Gauss-Newton method, Eq. (4), correspond to the real geological perceptions about the investigated region of the Urals.

The nearest gravity and magnetic interfaces (see Figure 3 and Figures 5 and 6) are rather different, but the deeper interfaces (see Figure 4 and Figures 7 and 8) are similar. We believe that, probably in the first case, the sources of the gravity and magnetic fields are different, and in the second case these sources are the same (or very close) and so we have the gravity and magnetic solutions very close to each other.

7. ACKNOWLEDGEMENTS

The authors are deeply grateful to the Russian Foundation for Basic Research (project No. 03-01-00099) for the financial support.

8. REFERENCES

[1] V.V. Vasin, G.Ya. Perestoronina, I.L. Prutkin and L.Yu. Timerkhanova, Reconstruction of the relief of geological boundaries in the three-layered medium using the gravitational and magnetic data, Proc. of the Conference on Geophysics and Mathematics, Institute of Mines, UrB RAS, Perm, pp. 35-41, 2001.

[2] A.B. Bakushinsky, A regularizing algorithm on the basis of the Newton- Kantorovich method for the solution of variational inequalities, *Zh. Vychisl. Mat. Mat. Fiz.*, Vol. 16, No. 6, pp. 1397-1404, 1976.

[3] G. Streng, *Linear Agebra and Its Applications*, Mir, Moskva, 1980.

[4] N.S. Bakhvalov, N.P. Zhidkov and G.M. Kobelkov, *Numerical Methods*, Nauka, Moskva, 1987.

[5] V.V. Vasin and A.L. Ageev, *Ill-posed Problems with a Priori Information*, Nauka, Ekaterinburg, 1993.

[6] E.N. Akimova, Parallelization of an algorithm for solving the gravity inverse problem, *Journal of Computational and Applied Mechanics*, Miskolc University Press, Vol. 4, No. 1, pp. 5-12, 2003.

[7] A.V. Baranov, A.O. Latsis, C.V. Sazhin and M.Yu. Khramtsov, The MVS-1000 System User's Guide, http://parallel.ru/mvs/user.html.

[8] J. Kwiatkowski, Evaluation of parallel programs by measurement of its granularity, Proc. of the Conference on Parallel Processing and Applied Mathematics, Lecture Notes in Computer Science, Vol. 2328, pp. 145-153, 2001.

STABILNI PARALELNI ALGORITMI ZA RJEŠAVANJE INVERZNIH GRAVIMETRIJSKIH I MAGNETOMETRIJSKIH PROBLEMA

SAŽETAK

U ovom radu ispituju se trodimenzionalni inverzni problemi gravimetrije i magnetometrije da bi se pronašli odnosi između medija pomoću gravitacijskih i magnetskih podataka. Pretpostavljamo da model donjeg poluprostora sadrži tri medija koji imaju stalne gustoće, a odvojeni su pomoću površina S_1 *i* S_2 *koje treba odrediti.*

Ovi inverzni problemi svedeni su na nelinearne integralne jednadžbe prve vrste, prema tome ovi problemi su loše postavljeni. Nakon diskretizacije integralne jednadžbe dobivamo sustav nelinearnih jednadžbi velike dimenzije. Koristimo iterativno reguliranu Gauss-Newton metodu da bi riješili ovaj problem. Moramo riješiti sustav linearnih algebarskih jednadžbi s punom matricom da bi realizirali jedan korak u ovoj metodi. Da se postigne ovaj cilj, primjenjuju se paralelne varijante Gaussove i Gauss-Jordan metode konjugiranog gradijenta.

Njihova realizacija iskoristila se u Massively Parallel Computing System MVS-1000. Izvršena je analiza efikasnosti paralelizacije iterativnih algoritama pomoću različitog broja procesora. Paralelizacija algoritama značajno skraćuje vrijeme rješavanja problema. Odnosi S_1 *i* S_2 *koji su postignuti Gauss-Newton-ovom metodom odgovaraju stvarnoj geološkoj predodžbi o području Urala koje se istražuje.*

Ključne riječi: *Paralelni algoritmi, gravimetrija, magnetometrija, paralelizacija.*