16-811 hw 3

Erica Weng

November 6, 2020

1

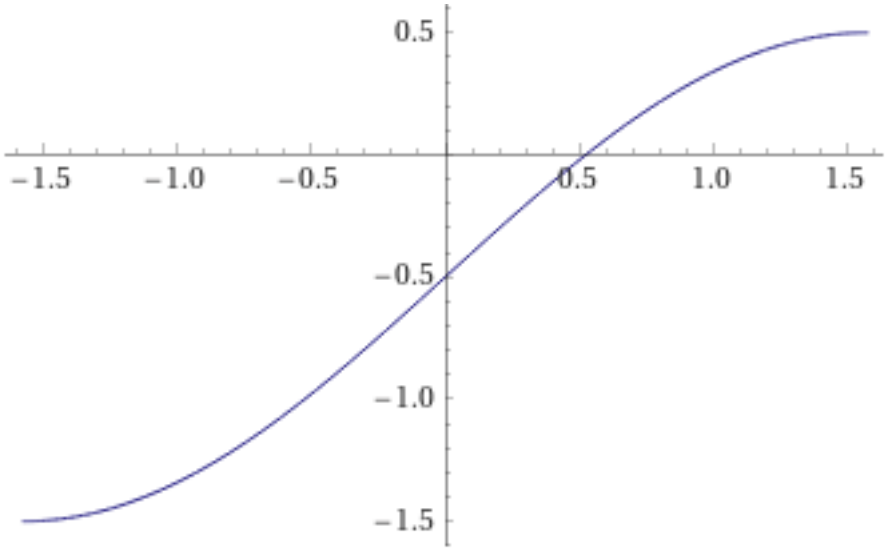
1a

$$f(x) \approx f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \dots$$
$$= -0.5 + x - \frac{1}{6}x^3 + \dots$$

or in full:

$$= -.5 + \sum_{j=0}^{\infty} \frac{(-1)^j x^{1+2j}}{(1+2k)!}$$

1b



1c

We want to find the best polynomial of degree $n = 2$ that approximates the fn $f(x) = \sin x - 0.5$ on the interval $[a, b] = [-\pi/2, \pi/2]$.
we have that the error $e(x_i) = |f(x_i) - p(x_i)|$ should achieve the max value at $n+2 = 2+2 = 4$ points x_i; $i = 0, 1, 2, 3$ such that $-\pi/2 \leq x_0 < x_1 < x_2 < x_3 \leq \pi/2$. Moreover, $e(x_i)$ alternates between positive and negative for adjacent i.
we have that if $f^{(2+1)} = f^{(3)}$ doesn't change sign on the interval $[-\pi/2, \pi/2]$, then $x_0 = a$ and $x_3 = b$.

$$f(x) = \sin x - 0.5$$
$$f'(x) = \cos x$$
$$f''(x) = -\sin x$$
$$f^{(3)}(x) = -\cos x$$

thus $f^{(3)}$ doesn't change sign on the interval $[-\pi/2, \pi/2]$ – it is negative in that interval. So we have the location of 2 of our 4 max error points, $x_0 = -\pi/2$ and $x_3 = \pi/2$.

So we have several equations:

$$f(x) = \sin x - 0.5$$
$$p(x) = ax^2 + bx + c$$
$$e(x) = \sin x - 0.5 - ax^2 - bx - c$$

$$x_0 = -\pi/2$$
$$x_3 = \pi/2$$
$$-\pi/2 \leq x_0 < x_1 < x_2 < x_3 \leq \pi/2$$
$$e(x_0) = -e(x_1) = e(x_2) = -e(x_3)$$

$$e'(x) = \cos x - 2ax - b$$

let's solve some:

$$e(x_0) = -e(x_3)$$
$$e(-\pi/2) = -e(\pi/2)$$
$$\sin(-\pi/2) - 0.5 - a(-\pi/2)^2 - b(-\pi/2) - c = -\sin(\pi/2) + 0.5 + a(\pi/2)^2 + b(\pi/2) + c$$
$$-0.5 - a(-\pi/2)^2 - c = 0.5 + a(\pi/2)^2 + c$$
$$\frac{\pi^2}{4}a + 2c + 1 = 0$$

$$c = \frac{-1 - \frac{\pi^2}{4}a}{2}$$

okay, at this point I'm just going to make a logical deduction and say that $a = 0$. the logic is that since f is a sine, it is radially symmetric about its intersection with the y-axis, that is $(0, -0.5)$. Thus, the BUA is going to be linear, not quadratic, because a quadratic function cannot be radially symmetric. I'm not sure how to prove this formally, but hand-wavily: given that two of the max-error xs are symmetric about the y-axis (that is, $x_0 = -x_3$; $x_0 = -\pi/2$ and $x_3 = \pi/2$), and given that $f(x)$ is radially symmetric, using the equioscillation thereom the other two max-error terms must also be symmetric about the y-axis – that is, $x_1 = -x_2$. Please let me know the flaws in my logic, i would like to know...

$$a = 0$$
$$c = \frac{-1 - \frac{\pi^2}{4}a}{2} = -\frac{1}{2}$$

Now we just have to find b. we'll use the first order stationary conditions.

$$e'(x_1) = e'(x_2) = 0$$
$$= \cos x_1 - b = 0$$
$$x_1 = \cos^{-1}(b)$$

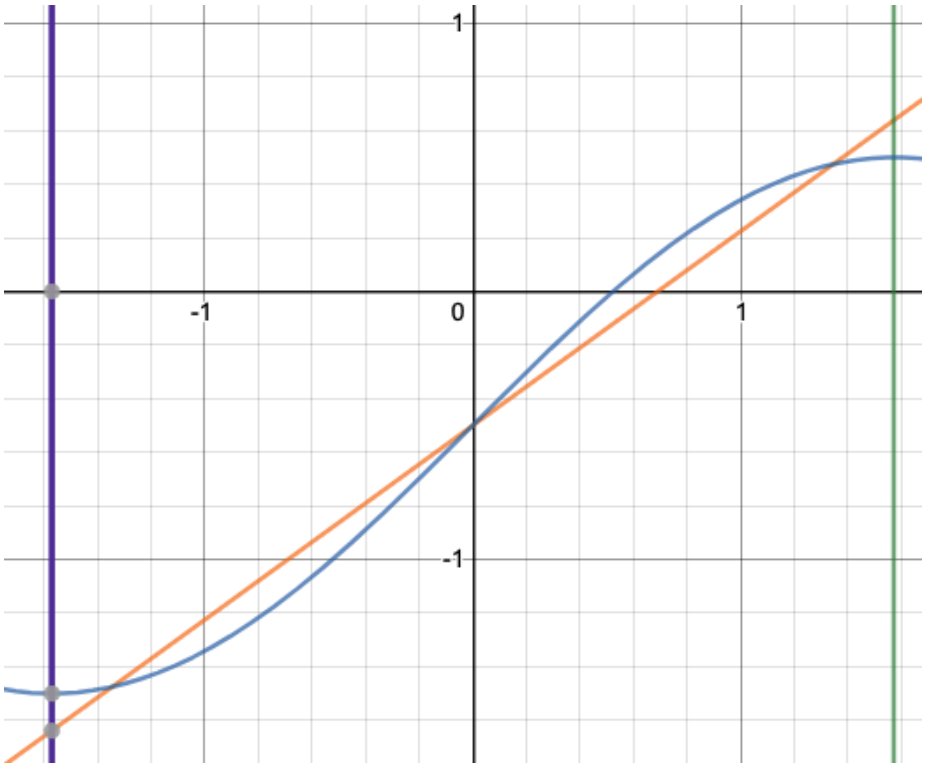
$$e(x_0) = e(x_2)$$
$$-1 + b(\pi/2) = \sin(x_1) - b(x_1)$$
$$-1 + b(\pi/2) = \sin(\cos^{-1}(b)) - b(\cos^{-1}(b))$$

$$b \approx 0.72461$$

alright! there we go. our approximating polynomial of max degree 2 is $\boxed{p(x) \approx 0.7246x - 0.5}$.

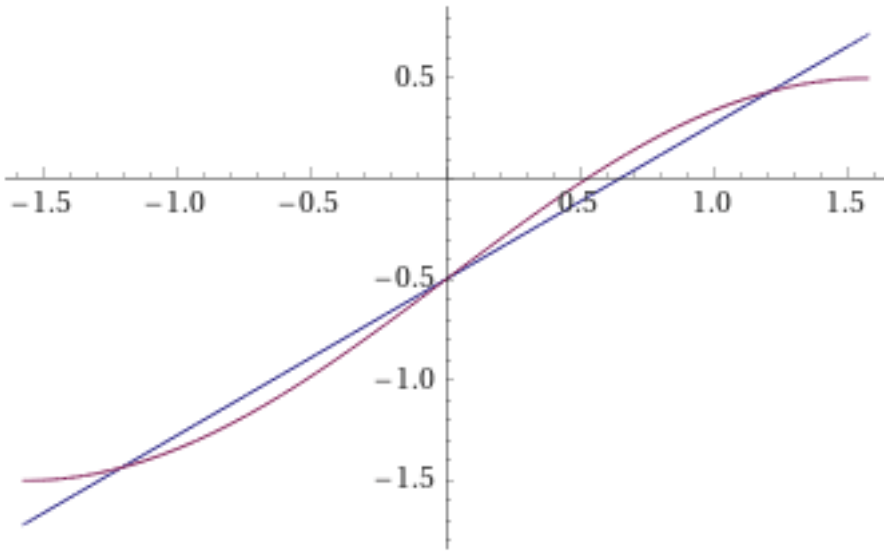
the L_∞ error is $e(x_0) = e(-\pi/2) = \sin(-\pi/2) - 0.7246(-\pi/2) \approx \boxed{0.1382}$
the L_2 error is

$$\sqrt{\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \sin(x) - 0.7246x \, dx} \approx \boxed{0.170404}$$



1d

I used gradient descent to optimize parameters a, b, and c according to least-squares loss, code in `code/q1.py` $-0.0000x^2 + 0.7740x + -0.5000$
L_2-error: 0.1508
L_∞-error: 0.2159



2

$$f(x) = -0.7 + x + 0.6 \cdot \sin(5\pi x)$$

found through gradient descent for optimization, code in `code/q2.py`

3

3a

$$T_0 = 0$$

$$T_1 = 1$$

$$T_2 = 2xT_1(x) - T_0(x) = 2x$$

$$T_3 = 2xT_2(x) - T_1(x) = 4x^2 - 1$$

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

$$T_5 = 2xT_4(x) - T_3(x) = \boxed{16x^4 - 12x^2 + 1}$$

3b

show that

$$\int_{-1}^{1} (1-x^2)^{-\frac{1}{2}}(16x^4 - 12x + 1)(8x^3 - 4x)dx$$

$= 0$ w/o evaluating the integral

change variables $x = \sin\theta$, $dx = \cos\theta d\theta$

$$\int_{\sin^{-1}(-1)}^{\sin^{-1}(1)} (1-\sin^2\theta)^{-\frac{1}{2}}(16\sin^4\theta - 12\sin\theta + 1)(8\sin^3\theta - 4\sin\theta)\cos\theta d\theta$$

cancel terms

$$\int_{\sin^{-1}(-1)}^{\sin^{-1}(1)} (16\sin^4\theta - 12\sin\theta + 1)(8\sin^3\theta - 4\sin\theta)d\theta$$

note that since $T_5(\theta) = 16\sin^4\theta - 12\sin\theta + 1$ is all even-powered sines, and $T_4(\theta) = 8\sin^3\theta - 4\sin\theta$ is all odd-powered sines, the product will be all odd-powered sines, which are radially symmetric about the origin. note also that we are evaluating from $\sin^{-1}(-1)$ to $\sin^{-1}(1)$, an interval that is symmetric about the y-axis. The integral evaluated on $[\sin^{-1}(-1), 0]$ will be negative, while that on $[0, \sin^{-1}(-1)]$ will be positive, and they will be equal to each other in magnitude, and thus will cancel out and be 0.

3c

the length of T_n is given by:

$$\int_{-1}^{1} (1-x^2)^{-\frac{1}{2}} T_n(x)^2 dx$$

change variables $x = \cos\theta$, $dx = -\sin\theta d\theta$

$$-\int_{\cos^{-1}(-1)}^{\cos^{-1}(1)} (1-\cos^2\theta)^{-\frac{1}{2}}\cos^2(n\theta)\sin\theta d\theta$$

cancel terms

$$\int_{\cos^{-1}(-1)}^{\cos^{-1}(1)} \cos^2(n\theta)d\theta$$

the integral is evaluated as:

$$\left(\frac{\theta}{2} + \frac{\sin(2n\theta)}{4n}\right)\Bigg|_{\cos^{-1}(-1)}^{\cos^{-1}(1)}$$

$$= \frac{\cos^{-1}(1) - \cos^{-1}(-1)}{2} + \frac{\sin(2n\theta)}{4n}\Bigg|_{\cos^{-1}(-1)}^{\cos^{-1}(1)}$$

$$\cos^{-1}(1) = 2\pi k \ ; \ k \in \mathbb{I}\ldots$$
$$\cos^{-1}(-1) = 2\pi k + \pi \ ; \ k \in \mathbb{I}\ldots$$

.

the value inside the sin will always be a multiple of 2π, and thus the sin will always evaluate to 0. So, we have shown that regardless of what n is, the length of the chebyshev polynomials are equal to a constant, that is they are all equal in length.

3d

we will show that following integral is 0, for arbitrary n (note we already did the change of variables $x = \cos\theta$, $dx = -\sin\theta d\theta$):

$$\int_{\cos^{-1}(-1)}^{\cos^{-1}(1)} \cos(n\theta)\cos((n+1)\theta)d\theta$$

$$\frac{1}{2}\int_{\cos^{-1}(-1)}^{\cos^{-1}(1)} \cos(\theta) + \cos(2n\theta+\theta)d\theta$$

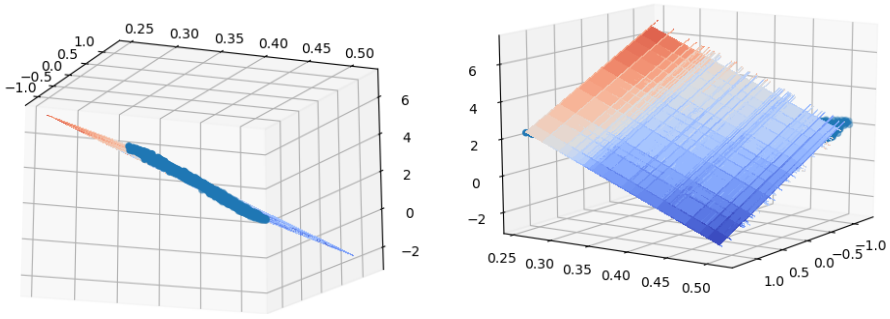
$$= \sin\theta + \frac{\sin(2n\theta+\theta)}{2n+1}\Bigg|_{\cos^{-1}(-1)}^{\cos^{-1}(1)} = 0$$

4

4a

plane

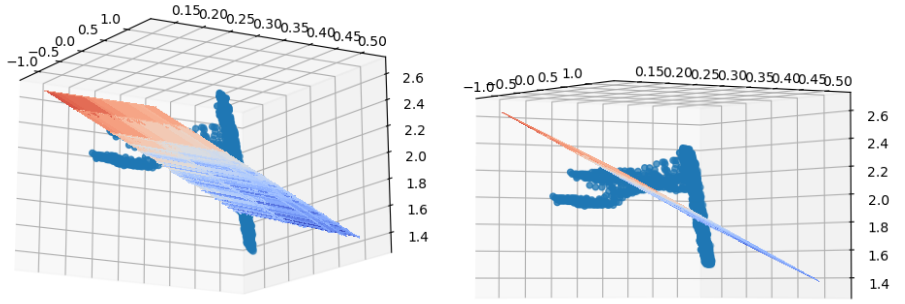
$$-1.793x + -18.745y + -z = -9.439$$



mean distance (length of normal vector from plane to each point; not the residual) is 0.00285

4b

$$-0.200x + -2.022y - z = -2.657$$



mean distance is 0.07715
since the point cloud is not planar, the plane really doesn't fit well to the data. it's trying to average out the "kinks" but thus it doesn't fit well anywhere

4c

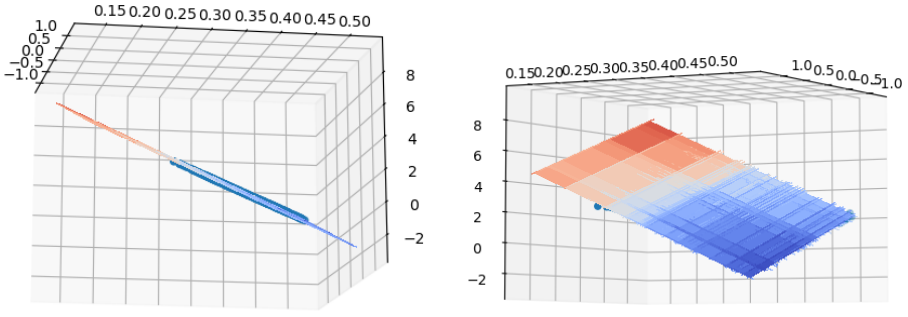
algorithm:

1. fit least-sqs plane to data
2. remove all points distance $>$ threshold (threshold is a hyperparam) and fit another least-squares plane to remaining data
3. repeat for N iterations. each step you put back in the points you removed in the previous round. decrease the distance threshold each iteration by a factor (another hyperparameter)

for 50 iterations, decreasing the distance threshold by .9 times each iteration, and with a starting distance threshold of .12, I ended up with the plane

$$-1.913x + -18.797y - z = -9.462$$

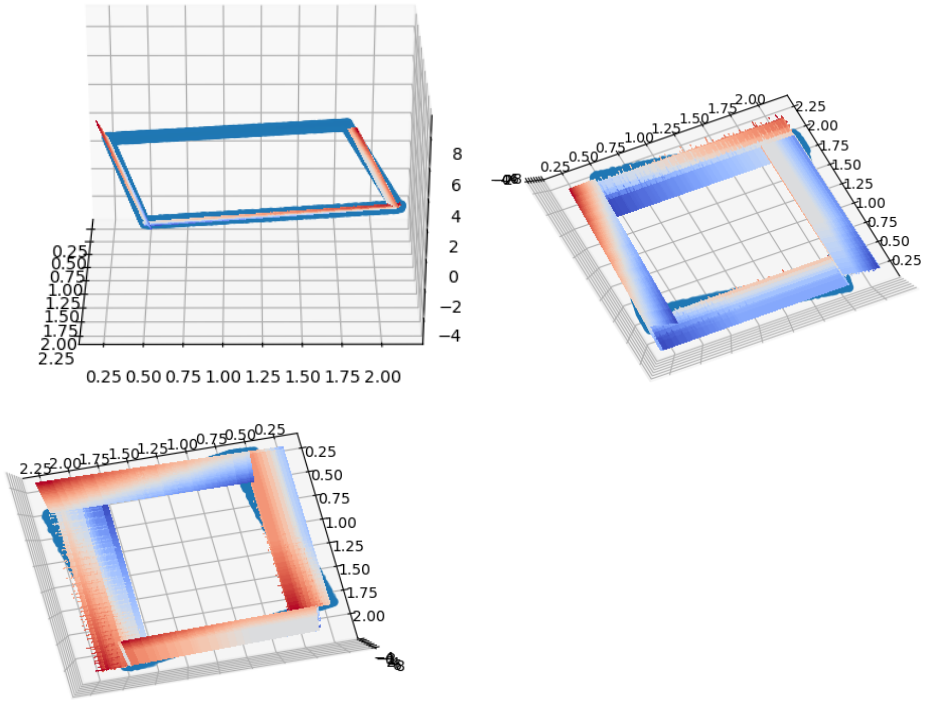
which is decently close to the original plane gotten in the `clear_table.txt` data



code in `code/q4.py`, function `q4c`

4d

I use the random sample consensus algorithm, which iteratively picks a small number of random points, computes the plane going through them, and if it accounts for at least a threshold fraction of the points (by "accounts for," i mean that at least a certain fraction of all points (about total_points / 5, to account for 4 walls plus some differences between the number of points in each wall) fall within a distance threshold of the plane), then I add it to the "candidates" pile that are in contention for the best fitting plane. After finding the best plane for a single wall, I remove the points "accounted for" from the set of total points and repeat for the other three walls.



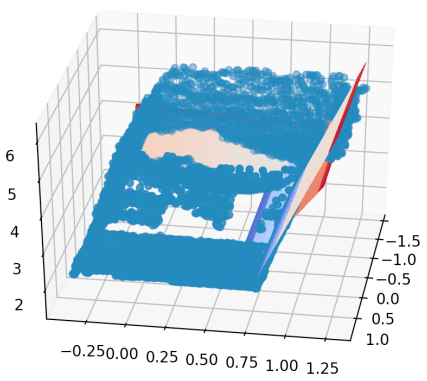
code in `code/q4.py`, function `q4d` and `viz_planes`

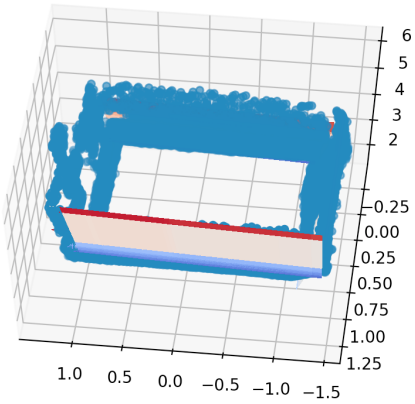
4e

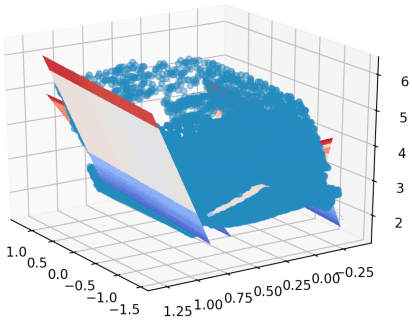
algorithm: same as in q4d, except at the end I take the plane with the smallest mean-squared error of all "inliers" of the plane (that is, for the points that are within a certain distance threshold to the plane, pick the plane with the smallest mean-squared distance of those points to it)

code in `code/q4.py`, function `q4d` and `viz_planes` the smoothest plane I got was $0.034x + 5.039y - z = 1.073$ which had a mean-squared error of 0.149, smallest of all the planes I found.

all planes:







the smoothest plane: