Vol4_23.2

April 11, 2024

[ ]:
```
import numpy as np
import scipy.linalg as la
from scipy.optimize import fsolve
```

1 23.3

[ ]:
```
A = np.array([[1, 0, 0, 0, -1, 0],
              [-2, 1, 0, 0, 0, -1],
              [0, -2, 1, 0, 0, 0],
              [0, 0, 0, 1, -2, 0],
              [1, 0, 0, 0, 1, -2],
              [0, 1, 0, 0, 0, 1]])

b = np.array([6, 0, 0, -3, 0, 0])

soln = la.solve(A, b)
l = ['x1', 'x2', 'x3', 'p0', 'p1 = u0', 'p2 = u1']
for i, var in enumerate(l):
    print(f'{var} = {soln[i]}')

print(f'u3 = p4 = 0')
```

```
x1 = 1.5
x2 = 1.5
x3 = 3.0
p0 = -12.0
p1 = u0 = -4.5
p2 = u1 = -1.5
u3 = p4 = 0
```

2 23.4

[ ]:
```
B_list = [1, 10, 20]
beta = 0.002
d2 = 0.1
d = 0.2
```

```
for B in B_list:
    S1, I1, R1 = 100, 5, 0
    p14, p24 = 0, -1
    def func(x):
        S2, S3, S4 = x[0], x[1], x[2]
        I2, I3, I4 = x[3], x[4], x[5]
        R2, R3, R4 = x[6], x[7], x[8]
        p11, p12, p13 = x[9], x[10], x[11]
        p21, p22, p23 = x[12], x[13], x[14]
        u1 = min(1-d, max(1/(2*B)*(-p12*S1 + beta*p12*S1*I1 - beta*p22*S1*I1),␣
↪0))
        u2 = min(1-d, max(1/(2*B)*(-p13*S2 + beta*p13*S2*I2 - beta*p23*S2*I2),␣
↪0))
        u3 = min(1-d, max(1/(2*B)*(-p14*S3 + beta*p14*S3*I3 - beta*p24*S3*I3),␣
↪0))

        return np.array([
            S1*(1-u1) - beta*(S1*(1-u1))*I1 - S2,
            S2*(1-u2) - beta*(S2*(1-u2))*I2 - S3,
            S3*(1-u3) - beta*(S3*(1-u3))*I3 - S4,
            I1 + beta*(S1*(1-u1))*I1 - d2*I1 - I2,
            I2 + beta*(S2*(1-u2))*I2 - d2*I2 - I3,
            I3 + beta*(S3*(1-u3))*I3 - d2*I3 - I4,
            R1 + u1*S1 - R2,
            R2 + u2*S2 - R3,
            R3 + u3*S3 - R4,
            p12*(1-u1-beta*(1-u1)*I1) + beta*p22*(1-u1)*I1 - p11,
            p13*(1-u2-beta*(1-u2)*I2) + beta*p23*(1-u2)*I2 - p12,
            p14*(1-u3-beta*(1-u3)*I3) + beta*p24*(1-u3)*I3 - p13,
            -beta*p12*S1*(1-u1) + p22*(1+beta*(1-u1)-d2) - 1 - p21,
            -beta*p13*S2*(1-u2) + p23*(1+beta*(1-u2)-d2) - 1 - p22,
            -beta*p14*S3*(1-u3) + p24*(1+beta*(1-u3)-d2) - 1 - p23
        ])

    root = fsolve(func, x0=np.zeros(15), xtol=1e-10)
    S2, S3, S4 = root[0], root[1], root[2]
    I2, I3, I4 = root[3], root[4], root[5]
    R2, R3, R4 = root[6], root[7], root[8]
    p11, p12, p13 = root[9], root[10], root[11]
    p21, p22, p23 = root[12], root[13], root[14]

    u1 = min(1-d, max(1/(2*B)*(-p12*S1 + beta*p12*S1*I1 - beta*p22*S1*I1), 0))
    u2 = min(1-d, max(1/(2*B)*(-p13*S2 + beta*p13*S2*I2 - beta*p23*S2*I2), 0))
    u3 = min(1-d, max(1/(2*B)*(-p14*S3 + beta*p14*S3*I3 - beta*p24*S3*I3), 0))


    print(f'B = {B}')
```

```
    print(f'S2 = {round(root[0],2)}, S3 = {round(root[1],2)}, S4 =␣
↪{round(root[2],2)}')
    print(f'I2 = {round(root[3],2)}, I3 = {round(root[4],2)}, I4 =␣
↪{round(root[5],2)}')
    print(f'R2 = {round(root[6],2)}, R3 = {round(root[7],2)}, R4 =␣
↪{round(root[8],2)}')

    print(f'Optimal Control\nu1 = {round(u1,2)}, u2 = {round(u2,2)}, u3 =␣
↪{round(u3,2)}\n')
```

```
B = 1
S2 = 19.8, S3 = 14.57, S4 = 13.52
I2 = 4.7, I3 = 4.37, I4 = 4.05
R2 = 80.0, R3 = 85.09, R4 = 86.02
Optimal Control
u1 = 0.8, u2 = 0.26, u3 = 0.06

B = 10
S2 = 72.39, S3 = 63.81, S4 = 60.95
I2 = 5.23, I3 = 5.38, I4 = 5.51
R2 = 26.88, R3 = 34.79, R4 = 36.98
Optimal Control
u1 = 0.27, u2 = 0.11, u3 = 0.03

B = 20
S2 = 85.11, S3 = 78.58, S4 = 75.96
I2 = 5.36, I3 = 5.68, I4 = 5.98
R2 = 14.03, R3 = 19.71, R4 = 21.46
Optimal Control
u1 = 0.14, u2 = 0.07, u3 = 0.02
```

The model remains valid since we don't get any negative populations