Option Pricing in the Presence of Transactions Costs

Chi Lee

1. Introduction

The widely accepted model for option pricing, as developed by Black and Scholes, uses the no-arbitrage framework whereby the payout of an option is replicated using a dynamic hedging strategy. The argument is that any price that differs from the cost of setting up this replicating portfolio, or the no-arbitrage price, leads to an opportunity to make an arbitrage profit.

At the heart of the model is the assumption that once the replicating portfolio is set up, no additional funds are needed. This self-financing would be achieved only if any adjustments in the hedge resulting from changes in the underlying can be financed exactly from the old portfolio. This is possible only if there are no transactions costs.

In the presence of transactions costs, the arbitrage argument fails. With transactions costs, continual infusion of funds is required into the replicating portfolio and regardless of how small the transactions costs are, hedging continuously would cause the overall transactions costs to grow without bound.

Discrete hedging would bound the effects of transactions costs, but this would lead to errors in the replication and would also invalidate the arbitrage argument.

An alternative approach to the no-arbitrage framework is to value options using economic theory where investors act according to their preferences. In arbitrage pricing, the risk tolerance of an investor is irrelevant because the investor is never exposed to any risk. If an arbitrage opportunity exists, investors with different levels of risk aversion would follow the same strategies to profit from an arbitrage.

In the preference-based approach, an investor follows a trading strategy that optimizes her preferences. So the value of an option is the expected cost of creating and hedging a portfolio that replicates the payout of an option. Non-zero transactions costs will influence the trading strategy of the investor and change the expected cost of the replication, but its effect will be bounded.

This paper will use the economic approach to model the effects of transactions costs on the valuation of options. Before presenting the model, we develop the ideas of a utility function to describe investor preferences and dynamic programming for optimizing a trading strategy.

2. Utility Functions

Borrowing from economic theory, the preferences of an investor can be represented by a utility function. Utility is the quantity that an investor is trying to maximize. So in making a decision, the investor is concerned with picking a strategy that will lead to the maximum amount of utility. For our analysis, we will be concerned only with the utility of wealth or monetary value.

Utility functions are assumed to be monotonically increasing and strictly concave. An increasing utility function means that an investor would always prefer having more wealth. The assumption of concavity translates to the view that each additional unit of wealth has diminishing utility to the investor. It also leads to the result that investors are risk averse. This is a desired result since investors are generally assumed to have an aversion to risk. Stated more formally, a utility function $U(\cdot)$ is risk averse at wealth W if for all gambles $\tilde{\varepsilon}$ with $E(\tilde{\varepsilon}) = 0$ if

$$U(W) > EU(W + \tilde{\varepsilon}).$$

The proof that a strictly concave function leads to risk aversion is proved as follows. Assuming $E(\tilde{\varepsilon}) = 0$ and $U(\cdot)$ is strictly concave at W, by Jensen's inequality

$$E[U(W + \tilde{\varepsilon})] < U(E[W + \tilde{\varepsilon}]) = U(W).$$

In choosing a utility function, it must be continuous and differentiable. Commonly used utility functions include the negative exponential utility function

$$U(W) = -e^{-aW}$$

and the power utility function

$$U(W) = W^{a}.$$

Typically, wealth is always assumed to be positive, however, in our option pricing model there are instances of negative wealth. In obtaining the results presented later in this paper, negative exponential utility function was used.

3. Dynamic Programming

An optimal trading strategy can be seen as a stochastic control problem where the control represents a trading strategy that effects the final outcome of a process. The control is optimized over all possible values to meet a chosen goal, usually determining the minimum or maximum expected value of the process.

Dynamic programming is a technique for solving stochastic optimization problems in a discrete setting. It is a useful method for computing both the optimal strategy and value when the problem is optimizing over some expected value at a later point in time.

The main idea behind dynamic programming is that when making a series of decisions, a decision made in an earlier period must be consistent with the intention of optimizing over all later periods. If the optimal strategy at time $t+1$ is known, then determining the optimal strategy at time t becomes a single period problem. So by working backwards one period at a time, a multi-period decision problem can be reduced to a sequence of one-period problems. Using a recursive procedure, the optimal controls with one period to go can be computed first, then the controls with two periods to go, and so forth.

While dynamic programming can be applied in a deterministic setting, we are concerned with how it operates in a Markov setting. Markov chains have the property that all the information relevant to predicting the evolution of the process at time t is simply the state at time t. Past information is not needed. A Markov control process X has a control function $\alpha(X,t)$ that depends on the state as well as time.

To solve for a maximum expected value of a function that is driven by a Markov control process at time T

$$f(x_t,t) = \max_{\alpha} E_{\alpha}[\, f(X(T)) \mid X(t) = x_t \,]$$

we compute $f(x,T-1)$ from $f(x,T)$ and solve recursively backwards. Since the optimal decision strategy for later periods must be known, only the control for the current time need to be solved. For transition probabilities $p(x_t, x_{t+1}, \alpha(t))$ and state space Ω ,the expected value at time $t+1$ is

$$E[\, f(X(t+1), t+1) \mid X(t) = x_t, \alpha(t)] = \sum_{x_{t+1}\in\Omega} f(x_{t+1}, t+1) p(x_t, x_{t+1}, \alpha(t)).$$

The optimal control leads to the maximum expected value at time t:

$$f(x_t, t) = \max_{\alpha(t)} \sum_{x_{t+1} \in \Omega} f(x_{t+1}, t+1)\, p(x_t, x_{t+1}, \alpha(t)).$$

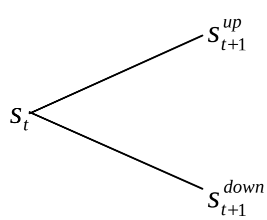
Starting with the final period when the values are known, this equation is used each period to compute the desired value at the current time.

4. Binomial Process

We follow the Black-Scholes theory and assume the underlying security follows a geometric brownian motion

$$\frac{dS}{S} = \mu dt + \sigma dZ$$

where Z is a normally distributed random variable with $E[Z] = 0$ and $\mathrm{var}[Z] = dt$.



Since dynamic programming is applicable only in a discrete setting, we model the underlying security as a recombining binomial process $S(t)$. So given that at time t $S(t) = s_t$, the asset price can either go up to $s_{t+1}^{up} = s_t u$ or down to $s_{t+1}^{down} = s_t d$ at time $t+1$. Since the process is recombining $u = 1/d$, we can match mean and variance of the discrete binomial process and the original continuous assumption to solve for the probability. The expected value of $S(t+1)$ is

$$E[S(t+1) \mid S(t) = s_t] = (pu + (1-p)d)s_t = e^{\mu dt} s_t$$

and the variance of $S(t+1)$ is

$$\begin{aligned} \mathrm{var}[S(t+1) \mid S(t) = s_t] &= (pu^2 + (1-p)d^2 - e^{2\mu dt})(s_t)^2 \\ &= e^{2\mu^2 dt}(e^{\sigma^{2dt}} - 1)(s_t)^2. \end{aligned}$$

Using the above equations we get the probability of an up move

$$p = \frac{e^{\mu dt} - d}{u - d}$$

5. General Model (no transactions cost)

Using the economic framework, the value of an option $V(s_t, t)$ is the expected cost of hedging an option with the payoff at maturity $V(s_T, T)$ using a trading strategy that maximizes the expected utility of the investor. The optimal control will solve for

$$\max_{\alpha} E_{\alpha}[U(V(s_T, T), s_T, \alpha)]$$

or for a single period

$$\max_{\alpha(t)} \sum_{s_{t+1} \in \Omega} U(V(s_{t+1}, t+1), s_{t+1}, \alpha) p(s_t, s_{t+1}, \alpha(t)).$$

To compute the value of the option, we take the expected utility and use the inverse utility function to map back into monetary units or wealth

$$V(s_t, t) = U^{-1} \max_{\alpha} E_{\alpha}[U(V(s_{t+1}, t+1), s_{t+1}, \alpha)]$$

so the recursive equation for dynamic programming would be

$$V(s_t, t) = U^{-1} \max_{\alpha(t)} \sum_{s_{t+1} \in \Omega} U(V(s_{t+1}, t+1), s_{t+1}, \alpha) p(s_t, s_{t+1}, \alpha(t)).$$

We can rewrite these formulas more explicitly for a binomial process.

6. Binomial Model (no transactions cost)

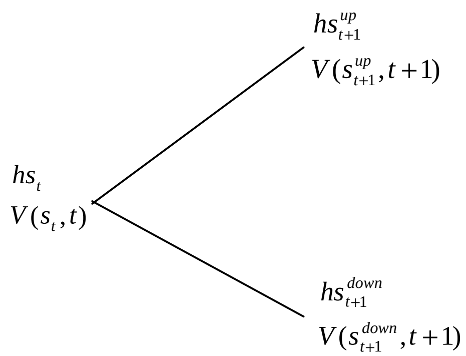
An option can be hedged by setting up a portfolio that closely replicates the payoff of the option in any given state. In our model, that replicating portfolio will simply be a holding of the underlying security $(h \in H)$. The control that the investor will use to optimize is the holding level at each period. In the Black-Scholes model, this holding level is set equal to the delta of the option. So at time t the investor will be long h amount of the underlying and short the option that she is trying to hedge

$$hs_t - V(s_t, t)$$

and assuming a binomial process two possible states can occur

$$\text{up state: } hs_{t+1}^{up} - V(s_{t+1}^{up}, t+1) \text{ or}$$

$$\text{down state: } hs_{t+1}^{down} - V(s_{t+1}^{down}, t+1).$$



We can compute the value of the option for the realization of each state by setting the portfolio at time t to the discounted value of the portfolio at time $t+1$ with the risk-free rate r and time interval dt

$$\text{up state: } hs_t - V(s_t, t) = (hs_{t+1}^{up} - V(s_{t+1}^{up}, t+1))e^{-rdt}$$

$$V(s_t, t) = V_t^{up} = (V(s_{t+1}^{up}, t+1) - hs_{t+1}^{up})e^{-rdt} - hs_t$$

$$\text{down state: } V(s_t, t) = V_t^{down} = (V(s_{t+1}^{down}, t+1) - hs_{t+1}^{down})e^{-rdt} - hs_t$$

The expected utility of a given hedge strategy is easy to solve because we already know the transition probabilities of the two states.

$$U(V_t^{up}) * p + U(V_t^{down})(1-p)$$

By maximizing the utility over all hedge strategies and converting back into units of wealth we arrive at the following equation for the value of an option at time t.

$$V(s_t, t) = U^{-1} \max_{h \in H}[U(V_t^{up}) * p + U(V_t^{down})(1-p)]$$

or

$$V(s_t, t) = U^{-1} \max_{h \in H} [U((V(s_{t+1}^{up}, t+1) - hs_{t+1}^{up})e^{-rdt} - hs_t) * p$$
$$+ U((V(s_{t+1}^{down}, t+1) - hs_{t+1}^{up})e^{-rdt} - hs_t)(1-p)]$$

In a world with no transactions costs, we expect arbitrage pricing to prevail and in order for this model hold any validity, it should match the Black-Scholes price. We can see that because investors have preferences and are averse to risk, the optimal hedging strategies are close to delta used for the replicating portfolio in the arbitrage model. In the arbitrage model, a replicating portfolio would have the same value in all the states. Using the binomial example above

$$hs_{t+1}^{up} - V(s_{t+1}^{up}, t+1) = hs_{t+1}^{down} - V(s_{t+1}^{down}, t+1)$$

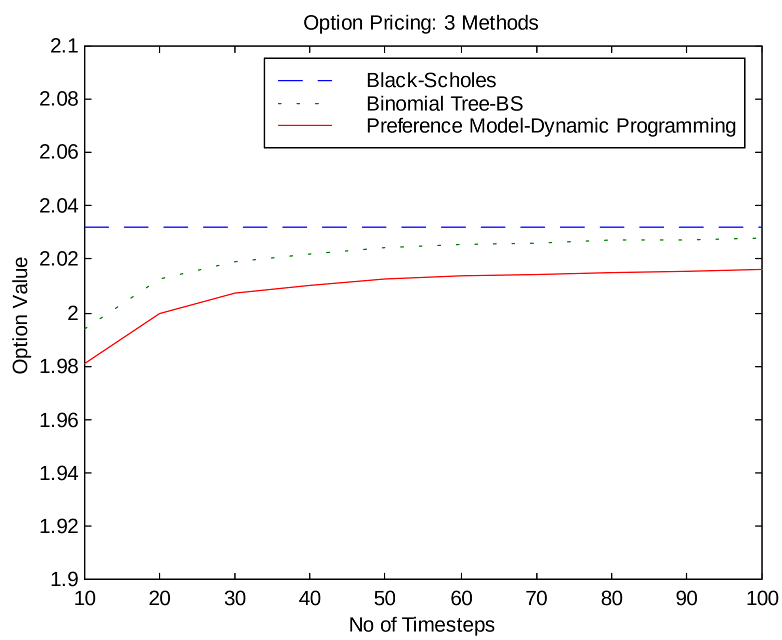
$$h = \frac{V(s_{t+1}^{up}, t+1) - V(s_{t+1}^{down}, t+1)}{s_{t+1}^{up} - s_{t+1}^{down}}$$

In our model, we assume that the investors are risk averse and their utility function is concave. As shown previously, this means that

$$U(W) > EU(W + \tilde{\varepsilon})$$

So assuming that the gamble is fair, the investor's optimal strategy is when h =delta, i.e. $Var[\tilde{\varepsilon}] = 0$. In the graph below we can see that the option price produced by the model converges to the Black-Scholes prices with smaller timesteps.

Figure 1: Comparison of Option Pricing Models



7. Binomial Model (with transactions cost)

Expanding the model that we have built, we can incorporate transactions cost. Previously, we have only been concerned with the amount of hedge held from time t to $t+1$. The amount of hedge held from time $t-1$ to t was not relevant for evaluating the price of the option at time t because there were no costs involved in changing the amount of the hedge. Once we assume that transactions costs are positive, the hedge level h_b set before time t becomes important in determining what the optimal hedge level h_a after time t should be. Using h_b, the transactions costs can be computed.

There are three general method for calculating transactions costs: i) transactions cost can be modeled as a percentage of the underlying security, ii) there can be a fixed charge for each share, or iii) there can be a single flat fee regardless of the number of shares. Obviously, the transactions cost can also be any combination of these three methods. We will be looking at all three ways of charging transactions cost. First, we focus on modeling transactions costs as a percentage c_1 of the underlying.

To move from one hedge level to another the following transactions cost is incurred

$$s_t \mid h_a - h_b \mid c_1$$

This cost is included in the portfolio

$$h_a s_t - V(s_t, t) + s_t \mid h_a - h_b \mid c_1.$$

In the two possible state at time $t+1$ the cost of the hedge becomes

$$\text{up state: } V(s_t, t, h_b) = (V(s_{t+1}^{up}, t+1, h_b) - h_a s_{t+1}^{up})e^{-rdt} + s_t \mid h_a - h_b \mid c_1 - h_a s_t$$

$$\text{down state: } V(s_t, t, h_b) = (V(s_{t+1}^{down}, t+1, h_b) - h_a s_{t+1}^{down})e^{-rdt} + s_t \mid h_a - h_b \mid c_1 - h_a s_t$$

and the value of the option is

$$V(s_t, t, h_b) = U^{-1} \max_{h_a \in H}[\, U((V(s_{t+1}^{up}, t+1, h_b) - h_a s_{t+1}^{up})e^{-rdt} + s_t \mid h_a - h_b \mid c_1 - h_a s_t) * p$$

$$+ U((V(s_{t+1}^{down}, t+1, h_b) - h_a s_{t+1}^{down})e^{-rdt} + s_t \mid h_a - h_b \mid c_1 - h_a s_t)(1-p)]$$

or more generally, assuming the total transactions costs of moving from one hedge state to another is tc, the equation can be written as

$$V(s_t, t, h_b) = U^{-1} \max_{h_a \in H} [\, U((V(s_{t+1}^{up}, t+1, h_b) - h_a s_{t+1}^{up})e^{-rdt} + tc - h_a s_t) * p$$

$$+ U((V(s_{t+1}^{down}, t+1, h_b) - h_a s_{t+1}^{down})e^{-rdt} + tc - h_a s_t)(1-p)].$$

So going back to our model of transactions costs as a percentage of the underlying,

$$tc = tc_1 = s_t \mid h_a - h_b \mid c_1.$$

In the two other types of transactions costs where a fixed cost c_2 is charged for each share and a flat fee c_3 is charged for each trade, the total transactions cost can be represented as

$$tc_2 = \mid h_a - h_b \mid c_2$$

$$tc_3 = c_3$$

, respectively.

8. Results

The following results were generated for the valuation of a call option assuming a spot stock price $s_0 = 10$, strike price $k = 10$, volatility $v = .4$, drift of the stock $\mu = .18$, risk-free rate $r = .10$, and time to expiration $T = 1$. In addition to the value of the option, we are also interested in obtaining the expected number of shares traded Ψ where the investor has no initial holdings $h_0 = 0$ and is assumed to follow the optimal trading strategy α

$$\Psi(h_0, \alpha) = E_\alpha[\sum_{t=0}^{T} \mid h_{t+1} - h_t \mid].$$

This number will provide insight into how transactions costs influence the hedging strategy. In Figure 2 and 3, we set the total number of timesteps n to 100 and the total hedge units m to 200, i.e. when $h = 200$ the replicating portfolio will have a delta of 1 and be fully hedged.

Consistent with our intuition, the value of the option in Figure 2 increases as the cost of transacting a trade (c_1, c_2, and c_2) goes up for all three types of costs. This was expected because, any increase in the transactions cost would increase the cost of creating and maintaining a replicating portfolio. Figure 3 also reaffirms our expectation of what happens to Ψ as transactions costs goes up. Opposite to the option value, as c_1 and c_2 decrease, Ψ increases. Theoretically, when there are no transactions costs, and $m = \infty$ (hedge units are not discrete) , Ψ should go to ∞ . This is the continuous hedging assumption of the Black and Scholes model. Interestingly, as c_3 grows, Ψ first goes down and then goes back up. This is due to the fact that a flat fee which disregards the number of shares to be traded, will encourage an investor to optimize her hedge by adjusting a larger number of shares with fewer trades. Thus, the expected number of shares traded will not necessarily go down with increased cost.

Figure 2: Option Prices with Transactions Costs

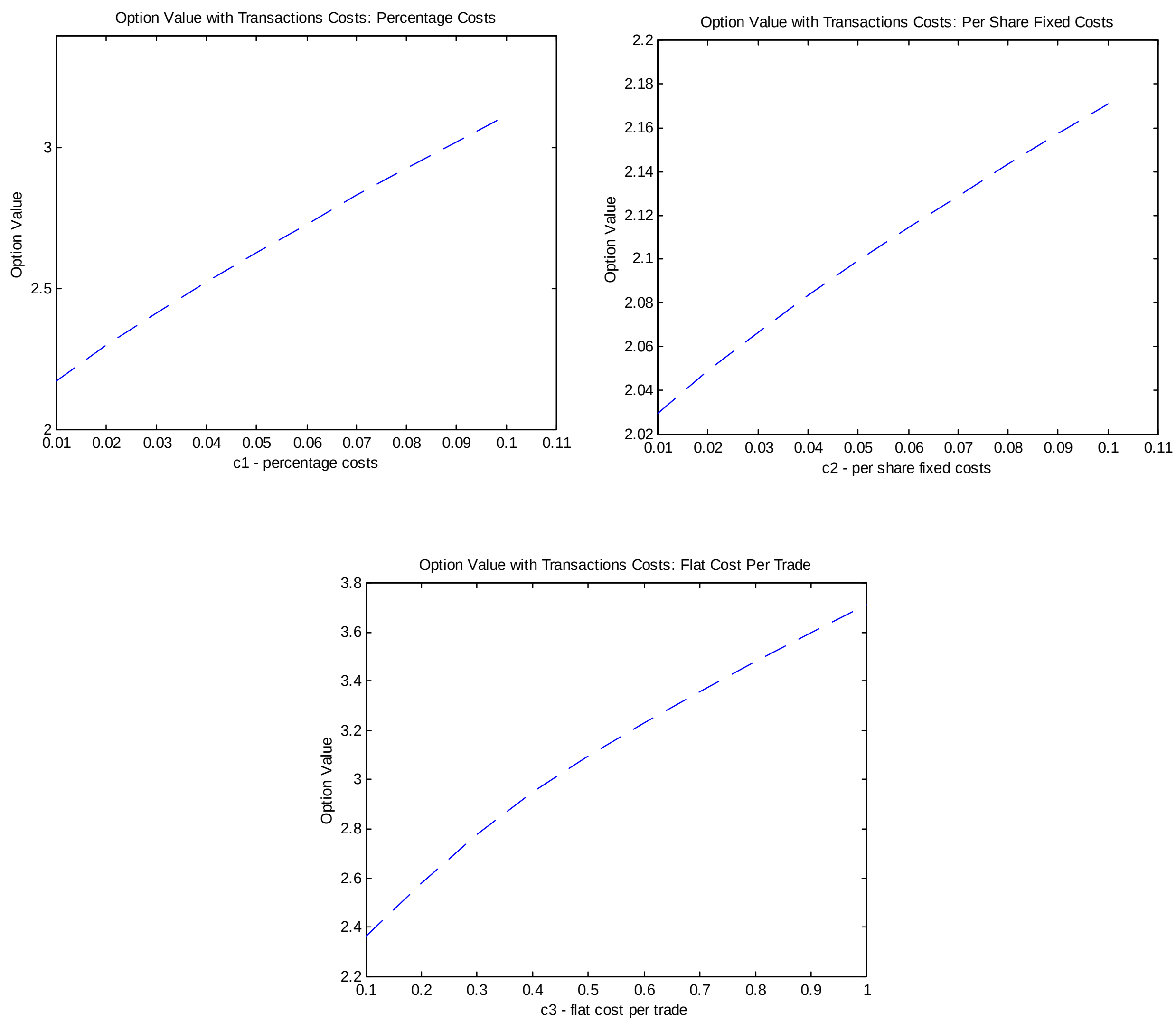
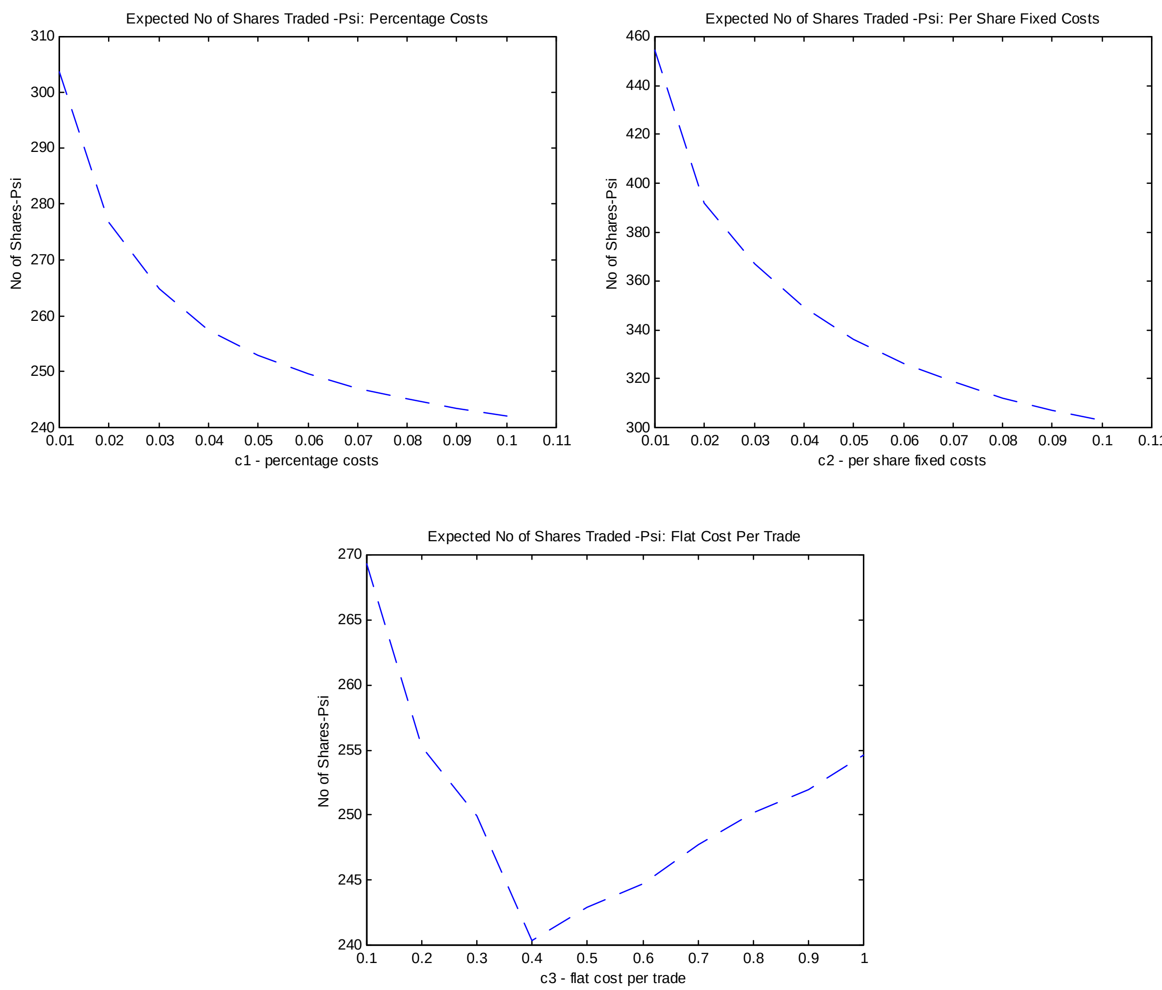
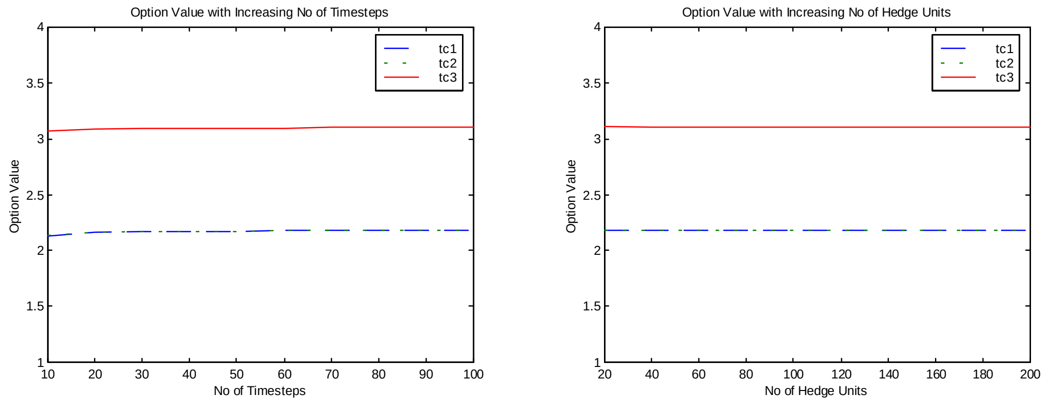


Figure 3: Expected No of Shares Traded - with Transactions Costs



In Figure 4, we show that unlike the discrete arbitrage models where prices grow without bound as the size of the timesteps or the hedge units get smaller, the option price converges to some value as m and n increase, i.e. size of the timesteps and hedge units decrease. For these computations as well as for Figures 5 to 8, $c_1 = 0.01$, $c_2 = 0.10$, and $c_3 = 0.50$. Note that in Figure 4, the option values for $tc = tc_1$ and $tc = tc_2$ are similar and the two graphs lie on top of each other.

Figure 4: Convergence with Finer Time Step and Hedge Unit Size



In Figures 5 to 8, we take a close look at the optimal hedging strategy when there are no transactions cost (Figure 5), and for the three transactions cost models: $tc = tc_1$ (Figure 6), $tc = tc_2$ (Figure 7), and $tc = tc_3$ (Figure 8). Both the number of time steps n and the number of hedge units m are set to 10. For each of the models, a cross section is taken at different timesteps t or periods between the initial time and maturity, i.e. $t = 2$, $t = 5$, and $t = 8$. For each of these timesteps, a contour graph illustrates the optimal adjustment in hedge $(h_a - h_b)$ for every hedge state h_b. These figures also show the optimal adjustment in hedge given particular hedge states, i.e. $h_b = 2$, $h_b = 5$, and $h_b = 8$, at all the nodes of the binomial tree.

Figure 5: Hedging Strategy with No Transactions Costs

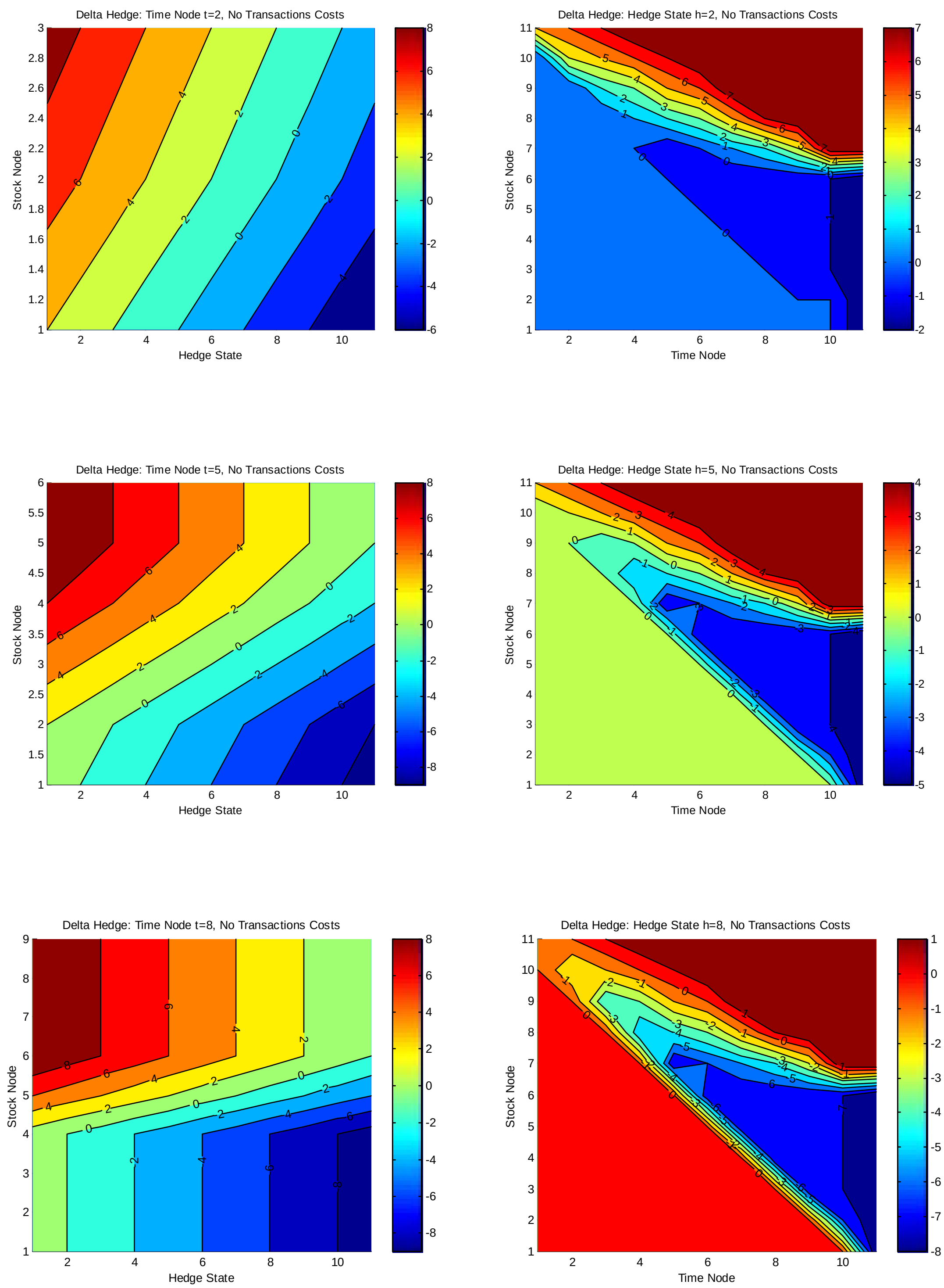


Figure 6: Hedging Strategy with Transactions Costs – Percentage Costs

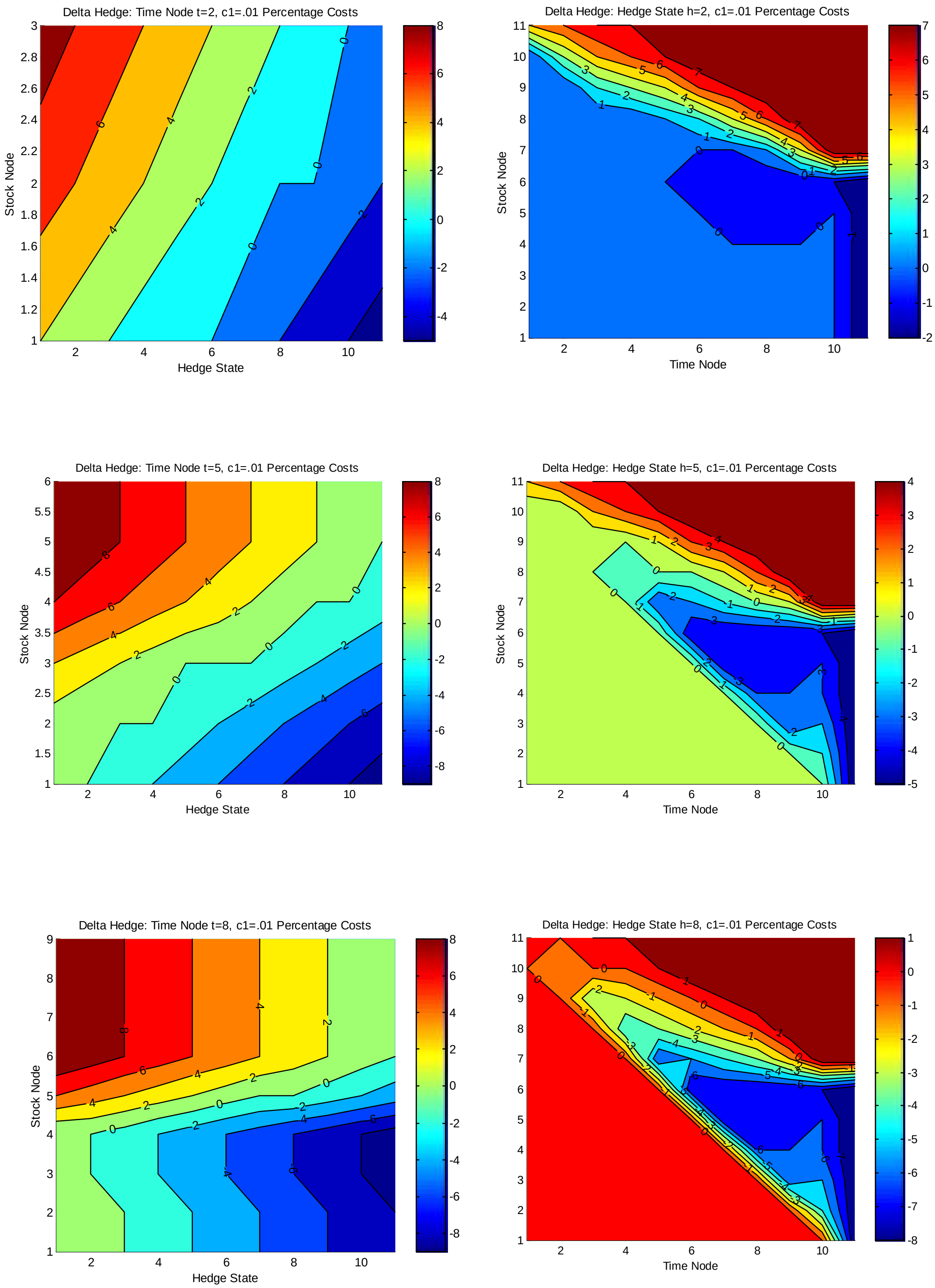


Figure 7: Hedging Strategy with Transactions Costs – Fixed Per Share Costs

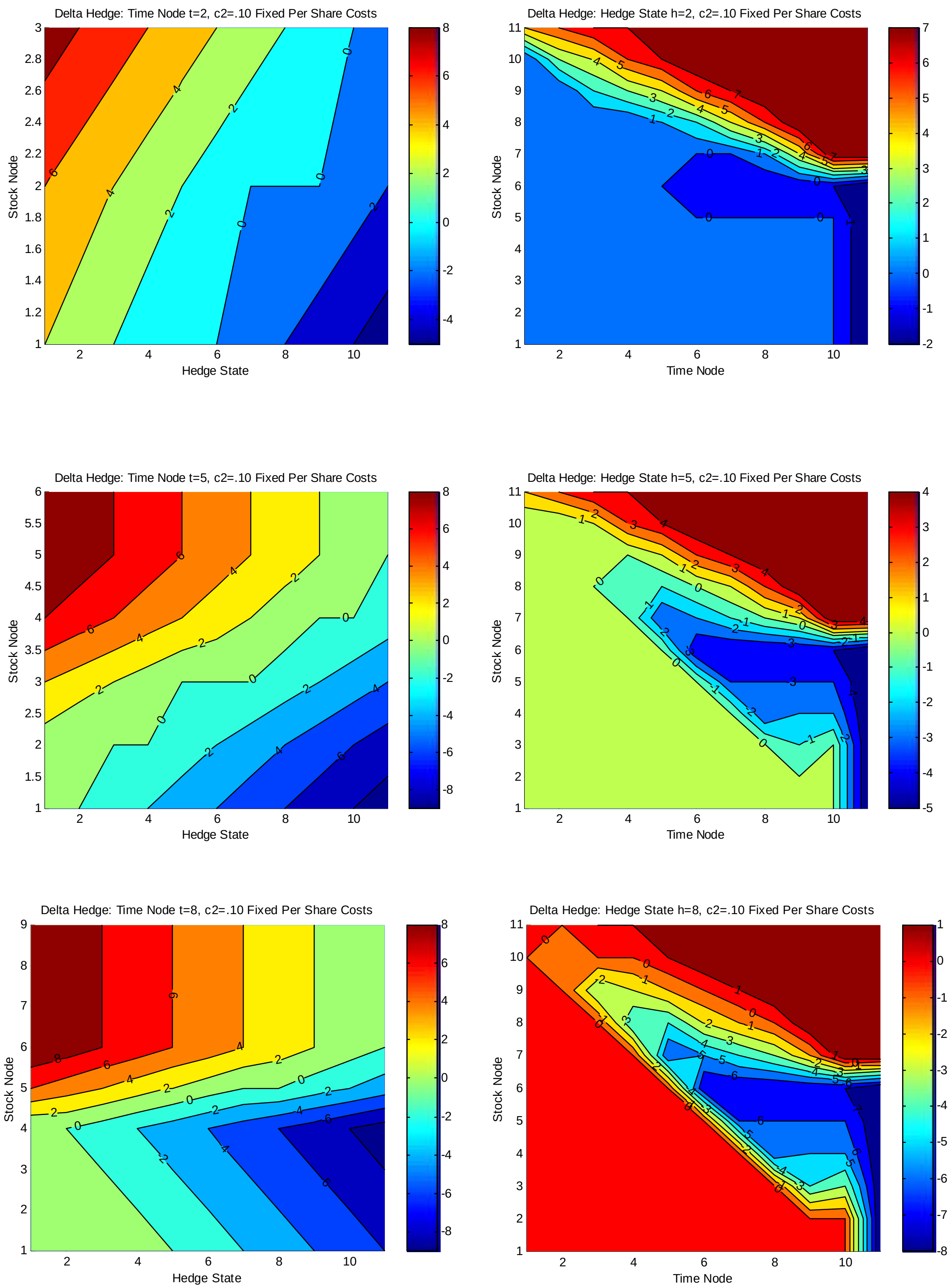
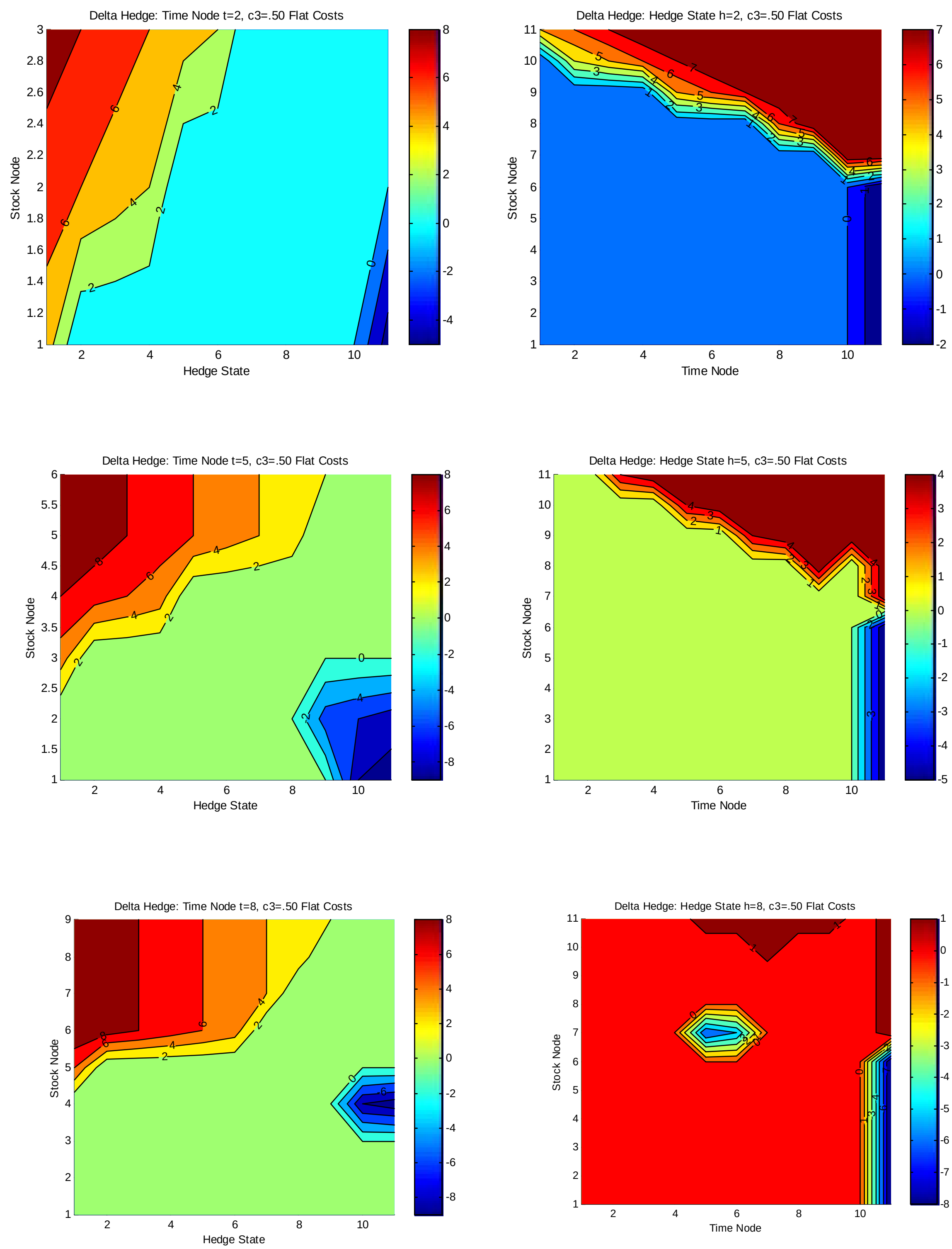


Figure 8: Hedging Strategy with Transactions Costs – Flat Cost Per Trade



These results clearly show that the optimal hedging strategy is effected by transactions costs. Depending on the type of transactions costs, the strategy is different. However, transactions costs generally lead to reduced number and size of readjustments to the portfolio.

Understanding the sensitivities of the valuation to its parameters is an important part of option pricing. Since the parameters are changing constantly, the hedging process is much more complicated in practice. Figures 9 to 13 displays the effect of transactions costs on the sensitivities (or Greeks) of the option, i.e. the first derivative of the option value w.r.t. s_0 (Figure 9), v (Figure 10), μ (Figure 11), r (Figure 12), and T (Figure 13). Each of the graphs include three curves for in the money ($k = 14$), at the money ($k = 10$) , and out of the money ($k = 6$) options.

For nearly all the parameters, greater transactions costs lead to an increase in sensitivities. An exception is the sensitivity to μ which has mixed results. Note that the sensitivity to T is often called Theta and it measures the sensitivity to a decrease in time to expiration. Figure 13 below shows the sensitivity to an increase in time to expiration.

Figure 9: Sensitivity to Spot Stock Price

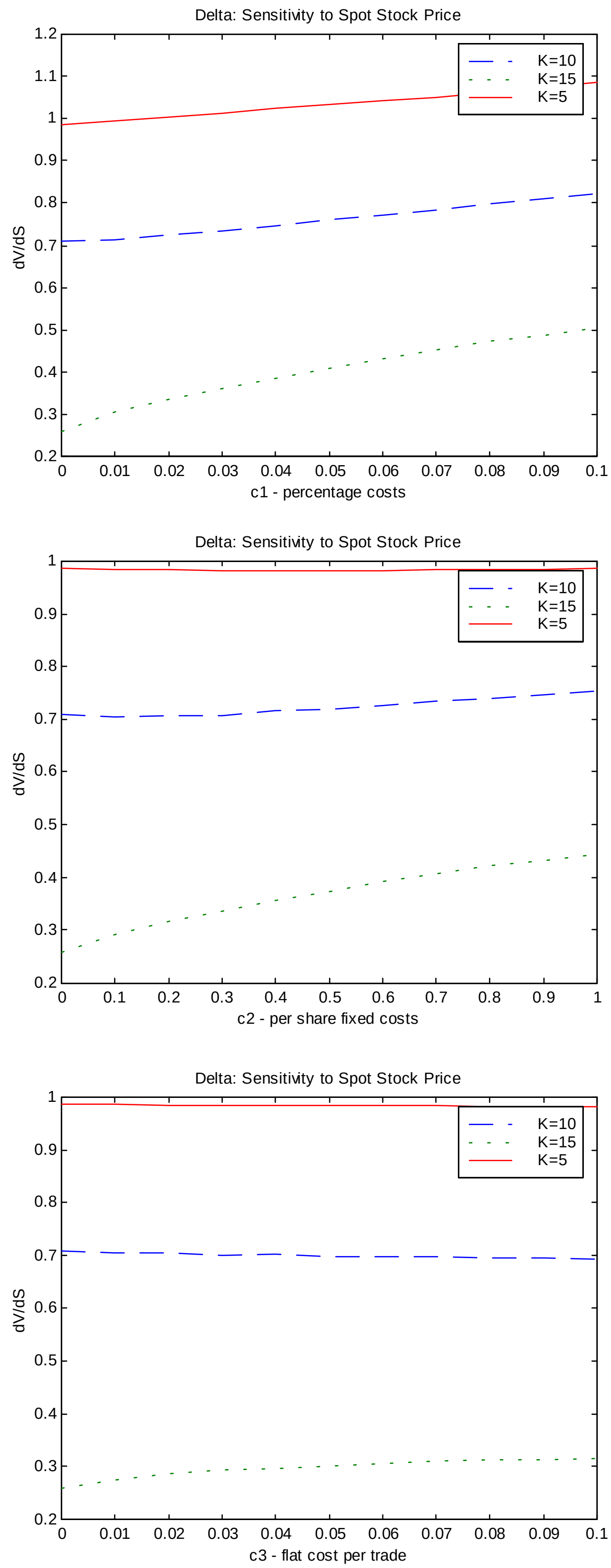


Figure 10: Sensitivity to Volatility

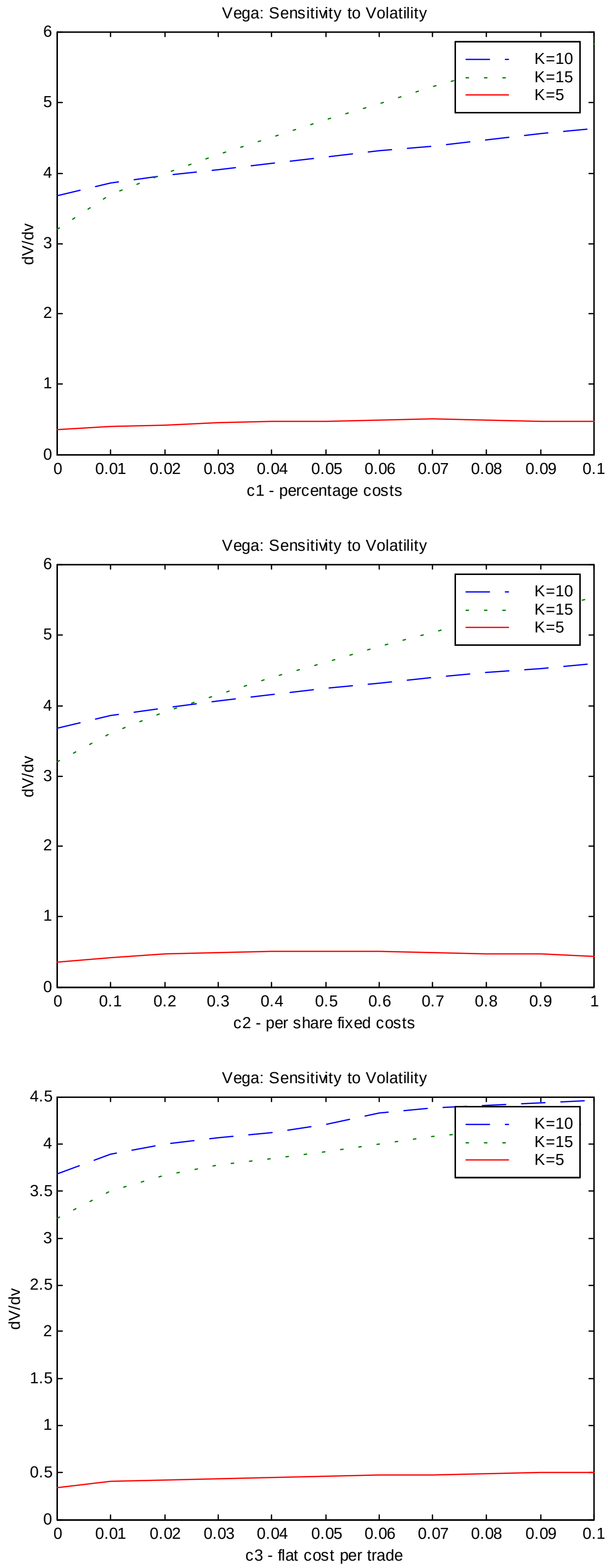


Figure 11: Sensitivity to Expected Return of Stock (mu)

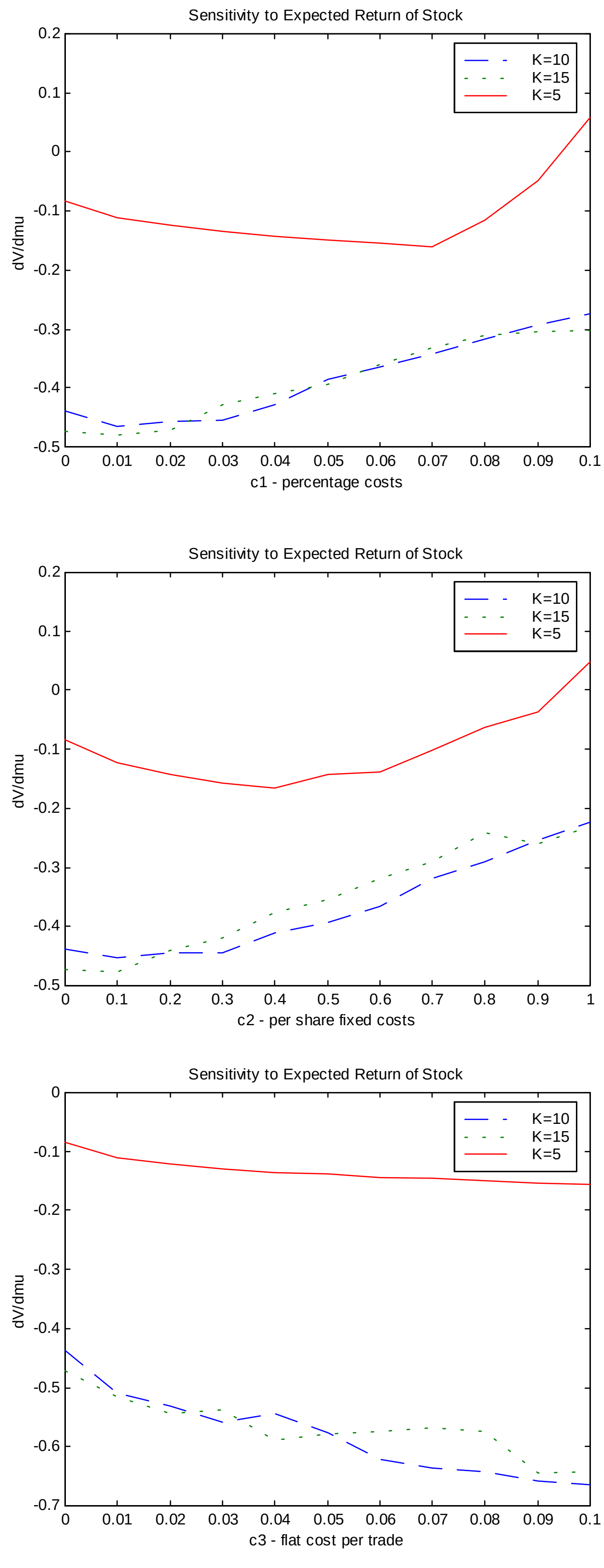


Figure 12: Sensitivity to Risk-Free Rate

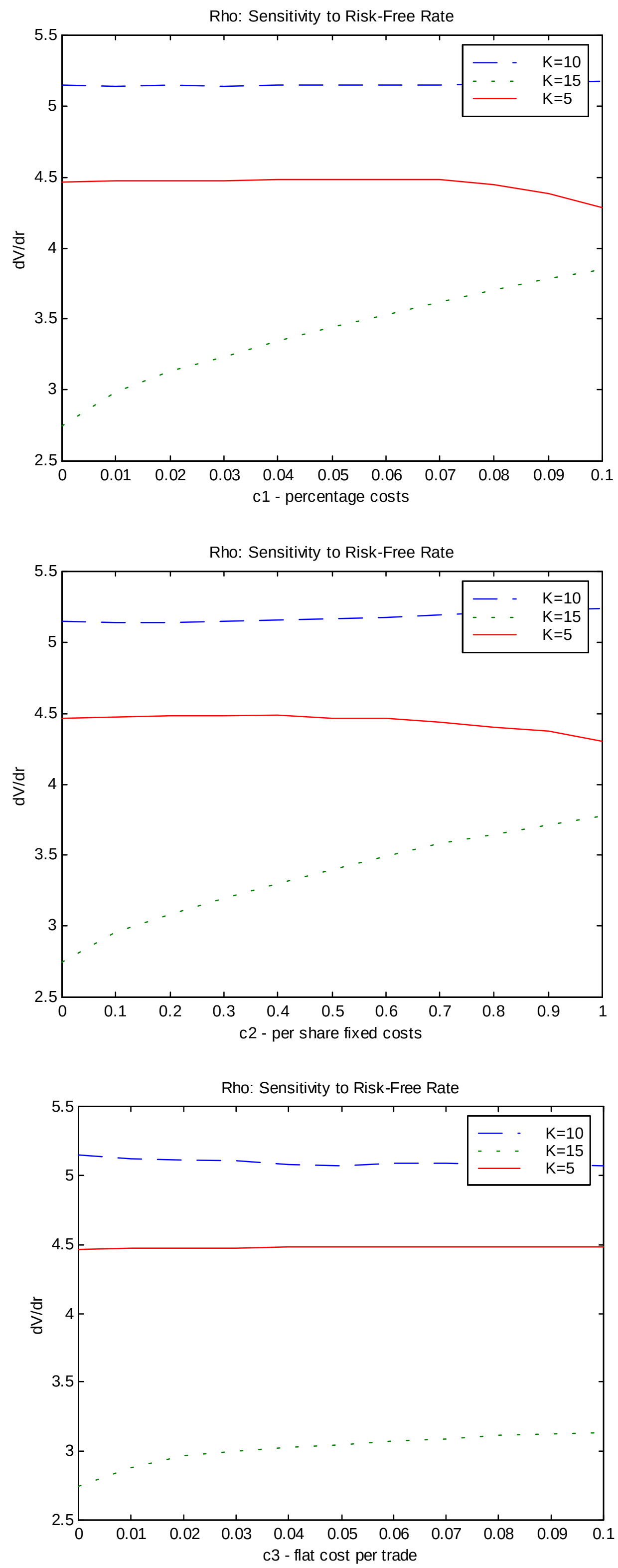
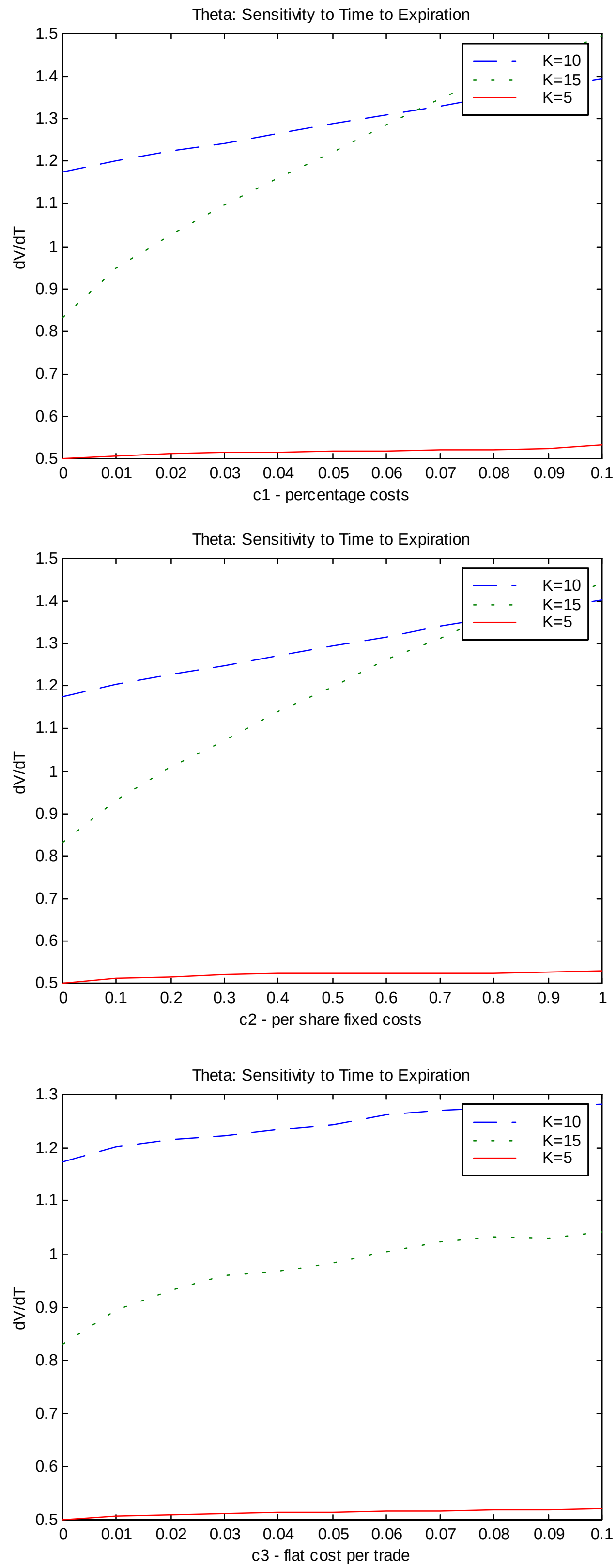


Figure 13: Sensitivity to Time to Expiration



9. Summary

Arbitrage pricing provides a powerful argument for the valuation of contingent claims. However, the model falls apart when its assumptions are relaxed. In particular, the option price grows without bound when the transactions costs are non-zero. The preference-based models, like the one that we have developed, provide a more flexible framework that is viable with less restrictive assumptions.

By using dynamic programming, we have been able to price contingent claims in a discrete setting as the expected cost of hedging the option over an optimal strategy. We have verified that when there are no transactions costs, our model produces a price that converges to the Black-Scholes price.

Given our assumptions, the results clearly indicate that transactions costs effect both the price of the option and the hedging behavior. Since transactions costs do exist in the marketplace, our model should provide useful insights into the pricing of options. In particular, it is a useful model of how hedging strategies change with different transactions costs.

10. Bibliography

Black, Fischer, and Myron Scholes. 1973. "The Pricing of Options and Corporate Liabilities." *Journal of Political Economy* 81: 637-54.

Ingersoll, Jr., Jonathan E. 1987. *Theory of Financial Decision Making*, Rowman & Littlefield, Totowa, NJ.

Pliska, Stanley R. 1997. *Introduction to Mathematical Finance*, Blackwell Publishers Inc., Malden, MA.

Appendix

The following MATLAB function prices a call option with transactions costs c_1, c_2, and c_3 using dynamic programming. The investor is assumed to have the utility function, `utility_fn()`.

```
function[value,dh]=option_pricer(So,K,v,mu,r,T,n,h,c1,c2,c3,a,type)
S(1,1)=So;
dt=T/n;
u=exp(v*sqrt(dt));
d=exp(-v*sqrt(dt));
p=(exp(mu*dt)-d)/(u-d);

%Stock Tree
for t=2:n+1
   for s=1:t-1
    S(s,t)=S(s,t-1)*u;
   end
    S(t,t)=S(t-1,t-1)*d;
end

for db=0:h
   fb(:,db+1)=max(S(:,t)-K,0);
   dh(:,n+1,db+1)= (S(:,t)>K)*h-db;
end

for t=n:-1:1                      %Time
for s=1:t
   fa=fb;                         %fa – option value previous timestep
  for db=0:h                      %fb – option value current timestep
     for da=0:h
        dS=da/h*S(s,t);
        cost=abs(da-db)/h*S(s,t)*c1+c2*abs(da-db)/h+c3;
        if da==db
          cost=0;
        end
   up=(fa(s,da+1)-da/h*S(s,t+1))*exp(-r*dt)+dS+cost;
   down=(fa(s+1,da+1)-da/h*S(s+1,t+1))*exp(-r*dt)+dS+cost;

utility(da+1)=utility_fn(up,a,type,utype,1)*p+utility_fn(down,a,type,uty
pe,1)*(1-p);
end  %da

fb(s,db+1)=utility_fn(max(utility),a,type,utype,-1);
[blank,opt_da]=max(utility);

opt_da=opt_da-1;
dh(s,t,db+1)=opt_da-db;

clear utility;
end  %db
end  %s
end  %t

value=fb(1,1)
```