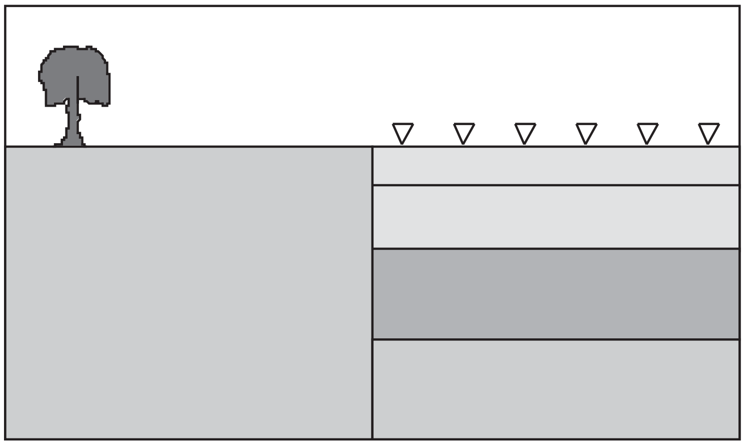
Exercise: The Vertical Fault – Part 3

Monte Carlo analysis of a nonlinear inverse problem



Again, using the same data as in Exercise 1, we wish to estimate $\Delta\rho(z)$ from observations of the horizontal gravity gradient along the x-axis. In this exercise we shall use the same parameterization of the problem as in Exercise 2, but this time we shall use MCMC (the Metropolis Algorithm) to sample solutions to the problem. The distribution of solutions is the *a posteriori* probability density $\sigma(\mathbf{m}) = \rho(\mathbf{m})L(\mathbf{m})$. In this exercise we will assume that the likelihood function $L(\mathbf{m})$ has the following form:

$$L(\mathbf{m}) = K_1 \exp\left(-\frac{\|\mathbf{d}_{obs} - \mathbf{g}(\mathbf{m})\|^2}{2s^2}\right), \tag{1}$$

where K_1 is a normalization constant, and s is the standard deviation of the noise. We assume that $s = 1.0 \cdot 10^{-9} s^{-2}$. Furthermore, we shall assume that the prior probability density is constant:

$$\rho(\mathbf{m}) = K_2 \tag{2}$$

indicating that no models are preferred over others.

If we define the *misfit*

$$S(\mathbf{m}) = -\ln(\sigma(\mathbf{m})) ,$$

each iteration of the Metropolis algorithm runs as follows:

1. Perturb the current model $\mathbf{m}_k$ (in the first iteration, choose a random model) by choosing a random component of $\mathbf{m}_k$ and adding a random number from the interval $[-\ell, \ell]$, where ℓ is a maximum step length (chosen by yourself).

2. The perturbed model $\mathbf{m}_k^{pert}$ is only accepted as the next model ($\mathbf{m}_{k+1}$) with the probability

$$p_{accept} = \begin{cases} \exp(-(S(\mathbf{m}_k^{pert}) - S(\mathbf{m}_k)) & \text{for } S(\mathbf{m}_k^{pert}) > S(\mathbf{m}_k) \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

   If $\mathbf{m}_k^{pert}$ is accepted, put $\mathbf{m}_{k+1} = \mathbf{m}_k^{pert}$. If, on the other hand, $\mathbf{m}_k^{pert}$ is rejected, put $\mathbf{m}_{k+1} = \mathbf{m}_k$.

3. Repeat (1) and (2) with $\mathbf{m}_{k+1}$ as the current model.

- Find, using the Metropolis Algorithm, a large number of models $\mathbf{m}^{(n)}$, $n = 1 \ldots N$, distributed according to the *a posteriori* distribution.