COMPSCI 371 Homework 0

Group Members: Mayur Sekhar, Ari Dixit

Problem 0 (3 points)

Part 1: Linear Algebra

Problem 1.1

If A has a nontrivial null space, then there exists a nonzero vector $\vec{v}$ such that $A\vec{v} = 0$. This implies that $\vec{v}$ is an eigenvector with eigenvalue $\lambda = 0$. Since $\vec{v}$ is a nonzero vector in $\mathbb{R}^d$, it is a real vector, which means A has a real eigenvector associated with eigenvalue 0, and thus, a square real matrix A with a nontrivial null space must have real eigenvectors.

Problem 1.2

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

The characterisitc polynomial of A is found by A - λI. The characteristic polynomial is λ^2 = 1 - 0. This means that since the the possible eigenvalue is less than 0 the only possible eigenvalues and eigne vectors would be imaginary as they are the square root of negative numbers. There is a maximum of 2 eigenvalues (since we have a 2x2 matrix), but both would still be imaginary as they are the squareroot of a negative number and thus imaginary numbers (not real).

$A\vec{v} = \lambda\vec{v}$, where λ is a real eigenvalue and $\vec{v}$ is the corresponding eigenvector.

$$A\vec{v} = \lambda\vec{v} \implies \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \lambda \begin{bmatrix} a \\ b \end{bmatrix}$$

This gives the system:

$-b = \lambda a$

$a = \lambda b$

Solve for b and substitute:

$b = -\lambda a$

$a = \lambda(-\lambda a)$

$a = -a\lambda^2$

Now solve for λ:

$\lambda^2 = -1 \quad \Rightarrow \quad \lambda = \pm\sqrt{-1}$

$\lambda = i$

Since λ is imaginary(not real), the eigenvalues of A are not real, and thus, the eigenvectors will also not be real.

Problem 1.3

$A\vec{x} = \vec{b}$, $A = \begin{bmatrix} 1 & 2 & 3 \\ -2 & 1 & -1 \\ 3 & 0 & 3 \end{bmatrix}$, & $\vec{b} = \begin{bmatrix} 5 \\ 0 \\ 3 \end{bmatrix}$

Write the augmented matrix: $\left[\begin{array}{ccc|c} 1 & 2 & 3 & 5 \\ -2 & 1 & -1 & 0 \\ 3 & 0 & 3 & 3 \end{array}\right]$

Step 1: Add 2 times Row 1 with Row 2 and Subtract 3 Row 1 from Row 3:
$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 5 \\ 0 & 5 & 5 & 10 \\ 0 & -6 & -6 & -12 \end{array}\right]$

Step 2: Divide Row 2 by 5 and Divide Row 6 by -6: $\left[\begin{array}{ccc|c} 1 & 2 & 3 & 5 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \end{array}\right]$

Step 3: Subtract Row 2 from Row 3: $\left[\begin{array}{ccc|c} 1 & 2 & 3 & 5 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right]$

$x_2 + x_3 = 2$

$x_2 = 2 - x_3$

$x_1 + 2x_2 + 3x_3 = 5$

$x_1 + 4 - 2x_3 + 3x_3 = 5$

$x_1 = 1 - x_3$

The set of all solutions to the given linear system cna be given by

$$\vec{x} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} + t \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$

Problem 1.4

The value of q that we think was used in the truncated Fourier series was q = 3. This is because when q = 3 the line of best fit which is graphed is most acurate to the data and most clearly close to mapping the actual data. While q = 4 was also quite good at mapping the data, q = 3 is just a little better at mapping the actual shape and curvature of the data. Since q = 3 is left than q = 4 this using q = 3 would be an example of trying to avoid the overfitting that might ocurr from the higher q values.

In [5]:
```
import urllib.request
import ssl
from os import path as osp
import shutil
import numpy as np
import matplotlib.pyplot as plt

def retrieve(file_name, semester='fall24', homework=0):
    if osp.exists(file_name):
        print('Using previously downloaded file {}'.format(file_name))
    else:
        context = ssl._create_unverified_context()
        fmt = 'https://www2.cs.duke.edu/courses/{}/compsci371/homework/{}/{}'
        url = fmt.format(semester, homework, file_name)
        with urllib.request.urlopen(url, context=context) as response:
            with open(file_name, 'wb') as file:
                shutil.copyfileobj(response, file)
        print('Downloaded file {}'.format(file_name))
```

In [6]:
```
import pickle

# retrieve is defined in a hidden cell
file_name = 'points.pkl'
retrieve(file_name)
with open(file_name, 'rb') as file:
    t = pickle.load(file)
```

```
Using previously downloaded file points.pkl
```

In [7]:
```
t
```

Out[7]: namespace(x=array([4.52751939e-02, 4.87577107e-02, 9.99176115e-01, 6.52369112e-01,
2.34510202e-01, 4.34947552e-01, 9.74186193e-01, 8.97677608e-01,
8.44231038e-01, 3.92404664e-01, 4.93023019e-01, 6.76689352e-01,
6.08027130e-02, 5.55596117e-01, 2.71451605e-01, 8.79651173e-01,
6.42144373e-02, 6.79181533e-01, 8.70088502e-01, 2.27318525e-01,
8.95448239e-01, 8.72195468e-01, 1.85172177e-02, 7.07495567e-01,
1.19968359e-03, 5.03363966e-01, 4.36667052e-01, 2.03252836e-01,
3.24942645e-01, 8.06215331e-01, 3.16452087e-01, 1.49038584e-01,
6.98511990e-01, 4.48544101e-01, 7.98939491e-01, 2.35516457e-01,
3.19784654e-01, 7.99879526e-01, 5.07068139e-01, 5.06385001e-01,
2.36194128e-01, 1.45362804e-02, 9.33223900e-01, 8.58257911e-02,
8.44926801e-01, 3.67880739e-01, 9.51022981e-01, 3.99425262e-01,
9.36442062e-01, 5.56159776e-01, 2.40135328e-01, 7.41421670e-01,
6.74389715e-01, 6.84205212e-01, 4.63824360e-01, 2.21888559e-01,
6.40940141e-01, 1.07089496e-01, 6.92222275e-01, 6.35386800e-01,
3.76512526e-01, 7.98523346e-01, 1.94025476e-01, 3.90458914e-01,
7.97933886e-01, 3.80475371e-01, 7.13257864e-01, 6.12517804e-01,
9.41000975e-01, 9.91676717e-01, 7.23676255e-01, 8.08843809e-01,
1.52865023e-01, 7.12890263e-01, 8.47624380e-01, 4.01225748e-01,
5.53250040e-01, 4.79487752e-01, 9.58523000e-01, 3.17277840e-01,
4.02084527e-01, 9.19789111e-04, 4.20184535e-01, 6.31436106e-01,
9.34961094e-01, 9.23676585e-01, 3.27349813e-01, 9.88861026e-01,
1.87674940e-01, 8.23252018e-01, 1.57259487e-01, 4.05098653e-01,
7.34733118e-02, 8.58038931e-01, 8.28798071e-01, 1.39789299e-01,
5.27106477e-01, 2.58141497e-01, 4.91775700e-01, 5.53206006e-01,
1.06262051e-01, 8.64375563e-01, 2.78696319e-01, 4.47123322e-01,
5.72585967e-02, 2.74520641e-03, 1.95159522e-01, 3.41675184e-01,
9.28113421e-01, 8.89728289e-01, 4.80501017e-01, 4.54759819e-01,
6.66993044e-01, 8.58754902e-01, 3.37297506e-01, 7.93654528e-01,
3.98987253e-01, 5.94053882e-01, 7.37466983e-01, 5.01357204e-01,
6.90775771e-01, 6.97313482e-01, 5.71268993e-03, 3.90655333e-02,
1.49038311e-01, 2.13635134e-01, 2.03214607e-01, 4.89655818e-02,
2.17211402e-01, 6.00947369e-01, 8.86075161e-01, 3.49503808e-01,
3.66404799e-01, 4.16741733e-01, 6.80341783e-01, 7.85273952e-01,
9.40958082e-01, 3.75700041e-01, 7.06774204e-01, 3.41634010e-01,
8.24001799e-01, 2.30648904e-01, 8.71013712e-01, 5.05105253e-01,
7.25344032e-01, 5.34993015e-01, 3.16121349e-01, 4.94301899e-01,
5.04504137e-02, 4.29925649e-02, 5.29963326e-01, 4.77174476e-01,
8.33137326e-01, 2.01226179e-02, 5.56883811e-01, 4.90637104e-01,
5.97997138e-01, 7.29578624e-01, 5.24740609e-01, 5.63038220e-01,
4.91783880e-01, 6.15206999e-01, 2.48127215e-01, 5.52613429e-01,
6.72386818e-01, 1.91112781e-01, 9.90367081e-01, 7.46459479e-01,
9.55331267e-01, 2.93946371e-01, 4.43492877e-01, 2.61438554e-01,
4.66916154e-02, 1.66768383e-02, 2.46747841e-01, 8.61262532e-01,
1.63694161e-01, 6.89622600e-01, 2.49381890e-01, 6.55829367e-02,
1.92486562e-01, 6.77926343e-01, 5.01656205e-01, 5.49031041e-01,
4.53164278e-01, 2.74347927e-01, 4.88347831e-01, 9.22446127e-01,
2.01494928e-01, 7.32734105e-01, 2.51129679e-01, 1.93416048e-01,
3.23845382e-01, 9.27791686e-02, 9.35374293e-01, 3.65133998e-01,
1.75837673e-01, 1.36012574e-06, 6.04275084e-02, 2.14651458e-01,
4.17653653e-01, 6.01570231e-01, 9.81030617e-01, 8.90477210e-01,
2.42254082e-01, 6.11545237e-01, 9.21541830e-01, 1.01602348e-01,
8.54038935e-01, 3.96070694e-01, 7.81687436e-01, 3.22597172e-01,
6.25722845e-01, 5.07015829e-01, 1.04421206e-01, 7.59802096e-01,
8.22944245e-01, 4.82984852e-01, 3.88010328e-01, 9.47491670e-01,
3.74309636e-01, 8.79618814e-01, 4.18574053e-01, 3.73516192e-01,

```
                   3.61427635e-01, 5.99974864e-02, 2.77318609e-01, 2.29480126e-01,
                   6.23443605e-02, 5.41568641e-01, 4.42201116e-01, 2.51083393e-02,
                   1.55919107e-01, 9.18707393e-01, 1.33605071e-01, 3.73328219e-01,
                   9.50776588e-01, 1.13355893e-01, 4.09828490e-01, 8.01113833e-01,
                   1.90834612e-02, 6.78240445e-02, 9.29928486e-01, 6.84231597e-01,
                   2.61037929e-01, 4.80239212e-01, 2.48947888e-01, 8.92438590e-01,
                   9.49807887e-01, 7.10268943e-02, 5.24084153e-01, 1.84038208e-01,
                   4.04382242e-01, 7.41194560e-01, 7.13327918e-01, 1.89814845e-01,
                   3.32277599e-01, 4.25267491e-01, 3.86285254e-01, 6.82141859e-01,
                   7.88353431e-01, 3.35430228e-01, 4.41216542e-01, 6.73039747e-02,
                   3.66807163e-01, 9.46957893e-01, 5.02178025e-01, 5.57190453e-01,
                   6.06116486e-01, 5.67179106e-01, 8.04456625e-01, 2.59576440e-01,
                   3.16581419e-01, 8.86823100e-01, 4.99931060e-01, 9.59837567e-01,
                   1.38942300e-01, 7.83158585e-01, 4.73015992e-01, 4.73850129e-01,
                   8.81075365e-01, 1.30645066e-01, 5.14447167e-01, 2.49576346e-01,
                   3.42828931e-01, 8.17322948e-01, 3.98160815e-01, 7.84412438e-01,
                   1.34893179e-02, 9.99338055e-01, 9.55266393e-01, 6.78500713e-01,
                   1.96814815e-01, 6.56830623e-01, 1.90225637e-01, 3.73413799e-01,
                   3.49775229e-02, 7.02310088e-02, 1.57736834e-01, 1.81855247e-0
1]),
          y=array([ 0.75008662,  0.09417666,  1.19650809,  0.37451642, -0.2041420
5,
                   -0.33819795,  1.31074879,  2.45861111,  2.29576375, -0.263602
,
                    1.03336449,  0.41345369,  0.02149154,  0.26629792, -0.7415750
6,
                    2.12843876,  0.13941933,  0.09278932,  2.51838715, -0.7198166
9,
                    1.81508872,  2.95715041,  0.9403663 ,  0.88464485,  0.9510272
7,
                    1.00430003,  0.03283229,  0.24982906, -1.05713485,  2.0501992
3,
                   -0.87266548,  0.15802106,  0.42383253, -0.01458517,  1.8882847
6,
                    0.25827833, -1.06560087,  1.94779533,  1.05094406,  0.5012033
,
                    0.35082674,  1.00883008,  2.16759907,  0.5479995 ,  1.9433025
7,
                   -0.76630404,  1.40032889, -0.95722583,  1.58811037,  0.2069161
2,
                   -0.54209596,  1.22573331,  0.45055963,  0.31622592, -0.1290601
,
                   -0.05739558,  0.26439071,  0.21315409,  0.56723925,  0.2542648
6,
                   -0.53176651,  2.13895587,  0.38070558, -0.76078936,  1.7898727
7,
                   -0.49588571,  0.73266684,  0.4147354 ,  0.91423061,  0.1219517
4,
                    1.02003245,  2.19985468,  0.78720916,  1.00390554,  2.4314800
1,
                   -0.47258907,  0.15706695,  0.56636952,  1.22159248, -0.5056823
4,
                   -0.41017043,  0.89879701, -0.03224255,  0.24182981,  1.6320631
4,
                    1.64486138, -0.80932087,  1.46861026,  0.19102303,  2.5917618
6,
```

```
                    0.61475082, -0.22510609,  0.57831924,  2.07325961,  2.3890995
6,
                    0.84393635,  0.35966613, -0.21784024,  0.19166585,  0.6802334
,
                    0.01125453,  2.85523373, -0.50120996, -0.47905816,  0.9153730
2,
                    1.08686096,  0.42365706, -0.5649983 ,  1.73545037,  2.1899127
3,
                    0.52724967,  0.33933601,  0.70999045,  2.4412068 , -0.8595892
2,
                    1.86303399, -0.16330038,  0.91673298,  1.00344378,  0.3181712
3,
                    0.94559214,  0.84420403,  0.97090454,  0.07365921,  0.2577464
2,
                    0.00852062,  0.28914043, -0.09277626,  0.52016791,  0.5524795
2,
                    2.41290328, -1.08092157, -0.69932589,  0.06598609,  0.0264712
1,
                    1.90004588,  1.87209224, -0.85955145,  0.9645649 , -0.8237625
3,
                    2.61092616,  0.19532082,  2.42380383,  1.08536541,  1.1900391
2,
                    0.97526329, -1.00775589,  0.22216921,  0.65675292,  0.7155777
1,
                    0.5578316 ,  0.51525039,  1.98699711,  0.67821596,  0.2368372
9,
                    0.49233245,  0.32607811,  0.80432993,  0.57934563,  1.1164474
7,
                    0.09494379,  0.23996905,  0.20740278,  0.45858985,  0.9844392
8,
                    0.3529023 ,  0.9762092 ,  1.34307885,  1.97262   , -0.3152587
6,
                    0.10008538, -0.15825726, -0.04477966,  0.38106601,  0.3444397
4,
                    2.62266996,  0.51244867,  0.75335377, -0.28724172,  0.4544488
,
                    0.00998747,  0.60035483,  0.24889306,  0.52390082,  0.1834536
1,
                   -0.44823863,  0.56737035,  2.0543575 ,  0.34402894,  1.4369735
,
                   -0.58209341, -0.20278313, -0.4992271 ,  0.60868399,  2.3458945
1,
                   -0.46880355,  0.58672397,  0.56320099,  0.59328128,  0.3583673
,
                   -0.64214144,  0.19156149,  1.04191856,  2.33779154, -0.9442920
6,
                    0.45147326,  1.99839218,  0.27786795,  2.4558642 , -0.3828809
6,
                    1.45461119, -0.55448458,  0.32190794,  0.20906081,  0.3959172
7,
                    1.88165981,  1.79557898, -0.1651964 , -1.07492764,  2.1599631
,
                   -0.73423223,  2.30265988, -0.55218738, -0.9446279 , -1.3146607
,
                    0.41998007, -0.29836355, -0.54649522,  0.0487746 ,  0.3650115
5,
```

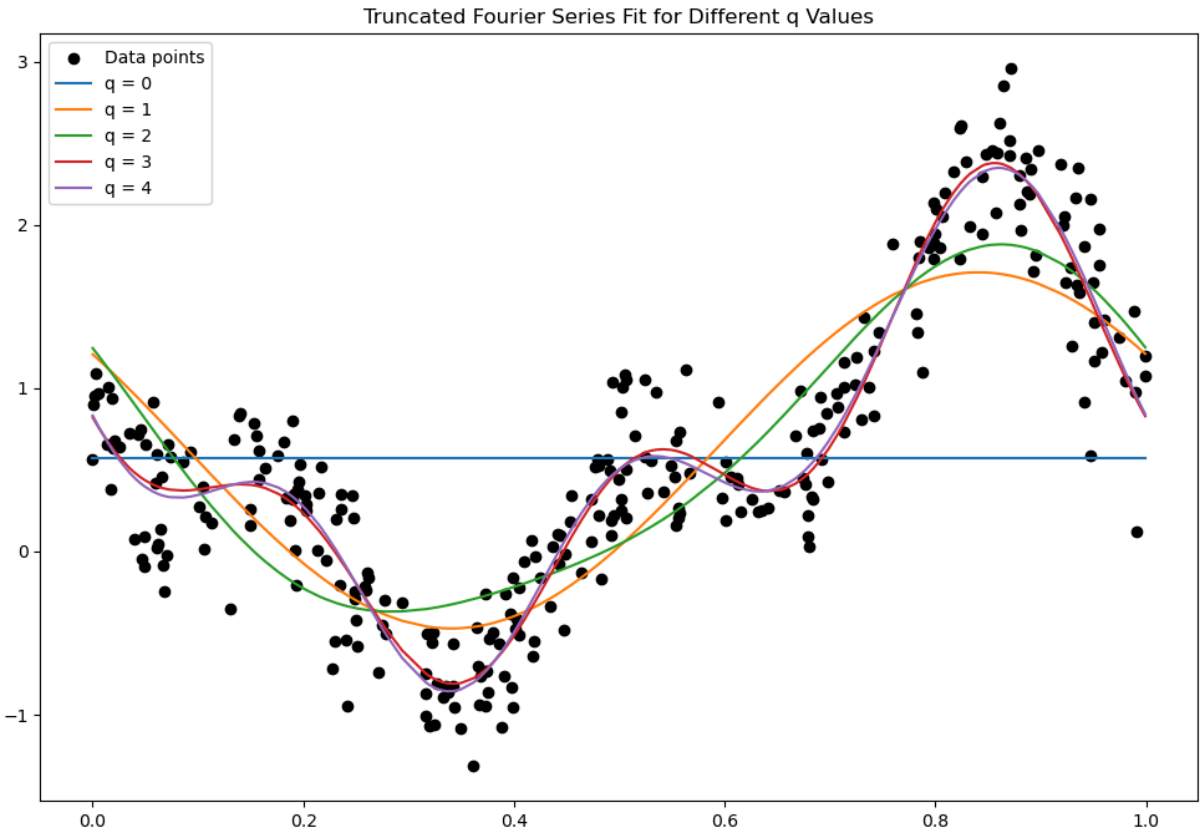
```
                   -0.07652518,  0.64332673,  0.70553691,  2.37414085,  0.6836807
8,
                   -0.26090539,  1.16816606,  0.17566758, -0.05773937,  2.1000611
2,
                    0.62932144, -0.24584656,  1.26036515,  0.74201814, -0.1276846
3,
                    0.21900769, -0.24415288,  1.71830561,  1.64383864,  0.6517989
4,
                    1.05331795,  0.32798735, -0.51148401,  0.83038452,  1.1595597
,
                    0.80359532, -0.89379803, -0.16082992, -0.56334176,  0.3366466
7,
                    1.10123772, -0.82630884,  0.10298674, -0.08398759, -0.9416521
6,
                    0.58958243,  0.85155419,  0.73070105,  0.45993522,  0.4816115
7,
                    1.86128881, -0.24045389, -0.74621631,  2.20294574,  0.4426067
4,
                    1.42077419,  0.83162906,  1.3440051 ,  0.32252839,  0.0641977
7,
                    1.97088699, -0.34990517,  0.7094769 , -0.42302771, -0.9534793
4,
                    2.32323048, -0.83097527,  1.80117135,  0.65288122,  1.0736560
7,
                    1.75142582,  0.22323105,  0.53234858,  0.3671811 ,  0.3236909
1,
                   -0.72972247,  0.72397887, -0.02371739,  0.44163203,  0.6702065
6]))
```

In [8]:

```python
def fit(x, y, q):
    n = len(x)
    A = np.ones((n, 2 * q + 1))
    for k in range(1, q + 1):
        A[:, k] = np.cos(2 * np.pi * k * x)
        A[:, q + k] = np.sin(2 * np.pi * k * x)

    c_star, _, _, _ = np.linalg.lstsq(A, y, rcond=None)
    return c_star
```

In [12]:

```python
plt.figure(figsize=(12, 8))
plt.scatter(t.x, t.y, color='black', label='Data points')

x_plot = np.sort(t.x)
for q in range(5):
    c_fit = fit(t.x, t.y, q)
    y_fit = np.ones_like(x_plot) * c_fit[0]
    for k in range(1, q + 1):
        y_fit += c_fit[k] * np.cos(2 * np.pi * k * x_plot) + c_fit[q + k] * np.sin(
    plt.plot(x_plot, y_fit, label=f'q = {q}')

plt.legend()
plt.title("Truncated Fourier Series Fit for Different q Values")
plt.show()
```



Part 2: Probability

Problem 2.1

To see if p(x, y) is a probability density function(PDF, we need to see if it meets the tw

Non-negativity means $p(x, y) \geq 0$ for all $(x, y) \in \mathbb{R}^2$.

Normalization means that the toal integral over the entire domain must $= 1$:

$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y)\, dxdy = 1$

Non-negativity:
$p(x, y) = ce^{(-1/2)(x-y)^2}$ is always non-negative because exponential functions are always

Normalization: $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y)\, dxdy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} ce^{(-1/2)(x-y)^2} dxdy = 1$

Let $u = x - y$ and $v = x + y$

$x = \frac{u+v}{2}$

$v = (u + y) + y$

$2y = v - u$

$y = \frac{v-u}{2}$

$\frac{\partial(x,y)}{\partial(u,v)} = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$

$\det\left(\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}\right) = \left(\frac{1}{2}\right)\left(\frac{1}{2}\right) - \left(-\frac{1}{2}\right)\left(\frac{1}{2}\right) = \frac{1}{4} - \left(-\frac{1}{4}\right) = \frac{1}{2}$

$\frac{1}{2}c\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}u^2} du dv$

$\int_{-\infty}^{\infty} e^{-\frac{1}{2}u^2} du$ is a Gaussian integral $= \sqrt{2\pi}$

$\frac{1}{2}c\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}u^2} du dv = \frac{1}{2}c\sqrt{2\pi}\int_{-\infty}^{\infty} dv = 2c\sqrt{2\pi}\int_{-\infty}^{\infty} dv = \infty$

Since $p(x,y)$ cannot be normalized it is not a probability density function.

Problem 2.2

Given that p(x,y) is a probability density function we can normalize in order to find C.

$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y) dx dy = 1 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Ce^{-\frac{1}{2}(x-y)^2} e^{-\frac{1}{2}(x+y)^2} dx dy$

$u = x - y \quad v = x + y$

$x = u + y \quad 2y = v - u$

$x = \frac{u+v}{2} \quad y = \frac{v-u}{2}$

$\left|\frac{\partial(x,y)}{\partial(u,v)}\right| = \left|\begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix}\right| = \left|\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}\right| = \frac{1}{2}$

$\frac{C}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}u^2} e^{-\frac{1}{2}v^2} du dv$

$= \frac{C}{2}\int_{-\infty}^{\infty} e^{-\frac{1}{2}u^2} du \int_{-\infty}^{\infty} e^{-\frac{1}{2}v^2} dv$

$= \frac{C}{2}\left(\sqrt{2\pi}\right)\left(\sqrt{2\pi}\right)$

$= \frac{C}{2}(2\pi) = C\pi = 1$

$C = \frac{1}{\pi}$

The above was unnecessary work but done nonetheless.

The density function is symmetric and centered at 0 and therefore the mean of both x and y is 0: m $= \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

The covariance matrix $\Sigma = \begin{bmatrix} \operatorname{var}(x) & \operatorname{cov}(x,y) \\ \operatorname{cov}(x,y) & \operatorname{var}(y) \end{bmatrix}$. Because x and y are independent and identically distributed, the variance can be given by the standard normal distribution:
$\operatorname{var}(x) = \operatorname{var}(y) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} x^2 e^{-x^2} dx$

$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} x^2 e^{-x^2} dx$

$u = e^{-x^2} \quad v = \frac{-x^3}{3}$

$du = -2xe^{-x^2} dx \quad dv = x^2 dx$

$\frac{x^3}{3} e^{-x^2} + \frac{2}{3} \int \left(x^3\right) \left(-2xe^{-x^2}\right) dx$

$= \frac{1}{\sqrt{\pi}} \left(\frac{\sqrt{\pi}}{2}\right) = \frac{1}{2}$

$\operatorname{cov}(x,y) = 0$ since x and y are independent.

$\therefore \Sigma = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$

$m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$

Problem 2.3

A has $\frac{2}{3}$ red, $\frac{1}{3}$ white.

B has $\frac{1}{3}$ red, $\frac{2}{3}$ white.

Fair coin has $\frac{1}{2}$ heads, $\frac{1}{2}$ tails.

Solve for $P(A|n\text{red})$.

$P(A) = P(\text{H}) = \frac{1}{2}, \quad P(B) = P(\text{T}) = \frac{1}{2}$

$P(R|A) = \frac{2}{3}, \quad P(W|A) = \frac{1}{3}, \quad P(R|B) = \frac{1}{3}, \quad P(W|B) = \frac{2}{3}$

Bayes' Theorem: $\quad P(A|n\text{red}) = \frac{P(n\text{red}|A) \cdot P(A)}{P(n\text{red})}$

Where $P(A|n\text{red})$ is the probability of die A being used if the first n die throws come up

Where $P(n\text{red}|A)$ is the probability of getting n reds given the die A is used.

$P(n\text{red})$ is the overall probability of getting n red outcomes.

$P(n\text{red}) = P(n\text{red}|A) \cdot P(A) + P(n\text{red}|B) \cdot P(B)$

$$P(n\text{red}|A) = \frac{2}{3}^n, \quad P(n\text{red}|B) = \frac{1}{3}^n$$

$$P(n\text{red}) = \frac{2}{3}^n \cdot \frac{1}{2} + \frac{1}{3}^n \cdot \frac{1}{2} = \frac{1}{2}((\frac{2}{3})^n + (\frac{1}{3})^n)$$

$$\therefore P(A|n\text{red}) = \frac{P(n\text{red}|A) \cdot P(A)}{P(n\text{red})} = \frac{(\frac{2}{3})^n \cdot \frac{1}{2}}{\frac{1}{2}((\frac{2}{3})^n + (\frac{1}{3})^n} = \frac{(\frac{2}{3})^n}{(\frac{2}{3})^n + (\frac{1}{3})^n}$$

$$P(A|n\text{red}) = \frac{(\frac{2}{3})^n}{(\frac{2}{3})^n + (\frac{1}{3})^n} = \frac{2^n}{3^n} \cdot \frac{3^n}{2^n + 1}$$

$$P(A|n\text{red}) = \frac{2^n}{2^n + 1}$$

Part 3: Geometry

Problem 3.1

In [15]:
```
# retrieve is defined in a hidden cell
file_name = 'lines.pkl'
retrieve(file_name)
with open(file_name, 'rb') as file:
    lines = pickle.load(file)
```

```
Using previously downloaded file lines.pkl
```

In [16]:
```
lines
```

Out[16]:
```
[namespace(A=array([[1., 0., 0.],
                    [0., 1., 0.]]),
           b=array([0., 0.])),
 namespace(A=array([[ 0.,  0., -2.],
                    [ 0.,  2.,  2.]]),
           b=array([4., 6.])),
 namespace(A=array([[2., 1., 3.],
                    [0., 5., 1.]]),
           b=array([14.,  4.]))]
```

In [33]:
```
from scipy.linalg import null_space

def parametric(A, b):
    p = np.linalg.lstsq(A, b, rcond=None)[0]
    n = null_space(A).flatten()
    n = n / np.linalg.norm(n)
    alpha_values = [0, 2]
    residuals = [np.linalg.norm(A @ (p + alpha * n) - b) for alpha in alpha_values]
    p_rounded = [float(np.round(val, 3)) for val in p]
    n_rounded = [float(np.round(val, 3)) for val in n]
    residuals_rounded = [float(np.round(res, 3)) for res in residuals]
    print(f"p = {p_rounded}, n = {n_rounded}, r = {residuals_rounded}")
    return p, n
```

```
for ns in lines:
    parametric(ns.A, ns.b)
```

```
p = [0.0, 0.0, 0.0], n = [0.0, 0.0, 1.0], r = [0.0, 0.0]
p = [0.0, 5.0, -2.0], n = [1.0, 0.0, 0.0], r = [0.0, 0.0]
p = [2.213, 0.173, 3.133], n = [-0.808, -0.115, 0.577], r = [0.0, 0.0]
```

Part 4: Calculus

Problem 4.1

$f_x = 3(x-1) - 3y^2$

Find where $f_x = 0$:

$f_x = 3(x-1)^2 - 3y^2 = 0$

$(x-1)^2 - y^2 = 0$

$x - 1 = \pm y$

$x - 1 = 0 \quad \Rightarrow \quad x = 1$

$f_y = -6xy + 6y$

Now, find where $f_y = 0$:

$f_y = -6xy + 6y = 0$

$y(1-x) = 0$

$y = 0$

Stationary point at $x = 1, y = 0$.

Find the second partial derivatives:

$f_{xx} = 6(x-1)$

$f_{yy} = -6x + 6 = 6(1-x)$

$f_{xy} = -6y$

Evaluate each at (1,0):

$f_{xx} = 6(2-1) = 0$

$f_{yy} = 6(1-1) = 0$

$f_{xy} = -6(0) = 0$

Second Derivative Test:

$f_{xx}(x,y)f_{yy}(x,y) - (f_{xy}(x,y))^2$

= $(0)(0) - (0)^2$

= 0

The second derivative test is inconclusive. Analysis of the actual structure of f(x,y) near (1,0) shows that the function behaves like a saddle point because of the cubic term of x combined with the quadratic term of y. The quadratic term of y would be at its minimum (the bottom of the parabolic function) and the cubic term of x would be a local minimum from one directions and a local maximium from another direction. Therefore the stationary point at (1,0) is a saddle point.