Performance-Effective Algorithm for Solving Large-Scale Forward Gravity Problem for Elliptical Objects

Petr S. Martyshko[1,2], Igor V. Ladovskii[1], Denis D. Byzov[1], and Alexander I. Chernoskutov[1,2]

[1] Bulashevich Institute of Geophysics, Ural branch of Russian Academy of Sciences, Yekaterinburg, Russia
[2] Yeltsin Ural Federal University, Yekaterinburg, Russia

Abstract. In this paper we purpose a performance-effective algorithm for solving forward gravity problem for elliptical objects. The algorithm is subject to parallel computations, it was implemented and tested with the CUDA technology. In general, the suggested method can be applied to density distribution models of arbitrary geometry that can be triangulated.

Keywords: computational geophysics, potential fields, gravity field, spherical Earth crust models

1 Problem statement

Modern studies of the internal structure of the Earth require density model reconstructions with high resolution and accuracy. This raises questions on error estimation due to spherical shape of the Earth and creates the need of refinement of the modelling results with sphericity taken into account. Thus, there is a need for an effective tool that allows solving the forward gravity problem for spherical and elliptical density models. In this paper, we propose an algorithm based on the new representations of the triangular plate potential in space that makes it possible to effectively use the capabilities of modern computational parallelization technologies (in particular, CUDA) and, as a result, to obtain high-performance gravity field solving software.

Define the 3D density model as follows. Upper boundary S of the model (ground-air interface) is a sector of a surface of an ellipsoid of revolution (for example, Krasovsky Ellipsoid), all points located at a distance of no more than H along the inner normal to S are forming a set D (see Fig. 1). In the domain $D \subset \mathbb{R}^3$, the density distribution $\rho(p)$ is defined, $p \in D$.

The vertical component of the gravitational field intensity Δg generaded by the domain D at the outer point $q \notin D \setminus \partial D$ is determined by the integral

$$\Delta g(q) = -\gamma \frac{\partial}{\partial \mathbf{n}_q} \int_D \frac{\rho(p) dV_p}{|\mathbf{r} - \mathbf{r}_0|}, \quad (1)$$

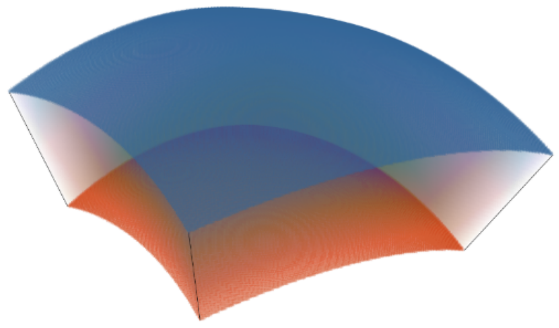

Fig. 1. Visual representation of the 3D density model

where γ is the gravitational constant, dV_p is a volume element of integration, $\mathbf{n}_q$ is the outer normal to S in the orthogonal projection of the point q to S, $\mathbf{r}$ and $\mathbf{r}_0$ are the radius vectors of the points p and q, respectively.

Assume the ellipsoid of rotation for our spherical model D has equatorial and polar radii equal to a and b. The position of the point on the surface of the ellipsoid of rotation is uniquely defined by the geodesic latitude $B \in [-\pi/2; \pi/2]$ and the longitude $L \in (-\pi; \pi]$ (except for the "poles", where longitude is not defined). As a third coordinate H we employ the distance between the given point and the ellipsoid surface; it has a plus sign if the point lies outside the ellipsoid, and the minus sign if the point lies inside it. Position of the given point in space is uniquely defined by three values: (L, B, H), where $H > -N(1 - e^2)$, $e = \sqrt{a^2 - b^2}/a$ (the eccentricity of the ellipsoid), and $N = a/\sqrt{1 - e^2 \sin^2 B}$.

2 Gravity field of a spherical model

Assume $D_{i,j,k}$ is a descrete element of D and the density of a descrete element be a constant $\rho_{i,j,k}$. Then the field Δg at point q generated by D is equals to

$$\Delta g(q) = \gamma \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} \rho_{i,j,k} G_{i,j,k}(q), \tag{2}$$

where $G_{i,j,k}(q)$ is the field at the point q of $D_{i,j,k}$ up to a coefficient γ. In order to calculate $G_{i,j,k}(q)$, we introduce in the space of a spherical model a rectangular Cartesian coordinate system $O'x'y'z'$. The center O' coincides with the center of the ellipsoid, $O'z'$ axis is ellipsoid's axis of rotation and directed from the "south pole" to the "north" (*i.e.* when $B = \pi/2$, then $z' > 0$), the $O'x'$ axis points to $(0, 0, 0)$ in (L, B, H), the axis $O'y'$ complements the system to the right-handed axis set. Equations for transition from (L, B, H) to $O'x'y'z'$ are then as follows:

$$\begin{cases} x' &= (N+H)\cos B\cos L \\ y' &= (N+H)\cos B\sin L \\ z' &= (N(1-e^2)+H)\sin B \end{cases}, \tag{3}$$

and

$$\mathbf{n} = \begin{pmatrix} \cos B\cos L \\ \cos B\sin L \\ \sin B \end{pmatrix}, \tag{4}$$

where $\mathbf{n}$ is a unit vector of the external normal to the ellipsoid at the point $(L, B, 0)$.

Integral (1) for the $G_{i,j,k}(q)$ field can not be expressed analytically. It is also problematic to calculate in numerically (with the help of the cubature formulas), since the boundaries of $D_{i,j,k}$ usually have a very complex description in the (L, B, H) or (x', y', z') coordinate system and, in order to achieve acceptable accuracy, a large number of nodes is required [1, 6]. Therefore, we propose to calculate integral (1) not for $D_{i,j,k}$, but for the polyhedron of approximation $\hat{D}_{i,j,k}$ formed by the triangulation of $D_{i,j,k}$. Thereby,

$$G_{i,j,k}(q) \approx \hat{G}_{i,j,k}(q) = -\frac{\partial}{\partial \mathbf{n}_q}\int_{\hat{D}_{i,j,k}} \frac{\rho(p)dV_p}{|\mathbf{r}-\mathbf{r}_0|}. \tag{5}$$

In formula (5), we proceed to integration over the surface, using the Ostrogradsky theorem

$$\hat{G}_{i,j,k}(q) = \left(\mathbf{n}_q, \int_{\hat{D}_{i,j,k}} \nabla_p \frac{1}{|\mathbf{r}-\mathbf{r}_0|} dV_p\right) = \left(\mathbf{n}_q, \oint_{\partial\hat{D}_{i,j,k}} \frac{\mathbf{n}_p}{|\mathbf{r}-\mathbf{r}_0|} dS\right).$$

Next, we divide the surface integral into parts along the faces of $\hat{D}_{i,j,k}$ and take into account that the outer normal $\mathbf{n}_p$ at the integration point is constant for each face

$$\hat{G}_{i,j,k}(q) = \sum_{S_{i1}\in\mathbb{S}(\hat{D}_{i,j,k})} (\mathbf{n}_q, \mathbf{n}_{i1}) \int_{S_{i1}} \frac{dS}{|\mathbf{r}-\mathbf{r}_0|}, \tag{6}$$

where $\mathbf{n}_{i1}$ is the outer normal to the face S_{i1}. Now we need to acquire a formula for the integral $\int_{S_{i1}} \frac{dS}{|\mathbf{r}-\mathbf{r}_0|}$ over a triangle (which is the gravitational potential of a triangle with a unit surface density without γ coefficient). Let $\mathbf{r}_i$ $(i = 1, 2, 3)$ be the radius vector of the vertices of the triangle $\langle p_1, p_2, p_3\rangle$; q is the field computation point; $\mathbf{a}_i = \mathbf{r}_i - \mathbf{r}_0$; $\mathbf{a}_{(j,i)} = \mathbf{a}_i - \mathbf{a}_j = \mathbf{r}_i - \mathbf{r}_j$; $\mathbf{N} = \mathbf{a}_{(i-1,i)} \times \mathbf{a}_{(i,i+1)}$ is the normal to the plane of the triangle with a length equal to its doubled area; $\mathbf{n} = \mathbf{N}/|\mathbf{N}|$ is the unit normal to the plane of the triangle; $\mathbf{A}_{(j,i)} = \mathbf{a}_{(j,i)}/|\mathbf{a}_{(j,i)}|$;

$(\mathbf{a}_i \cdot \mathbf{n})$ is the distance from the point q to the plane of the triangle (with the appropriate sign). Then

$$\int_{\langle p_1,p_2,p_3\rangle} \frac{dS}{|\mathbf{r}-\mathbf{r}_0|} = 2\left(\mathbf{n}; -\mathbf{a}_1 \arctan \frac{(\mathbf{A}_1;[\mathbf{A}_2;\mathbf{A}_3])}{1+(\mathbf{A}_3;\mathbf{A}_1)+(\mathbf{A}_1;\mathbf{A}_2)+(\mathbf{A}_2;\mathbf{A}_3)} + \right.$$
$$\left. + \sum_{i=1}^{3} \frac{[\mathbf{a}_{i-1};\mathbf{a}_i]}{|\mathbf{a}_{i-1,i}|} \operatorname{arctanh} \frac{|\mathbf{a}_{i-1,i}|}{|\mathbf{a}_{i-1}|+|\mathbf{a}_i|}\right). \tag{7}$$

Summarizing the process of calculating the vertical component of the field $\Delta g(q)$ from the model D, we have:

1. for each element $D_{i,j,k}$ using coordinates transformation (3), obtain the set of faces $\mathbb{S}(\hat{D}_{i,j,k})$ of the approximating polyhedron $\hat{D}_{i,j,k}$ (in the Cartesian system $O'x'y'z'$);
2. with the help of formulas (6), (7) calculate the vertical component of the field $\hat{G}_{i,j,k}(q)$ for the polyhedron $\hat{D}_{i,j,k}$ at the point q, the normal vector $\mathbf{n}(q)$ is calculated by formula (4);
3. assume $G_{i,j,k}(q) \approx \hat{G}_{i,j,k}(q)$ and by field summation of each $\hat{D}_{i,j,k}$ (2) acquire $\Delta g(q)$.

3 Implementation

In the presented algorithm, computation of formula (7) is taking most of the time of the running program. Indeed, in order to calculate the field of body that consists of N_b discrete elements in N_f points of space, the expression should be calculated $N = kN_bN_f$ times, where k is the amount of triangles in one element of the body. In practice, N is of order of 10^{15} for high-resolution Earth crust models (with surface area of order 1000×1000km and depth of 100km). Without using contemporary parallel technologies such computations would become senseless since it would take months or even years to run. Hence, our goal is to utilize most powerful computational devices available, which are GPUs.

Here, we assume that all the preprocessing steps have been done and as our input we have a set of triangles T (represented by triplets of points in space) that has been obtained as a result of triangulation of the discrete elements of the body. Additionally, we have a set of points Q, in which we want to calculate the normal component (or a vector) of gravitational field of the body. We are not going into details of this step because its computational complexity does not depend on N_f; thus, its becoming insignificant for large values of N_f.

The field computation requires minimum two nested cycles through Q and T. The iterations of the outer loop (over Q) are mutually independent and can be split between different computing nodes or/and GPUs. The inner loop (over T) can be expressed using map-reduce pattern: $g_q = \mathit{reduce}(\mathit{map}_q(T))$, where g is the resulting gravitational field vector the at the point q. The trick is that those operations are implemented with high efficiency in the CUDA Thrust library [5] in the form of a single function: *thrust::transform_reduce()*. All we were left to do is to implement formula (7) and pass it as a parameter.

4 Performance tests

All the tests have been performed using two regional Earth crust models that have been previously reconstructed in Bulashevich Institute of Geophysics (Ural Branch of Russian Academy of Sciences) as a result of solving inverse gravity problem [2–4]. Test models characteristics are shown in Table 1. Krasovsky Ellipsoid is taken as the reference one.

Table 1. Test models characteristics

Name	Physical size (length×width×height)	Discretization of the model (with respect to length, width, height)	Discretization of the field (with respect to length, width)
Workload 1	793km×1057km×81km	256×256×81	256×256
Workload 2	1336km×969km×81km	1336×969×81	334×243

Comparison of our most resent CPU available (i7-6900K 8 physical cores @ 3.2GHz) and our oldest GPU (GeForce GTX TITAN Black) showed that GPU outperforms CPU from 15 to 20 times even for considerably small workloads (Workload 1). So, future tests were conducted using only the GPUs.

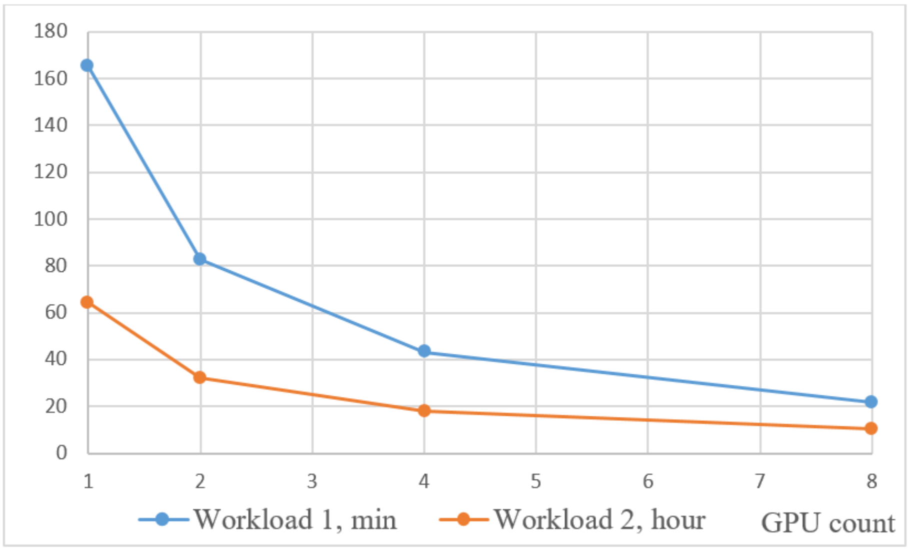

Fig. 2. Performance test results

Figure 2 shows scalability potential of the algorithm. A slight drop of performance after switching from 2 GPU to 4 and from 4 to 8 is mostly due to the fact that the load has been split between GPUs evenly, when actualy it needs to be carefully balanced (since our cluster consists of different models of GPUs[3]). The resulting field for the second workload is shown on Fig. 3.

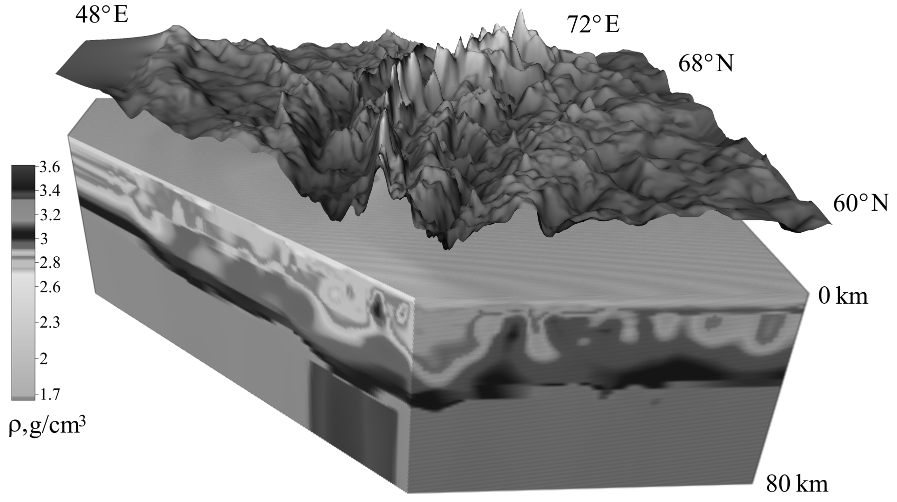

Fig. 3. The resulting field for the second workload

Acknowledgments

This work was partly supported by the Center of Excellence “Geoinformation technologies and geophisical data complex interpretation” of the Ural Federal University Program and by the The Russian Foundation for Fundamental Research (project 17-05-00916).

References

1. Heck, B., Seitz, K. A.: Comparison of the tesseroid, prism and point-mass approaches for mass reductions in gravity field modelling. Journal of Geodesy. 81, 121–136 (2007)
2. Ladovskii, I. V., Martyshko, P. S., Byzov, D. D., Kolmogorova, V. V.: On Selecting the Excess Density in Gravity Modeling of Inhomogeneous Media. Izvestiya, Physics of the Solid Earth, Vol. 53, No. 1, 130-139 (2017) `https://doi.org/10.1134/S1069351316060057`

[3] GPUs of the cluster: 2x Quadro M6000 24GB, 3x GeForce GTX TITAN Black, 3x GeForce GTX TITAN X

3. Martyshko, P. S., Ladovskii, I. V., Fedorova, N. V., Byzov, D. D., Tsidaev, A. G.: Theory and methods of complex interpretation of geophysical data. Yekaterinburg: Ural Department of Russian Academy of Sciences, 94p. (2016). ISBN 978-5-7691-2463-1.
4. Martyshko, P. S., Byzov, D. D., Ladovskii, I. V., Tsidaev, A. G.: 3D density models construction method for layered media. 15th International Multidisciplinary Scientific GeoConference SGEM 2015, www.sgem.org, SGEM2015 Conference Proceedings, Albena. Bulgaria. Book 2 Vol. 1, 425–432 (2015) https://doi.org/10.5593/SGEM2015/B21/S8.053
5. NVIDIA CUDA Toolkit Documentation. Trust.: http://docs.nvidia.com/cuda/thrust/index.html
6. Wild-Pfeiffer, F., Augustin, W. und Heck, B.: Optimierung der Rechenzeit bei der Berechnung der 2. Ableitungen des Gravitationspotentials von Massenelementen. Zeitschrift für Geodäsie. № 6 (132), 377–384 (2007)