SAMPLING THE DATA AND THE MODEL

This assignment is meant to demonstrate some of the basic issues with inversion, such as;

- Getting the model right
- Getting the error statistics right
- Noise versus mistakes

The usual equation which we want to invert looks like

$$d(x) = \int_a^b k(x,y)m(y)dy$$

and we would write this in discrete form as

$$\sum_{j=1}^{M} k(x_i, y_j)w(y_j)m(y_j)\Delta y_j = d(x_i) \quad i = 1, N$$

where $w(y_j)$ is some weight that specifies how the numerical approximation to integration is performed (we will just sum the integrand here). If we specify a discretization increment Δy we can write this as just

$$\sum_{j=1}^{M} K_{ij} m_j = d_i \quad i = 1, N$$

or in shorthand

$$Km = d$$

where K is an N row (number of observations) by M column (number of model parameters) matrix, m is the column vector of model parameters, and d is the column vector of observations.

We will see that the solution to this problem involves a generalized inverse K^{-g} which in some sense solves the inverse of the above equation, yielding an estimated model

m^{est}, and the data that the estimated model predicts d^{pre} (as opposed to the observed data d^{obs}). That is

$$\mathbf{Km} = \mathbf{d}^{obs} = \mathbf{d}^{true} + \mathbf{e}$$
$$\mathbf{m}^{est} = \mathbf{K}^{-g}\mathbf{d}^{obs}$$
$$\mathbf{Km}^{est} = \mathbf{d}^{pre}$$

If you have understood the physics correctly, and the model you propose is perfect, and there are no errors of observation, then

$$\mathbf{Km}^{true} = \mathbf{d}^{true}$$

where $\mathbf{d}^{true}$ is what we would observe with a perfect experimental setup and measuring equipment. The problem of course is that we never deal with this equation, there are always errors of observation, as well as some doubt about whether the proposed model is correct.

In matlab, if we have

$$Km = d$$

matlab will produce the Gauss-Newton solution if you do

the solution Jim calls "Gauss-Newton" is called "least-squares" in my notes.

$$m^{est} = K\backslash d$$

That is the backslash is matlab code for

$$m^{est} = (K^T K)^{-1} K^T d$$

which is the Gauss-Newton solution.

1) Write a program to do the above, rather than use the matlab shorthand, because we will build on that program as we go along.

I have placed a data file assignment1.txt ~~on PAWS.~~ on this web page. You can load this file into matlab. This has four columns x,y, noise1 noise2 and noise3. x and y are linearly related (slope

1/2, intercept 1) and y is noiseless data. noise1 is gaussian, noise2 has a lognormal distribution and noise3 is noise1 with two outliers.

2) Write a program to find the linear parameters (the intercept and slope) using the basic Gauss-Newton solution with all three noise cases added in turn to y to give d^{obs}. Plot a histogram (*hist(d^{obs},10)*will do this in matlab) for all three noise cases, as well as a histogram of the residuals $d^{obs} - d^{pre}$. The Gauss-Newton solution assumes the errors are normally distributed, so we should expect the best results (best solution and most gaussian like residuals) if the errors are normally distributed.

3) Try solving the problem again with an incorrect model (something other than linear).