

# A Study on Parallel Computation Based on 3D Forward Algorithm of Gravity

Mao Wang, Yuanman Zheng, Changli Yao

School of Geophysics and Geoinformation Technology, China University of Geosciences, Beijing, China

Email: wangmao@cugb.edu.cn

**How to cite this paper:** Wang, M., Zheng, Y.M. and Yao, C.L. (2017) A Study on Parallel Computation Based on 3D forward Algorithm of Gravity. *International Journal of Geosciences*, 8, 1073-1079.  
<https://doi.org/10.4236/ijg.2017.89060>

**Received:** August 11, 2017

**Accepted:** September 12, 2017

**Published:** September 15, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Interpretation of geophysical material is the prospecting method. Interpretation of Gravity-magnetic data is based on data processing and inversion. When the grid is divided into several million cells, the computing task is heavy and time-consuming. In order to increase efficiency of the 3D forward modeling, the paper will adopt MPI parallel algorithm and the several processes will deal with data in the method. Finally, we can gather the result. Through comparing the result of sequence algorithm with the result of MPI parallel algorithm, we can see the result is the same. When the number of processes is 2 to 8, the speed-up ratio is 1.97 to 5. The MPI parallel algorithm is very efficient.

## Keywords

Gravity, 3D Forward Modeling, Parallel Algorithm, MPI

---

## 1. Introduction

Interpretation of Gravity-magnetic data is based on data processing and inversion. Forward modeling is important and difficult in the interpretation of geophysical data [1] [2] [3], so we spend much time to do research on the problem and get much achievement. With the development of demand for geophysical data interpretation, 2D gravity-magnetic forward modeling can't meet the requirement of identifying the geological structure, so we need to develop the gravity 3D forward modeling [4] [5] [6]. 3D forward modeling is very difficult and the increased computation task makes it more difficult. Many researchers do research on the problem and raise treatment method. Changli Yao adopts the grid separation and storage technology, the method decreases the computation tasks and storage [7] [8], so it gets better effect. In the method we need to sup-

plement many observation points. Zhaoxi Chen adopts the GPU parallel technology to accelerate the processing speed and it gets good effects [9] [10]. The paper adopts the MPI parallel technology to accelerate the processing speed of 3D forward modeling. It gets good effects.

## 2. Gravity 3D Forward Modeling

We divide the source field underground into many cuboids (**Figure 1**). Forward modeling is the basic work of inversion [11] [12]. By means of traditional method that deals with the forward data one by one, the computation tasks will be huge and it will take a long time. The 3D density model forward equation illustrates the forward status. In the **Figure 1**, according the known equation we can get the gravity anomaly of the P(x,y,z) which is the *j*th cell.

$$\Delta g_j(x, y, z) = G\sigma_j \sum_{l=1}^2 \sum_{m=1}^2 \sum_{n=1}^2 (-1)^{l+m+n} \left\{ (x_l - x) \ln[(y_m - y) + R_{lmn}] + (y_m - y) \ln[(x_l - x) + R_{lmn}] + (z_n - z) \tan^{-1} \frac{(z_n - z)R_{lmn}}{(x_l - x)(y_m - y)} \right\} \quad (1)$$

G is gravitational constant,  $\sigma_j$  is the density of the *j*th cell.

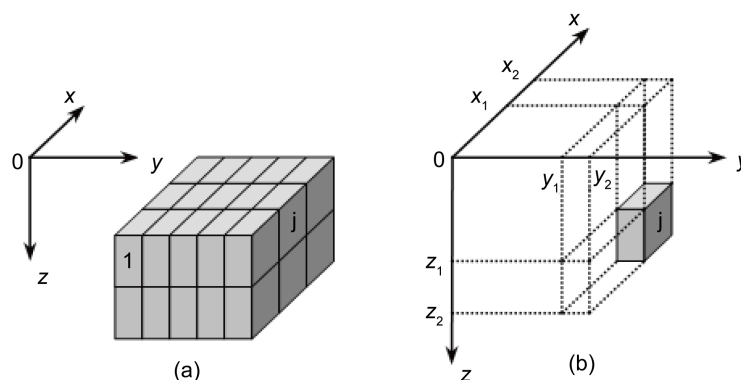
$$R_{lmn} = \sqrt{(x_l - x)^2 + (y_m - y)^2 + (z_n - z)^2} \quad (2)$$

By analyzing the equation, we know that in order to compute the gravity anomaly that the *j*th cell produces at the point P(x,y,z), we need to do so many computations: 16 times log, 8 times inverse trigonometric functions, 64 times multiplications, 8 times divisions, 8 times evolutions, 120 times additions or subtractions. The computation task is so huge for a cell at a point. If the number of the model cell is big, the total computation task will be very huge. The total computation is very huge, so the huge computation is a bottleneck problem.

## 3. MPI Parallel Algorithm

### 3.1. Introduction of MPI

MPI realize the data broadcasting, data sending, data receiving and data synchronization [13]. MPI support several data type, including complex. Although



**Figure 1.** (a) rectangular model, (b) a cell of the model.

the MPI program is code, the program is different and it can be identified by process ID. Different processes have different computation tasks.

In the MPI program, every process has all the variables and functions. The variables and functions have the same name, but the data in the variable is always not the same. If the process needs to know the data of other processes, it needs to communicate. Every process can allocate different memory for a pointer. When the task is different for a process in the 3D gravity forward algorithm, it needs to allocate memory more flexibly [14] [15].

### 3.2. The MPI Parallel Algorithm Based on 3D Gravity Forward

The gravity forward modeling is that the sum of the anomaly that every cuboid in the model produces. The serial program needs loop computation for 5 times. For the observing point, we need to calculate the point of x direction and y direction. For every point, we need to calculate the sum of anomaly that the cell produces for x direction, y direction and z direction. The pseudo-code is following.

```

For(i=1;i<nx_obs;i++) //observing x direction, nx_obs points
For(j=1;j<ny_obs;j++) //observing y direction, ny_obs points
For(l=1;l<nz_model;l++) //cells along the z direction, the number of cells
along z direction is nz_model
For(m=1;m<nx_model;m++) //cells along the x direction, the number of cells
along x direction is nx_model
For(n=1;n<ny_model;n++) //cells along the y direction, the number of cells
along y direction is ny_model
forwarddata(I,j) =forwarddata (I,j)+forward(I,j,l,m,n)

```

We analyze the 3d gravity forward algorithm and find that the algorithm needs loop computation for 5 times. The loop is the main work for parallel computation. We can divide the `nx_obs` into several parts for processes. The computation task that every process in parallel algorithm needs to do is less than that in the serial algorithm. Firstly, we need to divide the computation task into several parts.

```

count = nx_obs / size;           // observing along x direction, nx_obs
points. Size is the number of processes
segment = nx_obs %size;         //when tasks is allocated for every process
equally, segment is the number of remainder
for (i = 0; i < size; i++)
{
    if (i < segment)
        parr[i] = count + 1;
    else
        parr[i] = count;
}

```

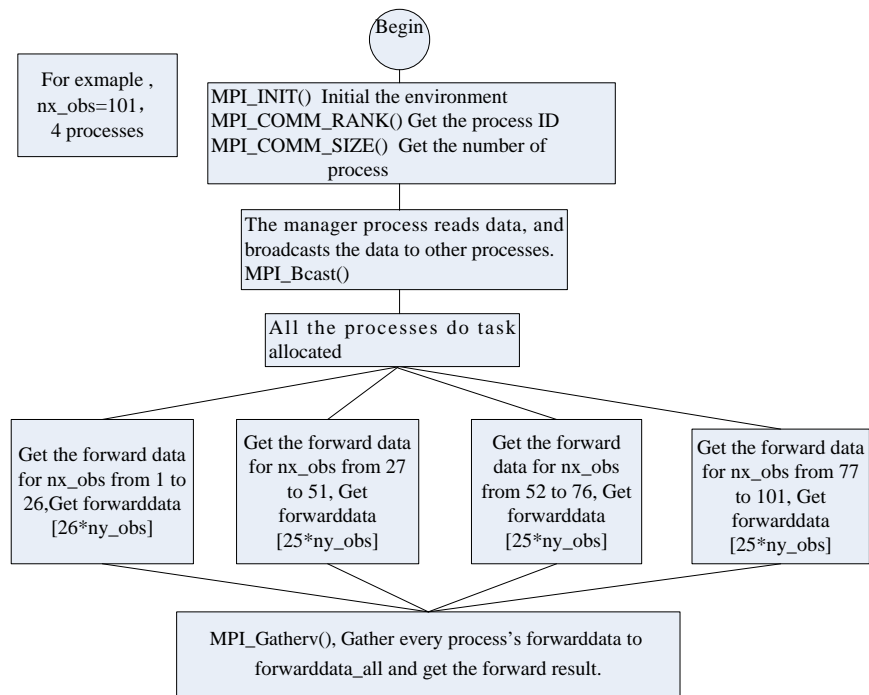
`Parr` is an array. The array records the computation task that every process needs to do. For example, `nx_obs` is 101, `size` is 4. We know `parr[0] = 26`,

$parr[1] = 25, parr[2] = 25, parr[3] = 25$ . The processes need to calculate 1 - 26, 27 - 51, 52 - 76, 77 - 101 row data, so every process do smaller task in the MPI algorithm than that in the serial algorithm.

The method uses MPI\_Gatherv function to gather result. In the parallel algorithm, the program allocates the computation tasks, and every process only has his data result. If we don't gather the data result, the process can write a file for the result data. For example the algorithm has 4 processes, we can get 4 files of data. If we want to get the final data result, we need merge 4 files manually. It is very difficult. MPI\_Gatherv (forwarddata, count, MPI\_FLOAT, forwarddata\_all, rcounts, displs, MPI\_FLOAT, 0, MPI\_COMM\_WORLD). The function MPI\_Gatherv gathers the forwarddata to forwarddata\_all and gathers the data from different process to one process. The length of the data can not the same. It is very useful.

**Figure 2** is gravity 3D forward MPI algorithm flow chart. The program allocates the computation tasks to every process, and process 0 is the manager process. It reads the parameter file, and allocates the computation tasks to every process. It broadcasts the parameter to processes, and gathers the data result from processes. Finally, it writes the result file. Other processes receive the parameter, and do the computation tasks. They send the data result to the manager process. At the same time, manager process also do computation task. We will introduce the performing process.

- (1) Initial the MPI environment, MPI\_INIT(). The manager process reads the model data and broadcasts the data to other process, MPI\_Bcast().
- (2) The manager process allocates the computation tasks according to the task



**Figure 2.** Gravity 3D forward MPI parallel flow chart.

allocate **Table 1**. Processes do the tasks separately and get the forward data for every observed point.

(3) After all the processes have done the tasks, the manager process gathers the data.

(4) The manager process writes the data file and finalize the MPI environment, MPI\_FINALIZE().

The MPI parallel environment

OS: windows 7

CPU: intel core i7 3.2 Ghz support 8 processes

Memory: 8 GB

Develop language: c

Compiler:visual studio 2010

Parallel environment: mpich2

Command: Mpiexec -np N ./gstd\_forward N is the number of processes.

## 4. Result and Discussions

The grid of the model is  $101 \times 101 \times 30$ . The model is 101 columns, 101 rows and 30 layers. The distance along x direction between 2 neighbor points is 10 m, the distance along y direction between 2 neighbor points is 10 m, the distance along z direction between 2 neighbor points is 2 m. The observed points are  $101 \times 101$  on the ground.

### 4.1. Validate of the Forward Result

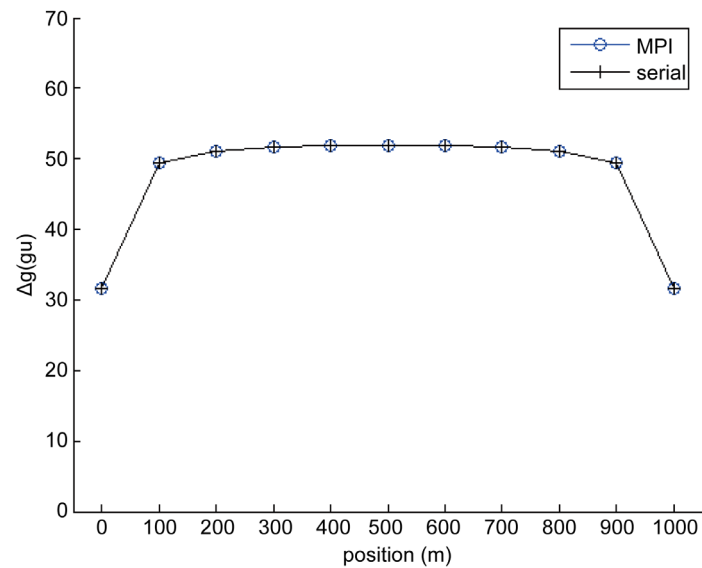
The parallel computation is based on the serial program. According to the characteristics, that process does computation task separately. It distributes the tasks to all processes and never changes other algorithm, so the parallel computation result is the same with the serial program's result. The study compares the forward result of two programs, the result is the same. We choose a line to draw a picture. In **Figure 3**, x axis is position of the observed point, y axis is the  $\Delta g$  forward data, dot is the MPI result and plus is the serial algorithm result. We can see that the forward data in 6th row is the same, so it proves the validity of the program.

### 4.2. Discussions of Parallel Efficiency

Analysis of the **Table 2** shows that the efficiency is the highest when the number of processes is 2. The speedup changes slightly and the efficiency declines when

**Table 1.** Task allocated Table.

Process ID	Computation task
0	1 - 26
1	27 - 51
2	52 - 76
3	77 - 101



**Figure 3.** Forward result figure in 6<sup>th</sup> row.

**Table 2.** 3D gravity forward parallel algorithm time used statistics.

The model of program	The amount of processes	The amount of the rows distributed for process	The size of 3D grid	The running time of program (minutes)	Parallel speedup	Parallel efficiency
Serial program	1	101	101 × 101 × 30	65	empty	empty
Parallel program	2	51, 50,	101 × 101 × 30	33	1.97	0.99
	4	26, 25, 25, 25	101 × 101 × 30	20	3.25	0.81
	8	13, 13, 13, 13, 13, 12, 12, 12	101 × 101 × 30	13	5	0.62

the number of processes changes from 4 to 8. The communication of the processes occupy much time when the number of processes increases from 4 to 8. We can see that the effect of parallel algorithm is very obvious. When the number of processes is 8, we can save 52 minutes compared to sequential algorithm.

## 5. Conclusion

The computation of 3D gravity forward for relatively big grid spends much time and the forward algorithm is called for dozens of times, so the key to the problems is parallel computation. The study realizes the parallel algorithm for 3D gravity forward in the MPI environment. The algorithm is proved correct and efficient. The study lays the foundation of the parallel computation for 3D gravity inversion.

## Funding

The paper is supported by China university of Geosciences (Beijing) basic research funding (2-9-2017-096).

## References

- [1] Yang, W.C., Shi, Z.Q., Hou, Z.Z., *et al.* (2001) Discrete Wavelet Transform and

- Gravity Anomaly Decomposition. *Chinese Journal of Geophysics*, **44**, 534-541.
- [2] Li, J. and Zhou, Y.X. (2000) Principle of Multi-Scale Gravity-Magnetic Field. *Global geology*, **19**, 379-391.
- [3] Yang, Y.S., Liu, T.Y. and Li, Y.Y. (2005) The Three-Dimensional Visualization Inversion of the Gravity-Magnetic Field for Arbitrary Shape by Means of Numerical Integration. *Geology and Prospecting*.
- [4] Lin, Z.M. and Chen, S.Q. (1996) 3D Gravity-Magnetic Interaction Interpretation and Separation of Regional and Local Anomalies. *Chinese Journal of Geophysics*, **39**, 705-711.
- [5] Yao, C.L., Zheng, Y.M. and Zhang, Y.W. (2007) 3D Gravity-Magnetic Inversion by Means of Random Subdomain Method. *Chinese Journal of Geophysics*, **50**, 1576-1583.
- [6] Yao, C.L., Hao, T.Y., et al. (2003) The High Speed Computation and Storage Method for 3D Gravity-Magnetic Genetic Inversion Algorithm. *Chinese Journal of Geophysics*, **46**, 252-258
- [7] Zheng, Y.M. (2011) The Research of 3D Gravity-Magnetic Real-Time and Visualization. PHD Paper, China University of Geosciences, Beijing.
- [8] Yao, C.L. and Guan, Z.N. (1997) Computation of Magnetic Gradients Due to Three-Dimensional Bodies. *Science in China Series D: Earth Sciences*, **40**, 293-299.
- [9] Zhang, S. (2009) High-Performance Computing, CUDA. China WaterPower Press, Beijing.
- [10] Chen, Z.X., Meng, X.H., Guo, L.H., Liu, G.F. (2012) GPU Parallel High Speed Computation for 3D Gravity and Gravity Gradient Forward and Inversion Algorithm.
- [11] Yao, C.L. and Zheng, Y.M. (2002) Dynamic Array Optimization for 3D Gravity-Magnetic Genetic Inversion Algorithm. *Computing Techniques for Geophysical and Geochemical*, **24**, 240-245.
- [12] Shen, N.H. and Guan, Z.N. (1985) Magnetic Prospecting. Geological Publish House, London.
- [13] Du, Z.H. (2001) Parallel Computation Technology.
- [14] Zhang, W.S. and Xue, W. (2009) Parallel Programming with MPI. Tsinghua University Press, Beijing.
- [15] Wang, M., Tan, H.D., Lin, C.H., et al. (2015) A Study on Parallel Computation Based on Finite element Forward Modeling of 2D Magnetotelluric. *International Journal of Geosciences*, **6**, 863-868. <https://doi.org/10.4236/ijg.2015.68070>

**Submit or recommend next manuscript to SCIRP and we will provide best service for you:**

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact [ijg@scirp.org](mailto:ijg@scirp.org)