

Memory Efficient Algorithm for Solving the Inverse Problem of Finding a Density in a Curvilinear Layer

Elena N. Akimova^{1,2}(ORCID 0000-0002-4462-5817),
Vladimir E. Misilov^{1,2}, and Maxim S. Arguchinsky²

¹ Ural Federal University, Ekaterinburg, Russia

² Krasovskii Institute of Mathematics and Mechanics, Ural Branch of RAS,
Ekaterinburg, Russia
aen15@yandex.ru, v.e.misilov@urfu.ru, deryabino.49@mail.ru

Abstract. A memory efficient algorithm is constructed and implemented for solving the inverse problem of finding a variable density in a curvilinear layer using gravitational data. The algorithm is based on the stabilized biconjugated gradient method. The modification based on approximation of the SLAE matrix by Toeplitz-block-Toeplitz one significantly reduces memory requirements. The parallel algorithms were implemented for the Uran supercomputer using the hybrid MPI+OpenMP technology. A model problem with synthetic data was solved.

Keywords: density reconstruction problem, linear inverse problem, ill-posed problems, SLAE, Toeplitz matrix, BiCGSTAB method, parallel computing, MPI, OpenMP

1 Introduction

The problem of finding a density in a horizontal or curvilinear layer between given depths was studied in works [1–7]. Solution process consists in two separate steps. The first one is to extract approximately anomalous effect of the considered layer from the observed gravitational field using special technique [3]. This technique was implemented for various multiprocessor systems [5, 6]. The extracted field is the right part of an integral equation of the desired density. After discretization of the area and approximation of the integral operator, both of these steps are reduced to solving ill-conditioned systems of linear algebraic equations.

In work [7], a parallel algorithm for solving density reconstruction problem was created on the basis of the stabilized biconjugate gradient (BiCGSTAB) method and implemented for multicore processor using the OpenMP technology.

In this work, this algorithm is implemented using the hybrid MPI+OpenMP technology for the Uran supercomputer installed at the Krasovskii Institute of Mathematics and Mechanics.

The memory efficient algorithm for solving this problem is constructed and implemented for multicore processor. The modification based on approximation

of the SLAE matrix by Toeplitz-block-Toeplitz one significantly reduces memory requirements.

2 Statement of density reconstruction problem

Let us assume a Cartesian coordinate system where plane xOy coincides with the Earth's surface, and axis z is directed downwards.

The vertical attraction of gravity $\Delta g(x, y, 0)$ generated by the layer $\Pi = \{(x, y, z) : H_1(x, y) \leq z \leq H_2(x, y)\}$ is given by the following equation [7]:

$$f \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left\{ \frac{1}{\sqrt{(x-x')^2 - (y-y')^2 - H_1^2(x, y)}} - \frac{1}{\sqrt{(x-x')^2 - (y-y')^2 - H_2^2(x, y)}} \right\} \sigma(x', y') dx' dy' = \Delta g(x, y, 0), \quad (1)$$

where f is the gravitational constant, $\sigma(x, y)$ is the density distribution in the layer.

Let us assume that the boundary surfaces $H_1(x, y)$ and $H_2(x, y)$ satisfy the conditions $H_1(x, y) < H_2(x, y)$, $\forall (x, y) \in R^2$, $\lim_{x, y \rightarrow \pm\infty} H_i(x, y) = \text{const}$, and the density is constant in areas $\{(x, y, z) : \overline{H_1} \leq z \leq H_1(x, y)\}$ and $\{(x, y, z) : H_2(x, y) \leq z \leq \overline{H_2}\}$, where $\overline{H_1} = \min_{x, y} H_1(x, y)$ and $\overline{H_2} = \max_{x, y} H_2(x, y)$.

Problem (1) is a linear two-dimensional Fredholm integral equation of the first kind of $\sigma(x, y)$; thus, the problem is ill-posed.

After discretization on a grid $n = M \times N$ and approximation of the integral operator using quadrature formulae, expression (1) takes the following form

$$f \sum_{i=1}^N \sum_{j=1}^M \left\{ \frac{1}{\sqrt{(x_v - x_i)^2 - (y_u - y_j)^2 - H_{1, (i-1)M+j}^2}} - \frac{1}{\sqrt{(x_v - x_i)^2 - (y_u - y_j)^2 - H_{2, (i-1)M+j}^2}} \right\} \sigma_{(i-1)M+j} \Delta x \Delta y = b_{(v-1)M+u}, \quad (2)$$

$u \in [1..M], v \in [1..N].$

The coefficient matrix of SLAE is an ill-conditioned nonsymmetric matrix of $n \times n = MN \times MN$ dimension.

After application of the regularization using the Lavrentiev scheme [1], we can rewrite this SLAE in the following form:

$$Az = b, \quad (3)$$

where $A = \overline{A} + \alpha E$, α is the regularization parameter.

3 Algorithms for solving the problem and their parallel implementation

BiCGSTAB algorithm. For solving system (3), we use BiCGSTAB algorithm (Fig.1) [8]. In work [7], we showed that this method is the most efficient of iterative gradient methods for solving SLAE with nonsymmetric matrix.

```


$$r^0 = b - Az^0, \quad \bar{r} = r^0, \quad \rho^0 = \alpha^0 = \omega^0 = 1, \quad v^0 = q^0 = 0$$

for ( $k = 0; k + +;$ ) do


$$\rho^k = \langle \bar{r}, r^{(k-1)} \rangle, \quad \beta^k = \frac{\rho^k}{\rho^{k-1}} \frac{\alpha^{k-1}}{\omega^{k-1} - 1}$$


$$q^k = r^{k-1} + \beta^k (q^{k-1} - \omega^{k-1} v^{k-1})$$


$$v^k = Aq^k, \quad \alpha^k = \frac{\rho^k}{\langle \bar{r}, v^k \rangle}$$


$$s^k = r^{k-1} - \alpha^k v^k$$


$$t^k = As^k, \quad \omega = \frac{\langle t^k, s^k \rangle}{\langle t^k, t^k \rangle}$$


$$z^k = z^{k-1} + \omega^k s^k + \alpha^k q^k, \quad r^k = s^k - \omega^k t^k$$

enddo

```

Fig. 1. BiCGSTAB algorithm

The initial estimate is $z^0 = 0$. The condition $\|Az^k - b\| / \|b\| < \varepsilon$ for some sufficiently small ε is taken as a termination criterion.

For large fine grids, storing of coefficient matrix A can take large memory space. *E.g.*, for the $2^9 \times 2^9$ grid, the matrix would be of $2^{18} \times 2^{18}$ dimension. Full storage of this matrix with double precision requires 512 GB of memory, which is more than one node of the Uran supercomputer has.

For solving the density reconstruction problem using the BiCGSTAB method, a parallel algorithm was developed utilizing the MPI technology. The coefficient matrix A is divided by a number of horizontal bars with respect to the number of used nodes, and each bar is stored and processed on its own node. The hybrid MPI+OpenMP technology was used to parallelize the algorithm for nodes' multicore processors.

Memory efficient algorithm. In work [6], the memory-efficient algorithm for solving the density reconstruction problem in a horizontal layer was proposed and implemented for multicore processor. This algorithm is based on exploiting the Toeplitz-block-Toeplitz structure of the coefficient matrix.

The elements of matrix can be written in the following form:

$$a_{k,p,l,q} = \gamma \Delta x \Delta y \left(\frac{1}{\sqrt{(x_k - x_l)^2 + (y_p - y_q)^2 + H_1^2}} - \frac{1}{\sqrt{(x_k - x_l)^2 + (y_p - y_q)^2 + H_2^2}} \right), \quad (4)$$

where $k, l = 1..M$ are the block indices and $p, q = 1..N$ are the indices of elements inside each block.

Apparently, the matrix elements depend only on the terms $(x_k - x_l)^2 + (y_p - y_q)^2$. Note that $a_{k,p,l,q} = a_{k,p+1,l,q+1}$ and $a_{k,p,l,q} = a_{k+1,p,l+1,q}$. The former equation means that in each block each descending diagonal from left to right is constant. The latter one means that each block diagonal is constant as well. In other words, the matrix is symmetric Toeplitz-block-Toeplitz.

The matrix structure for a 6×4 grid is shown in Fig. 2. The matrix has 4×4 blocks of 6×6 dimension. Equal elements are marked by the same colors.

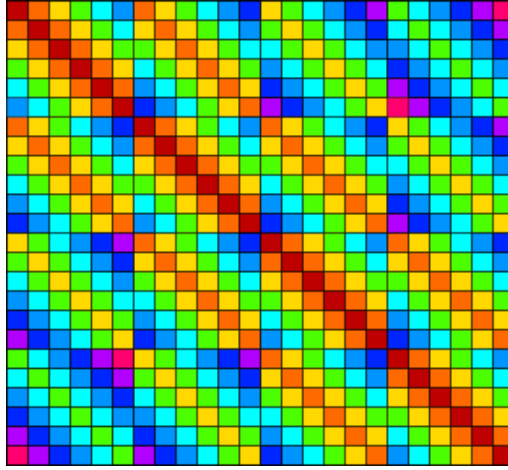


Fig. 2. Matrix structure for a 6×4 grid

The obvious way of storing this matrix is to store the first row only. Each subsequent row is obtained by double cycle shifting of elements. This method requires storing $M \cdot N$ elements, whilst full storage requires $M^2 \cdot N^2$.

In the case of a curvilinear layer, the matrix elements will have the form

$$a_{k,p,l,q} = \gamma \Delta x \Delta y \left(\frac{1}{\sqrt{(x_k - x_l)^2 + (y_p - y_q)^2 + H_{1,(p-1)M+k}^2}} - \frac{1}{\sqrt{(x_k - x_l)^2 + (y_p - y_q)^2 + H_{2,(p-1)M+k}^2}} \right), \quad (5)$$

where $H_{1,(p-1)M+k}$ and $H_{2,(p-1)M+k}$ are the values of depth of the boundary surfaces at the grid nodes. Obviously, these values vary, thus, matrix A will not be Toeplitz-block-Toeplitz.

In this work, we propose the following way to store this matrix approximately. Note that the farther the matrix element is from the main diagonal, the lower its value is. We compute the elements of the main block diagonal (*i.e.*, elements that has $p = q$) using formula (5), and the rest of the elements are replaced by their approximations computed by using formula (4) with some constant values $\overline{\overline{H_1}}$ and $\overline{\overline{H_2}}$. The mean values $\overline{\overline{H_w}} = \sum_{i=1}^{MN} H_{w,i}/MN$ can be used.

The number of stored elements will be $M(N^2 + 1)$. *E.g.*, to store the $2^{18} \times 2^{18}$ matrix, we need only 256 MB instead of 512 GB needed for full storage. Thus, to solve the problem using this algorithm, we need only one node with multicore processor. The algorithm was implemented utilizing the OpenMP technology

4 Numerical experiments

Let us demonstrate application of the constructed algorithms to solving the model density reconstruction problem on a large grid.

The model problem of reconstructing the density distribution in a curvilinear layer between the depths $\overline{H_1} = 9.5$ km and $\overline{H_2} = 11.5$ km is considered.

The boundary surfaces are shown in Fig. 3.

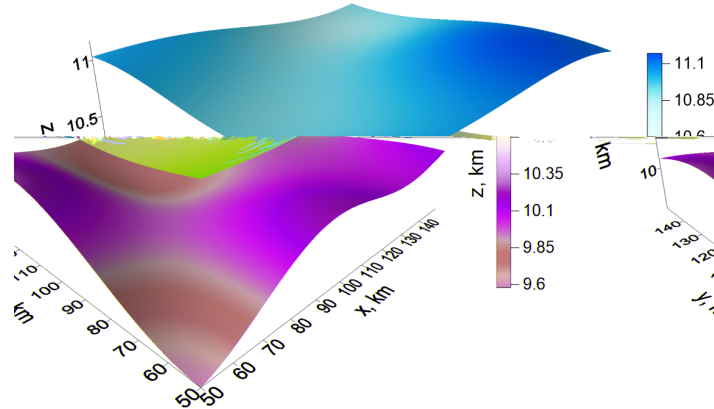


Fig. 3. Boundary surfaces

Figure 4 shows the model gravitational field. This field was obtained by solving the forward problem (1) using the model density distribution $\sigma(x, y)$.

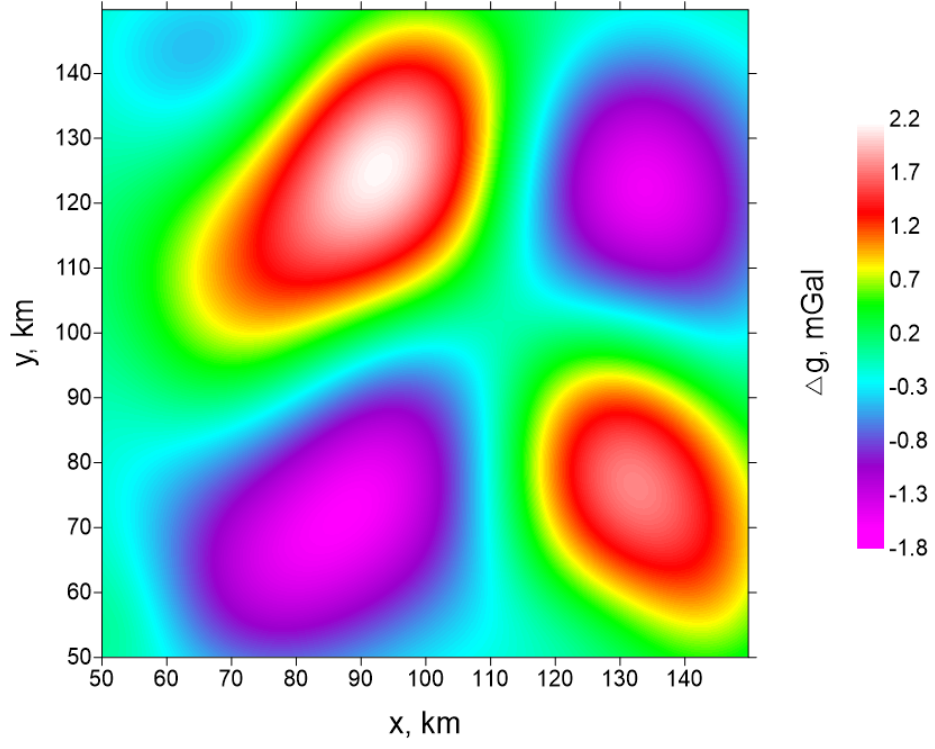


Fig. 4. Model gravitational field

Figure 5a shows the original density distribution $\sigma(x, y)$.

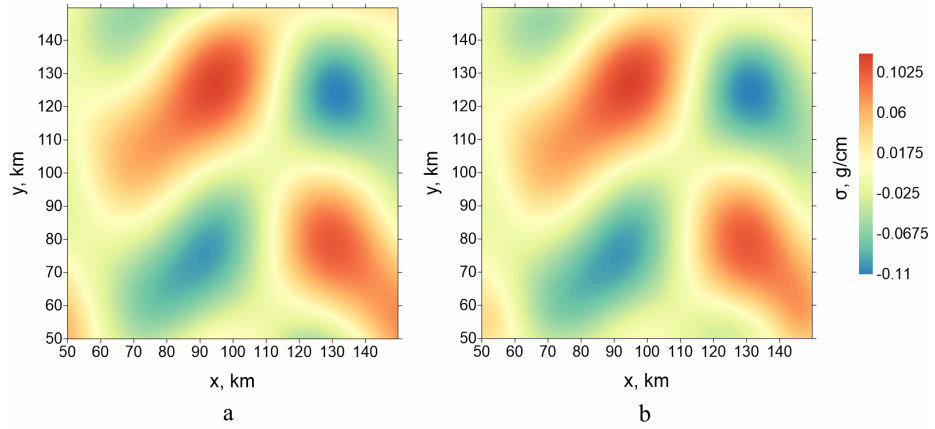


Fig. 5. Original σ (a) and reconstructed $\hat{\sigma}$ (b) density distributions

The inverse problem was solved on $2^9 \times 2^9$ grid by the BiCGSTAB algorithm and the memory efficient algorithm as described above.

Figure 5b shows the reconstructed density distribution $\hat{\sigma}$. The regularization parameter was $\alpha = 0.1$, $\varepsilon = 0.005$ was used as termination criterion.

Solution by the memory efficient algorithm has a relative error $\|\hat{\sigma} - \sigma\| / \|\sigma\|$ not greater than the error of solution by the unmodified BiCGSTAB. Both errors are lower than 0.05.

Table 1 shows the computing times for solving the problem by the BiCGSTAB method using the MPI technology with the Uran supercomputer. Each of supercomputer's nodes has Intel E5-2650 CPU and 192 GB of RAM. One core was used on each node. The coefficient matrix requires 512 GB for full storage. Thus, to solve the problem, we need at least 4 nodes.

Table 1. Computing times of BiCGSTAB algorithm

Number of nodes m	Computing time T_m , minutes	Relative speedup $S_m = T_4/T_m$
4	5.7	—
8	3	1.9
16	1.6	3.6

The last table line contains the time to solve the problem utilizing the hybrid MPI+OpenMP technology with 8 cores at each node. The computing time reduces to 20 seconds and the efficiency of this technology was 70 %.

The memory efficient algorithm requires 256 MB to store matrix; thus, to solve the problem, we need only one node with multicore processor.

Table 2 shows computing times, parallel speedup, and efficiency for solving the problem by the memory efficient algorithm using the node with two eight-core Intel E5-2650 CPUs. The parallel algorithm show good scaling.

Table 2. Computing times of memory efficient algorithm

Number of threads, m	Computing time T_m , minutes	Speedup $S_m = T_1/T_m$	Efficiency $E_m = S_m/m$
1	22.2	—	—
2	11.5	1.9	0.96
4	5.9	3.7	0.94
8	3	7.4	0.92
16	1.5	14.8	0.92

5 Conclusion

The memory efficient parallel algorithm was constructed on the basis of the Bi-CGSTAB method for solving the inverse problem of density reconstruction in a curvilinear layer. The modification is based on approximation of matrix blocks by the blocks of Toeplitz-block-Toeplitz matrix. Optimized method for matrix storage significantly reduces the memory requirements.

The parallel programs were developed for solving the inverse problem of density reconstruction in a curvilinear layer on the Uran supercomputer.

The first program is based on BiCGSTAB algorithm with full matrix storage using the hybrid MPI+OpenMP technology. The second program is based on the memory efficient algorithm with optimized matrix storage utilizing the OpenMP technology.

The model problem with synthetic data was solved, the speedup and efficiency of parallel algorithms was studied.

Acknowledgments

This work was partly supported by the Center of Excellence “Geoinformation technologies and geophysical data complex interpretation” of the Ural Federal University Program and by Ural Branch of RAS, project 18-1-1-8.

References

1. Lavrentiev, M. M., Romanov, V. G., Shishatskii, S. P.: Ill-Posed Problems of Mathematical Physics and Analysis. Translations of Math. Monographs 64. Amer. Math. Soc., Providence (1986)
2. Novoselitskii, V. M.: On the Theory of Determining Density Variations in a Horizontal Layer from Gravity Anomaly Data. *Izv. Akad. Nauk SSSR, Fiz. Zemli.* Vol. 5, 25–32 (1965)
3. Martyshko, P. S., Prutkin, I. L.: Technology of depth distribution of gravitational field sources (in Russian). *Geophysicheskii Journal.* Vol. 25 (3), 159–168 (2003)
4. Martyshko, P. S., Koksharov, D. E.: On finding the density in a layered medium using gravitational data (in Russian). *Geophysicheskii Journal.* Vol. 27 (4), 678–684 (2005)
5. 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. *Izvestiya, Physics of the Solid Earth.* Vol. 50(4), 508–513 (2014) <https://doi.org/10.1134/S1069351314040090>
6. Akimova, E. N., Martyshko, P. S., Misilov, V. E., Kosivets, R. A.: An efficient numerical technique for solving the inverse gravity problem of finding a lateral density. *Applied Mathematics and Information Sciences.* Vol. 10, Iss. 5, 1681–1688 (2016) <http://doi.org/10.18576/amis/100506>
7. Akimova, E. N., Martyshko, P. S., Misilov, V. E.: On finding a density in a curvilinear layer by biconjugate gradient type methods. *AIP Conference Proceedings.* Vol. 1863, 050009 (2017) <https://doi.org/10.1063/1.4992206>
8. Van der Vorst, H. A.: Iterative Krylov methods for large linear systems. Vol. 13. Cambridge University Press (2003)