# Lecture 2: Introduction to Monte Carlo methods

Mike Giles

Mathematical Institute

# Objective

There are many situations in which we want to estimate the average value of some random quantity

In general, we

- start with a random sample $\omega$ (which might correspond to a set of random numbers)
- usually compute some intermediate quantity $U$
- then evaluate a scalar output $f(U)$

$$\omega \;\rightarrow\; U \;\rightarrow\; f(U)$$

The objective is then to compute the **expected** (or average) value

$$\mathbb{E}[f(U)]$$

# Basics

In some cases, the random inputs are discrete: $X$ has value $x_i$ with probability $p_i$, and then

$$\mathbb{E}[f(X)] = \sum_i f(x_i)\, p_i$$

In other cases, the random inputs are continuous random variables: $X$ has probability density $p(x)$ if $\mathbb{P}(X \in (x, x+\mathrm{d}x)) \approx p(x)\,\mathrm{d}x$ and then

$$\mathbb{E}[f(X)] = \int f(x)\, p(x)\, \mathrm{d}x$$

In either case, if $a, b$ are random variables, and $\lambda, \mu$ are constants,

$$\begin{aligned} \mathbb{E}[a+\mu] &= \mathbb{E}[a] + \mu \\ \mathbb{E}[\lambda\, a] &= \lambda\, \mathbb{E}[a] \\ \mathbb{E}[a+b] &= \mathbb{E}[a] + \mathbb{E}[b] \end{aligned}$$

# Basics

The **variance** is defined as

$$\begin{aligned} \mathbb{V}[a] &\equiv \mathbb{E}\left[ (a - \mathbb{E}[a])^2 \right] \\ &= \mathbb{E}\left[ a^2 - 2a\,\mathbb{E}[a] + (\mathbb{E}[a])^2 \right] \\ &= \mathbb{E}\left[ a^2 \right] - (\mathbb{E}[a])^2 \end{aligned}$$

It then follows that

$$\begin{aligned} \mathbb{V}[a+\mu] &= \mathbb{V}[a] \\ \mathbb{V}[\lambda\, a] &= \lambda^2\, \mathbb{V}[a] \\ \mathbb{V}[a+b] &= \mathbb{V}[a] + 2\,\mathrm{Cov}[a,b] + \mathbb{V}[b] \end{aligned}$$

where

$$\mathrm{Cov}[a,b] \equiv \mathbb{E}\left[ (a - \mathbb{E}[a])\,(b - \mathbb{E}[b]) \right]$$

# Basics

$X_1$ and $X_2$ are independent continuous random variables if

$$p_{\rm joint}(x_1,x_2) = p_1(x_1)\ p_2(x_2)$$

We then get

$$\begin{aligned}
\mathbb{E}[f_1(X_1)\ f_2(X_2)] &= \iint f_1(x_1)\ f_2(x_2)\ p_{\rm joint}(x_1,x_2)\ {\rm d}x_1\ {\rm d}x_2 \\
&= \iint f_1(x_1)\ f_2(x_2)\ p_1(x_1)\ p_2(x_2)\ {\rm d}x_1\ {\rm d}x_2 \\
&= \left( \int f_1(x_1)\ p_1(x_1)\ {\rm d}x_1 \right) \left( \int f_2(x_2)\ p_2(x_2)\ {\rm d}x_2 \right) \\
&= \mathbb{E}[f_1(X_1)]\ \ \mathbb{E}[f_2(X_2)]
\end{aligned}$$

So, if $a,b$ are independent, ${\rm Cov}[a,b]\!=\!0 \Longrightarrow \mathbb{V}[a\!+\!b] = \mathbb{V}[a] + \mathbb{V}[b]$

More generally, the variance of the sum of independent r.v.'s is the sum of their variances.

# Random Number Generation

Monte Carlo simulation starts with random number generation, usually split into 2 stages:

- generation of independent uniform $(0,1)$ random variables
- conversion into random variables with a particular distribution (e.g. Normal)

**Very important:** never write your own generator, always use a well validated generator from a reputable source

- python
- MATLAB
- Intel MKL (Math Kernel Library)

# Uniform Random Variables

Pseudo-random generators use a deterministic (i.e. repeatable) algorithm to generate a sequence of (apparently) random numbers on $(0, 1)$ interval.

What defines a good generator?

- a long period – how long it takes before the sequence repeats itself $2^{32}$ is not enough (need at least $2^{40}$)
- various statistical tests to measure "randomness" – well validated software will have gone through these checks

For information see

- Intel MKL information
  `www.intel.com/cd/software/products/asmo-na/eng/266864.htm`
- Matlab information
  `www.mathworks.com/moler/random.pdf`
- Wikipedia information
  `en.wikipedia.org/wiki/Random_number_generation`

# Normal Random Variables

$N(0,1)$ Normal random variables (mean 0, variance 1) have the probability density function

$$p(x) \;=\; \frac{1}{\sqrt{2\pi}}\,\exp(-\,\tfrac{1}{2}x^2) \;\equiv\; \phi(x)$$

If $X$ is a $N(0,1)$ Normal random variable, then its CDF (Cumulative Distribution Function) is defined as

$$\mathbb{P}[X < x] = \int_{-\infty}^{x} \phi(x)\,\mathrm{d}x \equiv \Phi(x)$$

Many maths software libraries include the function $\Phi(x)$, along with $\sin, \cos, \exp, \log$ and others. In python it is norm.cdf from scipy.stats

# Normal Random Variables

The Box-Muller transformation method takes two independent uniform $(0, 1)$ random numbers $y_1, y_2$, and defines

$$\begin{aligned} x_1 &= \sqrt{-2\log(y_1)}\; \cos(2\pi y_2) \\ x_2 &= \sqrt{-2\log(y_1)}\; \sin(2\pi y_2) \end{aligned}$$

It can be proved that $x_1$ and $x_2$ are $N(0, 1)$ random variables, and independent:

$$p_{\rm joint}(x_1, x_2) = p(x_1)\; p(x_2)$$

# Inverse CDF

An alternative uses the cumulative distribution function $\Phi(x)$.

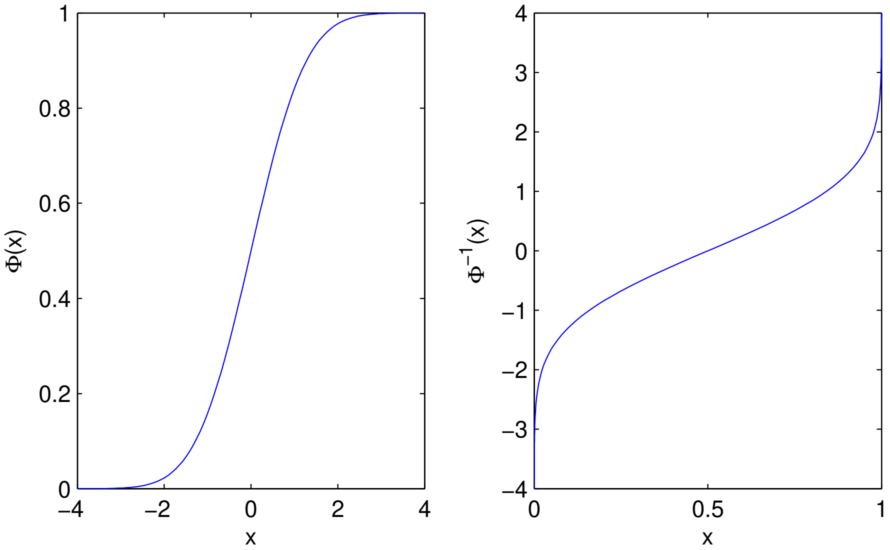
If $X$ is a $N(0,1)$ random variable, then $Y = \Phi(X)$ is a uniform $(0,1)$ random variable.

Hence, can start with a uniform $(0,1)$ random variable $Y$ and define $X$ by

$$X = \Phi^{-1}(Y)$$

$\Phi^{-1}(y)$ is approximated in software in a very similar way to other functions like $\cos, \sin, \log, \exp$. In python it is norm.ppf from scipy.stats

# Normal Random Variables



# Monte Carlo estimate

If we have a sequence $f_n$ of $N$ independent samples of $f$, the average

$$\overline{f} = N^{-1} \sum_{n=1}^{N} f_n.$$

is the Monte Carlo estimate of the expected value $\mathbb{E}[f]$

It is an unbiased estimate, since for each $n$,

$$\mathbb{E}[f_n] = \mathbb{E}[f] \quad \Longrightarrow \quad \mathbb{E}\left[\,\overline{f}\,\right] = \mathbb{E}[f]$$

We also have

$$\mathbb{V}[\,\overline{f}\,] = N^{-2}\, \mathbb{V}\left[\sum_{n=1}^{N} f_n\right] = N^{-2} \sum_{n=1}^{N} \mathbb{V}[f_n] = N^{-1} \mathbb{V}[f]$$

# Central Limit Theorem

The Central Limit Theorem says that if the variance $\sigma^2 \equiv \mathbb{V}[f]$ is finite, then the error

$$e_N(f) = \overline{f} - \mathbb{E}[f]$$

is approximately Normal in distribution for large $N$, i.e.

$$e_N(f) \sim \sigma \, N^{-1/2} \, Z$$

where $Z$ is a $N(0,1)$ random variable

# Central Limit Theorem

If $Z$ is a $N(0,1)$ random variable, then due to symmetry

$$\mathbb{P}[|Z| < s] = 1 - \mathbb{P}[|Z| > s] = 1 - 2\,\mathbb{P}[Z < -s] = 1 - 2\,\Phi(-s)$$

Table of probabilities for different $s$:

| $s$ | 1.0 | 2.0 | 3.0 | 4.0 |
|---|---|---|---|---|
| Prob | 0.683 | 0.9545 | 0.9973 | 0.99994 |

Hence, with probability 99.7%, $|e_N(f)| < 3\sigma N^{-1/2}$

$$\Longrightarrow \quad \mathbb{E}[f] \in (\overline{f} - 3\sigma N^{-1/2}, \overline{f} + 3\sigma N^{-1/2})$$

This is the confidence interval for $\mathbb{E}[f]$

# Estimated Variance

Given $N$ samples, the empirical variance is

$$\widetilde{\sigma}^2 = N^{-1} \sum_{n=1}^{N} \left(f_n - \overline{f}\,\right)^2 = \overline{f^2} - (\,\overline{f}\,)^2$$

where

$$\overline{f} = N^{-1} \sum_{n=1}^{N} f_n, \qquad \overline{f^2} = N^{-1} \sum_{n=1}^{N} f_n^2$$

$\widetilde{\sigma}^2$ is a slightly biased estimator for $\sigma^2$; an unbiased estimator is

$$\widehat{\sigma}^2 = (N{-}1)^{-1} \sum_{n=1}^{N} \left(f_n - \overline{f}\,\right)^2 = \frac{N}{N{-}1} \left(\overline{f^2} - (\,\overline{f}\,)^2\right)$$

# Finance Applications

Geometric Brownian motion for a single asset:

$$S_T = S_0 \ \exp\left((r - \tfrac{1}{2}\sigma^2)\, T + \sigma\, W_T\right)$$

$W_T$ is $N(0, T)$ random variable, so can put

$$W_T = \sqrt{T}\ Z$$

where $Z$ is a $N(0,1)$ random variable.

We are then interested in the price of financial options which can be expressed as

$$V = \mathbb{E}\left[f(S_T)\right]$$

for some "payoff" function $f(S)$.

# Finance Applications

For the European call option,

$$f(S) = \exp(-rT)\ \max(S-K, 0)$$

while for the European put option

$$f(S) = \exp(-rT)\ \max(K-S, 0)$$

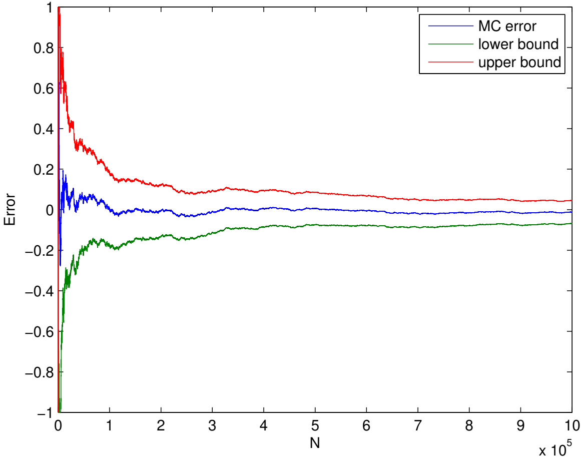
where $K$ is the strike price

For numerical experiments we will consider a European call with
$r = 0.05, \quad \sigma = 0.2, \quad T = 1, \quad S_0 = 110, \quad K = 100.$

The analytic value is known for comparison.

# Finance Applications

MC calculation with up to $10^6$ paths; true value = 17.663



# Finance Applications

The upper and lower bounds are given by

$$\text{Mean} \pm \frac{3\,\widetilde{\sigma}}{\sqrt{N}},$$

so more than a 99.7% probability that the true value lies within these bounds.

# SDEs in Finance

In computational finance, stochastic differential equations are used to model the behaviour of

- stocks
- interest rates
- exchange rates
- weather
- electricity/gas demand
- crude oil prices
- . . .

# SDEs in Finance

Stochastic differential equations are just ordinary differential equations plus an additional random source term.

The stochastic term accounts for the uncertainty of unpredictable day-to-day events.

The aim is **not** to predict exactly what will happen in the future, but to predict the probability of a range of possible things that **might** happen, and compute some averages, or the probability of an excessive loss.

This is really what is known more generally as Uncertainty Quantification – the finance industry has been doing it for a long time because they have so much uncertainty.

# SDEs in Finance

Examples:

- Geometric Brownian motion (Black-Scholes model for stock prices)

$$\mathrm{d}S = r\,S\,\mathrm{d}t + \sigma\,S\,\mathrm{d}W$$

- Cox-Ingersoll-Ross model (interest rates)

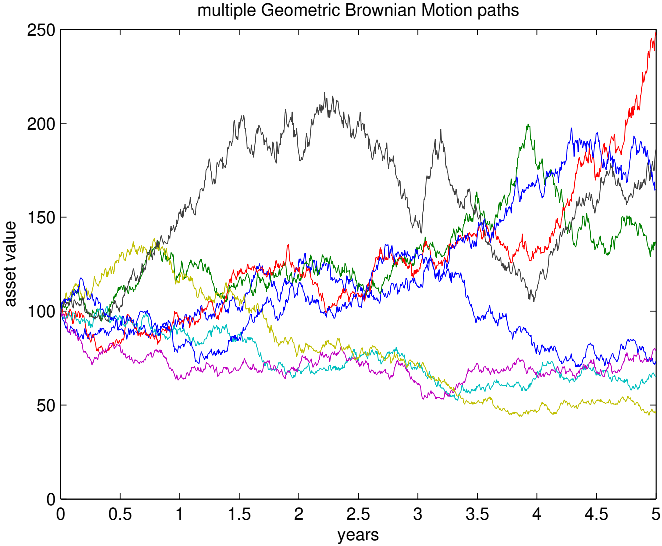
$$\mathrm{d}r = \alpha(b-r)\,\mathrm{d}t + \sigma\,\sqrt{r}\,\mathrm{d}W$$

- Heston stochastic volatility model (stock prices)

$$\begin{aligned}\mathrm{d}S &= r\,S\,\mathrm{d}t + \sqrt{V}\,S\,\mathrm{d}W_1 \\ \mathrm{d}V &= \lambda\,(\sigma^2 - V)\,\mathrm{d}t + \xi\,\sqrt{V}\,\mathrm{d}W_2\end{aligned}$$

with correlation $\rho$ between $\mathrm{d}W_1$ and $\mathrm{d}W_2$

# SDEs in Finance



# Generic Problem

Stochastic differential equation with general drift and volatility terms:

$$\mathrm{d}S_t = a(S_t, t)\,\mathrm{d}t + b(S_t, t)\,\mathrm{d}W_t$$

$W_t$ is a Wiener variable with the properties that for any $q{<}r{<}s{<}t$, $W_t{-}W_s$ is Normally distributed with mean 0 and variance $t{-}s$, independent of $W_r{-}W_q$.

In many finance applications, we want to compute the expected value of an option dependent on the terminal state $P(S_T)$

Other options depend on the average, minimum and/or maximum over the whole time interval.

# Euler discretisation

Given the generic SDE:

$$\mathrm{d}S_t = a(S_t)\,\mathrm{d}t + b(S_t)\,\mathrm{d}W_t, \quad 0 < t < T,$$

the Euler discretisation with timestep $\Delta t$ is:

$$\widehat{S}_{n+1} = \widehat{S}_n + a(\widehat{S}_n)\,\Delta t + b(\widehat{S}_n)\,\Delta W_n$$

where $\Delta W_n$ are independent Normal random variables with mean 0, variance $\Delta t$.

This will be our second model application in the course.