Chapter 6

Delta Hedging

I will assume that you are already familiar with the Black–Scholes theory of delta hedging. The aim of this Chapter is to test how delta hedging works in practice. What happens if we only hedge in discrete time? What if there are transaction costs.

If you would like to link this material with the previous reading on risk-neutral pricing, the motivation for delta hedging is described in section A6 of "C++ for Financial Mathematics" by John Armstrong. Section 14.1 of that book also derives the difference equations for discrete time delta hedging. No C++ is needed to understand this material.

In practice, delta hedging doesn't work perfectly so we would like to address the question of how much a trader should actually charge for an option in practice. We'll give an initial answer using a simple idea of my own invention called the 99%-profit price. At the end of the chapter we will redo the calculation using a more sophisticated idea called indifference pricing.

6.1 The 99% profit price

To test the effectiveness of delta hedging we will make an unconventional definition.

Definition. *The* 99%-profit price *is the amount that a trader needs charge for a financial product to ensure that they only make a loss* 1% *of the time.*

This definition is completely non-standard, so don't mention it in job interviews and expect it to be understood. It is my invention. It is a simplified version of the "indifference price" that we will discuss later.

Note that the 99%-profit price epends not only on the product being sold but also on the trader's strategy. If one trader has a better strategy than another trader, they will be able to charge a lower price. Exactly the same is true for the slightly more sophisticated notion of an indifference price.

To test the effectiveness of delta hedging we will consider the following very concrete situation.

Example 1: A stock follows geometric Brownian motion with drift $\mu = 0.05$, volatility $\sigma = 0.2$ risk free rate $r = 0.03$. The current stock price is \$100.

A speculator, Jancis, wishes to purchase a call option with strike \$110 and maturity 0.5. She contacts three banks and ask for quotes. The traders at each bank have been told by their boss that they can only make a loss 1% of the time, so they quote the 99%-profit-price for their strategy.

What prices are quoted by the following three banks?

(A) Bank A's trader, Amy, performs no hedging. She simply:

- Sells the call
- Puts the money paid by the customer into a risk free account option
- Crosses her fingers.

(B) Bank B's trader, Bob, uses the a stop loss strategy. He:

- Sells the call
- Each day he checks if the stock price is greater than the strike. If so he ensures that he is holding one unit of the stock, otherwise Bob ensures that he is holding zero units of the stock.
- Bob puts any remaining cash or debt into a risk free account.
- At maturity, he is holding the stock if and only if he is going to have to deliver it to the customer.

(C) Bank C's trader, Cesare, has the benefit of an MSc in mathematical finance. He:

- Sells the call
- Each day he uses the program he wrote in FM06 to compute the delta of a stock to compute the current delta. He ensures that he is holding precisely delta units of the stock.
- Cesare puts any remaining cash or debt into a risk free account.
- At maturity he delivers the stock to the customer if required and sells any outstanding stock holding.

The calculations for cases A and B as exercises, we will show how to perform the calculation for case C. Not unexpectedly, it turns out that Bank C is able to charge the lowest price.

This means that our speculator, Jancis, will be happy because she is able to get a lower quote for the option than would be possible if no one was following the delta hedging strategy. Cesare, the trader at Bank C, is also happy because 99% of the time he will make a profit.

Why doesn't Jancis simply delta hedge herself? Because she doesn't have the time. She is a speculator interested in pursuing her own (risky) investment strategies and is happy to pay for the service provided by Cesare who, in effect, is delta hedging on her behalf but at a price.

6.1.1 Our plan of campaign

To compute the 99% profit price we will begin by assuming the trader charges \$0. We will then simulate a large number of price paths in the $\mathbb{P}$ measure.

At each time point, we will follow the recipe given in the strategy to compute how much the investor puts into stock and into the bank. We will also compute any accrued interest. By working through to the final time point we will be able to compute the final bank balance—i.e. the trader's profit for every price path.

If we then compute the first percentile of the profit across the price paths, x, $-e^{-rT}x$ is an unbiased estimator of the 99%-profit-price. To see this, simply notice that if the trader had charged $-e^{-rT}x$ instead of 0 and put this in the bank their bank balance at each time t would be $-e^{r(t-T)}x$ higher than it was the last time. Therefore their final bank balance in the worst one percent of cases will be $-x + x = 0$.

6.1.2 Calculating the cashflows

Let us write down difference equations which allow us to calculate the bank balance of the trader at each point in time. We can then code up these difference equations in Matlab.

In our mathematical formulae we will use the following notation:

- S_0, K, σ, μ, r and T are the usual suspects.
- n is the number of time steps, so we have $n+1$ time points numbered from 0 to n.
- $\delta t = T/n$.
- P for the price paid by the customer.
- S_j for the stock price at time point j.
- Δ_j for the Black Scholes delta of the stock at time point j.
- b_j for the bank balance at the end of time point j.

We'll sometimes use more descriptive names in our code.

At time 0: the customer pays P; the trader purchases Δ_0 stocks. Therefore the trader's bank balance at time 0 is:

$$b_0 = P - \Delta_0 S_0 \tag{6.1}$$

At intermediate times t where $t \neq 0$ and $t \neq n$: the money in the bank earns interest; the trader purchases $\Delta_t - \Delta_{t-1}$ stocks. Therefore the bank balance at time i is:

$$b_t = e^{r\delta t} b_{t-1} - (\Delta_t - \Delta_{t-1}) S_t \tag{6.2}$$

We can use this recurrence relation to compute the bank balance at all future times.

At the final time: the money in the bank earns interest; the trader sells Δ_{n-1} shares; the trader fulfils the call option contract at cost

$$\max\{S_n - K, 0\}.$$

So the bank balance at the final time is:

$$b_n = e^{r\delta t} b_{n-1} + (\Delta_{n-1}) S_n - \max\{S_n - K, 0\} \tag{6.3}$$

6.1.3 Simulating delta hedging in MATLAB

The first part of our implementation is to write a function that computes the Black–Scholes delta. We have in fact decided to write our black scholes pricing function in such a way that it returns the price, the delta and the gamma all at once. This makes it quite a convenient function to use without actually adding much complexity.

```
function [ price, delta, gamma ] = ...
    blackScholesCallPrice( K, T, S0, r, sigma )
% Computes the price of an option using
% the Black Scholes formula.
% (The parameter order is contract terms
% then market data.)

numerator = log(S0./K) + (r+0.5*sigma.^2).*T;
denominator = sigma.*sqrt(T);
d1 = numerator./denominator;
d2 = d1 - denominator;
price = S0 .* normcdf(d1) - exp(-r.*T).*K.*normcdf(d2);

delta = normcdf(d1);
gamma = normpdf(d1) ./ (S0.*denominator);

end
```

With this preliminary, we are ready to write the code to simulate delta hedging and hence compute the 99% profit price. We begin by writing a function to simulate the profit and loss due to delta hedging when one charges a fixed amount. Here is how the function is declared.

```
function [ finalBankBalance] = simulateDeltaHedging( ...
      chargeToCustomer , ...
      K, T, ...
      S0, r, mu, sigma, nPaths, nSteps  )
```

Let us now write the body of this function. We begin by generating a large number of stock price paths. Notice that it is crucial that we are simulating in the $\mathbb{P}$ measure.

```
dt = T/nSteps;
times = dt:dt:T;
paths = generateBSPaths( T, S0, mu, sigma, nPaths, nSteps );
```

We next use equation (**??**) to compute the bank balance at time 0.

```
[~,delta] = blackScholesCallPrice(K,T,S0,r,sigma);
stockQuantity = delta;
cost = stockQuantity .* S0;
bankBalance = chargeToCustomer-cost;
```

The ~ symbol simply means that we have chosen to ignore the first returned value of the `blackScholesCallPrice` function namely the call price. As you can see we have chosen to write the code using long variable names. The code essentially re-derives (**??**). In my view this makes the code easy to read without needing to cross-refer to a mathematical document which describes what is going on.

We now loop through all the intermediate times using equation (6.2).

```
for t=1:(nSteps-1)
    S = paths(:,t);
    timeToMaturity = T-times(t);
    [~,delta] = blackScholesCallPrice(...
        K,timeToMaturity,S,r,sigma);
    newStockQuantity = delta;
    amountToBuy = newStockQuantity - stockQuantity;
    cost = amountToBuy .* S;
    bankBalance = exp(r*dt)*bankBalance - cost;

    stockQuantity = newStockQuantity;
end
```

Our code doesn't copy (6.2) to the letter. We have made a few changes.

Firstly, in our mathematics we have used separate variables b_t and b_{t-1} for each time. In our code we have just `bankBalance`. Reusing the same variable saves a little memory. In mathematics you can't "reuse" variables in this way because an equation like `bankBalance = bankBalance+1` doesn't make mathematical sense.

Secondly in our mathematics we have used the symbols Δ_t and Δ_{t-1} but in our code we have introduced variables `newStockQuantity` and `stockQuantity`. These help explain where the equation come from and will prove to be useful when you generalize the code to strategies other than delta hedging.

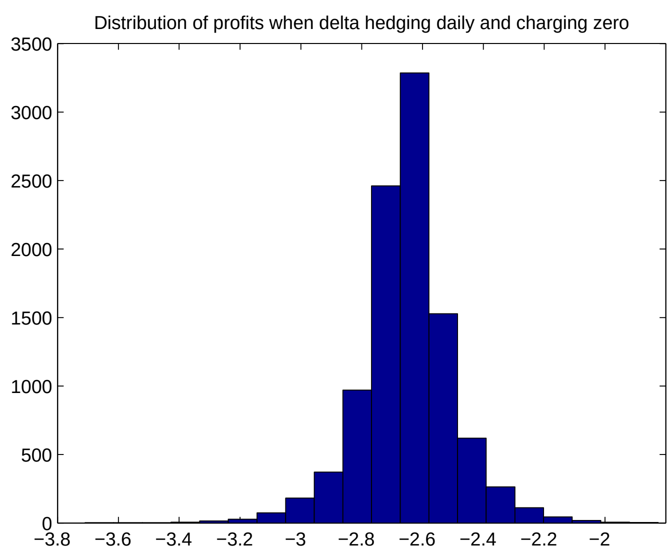
Finally we will use (6.3) to compute the final bank balance. This allows us to compute the profit and loss.

```
S = paths(:,nSteps);
stockValue = stockQuantity .* S;
liability = max(S-K, 0);

bankBalance = exp(r*dt)*bankBalance + stockValue - liability;

finalBankBalance = bankBalance;
```

This completes the `simulateDeltaHedging` function.

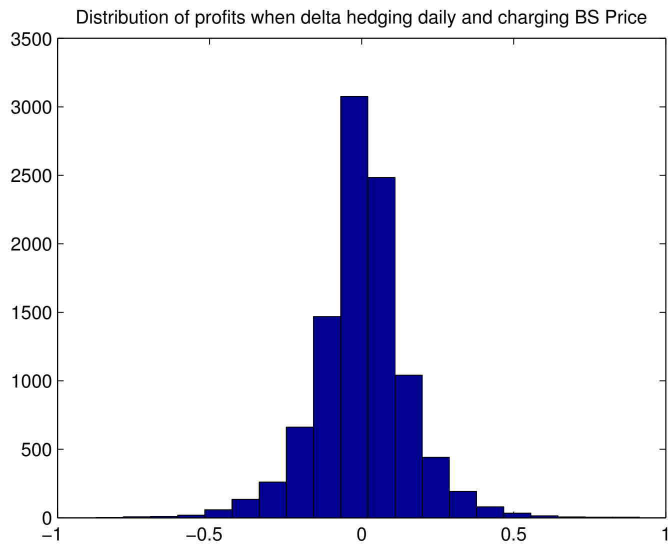
6.1.4 Computing the 99% profit price

We can now compute the 99% profit price. To do this we simply plot a graph of the discounted profit and loss if we charge zero. We can then read off the 1st percentile to see how much we need to charge. This can be automated using the `prctile` function in MATLAB. Doing this is left as an exercise.



6.2 Delta hedging - Theory and Practice

Let us suppose that instead of charging the 99% profit price, the trader had charged the Black-Scholes price. We assume that the trader still delta hedges in continuous time. The distribution of profit and loss is then as shown below. As one can see, the Black-Scholes price will ensure that on average one roughly breaks even.



However, discrete time delta hedging is definitely a risky strategy: this histogram is fairly wide. As you can check, the smaller δt is, the narrower the histogram.

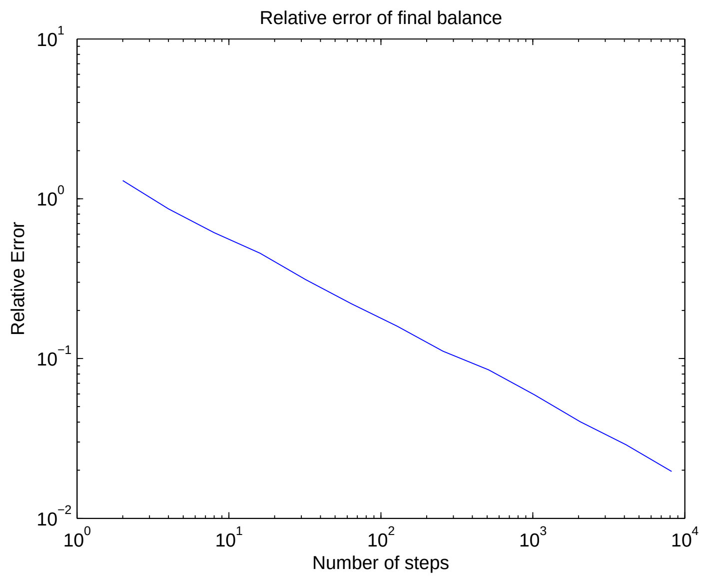
Probably all you have ever seen proved is that if you delta hedge in continuous time, then you will exactly break even.

A slightly more elaborate theory tells us that if a trader charges the Black Scholes price then performs the delta hedging trading strategy at `n` discrete times the expected profit should tend to zero as $n \to \infty$ and moreover the standard deviation of the profits should tend to zero. This can be proved by combining the theory on discrete time approximations to stochastic differential equations with Black-Scholes result.

Proving this is beyond the scope of this course, but we can at verify numerically whether it it is true.

To do this, we define the `error` of the strategy to be the profit or loss in each particular case. We define the relative error to be the root mean squared error divided by the initial option price. The theory above suggests that the relative error of delta hedging a call option at the Black-Scholes price should tend to zero as $\delta t \to 0$.

We plot a log-log plot of the relative error against the number of time steps n.



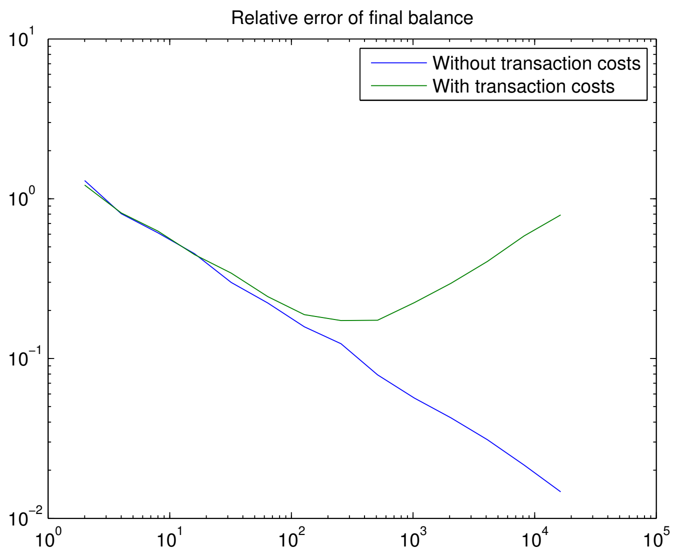
From the graph it appears from the graph that the relative error is of order $n^{-\frac{1}{2}}$. This can be proved using the theory of discrete time approximations to stochastic differential equations.

The fact that delta hedging cannot be performed in continuous time is just one possible criticism of the Black–Scholes theory. Using our simulations it is possible to quantify the importance of other assumptions in the theory. For example, how important our transaction costs?

A standard way to model transaction costs is to consider the bid-ask spread. We will model the ask price as following geometric Brownian motion as usual. We will model the bid price as being $(1-\epsilon)S_t$ for some ϵ. You can look at market data for the stock price to choose a sensible value of ϵ. This model is called "proportional transaction costs".

As well as the bid-ask spread, exchanges often levy charges. If these are proportional to the transaction cost, they can be modelled in the same way. Financial mathematicians use the phrase "transaction costs" to mean both the bid-ask spread and charges.

What happens if a trader follows the delta hedging trading strategy in the face of proportional transaction costs. Adapting the difference equations to include transaction costs is left as an exercise. Here is a plot of the result.



This shows that delta hedging is helpful if one rehedges reasonably infrequently. However, if one attempts to rebalance the portfolio too often, transaction costs begin to dominate.

In practice, delta hedging converges slowly. Once transaction costs are taken into account, the delta hedging strategy is not enough to explain observed call option prices. We conclude that, important though the idea of delta hedging is, there must be something more going on in the market.

6.3 Gamma Hedging

A trader can, if they want, buy exchange traded options to manager their own risk.

Why would they want to do this rather than just delta hedge? As we have seen, in practice delta hedging alone cannot explain market prices: it is cheaper to buy exchange traded options than to delta hedge.

But if even a derivatives trader cannot make money by delta hedging, where does this leave the Black-Scholes theory? The answer to this is that a trader does not simply buy a single call or put option. Throughout the day they regularly trade in options at different strikes and possibly exotic options too. Buy matching up people who want to buy and sell options, the trader is able to make a profit throughout the day because of the bid ask spread. Moreover they are able to meet market demand for less popular products such as out of the money options and exotics.

Where hedging comes in is that it means that the trader doesn't need to precisely match there buying and selling activity. At the end of the days trading, the trader will be left with some net risky position. But the trader can then hedge away this risk by buying stock, or if they prefer exchange traded options. Notice that the trader does not need to hedge each trade they have performed individually. They can rehedge an entire portfolio with just a couple of trades. This means that traders achieve an economy of scale, so banks are not as adversely affected by transaction costs as an individual trying to hedge their own portfolio.

A standard trading strategy is to *gamma hedge* a portfolio. This means to buy stock and exchange traded options in order to ensure that the portfolio has an overall delta and gamma of zero. Recall that the gamma is the second partial derivative of the price with respect to a change in the stock price.

To understand the motivation for the gamma hedging strategy, recall that when we delta hedge, we ensure that our portfolio consisting of

- Our stock holding
- Our liability
- Our bank balance

has an aggregate delta of zero. Intuitively this means that if the stock price changes, are portfolio's aggregate value won't change much—it won't change to first order in the stock price. If we could ensure that our portfolio had a delta of zero *and* a gamma of zero, we could ensure that it won't change to second order in the stock price. Presumably this will be a better strategy.

So, in detail the gamma hedging strategy is to write a portfolio of options and then at fixed time intervals, purchase stock and an exchange traded option so that our portfolio consisting of:

- Our stock holding
- Our holding in exchange traded options
- Our liability (the exotic option)
- Our bank balance

has an aggregate delta of zero and an aggregate gamma of zero.

For simplicity, let us suppose that the hedged portfolio consists of a single option and compute how well the gamma hedging strategy performs in this case.

Let us write the mathematical difference equations we need to solve to compute the effectiveness of gamma hedging.

At time point j we have the following variables. The option we are writing has strike $K1$. Its Black Scholes price is P^1_j, its delta is Δ^1_j and its gamma is Γ^1_j. The hedging option has price P^2_j, delta Δ^2_j and gamma Γ^2_j. The stock price is S_j. The trader's bank balance is b_j The trader's stock holding is q^S_j The trader's holding in option 1 is $q^1_j = -1$ since they have written the option. The trader's holding in option 2 is q^2_j.

By linearity of partial derivatives, the delta of our portfolio is:

$$q^S_j + \Delta^1_j q^1_j + \Delta^2_j q^2_j$$

The gamma of our portfolio is:

$$\Gamma^1_j q^1_j + \Gamma^2_j q^2_j$$

We know that $q^1_j = -1$. We require the portfolio to be delta and gamma neutral hence:

$$q^S_j - \Delta^1_j + \Delta^2_j q^2_j = 0$$
$$-\Gamma^1_j + \Gamma^2_j q^2_j = 0$$

We deduce that

$$q^2_j = \frac{\Gamma^1_j}{\Gamma^2_j}$$
$$q^S_j = \Delta^1_j - \Delta^2_j q^2_j$$

These recurrence relations allow us to simulate the strategy.

Note that we have solved two linear equations to get a gamma neutral portfolio. This is why we need to trade in two products (the stock and the hedging option) in order to obtain a gamma neutral portfolio.

One problem with our equations is that if Γ^2_j becomes very small then the equation

$$q^2_j = \frac{\Gamma^1_j}{\Gamma^2_j}$$

will start to cause numerical errors. This will happen if option 2 is a long way into the money or a long way out of the money we will start to see numerical errors. When we implement the gamma hedging strategy in Matlab we choose to modify our strategy to be to try to choose q^2_j using the above formula, but cap the value at 100 or -100 to avoid numerical errors.

Another problem we will face is that the gamma is the second derivative of the price and at maturity, the payoff of a call option is not differentiable. For this reason the gamma can sometimes tend to infinity near maturity. This too can lead to numerical errors. So in our simulation of gamma hedging, therefore, we stop the simulation a little before maturity and calculate the "market price" of the portfolio using the Black Scholes Formula. To accommodate this, introduce a variable `sellTime` to indicate when we sell our portfolio.

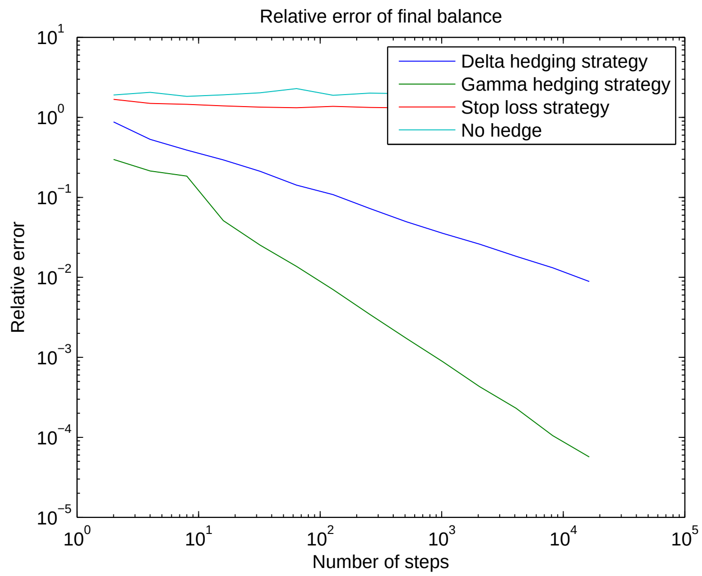
To test the effectiveness of gamma hedging we will see what happens if we charge the Black-Scholes price for option 1 at the start of trading and then liquidate our portfolio at the `sellTime`. Our aim is to generate a log-log plot of the "relative error" of this strategy exactly as for the delta hedging strategy. Here is the MATLAB code to do this.

```
dt = sellTime/nSteps;
times = dt:dt:sellTime;
paths = generateBSPaths( sellTime, S0, mu, sigma, nPaths, nSteps );

[~,delta1, gamma1] = blackScholesCallPrice(K1,T,S0,r,sigma);
[p2,delta2, gamma2] = blackScholesCallPrice(K2,T,S0,r,sigma);
```

```
option2Quantity = max(min(gamma1./gamma2,100),-100);
stockQuantity = delta1 - option2Quantity .* delta2;
stockCost = stockQuantity .* S0;
optionCost = option2Quantity .* p2;
bankBalance = chargeToCustomer -stockCost-optionCost;

for t=1:nSteps

    S = paths(:,t);
    timeToMaturity = T-times(t);
    [p1,delta1,gamma1] = blackScholesCallPrice(K1,timeToMaturity,S,r,sigma);
    [p2,delta2,gamma2] = blackScholesCallPrice(K2,timeToMaturity,S,r,sigma);
    newOption2Quantity = max(min(gamma1./gamma2,100),-100);
    newStockQuantity = delta1 - newOption2Quantity .* delta2;

    stockCost = (newStockQuantity - stockQuantity).* S;
    optionCost = (newOption2Quantity - option2Quantity).* p2;
    bankBalance = exp(r*dt)*bankBalance - stockCost - optionCost;

    stockQuantity = newStockQuantity;
    option2Quantity = newOption2Quantity;

    marketValue = bankBalance + stockQuantity.*S - p1 + option2Quantity.*p2;

end
```

In our delta hedging simulation, we had two special cases, time zero and maturity. In this simulation, because we stop the simulation before maturity, we have one less special case. Note also the `min` and `max` in the computation of `option2Quantity`. This prevents numerical errors.

We can now look at a log-log plot of the relative errors of the various trading strategies against the number of time steps, n.



We conclude that Gamma hedging allows one to achieve the price predicted by Black Scholes with much less rehedging. This means that Gamma hedging allows one to achieve a price much closer to the Black Scholes price when there are transaction costs in the model. It would appear that the rate of convergence of the gamma hedging strategy appears to be order n^{-1} and this can be proved mathematically.

6.4 Indifference pricing

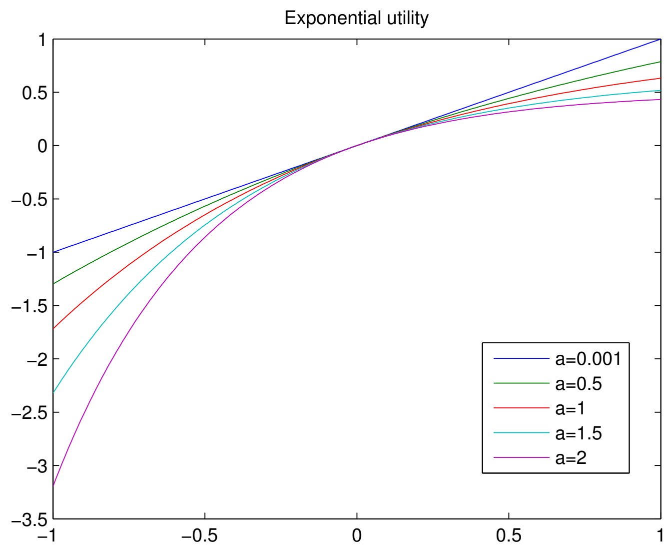
As we mentioned earlier, the concept of the 99% profit price is my own invention and it is in some ways a rather silly idea. Surely we should care about how much loss we may make in the 1% of bad cases? Indifference pricing gives a way of taking this into account. (All the code that we have just written to compute the 99% profit price will be reused to compute indifference prices, so we have not been wasting our time.)

To perform indifference pricing, one must first choose a *utility function* that describes one's risk preferences. A utility function $u : \mathbb{R} \longrightarrow \mathbb{R}$ is a function which sends the payoff to some measure of the value that an individual places in that payoff. Most people prefer making money to losing it, so utility functions are usually increasing. Most people are risk averse, they value making money less highly than they hate losing it.

An example of a utility function is the exponential utility function is:

$$u(x) = \frac{1 - e^{-ax}}{a}$$

The parameter a measures risk aversion. Some plots of this utility function for different values of a are shown below.



Suppose a trader is selling a financial product and

- Their utility function is u
- They plan to follow a particular strategy
- They believe the market follows some given stochastic process

Definition. *The indifference price given the strategy is the amount that they would have to charge so that their expected utility remains is the same whether or not they enter into the trade.*

The above definition which includes a strategy in the definition is non-standard. The indifference price (ignoring the strategy) is the price they would charge assuming that they choose the best possible strategy. We are ignoring all technical details about whether these prices are actually well defined.

In practice, finding the optimal strategy may be prohibitively difficult, so in practice it is very useful to be able to calculate the indifference price given the strategy one intends to pursue.

We note that the indifference price is quite different to the "risk-neutral" price that you have studies in FM02. Here are some key differences:

- The indifference price is subjective
- It depends upon your utility function
- It depends on your beliefs
- It depends upon the strategy you wish to follow

- We should really compute the total position of the trader including when computing the indifference price.
- e.g. if you have already sold a call, you may be more willing to buy a call as it will remove the risk from your books.
- You would quote different indifference prices for buying and for selling.

In real markets with transaction costs etc. there is no single "risk-neutral" price. One must use alternative pricing methods such as indifference pricing. However, if the theoretical model is sufficiently close to the real world model, it may well be that a real world indifference price is quite close to a theoretical risk-neutral price.

Let us compute the indifference price for a trader using a delta hedging strategy. We assume they have zero assets before the trade. We assume the utility function is the exponential utility function. We assume they invest any excess capital in the bank.

Let p be the payoff if the trader doesn't charge. So $e^{rT}P + p$ is the payoff if the trader charges P. The expected utility if they don't trade is 0. Therefore we must choose P to solve:

$$E(u(e^{rT}P + p)) = 0 \tag{6.4}$$

This is easy to do in MATLAB as MATLAB comes with a function called `fsolve` to solve equations numerically. For example, suppose we want to find a solution to $\sin(x) + \exp(x) = 0$ and we have an initial guess that $x = 0.1$ might be near a solution we would use the following code.

```
function ret=toSolve(x)
  ret = sin(x) + exp(x);
end
guess = 0.1;
solution = fsolve( @toSolve, guess );
assert( abs(toSolve( solution ) )<0.001);
```

The function `fsolve` can be used to find solutions of equations in MATLAB and is often useful. For example, you can use it to compute the implied volatility of an option. This is left as an exercise.

In our case we wish to solve (6.4) for P. This motivates the following code that computes the indifference price for a particular option given that we delta hedge with a certain frequency.

```
function indifferencePrice= computeIndifferencePrice( a )
%  Compute the indifference price when the trader uses
%  The delta hedging strategy


K = 110;
```

```
T = 0.5;
S0 = 100;
sigma = 0.2;
mu = 0.05;
r = 0.03;

nPaths = 10000;
nSteps = 133;

payoff = simulateDeltaHedging(0,K,T,S0,r,mu,sigma,nPaths,nSteps);

% Define the utility function
function utility=u( x )
    utility = (1-exp(-a.*x))/a;
end

% Write a function that computes the expected utility
% given the price
function ret=expectedUtility( price )
    ret = mean( u( exp(r*T)*price + payoff ) );
end

% Initial guess
[blackScholesPrice,~] = blackScholesCallPrice(K,T,S0,r,sigma);
% Use fsolve to find the solution
indifferencePrice = fsolve( @expectedUtility, blackScholesPrice );
end
```

6.5 Further Reading

See [2] for a discussion of delta and gamma hedging. The theory in this chapter can also be found in the chapter on delta hedging in [1]. Simply ignore the C++ code discussed in this chapter.

Bibliography

[1] John Armstrong. *C++ for Financial Mathematics*. CRC Press, 2017.

[2] M. S. Joshi. *The concepts and practice of mathematical finance*, volume 1. Cambridge University Press, 2003.