LA-UR-20-24374

Approved for public release; distribution is unlimited.

Title:	(U) Introduction to the Monte Carlo Method
Author(s):	Hill, James Lloyd
Intended for:	2020 X-CP Computational Physics Workshop lecture
Issued:	2020-06-17

Introduction to the Monte Carlo Method

Jim Hill
LANL

June 22, 2020

Introduction

- The goal here is to introduce you to the Monte Carlo (MC) method.
- MC is a very powerful simulation technique.
- Very simple examples will be used.
- Potential MC users have diverse backgrounds, but you should see from the examples how the method could be applied to your field of study.

Outline

What is Monte Carlo?

The Monte Carlo concept

- A Monte Carlo method is an algorithm that estimates the value of an integral by randomly sampling values within the problem domain.

The Monte Carlo concept

- A Monte Carlo method is an algorithm that estimates the value of an integral by randomly sampling values within the problem domain.
- The first (known) use of the method was Comte de Buffon in 1777.

The Monte Carlo concept

- A Monte Carlo method is an algorithm that estimates the value of an integral by randomly sampling values within the problem domain.
- The first (known) use of the method was Comte de Buffon in 1777.
- The method was formalized and its applicability to complex physics problems was established right here at Los Alamos during the Manhattan Project.

The Monte Carlo concept

- A Monte Carlo method is an algorithm that estimates the value of an integral by randomly sampling values within the problem domain.
- The first (known) use of the method was Comte de Buffon in 1777.
- The method was formalized and its applicability to complex physics problems was established right here at Los Alamos during the Manhattan Project.
- Mathematical rigor was largely developed in the postwar era.

The Monte Carlo concept

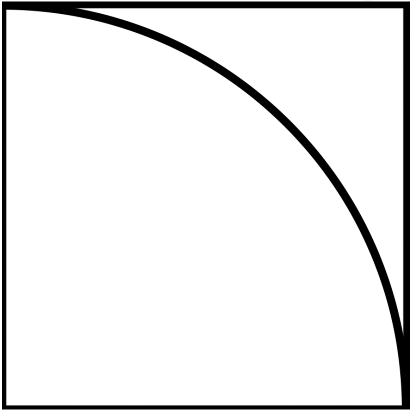
- A Monte Carlo method is an algorithm that estimates the value of an integral by randomly sampling values within the problem domain.
- The first (known) use of the method was Comte de Buffon in 1777.
- The method was formalized and its applicability to complex physics problems was established right here at Los Alamos during the Manhattan Project.
- Mathematical rigor was largely developed in the postwar era.
- The mathematical theory underpinning Monte Carlo is well-established … but this isn't a math lecture.

The Monte Carlo concept

- A Monte Carlo method is an algorithm that estimates the value of an integral by randomly sampling values within the problem domain.
- The first (known) use of the method was Comte de Buffon in 1777.
- The method was formalized and its applicability to complex physics problems was established right here at Los Alamos during the Manhattan Project.
- Mathematical rigor was largely developed in the postwar era.
- The mathematical theory underpinning Monte Carlo is well-established . . . but this isn't a math lecture.
- Today the method is used in physics, economics, epidemiology, you name it.

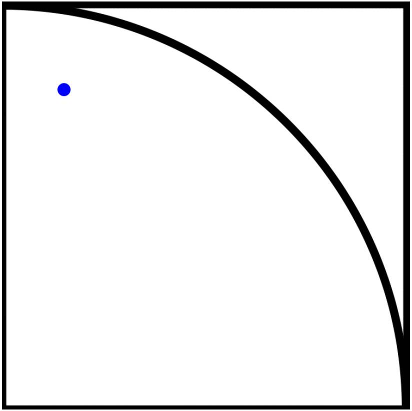
Estimation of π

Consider a quadrant of a unit circle inscribed within a square:



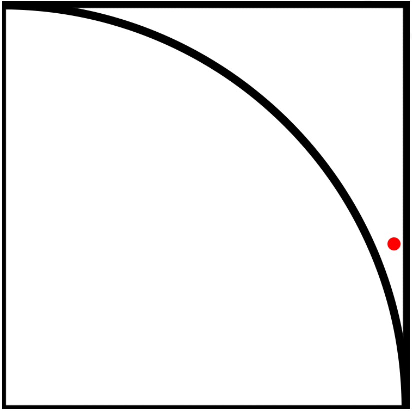
Estimation of π

Select 2 random Cartesian coordinates between 0 and 1. If their radius is within the circle, keep the point:



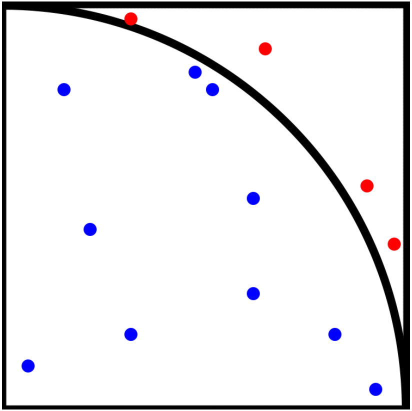
Estimation of π

If it isn't, reject the point:



Estimation of π

Over time, as you produce more and more points, you'll find the fraction of points you keep approaches the area of the quadrant to the area of the square, which is just $\pi/4$:



Estimation of π

```
#! /usr/bin/env python

import random
from math import sqrt as sqrt
from math import pi as pi

#  Given a quarter-circle inscribed within a unit square, calculate pi.
#  Intentionally inefficient for clarity.
#  Python's random() function returns a real number on [0,1).

for i in range(10):
  incircle = 0
  for samples in range(1,10**i+1):
    x = random.random(); y = random.random(); r = sqrt(x*x + y*y)
    if (r <= 1.0):
      incircle = incircle + 1
      frac = 1.0 * incircle / samples

  print "pi estimate for %10d samples = %.9f" % (samples,4.0*frac)
```

Estimation of π

```
pi estimate for         1 samples = 4.000000000
pi estimate for        10 samples = 2.400000000
pi estimate for       100 samples = 3.240000000
pi estimate for      1000 samples = 3.119119119
pi estimate for     10000 samples = 3.165600000
pi estimate for    100000 samples = 3.146351464
pi estimate for   1000000 samples = 3.140928000
pi estimate for  10000000 samples = 3.142132000
pi estimate for 100000000 samples = 3.141690040
```

This took about 11 minutes to run on a MacBook Pro.

Estimation of π

- The preceding is an example of what I call "intuitive Monte Carlo" (just my terminology).

Estimation of π

- The preceding is an example of what I call “intuitive Monte Carlo” (just my terminology).
- It feels right.

Estimation of π

- The preceding is an example of what I call "intuitive Monte Carlo" (just my terminology).
- It feels right.
- It sure looks right.

Estimation of π

- The preceding is an example of what I call "intuitive Monte Carlo" (just my terminology).
- It feels right.
- It sure looks right.
- Later we'll discuss why it *is* right.

Simple probability and countable events

- Given an elementary event with a countable number n of outcomes, each outcome $E_k, k = 1 \ldots n$ has a probability p_k of occurring.

Simple probability and countable events

- Given an elementary event with a countable number n of outcomes, each outcome $E_k, k = 1 \ldots n$ has a probability p_k of occurring.
- p_k is between 0 and 1, where 0 indicates the outcome cannot happen and 1 indicates the outcome must happen.

Simple probability and countable events

- Given an elementary event with a countable number n of outcomes, each outcome E_k, $k = 1 \dots n$ has a probability p_k of occurring.
- p_k is between 0 and 1, where 0 indicates the outcome cannot happen and 1 indicates the outcome must happen.
- Probabilities are generally normalized to 1; that is, $\sum_{k=1}^{n} p_k = 1$.

Simple probability and countable events

- Given an elementary event with a countable number n of outcomes, each outcome E_k, $k = 1 \ldots n$ has a probability p_k of occurring.
- p_k is between 0 and 1, where 0 indicates the outcome cannot happen and 1 indicates the outcome must happen.
- Probabilities are generally normalized to 1; that is, $\sum_{k=1}^{n} p_k = 1$.
- If for every outcome E_k there is an associated value x_k, then x_k is said to be a *random variable*.

Simple probability and countable events

- Given an elementary event with a countable number n of outcomes, each outcome E_k, $k = 1 \ldots n$ has a probability p_k of occurring.
- p_k is between 0 and 1, where 0 indicates the outcome cannot happen and 1 indicates the outcome must happen.
- Probabilities are generally normalized to 1; that is, $\sum_{k=1}^{n} p_k = 1$.
- If for every outcome E_k there is an associated value x_k, then x_k is said to be a *random variable*.
- The *expectation* of the random variable x is $\sum_{k=1}^{n} p_k x_k$, often written $\langle x \rangle$.

Simple probability and countable events

- Given an elementary event with a countable number n of outcomes, each outcome $E_k, k = 1 \dots n$ has a probability p_k of occurring.
- p_k is between 0 and 1, where 0 indicates the outcome cannot happen and 1 indicates the outcome must happen.
- Probabilities are generally normalized to 1; that is, $\sum_{k=1}^{n} p_k = 1$.
- If for every outcome E_k there is an associated value x_k, then x_k is said to be a *random variable*.
- The *expectation* of the random variable x is $\sum_{k=1}^{n} p_k x_k$, often written $\langle x \rangle$.
- The expectation is also called the *expectation value* or the *mean*.

Simple probability and countable events

- The expectation of a constant $\langle constant \rangle = constant$.

Simple probability and countable events

- The expectation of a constant $\langle constant \rangle = constant$.
- If x is a random variable, the real-valued function $g(x)$ is also a random variable with expectation $\langle g(x) \rangle = \sum_{k=1}^{n} p_k g(x_k)$.

Simple probability and countable events

- The expectation of a constant $\langle constant \rangle = constant$.
- If x is a random variable, the real-valued function $g(x)$ is also a random variable with expectation $\langle g(x) \rangle = \sum_{k=1}^{n} p_k g(x_k)$.
- If $g(x)$ is a linear combination $g(x) = \lambda_1 g_1(x) + \lambda_2 g_2(x)$, then $\langle g(x) \rangle = \lambda \langle g_1(x) \rangle + \lambda_2 \langle g_2(x) \rangle$.

Simple probability and countable events

- The expectation of a constant $\langle constant \rangle = constant$.
- If x is a random variable, the real-valued function $g(x)$ is also a random variable with expectation $\langle g(x) \rangle = \sum_{k=1}^{n} p_k g(x_k)$.
- If $g(x)$ is a linear combination $g(x) = \lambda_1 g_1(x) + \lambda_2 g_2(x)$, then $\langle g(x) \rangle = \lambda \langle g_1(x) \rangle + \lambda_2 \langle g_2(x) \rangle$.
- If $g(x)$ is a linear function of x, then $\langle g(x) \rangle = g(\langle x \rangle)$.

Simple probability and countable events

- The nth *moment* of x is the expectation value of the nth power of x; that is, $\langle x^n \rangle = \sum_k p_k x_k^n$.

Simple probability and countable events

- The nth *moment* of x is the expectation value of the nth power of x; that is, $\langle x^n \rangle = \sum_k p_k x_k^n$.

- The nth *central moment* of x is $\langle (x-\mu)^n \rangle = \sum_k p_k (x_k - \mu)^n = \sum_k p_k (x_k - \langle x \rangle)^n$.

Simple probability and countable events

- The nth *moment* of x is the expectation value of the nth power of x; that is, $\langle x^n \rangle = \sum_k p_k x_k^n$.

- The nth *central moment* of x is $\langle (x-\mu)^n \rangle = \sum_k p_k (x_k - \mu)^n = \sum_k p_k (x_k - \langle x \rangle)^n$.

- The second central moment $\sum_k p_k (x_k - \langle x \rangle)^n = \langle x^2 \rangle - \langle x \rangle^2$.

Simple probability and countable events

- The nth *moment* of x is the expectation value of the nth power of x; that is, $\langle x^n \rangle = \sum_k p_k x_k^n$.

- The nth *central moment* of x is $\langle (x-\mu)^n \rangle = \sum_k p_k (x_k - \mu)^n = \sum_k p_k (x_k - \langle x \rangle)^n$.

- The second central moment $\sum_k p_k (x_k - \langle x \rangle)^n = \left\langle x^2 \right\rangle - \langle x \rangle^2$.

- This quantity, also called the *variance* or *standard deviation* of x, is a measure of the dispersion of the random variable about its mean value.

Continuous random variables

- Let x be a continous random variable: $-\infty < x < \infty$.

Continuous random variables

- Let x be a continous random variable: $-\infty < x < \infty$.
- The *cumulative distribution function*, or CDF, is defined as $F(x)$ being the probability that a randomly-selected value $X < x$.

Continuous random variables

- Let x be a continous random variable: $-\infty < x < \infty$.
- The *cumulative distribution function*, or CDF, is defined as $F(x)$ being the probability that a randomly-selected value $X < x$.
- $F(x)$ is a nondecreasing function of x, with $F(-\infty) = 0$ and $F(\infty) = 1$.

Continuous random variables

- Let x be a continous random variable: $-\infty < x < \infty$.
- The *cumulative distribution function*, or CDF, is defined as $F(x)$ being the probability that a randomly-selected value $X < x$.
- $F(x)$ is a nondecreasing function of x, with $F(-\infty) = 0$ and $F(\infty) = 1$.
- The *probability distribution (density) function*, or PDF, is defined as dF/dx and is normalized to 1: $(\int_{-\infty}^{\infty} f(x)dx = F(\infty) - F(-\infty) = 1)$.

Continuous random variables

- Let x be a continous random variable: $-\infty < x < \infty$.
- The *cumulative distribution function*, or CDF, is defined as $F(x)$ being the probability that a randomly-selected value $X < x$.
- $F(x)$ is a nondecreasing function of x, with $F(-\infty) = 0$ and $F(\infty) = 1$.
- The *probability distribution (density) function*, or PDF, is defined as dF/dx and is normalized to 1: $(\int_{-\infty}^{\infty} f(x)dx = F(\infty) - F(-\infty) = 1)$.
- The expectation of a continuous random variable is defined as $E(x) = \int_{-\infty}^{\infty} xf(x)dx$.

Continuous random variables

- Let x be a continous random variable: $-\infty < x < \infty$.
- The *cumulative distribution function*, or CDF, is defined as $F(x)$ being the probability that a randomly-selected value $X < x$.
- $F(x)$ is a nondecreasing function of x, with $F(-\infty) = 0$ and $F(\infty) = 1$.
- The *probability distribution (density) function*, or PDF, is defined as dF/dx and is normalized to 1: $(\int_{-\infty}^{\infty} f(x)dx = F(\infty) - F(-\infty) = 1)$.
- The expectation of a continuous random variable is defined as $E(x) = \int_{-\infty}^{\infty} xf(x)dx$.
- Similarly, for a function $g(x)$, $E(g(x)) = \int_{-\infty}^{\infty} g(x)f(x)dx$.

Continuous random variables

- Let x be a continous random variable: $-\infty < x < \infty$.
- The *cumulative distribution function*, or CDF, is defined as $F(x)$ being the probability that a randomly-selected value $X < x$.
- $F(x)$ is a nondecreasing function of x, with $F(-\infty) = 0$ and $F(\infty) = 1$.
- The *probability distribution (density) function*, or PDF, is defined as dF/dx and is normalized to 1: $(\int_{-\infty}^{\infty} f(x)dx = F(\infty) - F(-\infty) = 1)$.
- The expectation of a continuous random variable is defined as $E(x) = \int_{-\infty}^{\infty} xf(x)dx$.
- Similarly, for a function $g(x)$, $E(g(x)) = \int_{-\infty}^{\infty} g(x)f(x)dx$.
- The variances are defined in the same manner.

Monte Carlo quadrature

- Suppose N random variables x_1, x_2, $\ldots x_N$ are sampled from the PDF $f(x)$.

Monte Carlo quadrature

- Suppose N random variables x_1, x_2, $\ldots x_N$ are sampled from the PDF $f(x)$.
- Let $G = \sum_{n=1}^{N} \lambda_n g_n(x)$.

Monte Carlo quadrature

- Suppose N random variables x_1, x_2, $\ldots x_N$ are sampled from the PDF $f(x)$.
- Let $G = \sum_{n=1}^{N} \lambda_n g_n(x)$.
- The expectation E of G is $E(G) = \langle G \rangle = \sum_{n=1}^{N} \lambda_n \langle g_n(x) \rangle$.

Monte Carlo quadrature

- Suppose N random variables $x_1, x_2, \ldots x_N$ are sampled from the PDF $f(x)$.
- Let $G = \sum_{n=1}^{N} \lambda_n g_n(x)$.
- The expectation E of G is $E(G) = \langle G \rangle = \sum_{n=1}^{N} \lambda_n \langle g_n(x) \rangle$.
- The variance V of G is $V(G) = \left\langle (G - \langle G \rangle)^2 \right\rangle = \sum_{n=1}^{N} \lambda_n^2 V(g_n(x))$.

Monte Carlo quadrature

- Suppose N random variables x_1, x_2, $\ldots x_N$ are sampled from the PDF $f(x)$.
- Let $G = \sum_{n=1}^{N} \lambda_n g_n(x)$.
- The expectation E of G is $E(G) = \langle G \rangle = \sum_{n=1}^{N} \lambda_n \langle g_n(x) \rangle$.
- The variance V of G is $V(G) = \left\langle (G - \langle G \rangle)^2 \right\rangle = \sum_{n=1}^{N} \lambda_n^2 V(g_n(x))$.
- Let all $\lambda_n = \frac{1}{N}$ and all $g_n(x) = g(x_n)$.

Monte Carlo quadrature

- Suppose N random variables x_1, x_2, $\ldots x_N$ are sampled from the PDF $f(x)$.
- Let $G = \sum_{n=1}^{N} \lambda_n g_n(x)$.
- The expectation E of G is $E(G) = \langle G \rangle = \sum_{n=1}^{N} \lambda_n \langle g_n(x) \rangle$.
- The variance V of G is $V(G) = \left\langle (G - \langle G \rangle)^2 \right\rangle = \sum_{n=1}^{N} \lambda_n^2 V(g_n(x))$.
- Let all $\lambda_n = \frac{1}{N}$ and all $g_n(x) = g(x_n)$.
- $E(G)$ is thus $\frac{1}{N} \sum_{n=1}^{N} \langle g(x) \rangle = \langle g(x) \rangle$.

Monte Carlo quadrature

- Suppose N random variables $x_1, x_2, \ldots x_N$ are sampled from the PDF $f(x)$.
- Let $G = \sum_{n=1}^{N} \lambda_n g_n(x)$.
- The expectation E of G is $E(G) = \langle G \rangle = \sum_{n=1}^{N} \lambda_n \langle g_n(x) \rangle$.
- The variance V of G is $V(G) = \left\langle (G - \langle G \rangle)^2 \right\rangle = \sum_{n=1}^{N} \lambda_n^2 V(g_n(x))$.
- Let all $\lambda_n = \frac{1}{N}$ and all $g_n(x) = g(x_n)$.
- $E(G)$ is thus $\frac{1}{N} \sum_{n=1}^{N} \langle g(x) \rangle = \langle g(x) \rangle$.
- Thus, the arithmetic average G has the same mean as $g(x)$. G is an *estimator* of $\langle g(x) \rangle$.

Monte Carlo quadrature

- Suppose N random variables $x_1, x_2, \ldots x_N$ are sampled from the PDF $f(x)$.
- Let $G = \sum_{n=1}^{N} \lambda_n g_n(x)$.
- The expectation E of G is $E(G) = \langle G \rangle = \sum_{n=1}^{N} \lambda_n \langle g_n(x) \rangle$.
- The variance V of G is $V(G) = \left\langle (G - \langle G \rangle)^2 \right\rangle = \sum_{n=1}^{N} \lambda_n^2 V(g_n(x))$.
- Let all $\lambda_n = \frac{1}{N}$ and all $g_n(x) = g(x_n)$.
- $E(G)$ is thus $\frac{1}{N} \sum_{n=1}^{N} \langle g(x) \rangle = \langle g(x) \rangle$.
- Thus, the arithmetic average G has the same mean as $g(x)$. G is an *estimator* of $\langle g(x) \rangle$.
- Likewise, $V(G) = \frac{1}{N} V(g(x))$.

Monte Carlo quadrature

- The million-dollar baby of Monte Carlo is therefore:

Monte Carlo quadrature

- The million-dollar baby of Monte Carlo is therefore:
- An integral $\int_{-\infty}^{\infty} g(x)f(x)dx$ can be approximated as $\frac{1}{N}\sum_{n=1}^{N} g(x_n)$, where the x_n are drawn from the PDF $f(x)$.

Monte Carlo quadrature

- The million-dollar baby of Monte Carlo is therefore:
- An integral $\int_{-\infty}^{\infty} g(x) f(x) dx$ can be approximated as $\frac{1}{N} \sum_{n=1}^{N} g(x_n)$, where the x_n are drawn from the PDF $f(x)$.
- The variance of this estimate decreases as the number of samples increases.

Monte Carlo quadrature

- The million-dollar baby of Monte Carlo is therefore:
- An integral $\int_{-\infty}^{\infty} g(x)f(x)dx$ can be approximated as $\frac{1}{N}\sum_{n=1}^{N} g(x_n)$, where the x_n are drawn from the PDF $f(x)$.
- The variance of this estimate decreases as the number of samples increases.
- The Monte Carlo method thus lets you compute solutions of integrals *with a bounded error estimate*, allowing you to determine when your solution has achieved a desired accuracy.

Monte Carlo quadrature

- The million-dollar baby of Monte Carlo is therefore:
- An integral $\int_{-\infty}^{\infty} g(x) f(x) dx$ can be approximated as $\frac{1}{N}\sum_{n=1}^{N} g(x_n)$, where the x_n are drawn from the PDF $f(x)$.
- The variance of this estimate decreases as the number of samples increases.
- The Monte Carlo method thus lets you compute solutions of integrals *with a bounded error estimate*, allowing you to determine when your solution has achieved a desired accuracy.
- For math reasons we more commonly use the *empirical variance* $\sigma^2 = \frac{1}{N-1}\left(\langle g^2 \rangle - \langle g \rangle^2\right)$, which has value ∞ for a single sample.

Sampling a random variable

- If we know the PDF $f(x)$, we can calculate the CDF $F(x)$.

Sampling a random variable

- If we know the PDF $f(x)$, we can calculate the CDF $F(x)$.
- If instead we know $F(x)$, we can calculate $f(x)$.

Sampling a random variable

- If we know the PDF $f(x)$, we can calculate the CDF $F(x)$.
- If instead we know $F(x)$, we can calculate $f(x)$.
- Either way, since the range of F is $(0, 1)$, we can generate a random number R in that range and calculate $x = F^{-1}(R)$, then $f(x)$. Depending on the form of F or f, this can be easy or painful.

Sampling a random variable

- If we know the PDF $f(x)$, we can calculate the CDF $F(x)$.
- If instead we know $F(x)$, we can calculate $f(x)$.
- Either way, since the range of F is $(0, 1)$, we can generate a random number R in that range and calculate $x = F^{-1}(R)$, then $f(x)$. Depending on the form of F or f, this can be easy or painful.
- Most random-number generation routines return a value on $[0, 1)$.

Estimation of π revisited

- The area of that quarter unit circle is $A = \int_0^1 \sqrt{(1-x^2)}dx = \frac{\pi}{4}$.

Estimation of π revisited

- The area of that quarter unit circle is $A = \int_0^1 \sqrt{(1-x^2)}dx = \frac{\pi}{4}$.
- If we identify the PDF $f(x) = 1$ and the function $g(x) = \sqrt{(1-x^2)}$, we can approximate the integral as $\mu = \frac{1}{N}\sum_{n=1}^{N}\sqrt{(1-x_n^2)}$.

Estimation of π revisited

- The area of that quarter unit circle is $A = \int_0^1 \sqrt{(1 - x^2)}dx = \frac{\pi}{4}$.
- If we identify the PDF $f(x) = 1$ and the function $g(x) = \sqrt{(1 - x^2)}$, we can approximate the integral as $\mu = \frac{1}{N} \sum_{n=1}^{N} \sqrt{(1 - x_n^2)}$.
- The value of the integral lies within $\mu (1 \pm \sigma) = \mu \pm \epsilon$.

Estimation of π revisited

- The area of that quarter unit circle is $A = \int_0^1 \sqrt{(1 - x^2)}dx = \frac{\pi}{4}$.
- If we identify the PDF $f(x) = 1$ and the function $g(x) = \sqrt{(1 - x^2)}$, we can approximate the integral as $\mu = \frac{1}{N} \sum_{n=1}^{N} \sqrt{(1 - x_n^2)}$.
- The value of the integral lies within $\mu\,(1 \pm \sigma) = \mu \pm \epsilon$.
- Since the variance decreases with N, every additional digit of improvement in the error estimate requires $100\times$ the samples.

Estimation of π revisited

```
#! /usr/bin/env python

import random
from math import sqrt as sqrt

#  Given a quarter-circle inscribed within a unit square, calculate pi
#  as integral(0,1) sqrt(1-x^2) dx.

for i in range(1,9):
  the_sum  = 0.0; the_sum2 = 0.0
  for idx in range(1,10**i+1):
    F = random.random(); x = F
    g = sqrt(1-x*x)
    the_sum = the_sum + g; the_sum2 = the_sum2 + g*g
  the_mean = the_sum / idx
  the_variance = 1.0/(idx-1)*(the_sum2/idx-the_mean*the_mean)
  the_error = sqrt(the_variance)
  my_pi = the_mean * 4.0; my_error = the_error * 4.0
  print "pi estimate for %09d counts = %.9f +/- %.9f" % (idx,my_pi,my_error)
```

Estimation of π

```
pi estimate for 000000010 counts = 3.419267129 +/- 0.202535995
pi estimate for 000000100 counts = 3.024936474 +/- 0.098122722
pi estimate for 000001000 counts = 3.150753001 +/- 0.028345405
pi estimate for 000010000 counts = 3.138572099 +/- 0.008995848
pi estimate for 000100000 counts = 3.141813753 +/- 0.002825196
pi estimate for 001000000 counts = 3.141734513 +/- 0.000892218
pi estimate for 010000000 counts = 3.141498800 +/- 0.000282275
pi estimate for 100000000 counts = 3.141540350 +/- 0.000089286
```

This took about 1 minute to run on a MacBook Pro.

Importance sampling

- Consider the integral $I = \int_0^1 \cos\left(\frac{\pi x}{2}\right) dx$.

Importance sampling

- Consider the integral $I = \int_0^1 \cos\left(\frac{\pi x}{2}\right) dx$.
- Analytically, we know $I = \frac{2}{\pi}$.

Importance sampling

- Consider the integral $I = \int_0^1 \cos\left(\frac{\pi x}{2}\right) dx$.
- Analytically, we know $I = \frac{2}{\pi}$.
- Let's use *analog Monte Carlo* to compute the integral.

Importance sampling

- Consider the integral $I = \int_0^1 \cos\left(\frac{\pi x}{2}\right) dx$.
- Analytically, we know $I = \frac{2}{\pi}$.
- Let's use *analog Monte Carlo* to compute the integral.
- Analog Monte Carlo describes an MC simulation that doesn't apply any numerical gimmickry to accelerate its convergence.

Importance sampling

```
#! /usr/bin/env python

import random
import time
from math import pi   as pi
from math import cos  as cos
from math import sqrt as sqrt

tstart = time.time()

random.seed(1999)
exact   = 2.0/pi
samples = 0
csum = 0.0; c2sum = 0.0; cerr = 1.0

# Continued on next slide
```

Importance sampling

```
# Continued from previous slide

while (cerr > 1.e-5):

  samples = samples + 1
  F = random.random()
  x = F
  val = cos(pi*x/2.0)
  csum  = csum + val; c2sum = c2sum + val**2

  cmean = csum/samples
  if (samples > 1):
    cvar = 1.0/(samples-1)*(c2sum/samples-cmean*cmean); cerr = sqrt(cvar)/cmean
  if (samples % 10000 == 0):
    print "%08d    %.8e    %.8e    %.8e" % (samples, cmean, cvar, cerr)

# Continued on next slide
```

Importance sampling

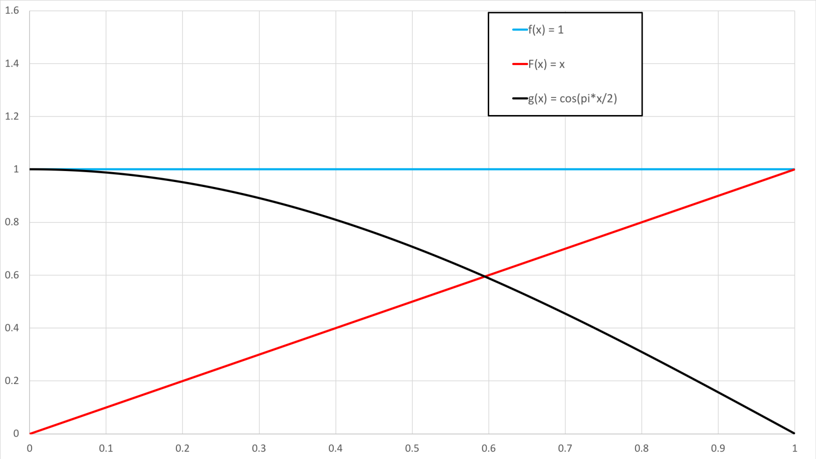
```
# Continued from previous slide

tend = time.time(); dt = tend - tstart; duration = dt
hours = dt // 3600; dt = dt % 3600
mins = dt // 60; secs = dt % 60

print("Samples = %d" % (samples))
print "Integral = %.8f  (Exact = %.8f)" % (cmean,exact)
print "Variance = %.8e" % (cvar)
print "Error = %.8e" % (cerr)
print "Calculation took %d hr, %d min, %.3f seconds (%.3e s/sample)." % \
       (hours, mins, secs, duration/samples)
```

Importance sampling

```
Samples = 2337099887
Integral = 0.63661544  (Exact = 0.63661977)
Variance = 4.05279220e-11
Error = 1.00000000e-05
Calculation took 0 hr, 32 min, 45.791 seconds (8.411e-07 s/sample).
```

Importance sampling

Why so many (2.3B) samples?



Importance sampling

- Because $f(x) = 1$, all points x are equally likely, including the ones near $x = 1$ that contribute very little to the sum used to approximate the integral.

Importance sampling

- Because $f(x) = 1$, all points x are equally likely, including the ones near $x = 1$ that contribute very little to the sum used to approximate the integral.

- Is there something we can do about that?

Importance sampling

- Because $f(x) = 1$, all points x are equally likely, including the ones near $x = 1$ that contribute very little to the sum used to approximate the integral.
- Is there something we can do about that?
- Sure is – it's called *importance sampling*.

Importance sampling

- Because $f(x) = 1$, all points x are equally likely, including the ones near $x = 1$ that contribute very little to the sum used to approximate the integral.
- Is there something we can do about that?
- Sure is – it's called *importance sampling*.
- This probably doesn't surprise you if you've been reading the slide titles.

Importance sampling

- Suppose you introduce a function $f^*(x)$ and rewrite the integral as $\int_0^1 \frac{g(x)f(x)}{f^*(x)} f^*(x)dx$.

Importance sampling

- Suppose you introduce a function $f^*(x)$ and rewrite the integral as $\int_0^1 \frac{g(x)f(x)}{f^*(x)} f^*(x)dx$.

- This has the same value as the original; however, we're now sampling a different integrand $\frac{g(x)f(x)}{f^*(x)}$ with a different PDF $f^*(x)$.

Importance sampling

- Suppose you introduce a function $f^*(x)$ and rewrite the integral as $\int_0^1 \frac{g(x)f(x)}{f^*(x)} f^*(x)dx$.
- This has the same value as the original; however, we're now sampling a different integrand $\frac{g(x)f(x)}{f^*(x)}$ with a different PDF $f^*(x)$.
- If we are judicious, this form will have a smaller variance than the original and thus you'll get to a "converged" solution faster.

Importance sampling

- Suppose you introduce a function $f^*(x)$ and rewrite the integral as $\int_0^1 \frac{g(x)f(x)}{f^*(x)} f^*(x) dx$.
- This has the same value as the original; however, we're now sampling a different integrand $\frac{g(x)f(x)}{f^*(x)}$ with a different PDF $f^*(x)$.
- If we are judicious, this form will have a smaller variance than the original and thus you'll get to a "converged" solution faster.
- $f^*(x) = g(x)$ would have zero variance, but in order to normalize $f^*(x)$ we'd need to calculate the very integral that defined the problem.

Importance sampling

- Suppose you introduce a function $f^*(x)$ and rewrite the integral as $\int_0^1 \frac{g(x)f(x)}{f^*(x)} f^*(x) dx$.
- This has the same value as the original; however, we're now sampling a different integrand $\frac{g(x)f(x)}{f^*(x)}$ with a different PDF $f^*(x)$.
- If we are judicious, this form will have a smaller variance than the original and thus you'll get to a "converged" solution faster.
- $f^*(x) = g(x)$ would have zero variance, but in order to normalize $f^*(x)$ we'd need to calculate the very integral that defined the problem.
- What if we choose $f^*(x)$ to be something pretty close to $g(x)$ but more easily integrated for normalization?

Importance sampling

- The Taylor series expansion of $\cos\left(\frac{\pi x}{2}\right) = 1 - \frac{\pi^2}{2 \times 2^2} x^2 + \ldots$, so let's try this:

Importance sampling

- The Taylor series expansion of $\cos\left(\frac{\pi x}{2}\right) = 1 - \frac{\pi^2}{2 \times 2^2} x^2 + \ldots$, so let's try this:

- $f^*(x) = \lambda\left(1 - x^2\right)$.

Importance sampling

- The Taylor series expansion of $\cos\left(\frac{\pi x}{2}\right) = 1 - \frac{\pi^2}{2 \times 2^2} x^2 + \ldots$, so let's try this:
- $f^*(x) = \lambda\left(1 - x^2\right)$.
- Normalize $f^*(x)$ to 1; that is, $\int_0^1 \lambda\left(1 - x^2\right) dx = 1$ or $\lambda = \frac{3}{2}$.

Importance sampling

- The Taylor series expansion of $\cos\left(\frac{\pi x}{2}\right) = 1 - \frac{\pi^2}{2 \times 2^2} x^2 + \ldots$, so let's try this:
- $f^*(x) = \lambda\left(1 - x^2\right)$.
- Normalize $f^*(x)$ to 1; that is, $\int_0^1 \lambda\left(1 - x^2\right) dx = 1$ or $\lambda = \frac{3}{2}$.
- $f^*(x) = \frac{3}{2}\left(1 - x^2\right)$.

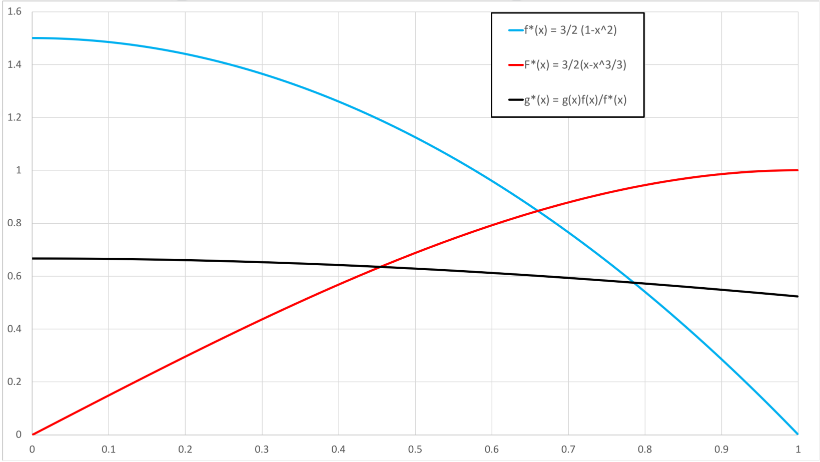
Importance sampling

- The Taylor series expansion of $\cos\left(\frac{\pi x}{2}\right) = 1 - \frac{\pi^2}{2 \times 2^2} x^2 + \ldots$, so let's try this:
- $f^*(x) = \lambda\left(1 - x^2\right)$.
- Normalize $f^*(x)$ to 1; that is, $\int_0^1 \lambda\left(1 - x^2\right) dx = 1$ or $\lambda = \frac{3}{2}$.
- $f^*(x) = \frac{3}{2}\left(1 - x^2\right)$.
- $F^*(x) = \frac{3}{2}\left(x - \frac{x^3}{3}\right)$.

Importance sampling

- The Taylor series expansion of $\cos\left(\frac{\pi x}{2}\right) = 1 - \frac{\pi^2}{2\times 2^2}x^2 + \ldots$, so let's try this:
- $f^*(x) = \lambda\left(1 - x^2\right)$.
- Normalize $f^*(x)$ to 1; that is, $\int_0^1 \lambda\left(1 - x^2\right) dx = 1$ or $\lambda = \frac{3}{2}$.
- $f^*(x) = \frac{3}{2}\left(1 - x^2\right)$.
- $F^*(x) = \frac{3}{2}\left(x - \frac{x^3}{3}\right)$.
- $g^*(x)$ is thus $\frac{\cos\left(\frac{\pi x}{2}\right)}{\frac{3}{2}(1-x^2)}$.

Variance reduction techniques



Importance sampling

- Whatever x is sampled, $g^*(x)$ will contribute meaningfully to the sum, driving more rapidly to "convergence".

Importance sampling

- Whatever x is sampled, $g^*(x)$ will contribute meaningfully to the sum, driving more rapidly to "convergence".
- In addition, we are more likely to sample x in regions where $g^*(x)$ has larger values.

Importance sampling

- Whatever x is sampled, $g^*(x)$ will contribute meaningfully to the sum, driving more rapidly to "convergence".
- In addition, we are more likely to sample x in regions where $g^*(x)$ has larger values.
- This practice is called *importance sampling*.

Variance reduction techniques

```
#! /usr/bin/env python

import random
import time
from math import pi   as pi
from math import cos  as cos
from math import sqrt as sqrt

tstart = time.time()

#  Calculate int_0^1 cos(pi * x / 2) dx with importance function f = 3/2(1-x^2)
#  F = 3/2(x-x^3/3), so solve x = (x^3+2R)/3 iteratively

random.seed(1999)
samples = 0; csum  = 0.0; c2sum = 0.0; cerr  = 1.0; third = 1.0/3.0

def rootfind(R):
  xold = 0.5; xnew = (xold**3 + 2.0*R)/3.0
  while (abs(xnew-xold)>1.e-10):
    xold = xnew; xnew = (xold**3 + 2.0*R)/3.0
  return xnew

# Continued on next slide
```

Importance sampling

```
# Continued from previous slide

while (cerr > 1.e-5):
  samples = samples + 1
  F = random.random(); x = rootfind(F)
  val = cos(pi*x/2.0) * 2.0/3.0 / (1.0 - x**2.0)
  csum  = csum + val; c2sum = c2sum + val**2
  cmean = csum/samples
  if (samples > 1 and cmean > 0.0):
    cvar = 1.0/(samples-1)*(c2sum/samples-cmean*cmean); cerr = sqrt(cvar)/cmean
  if (samples % 10000 == 0):
    print "%08d    %.8e    %.8e    %.8e" % (samples, cmean, cvar, cerr)

tend = time.time(); dt = tend - tstartA; duration = dt
hours = dt // 3600; dt = dt % 3600
mins = dt // 60; secs = dt % 60

print("Samples = %d" % (samples))
print "Integral = %.8f  (Exact = %.8f)" % (cmean,2.0/pi)
print "Variance = %.8e" % (cvar)
print "Error = %.8e" % (cerr)
print "Calculation took %d hr, %d min, %.3f seconds (%.3e s/sample) " %
       (hours, mins, secs, duration/samples)
```

Importance sampling

```
Samples = 24439299
Integral = 0.63662582  (Exact = 0.63661977)
Variance = 4.05292430e-11
Error = 9.99999996e-06
Calculation took 0 hr, 1 min, 40.222 seconds (4.101e-06 s/sample).
```

Importance sampling

- By using importance sampling, we converge to the proper result using a hundred times fewer samples.

Importance sampling

- By using importance sampling, we converge to the proper result using a hundred times fewer samples.
- "Convergence" here meaning the threshold error of 10^{-5}.

Importance sampling

- By using importance sampling, we converge to the proper result using a hundred times fewer samples.
- "Convergence" here meaning the threshold error of 10^{-5}.
- The *grind time* in seconds/sample is roughly $5\times$ greater because we're doing more work, but the decrease in samples more than makes up for it.

Importance sampling

- By using importance sampling, we converge to the proper result using a hundred times fewer samples.
- "Convergence" here meaning the threshold error of 10^{-5}.
- The *grind time* in seconds/sample is roughly $5\times$ greater because we're doing more work, but the decrease in samples more than makes up for it.
- Importance sampling is just one of a dizzying array of numerical techniques that can accelerate convergence to solution, collectively known as *variance reduction*.

Importance sampling

- By using importance sampling, we converge to the proper result using a hundred times fewer samples.
- "Convergence" here meaning the threshold error of 10^{-5}.
- The *grind time* in seconds/sample is roughly $5\times$ greater because we're doing more work, but the decrease in samples more than makes up for it.
- Importance sampling is just one of a dizzying array of numerical techniques that can accelerate convergence to solution, collectively known as *variance reduction*.
- Analog MC simulations are quite rare.

A simple physics application

- Consider the steady-state 1D problem of a slab exposed on its left side to an incident plane source of monoenergetic particles.

A simple physics application

- Consider the steady-state 1D problem of a slab exposed on its left side to an incident plane source of monoenergetic particles.
- The particles can scatter isotropically off of atoms in the wall.

A simple physics application

- Consider the steady-state 1D problem of a slab exposed on its left side to an incident plane source of monoenergetic particles.
- The particles can scatter isotropically off of atoms in the wall.
- We want to know what fraction of source particles come out the right side traveling 30 to 45 degrees off-normal.

A simple physics application

A simple physics application

- If the probability of scatter per unit length traveled by a particle in the wall is Σ, the probability of traveling a distance s before scattering is $1 - \exp(-\Sigma s)$. That's our first CDF F_1, with PDF $f_1 = \frac{1}{\Sigma}\exp(-\Sigma s)$.

A simple physics application

- If the probability of scatter per unit length traveled by a particle in the wall is Σ, the probability of traveling a distance s before scattering is $1 - \exp(-\Sigma s)$. That's our first CDF F_1, with PDF $f_1 = \frac{1}{\Sigma} \exp(-\Sigma s)$.
- Since the particle scatters isotropically when it scatters, the CDF for direction of travel μ on $(-1, 1)$ is $F_2 = \frac{1}{2}$ and the PDF is $f_2 = \frac{1}{2} (\mu - 1)$.

A simple physics application

- If the probability of scatter per unit length traveled by a particle in the wall is Σ, the probability of traveling a distance s before scattering is $1 - \exp(-\Sigma s)$. That's our first CDF F_1, with PDF $f_1 = \frac{1}{\Sigma}\exp(-\Sigma s)$.
- Since the particle scatters isotropically when it scatters, the CDF for direction of travel μ on $(-1, 1)$ is $F_2 = \frac{1}{2}$ and the PDF is $f_2 = \frac{1}{2}(\mu - 1)$.
- We start throwing particles and tracking them one at a time until it exits the slab. The left side is non-reentrant, so if the particle exits to the left, its history ends.

A simple physics application

- If the probability of scatter per unit length traveled by a particle in the wall is Σ, the probability of traveling a distance s before scattering is $1 - \exp(-\Sigma s)$. That's our first CDF F_1, with PDF $f_1 = \frac{1}{\Sigma}\exp(-\Sigma s)$.

- Since the particle scatters isotropically when it scatters, the CDF for direction of travel μ on $(-1, 1)$ is $F_2 = \frac{1}{2}$ and the PDF is $f_2 = \frac{1}{2}(\mu - 1)$.

- We start throwing particles and tracking them one at a time until it exits the slab. The left side is non-reentrant, so if the particle exits to the left, its history ends.

- If it exits to the right, its history also ends, but we check its angle and *tally* the particles with the proper angle.

A simple physics application

- If the probability of scatter per unit length traveled by a particle in the wall is Σ, the probability of traveling a distance s before scattering is $1 - \exp(-\Sigma s)$. That's our first CDF F_1, with PDF $f_1 = \frac{1}{\Sigma} \exp(-\Sigma s)$.
- Since the particle scatters isotropically when it scatters, the CDF for direction of travel μ on $(-1, 1)$ is $F_2 = \frac{1}{2}$ and the PDF is $f_2 = \frac{1}{2} (\mu - 1)$.
- We start throwing particles and tracking them one at a time until it exits the slab. The left side is non-reentrant, so if the particle exits to the left, its history ends.
- If it exits to the right, its history also ends, but we check its angle and *tally* the particles with the proper angle.
- Keep throwing particles until the answer converges to a desire

A simple physics application

```
#! /usr/bin/env python

import random
import time
from math import log  as ln
from math import sqrt as sqrt
from math import cos  as cos

tstart = time.time()

#  Given:
#          Slab, low-x = 0 cm, high-x = 40 cm
#          Sigma = 0.25 cm^{-1}  (isotropic)

random.seed(1999)
ptcl = 0; lsum  = 0.0; l2sum = 0.0

# Continued on next slide
```

A simple physics application

```
# Continued from previous slide

def trackparticle(particle,l):
   (ptcl, x, mu) = particle; (lsum, l2sum) = l
   Sigma = 0.25; width = 40.0
   cosmax = sqrt(3.0)/2.0; cosmin = sqrt(2.0)/2.0  #  cos 30, 45

   particle_is_tracking = 1
   while (particle_is_tracking):
      #  We need to track the particle to collision
      distance_to_collision = -ln(random.random()) / Sigma
      xprime = x + distance_to_collision * mu
      if (xprime > width): #  Particle exited to the right
         if (mu >= cosmin and mu <= cosmax): # Within 30 and 45 degrees
           lsum = lsum + 1; l2sum = l2sum + 1
         particle_is_tracking = 0
      elif (xprime < 0) : # Particle exited to the left
         particle_is_tracking = 0
      else: # Particle scattered, update location and direction
         x = xprime; mu = -1.0 + 2.0*random.random()
   return (lsum,l2sum)
# Continued on next slide
```

A simple physics application

```
# Continued from previous slide

lerr = 1.0
while (lerr >= 1.0e-3):
   #  Initialize particle
   ptcl = ptcl + 1; x = 0.0; mu = 1.0; particle = (ptcl, x, mu)
   (lsum,l2sum) = trackparticle(particle,(lsum,l2sum))

   lmean = lsum/ptcl
   if (ptcl > 1 and lmean > 0):
      lvar = 1.0/(ptcl-1)*(l2sum/ptcl-lmean*lmean); lerr = sqrt(lvar)/lmean

tend = time.time(); dt = tend - tstart; duration = dt
hours = dt // 3600; dt = dt % 3600
mins = dt // 60; secs = dt % 60

print("\nParticles = %d" % (ptcl))
print "Tally = %.9e, error = %.9e\n" % (lmean, lerr)
print "Calculation took %d hr, %d min, %.3f seconds (%.3e s/sample)." % \
       (hours, mins, secs, duration/ptcl)
```

A simple physics application

```
Particles = 23858000
Tally = 4.023006958e-02, error = 9.999796680e-04

Calculation took 0 hr, 3 min, 45.704 seconds (9.460e-06 s/sample).
```

More complicated physics applications

- In realistic neutron transport, there are seven independent variables (3 space, 2 direction, 1 energy, 1 time).

More complicated physics applications

- In realistic neutron transport, there are seven independent variables (3 space, 2 direction, 1 energy, 1 time).
- Many different reaction types occur between neutrons and nuclei.

More complicated physics applications

- In realistic neutron transport, there are seven independent variables (3 space, 2 direction, 1 energy, 1 time).
- Many different reaction types occur between neutrons and nuclei.
- Analytic solutions are essentially impossible to find.

More complicated physics applications

- In realistic neutron transport, there are seven independent variables (3 space, 2 direction, 1 energy, 1 time).
- Many different reaction types occur between neutrons and nuclei.
- Analytic solutions are essentially impossible to find.
- Monte Carlo sampling from a many-dimensional independent variable space is much more efficient than the usual discretization methods.

Coming soon

- Tomorrow Avneet Sood will talk about LANL's flagship code for neutral-particle transport, MCNP. Many of the concepts of today's lecture will be mentioned.

Coming soon

- Tomorrow Avneet Sood will talk about LANL's flagship code for neutral-particle transport, MCNP. Many of the concepts of today's lecture will be mentioned.
- Questions?