Problem 1

Function given is $f(x) = \frac{1}{3} + 2sinh(x)$ over the interval $[-3, 3]$

(a) The Taylor series expansion for any function $f(x)$ about a point $f(x_0)$ can be given as:

$$f(x) = f(x_0) + \frac{f^{'}(x_0)}{1!}(x - x_0) + \frac{f^{''}(x_0)}{2!}(x - x_0)^2 + \frac{f^{'''}(x_0)}{3!}(x - x_0)^3 + \dots$$

For the given function about a point $f(0)$ we have the taylor's series:

$$\begin{aligned} f(x) &= \frac{1}{3} + 2sinh(0) + \frac{2cosh(0)}{1!}(x - 0) + \frac{2sinh(0)}{2!}(x - 0)^2 + \frac{2cosh(0)}{3!}(x - 0)^3 + \dots \\ &= \frac{1}{3} + 2x + \frac{x^3}{3} + \frac{x^5}{60} + \dots \end{aligned}$$

(b) The graph for the function is as shown below:

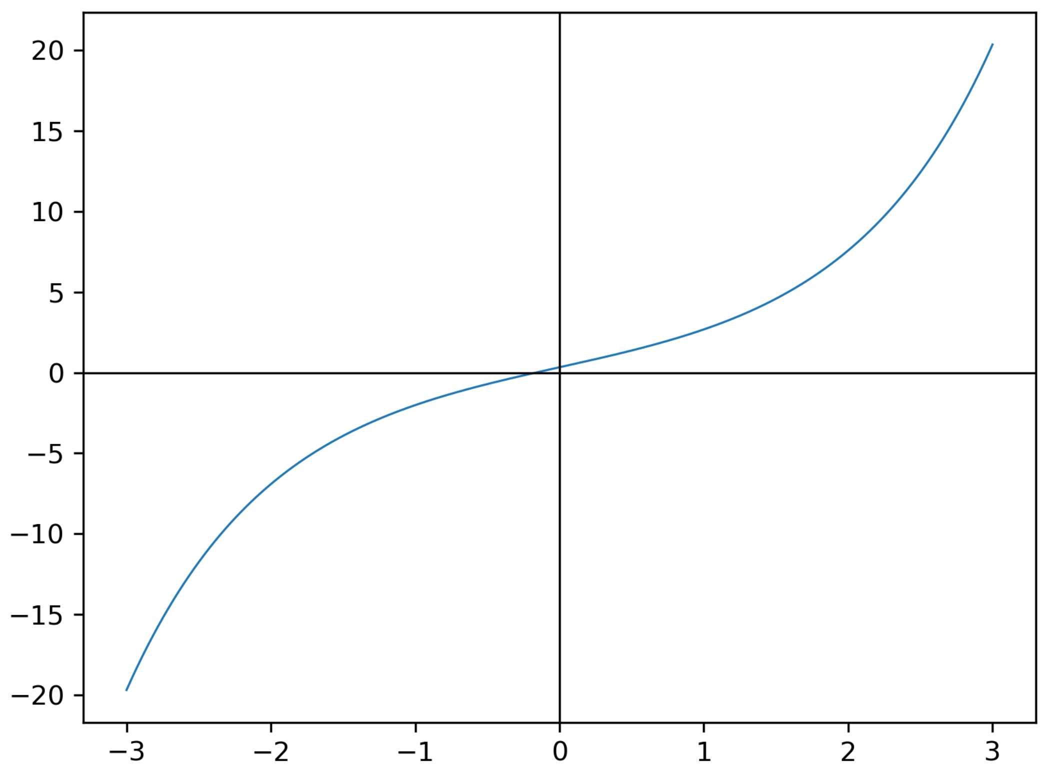


Figure 1: Graph for $f(x)$

(c) The best uniform approximation of the function $f(x) = \frac{1}{3} + 2sinh(x)$ which is quadratic can be written as $ax^2 + bx + c$.

We require that there are 4 points on the function which result in maximum error, as required by the Remez' theorem [1]. Further, we have that $f^{'''}(x)$ does not change sign on $[-3, 3]$, and thus we have the points $x_0 = -3$ and $x_3 = 3$. We need to find two more points x_1, x_2 at which we achieve maximal error.

From the theorem, we have the errors as:

$$\begin{aligned} e(x_0) = f(x_0) - p(x_0) \qquad &= f(-3) - p(-3) = \frac{1}{3} + 2sinh(-3) - 9a + 3b - c \\ e(x_3) = f(x_3) - p(x_3) \qquad &= f(3) - p(3) = \frac{1}{3} + 2sinh(3) - 9a - 3b - c \end{aligned}$$

We know that at x_0 and x_3 the errors are in opposite direction, since there are 3 intersecting points between $p(x)$ and $f(x)$, therefore:

$$\begin{aligned} e(-3) &= -e(3) \\ \frac{1}{3} + 2sinh(-3) - 9a + \cancel{3b} - c &= -\frac{1}{3} - 2sinh(3) + 9a + \cancel{3b} + c \\ \frac{2}{3} + 2sinh(-3) + 2sinh(3) &= 18a + 2c \\ 18a + 2c - \frac{2}{3} &= 0 \end{aligned}$$

We observe from the Taylor's expansion of $f(x)$ in part (a), and we know that there are no quadratic terms (function is odd) in the function. We set $a = 0$, and then we have $c = \frac{1}{3}$.

To find the value of b, we know that errors are maximum at the points x_1 and x_2. This means the error achieves extrema at these points:

$$\begin{aligned} e^{'}(x_1) &= 0 = e^{'}(x_2) \\ e^{'}(x) &= 2cosh(x) - (2ax + b) \\ e^{'}(x) &= 2cosh(x) - b \\ \implies b &= 2cosh(x) \end{aligned} \tag{1}$$

We also have:

$$\begin{aligned} e(-3) &= -e(x_1) \\ \cancel{\frac{1}{3}} + 2sinh(-3) + 3b - \cancel{\frac{1}{3}} &= -\cancel{\frac{1}{3}} - 2sinh(x_1) + bx_1 + \cancel{\frac{1}{3}} \\ \implies bx_1 - 2sinh(x_1) - 3b - 2sinh(-3) &= 0 \end{aligned} \tag{2}$$

Solving 1 and 2 we have $x_1 = -1.64765$ Similarly we solve for x_2 as follows:

$$\begin{aligned} & e(3) = -e(x_2) \\ & \cancel{\frac{1}{3}} + 2sinh(3) - 3b - \cancel{\frac{1}{3}} = -\cancel{\frac{1}{3}} - 2sinh(x_2) + bx_2 + \cancel{\frac{1}{3}} \\ \implies & bx_2 - 2sinh(x_2) + 3b - 2sinh(3) = 0 \end{aligned} \tag{3}$$

Solving 1 and 3 we have $x_2 = +1.64765$. Therefore we have $b = 2cosh(x_1) = 2cosh(x_2) = 5.38726$

Finally we have the best uniform approximating quadratic as follows:

$$p(x) = 5.38726x + \frac{1}{3} \tag{4}$$

We then calculate the L_∞ and L_2 error for this polynomial:

$$\begin{aligned} L_\infty &= \max_{-3\le x\le 3} |f(x) - p(x)| \\ L_\infty &= \max_{-3\le x\le 3} \left|\frac{1}{3} + 2sinh(x) - 5.38726x - \frac{1}{3}\right| \\ L_\infty &= 3.8739 \\ L_2 &= \sqrt{\int_{-3}^{3} \left|\frac{1}{3} + 2sinh(x) - 5.38726x - \frac{1}{3}\right|^2 dx} \\ L_2 &= 6.62518 \end{aligned}$$

(d) We want to find the quadratic polynomial which minimizes the error below:

$$\int_{-3}^{3} \left(\frac{1}{3} + 2sinh(x) - p(x)\right)^2 dx$$

where $p(x)$ ranges over all the polynomials of degree at most 2. We will construct an orthogonal basis of polynomials, using the recurrence as shown below:

$$p_{i+1}(x) = \left[x - \frac{< xp_i, p_i >}{< p_i, p_i >}\right] p_i(x) - \frac{< p_i, p_i >}{< p_{i-1}, p_{i-1} >} p_{i-1}(x) \qquad \text{where } i = 0, 1, \ldots$$

with $p_0(x) = 1$ and $p_{-1}(x) = 0$. We also define the inner product above as follows:

$$< g, h >= \int_{-3}^{3} g(x)h(x)dx$$

. Now we compute p_0, p_1 and p_2: Define $p_0(x) = 1$

$$\begin{aligned} \implies < p_0, p_0 > &= \int_{-3}^{3} 1 \cdot 1dx = 3 + 3 = 6 \\ < xp_0, p_0 > &= \int_{-3}^{3} x \cdot 1dx = \frac{9}{2} - \frac{9}{2} = 0 \end{aligned}$$

$$p_1(x) = [x-0]\cdot 1 - \frac{6}{1}\cdot 0 = x$$
$$< p_1, p_1 > = \int_{-3}^{3} x^2 dx = 9+9 = 18$$
$$< xp_1, p_1 > = \int_{-3}^{3} x^3 dx = \frac{81}{4} - \frac{81}{4} = 0$$

$$p_2(x) = [x-0]\cdot x - \frac{18}{6}\cdot 1 = x^2 - 3$$
$$< p_2, p_2 > = \int_{-3}^{3} (x^2-3)(x^2-3) = \left[\frac{x^5}{5} - \frac{2x^3}{9} + \frac{x}{9}\right]_{-3}^{3} = 85.867$$

This will construct a best approximation $p(x)$ by orthogonally projecting $f(x)$ onto the subspace of functions spanned by $\{p_0, p_1, p_2\}$ Therefore we have the polynomial

$$p(x) = \sum_{i=0}^{2} \frac{< \frac{1}{3} + 2sinh(x), p_i >}{< p_i, p_i >} p_i(x) \tag{5}$$

We compute

$$< f(x), p_0 > = \int_{-3}^{3} \frac{1}{3} + 2sinh(x) = \left[\frac{x}{3} + 2cosh(x)\right]_{-3}^{3} = 2$$

$$\begin{aligned} < f(x), p_1 > &= \int_{-3}^{3} \frac{x}{3} + 2x\cdot \sinh(x).dx \\ &= \left[\frac{x^2}{6} + 2(xcosh(x) - sinh(x))\right]_{-3}^{3} \\ &= 80.740 \end{aligned}$$

$$\begin{aligned} < f(x), p_2 > &= \int_{-3}^{3} \left(\frac{1}{3} + 2sinh(x)\right)\left(x^2-3\right) dx \\ &= \left[\frac{x^3}{9} - \frac{x}{9} - \frac{2}{3}cosh(x) + 2x^2cosh(x) - 4xsinh(x) + 4cosh(x)\right]_{-3}^{3} \\ &= 0 \end{aligned}$$

Finally, we have the polynomial from 5

$$p(x) = \frac{2}{6}\cdot 1 + \frac{80.740}{18}\cdot x + \frac{0}{85.867}\cdot (x^2-3)$$
$$p(x) = \frac{1}{3} + 4.4856x \tag{6}$$

For the best least squares approximation polynomial we found in 6, the errors are as below:

$$L_\infty = \max_{-3 \le x \le 3} |f(x) - p(x)|$$
$$L_\infty = \max_{-3 \le x \le 3} \left| \frac{1}{3} + 2sinh(x) - 4.4856x - \frac{1}{3} \right|$$

This error can be max when $\frac{dL_\infty}{dx} = 0$ or at end points

$$\frac{dL_\infty}{dx} = \frac{(2sinh(x) - 4.4856x)(2cosh(x) - 4.4856)}{|2sinh(x) - 4.4856x|} = 0$$
$$\implies x = \{-1.447, 1.447\}$$

Calculating the error at $x = \{-3, -1.447, 1.447, 3\}$ we obtain:

$$L_\infty = 6.5789$$

L_2 error can be solved trivially by computing the integral in the given limits:

$$L_2 = \sqrt{\int_{-3}^{3} \left| \frac{1}{3} + 2sinh(x) - 4.4856x - \frac{1}{3} \right|^2 dx}$$
$$L_2 = 5.409$$

Problem 2

We observe from the plot of the points given as in Figure 2, that the function is non-differentiable at $x = 3.0$, therefore we can approximate the points as a piece-wise combination of two curves. Further we can also see that the second order differential of the curve shows a large variation at $x = 3.0$ as shown in Figure 3

We will choose the basis of polynomials $\{\phi_1, \phi_2, \phi_2, \phi_3\} = \{1, x, x^2, x^3\}$ which spans the vector space of cubics to approximate the two parts of the plot, with $x \le 3.0$ and $x > 3.0$.

$$F(x) = c_1\phi_1(x) + c_2\phi_2(x) + c_3\phi_3(x) + c_4\phi_4(x)$$

and we need to minimize the least squares error as below:

$$\|f(x) - F(x)\|^2 = \sqrt{\sum_{i=1}^{n} |f_i - F(x_i; c_1, c_2, \dots c_k)|^2}$$

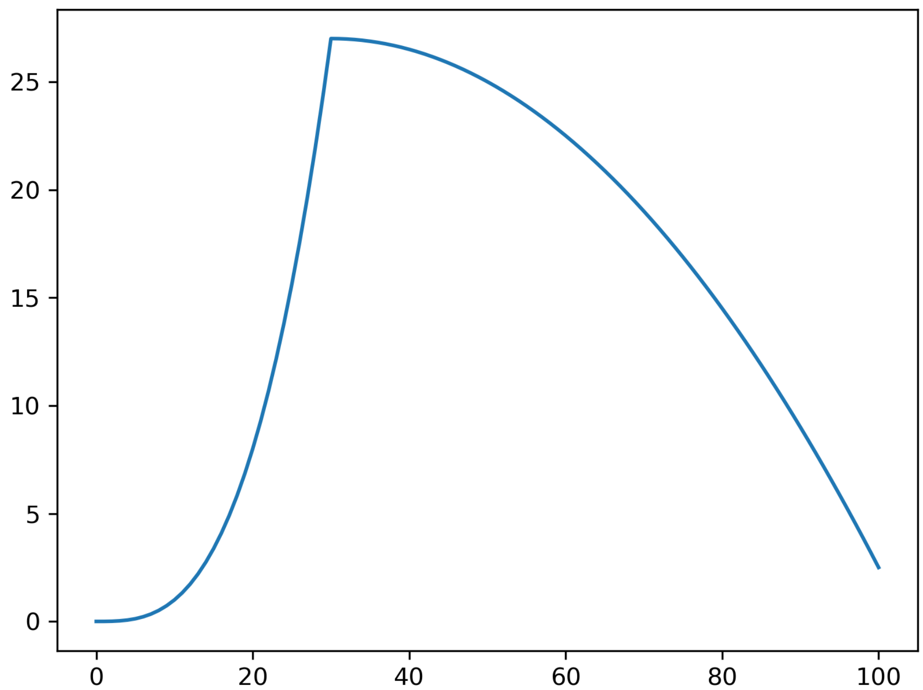


Figure 2: Plot of the give points in `problem2.txt`

We compute the normal equation matrix (for SVD), as derived in class, and compute the orthogonal projection of f. The equation is of form

$$\mathbf{A} \cdot \mathbf{c} = \mathbf{f}$$

and this is solved by taking the pseudo-inverse (by using SVD) of the system of equation to obtain the least squares solution.

The code required to solve this problem is given in `q2.py`. The approximated polynomial can now be given as:

$$F(x) = \begin{cases} x^3 & x \leq 3.0 \\ \frac{-x^2}{2} + 3x + 22.5 & x > 3.0 \end{cases}$$

Problem 3

(a) We are given that

$$T_n(\cos\theta) = \cos(n\theta) \qquad\qquad \text{for } n > 0$$

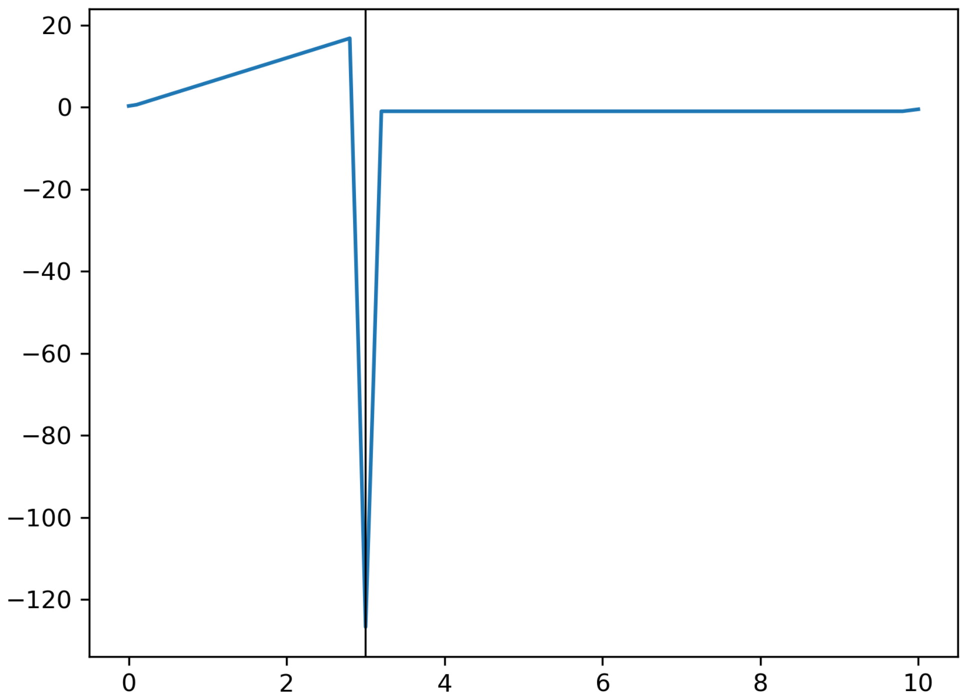


Figure 3: Second order differential of the given points. The black line shows the point at which the curve is non-differentiable.

and we obtain the recurrence relation

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \qquad\qquad \text{for } n > 0$$

We take $T_0(x) = 1$ and $T_1(x) = x$, then we compute:

$$T_2(x) = 2xT_1(x) - T_0(x) = 2x^2 - 1 \tag{7}$$
$$T_3(x) = 2xT_2(x) - T_1(x) = 2x(2x^2 - 1) - x = 4x^3 - 3x \tag{8}$$
$$T_4(x) = 2xT_3(x) - T_2(x) = 2x(4x^3 - 3x) - 2x^2 + 1 = 8x^4 - 8x^2 + 1 \tag{9}$$

(b) In the inner product given, we take $g = T_3(x)$ and $h = T_4(x)$ and write it as $f(x)$ as follows:

$$f(x) = \frac{1}{\sqrt{1-x^2}}(4x^3 - 3x)(8x^4 - 8x^2 + 1)$$

We can show that this function is odd, that is:

$$\begin{aligned} f(-x) &= \frac{1}{\sqrt{1-x^2}}(-4x^3+3x)(8x^4-8x^2+1) \\ &= -\left(\frac{1}{\sqrt{1-x^2}}(4x^3-3x)(8x^4-8x^2+1)\right) \\ &= -f(x) \end{aligned}$$

Now since $f(x) = -f(-x)$, the function is odd. When the integral of $f(x)$ is evaluated over any interval $[-a, a]$ the result is 0, therefore:

$$< T_3(x), T_4(x) > = \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}}(4x^3-3x)(8x^4-8x^2+1)dx = 0$$

This shows that the Chebyshev polynomials $T_3(x)$ and $T_4(x)$ are orthogonal relative to the inner product as defined by f(x)

(c) We calculate the length of T_n from the inner product $< T_n, T_n >$ as follows:

$$< T_n, T_n >= \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} T_n(x) \cdot T_n(x) dx$$

replacing $x = \cos(\theta)$ we have:

$$\begin{aligned} < T_n, T_n > &= \int_{\pi}^{0} \left(\frac{1}{\sqrt{1-\cos(\theta)^2}} \cos(n\theta) \cdot \cos(n\theta)\right)(-\sin(\theta))d\theta \\ &= \int_{\pi}^{0} \left(\frac{-\cos^2(n\theta)\cancel{\sin(\theta)}}{\cancel{\sqrt{1-\cos(\theta)^2}}}\right) d\theta \\ &= \int_{\pi}^{0} -\cos^2(n\theta)d\theta = -\frac{1}{2}\int_{\pi}^{0} (\cos(2n\theta)+1)\, d\theta \\ &= -\frac{1}{2}\left[\frac{sin(2n\theta)}{2n} + \theta\right]_{\pi}^{0} = \frac{\pi}{2} \end{aligned}$$

Since this integral is independent of n, it is proven that all the T_n have the same length.

(d) Now we calculate the inner product $< T_i, T_j >$ as follows:

$$< T_i, T_j >= \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} T_i(x) \cdot T_j(x) dx$$

replacing $x = \cos(\theta)$

$$\begin{aligned} < T_i, T_j > &= \int_{\pi}^{0} \left(\frac{1}{\sqrt{1-\cos(\theta)^2}} \cos(i\theta) \cdot \cos(j\theta) \right) (-\sin(\theta)) d\theta \\ &= \int_{\pi}^{0} -\cos(i\theta)\cos(j\theta) d\theta \\ &= -\frac{1}{2} \int_{\pi}^{0} \cos((i-j)\theta) + \cos((i+j)\theta) d\theta \\ &= -\frac{1}{2} \left[\frac{\sin((i-j)\theta)}{(i-j)} + \frac{\sin((i+j)\theta)}{(i+j)} \right]_{\pi}^{0} = 0 \end{aligned}$$

The above results shows that the Chebyshev polynomials are orthogonal to each other, and have the same length.

Problem 4

The solution to this problem is given in `q4.py`

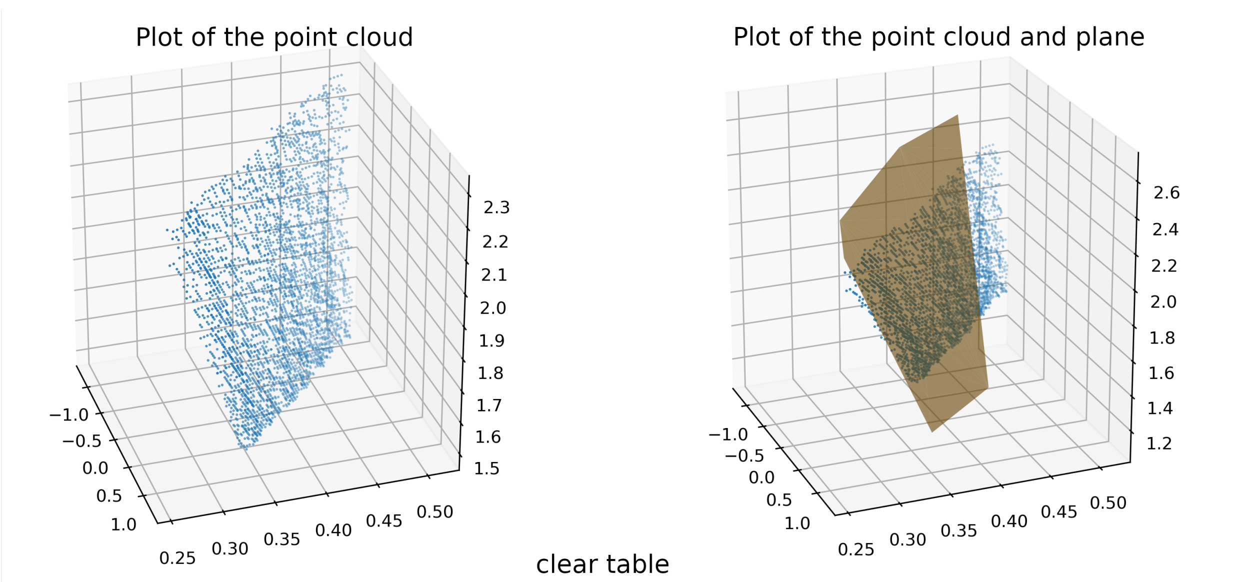


Figure 4: Plot of the original point cloud (blue) and plane overlapped (brown-orange) for `clean_table.txt`

(a) Similar to the data fitting plane, we fit a best plane in the linear least squares sense. The equation of a line is given as:

$$ax + by + cz = d$$

where x, y, z are the coordinates of the points passing through the plane and a, b, c, d are the parameters of the plane. However this being an over-constrained system, we can normalize $c = 1$ to obtain the equation:

$$ax + by + d = -z$$

With a set of given of points, to which a plane needs to be fit, we can write the above equation in matrix form as:

$$\overbrace{\begin{pmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \vdots & & \\ x_n & y_n & 1 \end{pmatrix}}^{A} \overbrace{\begin{pmatrix} a \\ b \\ d \end{pmatrix}}^{c} = \overbrace{\begin{pmatrix} -z_0 \\ -z_1 \\ \vdots \\ -z_n \end{pmatrix}}^{-z}$$

The least squares best parameters can be obtained by taking the SVD pseudo-inverse of $\mathbf{A}$, as was discussed in class. The solution is as shown in Figure 4.

Further the average distance of the points to the plane is calculate in `q4.py` and is 0.002855

(b) The plane that is fit to the noisy data `cluttered_table.txt` is as shown in Figure 5. Obviously from the image, we see that the plane that is fit is not a good one, owing to least squares fitting to the noisy points/outliers. Since the least squares metric is prone to outliers, we need to disregard them to get a good plane fit.

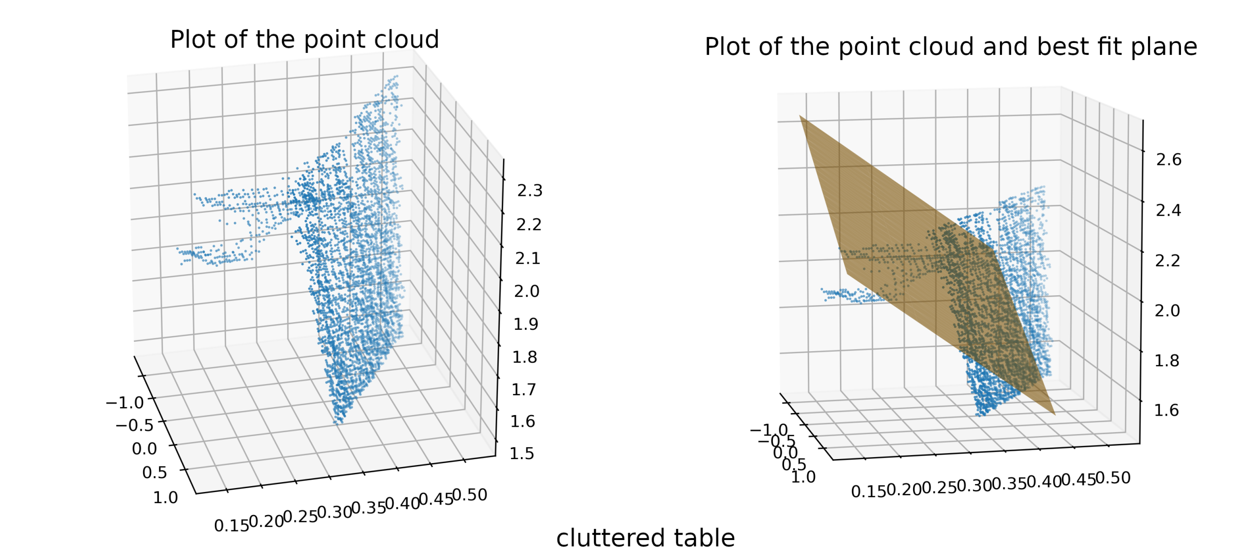
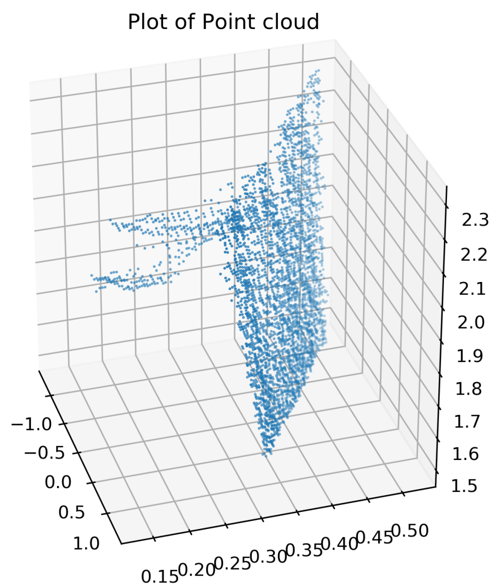


Figure 5: Plot of the original point cloud (blue) and plane overlapped (brown-orange) for `cluttered_table.txt` after using RANSAC

(c) A simple solution to find the dominant plane in the presence of few outliers, is to randomly sample the minimum number of points to solve the plane equation, and then find the best plane which retains the maximum number of inliers corresponding to a distance metric. This method is formally called RANSAC [2]. In this particular case, the minimum number of points required is 3. For a fixed number of iterations, we randomly sample 3 points, and generate plane parameters, and also compute distances to the plane. Conditioned on some tolerance value to the distance, we maintain the inliers and outliers. Finally we choose the plane parameters corresponding to the maximum inliers. Figure 6 shows the plane fit, after RANSAC method is used.



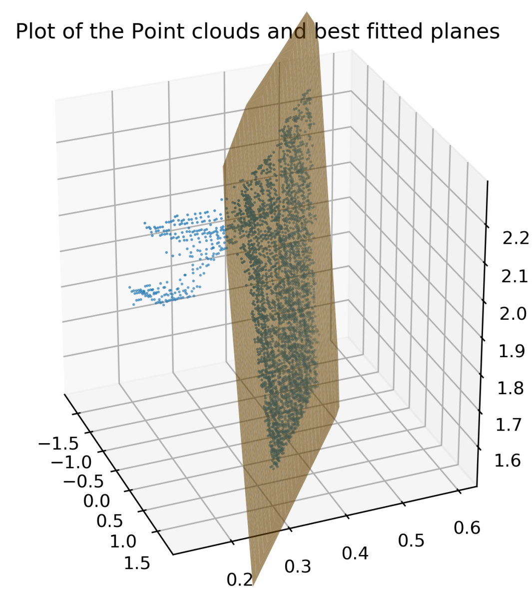


Figure 6: Plot of the original point cloud (blue) and plane overlapped plane (brown-orange) for `cluttered_table.txt`

(d) For multiple planes to be fit in the point cloud, we iterate the RANSAC algorithm as many times as required (4 in this case to fit 4 planes). In each iteration, we remove the points corresponding to the current best fit plane. This would run until there are very few points left. However, if the number of points between the multiple planes are largely unequal then we could hit cases wherein wrong planes could be fit owing to outliers being greater than the correct plane inliers. An example solution generated from `q4.py` is as shown in Figure 7

(e) As was described in the previous part, when there are largely unequal number of points, the fit planes are not always the correct ones. For this part of the problem, the tolerance required for each plane is manually tuned, to generate a fit as shown in Figure 8. I

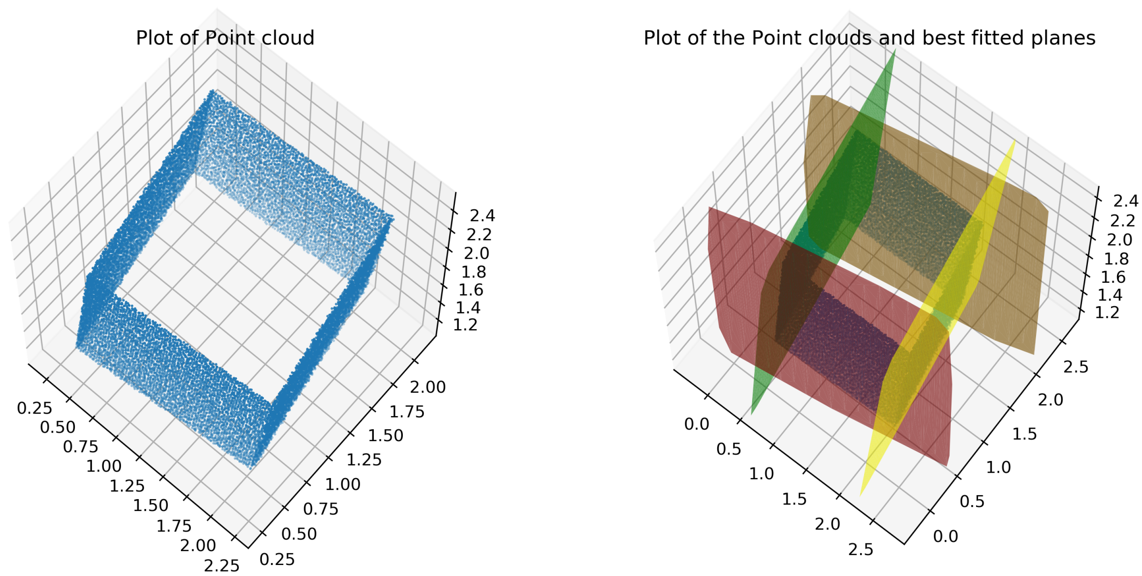


Figure 7: Plot of the original point cloud (blue) and plane overlapped planes for `clear_hallway.txt` Red = Plane 1, Orange = Plane 2, Green = Plane 3, Yellow = Plane 4

have used the following metric to obtain the smoothness of the plane:

$$smoothness = \frac{no.of inliers * tolerance}{\sum d_i}$$

where d_i is the distance of each point cloud in the inlier set of the plane. Intuitively, for a plane to be smooth, it is required that all the inliers of the plane have very low normal projection distances to the plane. Further, a larger number of inliers for a plane, corresponds to dense and rigid planes. Using the metric formulated above, from the code, the smoothest plane is the second plane (in Orange) obtained after running RANSAC, which generates a smoothness value of 3.94. The smoothness value of the planes are as given below:

```
[Red = 3.656, Orange = 3.947, Green = 2.399, Yellow = 2.173]
```

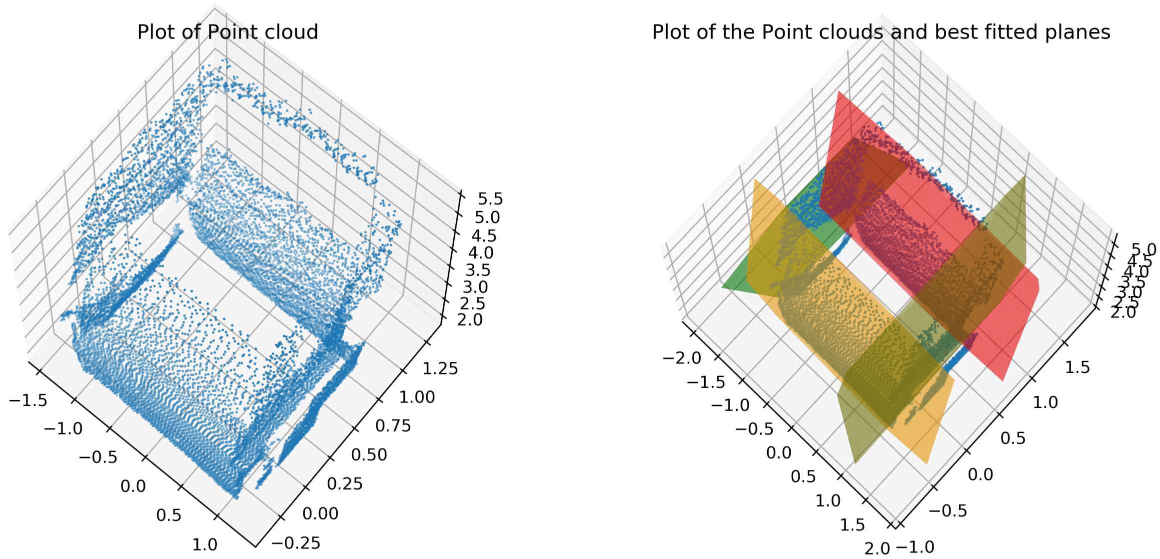


Figure 8: Plot of the original point cloud (blue) and plane overlapped planes for `cluttered_hallway.txt` Red = Plane 1, Orange = Plane 2, Green = Plane 3, Yellow = Plane 4

Discussion Citation

For this Homework set, I have discussed with *Tarasha Khurana* andrewID: `tkhurana@andrew.cmu.edu`, and *Viraj Parimi* andrewID: `vparimi@andrew.cmu.edu`

References

[1] Remez Algorithm
`https://en.wikipedia.org/wiki/Remez_algorithm`

[2] Fischler, Martin A. and Bolles, Robert C, *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography* Commun. ACM June 1981 vol. 24 num 6