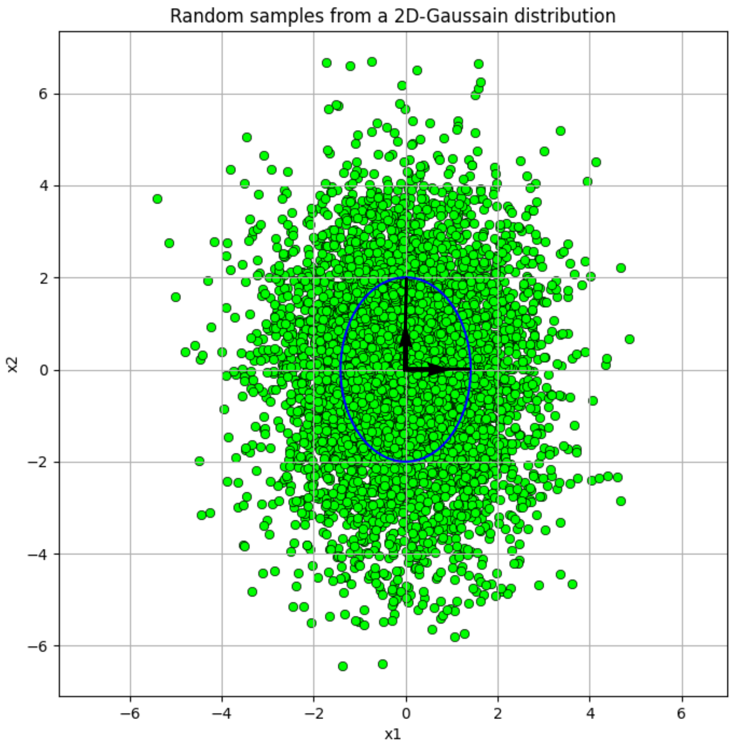
Q1

(a)



(b)

$$\begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} = 1$$

here $x = x_1$
$y = x_2$.

$$\begin{bmatrix} x & y \end{bmatrix} \times \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = 1$$

$[\frac{1}{2}x, \frac{1}{4}y] \times \begin{bmatrix} x \\ y \end{bmatrix}$

$\frac{1}{2}x^2 + \frac{1}{4}y^2$

$$\frac{1}{2}x^2 + \frac{1}{4}y^2 = 1$$

(c)

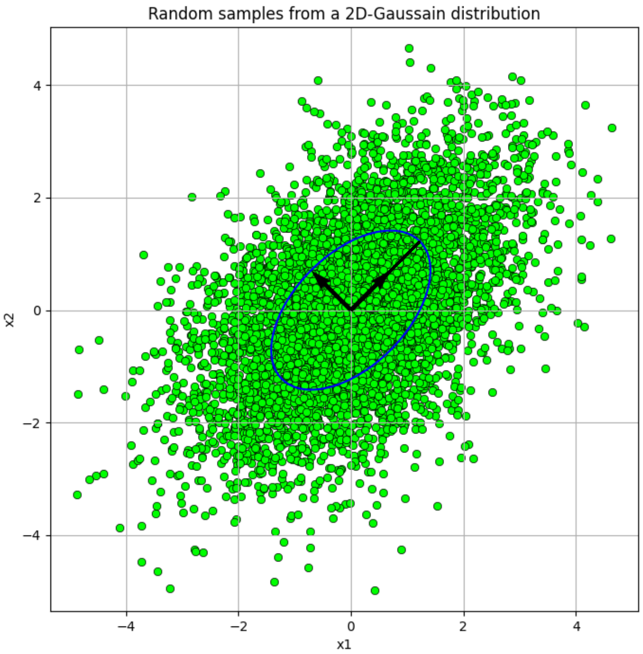
$$Q = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} \qquad \det(Q - \lambda I) = \lambda^2 - \frac{3}{4}\lambda + \frac{1}{8}$$

$\lambda_1 = \frac{1}{4} \qquad \lambda_2 = \frac{1}{2}$

$\lambda_1 = \frac{1}{4} \quad \bar{x}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ with length 2 $\qquad \lambda_2 = \frac{1}{2} \quad \bar{x}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ with length $\sqrt{2}$

(D) same graph with (a)

(e)



$$\begin{bmatrix} x & y \end{bmatrix} \times \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} = 1$$

$$\begin{bmatrix} \frac{2}{3}x-\frac{1}{3}y \,,\, -\frac{1}{3}x+\frac{2}{3}y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 1$$

$$\frac{2}{3}x^2 - \frac{1}{3}yx \quad -\frac{1}{3}xy + \frac{2}{3}y^2 = 1$$

$$\begin{bmatrix} x , & y \end{bmatrix} \times \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = 1$$

$$\frac{2}{3}x^2 - \frac{2}{3}xy + \frac{2}{3}y^2 = 1$$

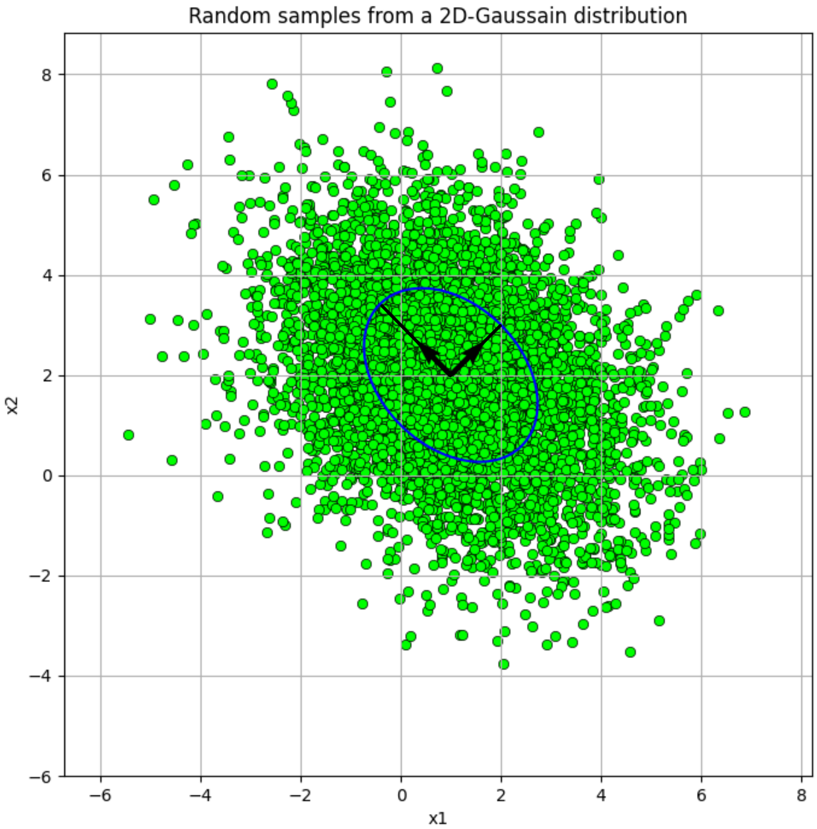
$$Q = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{bmatrix} \quad \det(Q-\lambda I) = \lambda^2 - \frac{4}{3}\lambda + \frac{1}{3}$$

$$\lambda_1 = \frac{1}{3} \quad \lambda_2 = 1$$

$$\lambda_1 = \frac{1}{3} \quad x_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \bar{x}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$ with length $\sqrt{3}$

$$\lambda_2 = 1 \quad x_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \bar{x}_2 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$ with length 1

(f)



$$\begin{bmatrix} x-1 \\ y-2 \end{bmatrix}^T \times \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}^{-1} \times \begin{bmatrix} x-1 \\ y-2 \end{bmatrix} = 1$$

$$\frac{3}{8}x^2 + \frac{3}{8}y^2 - \frac{10}{8}x - \frac{14}{8}y + \frac{2xy}{8} + \frac{19}{8} = 1$$

$$Q = \begin{bmatrix} \frac{3}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{3}{8} \end{bmatrix} \qquad \det(Q-\lambda I) = \lambda^2 - \frac{3}{4}\lambda + \frac{1}{8}$$

$$\lambda_1 = \frac{1}{4} \qquad \lambda_2 = \frac{1}{2}$$

$\lambda_1 = \frac{1}{4}$ $\quad x_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ $\quad \bar{x}_1 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ with length 2

$\lambda_2 = \frac{1}{2}$ $\quad x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\quad \bar{x}_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ with length $\sqrt{2}$

Code:

example code of Q1.(f)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal

plt.rcParams['figure.figsize']=8,8

# Define the equation function
def equation(x, y):
    return ((3*x**2-10*x+3*y**2+2*x*y-14*y+19)/8)-1

# Create a grid of x and y values
x = np.linspace(-6, 6, 500)
y = np.linspace(-6, 6, 500)
X, Y = np.meshgrid(x, y)
# Calculate the function values for each point on the grid
Z = equation(X, Y)

def generate_and_plot(kx,mu):
    distr = multivariate_normal(
        cov = Kx, mean = mu,
        seed = 1000
    )
    data = distr.rvs(size = 5000)
    plt.grid()

    plt.plot(data[:,0],data[:,1],'o',c='lime',markeredgewidth = 0.5, markeredgecolor = 'black',zorder=1)

    # Create a contour plot
    plt.contour(X, Y, Z, levels=[0], colors='b', zorder=2)

    #principal axes
    vector1 = np.array([-1, 1])
    #unit_vector1 = vector1 * ((3-2**0.5)/7)/ np.linalg.norm(vector1)
    unit_vector1 = vector1 / np.linalg.norm(vector1)
    plt.quiver(1, 2, unit_vector1[0], unit_vector1[1], angles='xy', scale_units='xy', scale=1, zorder=3)
    vector2 = np.array([1,1])
    # unit_vector2 = vector2 * ((3+2**0.5)/7)/ np.linalg.norm(vector2)
    unit_vector2 = vector2 / np.linalg.norm(vector2)
    plt.quiver(1, 2, unit_vector2[0], unit_vector2[1], angles='xy', scale_units='xy', scale=1, zorder=3)

    plt.title('Random samples from a 2D-Gaussain distribution')
    plt.xlabel('x1')
    plt.ylabel('x2')
    plt.axis('equal')

    plt.show()

Kx = np.array([[3.0, -1.0],[-1.0, 3.0]])
mu = np.array([1,2])
random_seed = 10

generate_and_plot(Kx, mu)
```

$2\times2 \quad 2\times1$

Q_2 $A = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & \frac{3}{4} \end{bmatrix} \begin{bmatrix} a_k \\ b_k \end{bmatrix}$

$\frac{1}{2}a_k + \frac{1}{4}b_k$
$\frac{1}{2}a_k + \frac{3}{4}b_k.$

(a) $\det(A - \lambda I) = \lambda^2 - \frac{5}{4}\lambda + \frac{1}{4}$ $\quad \lambda_1 = \frac{1}{4} \quad \lambda_2 = 1$

$\lambda_1 = \frac{1}{4} \quad x_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \qquad \lambda_2 = 1 \quad x_2 = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}$

$X = \begin{bmatrix} -1 & \frac{1}{2} \\ 1 & 1 \end{bmatrix} \qquad X^{-1} = \begin{bmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix}$

$A^k = \begin{bmatrix} -1 & \frac{1}{2} \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} \frac{1}{4}^k & 0 \\ 0 & 1^k \end{bmatrix} \times \begin{bmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix}$

when $k \to \infty$

$(\frac{1}{4})^k \Rightarrow 0 \qquad 1^k = 1$

$= \begin{bmatrix} -1 & \frac{1}{2} \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix}$

$= \begin{bmatrix} 0 & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix}$

$= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix}$

(b) $A^k \times \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix} \times \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix}$

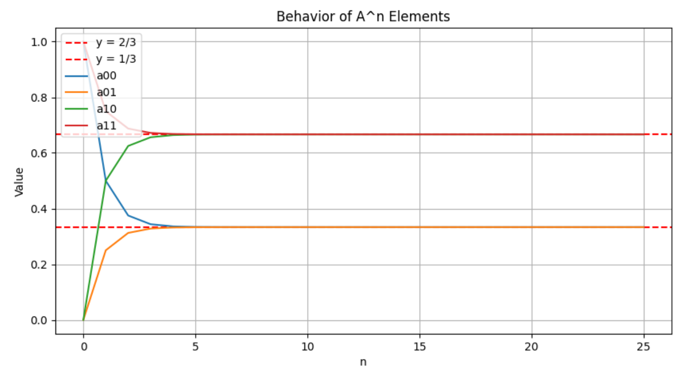
$\therefore$ city A will not be deserted eventually.

(c)

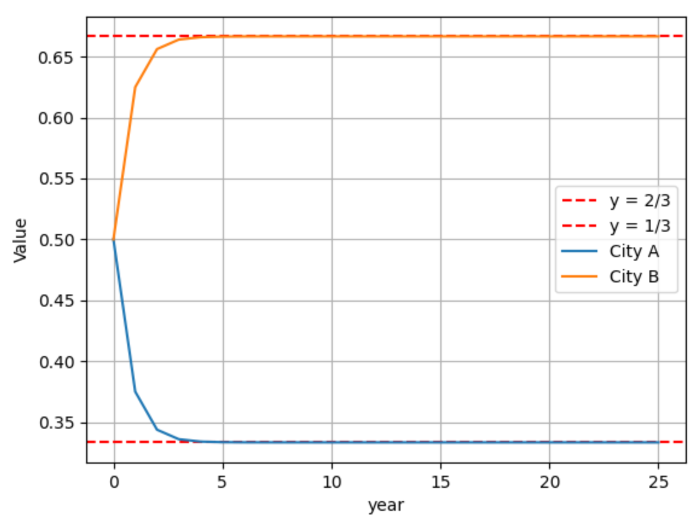
code

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

def plot_matrix_powers(A, n_max, int_p):
    results = []  # Store the results of A^n
    powers = range(n_max + 1)

    for n in powers:
        An = np.linalg.matrix_power(A, n)  # Calculate A^n
        results.append(An)

    # Extract the individual matrix elements for plotting
    elements = [[], [], [], []]

    pop_result = [[], []]

    for i in range(len(results)):
        elements[0].append(results[i][0, 0])
        elements[1].append(results[i][0, 1])
        elements[2].append(results[i][1, 0])
        elements[3].append(results[i][1, 1])

        pop_result[0].append((results[i]@int_p)[0,0])
        pop_result[1].append((results[i]@int_p)[1,0])

    plt.axhline(y=2/3, color='red', linestyle='--', label='y = 2/3')
    plt.axhline(y=1/3, color='red', linestyle='--', label='y = 1/3')
    plt.plot(powers,pop_result[0],label='City A')
    plt.plot(powers,pop_result[1],label='City B')
    plt.xlabel('year')
    plt.ylabel('Value')
    plt.legend()
    plt.grid(True)
    # Create the plot
    plt.figure(figsize=(10, 5))
    plt.axhline(y=2/3, color='red', linestyle='--', label='y = 2/3')
    plt.axhline(y=1/3, color='red', linestyle='--', label='y = 1/3')
    plt.plot(powers, elements[0], label='a00')
    plt.plot(powers, elements[1], label='a01')
    plt.plot(powers, elements[2], label='a10')
    plt.plot(powers, elements[3], label='a11')
    plt.xlabel('year')
    plt.ylabel('Value')
    plt.title('Behavior of A^n Elements')
    plt.legend()
    plt.grid(True)
    plt.show()
# Define your 2x2 matrix A
A = np.array([[0.5, 0.25],
              [0.5, 0.75]])
int_p = np.array([[0.5],
                  [0.5]])
# Choose the maximum power of A to calculate
n_max = 25
# Plot the behavior of A^n elements
plot_matrix_powers(A, n_max, int_p)
```

for Qa



for Qb



Q3 $\quad 13x^2+10xy+13y^2=72$ $\qquad$ 顶点 (0, 0)

$$Q=\begin{bmatrix}13 & 5\\ 5 & 13\end{bmatrix} \quad \det(Q-\lambda I)=\lambda^2-26\lambda+14$$

$$\lambda_1=8 \quad \lambda_2=18$$

$$\lambda_1=8 \quad x_1=\begin{bmatrix}-1\\ 1\end{bmatrix} \quad \bar{x}_1=\begin{bmatrix}-\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix} \quad \text{length}=\sqrt{\frac{72}{8}}=3$$

$$\lambda_2=18 \quad x_2=\begin{bmatrix}1\\ 1\end{bmatrix} \quad \bar{x}_2=\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix} \quad \text{length}=\sqrt{\frac{72}{18}}=2$$

$$\frac{1}{2}x^2-\frac{1}{2}xy+\frac{1}{2}y^2$$

$$[x \ \ y]\begin{bmatrix}\frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix}$$

$$[u, v]=[x, y]\begin{bmatrix}\frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix}$$

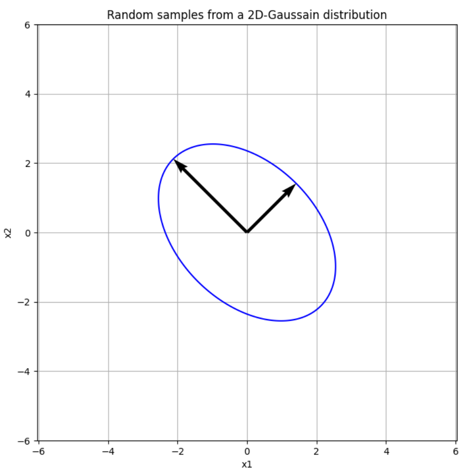
(a) $\quad a=3 \qquad b=2$

$$u=-\frac{1}{\sqrt{2}}x+\frac{1}{\sqrt{2}}y \qquad v=\frac{1}{\sqrt{2}}x+\frac{1}{\sqrt{2}}y$$

(b) principal axes are $3\times\begin{bmatrix}-\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix}$ with length 3

$2\times\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix}$ with length 2.

(c)



File Edit Selection View Go Run Terminal Help

codework1.py 3 Q3.py 3

C: > Users > wangs > OneDrive > 文档 > 510cod > Q3.py > ...

```python

import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal

plt.rcParams['figure.figsize']=8,8

# Define the equation function
def equation(x, y):
    return 13*x**2 + 10*x*y + 13*y**2-72

# Create a grid of x and y values
x = np.linspace(-6, 6, 500)
y = np.linspace(-6, 6, 500)
X, Y = np.meshgrid(x, y)
# Calculate the function values for each point on the grid
Z = equation(X, Y)

plt.grid()

plt.contour(X, Y, Z, levels=[0], colors='b', zorder=2)

#principal axes
vector1 = np.array([-1.0, 1.0])
#unit_vector1 = vector1 * ((3-2**0.5)/7)/ np.linalg.norm(vector1)
unit_vector1 = vector1 *((72/8)**0.5)/ np.linalg.norm(vector1)
plt.quiver(0, 0, unit_vector1[0], unit_vector1[1], angles='xy', scale_units='xy', scale=1, zorder=3)
vector2 = np.array([1.0,1.0])
# unit_vector2 = vector2 * ((3+2**0.5)/7)/ np.linalg.norm(vector2)
unit_vector2 = vector2 *((72/18)**0.5) / np.linalg.norm(vector2)
plt.quiver(0, 0, unit_vector2[0], unit_vector2[1], angles='xy', scale_units='xy', scale=1, zorder=3)

plt.title('Random samples from a 2D-Gaussain distribution')
plt.xlabel('x1')
plt.ylabel('x2')
plt.axis('equal')

plt.show()
```