ISSN 1995-4239, Numerical Analysis and Applications, 2013, Vol. 6, No. 2, pp. 98–110. © Pleiades Publishing, Ltd., 2013.
Original Russian Text © E.N. Akimova, D.V. Belousov, V.E. Misilov, 2013, published in Sibirskii Zhurnal Vychislitel'noi Matematiki, 2013, Vol. 16, No. 2, pp. 107–121.

Algorithms for Solving Inverse Geophysical Problems on Parallel Computing Systems

E. N. Akimova[1,2*], D. V. Belousov[1,2], and V. E. Misilov[1***]**

[1] *Institute of Mathematics and Mechanics, Ural Branch, Russian Academy of Sciences, ul. S. Kovalevskoi 16, Yekaterinburg, 620990 Russia*

[2] *Ural Federal University, ul. Mira 19, Yekaterinburg, 620002 Russia*

Received February 20, 2012; in final form, April 9, 2012

Abstract—For solving inverse gravimetry problems, efficient stable parallel algorithms based on iterative gradient methods are proposed. For solving systems of linear algebraic equations with block-tridiagonal matrices arising in geoelectrics problems, a parallel matrix sweep algorithm, a square root method, and a conjugate gradient method with preconditioner are proposed. The algorithms are implemented numerically on a parallel computing system of the Institute of Mathematics and Mechanics (PCS–IMM), NVIDIA graphics processors, and an Intel multi-core CPU with some new computing technologies. The parallel algorithms are incorporated into a system of remote computations entitled "Specialized Web-Portal for Solving Geophysical Problems on Multiprocessor Computers." Some problems with "quasi-model" and real data are solved.

DOI: 10.1134/S199542391302002X

Keywords: *inverse gravimetry problems, parallel algorithms, direct and iterative methods, parallel computing systems.*

1. INTRODUCTION

In investigating Earth's crust structure, important inverse problems of gravimetry are the problem of finding the density variable in a layer [1] and the structural problem of reconstructing the interface between some media [2]. The gravimetry problems are described by linear and nonlinear integral Fredholm equations of the first kind, that is, they are severely ill-posed problems. Some ideas of iterative regularization are used in the development of methods for solving such problems [3]. On discretization with iterative processes, the problems are reduced to systems of linear algebraic equations (SLAEs) with ill-conditioned dense matrices of large dimensions (several hundred thousand elements). Increasing the accuracy of the results of solving such problems, specifically, by using finer grids, greatly increases the calculation time.

Important problems in investigating Earth's crust inhomogeneity are those of geoelectrics. A key problem of geoelectrics is lateral logging sounding (LLS). The logging data are interpreted to provide bed resistivity data that are close to real ones. Paper [4] shows that a finite-difference approximation reduces a two-dimensional LLS problem to SLAE with a block-tridiagonal matrix of large dimensions.

The efficiency of solving geophysical problems can be greatly increased and the calculation time, decreased by paralleling the algorithms involved and using multiprocessor computation systems. At present, many Russian research institutes and universities have cluster-type distributed-memory massively parallel supercomputers. At the Institute of Mathematics and Mechanics UrB RAS (Yekaterinburg), parallel computing systems PCS–1000/17EK, PCS–IMM, and a supercomputer "Uran" have been successfully used to solve applied problems.

[*] E-mail: aen15@yandex.ru
[**] E-mail: rtfdeamon@mail.ru
[***] E-mail: out.mrscreg@gmail.com

There is a global tendency to use multicore hybrid computers with graphic processors (video cards) as computation systems to solve applied problems. In comparison to the PCS supercomputers, the hybrid computer systems based on graphic processors are cheaper and use less energy. The Institute of Mathematics and Mechanics UrB RAS has a hybrid computer cluster based on a video accelerator NVIDIA Tesla.

In this paper, we solve the linear inverse gravimetry problem of finding the density in a layer by iterative gradient-type methods and the nonlinear inverse gravimetry problem of reconstructing the interface between some media by using a regularized Newton's method. Some SLAEs with block-tridiagonal matrices arising in geoelectrics problems are also solved. For this, we construct efficient parallel direct and iterative algorithms. The algorithms are implemented on multiprocessor computing systems of various types: a multiprocessor complex PCS–IMM, graphic processors NVIDIA, and a multicore Intel processor. The efficiency and optimization of some parallel algorithms for solving geophysical problems on hybrid computing systems are investigated. The parallel algorithms are built in a system of remote calculation called "Specialized Web-Portal for Solving Problems on Multiprocessor Computers." Some problems with "quasi-model" and real data are solved.

2. METHODS FOR SOLVING INVERSE GRAVIMETRY PROBLEMS

2.1. Problem of Finding Density in a Layer

Consider the problem of finding the density variable $\sigma = \sigma(x, y)$ in a layer $\Pi = \{(x, y, z) \in R^3 : (x, y) \in D,\ H_1(x, y) \le z \le H_2(x, y)\}$ given gravitational data measured in an area $D = \{(x, y) \in R^2 : a \le x \le b,\ c \le y \le d\}$ of the Earth's surface (H_1, H_2 are constants for the horizontal layer). A priori assumptions are: no density anomalies outside the layer with boundaries $H_1 = H_1(x, y)$ and $H_2 = H_2(x, y)$ such that $H_1 < H_2\ \forall\,(x, y)$ and $H_i(x, y) \underset{\substack{x,y\to\pm\infty\\ y\to\pm\infty}}{\longrightarrow} h_i = \text{const}$.

Assume that the density distribution $\sigma = \sigma(x, y)$ in the layer does not depend on z (the z axis is directed downwards). This assumption provides the solution uniqueness for a laterally inhomogeneous density. If the density depends on z, in a general statement there exists a continuum of solutions.

The problem of finding an unknown density $\sigma(x, y)$ is reduced to solving the following linear two-dimensional integral Fredholm equation of the first kind:

$$A\sigma \equiv f \int_a^b \int_c^d \left\{ \frac{1}{[(x-x')^2 + (y-y')^2 + H_1^2(x', y')]^{1/2}} - \frac{1}{[(x-x')^2 + (y-y')^2 + H_2^2(x', y')]^{1/2}} \right\} \sigma(x', y')\, dx'\, dy' = \Delta g(x, y), \tag{1}$$

where f is the gravitational constant and $\Delta g(x, y)$ is the gravitational effect generated by sources in a horizontal or curvilinear layer. Preliminary processing of the gravitational data to separate an anomalous field is performed using a method proposed by P.S. Martyshko and I.L. Prutkin [5].

We discretize the equation on a grid where $\Delta g(x, y)$ is given, and approximate the integral operator using quadrature formulas. Then problem (1) is reduced to solving a system of linear algebraic equations (SLAEs) with a symmetric positive definite matrix (horizontal layer) or with a nonsymmetric matrix (curvilinear layer). Since Eq. (1) is an ill-posed problem, the SLAE obtained by discretizing the equation is ill-conditioned and can be transformed to the following form (the Lavrentiev scheme):

$$(A + \alpha E)z = b, \tag{2}$$

where α is a regularization parameter.

In the case of a curvilinear layer, the initial SLAE matrix is nonsymmetric. Therefore, the system is preliminarily transformed to the following form (the Tikhonov scheme):

$$(A^\top A + \alpha' E)z = A^\top b, \tag{3}$$

where $A^\top$ is the transpose of matrix A, and α' is a regularization parameter.

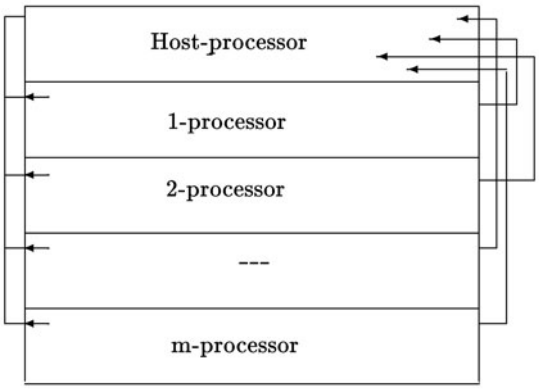

Fig. 1. Processor data distribution.

To solve the systems of equations (2) and (3), iterative gradient-type methods are used: the iteratively regularized simple-iteration method (SIM):

$$z^{k+1} = z^k - \frac{1}{\lambda_{\max}}[(A + \alpha E)z^k - b], \tag{4}$$

where $\lambda_{\max}$ is the maximal eigenvalue of matrix $A + \alpha E$ (symmetric case); the minimum residual method (MRM):

$$z^{k+1} = z^k - \frac{(A(Az^k - b), Az^k - b)}{\|A(Az^k - b)\|^2}(Az^k - b); \tag{5}$$

the minimum error method (MEM):

$$z^{k+1} = z^k - \frac{\|Az^k - b\|^2}{\|A^\top(Az^k - b)\|^2}A^\top(Az^k - b), \tag{6}$$

and the steepest descent method (SDM):

$$z^{k+1} = z^k - \frac{\|A^\top Az^k - A^\top b\|^2}{\|A(A^\top Az^k - A^\top b)\|^2}A^\top(Az^k - b). \tag{7}$$

In the regularized methods (5)–(7), matrix A is replaced by $A + \alpha E$. For the iterative processes, a residual termination criterion $\|Az^k - b\|/\|b\| < \varepsilon$ is used.

To solve the linear inverse gravimetry problem of density reconstruction in a layer, the stable iterative gradient-type methods have been numerically implemented on the multiprocessor PCS–IMM complex using the MPI library [6] and NVIDIA graphic processors with the CUDA parallel computing platform [7].

Paralleling of the gradient-type iterative methods is done by partitioning matrix A into m blocks by horizontal bands and dividing the solution vector z and the right-hand side vector b of the SLAE into m parts so that $n = m \cdot L$, where n is the dimension of the system of equations, m is the number of processors, and L is the number of matrix rows in the block (Fig. 1). At the current iteration, each of m processors calculates its part of the solution vector. In the multiplication of matrix A by a vector z, each of the m processors multiplies its part of the rows of matrix A by the vector z. In the matrix multiplication $A^\top A$, each of the m processors multiplies its part of the rows of the transpose $A^\top$ by the full matrix A. The host (master) processor sends the data and calculates its part of the solution vector. Notice that when the parallel iterative methods are implemented on the PCS–IMM for solving the gravimetry problem, the SLAE matrix is formed, and its parts are stored in the memory of each of the processors. This increases the paralleling efficiency.

The iteratively regularized simple-iteration method (SIM) on NVIDIA video accelerators in solving the linear gravimetry problem is parallelized and implemented in paper [8]. In the present paper, the

same principles are used to parallelize and implement the iterative methods MRM, MEM, and SDM on NVIDIA video accelerators.

The following two methods are used to optimize the memory in the calculations:

1. For grids of small dimensions (12100×12100), when the data can be stored in the video card memory, matrix A and vectors z of dimension n are increased to dimension M, by zeros added so that M is a multiple of the number of blocks. The block BLOCK_SIZE size (thread) is taken as a multiple of 16, since up to 512 threads are grouped in one block. Then the number of blocks is calculated by the formula blocks = M/BLOCK_SIZE. The calculations are made without data loading into the host processor memory. The data are stored only in the video card memory.

2. For grids of large dimensions (40000×40000), when the data cannot be stored in the video card memory, a method of calculating matrix A elements "immediately" is the fastest. This means that a matrix element is calculated at the time of access to this element, without storing it in the video card memory. This greatly decreases the frequency of access to the video card memory and accelerates the process of calculations, in comparison to the calculations when matrix A is stored in the host processor memory and the video accelerator is loaded portionwise.

2.2. The Problem of Finding the Interface between Media

Consider the three-dimensional structural inverse gravimetry problem of reconstructing the interface between some media, given a density jump and gravitational field measured in an Earth's surface area. Assume that the lower half-space consists of two layers of constant density divided by the sought-for surface S. Assume that a gravitational anomaly is created by a deviation of the sought-for surface S from the horizontal plane $z = H$ (the z axis is directed downwards). In a Cartesian system of coordinates the function $z = z(x, y)$ describing the sought-for interface satisfies the following nonlinear two-dimensional integral Fredholm equation of the first kind:

$$A[z] \equiv f\Delta\sigma \int_a^b \int_c^d \left\{ \frac{1}{[(x-x')^2 + (y-y')^2 + z^2(x',y')]^{1/2}} - \frac{1}{[(x-x')^2 + (y-y')^2 + H^2]^{1/2}} \right\} dx'\, dy' = G(x,y), \tag{8}$$

where f is the gravitational constant, $\Delta\sigma$ is the density jump at the interface, $G(x, y)$ is an anomalous gravitational field, and $z = H$ is the asymptotic plane for this interface. The gravimetry problem (8) is a severely ill-posed problem.

Discretizing equation (8) on a $n = M \times N$ grid where $G(x, y)$ is given, and approximating the integral operator using some quadrature formulas, we obtain the following system of nonlinear equations:

$$A_n[z] = F_n. \tag{9}$$

To solve the system of equations (9), we use the following iteratively regularized Newton's method [9]:

$$z^{k+1} = z^k - [A_n'(z^k) + \alpha_k I]^{-1}[A_n(z^k) + \alpha_k z^k - F_n]. \tag{10}$$

Here $A_n(z^k)$ and F_n are finite-dimensional approximations of the integral operator and right-hand side of equation (8), $A_n'(z^k)$ is the derivative of operator A at point z^k, I is the unit operator, and α_k is a sequence of positive regularization parameters.

Finding the next Newton's method approximation z^{k+1} from z^k is reduced to solving the SLAE

$$A_n^k z^{k+1} = F_n^k, \tag{11}$$

where $A_n^k = A_n'(z^k) + \alpha_k I$ is an ill-conditioned nonsymmetric dense $n \times n$ matrix, and $F_n^k = A_n^k z^k - (A_n(z^k) + \alpha_k z^k - F_n)$ is a vector of dimension n.

At first, the system of equations (11) is reduced to the form

$$B^k z^{k+1} \equiv [(A_n^k)^\top A_n^k + \alpha_k' I] z^{k+1} = (A_n^k)^\top F_n^k \equiv b, \tag{12}$$

where $(A_n^k)^\top$ is the transpose, and α_k' are regularization parameters.

At each step of Newton's method, SLAE (12) with a symmetric positive definite matrix is solved by the gradient-type iterative methods (4)–(7) and the regularized variant of the conjugate gradient method (CGM)

$$z^{k+1} = z^k - \gamma_k(B^k z^k - b) + \beta_k(z^k - z^{k-1}), \tag{13}$$

where γ_k and β_k are calculated from well-known formulas [10]. The termination criterion for the CGM iterative process is as follows: $\|Bz^k - b\|/\|b\| < \varepsilon$.

The Newton's method with the gradient methods for solving the nonlinear inverse gravimetry problem for finding the interface between media was numerically implemented and parallelized on the PCS–IMM system. It is planned to implement the Newton's method on NVIDIA graphic processors in the near future.

3. PARALLEL ALGORITHMS FOR SOLVING SLAES WITH BLOCK-TRIDIAGONAL MATRICES

As mentioned in the introduction, the LL problem is reduced by the finite-difference approximation to solving a SLAE with the block-tridiagonal matrix presented in Fig. 2.

To solve SLAEs with block-tridiagonal matrices in geoelectrics problems, the following methods are used: parallel matrix sweep algorithms (PMSAs), parallel square root algorithms (PSRAs), and a conjugate gradient method with preconditioner (PCGM) for solving a SLAE with a symmetric positive definite matrix. The parallel algorithms were implemented on NVIDIA graphic processors with the CUDA technology and on a multicore Intel processor with the OpenMP platform [6].

3.1. Parallel Matrix Sweep Algorithm

Consider the following system of equations with a block-tridiagonal matrix of general form:

$$\begin{cases} C_0\overline{Y}_0 - B_0\overline{Y}_1 = \overline{F}_0\,, & i = 0, \\ -A_i\overline{Y}_{i-1} + C_i\overline{Y}_i - B_i\overline{Y}_{i+1} = \overline{F}_i\,, & i = 1, 2, \ldots, N-1, \\ -A_N\overline{Y}_{N-1} + C_N\overline{Y}_N = \overline{F}_N\,, & i = N, \end{cases} \tag{14}$$

where $\overline{Y}_i$ are the sought-for vectors of dimension n, $\overline{F}_i$ are given vectors of dimension n, and A_i, B_i, C_i are square matrices of order n.

In paper [11], it is proposed to use a parallel matrix sweep algorithm to solve SLAE (14). The main idea of the parallel algorithm is as follows. The initial domain P (rectangle) is divided into L subdomains by vertical lines so that $N = L \times M$. Some vectors $\overline{Y}_K$, $K = 0, M, \ldots, N$, are taken as parametric

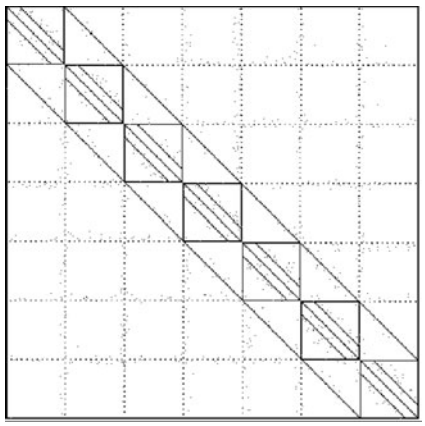

Fig. 2. SLAE matrix.

unknowns. They bind together the unknowns on the grid in the vertical direction. A reduced system of equations is constructed for $\overline{Y}_K$, which is smaller in dimension than the initial one. The system is solved by the classical matrix sweep method [12]. First, the vectors-parameters $\overline{Y}_K$ are found; then the other sought-for unknowns are expressed in terms of the parametric unknowns and found in each subdomain L independently.

3.2. Conjugate Gradient Method with Preconditioner

An efficient conjugate gradient method with preconditioner can be used to solve SLAE (14). Preconditioning is used to accelerate convergence of the iterative process. Here the initial system of equations $Ax = b$ is replaced by the following system:

$$C^{-1}Ax = C^{-1}b, \tag{15}$$

for which the iterative method converges much faster.

The following condition is used to choose the preconditioner C:

$$\mathrm{cond}(\tilde{A}) \ll \mathrm{cond}(A), \quad \mathrm{cond}(\tilde{A}) = \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}, \quad \mathrm{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}}, \tag{16}$$

where $\mathrm{cond}(A)$ and $\mathrm{cond}(\tilde{A})$ are the condition numbers of matrices A and $\tilde{A}$; $\lambda_{\max}$, $\tilde{\lambda}_{\max}$ and $\lambda_{\min}$, $\tilde{\lambda}_{\min}$ are the largest and smallest eigenvalues of matrices A and $\tilde{A}$, respectively.

The conjugate gradient method with preconditioner C for the system of equations (15) has the following form:

$$\begin{gathered} r^0 = b - Ax^0, \quad p^0 = C^{-1}r^0, \quad z^0 = p^0, \\ x^{k+1} = x^k + \alpha_k p^k, \quad \alpha_k = \tfrac{(r^k, z^k)}{(Ap^k, p^k)}, \quad r^{k+1} = r^k - \alpha_k Ap^k, \\ z^{k+1} = C^{-1}r^{k+1}, \ p^{k+1} = z^{k+1} + \beta_k p^k, \quad \beta_k = \tfrac{(r^{k+1}, z^{k+1})}{(r^k, z^k)}. \end{gathered} \tag{17}$$

Here the preconditioner is chosen by an incomplete LU-decomposition of matrix A. A residual termination criterion is used for termination of the CGM iterative process with preconditioner. The CGM is parallelized in a similar way as the other iterative gradient-type methods (see Fig. 1).

3.3. Square Root Method

The square root method is one of the fastest methods for solving a SLAE with a symmetric positive definite matrix [10]. The method is based on decomposing the symmetric matrix A into a product $A = S^\top S$, where S is an upper triangular matrix with positive elements on the main diagonal, and $S^\top$ is the transpose. The method consists in successive solving the following two systems of equations with triangular matrices:

$$S^\top y = b, \qquad Sz = y. \tag{18}$$

Solutions to the systems of equations (18) are found by the following recurrent formulas:

$$\begin{cases} y_1 = b_1/s_{11}, \quad y_i = \left(b_i - \sum\limits_{k=1}^{i-1} s_{ki}y_k\right) \Big/ s_{ii}, \quad i = 2, 3, \ldots, n; \\ z_n = y_n/s_{nn}, \ z_i = \left(y_i - \sum\limits_{k=i+1}^{n} s_{ik}z_k\right) \Big/ s_{ii}, \ i = n-1, n-2, \ldots, 1. \end{cases} \tag{19}$$

The main idea of the square root method parallelization for a multiprocessor shared-memory computer is parallel calculation of elements s_{ij}, $j = i, \ldots, n$, of every ith row of matrix S. The row with number i is divided into m parts so that $n - i = m \times L_i$, where i is the row number, n is the dimension of the system of equations, m is the number of processors, and L_i is the number of row elements calculated by each processor (Fig. 3).

1-processor	2-processor	...	m-processor

Fig. 3. The ith row processor distribution.

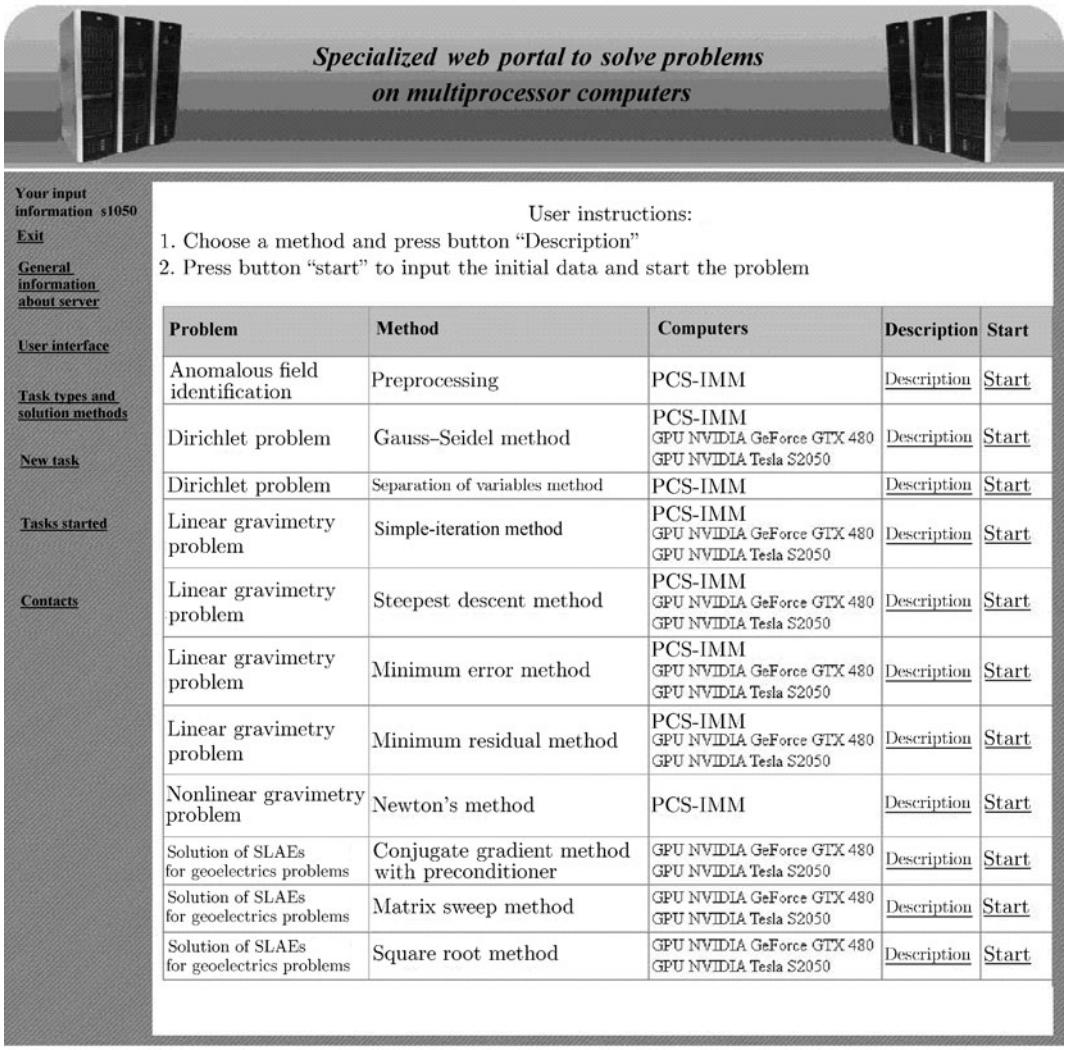

Fig. 4. Specialized web portal.

4. SPECIALIZED WEB PORTAL TO SOLVE PROBLEMS ON MULTIPROCESSOR COMPUTERS

A system for remote calculation called "Specialized Web-Portal for Solving Geophysical Problems on Multiprocessor Computers" has been developed and installed at the Department of Ill-Posed Problems of Analysis and Applications of the Institute of Mathematics and Mechanics UrB RAS (Fig. 4). The parallel algorithms have been built in this system to solve the inverse gravimetry problem of density reconstruction in a layer, the structural inverse gravimetry problem of reconstructing the interface between media, and SLAEs with block-tridiagonal matrices in geoelectrics problems.

Initially the web portal was used to perform tasks for solving gravimetry problems on a multiprocessor system, PCS–1000/17EK, with a web interface [13].

At present the web portal can be used to perform tasks for solving problems on a PCS–IMM parallel computing system, a hybrid computer NVIDIA Tesla cluster installed at the Institute of Mathematics and Mechanics UrB RAS, and an NVIDIA GeForce hybrid computer system (HCS) installed at the

Department of Computational Mathematics and Mathematical Physics Equations and Radiotechnical Faculty of the Institute of Radioelectronics and Information Technologies of Ural Federal University.

The PCS–IMM has 14 two-processor dual-core AMD Opteron 64 bit (2.6 GHz) modules, a GbitEthernet interface, and 112 gigabyte RAM. The NVIDIA Tesla cluster has 20 computational nodes with 8 GPU Tesla S2050, 50 gigabyte RAM, and two six-core CPUs. The HCS is a four-core processor Intel Core I7-950 with NVIDIA GeForce GTX 480 graphic processors.

4.1. General Description

The specialized web portal can be used to perform tasks for solving problems of gravimetry (identification of anomalous fields, finding of density in a layer, reconstruction of the interface between media) and solve SLAEs with block-tridiagonal matrices for geoelectrics problems on the multiprocessor complex PCS–IMM, the NVIDIA Tesla computer cluster, and the NVIDIA GeForce HCS system. The web portal, via the web interface, allows the user to choose the necessary computer type, specify the number of processor nodes, the problem type, and the method of its solution, load the input data, obtain the output data, and display the solution results using the Surfer and Gnuplot graphic packages. The calculation time for each task is also shown.

The web portal consists of three major parts: a HTTP-server IIS (Internet Information Services) with a web application; a data base SQL Server 2000 to store all user tasks with input and output data; a service to perform the data loading, start tasks on various multiprocessor computers, determine task status, and load the results into the web portal.

4.2. Web Application

The following services have been added to the initial web application (see [13]):

1. Authentication system. In the process of web portal registration, a homonymous account on PCS–IMM is checked. If the account is absent or the passwords do not coincide, the registration is canceled. SSL-encoding of traffic has been implemented.

2. Possibility of interaction with various types of computers and use problem solving methods on various multiprocessors communicating via ssh, transfer files via scp, and start programs via the mpirun or sbatch planner.

3. Modification of services responsible for file input and transfer and task run on various computers. The task run interface has been complemented by a possibility of choosing the computer to run the task. The browsing interface of started tasks can now indicate the computer on which the chosen task has been started. It is possible to delete the started task from the task list.

4. Data files loaded for calculation can be controlled, that is, another task can be started with the same data to decrease the server load (file transfer). The sets of task parameters are increased. For instance, there are two possibilities of solving linear gravimetry problems: introduction of two constants (layer boundaries) for a horizontal layer and loading of two files (layer boundaries) for a curvilinear layer.

5. Control of images: the user can choose input and output files to construct images of each specific problem to be solved. For gravimetry problems, a three-dimensional surface and level lines are constructed using program Scripter of the Surfer package.

6. Page design is modified. Detailed descriptions of problems and methods are available. A content control system of on-site editing of descriptions of the available methods for solving problems and adding new ones has been created.

5. RESULTS OF NUMERICAL EXPERIMENTS

5.1. Solving Gravimetry Problems with Real Data

A $59.4 \times 144\,\mathrm{km}^2$ array of gravitational data measured on area S of the Eastern Urals has been processed. This area contains the secular-variation Butkin anomaly and borders on the Urals from the East. This is a submeridional anomaly of the Earth's crust electrical conductivity. The data for calculating the anomalous gravitational field were obtained at the Institute of Geophysics UrB RAS (Yekaterinburg).

The following problems have been solved to study the nature of this anomaly with real data of observation: the problem of finding the density in a horizontal layer between $H_1 = 10$ km and $H_2 = 20$ km for domain S, and the problem of reconstructing the interface between the media. The grid spacings $\Delta x = 0.594$ km and $\Delta y = 1.44$ km, the gravitational constant $f = 6.67 \cdot 10^{-8}\mathrm{cm}^3/\mathrm{g}{\cdot}\mathrm{s}^2$. The distance to the asymptotic plane was $H = 15$ km. The density jump was $\Delta\sigma = 0.2\ \mathrm{g/cm}^3$.

With discretization of the initial equations on the grid, the problems were reduced to SLAEs with a symmetric (problem 1) and nonsymmetric (problem 2) 10000×10000 matrices. To solve problem 1, a parallel iteratively regularized MRM with a regularization parameter $\alpha = 0.001$ was used. To solve problem 2, an iteratively regularized Newton's method, with CGMP applied at each step, was used. Problem 1 was solved on the PCS–IMM, NVIDIA Tesla, and HCS GeForce. Problem 2 was solved on the PCS–IMM.

Figure 5 shows the layer density distribution reconstructed from an anomalous field for the domain S.

Figure 6 shows the reconstructed interface. The results were interpreted at the Institute of Geophysics UrB RAS (Yekaterinburg). The interpretation revealed a lengthy submeridional Earth's crust block of low density (0.2—$0.3\ \mathrm{g/cm}^3$) (see [14]).

Table 1 presents solution times for the linear gravimetry problem on the PCS–IMM and HCS NVIDIA GeForce GTX 480 at $\|Az^k - b\|/\|b\| \approx 0.011$ (749 iterations). The MPI technology was used to solve the problem on the PCS–IMM, the OpenMP technology, to solve it on the Intel multicore processor, and the CUDA technology, to solve it on the GeForce video accelerator.

To compare the calculation times for solving the problem, we introduce the following acceleration and efficiency coefficients of the parallel algorithms:

$$S_m = T_1/T_m, \qquad E_m = S_m/m, \qquad S = T_1/T_2,$$

where T_m is the execution time of the parallel algorithm on the PCS–IMM or on a multicore processor with m processors or cores ($m > 1$), T_1 is the execution time of the sequential algorithm with single processor or core, T_2 is the solution time of the problem on the video accelerator, and T_m is the sum of the actual calculation time and the overhead time: $T_m = T_c + T_o$.

Table 1. Solution times of the linear gravimetry problem

Computer	Time T_m, min	Acceleration S_m or S	efficiency E_m
Intel Core I7 (1 core)	7.73	—	—
Intel Core I7 (2 cores)	4.05	1.91	0.96
Intel Core I7 (4 cores)	2.1	3.68	0.92
NVIDIA GeForce GTX 480	0.2	38.7	—
PCS–IMM (1 processor)	24.33	—	—
PCS–IMM (2 processors)	12.18	1.99	0.99
PCS–IMM (4 processors)	6.1	3.99	0.99
PCS–IMM (10 processors)	2.46	9.89	0.99
PCS–IMM (20 processors)	1.24	19.6	0.98
PCS–IMM (50 processors)	0.5	48.7	0.97

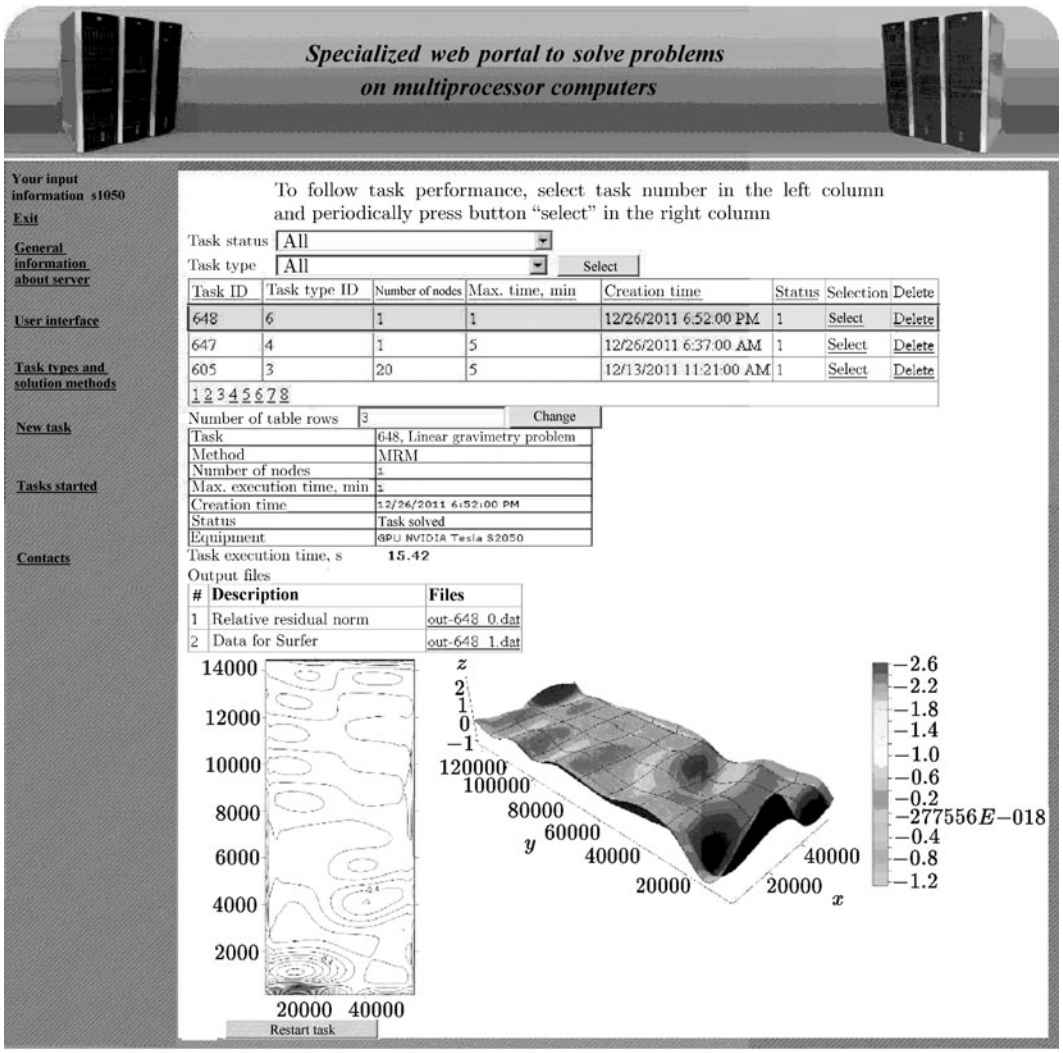

Fig. 5. Solution of the linear gravimetry problem.

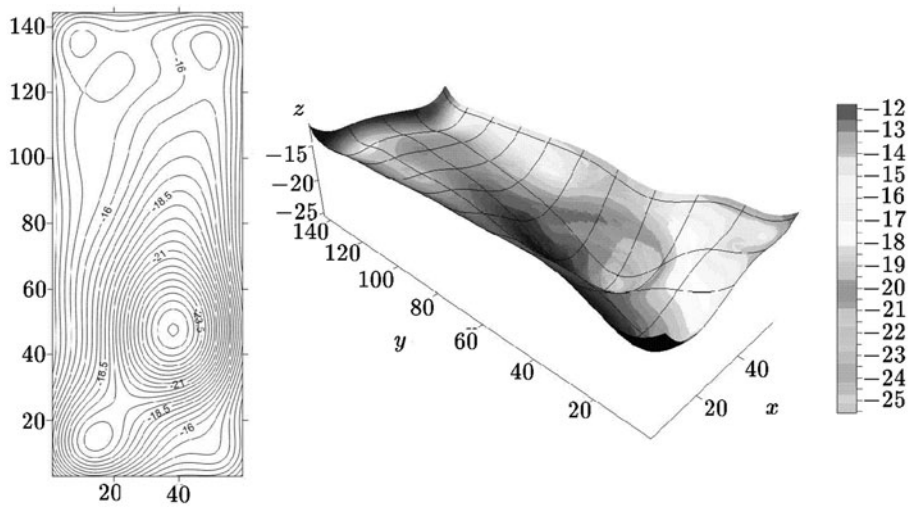

Fig. 6. Solution of the nonlinear gravimetry problem.

The results of calculations show that the use of Newton's method and gradient-type iterative methods to solve inverse gravimetry problems allows us to obtain correct solutions and determine the anomalous density parameters of deep Earth's crust zones. The use of parallel algorithms in solving gravimetry problems have greatly reduced the calculation time.

5.2. Solving the Potential Distribution Problem

The parallel matrix sweep algorithm, the preconditioned conjugate gradient method, and the square root method were used to solve the problem of finding the potential distribution in a conducting medium with a known quasi-model solution. The initial data and the solution to this problem were obtained at the Laboratory of Borehole Geophysics of the Trofimuk Institute of Petroleum Geology and Geophysics SB RAS (Novosibirsk).

The problem was discretized and reduced to solving a SLAE with an ill-conditioned symmetric positive definite block-tridiagonal 76136×76136 matrix with square blocks of order 248.

The approximate solution to this problem was compared to the model solution by calculating the relative error

$$\sigma = \|\overline{Y}^{\mathrm{M}} - \overline{Y}^{\mathrm{A}}\|/\|\overline{Y}^{\mathrm{M}}\|,$$

where $\overline{Y}^{\mathrm{M}}$ is the model solution, and $\overline{Y}^{\mathrm{A}}$ is the approximate solution to this problem.

The condition $\sigma < \varepsilon$ was taken as a termination criterion for the iterative CGMP. First, the condition number of the initial matrix A was found:

$$\mathrm{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}} \approx 1.3 \cdot 10^{11}, \qquad \lambda_{\max} \approx 1.4 \cdot 10^{6}, \qquad \lambda_{\min} \approx 1.1 \cdot 10^{-5} > 0,$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximal and minimal eigenvalues of the initial matrix.

For the case where the problem was solved with the preconditioned CGMP, to verify condition (16) the condition number of matrix $\tilde{A}$ was calculated:

$$\mathrm{cond}(\tilde{A}) = \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}} \approx 4.1 \cdot 10^{9} < \mathrm{cond}(A).$$

The problem was solved by the parallel conjugate gradient method with preconditioner, the parallel matrix sweep algorithm, and the parallel square root method at $\sigma_{\mathrm{CGMP}} \approx 10^{-7}$, $\sigma_{\mathrm{PMRA}} \approx 2 \cdot 10^{-7}$, and $\sigma_{\mathrm{PSRM}} \approx 6 \cdot 10^{-7}$. Figure 7 presents the numerical solution to the problem.

Table 2 presents the calculation times for solving the linear gravimetry problem on a HCS installed at the Department of Ill-Posed Problems of Analysis and Applications at the Institute of Mathematics and

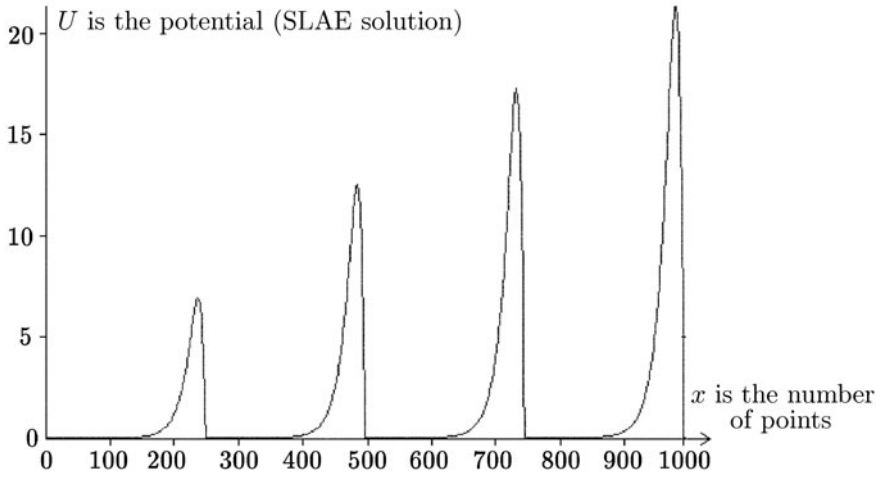

Fig. 7. Numerical solution to the problem.

Table 2. Problem solution times

Method	Computer	T_m, s (Windows API)	T_m, s (OpenMP)
CGMP	Intel Core I5 (1 core)	57	21
CGMP	Intel Core I5 (2 cores)	46	16
CGMP	Intel Core I5 (4 cores)	36	14
CGMP	NVIDIA GeForce GTX 285	—	—
PMRA	Intel Core I5 (1 core)	52	21
PMRA	Intel Core I5 (2 cores)	28	18
PMRA	Intel Core I5 (4 cores)	16	14
PMRA	NVIDIA GeForce GTX 285	—	10
PSRA	Intel Core I5 (1 core)	12	7.4
PSRA	Intel Core I5 (2 cores)	9	4.6
PSRA	Intel Core I5 (3 cores)	10	3.8
PSRA	Intel Core I5 (4 cores)	12	4.2

Mechanics UrB RAS. The system has a 4-core processor Intel Core I5-750 and an NVIDIA GeForce GTX 285 video accelerator. It is planned to include this HCS into the web portal. Notice that the time of solving the problem with the CGM without preconditioner with one core of Intel Core I5-750 at $\sigma_{\mathrm{CGM}} = 10^{-3}$ was 55 minutes, which is much longer than the solution times presented in Table 2.

Early in the development of the system, the CGMP, PMRA, and PSRA methods for a multicore Intel processor with common memory were parallelized by creating operating system (OS) threads using the Windows API development tools [15]. For parallel implementation of the program calculation, threads were created. Each of the threads was implemented on an OS "logical processor" to calculate its own data portion. At the end of each block, a barrier synchronization of the threads was done. To optimize the algorithm and decrease the calculation time, the OpenMP technology was used. The cycles were automatically parallelized using the function library OpenMP with special compiler directives. The interval of size L of the cycle variable i was divided into m parts. Each thread of the process calculated its own pth part of the data, where $p = L/m$ (Fig. 3).

To decrease the calculation time, the NVIDIA CUDA technology was also used. The program with CUDA differs from that with OpenMP in that the program on CUDA uses "thousands of threads." Such programs employ a massively parallel principle of programming. In the development of the algorithms, only "expensive" vector-matrix operations were used at the GPU. The principle of operation was as follows. The main thread of the program was implemented on the CPU, and the data from the main memory were loaded on the GPU where calculations were performed. Then the results of the calculations were loaded to the main memory. Parallelization on the GPU is often inefficient due to the formation of a memory bandwidth "bottleneck."

The PSRA method is the fastest. The time for solving a 76136×76136 SLAE on a 4-core Intel processor and video card is merely several seconds.

The results of the calculations have shown that the parallel methods PMRA, PSRA, and CGMP with preconditioner can solve problems with ill-conditioned matrices on multiprocessor computers very fast. These methods may be recommended for solving geoelectrics problems.

6. CONCLUSIONS

Some parallel direct and iterative algorithms were developed to solve the linear inverse problem of gravimetry of finding the density in a layer by iterative gradient-type methods, the nonlinear inverse gravimetry problem of reconstructing the interface between media using the regularized Newton's method, and SLAEs with block-tridiagonal matrices for geoelectrics problems. The algorithms were

implemented on the following multiprocessor computer systems: a multiprocessor complex PCS–IMM, NVIDIA graphic processors, and a multicore Intel processor. The parallel algorithms were built in a system developed for remote calculations called "Specialized Web-Portal for Solving Problems on Multiprocessor Computers." Problems with "quasi-model" and real data were solved.

The results of calculations have shown that the use of Newton's method and parallel gradient-type iterative methods to solve inverse gravimetry problems on multiprocessor systems allows us to obtain efficient correct solutions. With the parallel matrix sweep, square root, and conjugate gradient methods with preconditioner SLAEs with ill-conditioned block-tridiagonal matrices for geoelectrics problems can be solved very fast on multiprocessor computers.

ACKNOWLEDGMENTS

The paper was recommended for publication by the Program Committee of the International Conference "Parallel Computational Technologies 2012." This work was supported by the Ural Branch of the Russian Academy of Sciences under Fundamental Research Programs of RAS Presidium no. 15 (project no. 12-P-1-1023 18) and no. 18 (project no. 12-P-15-2019).

REFERENCES

1. Martyshko, P.S. and Koksharov, D.E., Determination of Density in a Layered Medium by Gravitational Data, *Geofiz. Zh.*, 2005, vol. 27, no. 4, pp. 678–684.
2. Numerov, B.V., Interpretation of Gravitational Observations in the Case of One Contact Surface, *Dokl. Akad. Nauk SSSR*, 1930, no. 21, pp. 569–574.
3. Vasin, V.V. and Eremin, I.I., *Operatory i iteratsionnye protsessy feierovskogo tipa. Teoriya i prilozheniya* (Operators and Iterative Processes of Fejer Type: Theory and Applications), Yekaterinburg: UrB RAS, 2005.
4. Dashevskii, Yu.A., Surodina, I.V., and Epov, M.I., Quasi-Three-Dimensional Mathematical Simulation of Diagrams of Nonaxisymmetric Direct Current Sondes in Anisotropic Media, *Sib. Zh. Ind. Mat.*, 2002, vol. 5, no. 3 (11), pp. 76–91.
5. Martyshko, P.S. and Prutkin, I.L., Technology of Separation of Gravitational Field Sources in Depth, *Geofiz. Zh.*, 2003, vol. 25, no. 3, pp. 159–168.
6. Voevodin, Vl.V., *Tekhnologii parallel'nogo programmirovaniya* (Parallel Programming Technologies), URL: http://parallel.ru/ (download date: 06.02.2012).
7. Berillo, A., *NVIDIA CUDA—negrafícheskie vychisleniya na grafícheskikh protsessorakh* (NVIDIA CUDA—Nongraphic Calculations on Graphic Processors), URL: http://www.ixbt.com/video3/cuda-1.shtml (download date: 06.02.2012).
8. Akimova, E.N. and Belousov, D.V., Parallelization of Algorithms for Solving the Linear Inverse Problem of Gravimetry on PCS–1000 and Graphic Processors, *Vestnik NNGU*, pt. 1, 2010, no. 5, pp. 193–200.
9. Bakushinsky, A. and Goncharsky, A., *Ill-Posed Problems: Theory and Applications*, London: Kluwer, 1994.
10. Faddeev, V.K. and Faddeeva, V.N., *Vychislitel'nye metody lineinoi algebry* (Computational Methods of Linear Algebra), Moscow: *Gostekhizdat*, 1963.
11. Akimova, E.N., Parallelization of the Matrix Sweep Algorithm, *Mat. Model.*, 1994, vol. 6, no. 9, pp. 61–67.
12. Samarskii, A.A. and Nikolaev, E.S., *Metody resheniya setochnykh uravnenii* (Methods for Solving Grid Equations), Moscow: Nauka, 1978.
13. Akimova, E.N. and Gemaidinov, D.V., Parallel Algorithms for Solving the Inverse Gravimetry Problem and Organization of Remote Communication between PCS–1000 and the User, *Vych. Met. Progr.*, 2008, vol. 9, no. 1, pp. 133–144.
14. Martyshko, P.S., Vasin, V.V., Akimova, E.N., and P'yankov, V.A., Complex Interpretation of Gravitational and Magneto-Variational Data (Using Pre-Ural Bashkiria as an Example), *Geofiz.*, 2011, no. 4, pp. 30–36.
15. *Metodika razrabotki mnogopotochnykh prilozhenii: printsipy i prakticheskaya realizatsiya* (Methods of Developing Multithread Applications: Principles and Practical Implementation), URL: http://www.rsdn.ru/article/baseserv/RUThreadingMethodology.xml (download date: 06.02.2012).