MATH 565 Monte Carlo Methods in Finance

Fred J. Hickernell **In-Class Final Exam** **Tuesday, December 4, 2012**

Instructions:

i. *This in-class part of the final exam has THREE questions for a total of* 65 *points possible. You should attempt them all.*

ii. *The time allowed is 120 minutes.*

iii. *Unless otherwise indicated, give answers to at least four significant digits.*

iv. *This test is closed book, but you may use 4 double-sided letter-size sheets of notes.*

v. *(Programmable) calculators are allowed, but they must not have stored text. Computers are also allowed, but only using MATLAB, Mathematica, or JMP. No internet access.*

vi. *Show all your work to justify your answers. Answers without adequate justification will not receive credit. Write out pseudo-code for the programs that you run to get your answers.*

vii. *In addition, as a precaution, submit soft copies of your programs to the Blackboard Dropbox after the exam is finished. If I have difficulty understanding your computational work, I may look at your programs.*

1. (25 marks)
Consider the problem of computing

$$\mu = \int_0^3 f(x)\,\mathrm{d}x.$$

Letting X_1, X_2 IID $\sim \mathcal{U}[0,3]$, define three very simple estimators:

$$\hat\mu_1 = \frac{3}{2}\left[f(0) + f(3)\right], \qquad \hat\mu_2 = \frac{3}{2}\left[f(X_1) + f(X_2)\right], \qquad \hat\mu_3 = \frac{3}{2}\left[f(X_1/2) + f((X_2+3)/2)\right].$$

a) Which of these estimators corresponds to *simple Monte Carlo* sampling and why?

Answer: $\hat\mu_2$*, the sampling is IID.*

b) Which of these estimators corresponds to *stratified* sampling and why?

Answer: $\hat\mu_3$*, there is one sample in* $[0, 3/2]$ *and one sample in* $[3/2, 3]$.

c) Each of these estimators involves the sum of two function values. Why is the coefficient in front of these estimators 3/2 rather than 1/2?

Answer: The length of the interval is 3*, not* 1.

d) For each of these estimators, prove that it is *unbiased* for general f or give an example of an f for which it is biased.

Answer: The first estimator is biased. For example, for $f(x) = x^2$, $\mu = 9$, *but*

$$\mathbb{E}(\hat\mu_1) = \hat\mu_1 = \frac{3}{2}\left[f(0) + f(3)\right] = \frac{3}{2}[0 + 9] = \frac{27}{2}.$$

Since $X_1, X_2 \sim \mathcal{U}[0,3]$, the probability density for these random variables is $1/3$ on the interval $[0,3]$. The other two estimators are unbiased because

$$\begin{aligned}
\mathbb{E}(\hat{\mu}_2) &= \frac{3}{2}\left[\mathbb{E}\{f(X_1)\} + \mathbb{E}\{f(X_2)\}\right] = 3\mathbb{E}\{f(X_1)\} = 3\int_0^3 f(x)\,\frac{\mathrm{d}x}{3} = \mu \\
\mathbb{E}(\hat{\mu}_3) &= \frac{3}{2}\left[\mathbb{E}\{f(X_1/2)\} + \mathbb{E}\{f((X_2+3)/2)\}\right] \\
&= \frac{3}{2}\left[\int_0^3 f(x/2)\,\frac{\mathrm{d}x}{3} + \int_0^3 f((x+3)/2)\,\frac{\mathrm{d}x}{3}\right] \\
&= \frac{3}{2}\left[\int_0^{3/2} f(t)\,\frac{2\mathrm{d}t}{3} + \int_{3/2}^3 f(z)\,\frac{2\mathrm{d}z}{3}\right] \qquad [t = x/2,\ z = (x+3)/2] \\
&= \int_0^3 f(t)\,\mathrm{d}t = \mu.
\end{aligned}$$

e) Compute the variance of each of these estimators *exactly* for the case where $f(x) = x$.

Answer: Since $\hat{\mu}_1$ is deterministic, its variance is always zero. For this particular f, note that

$$\begin{aligned}
\mathbb{E}(X_1) = \mathbb{E}(X_2) &= \int_0^3 x\,\frac{\mathrm{d}x}{3} = \frac{3}{2}, \\
\mathbb{E}(X_1^2) = \mathbb{E}(X_2^2) &= \int_0^3 x^2\,\frac{\mathrm{d}x}{3} = 3, \\
\mathrm{var}(X_1) = \mathrm{var}(X_2) &= \mathbb{E}(X_1^2) - [\mathbb{E}(X_1)]^2 = 3 - \frac{9}{4} = \frac{3}{4} \\
\mu &= \int_0^3 x\,\mathrm{d}x = \frac{9}{2} = 3\mathbb{E}(X_1).
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathrm{var}(\hat{\mu}_2) &= \mathrm{var}\left(\frac{3}{2}\left[f(X_1) + f(X_2)\right]\right) = \frac{9}{4}\,\mathrm{var}(X_1 + X_2) = \frac{9}{2}\,\mathrm{var}(X_1) = \frac{27}{8} = 3.375 \\
\mathrm{var}(\hat{\mu}_3) &= \mathrm{var}\left(\frac{3}{2}\left[f(X_1/2) + f((X_2+3)/2)\right]\right) = \frac{9}{4}\,\mathrm{var}\left(\frac{X_1 + X_2 + 3}{2}\right) \\
&= \frac{9}{16}\,\mathrm{var}\left(X_1 + X_2\right) = \frac{9}{8}\,\mathrm{var}(X_1) = \frac{27}{32} = 0.84375
\end{aligned}$$

Note that $\mathrm{var}(\hat{\mu}_3) < \mathrm{var}(\hat{\mu}_2)$.

2. (15 marks)
Suppose that $X_1, X_2, \ldots$ are IID random variables with logistic cumulative distribution function:

$$F(x) = \frac{1}{1 + \mathrm{e}^{-2x}}, \qquad -\infty < x < \infty.$$

Evaluate $\mathbb{P}(X_1 + \cdots + X_{10} \geq 1)$ using a Monte Carlo method with an error of no more than 0.01 with 99% confidence.

Answer: To generate X_i we may use the inverse cumulative distribution function method. Letting $u = F(x)$, we get

$$\begin{aligned} u &= \frac{1}{1+\mathrm{e}^{-2x}} \\ 1+\mathrm{e}^{-2x} &= \frac{1}{u} \\ x &= -\frac{1}{2}\log\left(\frac{1}{u}-1\right) =: F^{-1}(u) \end{aligned}$$

```
tic
n=1e5; %number of paths
d=10; %dimension
u=rand(n,d); %uniform random numbers
x=-0.5*log(1./u - 1); %transformed to get the logistic distribution
y=sum(x,2)>=1; %indicator random variable
p=mean(y); %estimated probability
errhat=2.58*sqrt(p*(1-p))/sqrt(n); %estimated error
disp(['The probability = ' num2str(p) ' +/- ' num2str(errhat)])
toc
disp(' ')

The probability = 0.36061 +/- 0.0039176
Elapsed time is 0.044213 seconds.
```

3. (25 marks)
A stock satisfies a geometric Brownian motion with an initial price of \$100, an interest rate of 2%, and a volatility of 40%.

a) Write a formula for the price of the stock a half year later.

Answer:

$$S(1/2) = 100\mathrm{e}^{(0.02-0.16/2)(1/2)+0.4\sqrt{1/2}X_1} = 100\mathrm{e}^{-0.03+.2828X_1}, \qquad X_1 \sim \mathcal{N}(0,1).$$

b) Write a formula to generate a stock price path where the stock is monitored every month for half of a year.

Answer:

$$\begin{gathered} t_j = j/12, \quad j=0,\ldots,6, \qquad S(0)=100, \qquad X_1,\ldots,X_6 \sim \mathcal{N}(0,1), \\ S(t_j) = S(t_{j-1})\mathrm{e}^{(0.02-0.16/2)(1/12)+0.4\sqrt{1/12}X_j} = S(t_{j-1})\mathrm{e}^{-0.005+0.1155X_j} \end{gathered}$$

c) Use Monte Carlo with 10^6 paths to estimate the price of a *European call* option maturing half a year later with a strike price of \$120. Estimate the error of your approximate option price. Is it better to use the formula from part a) or part b)? Why?

Answer: Both are equally accurate, but part a) is preferable because it uses less time.

```
%European option with half year maturity
tic
n=1e6; %number of paths
d=1; %number of time steps
s0=100; %initial stock price
r=0.02; %interest rate
vol=0.4; %volatility
K=120; %strike price
T=1/2; %time to maturity
dt=T/d; %time step
x=randn(n,d); %innovations in stock price
smat=s0*[ones(n,1) exp(cumsum((r-vol^2/2)*dt+vol*sqrt(dt)*x,2))]; %stock paths
payoff=max(smat(:,d+1)-K,0)*exp(-r*T); %European payoff
price=mean(payoff); %estimated option price
stdpay=std(payoff); %standard deviation of payoff
errhat=2.58*stdpay/sqrt(n); %estimated error
disp(['European call with ' num2str(T) ' year maturity and d = ' int2str(d)])
disp(['    takes ' num2str(toc) ' seconds'])
disp(['The price = ' num2str(price) ' +/- ' num2str(errhat)])
disp(['The standard deviation of the payoff divided by the option price = '...
    num2str(stdpay/price)])
disp(' ')

%European option with half year maturity, d=6
tic
d=6; %number of time steps
dt=T/d; %time step
x=randn(n,d); %innovations in stock price
smat=s0*[ones(n,1) exp(cumsum((r-vol^2/2)*dt+vol*sqrt(dt)*x,2))]; %stock paths
payoff=max(smat(:,d+1)-K,0)*exp(-r*T); %European payoff
price=mean(payoff); %estimated option price
stdpay=std(payoff); %standard deviation of payoff
errhat=2.58*stdpay/sqrt(n); %estimated error
disp(['European call with ' num2str(T) ' year maturity and d = ' int2str(d)])
disp(['    takes ' num2str(toc) ' seconds'])
disp(['The price = ' num2str(price) ' +/- ' num2str(errhat)])
disp(['The standard deviation of the payoff divided by the option price = '...
    num2str(stdpay/price)])
disp(' ')
```

```
European call with 0.5 year maturity and d = 1
    takes 0.080573 seconds
The price = 5.0865 +/- 0.035521
```

```
The standard deviation of the payoff divided by the option price = 2.7067

European call with 0.5 year maturity and d = 6
    takes 0.27357 seconds
The price = 5.1122 +/- 0.035549
The standard deviation of the payoff divided by the option price = 2.6952
```

d) Use Monte Carlo with 10^6 paths to estimate *standard deviation* of the payoff of a European call option maturing half a year later with a strike price of \$120. (You do not need to estimate the error of your approximate standard deviation.) How does the standard deviation of the payoff divided by the option price change if the time to maturity is increased to one year?

Answer: The ratio of the standard deviation to the payoff to the price decreases in size from 2.7 *to* 2.5 *as the time to maturity lengthens.*

```
%European option with one year maturity, d=1
tic
d=1; %number of time steps
T=1; %time to maturity
dt=T/d; %time step
x=randn(n,d); %innovations in stock price
smat=s0*[ones(n,1) exp(cumsum((r-vol^2/2)*dt+vol*sqrt(dt)*x,2))]; %stock paths
payoff=max(smat(:,d+1)-K,0)*exp(-r*T); %European payoff
price=mean(payoff); %estimated option price
stdpay=std(payoff); %standard deviation of payoff
errhat=2.58*stdpay/sqrt(n); %estimated error
disp(['European call with ' num2str(T) ' year maturity and d = ' int2str(d)])
disp(['    takes ' num2str(toc) ' seconds'])
disp(['The price = ' num2str(price) ' +/- ' num2str(errhat)])
disp(['The standard deviation of the payoff divided by the option price = '...
    num2str(stdpay/price)])
disp(' ')

European call with 1 year maturity and d = 1
    takes 0.084234 seconds
The price = 9.7879 +/- 0.062889
The standard deviation of the payoff divided by the option price = 2.4904
```

e) Use Monte Carlo with 10^6 paths to estimate the price of an *up and in barrier call* option monitored monthly, maturing half a year later, having a strike price of \$120 and having a barrier of \$140. How does this price compare to the price of the European call with the same strike price and maturity?

Answer: The barrier option is cheaper because fewer stock price paths breach the barrier.

```
%Barrier option with half year maturity
tic
```

```
d=6; %number of time steps
T=1/2; %time to maturity
dt=T/d; %time step
barrier=140; %barrier for up and in
x=randn(n,d); %innovations in stock price
smat=s0*[ones(n,1) exp(cumsum((r-vol^2/2)*dt+vol*sqrt(dt)*x,2))]; % stock paths
breachbarrier=any(smat>=barrier,2); %is the barrier breached?
payoff=max(smat(:,d+1)-K,0)*exp(-r*T).*breachbarrier; %barrier payoff
price=mean(payoff); %estimated option price
stdpay=std(payoff); %standard deviation of payoff
errhat=2.58*stdpay/sqrt(n); %estimated error
disp(['Barrier up and in call with ' num2str(T) ' year maturity and d = ' int2str(d)])
disp(['    takes ' num2str(toc) ' seconds'])
disp(['The price = ' num2str(price) ' +/- ' num2str(errhat)])
disp(['The standard deviation of the payoff divided by the option price = '...
    num2str(stdpay/price)])
disp(' ')

Barrier up and in call with 0.5 year maturity and d = 6
    takes 0.32784 seconds
The price = 4.2735 +/- 0.03526
The standard deviation of the payoff divided by the option price = 3.198
```