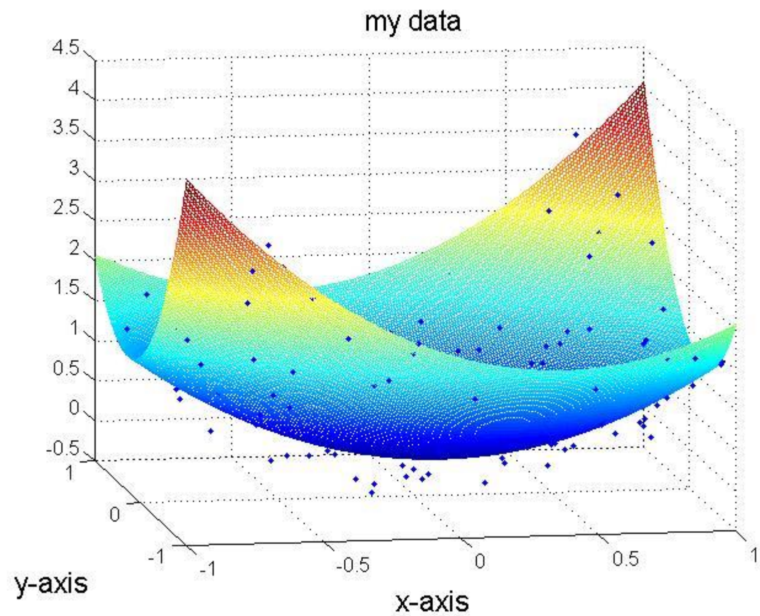
MACM 316 HW #9: Best Fit Surface

Mathew Teoh
mteoh@sfu.ca

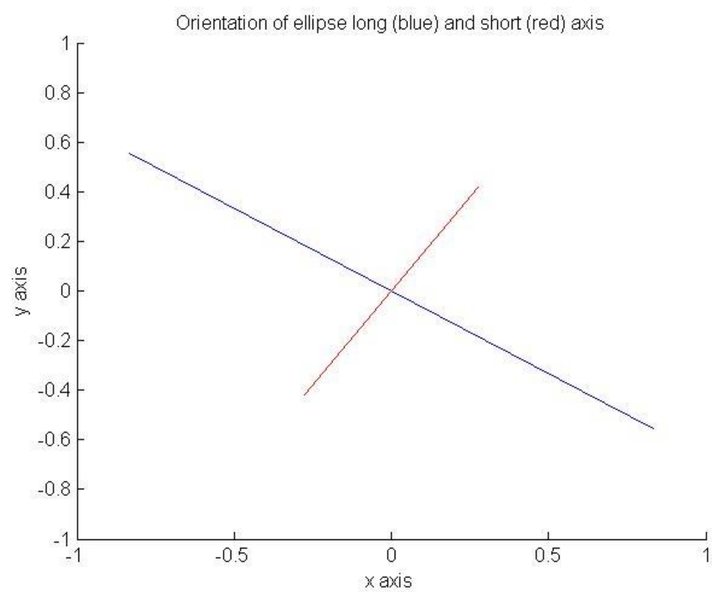
Method:

Using the protected function file *bfDataHW.p* I generated my own set of data based on my student number: **301165239**. My value of code was: **1866738.** Using Dr. Muraki's *bfqrDemo.m* as a starting point, I computed the best fit coefficients for an eilliptic paraboloid to fit the N data points. Following the procedure outlined in class, I constructed a 3 by N matrix called B, where the first column was the squared x values of the data, the second column was the product of the x and y data values, and the third column as the squared values of the y values. Then, performing a QR factorization on the matrix B, I minimized the squared error between the paraboloid and the actual data to obtain the coefficients. This was done by minimizing the quantity $|Q^T\vec{y} - R\vec{c}|^2$ following the code in the demo.



Observations and Analysis:

The plot of the surface with the generated points is shown on the right. I got an average squared error of **0.0758**. Using the matrix given in the assignment PDF, I determined how the ellipse in the paraboloid was rotated with respect to the z axis. I first computed the eigenvectors of the matrix, then noted that the eigenvectors pointed in the direction of the semimajor and semiminor axes of the ellipse. I got the angle of rotation from the x axis by taking the inverse tangent of the ratio of the first eigenvector's entries. Then I divided this angle by π to get the fraction of π that the ellipse was being rotated. I computed this fraction to be **0.8132**. A plot is shown on the right to describe the orientation of the ellipses axis. Note that the red and blue line segments are orthogonal, but the scaling of the figure does not reflect this because one axis is slightly longer than the other.



Conclusion:

Best fit lines are only the beginning to data approximation. Indeed, given data that one suspects to fit an elliptic paraboloid, one could fit an elliptic paraboloid to the data. This fit could have been accomplished by directly setting the gradient of the squared error to 0, but it turned out that QR factorization can be used to compute the paraboloid's coefficients instead. Since this took fewer lines of code to implement, this would be the sensible solution to follow.