Exercise1

November 3, 2021

1 Exercise 1. Dynamical Systems Theory in Machine Learning & Data Science

2 Name:- Kaushal Kumar (Email:- kaushal.kumar@stud.uni-heidelberg.de)

Remark: I am submitting this exercise group individually.

3 Task 1. A simple SIR model

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

[2]:
```
N=1000 # total population

# initial conditions : one infected, rest susceptible
I0=1
R0=0
S0= N-I0-R0

# parameter
beta=2.4
gamma=1.0/20

# time points (days)
t=np.linspace(0,199,200)  ## Grid of time points (in days)
```

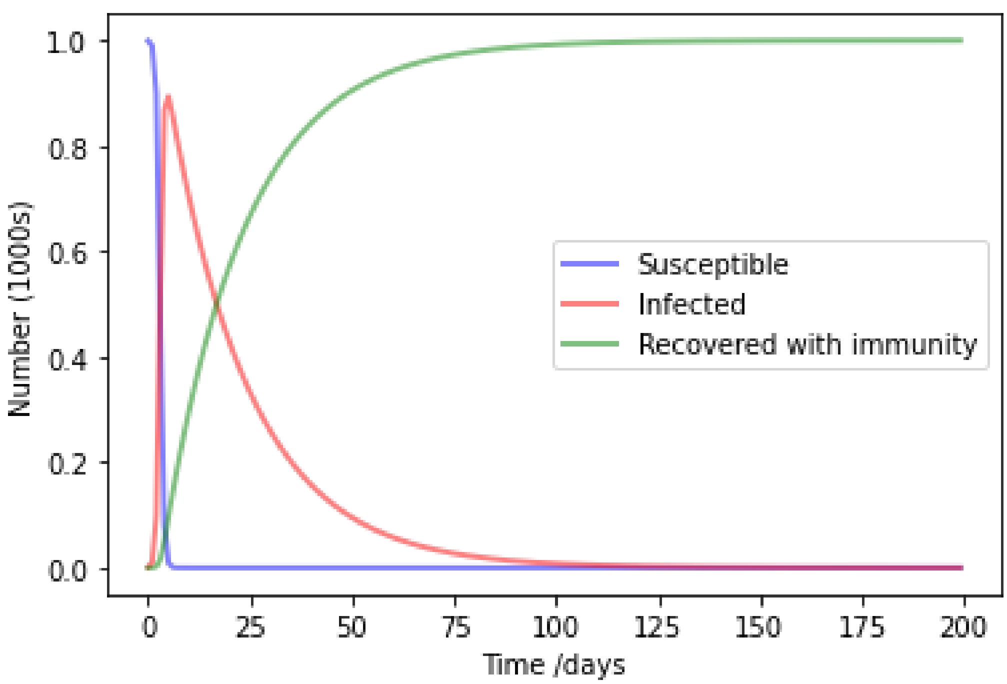
[3]:
```
# SIR diff equ

def SIR(y,t,N,beta,gamma):
    S,I,R= y
    dSdt= -beta*S*I/N
    dIdt= beta*S*I/N -gamma*I
```

```
    dRdt= gamma*I
    return dSdt, dIdt, dRdt
```

[4]:
```
# initial condition vector
y0= S0, I0, R0

# Integrate the SIR model over the time grid, t.

sol=odeint(SIR, y0, t, args=(N,beta,gamma))
S,I,R =sol.T
```

1. Plot for the time evolution untill the dynamics run into a fixed point

[5]:
```
# Plot the data on three separate curves for S(t), I(t) and R(t)

plt.plot(t, S/1000, 'b', alpha=0.5, lw=2, label='Susceptible')
plt.plot(t, I/1000, 'r', alpha=0.5, lw=2, label='Infected')
plt.plot(t, R/1000, 'g', alpha=0.5, lw=2, label='Recovered with immunity')
plt.xlabel('Time /days')
plt.ylabel('Number (1000s)')
plt.legend()
plt.show()
```



[6]:
```
# Determine fixed point for this systems
import sympy as sm
```

```python
S,I,R=sm.symbols('S,I,R')
P=-2.4*S*I/1000
Q=2.4*S*I/1000 -(1/20)*I
K=I*1.0/20

# set P(S,I,R)=0 , Q(S,I,R)=0 and K(S,I,R)=0

Peqn=sm.Eq(P,0)
Qeqn=sm.Eq(Q,0)
Keqn=sm.Eq(K,0)
criticalpoints=sm.solve((Peqn,Qeqn,Keqn),S,I,R)
print(criticalpoints)
```

```
[(S, 0.0, R)]
```

[ ]:

```python

```

3.Consider the case of mortality rate of 1 percentage. This is the case of SIRD model, where D is death

[7]:

```python
# SIRD diff equ
def SIRD(y,t,N,beta,gamma,mu):
    S,I,R,D= y
    dSdt= -beta*S*I/N
    dIdt= beta*S*I/N -gamma*I -mu*I
    dRdt= gamma*I
    dDdt= mu*I
    return dSdt, dIdt, dRdt, dDdt
```

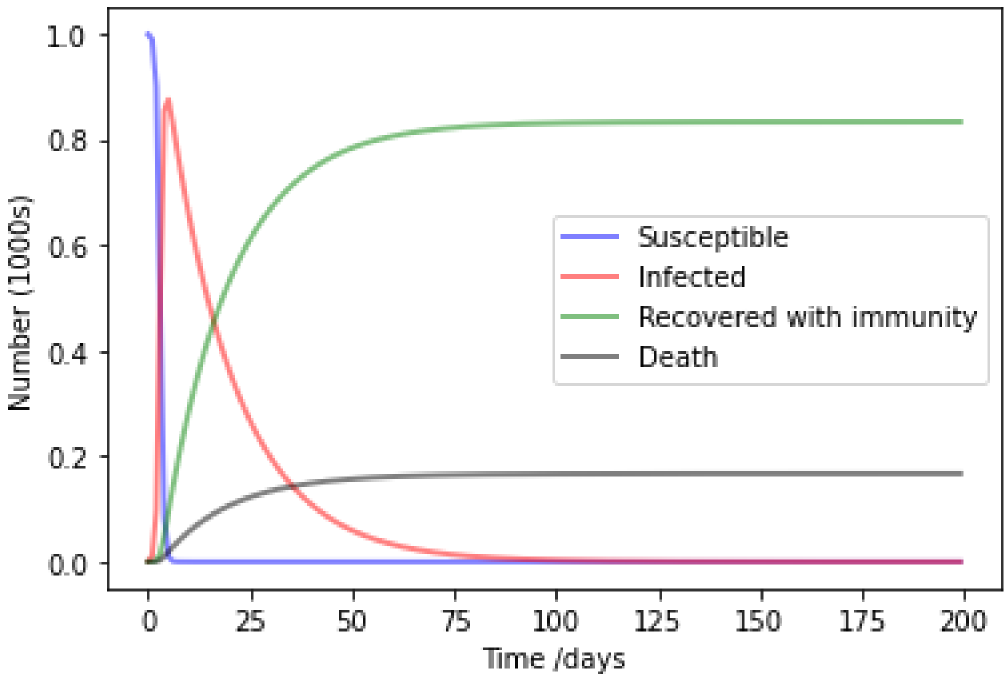
[8]:

```python
N = 1000
beta=2.4
gamma=1.0/20
mu= 0.01 # mortality rate
# 60 percentage of the total population are susceptible to infection initially.
S0, I0, R0, D0 =N-I0-R0, 1, 0, 0  # initial conditions: one exposed

# time points (days)
t=np.linspace(0,199,200)  ## Grid of time points (in days)
```

[9]:

```python
# initial condition vector
y0= S0, I0, R0, D0

# Integrate the SIR model over the time grid, t.

sol=odeint(SIRD, y0, t, args=(N,beta,gamma,mu))
S,I,R,D =sol.T
```

[10]:
```
# Plot the data on three separate curves for S(t), I(t) and R(t), D(t)

plt.plot(t, S/1000, 'b', alpha=0.5, lw=2, label='Susceptible')
plt.plot(t, I/1000, 'r', alpha=0.5, lw=2, label='Infected')
plt.plot(t, R/1000, 'g', alpha=0.5, lw=2, label='Recovered with immunity')
plt.plot(t, D/1000, 'k', alpha=0.5, lw=2, label='Death')
plt.xlabel('Time /days')
plt.ylabel('Number (1000s)')
plt.legend()
plt.show()
```



[ ]:

```
```

4 Task 4. Flow field of linear ODEs

Solution 4.1 (fixed point)

[11]:
```
# Determine fixed point for this systems
import sympy as sm
x1,x2=sm.symbols('x1,x2')
P=0.1*x1-0.3*x2
Q=0.2*x1-0.3*x2
```

```python
# set P(x1,x2)=0 and Q(x1,x2)=0.

Peqn=sm.Eq(P,0)
Qeqn=sm.Eq(Q,0)
criticalpoints=sm.solve((Peqn,Qeqn),x1,x2)
print(criticalpoints)
```

```
{x1: 0.0, x2: 0.0}
```

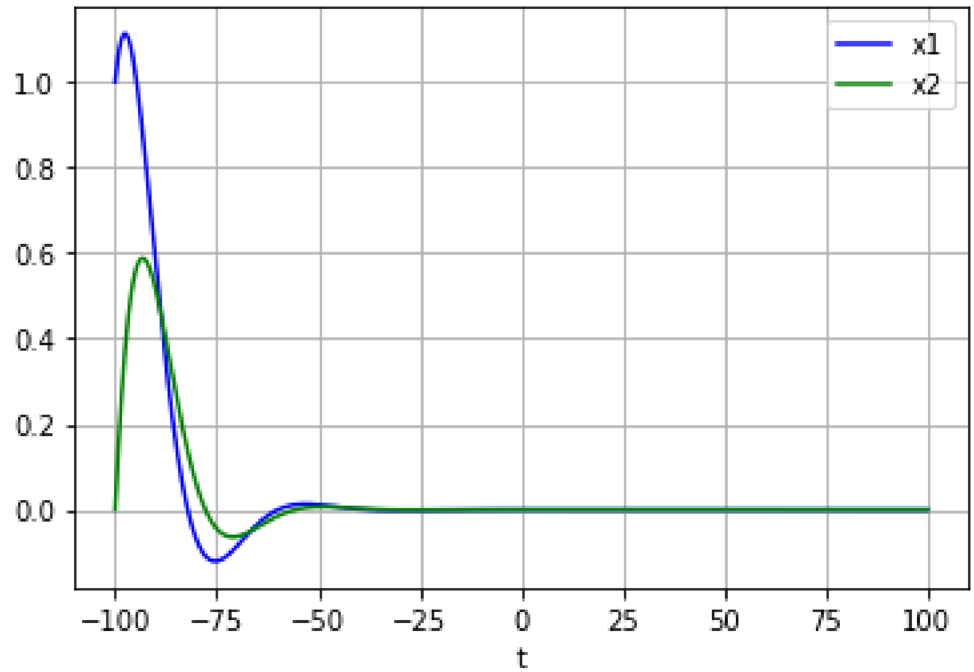
Note: I have attached the solution for fixed point calculated manually.

[12]:
```python
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint
import pylab as pl
```

[13]:
```python
# Linear ODEs
def dx_dt(x,t):
    return [0.1*x[0]-0.3*x[1], 0.2*x[0]-0.3*x[1]]
```

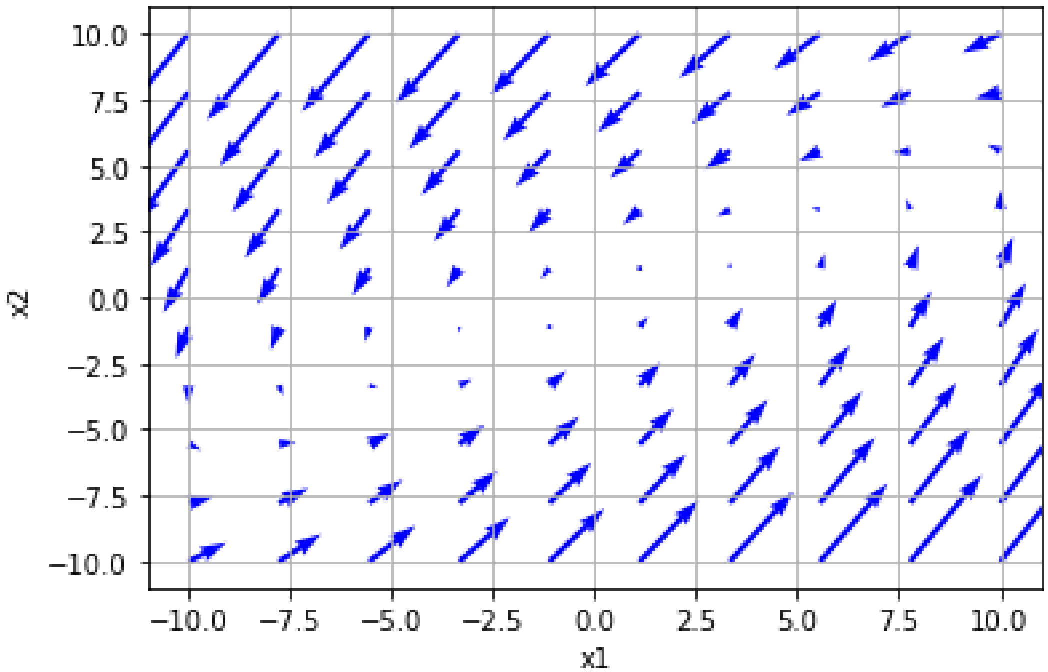
Solution: 4.2. (time evolution for the system)

[14]:
```python
# Trajectories
t = np.linspace(-100,100,1000)

# Initial conditions
x0=[1,0]

sol=odeint(dx_dt,x0,t)



plt.plot(t, sol[:, 0], 'b', label='x1')
plt.plot(t, sol[:, 1], 'g', label='x2')
plt.legend()
plt.xlabel('t')
plt.grid()
plt.show()
```



Above plot is the time evolution of the system for a given initial vector

[15]:
```
# Plot the flow field.
X,Y = np.mgrid[-10:10:10j, -10:10:10j]
x1 = 0.1*X -0.3*Y
x2= 0.2*X -0.3*Y
pl.quiver(X, Y, x1, x2, color = 'b')
plt.xlabel('x1')
plt.ylabel('x2')
plt.grid()
plt.show()
```



Above plot is the Flow field of the given system (solution 4.3)

Since the the trajectories spiral counterclockwise around the origin. Hence the fixed point (0,0) is called the stable focus.

(solution 4.4)

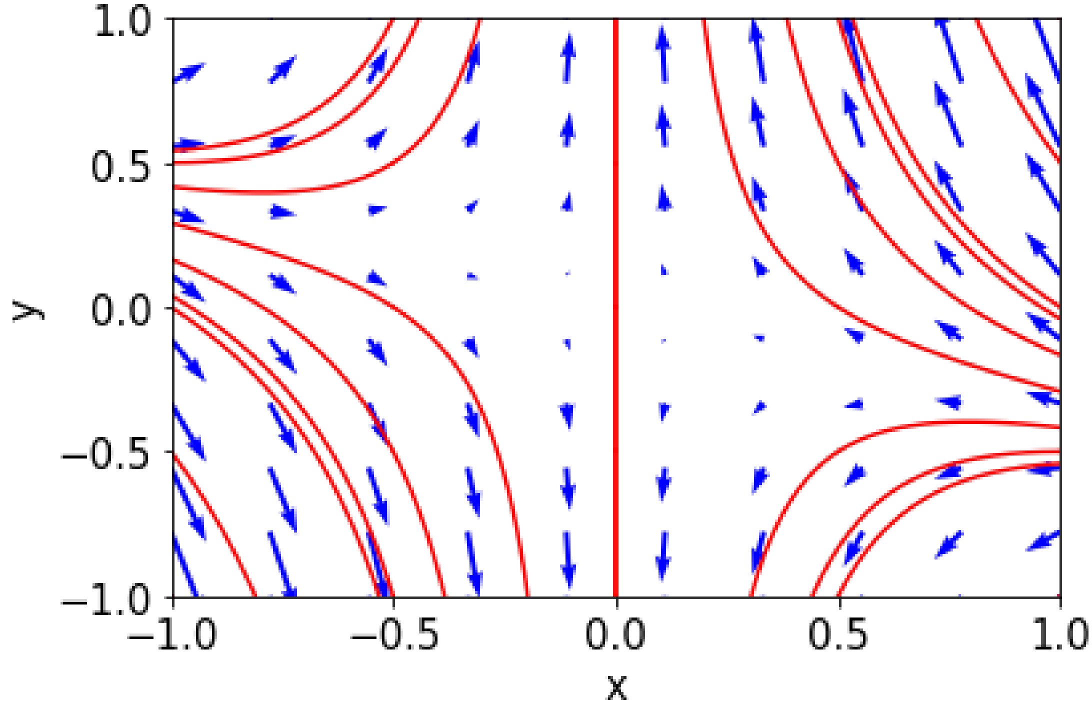
Different types of dynamics with own choice of parameter.

$\dot{X} = A.X$, where $A = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$ and $X = \begin{vmatrix} x1 \\ x2 \end{vmatrix}$.

[16]:
```
# Phase portrait with Field flow
```

[17]:
```
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint
import pylab as pl

# The 2-dimensional linear system.
a, b, c, d = -1, 0, 1, 2 # parameter of the system

def dx_dt(x, t):
 return [a*x[0] + b*x[1], c*x[0] + d*x[1]]

# Trajectories in forward time.
ts = np.linspace(0, 4, 100)
```

```python
ic = np.linspace(-1, 1, 5)
for r in ic:
    for s in ic:
        x0 = [r, s]
        xs = odeint(dx_dt, x0, ts)
        plt.plot(xs[:,0], xs[:,1], "r-")

# Trajectories in backward time.
ts = np.linspace(0, -4, 100)
ic = np.linspace(-1, 1, 5)
for r in ic:
    for s in ic:
        x0 = [r, s]
        xs = odeint(dx_dt, x0, ts)
        plt.plot(xs[:,0], xs[:,1], "r-")

# Label the axes and set fontsizes.
plt.xlabel('x', fontsize=15)
plt.ylabel('y', fontsize=15)
plt.tick_params(labelsize=15)
plt.xlim(-1, 1)
plt.ylim(-1, 1);


# Plot the Field flow
X,Y = np.mgrid[-1:1:10j, -1:1:10j]
u = a*X + b*Y
v = c*X + d*Y
pl.quiver(X, Y, u, v, color = 'b')
plt.show()
```



Above plot, fixed point (0,0) is the saddle point.

Note: Along with this jupyter notebook, hand written solution is also attached together

Kaushal kumar

Task 2: Higher order ODE systems

- Show that any n-th order ODE of the form $f(t, x, \frac{dx}{dt}, \ldots \frac{d^n x}{dt^n}) = 0$ can always be transformed into a system of n first order ODEs.

$\rightarrow$

Solution:- Consider the nth order differential equation in Explicit form

$$\frac{d^n x}{dt^n} = f(t, x, \frac{dx}{dt}, \frac{d^2 x}{dt^2}, \ldots, \frac{d^{n-1} x}{dt^{n-1}}), \quad (n \geq 1) \quad \text{—} (1)$$

A function $x(t)$ is a solution of this equation in an Interval I if it is n-times differential in I and satisfies equation —(1) identically for $x \in I$. Equation (1) can be transferred into a system of n first order differential equations for n function $x_1(t), x_2(t), \ldots, x_n(t)$:

Let $x_1 = x \quad , x_2 = \dot{x}$ where " $\cdot$ " $\rightarrow$ is derivative w.r.to time.

$$\left.\begin{aligned} \therefore \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= f(t, x_1, x_2, \ldots, x_{n-1}) \end{aligned}\right\} \quad \text{—} (2)$$

Hence, Equation (1) and equation (2) are equivalent in the following sense: If $x(t)$ is a soln of (1), then the vector function $x = (x_1, x_2, \ldots, x_n) := (x, \dot{x}, \ldots, x^{(n-1)})$ is a soln of (2).

$\square$

Task 3. ODEs on the real line

Solution :- The fact that $\dot{x} = f(x)$ corresponds to flow on a line

If a fixed point is there such that it is stable fixed point then the approach to the stable fixed point is always monotonic, ~~which mean~~ If we flow monotonically on a line, we will never come back to our starting place, which mean periodic solution are impossible.

Note :- (Reference : Nonlinear dynamics and chaos : Steven H. Strogatz)(Sec : 2.7)

Another proof that solution to $\dot{x} = f(x)$ can't oscillate :-

Let $\dot{x} = f(x)$ be a vector field on the line. Use the existence of of a potential function $V(x)$ to show that solution $x(t)$ can't oscillate.

→ Assume for, $\dot{x} = f(x) = -\frac{dV}{dx}$

there is an oscillatory solution $x(t)$ with period $T \neq 0$.

Then,

$$x(t) = x(t+T) \Rightarrow V(x(t)) = V(x(t+T))$$

but $\frac{dV}{dt} \leq 0$ $\quad \left(\because \frac{dV}{dt} = -\left(\frac{dV}{dx}\right)^2 \right)$

$$\Rightarrow x(t) \text{ is constant}$$

In other words, the solution $x(t)$ doesn't oscillate and never moves.

$\Rightarrow$ contradictions!

$\square$ //

Task 4: Flow field of linear ODEs.

$$\frac{dx}{dt} = A.X \quad \text{where} \quad A = \begin{bmatrix} 0.1 & -0.3 \\ 0.2 & -0.3 \end{bmatrix} \text{ and } X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

$-(*)$

(1.) Determine the fixed points of this system by hand.

Solution:- A fixed point is a point that satisfies the equation

$$\dot{x} = f(x) = 0$$

$$\Rightarrow \frac{dx}{dt} = A \cdot X = 0$$

$$\Rightarrow \begin{bmatrix} 0.1 & -0.3 \\ 0.2 & -0.3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\Rightarrow 0.1x_1 - 0.3x_2 = 0, \quad 0.2x_1 - 0.3x_2 = 0$$

ie, $0.1x_1 - 0.3x_2 = 0 \quad -(i)$

$0.2x_1 \cancel{+} 0.3x_2 = 0 \quad -(ii)$

Substract (ii) $-$ (i)

$$\Rightarrow 0.1x_1 = 0 \Rightarrow x_1 = 0$$

from (i) or (ii) put $x_1 = 0$

$$\Rightarrow x_2 = 0$$

$\therefore$ Critical or Fixed point $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Hence $(0,0)$ is the fixed point of $(*)$.

Answer.

4. $\frac{dx}{dt} = A \cdot X$, $A = \begin{pmatrix} 0.1 & -0.3 \\ 0.2 & -0.3 \end{pmatrix}$, $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

· Calculate Eigen values of the matrix.

The characteristic equation is given by

$\det(A - \lambda I) = 0$,

$\Rightarrow \begin{vmatrix} 0.1-\lambda & -0.3 \\ 0.2 & -0.3-\lambda \end{vmatrix} = 0$

$\Rightarrow (\lambda - 0.1)(\lambda + 0.3) + 0.3 \times 0.2 = 0$

$\Rightarrow \lambda^2 + 0.3\lambda - 0.1\lambda + 0.06 = 0$

$\Rightarrow \lambda^2 + 0.2\lambda + 0.06 = 0$

$\therefore \lambda = \frac{-0.2 \pm \sqrt{(0.2)^2 - 4 \times 0.06 \times 1}}{2}$

$= \frac{-0.2 \pm \sqrt{0.04 - 0.24}}{2}$

$= \frac{-0.2 \pm \sqrt{-0.20}}{2}$

$= -0.1 \pm \sqrt{-0.05}$

$= -0.1 \pm \sqrt{0.05}\, i$ = imaginary roots. (complex).

$\because$ Real part of roots are < 0.

Hence the fixed point (0,0) is a stable focus.

(0,0)

4. System parameter of my own choice :-

$$\dot{X} = AX, \quad \text{where } A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

characteristic equation:-

$$\det \begin{pmatrix} a-\lambda & b \\ c & d-\lambda \end{pmatrix} = 0$$

$$\Rightarrow (\lambda - a)(\lambda - d) - bc = 0$$

$$\Rightarrow \lambda^2 - (a+d)\lambda + ad - bc = 0$$

$$\therefore \lambda^2 - \tau\lambda + \Delta = 0$$

where $\tau = \text{trace}(A) = a+d$

$\Delta = \det(A) = ad - bc$

$$\lambda_1 = \frac{\tau + \sqrt{\tau^2 - 4\Delta}}{2}, \quad \lambda_2 = \tau - \sqrt{\tau^2 - 4\Delta}$$

Note:-

For the further analysis, please look the attached pdf of jupiter notebook.