Binomial Lattice Construction and Analysis

Author: Will Noone

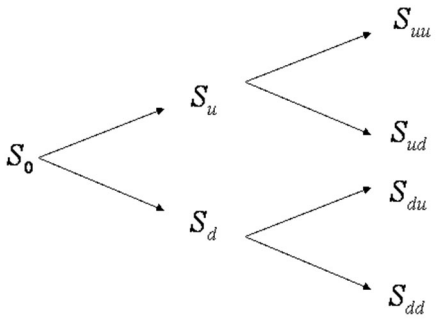
Abstract

This paper investigates the binomial model for option pricing, and the various models that have evolved from it. Part I describes the Binomial Lattice model, moment matching, CRR (Cox, Ross & Rubenstein), and the Jarrow-Rudd equal probability approach. The CRR and Jarrow-Rudd are two frameworks used for the binomial lattice model and will provide further insight into basic binomial modelling. Part II begins to elaborate on implementing the lattice and producing option prices with the Greeks as the convergence in price in regards to the Black-Scholes Model. Finally, Part III will examine the effects of dividends and exercise rights with respect to the valuation of American vs European options.

Part I
Binomial Lattice Model

A. Description of the Binomial Lattice Model

The binomial lattice model is an option pricing model that contains a binomial tree to show different paths that an underlying asset may observe over its life (see figure below). A lattice model can take into account expected changes in various restrictions such as volatility over the life, providing a better estimate of option prices compared to the Black-Scholes model. While the model requires more time than the Black–Scholes formula, it is more accurate, particularly for longer-dated options on securities with dividend payments, which makes it a better model for traders in the options markets.



This models flexibility in integrating expected volatility changes is useful in certain situations, such as pricing employee options at early-stage companies. Companies may expect lower volatility in their stock prices in the future as their businesses mature. This assumption can be factored into a lattice model, allowing more accurate option pricing than the Black-Scholes model, which contribute the same level of volatility over the life of the option[1].

A portfolio replicates a call option if the outcomes are the same at both states[2]:

$$\Delta S_u + B = C_u$$

$$\Delta S_d + B = C_d$$

B and Δ are the two unknowns. After isolating B, Δ can be found by the following equation[3]:

[1] Wikipedia: http://en.wikipedia.org/wiki/Binomial_options_pricing_model

[2] Chance, Don M., A Synthesis of Binomial Option Pricing Models for Lognormally Distributed Assets, p. 2

$$\Delta = \frac{C_u - C_d}{S(u-d)}$$

Since c_u and c_d are already known, we substitute Δ in either equation and solve for B. After solving for B, the resulting formula can be used as the value of a call option[4]:

$$c = \frac{\pi C_u + (1-\pi) C_d}{e^{rt}}$$

The risk-neutral probability, or pseudo-probability, may be calculated as follows:

$$\pi = \frac{e^{rt} - d}{u - d}$$

B. Moment Matching

The goal of moment matching is to approximate the sum of lognormal assets. Gilli and Schumann (2009) describe moment matching, which is related to finding values for the up movement and down movement. They go on to note one should match the mean and the variance of the underlying asset in a binomial tree with the mean and variance in a continuous time world. In the following formulas, M denotes the number of periods with M=1 and the current value of the underlying asset is S_0[5]:

$$E\left(\frac{S_{\Delta t}}{S_0}\right) = pu + (1-p)d$$

$$Var\left(\frac{S_{\Delta t}}{S_0}\right) = \frac{1}{S_0^2} Var(S_{\Delta t}) = pu^2 + (1-p)d^2 - (pu + (1-p)d)^2$$

The formula below is used to obtain the Var equation[6]:

$$Var(s_{\Delta t}) = S_0^2(pu^2 + (1-p)d^2) - s_0^2(pu + (1-p)d)^2$$

The mean gross return is $e^{r\Delta t}$. This observation allows for the derivation of the next equation[7]:

$$pu + (1-p)d = e^{r\Delta t}$$

The underlying asset has lognormally distributed returns with a variance of[8]:

---

[3] Chance, Don M., A Synthesis of Binomial Option Pricing Models for Lognormally Distributed Assets, p. 3
[4] Chance, Don M., A Synthesis of Binomial Option Pricing Models for Lognormally Distributed Assets, p. 4
[5] Gilli, M., and E. Schumann. p.3
[6] Gilli, M., and Shcumann, E., p.4
[7] Gilli, M., and Shcumann, E., p.4

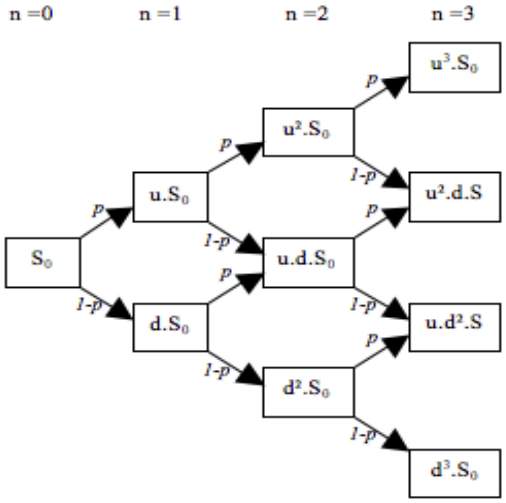
$$Var(S_{\Delta t}) = S_0^2 e^{2r\Delta t}\left(e^{\sigma^2 \Delta t} - 1\right) = S_0^2 (e^{2r\Delta t + \sigma^2 \Delta t} - e^{2r\Delta t})$$

If the equation above is divided by S_0^2 and equated to the original Var equation, one can obtain the following[9]:

$$pu^2 + (1-p)d^2 = e^{2r\Delta t + \sigma^2 \Delta t}$$

C. CRR Model

In 1979, Cox, Ross, and Rubinstein (CRR) developed one of the most commonly used methods of binomial modeling. The model uses a "discrete-time" (lattice based) model of the varying price over time of the underlying financial instrument in which binomial options pricing models do not have closed-form solutions[10]:



$$p = \frac{e^{rt/n} - d}{u - d}$$

$$u = e^{\sigma\sqrt{t/n}}$$

$$d = e^{-\sigma\sqrt{t/n}}$$

$$ud = 1$$

The assumption provided in the equation above then allows for the solving of the risk-neutral probability to be[11]:

$$p = \frac{e^{r\Delta t} - d}{u - d}$$

---

[8] Gilli, M., and Shcumann, E., p.4
[9] Gilli, M., and Shcumann, E., p.4
[10] http://en.wikipedia.org/wiki/Binomial_options_pricing_model
[11] Chance, Don M., A Synthesis of Binomial Option Pricing Models for Lognormally Distributed Assets, p. 7

Following CRR, u and d[12] can be written as follows:

$$u = e^{\sigma\sqrt{\Delta t}}$$

$$d = e^{-\sigma\sqrt{\Delta t}}$$

D. Jarrow-Rudd

Another popular binomial model is the Jarrow-Rudd model, or equal probability model. The third equation from their proposed approach is p = $\frac{1}{2}$. This leads to the following[13]:

$$u = e^{\left(r-\frac{\sigma^2}{2}\right)t+\sigma\sqrt{t}}$$

$$d = e^{\left(r-\frac{\sigma^2}{2}\right)t-\sigma\sqrt{t}}$$

Jarrow-Rudd risk neutralize their formulas by specifying $\mu = r - \frac{\sigma^2}{2}$, but are only consistent with a probability of $\frac{1}{2}$. Simply converting is not sufficient to ensure risk neutrality for a finite number of time steps[14].

[12] Chance, Don M., A Synthesis of Binomial Option Pricing Models for Lognormally Distributed Assets, p. 8
[13] http://www.goddardconsulting.ca/option-pricing-binomial-alts.html#jr
[14] Chance, Don M., A Synthesis of Binomial Option Pricing Models for Lognormally Distributed Assets, p. 10

Part II
Binomial Lattice Implementation

A. European Option Valuation and the Greeks Using BSM

In comparing various parameterizations of lattice structures, we first used MATLAB to create a simple Black Scholes model script to calculate a "base case" across initial asset prices of 80, 100, and 120 respectively for European call options, put options, and selected Greeks measures. We expect that the binomial lattice models (in particular Cox, Ross & Rubenstein and Jarrow-Rudd) will converge (will increasing time steps Δt) upon the Black Scholes estimates for call and put valuations as well as the Greeks.

The following figure displays our assumptions and results with respect the up, natural or at-the-money, and down states with respect to asset price, strike price, risk free rate, annual asset volatility, and time to maturity.

Simple Black Scholes Valuation			
INPUTS	Sd	S0	Su
Initial Stock Price (S)	80.00	100.00	120.00
Strike (X)	100.00	100.00	100.00
Risk Free (r')	0.05	0.05	0.05
Annual Volatility (sigma)	0.20	0.20	0.20
Time to Maturity (T)	1	1	1
European Call	1.8594	10.4506	26.1690
European Put	16.9824	5.5735	1.2920
Call Delta	0.2219	0.6368	0.8965
Put Delta	(0.7781)	(0.3632)	(0.1035)
Gamma	0.0186	0.0188	0.0075
Call Theta	(3.1753)	(6.4140)	(6.2303)
Put Theta	1.5809	(1.6579)	(1.4742)

B. European Option Valuations and Greeks:
Cox, Ross & Rubenstein & Jarrow-Rudd Lattice Parameterizations

Bearing in mind the assumptions from the BSM base case, we next consider the aforementioned CCR and JR parameterizations for a lattice structure of discrete time steps (5, 50, 500). We again used MATLAB to implement each lattice. As seen in the figures below, as Δt increases and approaches infinity, the estimates for option prices and Greeks produced by both CRR and JR converge upon the results of the Black-Scholes-Merton model (BSM).

Note that the CRR lattice converges upon the BSM estimates for nearly all valuations and Greeks after 500 times steps. The JR lattice behaves similarly except for estimates of Theta which often lag the BSM estimates even after 500 time steps. This is due to the varying treatment or "weighting" if you will of the Δt in the parameter calculations for [u , d, and p] in the JR lattice.

CRR Parameter Lattice				
M	5	50	500	BSM
Initial Stock Price (Sd = 80)				
European Call	1.8670	1.8303	1.8602	1.8594
European Put	16.9900	16.9532	16.9831	16.9824
Call Delta	0.1994	0.2182	0.2218	0.2219
Put Delta	(0.8006)	(0.7818)	(0.7782)	(0.7781)
Gamma	0.0177	0.0186	0.0186	0.0186
Call Theta	(2.9623)	(3.1632)	(3.1745)	(3.1753)
Put Theta	1.8418	1.5977	1.5822	1.5809
Initial Stock Price (S0 = 100)				
European Call	10.8059	10.4107	10.4466	10.4506
European Put	5.9289	5.5336	5.5695	5.5735
Call Delta	0.6239	0.6362	0.6368	0.6368
Put Delta	(0.3761)	(0.3638)	(0.3632)	(0.3632)
Gamma	0.0202	0.0191	0.0188	0.0188
Call Theta	(6.6140)	(6.4772)	(6.4202)	(6.4140)
Put Theta	(1.8100)	(1.7163)	(1.6636)	(1.6579)
Initial Stock Price (Su = 120)				
European Call	26.3534	26.1715	26.1691	26.1690
European Put	1.4763	1.2944	1.2920	1.2920
Call Delta	0.8885	0.8964	0.8965	0.8965
Put Delta	(0.1115)	(0.1036)	(0.1035)	(0.1035)
Gamma	0.0084	0.0076	0.0075	0.0075
Call Theta	(6.4684)	(6.2549)	(6.2327)	(6.2303)
Put Theta	(1.6643)	(1.4939)	(1.4761)	(1.4742)

Jarrow -Rudd Parameter Lattice				
M	5	50	500	BSM
Initial Stock Price (Sd = 80)				
European Call	2.0204	1.8720	1.8592	1.8594
European Put	17.1455	16.9951	16.9822	16.9824
Call Delta	0.2181	0.2215	0.2218	0.2219
Put Delta	(0.7819)	(0.7785)	(0.7782)	(0.7781)
Gamma	0.0185	0.0186	0.0186	0.0186
Call Theta	(2.6830)	(2.6507)	(2.6435)	(3.1753)
Put Theta	(0.2956)	(0.2915)	(0.2870)	1.5809
Initial Stock Price (S0 = 100)				
European Call	10.7557	10.4874	10.4534	10.4506
European Put	5.8813	5.6107	5.5763	5.5735
Call Delta	0.6359	0.6364	0.6368	0.6368
Put Delta	(0.3641)	(0.3636)	(0.3632)	(0.3632)
Gamma	0.0198	0.0188	0.0188	0.0188
Call Theta	(4.8052)	(4.5286)	(4.5065)	(6.4140)
Put Theta	(3.0219)	(2.7698)	(2.7501)	(1.6579)
Initial Stock Price (Su = 120)				
European Call	26.3057	26.1742	26.1682	26.1690
European Put	1.4318	1.2975	1.2912	1.2920
Call Delta	0.8965	0.8967	0.8965	0.8965
Put Delta	(0.1035)	(0.1033)	(0.1035)	(0.1035)
Gamma	0.0076	0.0075	0.0075	0.0075
Call Theta	(3.1070)	(3.0166)	(3.0046)	(6.2303)
Put Theta	(1.9278)	(1.8581)	(1.8482)	(1.4742)

Part III
Effects of Dividends and Exercise rights on Option Valuation

A. American Call vs European Call Valuation on Dividend Paying Assets

We next consider In-the-Money American and European calls placed upon a dividend paying asset across an increasing set of dividend yields. We again used MATLAB to implement an augmented CRR lattice for an American and European call.

Note that as dividend yield increases both American and European call valuations decline, albeit at different rates as evidences by the increasing premium. Said another way, European calls decline in value more rapidly that American calls with respect to increasing dividend yield. Intuitively, if the underlying asset pays a dividend, the asset price should decrease by the amount of the distribution. Consider that European options do not while American options do include an early exercise right. American call holders may exercise prior to a dividend payment, thus avoiding a decrease in profits relative to the constraint of the European call to exercise at maturity. This flexibility is manifested in the premium paid for an American call over a European call.

INPUTS		Dividend Paying Assets			
Initial Stock Price (S)	120.00	Div Yield	American Call	European Call	Premium
Strike (X)	100.00	0.060	24.9718	20.3737	4.5981
Risk Free (r')	0.05	0.065	24.2441	18.8343	5.4098
Annual Volatility (sigma)	0.20	0.070	23.5128	17.3725	6.1403
Time to Maturity (T)	5	0.075	22.7784	15.9865	6.7919
Time Steps (M)	5	0.080	22.0416	14.6745	7.3671

B. American Put vs European Put Valuation When Deep In-the-Money

Finally, we consider the value of deep In-the-Money American vs European puts on a non-dividend paying asset. We again used MATLAB to implement an augmented CRR lattice for an American and European call, in this instance without dividends. As stated earlier, the American option has the flexibility to exercise prior to maturity. Because the American put holder may exercise a deep In-the-Money put and reinvest the earnings prior to the option maturity, a premium is incorporated into the price.

INPUTS		Deep ITM Puts		
Initial Stock Price (S)	50.00	American Put	European Put	Premium
Strike (X)	100.00	50.0000	29.6766	20.3234
Risk Free (r')	0.05			
Annual Volatility (sigma)	0.20			
Time to Maturity (T)	5			
Time Steps (M)	5			

References

Chance, D.M., *A Synthesis of Binomial Option Pricing Models for Lognormally Distributed Assets*, 2008

Gilli, M., Schumann, E., *Implementing Binomial Trees*, Comisef Working Paper, 2009

http://www.goddardconsulting.ca/option-pricing-binomial-alts.html#jr

Wikipedia: http://en.wikipedia.org/wiki/Binomial_options_pricing_model

APPENDIX
MATLAB SYNTAX

1. BLACK SCHOLES OPTION VALUATION AND GREEKS

```
function [C,P,Delta,Gamma,Theta] = BlackScholes(S0,X,r,sigma,T)
%INPUTS
S0=100;
X=100;
r=0.05;
sigma=0.2;
T=1;

d1=(log(S0/X)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d2=d1-(sigma*sqrt(T));
C=S0*normcdf(d1)-X*(exp(-r*T)*normcdf(d2)) %EUROPEAN CALL
P=X*exp(-r*T)*normcdf(-d2)-S0*normcdf(-d1) %EUROPEAN PUT
[CallDelta,PutDelta]=blsdelta(S0,X,r,T,sigma) %DELTA
Gamma=blsgamma(S0,X,r,T,sigma) %GAMMA
[CallTheta,PutTheta]=blstheta(S0,X,r,T,sigma) %THETA
end
```

2. EUROPEAN CALL OPTION CRR VALUATION AND GREEKS

```
function [C0 , deltaE , gammaE , thetaE ] = EuropeanCallGreeks_CRR
(S0,X,r,T,sigma,M)

%INPUTS AND PARAMETERS
f7=1;
S0=100; %INITIAL STOCK PRICE
X=100; %STRIKE PRICE
r=0.05; %RISK FREE RATE
T=1; %TIME TO MATURITY
sigma=0.2; %ANNUAL VOLATILITY
M=1; %NUMBER OF TIME STEPS

%COMPUTE PARAMETERS
dt = T / M;
v = exp(-r * dt);
u = exp(sigma*sqrt(dt));
d = 1 /u;
p = (exp(r * dt) - d) / (u - d);

%SET ASSET PRICES AT PERIOD M (maturity)
S = zeros(M + 1,1);
S(f7+0) = S0 * d^M;
for j = 1:M
    S(f7+j) = S(f7+j - 1) * u / d;
end
```

```
%SET OPTION PRICES AT PERIOD M (maturity)
C = max(S - X, 0);

%REVERT STEPWISE THRU LATTICE
for i = M-1:-1:0
    for j = 0:i
        C(f7+j) = v * ( p * C(f7+j + 1) + (1-p) * C(f7+j));
        S(f7+j) = S(f7+j) / d;
    end
    if i==2
        %GAMMA
        gammaE = ((C(2+f7) - C(1+f7)) / (S(2+f7) - S(1+f7)) - ...
            (C(1+f7) - C(0+f7)) / (S(1+f7) - S(0+f7))) / ...
            (0.5 * (S(2+f7) - S(0+f7)));
        %THETA (aux)
        thetaE = C(1+f7);
    end
    if i==1
        %DELTA
        deltaE = (C(1+f7) - C(0+f7)) / (S(1+f7) - S(0+f7));
    end
    if i==0
        %THETA (final)
        thetaE = (thetaE - C(0+f7)) / (2 * dt);
    end

end
C0 = C(f7+0);
deltaE
gammaE
thetaE
```

3. EUROPEAN PUT OPTION CRR VALUATION AND GREEKS

```
function [P0,deltaE,gammaE,thetaE] = EuropeanPutGreeks_CRR(S0,X,r,T,sigma,M)

%INPUTS AND PARAMETERS
f7 = 1;
S0=120;
X=100;
r=0.05;
sigma=0.2;
T=1;
M=500;

%COMPUTE PARAMETERS
dt = T / M;
v = exp(-r * dt);
u = exp(sigma*sqrt(dt));
d = 1 /u;
p = (exp(r * dt) - d) / (u - d);

%SET ASSET PRICES AT PERIOD M (maturity)
```

```
S = zeros(M + 1,1);
S(f7+0) = S0 * d^M;
for j = 1:M
    S(f7+j) = S(f7+j - 1) * u / d;
end

%SET OPTION PRICES AT PERIOD M (maturity)
P = max(X-S, 0);

%REVERT STEPWISE THRU LATTICE
for i = M-1:-1:0
    for j = 0:i
        P(f7+j) = v * ( p * P(f7+j + 1) + (1-p) * P(f7+j));
        S(f7+j) = S(f7+j) / d;
    end
    if i==2
        %gamma
        gammaE = ((P(2+f7) - P(1+f7)) / (S(2+f7) - S(1+f7)) - ...
            (P(1+f7) - P(0+f7)) / (S(1+f7) - S(0+f7))) / ...
            (0.5 * (S(2+f7) - S(0+f7)));
        %theta (aux)
        thetaE = P(1+f7);
    end
    if i==1
        %delta
        deltaE = (P(1+f7) - P(0+f7)) / (S(1+f7) - S(0+f7));
    end
    if i==0
        %theta (final)
        thetaE = (thetaE - P(0+f7)) / (2 * dt);
    end
end
P0 = P(f7+0);
deltaE
gammaE
thetaE
```

4. EUROPEAN CALL OPTION JR VALUATION AND GREEKS

```
function [C0 , deltaE , gammaE , thetaE ] = EuropeanCallGreeks_JR
(S0,X,r,T,sigma,M)

%INPUTS AND PARAMETERS
f7=1;
S0=100; %INITIAL STOCK PRICE
X=100; %STRIKE PRICE
r=0.05; %RISK FREE RATE
T=1; %TIME TO MATURITY
sigma=0.2; %ANNUAL VOLATILITY
M=1; %NUMBER OF TIME STEPS

%COMPUTE PARAMETERS
dt = T / M;
```

```
v = exp (-r * dt);
u = exp((r-1/2*sigma^2)*dt+( sigma * sqrt (dt)));
d = exp((r-1/2*sigma^2)*dt-( sigma * sqrt (dt)));
p = 1/2;

%SET ASSET PRICES AT PERIOD M (maturity)
S = zeros(M + 1,1);
S(f7+0) = S0 * d^M;
for j = 1:M
    S(f7+j) = S(f7+j - 1) * u / d;
end

%SET OPTION PRICES AT PERIOD M (maturity)
C = max(S - X, 0);

%REVERT STEPWISE THRU LATTICE
for i = M-1:-1:0
    for j = 0:i
        C(f7+j) = v * ( p * C(f7+j + 1) + (1-p) * C(f7+j));
        S(f7+j) = S(f7+j) / d;
    end
    if i==2
        %GAMMA
        gammaE = ((C(2+f7) - C(1+f7)) / (S(2+f7) - S(1+f7)) - ...
            (C(1+f7) - C(0+f7)) / (S(1+f7) - S(0+f7))) / ...
            (0.5 * (S(2+f7) - S(0+f7)));
        %THETA (aux)
        thetaE = C(1+f7);
    end
    if i==1
        %DELTA
        deltaE = (C(1+f7) - C(0+f7)) / (S(1+f7) - S(0+f7));
    end
    if i==0
        %THETA (final)
        thetaE = (thetaE - C(0+f7)) / (2 * dt);
    end

end
C0 = C(f7+0);
deltaE
gammaE
thetaE
```

5. EUROPEAN PUT OPTION JR VALUATION AND GREEKS

```
function [P0,deltaE,gammaE,thetaE] = EuropeanPutGreeks_JR(S0,X,r,T,sigma,M)

%INPUTS AND PARAMETERS
f7 = 1;
S0=120;
X=100;
r=0.05;
```

```
sigma=0.2;
T=1;
M=5;

%COMPUTE PARAMETERS
dt = T / M;
v = exp (-r * dt);
u = exp((r-1/2*sigma^2)*dt+( sigma * sqrt (dt)));
d = exp((r-1/2*sigma^2)*dt-( sigma * sqrt (dt)));
p = 1/2;

%SET ASSET PRICES AT PERIOD M (maturity)
S = zeros(M + 1,1);
S(f7+0) = S0 * d^M;
for j = 1:M
    S(f7+j) = S(f7+j - 1) * u / d;
end

%SET OPTION PRICES AT PERIOD M (maturity)
P = max(X-S, 0);

%REVERT STEPWISE THRU LATTICE
for i = M-1:-1:0
    for j = 0:i
        P(f7+j) = v * ( p * P(f7+j + 1) + (1-p) * P(f7+j));
        S(f7+j) = S(f7+j) / d;
    end
    if i==2
        %GAMMA
        gammaE = ((P(2+f7) - P(1+f7)) / (S(2+f7) - S(1+f7)) - ...
            (P(1+f7) - P(0+f7)) / (S(1+f7) - S(0+f7))) / ...
            (0.5 * (S(2+f7) - S(0+f7)));
        %THETA (aux)
        thetaE = P(1+f7);
    end
    if i==1
        %DELTA
        deltaE = (P(1+f7) - P(0+f7)) / (S(1+f7) - S(0+f7));
    end
    if i==0
        %THETA (final)
        thetaE = (thetaE - P(0+f7)) / (2 * dt);
    end
end
P0 = P(f7+0);
deltaE
gammaE
thetaE
```

6. EUROPEAN CALL OPTION DIVIDENDS

```
function C0 = EuropeanCallDiv(S0,X,r,q,T,sigma,M)
```

```
%INPUTS AND PARAMETERS
f7=1;
S0=120; %INITIAL STOCK PRICE
X=100; %STRIKE PRICE
r=0.05; %RISK FREE RATE
q=0.06; %DIVIDEND YEILD
T=5; %TIME TO MATURITY
sigma=0.2; %ANNUAL VOLATILITY
M=5; %NUMBER OF TIME STEPS

%COMPUTE PARAMETERS
dt = T / M;
v = exp (-r * dt);
u = exp( sigma * sqrt (dt));
d = 1 / u;
p = ( exp ((r-q) * dt) - d) / (u - d);

%SET ASSET PRICES AT PERIOD M (maturity)
S = zeros (M + 1 ,1);
S(f7 +0) = S0 * d^M;
for j = 1:M
    S(f7+j) = S(f7+j - 1) * u / d;
end

%SET OPTION PRICES AT PERIOD M (maturity)
C = max (S - X, 0);

%REVERT STEPWISE THRU LATTICE
for i = M -1: -1:0
    for j = 0:i
        C(f7+j) = v * (p * C(f7+j + 1) + (1-p) * C(f7+j));
    end
end
C0 = C(f7 +0);
```

7. AMERICAN CALL OPTION DIVIDENDS

```
function C0 = AmericanCallDiv(S0,X,r,q,T,sigma,M)

%INPUTS AND PARAMETERS
f7 = 1;
S0=120; %INITIAL STOCK PRICE
X=100; %STRIKE PRICE
r=0.05; %RISK FREE RATE
q=0.06; %DIVIDEND YEILD
T=5; %TIME TO MATURITY
sigma=0.2; %ANNUAL VOLATILITY
M=5; %NUMBER OF TIME STEPS
```

```
%COMPUTE PARAMETERS
dt = T / M;
v = exp (-r * dt);
u = exp ( sigma * sqrt (dt));
d = 1 / u;
p = ( exp ((r-q) * dt) - d) / (u - d);

%SET ASSET PRICES AT PERIOD M (maturity)
S = zeros (M + 1 ,1);
S(f7 +0) = S0 * d^M;
for j = 1:M
S(f7+j) = S(f7+j - 1) * u / d;
end

%SET OPTION PRICES AT PERIOD M (maturity)
C = max (S - X, 0);

%REVERT STEPWISE THRU LATTICE
for i = M -1: -1:0
for j = 0:i
C(f7+j) = v * (p * C(f7+j + 1) + (1-p) * C(f7+j));
S(f7+j) = S(f7+j) / d;
C(f7+j) = max (C(f7 + j), S(f7+j)-X);
end
end
C0 = C(f7 +0);
```

8. AMERICAN PUT OPTION

```
function P0 = AmericanPut(S0,X,r,T,sigma,M)

%INPUTS AND PARAMETERS
f7 = 1;
S0=100; %INITIAL STOCK PRICE
X=100; %STRIKE PRICE
r=0.05; %RISK FREE RATE
T=5; %TIME TO MATURITY
sigma=0.2; %ANNUAL VOLATILITY
M=5; %NUMBER OF TIME STEPS

%COMPUTE PARAMETERS
dt = T / M;
v = exp (-r * dt);
u = exp ( sigma * sqrt (dt));
d = 1 / u;
p = ( exp (r * dt) - d) / (u - d);

%SET ASSET PRICES AT PERIOD M (maturity)
S = zeros (M + 1 ,1);
```

```
S(f7 +0) = S0 * d^M;
for j = 1:M
S(f7+j) = S(f7+j - 1) * u / d;
end

%SET OPTION PRICES AT PERIOD M (maturity)
P = max (X - S, 0);

%REVERT STEPWISE THRU LATTICE
for i = M -1: -1:0
for j = 0:i
P(f7+j) = v * (p * P(f7+j + 1) + (1-p) * P(f7+j));
S(f7+j) = S(f7+j) / d;
P(f7+j) = max (P(f7 + j), X-S(f7+j));
end
end
P0 = P(f7 +0)
```

9. European Put Option

```
function P0 = EuropeanPut(S0,X,r,T,sigma,M)

%INPUTS AND PARAMETERS
f7=1;
S0=100; %INITIAL STOCK PRICE
X=100; %STRIKE PRICE
r=0.05; %RISK FREE RATE
T=5; %TIME TO MATURITY
sigma=0.2; %ANNUAL VOLATILITY
M=5; %NUMBER OF TIME STEPS

%COMPUTE PARAMETERS
dt = T / M;
v = exp (-r * dt);
u = exp( sigma * sqrt (dt));
d = 1 / u;
p = ( exp (r * dt) - d) / (u - d);

%SET ASSET PRICES AT PERIOD M (maturity)
S = zeros (M + 1 ,1);
S(f7 +0) = S0 * d^M;
for j = 1:M
    S(f7+j) = S(f7+j - 1) * u / d;
end

%SET OPTION PRICES AT PERIOD M (maturity)
P = max (X - S, 0);
```

```
%REVERT STEPWISE THRU LATTICE
for i = M -1: -1:0
    for j = 0:i
        P(f7+j) = v * (p * P(f7+j + 1) + (1-p) * P(f7+j));
    end
end
P0 = P(f7 +0);
```