16-811 Assignment 3: Resubmission 1

Emma Benjaminson

31 October 2019

1 Problem 1

1.1 Part (c)

Note that the code used to solve this problem is included in code/q1.m.

If I want to use a quadratic function to approximate $f(x)$, then I need a polynomial with $n = 2$, and therefore $n+2 = 4$ points that meet the requirement:

$$(-1)^i[f(x_i) - p(x_i)] = \epsilon||f - p||_\infty \tag{1}$$

I also know (see Part (a)) that the $n + 1 = 3$ derivative of $f(x)$ is $f'''(x) = 2cosh(x)$ which is positive for the entire interval $[-3, 3]$. Therefore, $x_0 = -3$ and $x_3 = 3$. Let me represent my quadratic function as the polynomial $p(x) = a + bx + cx^2$.

As I was working on solving this problem initially, I quickly realized that it is difficult to find 3 equations to find the 3 unknown coefficients in the polynomial expression above. Thanks to a suggestion from the TAs, I propose approximating the function $f(x)$ using a linear polynomial where the coefficient $c = 0$. I will still look for 4 points and show that they satisfy the error requirement and that therefore a linear polynomial is in fact the best quadratic approximation to the function $f(x)$. I will relax the constraint that these 4 points must be uniformly spaced.

The first step to finding coefficients a and b is to realize that the error at the endpoints of the interval is the same but with opposing signs, thus:

$$e(x_0) = -e(x_3) \tag{2}$$

$$e(x_0) = f(x_0) - p(x_0) = 2sinh(-3) + \frac{1}{3} - (a + b(-3)) = -19.7024 - a + 3b \tag{3}$$

$$e(x_3) = 2sinh(3) + \frac{1}{3} - (a + b(3)) = 20.3691 - a - 3b \tag{4}$$

$$-19.7024 - a + 3b = -(20.3691 - a - 3b) \tag{5}$$

$$a = \frac{1}{3} \tag{6}$$

Now that we have the y-intercept of our linear approximation, I need a second equation to find the slope, b. Similarly to above, we also know that the error at x_0 and x_2 are equal and of the same sign, so let's solve that equation:

$$e(x_0) = e(x_2) = f(x_0) - p(x_0) = f(x_2) - p(x_2) \tag{7}$$

$$-19.7024 - a + 3b = 2sinh(x_2) + \frac{1}{3} - a - bx_2 \tag{8}$$

We can solve this equation as follows:

$$2sinh(x_2) - (x_2 + 3)b + 20.0357 = 0 \tag{9}$$

I know that the error at each of the 4 points, including x_2, must be minimized, which means that the first derivative of this expression must be equal to 0. Let's solve that out:

$$\frac{d}{dx_2}e(x_2) = \frac{d}{dx_2}\big(2sinh(x_2) + \frac{1}{3} - (a + b(x_2))\big) = 2cosh(x_2) - b = 0 \tag{10}$$

Therefore I can solve for b in this expression:

$$b = 2cosh(x_2) \tag{11}$$

Now I can substitute this expression for b into Equation 9 above and solve for x_2:

$$2sinh(x_2) - 2x_2cosh(x_2) - 6cosh(x_2) + 20.0357 = 0 \tag{12}$$

Solving this gives us three possible roots, $x = -3.00223, -2.99777, 1.64764$. The first root is outside the interval defined in the problem statement. The second root is close enough to $x_0 = -3$ that I will assume it is not distinct from -3 and so not the correct root for x_2. That leaves me with $x_2 = 1.64764$. Plugging this back into my expression for b, I find that $b = 5.3872$.

Now my full polynomial expression is $p(x) = 5.3872x + \frac{1}{3}$. The plot of this approximation against the original function is shown below.

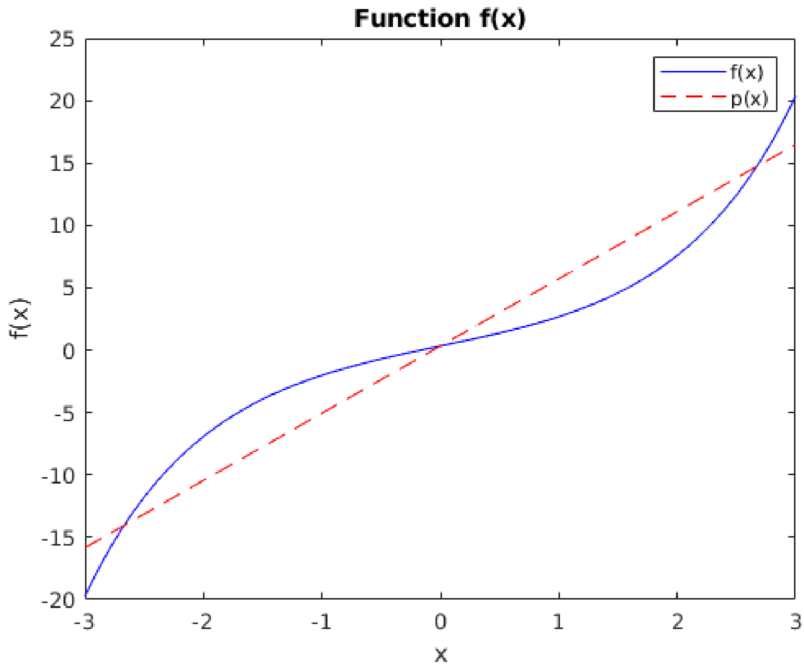

Figure 1: Plot of $f(x)$ and $p(x)$ over $[-3, 3]$.

The L_∞ error is calculated as:

$$L_\infty = \max_{a \le x \le b} |f(x) - p(x)| = \max_{-3 \le x \le 3} |(\frac{1}{3} + 2sinh(x)) - (\frac{1}{3} + 5.3872x)| = 3.8741 \tag{13}$$

The L_2 error is calculated as:

$$\sqrt{\int_a^b |e(x)|^2 dx} = \sqrt{\int_{-3}^3 |2sinh(x) - 5.3872x|^2 dx} = 6.625 \tag{14}$$

2 Problem 2

This function can be approximated in a piecewise fashion by dividing it into 2 sub-functions that intersect at x = 3. I chose to build 3rd order polynomials to approximate each part of this function. I used a least squares approach with SVD to find the coefficients of the polynomials. The complete code for this problem is given in code/q2.m and the plot of my approximations are shown below.

I built the matrix A as follows:

$$A = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 \end{bmatrix} \tag{15}$$

Where:

$$\phi_1 = \begin{bmatrix} 1 \\ . \\ . \\ . \\ 1 \end{bmatrix} \tag{16}$$

$$\phi_2 = \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_n \end{bmatrix} \tag{17}$$

$$\phi_3 = (\phi_2)^2, \phi_4 = (\phi_2)^3 \tag{18}$$

Then I used SVD to obtain the S, V, U matrices and calculated the coefficients as:

$$\bar{x} = V \cdot S^{-1} \cdot U' \cdot f_i \tag{19}$$

And I obtained the final expressions for functions A and B (as shown in figure below):

$$p_A(x) = 7.9936e^{-15} - 1.954e^{-14}x + 1.5987e^{-14}x^2 + x^3 \tag{20}$$

$$p_B(x) = 22.5 + 3x - 0.5x^2 - 5.2215e^{-16}x^3 \tag{21}$$

It is possible that the extremely small terms could be rounded off to 0.

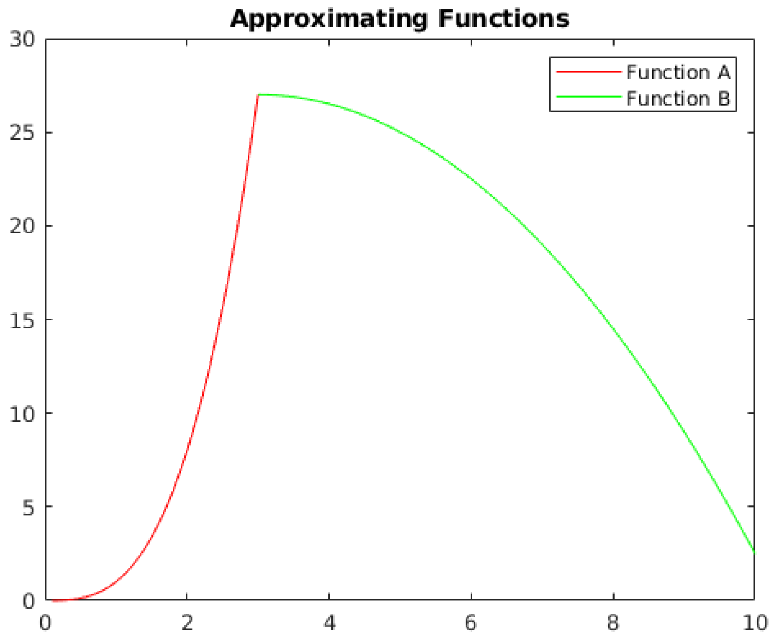

Figure 2: Plot of the two approximating functions used to represent the given function.

3 Problem 3

3.1 Part c)

The definition of the length of T_n is:

$$||T_n|| = \sqrt{< T_n, T_n >} \tag{22}$$

I can compute the inner product of T_n using the expression shown in part b), but first I will perform a substitution of variables: $x = cos(n\theta)$. This substitution of variables means that my integral now runs from 0 to π and $dx = sin(n\theta)d(n\theta)$.

$$< T_n, T_n >= \int_0^\pi (1 - cos(n\theta)^2)^{-\frac{1}{2}} cos(n\theta)cos(n\theta)sin(n\theta)d(n\theta) \tag{23}$$

$$< T_n, T_n >= \int_0^\pi \frac{cos^2(n\theta)}{sin(n\theta)} sin(n\theta)d(n\theta) = \frac{1}{2n}(n\theta + sin(n\theta)cos(n\theta))|_0^\pi = \frac{\pi}{2} \tag{24}$$

Therefore the length is:

$$||T_n|| = \sqrt{\frac{\pi}{2}} \tag{25}$$

3.2 Part d)

Please note that I used the following reference to help me solve this problem:

(1) Mahaffy, Joe. Numerical Analysis and Computing: Lecture Notes 12, Approximation Theory, Chebyshev Polynomials and Least Squares, redux. Department of Mathematics, San Diego State University. Spring 2010. `https://jmahaffy.sdsu.edu/courses/s10/math541/lectures/pdf/week12/lecture.pdf`

I want to show that any 2 terms, T_i and T_j, where $i \neq j$, are orthogonal. I can do this by substituting terms into the expression for the inner product as I did in part c):

$$< T_i, T_j >= \int_{-1}^{1} T_i \cdot T_j dx = \int_{0}^{\pi} cos(i\theta)cos(j\theta)d\theta \tag{26}$$

Using a trigonometric identity, I can rewrite this expression as:

$$< T_i, T_j >= \int_{0}^{\pi} \frac{cos((i+j)\theta) + cos((i-j)\theta)}{2} d\theta \tag{27}$$

Then we integrate:

$$< T_i, T_j >= \frac{1}{2(i+j)} sin((i+j)\theta) + \frac{1}{2(i-j)} sin((i-j)\theta)\Big|_0^{\pi} \tag{28}$$

We find that the $sin(\theta)$ terms are always zero at π and 0, so the entire integral is 0. Therefore, T_i and T_j are always orthogonal.

4 Problem 4

I used the following references to help me solve this problem:

(1) Derpanis, K. "Overview of the RANSAC Algorithm." May 13, 2010. `http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf` Visited 10/29/2019.

(2) Wikipedia. "Random sample consensus." `https://en.wikipedia.org/wiki/Random_sample_consensus` Visited 10/29/2019.

(3) Math Insight. "Distance from Point to Plane." `https://mathinsight.org/distance_point_plane` Visited 10/29/2019.

(4) Rosenberg, J. "Lines, Planes and MATLAB." 2009. `http://www2.math.umd.edu/~jmr/241/lines_planes.html` Visited 10/29/2019.

Please note that the code for all parts of this problem is contained in code/q4.m.

4.1 Part a)

I wrote a function that uses the Random Sample Consensus algorithm (RANSAC) to compute the best fit plane to the data set. My fitted plane and the data is shown in the figures below. I calculated the average distance of a point in the data set to the fitted plane as 0.0159 m.

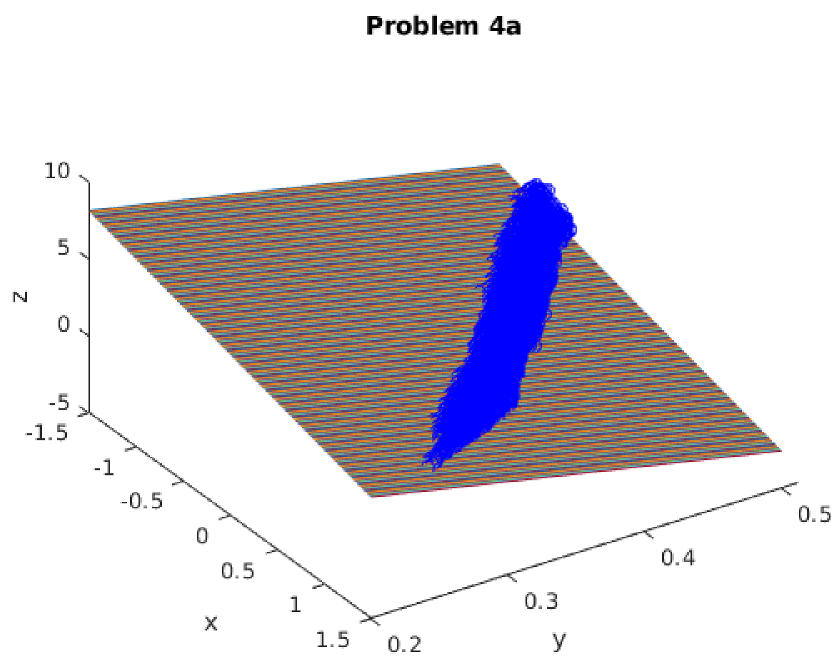

Figure 3: Plot for 4a).

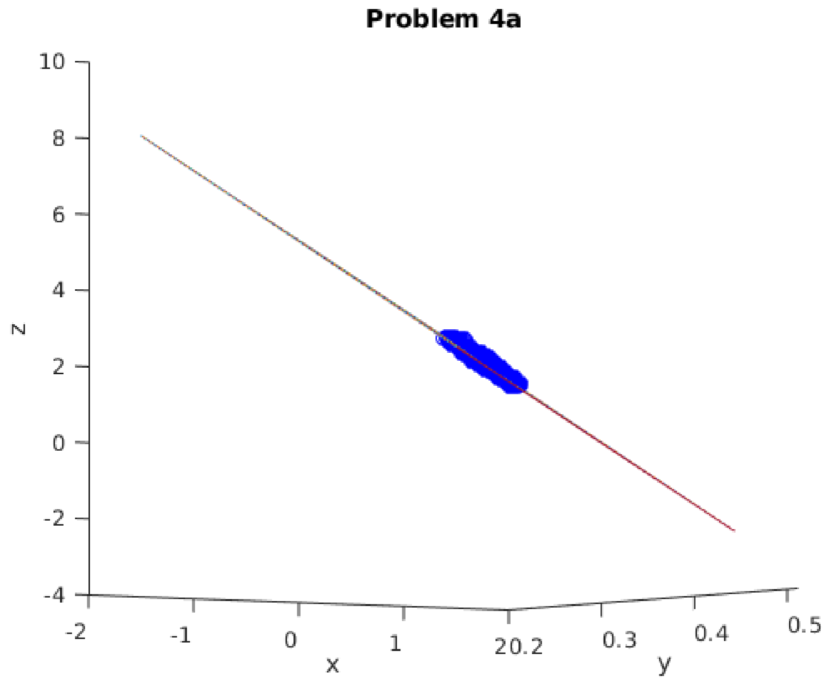

Figure 4: Plot for 4a).

4.2 Part b)

The results of my RANSAC algorithm with the same hyperparameters as used in part a (max number of iterations, threshold distance and inlier ratio) is shown in the plots below. The algorithm was mostly able to ignore the influence of the cat on the data set. However, you can see that the plane fitted to the non-outlier data is a slightly worse fit than that shown in part (a). The algorithm was likely influenced by some of the outlier points that were not filtered out by the hyperparameters, which were loose in part (a) because the original data set did not have significant outliers.

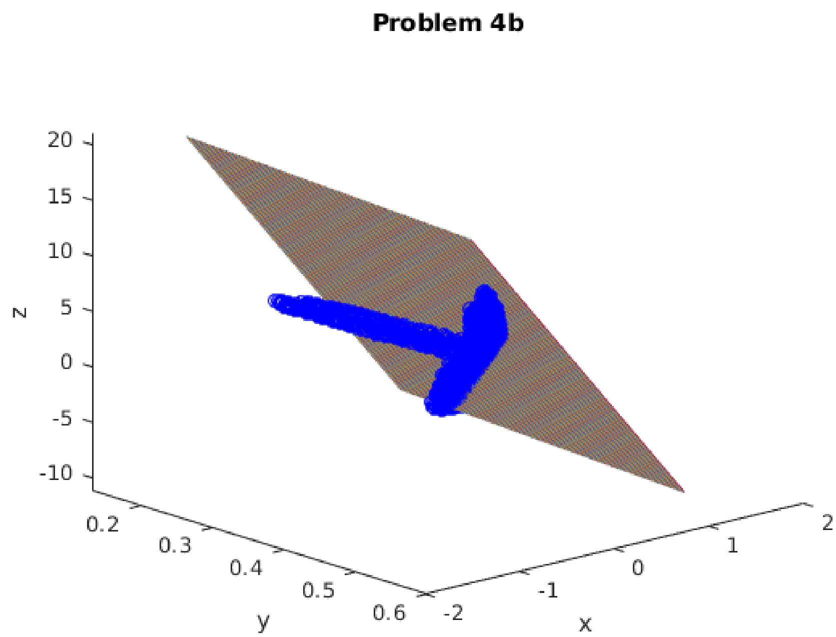

Figure 5: Plot for 4b).

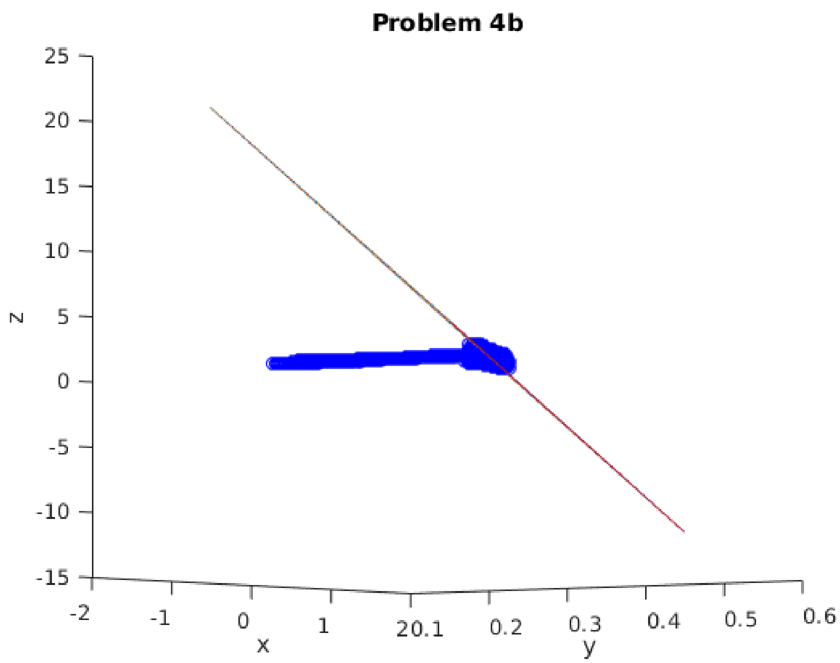

Figure 6: Plot for 4b).

4.3 Part c)

To answer this question I tried to make the RANSAC algorithm's hyperparameters more stringent, and I observed several things in the process. The first is that the RANSAC algorithm does not return exactly the same fitted plane every time, because it takes a random sampling of 3 points on each iteration from which to build the fitted plane. Depending on the initial random sampling, the final converged plane can look slightly different.

I increased the number of iterations from 5 to 10 to try to improve the fit. I think this change did show a slight improvement in the plane fit. However, increasing the number of iterations also increased the likelihood of converging to a plane that fit with the outlier data points rather than the inlier data points.

I also reduced the threshold distance from 2.0 m to 1.9 m. It seemed that 1.9 m was the lowest value I could use - the algorithm could not converge on a better plane fit than the starting value for smaller threshold distances. (Note that I started my algorithm with all the plane equation coefficients equal to 1 instead of 0. This may influence the algorithm's ability to converge on a solution.)

Next I tried to increase the inlier ratio to make the requirements for convergence more stringent. However, this also prevented the algorithm from converging at all, so I had to keep my inlier ratio at 0.5.

The final parameters given above tended to result in a good plane fit, taking into account the fact that the random sampling step in the algorithm did cause it to converge to a poor fit occasionally.

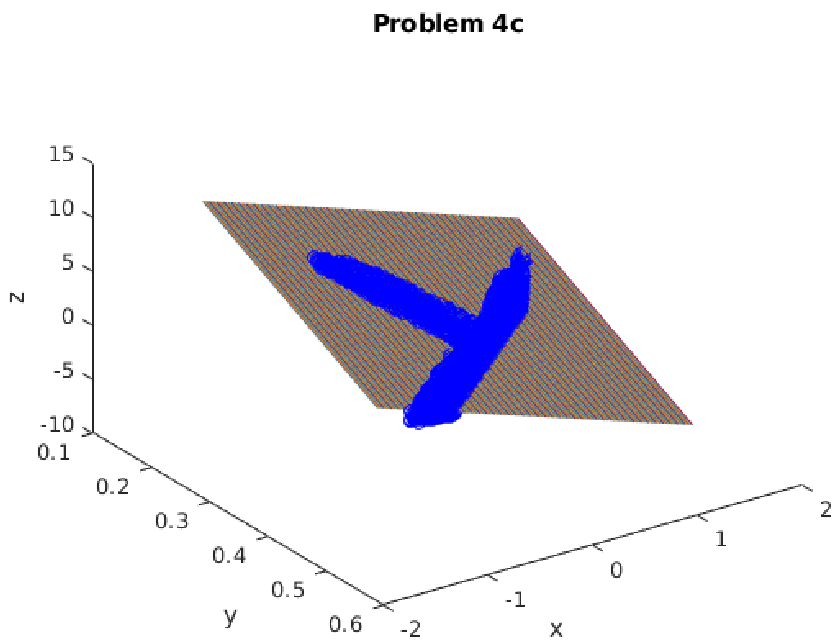

Figure 7: Plot for 4c).

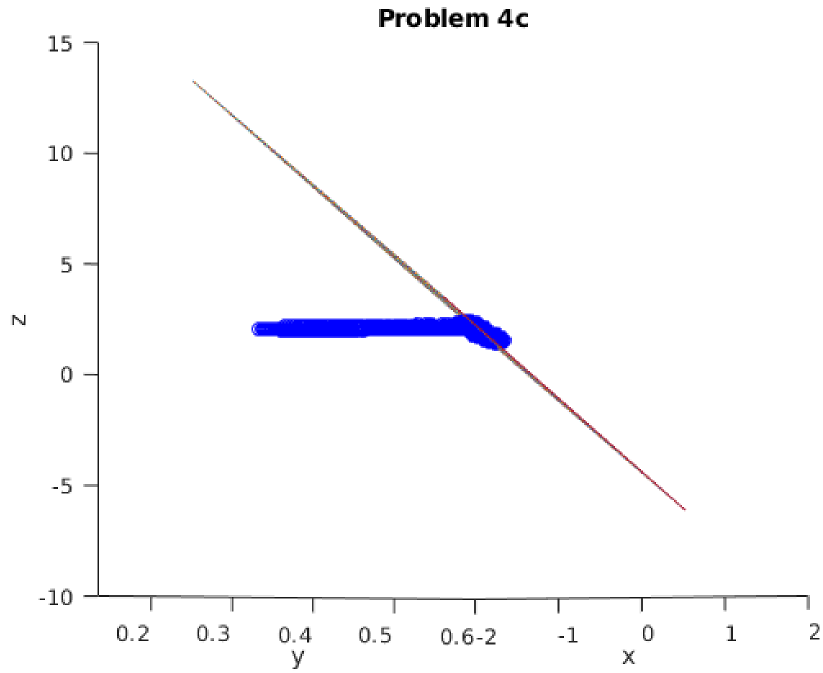

Figure 8: Plot for 4c).

4.4 Part d)

I noticed that the data was clustered into 4 groups for each of the walls, so I simply divided the data set into 4 groups and used my RANSAC algorithm to fit a plane to each group of points. The results are shown in the figures below.

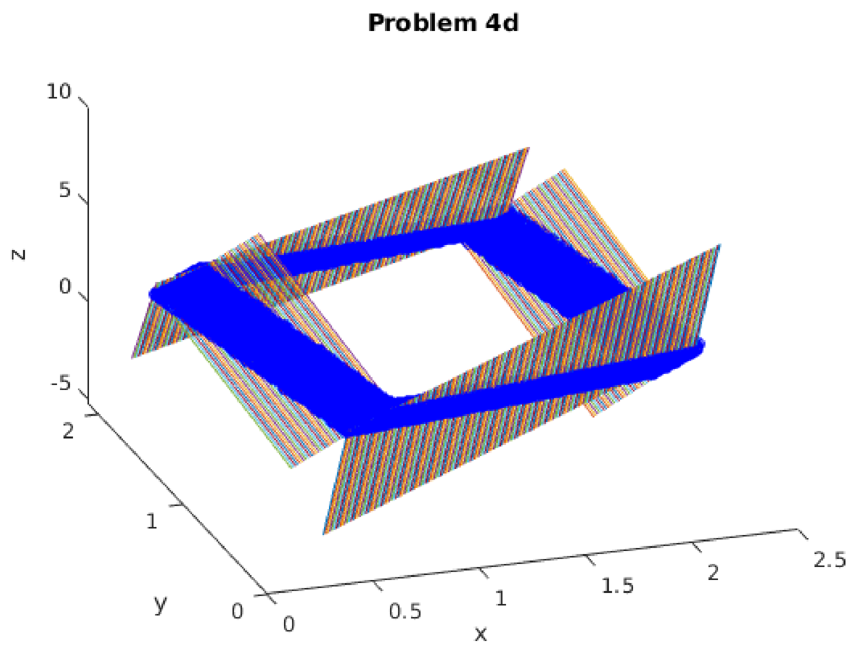

Figure 9: Plot for 4d).

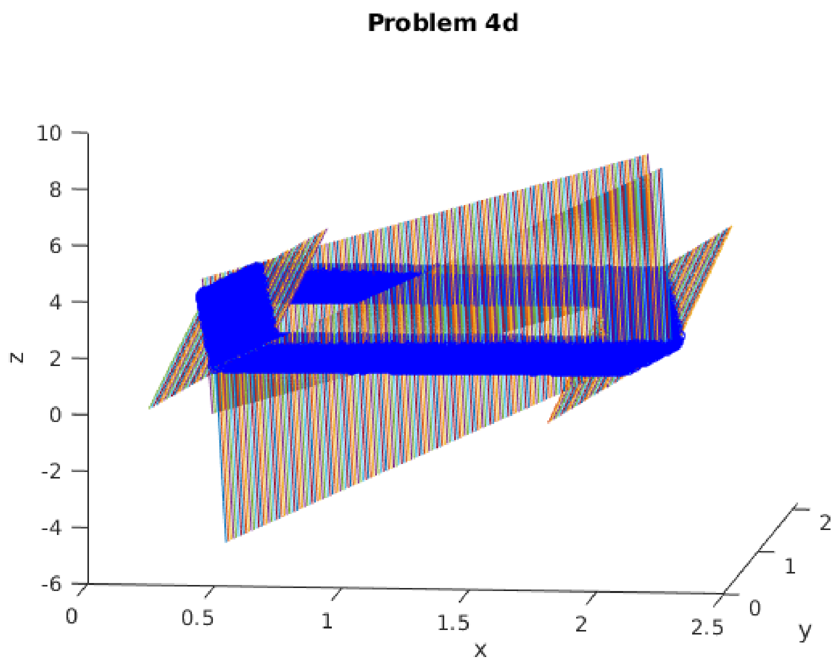

Figure 10: Plot for 4d).

4.5 Part e)

My plane fits to the cluttered hallway data set are shown in the figure below. I chose to characterize the smoothness of each surface as the average distance of the subgroups of points from their respective plane fit. If the average distance is larger, it means that the surface of the wall is more rough. For the plots shown below, the average smoothness characteristic for each wall was $[0.1247m, 1.1381m, 0.7706m, 0.0230m]$. From these values, we can see that the second wall is the most rough and the final wall is the smoothest.

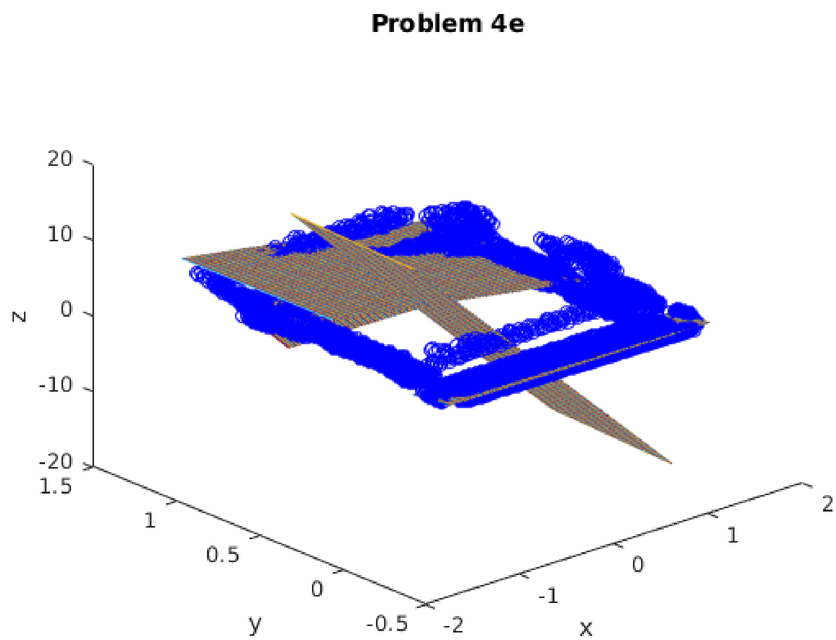

Figure 11: Plot for 4e).

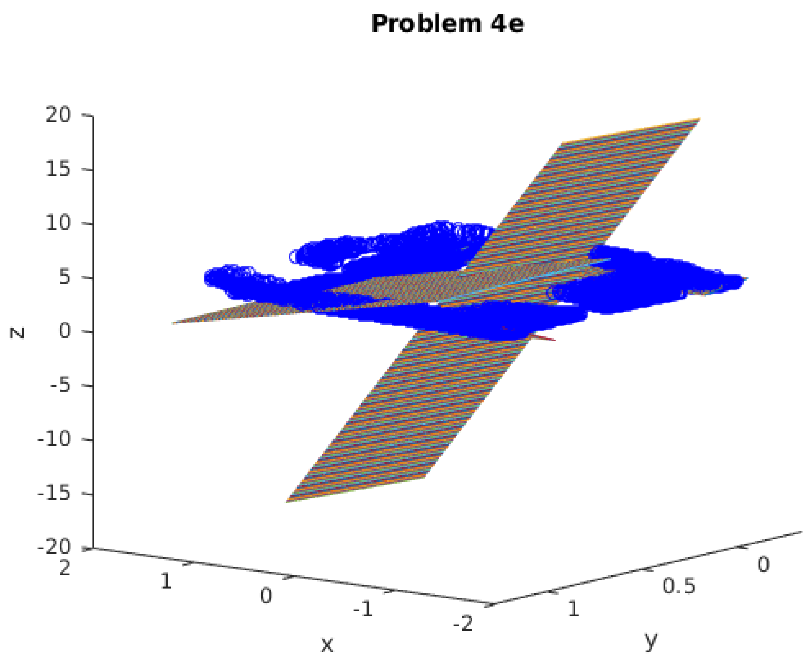

Figure 12: Plot for 4e).