Elementos de Cálculo Numérico / Cálculo Numérico

Primer Cuatrimestre 2020

Cuarto ejercicio computacional
Lunes 04/04/20 al Lunes 11/04/20
Recuerde subir el archivo en formato ejercicioX_NOMBREAPELLIDO.py
Recuerde al hacer consultas enviar su código

Podrá encontrar un ejemplo de resolución de un sistema de ecuaciones ordinarias en varias dimensiones en los videos de la parte computacional.

Esta semana simularemos un modelo SIR. Un modelo SIR (Suceptible - Infectado - Recuperado) consta de tres variables S, I, R, que dan cuenta del número de personas suceptibles, infectadas y recuperadas de una enfermedad. Es uno de los modelos más básicos que se emplean en la actualidad para dar cuenta del desarrollo y expansión de enfermedades.

El modelo consta de tres ecuaciones:

$$S' = -\beta \frac{SI}{N}$$
$$I' = \beta \frac{SI}{N} - \gamma I$$
$$R' = \gamma I$$

donde β^{-1} es el tiempo medio entre contactos del par suceptible - infectado, γ^{-1} el tiempo medio de recuperación, y N la población total. Noten que $(S + I + R)' = 0$ y por lo tanto $S + I + R = N$ es constante (no cambia la población).

Para simular este modelo, emplearemos Runge-Kutta de orden 4, y consideraremos un vector $V = (S, I, R)$ de forma que

$$V' = F(V) \tag{1}$$

Considere los siguientes pasos para construir el modelo:

a Construya una función `V_prima(coefs,V)` que reciba una lista `coefs=[beta,gamma,N]`, con los valores de β y γ y el tamaño de población `N`, y un np.array `V`, de forma que `V[0]` sea S, `V[1]` sea I y `V[2]` sea R, y retorne un array `np.array([S_prima,I_prima,R_prima])`, donde cada componente contiene el valor de la derivada en calculados en función de `V` y `coefs`

b Construya una funcion `paso_runge_kutta(coefs,V,dt,t)` siguiendo el siguiente modelo:

```
def paso_runge_kutta(coefs,V,dt,t):
    k1 = V_prima(coefs,V,t)
    k2 = # COMPLETAR
    k3 = # COMPLETAR
```

```
        k4 = # COMPLETAR
        V = V + # COMPLETAR
        t = t+dt
        return([t,V])
```

(si no tiene idea de como completar esto, revise la teórica, clase 4)

c Una función `integra_runge_kutta(coefs,V0,t0,T,dt)` que reciba los coeficientes, la condicion inicial, el paso temporal y el tiempo máximo de integración, y retorne dos listas, una con los tiempos, y otra con los vectores `V`:

```
def integra_runge_kutta(coefs,V0,t0,T,dt):
    t = [t0]
    V = [V0]
    while t[-1]<T:
        #COMPLETAR
    return([t,V])
```

d Para recuperar los valores de S, I, y R, aquí una ayudita más:

```
def despliega_integracion(integracion):
    t = integracion[0]
    V = integracion[1]
    S = []
    I = []
    R = []
    for i in range(len(V)):
        S.append(V[i][0])
        I.append(V[i][1])
        R.append(V[i][2])
    return([np.array(t),np.array(S),np.array(I),np.array(R)])
```

Realice la simulación considerando los valores de $\beta = 1.3$, $\gamma = 1$, `t0=0`, `T=1000`,`dt=0.15`,`N=10**4` `S0=N*0.99`,`I0=N*0.01`,`R0=0`. Grafique las tres variables en función del tiempo, así como la suma `S+I+R` ¿Se mantiene constante en el tiempo? Explore el efecto de cambiar el valor de β ¿Cómo cambia el valor número máximo de infectados, así como el momento en el que se alcanza, para distintos valores de β?¿Qué pasa si $\beta/\gamma < 1$? ¿Siempre se llega a infectar a toda la población?

Por último, pruebe el efecto de agregar la posibilidad de que los recuperados se vuelvan suceptibles nuevamente, a través de un coeficiente α.

$$
\begin{aligned}
S' &= -\beta\frac{SI}{N} + \alpha R \\
I' &= \beta\frac{SI}{N} - \gamma I \\
R' &= \gamma I - \alpha R
\end{aligned}
$$

Incorpore α a los valores en `coefs`. ¿Qué efecto tiene en la evolución?