Problem 1

Function:

function u = halton(p,n)

b = zeros(ceil(log(n)/log(p)),1);

for j = 1:n

i =1;

b(1) = b(1) + 1;

while b(i) > p - 1 + eps

b(i) = 0;

i = i + 1;

b(i) = b(i) + 1;

end

u(j) = 0;

for k = 1: length(b(:))

u(j) = u(j) + b(k) \* p ^ (-k);

end

end

end

Main:

Method 1

n = 10^5; x = rand(4,n); b = sum(x.\*x); s = 0;

for i = 1:n

if b(i) <= 1

s = s+1;

end

end

s/n \* 16

Method 2

p1 = 2; p2 = 3; p3 = 5; p4 = 7; n = 10 ^ 5;

w = halton(p1,n);

x = halton(p2,n);

y = halton(p3,n);

z = halton(p4,n);

s = 0;

for i = 1:10 ^5

if (2 \* w(i) - 1)^2 + (2 \* x(i) - 1)^2 + (2 \* y(i) - 1)^2 + (2 \* z(i) - 1)^2 < 1

s = s + 1;

end

end

s / 10^5 \* 16

pi ^ 2 / 2

Result:

The approximation of the volume of the ball with Monte Carlo is 4.8922

The approximation of the volume of the ball with quasi - Monte Carlo is 4.9293

The exact volume is . It shows that our results are very close to the exact number. Specially, quasi - Monte Carlo approximation is better than the original one.

Problem 2

Main:

m = 0; n = 10000;

for j = 1:n

s = 0; b = 0; k = 1000; sqdelt = sqrt(0.01);

for i = 1:k

a = b;

b = b + sqdelt \* randn;

if i >= 300 & i <= 500 % change interval for different questions

if a \* b <= 0

s = 1;

end

end

end

if s == 0;

m = m + 1;

end

end

(2 / pi) \* asin(sqrt(3/5))

m/10000

1. For interval [3, 5], the result we get is 0.5788, and the exact probability is . It shows that they are very close to each other.
2. For interval [2, 10], the result we get is 0.3111, and the exact probability is . It show that they are very close to each other.
3. For interval [8, 10], the result we get is 0.7189, and the exact probability is . It show that they are very close to each other.

Problem 3 (on the previous page)

Problem 4

Function:

function E = european\_option\_price(g, S0, r, T, sigma, zmin, zmax, m)

h = (zmax - zmin)/m;

z = zmin : h : zmax; % m+1 points

f = @(z)exp(-r \* T) \* g(S0 \* exp(sigma \* sqrt(T) \* z + (r - sigma ^ 2 / 2) \* T)) \* normpdf(z);

y0 = f(zmin); ym = f(zmax);

sumyi = 0;

for i = 2:m

sumyi = sumyi + f(z(i));

end

E = h / 2 \* (y0 + ym + 2 \* sumyi);

end

function price = Cbs(T,S,K,r,sigma)

d1 = (log(S/K) + (r + sigma^2/2) \* T)/(sigma \* sqrt(T));

d2 = d1 - sigma \* sqrt(T);

price = S \* normcdf(d1) - K\*exp(-r \* T) \* normcdf(d2);

end

Main:

S0 = 100; r = 0.01; sigma = 0.2;

K = 100; T = 1;

g = @(x)max(x - K, 0);

zmin = -10; zmax = 10; h = []; e = [];

for i = 1:15

m = 2^i;

E = european\_option\_price(g, S0, r, T, sigma, zmin, zmax, m);

price = Cbs(T,S0,K,r,sigma);

error = abs(E - price);

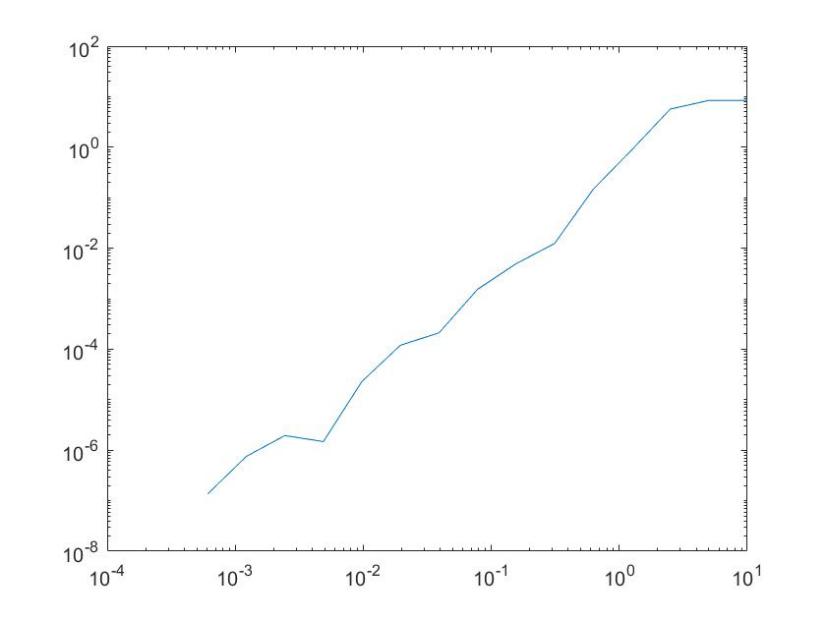
e = [e, error];

h = [h, (zmax - zmin)/m];

end

loglog(h, e)

The plot is as follows:



Finding that absolute error approach zero as h tends to zero

Problem 5

(b)

T =1; K = 100; B = 120; r = 0.1; sigma = 0.25; S0 = 100;

M = 1000; delt = T/M;

t = 0:delt:T; % M + 1 points

N = 1000;

k = M;

S = [];

for i = 1:N

b = 0; sqdelt = sqrt(delt); St = [];

for j = 1:k+1

tk = t(j);

b = b + sqdelt \* randn;

Stk = S0 \* exp((r - 1/2 \* sigma^2) \* tk + sigma \* b);

St = [St, Stk];

end

S = [S; St];

end

(c):

T =1; K = 100; B = 120; r = 0.1; sigma = 0.25; S0 = 100;

M = 1000; delt = T/M;

t = 0:delt:T; % M + 1 points

N = 10000;

k = M; VT = 0;

for i = 1:N

b = 0; sqdelt = sqrt(delt); l = 0;

for j = 1:k+1

tk = t(j);

b = b + sqdelt \* randn;

Stk = S0 \* exp((r - 1/2 \* sigma^2) \* tk + sigma \* b);

if Stk >= B

l = 1;

end

end

if l == 0;

VT = VT + max(Stk - K, 0);

end

end

E = VT/N \* exp(-r \* T)

When N = 10000, the result of our estimation is 0.7450;

When N = 100000, the result of our estimation is 0.7381.