# Tarea 1: De la regla 1/2 al método bunching

Sofía Thiel Pizarro
B36953

26 de febrero de 2021

## 1. Muestreo Directo

### 1.1 A1

Al correr el código del enunciado se obtiene la siguiente gráfica:

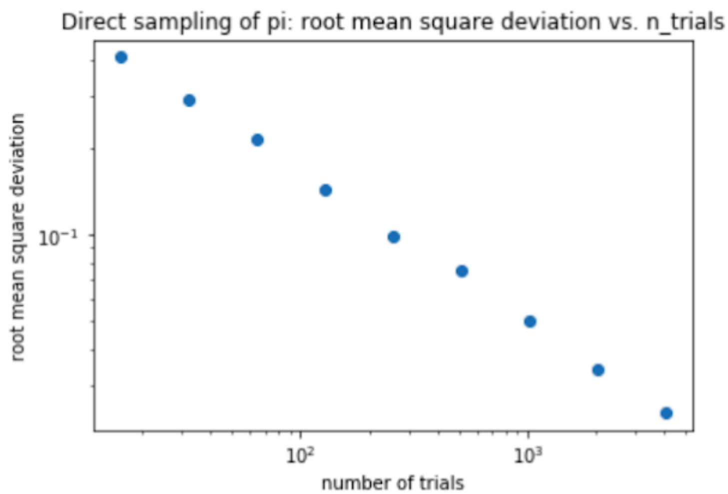


Figura 1: Muestreo Directo

### 1.2 A2

Graficando ahora también la función $\frac{1,642}{\sqrt{N_{trials}}}$ se obtiene la siguiente gráfica:

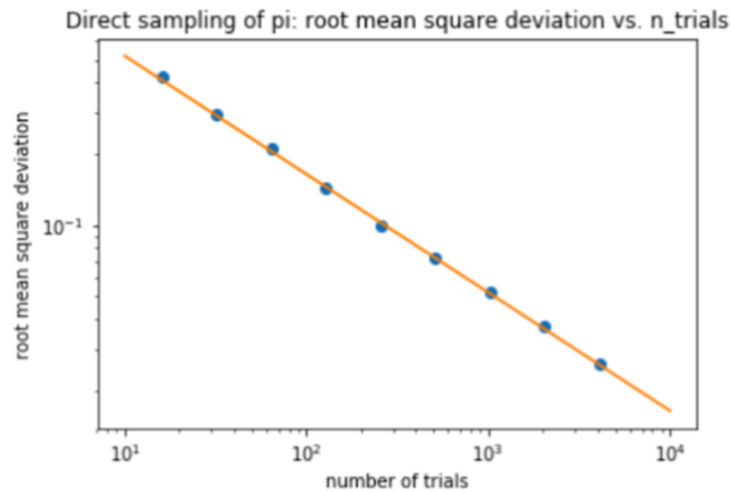


Figura 2: Gráfica que proyecta el muestreo directo y la función $\frac{1{,}642}{\sqrt{N_{trials}}}$

De la gráfica 2 se puede notar que la recta de color naranja se ajusta perfectamente con los datos obtenidos por el método de monte carlo de la pregunta 1.1 A1.

- La línea de código utilizada para graficar la recta naranja se colocó en la sección de pylab antes de pylab.show() del código. Se utiliza así para tener un orden, asimismo para que python, en este caso, lea el código correctamente y lo interprete línea a línea.

- Sabemos que el error de muestreo de datos estadísticos está dado por la desviación estandar entre la raíz del número de veces que se realiza el muestreo o en otras palabras la variancia de la media. Por lo que para el error del cálculo de pi la desviación estandar tiene un valor de 1,642 y como se observa en la gráfica la función se ajusta perfectamente a los datos experimentales y la dispersión de los datos es muy pequeña, lo cuál es lo esperado.

  Al tomar el límite de N obtenemos:
  $$\lim_{x\to\infty} \frac{1,642}{\sqrt{N}} = 0$$

  Al aumentar el número de intentos se hace más grande la probabilidad de que el estimado de pi se acerque a su valor exacto y el error disminuya.

# 2. Cadenas de Markov

## 2.1 B1

En esta sección se utiliza el mismo propósito de montecarlo pero por medio de cadenas de Markov. Se espera un mayor error que por el método de muestreo directo ya que las cadenas de Markov presentan un factor de correlación. En la gráfica 3 se observa el resultado para diferentes deltas, donde: azul $\delta = 0{,}062$, naranja $\delta = 0,125$, café $\delta = 4{,}0$, verde $\delta = 0,25$, morado $\delta = 2{,}0$, rojo $\delta = 1{,}0$

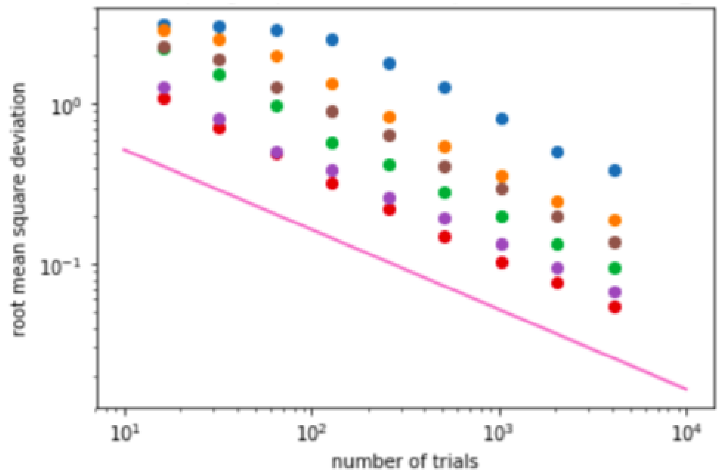


Figura 3: Gráfica para $\delta = 0{,}062$, $\delta = 0{,}125$, $\delta = 0{,}25$, $\delta = 1{,}0$, $\delta = 2{,}0$, $\delta = 4{,}0$ junto con la recta de error obtenida en la sección anterior por método directo.

- Los valores de delta más precisos son verde $\delta = 0{,}25$ y rojo $\delta = 1{,}0$. Como se puede ver en la gráfica 3 son los que están más rectos.
- Los valores para delta muy grande o muy pequeño dan resultados menos precisos ya que lo que se está haciendo es escoger una posición fija dentro del cuadrado o área de interés y de ahí dar pasos grandes o pequeños en direcciones aleatoreas. Si el lado del cuadrado mide uno entonces los pasos que sean más grandes que uno son más probables que se salgan del área que se definió, es como si se quedara en el mismo lugar y al volver a dar un paso la probabilidad aumenta de que salga del área, por ende aumenta el error. Para pasos muy pequeños es como si se quedara en el mismo lugar y pasa lo mismo que en el caso anterior donde el error aumenta. Por esto los valores más precisos son los cercanos a 1.0 y así se presenta en la gráfica 3.
- El error aumenta con este método ya que existen condiciones iniciales en el sistema que no hay con el método directo, como por ejemplo se necesita de una posición inicial o un valor para el paso, entre otras. El hecho de que no hayan condiciones iniciales hace que el programa no almacene tanta información, por lo tanto la convergencia del método es mucho más rápida. Asimismo, otra situación que hace que aumente el error es que cuando el paso se sale del área estudiada es como si se quedara en la misma posición como se ha comentado y los pasos se agrupen provocando que la desviación estandar aumente.

## 3. Regla 1/2

| Delta | Tasa de aceptación |
|---|---|
| 0.062 | 0.878 |
| 0.125 | 0.952 |
| 0.25 | 0.626 |
| 1.0 | 0.556 |
| 2.0 | 0.195 |
| 4.0 | 0.713 |

Cuadro 1: Tabla de la tasa de aceptación para distintos valores de delta

Las tasas de aceptación obtenidas concuerdan con lo mencionado en las secciones anteriores. Según se observa en la tabla las tasas de aceptación de los valores 0.25 y 1.0 se acercan a un medio. Lo cual propone que el error de estos valores es menor.

# 4. Método Bunching

## 4.1 C1

Para esta sección se obtuvo 3.1455 como valor aproximado de $\pi$ y 1.6432 como el valor de la desviación estandar. El valor aproximado de $\pi$ que se obtuvo es bastante cercano al valor teórico que se conoce.

Al inicio de la tarea se toma la curva de error como $\frac{1{,}642}{\sqrt{N}}$ donde el término de 1,642 representa la desviación estandar y comparandolo con el valor obtenido se nota que es bastante cercano a lo esperado.

## 4.2 C2

Se muestra el código modificado para obtener la raíz cuadrada de la variancia:

```
import random, math

n_trials = 40000
n_hits = 0
var=0.0
var1=0.0
for iter in range(n_trials):
    x, y = random.uniform(-1.0, 1.0), random.uniform(-1.0, 1.0)
    obs=0.0
    if x**2 + y**2 < 1.0:
        n_hits += 1
        obs = 4.0
    var1 += obs
    var +=(obs)**(2)
var2 = ((var/n_trials)-(var1/n_trials)**2 )

print(math.sqrt(var2))
```

El valor obtenido de la raíz cuadrada de la variancia es de: 1.6421

## 4.3 C3

El valor estimado de $\pi$ obtenido en esta sección es de 3.1511. Si realizamos $3{,}1511 - \pi$ obtenemos 0.0096, se puede notar de la gráfica que el error aparente hace plateau en un valor cercano a este lo que valida una gran estimación del verdadero error. Asimismo, podemos notar que el valor estimado es bastante cercano al valor teórico de $\pi$.

La gráfica 4 muestra perfectamente el plateau, que es donde el error se minimiza y donde ocurre una convergencia de los valores. Al aumentar las iteraciones el error aparente se esparce. En el bunching los datos no están correlacionados por lo que el error aparente tiende al error real para un número alto de iteraciones.

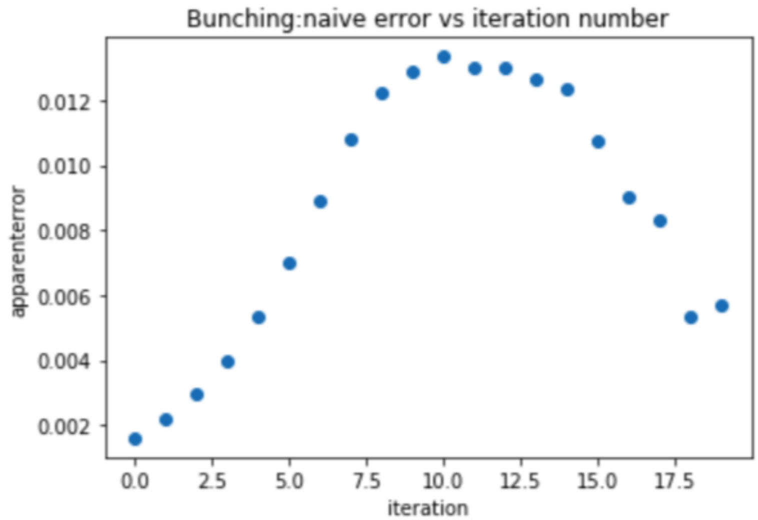


Figura 4: Error aparente del método de bunching