

Here's a simple example of a non-linear, least squares inverse problem. To set up the problem, I will first generate some gravity values over a buried sphere. Then I will add some random noise to those values. Finally, I will use a couple different techniques to determine the geometric parameters of the sphere (radius, depth, offset from the origin) directly from the original and/or noisy values. For now, start with a sphere of radius 500 m, depth to center of 750 m, and density contrast of 250 kg/m³ and specify its center, xx, at 1500 meters from the origin. G is the gravity constant:

First, some initial information and constants: $i := 0..15$ $x_i := 200 \cdot i \cdot \text{m}$ $G := 6.67 \cdot 10^{-11} \frac{\text{m}^3}{\text{kg} \cdot \text{sec}^2}$

Now the parameters of the model: $\rho := 250 \frac{\text{kg}}{\text{m}^3}$ $z := 750 \cdot \text{m}$ $xx := 1500 \cdot \text{m}$ $R := 500 \cdot \text{m}$

The equation for gravity over a buried sphere is: $g_i := \frac{4}{3} \cdot \pi \cdot G \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + \left((x_i - xx) \right)^2 \right]^{\frac{3}{2}}}$ $K1 := \frac{4}{3} \cdot \pi \cdot G$

If we add a little random noise to the signal we get: $n := .5 \cdot \text{mean}(g)$ $h_i := \left(g_i + \text{rnd}(1) \cdot n \right) - \frac{n}{2}$

Now we have two vectors of data, g(i) holds the original values, h(i) holds those same values with random noise, as a percent of the mean value of g(i), included. The next step is to see how to invert those data, with or without noise, to recover the original parameters of the model (R, z, xx, and/or density) directly. Because of the inherent ambiguity, we will assume we know the density contrast of the source and invert for the geometric parameters. There are a few different ways to do curve fitting, or inversion, with Mathcad.

Mathcad's *Find* function returns the solution to a system of equations when there is an exact solution or the number of equations equals the number of unknowns. *Find* should recover the unknown parameters if we use the exact Bouguer values, g(i).

First, provide initial guesses for the "unknown" parameters: $R := 425 \cdot \text{m}$ $xx := 1000 \cdot \text{m}$ $z := 500 \cdot \text{m}$

The next step is to build a solve block which starts with *Given* and includes the equations and inequalities. Because there will only be three unknowns, and we are using exact data, we only need three equations. Each of the three will include the unknowns and some real values:

Given

$$K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + \left(x_3 - xx \right)^2 \right]^{\frac{3}{2}}} = g_3 \quad K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + \left(x_6 - xx \right)^2 \right]^{\frac{3}{2}}} = g_6 \quad K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + \left(x_9 - xx \right)^2 \right]^{\frac{3}{2}}} = g_9$$

Now use *Find* to solve the system:

$$\begin{bmatrix} z \\ xx \\ R \end{bmatrix} := \text{find}(z, xx, R) \quad \text{So Mathcad determines:} \quad R = 500 \text{ m} \quad z = 750 \text{ m} \quad xx = 1.5 \cdot 10^3 \text{ m}$$

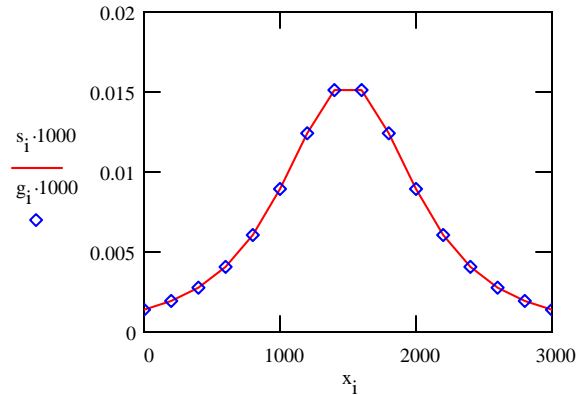
Since we used perfect gravity values for the sphere we get what we expect, the starting values.

To check the fit, we can put R, z and xx into a new equation, s(i), and graph it along with the original data.

$$s_i := K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + (x_i - xx)^2 \right]^{\frac{3}{2}}}$$

The error for the fit is: $\left[\sum_i (s_i - g_i)^2 \right]^{.5} = 0 \cdot \text{m} \cdot \text{sec}^{-2}$

So, even with wild initial guesses, we recover the original parameters using three equations for the three unknowns; such is the value of perfect data. As the figure to the right shows, the inversion ($s(i)$) lies exactly on the data (diamonds). Thus we accurately recovered the x -offset, depth, and radius of the buried sphere. The next steps are to look at the noisy data, and an overdetermined system.



Now, if we follow the previous example but use the noisy data instead of the original data we should get parameters that are close to the original data and yield a pretty good fit. To start use the same original values but specify $h(i)$ as the data rather than $g(i)$:

These are the original values: $\rho := 500 \frac{\text{kg}}{\text{m}^3}$ $xx := 1500 \cdot \text{m}$ $z := 750 \cdot \text{m}$

Start the solve block with *given* and then put in some equations and data as constraints:

Given

$$K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + (x_4 - xx)^2 \right]^{\frac{3}{2}}} = h_4$$

$$K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + (x_7 - xx)^2 \right]^{\frac{3}{2}}} = h_7$$

$$K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + (x_{10} - xx)^2 \right]^{\frac{3}{2}}} = h_{10}$$

$$K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + (x_{13} - xx)^2 \right]^{\frac{3}{2}}} = h_{13}$$

Now use *Find* to solve the system:

$$\begin{bmatrix} z \\ xx \\ R \end{bmatrix} := \text{find}(z, xx, R) \quad \text{The vector } [z, xx, R] \text{ is defined as the result of the } \textit{find} \text{ operation.}$$

The result, from using four of the noisy values: $R = 431.919 \cdot \text{m}$ $z = 862.201 \cdot \text{m}$ $xx = 1.52 \cdot 10^3 \cdot \text{m}$

The parameters from using *find* on four noisy values are pretty close to the original values.

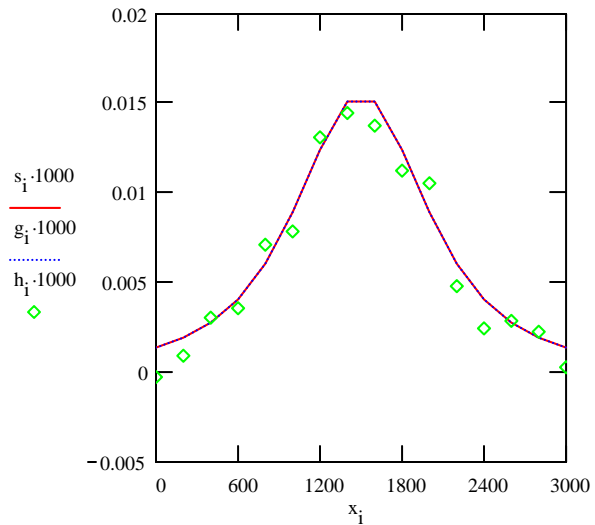
.....

Put ρ , z and xx into a new equation, $s(i)$, and graph it along with the original data to check the fit:

$$s_i := K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + (x_i - xx)^2 \right]^{\frac{3}{2}}}$$

$$\left[\sum_i (g_i - s_i)^2 \right]^{\frac{1}{2}} = 2.969 \cdot 10^{-6} \text{ m} \cdot \text{sec}^{-2}$$

The fit is poorly constrained with only four points yet picking four well distributed points gets it pretty close. What we want is to use all the available data to best constrain the fit and minimize the difference between the noisy data and the calculated results from the model.



Now we need a method that uses all the available data. *Minerr* works exactly like *Find* but *Minerr* will return a non-linear least squares solution. Unfortunately, *Minerr* will not accept a range variable in the arguments. To use it like the previous example, you would need to write an equation for each measurement. That can be done using the copy command but it takes a few pages if there are many observations. A faster way is to pose the problem slightly differently.

First, the parameters of the model and the tolerance (TOL) for the least squares fit:

$$\rho := 250 \frac{\text{kg}}{\text{m}^3} \quad z := 750 \text{ m} \quad xx := 1500 \text{ m} \quad R := 500 \text{ m} \quad \text{TOL} := .001$$

Now we proceed a bit differently than before. To use *MINERR*, we have to define the function to be fit:

$$F(x, z, xx, R) := K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + (x - xx)^2 \right]^{\frac{3}{2}}}$$

Next, denote the sum of squared errors to be minimized. This is the squared differences between the data with random errors and the values in the defined function:

$$\text{SSE}(z, xx, R) := \sum_i \left(h_i - F(x_i, z, xx, R) \right)^2$$

Now provide some initial guesses for the parameters, get them close to avoid local minima. The more reasonable these values are, with respect to the "real" results, the faster the solution and the less likely Mathcad is to find a local minima. In a more general problem, you may have to experiment with the guesses and TOL to get a satisfactory solution.

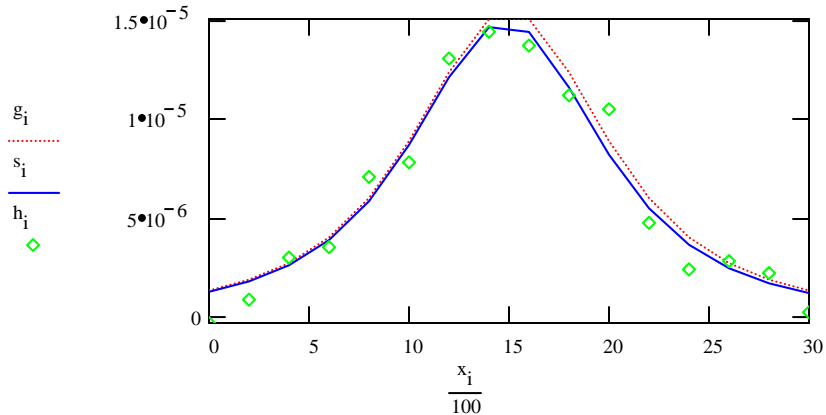
These are my initial guesses for the parameters: $R := 425 \cdot m$ $z := 650 \cdot m$ $xx := 1600 \cdot m$

As usual, start the solve block with a "given" and provide as many constraints as unknowns:

Given

$$SSE(z, xx, R) = 0 \cdot \frac{m^2}{cm^4} \quad l=1 \quad 2=2 \quad \text{The last two are necessary dummy constraints.}$$

$$\begin{bmatrix} z \\ xx \\ R \end{bmatrix} := \text{Minerr}(z, xx, R) \quad ERR = 1.617 \cdot 10^{-11} \quad s_i := K1 \cdot R^3 \cdot \rho \cdot \frac{z}{\left[z^2 + (x_i - xx)^2 \right]^{\frac{3}{2}}}$$



The figure above shows that the least squares solution fits well. The resulting parameters are:

$$z = 737.385 \cdot m \quad R = 488.703 \cdot m \quad xx = 1.485 \cdot 10^3 \cdot m$$

Obviously, from the figure, the derived parameters offer a model that fits the noisy data well and is close to the original model we used to generate the original values.

The accuracy and speed of the result is dependent on the initial guesses, which should be as good as you can get them, and TOL. Sometimes it is advantageous to start with large tolerance and reduce it when you get reasonable parameters. Graphing the result provides a quick check to see if Minerr has found a local minimum.

Now, let's make it more complicated and bury two adjacent spheres:

$$i := 0..15 \quad x_i := 300 \cdot i \cdot m \quad TOL := .001 \quad G := 6.67 \cdot 10^{-11} \cdot \frac{m^3}{kg \cdot sec^2} \quad K1 := \frac{4}{3} \cdot \pi \cdot G \quad \rho := .5 \cdot \frac{gm}{cm^3}$$

Now the parameters: $z1 := 650 \cdot m$ $xx1 := 1400 \cdot m$ $R1 := 450 \cdot m$ $z2 := 750 \cdot m$ $xx2 := 3100 \cdot m$ $R2 := 500 \cdot m$

The equation for two adjacent buried spheres is:

$$g_i := K1 \cdot \rho \cdot \left[\frac{R1^3 \cdot z1}{\left[z1^2 + (x_i - xx1)^2 \right]^{\frac{3}{2}}} + \frac{R2^3 \cdot z2}{\left[z2^2 + (x_i - xx2)^2 \right]^{\frac{3}{2}}} \right]$$

..... / \ n

Adding some random noise as before yields:

$$n := .5 \cdot \text{mean}(g) \quad h_i := (g_i + \text{rnd}(1) \cdot n) - \frac{n}{2}$$

Define the function for *MINERR*: $F(x, z1, z2, xx1, xx2, R1, R2) := K1 \cdot \rho \cdot \left[\frac{R1^3 \cdot z1}{[z1^2 + (x - xx1)^2]^{\frac{3}{2}}} + \frac{R2^3 \cdot z2}{[z2^2 + (x - xx2)^2]^{\frac{3}{2}}} \right]$

Next, denote the sum of squared errors to be minimized. This is the squared differences between the data with random errors and the values in the defined function:

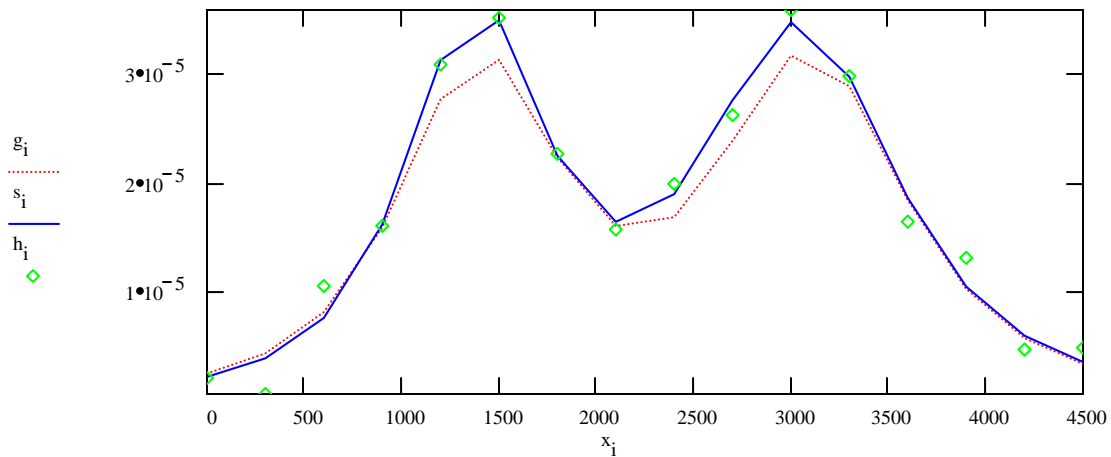
$$SSE(z1, z2, xx1, xx2, R1, R2) := \sum_i (h_i - F(x_i, z1, z2, xx1, xx2, R1, R2))^2$$

Then provide some initial guesses for the parameters and set up the solve block, starting with a *given*:

$$z1 := 550 \cdot \text{m} \quad xx1 := 1200 \cdot \text{m} \quad R1 := 400 \cdot \text{m} \quad z2 := 850 \cdot \text{m} \quad xx2 := 3300 \cdot \text{m} \quad R2 := 550 \cdot \text{m}$$

Given $SSE(z1, z2, xx1, xx2, R1, R2) = 0 \cdot \frac{\text{m}^2}{\text{sec}^4}$ 1=1 2=2 3=3 4=4 5=5

$$\begin{bmatrix} z1 \\ z2 \\ xx1 \\ xx2 \\ R1 \\ R2 \end{bmatrix} := \text{Minerr}(z1, z2, xx1, xx2, R1, R2) \quad s_i := K1 \cdot \rho \cdot \left[\frac{R1^3 \cdot z1}{[z1^2 + (x_i - xx1)^2]^{\frac{3}{2}}} + \frac{R2^3 \cdot z2}{[z2^2 + (x_i - xx2)^2]^{\frac{3}{2}}} \right]$$



The parameters determined from the noisy data are:

$$\begin{aligned} z1 &= 563.161 \text{ m} & xx1 &= 1.382 \cdot 10^3 \text{ m} & R1 &= 426.647 \text{ m} \\ z2 &= 772.842 \text{ m} & xx2 &= 3.048 \cdot 10^3 \text{ m} & R2 &= 525.235 \text{ m} & \text{ERR} &= 3.914 \cdot 10^{-11} \end{aligned}$$

Judging by the graph and ERR, the fit is pretty good. This general application of *Minerr* can be used in most curve-fitting situations to find the least-squares, best-fit parameters for a model.

Here is the application of the preceding techniques, using *MINERR*, to the equation for a buried horizontal

thin slab. The equation is from Dobrin and Savit (1988, page 523). The parameters of the model are the thickness (t) of the slab, the depth (z) to the center of the slab, and the offset (xx) of the edge of the slab from the origin. Again, applying non-linear least squares to some noisy data does a fine job of recovering the original parameters.

First some constants: $i := 0..15$ $x_i := 100 \cdot i \cdot \text{m}$ $G := 6.67 \cdot 10^{-11} \frac{\text{m}^3}{\text{kg} \cdot \text{sec}^2}$

Now the parameters: $z := 100 \cdot \text{m}$ $t := 100 \cdot \text{m}$ $xx := 750 \cdot \text{m}$ $\rho := 500 \frac{\text{kg}}{\text{m}^3}$

The slab equation: $g_i := 2 \cdot G \cdot \rho \cdot t \cdot \left(\frac{\pi}{2} - \text{atan} \left(\frac{x_i - xx}{z} \right) \right)$ Add noise: $n := .75 \cdot \text{mean}(g)$ $h_i := (g_i + \text{rnd}(1) \cdot n) - \frac{n}{2}$

Define the function for MINERR to iterate on: $F(x, z, t, xx) := 2 \cdot G \cdot \rho \cdot t \cdot \left(\frac{\pi}{2} - \text{atan} \left(\frac{x - xx}{z} \right) \right)$

Define the sum of squares to be minimized. This is just the squared differences between noisy data and the function: $\text{SSE}(z, t, xx) := \sum_i (h_i - F(x_i, z, t, xx))^2$

Provide some initial guesses, the closer the better: $z := 75 \cdot \text{m}$ $t := 75 \cdot \text{m}$ $xx := 690 \cdot \text{m}$

Now the solve block: Given

Constraints: $\text{SSE}(z, t, xx) = 0 \cdot \frac{\text{m}^2}{\text{sec}^4}$ $1=1$ $2=2$

And solution: $\begin{bmatrix} z \\ t \\ xx \end{bmatrix} := \text{Minerr}(z, t, xx)$ Now: $z = 84.44 \cdot \text{m}$ $t = 97.843 \cdot \text{m}$ $xx = 780.197 \cdot \text{m}$

Calculate the function with the new parameters: $s_i := 2 \cdot G \cdot \rho \cdot t \cdot \left(\frac{\pi}{2} - \text{atan} \left(\frac{x_i - xx}{z} \right) \right)$ ERR = $4.912 \cdot 10^{-11}$

Put the results in milligals: $s := 10^5 \cdot s$ $g := 10^5 \cdot g$ $h := 10^5 \cdot h$

Then plot them (original values, noisy values, and least squares solution for the parameters) to check the fit:

