HW3

Lifan Yu lifany

October 2023

1

To run program, use the following command. Output images are saved in q1-b.png, q1-c.png, q1-d.png

```
python q1.py
```

(a)

Taylor series expansion of $f(x)$ around a point c is $f(x) = f(c) + f'(c)(x-c) + \frac{f''(c)}{2!}(x-c)^2 + ...$

For $f(x) = cos(\frac{\pi}{2}x), x \in [0,2]$ around $x = 1$:

$f(1) = cos(\frac{\pi}{2}) = 0$

$f'(1) = -\frac{\pi}{2}sin(\frac{\pi}{2}) = -\frac{\pi}{2}$

$f''(1) = -\frac{\pi^2}{4}cos(\frac{\pi}{2}) = 0$

$f'''(1) = \frac{\pi^3}{8}sin(\frac{\pi}{2}) = \frac{\pi^3}{8}$

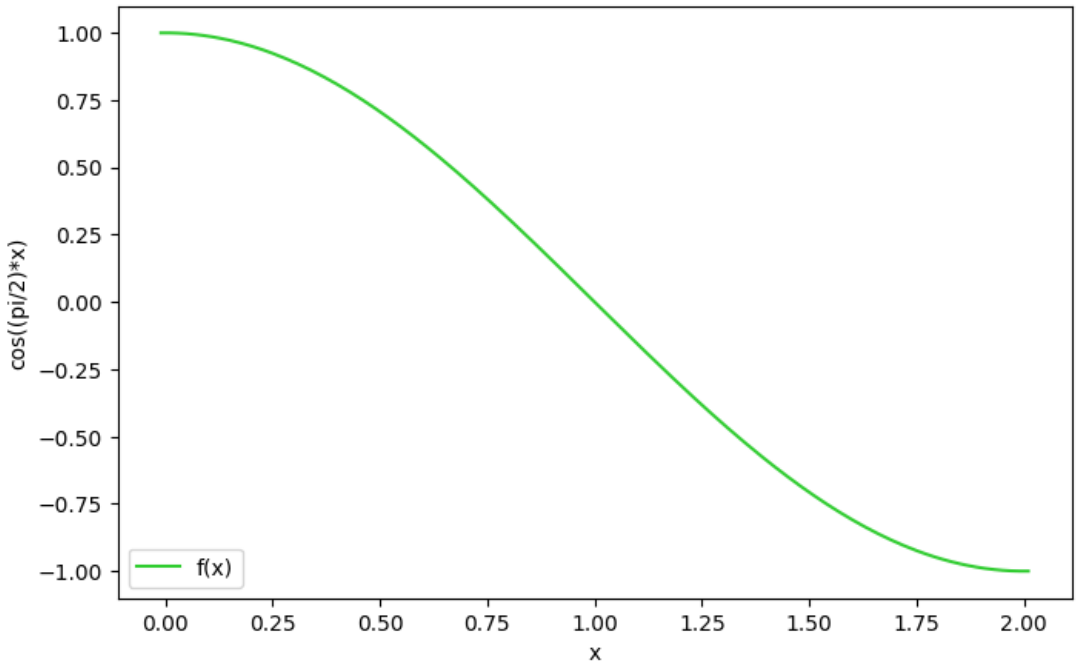
Taylor series expansion of f(x) around x = 1 is

$$f(x) = 0 - \frac{\pi}{1! * 2}(x-1) + 0 + \frac{\pi^3}{3! * 2^3}(x-1)^3 + 0 - \frac{\pi^5}{5! * 2^5}(x-1)^5 + ...$$

$$= \sum_{k=0}^{\infty}(-1)^{k+1}\frac{\pi^{1+2k}}{(1+2k)! * 2^{1+2k}}(x-1)^{1+2k}$$

(b)

Graph of $f(x) = cos(\frac{\pi}{2}x), x \in [0,2]$



(c)

$f'''(x) = \frac{\pi^3}{8} sin(\frac{\pi}{2}x)$ does not change sign over $[0, 2]$. therefore, there are 4 points x_0, x_1, x_2, x_3 at which the error $f(x) - p(x)$ is greatest, with equal magnitude and alternating sign. Also, $x_0 = 0, x_3 = 2$. The 4 points are then $-\frac{\pi}{2}, x_1, x_2, \frac{\pi}{2}$. Write the polynomial as $p(x) = ax^2 + bx + c$.

We compute the errors

$e(x_0) = f(x_0) - p(x_0) = f(0) - p(0) = cos(0) - c = 1 - c$
$e(x_1) = f(x_1) - p(x_1) = cos(\frac{\pi}{2}x_1) - ax_1^2 - bx_1 - c$
$e(x_2) = f(x_2) - p(x_2) = cos(\frac{\pi}{2}x_2) - ax_2^2 - bx_2 - c$
$e(x_3) = f(x_3) - p(x_3) = f(2) - p(2) = cos(\pi) - (4a + 2b + c) = -1 - 4a - 2b - c$

For the errors at these 4 points, we have

$e(x_0) = -e(x_1) = e(x_2) = -e(x_3) = ||e(x)||_\infty$
$1 - c = -cos(\frac{\pi}{2}x_1) + ax_1^2 + bx_1 + c = cos(\frac{\pi}{2}x_2) - ax_2^2 - bx_2 - c = 1 + 4a + 2b + c$

We then solve the equations

Solving $e(x_0) = e(x_3)$, $1 - c = 1 + 4a + 2b + c$ gives $2b + 2c = -4a$ (1)

We know $cos(\frac{\pi}{2} + m) = -cos(\frac{\pi}{2} - m), m \in R$, then $cos(\frac{\pi}{2}(1 + k)) = -cos(\frac{\pi}{2}(1 - k)), k \in R$, in order to approximate well, it is best to have x_1, x_2 symmetric around 1 so $cos(\frac{\pi}{2}x_1) = -cos(\frac{\pi}{2}x_2)$ (2), thus $x_1 + x_2 = 2$ (3)

Solving $e(x_1) = e(x_2)$, $-cos(\frac{\pi}{2}x_1) + ax_1^2 + bx_1 + c = cos(\frac{\pi}{2}x_2) - ax_2^2 - bx_2 - c$ gives $a(x_1^2 + x_2^2) + 2b + 2c = cos(\frac{\pi}{2}x_1) + cos(\frac{\pi}{2}x_2)$ (4)

Substitute (1) into (4) gives $a(x_1^2 + x_2^2) - 4a = cos(\frac{\pi}{2}x_1) + cos(\frac{\pi}{2}x_2)$, substitute (2) into it gives $a(x_1^2 + x_2^2 - 4) = 0$

Because (3) gives us $(x_1 + x_2)^2 = 4$, we have $a(4 - 2x_1x_2 + 4) = 0 \rightarrow ax_1x_2 = 0$. Because $x_1, x_2 \in [0, 2], x_1 * x_2 \neq 0$, the only possible solution is

$$a = 0 (5)$$

Substituting (5) into (1), we get

$$b = -c (6)$$

We want to minimize e_i, so we can take the derivative of e_1 and let it be 0, and substituting (6), $\frac{\partial}{\partial x_1} e(x_1) = \frac{\pi}{2} sin(\frac{\pi}{2}x_1) - c = \frac{\pi}{2} sin(\frac{\pi}{2}x_1) + b = 0$, which gives us

$$x_1 = \frac{2}{\pi} sin^{-1}(\frac{2b}{\pi}) (7)$$

Substituting (7) and (6) into $e(x_1) = e(x_2)$ gives $1 + b = -cos(\frac{\pi}{2}x_1) + bx_1 - b$
$= -cos(sin^{-1}(\frac{2b}{\pi})) - \frac{2b}{\pi} sin^{-1}(\frac{2b}{\pi}) - b$
$= -\sqrt{1 - (\frac{2b}{\pi})^2} - \frac{2b}{\pi} sin^{-1}(\frac{2b}{\pi}) - b$
Arranging into
$\sqrt{1 - (\frac{2b}{\pi})^2} + b(2 + \frac{2}{\pi} sin^{-1}(\frac{2b}{\pi})) + 1 = 0$
Using Muller's method to find roots gives us

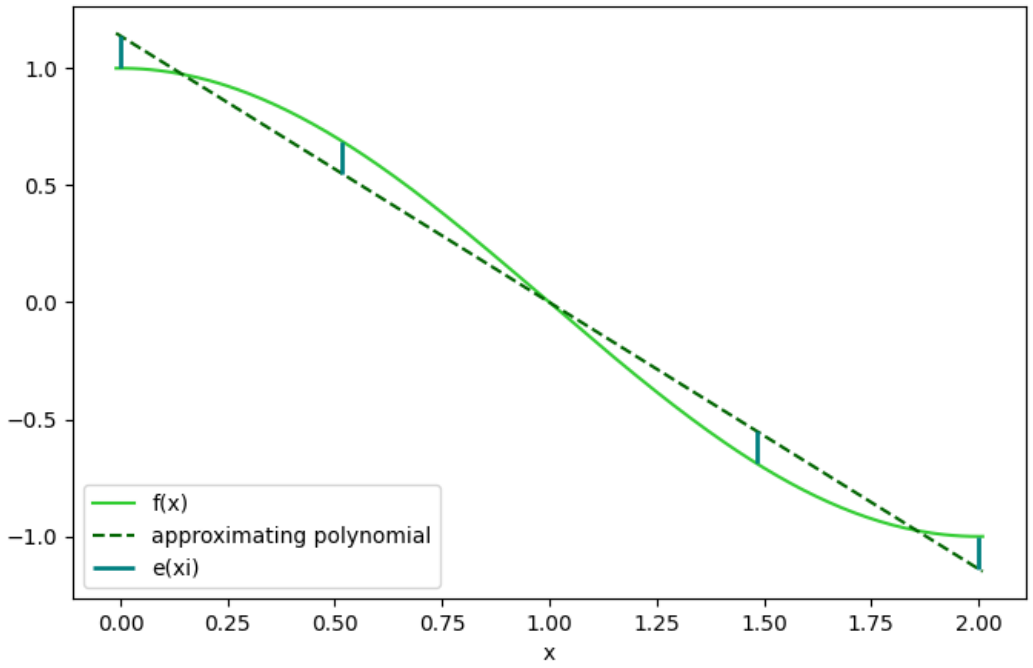
$$b \approx -1.13821685$$

Now we have

$$\boxed{\text{p(x) = -1.13821685x + 1.13821685}}$$

A graph is shown below:

Graph of $f(x) = cos(\frac{\pi}{2}x)$ and approximating polynomial (dotted line)



$x_1 = \frac{2}{\pi} sin^{-1}(\frac{2b}{\pi}) \approx 0.515961406$

L_∞ error:

$L_\infty = max(e(x)) = f(x_1) - (bx_1 + c)$
$\approx cos(\frac{\pi}{2}x_1) - (-1.13821685x_1 + 1.13821685)$
$= \boxed{0.13821685}$

L_2 error:

$L_2 = \sqrt{\int_0^2 |e(x)|^2 \, dx}$
$= \sqrt{\int_0^2 (cos(\frac{\pi}{2}x) + 1.13821685x - 1.13821685)^2 \, dx}$
$\approx \sqrt{0.0184841} \approx \boxed{0.135956}$

```
import numpy as np
import matplotlib.pyplot as plt
low, high = 0, 2
def f(x):
  return np.cos((np.pi/2)*(x))
def p(x):
  return -1.13821685*x + 1.13821685
x_plot = np.arange(low-0.01, high+0.02, .01)
y_plot = [f(item) for item in x_plot]
plt.figure(figsize = (8, 5))
plt.plot(x_plot, y_plot, "limegreen", label = "f(x)")
y_line = [p(x) for x in x_plot]
plt.plot(x_plot, y_line, "darkgreen", linestyle='--', label="approximating polynomial")
xi = [0, 0.515961406, 1.484038594, 2]
yi1 = [f(xi[0]), p(xi[1]), f(xi[2]), p(xi[3])]
yi2 = [p(xi[0]), f(xi[1]), p(xi[2]), f(xi[3])]
plt.vlines(x=xi, ymin=yi1, ymax=yi2, colors='teal', ls='-', lw=2, label='e(xi)')
plt.xlabel("x")
plt.ylabel("cos((pi/2)*x)")
leg = plt.legend(loc='lower left')
plt.show()
```

(d)

Calculating Legendre polynomials on [0, 2]

Define

$$p_{-1}(x) = 0, p_0(x) = 1$$

We have

$< p_0, p_0 >= \int_0^2 1 * 1\, dx = 2$

$< xp_0, p_0 >= \int_0^2 x\, dx = \frac{1}{2}x^2|_0^2 = 2$

$$p_1(x) = [x - \frac{< xp_0, p_0 >}{< p_0, p_0 >} p_0(x) - \frac{< p_0, p_0 >}{< p_{-1}, p_{-1} >} p_{-1}(x)] = (x - \frac{2}{2}) * 1 - 0 = x - 1$$

$< p_1, p_1 >= \int_0^2 (x-1)^2\, dx = \frac{1}{3}x^3 - x^2 + x|_0^2 = \frac{2}{3}$

$< xp_1, p_1 >= \int_0^2 x(x-1)^2\, dx = \int_0^2 \frac{1}{4}x^4 - \frac{2}{3}x^3 + \frac{1}{2}x^2|_0^2 = \frac{2}{3}$

$$p_2(x) = [x - \frac{< xp_1, p_1 >}{< p1, p1 >}]p_1(x) - \frac{< p_1, p_1 >}{< p_0, p_0 >} p_0(x) = (x-1)(x-1) - \frac{\frac{2}{3}}{2} = (x-1)^2 - \frac{1}{3}$$

$< p_2, p_2 >= \int_0^2 [(x-1)^2 - \frac{1}{3}]^2\, dx = \frac{8}{45}$

$$< cos(\frac{\pi}{2}x), p_0(x) >= \int_0^2 cos(\frac{\pi}{2}x) * 1\, dx = 0$$

$$< cos(\frac{\pi}{2}x), p_1(x) >= \int_0^2 cos(\frac{2}{\pi}x) * (x-1)\, dx = -\frac{8}{\pi^2}$$

$$< cos(\frac{\pi}{2}x), p_2(x) >= \int_0^2 cos(\frac{2}{\pi}x) * [(x-1)^2 - \frac{1}{3}]\, dx = 0$$

$$p(x) = \sum_{i=1}^{2} d_i p_i(x) = \sum_{i=0}^{2} \frac{< cos(\frac{2}{\pi}x), p_i(x) >}{< p_i, p_i >} p_i(x) = 0 + \frac{< cos(\frac{2}{\pi}x), p_1(x) >}{< p_1, p_1 >} p_1(x) + 0 = -\frac{8}{\pi^2} * \frac{3}{2}(x-1)$$

$$\boxed{p(x) \approx -1.215854x + 1.215854}$$

L_∞ error:

$L_\infty = max(e(x)) = max(cos(\frac{\pi}{2}x) + 1.215854x - 1.215854)$

To take the maximum, let $\frac{\partial}{\partial x}(e(x)) = \frac{\partial}{\partial x}(cos(\frac{\pi}{2}x) + 1.215854x - 1.215854) = 0$

We have $x \approx 0.56665$

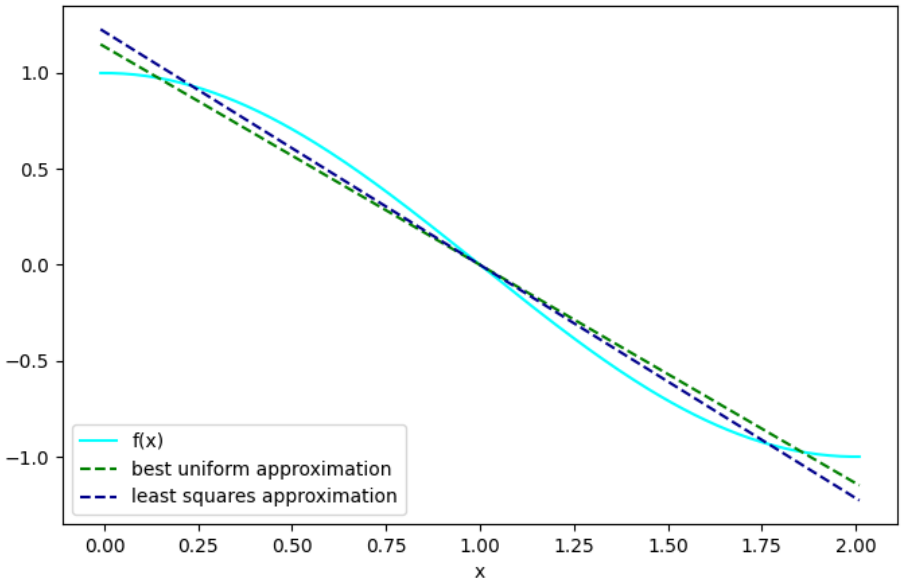
$$\boxed{\mathrm{L}_\infty \approx e(0.56665) \approx 0.102450}$$

L_2 error:

$L_2 = \sqrt{\int_0^2 |e(x)|^2\, dx}$

$= \sqrt{\int_0^2 (cos(\frac{\pi}{2}x) + 1.215854(x-1))^2\, dx}$

$$\boxed{\mathrm{L}_2 \approx 0.120273}$$



To run:

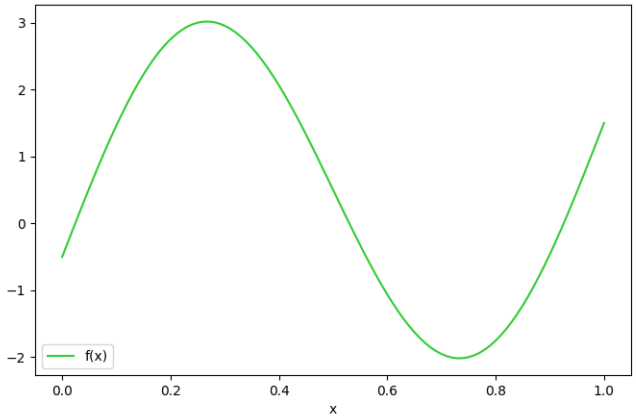
```
python q1.py
```

Program:

```
import numpy as np
import matplotlib.pyplot as plt

low, high = 0, 2

def f(x):
  return np.cos((np.pi/2)*(x))

def p(x):
  return -1.215854*x + 1.215854

x_plot = np.arange(low-0.01, high+0.02, .01)
y_plot = [f(item) for item in x_plot]
plt.figure(figsize = (8, 5))
plt.plot(x_plot, y_plot, "limegreen", label = "f(x)")
y_line = [p(x) for x in x_plot]
plt.plot(x_plot, y_line, "darkgreen", linestyle='--', label="approximating polynomial (least squares
                                                    )")
plt.xlabel("x")
plt.ylabel("cos((pi/2)*x)")
leg = plt.legend(loc='lower left')
plt.show()
```

2

We try to express $f(x)$ as $f(x) = c_0 + c_1x + c_2x^2 + c_3cos(\pi x) + c_4cos(\pi x) + ... + c_k\phi_k(x)$, as a linear combination of several candidate functions. A graph of $f(x)$ is shown below:



We can first observe the data and pick some possible basis. After plotting $f(x)$, observing $f(x)$'s periodicity is about 1, and $f(x)$ has center $x = 0$, we assume that $sin(2\pi x)$. The plot is slightly shifted to the positive y direction, so another possible basis is 1. $f(x)$ has slightly higher values in the positive x direction, so x might be another possible basis. We will approximate using a linear combination of the basis functions $sin(2\pi x), x, 1$.

To run the Python program:

```
python q2.py
```

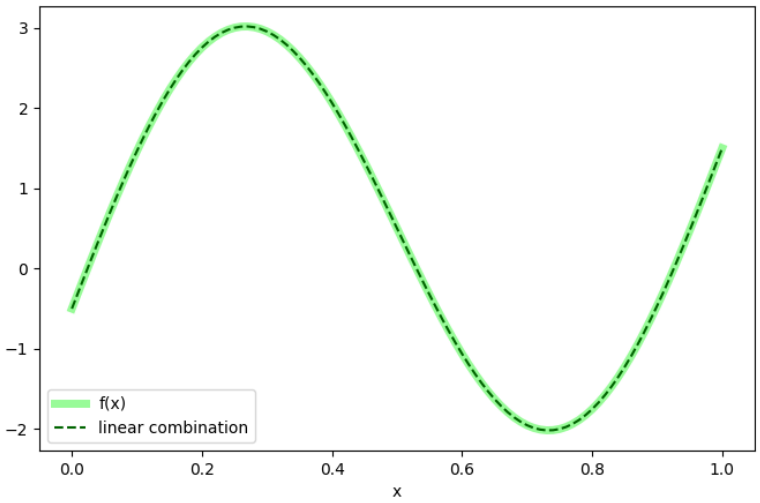
The output is as follows:

```
functions: 1, x, sin(2pi x)
coefficients:
[[-0.5]
 [ 2. ]
 [ 3. ]]
```

Which gives us

$$\boxed{\text{f(x) = -0.5 + 2*x + 3 * sin(2 } \pi x)}$$

Plotting the linear combination tells us that we have found a suitable linear combination of functions.



3

(a)

$$x = cos(\theta) \Longrightarrow \theta = cos^{-1}(x)$$

$$T_0(x) = cos(0 * \theta) = cos(0) = 1$$

$$T_1(x) = cos(1 * cos^{-1}(x)) = x$$

$$T_2(x) = 2xT_1(x) - T_0(x) = \boxed{2x^2 - 1}$$

$$T_3(x) = 2xT_2(x) - T_1(x) = 2x(2x^2 - 1) - x = \boxed{4x^3 - 3x}$$

$$T_4(x) = 2xT_3(x) - T_2(x) = 2x(4x^3 - 3x) - (2x^2 - 1) = \boxed{8x^4 - 8x^2 + 1}$$

(b)

Take $g(x) = T_3(x), h(x) = T_4(x)$

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

$\frac{8x^4 - 8x^2 + 1}{\sqrt{1-x^2}}$ is symmetric around x = 0 on [-1, 1] because both $8x^4 - 8x^2 + 1$ and $\sqrt{1 - x^2}$ are symmetric around x = 0, and denote $h(x) = 4x^3 - 3x$,

$$h(x) = 4x^3 - 3x = -(4x(-x)^3 - 3(-x)) = -h(-x), x \in [-1, 1]$$

This gives us $\int_{-1}^{1} h(x)\, dx = 0$, and $\int_{-1}^{1} \frac{8x^4 - 8x^2 + 1}{\sqrt{1-x^2}} h(x)\, dx = 0$. Therefore,the above integral equals 0.

Because their inner product $\boxed{< T_3(x), T_4(x) >=< g, h >= 0}$, $T_3(x), T_4(x)$ are orthogonal.

(c)

Let $x = cos(\theta)$, because $x \in [-1, 1]$, $\theta \in [cos^{-1}(-1), cos^{-1}(1)]$

$$< T_n(x), T_n(x) >= \int_{-1}^{1} \frac{T_n(x)T_n(x)}{\sqrt{1 - x^2}}\, dx = \int_{cos^{-1}(-1)}^{cos^{-1}(1)} \frac{cos(n\theta)cos(n\theta)}{\sqrt{1 - cos^2(\theta)}}(-sin(\theta))\, d\theta = \int_{cos^{-1}(-1)}^{cos^{-1}(1)} cos^2(n\theta)\, d\theta$$

$$= -(\frac{sin(2n\theta)}{4n} + \frac{\theta}{2} + c)|_{2k\pi+\pi}^{2k\pi} = \frac{sin[2n(2k\pi + \pi)]}{4n} - \frac{sin[2n(2k\pi)]}{4n} - \frac{2k\pi + \pi}{2} - \frac{2k\pi}{2}$$

$$= \frac{1}{4n}[sin(4nk\pi + 2n\pi) - sin(4nk\pi))] + \frac{\pi}{2}$$

Here k is an integer. Because $sin(2n\pi) = 0$ for any integer n.

$$< T_n(x), T_n(x) > \quad = \frac{1}{4n}(0 - 0) + \frac{\pi}{2} = \frac{\pi}{2}$$

We now derived that all $T_n(x)$ have the same length $\boxed{|T_n(x)| = \sqrt{< T_n(x), T_n(x) >} = \sqrt{\frac{\pi}{2}}}$

(d)

Let $x = cos(\theta)$, because $x \in [-1, 1]$, $\theta \in [cos^{-1}(-1), cos^{-1}(1)]$

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

$= \int_{cos^{-1}(-1)}^{cos^{-1}(1)} \frac{1}{\sqrt{1-cos^2(\theta)}} cos(i\theta) cos(j\theta)(-sin(\theta))\, d\theta$

$= -\int_{cos^{-1}(-1)}^{cos^{-1}(1)} cos(i\theta) cos(j\theta)\, d\theta$

$= -\frac{sin[(i-j)x]}{2(i-j)} - \frac{sin[(i+j)x]}{2(i+j)} + c|_{\pi+2k\pi}^{2k\pi}$

$= \frac{sin[(i-j)(\pi+2k\pi)]}{2(i-j)} + \frac{sin[(i+j)(\pi+2k\pi)]}{2(i+j)} - \frac{sin[(i-j)(2k\pi)]}{2(i-j)} - \frac{sin[(i+j)(2k\pi)]}{2(i+j)} \boxed{= 0}$

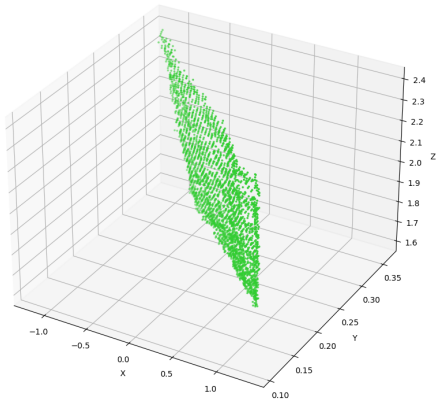
Because $i \geq j \geq 0$ and $i \neq j$, $i - j$ and $i + j$ are both non-zero integers, because $sin(n\pi) = 0$ for any integer n, the above expression equals zero. We have shown that $< T_i, T_j >= 0$ for all i, j such that $i \geq 0, j \geq 0, i \neq j$

4

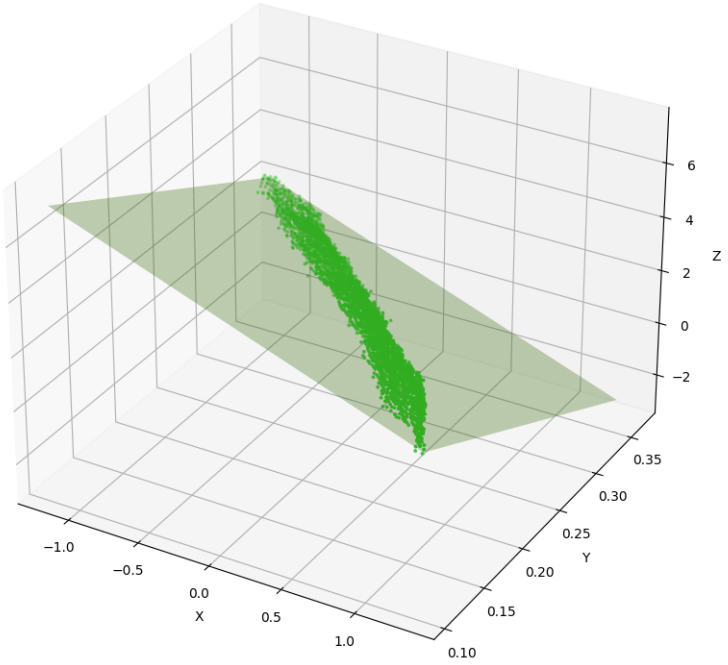
To run the code for all subquestions of question 4: `python q4.py`. All result images are saved in results/ folder with corresponding question number.

(a)

Raw data:

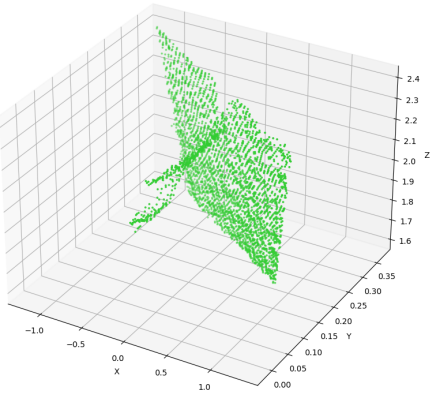


Least squares approximation fitted plane is shown below. Average distance from point to plane: 0.0027447682335793883

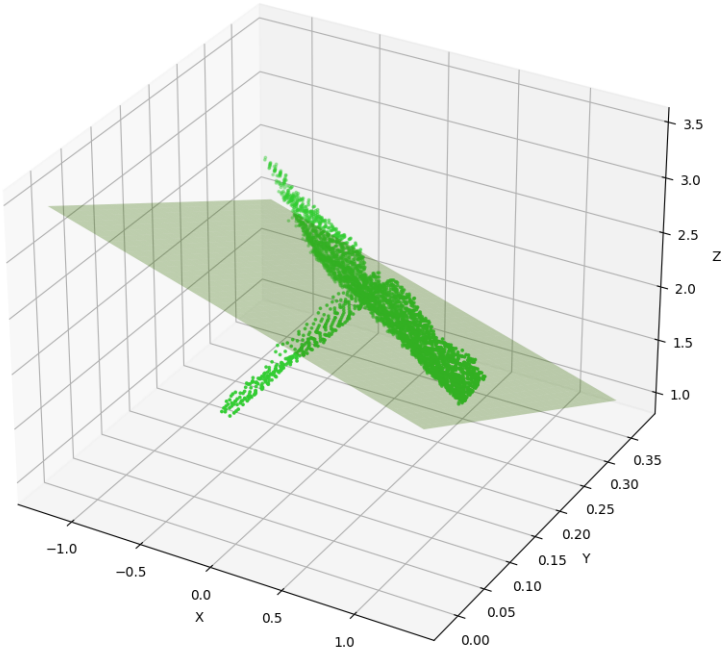


(b)

Raw data:



Least squares approximation fitted plane is shown below. It is not accurate since the distance of the outliers (points that indicate the cat) is taken into account in the calculations of the approximating plane. The least squares calculations also tries to minimize the outliers' distance to the fitted plane as well. Many of the outlier points are far from dominant plane, and thus have a huge impact in changing the coefficients of the plane expression, making the fitted plane shift towards their positions. The plane will then be incorrectly fitted.



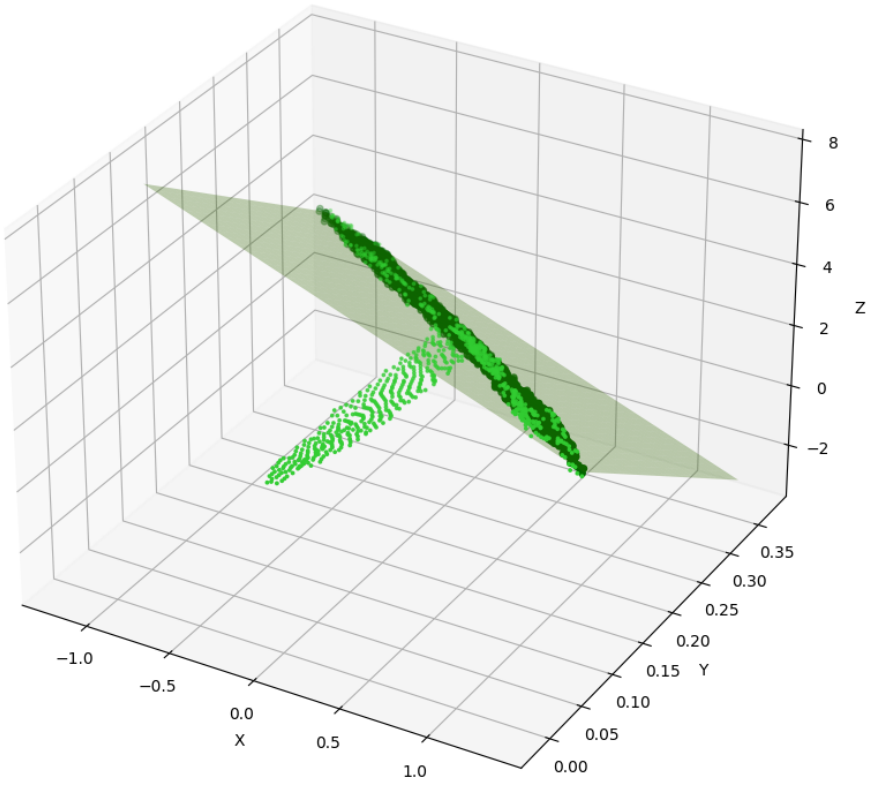
(c)

We can use RANSAC to randomly sample points and fit the plane from the few points we sampled, We then calculate the distance from all the points to the plane and differentiate between inliers and outliers. Under the assumption that there are significantly more points on the plane than those that are not, we can:

(1) Set a threshold for the ratio of inliers over the total number of points, and select the plane of which a ratio higher than the threshold

(2) Or sample points and fit a plane for many (say, 500) ierations, and ultimately take the plane with the highest number of inliers (meaning, that it fits most of the points)

Approach (b) is programmed in q4.py file. The fitting result is shown below. Inliers are colored in darkgreen.



(d)

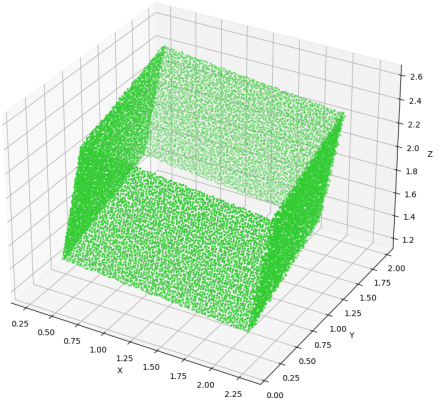
The extended solution to find 4 plane works as follows:

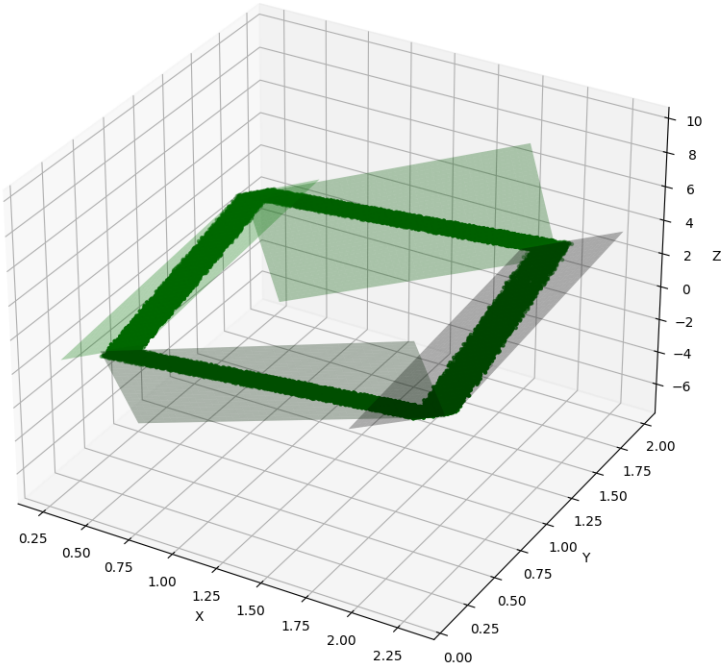
(1) Use RANSAC to identify one plane first. Assuming that there is an equal amount of points in each plane, each plane found should satisfy the condition that the number of inliers is roughly 1/4 of the number of all points. I set the ratio threshold to 0.18, slightly lower than $1/4 = 0.25$. We keep sampling and fitting the plane until this condition is satisfied.

(2) Exclude the inliers found in the previous plane and run RANSAC on the rest of the points. Repeat step (1)

(3) We stop fitting new planes once we have found 4 planes.

This approach is implemented in q4.py. The raw data and result image of a clean hallway are shown below:





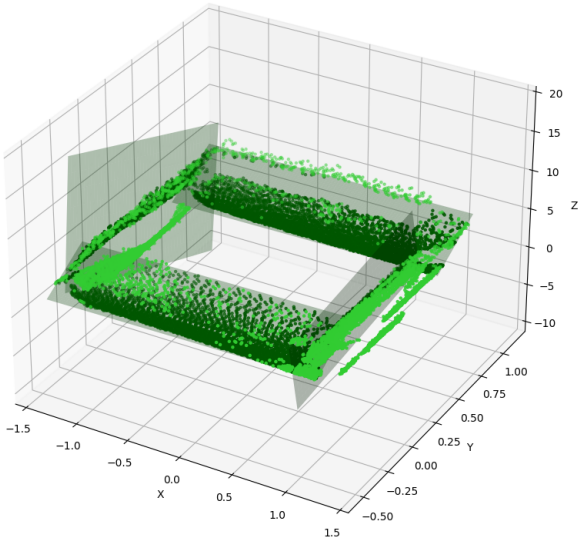
(e)

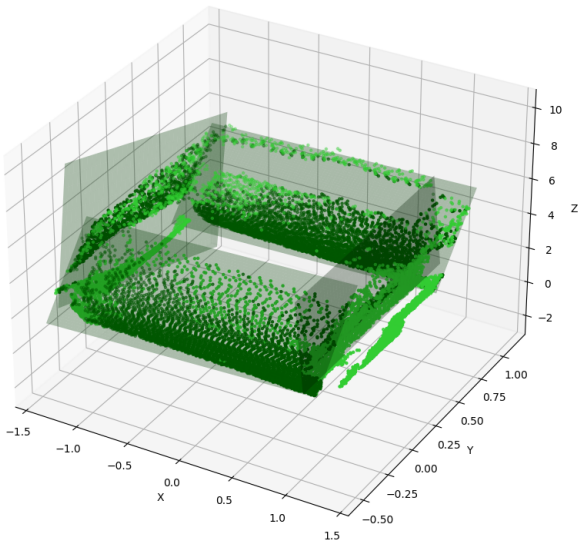
For noisy hallway, the algorithm is slightly modified:

(1) When fitting the initial 2 planes with RANSAC, a new "best plane" should have a lower mean distance of inliers than the best plane already found. Also, it needs to capture more inliers than the previous "best plane". Find the first two planes with such strict conditions.

(2) Same as the previous algorithm, we fit a new plane using the outliers of the previous plane.

(3) Because of the noisy data, find the rest of the planes with more relaxed conditions. Now a new best plane can have a larger mean distance for its inlier points.





4 planes with inliers and outliers, light green indicates outliers

The smoothest score (using mean distance of inliers to the plane) is around 0.0037362285342271576. For each experiment, this number might vary from the 4th digit after the decimal point due to the random selection of points. Other planes have values around 0.004xxx, 0.005xxx, 0.006xxx. The larger the mean distance, the noisier the plane.

The smoothest plane is visualized below.