HW1 Monte Carlo Method

February 26, 2025

[2]:
```python
import numpy as np

def estimate_pi(m):
    # Random points in the unit square (x, y)
    # Function to estimate pi using Monte Carlo method
    x = np.random.random(m) # Generating m floating points between [0,1)
    y = np.random.random(m)

    # Check how many points fall inside the unit circle
    inside_circle = (x**2 + y**2) <= 1
    return 4 * np.sum(inside_circle) / m  # Estimate of one pi

def calculate_rmse(m, repetitions=500):
    # Function to calculate RMSE for given sample size and number of repetitions
    pi_true = np.pi
    estimates = np.array([estimate_pi(m) for _ in range(repetitions)])
    # Calculate the Root Mean Squared Error (RMSE)
    rmse = np.sqrt(np.mean((estimates - pi_true) ** 2))
    return rmse
```

[4]:
```python
# Sample sizes
sample_sizes = [10000, 20000, 40000]

# Repeat the experiment for each sample size and calculate RMSE
rmse_results = {m: calculate_rmse(m) for m in sample_sizes}
print(rmse_results)
```

```
{10000: 0.01630214231932149, 20000: 0.011414864469666282, 40000:
0.00842188559981628}
```

[ ]:

[ ]: