Desinfección mediante caminantes aleatorios

Aldo Sayeg Pasos Trejo

29 de septiembre de 2021

Índice

1. Simulacion de desinfección

La simulación que realizamos puede primero describirse en términos más simples mediante el siguiente algoritmo:

2. Posiciones del virus como resultado de la simulación de un estornudo

Las posiciones iniciales del virus se generan a través de la solución a un sistema de ecuaciones diferenciales, correspondiente al tiro parabólico en 3D con fricción lineal en la velocidad. Explícitamente, el sistema de ecuaciones

Algorithm 1 Simulacion de desinfeccion

function DESINFECCIÓN(n_g,n_v,t_s,RadioDecrece)
  Genera posiciones iniciales para n_v particulas de virus
  Genera posiciones iniciales para n_g particulas de gel
  for $i = 1, \ldots, t_s$ **do**
    for $j = 1, \ldots, n_g$ **do**
      Avanza la j-ésima partícula de gel en una dirección aleatoria
      Checa cuantas de las n_v partículas de virus están a distancia r_j de la j-ésima partícula de gel.
      Aumenta el número de visitas la partículas de virus que estén dentro del radio r_j
      if RadioDecrece **then**
        Disminuye r_j
      end if
    end for
  end for
end function

es:

$$\begin{cases} \ddot{x} + \dfrac{3\pi D\mu}{6}\dot{x} = 0 \\ \ddot{y} + \dfrac{3\pi D\mu}{6}\dot{y} = 0 \\ \ddot{z} + \dfrac{3\pi D\mu}{6}\dot{z} + g = 0 \end{cases} \tag{1}$$

Las condiciones iniciales para el sistema de EDOs corresponden a

$$\begin{gathered} (x, y, z) = (0, 0, h) \\ (\dot{x}, \dot{y}, \dot{z}) = (R\sin(\Theta)\cos(\Phi), R\sin(\Theta)\sin(\Phi), R\cos(\Theta)) \end{gathered} \tag{2}$$

$$\begin{gathered} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix} \\ \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} R\sin(\Theta)\cos(\Phi) \\ R\sin(\Theta)\sin(\Phi) \\ R\cos(\Theta) \end{pmatrix} \end{gathered} \tag{3}$$

con h la altura de una persona que queramos simular, R (la magnitud de la velocidad inicial) es una variable aleatoria tomada uniformemente en el intervalo $[2, 3]$, Θ el angulo polar tomado uniformemente en el intervalo

$[\pi/8, \pi/4]$ y Φ es el angulo azimutal, tomado uniformemente en el intervalo $[-\pi/4, \pi/4]$.

Para obtener la posición inicial de una partícula de virus, resolvemos el sistema 1 con condiciones iniciales 3. La posición inicial es el valor $(y(t^*), z(t^*))$ donde t^* es el primer tiempo discreto a partir del cual $x(t^*) > 1.0$ y $z(t^*) > 0.5$

3. Caminantes aleatorios y movimiento browniniano

Una variable aleatoria es una función $X : \Omega \to \mathbb{R}$. En palabras simples, es un posible número real que toma uno de varios valores posibles (inclusive una infinidad), cada uno con distinta probabilidad. Ejemplos de variables aleatorias son:

- El resultado de tirar un dado.
- El resultado de tirar un volado codificamos si "águila" como 0 y "sol" como 1.
- El resultado de tomar un punto al azar en un círculo.
- El resultado de tomar un punto al azar en una recta.

Una cosa importante que deben distinguir es que la variable aleatoria es el posible número con todos sus valores, no es el resultado en sí. Un ejemplo sencillo es el siguiente: si tiran un dado y cae un número 6, el número 6 **no es un variable aleatoria**, la variable aleatoria es **el posible resultado de tirar un dado**. Número 6, es decir, al resultado en sí de tirar el dado, le decimos **muestra** de la variable aleatoria de tirar el dado.

Cuando tenemos muchas variables aleatorias $X_1, X_2, \ldots$, es decir, tenemos un conjunto $\{X_n \mid n \in \mathbb{N}\}$, decimos que tenemos un **proceso estocástico**. Ya que $n \in \mathbb{N}$, en particular decimos que tenemos un proceso estocástico **a tiempo discreto** o a pasos discretos. Uno de los procesos estocásticos más sencillos es la **caminata aleatoria 1D**. Una caminata aleatoria 1D se define de forma recursiva como:

$$X_0 = 0$$
$$X_n = \begin{cases} X_{n-1} + 1 & \text{con probabilidad } 1/2 \\ X_{n-1} - 1 & \text{con probabilidad } 1/2 \end{cases} \tag{4}$$

La siguiente figura ilustra un conjunto de 10 caminatas aleatorias independientes. En general, no es importante que el punto inicial de una caminata

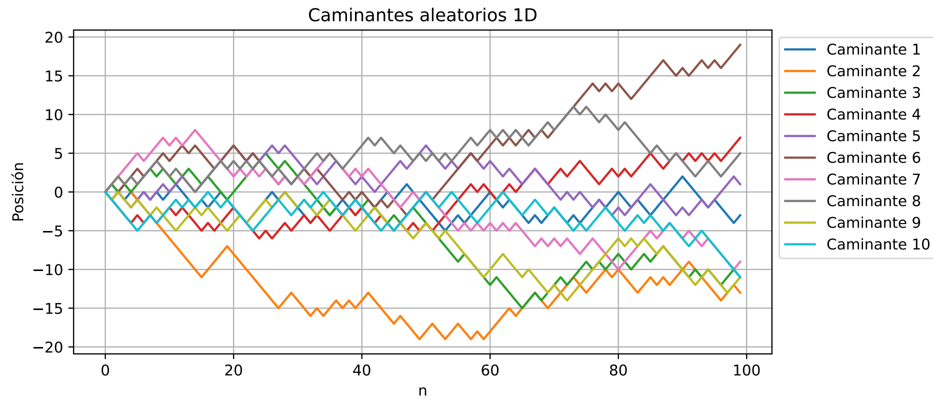

Figura 1: 10 caminantes aleatorios independientes

aleatoria sea 0. Basta con que sea un punto constante cualquiera. La caminata aleatoria se puede extender de forma trivial a varias dimensiones. Aunque hay distintas formas de extenderla, nosotros en la simulación hacemos una caminata aleatoria en $\mathbb{R}^2$ de la siguiente forma:

- El punto inicial es un punto (X_0, Y_0) escogido al azar de forma uniforme en el rectángulo más pequeño que contiene a todas las partículas de virus.

- En cada paso, hacemos $(X_{n+1}, Y_{n+1}) = (X_n, Y_n) + (R\cos(\Theta), R\sin(\Theta))$ donde R es una variable aleatoria uniforme en el intervalo $[0, 0.05]$ y Θ un ángulo aleatorio en el intervalo $[0, 2\pi)$

4. Búsquedas de vecinos: árboles KD

En cada paso de la simulación se realiza el siguiente procedimiento: para cada partícula de gel, con un radio dado r, buscamos cuántas partículas de virus se encuentran a una distancia r de ella. Si tenemos n_g partículas de gel y n_v partículas de virus, entonces necesitaríamos realizar $n_g \times n_v$ operaciones de búsqueda para contabilizar cuántas de las n_v partículas de virus están en contacto con las m_g partículas de gel.

Sin embargo, podemos resolver esto de forma más rápida si en lugar de buscar si cada partícula de virus está en el radio de cada partícula de gel, utilizamos un árbol KD [1] sobre las posiciones de los viruses para poder buscar entonces, dada una particula de gel y su radio, todas las partículas de virus a una distancia menor a r de la partícula de gel.

Un árbol KD es una **estructura de datos** (una forma ingeniosa de representar información en una computadora) que permite, dado un conjunto fijo de n puntos en $\mathbb{R}^d$, buscar los puntos más cercanos a otro. La estructura permite que encontremos todos el punto más cercano del conjunto a cualquier otro con tan solo $\log(n)$ operaciones, a diferencia de las usuales n.

5. Consideraciones extra

- Para darle más realismo, una posibilidad es tomar los radios no iguales para todas las partículas de gel si no muestrear los radios de una variable aleatoria normal (gaussiana) con promedio distinto de 0. En las animaciones, el radio promedio es 0.05.
- También es posible, en una metáfora directa con la evaporación del alcohol, hacer una simulación en la cuál los radios de las partículas se reduzcan linealmente hasta volverse 0 en el paso final de la simulación.
- Las partículas de virus pueden tener distintos perfiles de vitalidad, es decir, la forma en la que dejan de funcionar como función del número de visitas de partículas de gel que han tenido. La figura 2 muestra perfiles exponenciales, lineales y logarítmos para la vitalidad de las partículas.

6. Simulación revisada

Con toda la información que acabamos de mencionar, podemos volver a escribir el código de la simulación.

7. Trabajo a futuro

Hay distintas cosas que se podrían realizar para hacer más realista la simulación

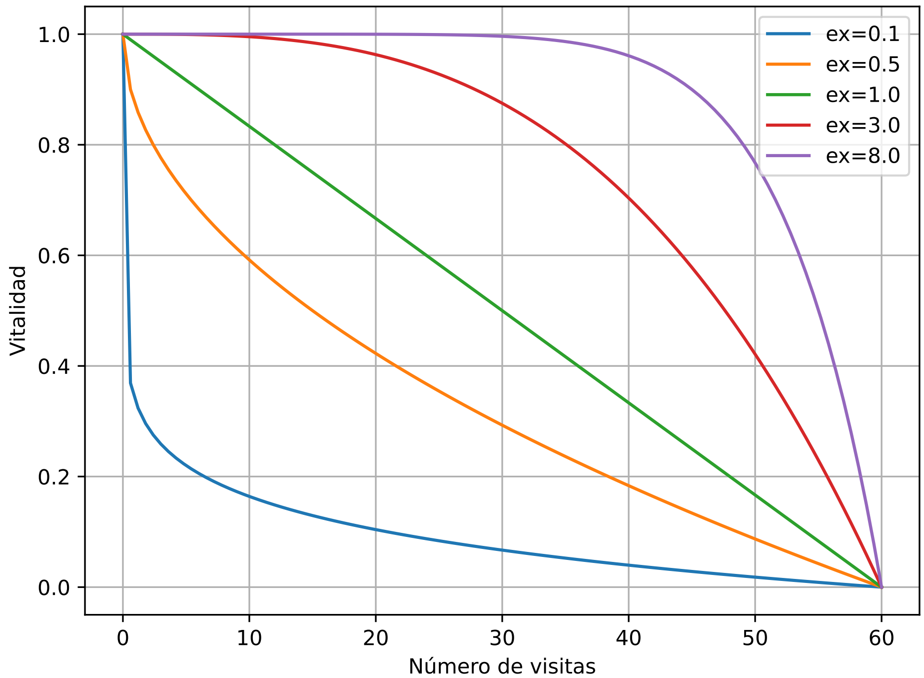

Figura 2: Distintos perfiles de vitalidad para las partículas de virus. En todos los casos, los virus mueren cuando son visitados 60 veces

- Evitar que se empalmen o se encimen las partículas de gel. Esto consiste en modelarlas como discos duros para hacer más realista su movimiento.
- Añadir una fuerza electrostática repulsiva entre las partículas de Gel. Esto evitara que se encimen además de que provocará que se expandan una de la otra. Así la simulación será realista para un limpiador basado en partículas con carga.
- Dejar de modelar las partículas de virus como puntuales y darles mayor movilidad.

Referencias

[1] Árbol kd - Wikipedia, la Enciclopedia Libre. `https://es.wikipedia.org/wiki/%C3%81rbol_kd`. Accesado el 25-09-2021

”