MATH 565 Monte Carlo Methods in Finance

Fred J. Hickernell **Fall 2009**
Take-Home Part of Final Exam **Due Noon, Friday, December 11**

Instructions:

i. *This take-home part of the final exam consists of TWO questions for a total possible of* 50 marks. *Answer both of them.*

ii. *You may consult any book, web page, software repository or other inanimate object. You may use the m-files on Blackboard. You* may not *consult any other person face-to-face, by telephone, by email, or by any other means.*

iii. *Show all your work to justify your answers. Submit hard copies of your derivations, programs, output, and explanations. Answers without adequate justification will not receive credit.*

1. (25 marks)
Consider an asset price $S(t)$ modeled by a geometric Brownian motion with an inital price of \$100, a risk-free interest rate of 3% and a volatility of 50%. Suppose that you sell an arithmetic mean put option with a strike price of \$100, an expiry time of one year in the future, and quarterly monitoring. What is the probability of you having to pay out *more than* \$30 *in today's dollars?* Compute the answer to an error of less than 1%.

```
n=10000; %number of samples
d=4; %number of time steps
x=randn(n,d); %Gaussian random numbers
s0=100; %initial stock price
T=1; %time to expiry
delta=T/d; %length of time step
r=0.03; %interest rate
sigma=0.5; %volatility
strike=100; %strike price
losstol=30; %tolerance for loss
s=s0*cumprod(exp((r-sigma*sigma/2)*delta+sigma*sqrt(delta)*x),2);
  %stock price paths
savg=mean(s,2); %arithmentic mean of stock price
payoff=max(strike-savg,0)*exp(-r*T); %payoff of arithmetic mean option
bigloss=payoff>30; %paths which give a loss
probbigloss=mean(bigloss) %probability of a loss
ciwidth=1.96*sqrt(probbigloss*(1-probbigloss))/sqrt(n)
  %confidence interval width of probability

probbigloss =
   0.168400000000000
ciwidth =
   0.007334733777745
```

2. (25 marks)
For a d-dimensional point set, $\{\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=1}^n \subset [0,1]^d$, the square discrepancy is

defined as

$$D^2(\{\boldsymbol{x}_i\}_{i=1}^n) = \left(\frac{13}{12}\right)^d - \frac{2}{n}\sum_{i=1}^n \prod_{j=1}^d \left\{1 + \frac{1}{2}\left|x_{ij} - \frac{1}{2}\right| \left[1 - \left|x_{ij} - \frac{1}{2}\right|\right]\right\}$$
$$+ \frac{1}{n^2}\sum_{i,k=1}^n \prod_{j=1}^d \left\{1 + \frac{1}{2}\left[\left|x_{ij} - \frac{1}{2}\right| + \left|x_{kj} - \frac{1}{2}\right| - |x_{ij} - x_{kj}|\right]\right\}$$

The discrepancy tells us the error of approximating a multidimensianal integral by a sample average. You do not need to show it, but the mean square discrepancy of a simple random sample is

$$E[D^2(\{\boldsymbol{x}_i\}_{i=1}^n)] = \frac{1}{n}\left[\left(\frac{5}{4}\right)^d - \left(\frac{13}{12}\right)^d\right].$$

Compute the square discrepancy of an unscrambled 4-dimensional Sobol' set for $n = 2, 4, \ldots, 2^{10}$ points and compare it to the mean square discrepancy of a simple random sample in terms of relative magnitudes and trends.

```
%% Problem 2
d=4; %dimension
nvec=2.^(1:10)'; %vector of sample sizes
nmax=nvec(end); %largest sample size
nn=length(nvec); %number of sample sizes
p=sobolset(d); %generate Sobol' sequence
x=p(1:nmax,1:d); %take first nmax values
%xminhalf=x-1/2; %x minus half
absxminhalf=abs(x-1/2);
constterm=(13/12)^d; %constant term in discrepancy
singlesum=cumsum(prod(1+0.5*absxminhalf.*(1-absxminhalf),2),1);
  %single sum term in discrepancy
%Now compute the double sum
for m=1:nn
    temp3=zeros(nmax,nn);
    for i=1:nmax
        temp1=1+0.5*(repmat(absxminhalf(i,:),nmax,1)+absxminhalf ...
            -abs(repmat(x(i,:),nmax,1)-x));
        temp2=cumsum(prod(temp1,2),1);
        temp3(i,:)=temp2(nvec)';
    end
    temp4=cumsum(temp3,1);
    temp5=temp4(nvec,:);
end
disc2=constterm-2*singlesum(nvec)./nvec+diag(temp5)./(nvec.*nvec)
  %add all terms together for discrepancy of Sobol set
disc=sqrt(disc2)
disc2rand=((5/4)^d-constterm)./nvec %discrepancy of random set
discrand=sqrt(disc2rand)
figure(1); %plot the two discrepancies
loglog(nvec,disc,'b-',nvec,discrand,'k--','linewidth',2)
```

```
xlabel('Sample Size')
ylabel('Square Discrepancy')
print -deps CompareDiscrepancy.eps

disc2 =
   0.791181399498456
   0.220059230003828
   0.064536224130863
   0.015300440831781
   0.004594718297612
   0.001446829534721
   0.000395854987815
   0.000115177678008
   0.000035005780713
   0.000010307910678
disc =
   0.889483782594408
   0.469104711129432
   0.254039808161758
   0.123694950712552
   0.067784351421339
   0.038037212499359
   0.019896104840279
   0.010732086377233
   0.005916568322330
   0.003210593508669
disc2rand =
   0.532021604938272
   0.266010802469136
   0.133005401234568
   0.066502700617284
   0.033251350308642
   0.016625675154321
   0.008312837577160
   0.004156418788580
   0.002078209394290
   0.001039104697145
discrand =
   0.729398111416716
   0.515762350767421
   0.364699055708358
   0.257881175383710
   0.182349527854179
   0.128940587691855
   0.091174763927089
   0.064470293845928
   0.045587381963545
   0.032235146922964
```

We see from the graph below that the discrepancy of the Sobol' sequence decays much more quickly than that of the simple random set.