Bermudan Option Pricing using Binomial Models
Seminar in Analytical Finance I

Jessica Radeschnig, Hossein Nohrouzian, Amir Kazempour Esmati

HT 2012

Contents

Abstract

This paper aims at giving an overview of the binomial option pricing model. Further, the model is used to find the fair price for three different financial derivatives,namely, American, European and Bermudan. Moreover, binomial option pricing is implemented in MATLAB. The examples provided, will show that the price of the Bermudan option, lies between the one of American and European.

1 Introduction

Binomial option pricing is a simple but powerful technique that can be used to solve many complex option-pricing problems. In contrast to the Black-Scholes and other complex option-pricing models that require solutions to stochastic differential equations, the binomial option-pricing model (two state option-pricing model) is mathematically simple. It is based on the assumption of no arbitrage. (Conroy 2003) First, the assumptions and characteristics of the binomial model are explained. Second, the model is utilized to price an option. At last, a MATLAB application is proposed to perform the calculations and give the prices for three different types of options. The MATLAB codes are provided as an appendix to this paper.

2 Probability Theory

Since the future price of a financial security are unknown at time $t = 0$, the price represent a *Stochastic Variable*. With the use of *Probability Theory*, one is able to estimate the expected future prices. Once having these prices, one can price different derivatives with the financial security as the underlying asset.

2.1 Random Walks $\in \mathbb{Z}$

If the random variables, $X_i, \quad i \in \{1, 2, ..., n\}$ are independent and identically distributed(IID), a partial sum process is defined through

$$W_n = \begin{cases} 0, & n = 0 \\ X_1 + X_2 + \cdots + X_n, & n \geq 1. \end{cases}$$

The random walk in $\mathbb{Z}$ space is a walk that moves along a line, either up (u) or down (d) and is defined to be a partial sum process, where $P\{X_1 = \omega\}$,where $\omega \in \{u, d\}$, denotes the probability distribution of X_1 at time t=1. The possible outcomes for the first three steps are illustrated in Table 1, where possible values of W_i is denoted by ω_i.(Kijima 2002, p.95)

$P\{\emptyset\} = 0 \quad P\{\Omega\{\omega_i\}\} = 1$	
$F_1 = \{\emptyset, \Omega\{\omega_1, \omega_2\}\}$	$n(\Omega) = 2^i = 2 \implies \omega_i \in$\{u, d\}
$F_2 = \{\emptyset, \Omega\{\omega_1, ..., \omega_4\}\}$	$n(\Omega) = 2^i = 4 \implies \omega_i \in$\{uu, ud, dd, du \}
$F_3 = \{\emptyset, \Omega\{\omega_1, ..., \omega_8\}\}$	$n(\Omega) = 3^i = 8 \implies \omega_i \in$\{uuu, uud, udu, udd, ddd, ddu, duu, dud \}

Table 1: **Random Walks - Information at Time t** The table is a summery of the known information at the probability space for the first t=3 time-periods.

Section 2.2 will further describe how the probability is distributed over these values of $W_n \in \Omega$.

2.2 Binomial Probability Distribution

The probability $P(W_n = \omega_i)$, is defined to be the sum of probabilities of all the sample points $\in \Omega$ that are assigned the value $\omega_i = x$. Since the $\mathbb{Z}$ space has only two directions, it is consistent with the binomial probability distribution, hence, the probability function of W_n is $P\{W_n = x\} = \begin{pmatrix} n \\ x \end{pmatrix} p^x(1-p)^{n-x}$

Probability Distribution Function

$$P\{W_n = x\} = \begin{pmatrix} n \\ x \end{pmatrix} p^x(1-p)^{n-x} \tag{1}$$

where,

$$\begin{pmatrix} n \\ k \end{pmatrix} = \frac{n!}{k!(n-k)!} \tag{2}$$

is the *Binomial Coefficient*. This coefficient can either be calculated through (2) or read from a table such as *Pascal's Triangle*. This triangle is illustrated in Figure 1. From now on, $P(W_n = \omega_i)$ will

n	$\binom{n}{k}$															
0	1															
1	1	1														
2	1	2	1													
3	1	3	3	1												
4	1	4	6	4	1											
5	1	5	10	10	5	1										
6	1	6	15	20	15	6	1									
7	1	7	21	35	35	21	7	1								
8	1	8	28	56	70	56	28	8	1							
9	1	9	36	84	126	126	84	36	9	1						
10	1	10	45	120	210	252	210	120	45	10	1					
11	1	11	55	165	330	462	462	330	165	55	11	1				
12	1	12	66	220	495	792	924	792	495	220	66	12	1			
13	1	13	78	286	715	1287	1716	1716	1287	715	286	78	13	1		
14	1	14	91	364	1001	2002	3003	3432	3003	2002	1001	364	91	4	1	
15	1	15	105	455	1365	3003	5005	6435	6435	5005	3003	1365	455	105	15	1

Figure 1: **Pascal's Triangle** The table gives the corresponding binomial coefficient for (the first 15) different values of n.

be denoted as $P(x)$.(Wackerly, Mendenhall, and Scheaffer 2007)

Further, let X be a discrete random variable, then the expected value of X is

$$E[W_n] = \sum_{i}^{n} X_i P(X_i) \tag{3}$$

The following properties will also be consistent with the binomial probability distribution:

Mean:

$$\mu = np \tag{4}$$

Variance:

$$\sigma^2 = np(1-p) \tag{5}$$

Moment Generating Function:

$$m(t) = [pe^t + (1-p)]^n \tag{6}$$

2.3 The Binomial Tree

The information from section 2.1 - 2.2 is now used in order to construct the *Binomial Tree*. Let $W_n = S(t)$ be the spot price of the financial security at $t = 0$ with the initial value, $S(0) = S_0$. Let's begin by assuming that the stock price follows a multiplicative binomial process over discrete periods, the price can have two possible values at time t: uS_{t-1} with probability p, or dS_{t-1} with probability $1-p$. Evolution of such process can be seen in Figure 2. The probability that the security price be equal to the value of a specific node in the binomial tree at time $t = n$, can be calculated by equation (2).

Since, an up move, followed by a down move and a down move followed by an up move have exactly the same effect on the price, the binomial tree, as characterized above, is called a recombining tree. The recombining property, assures that expanding the tree with one more step, will results in one extra node in the final values.

Figure 2 illustrates all possible paths the random walk can take and also the equations denoting the value of the underlying stock at each node, all from the information known at time 0 alone.

Example 1: Constructing the Binomial Tree Let the *Current Stock Price*, $S_0 = \$40$, the *Default-Free Interest-Rate*, $r = 0.12$, and $u = 1.1, d = 0.1$. Using the values given in Figure 2, Figure 3 represents the Binomial Tree of this example.

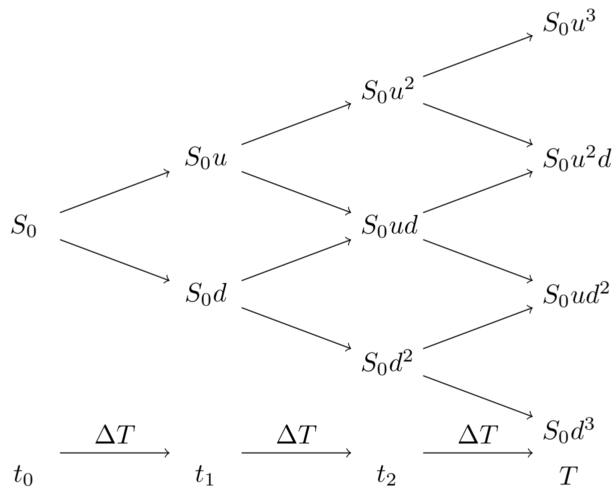


Figure 2: **The Binomial Tree** The possible paths of the random walk, each node having the probability found by (1).

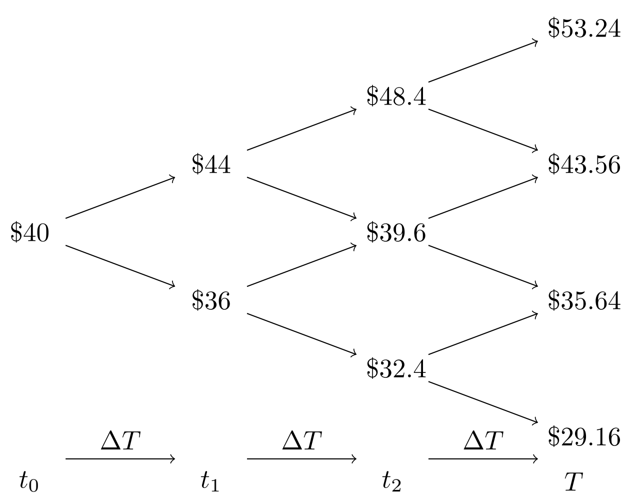


Figure 3: **The Binomial Tree** The stock price at time t, $S_0 = \$40$, $u = 1.1$ and $d = 0.9$.

3 Option Pricing

An option is a financial derivative that gives the holder the right - but not the obligation, to exercise at - or - before maturity at a specified price. A *Put-Option* gives the holder the right to sell the underlying asset while a *Call-Option* gives the right to buy it, and each agreement usually involves the trade of 100 shares of the underlying security. The holder of a contract is said to have a *Long Position* if she is the buyer of the option and a *Short Position* if she is the seller. In other words, the seller in an agreement has the short position while the buyer has the long position. There exist many different kinds of options, some examples would be European, American, Asian, Bermudan, and Currency - Options.

3.1 Fair price of financial derivative

The basic idea behind fair pricing is that the price of an option is martingale, meaning that the price today is equal the expected price of tomorrow. If all options are priced according to this theory, there are no arbitrage opportunities; there are no free lunch.

The Pay-Off Depending on if the option is a put or a call an the position held is long or short, there are different functions giving the pay-off for each participant:

Short Call:

$$-max\{S_T - K, 0\} \tag{7}$$

Long Call:

$$max\{S_T - K, 0\} \tag{8}$$

Short Put:

$$-max\{K - S_T, 0\} \tag{9}$$

Long Put:

$$max\{K - S_T, 0\} \tag{10}$$

Where S_T represent the price of the underlying at maturity, and K is the *Strike Price* (the price specified in the contract). These functions are summarized in Table 2.

Matching Volatility with u and d In practice, the volatility of a financial security can be estimated by the historical market data. Thus, one should calculate the u and d which are matched to such volatility. The corresponding values are calculated by (Cox, Ross, and Rubinstein 1979) and are given by

$$u = e^{\sigma\sqrt{\Delta t}} \quad d = e^{-\sigma\sqrt{\Delta t}}. \tag{11}$$

Risk-Neutral Probability According to the *risk-neutral valuation* principle, one can with complete impunity assume the world is risk neutral when pricing options. In a risk neutral world all individuals are indifferent to risk, and the expected return on all securities is the risk free interest rate. Given the up and down factor by eq.(11) the risk neutral probability is given by (ibid.), and is given by

$$p^* = \frac{e^{r\Delta T} - d}{u - d}, \tag{12}$$

where r is the risk-free interest rate.

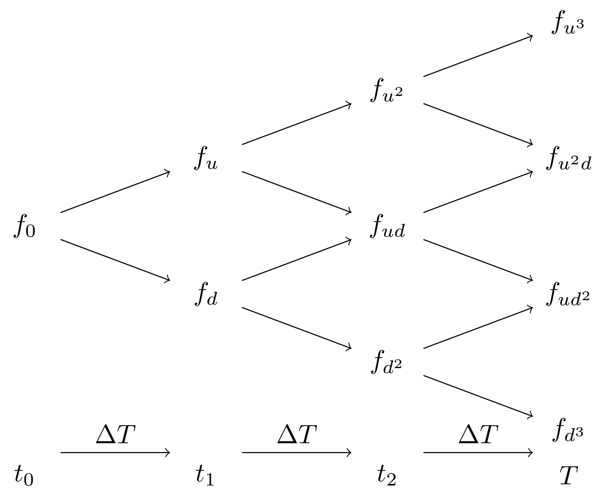


Figure 4: **The Binomial Tree-Option Value**

Pricing the option The option price at each node is given by

$$f(t) = e^{-r\Delta T}[p^* f_u(t+1) + \Big(1 - p^*\Big) f_d(t+1)], \qquad 0 \le t \le T-1. \tag{13}$$

Equation (13) is simply the expected pay-off under the risk neutral probability measure, discounted at the risk-free interest rate. Analogously, one can trace back through the nodes in the tree, which gives the option's price at time 0.(Hull 2009)

Because it is the holder of the long position that has the option to exercise or not, the fair value of the option at each node is given by (8) and (13), $t = T$ for Call- and Put- options respectively.

The remaining nodes are priced through (13),

where f represents the pay-off for the option in question. The use of this formula is through finding the latest values first, and then work backwards until the initial time 0.

4 Three Different Types of Options

In this section, European, American and Bermudan - Options will be described one by one. Every option-class will also be priced in an example; by the methods of Section 3, in order to demonstrate the differences between them.

4.1 European Options

A European option, is an option that may only be exercised on expiration date. Therefore, pricing such an option, using the binomial model, requires only the pay-off values of the last period's nodes in the binomial tree.

Example 2: European Option Pricing Let the stock in Example 1 be the underlying asset, while the *Strike Price*, $K = \$42$. Since $\Delta T = 3/12$, using (12), the risk-neutral probability is found to be ≈ 0.65. The values of f_T are found through (10) and all $f_t \neq F_T$ is found using (13)[1]

[1]Since the option is European with only possible exercise at maturity, the price of the option could be calculated directly from $f_0 = e^{-3r\Delta T}[p^{*3} f_{u^3} + 3p^{*2}(1-p^*) f_{u^2 d} + 3p^*(1-p^*)^2 f_{ud^2} + (1-p^*)^3 f_{d^3}]$ (in this example) which would make the calculation more simple. But for demonstration purposes, we stick to the recursive process through the binomial tree. The pay-off at each node for a European put-option are illustrated in Figure 4. The result is that the investor taking the long position pays \$1.8687 at time 0.

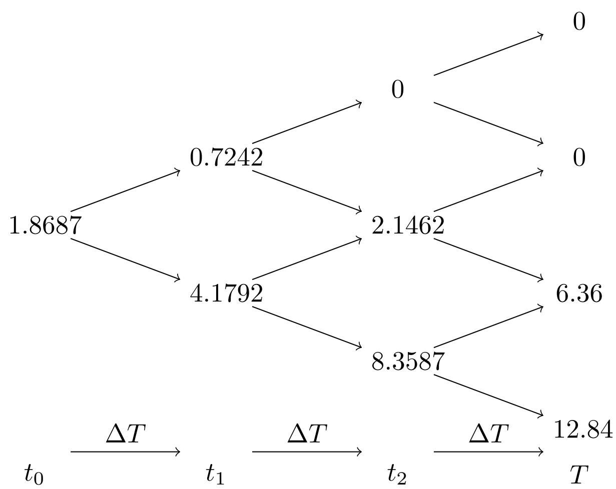


Figure 5: **European Put-Option** With $S_0 = 40$, $u = 1.1$, $d = 0.9$, $K = 42$, and $T = 3$, the holder of the long position pays \$1.8687 at time 0, to have the option to exercise after 3 months.

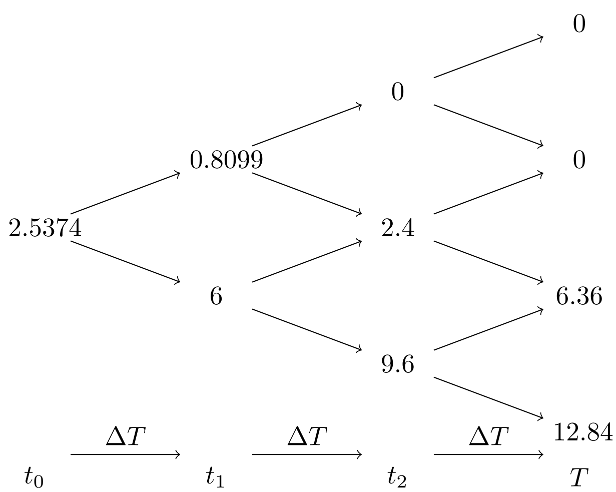


Figure 6: **American Put-Option** With $S_0 = 40$, $u = 1.1$, $d = 0.9$, $K = 42$, and $T = 3$, the holder of the long position pays \$2.5374 at time 0, to have the option to exercise after 1, 2, or 3 months.

4.2 American Options

The American option, is an option which gives the holder the right to exercise at any time on or before the maturity date. In other words, the option can be exercised at every node in the tree, thus it should be decided at each time step, if the holder wants to exercise the option or keep it until the next period. The decision is made by comparing the expected pay-off of the two successive nodes and the pay-off to an early exercise.[2]

Example 3: American Option Pricing With the same settings as for Example 2 and 3, an American put-option will be priced. The results are displayed in Figure 6.

4.3 Bermudan Options

Bermudan options take an intermediate place between American and European options. In American options exercise is permitted at any time, while, a Bermudan option has finite set dates at which the option can be exercised,e.g., annually, quarterly, or monthly. (Schweizer 2012)

[2]For instance, the value of the option at node, dd, is determined by $f_{d^2} = max\{k - S_t, e^{-r\Delta T}[p^* f_{ud^2} + (1-p^*)f_{d^3}]\}$.

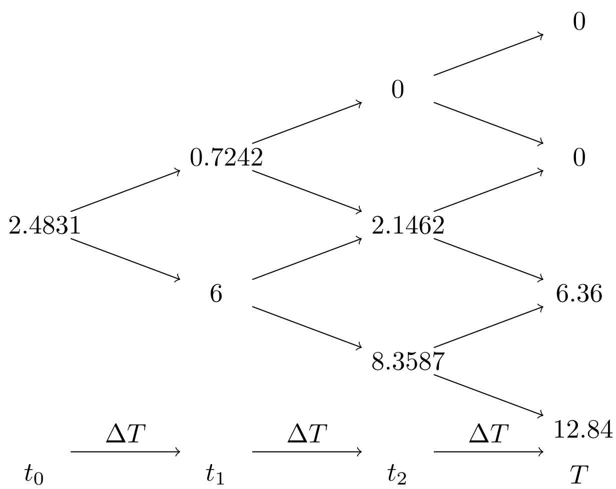
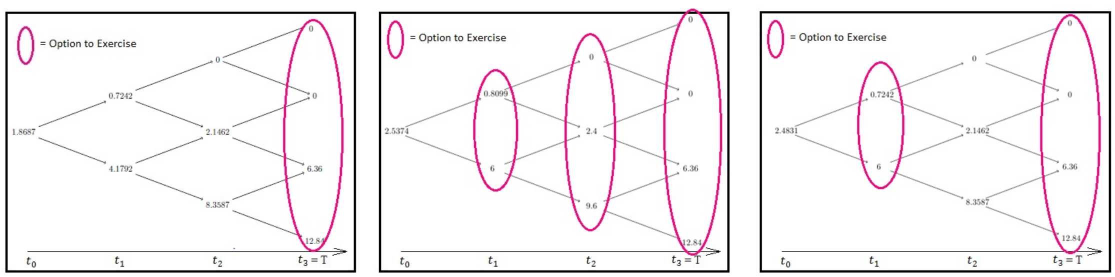


Figure 7: **Bermudan Put-Option** With $S_0 = 40$, $u = 1.1$, $d = 0.9$, $K = 42$, and $T = 3$, the holder of the long position pays \$2.4831 at time 0, to have the option to exercise after 1 or 3 months.



(a) European Put

(b) American Put

(c) Bermudan Put

Figure 8: **Bermudan Vs. European** & **American Options**

Example 4: Bermudan Option Pricing With the same settings as previous examples, an Bermudan put-option will now be priced[3]. The results are displayed in Figure 7.

5 MATLAB Implementation

Purpose The aim here is to develop an application, which calculates the fair price for three types of financial derivatives considered in Section 4. The application, consists of a function, which returns the prices and a graphical user interface that read the option parameters from the user input.

Method The binCalculator function takes in the parameters of the option(e.g., Stock price, Strike price). Then, it creates the binomial tree, using the risk neutral probability and the up and down factor proposed by the CRR model. Using the binomial tree, American, European and Bermudan pay-off trees are created.

There is one condition applied to the exercise frequency per year for the Bermudan option, it should be chosen in a way that makes the $NumberofSteps/(T \times ExerciseFrequency)$ an integer. This makes sure that is possible to exercise the option at the chosen nodes. Otherwise, theoretically, the Bermudan option cannot be exercised in any of the intermediate nodes.

The focus in this application was not on the efficiency of the codes, therefore, the pricing process will consume a significant amount of system resources. This is partly due to the purpose that we wanted to store all the nodes in the binomial tree, which can be used later to understand the mechanisms of the pricing the model.

[3] As with the European option, there is a short-cut formula for finding the value between the non-exercisable nodes. In this particular example, it would not make the calculations easier thus.

Further, the GUI will read the parameters from the text-boxes and calculate the values, by calling the function and shows them on the screen.

Figure (9) illustrates the GUI along with the prices calculated for a put option.

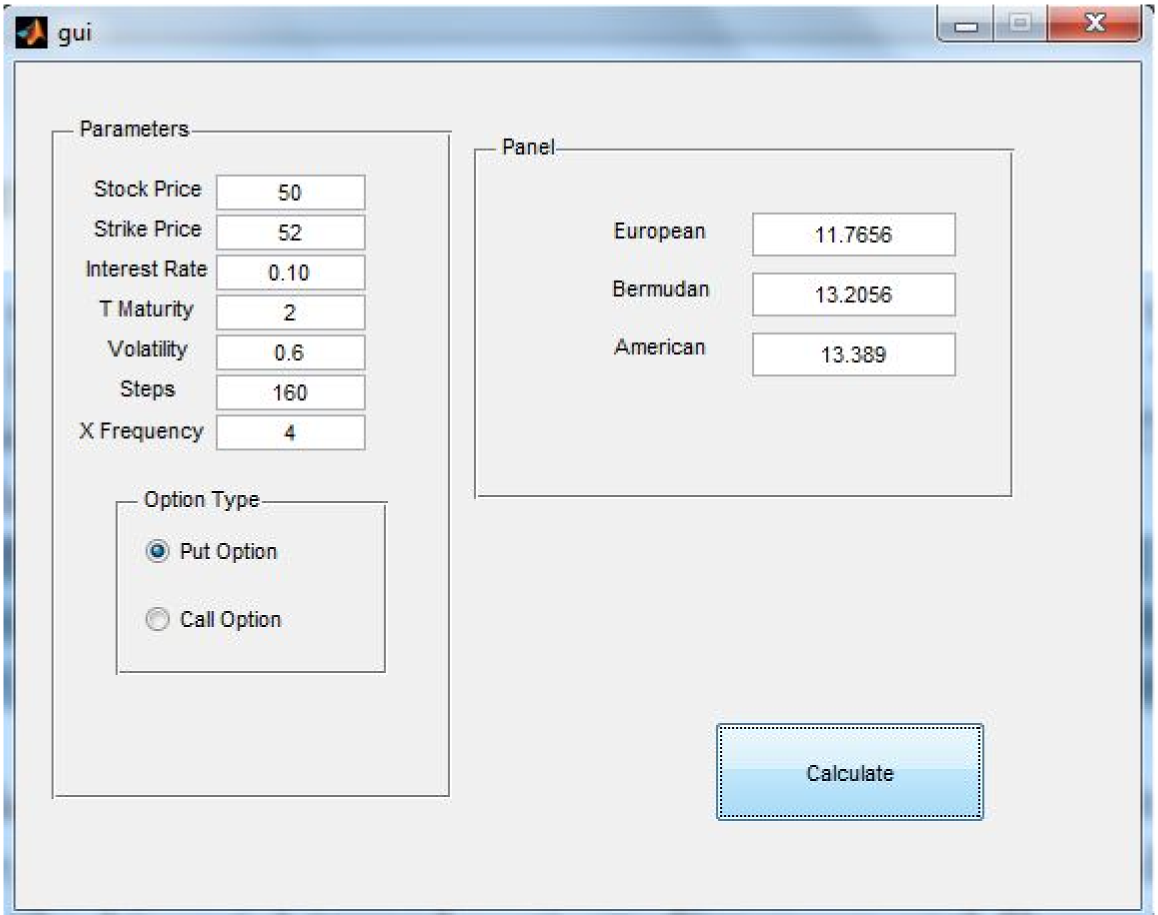


Figure 9: **Option-Pricing with MATLAB** $S_0 = 50$,$\sigma = 0.6$, $K = 52$, and $T = 2$, with a 160 steps binomial tree, and quarterly exercise dates for the Bermudan option

6 Conclusion

In this paper, we studied how a financial derivative can be priced using the binomial trees. The binomial model provides a generalizable numerical method for the valuation of options. Since, the model is based on few general assumption, it can easily be used to price different options such as, American, European, and Bermudan. As it was expected the price of the Bermudan option lied between the ones of European and American. Therefore, one can claim that the more exercise dates, cet.par., means a higher price for the option. Further we have developed an application in MATLAB, that can numerically confirm the above result.

References

[1] Robert M. Conroy. *Binomial option pricing*. University of Virginia. Aug. 2003. URL: `http://faculty.darden.virginia.edu/conroyb/derivatives/binomial%20option%20pricing%20_f-0943_.pdf`.

[2] John C. Cox, Stephen A. Ross, and Mark Rubinstein. "Option pricing: A simplified approach". In: *Journal of Financial Economics* 7.3 (1979), pp. 229–263. ISSN: 0304-405X. DOI: 10.1016/0304-405X(79)90015-1. URL: http://www.sciencedirect.com/science/article/pii/0304405X79900151.

[3] John Hull. "Options, futures, and other derivatives". In: *Options, futures, and other derivatives.* 2009.

[4] Masaaki Kijima. *Stochastic Processes with Applications to Finance.* Chapman and Hall/CRC, 2002. ISBN: 1584882247. URL: http://www.amazon.com/Stochastic-Processes-Applications-Finance-Masaaki/dp/1584882247%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D1584882247.

[5] Martin Schweizer. *On Bermudan Options.* ETH Zurich. Aug. 2012. URL: http://www.math.ethz.ch/~mschweiz/Files/bermuda_final.pdf.

[6] Dennis Wackerly, William Mendenhall, and Richard L. Scheaffer. *Mathematical Statistics with Applications.* Duxbury Press, 2007. ISBN: 0495110817. URL: http://www.amazon.com/Mathematical-Statistics-Applications-Dennis-Wackerly/dp/0495110817%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0495110817.

A Appendix I: MATLAB Codes

```
function [euPrice, amPrice, brPrice] =
binCalculator(S,K,r,sigma,T,N,exercise_Frequency,put_True)
% Course: MMA707 Analytical Finance I
%Seminar in Analytical Finance I
%Group: Jessica Radeschnic, Hossein Nohrouzian,
                  Amir KazempourEsmati

                %%%%%%%%%%%% binCalculator%%%%%%%%%%%%%
%binCalculator implements the CRR binomial tree to calculate the fair price
%% for American,European, and Bermudan options. Please refer to the seminar
% report for the calculation details. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% Parameters Guide%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% S: Spot price, e.g., 50.
% K: Strike pricem, e.g., 50.
% r: Risk-free interest rate e.g., 0.1 for 10%.
% sigma: Volatility, e.g., 0.3 for 30%.
% T: Years to maturity, e.g., 1 for 1 year.
% N: Number of steps in the binomial tree.
% Exercise_Frequency: Number of times that the Bermudan option can be
% exercised in a year, e.g., 12 for the monthly exercise.
% put_True: 1 for put option, 0 for a call.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%Author: Amir Kazempour, akr10001@student.mdh.se%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dt= T/N;
d = exp(-sigma*sqrt(dt)); u = exp(sigma*sqrt(dt));
p = (exp(r*dt)-d)/(u-d);
discount_Factor = exp(-r*dt);
step_Length = T/N;
exercise_True= zeros(1,N); %This is used later,
%to check if the exercise is possible at a certain node(in Bermudan tree)
mCoeff = N / (T*exercise_Frequency);

for i = mCoeff :mCoeff: N
    exercise_True(i) = 1;
end

%Creating the binomial tree from the last node to the first node
for j = N: -1:1
    for i = j+1: -1 :1
        binTree(i,j) = S * (u ^ (j-2*(i-1)));

    end
end

%binTreeEE calculates the payoff, taking into consideration
% only the spot prices at each node.
if put_True
    for j = N: -1:1
        for i = j+1: -1 :1
            binTreeEE(i,j) = max(0,K- binTree(i,j,1) );
        end
```

```
    end
else
    for j = N: -1:1
        for i = j+1: -1 :1
            binTreeEE(i,j) = max(0,binTree(i,j,1)- K );
        end
    end
end


%binTreeNE gives the pay-offs in case that the option can not be exercised
%until the last node.
binTreeNE(:,N) = binTreeEE(:,N);
for j = N-1: -1:1
    for i = j+1: -1 :1
        binTreeNE(i,j) = discount_Factor * ( binTreeNE(i,j+1)*p +
          binTreeNE(i+1,j+1)*(1-p) );
    end
end

binTreeAm(:,N) = binTreeEE(:,N);
%Code for the American tree
for j = N-1: -1:1
    for i = j+1: -1 :1
        binTreeAm(i,j) = max (binTreeEE(i,j),
                discount_Factor * ( binTreeAm(i,j+1)*p +  binTreeAm(i+1,j+1)*(1-p)));
     end
end

%Code for the Bermudan option,
binTreeBr(:,N) = binTreeEE(:,N);
for j = N-1: -1:1
    for i = j+1: -1 :1
        if exercise_True(j)
            binTreeBr(i,j) = max (binTreeEE(i,j),
            discount_Factor * ( binTreeBr(i,j+1)*p +  binTreeBr(i+1,j+1)*(1-p)));
        else
            binTreeBr(i,j) = discount_Factor *
            ( binTreeBr(i,j+1)*p +  binTreeBr(i+1,j+1)*(1-p));
        end
     end
end

%Options prices are stored in the cell(4,1) of the respective tree.
euPrice = discount_Factor * ( binTreeNE(1,1)*p +  binTreeNE(2,1)*(1-p) );

amPrice = discount_Factor * ( binTreeAm(1,1)*p +  binTreeAm(2,1)*(1-p) );

brPrice = discount_Factor * ( binTreeBr(1,1)*p +  binTreeBr(2,1)*(1-p) );
%Exporting all the prices to the Prices vector to be used for illustration.
% Prices = [binTreeNE(4,1); binTreeBr(4,1);binTreeAm(4,1)];
%      bar3(Prices);
end
```