Samin Shapour Miandouab

115232017  
HW1

Q1

In this part, the goal was to compute Monte Carlo estimates of π with a **fixed number of random points** and analyze the tradeoff between accuracy and computational cost.

**Method**

We looped over increasing sample sizes , generated random points in the unit square , and counted the number of “hits” inside the quarter circle . The estimate is

where is the number of hits. For each , we also recorded:

* The **absolute error** ,
* The **runtime** using tic/toc.

*x = rand(N,1);*

*y = rand(N,1);*

*H = sum(x.^2 + y.^2 <= 1); % quarter-circle hits*

*pi\_hat(k) = 4\*H/N; % Monte Carlo estimate*

*abs\_err(k) = abs(pi\_hat(k) - pi);*

*t\_sec(k) = toc;*

**Plot 1 — π estimate vs N:**

* As increases, converges toward the true value of π.
* At small , estimates fluctuate due to randomness, but by , the results are very close to π.

A graph on a black background

AI-generated content may be incorrect.

**Plot 2 — Absolute error vs N:**

* Error decreases roughly like , consistent with Monte Carlo theory.
* Larger leads to smaller deviations.

A graph on a black background

AI-generated content may be incorrect.

**Plot 3 — Precision vs computational cost:**

* Shows the trade-off: higher accuracy requires more runtime.
* The curve illustrates diminishing returns — doubling precision needs much more computation.

A graph with a line going up

AI-generated content may be incorrect.

Q2

**Goal.** Instead of fixing , keep sampling until the estimate is precise to a **user-chosen number of significant figures** . Precision is enforced with a **95% confidence-interval (CI) stop rule**, not by comparing to the true .

**Stop rule.** After each batch of points, compute a 95% CI for :

Stop when the relative half-width .

Figure 1 shows the Monte Carlo sampling in the unit square, highlighting points inside/outside the quarter circle.

A screen shot of a graph

AI-generated content may be incorrect.

Figure 2 shows the **convergence of π̂**, which gradually stabilizes around the true value as more samples are drawn.

A graph on a black background

AI-generated content may be incorrect.

Figure 3 shows the **95% confidence interval half-width shrinking** with sample size, crossing the stopping threshold.

A graph on a black background

AI-generated content may be incorrect.

Figure 4 compares the number of samples and runtime needed to reach **2 vs 3 significant figures**. Achieving higher precision required dramatically more samples and runtime.

A screenshot of a graph

AI-generated content may be incorrect.

Q3

A screen shot of a computer

AI-generated content may be incorrect.A screen shot of a graph

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

* For **2 significant figures**, π ≈ 3.1 was reached after about **5,000 samples**.
* For **3 significant figures**, π ≈ 3.13 required about **45,000 samples**.
* For **4 significant figures**, π ≈ 3.142 required nearly **4.2 million samples**.

This demonstrates the rapidly increasing computational cost as higher precision is requested. Each additional significant figure requires approximately an order of magnitude more samples.

The implementations are separated into 3 segments and in total of 4 MATLAB files, which for question 3, for better separation of concerns, I divided it to two parts. To run each question’s implementation, one can follow the bellow instructions:

Q1: run HW1Q1.m

Q2: HW1Q2.m

Q3: mcPi\_sigfigs.m