Take Home Exam 1: exercise 4

Alessandro La Farciola

October 13, 2022

In this file there are the Matlab codes and the final plot for *Exercise 4*.

Exercise 4a

```
function P = BinomialpriceBarrierUODM(r,d,u,N,T,s,K,b)
% Input:
% r = risk free interest rate
% d,u = possible returns on the underlying asset
% N = number of time-intervals
% T = maturity
% s = initial asset price
% K = strike value
% b = value of the barrier

% Output:
% P = price of an up-and-out option in the multi-period binomial model

%define needed parameters
r_tilde=r*T/N;
disc=1/(1+r_tilde);
q_u=((1+r_tilde)-d)/(u-d);

%vector of intermediate prices
X=zeros(N+1,1);

%payoff at maturity
for i= 1:N+1
    if(s*u^(i-1)*d^(N+1-i)<b)
        X(i)=max(s*u^(i-1)*d^(N+1-i)-K,0);
    else
        X(i)=0;
    end
end

%calculation of expectation by iterative procedure

for k=N:-1:1
    for i= 1:k
```

```
            if(k==(N/2+1))
                if(s*u^(i-1)*d^(k-i)<b)
                  X(i)=disc*(q_u*X(i+1)+(1-q_u)*X(i));
                else
                  X(i)=0;
                end
            else
                X(i)=disc*(q_u*X(i+1)+(1-q_u)*X(i));
            end
        end
    end

%final option price
P=X(1);
end
```

Exercise 4b

```
function P = MCpriceBarrierUODM(r,sigma,Nt,Ns,T,s,K,b)
% Input:
% r = risk-free interest rate
% sigma = volatility
% Nt = number of time interval
% Ns = number of simulations
% s = initial asset price
% T = maturity
% K = strike value
% b = value of the barrier

% Output:
% P = price of an up-and-out option in the multi-period binomial model

%definition of vector of payoffs
p=zeros(Ns,1);

%simulations of paths using the function SimBS of exercise 2.5 a)
%and computation of payoffs
for i=1:Ns
    [t,X]=SimBS(r,sigma,s,T,Nt);
    if(X(Nt+1)<b && X(Nt/2+1)<b)
        p(i)=max(X(Nt+1)-K,0);
    else
        p(i)=0;
    end
end

%Monte Carlo price at t=0
P= (exp(-r*T)/Ns)*sum(p);
end
```

Exercise 4c

In this section, I firstly present the script to obtain the plot.

```
%compare the binomial prices with the MC price for N->infty

%set parameters
s=1; r=0.1; T=0.5; K=0.9; sigma=0.1; b=1.3;

%MC price
Nt=100;
Ns=10^6;
PMC=MCpriceBarrierUODM(r,sigma,Nt,Ns,T,s,K,b);

%Binomial prices
P=zeros(1,100);
for N=2:2:200
    d=1+r*T/N-sigma*sqrt(T)/sqrt(N);
    u=1+r*T/N+sigma*sqrt(T)/sqrt(N);
    P(N/2)=BinomialpriceBarrierUODM(r,d,u,N,T,s,K,b);
end

%plot
plot((2:2:200),P,(2:2:200),PMC.*ones(1,100))
title('Comparison between binomial prices and MC price of UODM')
xlabel('N')
ylabel('UODM option price')
legend('Binomial prices','MC price')
```

Now, the following figure shows the plot obtained with the previous script.

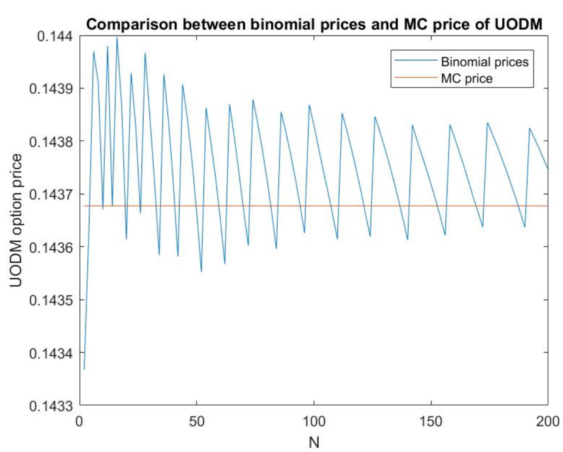
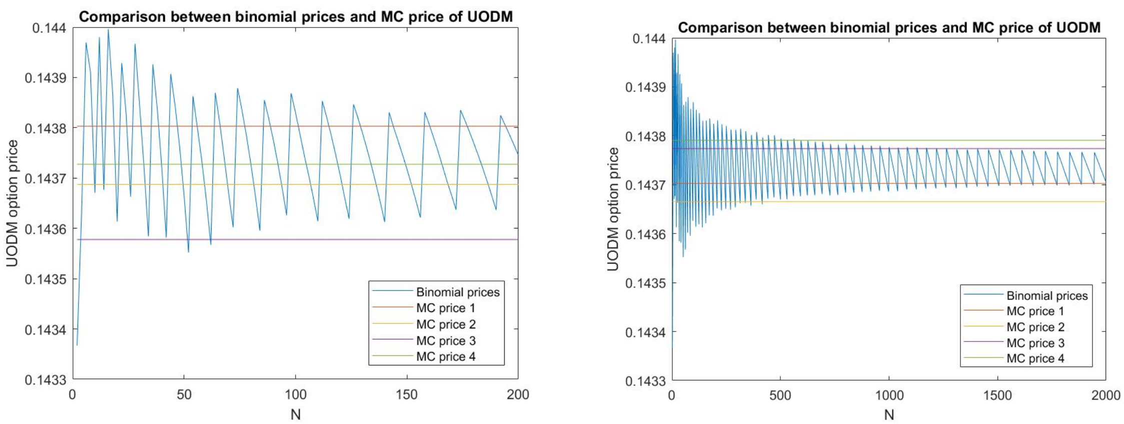


Figure 1: Comparison between the binomial prices and MC price

In the previous figure 1 we can observe the differences between the prices obtained with the binomial method and the Monte Carlo one. We can note that the binomial method tends to return a price of the up-and-out call barrier option a little bit higher than the other simulation method. Moreover, we can say that when N is bigger, the binomial price is getting a little bit closer to the MC price, even if it is oscillating. However, this plot does not guarantee us its convergence.

Since the second method is a simulation method and our option is a path-dependent option, it seems reasonable to make different tests for the MC price.



a) $N \leq 200$ b) $N \leq 2000$

Figure 2: Comparison between the binomial prices and 4 simulations of MC price

As we can see from figure 2a we obtain 4 different values of the price with MC simulation which complicate the convergence argument of the binomial method. The same result is visible in figure 2b, where I computed the binomial price until $N = 2000$. However, in this plot it seems more likely a convergence of binomial prices.