

UNIVERSIDAD DE CÓRDOBA

Facultad de Ciencias

Grado de Física

Trabajo Fin de Grado

Física estadística de las epidemias

Código del TFG: **FS22-13-FSC**

Tipo de TFG: **Trabajo teórico-práctico general**

Autor: Antonio Rivas Blanco



8 de junio de 2023

Agradecimientos

Si bien este trabajo es otro trámite para conseguir los últimos seis créditos del grado, me gustaría tener unas palabras de agradecimiento con todas las personas que han hecho de estos cuatro años una preciosa etapa.

Primero me gustaría agradecer el apoyo de mis padres, siempre han confiado en mí, incluso cuando ni siquiera yo lo hacía. Han sido capaces de animarme cuando todo iba cuesta abajo como también lo han hecho todos mis amigos y amigas. Son un pilar fundamental cuando necesito desconectar. Gracias también a todos los compañeros de clase, por la solidaridad característica de los alumnos de este grado y por el genial ambiente de clase que hemos creado.

No puedo olvidarme de todos los profesores, con sus más y sus menos, han hecho del grado un camino muy bonito gracias a la cercanía que muestran hacia nosotros. En particular quiero darle las gracias a Jorge, que ha hecho que este trámite haya sido mucho más liviano, además haber hecho las clases algo diferente, divertido y enriquecedor. Gracias a docentes como tú muchos recuperamos las ganas de aprender y conseguís que nos enamoremos de lo que estudiamos.

Por último, gracias Laura por ser el hombro en el que siempre puedo apoyarme y Carla por sacar mi mejor versión.

Índice general

Índice general	3
Índice de figuras	5
Resumen. Palabras clave	6
Abstract. Keywords	7
1. Introducción	8
2. Objetivos	10
3. Métodos y materiales	11
3.1. Generación de números aleatorios	11
3.1.1. Generación de números pseudoaleatorios uniformes	12
3.1.2. Generación de números pseudoaleatorios no uniformes	13
3.2. Entorno de programación	14
3.3. Modelo SIS	15
3.4. Procesos estocásticos	16
3.4.1. Ecuación maestra	17
3.4.2. Ecuación de Fokker-Planck	18
3.4.3. Ecuación de Langevin	19
3.5. Métodos de generación de trayectorias	20
3.5.1. Algoritmo de Gillespie	20
3.5.2. Integración numérica de la ecuación de Langevin	22
4. Resultados	23
4.1. Series temporales	23
4.2. Efecto de las condiciones iniciales	26
4.3. Diagrama de fase	27
4.4. Desviación estandar	28
5. Conclusiones	30

Bibliografía	34
Anexo: Códigos de las distintas simulaciones	36

Índice de figuras

1.1. Línea temporal con las principales pandemias, adaptado de los datos de la tabla 1 de [1].	8
3.1. Distribución de los números pseudoaleatorios uniformes con el generador lineal congruencial para diferentes valores de los parámetros.	12
3.2. Diagrama del método de la transformada inversa. (a) Distribución de probabilidad. (b) Probabilidad acumulada.	13
3.3. Muestreo de 10^6 números aleatorios distribuidos gaussianamente usando el método de la transformada inversa.	14
3.4. Características del modelo SIS. (a) Dinámica del sistema. (b) Diagrama de fase del modelo SIS.	16
4.1. Número de infectados frente al tiempo para una población de 1000 individuos.	24
4.2. Número de infectados frente al tiempo para una población de 10000 individuos.	25
4.3. Número de infectados frente al tiempo para diferentes condiciones iniciales.	26
4.4. Número medio de infectados frente a la tasa de infección para distintas poblaciones.	27
4.5. Desviación estándar frente a la tasa de infección para diferentes poblaciones.	28

Resumen

Las epidemias han tenido un impacto de indudable importancia en los últimos tiempos, por lo que es necesario controlar la expansión de las enfermedades para un desarrollo próspero de la sociedad.

En este trabajo se dará una introducción a diferentes modelos de transmisión de enfermedades y nos centraremos en el estudio del modelo más simple, el susceptible-infectado-susceptible (SIS). Se hablará también sobre el tratamiento matemático para la resolución del modelo, que realizaremos mediante dos métodos: la simulación de trayectorias compatibles con el modelo usando el algoritmo de Gillespie y resolviendo numéricamente la ecuación de Fokker-Planck. Para ello, utilizaremos diferentes algoritmos implementados en lenguaje Python.

Finalmente, se representarán las simulaciones para diferentes valores de los parámetros las cuales nos permitirán sacar conclusiones sobre el comportamiento estocástico del sistema y posibles métodos de control de la enfermedad.

Palabras clave: procesos estocásticos; ecuación maestra; cambios de fase; algoritmo de Gillespie; ecuación de Fokker-Planck; ecuación de Langevin

Abstract

Epidemics have had an impact of doubtless importance in recent times, so it is necessary to control the expansion of diseases for a prosperous development of society.

In this paper an introduction about different transmission models will be given. We will focus on the study of the simplest one, the susceptible-infected-susceptible model (SIS). We will talk about the mathematical treatment for solving the model, which we will do by two methods: simulating compatible paths with the model using Gillespie's algorithm and solving numerically Langevin's equation. To do this, we will use different algorithms programmed in Python language.

Finally, the simulations will be represented for different values of the parameters which will allow us to draw conclusions about the system's stochastic behaviour and different control methods.

Keywords: stochastic processes; phase transitions; master equation; Gillespie's algorithm; Fokker-Plank equation; Langevin's equation

CAPÍTULO 1

Introducción

Las epidemias han sido y son un problema que además de afectar a la salud pública, perjudican además otros aspectos de la sociedad como la economía, la educación o la política. En la figura 1.1 podemos encontrar una línea temporal con algunas de las principales pandemias o epidemias que han sucedido a lo largo de la historia.



Figura 1.1. Línea temporal con las principales pandemias, adaptado de los datos de la tabla 1 de [1].

Por eso es importante entender como estas se transmiten, cómo podemos prevenir su expansión o controlarla en el caso en el que se haya producido la infección de una parte considerable de la población y así minimizar el impacto que tenga en la sociedad. Para lograr entender el comportamiento de la expansión de las enfermedades, haremos uso de la física estadística como una herramienta para el estudio y modelado de estas. Podemos relacionar las epidemias con la física estadística, ya que tratamos a la población como un conjunto de partículas interaccionantes, al cual aplicaremos distintos modelos para pronosticar el comportamiento de la enfermedad. Tal y como hacemos con la física estadística, estudiamos las interacciones microscópicas y a partir de estas obtenemos valores de magnitudes macroscópicas, lo que nos permitirá planificar diferentes estrategias de prevención y control de esta. En este trabajo, nos centraremos en un modelo simple, el susceptible-infectado-susceptible (SIS) [2]. En este modelo encontramos dos posibles estados de los individuos, infectado o susceptible a ser infectado. Los individuos infectados podrán pasar a ser susceptibles con cierta

probabilidad por unidad de tiempo y si un infectado interacciona con un susceptible podrá contagiarlo bajo otra probabilidad por unidad de tiempo.

Para estudiar la evolución del sistema, utilizaremos el algoritmo de Gillespie [3], un algoritmo que nos permitirá obtener trayectorias compatibles con las soluciones de la ecuación maestra [4, 5], una ecuación diferencial determinista para la probabilidad de encontrar a nuestro sistema en cierto estado discreto. Esta técnica tiene en cuenta la naturaleza estocástica e impredecible de las epidemias, debido a la incertidumbre probabilística de los contagios y recuperaciones. Además de usar método, obtendremos la ecuación de Fokker-Planck [4, 5] a partir de la ecuación maestra, ya que es un desarrollo en serie a segundo orden de dicha ecuación, en este caso es una ecuación en derivadas parciales que nos permite obtener la probabilidad de encontrar nuestro sistema en cierto estado pero en este caso para una variable cuasicontinua, el número de infectados normalizado. Finalmente, partiendo de la ecuación de Fokker-Planck obtendremos la ecuación de Langevin [4], una ecuación diferencial estocástica para nuestra variable estocástica, el número de infectados normalizado. La resolución de esta ecuación también nos permitirá describir la evolución del sistema.

Para la resolución de dichas ecuaciones, debido a la complejidad o inexistencia de soluciones analíticas, recurriremos a resoluciones numéricas. Para ello, nos ayudaremos de los métodos Montecarlo [6], los cuales nos permiten resolver este tipo de problemas. Estos métodos se basan en la generación de números aleatorios y repetición de cálculos para obtener una solución aproximada del problema. Además, nos permitirán predecir comportamientos de nuestro sistema de manera probabilística para poder estudiar los sistemas complejos.

A partir de la resolución de estas ecuaciones podremos obtener información cuantitativa sobre la propagación de la enfermedad, como la evolución temporal del número de infectados, la velocidad de transmisión, la probabilidad de que exista un brote epidémico. Distinguimos dos zonas en el problema, la zona absorbente y la zona activa. La primera se da cuando la velocidad de recuperación es mayor a la de infección, por lo que con el paso del tiempo, el número de individuos infectado es nulo. Por otro lado, nos encontramos en la zona activa cuando la velocidad de infección es mayor que la de recuperación, por lo que el número de infectados con el paso del tiempo no será nulo. Un brote epidémico cuando se produce con el paso de la zona absorbente a la activa, ya que la probabilidad de infección será mayor a la de recuperación y, por lo tanto, se producirá un aumento del número de infectados de la población.

Además, podemos usar la física estadística con otros aspectos de la vida. Por ejemplo, podemos comprender la propagación de información y rumores en redes sociales [7] o el contagio de comportamientos y opiniones. Los principios y enfoques utilizados para modelar las epidemias pueden ser aplicados a una amplia gama de fenómenos sociales y biológicos [8] en los que la propagación y la interacción juegan un papel crucial.

CAPÍTULO 2

Objetivos

En este Trabajo Fin de Grado se han planteado los siguientes objetivos:

- Estudiar el modelo susceptible-infectado-susceptible (SIS) en un sistema de tamaño finito y por tanto, sujeto a fluctuaciones. Para ello, haremos uso de la teoría de procesos estocásticos
- Utilizar métodos Montecarlo, la ecuación de Fokker-Planck, la de Langevin y el algoritmo de Gillespie para estudiar la evolución del sistema.
- Implementar en lenguaje Python dichas ecuaciones y algoritmo para su resolución.
- Visualizar y analizar los resultados obtenidos para distintos órdenes de población e interacción. En concreto, nos centraremos en el estudio del diagrama de fase y la amplitud de las fluctuaciones.

CAPÍTULO 3

Métodos y materiales

En este capítulo se mostrarán las herramientas matemáticas necesarias para obtener los resultados, se comienza con una breve introducción a la generación de números aleatorios. Se describirá el modelo SIS y se relacionará con la teoría de procesos estocásticos; relativos a estos, hablaremos de la ecuación maestra, la de Fokker-Planck y la de Langevin. Finalmente, se describirá el algoritmo de Gillespie.

3.1. GENERACIÓN DE NÚMEROS ALEATORIOS

Necesitamos hacer un pequeño preámbulo a los números aleatorios, ya que tienen un papel fundamental en los métodos Montecarlo, los cuales nos permitirán usar estos números aleatorios para obtener soluciones numéricas de nuestros modelos, en particular, a generar variables estocásticas a las que les asignaremos una distribución determinada.

Estos métodos dependen de generadores de números pseudopaleatorios, que son algoritmos que nos permiten obtener una secuencia de números determinista y periódica cuyas propiedades son muy parecidas a una secuencia totalmente aleatoria. Estas secuencias dependen de un valor inicial, que llamaremos “semilla”. En muchos casos la elección de esta es fundamental porque la “calidad” de los números obtenidos puede depender del valor inicial.

En este trabajo se han utilizado generadores de números aleatorios implementados en la biblioteca “numpy” de Python. Aun así, vamos a estudiar algún ejemplo para entender los fundamentos de dichos métodos.

3.1.1. GENERACIÓN DE NÚMEROS PSEUDOALEATORIOS UNIFORMES Es uno de los primeros métodos descubiertos, el generador lineal congruencial, para un estudio más profundo véase el Anexo A de [5]. Mediante este método se obtienen enteros aleatorios que llamaremos $x \in [0, M - 1]$. Donde M es el entero que actuará como cota superior. De forma general es interesante que M sea un número grande, ya que si hacemos el cambio de variable x/M obtendremos números cuasiuniformes en el intervalo $[0, 1]$, que es la base de muchos otros, como el método de la transformada inversa que discutiremos posteriormente. La secuencia de números está definida por la siguiente relación de recurrencia:

$$n_k = (a n_{k-1} + b) \bmod M \quad (3.1)$$

Donde n_0 es la semilla y a y b son parámetros, multiplicador e incremento respectivamente. En función de los valores que tomen, obtendremos números mejor o peor distribuidos uniformemente. Un inconveniente de este método es que es muy sensible a la elección de los parámetros, tanto que puede ser que los números lleguen a estar correlacionados o que el periodo sea demasiado pequeño, por lo que no tendrían las propiedades de número aleatorio que buscamos. A continuación, en la figura 3.1 podemos encontrar como podemos observar como se modifica la distribución de números aleatorios para diferentes valores de los parámetros.

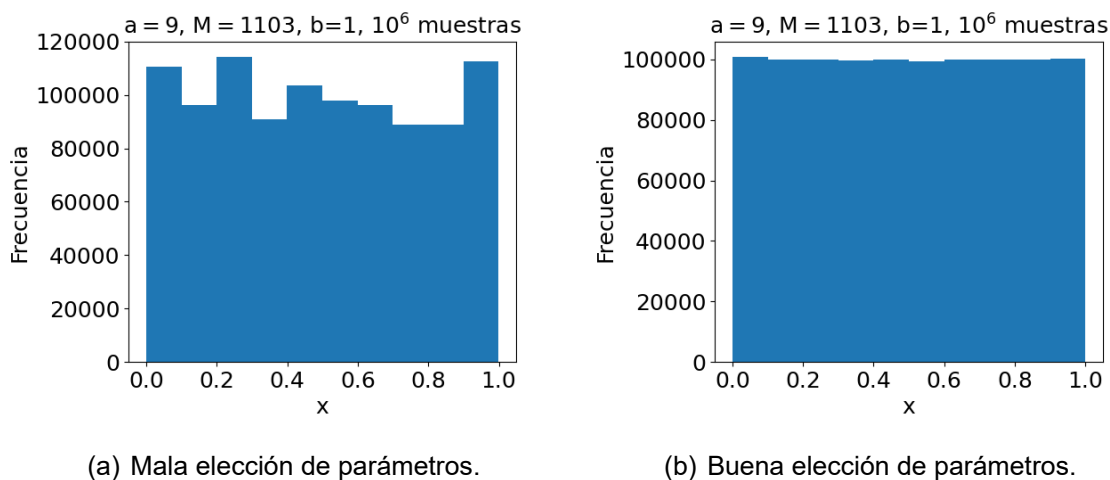


Figura 3.1. Distribución de los números pseudoaleatorios uniformes con el generador lineal congruencial para diferentes valores de los parámetros.

Para consultar otros valores buenos y malos de estos parámetros, véase [9].

3.1.2. GENERACIÓN DE NÚMEROS PSEUDOALEATORIOS NO UNIFORMES Dependen de los generadores de números uniformes, ya que a partir de estos, aplicándoles otros métodos, lograremos simular las distribuciones no uniformes. En el caso de la generación de números pseudoaleatorios uniformes, la secuencia de enteros pertenece al intervalo $[0, M - 1]$. Sin embargo, será interesante definir una nueva variable, $u = x/M - 1$ de tal forma que $u \in [0, 1]$ es una variable cuasicontinua.

Nos centraremos en el método de la transformada inversa, capítulo 2 de [5]. Permite obtener un número no uniformemente distribuido por cada número uniformemente distribuido. El inconveniente de este método es que no siempre podemos encontrar fácilmente la función inversa, por lo que generalmente, tendremos que acudir a otros métodos.

Supongamos que queremos obtener una variable $x \in [a, b]$ que esté distribuida como una función cualquiera $p(x)$. Para ello, partimos de una variable cuasicontinua en el intervalo $[0, 1]$, de tal forma que u es un número aleatorio que pertenece a este intervalo, $u \sim U[0, 1]$. Tendremos que encontrar el valor de x tal que $P(x) = u$, por lo que podremos despejar $x = P^{-1}(u)$, de esta forma x estará distribuida como nosotros le pedimos. Se ve a continuación la demostración.

$$p_x(x) = \frac{1}{p_u(u)} \frac{du}{dx} = \frac{d}{dx} \underbrace{\int_a^x p(x') dx'}_{P(x)} = p(x)$$

A continuación, en la figura 3.2 se muestra un diagrama del método de la transformada inversa.

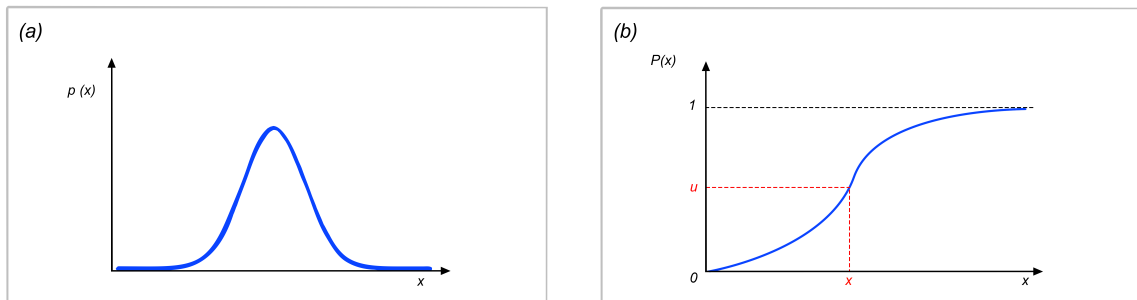


Figura 3.2. Diagrama del método de la transformada inversa. (a) Distribución de probabilidad. (b) Probabilidad acumulada.

En nuestro caso utilizaremos un generador de números aleatorios que sigan una distribución gaussiana, en la figura 3.3 encontramos dicha distribución obtenida mediante este método.

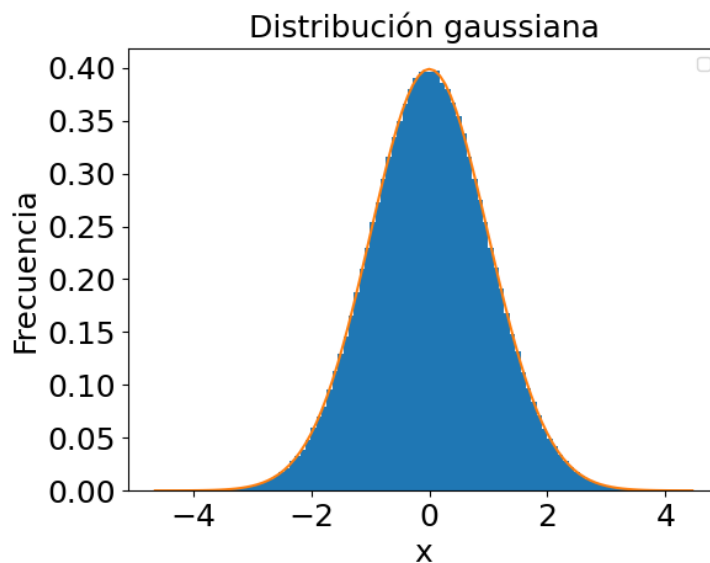


Figura 3.3. Muestreo de 10^6 números aleatorios distribuidos gaussianamente usando el método de la transformada inversa.

3.2. ENTORNO DE PROGRAMACIÓN

Para la implementación de los métodos de resolución se ha utilizado el lenguaje Python. Concretamente, se ha utilizado python instalado en Visual Studio Code, que también ha permitido realizar este documento en formato \LaTeX . Se ha elegido este lenguaje de programación debido a la simplicidad de su sintaxis lo cual ha hecho su aprendizaje más llevadero. Además, puesto que es uno de los lenguajes en auge en la actualidad [10], podremos encontrar gran cantidad de bibliotecas que nos facilitarán el trabajo. En el caso de este trabajo, hemos utilizado las bibliotecas “numpy” y “matplotlib”.

La primera proporciona un amplio abanico de funciones matemáticas, además de un soporte más eficiente para cálculos que el perteneciente a Python lo que hará que nuestras simulaciones sean más rápidas, algo que nos interesa, ya que cuanto mayor sea la población que queramos analizar, se necesitarán en general, hacer más cálculos. Además, esta biblioteca incluye generadores de números aleatorios que utilizaremos para resolver las ecuaciones de nuestro modelo; también nos permitirá obtener medias y desviaciones de nuestros conjuntos de datos rápidamente.

Por otro lado, la biblioteca matplotlib nos permite graficar los resultados obtenidos, nos hará posible estudiar y analizar la expansión de la epidemia de manera visual. En general, la combinación de estas dos bibliotecas junto con el lenguaje Python permitirá

resolver problemas de manera eficiente con el uso de un lenguaje de programación sencillo.

Para una descripción más profunda de estas bibliotecas, véase [11] y/o [12].

3.3. MODELO SIS

El modelo que utilizaremos para caracterizar la epidemia es el SIS: susceptible-infectado-susceptible. Además, consideraremos el caso en el que todos los individuos pueden interactuar con todos y que el número total de individuos no cambia, para un estudio de este modelo en redes heterogéneas, véase [13].

Bajo este modelo, los individuos pueden estar en dos estados, infectado y susceptible a ser infectado. Como el número de individuos totales que denotamos como N es constante, el estado quedará determinado por el número de infectados (n_i) o susceptibles (n_s), ya que $N = n_i + n_s$.

En este modelo se pueden dar dos interacciones que alteren el estado del sistema. La primera es que un individuo susceptible interactúe con un infectado y este lo infecte con una probabilidad por unidad de tiempo que denotamos λ . Podemos esquematizar esta interacción de la siguiente forma:



La otra interacción es que un individuo infectado se recupere y pase a volver susceptible, con una probabilidad por unidad de tiempo de ocurrir que llamamos μ . El esquema de esta interacción queda:



Este es el modelo más simple. Podemos estudiar variantes de este como el SIR, en el que cuando el individuo se recupera no es susceptible a infectarse durante un determinado periodo de tiempo, ya que ha adquirido una inmunidad. Para el estudio más detallado de este y otros modelos, diríjase a [13].

Nosotros estudiaremos el modelo SIS, al que le haremos una pequeña modificación que consiste en introducir un individuo infectado en la población en el momento en que el número de infectados sea nulo. Esta modificación es consecuencia de que la dinámica del sistema cuando no hay individuos infectados se para, ya que no se pueden dar ninguna de las dos interacciones 3.2 y 3.3. En la figura 3.4 se muestra un diagrama de las interacciones del modelo SIS y el diagrama de fase que obtendremos bajo nuestras consideraciones.

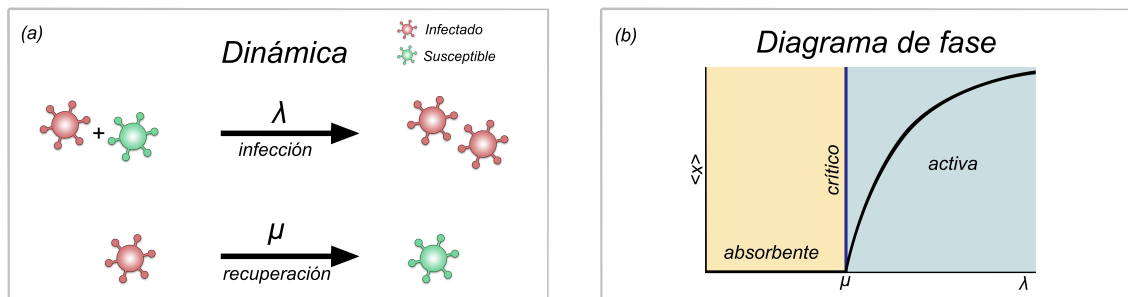


Figura 3.4. Características del modelo SIS. (a) Dinámica del sistema. (b) Diagrama de fase del modelo SIS.

Como se ve en la figura (b) distinguimos dos fases, la fase absorbente es en la que $\mu > \lambda$, lo que se traduce en que la probabilidad por unidad de tiempo de recuperación es mayor a la de infección, por lo que con el paso del tiempo, se recuperará toda la población y por ello el número medio de infectados es cero, por lo que es la zona en la que queremos situarnos en la realidad, puesto que aunque existan individuos infectados en la población, se producirá la recuperación total de estos.

Encontramos también un punto crítico cuando $\lambda = \mu$. Aquí es donde es más relevante la parte estocástica de nuestro problema, ya que las fluctuaciones del sistema pueden llevarnos a la zona activa o a la absorbente. Estas fluctuaciones toman mayor importancia cuanto menor es la población, ya que el cambio de estado de un sistema hace que el porcentaje de individuos infectados sea mucho mayor o menor relativo a la población total. No es conveniente encontrarnos en esta zona, debido a que es difícil pronosticar cuál será la evolución de la enfermedad, por lo que es más complicado tomar decisiones para paliar la expansión de esta.

Por último, la zona activa se da cuando $\lambda > \mu$, lo que indica que se producirán más infecciones que recuperaciones con el paso del tiempo. Esta es la zona más peligrosa puesto que pese a que el tiempo pase, el número de infectados no bajará naturalmente, sino que se tendrán que poner remedios como el aislamiento para que no se produzcan interacciones o bien tratamientos médicos que disminuyan la probabilidad de infección para poder volver a la zona absorbente.

3.4. PROCESOS ESTOCÁSTICOS

Como se ha dicho anteriormente, la evolución de una epidemia es un proceso estocástico, ya que nuestra variable, el número de infectados, evoluciona de una manera

no determinista. Concretamente, se trata de un proceso de Markov [4], en el que la evolución no depende de la historia pasada del sistema, sino solo del estado en el que se encuentra.

3.4.1. ECUACIÓN MAESTRA La ecuación maestra es una ecuación diferencial, por lo que es determinista, que describe la evolución de la probabilidad en un proceso de Markov para sistemas que pasan de un estado a otro de manera continua en el tiempo [4]. Esta ecuación es útil para procesos estocásticos en los que tenemos un número discreto de estados accesibles, como es nuestro caso.

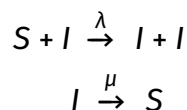
Para obtener esta ecuación, denotamos $\vec{n}(t)$ como el estado del sistema y llamamos $P(\vec{n}, t | \vec{n}_0, t_0)$ a la probabilidad de encontrar el sistema en el estado \vec{n} en el tiempo t dadas las condiciones iniciales $\vec{n}(t = t_0) = \vec{n}_0$. Para obtener la ecuación maestra, consideraremos que todos los individuos pueden interactuar con el resto, así no nos tendremos que preocupar de la posición de los individuos.

La ecuación maestra es una ecuación de continuidad para el cambio de la probabilidad $P(\vec{n}, t | \vec{n}_0, t_0)$ en el tiempo, por lo que la variación temporal de dicha probabilidad vendrá dada por el flujo de entrada de probabilidad menos el flujo de salida de esta. Si llamamos $W(\vec{n} \rightarrow \vec{n}')$ la probabilidad por unidad de tiempo de pasar del estado \vec{n} al estado \vec{n}' la ecuación maestra¹ queda:

$$\frac{\partial P(\vec{n}, t)}{\partial t} = \sum_{\vec{n}'} [W(\vec{n}' \rightarrow \vec{n})P(\vec{n}', t) - W(\vec{n} \rightarrow \vec{n}')P(\vec{n}, t)]. \quad (3.4)$$

Tendremos un sistema de ecuaciones diferenciales acopladas, una para la probabilidad de cada estado del sistema, por lo que solo tendrá solución analítica en muy pocos casos y será costoso de resolver computacionalmente. Debido a esto, utilizaremos el algoritmo de Gillespie, ver 3.5, para obtener trayectorias compatibles con la solución de esta ecuación. Con este algoritmo obtendremos las trayectorias a partir de la evolución de nuestra variable estocástica, el número de infectados.

Vamos a obtener la ecuación maestra del modelo SIS, para posteriormente, tomaremos la aproximación de tamaños grandes, la cual nos llevará a una ecuación más sencilla de tratar, la ecuación de Fokker-Planck. Recordamos las posibles interacciones que se pueden dar en el modelo.



El sistema quedará determinado por el número de individuos infectados, ya que supondremos que la población es constante, es decir, $N = n_I + n_S$. Para no cargar la notación usaremos $n \equiv n_I$.

¹Se ha omitido para no sobrecargar la notación la parte de las condiciones iniciales.

Definimos las tasas de probabilidad W_+ y W_- de la siguiente manera para poder obtener la ecuación maestra.

$$\begin{aligned} W_+(n) &\equiv W_+(n \rightarrow n+1) = \lambda n \left(1 - \frac{n}{N}\right) \\ W_-(n) &\equiv W_-(n \rightarrow n-1) = \mu n \end{aligned} \quad (3.5)$$

Estas tasas son la probabilidad de un cambio de estado durante un intervalo de tiempo Δt . Como se ve en la ecuación 3.5, es la misma expresión que para la colisión de partículas. Producto de la probabilidad de que se produzca cada uno de los procesos de 3.2 y 3.3 multiplicado por el número de individuos en el estado de partida.

De esta forma, la ecuación maestra queda:

$$\begin{aligned} \partial_t P(n, t) &= W_+(n-1)P(n-1, t) + W_-(n+1)P(n+1, t) - \\ &- W_-(n)P(n, t) - W_+(n)P(n, t) \end{aligned} \quad (3.6)$$

los dos primeros términos son las probabilidades por unidad de tiempo de que se produzca una interacción que de lugar al estado (n, t) . A este estado se puede llegar a partir del estado $(n-1)$ o desde el estado $(n+1)$, correspondientes al primer y segundo sumando respectivamente. Los otros dos términos, que están restando, son las probabilidades de que se produzcan interacciones que dan lugar a otro estado. Partiendo del estado n , podremos ir a los estados $(n-1)$ o al estado $(n+1)$, correspondientes al último y penúltimo sumando respectivamente. Hacemos el cambio de variable discreta (n) a cuasicontinua (x) adimensionalizando n , $x = n/N$. Por lo que las probabilidades cambian también, quedando de esta forma $P(n, t) = \tilde{P}(x, t) \frac{dx}{dn} = \tilde{P}(x, t) \frac{1}{N}$ y se define también las nuevas tasas \tilde{W} de la siguiente forma:

$$\begin{aligned} N\tilde{W}_+(x, t) &= W_+(n, t) = \lambda n \left(1 - \frac{n}{N}\right) \implies \tilde{W}_+(x, t) = \lambda x(1 - x) \\ N\tilde{W}_-(x, t) &= W_-(n, t) = \mu n \implies \tilde{W}_-(x, t) = \mu x \end{aligned}$$

Teniendo en cuenta esto, la ecuación maestra queda:

$$\begin{aligned} \partial_t \left(\tilde{P}(x, t) \frac{1}{N} \right) &= \overbrace{N\tilde{W}_+ \left(x - \frac{1}{N}, t \right) \tilde{P} \left(x - \frac{1}{N}, t \right) \frac{1}{N}}^{f(x-1/N)} + \overbrace{N\tilde{W}_- \left(x + \frac{1}{N}, t \right) \tilde{P} \left(x + \frac{1}{N}, t \right) \frac{1}{N}}^{f(x+1/N)} \\ &- N\tilde{W}_+(x, t) \tilde{P}(x, t) \frac{1}{N} - N\tilde{W}_-(x, t) \tilde{P}(x, t) \frac{1}{N} = \\ &= f \left(x - \frac{1}{N} \right) + f \left(x + \frac{1}{N} \right) - N\tilde{P}(x, t) (\tilde{W}_+(x) + \tilde{W}_-(x)) \end{aligned} \quad (3.7)$$

3.4.2. ECUACIÓN DE FOKKER-PLANCK Para obtener la ecuación de Fokker-Planck debemos hacer un desarrollo en serie a segundo orden en torno a los puntos $x \pm \frac{1}{N}$ de la nueva función f que hemos definido. Podemos aceptar el desarrollo a segundo orden

siempre que tengamos una población grande, ya que hacemos el resto de términos los podremos despreciar.

$$f\left(x \pm \frac{1}{N}, t\right) \equiv f(x \pm \Delta x, t) \approx f(x, t) \pm \frac{1}{N} \partial_x f(x, t) + \frac{1}{2N^2} \partial_x^2 f(x, t) + \dots$$

Y la ecuación maestra pasa a ser

$$\begin{aligned} \partial_t \tilde{P}(x, t) = & \cancel{N\tilde{W}_+(x)P(x, t)} - \frac{1}{N} \partial_x [N\tilde{W}_+(x)P(x, t)] + \frac{1}{2N^2} \partial_x^2 [N\tilde{W}_+(x)P(x, t)] + \\ & + \cancel{N\tilde{W}_-(x)P(x, t)} - \frac{1}{N} \partial_x [N\tilde{W}_-(x)P(x, t)] + \frac{1}{2N^2} \partial_x^2 [N\tilde{W}_-(x)P(x, t)] - \\ & - \cancel{N\tilde{W}_+(x)P(x, t)} - \cancel{N\tilde{W}_-(x)P(x, t)} \end{aligned} \quad (3.8)$$

Reagrupando términos y particularizando para nuestras tasas de probabilidad, finalmente obtenemos la ecuación de Fokker-Planck.

$$\begin{aligned} \partial_t \tilde{P}(x, t) = & -\partial_x [(\tilde{W}_+ - \tilde{W}_-) \tilde{P}(x, t)] + \frac{1}{2N} \partial_x^2 [(\tilde{W}_+ + \tilde{W}_-) \tilde{P}(x, t)] \\ = & -\partial_x [(\lambda x(1-x) - \mu x) \tilde{P}(x, t)] + \frac{1}{2N} \partial_x^2 [(\lambda x(1-x) + \mu x) \tilde{P}(x, t)] \end{aligned} \quad (3.9)$$

Debido a que es un desarrollo en serie de la ecuación Maestra, la ecuación de Fokker-Planck sigue siendo una ecuación determinista para la probabilidad, pero en este caso de una variable cuasicontinua. Es por esto que puede ser costosa de resolver tanto analítica como numéricamente, por lo que de manera análoga a como haremos con el algoritmo de Gillespie, podemos encontrar trayectorias compatibles con la solución de esta ecuación. Estas trayectorias las obtendremos resolviendo la ecuación de Langevin, una ecuación diferencial estocástica para la nueva variable estocástica, el número de infectados normalizado, x .

3.4.3. ECUACIÓN DE LANGEVIN Partiendo de la ecuación 3.7, se pueden definir las siguientes funciones:

$$\begin{aligned} A(x) &= \lambda x(1-x) - \mu x \\ B(x) &= \lambda x(1-x) + \mu x \end{aligned} \quad (3.10)$$

obteniéndose así la siguiente expresión para la ecuación de Fokker-Planck.

$$\partial_t P(x, t) = -\partial_x (A(x)P(x)) + \frac{1}{2} \partial_x^2 (B(x)P(x))$$

Podemos ver como se relacionan la ecuación de Langevin y de Fokker-Planck en [4, 5]. Si realizamos la integración bajo la interpretación de Itô² la ecuación de Langevin pasa a ser:

$$\dot{x} = A(x) + \sqrt{B(x)}\xi(t) \quad (3.11)$$

Donde $\xi(t)$ se denomina ruido blanco gaussiano, [4, 5]. Las principales características de este tipo de ruido son que su media es nula y no está correlacionado con él mismo, por lo que su función de autocorrelación es una delta de Kronecker. Físicamente, lo podemos identificar con el límite al continuo de la velocidad de una partícula que sigue un movimiento Browniano, es decir, cambia su dirección aleatoriamente y de forma uniforme en el tiempo.

Como vemos, es una ecuación diferencial estocástica, por lo que será mucho más fácil la integración numérica.

Para nuestro caso, la ecuación queda:

$$\dot{x} = \lambda x(1 - x) - \mu x + \frac{1}{\sqrt{N}} \sqrt{\lambda x(1 - x) + \mu x} \xi(t) \quad (3.12)$$

Es interesante tomar el límite de población $N \rightarrow \infty$, para no encontrar fluctuaciones demográficas. De esta forma, el segundo término, relacionado con la aleatoriedad de los procesos estocásticos tiende a cero. Esto nos permitirá obtener el número medio de infectados una vez alcanzado el estado estacionario. Para obtener dicho valor, igualamos $\dot{x} = 0$, por lo que la ecuación de Langevin queda:

$$\begin{aligned} \dot{x} &= \lambda x(1 - x) - \mu x = 0 && \text{Omitimos la solución}^3 x^{est} = 0 \\ \lambda(1 - x^{est}) - \mu &= 0 \\ x^{est} &= 1 - \frac{\mu}{\lambda} \end{aligned} \quad (3.13)$$

3.5. MÉTODOS DE GENERACIÓN DE TRAYECTORIAS

3.5.1. ALGORITMO DE GILLESPIE Como se ha comentado anteriormente, el costo computacional para resolver numéricamente la ecuación maestra es muy alto, es por eso que utilizaremos este algoritmo para poder simular trayectorias temporales de la variable estocástica compatibles con la solución de la ecuación maestra.

Para esto, en lugar de estudiar $P(\vec{n}, t | \vec{n}_0, t_0)$, estudiaremos una nueva probabilidad, $p(\tau, t | \vec{n}, t)$, que se define tal que $p(\tau, t | \vec{n}, t)d\tau$ es la probabilidad de que ocurra la reacción j en el intervalo de tiempo $d\tau$. Esta probabilidad vendrá dada por la siguiente expresión [3]:

²Básicamente significa que el ruido “actúa” en el momento inicial de cada paso de la integración.

$$P(\tau, \vec{n}, t) = \sum_j W_j(\vec{n}) \exp\left(-\sum_j W_j \tau\right) \quad (3.14)$$

Donde W_j es la probabilidad de que la interacción j se produzca en un intervalo de tiempo $\delta\tau$

Si sumamos para todos los valores de j , obtendremos la probabilidad de que cualquier reacción ocurra en el intervalo de tiempo $d\tau$, que será la base para obtener las trayectorias compatibles con la solución de la ecuación maestra.

$$P(\tau|\vec{n}, t) = \sum_j W_j(\vec{n}) \exp\left(-\sum_j W_j(\vec{n})\tau\right) \quad (3.15)$$

Como vemos en la ecuación 3.15, τ se distribuye como una variable aleatoria exponencial de media $W_0(\vec{n}) \equiv \frac{1}{\sum_j W_j(\vec{n})}$ y j es un número entero aleatorio cuya probabilidad es W_j/W_0 .

Para generar valores de τ y j compatibles utilizaremos el método directo, el cual se basa en el método de la transformada inversa mencionado anteriormente. Primero debemos obtener dos enteros aleatorios que llamamos r_1 y r_2 pertenecientes a la distribución $U(0, 1)$. A partir de estos obtenemos τ y j :

$$\tau = \frac{1}{W_0(\vec{n})} \ln\left(\frac{1}{r_1}\right) \quad (3.16)$$

$$j = \text{el entero más pequeño tal que } \sum_{j'=1}^j W_{j'}(\vec{n}) > r_2 W_0(\vec{n})$$

Una vez obtenidos, el algoritmo que tenemos que realizar es el siguiente:

1. Establecemos las condiciones iniciales, $t = t_0$ y $\vec{n} = \vec{n}_0$.
2. Evaluamos todos los posibles $W_j(\vec{n})$ y calculamos $W_0(\vec{n})$.
3. Obtenemos los valores de τ y j usando la ecuación 3.16.
4. Elegimos qué reacción ocurrirá utilizando la probabilidad de estas.
5. Guardamos los datos.
6. Actualizamos el sistema tomando el estado actualizado como estado inicial.
7. Repetir el proceso desde el paso 2 o terminar la simulación.

3.5.2. INTEGRACIÓN NUMÉRICA DE LA ECUACIÓN DE LANGEVIN Debemos distinguir las dos partes de la ecuación de Langevin, el sumando que contiene el término de ruido y el que no lo tiene. Para la integración del sumando sin el término de ruido podemos utilizar cualquier método de integración, en nuestro caso, hemos utilizado el método de Euler.

Por otro lado, para el sumando que contiene el término de ruido no tenemos tal libertad, ya que la integral de este término es una integral estocástica. En nuestro caso, tomaremos la siguiente integral:

$$\int_t^{t+dt} \xi(t') dt' = G_t \sqrt{dt} \quad \text{Donde } G_t \text{ es un número gaussiano } (0, 1).$$

Esta integral la tenemos que hacer una vez que sacamos fuera de la integral el término $B(x)$ de la ecuación 3.11. Es decir, estamos realizando la integral bajo la interpretación de Itô [4]. Existen otras formas de integrar la ecuación de Langevin, pero la equivalencia que hemos hecho entre esta y la ecuación no es exactamente [5].

CAPÍTULO 4

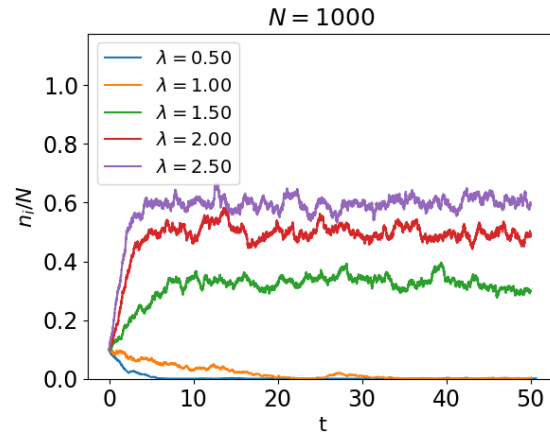
Resultados

En este capítulo se mostrarán todas las figuras¹ obtenidas a partir de las simulaciones. Se estudiará la evolución del número de infectados frente al tiempo, cómo afectan las condiciones iniciales, el número medio de infectados frente a la tasa de infección y el cambio de la desviación estándar de diferentes tamaños de población.

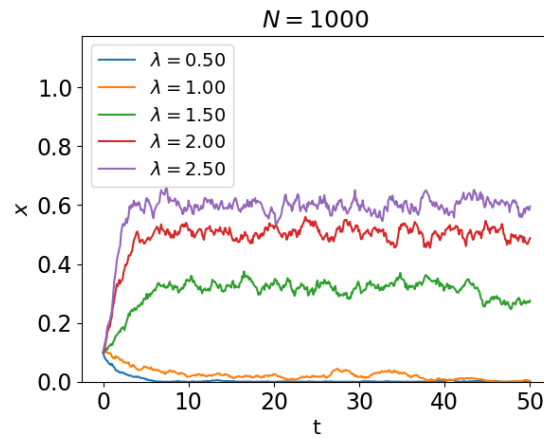
4.1. SERIES TEMPORALES

En las figuras 4.1(a) y 4.1(b) se muestra cómo evoluciona el número de infectados frente al tiempo para diferentes tasas de infección λ en una población de $N = 1000$ individuos. Encontraremos un estado transitorio en el que se produce la infección o recuperación de la población de la población en función de la tasa de infección y pasado este tiempo, se llega al estado estacionario, donde el número de infectados oscilará en torno al valor estacionario obtenido en la ecuación 3.12.

¹Para todas las simulaciones $\mu = 1$. Las dimensiones de este parámetro y de λ son de *tiempo*⁻¹. Se utilizará un sistema de unidades arbitrario, lo que quiere decir que se producirá una recuperación por hora, día, mes... Debido a esta arbitrariedad en las unidades, hemos decidido omitirlas en todas las gráficas. Además, la condición inicial para todos los casos exceptuando la sección 4.2, la condición inicial es que el número de infectados inicial es $N/10$



(a) Gillespie.

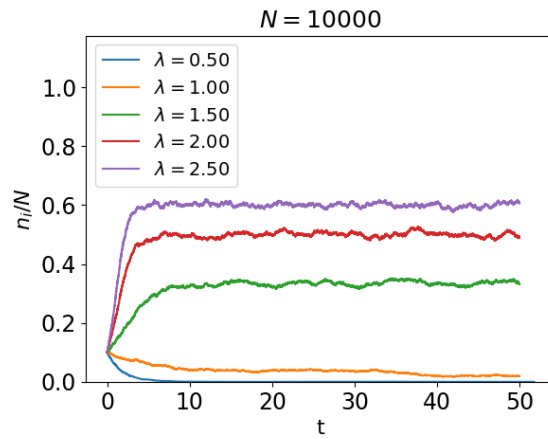


(b) Langevin.

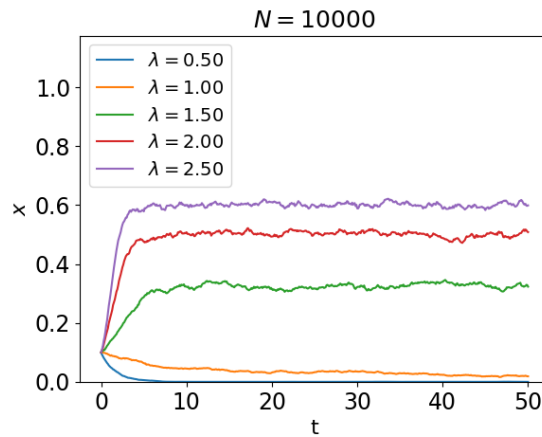
Figura 4.1. Número de infectados frente al tiempo para una población de 1000 individuos.

Como se puede comprobar, encontramos un período transitorio hasta que llegamos al estado estacionario, en el que para el caso en el que $\mu > \lambda$ el número de infectados tiende a cero, ya que hay mayor probabilidad de recuperación que de infección. Por otro lado, cuando $\mu < \lambda$ el número de infectados oscilará en torno al valor estacionario que hemos obtenido en la ecuación 3.13.

Podemos analizar también cómo cambian las fluctuaciones para poblaciones más grandes. Aumentando en un orden de magnitud la población, las fluctuaciones se reducirán en un factor $1/\sqrt{N}$ tal y como aparece en la ecuación de Langevin 3.12. En las figuras 4.2(a) y 4.2(b) podemos ver la atenuación de estas fluctuaciones. En general, todas las trayectorias para poblaciones mayores serán similares, pero con fluctuaciones más pequeñas, por lo que no es necesario conocer el estado estacionario dichas poblaciones. Debido a que el tiempo de simulación aumenta con el N en la resolución con el algoritmo de Gillespie, nos restringiremos a poblaciones de 10, 100 y 1000 individuos.



(a) Gillespie.

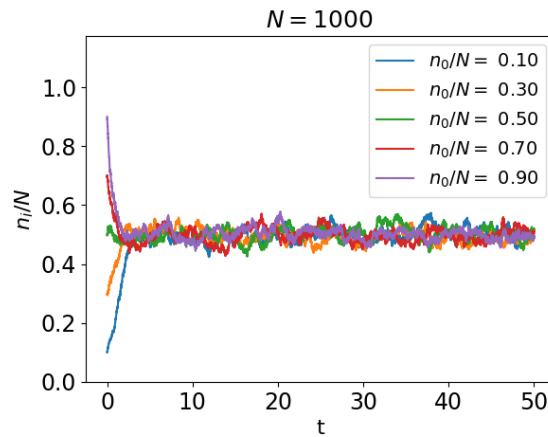


(b) Langevin.

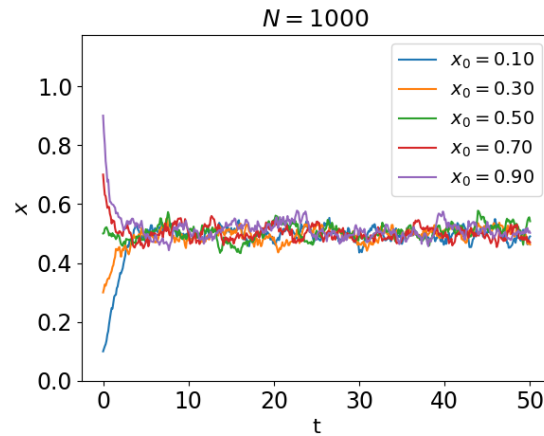
Figura 4.2. Número de infectados frente al tiempo para una población de 10000 individuos.

4.2. EFECTO DE LAS CONDICIONES INICIALES

A continuación, veremos cómo el estado estacionario es independiente al número de infectados inicial, al estar tratando con procesos de Markov, los cuales solo dependen del estado del sistema en el tiempo anterior. Además, esto también queda explícito en la ecuación de Langevin 3.12, donde no aparece ningún término referido a las condiciones iniciales. Se muestran en la figura 4.3 la evolución del número de infectados frente al tiempo para diferentes condiciones iniciales.



(a) Gillespie.



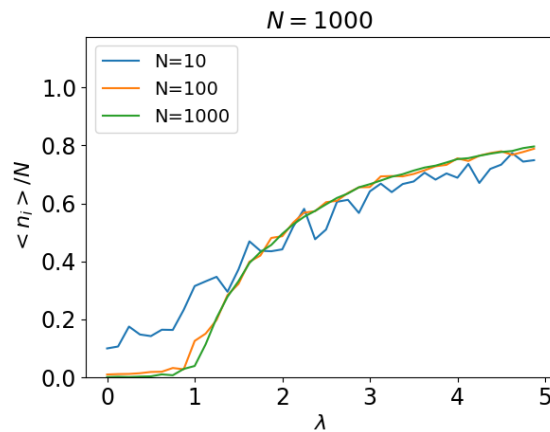
(b) Langevin.

Figura 4.3. Número de infectados frente al tiempo para diferentes condiciones iniciales.

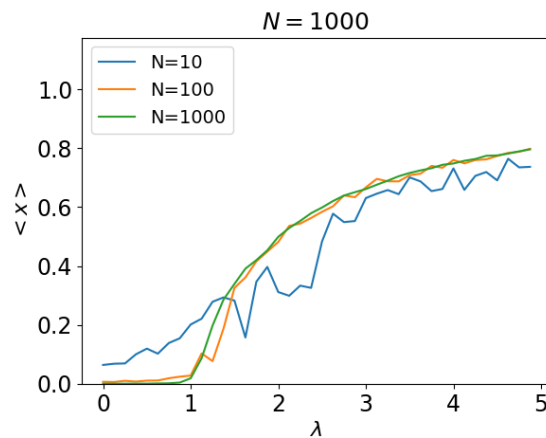
Como vemos, usando tanto el algoritmo de Gillespie como la solución de la ecuación de Langevin se obtiene el mismo comportamiento llegando a un mismo estado estacionario, que se corresponde con el valor dado de la ecuación 3.13. En este caso, las tasas de infección y recuperación son $\lambda = 2$ y $\mu = 1$.

4.3. DIAGRAMA DE FASE

En este apartado estudiaremos el número medio de infectados frente a la tasa de infección, obteniendo así curvas similares a la de la figura 3.4. En las figuras 4.4(a) y 4.4(b) se encuentran dichas curvas.



(a) Gillespie.



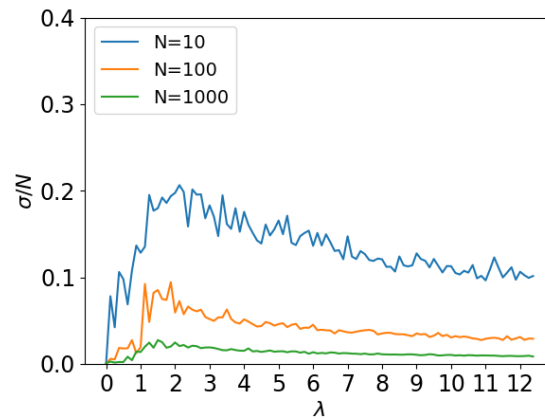
(b) Langevin.

Figura 4.4. Número medio de infectados frente a la tasa de infección para distintas poblaciones.

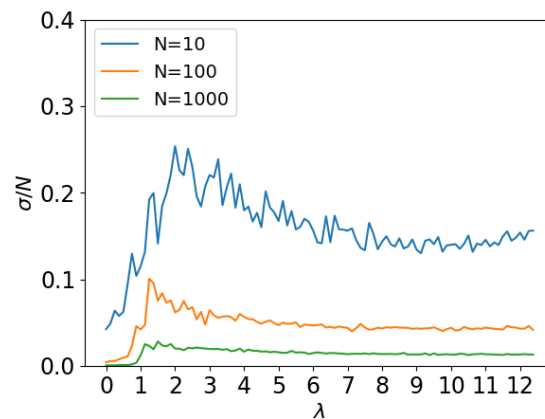
Como se puede comprobar, ambas figuras se ajustan bastante bien a lo esperado, si bien en el caso en el que la población es de 10 individuos esta curva es levemente distinta, ya que el cambio de estado de un individuo cambia el 10 % del sistema. A esto tenemos que sumarle que nuestra condición inicial para esta población es que un individuo está infectado, por lo que las propias fluctuaciones llevan a la extinción de infectados y nuestra simulación añade uno infectado para seguir ejecutándose, lo que provoca que el número medio de infectados sea no nulo en la zona absorbente.

4.4. DESVIACIÓN ESTANDAR

Es importante estudiar el comportamiento de las fluctuaciones a lo largo del diagrama de fase, ya que estas en el entorno del punto crítico podrán cambiar el sistema de la zona absorbente a la activa. A continuación, en la figura 4.5 se muestra el diagrama de fase para la desviación estándar.



(a) Gillespie.



(b) Langevin.

Figura 4.5. Desviación estándar frente a la tasa de infección para diferentes poblaciones.

Como se comprueba en las figuras, tanto el algoritmo de Gillespie como la resolución de la ecuación de Langevin proporcionan un comportamiento similar. Estas fluctuaciones son mayores en el entorno del punto crítico, tal y como se ha comentado en la sección 3.3, ya que es donde los procesos estocásticos toman mayor importancia. Por otro lado, se puede comprobar que en el caso de la solución de la ecuación de Langevin para diez individuos tenemos una pequeña sobreestimación de la desviación estándar,

pues para obtener esta, hemos supuesto la aproximación de grandes tamaños para llegar a la ecuación de Fokker-Planck y en este caso, la aproximación no es correcta. Podemos utilizar esta desviación estándar como las barras de error, pero hemos decidido no incluirlas en el diagrama de fase para no sobrecargar la imagen.

Conclusiones

Con la realización de este trabajo hemos podido iniciarnos en la teoría de procesos estocásticos, teoría fundamental en gran cantidad de ramas de la física. Hemos podido comprobar como podemos encontrar y/o aplicar la física en otras ramas del conocimiento, como es el caso de este trabajo y la epidemiología.

El uso de modelos matemáticos nos ha permitido caracterizar el comportamiento de las epidemias, algo imprescindible para evitar o minimizar los daños que estas causan en la sociedad. Para la resolución de las diferentes ecuaciones del modelo, como en la mayoría de problemas actuales; es necesario el conocimiento de lenguajes de programación, ya que cuando consideramos situaciones no ideales, la mayoría de problemas no tienen solución analítica.

Como punto de estudio principal, hemos estudiado los diagramas de fase bajo el modelo elegido. Obteniéndolo así gran cantidad de información para distintos valores de los parámetros como lo son el tamaño de la población, las condiciones iniciales o las tasas de interacción. Esto nos permitirá tomar medidas con suficiente antelación para disminuir lo máximo posible los efectos de epidemias que podamos modelar con las condiciones estudiadas.

Para finalizar, haremos una pequeña discusión sobre los dos métodos utilizados para obtener las trayectorias. Los resultados obtenidos son muy similares a los esperados, por lo que ambos métodos reproducen bien lo esperado, pero es interesante hacer una pequeña discusión sobre las ventajas e inconvenientes de sendos métodos.

Por una parte, con el algoritmo de Gillespie no realizamos ninguna aproximación, por lo que los resultados obtenidos serán buenos para cualquier valor de los diferentes parámetros de nuestro sistema. Un inconveniente de este método es el tiempo de cálculo. Para poblaciones de órdenes mayores a los mil individuos, el tiempo de simulación aumenta considerablemente. Además, para altas tasas de infección también aumenta el tiempo de simulación con el algoritmo de Gillespie.

Por otra parte, en el caso de la ecuación de Langevin tenemos la restricción de que debemos tener una población grande para poder aplicar la aproximación. Sin embargo, para poblaciones mayores al centenar de individuos esta aproximación es asumible, por lo que para la gran mayoría de casos de interés de nuestro modelo podemos aplicar esta aproximación. La principal ventaja de este método es el tiempo de simulación, el cual es prácticamente fijo, ya que nosotros elegimos el paso temporal para la resolución de la ecuación diferencial, por lo que al contrario del algoritmo de Gillespie, es independiente al tamaño de la población y de la tasa de infección.

En general, como se ha comentado, ambos son buenos métodos, pero debido a que en el caso que nos concierne, el estudio de las epidemias, es mejor utilizar la ecuación de Langevin, ya que se reduce considerablemente el tiempo de simulación obteniendo aun así resultados igual de satisfactorios que con el algoritmo de Gillespie.

Conclusions

With the completion of this work, we have been able to delve into the theory of stochastic processes, a fundamental theory in many branches of physics. We have verified how we can find and/or apply physics in other fields of knowledge, such as this work and epidemiology.

The use of mathematical models has allowed us to characterize the behavior of epidemics, something essential to prevent or minimize the damages they cause in society. To solve the different equations of the model, as in most current problems, knowledge of programming languages is necessary, because when we consider non-ideal situations, the majority of problems do not have an analytical solution.

As the main focus of study, we have examined phase diagrams, obtaining a wealth of information for different parameter values such as population size, initial conditions, or interaction rates. This will enable us to take actions to minimize the effects of epidemics that we can model with the studied conditions.

Finally, we will have a brief discussion on the two methods used to obtain the trajectories. The results obtained are very similar to the expected ones, so both methods accurately reproduce the expected outcomes. However, it is interesting to discuss the advantages and disadvantages of each method.

On one hand, with the Gillespie algorithm, we do not make any approximations, so the results obtained will be good for any value of the different parameters in our system. One drawback of this method is the computational time. For populations larger than a thousand individuals, the simulation time increases considerably. Additionally, for high infection rates, the simulation time also increases with the Gillespie algorithm.

On the other hand, in the case of the Langevin equation, we have the restriction that we need a large population to apply the approximation. However, for populations larger than a hundred individuals, this approximation is reasonable, so for the vast majority of cases of interest in our model, we can apply this approximation. The main advantage of this method is the simulation time, which is practically fixed since we choose the time step for the resolution of the differential equation. Thus, unlike the Gillespie algorithm, it is independent of the population size and infection rate.

In general, as mentioned, both methods are good, but due to the specific nature of our study of epidemics, it is better to use the Langevin equation. This significantly reduces the simulation time while still obtaining equally satisfactory results as with the Gillespie algorithm.

Bibliografía

- [1] Carlos Castañeda Gullot y Gerardo Ramos Serpa. "Principales pandemias en la historia de la humanidad". En: *Revista cubana de Pediatría* 92 (2020).
- [2] Gerardo Chowell et al. "Mathematical models to characterize early epidemic growth: A review". En: *Physics of life reviews* 18 (2016), págs. 66-97.
- [3] Daniel T Gillespie. "Stochastic simulation of chemical kinetics". En: *Annu. Rev. Phys. Chem.* 58 (2007), págs. 35-55.
- [4] Alan J McKane. *Stochastic Processes*. 2009.
- [5] Raúl Toral y Pere Colet. *Stochastic numerical methods: an introduction for students and scientists*. John Wiley & Sons, 2014.
- [6] Dirk P Kroese et al. "Why the Monte Carlo method is so important today". En: *Wiley Interdisciplinary Reviews: Computational Statistics* 6.6 (2014), págs. 386-392.
- [7] Claudio Castellano, Santo Fortunato y Vittorio Loreto. "Statistical physics of social dynamics". En: *Reviews of modern physics* 81.2 (2009), pág. 591.
- [8] Sandro Azaele et al. "Statistical mechanics of ecological systems: Neutral theory and beyond". En: *Reviews of Modern Physics* 88.3 (2016), pág. 035003.
- [9] Stephen K. Park y Keith W. Miller. "Random number generators: good ones are hard to find". En: *Communications of the ACM* 31.10 (1988), págs. 1192-1201.
- [10] Rizel Scarlett. "Why Python keeps growing, explained". En: (2023). URL: <https://github.blog/2023-03-02-why-python-keeps-growing-explained>.
- [11] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. "O'Reilly Media, Inc.", 2012.
- [12] Jake VanderPlas. *Python data science handbook: Essential tools for working with data*. "O'Reilly Media, Inc.", 2016.

- [13] Romualdo Pastor-Satorras et al. "Epidemic processes in complex networks". En: *Reviews of modern physics* 87.3 (2015), pág. 925.

Códigos

A continuación se muestran los códigos del algoritmo de Gillespie y de la resolución de la ecuación de Langevin.

```
1      def gillespie_SIS(infected, recovery_rate, infecting_rate, tmax):
2          time = np.array([0.])
3          inf = np.array([infected])
4          t = 0
5          while t < tmax and inf[-1] > 0:
6              r1 = infecting_rate * infected * (individuos-infected) /
individuos
7              r2 = recovery_rate * infected
8              r_total = r1 + r2
9              delta_t = np.random.exponential(scale=1/r_total)
10             prob = np.random.rand()
11             if prob < r1/r_total:
12                 infected += 1
13             else:
14                 infected -= 1
15                 if infected == 0:
16                     infected +=1
17             t += delta_t
18             time=np.append(time,t)
19             inf=np.append(inf,infected)
20         return time, inf
```

Code Listing 1. Algoritmo de Gillespie.

```

1  import numpy as np
2  import matplotlib.pyplot as plot
3
4  def langevin(deltat,x_0,individuos,infecting_rate,recovery_rate,tmax):
5      t=0.
6      time = np.array([0.])
7      x = np.array([x_0])
8      sqrt_deltat=np.sqrt(deltat)
9      while t < tmax:
10         f = x[-1] + deltat*(infecting_rate*x[-1]*(1-x[-1])-
recovery_rate*x[-1])+sqrt_deltat*np.sqrt((infecting_rate*x[-1]*(1-x[-1])+
recovery_rate*x[-1])/individuos)*np.random.normal()
11         if f<0:
12             f=1
13         x = np.append(x,f)
14         t += deltat
15         time = np.append(time,t)
16     return x, time

```

Code Listing 2. Resolución de la ecuación de Langevin.

Con el siguiente código podremos obtener los puntos para la representación de los diagramas de fases con y sin barras de error para el número medio de infectados y el diagrama de fase para la desviación estándar. De la misma forma podremos obtenerlo utilizando la función langevin.

```
1 puntos = 100
2 for j in range (3):
3     medias=np.array([]) #Vector que contendrá las medias
4     inf_rates=np.array([])#Vector que contendrá los lambdas
5     desviacion=np.array([])
6     individuos = 10**(j+1)
7     infectados_inicio = individuos/10
8     tmax = 100
9     for i in range(puntos):
10         time, inf= gillespie_SIS(infectados_inicio, recovery_rate,
11 0.+0.125*i,tmax)
12         inf_est=np.array(inf[int(np.ceil(np.size(inf)/2)):np.size(inf)
13 :1])
14         medias=np.append(medias,[np.mean(inf_est)/individuos])
15         desviacion=np.append(desviacion,[np.std(inf_est)/individuos])
16         inf_rates=np.append(inf_rates,0.+0.125*i)
17         print(np.size(inf_rates))
18 plot.plot(inf_rates,medias,'-', label='N=%d'%(individuos))
19 plot.plot(inf_rates,desviacion,'-', label='N=%d'%(individuos))
20 plot.errorbar(inf_rates, medias, yerr=desviacion, label='N=%d'%(
21 individuos), fmt = '-', ecolor='k', capsize=3, capthick=1)
```

Con el siguiente código podremos obtener los datos las trayectorias temporales para diferentes condiciones iniciales y las series temporales. De igual manera que anteriormente, se pueden obtener utilizando la función `gillespie_SIS`

```
1 deltat=0.1
2 x_0=0.1
3 individuos=1000
4 infecting_rate=2
5 recovery_rate=1
6 tmax=50
7 trayectorias=5
8 #Condiciones iniciales
9 for j in range(trayectorias) :
10     x_0=0.1+0.2*j
11     x , time = langevin(deltat,x_0,individuos,infecting_rate,
12         recovery_rate,tmax)
13     plot.plot(time,x,'-',label=f"$x_0= %.2f"%(x_0))
14 #Series temporales
15 for j in range(trayectorias) :
16     infecting_rate=0.5+0.5*j
17     x , time = langevin(deltat,x_0,individuos,infecting_rate,
18         recovery_rate,tmax)
19     plot.plot(time,x,'-',label=f"$ \lambda= %.2f"%(infecting_rate))
```