

Análise Quali

May 18, 2020

0.0.1 Sympy

Achar os pontos fixos, jacobiana e estabilidade dos pontos fixos.

$$\begin{aligned}\dot{y}_1 &= 2y_1 + y_1y_2 \\ \dot{y}_2 &= y_1y_2 - y_2^2\end{aligned}$$

então, temos

$$\begin{aligned}\dot{y}_1 &= f_1(y_1, y_2) \\ \dot{y}_2 &= f_2(y_1, y_2)\end{aligned}$$

```
[19]: from sympy import *
```

```
[20]: y1,y2 = symbols('y1 y2')
```

```
[21]: y2
```

```
[21]:  $y_2$ 
```

```
[22]: f1 = 2*y1 + y1*y2
```

```
[23]: f1
```

```
[23]:  $y_1y_2 + 2y_1$ 
```

```
[24]: f2 = y1*y2 - y2**2
```

```
[25]: f2
```

```
[25]:  $y_1y_2 - y_2^2$ 
```

• Ponto fixo:

```
[26]: pf1,pf2 = solve([f1,f2])
```

```
[27]: pf1
```

```
[27]: {y1: -2, y2: -2}
```

```
[28]: pf2
```

```
[28]: {y1: 0, y2: 0}
```

- Matriz Jacobiana:

```
[31]: S = Matrix([f1,f2]); S
```

```
[31]: 
$$\begin{bmatrix} y_1 y_2 + 2y_1 \\ y_1 y_2 - y_2^2 \end{bmatrix}$$

```

```
[32]: JS = S.jacobian([y1,y2])
```

```
[33]: JS
```

```
[33]: 
$$\begin{bmatrix} y_2 + 2 & y_1 \\ y_2 & y_1 - 2y_2 \end{bmatrix}$$

```

- Avaliar os pontos fixos:

```
[34]: pf1
```

```
[34]: {y1: -2, y2: -2}
```

```
[35]: JSpf1 = JS.subs([(y1,-2),(y2,-2)]); JSpf1
```

```
[35]: 
$$\begin{bmatrix} 0 & -2 \\ -2 & 2 \end{bmatrix}$$

```

```
[36]: pf2
```

```
[36]: {y1: 0, y2: 0}
```

```
[37]: JSpf2 = JS.subs([(y1,0),(y2,0)]); JSpf2
```

```
[37]: 
$$\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

```

- Calcular os autovalores:

```
[38]: JSpf1
```

```
[38]: 
$$\begin{bmatrix} 0 & -2 \\ -2 & 2 \end{bmatrix}$$

```

```
[39]: JSpf1.eigenvals()
```

```
[39]: {1 - sqrt(5): 1, 1 + sqrt(5): 1}
```

```
[40]: pf2
```

```
[40]: {y1: 0, y2: 0}
```

```
[41]: JSpf2
```

```
[41]:  $\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$ 
```

```
[42]: JSpf2.eigenvals()
```

```
[42]: {2: 1, 0: 1}
```

```
[47]: import numpy as np
```

```
[49]: JSpf1
```

```
[49]:  $\begin{bmatrix} 0 & -2 \\ -2 & 2 \end{bmatrix}$ 
```

```
[67]: JSpf1arr = np.array(JSpf1, dtype=int)
```

```
[68]: JSpf1arr
```

```
[68]: array([[ 0, -2],  
         [-2,  2]])
```

```
[75]: eigval, eigvec = np.linalg.eig(JSpf1arr)
```

```
[76]: eigval
```

```
[76]: array([-1.23606798,  3.23606798])
```

```
[ ]:
```