Computing a gravity anomaly

C. Thieulot

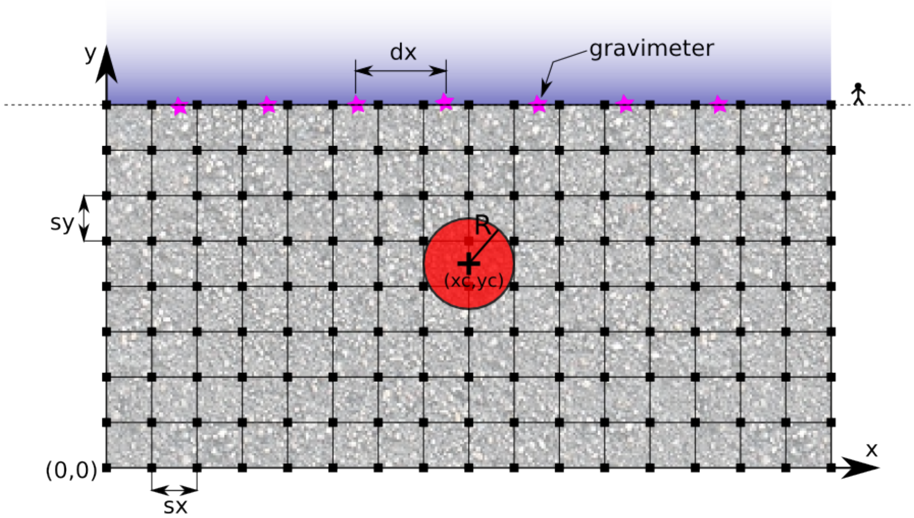
September 28, 2016

After three weeks of file opening, array allocating and do-looping, it is time we apply these new skills to a more concrete geophysical problem: gravimetry. Before you proceed further, please read :

http://en.wikipedia.org/wiki/Gravity_anomaly

http://en.wikipedia.org/wiki/Gravimeter

Before you start coding, read and re-read this document thoroughly. Subroutines are provided to you and should **not** be modified.



Let us consider a vertical domain $Lx \times Ly$ where $L_x = 1000km$ and $Ly = 500km$. This domain is discretised by means of a grid which counts $nnx \times nny$ nodes. This grid then counts $ncellx \times ncelly = (nnx - 1) \times (nny - 1)$ cells.

(1) declare in your program the nnx, nny, Lx, Ly variables and set them to (self-chosen) meaningful values.

(2) the total number of nodes is stored in np and the total number of cells in $ncell$. Compute np and $ncell$. The horizontal spacing between nodes is sx and the vertical spacing is sy. Compute sx, sy.

(3) Create the **one-dimensional** arrays $xgrid$, $ygrid$ to store the $x-$ and $y-$coordinates of **all** the nodes. Write a double do-loop to fill these arrays. (See **Appendix**) Use the subroutine *write_two_columns* to create the file *grid_init.dat* and visualise it with gnuplot (See Fig. 1).

(4) Create the one-dimensional arrays $xcgrid$ and $ycgrid$ which will store the coordinates of the centers of the cells. Write another double do-loop to fill these arrays. Use the subroutine *write_two_columns* to create the file *gridc_init.dat* and visualise it with gnuplot (See Fig. 2).

Assume that this domain is filled with a rock type which mass density is given by $\rho_1 = 3000kg/m^3$, and that there is a circular inclusion of another rock type ($\rho_2 = 3200kg/m^3$) at location (xci, yci) of radius rci. The density in the system is then given by

$$\rho(x, y) = \begin{cases} \rho_2 \text{ inside the circle} \\ \rho_1 \text{ outside the circle} \end{cases}$$

We will then use an array rho to store the density of the material in each cell.

(5) allocate the array rho which contains the density of the material present in all cells. Fill the array as follows: if the center of the cell is within the inclusion, the density of the whole cell is set to ρ_2, otherwise it is set to ρ_1. Use the *write_three_columns* subroutine to output this array and visualise it with gnuplot (See Fig. 3).

Let us now assume that we place $nsurf$ gravimeters at the surface of the model. These are placed between coordinates $x = 0$ and coordinates $x = Lx$. We will use the arrays $xsurf$ and $ysurf$ to store the coordinates of these locations.

(6) compute the spacing dx between the gravimeters as a function of $nsurf$ and Lx

(7) allocate the $xsurf$ and $ysurf$ arrays and store the $x-$ and $y-$coordinates of the gravimeters in them. Use the subroutine *write_two_columns* to create the file *surf_init.dat*.

(8) use gnuplot to visualise simultaneously the nodes, the cell centers and the surface points (See Fig. 4)

At any given point (x_i, y_i) in a 2D space, one can show that the gravity anomaly due to the presence of a circular inclusion can be computed as follows:

$$g(x_i, y_i) = 2\pi G(\rho_2 - \rho_0)R^2 \frac{y_i - y_c}{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{1}$$

where R is the radius of the inclusion, (x_c, y_c) are the coordinates of the center of the inclusion, and ρ_0 is a reference density.

However, the general formula to compute the gravity anomaly at a given point (x_i, y_i) in space due to a density anomaly of any shape is given by:

$$g(x_i, y_i) = 2G \int\int_\Omega \frac{\Delta\rho(x, y)(y - y_i)}{(x - x_i)^2 + (y - y_i)^2} dxdy \tag{2}$$

where Ω is the area of the domain on which the integration is to be carried out. Furthermore the density anomaly can be written : $\Delta\rho(x, y) = \rho(x, y) - \rho_0$. We can then carry out the integration for each cell and sum their contributions:

$$g(x_i, y_i) = 2G \sum_{ic=1}^{ncell} \int\int_{\Omega_e} \frac{(\rho(x, y) - \rho_0)(y - y_i)}{(x - x_i)^2 + (y - y_i)^2} dxdy \tag{3}$$

where Ω_e is now the area of a single cell. Finally, one can assume the density to be constant within each cell so that $\rho(x, y) \to \rho(ic)$ and $\int\int_{\Omega_e} dxdy \to sx \times sy$ and then

$$g(x_i, y_i) = 2G \sum_{ic=1}^{ncell} \frac{(\rho(ic) - \rho_0)(y(ic) - y_i)}{(x(ic) - x_i)^2 + (y(ic) - y_i)^2} s_x s_y \tag{4}$$

We will then use the array $gsurf$ to store the value of the gravity anomaly measured at each gravimeter at the surface.

(9) Allocate the array and use a double do-loop to compute the gravity anomaly at each gravimeter using the above formula. The outer loop will concern the surface points while the inner loop will concern the gravity calculation. Use $\rho_0 = 3000 kg/m^3$ to start with.

(10) Use the *write_two_columns* subroutine to output the $xsurf$ and $gsurf$ arrays in the file *gravity.dat*

(11) We will now proceed to benchmark our calculation by plotting our results against the theoretical curve given by equation (1). In gnuplot we enter the following:

```
> plot 'gravity.dat' , 2*6.6738480e-11*pi* 50000**2 *200/( (x-500e3)**2 + (250e3)**2  ) * 2.5e5
```

(This is in the case where xci=500km, yci=250km). It should then look like Fig. 5.

To go further I expect all the groups to have come so far. Hereafter are listed a few more questions for groups with time on their hands or curious students.

- explore the effect of the size of the inclusion on the gravity profile.
- explore the effect of the ρ_0 value.
- explore the effect of the grid resolution.
- measure the time that is required to complete task 9 by means of the *cpu_time* subroutine (google it). How does this time vary with nsurf ? how does it vary when the grid resolution is doubled ?
- Assume now that $\rho_2 < \rho_1$. What does the gravity profile look like ?
- what happens when the gravimeters are no more at the surface of the Earth but in a satellite ?
- if you really can't get enough, redo the whole exercise in 3D...

A quick reminder about gnuplot If you wish to plot a simple 2D figure whose values are stored in a two-colum file (ex. Figs. 1,2,4):

```
> plot 'file.dat' u 1:2 w lp t 'title'
```

If you wish to plot a simple 3D figure whose values are stored in a two-colum file (ex. Fig. 3):

```
> splot 'file.dat' u 1:2:3 w lp t 'title'
```

To export to the plots to a eps or pdf file, use the script given in appendix B.1 of the syllabus.

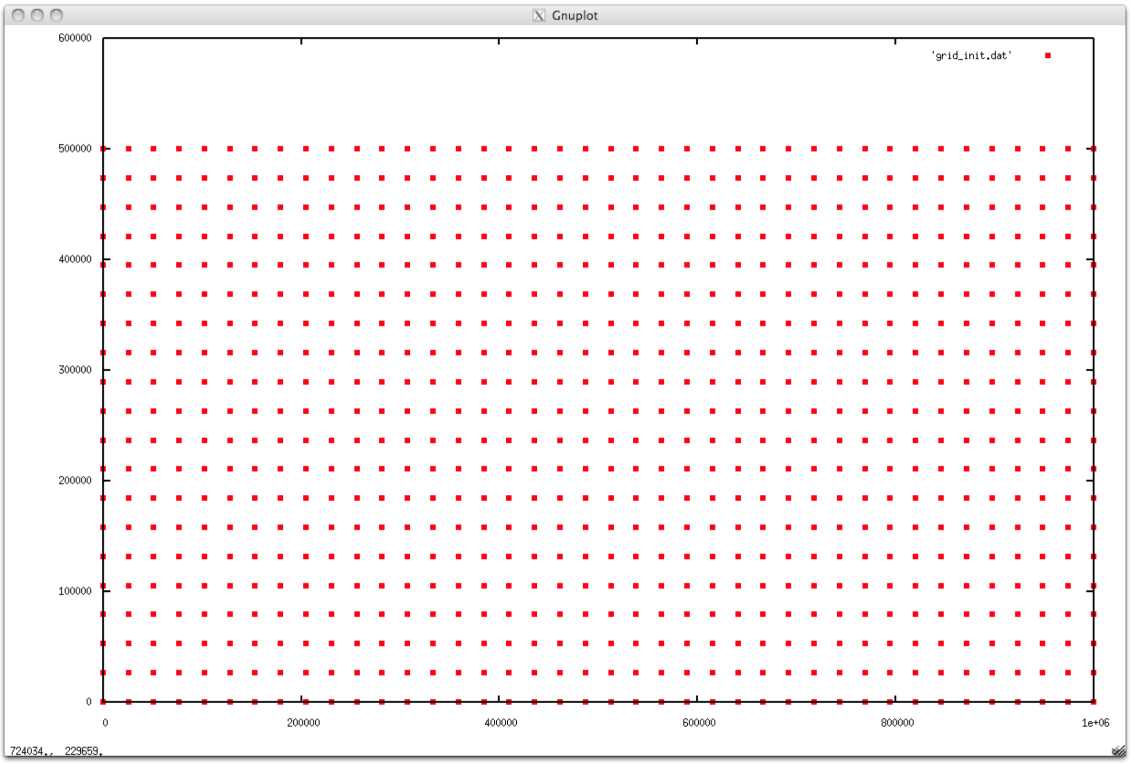


Fig1: example of a 40x20 grid (nodes)

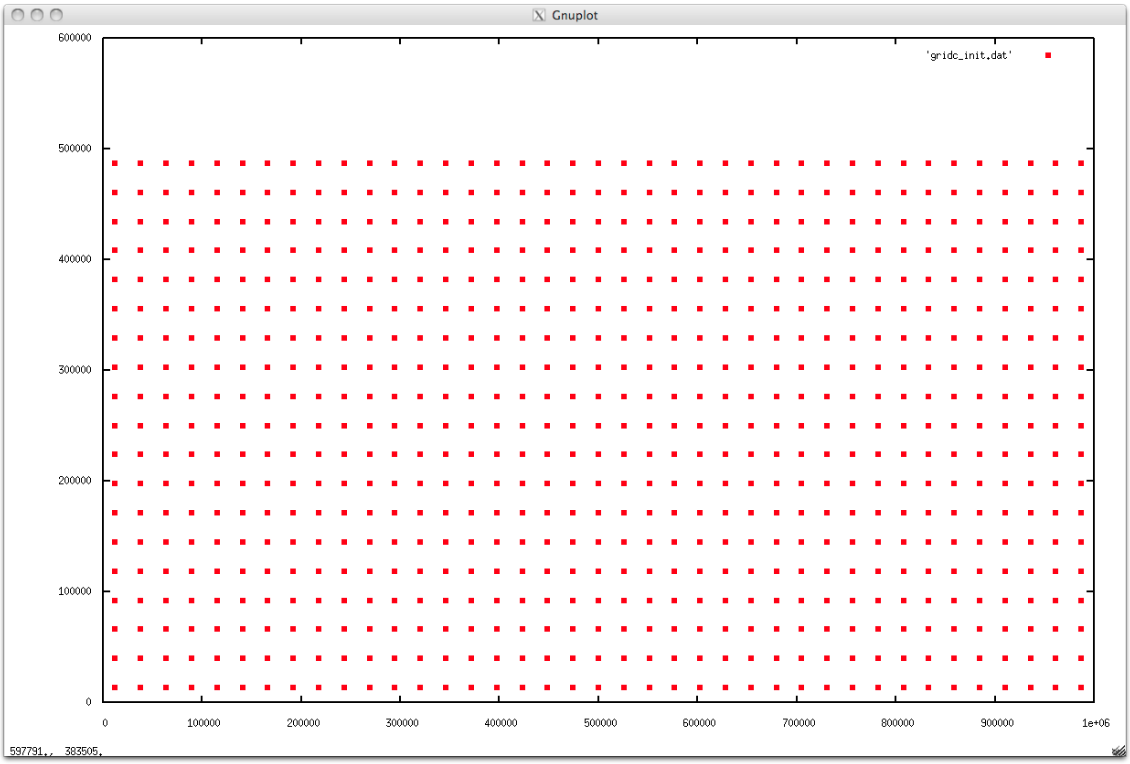


Fig2: example of a 40x20 grid (centers of cells)

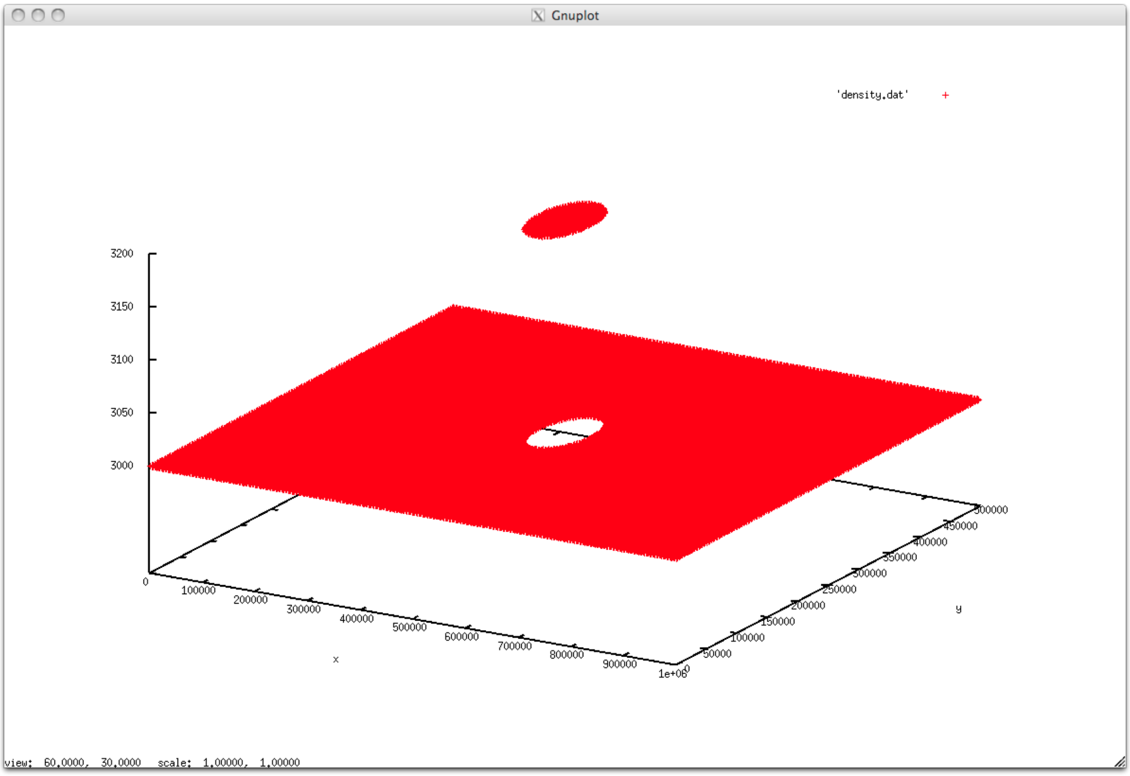


Fig3: example of the density field for a 200x100 grid

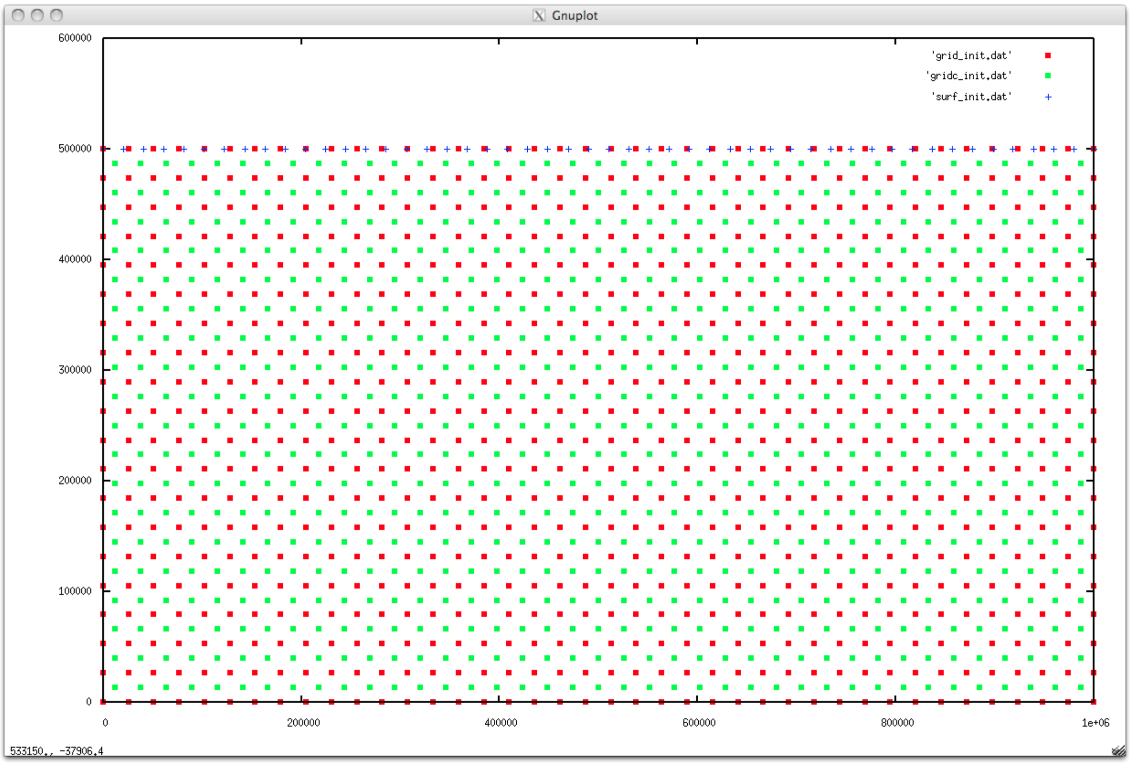


Fig4: example of a 40x20 grid with nodes, centers of cells and gravimeters locations

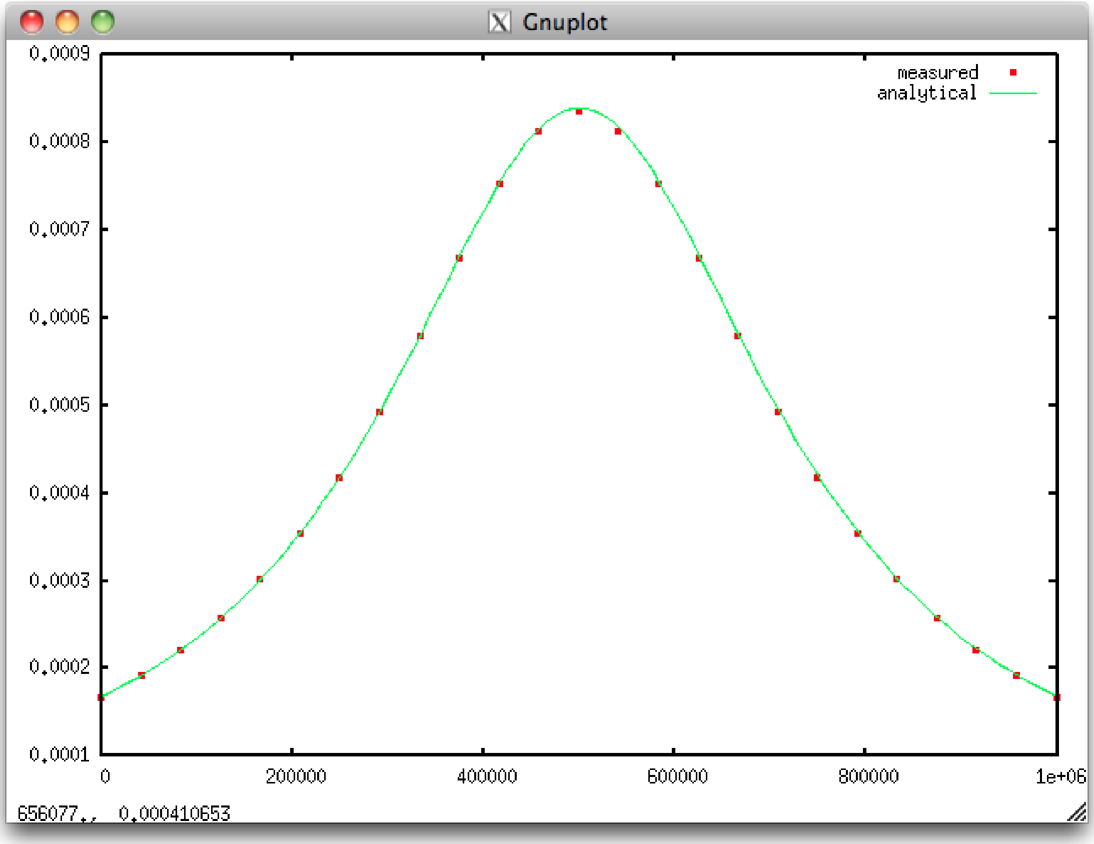
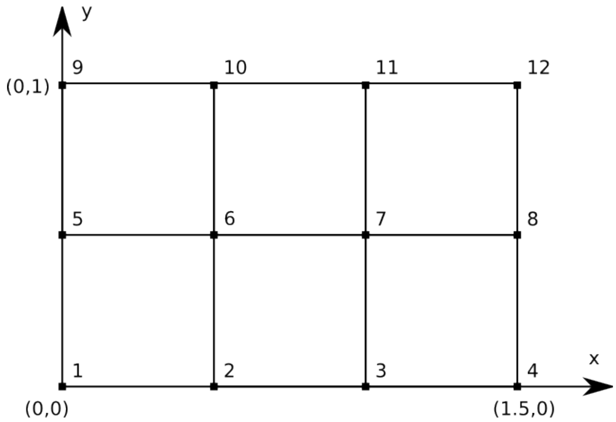


Fig5: Example of an analytical gravity profile obtained with Eq. (1) alongside measurements obtained with Eq. (4)

Appendix

Let us assume $Lx = 1.5$, $Ly = 1$, $nnx = 4$, $nny = 3$ as shown on the figure hereunder:



In this particular case, the arrays $xgrid$ and $ygrid$ should look like this:

$$xgrid = (0, 0.5, 1.0, 1.5, 0, 0.5, 1.0, 1.5, 0, 0.5, 1.0, 1.5)$$

$$ygrid = (0, 0, 0, 0, 0.5, 0.5, 0.5, 0.5, 1.0, 1.0, 1.0, 1.0)$$

Indeed, the coordinates of node 7 are $xgrid(7) = 1.0$ and $ygrid(7) = 0.5$.