Options Pricing In A Binomial Model With Transaction Costs

Duy Nguyen*

February 16, 2015

Abstract

This note was developed when I read the papers [2], [5] and [7]. In [2] the authors develop a discrete time approach to options pricing with transaction costs. Their model represents an extension of binomial CRR option pricing model [3]. They also provided various examples to support the theory and methodologies. The main purpose of this note is to replicate numerical results in [2] and collect some well-known facts about options pricing with transaction costs. The note follows [2] closely.

Keywords: Option pricing, regime switching model, recombined tree, stochastic approximation, financial derivatives, transaction cost

AMS subject classifications: 91G80, 93E11, 93E20

*Department of Mathematics, Massachusetts College of Liberal Arts, 375 Church Street, North Adams, MA 01247, Email: d.nguyen@mcla.edu
1 Introduction and problem formulation

The binomial model was originally presented in 1979 by Cox, Ross, and Rubinstein [3] (henceforth the CRR model) as a simplification to the more complicated Black-Scholes model. Since the binomial tree model was introduced, due to its computational efficiency and simplicity, tree methods and various versions of it have been considered by a number of researchers and are widely used by practitioners in option markets. In fact, entire theory of derivative pricing and hedging has been developed with thin the discrete tree modeling framework. However, CRR model assumes perfect frictionless markets, in particular that there is no transaction costs associated to trades. People therefore have been trying to incorporate transaction costs in Black-Scholes model. One of the first models embodying transaction costs is the continuous time Leland model [5]. Leland reformulates Black-Scholes model by introducing of augmented volatility

\[ \sigma_L = \sigma \sqrt{1 + \frac{2\pi k}{\sigma \sqrt{h}}} \]

where \( k \in [0, 1) \) indicates transaction cost measured as a fraction of the amount traded, \( \sigma \) is the volatility of the asset, and \( h \) is the time step size. Then replacing \( \sigma \) in Black-Scholes’ formula by \( \sigma_L \), he can use Black-Scholes’ formula to give an approximation value of an European option. Building on work of [6], Boyle and Vorst employ a discrete time framework and construct the portfolio to replicate a long and short European call.

This note is organized as follow: we revisit the formulation in [2] and derive a recursive algorithm which enables us to compute European call options with transaction codes. Matlab code is provided for ease of computation. We also compare discrete method with various continuous approximation methods in the last section.

2 Problem formulation

Assume that there are two investment securities available to investors, one is the risk-less asset \( B \) and the other is the stock \( S \). Let \( T > 0 \) be the time to expiration of options under consideration. \( n > 0 \) be the number of time steps, \( h := T/n \) be the time step size, \( S \) be the initial asset price, \( K \) be the strike price, \( u := e^{\sigma \sqrt{h}} \), \( d := e^{-\sigma \sqrt{h}} \) be the upward move and downward move factors, respectively. Let \( R := e^{rh} \) be the one-period interest rate. In a 2-period binomial tree framework, the discrete evolution of the underlying asset price is given in the Figure 1 below.

![Figure 1: 2-step binomial tree](image-url)
A dynamic hedging strategy is employed to replicate the payoff to an European call option. The replicating portfolio will be constructed backward from the maturity day, i.e., if know the portfolio at the terminal nodes, we will construct the portfolio at the initial node $S$. Let $\Delta_i$ denote the number of risky assets and $B_i$ denote the number of bonds after the initial node $S$. In Boyle-Vorst they assume that the institution creating the replicating portfolio does not have to buy the initial amount of the risky asset $\Delta$. In addition, they assume that the replicating portfolio at option expiration for in-the-money call option (i.e. $Su^d^n > K$) consists of one unit of the risky asset and short position in the riskless asset equal to exercise price. That is the terminal values for $(\Delta_i, B_i) = (1, -K)$ if the terminal price $Su^d^n > K$. Figure 2 gives the $\Delta$’s and $B$’s at each node.

![Figure 2: 2-step portfolio](image)

We must select the initial portfolio $(\Delta, B)$ in such a way that $(\Delta_1, B_1)$ can be bought if the up-state $Su$ occurs and $(\Delta_2, B_2)$ can be bought if the down-state $Sd$ occurs. Hence for the strategy to be self-financing, the equations

\[
\Delta Su + BR = \Delta_1 Su + B_1 + k|\Delta - \Delta_1|Su
\]

\[
\Delta Sd + BR = \Delta_2 Sd + B_2 + k|\Delta - \Delta_2|Sd
\]

must be satisfied. Note that (1) and (2) are two non-linear equations containing two unknown variable $\Delta$ and $B$. The solution $(\Delta, B)$ might not exist. Hence we need to find conditions imposed on $u, d, k, \Delta_1, \Delta_2$ such that the solution exists and uniquely determined by given parameters. The following proposition gives sufficient conditions that guarantee the existence of the solution $(\Delta, B)$. The readers are invited to refer to [7] for its simple proof.

**Proposition 1 ([7])**

1. If $\Delta_1 < \Delta_2$ and $k(u + d)/(u - d) < 1$

2. or if $\Delta_1 \geq \Delta_2$ and $k < 1$

then equations (1) and (2) have a unique solution for $\Delta$ and $B$. Moreover we have $\Delta_2 \leq \Delta \leq \Delta_1$. 

3
3 Option pricing under transaction costs

In this section, by following the work of [2] an algorithm that enable us to compute a long European option is provided.

Now let $\bar{u} = u(1 + k)$ and $\bar{d} = d(1 - k)$. Proposition 1 reduces non-linear equations to following linear equations

$$\Delta S\bar{u} + BR = \Delta_1 S\bar{u} + B_1$$ (3)

$$\Delta S\bar{d} + BR = \Delta_2 S\bar{d} + B_2$$ (4)

Solving the system (3) and (4) we have

$$\Delta = \frac{\Delta_1 S\bar{u} + B_1 - \Delta_2 S\bar{d} - B_2}{R}$$ (5)

$$B = \frac{\Delta_1 S\bar{u} + B_1 - \Delta S\bar{u}}{R}$$

Hence the European option value at the node $S$ is given be

$$C = \Delta S + B$$ (6)

4 Numerical Example

4.1 Discrete tree approach

By combing (3), (4), and (6), we write the following Matlab program to compute the European call options with transaction cost. The code can find the option values at each node on the binomial tree. In addition, it also provides the portfolio values at each node. Using the parameters in [2] with the initial asset price $S = 100$, strike price $K = 80$, volatility $\sigma = 0.2$, time to expiry $T = 1$, upward move factor $u = e^{\sigma \sqrt{T/N}}$, downward move factor $d = e^{-\sigma \sqrt{T/N}}$, time step $N = 6$, and interest rate $r = \log(1.1)$. In Table 1, we report the stock prices created by a 6-step binomial tree. In Table 2 we record portfolio values at each node on the binomial tree, finally European call options at each node on the binomial tree is tabulated in Table 3.

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>108.5076</td>
<td>117.7389</td>
<td>127.7556</td>
<td>138.6245</td>
<td>150.4181</td>
<td>163.2150</td>
</tr>
<tr>
<td>2</td>
<td>92.1595</td>
<td>100</td>
<td>108.5076</td>
<td>117.7389</td>
<td>127.7556</td>
<td>138.6245</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>84.9337</td>
<td>92.1595</td>
<td>100</td>
<td>108.5076</td>
<td>117.7389</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>78.2744</td>
<td>84.9337</td>
<td>92.1595</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>72.1371</td>
<td>78.2744</td>
<td>84.9337</td>
<td>92.1595</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>66.4814</td>
<td>72.1371</td>
<td>78.2744</td>
<td>84.9337</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>61.2689</td>
<td>66.4814</td>
<td>72.1371</td>
<td>78.2744</td>
<td>84.9337</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Stock prices on a 6-step binomial tree
Table 2: Portfolio values on a 6-step binomial tree with transaction cost

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27.7353</td>
<td>34.7242</td>
<td>42.6640</td>
<td>51.4786</td>
<td>61.1262</td>
<td>71.6788</td>
<td>83.2150</td>
</tr>
<tr>
<td>2</td>
<td>19.2182</td>
<td>25.1802</td>
<td>32.2306</td>
<td>40.2406</td>
<td>49.0164</td>
<td>58.6245</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11.7711</td>
<td>16.4825</td>
<td>22.5017</td>
<td>29.7683</td>
<td>37.7389</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5.7348</td>
<td>8.8470</td>
<td>13.4202</td>
<td>20.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.6399</td>
<td>2.8560</td>
<td>4.9337</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: European call options on a 6-step binomial tree with transaction cost

4.2 Continuous approximation approach

As we have mentioned, Leland consider European call options with transaction costs. His key insight is to replace $\sigma$ in Black-Schole’s formula by $\sigma_L = \sigma \sqrt{1 + \frac{\sqrt{2/\pi} \cdot k}{\sigma \sqrt{h}}}$ and then European call options with proportion transaction cost can be approximated. In this formula, both $k$ and $h$ are assume to be small while keeping the ratio $\frac{k}{h}$ order one. Similarly, Boyle-Vorst in their paper [2] showed that European call options can also be approximated using Black-Schole’s formula with an adjusted volatility $\sigma_{VB} = \sigma \sqrt{1 + \frac{k}{\sigma \sqrt{h}}}$. The crucial step in Leland’s and Boyle-Vorst ’s argument (see [1]) is the implicit use of the approximation

$$ W(t + h) - W(t) = c^* \sqrt{h} \quad (7) $$

with an arbitrary constant $c^*$. Then the option value with transaction cost has the adjusted volatility

$$ \hat{\sigma} = \sigma \sqrt{1 + c^* \frac{k}{\sigma \sqrt{h}}} \quad (8) $$

Hence the “optimal” choice of $c^*$ is an interesting question related to the risk inherent in markets with transaction costs (see also Kusuoka [4]).
<table>
<thead>
<tr>
<th>Transaction cost</th>
<th>Num of revision times (N)</th>
<th>6</th>
<th>13</th>
<th>52</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 0.00%$</td>
<td></td>
<td>80</td>
<td>27.703</td>
<td>27.701</td>
<td>27.665</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>12.655</td>
<td>13.093</td>
<td>12.953</td>
</tr>
<tr>
<td></td>
<td></td>
<td>110</td>
<td>8.129</td>
<td>8.026</td>
<td>7.972</td>
</tr>
<tr>
<td></td>
<td></td>
<td>120</td>
<td>4.216</td>
<td>4.427</td>
<td>4.548</td>
</tr>
<tr>
<td>$k = 0.125%$</td>
<td></td>
<td>80</td>
<td>27.735</td>
<td>27.747</td>
<td>27.753</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90</td>
<td>19.894</td>
<td>19.842</td>
<td>19.865</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>12.770</td>
<td>13.248</td>
<td>13.256</td>
</tr>
<tr>
<td></td>
<td></td>
<td>110</td>
<td>8.254</td>
<td>8.205</td>
<td>8.324</td>
</tr>
<tr>
<td></td>
<td></td>
<td>120</td>
<td>4.329</td>
<td>4.595</td>
<td>4.882</td>
</tr>
<tr>
<td>$k = 0.5%$</td>
<td></td>
<td>80</td>
<td>27.837</td>
<td>27.894</td>
<td>28.047</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90</td>
<td>20.113</td>
<td>20.149</td>
<td>20.453</td>
</tr>
<tr>
<td></td>
<td></td>
<td>110</td>
<td>8.618</td>
<td>8.721</td>
<td>9.300</td>
</tr>
<tr>
<td></td>
<td></td>
<td>120</td>
<td>4.663</td>
<td>5.084</td>
<td>5.820</td>
</tr>
<tr>
<td>$k = 2%$</td>
<td></td>
<td>80</td>
<td>28.297</td>
<td>28.563</td>
<td>29.409</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90</td>
<td>20.983</td>
<td>21.346</td>
<td>22.643</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>14.358</td>
<td>15.333</td>
<td>16.966</td>
</tr>
<tr>
<td></td>
<td></td>
<td>110</td>
<td>9.965</td>
<td>10.555</td>
<td>12.469</td>
</tr>
<tr>
<td></td>
<td></td>
<td>120</td>
<td>5.926</td>
<td>6.859</td>
<td>8.950</td>
</tr>
</tbody>
</table>

Table 4: European call with varying transaction costs

In this section, we will consider approximating European option with binomial tree method with various choice of up/down move factors. We also compare the results obtained using tree method with continuous approximation methods. More specifically, using the parameters in [2] with the initial asset price $S = 100$, strike price $K = 100$, volatility $\sigma = 0.2$, transaction cost $k = 0.125\%$, time to expiry $T = 1$, time step $N = 52$, and interest rate $r = \log(1.1)$. Tree method with different choices for $u$ and $d$ converges very fast. It is clear that the choice for $ud = 1$ works best since the tree is recombined; hence it is more efficient.
\[ K \quad Tree \quad u = e^{\sigma \sqrt{h}} \quad d = e^{-\sigma \sqrt{h}} \quad Tree \quad u = e^{(r-0.5\sigma^2)h+\sigma \sqrt{h}} \quad d = e^{(r-0.5\sigma^2)h-\sigma \sqrt{h}} \quad Tree \quad u = e^{\sqrt{e^{\sigma h} - 1 + r h}} \quad d = e^{-\sqrt{e^{\sigma h} - 1 + r h}} \quad \text{Leland} \quad \sigma_L = \sigma \sqrt{1 + \sqrt{2/\pi} \frac{1}{\sigma \sqrt{h}}} \quad \text{BV} \quad \sigma_{BV} = \sigma \sqrt{1 + 2k \sigma \sqrt{h}} \]

<table>
<thead>
<tr>
<th></th>
<th>80</th>
<th>90</th>
<th>100</th>
<th>110</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>27.753</td>
<td>27.765</td>
<td>27.762</td>
<td>27.745</td>
<td>27.764</td>
</tr>
<tr>
<td>110</td>
<td>8.324</td>
<td>8.349</td>
<td>8.284</td>
<td>8.246</td>
<td>8.316</td>
</tr>
<tr>
<td>120</td>
<td>4.882</td>
<td>4.871</td>
<td>4.925</td>
<td>4.822</td>
<td>4.889</td>
</tr>
</tbody>
</table>

Table 5: European call option with various approximation methods

5 Concluding and remarks

In this note, we revisit the work done [2]. We provide a Matlab program that can be used to replicate the results in Boyle-Vorst’s paper. Finally, we compare the numerical results obtained by tree method and continuous approximation methods of Leland’s and Boyle-Vorst’s. It is interesting if we could extend the work for a binomial model to a trinomial model and it is worth to investigate further the “optimal” choice of \( c^* \) in \( \hat{\sigma} = \sigma \sqrt{1 + c^* \frac{k}{\sigma \sqrt{h}}} \).

---

```matlab
% This program is used to replicate the results (Call options with transaction cost) in the paper "Option Replication in Discrete Time with Transaction Costs" by Phelim P. Boyle and Ton Vorst, The Journal of Fianance, Page 271 of ...
% by Phelim P. Boyle and Ton Vorst, The Journal of Fianance, Page 271 of ...
% Code writen by : Duy Nguyen,
% Department of Mathematics
% Massachusset College of Liberal Art
% email: nducduy@gmail.com

%%%%%%%%%% Problem and method parameters %%%%%%%%%%%
S = 100; % asset price
K=80; % strike price
sigma=0.20; % volatility
T=1; % time to expiration
r=log(1.1); % interest rate
N=6; % number of period
dt=T/N; % time step
u=exp(sigma*sqrt(dt)); % up move factor
d=exp(-sigma*sqrt(dt)); % down move factor
kappa=0.125/100; % transaction cost
ubar=u*(1+kappa);
dbar=d*(1-kappa);
%d when kappa=0 we have the CRR model
ubar=u*(1+kappa);
dbar=d*(1-kappa);
```
N=N+1;
P=zeros(N,N);

discountFactor=exp(-r*dt); %discount factor
P(1,1)=S;%initial price

% P is a upper triangle matrix that contains all possible stock prices from
% the binomial tree
for j=2:1:N
    for i=1:j
        P(i,j)=S*u^(j-i)*d^(i-1);
    end
end;

v=zeros(N,N);% contains all option values from the binomial tree
%we only care about the value at (1,1)
portFolio=zeros(2,1,N); % the 2-d matrix that contains the portfolio at
%the terminal time
%v is a upper triangle matrix
for i=1:1:N
    diff=P(i,N)-K;
    v(i,N)=max(diff,0);
    if diff>0
        portFolio(1,1,i)=1;% contain number of risky assets
        portFolio(2,1,i)=-1; % contains number of bonds
    end
end

for j=N−1:-1:1
    pFolio=zeros(2,1,j);
    for i=1:1:j
        %Compute the portfolio on each node after the terminal nodes
        %pFolio(1,1,i) : contains the number of risky assets
        %pFolio(2,1,i) : contains the number of bonds
        pFolio(1,1,i)=(portFolio(1,1,i)*P(i,j+1)*ubar/u+portFolio(2,1,i)*K−...
                      portFolio(1,1,i+1)*P(i,j+1)*dbar/u−portFolio(2,1,i+1)*K)/...
                      (P(i,j+1)*ubar/u−P(i,j+1)*dbar/u);
        pFolio(2,1,i)=(portFolio(1,1,i)*P(i,j+1)*ubar/u+...
                      portFolio(2,1,i)*K−pFolio(1,1,i)*P(i,j+1)*ubar/u)*discountFactor ... 
                      ;
        %v(i,j) is the European call option at the node (i,j)
        v(i,j)=pFolio(1,1,i)*P(i,j+1)/u+pFolio(2,1,i);
    end
portFolio=pFolio;
K=1;
end
v(1,1)
References


