CARNEGIE MELLON UNIVERSITY

SUMMER UNDERGRADUATE APPLIED MATHEMATICS INSTITUTE

Calibrating a Trinomial Maple Model to Compute the Current Values of Call Options

Authors:
Rebecca Spencer-Strong
Alexandra Van Neste

Faculty Advisor:
Dr. David Handron

July 21, 2014

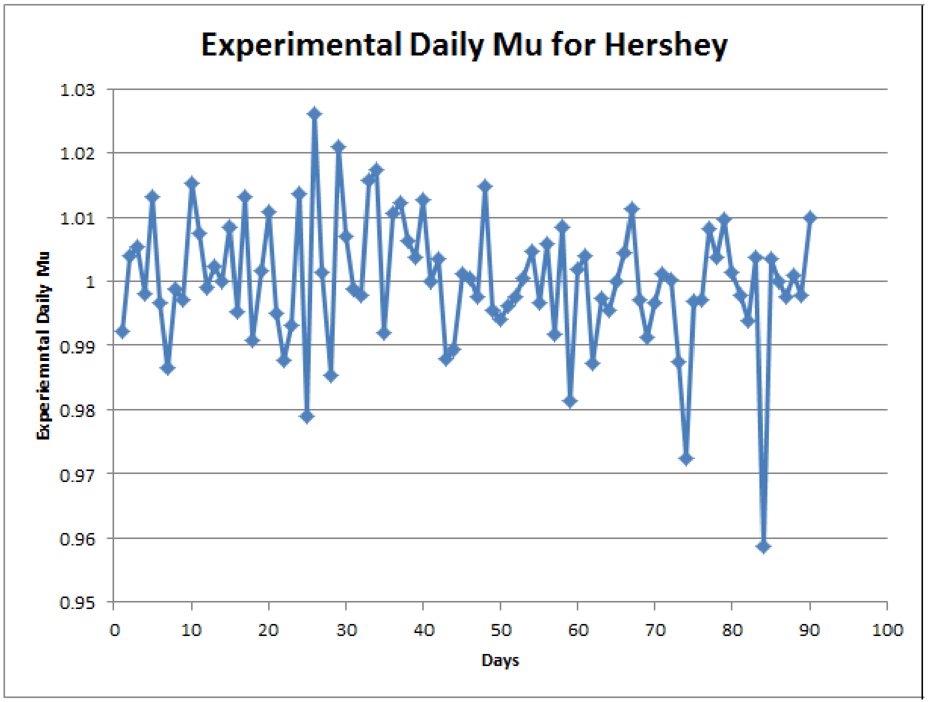
Introduction

For this project, we looked into a trinomial model that would compute time-zero call prices based on real life data. Our approach involved using a Maple program. This program was a set of procedures that allowed us to input lists of historical stock prices and current strike prices from Yahoo Finance, a maturity date, an interest rate, and the current stock price. The output of the program was a list of call option prices corresponding to the inputted strike prices. The program began by computing a variance for the list of stock prices based on the expiration date. Once it had this value for the variance of the growth rate, the program computed future stock price values to build a trinomial tree and then performed backwards induction on that tree. The backwards induction process calculated the call prices at time N in the tree and used those values to determine the time-zero call price of each call option. Our trinomial model gave us some flexibility because our risk-neutral probabilities had a free parameter. Having this free parameter allowed us to use the optimal risk-neutral values that resulted in the calculated call prices closest to the quoted call price values. The goal was to be able to calibrate our model so that it would fit market data well and potentially be useful in finding mispriced options. Maple allowed us to make the computations fast and relatively easy to adjust.

Initial Considerations: One Step Trinomial Model

Our initial plan for the model was to use the growth rate, μ, and the variance, σ^2, to calculate the risk-neutral probabilities. Our assumptions were that the observed value of μ was $\mathbb{E}(1+\rho)$ and the observed value of σ^2 was $Var^{\mathbb{E}}(1+\rho)$. In this model, the values of u, m, and d were $1+\rho(\omega_1), 1+\rho(\omega_2), 1+\rho(\omega_3)$ respectively. We set $\rho(\omega_2)$ to be 0 so that the value of our m factor would be 1. The observed μ was calculated using Excel, originally by finding the rate of return each day over a period of 90 days and then computing the average of those values. The observed variance σ^2 was computed using the var.s function of Excel. Using the observed rate of return and the observed variance, we calculated the value of ρ which we used to determine the values for u, m, and d.

We ended up having several issues with this approach. Estimating μ was difficult because the rate of return was relatively volatile for each of our three example stocks. The following example was the graph showing the volatility of the observed rate of return over a 90 day period for Hershey.



Another issue of potential concern was whether our model would be arbitrage-free. The following were the options for the value of a portfolio at time 1 that started with a stockprice of S_0 and a portfolio value $X_0 = 0$.

$$X_1(\omega) = \begin{cases} \alpha(u - (1+r))S_0 \\ \alpha(m - (1+r))S_0 \\ \alpha(d - (1+r))S_0 \end{cases}$$

In order for the model to have been arbitrage-free, unless all three of the cases were equal to zero, at least one of the cases must have been positive and at least one of the cases must have been negative. Since u was the ”up factor” and d was the ”down factor”, we said that the first case was strictly positive and the third case was strictly negative.

$$X_1(\omega) = \begin{cases} \alpha(u - (1+r))S_0 > 0 \\ \alpha(m - (1+r))S_0 \\ \alpha(d - (1+r))S_0 < 0 \end{cases}$$

Since the value of m did not affect the arbitrage conditions of the model, we had the same arbitrage-free conditions as the binomial model with an additional, flexible factor.

$$u > 1 + r > m > d > 0$$
$$u > m > 1 + r > d > 0$$

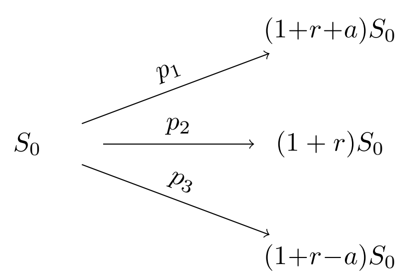
These cases showed that we had the same arbitrage-free conditions as the binomial model.

Our most significant problem was that our original solutions for u and d were illogical. We had two equations: $3\mu - 1 = u + d$ and $3\sigma^2 + 3\mu^2 - 1 = u^2 + d^2$. We knew that the value of σ^2 would be around 0 and that the value of μ would be around 1 which would give us a model that intersects only at the point (1,1). The model would not be arbitrage-free if u and d had the same values. This was an ambiguous model because it was not guaranteed that the circle and the line would ever intersect based on our observed values of μ and σ^2. We observed examples in which the circle and the line did not meet which supports the idea that our model would not always have solutions.

New plan

For this reason, we decided to adjust our model conveniently so that $m = (1 + r)$ and u, and d were symmetric around that value as $u = (1 + r) + a$ and $d = (1 + r) - a$. We solved for μ which is shown below and determined that it was only dependent on the values of the interest rate, r, and the number of steps in the model, N. We then derived a formula for a using the variance formula which is also shown below.

Multistep Models



Solving for μ

$$\mu = \mathbb{E}[\frac{S_N - S_0}{S_0}] + 1$$
$$\mu = (\mathbb{E}[\frac{S_N}{S_{N-1}}] * ... * \mathbb{E}[\frac{S_1}{S_0}] - 1) + 1$$
$$\mu = (\frac{1}{3}(\frac{uS_{N-1}}{S_{N-1}}) + \frac{1}{3}(\frac{mS_{N-1}}{S_{N-1}}) + \frac{1}{3}(\frac{dS_{N-1}}{S_{N-1}}))^N$$
$$\mu = (\frac{1}{3}(u + m + d))^N$$
$$\mu = (\frac{1}{3}((1 + r + a) + (1 + r) + (1 + r - a)))^N$$
$$\mu = (\frac{1}{3}(3(1 + r)))^N$$
$$\mu = (1 + r)^N$$

Solving for a using the variance formula

$$u = 1 + r + a$$
$$m = 1 + r$$
$$d = 1 + r - a$$

$$Var(\frac{S_N - S_0}{S_0}) = \mathbb{E}[(\frac{S_N - S_0}{S_0})^2] - (\mathbb{E}[\frac{S_N - S_0}{S_0}])^2$$

Breaking it down, the first part is equivalent to the following:

$$\mathbb{E}[(\frac{S_N - S_0}{S_0})^2] = \mathbb{E}[\frac{(S_N)^2}{(S_0)^2}] - 2\mathbb{E}[\frac{S_N}{S_0}] + 1$$
$$\mathbb{E}[(\frac{S_N - S_0}{S_0})^2] = (\mathbb{E}[\frac{(S_N)^2}{(S_{N-1})^2}] * ... * \mathbb{E}[\frac{(S_1)^2}{(S_0)^2}]) - 2(1 + r)^N + 1$$
$$\mathbb{E}[(\frac{S_N - S_0}{S_0})^2] = (\frac{1}{3}(u^2 + m^2 + d^2))^N - 2(1 + r)^N + 1$$
$$\mathbb{E}[(\frac{S_N - S_0}{S_0})^2] = (\frac{1}{3}((1 + r + a)^2 + (1 + r)^2 + (1 + r - a)^2))^N - 2(1 + r)^N + 1$$
$$\mathbb{E}[(\frac{S_N - S_0}{S_0})^2] = ((1 + r)^2 + \frac{2}{3}a^2)^N - 2(1 + r)^N + 1$$

The second part is equivalent to the following:

$$
\begin{aligned}
(\mathbb{E}[\frac{S_N - S_0}{S_0}])^2 &= (\mu - 1)^2 \\
&= ((1+r)^N - 1)^2 \\
&= ((1+r)^{2N} - 2(1+r)^N + 1)
\end{aligned}
$$

Putting it back together:

$$Var(\frac{S_N - S_0}{S_0}) = \mathbb{E}[(\frac{S_N - S_0}{S_0})^2] - (\mathbb{E}[\frac{S_N - S_0}{S_0}])^2$$

$$Var(\frac{S_N - S_0}{S_0}) = (((1+r)^2 + \frac{2}{3}a^2)^N - 2(1+r)^N + 1) - ((1+r)^{2N} - 2(1+r)^N + 1)$$

Our observed variance is σ^2, and we want to solve our equation for a in order to input the formula into our Maple program

$$\sigma^2 = ((1+r)^2 + \frac{2}{3}a^2)^N - (1+r)^{2N}$$

$$a = \sqrt{\frac{3}{2}((\sigma^2 + (1+r)^{2N})^{\frac{1}{N}} - (1+r)^2)}$$

Our Maple program was designed to do the calculations from this new starting point where it used the values of u, m, d, and r to compute the values of the risk-neutral probabilities.

Solving for the risk-neutral probabilities, we only had two conditions for three variables.

1)

$$\tilde{p_1} + \tilde{p_2} + \tilde{p_3} = 1$$

2)

$$
\begin{cases}
\tilde{\mathbb{E}}[S_1] & = (1+r)S_0 \\
\tilde{p_1}(uS_0) + \tilde{p_2}(mS_0) + \tilde{p_3}(dS_0) & = (1+r)S_0 \\
\tilde{p_1}u + \tilde{p_2}m + \tilde{p_3}d & = (1+r)
\end{cases}
$$

In order to solve the system of equations for p_1, p_2, p_3, we set $p_3 = t$.

We designed a Maple procedure to compute the range of t so that we could test different values within that range.

Derivation of Newton's Method

With a range for our free parameter, we used Newton's Method, an approximation strategy, in order to determine the best value for the free variable within it's range. Applying this method allowed us to find the t corresponding to the minimum value of our error plot. The calculations for this method are explained below.

Our quoted values for our sample stock were $\hat{C}_0^{k1}, \hat{C}_0^{k2}...\hat{C}_0^{k5}$ Therefore we could assume our least squares function to be,

$$f(t) = \sum_{i=1}^{5}(\hat{C}_0^{ki} - C_0^{ki}(t))^2$$

Since we wanted to minimize the function for $t\epsilon(t_{min}, t_{max})$ we applied Newtons Method.

$$t_n = t_{n-1} - \frac{f'(t_{n-1})}{f''(t_{n-1})}$$

We began by approximating our numerator. From the definition of a derivative we knew that

$$f'(t) \approx \frac{f(t+\delta) - f(t-\delta)}{2\delta}$$

Further, looking at the denominator we were able to approximate that as well.

$$f''(t) \approx \frac{f'(t+\delta) - f'(t-\delta)}{2\delta}$$

Through subsitition from the above calculations, this simplified to,

$$f''(t) \approx \frac{f(t+\delta) - 2f(t) + f(t-\delta)}{\delta^2}$$

In the end, our final equations for finding the minimum t resulted in

$$t_n = t_{n-1} - \frac{\frac{f(t_{n-1}+\delta)-f(t_{n-1}-\delta)}{2\delta}}{\frac{f(t_{n-1}+\delta)-2f(t_{n-1})+f(t_{n-1}-\delta)}{\delta^2}}$$

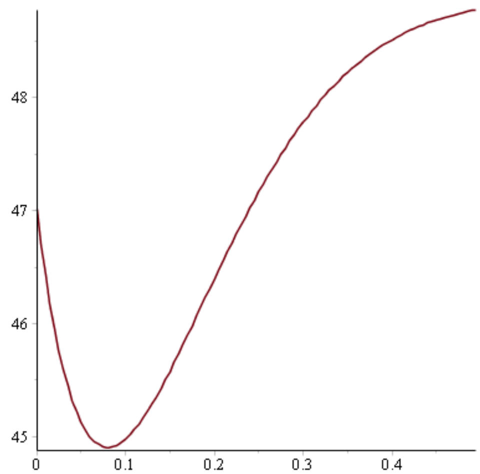
$$t_n = t_{n-1} - \frac{\delta(f(t_{n-1}+\delta) - f(t_{n-1}-\delta))}{2(f(t_{n-1}+\delta) - 2f(t_{n-1}) + f(t_{n-1}-\delta)}$$

After several iterations of Newton's Method, we then used the resulting t, which we referred to as the optimal t value, to compute the final step in the program. The last procedure calculated the risk-neutral probabilities based on the optimal t value and then used backwards induction to determine the time-zero prices of the call options.

Results

We had a variety of different results. Before computing our call option prices, Maple produced the graph of our function so that we could determine if it had a clear minimum or not. We found that some of our plots had clear minimums while other plots had no minimum. The Hershey plot shown below was an example of one of our plots with a clear minimum.

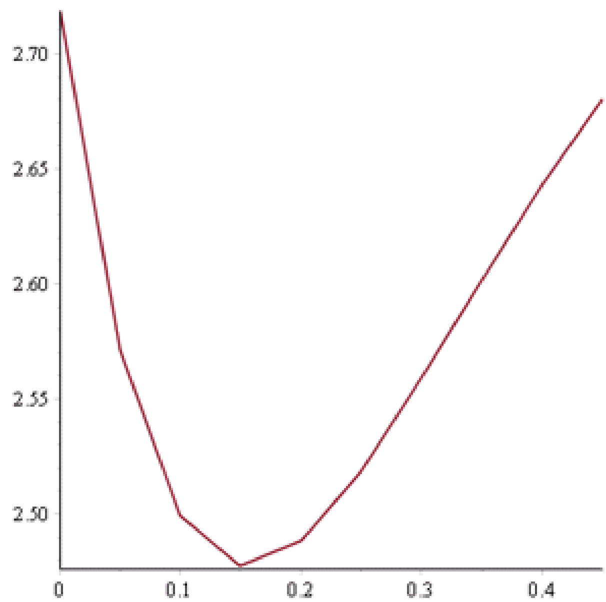
Ten step hershey:



When the graph did not have a clear minimum, the optimal t value converged to a number that was outside of the potential domain of t. Sometimes the optimal t converged to the wrong number if the estimated minimum was too far away from the actual minimum of the data. This occurence led us to question what factor was impacting our error plot the most. A significant finding that we came across was that these graphs differed depending on the strike prices that we initially chose. For our program, we found that the list of strike prices required symmetry around the center value. We also discovered that the value of the center strike price should be the strike value that was closest to the current stock price. We found that there was a ”window” range that we could chose from for the other strike prices. The window had to be a certain range that was not too close, but also not too far from the stock price. Unfortunately we did not discover how exactly to determine this ”window” without guessing and checking. Once we discovered that symmetry played an essential role in choosing our strike prices, the graphs of error became more consistent, with minimum values in the t domain.

We were able to look at different types of stocks and see how close our calculated results were to the quoted values.
6 step JCPenny:

```
S0 := 8.56 :
KL := [7, 8, 8.5, 9, 9.5] :
quotelist := [1.57, .57, .09, .01, .02] :
jList, G := test(sigma, .0002, N, S0, KL, quotelist) :
with(plots) :
plot(jList, G);
```

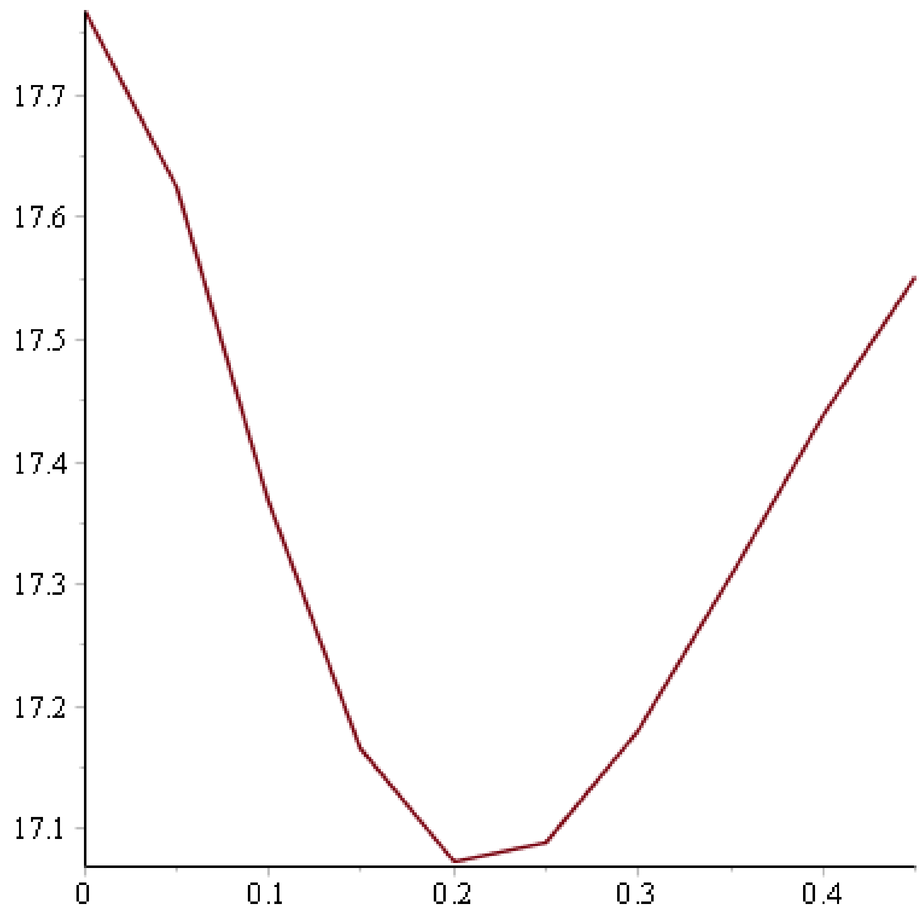



```
> m := .15 :
  optimalT(m, .0025, N, S0, KL, quotelist) :
  finalstep(OT, N, S0, KL);

  [1.568394123, 0.5782694415, 0.1886501026, 0.02831829810, 0.001235657979]
```

Comparing the *quotelist* in the picture to our blue, calculated list of numbers, four of our calculated values were less than 2 cents away from the market call prices. Since the middle value was nine cents away from the quoted value, we believe that this option was mispriced given the fact that our model was accurate for the four other values. This output was our best result. We had outputs for the other stocks, but they were generally further away from the quoted values which is not useful for our model.

AT&T:



0.2165387166

[4.221182196, 0.9354902445, 0.6257084380, 0.1044328109, 0.0005861781481]

This next example was a more accurate depiction of results that we were computing. The calculated values in blue were relatively close to the values in the quoted list: [4.05, 0.98, 0.63, 0.12, 0.01]. Although these results were relatively consistent, they were still not significant enough to make any clear judgements on whether options were mispriced or not. Unfortunately, the majority of our results matched this pattern.

Conclusion

Although Maple as a computer program was very useful and fast, there were a few things that we could improve upon in our project that would be beneficial in the future. In the Maple procedure where the stock values were computed at future times based on u, m and d, each value was put in a list where equivalent amounts were not combined, so there were many repeats. These list could be reduced in size since multiplication is commutative so that our lists would not grow exponentially and slow Maple's computation time when it referred back to numbers in that list. We set up our program like this because it made the backwards induction process more simple to compute. Turning the lists into sequences would be an alternate option that would prevent repeats and possibly make our procedure faster and more efficient.

In future exploration of this project, it would be beneficial to see what factors of stocks cause some optimal free parameter values to be outside of it's own range, while others show a smooth parabola with a clear minimum. Once this question can be answered, then it would be interesting to see if and how the model can be altered to calculate option prices for those stocks that have a resulting free parameter value outside of it's range. Another factor that requires further inquiry is whether using different approximation methods, such as Householder's Method or the Secant Method, would have an impact on the accuracy of the results. We specifically used Newton's Method because we were familiar with it, and it is known to converge quickly. We believe that with further exploration into these questions, our model could become very useful in determining the values of call options.