AGUJAS DE BUFFON

Arturo Hesiquio Casarrubias
Erick Gabriel Vega Cervantes
Israel Santiago Gómez

OBJETIVOS

Objetivo General

Desarrollar una Simulación Precisa: Crear un programa de simulación que estime el valor de π utilizando el problema de las agujas de Buffon. Asegurarse de que la simulación sea eficiente y pueda manejar un gran número de agujas para obtener estimaciones precisas.

Objetivos Específicos

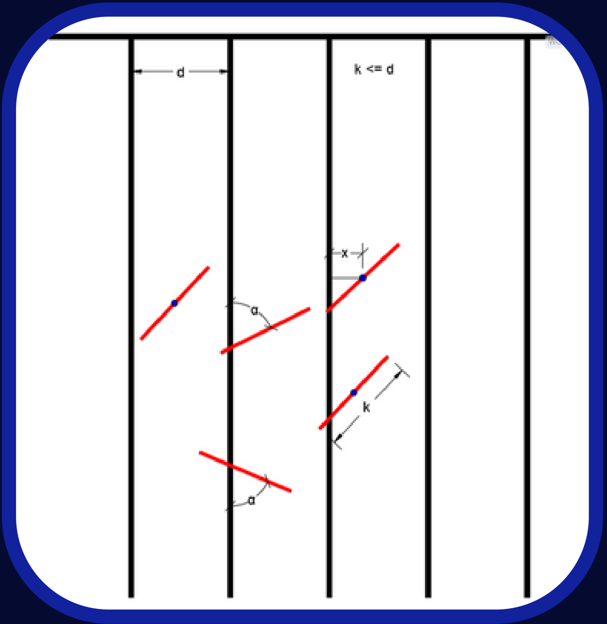
Análisis de Resultados: Realizar un análisis estadístico de los resultados de la simulación, incluyendo la probabilidad estimada de que una aguja cruce una línea y la estimación de π.

Visualización de Datos: Generar gráficos y visualizaciones que muestren la relación entre el número de agujas y la estimación de π. Esto ayudará a comprender mejor cómo la precisión se ve afectada por el tamaño de la muestra.

¿DE QUÉ TRATA EL PROBLEMA?

El problema de la aguja de Buffon es un famoso problema geométrico y estadístico que lleva el nombre del matemático francés Georges-Louis Leclerc, conde de Buffon. Este problema se planteó por primera vez en el siglo XVIII y se ha convertido en un clásico en la teoría de la probabilidad.

Se trata de lanzar una aguja sobre un papel en el que se han trazado rectas paralelas distanciadas entre sí de manera uniforme. Se puede demostrar que si la distancia entre las rectas es igual a la longitud de la aguja, la probabilidad de que la aguja cruce alguna de las líneas es:



$$P = \frac{2}{\pi}$$

De esta manera:

$$\pi \approx \frac{2N}{A}$$

HIPÓTESIS

Al aumentar el número de agujas lanzadas en el experimento de las agujas de Buffon, la estimación de π se acercará más al valor real de π y mostrará una mayor precisión. Se espera que la convergencia de las estimaciones sea más cercana a medida que el tamaño de la muestra se vuelve más grande, lo que se traducirá en una disminución de la variabilidad de las estimaciones.

LIMITACIONES

1. Variabilidad Estadística: La estimación de π es probabilística y su precisión varía con la cantidad de agujas lanzadas.
2. Dependencia de Parámetros: Los valores de k y d afectan significativamente la precisión de la estimación de π.
3. Limitaciones en Precisión Práctica: No es adecuado para cálculos altamente precisos de π debido a la variabilidad y la necesidad de recursos significativos.
4. Sesgo Aleatorio: La estimación de π puede ser sesgada debido a la aleatoriedad en la simulación.
5. Sensibilidad a Parámetros: Los valores de k y d son críticos y su elección afecta la precisión.
6. Limitado a Experimentos Específicos: Solo es aplicable a problemas que siguen el modelo de agujas lanzadas sobre líneas paralelas, limitando su alcance a problemas más generales.

PARÁMETROS

k: La longitud de las agujas lanzadas.

d: El espaciado entre las líneas verticales en la superficie sobre la que se lanzan las agujas.

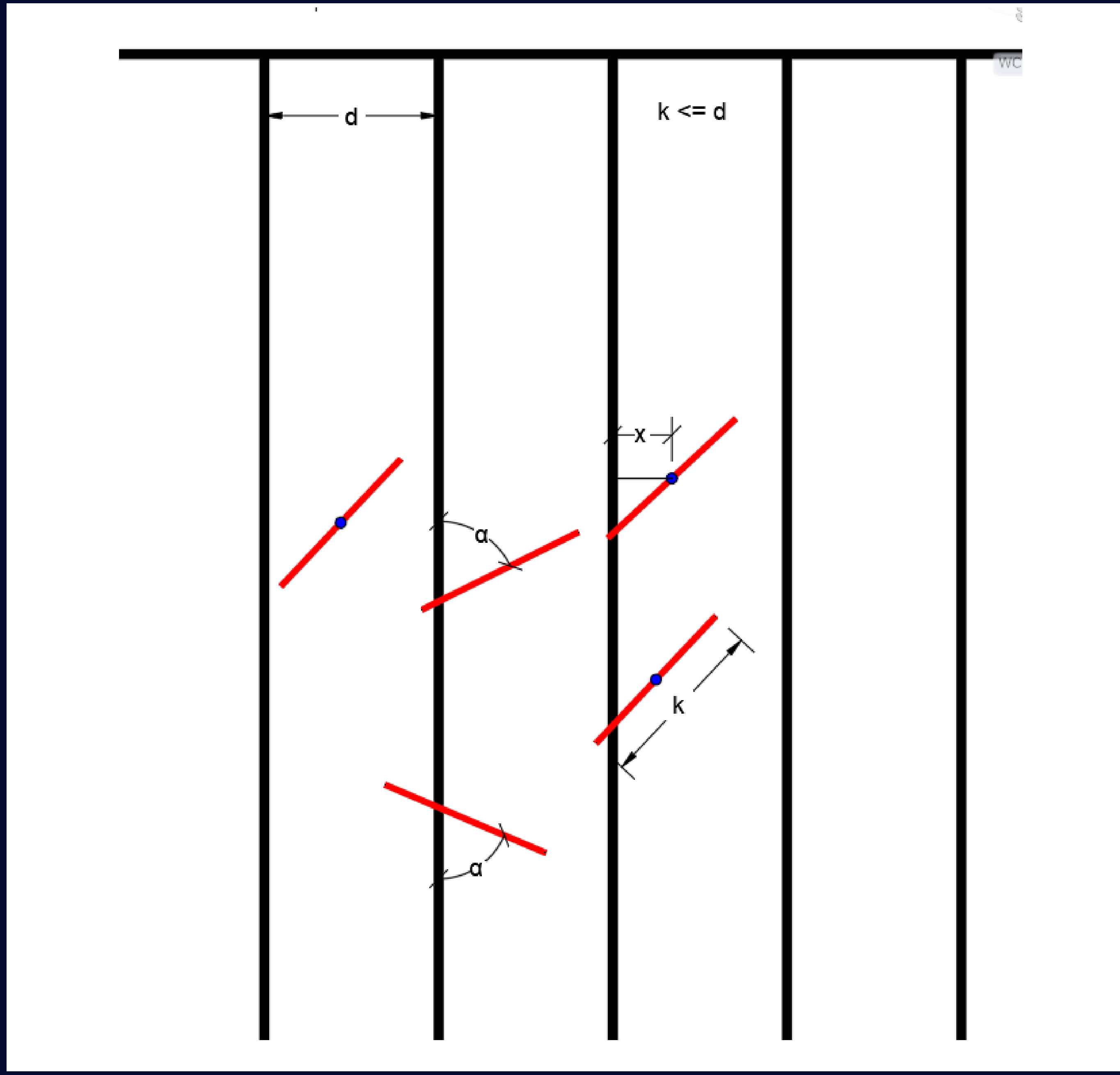
Estos son los dos parámetros principales que afectan directamente la simulación y la estimación de π en el problema. Las estimaciones de π se basan en la combinación de estos parámetros con los resultados del experimento. Otros valores numéricos en el código son constantes o límites para la simulación y no son parámetros en sí.

DATOS

1. Lineas paralelas
2. Distancia entre las lineas
3. Longitud de la aguja
4. Numero de simulaciones

SOLUCIÓN

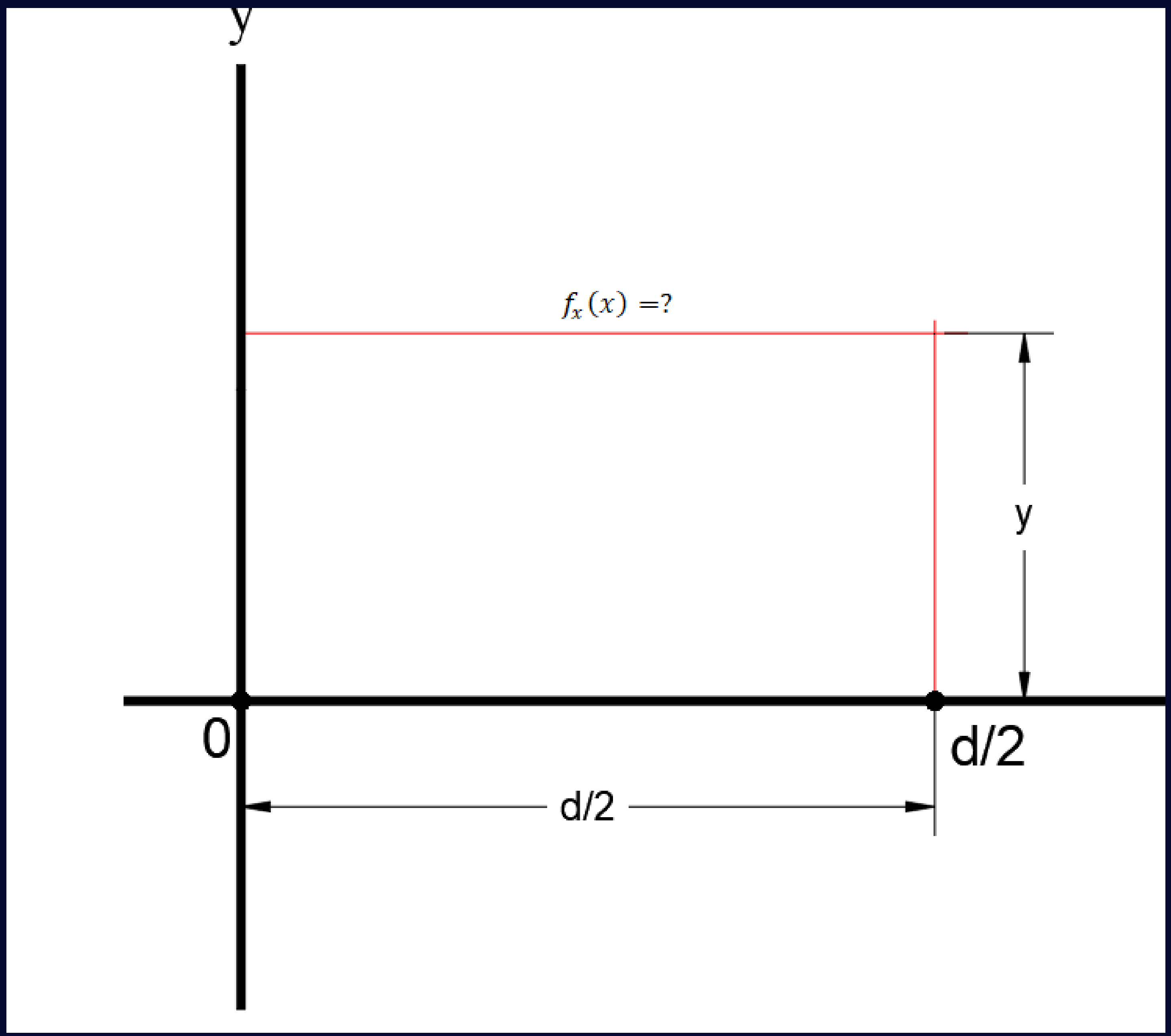
Sea una aguja de longitud k lanzada sobre un plano segmentado por líneas paralelas separadas d, unidades. ¿Cuál es la probabilidad que la aguja cruce alguna línea?



- Sabiendo esto podemos definir lo siguiente:
- Si x es la distancia entre el centro de la aguja y la línea más cercana, $x \in [0, d/2]$, dicho de otra manera: $0 \leq x \leq d2$
- Siendo α el ángulo entre la aguja y las líneas, $\alpha \in [0, \pi/2]$, dicho de otra manera: $0 \leq \alpha \leq \pi 2$.

SOLUCIÓN

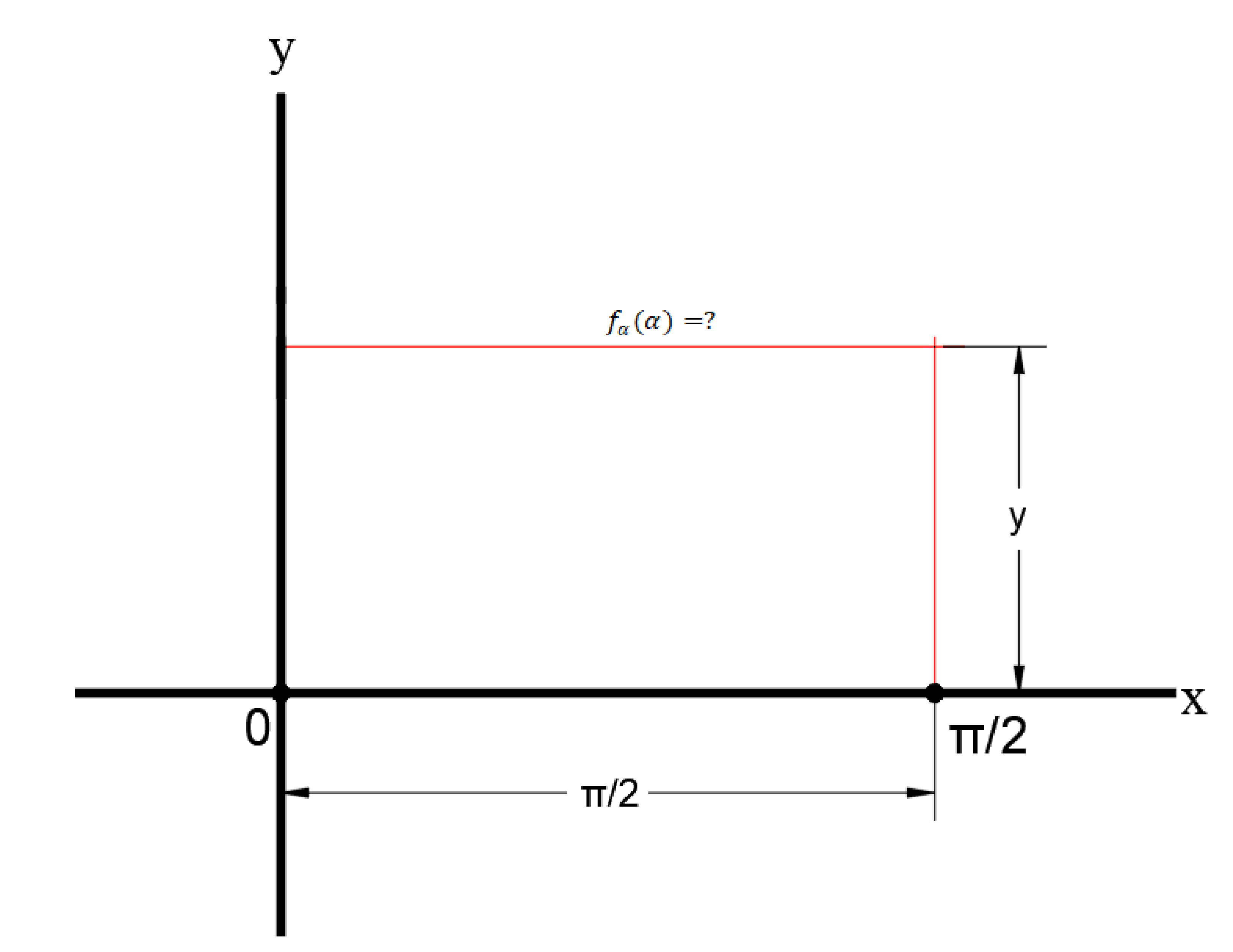
- Ahora procederemos a generar las funciones de densidades de las variables anteriormente descritas:
- Sabemos que x toma valores entre 0 y $d/2$ por lo tanto la funcion de densidad de manera grafica nos queda de la siguiente manera:



- Si sabemos que si es una funcion de densidad el area encerrada vale 1, por lo tanto:
- $d2 * y = 1$
- Despejando y:
- $y = 2d$
- Por lo tanto la funcion de densidad queda como:
- $fx(x) = 2d$

SOLUCIÓN

- Procediendo de la misma manera para α:



- $\pi 2 * y = 1$
- Despejando y:
- $y = 2\pi$
- Por lo tanto la funcion de densidad queda como:

- $f\alpha(\alpha) = 2\pi$

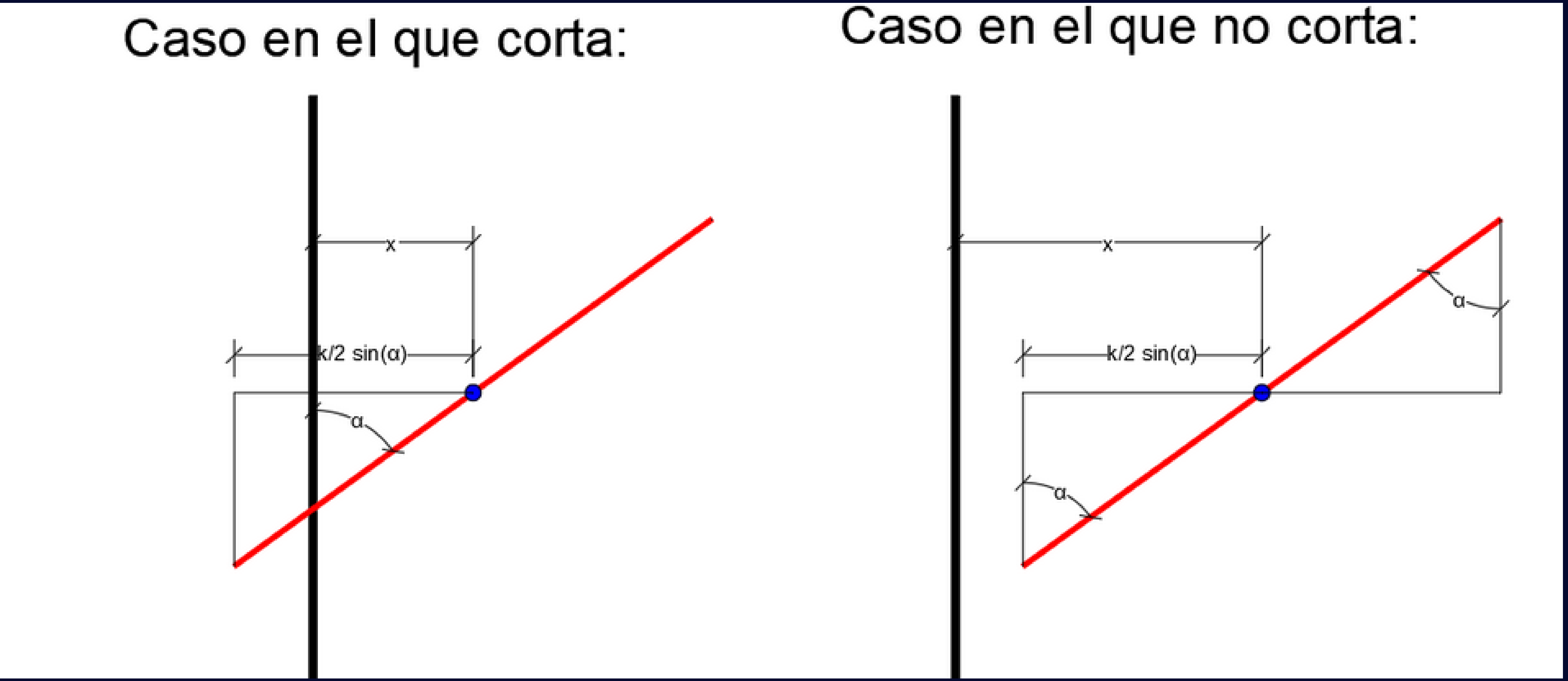
- Ahora calculamos la funcion de densidad conjunta:
- Al ser x y α variables aleatorias independientes la funcion de densidad conjunta se puede determinar de la siguiente manera:
- $fx,\alpha(x,\alpha) = fx * f\alpha$

- $fx,\alpha(x,\alpha) = 2d * 2\pi = 4d\pi$

SOLUCIÓN

- Ahora calculamos la funcion de densidad conjunta:
- Al ser x y α variables aleatorias independientes la funcion de densidad conjunta se puede determinar de la siguiente manera:
- $fx,\alpha(x,\alpha)=fx*f\alpha$

- $fx,\alpha(x,\alpha)=2d*2\pi=4d\pi$
- La probabilidad se define como la integral dentro del dominio de la funcion de densidad, sin embargo aun nos falta determinal el dominio de fx el cual determinaremos con trigonometria



SOLUCIÓN

- Por lo tanto el dominio de x
- para cuando una aguja corta la linea es el siguiente:
- $0 \leq x \leq k2\sin\alpha$

- Ahora ya podemos resolver la integral de probabilidad

$$P = \int_0^{\frac{\pi}{2}} \int_0^{\frac{k}{2}\sin\alpha} \frac{4}{d\pi}\, dx\, d\alpha$$

- Resolviendo la integral nos queda de la siguiente manera:

$$\frac{2k}{\pi d}$$

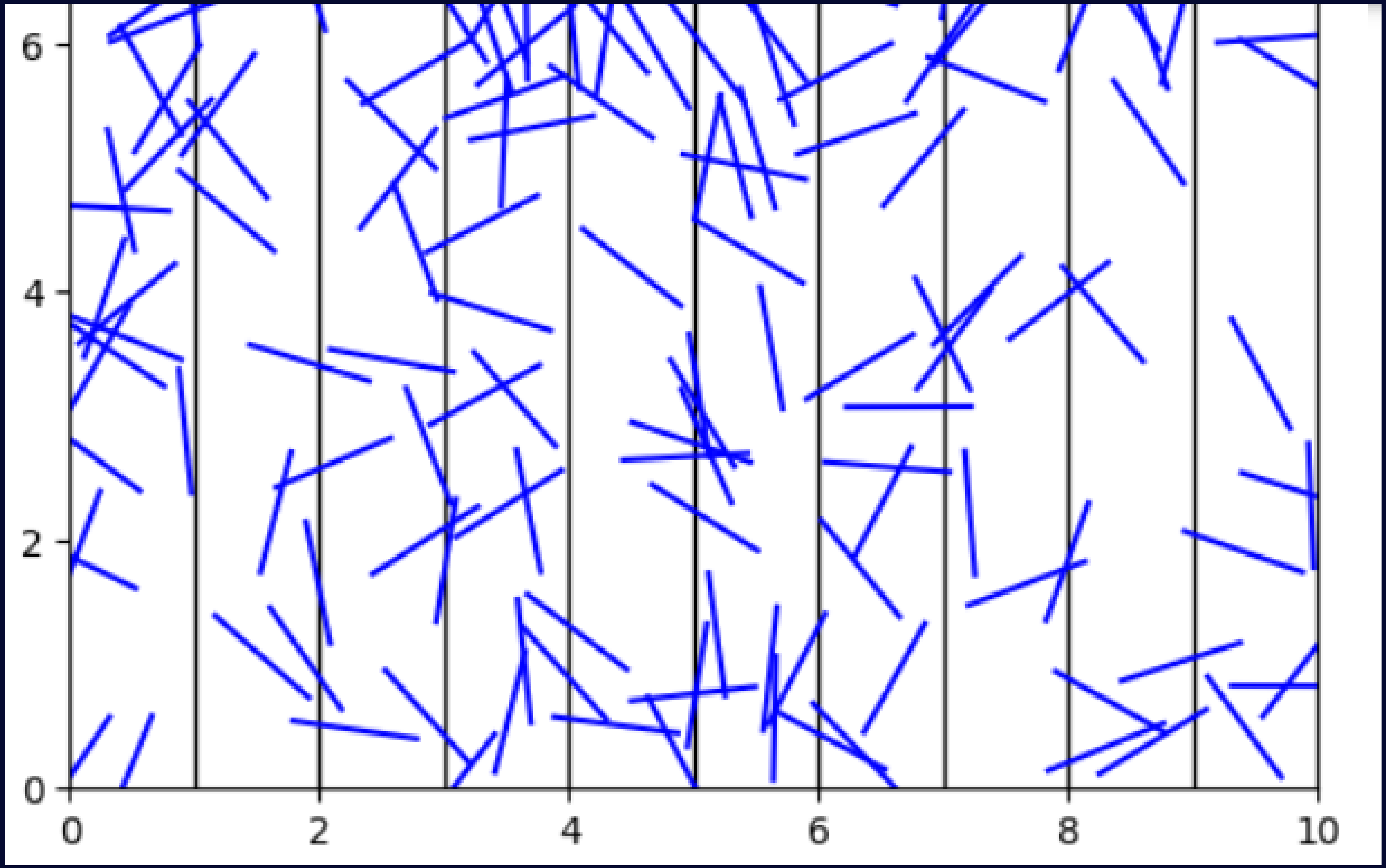
$$P = \frac{2k}{\pi d}$$

CÓDIGO

```
import matplotlib.pyplot as plt
import random
import math

plt.figure(figsize=(6, 6))
plt.vlines(range(11), ymin=0, ymax=10, colors='black', linewidth=1)

lineas_verticales = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
k = 1  # Longitud de las agujas
d = 1  # Espaciemiento entre líneas verticales

plt.xlim(0, 10)
plt.ylim(0, 10)

# Contadores para el número total de agujas y el número de agujas que cruzan líneas
total_agujas = 0
aguja_cruza_linea = 0

def generar_aguja():
    x = random.uniform(0, 10)
    y = random.uniform(0, 10)
    angulo = random.uniform(0, math.pi)
    return x, y, angulo

def dibujar_aguja(x, y, angulo):
    x1 = x - (1/2) * math.cos(angulo)
    x2 = x + (1/2) * math.cos(angulo)
    y1 = y - (1/2) * math.sin(angulo)
    y2 = y + (1/2) * math.sin(angulo)
    plt.plot([x1, x2], [y1, y2], 'b-')
```

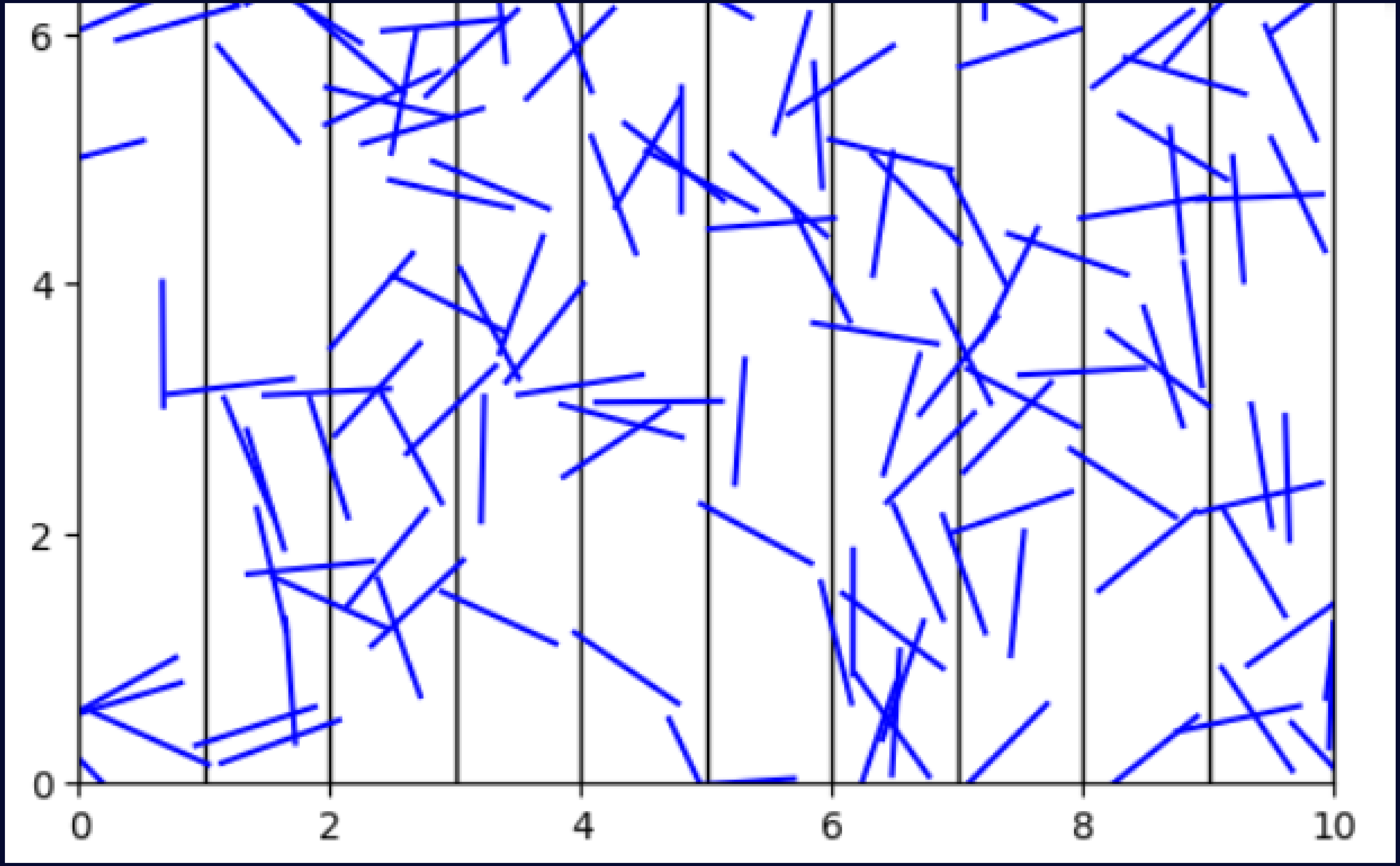
```
# Verificar si la aguja cruza una línea vertical
    global aguja_cruza_linea
    x_min = min(x1, x2)
    x_max = max(x1, x2)
    for linea in lineas_verticales:
        if x_min <= linea <= x_max:
            aguja_cruza_linea += 1

num_agujas = 200

for i in range(num_agujas):
    x, y, angulo = generar_aguja()
    dibujar_aguja(x, y, angulo)
    total_agujas += 1

pi_estimado = (2 * k * total_agujas) / (d * aguja_cruza_linea)
print("Estimación de π:", pi_estimado)


plt.show()
```

RESULTADOS



Estimación de π: 3.1496062992125986

RESULTADOS



Estimación de π: 3.278688524590164

CONCLSUIÓN

A través de este proyecto, obtuvimos una comprensión profunda de la geometría, la probabilidad y las técnicas de programación mientras planteamos un problema clásico que desafía nuestra comprensión de la estadística y la teoría de la probabilidad.

El desarrollo de este programa no sólo nos permite aplicar conceptos matemáticos a entornos del mundo real, sino que también nos enseña cómo diseñar algoritmos eficientes y realizar simulaciones numéricas utilizando herramientas de programación. Además, al compartir y discutir los resultados con otros estudiantes y profesionales, ampliamos nuestra comprensión y aprendizaje colaborativo.

En última instancia, este proyecto nos recuerda la importancia de cómo la informática y las matemáticas convergen para resolver problemas del mundo real