Bifurcation_Point

May 27, 2020

1 Ananysis of the Bifurcation Point

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sympy import *
from tqdm import tqdm
from mpl_toolkits.mplot3d import Axes3D

init_printing()

x, t, d, Pe, Pm, v,  , N, v1, v0,  ,  , xd = symbols('x t d P_e P_m v delta N␣
↪v_1 v_0 rho, sigma \dot{x}')
```

We know that the dynamical equation is given as

[2]:
```
f = Pe*v1*(1-x) + Pm*v0*N*(1-x)*x/(1+Pm*N*(1-x)) - Pm*v1*N*x*(1-x)/(1+Pm*N*x)
Eq(xd,f)
```

[2]:

$$\dot{x} = \frac{NP_m v_0 x\left(-x+1\right)}{NP_m\left(-x+1\right)+1} - \frac{NP_m v_1 x\left(-x+1\right)}{NP_m x+1} + P_e v_1\left(-x+1\right)$$

Substituting v_0 and v_1 with v and $v+\delta$, we get

[3]:
```
F = f.subs([(v0, v+ ), (v1,v)])
Eq(xd,F)
```

[3]:

$$\dot{x} = -\frac{NP_m v x\left(-x+1\right)}{NP_m x+1} + \frac{NP_m x\left(\delta+v\right)\left(-x+1\right)}{NP_m\left(-x+1\right)+1} + P_e v\left(-x+1\right)$$

The fixed points of this system are obatined by solving

[4]:
```
Eq(F)
```

[4]:

$$-\frac{NP_m v x\left(-x+1\right)}{NP_m x+1} + \frac{NP_m x\left(\delta+v\right)\left(-x+1\right)}{NP_m\left(-x+1\right)+1} + P_e v\left(-x+1\right) = 0$$

It is easy to see that $(x-1)$ is common in the left hand side. **Hence $x = 1$ is the trivial fixed point of the system.** Dividing the equation by $(x-1)$, we get

[5]:
```
F1 = simplify(F/(x-1))
Eq(F1)
```

[5]:

$$\frac{NP_m vx\left(-NP_m\left(x-1\right)+1\right)-NP_m x\left(\delta+v\right)\left(NP_m x+1\right)-P_e v\left(NP_m x+1\right)\left(-NP_m\left(x-1\right)+1\right)}{\left(NP_m x+1\right)\left(-NP_m\left(x-1\right)+1\right)}=0$$

The equation would be much easier to analyze if we multiply by the denominator. This can be done as long as the denominator is not zero. We find the roots of the denominator to be

[6]:
```
solve(denom(F1),x)
```

[6]:

$$\left[-\frac{1}{NP_m},\quad 1+\frac{1}{NP_m}\right]$$

Both of these roots are outside the physically relevant range of $0 \leq x \leq 1$. Therefore, we can multiply by the denominator without any problem. This gives us

[7]:
```
F2 = F1*denom(F1)
Eq(F2)
```

[7]:

$$NP_m vx\left(-NP_m\left(x-1\right)+1\right)-NP_m x\left(\delta+v\right)\left(NP_m x+1\right)-P_e v\left(NP_m x+1\right)\left(-NP_m\left(x-1\right)+1\right)=0$$

The left hand side is a quadratic polynomial is x. Therefore the system has two additional roots. To find out when the roots are real, we collect the coefficients of the quadratic polynomial

[8]:
```
cf = Poly(F2,x).coeffs()
cf
```

[8]:

$$\left[N^2P_eP_m^2v-N^2P_m^2\delta-2N^2P_m^2v,\quad -N^2P_eP_m^2v+N^2P_m^2v-NP_m\delta,\quad -NP_eP_mv-P_ev\right]$$

The bifurcation occurs when the determinant is zero (because the number of real roots of the system changes from 1 to 3 as determinant goes from being negative to being positive). Hence the equation determining the bifurcation point is

[9]:
```
det = cf[1]**2-4*cf[0]*cf[2]
Eq(det)
```

[9]:

$$-\left(-NP_eP_mv-P_ev\right)\left(4N^2P_eP_m^2v-4N^2P_m^2\delta-8N^2P_m^2v\right)+\left(-N^2P_eP_m^2v+N^2P_m^2v-NP_m\delta\right)^2=0$$

The bifurcation points (say in terms of N) are, therefore,

[10]:
```
bp = solve(det,N)
bp
```

[10]:

$$\left[0,\quad -\frac{2\sqrt{-P_e\left(\delta+v\right)\left(P_e v-\delta-2v\right)}}{P_m v\left(P_e^2-2P_e+1\right)}-\frac{2P_e^2 v-P_e\delta-4P_e v-\delta}{P_m v\left(P_e-1\right)^2},\quad \frac{2\sqrt{-P_e\left(\delta+v\right)\left(P_e v-\delta-2v\right)}}{P_m v\left(P_e^2-2P_e+1\right)}-\frac{2P_e^2 v-P_e\delta -}{P_m v\left(P_e\right.}\right.$$

Let us substitute some typical values to find out which roots are valid

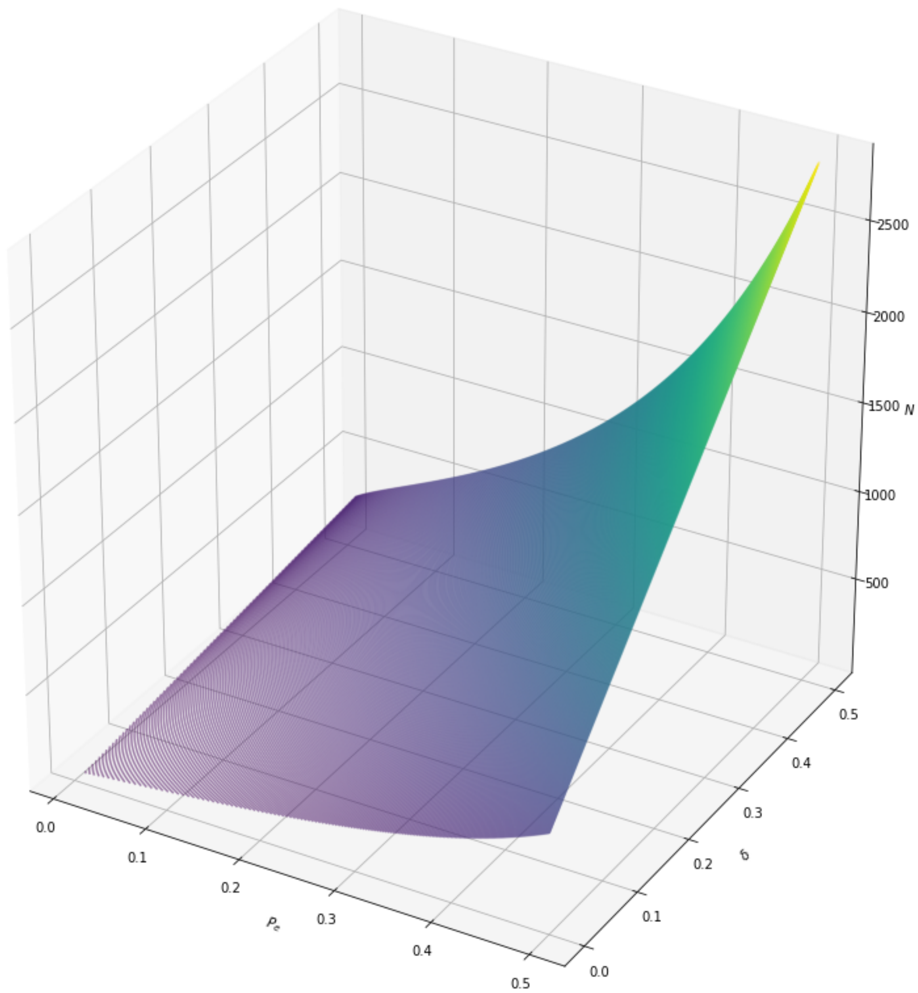
[11]:
```
[
    bp[0].subs([(Pe,0.12), (Pm,0.025), (v,0.7), ( ,0.1)]),
    bp[1].subs([(Pe,0.12), (Pm,0.025), (v,0.7), ( ,0.1)]),
    bp[2].subs([(Pe,0.12), (Pm,0.025), (v,0.7), ( ,0.1)]),
]
```

[11]:

$$[0,\quad -22.8416463839571,\quad 85.9821422517257]$$

We can see that the first and the second roots are invalid. Hence, the actual bifurcation point is the last one. Finally, we plot it as a function of P_e and δ

[12]:
```
fun = bp[2]

pp = fun.subs([(Pm,0.025),(v,0.1)])

pf = lambdify([Pe, ], pp)

X = np.linspace(0.01,0.5,200)
Y = np.linspace(0,0.5,200)

XX, YY = np.meshgrid(X,Y)
ZZ = pf(XX,YY)

fig = plt.figure(figsize=(12,12))
ax = plt.axes(projection='3d')
ax.contour3D(XX, YY, ZZ, 1000, alpha=0.5)
ax.set_xlabel('$P_e$')
ax.set_ylabel('$\delta$')
ax.set_zlabel('$N$')
plt.tight_layout()
plt.show()
```


0.0
0.1
0.2
0.3
0.4
0.5
P_e
0.0
0.1
0.2
0.3
0.4
0.5
δ
500
1000
1500
2000
2500
N

[ ]: