16811 A3

October 2022

1.

(a) Taylor expansion:

$$\begin{aligned} f(x) &= \sum_{n=0}^{\infty} \frac{f^n(0)}{n!} x^n \\ &= f^0(0) + \frac{f^1(0)}{1} x^1 + \frac{f^2(0)}{2!} x^2 + \frac{f^3(0)}{3!} x^3 + ... \end{aligned}$$

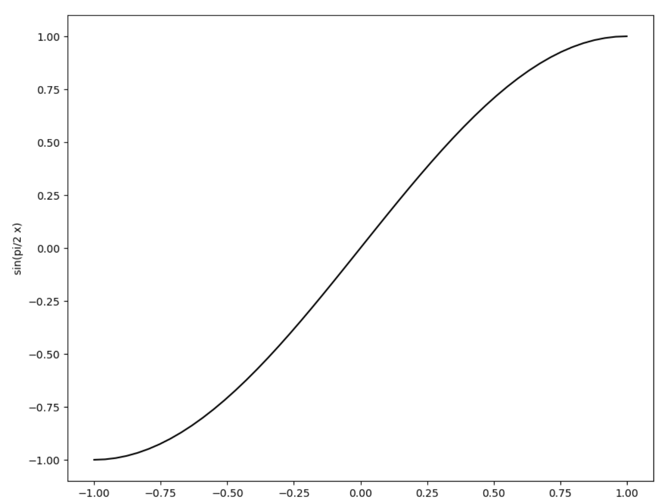
Since

$$\begin{aligned} f(x) &= sin(\frac{\pi}{2} x) \\ f^0(0) &= 0 \\ f^1(0) = \frac{\pi}{2} cos(0) &= \frac{\pi}{2} \\ f^2(0) = -(\frac{\pi}{2})^2 sin(0) &= 0 \\ f^3(0) = -(\frac{\pi}{2})^3 cos(0) &= -(\frac{\pi}{2})^3 \\ &... \end{aligned}$$

we can get that

$$\begin{aligned} f(x) &= f^0(0) + \frac{f^1(0)}{1} x^1 + \frac{f^2(0)}{2!} x^2 + \frac{f^3(0)}{3!} x^3 + ... \\ &= 0 + \frac{\pi}{2} x^1 + 0 - \frac{1}{3!} (\frac{\pi}{2})^3 x^3 + 0 + \frac{1}{5!} (\frac{\pi}{2})^5 x^5 + 0... \\ &= \sum_{n=1}^{\infty} (-1)^{n+1} \frac{1}{(2n-1)!} (\frac{\pi}{2})^{2n-1} x^{2n-1} \end{aligned}$$

(b) Run *python q1.py* to get the plot.



(c)

$$n = 2, p(x) = a + bx + cx^2$$
$$e(x) = sin(\frac{\pi}{2}x) - a - bx - cx^2$$

From (a) we know that the coefficient of x^2 is $0,$ therefore $x^2 \in$ null space of $f(x)$ and $c = 0$

$$\Rightarrow p(x) = a + bx, e(x) = sin(\frac{\pi}{2}x) - a - bx$$

$n + 2 = 3$, let $-1 \leq x_0 < x_1 < x_2 \leq x_3 \leq 1$, and since f^{n+1} does not change the sign on $[-1, 1]$, then $x_0 = -1, x_3 = 1$.
We need to find a, b, x_1, x_2.

$$e(-1) = -e(x_1) = e(x_2) = -e(1)$$
$$e(-1) = -e(1) \Rightarrow sin(-\frac{\pi}{2}) - a + b = -sin(\frac{\pi}{2}) + a + b \Rightarrow a = 0$$
$$e'(x_1) = 0 \Rightarrow \frac{\pi}{2}cos(\frac{\pi}{2}x_1) - b = 0 \ (1)$$
$$e'(x_2) = 0 \Rightarrow \frac{\pi}{2}cos(\frac{\pi}{2}x_2) - b = 0 \ (2)$$
$$e(-1) = -e(x_1) \Rightarrow -1 + b = -sin(\frac{\pi}{2}x_1) + bx_1 \ (3)$$
$$e(-1) = e(x_2) \Rightarrow -1 + b = sin(\frac{\pi}{2}x_2) - bx_2 \ (4)$$

From (1) and (2), we can get $cos(\frac{\pi}{2}x_1) = cos(\frac{\pi}{2}x_2) \Rightarrow x_1 = -x_2$ (5)
From (1) and (3), solve two unkowns using 2 equations, we can get $b = 1.138, x_1 = -0.484$.
From (5), we get $x_2 = 0.484$.

$L_\infty = e(-1) = -1 + 1.138 = 0.138$
$L_2 = \sqrt{\int_{-1}^{1} |e(x)|^2 dx} = \sqrt{\int_{-1}^{1} |e(x)|^2 dx} = \sqrt{\int_{-1}^{1} |sin(\frac{\pi}{2}x) - 1.138x|^2 dx} = 0.136$

(d)

$$p_{i+1} = \left(x - \frac{\langle x p_i, p_i\rangle}{\langle p_i, p_i\rangle}\right) p_i(x) - \frac{\langle p_i, p_i\rangle}{\langle p_{i-1}, p_{i-1}\rangle} p_{i-1}(x)$$

$$\langle p_0, p_0\rangle = \int_{-1}^{1} 1\, dx = 2$$

$$\Rightarrow p_1(x) = x$$

$$\langle p_1, p_1\rangle = \int_{-1}^{1} x^2 dx = \frac{2}{3}$$

$$\Rightarrow p_2(x) = x^2 - \frac{1}{3}$$

$$\langle p_2, p_2\rangle = \int_{-1}^{1} (x^2 - \tfrac{1}{3})^2 dx = \frac{8}{45}$$

$\Rightarrow$ orthogonal approximation $p(x) = \sum_{i=0}^{2} \frac{\langle \sin(\frac{\pi}{2}x), p_i\rangle}{\langle p_i, p_i\rangle} p_i(x)$

$$\langle \sin(\tfrac{\pi}{2}x), p_0\rangle = \int_{-1}^{1} \sin(\tfrac{\pi}{2}x) dx = 0$$

$$\langle \sin(\tfrac{\pi}{2}x), p_1\rangle = \int_{-1}^{+1} \sin(\tfrac{\pi}{2}x) x\, dx = 8 \cdot \frac{1}{\pi^2}$$

$$\langle \sin(\tfrac{\pi}{2}x), p_2\rangle = \int_{-1}^{1} \sin(\tfrac{\pi}{2}x) \cdot (x^2 - \tfrac{1}{3}) dx = 0$$

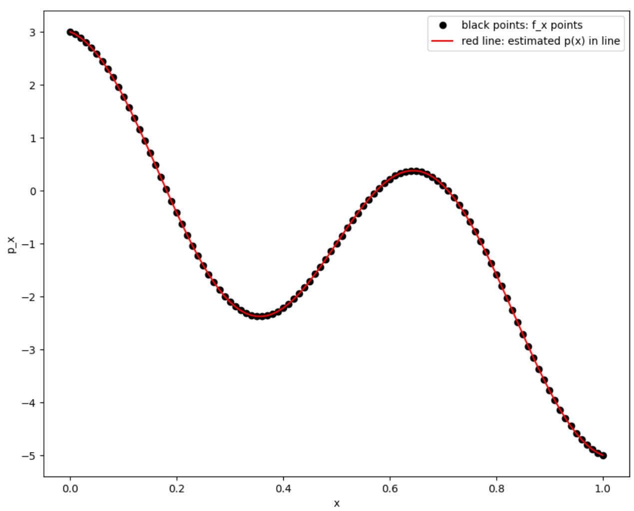
$$\Rightarrow p(x) = 0 \cdot p_0(x) + \frac{8}{\pi^2} \cdot \frac{3}{2} \cdot p_1(x) + 0 \cdot p_2(x) \approx 1.216x$$

Similarly to formula in c, we get L_∞ norm $= 0.2159$

L_2 norm $= 0.1203$

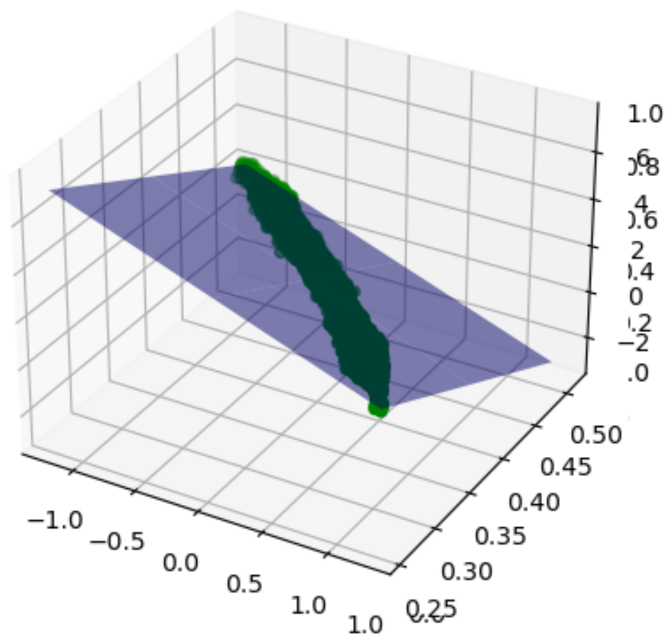
2.

Run *python q2.py* to get the estimated p(x), and the graph below: the black points represents the original f_x points, and the red line represents the estimated p(x) in line.
Explanation:
I first plotted the graph of f_x.
From the observation of start and points $f\_(x_0) = 3, f\_(x_{100}) = -5$, and the magnitude of frequency, I guess that '$cos(2\pi x)$ and x' might be in f_x.
Therefore I chose the basis as $1, cos(2\pi x), x$.
In the code, in order to find the estimated coefficients c, I used the SVD to solve the equation $\mathbf{Ac} = \mathbf{f\_x}$, where $\mathbf{A} = [cos(3\pi\mathbf{x}), \mathbf{x}, \mathbf{1}].T$.
The coefficients I found are [2, -4, 1]



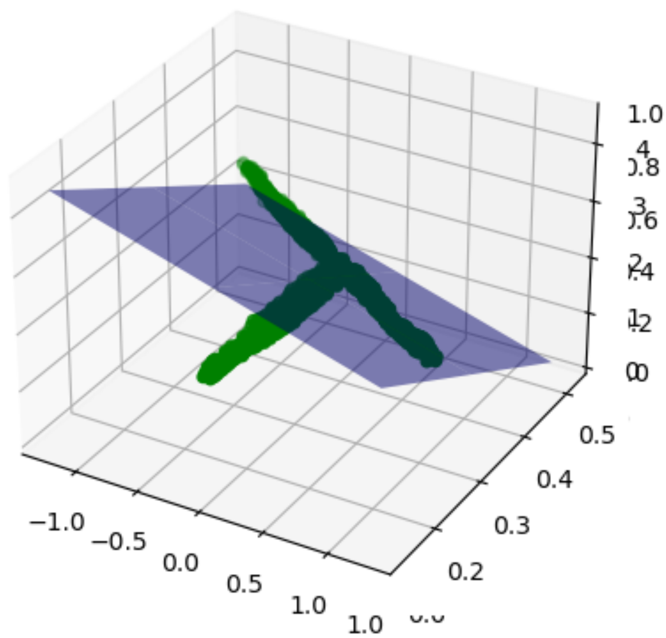
4.

(a) Run *python q4_ab.py*.
First, from the output of function *get_coef*, we can see that the coefficients are $0.0, 0.0, -0.0, -0.0$.
To avoid the degeneracies, we let one of the coefficients $c = 0$, which is implemented in the function *get_coef1*, and we get the results $a : 1.9569956342147987, b : 20.456821816948395, c : 1, d : -10.132444809792128$.
Therefore, the fitted plane is $1.9569956342147987x + 20.456821816948395y + z - 10.132444809792128 = 0$. The average distance of a point in the data set to the fitted plane is 0.002736957327608017.
The fitted plane along with the data is shown below:



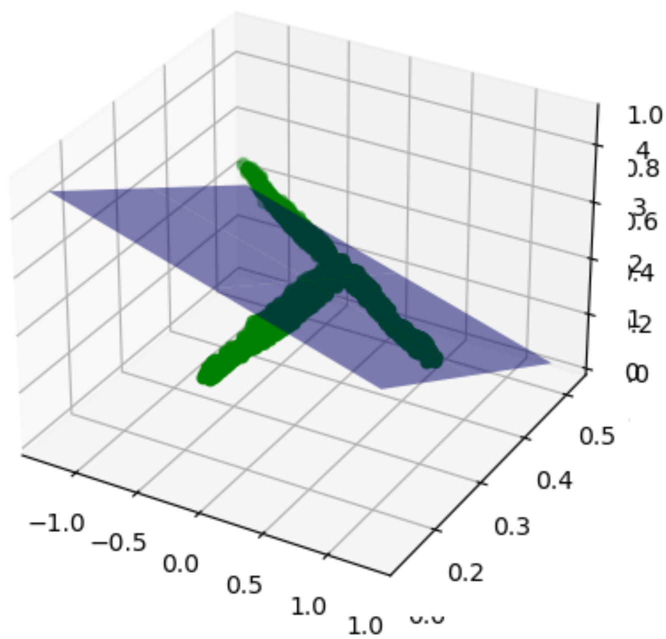
(b) Run *python q4_ab.py*.
The results are: $a : 0.6591950697313653, b : 6.542071310537433, c : 1, d : -4.43731174905677$.
Therefore, the fitted plane is $0.6591950697313653x + 6.542071310537433y + z - 4.43731174905677 = 0$.
The average distance of a point in the data set to the fitted plane is 0.02824161326033477.

The fitted plane along with the data is shown below:
As we can se from the graph, the plane does not fit well to the data in this example, mainly because there are some outliers which caused the the fitted plane shift from the desired position.

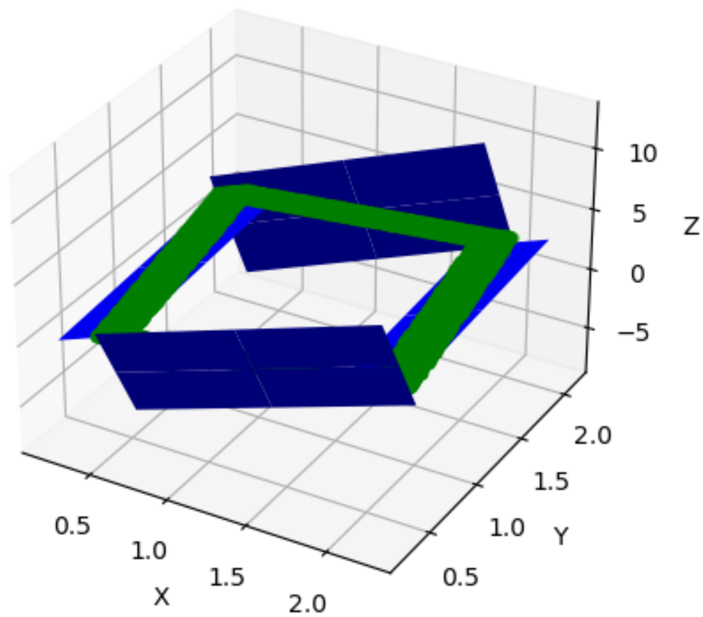


(c) Since there are outilers in the dataset, I chose RANSAC to find the plane that best fit the points:
In each iteration of RANSAC: sample randomly the number of points requried to fit the plane; then solve for the plane coefficients using samples; score by the fraction of inliers within a preset threshold and return the coefficients with the maximum inliers after the last iteration.
The results are: $a : 0.6591950697313653, b : 16.74630218575813, c : 1, d : -8.19351730275052$.
Therefore, the fitted plane is $0.6591950697313653x + 16.74630218575813y + z - 8.19351730275052 = 0$.
The average distance of a point in the data set to the fitted plane is 0.004020165477250292.

The fitted plane along with the data is shown below:



(d) By using the same algorithm RANSAC as part c, and after we found the first plane, we use the remaining points other than the first set of bestfit points to fit the second plane, same repeadly for the third and forth plane.
We found the four planes (after normalization) and the average distance as follows: $-0.17084976427098508X + 0.984705127017284y + 0.03414924412667991z + 1.800185004436954 = 0$ average distance: 0.010264269685697795; $-0.16754808173722976x + 0.9850746032088846y + 0.03944193731338481 4z + 0.21779307412880713 = 0$, average distance: 0.011036758591514815; $-0.963324218997436x + -0.17313903733524658y + 0.20501054324986318z + -1.8155706368583644 = 0$, average distance: 0.00973912681464682; $-0.9633946015831143x + -0.173828429971411y + 0.2040943864347734z + -0.21168427643036922 = 0$, average distance: 0.005994844854619448
The fitted planes along with the data is shown below:



(e) We can define the smoothness of a plane as the average distance of the points on the plane to this plane. If this average distance is small than a defined threshold, then the plane should be smooth.
We can use the same RANSAC algorithm as before, but for each plane we tested with different number of iterations and threhold depending on its smoothness.
We found the four planes (after normalization) and the average distance as follows:
$-0.011480633237222856x + -0.9950065892501466y + 0.09914677205669399z + -0.5878693318418458 = 0$ average distance: 0.004000038159456176; $-0.016731113753420682x + -0.9957700256935464y + 0.09034448385399209z + 0.5278183038415216 = 0$, average distance: 0.00443713890278853; $-0.9957493425800125x + -0.002579630738421659y + -0.092068410742915z + 0.9442957693976075 = 0$, average distance: 0.052391909572069635; $-0.9985728802463664x + -0.0290621631496907y + 0.0448061771359361z + -0.8339193974087117 = 0$, average distance: 0.04878216917947904
The fitted planes along with the data is shown below: