GPU Optimized Software for
Forward and Inverse Gravity Problems Solution
for Contact Boundary between Two Layers

Alexander G. Tsidaev[1,2]

[1]*Ural Federal University, Yekaterinburg, Russia*
[2]*Bulashevich Institute of Geophysics, Yekaterinburg, Russia*

Corresponding author: amita2019@tsidaev.ru

Abstract. The paper presents methods overview and software for forward and inverse gravity problems solution for contact surfaces. Forward calculations are improved by introducing the finite elements method. This allows one to calculate the gravity field of shallow boundaries. Inversion is performed by the method of local corrections. Both methods are optimized with CUDA, which significantly increased efficiency. The software CUGRACID is distributed under open license. Usage information for forward and inverse problems solution programs is provided.

INTRODUCTION

Infinite flat layer with constant density value produces zero gravity anomaly field. Obviously, two flat layers with flat contact boundary also produce zero anomaly field. Situation is changed when the contact boundary becomes curvilinear. Masses from the lower layer wedges into the upper one and vice versa (Fig. 1). This produces the density inhomogeneity, which causes gravity anomaly to appear. This anomaly is called the gravity field of contact boundary and can be written as [1,2]

$$U(x',y') = f\Delta\sigma \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\frac{1}{\sqrt{(x-x')^2+(y-y')^2+z(x,y)^2}} - \frac{1}{\sqrt{(x-x')^2+(y-y')^2+H^2}}\right)dxdy \ . \tag{1}$$

Here, $U(x',y')$ is the gravity anomaly field of the boundary surface calculated at the point (x',y');

$\Delta\sigma = \sigma_{lower} - \sigma_{upper}$ is the density jump on the boundary, which is calculated from absolute density values σ_{lower}

and σ_{upper} of lower and upper layers, correspondingly; f is gravitational constant; $z(x,y)$ is the contact boundary of the layers with asymptote H. In practices, boundaries without asymptote may be used, too. Average value of the boundary is taken as H.

METHODS

Forward calculation of the field is performed by Eq. 1 directly. The inverse problem is to find $z(x,y)$ position by the given observed gravity field, asymptote depth, and density jump. In this case, formula (1) acts as an integral equation. It can be solved by various common methods. However, there is the method of local corrections [2, 3], which was developed especially for this kind of problem. Its iterative formula is

$$z^{n+1}(x,y) = \frac{z^n(x,y)}{1+\alpha \ Sign(\Delta\sigma)\cdot z^n(x,y)\left(U(x,y)-U^n(x,y)\right)} \ . \tag{2}$$

Here, z^n is the position of contact boundary on n-th step, U^n is the gravity field of z^n boundary, U is the observed gravity field, α is a regularizing coefficient, which should be selected small enough.
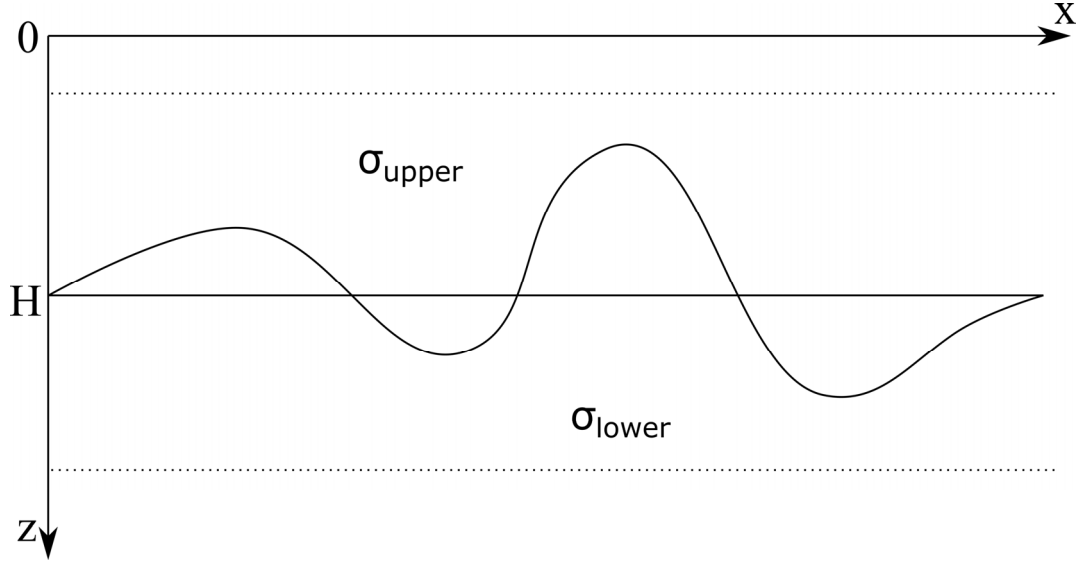

FIGURE 1. Contact boundary z(x,y) between two layers.

Formula (1) has weak singularity at the point $x = x', y = y'$ when $z(x,y) = 0$ or $H = 0$. This makes the formula inapplicable for shallow contact surfaces. To avoid this limitation, we used finite elements method. The volume between the contact boundary and its asymptote (or average value) is approximated with the prismatic elements (Fig. 2). Positions of elements are defined by discretization $x_i = \xi_i, y_j = \eta_j, 0 \leq i < N, 0 \leq j < M$. Gravity effect of each prism may be calculated as [4]

$$\Delta g_{ij}(x',y') = -(\eta - y')\cdot \ln\left(\xi - x' + R\right) - (\xi - x')\cdot \ln\left[\eta - y' + R\right] + \zeta \cdot arctg\left[\frac{(\xi - x')\cdot(\eta - y')}{\zeta \cdot R}\right]\Bigg|_{\xi_i}^{\xi_{i+1}}\Bigg|_{\eta_j}^{\eta_{j+1}}\Bigg|_{H}^{z(\xi_i,\eta_j)} \tag{3}$$

Here, $R = \sqrt{(\xi - x')^2 + (\eta - y')^2 + \zeta^2}$. Thus, the gravity field of the contact boundary may be calculated as a sum of gravity effects of the prisms

$$U(x',y') = f\Delta\sigma\sum_{ij}\Delta g_{ij}(x',y'). \tag{3}$$

The field of boundary is replaced with the field of the limited object and this eliminates need in the "real" asymptote completely.

OPTIMIZATION

For large models, the field calculation may take a significant amount of time. When this calculation is a part of iteration process (*e.g.*, for local corrections method (2)), its total consumed time increases by an order of magnitude. Author used the CUDA technology for acceleration of the algorithm execution with GPU [5]. Table 1 shows the of performance for surface grid of 256x256 points.

TABLE 1. Duration of the forward problem calculation on different hardware.

Hardware	Calculation time, s	Speed up factor
1 CPU core (Intel Xeon E5520)	3968.16 (more than 66 minutes)	1x
1 GPU (nVidia GTX580)	62.412	64x
1 GPU (nVidia GTX780 Ti)	32.711	121x
1 GPU (Tesla M2050)	27.841	143x
2 GPUs (nVidia GTX780 Ti)	16.862	235x
2 GPUs (Tesla M2050)	13.693	290x
4 GPUs (Tesla M2050)	7.528	527x
8 GPUs (Tesla M2050)	5.274	752x

Optimization is performed by splitting the calculation of components of sum (3) to different CUDA threads. Thus, each CUDA thread calculates $\Delta g_{ij}(x',y')$, and then the main thread of the current CUDA block combines these partial results into the field value $U(x',y')$ at one observation point. To get the field values at all observation points, NxM blocks should work.
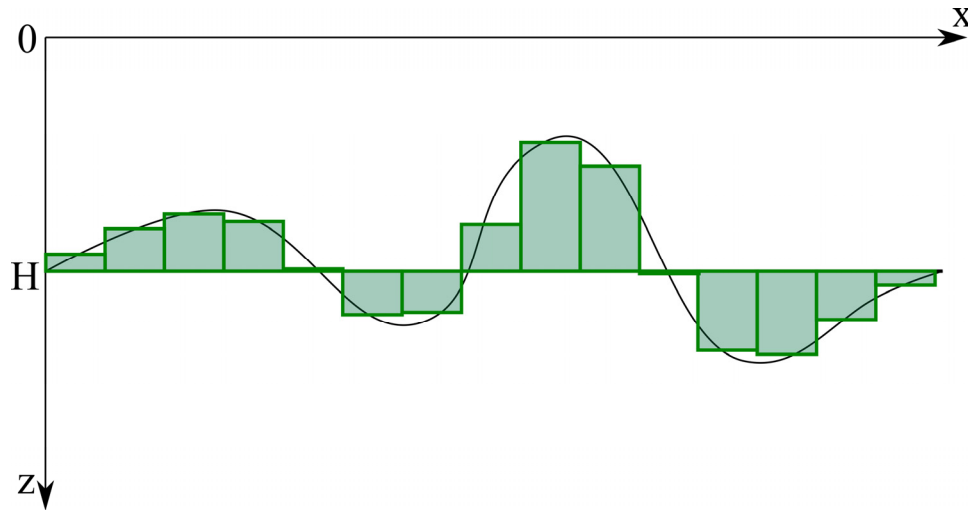

FIGURE 2. Approximation of volume between contact boundary and its asymptote with prisms.

SOFTWARE

The CUGRACID package for forward and inverse gravity problems solution includes
- *v3* program for calculation of gravity field of contact boundary;
- *lc* program for inverse problem solution with the method of local corrections and optimized forward problem algorithm.

Both programs use the CUDA for calculation. The source code is cross platform. For the compilation under OS Windows, all the required libraries (*e.g.* getopt) are provided with the source. Programs are available at https://github.com/atsidaev/cugracid. The compiled binary contains both programs. The required one should be selected as first command line argument

```
$ cugracid <v3|lc> …
```

Also, it is possible to create symbolic link to CUGRACID binary (or simply rename) with *v3* or *lc* name. In this case, it is not needed to specify required program on the command line. The *v3* program should be executed as

```
$ cugracid v3 <boundary surface filename> <delta sigma> [field filename]
```

All boundaries and fields should be in the Surfer binary GRD format. All distances and depths should be in kilometers. Density should be in g/cm³.

The *lc* program has complex command line options

```
-f FIELD
    File name of given gravity field
-s DELTA_SIGMA
    Density jump value of bounary
-b BOUNDARY
    Initial boundary position
-t ASYMPTOTE
    Depth value of asymptote of the boundary
-o RESULT
    Output file for resulting boundary
-e EPSILON
    Required accuracy (stopping condition)
-I ITERATIONS
    Iterations count
```

Options *–e* and *–i* are mutually exclusive. Option *–t* may or may not be specified if *–b* is set. Otherwise, it is required.

The program is distributed under GNU Public License v2.

REFERENCES

1. Numerov, B.V., "Interpretation of Gravitational Observations in the Case of One Contact Surface", Dokl.Akad. Nauk SSSR, no. 21, pp. 569–574 (1930)
2. Martyshko, P.S., Ladovskiy, I.V., Tsidaev, A.G., "Construction of Regional Geophysical Models Based on the Joint Interpretation of Gravitaty and Seismic Data", Izvestiya, Phys. Solid Earth, 46 (11), 931-942 (2010).
3. Prutkin, I.L., "The solution of three-dimensional inverse gravimetric problem in the class of contact surfaces by the method of local corrections", Izvestiya. Phys. Solid Earth 22 (1), 49-55 (1986)
4. Nagy, D. "Gravitational attraction of a right rectangular prism", Geophysics, 31, 362–371 (1966)
5. Tsidaev, A., CUDA Parallel Algorithms for Forward and Inverse Structural Gravity Problems", in *Proceedings of the 1st Ural Workshop on Parallel, Distributed, and Cloud Computing for Young Scientists*, CEUR Workshop Proceedings 1513, edited by Andrey Sozykin *et al.* (Ural Federal University, Yekaterinburg, Russia, 2015), pp. 50-56.