

December 10, 2024

```
[7]: import numpy as np
import math
from decimal import getcontext

from scipy.stats import norm

N = 10_000_00
x = np.random.rand(N)
fx = 4 / (1 + x**2)
estimate_a = np.mean(fx)

print("Monte Carlo estimate result (a):", estimate_a)
print("Theoretical value:", np.pi)

def integrand(x):
    return np.sqrt(x + np.sqrt(x + np.sqrt(x + np.sqrt(x))))

x = np.random.uniform(0, 4, N)
fx = integrand(x)
estimate_b = (4 - 0) * np.mean(fx)

print("Monte Carlo estimate result (b):", estimate_b)

theoretical_c = 1 - norm.cdf(8)

N = 100_0000
mu = 8
y_shifted = np.random.randn(N) + mu

w = norm.pdf(y_shifted, 0, 1) / norm.pdf(y_shifted, 8, 1)

I = (y_shifted > 8).astype(float)

estimate_c = np.mean(I * w)
print("Importance sampling Monte Carlo estimate (c):", estimate_c)
print("Theoretical value:", theoretical_c)
```

Monte Carlo estimate result (a): 3.141655418534438

Theoretical value: 3.141592653589793
Monte Carlo estimate result (b): 7.67652499380115
Importance sampling Monte Carlo estimate (c): 6.211953979014197e-16
Theoretical value: 6.661338147750939e-16