Computación científica LF-214

Profesor: Julio C. Marín

Departamento de Meteorología, Universidad de Valparaíso

Primer Semestre

Temas de la clase

- ODEs de más de una variable
- Ejemplos y Ejercicios

ODEs de más de una variable

- En muchos problemas físicos tenemos más de una variable. Por ejemplo:

$$\frac{dx}{dt} = xy - x, \qquad \frac{dy}{dt} = y - xy + \sin^2(\omega t)$$

- Derivada de cada variable (x, y) depende de las 2 variables (x, y). Hay una sola variable independiente t.
- La forma general para 2 ODEs de 1er orden es:

$$\frac{dx}{dt} = f_x(x, y, t), \qquad \frac{dy}{dt} = f_y(x, y, t)$$

donde f_x y f_y pueden ser funciones no lineales de x, y y t.

ODEs de más de una variable

- Usando notación vectorial, ecs. para un número arbitrario de variables queda:

$$\frac{d\mathbf{r}}{dt} = \mathbf{f}(\mathbf{r}, t)$$

donde $r = (x, y, \ldots)$ y $f(r, t) = (f_x(x, y, t), f_y(x, y, t), \ldots)$

- Resolver ODEs simultáneas de manera analítica es más difícil en general
- PERO resolverlas numéricamente no implica mucha mayor dificultad que el caso de una variable

ODEs de más de una variable. Ejemplo

Ejemplo 8.5: Calcular solución de ecs:

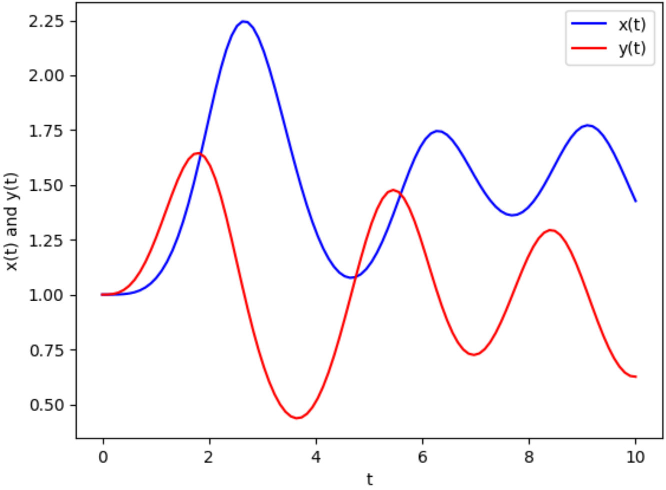
$$\frac{dx}{dt} = xy - x, \qquad \frac{dy}{dt} = y - xy + \sin^2(\omega t)$$

desde $t = 0$ hasta $t = 10$ para el caso $\omega = 1$ con la condición inicial $x = y = 1$ en $t = 0$

ODEs de más de una variable. Ejemplo

- import numpy as np
- from scipy.integrate import odeint
- import matplotlib.pyplot as plt
- def model(r,t):
-     x = r[0]; y = r[1]
-     fx = x*y - x; fy = y - x*y + np.sin(t)**2
-     return np.array([fx,fy], float)
- r0 = [1.0, 1.0] # Condición inicial
- t = np.linspace(0,10,100)
- rp = odeint(model, r0, t) # Resolver ODE
- plt.clf()
- plt.plot(t, rp[:,0], '-b', t, rp[:,1], '-r')
- plt.ylabel('x(t) and y(t)'); plt.xlabel('t')
- plt.legend(('x(t)','y(t)'), loc=1)

ODEs de más de una variable. Ejemplo



Ejercicio: Resolver el sistema de ecuaciones:

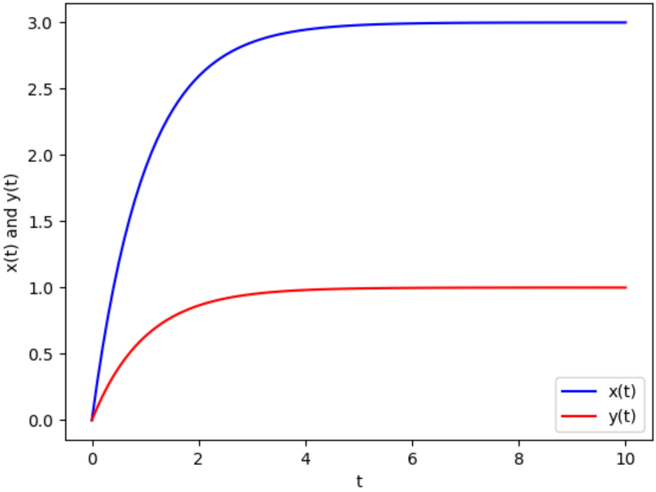
$$\frac{dx}{dt} = 3e^{-t}, \qquad \frac{dy}{dt} = 1 - y$$

desde $t = 0$ hasta $t = 10$ con la condición inicial $x = y = 0$ en $t = 0$

ODEs de más de una variable. Ejercicio

- import numpy as np
- from scipy.integrate import odeint
- import matplotlib.pyplot as plt
- def model(r,t):
-     x = r[0]; y = r[1]
-     fx = 3*np.exp(-t)
-     fy = 1 - y
-     return np.array([fx,fy], float)
- r0 = [0, 0] # Condición inicial
- t = np.linspace(0,10,100)
- rp = odeint(model, r0, t) # Resolver ODE
- plt.clf()
- plt.plot(t, rp[:,0], '-b', t, rp[:,1], '-r')
- plt.ylabel('x(t) and y(t)'); plt.xlabel('t')
- plt.legend(('x(t)','y(t)'), loc=4)

ODEs de más de una variable. Ejemplo



Ejercicio 8.2: Ecuaciones de Lotka-Volterra

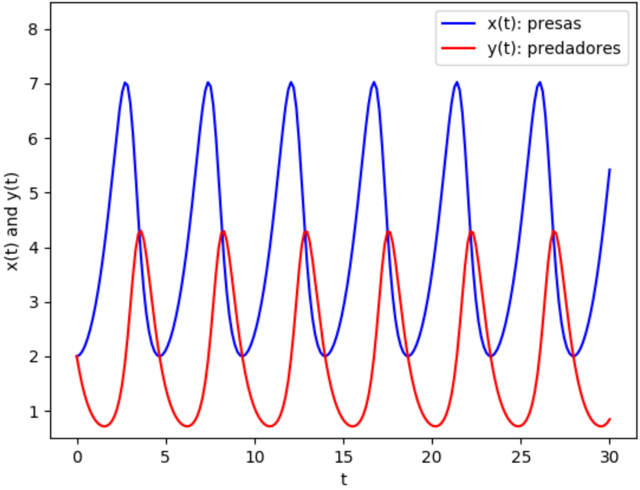
En el modelo de Lotka-Volterra, los conejos (presas) se reproducen a una tasa proporcional a su población, pero los zorros (predadores) se los comen a una tasa proporcional a la población de los zorros y la de los conejos: $dx/dt = \alpha x - \beta xy$. α y β son constantes. Al mismo tiempo, los zorros se reproducen a una tasa proporcional a la tasa a la que comen los conejos, pero también mueren de vejez a una tasa proporcional a su población: $dy/dt = \gamma xy - \delta y$. γ y δ son constantes.

a) Escriba un programa que resuelva las ecuaciones para el caso $\alpha = 1$, $\beta = \gamma = 0.5$ y $\delta = 2$, comenzando con la condición inicial $x = y = 2$. Haga un gráfico que muestre ambas curvas $x(t)$ y $y(t)$ en función de t desde $t = 0$ a $t = 30$.

Resolución de sists. de ODEs. Ejercicio

- import numpy as np
- from scipy.integrate import odeint
- import matplotlib.pyplot as plt
- def model(r,t):
-     x = r[0]; y = r[1]
-     fx = x - 0.5*x*y
-     fy = 0.5*x*y - 2*y
-     return np.array([fx,fy], float)
- r0 = [2, 2] # Condición inicial
- t = np.linspace(0,30,200)
- rp = odeint(model, r0, t) # Resolver ODE
- plt.clf(); plt.plot(t, rp[:,0], '-b', t, rp[:,1], '-r')
- plt.ylabel('x(t) and y(t)'); plt.xlabel('t')
- plt.legend(('x(t): presas','y(t): predadores'), loc=1)
- plt.ylim(0.5,8.5)

Resolución de sists. de ODEs. Ejercicio



Ejercicio: Resolver el sistema de ecuaciones:

$$(2-x)\frac{dx}{dt} = -x + u, \qquad (y-1)\frac{dy}{dt} = -y + x$$

desde $t = 0$ hasta $t = 15$ con la condición inicial $x = y = 0$ en $t = 0$. El parámetro $u = 0$ para $t < 5$ y $u = 2$ para $t \geq 5$

Resolución de sists. de ODEs. Ejercicio

```
def model(r,t):
    if t < 5:
        u = 0
    else:
        u = 2
    x = r[0]; y = r[1]
    fx = (-x + u)/(2-x)
    fy = (-y + x)/(y-1)
    return np.array([fx,fy], float)
r0 = [0, 0] # Condición inicial
t = np.linspace(0, 15, 100)
rp = odeint(model, r0, t) # Resolver ODE
plt.clf(); plt.plot(t, rp[:,0], '-b', t, rp[:,1], '-r')
plt.ylabel('x(t), y(t)'); plt.xlabel('t')
plt.legend(('x(t)', 'y(t)'), loc=2)
```

Resolución de sists. de ODEs. Ejercicio