Monte-Carlo Methods and Sampling for Computing

Introduction

Francesco Banterle, Ph.D.

Meet Your Instructor

Francesco Banterle

- Ph.D. in Engineering from Warwick University, UK.

- Monte-carlo and sampling are daily tools for my research:
  - Computer Graphics;
  - Computer Vision;
  - Imaging.

Course

Reference Material

- The beautiful book by prof. Art Owen:
  - “Monte Carlo theory, methods and examples”
    - https://statweb.stanford.edu/~owen/mc/
    - @book{mcbook,

      author = {Art B. Owen},

      year = 2013,

      title = {Monte Carlo theory, methods and examples}

      }
- Other references:
  - Christian P. Robert, George Casella. “Monte Carlo Statistical Methods”. Springer Texts in Statistics. 2004.
  - Kurt Binder, Dieter Heermann. “Monte Carlo Simulation in Statistical Physics”. Springer. 2010.

Course

Exam

- Different options:
  - Seminar on a paper;
  - Programming project 1-2 people maximum;
  - Literature review on a few papers;
  - Interview.

Course
Schedule

- 23/05/2023: 9:00–13:00:
    - INTRODUCTION
    - UNIFORM RANDOM NUMBERS
- 30/05/2023: 09:00 —13:00:
    - NON-UNIFORM RANDOM NUMBERS
    - LOW DISCREPANCY SEQUENCES
- 06/06/2023: 09:00 —13:00:
    - VARIANCE REDUCTION TECHNIQUES
- 13/06/2023 10:30 —12:30:
    - METROPOLIS SAMPLING
    - MONTE-CARLO APPLICATIONS

What is the most visible application of Monte-Carlo today?

Monte-Carlo

Everyday

- Movies;
- Cars advertisement;
- IKEA Catalog;

Randomized Algorithms

Randomized Algorithms
The Basics

- Randomized algorithms try to solve a problem using randomness.
  - Why?
    - It may be too computationally expensive without.
- Typically, we have two classes of randomized algorithms:
  - Las Vegas Methods
  - Monte-Carlo Methods
- They both use pseudo-random number generators as source of randomness.

Las Vegas Algorithms

Main Idea

- A Las Vegas algorithm outputs **a correct solution** for a given problem.
- The running time may be **unbounded**; the expected running time is required to be bounded.
- A classic Las Vegas algorithms:
  - QuickSort;
  - Karger's algorithm (Minimum cut of a connected graph);
  - etc.

Monte-Carlo Algorithms
Main Idea

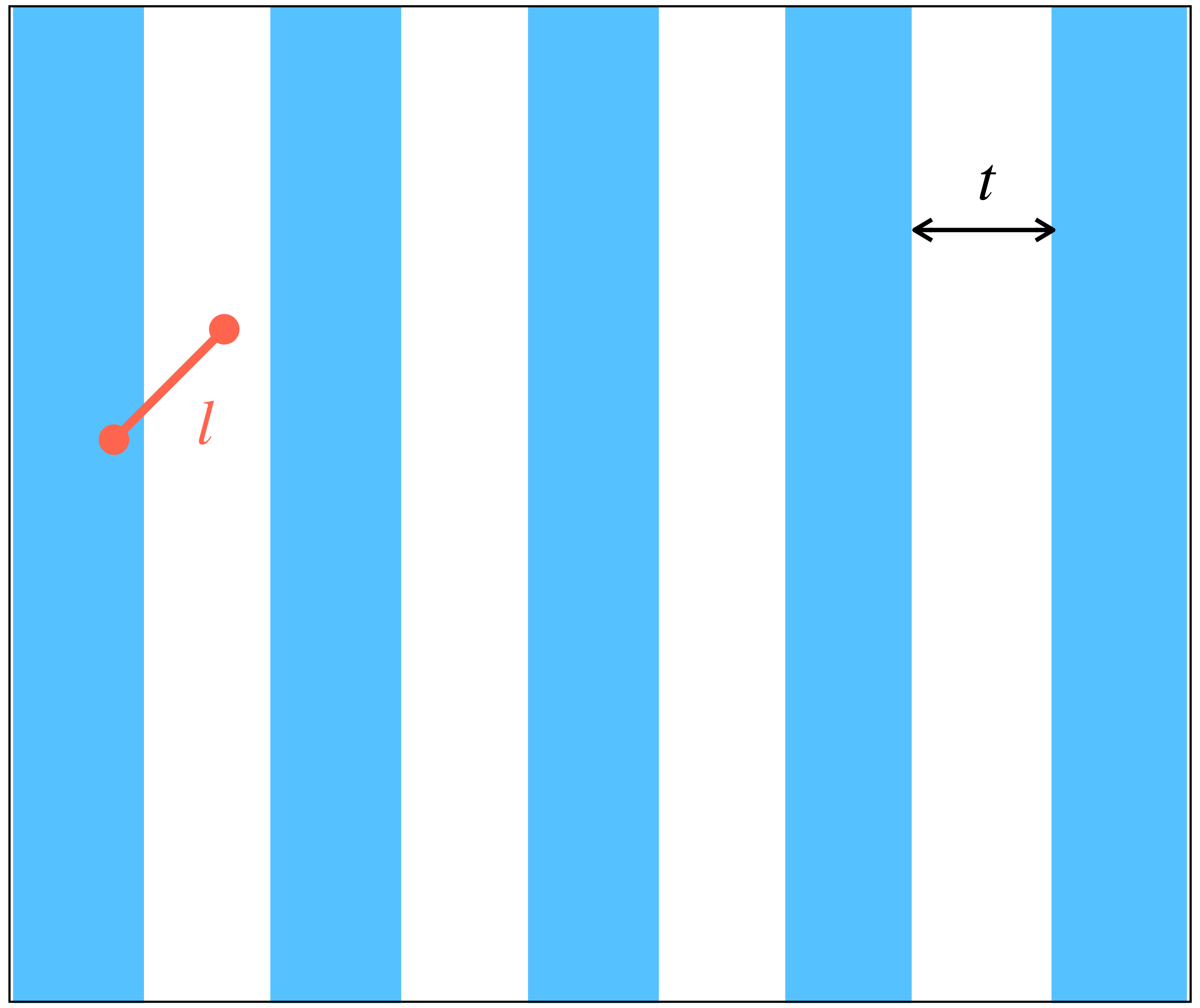
- A Monte-Carlo algorithm outputs **an approximated solution** for a given problem.
- Typically, we want to compute a quantity of interest:
  - The average of some random variable;
  - Quantiles;
  - Ratio
- The running time is **bounded**.

Monte-Carlo History

Monte-Carlo Algorithms

History

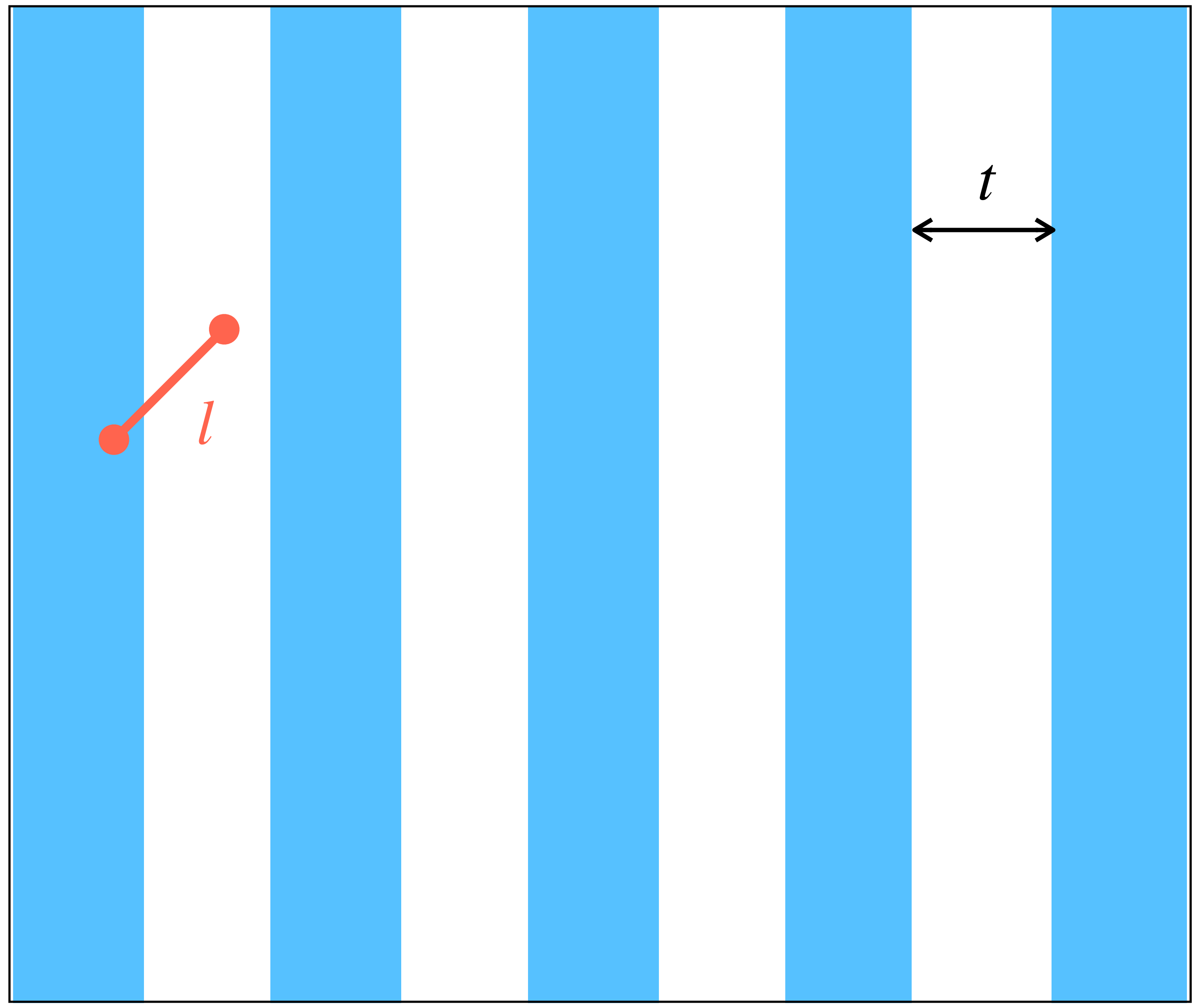
- 18th Century: Buffon's Needle
  - Based on a question by Georges-Louis Leclerc, Comte de Buffon:
  - *"What's the probability that a needle (that we threw on the floor) will lie across two strips on a floor made of parallel strips of wood?"*



Monte-Carlo Algorithms

History

- 18th Century: Buffon's Needle
    - Based on a question by Georges-Louis Leclerc, Comte de Buffon:
    - *"What's the probability that a needle (that we threw on the floor) will lie across two strips on a floor made of parallel strips of wood?"*



$$P(X = \text{btwstrips}) = \frac{2}{\pi}\frac{l}{t}$$

This holds for short needles: $l < t$

Monte-Carlo Algorithms

History

- 1900s: Gosset with pen-name Student; while developing the Student's t-distribution ran some simulations;
- 1930s: Fermi first experiments with Monte-Carlo;
- 1940s: Ulam, von Neumann, Metropolis during Manhattan project developed the modern Monte-Carlo especially for running simulations of nuclear weapons.
- 1950s: The method becomes popular in different fields such as physics, chemistry, etc.

Monte-Carlo Algorithms

History

- Montecarlo algorithms won three technical Oscars:
  - 1997: Ken Perlin for “solid noise” used in the movie Tron (1982);
  - 2003: Thomas Driemeyer’s team for MentalRay that uses quasi-montecarlo;
  - 2014: Eric Veach for multiple importance sampling;
  - 2014: Matt Pharr, Pat Hanrahan, and Greg Humphreys for formalization and reference implementation of Montecarlo methods for Computer Graphics.

Basics

Monte-Carlo Algorithms

Probability Theory Review

- A variable, X, is random/stochastic if its value cannot be determined before observing it; i.e., it depends on a **random phenomenon**.
- Even though we cannot know in advance the value of a variable X, we can say something about it in terms of probabilities.
  - In general, $P(E)$, is the probability of an event E to happen.
- Our main focus will be on continuous random variables.

Monte-Carlo Algorithms

Probability Theory Review

- A random variable X has an uncountably infinite number of possible values.
- Each variable has a probability density function (PDF) or $p_X(x)$ defined as:
  - a non-negative function defined on an interval (e.g., $[x_s, x_e]$);
  - normalized in such interval: $\int_{x_s}^{x_e} p_X(x)dx = 1$;
  - $P(a \leq X \leq b) = \int_a^b p_X(x)dx$
    - $p_X(x) = P(x \leq X \leq x + dx)$.

Monte-Carlo Algorithms

Probability Theory Review

- The cumulative distribution function (CDF) of a single random variable, X, is defined as:

$$F_X(x) = \int_{x_s}^{x} p_X(x)dx.$$

  - Note that:
    - $P(a \le X \le b) = \int_a^b p_X(x)dx = F_X(b) - F_X(a);$
    - $P(X \le a) = \int_{-\infty}^{a} p_X(x)dx;$
    - $P(X \ge a) = \int_a^{\infty} p_X(x)dx;$
    - $P(X = a) = 0.$
    - F_X is monotonically increasing.
    - $F_X(x_s) = 0$ and $F_X(x_e) = 1.$

Monte-Carlo Algorithms

Probability Theory Review

- Important measures of a PDF are its mean and its variance.
- The mean is defined as:

$$\mathbb{E}(X) = \mu(X) = \int_a^b x p_X(x) dx.$$

- The variance is defined as:

$$\sigma^2(X) = \mathbb{E}\left(\left(X - \mathbb{E}(X)\right)^2\right) = \mathbb{E}(X^2) - \mathbb{E}(X)^2,$$

where $\mathbb{E}(X^2) = \displaystyle\int_a^b x^2 p_X(x) dx$.

Some Practical Examples

Monte-Carlo Algorithms

An Example: Nagel-Schreckenberg Traffic Model

- This simulation has n cars running on a ring track.
- For each car at position x and speed v with distance d from the car ahead, we have the following rules:
  - $v \leftarrow \min(v+1, v_{\max})$
  - $v \leftarrow \min(v, d-1)$
  - $v \leftarrow \max(0, v-1)$ with p
  - $x \leftarrow x + v$

Monte-Carlo Algorithms

An Example: Nagel-Schreckenberg Traffic Model

- Let’s simulate this system with a track long $m = 1000$ and $n = 100$ cars.

- All cars have speed $v = 0$.

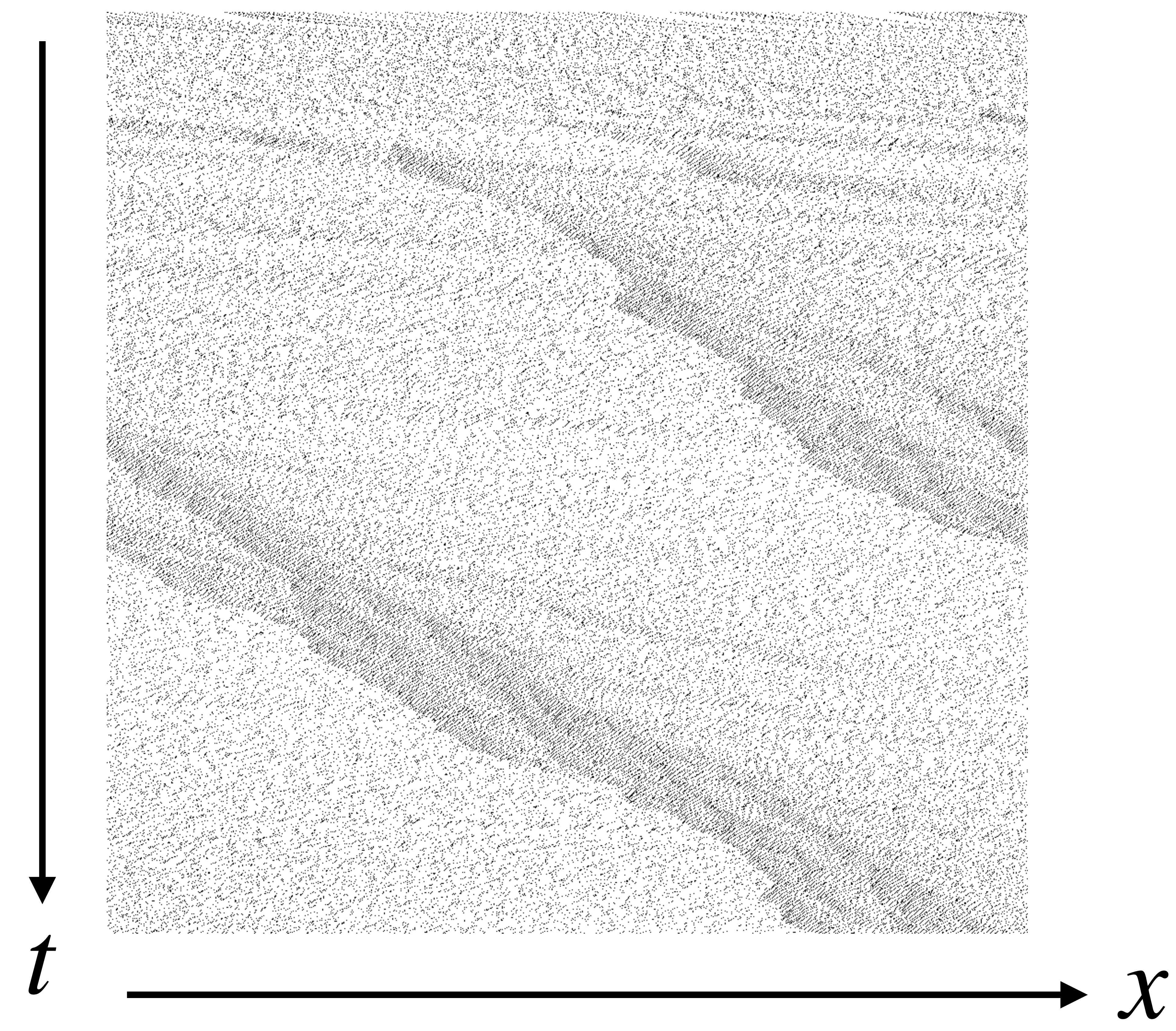
- All cars are placed on the track randomly without repetition.

- An image in some cases is more important to understand how the simulations behaves.

Monte-Carlo Algorithms

An Example: Nagel-Schreckenberg Traffic Model

- Let's simulate this system with a track long $m = 1000$ and $n = 100$ cars.
- All cars have speed $v = 0$.
- All cars are placed on the track randomly without repetition.
- An image in some cases is more important to understand how the simulations behaves.



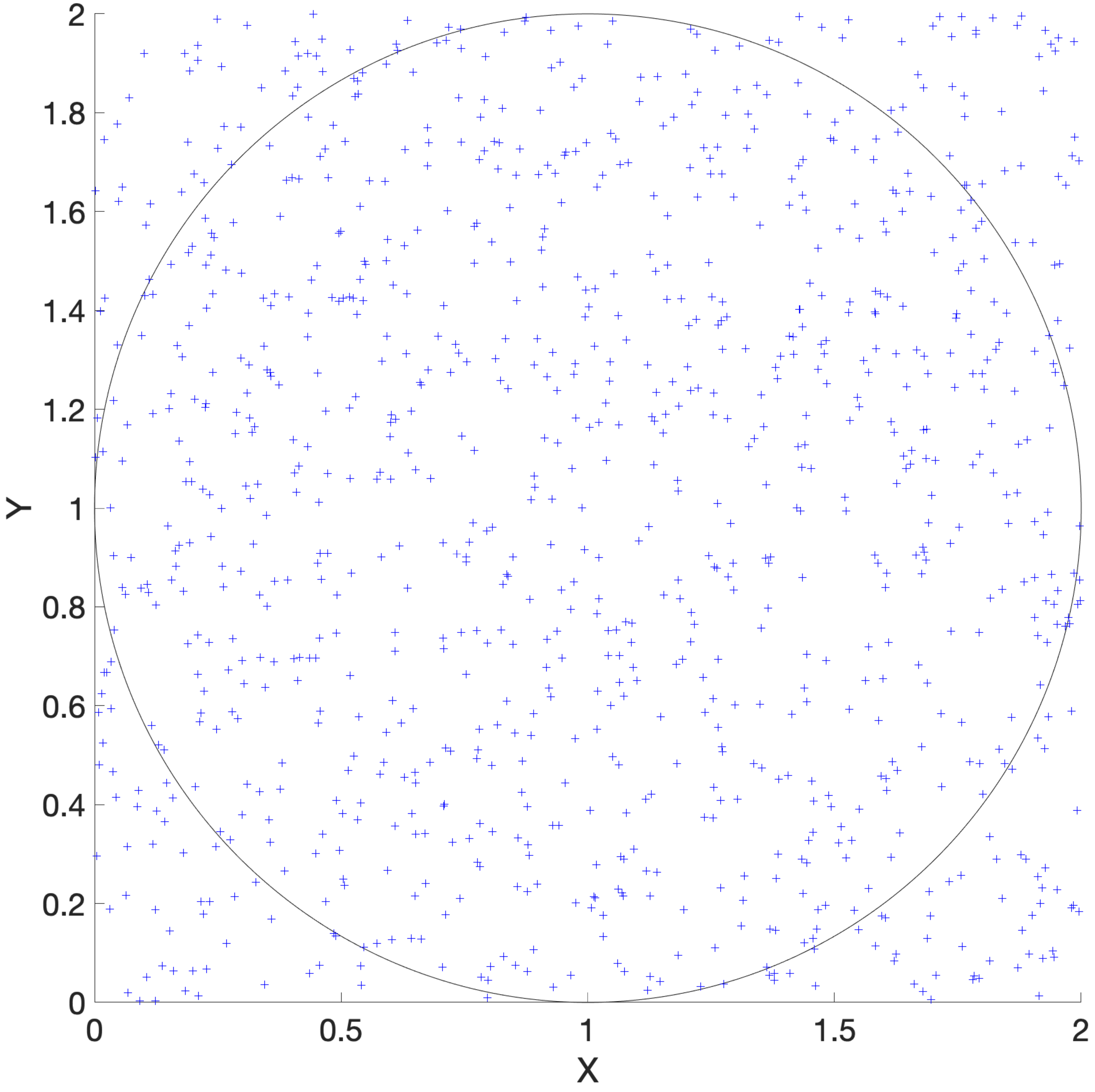
Monte-Carlo Algorithms

An Example: Estimating π

- We want to estimate π using Monte-Carlo.
- We know that the area of a circle is $A = \pi r^2$.
- We draw samples in a square; $[0,2] \times [0,2] -> r = 1$
- Samples that falls inside a circle with $r = 1$ and center in $(1,1)$ are used to estimate π.

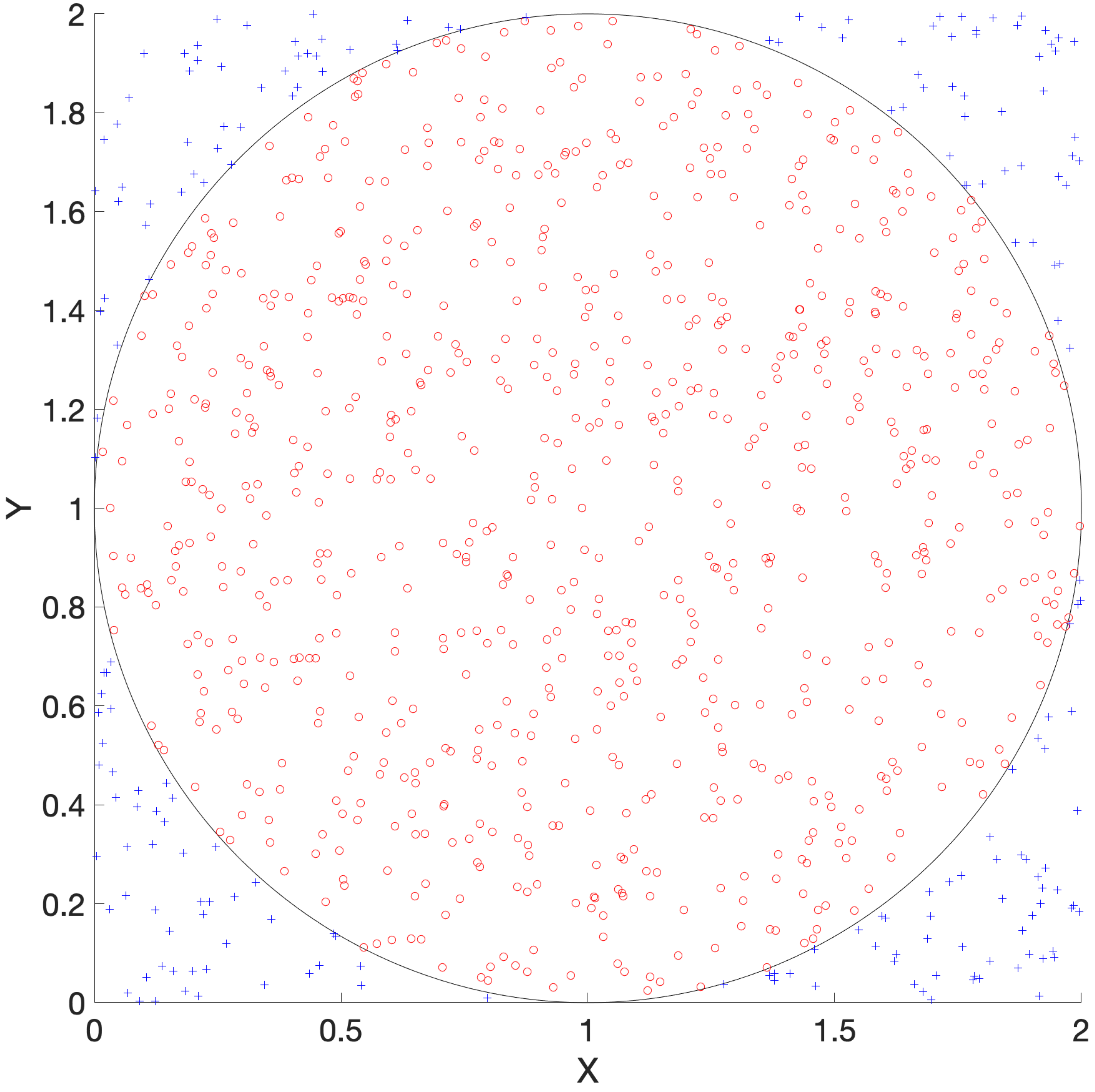
Monte-Carlo Algorithms

An Example: Estimating π



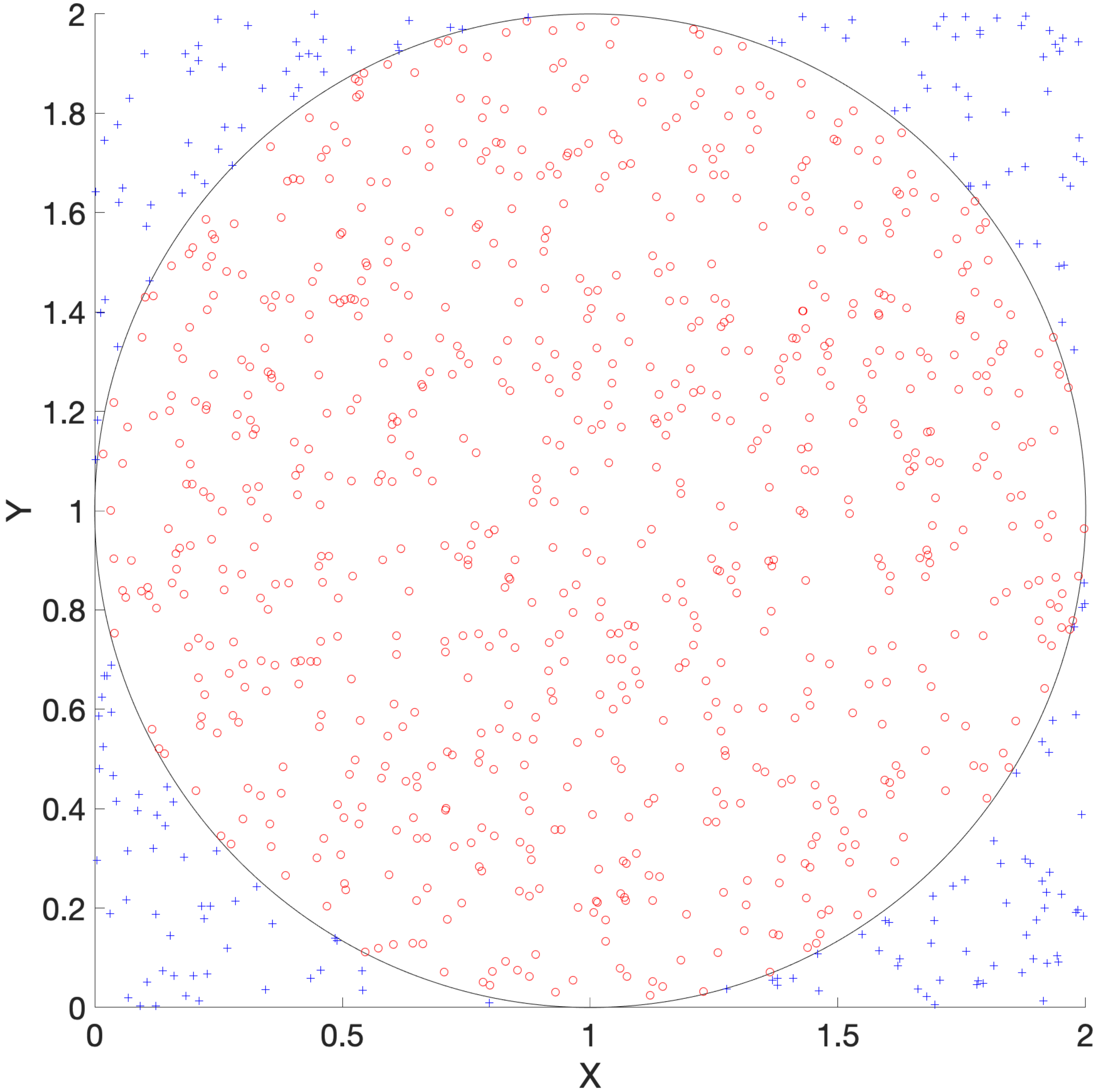
Monte-Carlo Algorithms

An Example: Estimating π



Monte-Carlo Algorithms

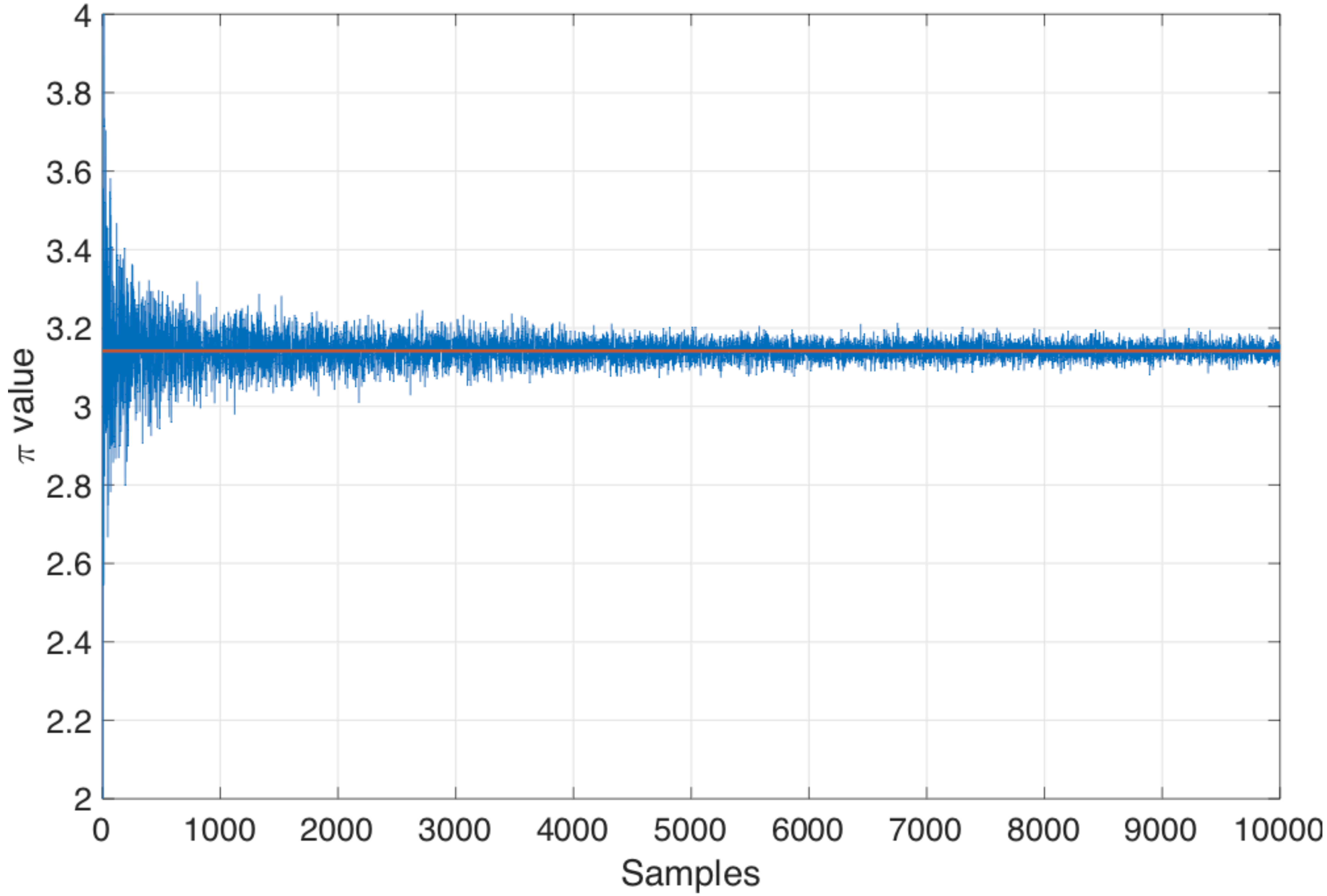
An Example: Estimating π



$$\pi_e = 4\frac{|\text{red\_samples}|}{|\text{blue\_samples}| + |\text{red\_samples}|}$$

Monte-Carlo Algorithms

An Example: Estimating π



Monte-Carlo Algorithms

An Example: Interpoint Distances

- We have two points; $\mathbf{x} = \big(x_1, x_2\big)$ and $\mathbf{y} = \big(y_1, y_2\big)$, where both are in $[0,a] \times [0,b]$.
- We define $D(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$.
- The mean of D can be approximated as:

$$\mathbb{E}(\hat{D}) = \frac{1}{n} \sum_{i=1}^{n} d(\mathbf{x}_i, \mathbf{y}_i),$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ are independent and uniformly distributed samples in $[0,a] \times [0,b]$.

Monte-Carlo Algorithms

An Example: Interpoint Distances

- Let's draw 1,000,000 samples in $[0,3] \times [0,2]$.

$$\mathbb{E}(\hat{D}) = 1.3171...$$

- This problem has a closed form introduced by Ghosh in 1951. In this case, the correct expected value for D would be:

$$\mathbb{E}(D) = 1.3171...$$

- If we compute the relative error, we get:

$$\frac{\mathbb{E}(\hat{D}) - \mathbb{E}(D)}{\mathbb{E}(D)} = 6.44 \times 10^{-4}.$$

- In many cases, we do not have a closed form for a problem!

Bibliography

- Art Owen. “Chapter 1: Introduction” from the book “Monte Carlo theory, methods and examples”. 2013.
- Art Owen. “Chapter 2: Simple Monte Carlo” from the book “Monte Carlo theory, methods and examples”. 2013.
- Peter Shirley, Changyaw Wang, Kurt Zimmerman. “Monte Carlo Techniques for Direct Lighting Calculations”. ACM Transactions on Graphics. Volume 15. Issue 1. Jan. 1996.

Thank you for your attention!