9 Monte Carlo method for option pricing (2 lectures)

We have seen in (2.36) and (8.6) that

$$c(0, S(0)) = \tilde{\mathbb{E}}[e^{-\int_0^T r(s,\omega)ds} c(T, S(T))] = \tilde{\mathbb{E}}[e^{-\int_0^T r(s,\omega)ds} (S(T) - K)^+]. \tag{9.1}$$

This means that the price of a derivative security can be usefully represented as an expected value. Valuing derivatives thus reduces to computing expectations. The expectation can be approximated by sample averages.

In general, valuing a derivative security by Monte Carlo typically involves simulating paths of stochastic processes used to describe the evolution of underlying asset prices, interest rates, model parameters, and other factors relevant to the security in question.

9.1 What is a Monte Carlo method

Suppose we want to calculate

$$\mu = \mathbb{E}[h(X)] \tag{9.2}$$

with $h(x) = (\cos(50x) + \sin(20x))^2$ and $X \sim U(0, 1)$ being a random variable with uniform distribution on $[0, 1]$. We can write

$$\mu = \int_0^1 h(x)dx$$

and it is possible to integrate this function analytically for this special example. We will use (9.2) as an example to explain the main steps in Monte Carlo simulation: (1) We generate $X_1, X_2, \dots X_n$ independent identically distributed (i.i.d.) $U(0, 1)$ random variables. (2) We approximate $\mu = \mathbb{E}[h(X)]$ with $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^n h(X_i)$.

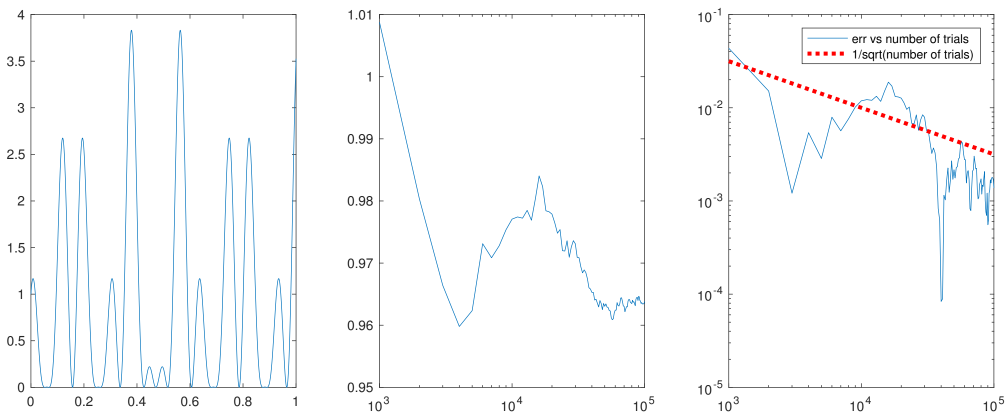


Figure 9.1: Calculation of $\mathbb{E}[h(X)]$ with $X \sim U(0, 1)$. Left: plot of $h(x) = (\cos(50x) + \sin(20x))^2$. Middle: $\hat{\mu}_n$ vs n. Right: $|\hat{\mu}_n - \mu|$ vs n.

```
% syms x; mu=int((cos(50*x)+sin(20*x)).^2,0,1)
% the following mu is the exact value of E[h(X)]
mu=1/200*sin(100)+103/105-1/70*cos(70)+1/30*cos(30)-1/80*sin(40);

% now, we pretend that we do not know the exact value of mu
% we try Monte Carlo method to estimate mu
x=linspace(0,1,1000);
h=(cos(50*x)+sin(20*x)).^2;
figure; subplot(1,3,1); plot(x,h);

rand('state',5);
N=100; L=1000;
n=zeros(N,1); muEst=zeros(N,1);
r=rand(N*L,1);
for k=1:N
    n(k)=L*k;  % number of samples
    x=r(1:n(k));
    h=(cos(50*x)+sin(20*x)).^2;
    muEst(k)=sum(h)/n(k);
end
err=abs(muEst-mu);

subplot(1,3,2); semilogx(n,muEst);

subplot(1,3,3); loglog(n,err); hold on;
loglog(n,1./sqrt(n),'r:','LineWidth',3);
legend('err vs number of trials','1/sqrt(number of trials)');
```

9.2 Error Estimates for Monte Carlo Method

In Question 1 of Homework IV, we have proved a form of weak law of large number, which states that if $X_1, ..., X_i, ...$ are i.i.d. random variables with finite mean μ, then

$$\frac{\sum_{i=1}^n (X_i - \mu)}{n} \to 0 \qquad \text{or} \qquad \bar{X} \stackrel{\text{def}}{=} \frac{\sum_{i=1}^n X_i}{n} \to \mu$$

in mean square sense.

The central limit theorem says that for mutually independent random variable $X_1, ..., X_i, ...$ with mean μ and variance σ^2,

$$\frac{\sum_{i=1}^n (X_i - \mu)}{\sqrt{n}} \to N(0, \sigma^2) \qquad \text{in distribution}^{73}. \tag{9.3}$$

[73]We say $Y_i \to Y$ in distribution if $\lim_{i\to\infty} F_i(y) = F(y)$ where $F_i(y) \stackrel{\text{def}}{=} \mathbb{P}(Y_i \leq y)$ is the cdf of Y_i and $F(y)$ is the cdf of Y.

(9.3) can also be written as

$$\frac{(\frac{1}{n}\sum_{i=1}^{n} X_i) - \mu}{\sqrt{\sigma^2/n}} \to N(0,1) \qquad \text{in distribution.} \tag{9.4}$$

Using the central limit theorem, we can determine the error related to Monte Carlo estimator. Suppose we are trying to use Monte Carlo to estimate

$$\theta = \mathbb{E}[X]. \tag{9.5}$$

Let X_1, X_2, ..., X_n be n simulations of X with mean μ and variance σ^2 . In addition, suppose X_i's are independent. By (9.4), $Y = \frac{\bar{X}-\theta}{\sqrt{\sigma^2/n}}$ is approximately $N(0,1)$ distributed. From a $N(0,1)$ table we notice that with probability 0.95 a normal sample lies within 1.96 standard deviations of the mean; hence

$$P\left(-1.96 < \frac{\bar{X}-\theta}{\sqrt{\sigma^2/n}} < 1.96\right) \approx 0.95.$$

In other words, with probability 0.95, θ lies in the interval

$$\bar{X} - 1.96\sqrt{\sigma^2/n} < \theta < \bar{X} + 1.96\sqrt{\sigma^2/n}. \tag{9.6}$$

So, given a value for σ^2, we can compute the probabilistic error bounds for θ, or confidence intervals as they are called.

Since we are only interested in $\theta = \mathbb{E}[X]$, we may choose different random variable X with the same mean. An efficient Monte Carlo method requires the variance $\sigma = \mathrm{Var}[X]$ to be small so that the size of the confidence interval in (9.6), which is about $4\sqrt{\sigma^2/n}$, is small.

In practice, usually σ^2 must itself be estimated from the data. The value of σ^2 can be estimated by the sample variance

$$s_X^2 \stackrel{\text{def}}{=} \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2. \tag{9.7}$$

In Question 5 of Homework IX, we prove $E(s_X^2) = \sigma^2 = \mathbb{E}(X_i - \mathbb{E}X_i)^2$.

There is a theorem (Slutsky's theorem) in probability, which says that if $X_n \to X$ in distribution and $Y_n \to$ constant a in distribution, then $X_nY_n \to aX$ in distribution. Hence (see Example 5.5.18 of "Statistical Inference" 2nd edition by Casella and Berger for the proof of the following statement)

$$\frac{(\frac{1}{n}\sum_{i=1}^{n} X_i) - \mu}{\sqrt{s_X^2/n}} = \frac{(\frac{1}{n}\sum_{i=1}^{n} X_i) - \mu}{\sqrt{\sigma^2/n}}\frac{\sigma}{s_X} \to N(0,1) \qquad \text{in distribution.} \tag{9.8}$$

So, when n is large, with probability close to 0.95, θ lies in the interval

$$\bar{X} - 1.96\sqrt{s_X^2/n} < \theta < \bar{X} + 1.96\sqrt{s_X^2/n}. \tag{9.9}$$

Example 9.1 *If you have taken probabilitiy or statistics before, you might have learned the following fact (for example, see Proposition 8.I in Chapter 7 of "A First Course in Probability" by Sheldon Ross. A proof can be also found in Question 5 of Homework IX): If* X_1*,* X_2*, …,* X_n *are i.i.d. normal random variables with mean* μ *and variance* σ^2*, then*

- $\bar{X}$ *is a normal random variable with mean* μ *and variance* σ^2/n*;*
- *the sample mean* $\bar{X}$ *and the sample variance* s_X^2 *are independent;*
- $(n-1)s_X^2/\sigma^2 \sim \chi^2_{n-1}$ *(whose pdf is some constant* $\times e^{-x/2}x^{\frac{n-3}{2}}$*,* $x > 0$*)* [74]*;*
- $\frac{\bar{X}-\mu}{\sqrt{s_X^2/n}} = \frac{\frac{\bar{X}-\mu}{\sqrt{\sigma^2/n}}}{\sqrt{\frac{(n-1)s_X^2}{\sigma^2}\frac{1}{n-1}}}$ *has t-distribution with* $n-1$ *degrees of freedom* [75] *whose pdf is some constant* $\times(1+\frac{t^2}{n-1})^{-\frac{n}{2}}$*. When* $n \to \infty$*,* $(1+\frac{t^2}{n-1})^{-\frac{n}{2}} = [(1+\frac{t^2}{n-1})^{n-1}]^{-\frac{n}{2(n-1)}} \to [e^{t^2}]^{-1/2} = e^{-t^2/2}$*. Hence* $\frac{\bar{X}-\mu}{\sqrt{s_X^2/n}} \to N(0,1)$ *in distribution.*

9.3 First Example of Risk-Neutral Pricing by Monte Carlo

The following is a overly simplified summary of risk-neutral pricing by Monte Carlo. (See Page 30 of "Monte Carlo Methods in Financial Engineering" by Glasserman.)

1) Replace drift α in $dS_t = S_t(\alpha dt + \sigma dW_t)$ by risk-free interest rate r and simulate path;
2) Calculate the payoff of derivative security on each path;
3) discount payoffs at the risk-free rate;
4) Calculate average over paths.

In this section, unless otherwise stated, the Matlab codes are taken from the textbook of Choe.

```
S0 = 100;
K = 110;
sigma = 0.3;
r = 0.05;
T = 0.5;
% Black-Scholes-Merton formula
d1 = (log(S0/K)+(r+0.5*sigma^2)*T)/(sigma*sqrt(T));
d2 = d1 - sigma*sqrt(T);
N1 = normcdf(d1);
N2 = normcdf(d2);
```

[74] If Z_i $i = 1, \cdots, n$ are i.i.d. $N(0,1)$ random variable, then $\sum_{i=1}^n Z_i^2 \sim \chi^2_n$

[75] If $Z \sim N(0,1)$ and $Y \sim \chi^2_n$ are independent, then $\frac{Z}{\sqrt{Y/n}}$ has t-distribution with n degrees of freedom.

```
Call_formula = S0*N1 - K*exp(-r*T)*N2;
% option price
J = 16;
L = 2^J;
rng(0);
W = sqrt(T)*randn(L,1);
S = S0*exp((r-0.5*sigma^2)*T + sigma*W); % S_T
V = exp(-r*T)*max(S - K,0);
figure(1)

for j = 9:J
    M(j) = 2^j; % number of samples
    a(j) = mean(V(1:M(j)));
    b(j) = 1.96*std(V(1:M(j)))/sqrt(M(j));
end
x = 8:0.01:J+1;
semilogx(x,Call_formula*ones(length(x),1),'r')
axis([min(x),max(x),4,6.5])
hold on
errorbar(9:J,a(9:J),b(9:J));
xlabel('log_2(N)')
ylabel('call price');
```

The result is shown in Figure 9.2.

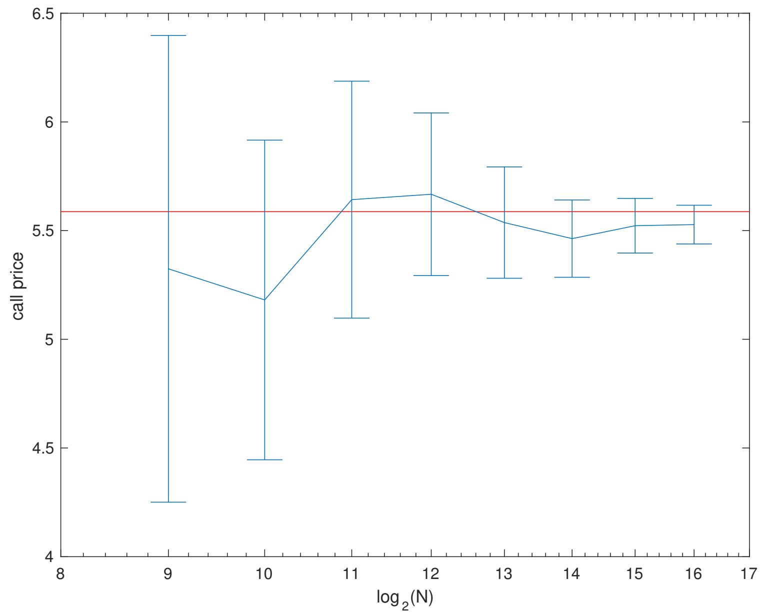


Figure 9.2: Monte Calo simulation for European call option.

9.4 Volatility and Maximum Likelihood Estimation

(We are following Chapter 25 of Choe's book.) Volatility is the most important parameter in the geometric Brownian motion stock price model. Historical volatility is an estimate of volatility based on the past data, usually collected over several years up to the current date. Assume that the asset price follows geometric Brownian motion $dS_t = S_t\,(\alpha dt + \sigma dW_t)$,

$$S_t = S_0 e^{(\alpha - \frac{1}{2}\sigma^2)t + \sigma W_t}.$$

Since

$$\log \frac{S_t}{S_u} = \left(\alpha - \frac{1}{2}\sigma^2\right)(t-u) + \sigma(W_t - W_u)$$

for $0 \le u \le t$, we observe that $\log \frac{S_t}{S_u} \sim N(\left(\alpha - \frac{1}{2}\sigma^2\right)(t-u), \sigma^2(t-u))$

Suppose that historical asset price data $S(t_i)$ is available at equally spaced time values $t_i = i\delta t$, $1 \le i \le n$, and that $t_n = n\delta t$ is the present time. Define

$$L_i = \log \frac{S(t_i)}{S(t_{i-1})}. \tag{9.10}$$

Then $L_i \sim N((\alpha - \frac{1}{2}\sigma^2)\delta t, \sigma^2 \delta t)$.

$$L_i/\sqrt{\delta t} \sim N\left((\alpha - \frac{1}{2}\sigma^2)\sqrt{\delta t}, \sigma^2\right). \tag{9.11}$$

Let $X_1, \cdots, X_n$ be a sample from a population with pdf or pmf (probability mass function) $f(x|\theta_1, \cdots, \theta_k)$ where $\theta_1, \cdots, \theta_k$ are the parameters in f. For example, $X_i = L_i/\sqrt{\delta t}$, $f(x|\theta_1, \theta_2)$ is the pdf of $N((\alpha - \frac{1}{2}\sigma^2)\sqrt{\delta t}, \sigma^2)$. Hence $\theta_1 = (\alpha - \frac{1}{2}\sigma^2)\sqrt{\delta t}$, $\theta_2 = \sigma^2$.

We want to find an estimation of the point $(\theta_1, \cdots, \theta_k)$ using a function of $X_1, \cdots, X_n$. This function is denoted by $W(X_1, \cdots, X_n)$ and is called a point estimator. (The rest of this subsection is taken from Section 7.2.2 of "Statistical Inference", 2nd edition, by Casella and Berger.)

The method of maximum likelihood is by far, the most popular technique for deriving estimators. The likelihood function is defined by

$$L(\vec{\theta}|\vec{x}) = L(\theta_1, \cdots, \theta_k | x_1, \cdots, x_n) = \prod_{i=1}^{n} f(x_i|\theta_1, \cdots, \theta_k). \tag{9.12}$$

Suppose $\vec{X} \stackrel{\text{def}}{=} (X_1, \cdots, X_n) = (x_1, \cdots, x_n) \stackrel{\text{def}}{=} \vec{x}$ is observed. Let $\hat{\theta}(\vec{x})$ be a parameter value at which $L(\vec{\theta}|\vec{x})$ attains its maximum as a function of $\vec{\theta}$, with $\vec{x}$ held fixed. A maximum likelihood estimator (MLE) of the parameter θ based on a sample $\vec{X}$ is $\hat{\theta}(\vec{X})$.

Note that $L(\vec{\theta}|\vec{X})$ attains its maximum at $\vec{\theta}$ if and only if $\log L(\vec{\theta}|\vec{X})$ attains its maximum at $\vec{\theta}$.

Example 9.2 *Let $X_1, \cdots, X_n$ be i.i.d. Bernoulli random variables. This means $\mathbb{P}(X_i = 1) = p$ and $\mathbb{P}(X_i = 0) = 1 - p$. Hence its pmf is $\mathbb{P}(X_i = x) = f(x|p) = p^x(1-p)^{1-x}$. So when $(X_1, \cdots, X_n) = (x_1, \cdots, x_n) \overset{def}{=} \vec{x}$ is observed,*

$$L(p|\vec{x}) = \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} = p^y(1-p)^{n-y}$$

where $y = \sum_{i=1}^n x_i$. $\log L(p, |\vec{x}) = y \log p + (n-y)\log(1-p)$. $\frac{1}{L(p|\vec{x})}\frac{\partial}{\partial p}L(p|\vec{x}) \overset{chain\ rule}{=} \frac{\partial}{\partial p}\log L(p|\vec{x}) = \frac{y}{p} - \frac{n-y}{1-p} = 0$ implies $p = \frac{y}{n}$. From calculus, we know the global maximum of $L(p|\vec{x})$ on $[0,1]$ must be attained at either the point where $\frac{\partial}{\partial p}L(p|\vec{x}) = 0$ or the boundary points $p = 0$ or 1. But $L(\frac{y}{n}|\vec{x}) > 0 = L(1|\vec{x}) = L(0|\vec{x})$. Hence $(\sum_{i=1}^n x_i)/n$ is the MLE of p.

Example 9.3 *Let $X_1, \cdots, X_n$ be i.i.d. $N(\mu, \sigma^2)$ with both μ and σ^2 unknown. Then*

$$\begin{aligned} L(\mu, \sigma^2|\vec{x}) &= \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{1}{2}\sum_{i=1}^n (x_i-\mu)^2/\sigma^2}, \\ \log L(\mu, \sigma^2|\vec{x}) &= -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma^2 - \frac{1}{2}\sum_{i=1}^n (x_i-\mu)^2/\sigma^2. \end{aligned}$$

Then

$$\begin{aligned} \frac{\partial}{\partial\mu}\log L(\mu, \sigma^2|\vec{x}) =& \frac{1}{\sigma^2}\sum_{i=1}^n (x_i - \mu), \\ \frac{\partial}{\partial\sigma^2}\log L(\mu, \sigma^2|\vec{x}) =& -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^n (x_i-\mu)^2. \end{aligned}$$

Setting these partial derivatives equal to 0 and solving yields the solution

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^n x_i, \tag{9.13}$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n (x_i - \hat{\mu})^2. \tag{9.14}$$

One can verify that this solution is indeed a global maximum. See Example 7.2.11 of Casella and Berger for the verification.

MLE has the following nice property

Theorem 9.1 *(Theorem 10.1.12 of Casella and Berger and Theorem 4.17 of J. Shao "Mathematical Statistics", 2nd edition) Let $X_1, X_2, \cdots, X_n, \cdots$ be i.i.d. $f(x|\vec{\theta})$. Let $\hat{\theta}$ denote the MLE of $\vec{\theta}$, and let $\tau(\vec{\theta})$ be a continuous function of $\vec{\theta}$. Under some reasonable conditions on $f(x|\vec{\theta})$,*

$$\sqrt{n}\left(\tau(\hat{\theta}) - \tau(\vec{\theta})\right) \to N(0, v(\vec{\theta})) \tag{9.15}$$

where $v(\vec{\theta})$ is some function of $\vec{\theta}$. See Theorem 10.1.12 of Casella and Berger and Theorem 4.17 of Shao for the precise formula of $v(\vec{\theta})$ when $\vec{\theta} \in \mathbb{R}$ and $\vec{\theta} \in \mathbb{R}^k$ respectively.

Continue with (9.11): $L_i/\sqrt{\delta t} \sim N\left((\alpha - \frac{1}{2}\sigma^2)\sqrt{\delta t}, \sigma^2\right)$. People add a correction to (9.14) and use

$$\frac{1}{m-1}\sum_{k=1}^{m}\left(\frac{L_{n-k+1}}{\sqrt{\delta t}} - a\right)^2 \tag{9.16}$$

to estimate σ^2. Here $a = \frac{1}{m}\sum_{k=1}^{m}\frac{L_{n-k+1}}{\sqrt{\delta t}}$ is an estimation of $(\alpha - \frac{1}{2}\sigma^2)\sqrt{\delta t}$. Please see Question 1 of the homework to see why we use $\frac{1}{m-1}$ instead of $\frac{1}{m}$ in (9.16). Note that if $(\mathbb{E}[X])^2$ is small comparing with $\mathrm{Var}[X]$, $\mathrm{Var}[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \approx \mathbb{E}[X^2]$. Since a is close to zero, in practice, people may ignore a and simply use

$$\sqrt{\frac{1}{m-1}\sum_{k=1}^{m}\left(\frac{L_{n-k+1}}{\sqrt{\delta t}}\right)^2} = \sqrt{\frac{1}{\delta t}\frac{1}{m-1}\sum_{k=1}^{m}L_{n-k+1}^2}.$$

to estimate σ. If we want to find volatility per year using the daily data, we choose $\delta t = 1/N$ where N is the number of trading days in a year, usually $N = 252$.

The following code perform the above estimation:
`vola = np.sqrt(np.sum(np.square(goog['Log_Ret']))/(len(goog['Log_Ret'])-1)*252)`
The following Python code is a slightly modified version of code from "Python for Finance: Analyze Big Financial Data" by Y. Hilpisch.

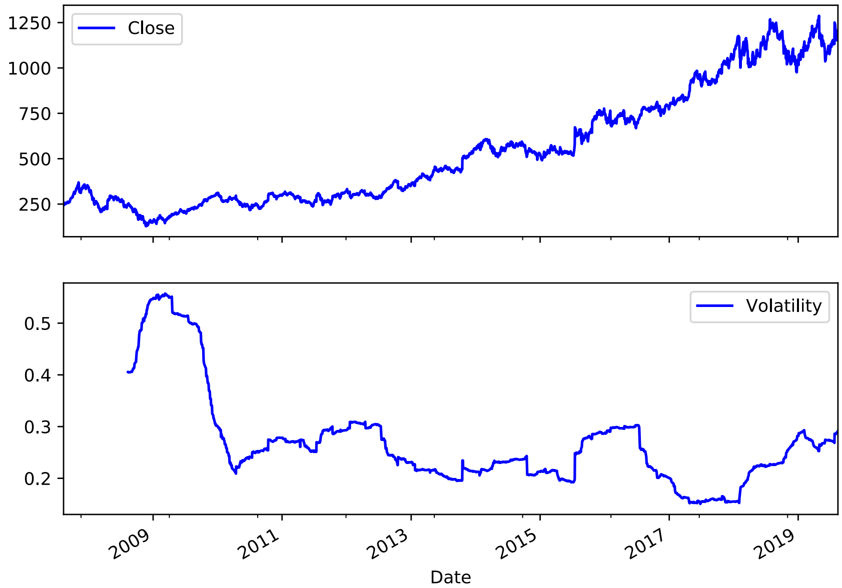
```
import numpy as np
import pandas as pd
import pandas_datareader.data as web
import matplotlib.pyplot as plt

goog = web.DataReader('GOOG', data_source='yahoo', start='8/14/2007', end='8/14/2019')
# type(goog) is pandas.core.frame.DataFrame

# print(goog.tail())
goog['Log_Ret'] = np.log(goog['Close'] / goog['Close'].shift(1))
vola = np.sqrt( np.sum(np.square(goog['Log_Ret']))/(len(goog['Log_Ret'])-1)*252 )
print('Volatility = '+str(vola))

goog['Volatility'] = pd.Series(goog['Log_Ret']).rolling\
    (window=252,center=False).std() * np.sqrt(252)
goog[['Close', 'Volatility']].plot(subplots=True, color='blue', figsize=(8, 6))
plt.show()
```

Download `Sect_9p4.py` from luminus and install Python 3. Then run `python3 Sect_9p4.py`. The above code gives `Volatility = 0.289367985503` and the following Figure. As you can tell that the volatility is not a constant.



9.5 The Antithetic Variate Method

This subsection is based on Chapter 28 of "Stochastic Analysis for Finance with Simulations" by Choe. Recall that to estimate $\theta = \mathbb{E}[X]$, we may choose different random variable X with the same mean. An efficient Monte Carlo method requires the variance $\sigma^2 = \mathrm{Var}[X]$ to be small so that the size of the confidence interval $\bar{X} - 1.96\sqrt{\sigma^2/n} < \theta < \bar{X} + 1.96\sqrt{\sigma^2/n}$ is small.

Here is a basic example of the antithetic variate method for reducing variance. Given a function $f(X)$ defined on the unit interval $[0, 1]$, choose a random variable U uniformly distributed in $[0, 1]$. Note that $1 - U$ is also uniformly distributed. To find the expectation $\mathbb{E}[f(U)]$ we compute

$$\frac{\mathbb{E}[f(U)] + \mathbb{E}[f(1-U)]}{2} = \mathbb{E}\left[\frac{f(U) + f(1-U)}{2}\right] \tag{9.17}$$

In this case, $1 - U$ is called an antithetic variate. This method is effective especially when f is monotone.

Example 9.4 *Suppose we want to find* $\mathbb{E}[e^U]$ *with* $U \sim U(0,1)$ *by the Monte Carlo method. (For this special case, we know that* $\mathbb{E}[e^U] = \int_0^1 e^x dx = e - 1$.*) Let*

$$X = \frac{e^U + e^{1-U}}{2}.$$

$\mathbb{E}[X] = \frac{1}{2}\left(\mathbb{E}[e^U] + \mathbb{E}[e^{1-U}]\right) = \mathbb{E}[e^U]$.

$$Var(X) = \frac{Var(e^U) + Var(e^{1-U}) + 2Cov(e^U, e^{1-U})}{4} = \frac{Var(e^U) + Cov(e^U, e^{1-U})}{2}.$$

Note that

$$\mathbb{E}[e^U e^{1-U}] = \int_0^1 e^x e^{1-x} dx = e \approx 2.71828$$

and

$$Cov(e^U, e^{1-U}) = \mathbb{E}[e^U e^{1-U}] - \mathbb{E}[e^U]\mathbb{E}[e^{1-U}] = e - (e-1)^2 \approx -0.2342.$$

Since

$$Var(e^U) = \mathbb{E}[e^{2U}] - \left(\mathbb{E}[e^U]\right)^2 = \frac{e^2 - 1}{2} - (e-1)^2 \approx 0.2420,$$

we have

$$Var(X) \approx 0.0039.$$

Hence the variance of X *is greatly reduced in comparison with the variance of* e^U.

Theorem 9.2 *For any monotonically increasing (or decreasing) functions* f *and* g*, and for any random variable* X*, we have*

$$\mathbb{E}[f(X)g(X)] \geq \mathbb{E}[f(X)]\mathbb{E}[g(X)], \tag{9.18}$$

or, equivalently,

$$Cov(f(X), g(X)) \geq 0. \tag{9.19}$$

Proof: First, consider the case when f and g are monotonically increasing. Note that

$$(f(x) - f(y))(g(x) - g(y)) \geq 0$$

for every x, y. Thus, for any random variables X, Y, we have

$$(f(X) - f(Y))(g(X) - g(Y)) \geq 0$$

and hence

$$\mathbb{E}[(f(X) - f(Y))(g(X) - g(Y))] \geq 0.$$

This leads to

$$\mathbb{E}[f(X)g(X)] + \mathbb{E}[f(Y)g(Y)] \geq \mathbb{E}[f(X)g(Y)] + \mathbb{E}[f(Y)g(X)]. \tag{9.20}$$

If X and Y are independent and identically distributed, then

$$\mathbb{E}[f(X)g(X)] = \mathbb{E}[f(Y)g(Y)]$$

and

$$\begin{aligned}\mathbb{E}[f(X)g(Y)] &= \mathbb{E}[f(X)]\mathbb{E}[g(Y)] = \mathbb{E}[f(X)]\mathbb{E}[g(X)]\\ \mathbb{E}[f(Y)g(X)] &= \mathbb{E}[f(Y)]\mathbb{E}[g(X)] = \mathbb{E}[f(X)]\mathbb{E}[g(X)]\end{aligned}$$

Hence (9.20) implies

$$2\mathbb{E}[f(X)g(X)] \geq 2\mathbb{E}[f(X)]\mathbb{E}[g(X)].$$

Therefore,

$$\mathrm{Cov}(f(X), g(X)) = \mathbb{E}[f(X)g(X)] - \mathbb{E}[f(X)]\mathbb{E}[g(X)] \geq 0.$$

When f and g are monotonically decreasing, then we simply replace f and g by $-f$ and $-g$. Then $\mathrm{Cov}(f(X), g(X)) = \mathrm{Cov}(-f(X), -g(X)) \geq 0$. □

Corollary 9.1 *Let $f(x)$ be a monotone function and X be any random variable. Then we have*

$$Cov(f(X), f(1-X)) \leq 0, \tag{9.21}$$

$$Cov(f(X), f(-X)) \leq 0. \tag{9.22}$$

Proof: First, consider the case when f is monotonically increasing. Define $g(x) = -f(1-x)$. Then $g(x)$ is also increasing. Hence

$$\mathrm{Cov}(f(X), f(1-X)) = \mathrm{Cov}(f(X), -g(X)) = -\mathrm{Cov}(f(X), g(X)) \leq 0.$$

If $f(x)$ is decreasing, then consider $-f(x)$ which is increasing. The proof of $\mathrm{Cov}(f(X), f(-X)) \leq 0$ is similar. □

Remark: (i) If $f(x)$ is monotone and U is uniformly distributed in $[0, 1]$, then $\mathrm{Cov}(f(U), f(1-U)) \leq 0$.

(ii) If $f(x)$ is monotone and Z is a standard normal variable, then $\mathrm{Cov}(f(Z), f(-Z)) \leq 0$.

Example 9.5 *Consider a European call option with expiry date T and payoff $C(S_T) = (S_T - K)^+$ where S_T is the asset price at T which is modelled by the Geometric Brownian motion. If Z denotes a standard normal variable, we consider Z and $-Z$ to apply the antithetic variate method. Since $S_T = S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z}$ and $\tilde{S}_T = S_0 e^{(r-\frac{1}{2}\sigma^2)T-\sigma\sqrt{T}Z}$ are correlated, two payoffs $C(S_T)$ and $\tilde{C}(S_T)$ are also correlated. Since*

$$\tilde{\mathbb{E}}[C(S_T)] = \tilde{\mathbb{E}}[\tilde{C}(S_T)],$$

we have

$$\tilde{\mathbb{E}}[C(S_T)] = \frac{1}{2}\tilde{\mathbb{E}}[C(S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z}) + C(S_0 e^{(r-\frac{1}{2}\sigma^2)T-\sigma\sqrt{T}Z})].$$

We use the antithetic variate method to compute the price of a European call option, and compare the results with the Black-Scholes-Merton formula. The price of the vanilla call option according to the Black-Scholes-Merton formula is approximately equal to `call_vanilla=10.0201`*, while the estimated values by the standard Monte Carlo method and by the antithetic variate method are* `a = 10.1005` *and* `a_anti = 10.0645`*, respectively. The corresponding variances are* `b=382.2483` *and* `b_anti =139.9497` *with their ratio given by* `2.7313`*.*

```
S0 = 100;
K = 110;
r = 0.05;
sigma = 0.3;
T = 1;
d1 = (log(S0/K) + (r + 0.5*sigma^2)*T)/(sigma*sqrt(T));
d2 = (log(S0/K) + (r - 0.5*sigma^2)*T)/(sigma*sqrt(T));
call_vanilla = S0*normcdf(d1) - K*exp(-r*T)*normcdf(d2)

N = 1000;
dt = T/N;
M = 10^5;
S = ones(1,N+1);
S2 = ones(1,N+1);
V = zeros(M,1);
V2 = zeros(M,1);
V_anti = zeros(M,1);
rng(1)
for i = 1:M
    S(1,1) = S0;
    S2(1,1) = S0;
    dW = sqrt(dt)*randn(1,N);
    for j = 1:N
        S(1,j+1) = S(1,j)*exp((r-0.5*sigma^2)*dt + sigma*dW(1,j));
        S2(1,j+1) = S2(1,j)*exp((r-0.5*sigma^2)*dt - sigma*dW(1,j));
    end
    V(i) = exp(-r*T)*max(S(1,N+1)-K,0);
    V2(i) = exp(-r*T)*max(S2(1,N+1)-K,0);
    V_anti(i) = (V(i)+V2(i))/2;
end
a = mean(V)
b = var(V)
a_anti = mean(V_anti)
b_anti = var(V_anti)
ratio = b/b_anti
```

Example 9.6 *We estimate the reduction rate of variance in the antithetic variate method for* *various choices of parameters. The option price is given by taking the average of* $e^{-rT}(C_1 + C_2)/2$ *with respect to the standard normal density.*

```
S0 = 10;
K = 8; %Choose other values for K.
r = 0.05;
sigma = 0.3;
T = 1;
Z = randn(10^7,1);
C1 = max(S0*exp((r-0.5*sigma^2)*T +sigma*sqrt(T)*Z)-K,0);
C2 = max(S0*exp((r-0.5*sigma^2)*T -sigma*sqrt(T)*Z)-K,0);
C = (C1+C2)/2;
Variance_classical = var(C1)
Variance_anti = var(C)
reduction_rate = 1- Variance_anti / Variance_classical
```

When `K=8`, `Variance_classical = 8.4800`, `Variance_anti =1.1732`, `reduction_rate =0.8616`.

When `K=10`, `Variance_classical = 5.5955`, `Variance_anti =1.6810`, `reduction_rate =0.6996`.

When `K=12`, `Variance_classical = 3.0198`, `Variance_anti =1.2475`, `reduction_rate =0.5869`.

Example 9.7 *A barrier option has a barrier or barriers, and becomes effective only when the asset price stays within the barriers or goes outside the boundary depending on the type of the option during the lifetime of the option. For a knock-out option, if the asset price crosses the barrier or barriers, the option loses its value immediately. For a knock-in option the barrier option has value only when the asset price reaches a certain level before or at expiry date.*

We use the antithetic variate method to compute the price of a down-and-out put barrier option with maturity T *and payoff*

$$(K - S_T)^+ \mathbf{1}_{\{\min_{0\le t\le T} S_t > L\}} \tag{9.23}$$

According to the formula in Theorem 18.3 of Choe's book, its price is `P_do = 10.6332`, *while the estimated values by the standard Monte Carlo method and by the antithetic variate method are* `a = 10.7168` *and* `a_anti = 10.6846`, *respectively. The corresponding variances are* `b = 175.5960` *and* `b_anti = 36.8322` *with their ratio given by* `4.7675`.

```
S0 = 100;
K = 110;
r = 0.05;
sigma = 0.3;
T = 1;
```

```
L = 60; % a lower barrier
d1K = (log(S0/K) + (r + 0.5*sigma^2)*T)/(sigma*sqrt(T));
d2K = (log(S0/K) + (r - 0.5*sigma^2)*T)/(sigma*sqrt(T));
d1L = (log(S0/L) + (r + 0.5*sigma^2)*T)/(sigma*sqrt(T));
d2L = (log(S0/L) + (r - 0.5*sigma^2)*T)/(sigma*sqrt(T));
d3 = (log(L/S0) + (r + 0.5*sigma^2)*T)/(sigma*sqrt(T));
d4 = (log(L/S0) + (r - 0.5*sigma^2)*T)/(sigma*sqrt(T));
d5 = (log(L^2/S0/K) + (r + 0.5*sigma^2)*T)/(sigma*sqrt(T));
d6 = (log(L^2/S0/K) + (r - 0.5*sigma^2)*T)/(sigma*sqrt(T));
put_vanilla = K*exp(-r*T)*normcdf(-d2K) - S0*normcdf(-d1K);
P2 = - K*exp(-r*T)*normcdf(-d2L) + S0*normcdf(-d1L) ;
P3 = - K*exp(-r*T)*(L/S0)^(2*r/sigma^2-1)*(normcdf(d4)-normcdf(d6));
P4 = S0*(L/S0)^(2*r/sigma^2+1)*(normcdf(d3)-normcdf(d5));
P_do = put_vanilla + P2 + P3 + P4
% the above exact price of a down-and-out put option is derived
% in Theorem 18.3 of Choe's book.

N = 10000;
dt = T/N;
M = 10^4;
V1 = zeros(M,1);
V2 = zeros(M,1);
V_anti = zeros(M,1);
S1 = ones(1,N+1);
S2 = ones(1,N+1);
rng(1)
for i=1:M
    S1(1,1) = S0;
    S2(1,1) = S0;
    dW = sqrt(dt)*randn(1,N);
    for j = 1:N
        S1(1,j+1) = S1(1,j)*exp((r-0.5*sigma^2)*dt + sigma*dW(1,j));
        S2(1,j+1) = S2(1,j)*exp((r-0.5*sigma^2)*dt - sigma*dW(1,j));
    end
    S1_min = min(S1(1,:));
    S2_min = min(S2(1,:));
    if S1_min > L
        V1(i) = exp(-r*T)*max(K - S1(1,N+1),0);
    else
        V1(i) = 0;
    end
```

```
    if S2_min > L
        V2(i) = exp(-r*T)*max(K - S2(1,N+1),0);
    else
        V2(i)=0;
    end
    V_anti(i) = (V1(i)+V2(i))/2;
end

a = mean(V1)
b = var(V1)
a_anti = mean(V_anti)
b_anti = var(V_anti)
ratio = b/b_anti
```

9.6 Euler-Maruyama for generating sample paths

Note that in the above example, we have to simulate the whole path of S_t. For the Geometric Brownian motion case, we know the exact formula of S_t. Hence we can set

$$S_{t_{n+1}} = S_{t_n} e^{(r-\frac{1}{2}\sigma^2)\delta t + \sigma\sqrt{\delta t} w}.$$

Here $\delta t = T/N$, $t_n = n\delta t$, $w \sim N(0,1)$. In the more general case, suppose X_t, which can represent stock price, interest rate, or volatility, satisfies

$$dX_t = f(t, X_t)dt + g(t, X_t)dW_t.$$

The following Euler-Maruyama method will generate a sample path: Suppose we have X_{t_n}, we first generate a random variable $w \sim N(0,1)$, then compute $X_{t_{n+1}}$ by

$$X_{t_{n+1}} = X_{t_n} + f(t_n, X_{t_n})\delta t + g(t_n, X_{t_n})\sqrt{\delta t} w.$$

For more details, see the standard textbook "Numerical Solution of Stochastic Differential Equations" by Kloeden.

9.7 The Control Variate Method

The control variate method for variance reduction in estimating $\mathbb{E}[X]$ employs another random variable Y whose properties are well-known. The random variable Y is called a control variate.

Suppose that $\mathbb{E}[Y]$ is already known. Define

$$\tilde{X} = X + c(Y - \mathbb{E}[Y])$$

for a real constant c. Then

$$\mathbb{E}[\tilde{X}] = \mathbb{E}[X].$$

We try to find an optimal value $\tilde{c}$ of c for which

$$\operatorname{Var}(\tilde{X}) = \operatorname{Var}(X) + c^2\operatorname{Var}(Y) + 2c\operatorname{Cov}(X, Y)$$

become minimum. Differentiating with respect to c, we have

$$\tilde{c} = -\frac{\operatorname{Cov}(X, Y)}{\operatorname{Var}(Y)}, \tag{9.24}$$

and by choosing $c = \tilde{c}$ we obtain

$$\begin{aligned}\min_c \operatorname{Var}(\tilde{X}) =& \operatorname{Var}(X) - \frac{\operatorname{Cov}(X, Y)^2}{\operatorname{Var}(Y)} = \operatorname{Var}(X)\left(1 - \frac{\operatorname{Cov}(X, Y)^2}{\operatorname{Var}(X)\operatorname{Var}(Y)}\right) \\ =& \operatorname{Var}(X)\left(1 - \rho(X, Y)^2\right) \le \operatorname{Var}(X).\end{aligned}$$

And the more X and Y are correlated, the better the variance reduction we get.

Example 9.8 *Let U be uniformly distributed in the interval $[0, 1]$ and*

$$X = e^U.$$

We want to estimate $\mathbb{E}[X]$. Take a control variate $Y = U$. Note that $\mathbb{E}[Y] = 1/2$. Define

$$\tilde{X} = e^U - \tilde{c}\left(U - \frac{1}{2}\right)$$

for the optimal value $\tilde{c}$ given by (9.24). Note that

$$\tilde{c} = -\frac{Cov(e^U, U)}{Var(U)} = -\frac{\mathbb{E}[e^U U] - \mathbb{E}[e^U]\mathbb{E}[U]}{1/12} = 6(e-3) \approx -1.690309.$$

Recall that

$$Var(e^U)] = \mathbb{E}[e^{2U}] - \mathbb{E}[e^U]^2 = \frac{e^2 - 1}{2} - (e-1)^2 \approx 0.242036.$$

Hence the variance reduction rate is approximately equal to

$$\rho(e^U, U)^2 = \frac{Cov(e^U, U)^2}{Var(e^U)\, Var(U)} \approx 0.9837.$$

and the variance is greatly reduced by 98.37%.

Example 9.9 *(European Call Option) To estimate $\mathbb{E}[X]$ for $X = (S_T - K)^+$, we choose the underlying asset itself as a control variate, i.e., we take $Y = S_T$. Recall that $\mathbb{E}[Y] = S_0 e^{rT}$ and $Var(Y) = S_0^2 e^{2rT}(e^{\sigma^2 T} - 1)$. To compute the covariance of X and Y we may use the Monte Carlo method since we do not need the precise value for $\tilde{c}$ in (9.24). For exotic options with complicated payoffs, sometimes it might be convenient to take the payoffs of standard European calls and puts as control variates.*

The payoff of an Asian option with expiry date T is determined by a suitably defined average of underlying asset prices $S_1, \cdots, S_n$ measured on predetermined dates $0 < t_1 < t_2 < \cdots < t_n = T$. For example, we can choose any of the following definitions of average to define a payoff of an Asian option

$$\begin{aligned} A_1 &= \text{arithmetic average } = \frac{S_1 + \cdots + S_n}{n} \\ A_2 &= \text{gemetric average } = (S_1 \times \cdots \times S_n)^{1/n} \end{aligned}$$

or even some weighted average. One can also define the corresponding continuous versions of averages.

Theorem 9.3 *(Theorem 18.1 of Choe's book) Suppose that the price of the underlying asset S_t follows a geometric Brownian motion, i.e., $S_t = S_0 \exp{(\mu - \frac{1}{2}\sigma^2)t + \sigma W_t}$ for some μ and σ. Then the price of an Asian call option with geometric average, with strike price K and expiry date T, is given by*

$$S_0 e^{(\bar{u}-r)T} N(d_1) - K e^{-rT} N(d_2)$$

where $d_1, d_2 = \frac{\log(S_0/K) + \left(\bar{\mu} \pm \frac{1}{2}\bar{\sigma}^2\right)T}{\bar{\sigma}\sqrt{T}}$, $\bar{\sigma}^2 = \frac{1}{6}\sigma^2 \left(1 + \frac{1}{n}\right)\left(2 + \frac{1}{n}\right)$, $\bar{\mu} = \frac{1}{2}\bar{\sigma}^2 + \frac{1}{2}\left(r - \frac{1}{2}\sigma^2\right)\left(1 + \frac{1}{n}\right)$.

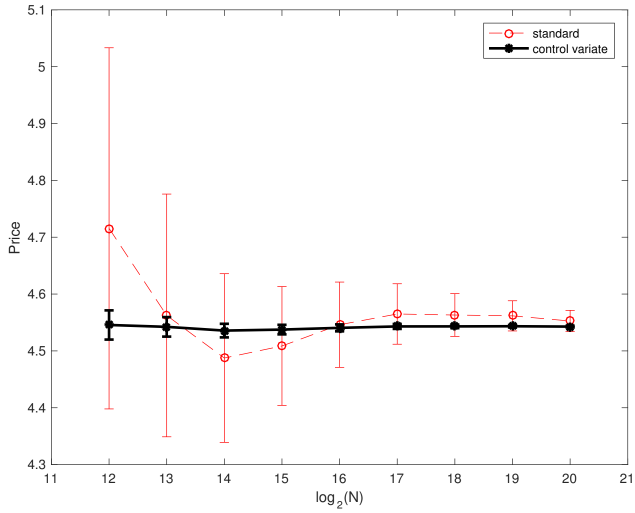
Example 9.10 *(Asian Option with Arithmetic Average) We use the control variate method to compute the price of an Asian option with arithmetic average, denoted by V_{arith}, using the price of the corresponding Asian option with geometric average, V_{geo}, as a control variate. The formula for the price of an Asian option with geometric average, V_{geo} formula, is given by the previous Theorem. Since the expectation of V_{geo} is equal to $V_{geoformula}$, the expectation of*

$$\tilde{V} = V_{arith} + c\,(V_{geo} - V_{geoformula})$$

is equal to the expectation of V_{arith}, however, with a suitable choice of c the variance of $\tilde{V}$ is reduced in comparison to the variance of V_{arith} which is computed by the standard Monte Carlo method. The optimal value for c is given by (9.24) which is not known. For our simulation we simply choose $\tilde{c} = 1$, which is not the best choice, however, we have a reasonable reduction in variance.

```
S0 = 100;
K = 110;
r = 0.05;
sigma = 0.3;
T = 1;
L = 12; % number of observations
dt = T/L;
sigma_bar = sqrt( sigma^2*(L+1)*(2*L+1)/(6*L^2));
mu_bar = 1/2*sigma_bar^2 + (r-1/2*sigma^2)*(L+1) / (2*L);
d1 = (log(S0/K) + (mu_bar+1/2*sigma_bar^2)*T)/(sigma_bar*sqrt(T));
```

```
d2 = (log(S0/K) + (mu_bar-1/2*sigma_bar^2)*T)/(sigma_bar*sqrt(T));
V_geo_formula = S0*exp((mu_bar -r)*T)*normcdf(d1) -K*exp(-r*T)*normcdf(d2);
J = 20;
ave = zeros(J,1);
ave_control = zeros(J,1);
error = zeros(J,1);
error_control = zeros(J,1);
ratio = ones(J,1);
S = ones(2^J,L);
rng(1);
dW = sqrt(dt)*randn(2^J,L);
for i=1:2^J
    S(i,1) = S0*exp((r-1/2*sigma^2)*dt +sigma*dW(i,1)); %asset price at T_1
    for j=2:L
        S(i,j) = S(i,j-1) *exp((r-1/2*sigma^2)*dt+ sigma*dW(i,j));
    end
end
J1 = 12;
for n=J1:J
    N = 2^n;
    V_arith = exp(-r*T) * max( mean(S(1:N,:),2) - K , 0);
    ave(n) = mean(V_arith);
    var_V_arith = var(V_arith);
    error(n) = 1.96*sqrt(var_V_arith)/sqrt(N);
    V_geo = exp(-r*T) * max( exp(mean(log(S(1:N,:)),2)) - K , 0);
    V = V_arith - V_geo + V_geo_formula;
    ave_control(n) = mean(V);
    var_control = var(V);
    error_control(n) = 1.96*sqrt(var_control)/sqrt(N);
    ratio(n) = var_V_arith/var_control;
end
errorbar(J1:J, ave(J1:J), error(J1:J), 'ro--')
hold on
errorbar(J1:J, ave_control(J1:J), error_control(J1:J),'k*-','linewidth',2)
legend('standard', 'control variate');
xlabel('log_2(N)');
ylabel('Price');
```



9.8 The Importance Sampling Method

Importance sampling is often referred to as a change of measure technique. To be more concrete, consider the problem of estimating the expected value

$$\mu = \mathbb{E}h(X)$$

with X has pdf $f(x)$. The plain Monte Carlo scheme will simulate i.i.d. samples X_i with pdf f and take the sample average of $\{h(X_i)\}$ as the estimate.

The basic idea of importance sampling comes from the observation that for an arbitrary probability density function g, one can write

$$\begin{aligned} \mu =& \mathbb{E}h(X) = \int_{\mathbb{R}} h(x)f(x)dx = \int_{\mathbb{R}} h(x)\frac{f(x)}{g(x)} \cdot g(x)dx \\ =& \int_{\mathbb{R}} h(y)\frac{f(y)}{g(y)} \cdot g(y)dy = \mathbb{E}\left(h(Y)\frac{f(Y)}{g(Y)}\right), \end{aligned} \tag{9.25}$$

where Y is a random variable with pdf g. Therefore, one can draw i.i.d. samples Y_i from the alternative pdf g and use the sample average of

$$\frac{1}{\#\text{ of samples}} \sum_i h(Y_i)\frac{f(Y_i)}{g(Y_i)}$$

to estimate μ. Importance sampling is different from plain Monte Carlo in that samples are now generated from a different probability distribution (hence the name "change of measure").

$$\frac{f(Y_i)}{g(Y_i)}.$$

is called likelihood ratio in some book. The variance of the importance sampling estimate is

$$\mathrm{Var}\left[h(Y)\frac{f(Y)}{g(Y)}\right] \tag{9.26}$$

where Y is a random variable with pdf g. To reduce the variance, the pdf g should be chosen to "mimic" $h(x)f(x)$. g should be large when hf is large and this gives the name "importance sampling".

Example 9.11 *Let* $h(x) = 4\sqrt{1-x^2}$. $\mu = \mathbb{E}_X[h(X)]$ *with* $X \sim U(0,1)$ *which means* $f = 1$ *on* $[0,1]$. *If we use simple sampling,* $Var[h(X)] = \mathbb{E}[h(X)^2] - (\mathbb{E}h(X))^2 = \int_0^1 16(1-x^2)dx - \pi^2 = 0.797$. *Let* $g(y) = \frac{4-2y}{3}$ *which imitates the relative heights of the graph* $h(y)f(y)$ *moderately well.* $Var[h(Y)\frac{f(Y)}{g(Y)}] = \mathbb{E}\left(\frac{h(Y)}{g(Y)}\right)^2 - \left(\mathbb{E}\frac{h(Y)}{g(Y)}\right)^2 = \int_0^1 \frac{h(y)^2}{g(y)}dy - \pi^2 \approx 0.224$.

9.9 Importance Sampling for Normal Distributions and Girsanov Theorem Revisit

Consider the problem of estimating $\mathbb{E}[h(X)]$ where $X \sim N(0,\sigma^2)$. (We use f to denote the pdf of X.) The alternative sampling distribution is chosen $Y = X + \theta \sim N(\theta, \sigma^2)$ for some $\theta \in \mathbb{R}$. Denote the pdf of Y by $g(y)$. The corresponding likelihood ratio is

$$\frac{f(Y)}{g(Y)} = \frac{e^{-\frac{Y^2}{2\sigma^2}}}{e^{-\frac{(Y-\theta)^2}{2\sigma^2}}} = e^{-\frac{\theta}{\sigma^2}Y + \frac{1}{2}\frac{\theta^2}{\sigma^2}}.$$

Example 9.12 *(Rare event simulation) Suppose we want to estimate* $\mu = \mathbb{P}(X \in B)$ *with* $X \sim N(0,1)$ *and* $B = (b,\infty)$ *with* b *very large. Let* $h(X) = \mathbf{1}_{\{X\in B\}}$ *which is* 1 *if* $X \in B$ *and* 0 *otherwise. Then*

$$\mathbb{E}[h(X)] = \mathbb{E}[\mathbf{1}_{\{X\in B\}}] = \int_{\mathbb{R}} \mathbf{1}_{\{x\in B\}} f(x)dx = \int_B f(x)dx = \mathbb{P}(X \in B) = \mu.$$

If we directly generate N *standard normal distributed sample* X_i*'s and estimate* μ *by* $\frac{1}{N}\sum_{i=1}^N \mathbf{1}_{\{X_i\in B\}}$, *most of the* X_i*'s will end up with* $\mathbf{1}_{\{X_i\in B\}} = 0$ *and are wasted in the sense that they are not counted. We have to take* N *very large to get a good estimation of* μ. *Now, if we use*

$$\mathbb{E}[h(X)] = \mathbb{E}\left[h(Y)\frac{f(Y)}{g(Y)}\right]$$

with $Y \sim N(\theta, 1)$. *Then* $\frac{f(Y)}{g(Y)} = e^{-\theta Y + \frac{1}{2}\theta^2}$. *So, we generate* N *i.i.d. sample* Y_i *of type* $N(\theta,1)$ *and estimate* μ *by*

$$\mu \approx \frac{1}{N}\sum_{i=1}^N \mathbf{1}_{\{Y_i\in B\}} e^{-\theta Y_i + \frac{1}{2}\theta^2}.$$

If $\theta \geq b$, *then most of the* Y_i*'s won't be wasted.*

```
b = 5;
N = 10000;
rng(1);
X = randn([N,1]);
Z1 = X>b;
mu1 = sum(Z1)/N
var1 = var(Z1)

theta = 6;
Y = X + theta;
Z2 = (Y>b).*exp(-theta*Y+0.5*theta^2);
mu2 = sum(Z2)/N
var2 = var(Z2)

% confidence interval
c = 2*sqrt(var2/N);
[mu2-c, mu2+c]
```

The first method gives $\mu \approx 0$ hence is totally useless. The second method gives $\mu \approx 2.7285 \times 10^{-7}$ and the variance is 6.9506×10^{-13}. By (9.6), with probability 0.954, μ lies in the interval $[2.562, 2.895] \times 10^{-7}$. The exact value is indeed 2.867×10^{-7}.

Example 9.13 (Deep Out-of-the-Money Asian Option) *Consider an option Let*

$$C_T(S_{t_1}, \cdots, S_{t_L}) = \max\left\{\frac{1}{L}(S_{t_1} + \cdots + S_{t_L}) - K, 0\right\}$$

be the payoff of an Asian option with arithmetic average. Here $L = T/\delta t$ and $t_i = n\delta t$. The standard Monte Carlo method for option pricing computes

$$e^{-rT}\mathbb{E}_r[C_T(S_{t_1}, \cdots, S_{t_L})]$$

using the sample paths of the geometric Brownian motion $dS_t = rS_t dt + \sigma S_t dW_t$. Since $S_{t_i} = S_0 e^{(r-\frac{1}{2}\sigma^2)t_i + \sigma W_{t_i}}$, if we let $X = (W_{t_1}, W_{t_2}, \cdots, W_{t_L}) = (X_1, \cdots, X_L)$, then

$$\mathbb{E}[C_T(S_{t_1}, \cdots, S_{t_L})] = \mathbb{E}[h(X)],$$

with

$$h(X) = \max\left\{\frac{S_0}{L}\left(e^{(r-\frac{1}{2}\sigma^2)t_1 + \sigma X_1} + \cdots + e^{(r-\frac{1}{2}\sigma^2)t_L + \sigma X_L}\right) - K,\ 0\right\}.$$

In computing the price of such an option using the standard Monte Carlo method, too many sample asset paths are wasted without contributing in option valuation, if K is very high compared to S_0 so that the option is deep out-of-the-money.

We move upward the asset paths as time increases by replacing a given drift coefficient r by a larger value r_1 in simulating geometric Brownian motion so that more sample paths *thus generated would have nonzero contribution in evaluating the average of payoff. That means we introduce* $Y_i = X_i + \frac{(r_1-r)t_i}{\sigma} = W_{t_i} + \frac{(r_1-r)t_i}{\sigma}$ *and* $Y = (Y_1, \cdots, Y_L)$*. Then by (9.25),*

$$\mathbb{E}[C_T(S_{t_1}, \cdots, S_{t_L})] = \mathbb{E}[h(X)] = \mathbb{E}\left[\frac{h(Y)f(Y)}{g(Y)}\right] \tag{9.27}$$

with

$$\begin{aligned} h(Y) =& \max\left\{\frac{S_0}{L}(e^{(r-\frac{1}{2}\sigma^2)t_1+\sigma Y_1} + \cdots + e^{(r-\frac{1}{2}\sigma^2)t_L+\sigma Y_L}) - K, 0\right\} \\ =& \max\left\{\frac{S_0}{L}(e^{(r_1-\frac{1}{2}\sigma^2)t_1+\sigma W_1} + \cdots + e^{(r_1-\frac{1}{2}\sigma^2)t_L+\sigma W_{t_L}}) - K, 0\right\}. \end{aligned}$$

which is no longer deep out-of-the-money if r_1 *is large.*

Now, we need to figure out what $f(Y)$ *is and what* $g(Y)$ *is. We stress again that* f *is the pdf of* X *while* g *is the pdf of* Y*.*

Note that if W *is Brownian motion, the pdf of* $X = (W_{\delta t}, W_{2\delta t}, W_{3\delta t}, \cdots, W_{L\delta t})$ *is (see Theorem 3.2)*

$$f(x) = f(x_1, x_2, \ldots, x_L) = ce^{-\sum_{i=0}^{L-1}\frac{(x_{i+1}-x_i)^2}{2\delta t}}.$$

Since $Y = X + \frac{r_1-r}{\sigma}(t_1, t_2, \cdots, t_L)$*, the pdf of* Y *is*

$$\begin{aligned} g(y) =& f\left(y_1 - \frac{(r_1-r)t_1}{\sigma}, \cdots, y_L - \frac{(r_1-r)t_L}{\sigma}\right) = ce^{-\sum_{i=0}^{L-1}\frac{(y_{i+1}-y_i-\frac{(r_1-r)\delta t}{\sigma})^2}{2\delta t}} \\ =& ce^{-\sum_{i=0}^{L-1}\frac{(y_{i+1}-y_i)^2}{2\delta t}} e^{\sum_{i=0}^{L-1}(y_{i+1}-y_i)\frac{r_1-r}{\sigma}} e^{-\sum_{i=0}^{L-1}\frac{(r_1-r)^2}{2\sigma^2}\delta t}. \end{aligned} \tag{9.28}$$

Let $\theta = \frac{r_1-r}{\sigma}$ *(which, by the way, is the minus of the* `theta` *in the Matlab code of Choe to be presented later). Then* $Y_i = X_i + \theta t_i = W_{t_i} + \theta t_i$ *and*

$$\begin{aligned} \frac{f(Y)}{g(Y)} =& e^{-\sum_{i=0}^{L-1}(Y_{i+1}-Y_i)\frac{r_1-r}{\sigma}} e^{\sum_{i=0}^{L-1}\frac{1}{2}\theta^2\delta t} \\ =& e^{-\sum_{i=0}^{L-1}(W_{t_{i+1}}-W_{t_i})\theta} e^{-\sum_{i=0}^{L-1}\frac{1}{2}\theta^2\delta t}. \end{aligned}$$

So, we now know how to use (9.27) to compute. By the way, if we let $\delta t \to 0$, $\frac{f(Y)}{g(Y)} \to e^{-\int_0^T \theta dW_t - \int_0^T \frac{1}{2}\theta^2 dt}$*.*

Another derivation using Girsanov Theorem*:*

Now, we present another approach that uses Girsanov Theorem. Then you can see that our derivation above hence derived Girsanov theorem from scratch.

We want to estimate

$$\begin{aligned} &\mathbb{E}[C_T(S_{t_1}, \cdots, S_{t_L})] \\ =&\mathbb{E}\left[\max\left\{\frac{S_0}{L}\left(e^{(r-\frac{1}{2}\sigma^2)t_1+\sigma W_{t_1}} + \cdots + e^{(r-\frac{1}{2}\sigma^2)t_L+\sigma W_{t_L}}\right) - K,\ 0\right\}\right] \\ =&\mathbb{E}\left[\max\left\{\frac{S_0}{L}\left(e^{(r_1-\frac{1}{2}\sigma^2)t_1+\sigma \tilde{W}_{t_1}} + \cdots + e^{(r_1-\frac{1}{2}\sigma^2)t_L+\sigma \tilde{W}_{t_L}}\right) - K,\ 0\right\}\right] \end{aligned}$$

where $r_1 t + \sigma\tilde{W}_t = rt + \sigma W_t$, or $\tilde{W}_t = \frac{(r-r_1)t}{\sigma} + W_t = -\theta t + W_t$ with the θ introduced before.

By Girsanov Theorem, if we define $Z = e^{\int_0^T \theta dW_t - \int_0^T \frac{1}{2}\theta^2 dt}$ and introduce $\tilde{\mathbb{P}}$ by (8.24)

$$\tilde{\mathbb{P}}(A) = \int_A Z(\omega) d\mathbb{P}(\omega),$$

then $\tilde{W}_t$ is a Brownian motion under the new measure $\tilde{\mathbb{P}}$. Now, $\tilde{\mathbb{E}}[X] = \int_\Omega X(\omega)d\tilde{\mathbb{P}}(\omega) = \int_\Omega X(\omega)Zd\mathbb{P}(\omega) = \mathbb{E}[XZ]$, or $\mathbb{E}[Y] = \tilde{\mathbb{E}}[YZ^{-1}]$ for any random variable Y. So,

$$\begin{aligned}&\mathbb{E}[C_T(S_{t_1}, \cdots, S_{t_L})]\\ =&\tilde{\mathbb{E}}\left[\max\left\{\frac{S_0}{L}\left(e^{(r_1-\frac{1}{2}\sigma^2)t_1+\sigma\tilde{W}_{t_1}} + \cdots + e^{(r_1-\frac{1}{2}\sigma^2)t_L+\sigma\tilde{W}_{t_L}}\right) - K,\ 0\right\} Z^{-1}\right].\end{aligned}$$

Note that $Z^{-1} = e^{-\int_0^T \theta dW_t + \int_0^T \frac{1}{2}\theta^2 dt} = e^{-\int_0^T \theta d\tilde{W}_t - \int_0^T \frac{1}{2}\theta^2 dt}$ and $\tilde{W}_t$ is the standard Brownian motion under $\tilde{\mathbb{E}}$. Hence we get exactly the same result as the previous approach which directly uses (9.25).

Here is the Matlab code of Choe for computing Asian option. I did not modify the code at all. We set $\theta = \frac{r_1 - r}{\sigma}$ while Choe's `theta` *is the minus of our θ. Hence the factor $f(Y)/g(Y)$ or Z^{-1} is exactly the* `RN` *in Choe's code.*

```
S0 = 100;
K = 150;
r = 0.05;
sigma = 0.3;
T = 1;
L = 12; % number of measurements
dt = T/L;
r1 = r + 0.5;
theta = (r - r1)/sigma;
M = 10^6;
rng(1);
dW = sqrt(dt)*randn(M,L);
W = sum(dW,2);
RN = exp(-0.5*theta^2*T + theta*W); % Radon-Nikodym derivative dQ/dQ1
S = zeros(M,L);
for i=1:M
    S(i,1) = S0*exp((r-1/2 *sigma^2)*dt + sigma*dW(i,1));
    for j=2:L
        S(i,j) = S(i,j-1) * exp((r-1/2*sigma^2)*dt+ sigma*dW(i,j));
    end
end
S1 = zeros(M,L);
```

```
for i=1:M
    S1(i,1) = S0*exp((r1-1/2 *sigma^2)*dt + sigma*dW(i,1));
    for j=2:L
        S1(i,j) = S1(i,j-1) * exp((r1-1/2*sigma^2)*dt+ sigma*dW(i,j));
    end
end
V = exp(-r*T) * max( mean(S(1:M,:),2) - K, 0);
price = mean(V)
variance = var(V)
V1 = exp(-r*T) * max( mean(S1(1:M,:),2) - K, 0);
price1 = mean(V1.*RN)
variance1 = var(V1.*RN)
ratio = variance / variance1
```

The results are `price = 0.1978, variance = 4.1783, price1 = 0.1952, variance1 = 0.2326, ratio = 17.9624`.

9.10 American Put Option

Recall that the American put gives the holder of the option the right, but not the obligation, to sell the underlying security for a fixed strike price K at any time before a given expiration time K. The decision on whether to exercise the American put or not may depend on all the information available at that time but may not depend on future information. This motivates the following definition

Definition 9.1 *(Defintion 8.2.1 of Shreve II) A stopping time τ is a random variable that takes value in $[0, \infty]$ and satisfies $\{\tau \leq t\} \in \mathcal{F}_t$ for all $t \geq 0$.*

That means, conditional on the information $\mathcal{F}_t$ at time t, one can determine whether the event $\{\tau \leq t\}$ has occured or not. For example, given $a > 0$, $\tau = \inf\{t \geq 0 | e^{(r-\frac{1}{2}\sigma^2)t+\sigma\tilde{W}_t} \geq a\}$ is a stopping time while $\tau = \sup\{t \in [0, 10] | e^{(r-\frac{1}{2}\sigma^2)t+\sigma\tilde{W}_t} \geq a\}$ is not a stopping time.

Note that $\{\tau > t - \frac{1}{n}\} = \{\tau \leq t - \frac{1}{n}\}^c \in F_{t-\frac{1}{n}} \subset \mathcal{F}_t$ by property *ii*) of Defintion 2.5 and Defintion 2.6. Hence $\{\tau = t\} = \{\tau \leq t\} \cap \left(\cap_{n=1}^{\infty}\{\tau > t - \frac{1}{n}\}\right) \in \mathcal{F}_t$ by property *iv*) of Defintion 2.5. This means given the information at time t, one can determine whether the event $\{\tau = t\}$ has occured or not.

τ can represent some exercise policy of an option. For example, for perpetual American put (means expiration date $T = \infty$), if the initial stock price is above $L^* \stackrel{\text{def}}{=} \frac{2r}{2r+\sigma^2}K$, one can define $\tau = \min\{t \geq 0 : S(t) = L^*\}$. So the first time the stock price drops to L^* is the time to exercise. Then at any time t, one can determine whether to exercise the perpetual American put or not. This example is from (8.3.9) and (8.3.12) of Shreve II.

We now give three characterization of American option pricing problem.

Part I The value of an American put option at $t = 0$ can be formulated as an optimal stopping problem

$$V(t,x) = \sup_{\tau \in T_{[t,T]}} \tilde{\mathbb{E}}\left[e^{-r(\tau-t)}(K - S_\tau)^+ | S_t = x\right]. \tag{9.29}$$

where $T_{[t,T]}$ denotes all stopping time within $[t,T]$. See (8.4.1) of Shreve II [76] or Section 8G of Duffie or Page 340 of Choe's book. Here $\tilde{\mathbb{E}}$ is calculated using risk-neutral probability $\tilde{\mathbb{P}}$. For Monte Carlo simulation based on continuous stock price model, it simply means that we will use the risk-free interest rate r as the drift coefficient.

Part II: Comparing with the European option pricing problem, we now have an additional constraint $V(t,x) \geq (K-x)^+$. This allows one to formulate American option pricing problem as a linear complementarity problem:

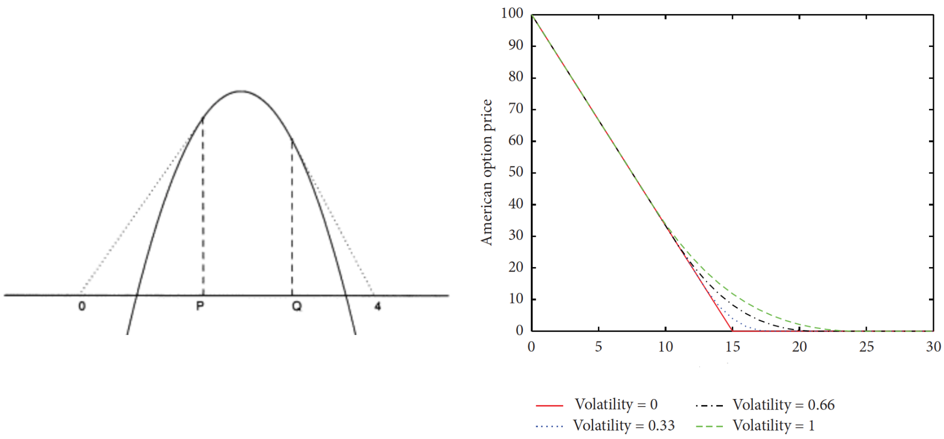


Figure 9.3: Left: An elastic string stretched over a parabola shaped obstacle is a typical linear complementarity problem. Right: A typical plot for $V(0,x)$ with different σ. (See also Figure 19.2 of Choe.) Note that $V(0,x)$ is bounded from below by $(K-x)^+$. When initial S_0 (which is the x-value in the plot) is small enough, one needs to exercise the option immediately to collect payoff $K - S_0$. Otherwise, the put is more valuable than its intrinsic value, and the owner can capture this extra value by waiting until the stock price enters $\mathcal{S}$ (defined in the next figure) to exercise. If the option is not exercised when it should be, its value tends downward.

$V(t,x)$ in (9.29) satisfies the following three conditions (Section 8.4.1 of Shreve II, Section 8H of Duffie.)

(a) $V(t,x) \geq (K-x)^+$ for all $t \in [0,T]$, $x \geq 0$.

(b) $rV(t,x) - \frac{\partial V}{\partial t}(t,x) - rx\frac{\partial V}{\partial x}(t,x) - \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V}{\partial x^2}(t,x) \geq 0$ for all $t \in [0,T]$, $x \geq 0$.

[76] In Shreve II, there is no $^+$ in $(K - S_\tau)^+$ because in (8.4.1) τ is allowed to be ∞ when the exercise criterion is never met. When $\tau = \infty$, $e^{-r(\tau-t)}(K - S_\tau) = 0$ since $r > 0$. So, Shreve's $T_{[t,T]}$ includes an additional value ∞.

(c) For each $t \in [0,T]$ and $x \geq 0$, equality holds in either (a) or (b).

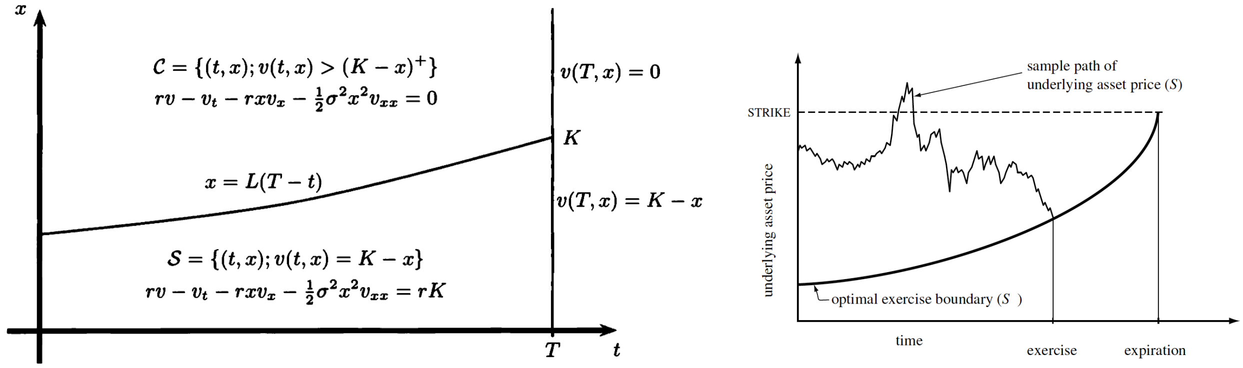


Figure 9.4: The set $\{(t,x) : 0 \leq t \leq T, x \geq 0\}$ can be divided into two regions, the stopping set $\mathcal{S} = \{(t,x) : V(t,x) = (K-x)^+\}$ and the continuation set $\mathcal{C} = \{(t,x) : V(t,x) > (K-x)^+\}$. The graph of the function $x = L(T-t)$ forms the boundary between $\mathcal{C}$ and $\mathcal{S}$. $T-t$ is the time to expiration. $L(0) = K$. If the option never expires, $\lim_{T\to\infty} L(T-t) = \frac{2r}{2r+\sigma^2}K$ (See Shreve II (8.3.12)).

Here is the argument by Duffie why we should have Property (b) above. This argument together with the following Remark are for your information only. They won't be tested. But I hope you can see why we see Black-Scholes-Merton again in Property (b). By Itô formula, when $dS_t = rS_t dt + \sigma S_t d\tilde{W}_t$,

$$
\begin{aligned}
e^{-rt}V(t,S_t) =& V(0,S_0) \\
&+ e^{-rt}\left(\int_0^t \left[-rV + \frac{\partial V}{\partial t} + rS_u\frac{\partial V}{\partial x} + \frac{1}{2}\sigma^2 S_u^2 \frac{\partial^2 V}{\partial x^2}\right] du + \int_0^t \sigma S_u \frac{\partial V}{\partial x} d\tilde{W}_u\right).
\end{aligned}
$$

By (9.29), one can manage to show that $V(0,S_0) \geq \tilde{\mathbb{E}}[e^{-rt}V(t,S_t)|\mathcal{F}_0]$. So, the integrand in $\int_0^t \bullet du$ is ≤ 0 which leads to Property (b).

Remark: We want to prove $V(0,S_0) \geq \tilde{\mathbb{E}}[e^{-rt}V(t,S_t)|\mathcal{F}_0]$ which is claimed to be obvious in Section 8H of Duffie. Here is argument if you do not have the intuition: By definition, $\sup_v f(v)$ denotes the smallest upper bound for all $f(v)$. Since it is the smallest upper bound, if one lowers it a little bit, there must be an $f(v^*)$ that is larger than that value. Hence for any $\epsilon > 0$, there must be a rational exercise policy $\tau^* \in T_{[t,T]}$ so that

$$
\tilde{\mathbb{E}}[e^{-r\tau^*}(K - S_{\tau^*})^+|\mathcal{F}_t] \geq \sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[e^{-r\tau}(K-S_\tau)^+|\mathcal{F}_t\right] - \epsilon.\text{[77]} \qquad (9.30)
$$

[77]Indeed, $\sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[e^{-r\tau}(K-S_\tau)^+|\mathcal{F}_t\right]$ can be attained at some τ^*: By (8.4.1) and (8.4.17) of Shreve II, we can choose $\tau^* = \min\{u \in [t,T] : (u,S(u)) \in \mathcal{S} \text{ in Figure 9.4}\} = \begin{cases} \min\{u \in [t,T] : S(u) = L(T-u)\} & \text{if } S(t) > L(T-t) \\ t & \text{if } S(t) \leq L(T-t) \end{cases}$ so that $\tilde{\mathbb{E}}[e^{-r\tau^*}(K - S_{\tau^*})^+|\mathcal{F}_t] = \sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[e^{-r\tau}(K-S_\tau)^+|\mathcal{F}_t\right]$.

Then,

$$\sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[\tilde{\mathbb{E}}[e^{-r\tau}(K-S_\tau)^+|\mathcal{F}_t]\Big|\,\mathcal{F}_0\right] \geq \tilde{\mathbb{E}}\left[\tilde{\mathbb{E}}[e^{-r\tau^*}(K-S_{\tau^*})^+|\mathcal{F}_t]\Big|\,\mathcal{F}_0\right]$$
$$\stackrel{(9.30)}{=} \tilde{\mathbb{E}}\left[\sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[e^{-r\tau}(K-S_\tau)^+|\mathcal{F}_t\right]\Bigg|\,\mathcal{F}_0\right]-\epsilon. \quad (9.31)$$

So,

$$\begin{aligned}
V(0,S_0) &= \sup_{\tau\in T_{[0,T]}} \tilde{\mathbb{E}}\left[e^{-r\tau}(K-S_\tau)^+\big|\,\mathcal{F}_0\right] \\
&\stackrel{\text{take sup over a smaller set}}{\geq} \sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[e^{-r\tau}(K-S_\tau)^+\big|\,\mathcal{F}_0\right] \\
&\stackrel{\text{iterated conditioning}}{=} \sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[\tilde{\mathbb{E}}[e^{-r\tau}(K-S_\tau)^+|\mathcal{F}_t]\Big|\,\mathcal{F}_0\right] \\
&\stackrel{(9.31)}{\geq} \tilde{\mathbb{E}}\left[\sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[e^{-r\tau}(K-S_\tau)^+|\mathcal{F}_t\right]\Bigg|\,\mathcal{F}_0\right]-\epsilon \\
&= \tilde{\mathbb{E}}\left[e^{-rt}\sup_{\tau\in T_{[t,T]}} \tilde{\mathbb{E}}\left[e^{-r(\tau-t)}(K-S_\tau)^+|\mathcal{F}_t\right]\Bigg|\,\mathcal{F}_0\right]-\epsilon \qquad (9.32) \\
&\stackrel{(9.29)}{=} \tilde{\mathbb{E}}[e^{-rt}V(t,S_t)\big|\,\mathcal{F}_0]-\epsilon. \qquad (9.33)
\end{aligned}$$

i.e., $V(0,S_0)\geq \tilde{\mathbb{E}}[e^{-rt}V(t,S_t)|\mathcal{F}_0]-\epsilon$. Since this ϵ is arbitrary, we can let $\epsilon\downarrow 0$ and obtain $V(0,S_0)\geq \tilde{\mathbb{E}}[e^{-rt}V(t,S_t)|\mathcal{F}_0]$. This ends the remark.

Part III. From conditions (a),(b),(c) in Part II, one can also characterize American option pricing problem as a free-boundary problem: $V(t,x)$ in (9.29) satisfies (Section 8.4.1 of Shreve II, Section 8H of Duffie.)

$$\begin{aligned}
& rV(t,x)-\frac{\partial V}{\partial t}(t,x)-rx\frac{\partial V}{\partial x}(t,x)-\frac{1}{2}\sigma^2x^2\frac{\partial^2 V}{\partial x^2}(t,x)=0, \quad \text{for } x\geq L(T-t), \\
& V(t,x)=K-x \quad \text{for } 0\leq x\leq L(T-t) \\
& \frac{\partial V}{\partial x}(t,x)=-1 \quad \text{for } x=L(T-t) \\
& L(0)=K, \qquad V(T,x)=(K-x)^+ \\
& \lim_{x\to\infty} V(t,x)=0.
\end{aligned}$$

For example, the fact that V and its $x-$derivative are continuous across the exercise boundary $L(T-t)$ with $\frac{\partial}{\partial x}V(t,x)=-1=\frac{\partial}{\partial x}(K-x)$ can be easily seen from the right plot of Figure 9.3.

9.11 Binomial Tree Method for American Put Option

To value American options, the optimal stopping problem (9.29) also has to be discretized:

$$V_0 = \sup_{\tau \in \{0, \delta t, 2\delta t, \cdots, T\}} \tilde{\mathbb{E}}[e^{-r\tau}(K - S_\tau)^+] \tag{9.34}$$

Indeed, we are approximate an American option with Bermudan options that can be exercised only at a set of discrete times $\{0, \delta t, 2\delta t, \cdots, n\delta t = T\}$.

Before we discuss the Monte Carlo method, let us first introduce the binomial tree method for American put options with expiration date $\mathtt{T}$: At every step we compare the price of the corresponding European option if the option is not exercised and the profit if it is exercised, and choose the more profitable action. Using our previous notation in Chapter 2, at the expiration date, $n\delta t = \mathtt{T}$,

$$ⓟ_n(\omega_1\omega_2\cdots\omega_n) = (K - S_n(\omega_1\omega_2\cdots\omega_n))^+.$$

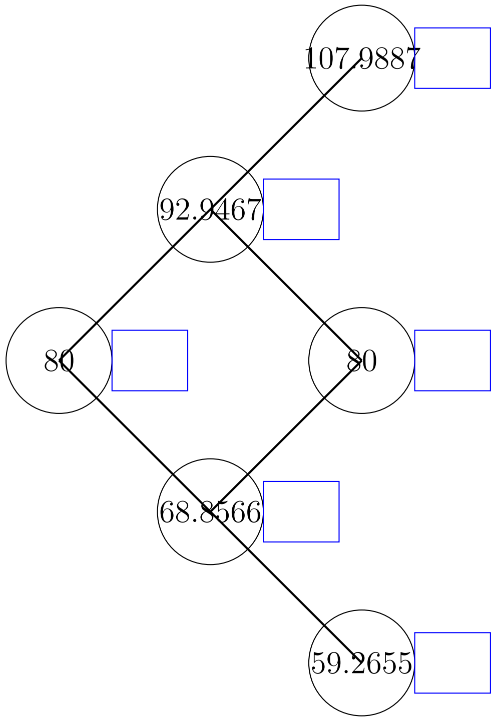
Then we move backward in time to determine the American put option price before the expiration date. For $n\delta t < \mathtt{T}$,

$$\begin{aligned} ⓟ_n(\omega_1\omega_2\cdots\omega_n) = \max\Big\{&(K - S_n(\omega_1\omega_2\cdots\omega_n))^+, \\ &\frac{1}{\rho}\left(q_u ⓟ_{n+1}(\omega_1\omega_2\cdots\omega_n H) + q_d ⓟ_{n+1}(\omega_1\omega_2\cdots\omega_n T)\right)\Big\}. \end{aligned}$$

The above formula should be compare with the following formula (which is (2.7)) for European put:

$$ⓟ_n(\omega_1\omega_2\cdots\omega_n) = \frac{1}{\rho}\left(q_u ⓟ_{n+1}(\omega_1\omega_2\cdots\omega_n H) + q_d ⓟ_{n+1}(\omega_1\omega_2\cdots\omega_n T)\right).$$

Example 9.14 *Consider the problem of using binomial tree method to calculate the American put option with $S_0 = 80$, $K = 90$, $\mathtt{T} = 0.5$, $r = 0.05$, $\sigma = 0.3$. We set $\delta t = \mathtt{T}/2$ and construct a two step binomial tree with $u = e^{\sigma\sqrt{\delta t}} \approx 1.16183$, $d = e^{-\sigma\sqrt{\delta t}} \approx 0.860708$, $\rho = e^{r\delta t} \approx 1.01258$. Evaluate the American put option price based on the binomial tree. Keep at least 5 significant digits in your calculation.*



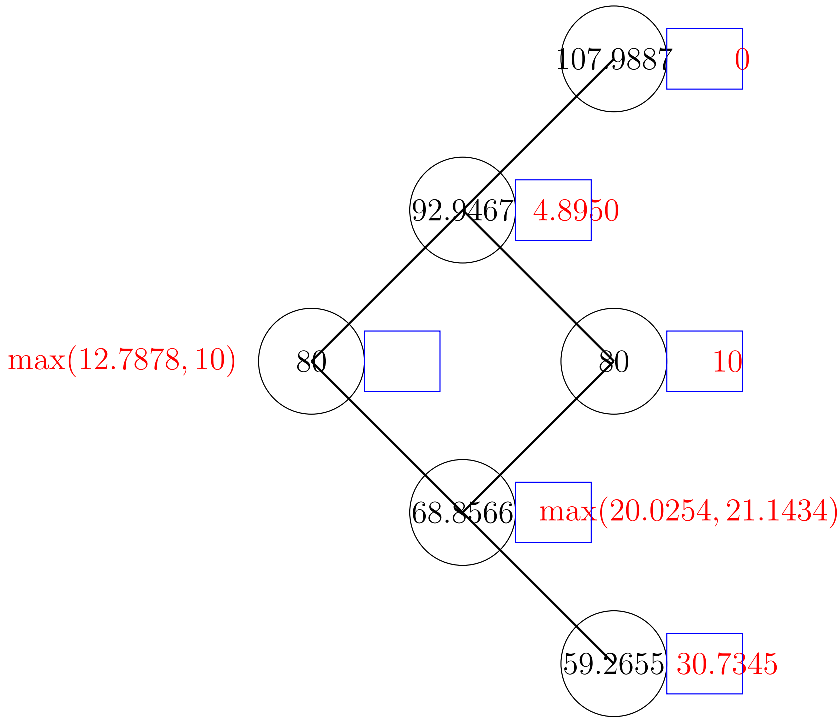
Solution

$q_u = \frac{\rho - d}{u - d} \approx 0.504342$, $q_d = 1 - q_u$.

Put_Am = 12.7878

Matlab code for another example:

```
% code from Choe.
% binomial tree method for American and European put options
S0 = 100;
K = 110;
T = 1;
r = 0.05;
sigma = 0.3;
M = 1000; % number of time steps
dt = T/M;
u = exp(sigma*sqrt(dt));
d = exp(-sigma*sqrt(dt));
q = (exp(r*dt)-d)/(u-d);
Put_Am = max(K - S0*u.^([M:-1:0]').*d.^([0:1:M]'),0);
Put_Eu = max(K - S0*u.^([M:-1:0]').*d.^([0:1:M]'),0);
for i = M:-1:1
    S = S0*u.^([i-1:-1:0]').*d.^([0:1:i-1]');
    Put_Am =max(max(K-S,0),exp(-r*dt)*(q*Put_Am(1:i)+(1-q)*Put_Am(2:i+1)));
```




```
    Put_Eu =exp(-r*dt)*(q*Put_Eu(1:i)+(1-q)*Put_Eu(2:i+1));
end
Put_Am
Put_Eu

d1 = (log(S0/K)+(r+0.5*sigma^2)*T)/(sigma*sqrt(T));
d2 = d1 - sigma*sqrt(T);
Put_Eu_ex = K*exp(-r*T)*normcdf(-d2) - S0*normcdf(-d1)
```

The result is `Put_Am = 15.6167`, `Put_Eu = 14.6526`, and `Put_Eu_ex = 14.6553`

9.12 Monte Carlo Method for American Put Option

The most influential paper on Monte Carlo method for American option is perhaps the paper "Valuing American Options by Simulation: A Simple Least-Squares Approach" by Longstaff and Schwartz. It starts with an illustrative example. The authors go through the derivation step by step carefully. You will read the first few pages of it in order to answer Question 10 of Homework IX.

In the paper of Longstaff and Schwartz, they first find the optimal exercise time τ^i for each stock price path S^i, $i = 1, 2, ..., m = 8$ (see the second table on page 119 of their paper) and then use

$$V_0 = \frac{1}{m}\sum_{i=1}^{m} e^{-r\tau^i}(K - S^i_{\tau^i})^+ \tag{9.35}$$

to estimate V_0. Hence the Monte Carlo method is based on (9.29).

Here is how to determine τ^i for each path i: Given a stock price path S^i, if at time t_k, the price is out-of-the-money (which means $K - S_{t_k} < 0$), then τ^i will not be t_k. If the stock price is in-the-money (which means $K - S_{t_k} > 0$) at time t_k, then one needs to decide if $\tau^i = t_k$ or not. In that situation, one compares the current pay-off $K - S_{t_k}$ with the value of continuation which we now introduce.

We use $q_t(x)$ to denote the value of continuation which is the value of not exercising the option at time t (when $S_t = x$) and then wait for another δt:

$$q_t(x) = \tilde{\mathbb{E}}[e^{-r\delta t} V_{t+\delta t}(S_{t+\delta t})|S_t = x]. \tag{9.36}$$

It is good to know that (See Remark 1 of the paper "A Review on Regression-based Monte Carlo Methods for Pricing American Options" by Michael Kohler or the first four equations of the paper "Simulation for American Options: Regression Now or Regression Later?" by Paul Glasserman and Bin Yu. This is the so called dynamic programming.)

$$V_t(x) = \max\left((K - x)^+, q_t(x)\right). \tag{9.37}$$

In principel, since we know $V_T(z) = \sup_{\tau\in[T,T]} \tilde{\mathbb{E}}\left[e^{-r(\tau-T)}(K - S_\tau)^+|S_T = z\right] = (K - z)^+$ for any z, we can evaluate $q_{T-\delta t}(x)$ by (9.36) and then $V_{T-\delta t}(x)$ by (9.37) for any x. So, by induction, we can travel backward in time and eventually find V_0.

Back to Longstaff and Schwartz. Their idea is to assume , say, $q_{t_k}(x) = a_0 + a_1 x + a_2 x^2$ (which is the $E[Y|X] = a_0 + a_1 X + a_2 X^2$ in their paper), and then use least square to estimate a_0, a_1, a_2. After that, they can evaluate $q_{t_k}(S^i_{t_k})$ and compare it with $(K - S^i_{t_k})$ which we now know is already positive. In some sense, they assume $q_{t_k}(x) = a_0 + a_1 x + a_2 x^2$ only when $K - x > 0$. When $K - x < 0$, they do not care how $q_t(x)$ looks like since τ^i won't be t_k and they do not need to know the value of continuation. That is why they use only in-the-money paths when estimating $q_{t_k}(x)$.

Here is the Matlab code by Mark Hoyle that implement Longstaff and Schwartz's algorithm.

```
function [Price,CF,S,t] = AmericanOptLSM(S0,K,r,T,sigma,N,M,type)

%AmericanOptLSM - Price an american option via Longstaff-Schwartz Method
%
%  By Mark Hoyle
%  https://www.mathworks.com/matlabcentral/fileexchange/
%  16476-pricing-american-options?focused=6781443&tab=function
%
% Inputs:
%
%   S0       Initial asset price
```

```
%   K       Strike Price
%   r       Interest rate
%   T       Time to maturity of option
%   sigma   Volatility of underlying asset
%   N       Number of points in time grid to use (minimum is 3, default is 50)
%   M       Number of points in asset price grid to use (minimum is 3, default is 50)
%   type    True (default) for a put, false for a call

if nargin < 6 || isempty(N), N = 50; elseif N < 3, error('N has to be at least 3'); end
if nargin < 7 || isempty(M), M = 50; elseif M < 3, error('M has to be at least 3'); end
if nargin < 8, type = true; end
dt = T/N;

t = 0:dt:T;
t = repmat(t',1,M);

rng(1);
R = exp((r-sigma^2/2)*dt+sigma*sqrt(dt)*randn(N,M));
S = cumprod([S0*ones(1,M); R]);

ExTime = (M+1)*ones(N,1);

% Now for the algorithm
CF = zeros(size(S)); % Cash flow matrix

CF(end,:) = max(K-S(end,:),0); % Option only pays off if it is in the money

for ii = size(S)-1:-1:2
    if type
        Idx = find(S(ii,:) < K); % Find paths that are in the money at time ii
    else
        Idx = find(S(ii,:) > K); % Find paths that are in the money at time ii
    end
    X = S(ii,Idx)'; X1 = X/S0;
    Y = CF(ii+1,Idx)'*exp(-r*dt); % Discounted cashflow
    R = [ ones(size(X1)) (1-X1) 1/2*(2-4*X1-X1.^2)];
    a = R\Y; % Linear regression step
    C = R*a; % Cash flows as predicted by the model
    if type
        Jdx = max(K-X,0) > C; % Immediate exercise better than predicted cashflow
    else
        Jdx = max(X-K,0) > C; % Immediate exercise better than predicted cashflow
```

```
    end
    nIdx = setdiff((1:M),Idx(Jdx));
    CF(ii,Idx(Jdx)) = max(K-X(Jdx),0);
    ExTime(Idx(Jdx)) = ii;
    CF(ii,nIdx) = exp(-r*dt)*CF(ii+1,nIdx);
end

Price = mean(CF(2,:))*exp(-r*dt);
end
```

Final Remark: You can read or try Question 15 of Homework III to get a little bit feeling about jump diffusion model for option price. After that, if you are interested in it, you can read, for example, the paper "Jump-diffusion models: a practitioner's guide" by Peter Tankov and Ekaterina Voltchkova. Google it and you should be able to find the paper on Tankov's webpage.

Python code by Y. Hilpisch based on Merton's jump diffusion model proposed in 1976 can be downloaded from `https://github.com/yhilpisch/dawp`

9.13 Homework IX

(Only submit solutions to Questions 1,8,9,10.)

1. (Random number generator) A linear congruential (pseudo-)random number generator generate a sequence of integers by $x_{n+1} = f(x_n)$ with

$$f(x) = (ax + c) \mod m$$

where x_n is an integer and $a > 1$, $c \geq 0$, $m >> 1$ are fixed integers. Once x_i is obtained, $u_i = x_i/m$ will be output and be used as a random number between 0 and 1. We use the notation $m|x$ to mean that m divides x (exactly, with no remainder) and we write $a = b \mod m$, if $m|(a-b)$. One says that a is congruent to b modulo m. For example, $21 = 5 \mod 8$.

Obviously, the iteration $x_{n+1} = f(x_n)$ eventually cycles: there is a positive integer p such that $x_i = x_{i+p}$ for all sufficiently large i. The smallest such p is called the cycle length or period of the random number generator. Ideally we want p as large as possible. But obviously, it cannot bigger than m.

When $a = 7^5 = 16807$, $c = 0$, and $m = 2^{31} - 1$ (which is a prime number), we obtain the random number generator used in Matlab version 4 in the 1990s. The cycle length is m which is around 2×10^9, perhaps sufficient for the 20th century, but not sufficient nowadays.

Now, let $a = 5$, $c = 1$, and $m = 8$ and let the generator be seeded with $x = 0$. Compute the sequence generated.

2. Suppose we want to generate a continuous random variable whose cumulative distribution function is the desired function F. One can first generate a uniformly distributed random variable $U \sim U(0,1)$ and then let $X = F^{-1}(U)$. Here $F^{-1}(u)$ denotes the inverse function of F. Show that the cdf of X is F.

Proof: Since F is monotonically non-decreasing,

$$\mathbb{P}(X \leq x) = \mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x).$$

3. (Box-Muller algorithm for generating normally distributed random variables) (This is for your information only and won't be tested.) Box-Muller algorithm generates two independent standard normal random variables. The idea is to first uniformly choosing an angle in $\mathbb{R}^2$ and then generating the square distance from an exponential distribution with parameter $\frac{1}{2}$ [78]. Hence the algorithm is as follows

[78]The pdf is $f(x) = \frac{1}{2}e^{-\frac{1}{2}x}$, $x \geq 0$. Hence its cdf is $F(x) = \int_0^x f(u)du = 1 - e^{-\frac{1}{2}x}$. Setting $y = 1 - e^{-\frac{1}{2}x}$, we get $x = -2\ln(1-y)$. Hence by Question 2, if $U_2 \sim U(0,1)$, $F^{-1}(U_2) = -2\ln(1-U_2)$ is exponentially distributed with parameter $1/2$. But since U_2 and $1-U_2$ have the same distribution, we directly use $-2\ln U_2$ to be the square distance $X_1^2 + X_2^2$.

1) Sample independent random variables $U_1,\ U_2 \sim U(0,1)$.

2) Let $\theta = 2\pi U_1$ and $r = \sqrt{-2\ln U_2}$.

3) $X_1 = r\cos(\theta),\ X_2 = r\sin(\theta)$.

Prove that X_1 and X_2 are independent identical distribution and $X_i \sim N(0,1)$.

Proof: Denote the joint pdf of X_1 and X_2 by $f_{(X_1,X_2)}$. All we need to show is $f_{(X_1,X_2)}(x_1,x_2) = \frac{1}{\sqrt{2\pi}}e^{-x_1^2/2}\frac{1}{\sqrt{2\pi}}e^{-x_2^2/2}$. In another word, we want to prove

$$\begin{aligned} P\left((X_1,X_2)\in A\right) &= \int_A f_{(X_1,X_2)}(x_1,x_2)dx_1dx_2 \\ &= \int_A \frac{1}{\sqrt{2\pi}}e^{-x_1^2/2}\frac{1}{\sqrt{2\pi}}e^{-x_2^2/2}dx_1dx_2 \end{aligned} \tag{9.38}$$

for any region $A \subset \mathbb{R}^2$.

Consider the mapping $(u_1,u_2) \to (x_1,x_2)$ defined by $x_1 = \cos(2\pi u_1)\sqrt{-2\ln u_2}$ and $x_1 = \sin(2\pi u_1)\sqrt{-2\ln u_2}$. Suppose the mapping $(u_1,u_2) \to (x_1,x_2)$ maps region B to region A.

If we want to change the integration $\int_A \cdots dx_1dx_2$ on the right hand side of the above equation to $\int_B \cdots du_1du_2$, we need to add the Jacobian $\det\left(\frac{\partial x_i}{\partial u_j}\right) \overset{\text{def}}{=} \det\begin{bmatrix} \frac{\partial x_1}{\partial u_1} & \frac{\partial x_1}{\partial u_2} \\ \frac{\partial x_2}{\partial u_1} & \frac{\partial x_2}{\partial u_2} \end{bmatrix}$.

Straightforward computation shows $\det\left(\frac{\partial x_i}{\partial u_j}\right) = \frac{2\pi}{u_2}$. Note that $\frac{1}{\sqrt{2\pi}}e^{-x_1^2/2}\frac{1}{\sqrt{2\pi}}e^{-x_2^2/2}$ becomes $\frac{1}{2\pi}u_2$. Hence

$$\begin{aligned} \int_A \frac{1}{\sqrt{2\pi}}e^{-x_1^2/2}\frac{1}{\sqrt{2\pi}}e^{-x_2^2/2}dx_1dx_2 &= \int_B \frac{1}{2\pi}u_2\frac{2\pi}{u_2}du_1du_2 = \int_B du_1du_2 \\ &= \mathbb{P}\left((U_1,U_2)\in B\right). \end{aligned}$$

Hence (9.38) follows from $\mathbb{P}\left((X_1,X_2)\in A\right) = \mathbb{P}\left((U_1,U_2)\in B\right)$.

4. Recall the Central Limit Theorem: Let $X_1, X_2, \ldots,$ be independent identically distributed random variables having mean μ and finite variance σ^2. Let $S_n = X_1 + \cdots + X_n$. Then

$$\lim_{n\to\infty} P\left(a \le \frac{S_n - n\mu}{\sigma\sqrt{n}} \le b\right) = P(a \le X \le b)$$

with $X \sim N(0,1)$.

Let S be the number of heads in 1,000,000 tosses of a fair coin. Use the Central Limit Theorem to estimate (i) the probability that S lies between 499,500 and 500,500; (ii) the probability that S lies between 499,000 and 501,000; and (iii) the probability that S lies between 498,500 and 501,500.

Solution:

X has Bernoulli distribution with parameter $\frac{1}{2}$, i.e., $P(X=1)=\frac{1}{2}$ and $P(X=0)=\frac{1}{2}$. $X=1$ means head and $X=0$ means tail. $\mathbb{E}X=\frac{1}{2}$. $\mathrm{Var}X=\sigma^2=\frac{1}{4}$. $S=\sum_{i=1}^{1{,}000{,}000} X_i$. $\mathrm{Var}S=\sum_{i=1}^{1{,}000{,}000}\mathrm{Var}X=250,000$

By the central limit theorem, $P(|\frac{S-500000}{\frac{1}{2}\times 1000}|\le a)\approx P(|X|\le a)$.

$a=1,2,3$ in the three different cases.

$P(|X|\le a)\approx 0.6826,\ 0.9544,\ 0.9972$ respectively.

5. Suppose X_i $(i=1,...,n)$ are i.i.d. random variables with mean μ and variance σ^2. Let $\bar{X}=\frac{1}{n}\sum_{i=1}^n X_i$

$$s_X^2=\frac{1}{n-1}\sum_{i=1}^n\left(X_i-\bar{X}\right)^2.$$

a) Show that $\bar{X}$ and $X_i-\bar{X}$ are uncorrelated, i.e.,

$$\mathrm{Cov}(\bar{X},X_i-\bar{X})=\mathbb{E}[(\bar{X}-\mu)(X_i-\bar{X})]=0. \tag{9.39}$$

By the way, **if** $X_i\sim N(\mu,\sigma)$, then $\bar{X}$ and $X_i-\bar{X}$ are independent. That essentially the reason [79] why $\bar{X}$ and $s_X^2=\frac{1}{n-1}\sum_{i=1}^n(X_i-\bar{X})^2$ are independent in Example 9.1.

b) Show that

$$(n-1)s_X^2=\left[\sum_{i=1}^n(X_i-\mu)^2\right]-n(\bar{X}-\mu)^2. \tag{9.40}$$

Sometimes, people will rewrite the above equation as

$$\frac{(n-1)s_X^2}{\sigma^2}+\left(\frac{\bar{X}-\mu}{\sigma/\sqrt{n}}\right)^2=\sum_{i=1}^n\left(\frac{X_i-\mu}{\sigma}\right)^2. \tag{9.41}$$

So, **if** $X_i\sim N(\mu,\sigma)$, $\frac{X_i-\mu}{\sigma}\sim N(0,1)$ and the right hand side of (9.41) $\sim\chi_n^2$. $\left(\frac{\bar{X}-\mu}{\sigma/\sqrt{n}}\right)^2\sim N(0,1)$ and is independent of $\frac{(n-1)s_X^2}{\sigma^2}$. Hence $\frac{(n-1)s_X^2}{\sigma^2}\sim\chi_{n-1}^2$

c) Show that s_X^2 is an unbiased estimation of the variance σ^2, namely,

$$E(s_X^2)=\sigma^2. \tag{9.42}$$

Proof: (a)

$$\mathrm{Cov}(\bar{X},X_i-\bar{X})=\mathrm{Cov}(\bar{X},X_i)-\mathrm{Cov}(\bar{X},\bar{X})=\frac{1}{n}\sum_{j=1}^n\mathrm{Cov}(X_j,X_i)-\mathrm{Var}[\bar{X}]=\frac{1}{n}\sigma^2-\frac{1}{n}\sigma^2=0.$$

[79] Because $\sum_{i=1}^n(X_i-\bar{X})=0$, one can write s_X^2 using only $\{X_i-\bar{X},i=2,..,n\}$. Because $\bar{X}$ and the $n-1$ dimensional random vector $(X_2-\bar{X},X_3-\bar{X},\cdots,X_n-\bar{X})$ are jointly normal and are uncorrelated, they are independent. So $\bar{X}$ and s_X^2 are independent.

(b) Note that $\sum_{i=1}^n (X_i - \mu)(\bar{X} - \mu) = n(\bar{X} - \mu)^2$.

$$\begin{aligned}
(n-1)s_X^2 &= \sum_{i=1}^n \left(X_i - \bar{X}\right)^2 = \sum_{i=1}^n \left((X_i - \mu) - (\bar{X} - \mu)\right)^2 \\
&= \sum_{i=1}^n \left[(X_i - \mu)^2 - 2(X_i - \mu)(\bar{X} - \mu) + (\bar{X} - \mu)^2\right] \\
&= \left[\sum_{i=1}^n (X_i - \mu)^2\right] - 2n(\bar{X} - \mu)^2 + n(\bar{X} - \mu)^2 \\
&= \left[\sum_{i=1}^n (X_i - \mu)^2\right] - n(\bar{X} - \mu)^2
\end{aligned}$$

(c) Note that $\mathbb{E}[(\bar{X} - \mu)^2] = \mathrm{Var}[\bar{X}] = \frac{1}{n^2}\sum_{i=1}^n \mathrm{Var}[X_i] = \frac{1}{n}\sigma^2$. Taking $\mathbb{E}$ on both sides of (9.40), we obtain

$$(n-1)\mathbb{E}[\sigma^2] = n\sigma^2 - n\frac{1}{n}\sigma^2.$$

Hence $\mathbb{E}[\sigma^2] = \sigma^2$.

6. (Implied volatility) Since by (7.26),

$$\text{Vega} = \frac{\partial c}{\partial \sigma}(t, S_0) = S_0\sqrt{T-t}N'(d_+) > 0,$$

the option price c is an increasing function of σ when other parameters K, r, and S_0 are fixed. We can use Bisection or Newton method to solve for $f(\sigma) = c(\sigma) - c_{\text{given}} = 0$ when c_{given} is a given option price.

Here is Newton's method for solving $f(x) = 0$ (here $x = \sigma$)

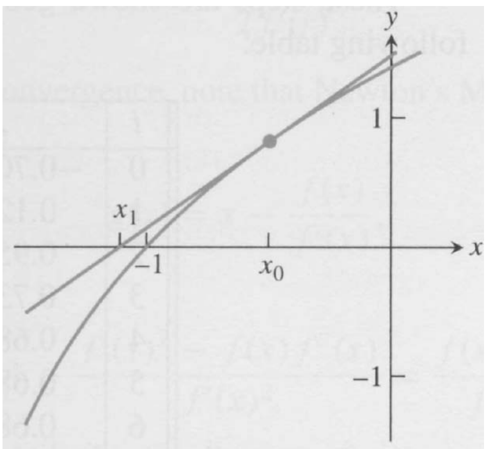


Figure 9.5: One step of Newton's method

Suppose x_* is the root of function $f(x)$. Since $0 = f(x_*) \overset{\text{Taylor}}{\approx} f(x_n) + f'(x_n)(x_n - x_*)$, $x_* \approx x_n - f(x_n)/f'(x_n)$. Hence we obtain the so called Newton iteration with initial guess x_0:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{9.43}$$

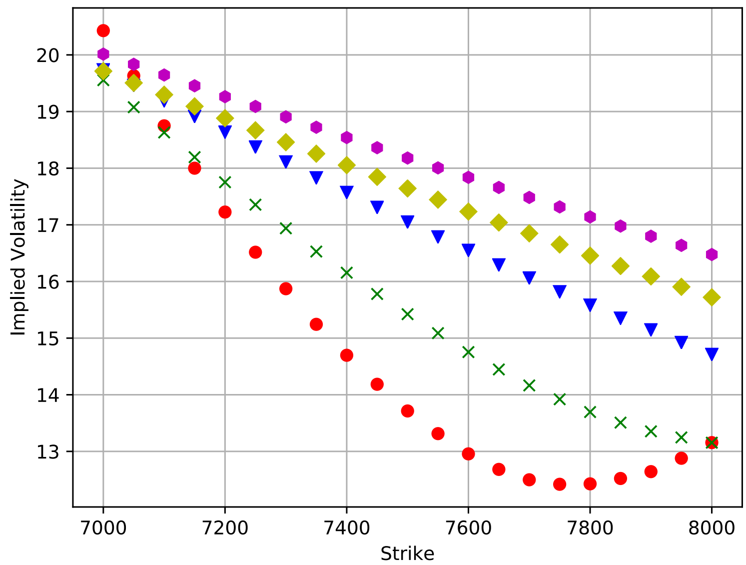
If you know how to play with Python code, download the two Python files `bsm_functions.py` and `DAX_imp_vol.py` from luminus. (They are written by Hilpisch. See his book "Derivatives Analytics with Python".) Try to understand the code and then answer the following questions:

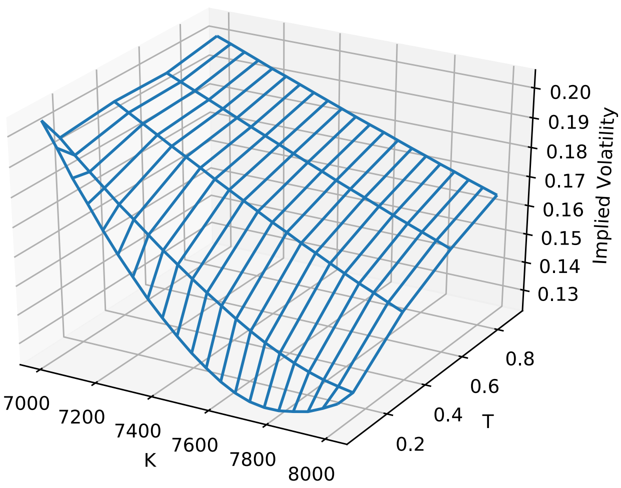
(a) Figure out which line of which file performs the calculation (9.43). Write down your answer besides the plots you need to submit in (b).

(b) Run the last Python code, printout the two figures, and then submit. Please note that the first plot shows that the implied volatility is not a constant.

However, if you do not know Python, download the Matlab code `Sect_9_ex.m` (this code is from Choe's book) from luminus. Print out the plot. Under the plot, write down the value of σ and write down the number of the line where (9.43) is performed. Then submit.

Solution:

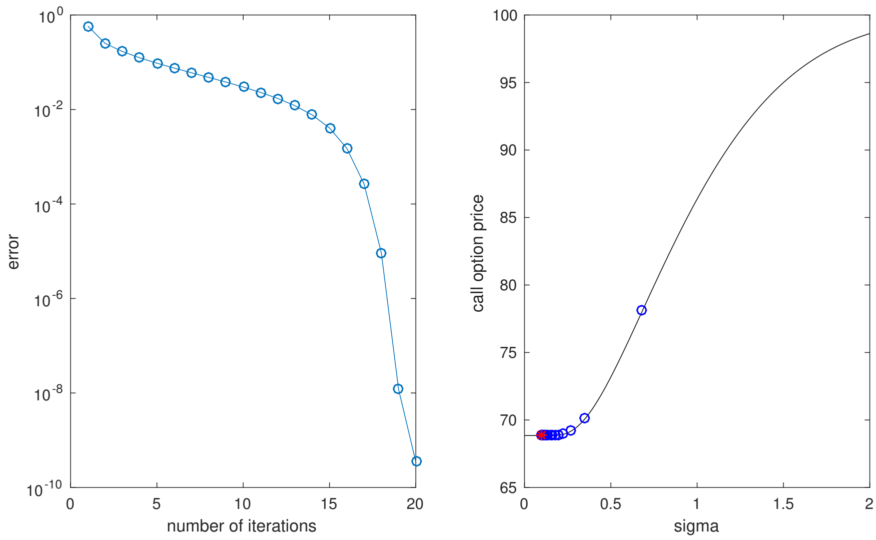




Lines -2 and -3 (or equivalently, lines 105 and 106) of bsm_functions.py performs (9.43).

```
sigma_est -= ((bsm_call_value(S0, K, T, r, sigma_est) - C0)
                    / bsm_vega(S0, K, T, r, sigma_est))
```

If you use the Matlab code Sect_9_ex.m, $\sigma = 0.1$. (Initial value $\sigma_0 = 0.6830$.)



line 18 performs (9.43).

```
sigma(i+1)=sigma(i) - F(sigma(i))/vega(sigma(i));
```

7. (Exponential Tilting) We say that $g = g_\theta$ is an exponential tilted density of f in (9.25) if

$$g_\theta = \frac{e^{\theta x} f(x)}{\mathbb{E}[e^{\theta X}]}, \qquad f \text{ is the pdf of } X. \tag{9.44}$$

Then

$$f(x)/g_\theta(x) = \mathbb{E}[e^{\theta X}]e^{-\theta x} = e^{\varphi(\theta)-\theta x} \tag{9.45}$$

where $\varphi(\theta) = \log \mathbb{E}[e^{\theta X}]$ is the log of the moment generating function (mgf) of $X \sim f$ (mgf has appeared before in Question 1 of Homework III).

a) Show that g_θ is a density, namely, $g_\theta \geq 0$ and $\int_{-\infty}^{\infty} g_\theta(x)dx = 1$.

b) Let $Y \sim g_\theta$. Prove that $\mathbb{E}[Y] = \varphi'(\theta)$.

c) If $f \sim N(\mu, \sigma^2)$, find g_θ.

Proof: (a) It is obvious that $g_\theta \geq 0$.

$$\int_{-\infty}^{\infty} g_\theta(x)dx = \frac{1}{\mathbb{E}[e^{\theta X}]} \int_{-\infty}^{\infty} e^{\theta x} f(x)dx = \frac{\mathbb{E}[e^{\theta X}]}{\mathbb{E}[e^{\theta X}]} = 1.$$

(b)

$$\begin{aligned}
\varphi'(\theta) &= \frac{1}{\mathbb{E}[e^{\theta X}]} \frac{d}{d\theta}\mathbb{E}[e^{\theta X}] = \frac{1}{\mathbb{E}[e^{\theta X}]} \int_{-\infty}^{\infty} \frac{d}{d\theta} e^{\theta x} f(x)dx \\
&= \frac{1}{\mathbb{E}[e^{\theta X}]} \int_{-\infty}^{\infty} x e^{\theta x} f(x)dx = \int_{-\infty}^{\infty} x g_\theta(x)dx = \mathbb{E}[Y].
\end{aligned}$$

(c) By Question 1 of Homework III, $\mathbb{E}[e^{\theta X}] = e^{\theta\mu}\mathbb{E}[e^{\theta(X-\mu)}] = e^{\mu\theta + \frac{1}{2}\sigma^2\theta^2}$. Hence $\varphi(x) = \log \mathbb{E}[e^{\theta X}] = \mu\theta + \frac{1}{2}\sigma^2\theta^2$.

$$g_\theta(x) = f(x)e^{-\varphi(x)+\theta x} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} e^{-\mu\theta - \frac{1}{2}\sigma^2\theta^2 + \theta x} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[x-(\mu+\theta\sigma^2)]^2}{2\sigma^2}}.$$

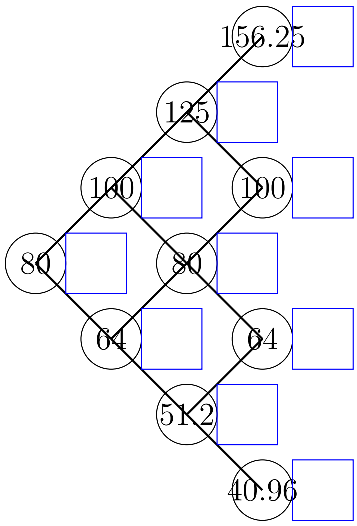
In other words, $g_\theta \sim N(\mu + \theta\sigma^2, \sigma^2)$.

8. Suppose you want to approximate $\mu = \mathbb{E}[h(X)]$ with $X \sim U(0,1)$ and $h(x) = \frac{1}{x^{1/3}} + \frac{x}{10}$ by Monte Carlo method.

(a) If you use simple sampling to approximate μ by $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^n h(X_i)$, where $X_1, ..., X_n$ are i.i.d. $U(0,1)$ random variables, what is the variance of $\hat{\mu}_n$? How is the variance of $\hat{\mu}_n$ related to $|\hat{\mu}_n - \mu|$? [Hint: In other words, find the number a so that with probability 0.95, $|\hat{\mu}_n - \mu| < a\sqrt{\mathrm{Var}(\hat{\mu}_n)}$.]

(b) If you use importance sampling based on the pdf $g(y) = \frac{2}{3y^{1/3}}$, what is the new formula of $\hat{\mu}_n$ once you have generated n i.i.d. random variables $Y_1, Y_2, ..., Y_n$ whose common pdf is $g(y)$? What is the variance of this new $\hat{\mu}_n$? (This variance can depend on n.)

9. Consider the problem of using binomial tree method to calculate the European and American put options with $S_0 = 80$, $K = 90$, and some given $\mathtt{T}$, r and σ. We construct a three step binomial tree with $\delta t = \mathtt{T}/3$, $u = e^{\sigma\sqrt{\delta t}} = 5/4$, $d = 1/u$, $\rho = e^{r\delta t} = 1.025$. Evaluate the European and American put option prices based on the binomial tree. Keep at least 5 significant digits in your calculation. (Note that the problems in Chapter 2 asked you to calculate call options, but here is a put option.)



10. Read the first 8 pages of the paper "Valuing American Options by Simulation: A Simple Least-Squares Approach" by Longstaff and Schwartz [80] and play with the Matlab code `Sect_9p12.m` on luminus. Answer the following two questions:

(a) Let $S_0 = 100$, $K = 100$, $r = 0.03$, $T = 1$. Choose 200 points in the time grid and generate 10000 sample paths. What are the American put option prices if $\sigma = 0.1$, 0.2, 0.3 respectively?

(b) On page 117 of the paper, the author mentioned "we regress Y on a constant, X, and X^2" which means that he assumes $\mathbb{E}[Y|X] = c_0 + c_1 X + c_2 X^2 = f(X)$ with some undetermined constant c_0, c_1, c_2. Then he tries to find c_0, c_1, c_2 so that $f(X_i) = Y_i$, where $(X_1, X_2, X_3, X_4, X_5) = (1.08, 1.07, 0.97, 0.77, 0.84)$ and $(Y_1, Y_2, Y_3, Y_4, Y_5) = (0, 0.07, 0.18, 0.20, 0.09) \times 0.94176$ according to the paper. It means that he is hoping to determine (c_0, c_1, c_2) so that

$$\begin{aligned} c_0 + c_1 X_1 + c_2 X_1^2 &= Y_1 \\ c_0 + c_1 X_2 + c_2 X_2^2 &= Y_2 \\ &\cdots\cdots \\ c_0 + c_1 X_5 + c_2 X_5^2 &= Y_5 \end{aligned}$$

[80] https://escholarship.org/uc/item/43n1k4jb or http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.155.3462

or, written in matrix-vector form

$$Az = b$$

with

$$A = \begin{pmatrix} 1 & X_1 & X_1^2 \\ 1 & X_2 & X_2^2 \\ \vdots & \vdots & \vdots \\ 1 & X_5 & X_5^2 \end{pmatrix},$$

$z = (c_0, c_1, c_2)^\top$, $b = (Y_1, Y_2, Y_3, Y_4, Y_5)^\top$. Apparently, there are too many equations than unknowns. Hence one can only find z so that the distance between Az and b is small, which means one wants to choose z so that

$$z = \text{argmin}_z \|Az - b\|_2^2,$$

where $\|d\|_2^2 = \sum_{i=1}^k d_i^2$ if $d = (d_1, \cdots, d_k)$. From calculus, we know that at the optimal z, the gradient of $\|Az - b\|_2^2 \stackrel{\text{def}}{=} g(z)$ should vanish. We also know that the gradient of g is a vector and its value is $\nabla g(z) = A^\top(Az - b)$. Hence the optimal z satisfies

$$A^\top Az = A^\top b.$$

Solving for z can be done in Matlab

```
x=[1.08,1.07,0.97,0.77,0.84]'
y=[0,0.07,0.18,0.20,0.09]'*0.94176
A=[ones(5,1),x,x.^2]
z=A\y
```

The last line is equivalent to `z=(A'*A)\(A'*y)` because in Matlab, if A is not a square matrix, `A\y` returns the least square solution z to `Az=y`.

Answer the following questions: What is the value of z you get from the above Matlab code? Does the result agree with what you have seen from the paper of Longstaff and Schwartz?