MATLAB

Lecture 5

Financial engineering : European and American Options

- **A European call (put) option** is a contract which gives the buyer
- the right to buy (sell) an asset at a future time T for a price K. The
- underlying asset, the maturity time T and the strike price K are
- specified in the contract.

- **An American call (put) option** is a contract which gives the buyer
- the right to buy (sell) an asset at a future time $t <= T$ for a price K
- .The underlying asset, the maturity time T and the strike price K are
- specified in the contract.

European call option

- **A European call option** is a contract which gives the buyer the right to buy an asset at a future time T for a price K.

- This is a right but not obligation!

- If at time T price of asset $S(t)>K$ we use this right and have a profit $S(t)-K$
- If at time T price of asset $S(t)<K$ we are not using this right

European call option

- How much we have to pay now for this right (financial instrument)?

European call option

- How much we have to pay now for this right (financial instrument)?

European call option

- Let we discuss discrete time in a very simple setting
- At the time zero (now) t=0, we have an asset with price S_0
- At the time t=1 price can
  - go up and become S_0*u
  - or can go down and become S_0*d

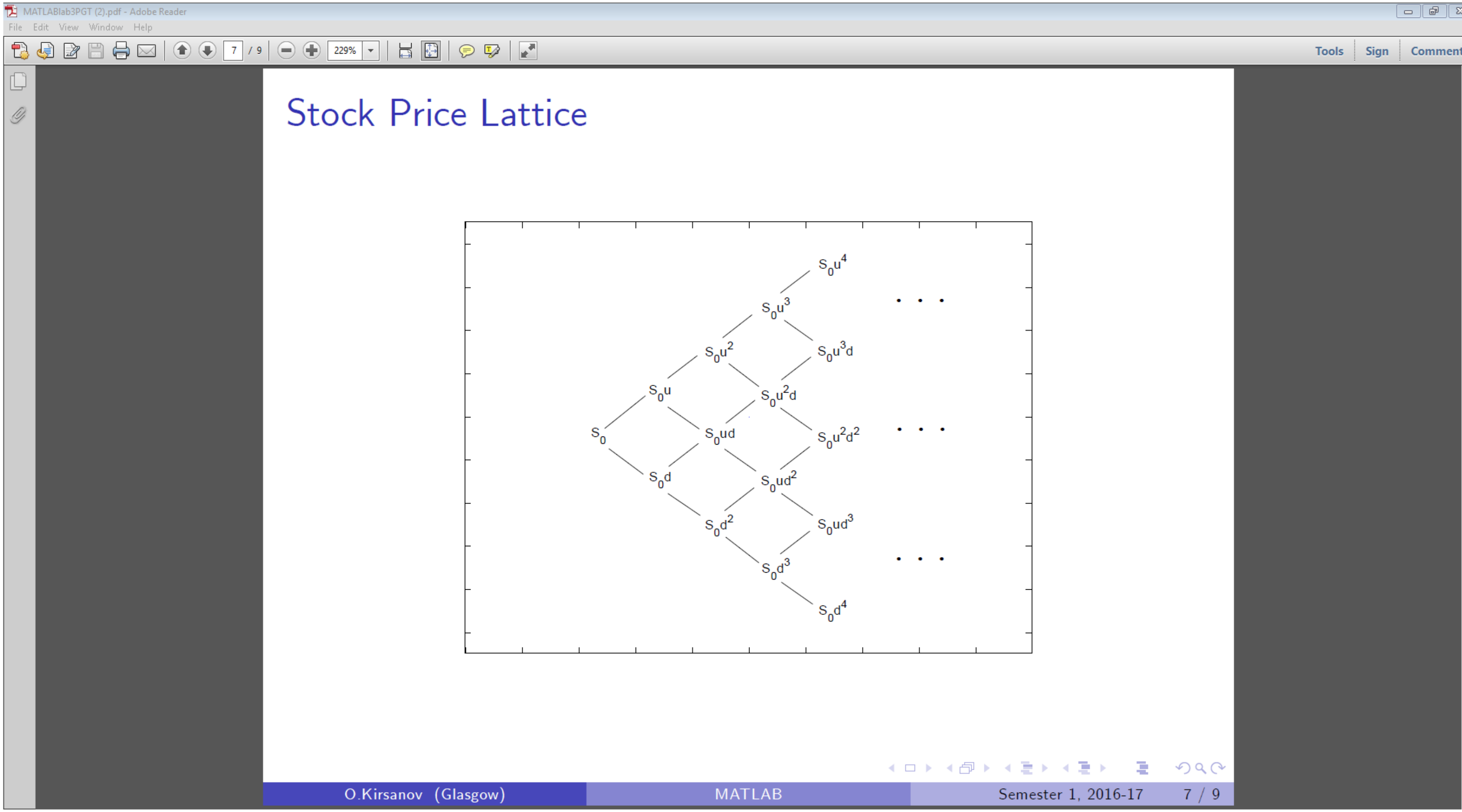
$$
\begin{tikzcd}
 & S_0\text{*u} \\
S_0 \arrow[ur] \arrow[dr] & \\
 & S_0\text{*d}
\end{tikzcd}
$$

European call option

- At the time t=2 price again can up or down with the same numbers u , d

$$
\begin{tikzcd}[column sep=large]
 & & S_0\text{*u*u} \\
 & S_0\text{*u} \arrow[ur] \arrow[dr] & \\
S_0 \arrow[ur] \arrow[dr] & & S_0\text{*u*d (we assuming } S_0\text{*u*d= } S_0\text{*d*u)} \\
 & S_0\text{*d} \arrow[ur] \arrow[dr] & \\
 & & S_0\text{*d*d}
\end{tikzcd}
$$

European call option



- Up until final time T, t=T

European call option

- How we can describe the price process?
- We know S_0, u and d.
- Let us create matrix S:

$$\begin{bmatrix} S_0 & S_0*u & S_0*u*u & S_0*u*u*u \\ & S_0*d & S_0*u*d & S_0*u*u*d \\ & & S_0*d*d & S_0*u*d*d \\ & & & S_0*d*d*d \end{bmatrix}$$

European call option

- Each element of this matrix is:
  - $S_0 u^i d^j$ where (0 <= i , j <= N)

- Let us consider the first row:
  - You can see that each element is previous one multiplied by u
  - Starting from S(1,1)= S_0 ,

    S(1,2)= S(1,1)*u, S(1,3)= S(1,2)*u, …..

- Let us consider the second row:
  - You can see that each element is previous one multiplied by u but starting from S(2,2)= $S_0 * d$

- Third row:

– starting from S(3,3)= $S_0 * d * d$

– each element is previous one multiplied by u …….

European call option

- We can create diagonal matrix with elements on diagonal:
- S_0, S_0*d, S_0*d*d,……
- Than follow the idea we discussed in previous slides .

- Alternatively :

```
SS = zeros(N,T);
SS(1,1) = S0;
for t = 2:T;
    for n = 1:t;
        SS(n,t) = S0*u^(t-n)*d^(n-1);
    end;
end;
```

European call option

- At the final time t=T

- We ether use the right to buy assets (**executing option**) at the price K
  - In this case we have profit S_T-K (**payoff**)
- or we are not buying assets (**payoff=0**)

- Summarize : At the final time t=T we have payoff vector
- **max(0,** S_T **−K)**

European call option

- In matlab:

- Final time correspond to the last column in our matrix, so:

- ```
  payoff = max((SS(:,T)-strike),0);
  ```
  ```

European call option

- The question we need to answer:
- (**what is the price of European option?)**
- Is the same question as :
- What is the price of financial instrument in time $t=0$ which provide that payoff at time $t=T$

European call option

- Let us discuss single elementary period model:

- $max(S^1{}_1$-K,0)
- C_0
- $max(S^2{}_1$-K,0)
- payoff

- We know $S^2{}_1 S^2{}_1$,K . How to find out C_0?

European call option

- Let price C_0 go up with RNP probability P
- p
- $max(S^1{}_1$-K,0)
- C_0
- $max(S^2{}_1$-K,0)
- (1-p) payoff

- Then

C_0=(p* $max(S^2{}_1$-K,0)+$(1-p) *$ $max(S^1{}_1$-K,0)) /(1+r)

European call option

- In this simple model and under assumption of arbitrage free market

- P- Risk Neutral Probability

$$P = \frac{1+r-d}{u-d}$$

Where r is interest rate

European call option

- Dealing with multi period model:

$$max(S^{1}{}_{1}\text{-K,0})$$

$$C_{T-1}{}^{1}$$

$$max(S^{2}{}_{1}\text{-K,0})$$

$$C_{T-2}{}^{1}$$

$$C_{T-1}{}^{2}$$

$$max(S^{2}{}_{1}\text{-K,0})$$

$$C_{T-1}{}^{3}$$

$$max(S^{2}{}_{1}\text{-K,0})$$

....

- This called backwards induction process

European call option

- backwards induction process –

- Start with time t=T, we go to t=T-1
- We continue until t=0

  (time zero- is time “now”)

  At the time t=0

  PE ← **price of European call option**

  Any single time we are dealing only with set of elementary models.

European call option

- Algorithm:
- Let us have column vector payoff and probability p.

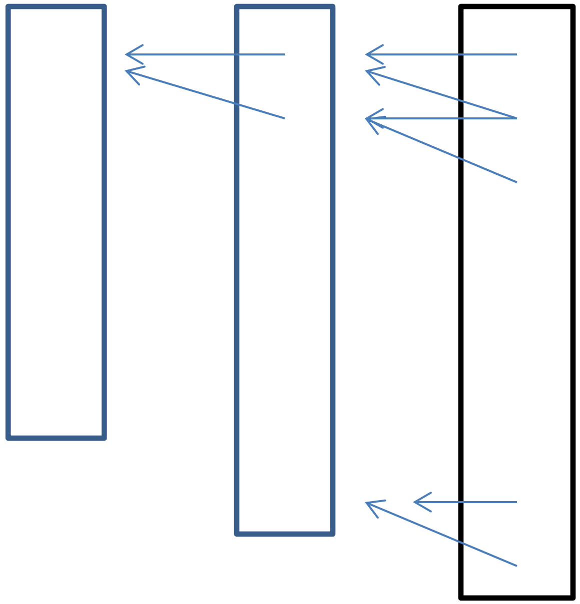
Payoff(1)= (p*payoff(1)+ (1-p)*payoff(2)) /(1+r)

Payoff(2)= (p*payoff(2)+ (1-p)* payoff(3)) /(1+r)

.....

.....

Payoff(n-1)= (p*payoff(n-1)+ (1-p)* payoff(n))//(1+r)



European call option

- Alternatively
-

```
PayOffMatr(:,T) = payoff;
for j=T-1:-1:1
        for k = 1:N-1
             PayOffMatr(k,j) = (PayOffMatr(k,j+1)*rnp ...
             + PayOffMatr(k+1,j+1)*(1-rnp))/(1+r);
        end;
end;
```

In this case we save entire matrix of payoffs

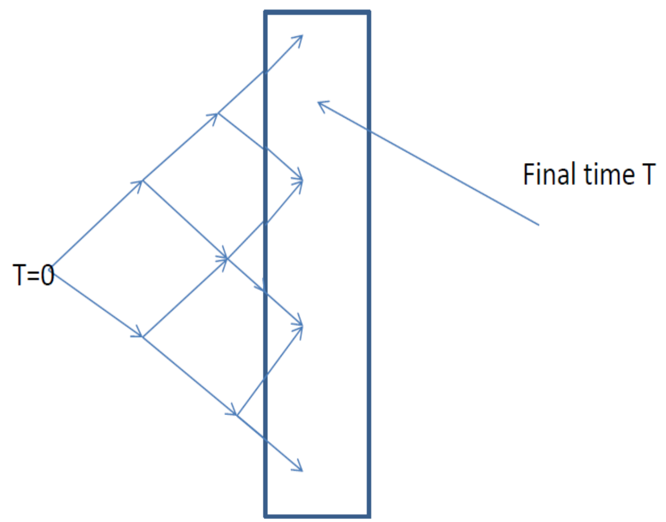
European call option

- r = 0.05; u = 11/10; d = 10/11; strike = 0.9; S0 = 1; T = 5; N = T;
- SS = zeros(N,T);PayOffMatr = zeros(N,T); SS(1,1) = S0;
- for t = 2:T;
- for n = 1:t;
- SS(n,t) = S0*u^(t-n)*d^(n-1);
- end;
- end;
- rnp = (1+r-d)/(u-d);payoff = max((strike-SS(:,T)),0);
- PayOffMatr(:,T) = payoff;
- for j=T-1:-1:1
- for k = 1:N-1
- PayOffMatr(k,j) = (PayOffMatr(k,j+1)*rnp ...
- + PayOffMatr(k+1,j+1)*(1-rnp))/(1+r);
- end;
- end;

European call option

- **Knowing payoffs and RNP**
  - **You do not need to know u, d, S_0 to calculate price of option**

European call option



European option

Price of European option depend only on prices in final time T

European put option

- **A European put option** is a contract which gives the buyer the right to **sell** an asset at a future time T for a price K

- **Exactly the same logic but different payoff:**

$$\max(0, K - S_T)$$

American options

- Difference between American and European options:

- American option can be executed any time
- European option can be executed at final (maturity) time only

American options

- Price of American **call** option is exactly the same as price of European call option (no dividends).

- Let we discuss American put option

American put option

- **An American put option** is a contract which gives the buyer the right to sell an asset at a future time t <= T for a price K

- The underlying asset, the maturity time T and the strike price K are specified in the contract.

American put option

- At any time t the asset holder can sell asset at price K or keep it until time t+1.
- asset holder sells an asset if it will be more profitable than keep an asset one more period.
- What does this mean?

American put option

- 2 cases
- If we keep the asset we follow the same logic as in case of European option and
- $\boldsymbol{PA_{keep}}$=(p* $\boldsymbol{max(K - S^2{}_t}$,0)+$(1-p) *$ $\boldsymbol{max(K - S^1{}_t}$K,0)) /(1+r)
- $max(K - S^1{}_t$,0)
- PA_{keep}
- $max(K - S^2{}_t$,0)

- If we sell the asset at time t-1
- We have a profit **K-$\boldsymbol{S_{t-1}}$**

American put option

- asset holder sells the asset if it will be more profitable than keep the asset one more period

- Mathematically this means we select the biggest number of these two:

(p* $\boldsymbol{max(K - S^2{}_t},$**0)+**$\boldsymbol{(1-p) * max(K - S^1{}_t}$**K,0)) /(1+r)**

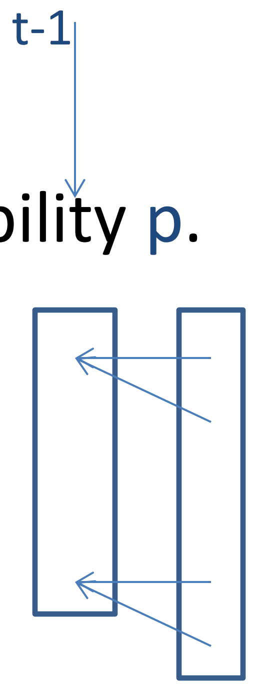
and

K-$\boldsymbol{S_{t-1}}$

- Max(**K-**$\boldsymbol{S_{t-1}}$**, (p*** $\boldsymbol{max(K - S^2{}_t},$**0)+**$\boldsymbol{(1-p) * max(K - S^1{}_t}$**K,0)) /(1+r))**

American put option

- European option
- Algorithm:
- Let us have column vector payoff and probability p.



Payoff(1)= (p*payoff(1)+ p*payoff(2)) /(1+r)

Payoff(2)= (p*payoff(2)+ p*payoff(3)) /(1+r)

…..

…..

Payoff(n-1)= (p*payoff(n-1)+ p*payoff(n))//(1+r)

- American option
- **Payoff(1)= max((p*payoff(1)+ p*payoff(2)) /(1+r), K- S_{t-1})**
- ..
- **Payoff(n-1)= max((p*payoff(n-1)+ p*payoff(n)) /(1+r), K- S_{t-1})**

American put option

- Alternatively

- **PayOffMatr(:,T) = payoff;**
- **for j=T-1:-1:1**
- **for k = 1:N-1**
- **PayOffMatr(k,j) = max(((PayOffMatr(k,j+1)*rnp ...**
- **+ PayOffMatr(k+1,j+1)*(1-rnp))/(1+r)),(strike-SS(k,j)));**
- **end;**
- **end;**

American put option

- Alternatively

```
for j=T-1:-1:1
        for k = 1:N-1
          PayOff0 = (PayOffMatr(k,j+1)*rnp ...
          + PayOffMatr(k+1,j+1)*(1-rnp))/(1+r);
          if PayOff0 > (strike-SS(k,j))
            PayOffMatr(k,j) = PayOff0; IndWait(k,j) = 1;
          else
            PayOffMatr(k,j) = (strike-SS(k,j)); IndWait(k,j) = 0;
          end;
        end;
end;
```

American put option

- Example:

```
r = 0.05; u = 11/10; d = 10/11; strike = 0.9; S0 = 1; T = 5; N = T;
SS = zeros(N,T);PayOffMatr = zeros(N,T); SS(1,1) = S0;
for t = 2:T;
for n = 1:t;
SS(n,t) = S0*u^(t-n)*d^(n-1);
end;
end;
rnp = (1+r-d)/(u-d);payoff = max((strike-SS(:,T)),0);
PayOffMatr(:,T) = payoff;
for j=T-1:-1:1
for k = 1:N-1
PayOffMatr(k,j) = max(((PayOffMatr(k,j+1)*rnp ...
+ PayOffMatr(k+1,j+1)*(1-rnp))/(1+r)),(strike-SS(k,j)));
end;
end;
```