Example: Calculate the gravity gradient tensor from a DEM file

Leonardo Uieda

April 7, 2011

This document intends to demonstrate how to calculate the gravity gradient tensor (GGT) due to topographic masses using tesseroids. To do that we need:

1. A DEM file with lon, lat, and height information;

2. Assign correct densities to continents and oceans (we'll be using a little Python for this);

3. Convert the DEM information into a tesseroid model;

4. Calculate the 6 components of the GGT;

5. Make nice color plots of the results (we'll be using Python for this too);

The file run_example.sh is a small shell script that executes all the above (we'll be looking at each step in more detail):

run_example.sh

```
1  #!/bin/bash
2
3  # First, insert the density information into the DEM file using a Python script
4  python dem_density.py dem−10min.xyz > dem−10min−dens.xyz
5
6  # Now, use tessmodgen to create a tesseroid model
7  tessmodgen −v −s0.166666667/0.166666667 −z0 < dem−10min−dens.xyz > dem−10min−mod.txt
8
9  # Calculate the GGT
10 tessgrd −r−60/−45/−30/−15 −b50/50 −z250e03 | \
11 tessgxx dem−10min−mod.txt −lgxx.log | tessgxy dem−10min−mod.txt −lgxy.log | \
12 tessgxz dem−10min−mod.txt −lgxz.log | tessgyy dem−10min−mod.txt −lgyy.log | \
13 tessgyz dem−10min−mod.txt −lgyz.log | tessgzz dem−10min−mod.txt −lgzz.log > dem−10min−
       ggt.xyz
14
15 # Use another Python script to make some nice plots
16 python mkplots.py
```

1 Python

Python is a modern programming language that is very easy to learn and extremely productive. We'll be using it to make our lives a bit easier during this example but it is by no means a necessity. The same thing could have been accomplished with Unix tools and the Generic Mapping Tools (`http://www.soest.hawaii.edu/gmt`) or other plotting program.

If you have interest in learning Python we recommend the excelent video lectures in the Software Carpentry (`http://software-carpentry.org`) course. There you will also find lectures on various scientific programming topics. I strongly recommend taking this course to anyone who works with scientific computing.

2 The DEM file

For this example we'll use ETOPO1 for our DEM. The file dem-10min.xyz contains the DEM as a 10'
grid. Longitude and latitude are in decimal degrees and heights are in meters. This is what the DEM
file looks like (first few lines):

<div align="center">dem–10min.xyz</div>

```
 1  # This is the DEM file from ETOPO1 with 10' resolution
 2  # points in longitude: 151
 3  # Columns:
 4  #  lon           lat         height(m)
 5  −65.000000  −10.000000     157
 6   −64.833333  −10.000000     168
 7   −64.666667  −10.000000     177
 8   −64.500000  −10.000000     197
 9   −64.333333  −10.000000     144
10   −64.166667  −10.000000     178
```

Notice that you can include comments in the files by starting a line with #. Figure 1 shows the DEM
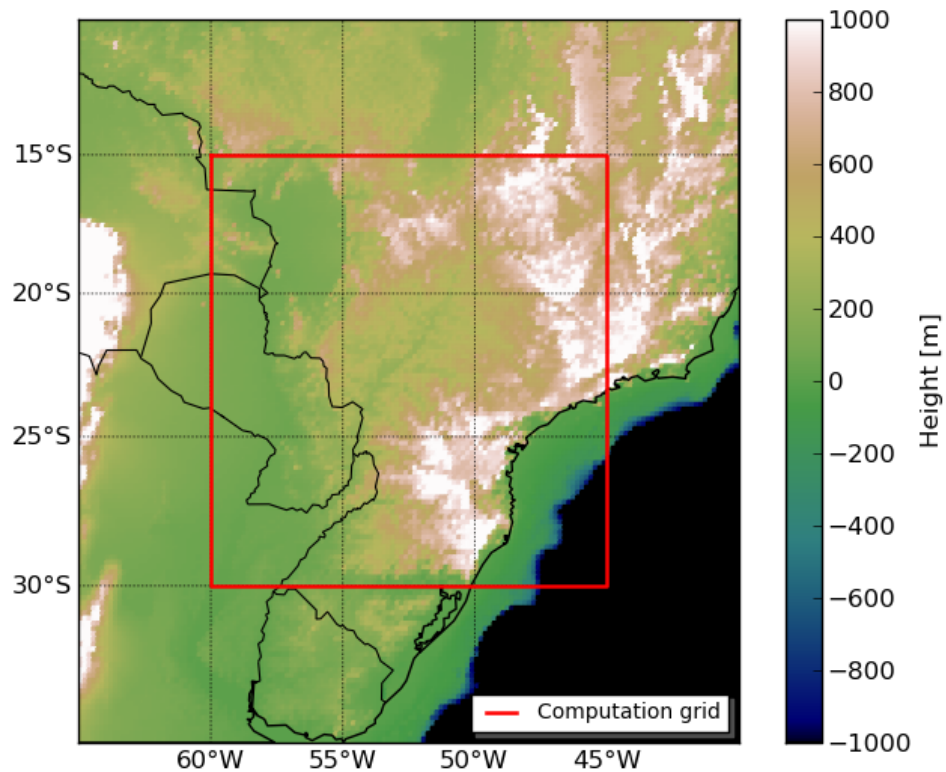ploted in pseudocolor. The red rectangle is the area in which we'll be calculating the GGT.

Figure 1: The ETOPO1 10" DEM of the Paraná Basin, southern Brasil.

3 Assigning densities

Program tessmodgen allows us to provide the density value of each tesseroid through the DEM file. All
we have to do is insert an extra column in the DEM file with the density values of the tesseroids that

will be put on each point. This way we can have the continents with 2.67 $g.cm^{-3}$ and oceans with 1.67 $g.cm^{-3}$. Notice that the density assigned to the oceans is positive! This is because the DEM in the oceans will have heights bellow our reference (h=0km) and tessmodgen will automatically invert the sign of the density values.

We will use the Python script dem_density.py to insert the density values into our DEM and save the result to dem-10min-dens.xyz:

run_example.sh

```
3  # First, insert the density information into the DEM file using a Python script
4  python dem_density.py dem−10min.xyz > dem−10min−dens.xyz
```

If you don't know Python, you can easily do this step in any other language or even in Excel. This is what the dem_density.py script looks like:

dem_density.py

```
1  """
2  Assign density values for the DEM points.
3  """
4  import sys
5  import numpy
6
7  lons, lats, heights = numpy.loadtxt(sys.argv[1], unpack=True)
8
9  for i in xrange(len(heights)):
10     if heights[i] >=0:
11         print "%lf %lf %lf %lf" % (lons[i], lats[i], heights[i], 2670.0)
12     else:
13         print "%lf %lf %lf %lf" % (lons[i], lats[i], heights[i], 1670.0)
```

The result is a DEM file with a forth column containing the density values (Figure 2):

dem−10min−dens.xyz

```
1   −65.000000  −10.000000  157.000000  2670.000000
2   −64.833333  −10.000000  168.000000  2670.000000
3   −64.666667  −10.000000  177.000000  2670.000000
4   −64.500000  −10.000000  197.000000  2670.000000
5   −64.333333  −10.000000  144.000000  2670.000000
6   −64.166667  −10.000000  178.000000  2670.000000
7   −64.000000  −10.000000  166.000000  2670.000000
8   −63.833333  −10.000000  164.000000  2670.000000
9   −63.666667  −10.000000  189.000000  2670.000000
10  −63.500000  −10.000000  210.000000  2670.000000
```

4 Making the tesseroid model

Next, we'll use our new file dem-10min-dens.xyz and program tessmodgen to create a tesseroid model of the DEM:

run_example.sh

```
6  # Now, use tessmodgen to create a tesseroid model
7  tessmodgen −v −s0.166666667/0.166666667 −z0 < dem−10min−dens.xyz > dem−10min−mod.txt
```

tessmodgen places a tesseroid on each point of the DEM. The bottom of the tesseroid is placed on a reference level and the top on the DEM. If the height of the point is bellow the reference, the top and bottom will be inverted so that the tesseroid isn't upside-down. For topographic masses, the reference surface is h=0km (argument -z). The argument -s is used to specify the grid spacing (10') which will be used to set the horizontal dimensions of the tesseroid. Since we didn't pass the -d argument with the density of the tesseroids, tessmodgen will expect a fourth column in the input with the density values.

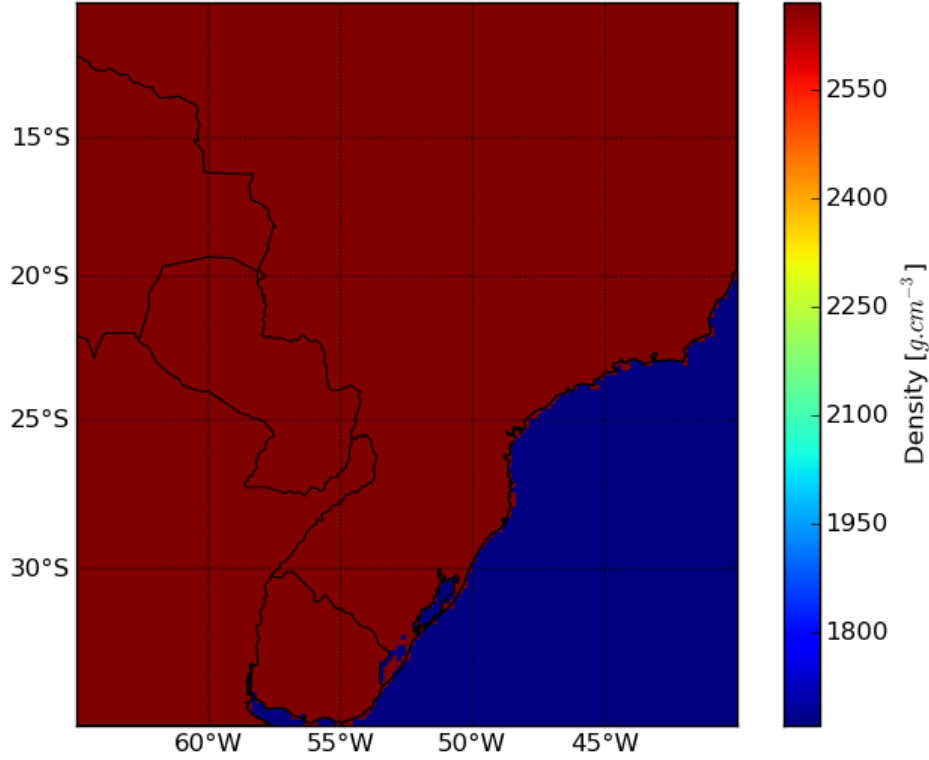The result is a tesseroid model file that should look somthing like this:

Figure 2: Density values. 2.67 $g.cm^{-3}$ in continents and 1.67 $g.cm^{-3}$ in the oceans.

dem–10min–mod.txt

```
1  # Tesseroid model generated by tessmodgen 1.0:
2  #    local time: Thu Apr  7 15:30:33 2011
3  #    grid spacing: 0.166667 deg lon / 0.166667 deg lat
4  #    reference level (depth): 0
5  #    density: read from input
6  −65.0833333335 −64.9166666665 −10.0833333335 −9.9166666665 157 0 2670
7  −64.9166663335 −64.7499996665 −10.0833333335 −9.9166666665 168 0 2670
8  −64.7500003335 −64.5833336665 −10.0833333335 −9.9166666665 177 0 2670
9  −64.5833333335 −64.4166666665 −10.0833333335 −9.9166666665 197 0 2670
10 −64.4166663335 −64.2499996665 −10.0833333335 −9.9166666665 144 0 2670
```

5   Calculating the GGT

Tesseroids now allows use of custom computation grids by reading the computation points from standard input. This way, if you have a file with lon, lat, and height coordinates and wish to calculate any gravitational field in those points, all you have to do is redirect stardard input to that file (using <).

For convience, we added to the set of tools the program tessgrd to create regular grids and print them to standard output. So if you don't want to compute on a custom grid (like us), you can simply pipe the output of tessgrd to the tessg* programs. This also means that you can pipe the results from one tessg* to the other:

run_example.sh

```
9  # Calculate the GGT
10 tessgrd −r−60/−45/−30/−15 −b50/50 −z250e03 | \
```

```
11    tessgxx dem−10min−mod.txt −lgxx.log | tessgxy dem−10min−mod.txt −lgxy.log | \
12    tessgxz dem−10min−mod.txt −lgxz.log | tessgyy dem−10min−mod.txt −lgyy.log | \
13    tessgyz dem−10min−mod.txt −lgyz.log | tessgzz dem−10min−mod.txt −lgzz.log > dem−10min−
         ggt.xyz
```
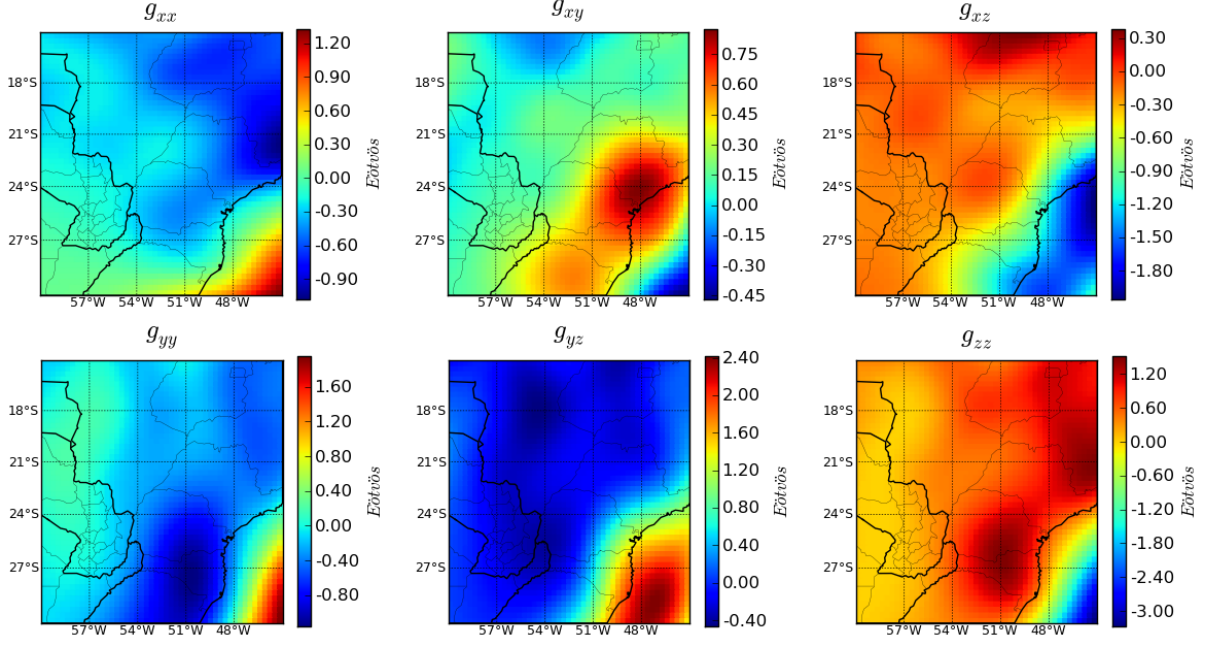
The result of this is shown in Figure 3.

Figure 3: GGT of caused by the topographic masses.

6   Making the plots

The plots were generated using the powerfull Python library Matplotlib (`http://matplotlib.sourceforge.net/index.html`). The script mkplots.py is somewhat more complicated that dem_density.py and requires a bit of "Python Fu". The examples in the Matplotlib website should give some insight into how it works. To hanble the map projections we used the Basemap toolkit of Matplotlib (`http://matplotlib.sourceforge.net/basemap/doc/html/`).

7   Conclusions

To wrap up our example, I would like to note that this same processes can be applied for any sort of interface, like Moho and sediment thickness maps. The difference is all in the choice of the reference level. For Moho it could be, say, h=-35km. Another thing to note is that the interface relief should be given as heights. So sediment thickness would have to be converted into height of the basin's basement (which would be negative values).