16-811 hw 3

Erica Weng

October 28, 2020

1

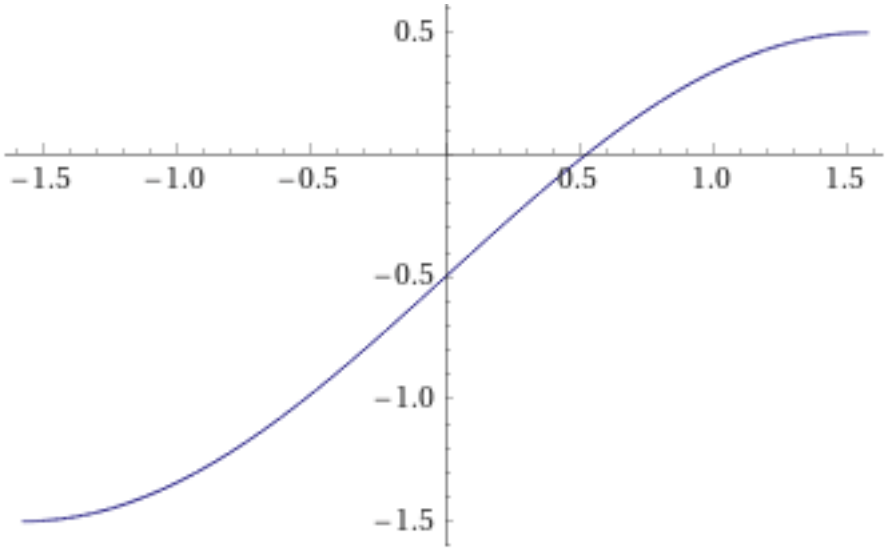
1a

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

or in full:

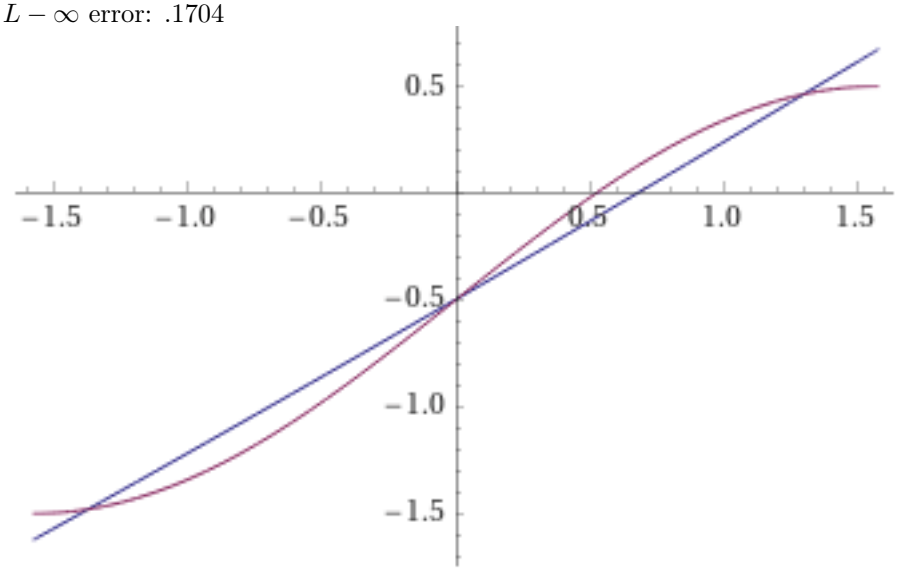
$$= -.5 + \sum_{j=0}^{\infty} \frac{(-1)^j x^{1+2j}}{(1+2k)!}$$

1b



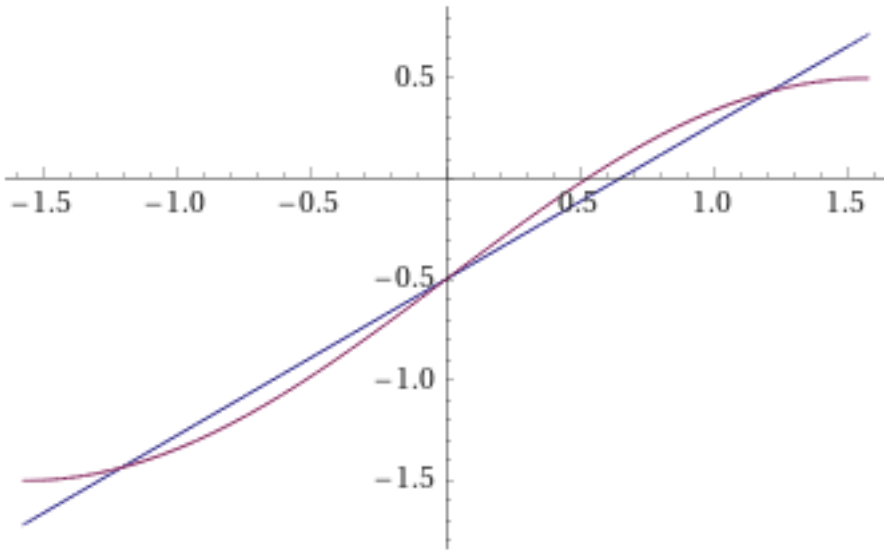
1c

I used gradient descent to optimize parameters a, b, and c according to BUA loss, code in `code/q1.py` bua: $p(x) = 0.0000x^2 + 0.7685x - 0.5198$
$L2$ error: .1385
$L - \infty$ error: .1704



1d

I used gradient descent to optimize parameters a, b, and c according to least-squares loss, code in `code/q1.py` $-0.0000x^2 + 0.7740x + -0.5000$
$L2$-error: 0.1508
$L - \infty$-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. Then their integral will all be cosines. since we are evaluating from -1 to 1 which is symmetric about the origin, then thus the integral will be 0.

3c

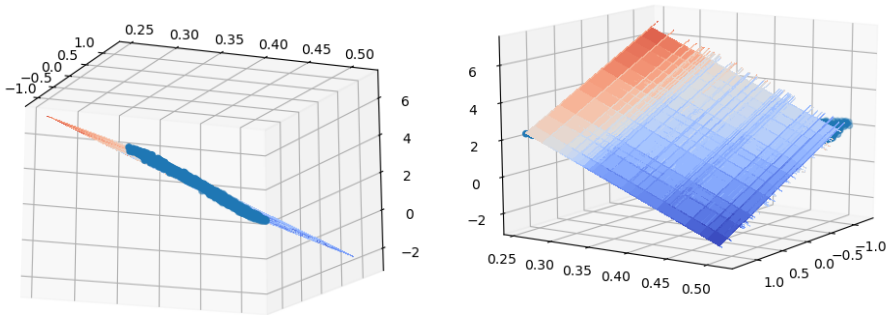
3d

4

4a

plane

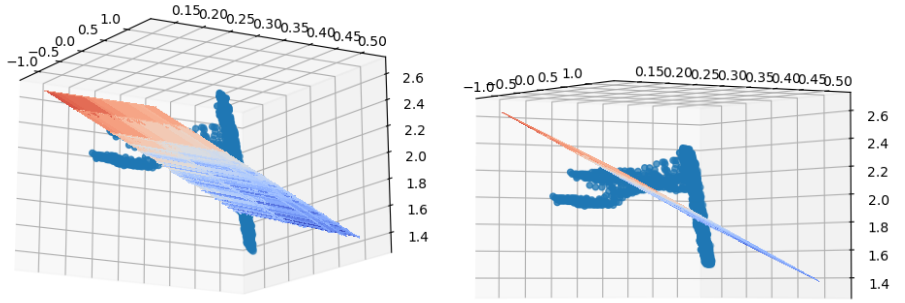
$$-1.793x + -18.745y + -z = -9.439$$



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

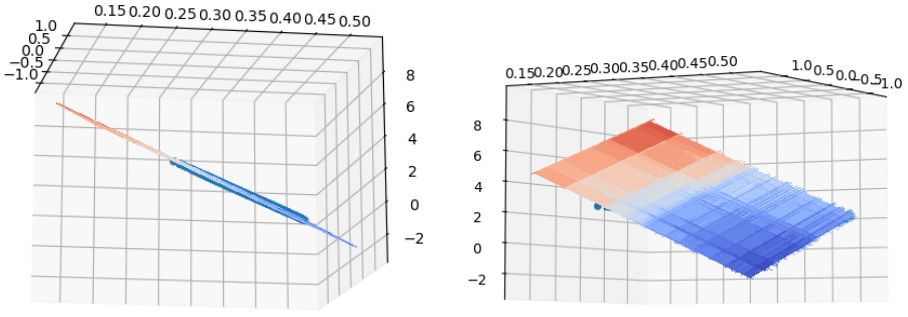
algorithm:

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

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

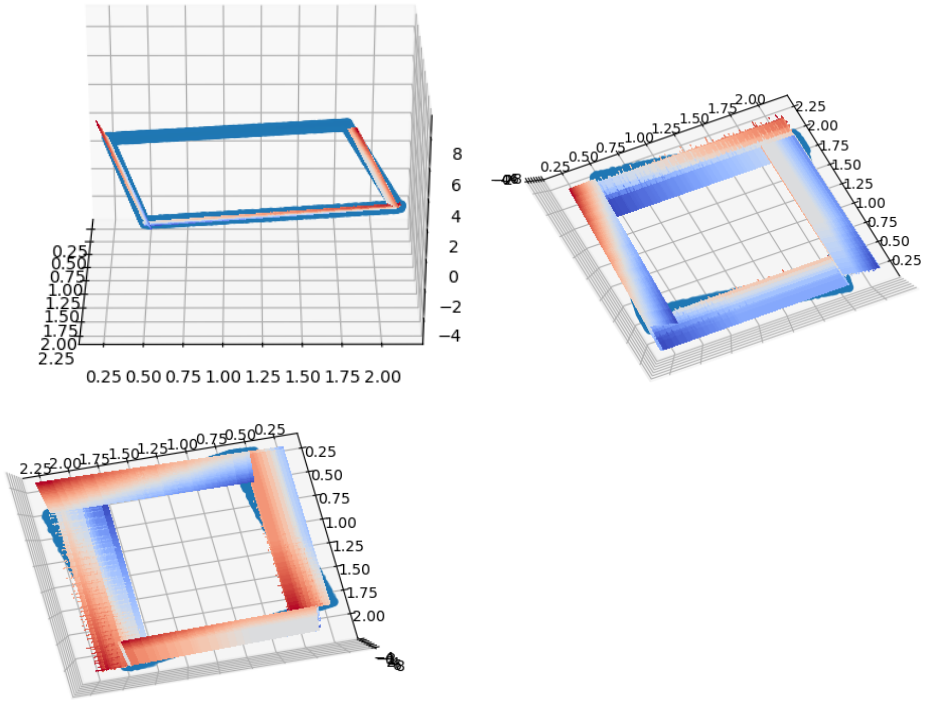
which is decently close to the original plane gotten in the `clear_table.txt` data



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.



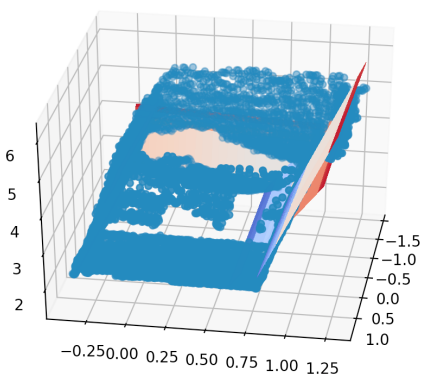
code in `code/q4.py`, function `q4d` and `viz_planes`

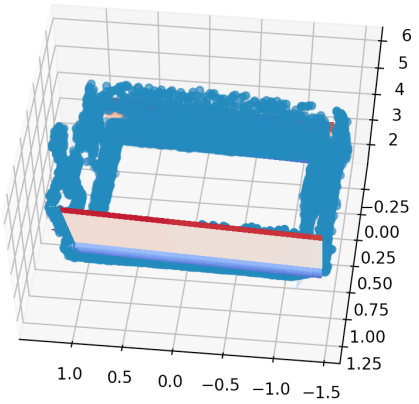
4e

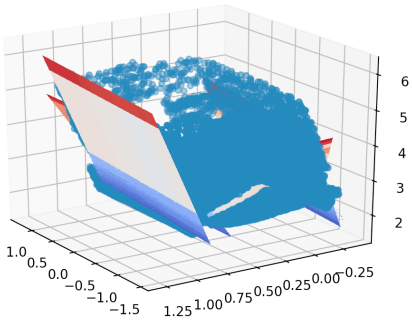
algorithm: same as in q4d, except at the end I take the plane with the smallest mean-squared error of all "inliers" of the plane (that is, for the points that are within a certain distance threshold to the plane, pick the plane with the smallest mean-squared distance of those points to it)

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: