Universidad del Estado de Sonora

División de Ciencias Exactas y Naturales.

Licenciatura en Física.

Física Computacional 1

Hedwin Aaron Encinas Acosta

1 Introduccion

En el estudio científico es muy común encontrarse modelos que describan el comportamiento de diverso tipos de fenómenos. En la ciencia esto es una herramienta muy útil pues nos permite estudio de sistemas que, sin esta importante herramienta seria muy difícil, si no es que imposible, comprender sistemas tan complejos como lo son los campos magnéticos o el flujo de calor. Pero demos una definición mas clara de modelo.

1.1 Modelos científico

Un modelo científico se define como una representación abstracta, conceptual, gráfica o visual , física, de fenómenos, sistemas o procesos a fin de analizar, describir, explicar esos fenómenos o procesos. Un modelo permite determinar un resultado final a partir de unos datos de entrada. Se considera que la creación de un modelo es una parte esencial de toda actividad científica. Para hacer un modelo es necesario plantear una serie de hipótesis, de manera que lo que se quiere estudiar esté suficientemente plasmado en la representación, aunque también se busca, normalmente, que sea lo bastante sencillo como para poder ser manipulado y estudiado.[1]

2 Actividad

En esta actividad se modificara un código de ejemplo de la pagina *SciPy CookBook*[2], el cual modela lo que sucedería en el caso de que un virus zombie se esparciera por la población. Se darán diferentes escenarios iniciales y se vera como se comporta la población de zombies y humanos durante un periodo de tiempo.Cabe menciona que el programa esta escrito en el lenguaje de programación python.

3 Parámetros

Debido a que en esta actividad se vera como se desarrolla una epidemia zombie contemplando diferentes iniciales que se presentaran mas adelante, definiremos cual es el significado de cada parámetro que se usara en las ecuaciones para que todo quede mas claro:

- **S:** Población susceptible a infectarse o morir.
- **Z:** Población Zombie
- **R:** Población de individuos removidos o eliminados.
- **I:** Población de infectados pero no zombies.
- **Q:** Población en cuarentena.
- $\prod$: Tasa de natalidad.
- δ: Tasa de muertes naturales.
- β: Tasa de transmisión.
- ζ: Tasa de resucitación.
- α: Tasa de destrucción **Z**.
- ρ: Tasa de conversión de **I** a **Z**.
- κ: Tasa de ingreso de **I** a **Q**.
- σ: Tasa de ingreso de **Z** a **Q**.
- γ: Tasa de muertes en **Q**.
- c: Tasa de cura

4 Modelo basico de Zombis

En este modelo es el mas simple, para lo cual se asuma que os zombies solo pueden venir de dos formas:

- De los individuos que han muerto recientemente.
- Todos aquellos que han perdido un encuentro con un

El sistema de ecuaciones para este caso es el siguiente:

$$
\begin{aligned}
S' &= \Pi - \beta S Z - \delta S \\
Z' &= \beta S Z + \zeta R - \alpha S Z \\
R' &= \delta S + \alpha S Z - \zeta R\,.
\end{aligned}
$$

4.1 Código

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0         # Nacimientos Diarios
D = 0.0001   # Muertes Naturales %  (Por dia)
B = 0.0095   # Transmision       %  (Por dia)
G = 0.0001   # Removidos         %  (Por dia)
A = 0.005    # Destruidos        %  (Por dia)

#Sistema de Ecuaciones Diferenciales
def f(y, t):
    Si = y[0]
    Zi = y[1]
    Ri = y[2]
    # Modelo
    f0 = P - B*Si*Zi - D*Si              #Si
    f1 = B*Si*Zi + G*Ri - A*Si*Zi       #Zi
    f2 = D*Si + A*Si*Zi - Z*Ri          #Ri
    return [f0, f1, f2]

S0 = 500.                          # Poblacion Inicial
Z0 = 0                             # Zombie Inicial
R0 = 0                             # Muertos Inicial
y0 = [S0, Z0, R0]                  # Condicion Inicial
t  = np.linspace(0, 14., 1000)    #Tiempo
```

```
# Solucion E.D
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]
# Grafica
plt.figure()
plt.ylim(0,600)
plt.grid(True)
plt.plot(t, S, label='Vivos')
plt.plot(t, Z, label='Zombies')
plt.xlabel('Tiempo')
plt.ylabel('Poblacion')
plt.title('Apocalipsis Zombie - Modelo Basico')
plt.legend(loc="best")
```

5 Modelo con Infeccion Latente

En el caso de infección latente se hacen algunos cambios a las condiciones descritas anteriormente:

- Los susceptibles primero se mueven a la clase de infectados, una ves así permanecen de ese modo por un tiempo.
- Los individuos infectados aun pueden morir por causas naturales, de no ser así se convierten en zombies.

El sistema de ecuaciones para este caso es el siguiente:

$$\begin{aligned} S' &= \Pi - \beta S Z - \delta S \\ I' &= \beta S Z - \rho I - \delta I \\ Z' &= \rho I + \zeta R - \alpha S Z \\ R' &= \delta S + \delta I + \alpha S Z - \zeta R \end{aligned}$$

5.1 Código

```
#modelo de infeccion latente
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8


P = 0        # Nacimientos
D = 0.0001  # Muertes Naturales
B = 0.0095  # Transmision
G = 0.0001  # Removidos
A = 0.005   # Destruidos
o = 0.0001  #infectados
# solve the system dy/dt = f(y, t)
def f(y, t):
    Si = y[0]
    Zi = y[1]
    Ri = y[3]
    Ii = y[2]
    # the model equations (see Munz et al. 2009)
    f0 = P - B*Si*Zi - d*Si
    f1 = B*Si*Zi + G*Ri - A*Si*Zi
    f3 = B*Si*Zi-o-d*Ii
    f2 = d*Si+d*Ii+ A*Si*Zi - G*Ri

    return [f0, f1, f2,f3]

# initial conditions
S0 = 500.                        # Poblacion Inicial
Z0 = 0                           # Zombie Inicial
R0 = 0                           # Muertos Inicial
y0 = [S0, Z0, R0]                # Condicion Inicial
t  = np.linspace(0, 14., 1000)   #Tiempo

# solucion
```

```
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]
I = soln[:, 3]
# plot
plt.figure()
plt.ylim(0,600)
plt.grid(True)
plt.plot(t, S, label='Vivos')
plt.plot(t, Z, label='Zombies')
plt.xlabel('Tiempo')
plt.ylabel('Poblacion')
plt.title('Apocalipsis Zombie - Infeccion latente')
plt.legend(loc="best")
```

6 Modelo con Cuarentena

Para este escenario se decidió poner en cuarentena a los infectados, por lo tanto son aislados de la población general y por lo tanto no pueden infectar a otro individuos. Para esto es necesario definir las siguientes condiciones:

- El área de cuarentena solo contiene a miembros infectados o zombies.
- Existe la posibilidad de escape, pero cualquier intento resultara en la muerte del sujeto.
- Los sujetos que intentaron escapar y fueron eliminados pueden revivir como zombies.

El sistema de ecuaciones para este caso es el siguiente:

$$
\begin{aligned}
S' &= \Pi - \beta S Z - \delta S \\
I' &= \beta S Z - \rho I - \delta I - \kappa I \\
Z' &= \rho I + \zeta R - \alpha S Z - \sigma Z \\
R' &= \delta S + \delta I + \alpha S Z - \zeta R + \gamma Q \\
Q' &= \kappa I + \sigma Z - \gamma Q\,.
\end{aligned}
$$

6.1 Código

```
#modelo con cuerentena
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0          # Nacimientos
D = 0.0001    # Muertes Naturales
B = 0.0095    # Transmision
G = 0.0001    # Removidos
A = 0.0001    # Destruidos
o = 0.05      # Infected
K = 0.15      # Infectados
S = 0.10      # Infected
M = 0.001     # Infected

#Sistema de Ecuaciones
def f(y, t):
    Si = y[0]
    Zi = y[1]
    Ri = y[2]
    Ii = y[3]
    Qi = y[4]
    # Modelo
```

```
    f0 = P - *Si*Zi - D*Si
    f1 = R*Ii + G*Ri - A*Si*Zi - S*Zi
    f2 = Del*Si + Del*Ii + Alf*Si*Zi - G*Ri + M*Qi
    f3 = B*Si*Zi -oIi - D*Ii - K*Ii
    f4 = K*Ii + S*Zi - M*Qi
    return [f0, f1, f2, f3, f4]

S0 = 500.                        # Poblacion Inicial
Z0 = 0.                          # Zombie Inicial
R0 = 0.                          # Muertos Inicial
I0 = 1.                          # Infectados Inicial
Q0 = 0.                            # Cuarentena Inicial
y0 = [S0, Z0, R0, I0, Q0]        # Condiciones Iniciales
t  = np.linspace(0., 30., 1000)  # Tiempo

# Solucion
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]
I = soln[:, 3]
Q = soln[:, 4]
# plot
plt.figure()
plt.ylim(0,600)
plt.grid(True)
plt.plot(t, S, label='Vivos')
plt.plot(t, Z, label='Zombies')
plt.xlabel('Tiempo')
plt.ylabel('Poblacion')
plt.title('Apocalipsis Zombie - Modelo Caurentena.')
plt.legend(loc="best")
```

7 Modelo con Tratamiento

Ahora se toma en consideración le creación de una cura, la cual puede curar a cualquier zombie y volverlo humano, por lo tanto se elimina la necesidad de una cuarentena. Las nuevas consideraciones son:

- La cura puede curar a cualquier zombie y volverlo humano sin importar cual fue la causa de su transformación en zombie.
- La cura no da inmunidad al virus al sujeto, por lo cual puede volver a convertirse en zombie.

El sistema de ecuaciones para este caso es el siguiente:

$$
\begin{aligned}
S' &= \Pi - \beta S Z - \delta S + c Z \\
I' &= \beta S Z - \rho I - \delta I \\
Z' &= \rho I + \zeta R - \alpha S Z - c Z \\
R' &= \delta S + \delta I + \alpha S Z - \zeta R \,.
\end{aligned}
$$

7.1 Código

```
#modelo con cura
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

  = 0        # Nacimientos
D = 0.0001   # Muertes Naturales
B = 0.0095   # Transmision
G = 0.0001   # Removidos
A = 0.0001   # Destruidos
o = 0.05     # Infected
C  = 0.05     # Cura

#Sistema de Ecuaciones
def f(y, t):
    Si = y[0]
    Zi = y[1]
```

```
    Ri = y[2]
    Ii = y[3]
    # Modelo
    f0 = P - B*Si*Zi - D*Si +C*Zi
    f1 = Ii + G*Ri - Alf*Si*Zi -C*Zi
    f2 = D*Si + D*Ii + A*Si*Zi - G*Ri
    f3 = B*Si*Zi -o*Ii - D*Ii

    return [f0, f1, f2, f3]

S0 = 500.                         # Poblacion Inicial
Z0 = 0.                           # Zombie Inicial
R0 = 0.                           # Muertos Inicial
I0 = 1.                           # Infectados Inicial
y0 = [S0, Z0, R0, I0]             # Condiciones Iniciales
t  = np.linspace(0., 30., 1000)  # Tiempo

# Solucion
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]
I = soln[:, 3]
# plot
plt.figure()
plt.ylim(0,500)
plt.grid(True)
plt.plot(t, S, label='Vivos')
plt.plot(t, Z, label='Zombies')
plt.xlabel('Tiempo')
plt.ylabel('Poblacion')
plt.title('Apocalipsis Zombie - Modelo Tratamiento.')
plt.legend(loc="best")
```

Referencias

[1] Wikipedia, *Modelo científico*
https://es.wikipedia.org

[2] SciPy CookBook
http://scipy-cookbook.readthedocs.org/index.html