

EID Parte 2

Vicente Rivera — Matias Vasquez — Juan Sepúlveda

I. MODELO SIR LINEAL PARA VIRUS INFORMÁTICOS

Sea N el número total de equipos de la red. Del contexto de la parte anterior se tiene $N = 1000$ equipos. La población inicial de infectados es $I_0 = 1$ equipo, por lo que la población inicial de susceptibles es $S_0 = 999$ equipos y $R_0 = 0$.

Además, se sabe que inicialmente se observan aproximadamente 3 infecciones por hora en la red completa. En el modelo lineal, las nuevas infecciones por hora vienen dadas por $\alpha S(t)$, de modo que al inicio

$$\alpha S_0 \approx 3 \Rightarrow \alpha = \frac{3}{1000} = 0.003 \text{ h}^{-1}.$$

Por otro lado, se propone una tasa de recuperación

$$\beta = 0.05 \text{ h}^{-1},$$

El valor de β corresponde a un tiempo medio de recuperación de $1/\beta \approx 20$ horas, adecuado para entornos corporativos con respuesta activa.

A. Modelo A

- $S(t)$: número de computadores susceptibles,
- $I(t)$: número de computadores infectados.

El modelo lineal simple sin recuperación es

$$\begin{cases} S'(t) = -\alpha S(t), \\ I'(t) = \alpha S(t), \end{cases} \quad (1)$$

con condiciones iniciales $S(0) = S_0 = 999$, $I(0) = I_0 = 1$.

1) Ecuación $S(t)$:

$$S'(t) = -\alpha S(t) \Rightarrow S(t) = S_0 e^{-\alpha t}.$$

2) Ecuación $I(t)$:

$$I'(t) = \alpha S(t) \Rightarrow I(t) = I_0 + S_0 (1 - e^{-\alpha t}).$$

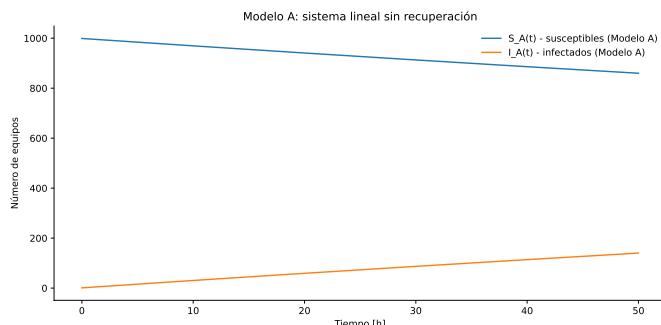


Fig. 1. Modelo A: sistema lineal sin recuperación

B. Modelo B

Se propone ahora un modelo lineal con recuperación:

- $S(t)$: número de computadores susceptibles,
- $I(t)$: número de computadores infectados,
- $R(t)$: número de computadores recuperados.

El sistema es

$$\begin{cases} S'(t) = -\alpha S(t), \\ I'(t) = \alpha S(t) - \beta I(t), \\ R'(t) = \beta I(t), \end{cases} \quad (2)$$

con $S(0) = 999$, $I(0) = 1$, $R(0) = 0$.

Como antes, se tiene

$$S(t) = S_0 e^{-\alpha t}.$$

1) Ecuación $I(t)$: Partiendo de

$$I'(t) + \beta I(t) = \alpha S_0 e^{-\alpha t},$$

se obtiene la solución

$$I(t) = I_0 e^{-\beta t} + \frac{\alpha S_0}{\beta - \alpha} (e^{-\alpha t} - e^{-\beta t}).$$

2) Ecuación $R(t)$: Como el total se conserva,

$$N = S(t) + I(t) + R(t),$$

se tiene

$$R(t) = N - S(t) - I(t).$$

Sustituyendo las soluciones anteriores:

$$R(t) = N - S_0 e^{-\alpha t} - I_0 e^{-\beta t} - \frac{\alpha S_0}{\beta - \alpha} (e^{-\alpha t} - e^{-\beta t}).$$

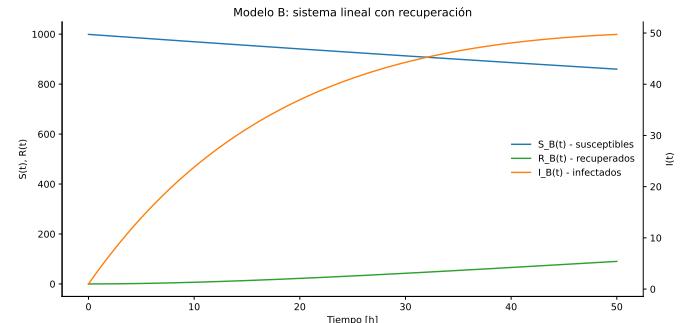


Fig. 2. Modelo B: sistema lineal con recuperación (eje secundario para I(t))

II. MODELO SIR NO LINEAL PARA VIRUS INFORMÁTICOS

A. Modelo

En esta sección modelamos la propagación del troyano utilizando el modelo clásico SIR no lineal, adaptado a redes de computadores. Dividimos la red en tres compartimentos:

- $S(t)$: número de computadores susceptibles (vulnerables pero no infectados).
- $I(t)$: número de computadores infectados que pueden propagar el malware.
- $R(t)$: número de computadores recuperados (limpiados, parcheados o aislados).

El sistema de ecuaciones diferenciales no lineal es:

$$\begin{cases} S'(t) = -\alpha S(t)I(t), \\ I'(t) = \alpha S(t)I(t) - \beta I(t), \\ R'(t) = \beta I(t), \end{cases} \quad (3)$$

donde los parámetros tienen la siguiente interpretación:

- $\alpha > 0$: tasa de infección, que mide cuántos contactos efectivos susceptible–infectado generan una nueva infección por unidad de tiempo.
- $\beta > 0$: tasa de recuperación, que mide la probabilidad de que un computador infectado sea detectado, limpiado o aislado por unidad de tiempo.

Se cumple que la población total permanece aproximadamente constante:

$$N = S(t) + I(t) + R(t) \approx 1000.$$

Los parámetros adoptados en este estudio son:

$$N = 1000, \quad S(0) = 999, \quad I(0) = 1, \quad R(0) = 0,$$

$$\alpha = 0.003 \text{ h}^{-1}, \quad \beta = 0.25 \text{ h}^{-1}.$$

h^{-1} : Tasa, cantidad que ocurre en una hora.

El valor de α se eligió de forma que la tasa inicial de contagio $\alpha S(0)I(0) \approx 3$ infecciones/hora sea coherente con el modelo lineal usado en la Parte 1. El valor de β corresponde a un tiempo medio de recuperación de $1/\beta \approx 20$ horas, adecuado para entornos corporativos con respuesta activa.

B. No linealidad y método de resolución

El sistema SIR es no lineal debido al término $\alpha S(t)I(t)$, que representa la interacción simultánea entre susceptibles e infectados. La presencia de este producto de variables impide aplicar métodos lineales de superposición y hace que no exista una solución cerrada simple para (S, I, R) . Así, la dinámica no puede expresarse como combinación lineal de soluciones particulares, y requiere el uso de métodos numéricos para aproximar la trayectoria temporal del sistema.

Para resolver este sistema se utilizó un integrador numérico basado en métodos de Runge–Kutta adaptativos. En particular, la función `solve_ivp` de la biblioteca `scipy.integrate` implementa el método Dormand–Prince (RK45), que estima

su propio error en cada paso comparando dos aproximaciones de distinto orden y ajusta automáticamente el tamaño del paso para satisfacer las tolerancias relativa (`rtol`) y absoluta (`atol`). Esto permite integrar eficientemente sistemas no lineales como el SIR, reduciendo el paso en zonas donde la solución varía rápidamente (por ejemplo, cerca del pico de infectados) y aumentándolo cuando la solución es suave.

C. Resultados de la simulación

La simulación muestra el comportamiento característico de una propagación tipo SIR. El número de susceptibles $S(t)$ disminuye rápidamente durante las primeras horas, mientras que la cantidad de infectados $I(t)$ crece hasta alcanzar un pico definido antes de comenzar a descender debido a la acción del término de recuperación $\beta I(t)$ y al agotamiento de susceptibles. Finalmente, el número de recuperados $R(t)$ crece monótonamente hasta estabilizarse cuando prácticamente no quedan equipos infectados.

El comportamiento observado refleja adecuadamente la dinámica de un brote de malware dentro de una red cerrada: un aumento brusco durante la fase inicial, un máximo de infecciones en un tiempo intermedio y una disminución progresiva a medida que los equipos son limpiados o aislados.

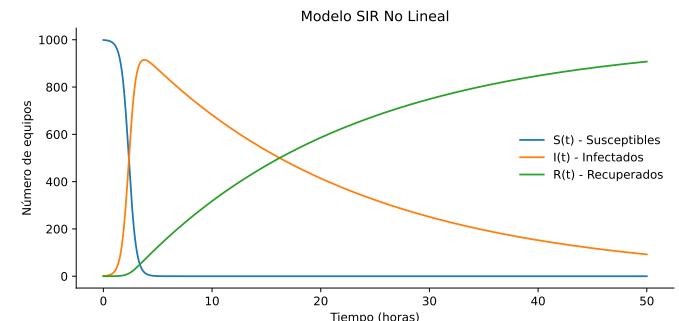


Fig. 3. Modelo SIR no lineal para un troyano ($N = 1000$)

D. Comparación con los modelos lineales

En el Modelo A (lineal sin recuperación), el número de infectados crece monótonamente, sin presentar un pico, porque la tasa de contagio depende sólo de $S(t)$ y no de la cantidad de infectados.

En el Modelo B (lineal con recuperación), $I(t)$ puede aumentar inicialmente y luego disminuir, pero el crecimiento es más suave, pues el término de contagio sigue siendo proporcional únicamente a $S(t)$.

En contraste, el modelo SIR no lineal incorpora el término αSI , haciendo que la tasa de infección sea proporcional a la interacción entre susceptibles e infectados. Esto produce un crecimiento inicial mucho más rápido y un máximo pronunciado de $|I(t)|$, seguido de una disminución marcada cuando los susceptibles se agotan y la recuperación domina. Así, el modelo SIR reproduce de manera más realista el comportamiento de un brote en una red cerrada.

III. ANEXO

```

1 import numpy as np
2
3 # condiciones iniciales
4 N = 1000 # nro de equipos
5 S0 = 999 # nro inicial de equipos
6     susceptibles
7 I0 = 1 # nro inicial de equipos
8     infectados
9 R0 = 0 # nro inicial de equipos
10    recuperados
11
12 # parametros
13 alpha = 0.003 # tasa de infeccion
14 beta = 0.05 # tasa de recuperacion
15
16 t = np.linspace(t0, t_final, 1000)

```

Código 1. parameters.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from parameters import alpha, beta, N, S0
4     , I0, t
5
6 # MODELO A (2 variables)
7 S_A = S0 * np.exp(-alpha * t)
8 I_A = I0 + S0 * (1 - np.exp(-alpha * t))
9
10 # MODELO B (3 variables)
11 S_B = S0 * np.exp(-alpha * t)
12 I_B = I0 * np.exp(-beta * t) + (alpha *
13     S0 / (beta - alpha)) * (
14     np.exp(-alpha * t) - np.exp(-beta * t)
15     )
16 R_B = N - S_B - I_B

```

Código 2. lineal.py

```

1 # GRAFICOS
2 fig, ax = plt.subplots(figsize=(10, 5))
3 ax.plot(t, S_A, label="S_A(t) -
4     susceptibles (Modelo A)")
5 ax.plot(t, I_A, label="I_A(t) -
6     infectados (Modelo A)")
7 ax.set_xlabel("Tiempo [h]")
8 ax.set_ylabel("Numero de equipos")
9 ax.set_title("Modelo A: sistema lineal
10    sin recuperacion")
11 ax.legend(frameon=False, loc="center
12    right")
13 ax.spines["top"].set_visible(False)
14 ax.spines["right"].set_visible(False)
15 plt.tight_layout()
16 plt.savefig("parte2/modelo_lineal_a.pdf",
17             format="pdf", bbox_inches="tight")
18
19 fig, ax = plt.subplots(figsize=(10, 5))
20 ax.plot(t, S_B, label="S_B(t) -
21     susceptibles", color="C0")
22 ax.plot(t, R_B, label="R_B(t) -
23     recuperados", color="C2")
24
25 # eje secundario para I(t)
26 ax2 = ax.twinx()
27 ax2.plot(t, I_B, label="I_B(t) -
28     infectados", color="C1")
29
30 ax.set_xlabel("Tiempo [h]")
31 ax.set_ylabel("S(t), R(t)")
32 ax2.set_ylabel("I(t)")
33
34 # combinar leyendas
35 lines, labels = ax.
36     get_legend_handles_labels()
37 lines2, labels2 = ax2.
38     get_legend_handles_labels()
39 ax.legend(lines + lines2, labels +
40     labels2, frameon=False, loc="center
41     right")
42 ax.set_title("Modelo B: sistema lineal
43    con recuperacion")
44
45 ax.spines["top"].set_visible(False)
46 ax.spines["right"].set_visible(False)
47 ax2.spines["top"].set_visible(False)
48 plt.tight_layout()
49 plt.savefig("parte2/modelo_lineal_b.pdf",
50             format="pdf", bbox_inches="tight")
51 plt.show()

```

Código 3. parameters.py

```

1 # ruff: noqa: E741
2 from scipy.integrate import solve_ivp
3 import matplotlib.pyplot as plt
4 from parameters import alpha, beta, S0,
5     I0, R0, t0, t_final, t
6
7
8
9 def sir_nolineal(t, y):
10     S, I, R = y
11     dSdt = -alpha * S * I
12     dIdt = alpha * S * I - beta * I
13     dRdt = beta * I
14     return [dSdt, dIdt, dRdt]
15
16
17 sol = solve_ivp(sir_nolineal, (t0,
18     t_final), y0, t_eval=t, rtol=1e-6,
19     atol=1e-8)
20
21 # vectores
22 t = sol.t
23 S, I, R = sol.y
24
25 # grafiquito
26 fig, ax = plt.subplots(figsize=(8, 4))
27 ax.plot(t, S, label="S(t) - Susceptibles")
28 ax.plot(t, I, label="I(t) - Infectados")
29 ax.plot(t, R, label="R(t) - Recuperados")
30 ax.set_xlabel("Tiempo (horas)")
31 ax.set_ylabel("Numero de equipos")
32 ax.legend(frameon=False)
33 ax.spines["top"].set_visible(False)
34 ax.spines["right"].set_visible(False)
35 ax.set_title("Modelo SIR No Lineal")
36 plt.tight_layout()
37 plt.savefig("parte2/no_lineal.pdf",
38     format="pdf", bbox_inches="tight")
39 plt.show()

```

Código 4. nolineal.py