



UNIVERSIDAD  
SERGIO ARBOLEDA

MATEMÁTICAS

ECUACIONES DIFERENCIALES ORDINARIAS

---

## Taller 7

---

*Alexander Mendoza*

*Alejandro Garzón*

17 de noviembre de 2024

```

1      import numpy as np
2      import matplotlib.pyplot as plt
3
4
5      def sistema_depredador_presa(Y):
6          x, y = Y
7          return np.array([2*x - 1.2*x*y, -y + 1.2*x*y])
8
9
10     def resolver_sistema(Y0, dt, nt):
11         Yt = np.zeros((2, nt+1))
12         tt = np.zeros(nt+1)
13         Yt[:, 0] = Y0
14         t = 0
15         Y = Y0.copy()
16         for i in range(nt):
17             Ym = Y + dt * sistema_depredador_presa(Y)
18             Y = Y + dt / 2 * (sistema_depredador_presa(Y) +
19                             sistema_depredador_presa(Ym))
19             t += dt
20             Yt[:, i+1] = Y
21             tt[i+1] = t
22         return Yt, tt
23
24
25     def funcion_lyapunov(x, y):
26         return -1.2*x - np.log(x) + 1.2*y - 2*np.log(y)
27
28
29     def grafica_evolucion_tiempo(tt, x, y):
30         plt.figure(figsize=(8, 6))
31         plt.plot(tt, x, linestyle='--', color='tab:red', label=
32                 'x(t)', linewidth=2)
33         plt.plot(tt, y, linestyle='-.', color='tab:blue', label=
34                 'y(t)', linewidth=2)
35         plt.legend(fontsize=12)
36         plt.xlabel('$t$', fontsize=14)
37         plt.ylabel('$x, y$', fontsize=14)
38         plt.title('Grafica x(t) y(t)', fontsize=16)
39         plt.grid(True, linestyle=':', color='gray', alpha=0.7)
40         # More subtle grid lines
41         plt.tight_layout()
42
43
44     def grafica_plano(x, y):
45         plt.figure(figsize=(8, 6))
46         plt.plot(x, y, linestyle='-', color='darkviolet',
47                 linewidth=2)
48         plt.axis('equal')
49         plt.xlabel('$x$', fontsize=14)
50         plt.ylabel('$y$', fontsize=14)
51         plt.title('Orbita', fontsize=16)
52         plt.grid(True, linestyle=':', color='gray', alpha=0.7)
53         plt.tight_layout()
54
55     def grafica_plano_curva(x, y):

```

```

53     xgrid = np.linspace(0.5, 3, 100)
54     ygrid = np.linspace(0.5, 3, 100)
55     X, Y = np.meshgrid(xgrid, ygrid)
56     Z = funcion_lyapunov(X, Y)
57
58     plt.figure(figsize=(8, 6))
59     cp = plt.contour(X, Y, Z, levels=20, cmap='viridis')
60     plt.plot(x, y, 'r--', linewidth=2)
61     plt.colorbar(cp)
62     plt.axis('equal')
63     plt.xlabel('$x$', fontsize=14)
64     plt.ylabel('$y$', fontsize=14)
65     plt.title('Orbita con curva', fontsize=16)
66     plt.grid(True, linestyle=':', color='gray', alpha=0.7)
67     plt.legend(fontsize=12)
68     plt.tight_layout()
69
70
71
72     Y0 = np.array([1.75, 1.0])
73     dt = 0.01
74     nt = 1000
75
76     Yt, tt = resolver_sistema(Y0, dt, nt)
77     x, y = Yt[0, :], Yt[1, :]
78
79     grafica_evolucion_tiempo(tt, x, y)
80     grafica_plano(x, y)
81     grafica_plano_curva(x, y)
82
83     plt.show()

```



