Monte Carlo Methods

Prof. Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Monte Carlo methods

In option pricing there are two main approaches:

- Monte Carlo methods for estimating expected values of financial payoff functions based on underlying assets.

  E.g., we want to estimate $\mathbb{E}[f(S(T))]$ where

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

  and $W(T)$ is driving Brownian motion at terminal time T
- Numerical approximation of the PDE which describes the evolution of the expected value.

  $$u(s,t) = \mathbb{E}\left[f(S(T)) \mid S(t) = s\right]$$

  Usually less costly than MC when there are very few underlying assets ($M \leq 3$), but much more expensive when there are many.

Geometric Brownian Motion

In this first lecture, we consider M underlying assets, each modelled by Geometric Brownian Motion

$$\mathrm{d}S_i = r\, S_i\, \mathrm{d}t + \sigma_i\, S_i\, \mathrm{d}W_i$$

so Ito calculus gives us

$$S_i(T) = S_i(0)\ \exp\left((r - \tfrac{1}{2}\sigma_i^2)T + \sigma_i W_i(T) \right)$$

in which each $W_i(T)$ is Normally distributed with zero mean and variance T.

We can use standard Random Number Generation software (e.g. `randn` function in MATLAB) to generate samples of each $W_i(T)$, but there is a problem . . .

Correlated Brownian Motions

Different assets do not behave independently – on average, they tend to move up and down together.

This is modelled by introducing correlation between the driving Brownian motions so that

$$\mathbb{E}\left[W_i(T)\ W_j(T)\right] = \Omega_{i,j}\, T$$

where $\Omega_{i,j}$ is the correlation coefficient, and hence

$$\mathbb{E}\left[W(T)\ W(T)^T\right] = \Omega\, T.$$

How do we generate samples of $W_i(T)$?

Correlated Normal Random Variables

Suppose x is a vector of independent $N(0,1)$ variables, and define a new vector $y = L\,x$.

Each element of y is Normally distributed, $\mathbb{E}[y] = L\,\mathbb{E}[x] = 0$, and

$$\mathbb{E}[y\,y^T] = \mathbb{E}[L\,x\,x^T\,L^T] = L\,\mathbb{E}[x\,x^T]\,L^T = L\,L^T.$$

since $\mathbb{E}[x\,x^T] = I$ because

- elements of x are independent $\Longrightarrow \mathbb{E}[x_i\,x_j] = 0$ for $i \neq j$
- elements of x have unit variance $\Longrightarrow \mathbb{E}[x_i^2] = 1$

Correlated Normal Random Variables

To get $\mathbb{E}[y\, y^T] = \Omega$, we need to find L such that

$$L\, L^T = \Omega$$

L is not uniquely defined – simplest choice is to use a Cholesky factorization in which L is lower-triangular, with a positive diagonal.

In MATLAB, this is achieved using

```
L = chol(Omega,'lower');
```

Basket Call Option

In a basket call option, the discounted payoff function is

$$f = \exp(-r\,T) \left(\frac{1}{M} \sum_i S_i(T) - K \right)^+$$

where K is the strike, r is the risk-free interest rate, and $(x)^+ \equiv \max(x,0)$.

Monte Carlo estimation of $\mathbb{E}[f(S(T))]$ is very simple – we generate N independent samples of $W(T)$, compute $S(T)$, and then average to get

$$\overline{f}_N \equiv N^{-1} \sum_{n=1}^{N} f^{(n)} \;\approx\; \mathbb{E}[f]$$

Monte Carlo Error and CLT

This MC estimate is unbiased, meaning that the average error is zero

$$\mathbb{E}[\varepsilon_N] = 0$$

where ε_N is the error $\overline{f}_N - \mathbb{E}[f]$.

In addition, the Central Limit Theorem proves that for large N the error is asymptotically Normally distributed

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

with Z a $N(0,1)$ random variable and σ^2 the variance of f:

$$\sigma^2 = \mathbb{V}[f] \equiv \mathbb{E}\left[(f - \mathbb{E}[f])^2 \right].$$

CLT

This means that

$$\mathbb{P}\left[\left|N^{1/2}\sigma^{-1}\varepsilon_N\right| < s\right] \;\approx\; 1 - 2\,\Phi(-s),$$

where $\Phi(s)$ is the Normal CDF (cumulative distribution function).

Typically we use $s=3$, corrsponding to a 3-standard deviation confidence interval, with $1-2\,\Phi(-s) \approx 0.997$.

Hence, with probability 99.7%, we have

$$\left|N^{1/2}\sigma^{-1}\varepsilon_N\right| < 3 \;\Longrightarrow\; |\varepsilon_N| < 3\,\sigma\,N^{-1/2}$$

This bounds the accuracy, but we need an estimate for σ.

Empirical Variance

Given N samples, the empirical variance is

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

where

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

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

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

Applications

Geometric Brownian motion for single asset:

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

For the European call option,

$$f(S) = \exp(-rT)\ (S-K)^+$$

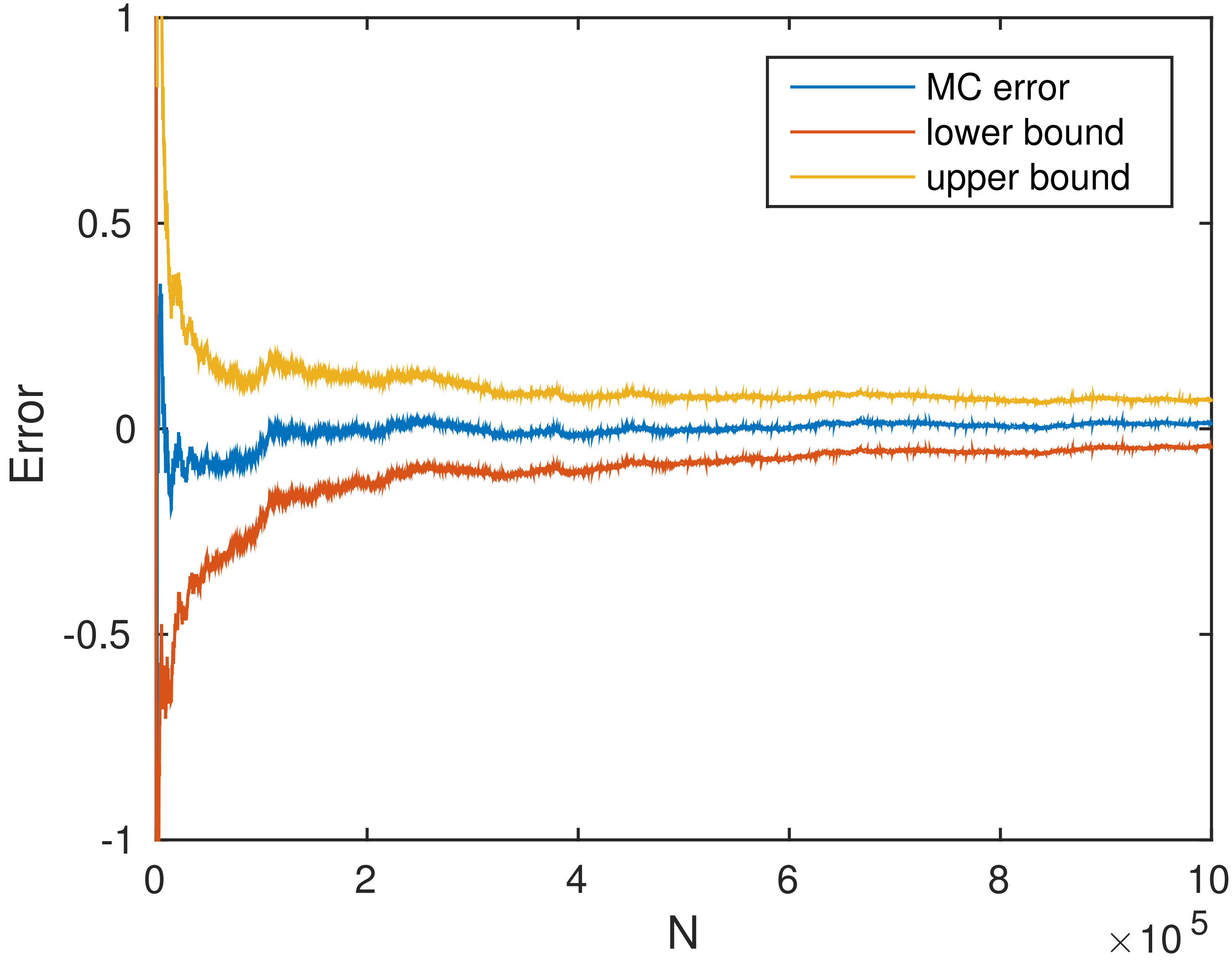
where K is the strike price.

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

The analytic value is known for comparison.

Applications

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



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.

Applications

MATLAB code:

```
r=0.05;  sig=0.2;  T=1;  S0=110;  K=100;
N = 1:1000000;
Y = randn(1,max(N));    % Normal random variables
S = S0*exp((r-sig^2/2)*T + sig*sqrt(T)*Y);
F = exp(-r*T)*max(0,S-K);

sum1 = cumsum(F);      % cumulative summation of
sum2 = cumsum(F.^2);   % payoff and its square
val  = sum1./N;
sd   = sqrt(sum2./N - val.^2);
```

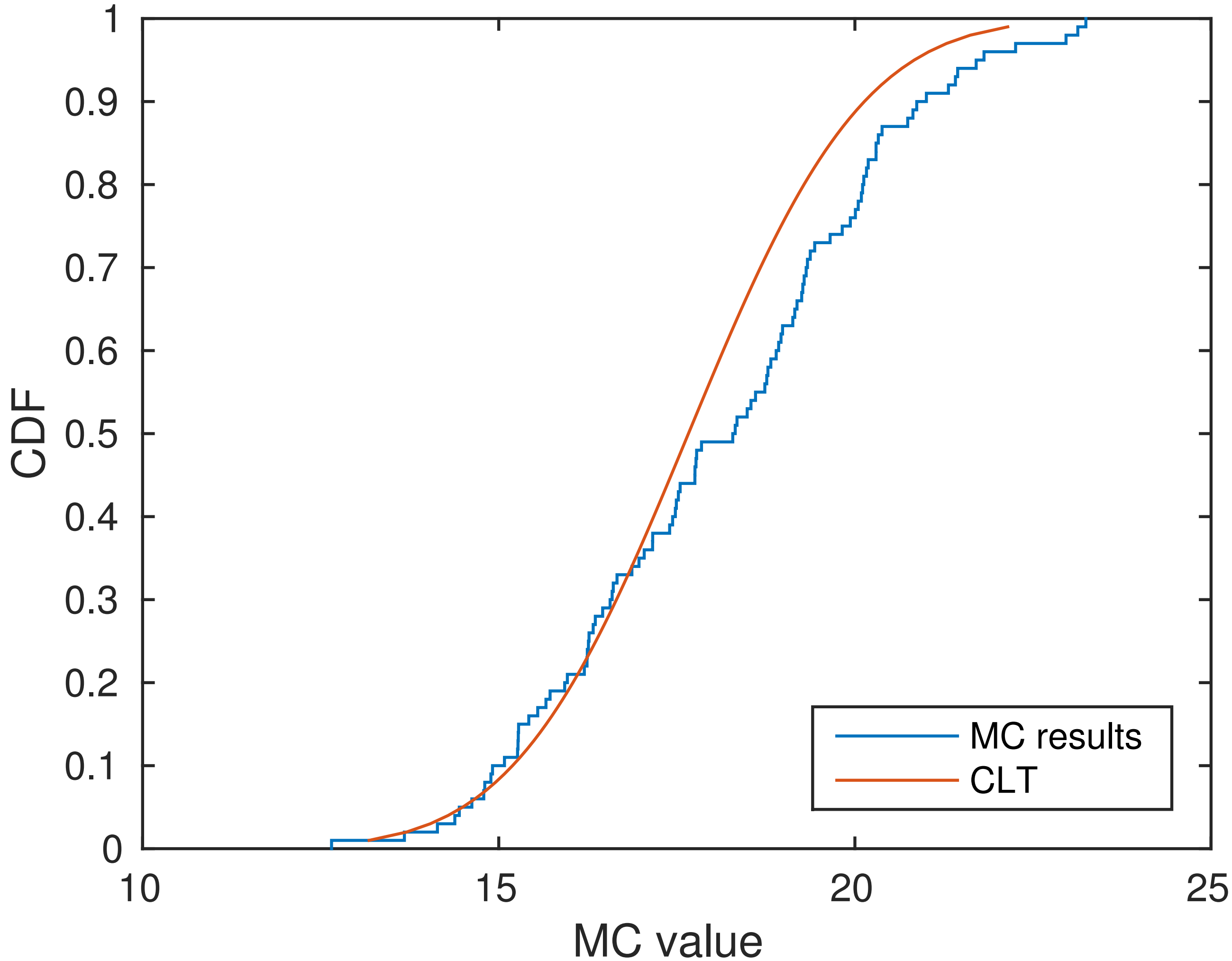
Applications

```
err = european_call(r,sig,T,S0,K,'value') - val;

plot(N,err,                    ...
     N,err-3*sd./sqrt(N),   ...
     N,err+3*sd./sqrt(N))
axis([0 length(N) -1 1])
xlabel('N'); ylabel('Error')
legend('MC error','lower bound','upper bound')
```

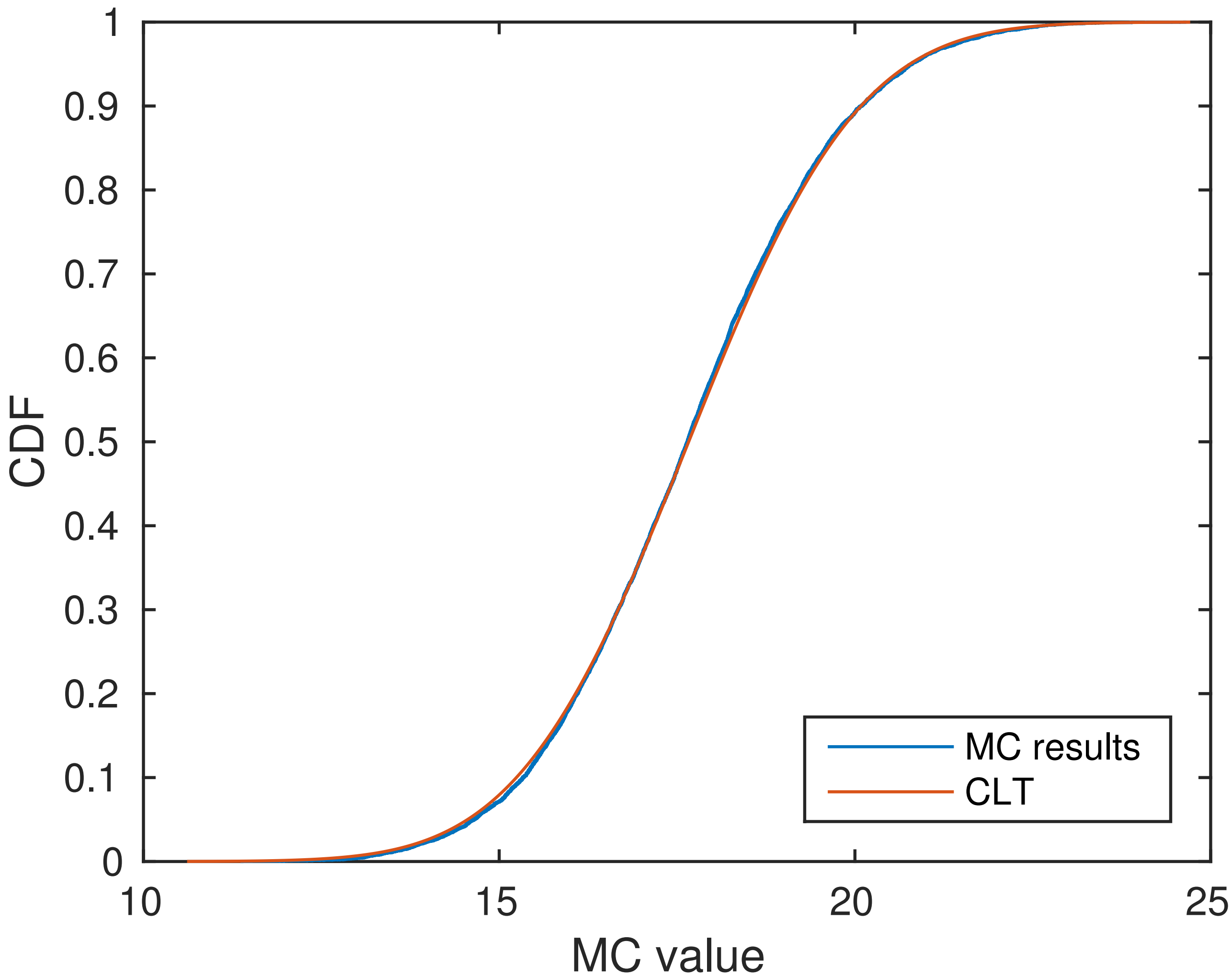
Applications

CLT: 100 independent tests, each with 100 samples



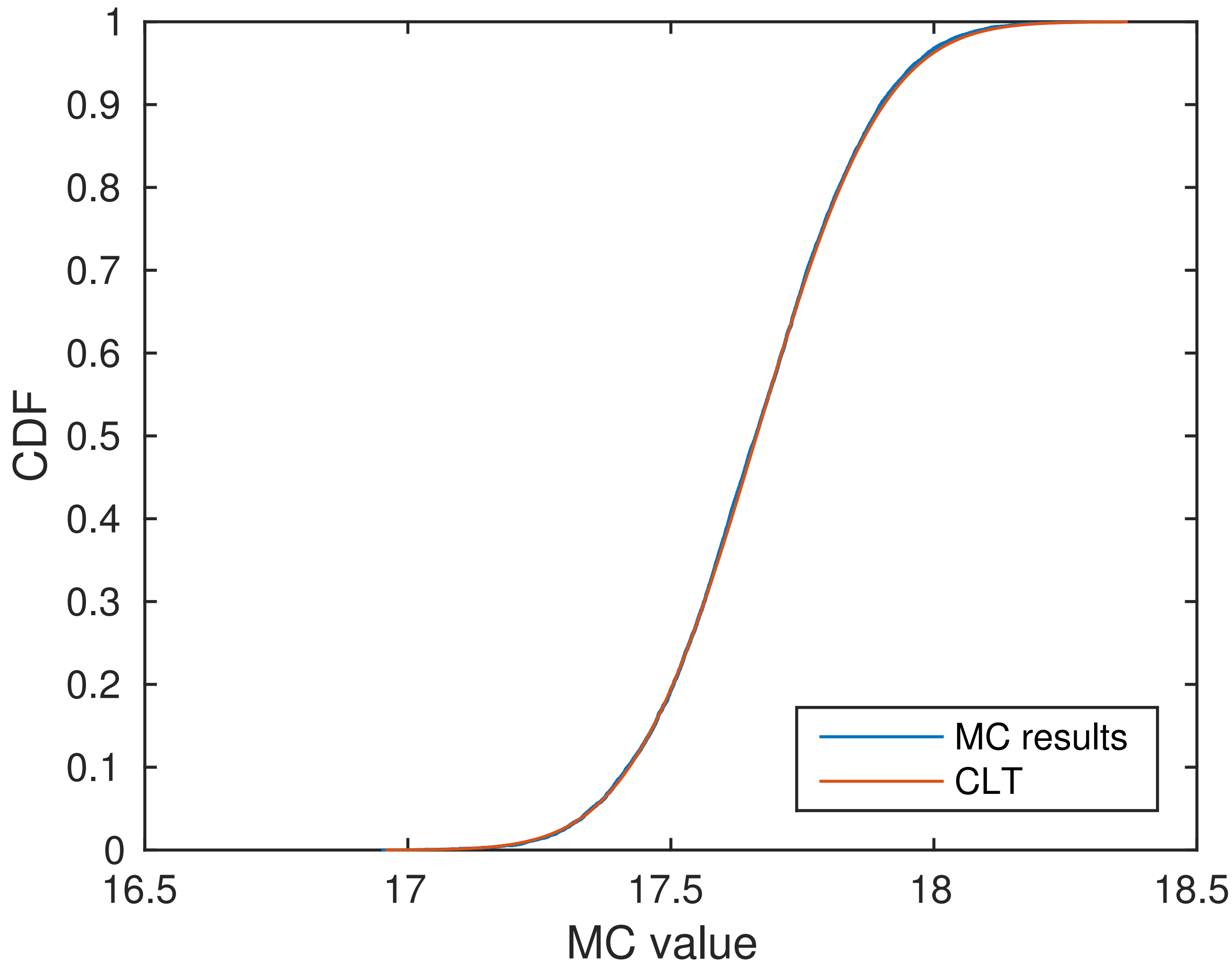
Applications

CLT: 10^4 independent tests, each with 100 samples



Applications

CLT: 10^4 independent tests, each with 10^4 samples



Basket call option

- 5 underlying assets starting at $S_0 = 100$, with call option on arithmetic mean with strike $K = 100$
- Geometric Brownian Motion model, $r = 0.05, T = 1$
- volatility $\sigma = 0.2$ and correlation matrix

$$\Omega = \begin{pmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{pmatrix}$$

Applications

```
r=0.05; sigma=0.2; rho=0.1; T=1; K=100; S0=100;

N = 10^5;                         % number of MC samples
Omega = eye(5) + rho*(ones(5)-eye(5));
L = chol(Omega,'lower');   % Cholesky factorisation
W = sqrt(T)*L*randn(5,N);
S = S0.*exp((r-0.5*sigma^2)*T + sigma*W);

S = 0.2*sum(S,1);                 % average asset value
F = exp(-r*T)*max(S-K,0); % call option

val = sum(F)/N                    % mean and its std. dev.
sd  = sqrt( (sum(F.^2)/N - val.^2)/(N-1) )
```

Variance Reduction

Monte Carlo is a very simple method; it gets complicated when we try to reduce the variance, and hence the number of samples required.

There are several approaches:

- antithetic variables
- control variates
- importance sampling
- stratified sampling
- Latin hypercube
- quasi-Monte Carlo

We will discuss control variates.

Review of elementary results

If a, b are random variables, and λ, μ are constants, then

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

where

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

Review of elementary results

If a, b are independent random variables then

$$\mathbb{E}[f(a)\; g(b)] = \mathbb{E}[f(a)]\; \mathbb{E}[g(b)]$$

Hence, Cov$[a,b]=0$ and therefore $\mathbb{V}[a+b] = \mathbb{V}[a] + \mathbb{V}[b]$

Extending this to a set of N iid (independent identically distributed) r.v.'s x_n, we have

$$\mathbb{V}\left[\sum_{n=1}^{N} x_n\right] = \sum_{n=1}^{N} \mathbb{V}[x_n] = N\, \mathbb{V}[x]$$

and so

$$\mathbb{V}\left[N^{-1} \sum_{n=1}^{N} x_n\right] = N^{-1}\, \mathbb{V}[x]$$

Control Variates

Suppose we want to approximate $\mathbb{E}[f]$ using a simple Monte Carlo average $\overline{f}$.

If there is another payoff g for which we know $\mathbb{E}[g]$, can use $\overline{g} - \mathbb{E}[g]$ to reduce error in $\overline{f} - \mathbb{E}[f]$.

How? By defining a new estimator

$$\widehat{f} = \overline{f} - \lambda \left(\overline{g} - \mathbb{E}[g]\right)$$

Again unbiased since $\mathbb{E}[\widehat{f}] = \mathbb{E}[\overline{f}] = \mathbb{E}[f]$

Control Variates

For a single sample,

$$\mathbb{V}[f - \lambda\,(g - \mathbb{E}[g])] = \mathbb{V}[f] - 2\,\lambda\,\mathsf{Cov}[f,g] + \lambda^2\,\mathbb{V}[g]$$

For an average of N samples,

$$\mathbb{V}[\overline{f} - \lambda\,(\overline{g} - \mathbb{E}[g])] = N^{-1}\left(\mathbb{V}[f] - 2\,\lambda\,\mathsf{Cov}[f,g] + \lambda^2\,\mathbb{V}[g]\right)$$

To minimise this, the optimum value for λ is

$$\lambda = \frac{\mathsf{Cov}[f,g]}{\mathbb{V}[g]}$$

Control Variates

The resulting variance is

$$N^{-1}\,\mathbb{V}[f]\left(1-\frac{(\textsf{Cov}[f,g])^2}{\mathbb{V}[f]\,\mathbb{V}[g]}\right)=N^{-1}\,\mathbb{V}[f]\left(1-\rho^2\right)$$

where ρ is the correlation between f and g.

The challenge is to choose a good g which is well correlated with f – the covariance, and hence the optimal λ, can be estimated from the data.

Control Variates

For a basket option with M underlying assets, we know that for each asset

$$\mathbb{E}[\exp(-r\,T)\,S_i(T)] = S_i(0)$$

so we could use

$$g = \exp(-r\,T)\,\frac{1}{M}\sum_{m=1}^{M} S_i(T)$$

with

$$\mathbb{E}[g] = \frac{1}{M}\sum_{m=1}^{M} S_i(0)$$

Numerical test will do the simpler scalar case with $M\!=\!1$.

Application

MATLAB code, part 1 – estimating optimal λ:

```
r=0.05; sig=0.2; T=1; S0=110; K=100;

N = 1000;
Y = randn(1,N);          % Normal random variable
S = S0*exp((r-sig^2/2)*T + sig*sqrt(T)*Y);
F = exp(-r*T)*max(0,S-K);
C = exp(-r*T)*S;
Fave = sum(F)/N;
Cave = sum(C)/N;
lam = sum((F-Fave).*(C-Cave)) / sum((C-Cave).^2);
```

Application

MATLAB code, part 2 – control variate estimation:

```
N  = 1e5;
Y  = randn(1,N);              % Normal random variable
S  = S0*exp((r-sig^2/2)*T + sig*sqrt(T)*Y);
F  = exp(-r*T)*max(0,S-K);
C  = exp(-r*T)*S;
F2 = F - lam*(C-S0);

Fave   = sum(F)/N;
F2ave  = sum(F2)/N;
sd     = sqrt( sum((F -Fave ).^2)/(N*(N-1)) );
sd2    = sqrt( sum((F2-F2ave).^2)/(N*(N-1)) );
```

Application

Results:

```
>> cv

estimated price (without CV) = 17.624089 +/- 0.178
estimated price (with CV)    = 17.651112 +/- 0.045
exact price                  = 17.662954
```

Final words

- Monte Carlo estimation is very simple, and efficient when there are multiple underlying assets
- Need to generate correlated Normal random variables
- Central Limit Theorem (CLT) is very important in giving a confidence interval for the computed value
- The use of a control variate with known expected value can greatly reduce the number of MC samples required