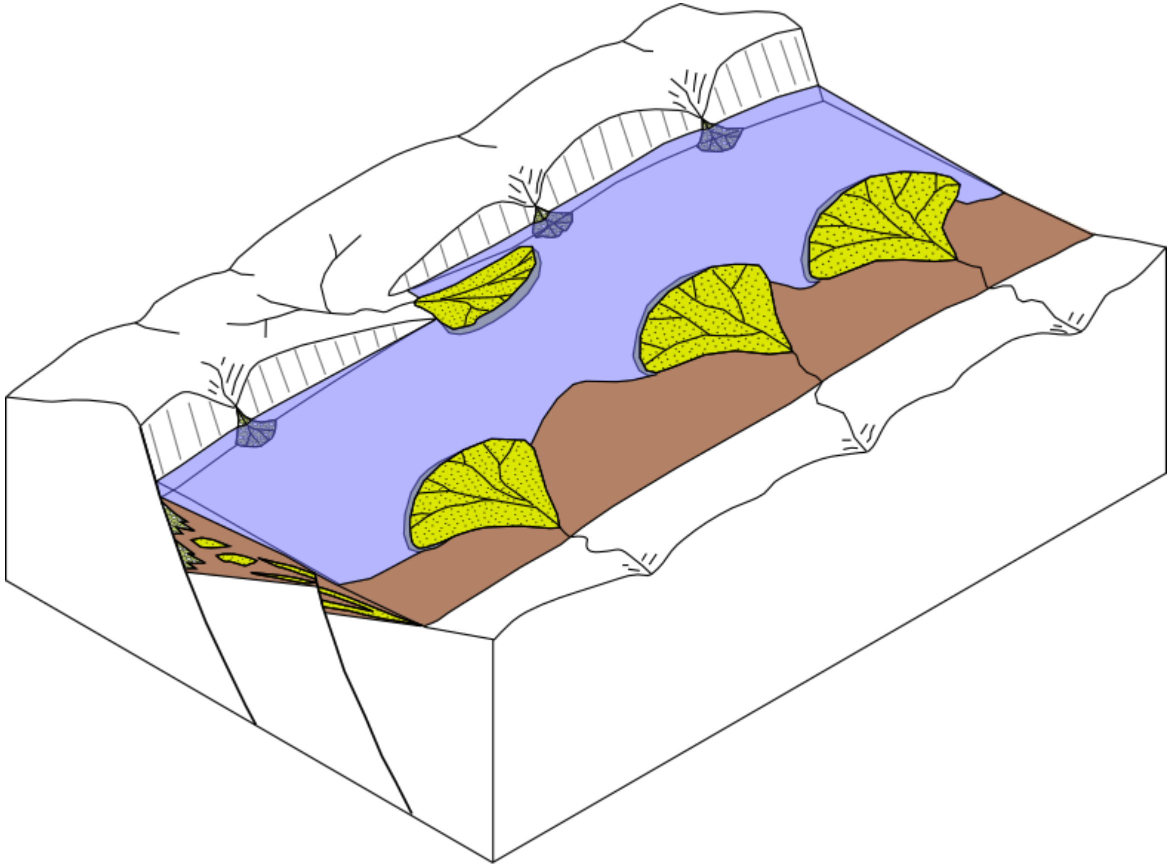
A practical crash-course into the maths and Python implementation of

Non-linear gravity inversion for the relief of a sedimentary basin

Sedimentary basins



Depressions filled by sediments*

Over time, sediments become rocks

Important natural resources:
Water, minerals, geothermal, oil & gas, etc.

Often ~1-15km deep & 100s wide

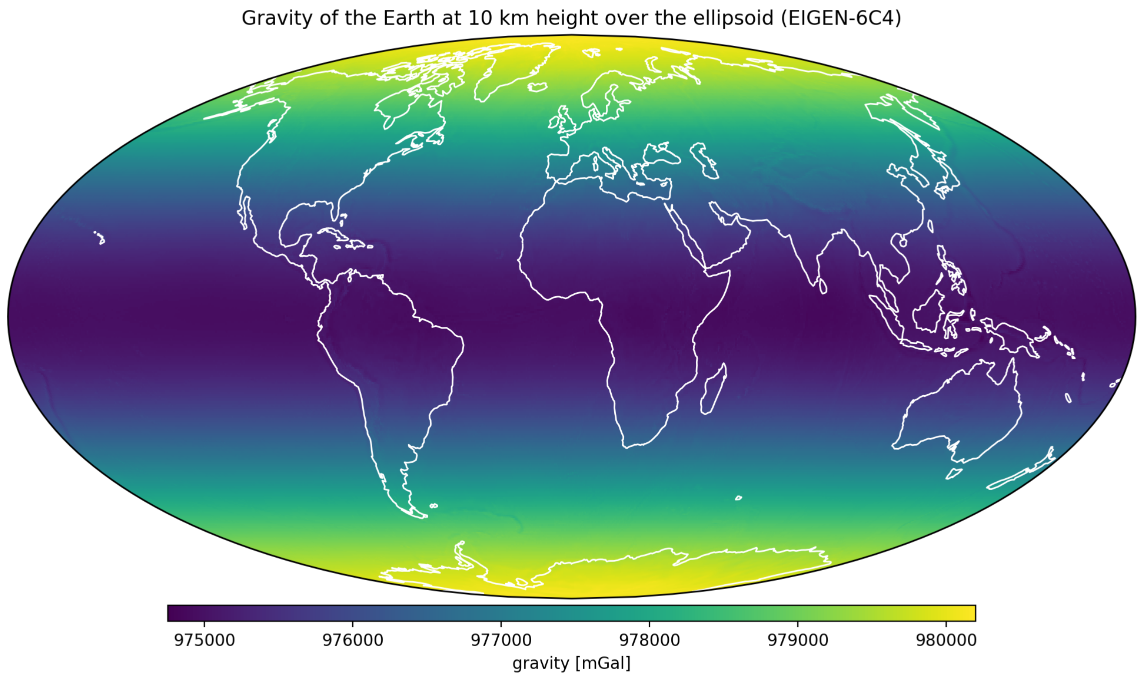
Need to understand their structure
to manage resources well

How can we study it without digging?

Image credit: https://commons.wikimedia.org/wiki/File:Half-graben_sedimentation.png (CC-BY-SA)

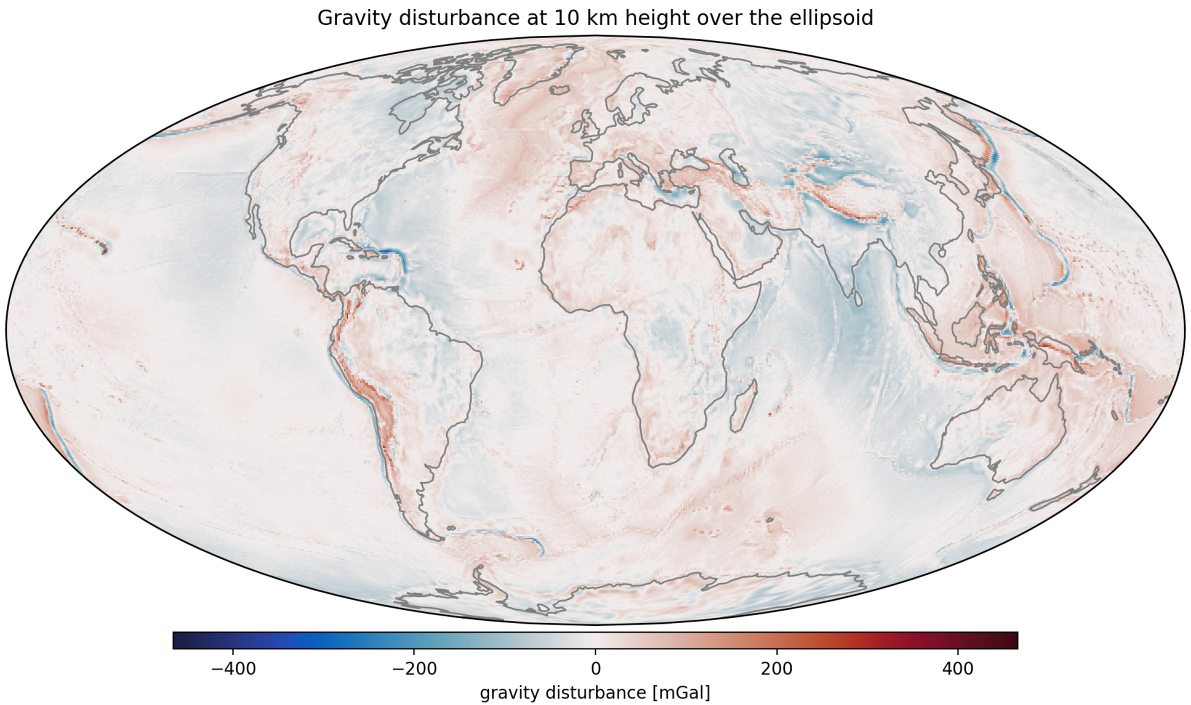
*I'm not a geologist.

Gravity disturbances



Measure gravity globally with very high accuracy ($1e-5$ - $1e-11$ m/s^2)

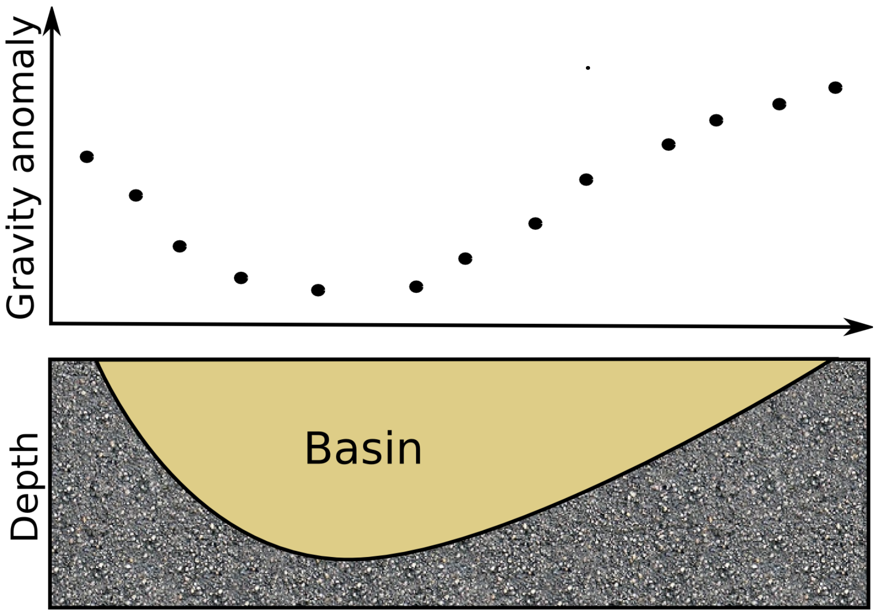
Subtract the gravity of a theoretical homogenous Earth to get the gravity disturbance



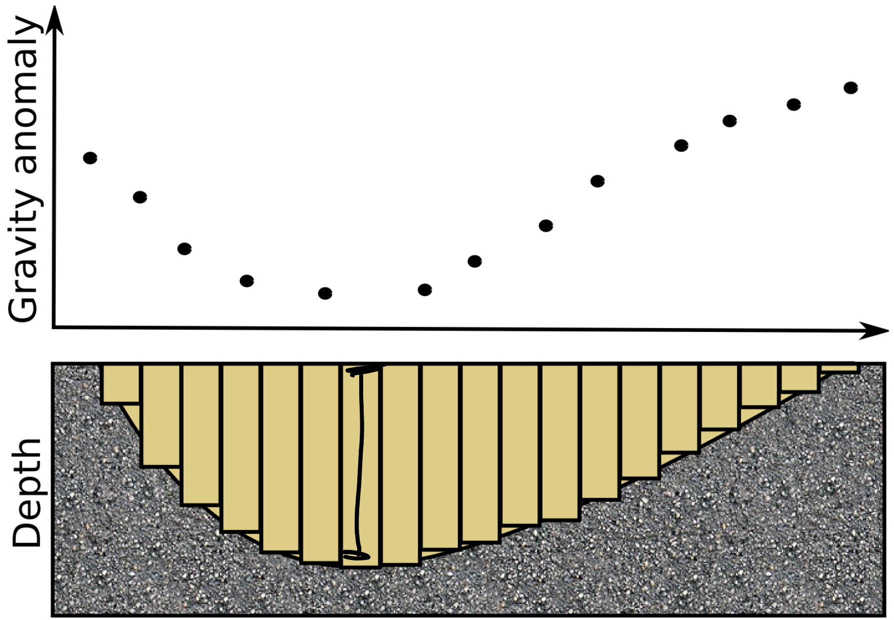
Disturbances relate to density anomalies inside the Earth

Sedimentary basin rocks have low density causing a negative disturbance

Modeling basins with gravity



Discretize



What is the height of the prisms (with known density) that fits the observed gravity disturbances?

To the maths!

Non-linear parametric model: $g(x,y,z) = \sum_{j=1}^{M} G\Delta\rho \int_{x_1}^{x_2}\int_{y_1}^{y_2}\int_{0}^{h_j} \frac{z-z'}{\ell^3}\,dx'\,dy'\,dz'$

$d_i = f_i(\bar{p})$

$\bar{p} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_M \end{bmatrix}_{M\times 1}$

$\bar{d} = \bar{f}(\bar{p})$

$\phi(\bar{p}) = [\bar{d}^o - \bar{f}(\bar{p})]^T [\bar{d}^o - \bar{f}(\bar{p})]$

misfit function

$\bar{d}^o \Rightarrow$ observed gravity disturbances

$\phi(\bar{p}) = \|\bar{d}^o - \bar{f}(\bar{p})\|^2 \Rightarrow \min_{\bar{p}} \phi$

Traditional approach is to linearize the model and solve several linear problems:

$$\bar{f}(\bar{p}) \approx \bar{f}(\bar{p}_0) + \bar{\bar{A}}(\bar{p}_0)\,\overline{\Delta p} + \cancel{O^2} \rightarrow 0$$

Here, we'll take a slightly different route to the solution.

Non-linear inverse problems with the Gauss-Newton method

Expand the misfit function in a Taylor series: $\phi(\bar{p}) = [\bar{d}^o - \bar{f}(\bar{p})]^T[\bar{d}^o - \bar{f}(\bar{p})]$

$$\phi(\bar{p}) \approx \phi(\bar{p}_0) + \bar{\nabla}\phi(\bar{p}_0)\bar{\Delta p} + \frac{1}{2}\bar{\Delta p}^T \underbrace{\bar{\bar{\nabla}}\phi(\bar{p}_0)}_{\text{Hessian}} \bar{\Delta p} + \cancel{O^3} \rightarrow 0 = \Gamma(\bar{p})$$

Γ is now a quadratic function of $\bar{\Delta p}$

Solve for $\bar{\Delta} p$ like we would in a linear problem.

$$\bar{\nabla}\Gamma = \bar{0}$$

$$\bar{\nabla}\Gamma = \bar{0} = \bar{\nabla}\phi(\bar{p}_0) + \frac{1}{\cancel{2}}\cancel{2}\,\bar{\bar{\nabla}}\phi(\bar{p}_0)\,\bar{\Delta p}$$

$$\underline{\underline{\bar{\bar{\nabla}}\phi(\bar{p}_0)\,\bar{\Delta p} = -\bar{\nabla}\phi(\bar{p}_0)}}$$

Newton's method

Derive the expression for the gradient of the misfit function with respect to the parameters.

the gradient vector

$$\bar{\nabla}_p = \begin{bmatrix} \frac{\partial}{\partial p_1} \\ \vdots \\ \frac{\partial}{\partial p_M} \end{bmatrix}$$

tip: start by calculating $\frac{\partial \phi}{\partial p_1}$

$$\phi(\bar{p}) = [\bar{d}^o - \bar{f}(\bar{p})]^T [\bar{d}^o - \bar{f}(\bar{p})]$$

$$\frac{\partial \phi}{\partial p_1} = -\frac{\partial \bar{f}}{p_1}^T [\bar{d}^o - \bar{f}(\bar{p})] - [\bar{d}^o - \bar{f}(\bar{p})]^T \frac{\partial \bar{F}}{\partial p_1}$$

$$= -2 \frac{\partial \bar{f}}{\partial p_1}^T [\bar{d}^o - \bar{F}(\bar{p})]$$

tip: the transpose of a scalar is itself

$$\bar{\nabla}\phi = \begin{bmatrix} -2\,\frac{\partial \bar{f}}{p_1}^T\left[\bar{d}^o - \bar{f}(\bar{p})\right) \\ \vdots \\ -2\,\frac{\partial \bar{f}}{\partial p_M}^T\left[\bar{d}^o - \bar{f}(\bar{p})\right] \end{bmatrix} = -2\begin{bmatrix} \frac{\partial \bar{f}}{\partial p_1}^T \\ \vdots \\ \frac{\partial \bar{f}}{\partial p_M}^T \end{bmatrix}\left[\bar{d}^o - \bar{f}(\bar{p})\right] = -2\underbrace{\begin{bmatrix} \frac{\partial f_1}{\partial p_1} & \frac{\partial f_2}{\partial p_1} & \cdots & \frac{\partial f_N}{\partial p_1} \\ \vdots & & & \\ \frac{\partial f_1}{\partial p_M} & \frac{\partial f_2}{\partial p_M} & \cdots & \frac{\partial f_N}{\partial p_M} \end{bmatrix}}_{\text{Jacobian } \bar{\bar{A}}}\left[\bar{d}^o - f(\bar{p})\right]$$

$$\bar{\nabla}\phi = -2\,\bar{\bar{A}}^T(\bar{p})\left[\bar{d}^o - \bar{f}(\bar{p})\right]$$

$$\bar{\bar{\nabla}}\phi \approx 2\,\bar{\bar{A}}(\bar{p})^T\,\bar{\bar{A}}(\bar{p})$$

Gauss-Newton method

$$\underbrace{\cancel{2}\,\bar{\bar{A}}^T\,\bar{\bar{A}}\,\Delta p = \cancel{2}\,\bar{\bar{A}}^T\left(\vec{d}^{\,o} - \bar{F}(\bar{p}_0)\right)}$$

System of normal equations

Solve for $\bar{\Delta} p$ and repeat.

To the code!

Now we'll see how to code all of this up in Python.

We'll cheat and use ready-made forward modelling.

But you can see the code for all of that in the repository.

https://github.com/GeophysicsLibrary/non-linear-gravity-inversion

Smoothness regularization

$\lambda \; \Theta(\bar{p})$

$$\phi(\bar{p}) = [\bar{d}^o - \bar{f}(\bar{p})]^T [\bar{d}^o - \bar{f}(\bar{p})] + \underbrace{\lambda \; \bar{p}^T \bar{\bar{R}}^T \bar{\bar{R}} \; \bar{p}}_{\text{Regulizing function}}$$

$$+ \lambda \, \| \nabla_x \bar{p} \|^2$$

What is R? $\quad \bar{\bar{R}} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \\ & & & & & -1 & 1 \end{bmatrix}_{(M-1,\,M)}$

$$\bar{\bar{R}}\bar{p} = \begin{bmatrix} p_2 - p_1 \\ p_3 - p_2 \\ \vdots \\ p_M - p_{M-1} \end{bmatrix} \qquad \frac{\partial \bar{p}}{\partial x} \approx \nabla_x \bar{p}$$

$$\bar{\nabla}\phi = \underline{\bar{\nabla}\,\text{DATA MISFIT}} + \underbrace{\bar{\nabla}\left[\lambda p^T R^T R p\right]}_{\Theta} \qquad \bar{\bar{\nabla}}\phi = \underline{A^T A} + \underline{\bar{\bar{\nabla}}\left[\lambda p^T R^T R p\right]}$$

$$\frac{\partial \Theta}{\partial p_1} = \dots \qquad \bar{\nabla}\Theta = 2\lambda \bar{\bar{R}}^T \bar{\bar{R}}\,\bar{p} \qquad \bar{\bar{\nabla}}\Theta = 2\lambda \bar{\bar{R}}^T \bar{\bar{R}}$$

$$\underbrace{\left(\bar{\bar{A}}^T \bar{\bar{A}} + \lambda \bar{\bar{R}}^T \bar{\bar{R}}\right)}_{\bar{\bar{\nabla}}\phi} \Delta \bar{p} = \underbrace{\bar{\bar{A}}^T\left(\bar{d}^0 - \bar{F}(p_0)\right) - \lambda \bar{\bar{R}}^T \bar{\bar{R}}\,\bar{p}_0}_{-\bar{\nabla}\phi(\bar{p}_0)}$$

Regularized Gauss-Newton solution