FMO6 — Web: https://tinyurl.com/ycaloqk6 Polls: https://pollev.com/johnarmstron561

Lecture 5

Dr John Armstrong

King's College London

August 22, 2020

Delta Hedging

Lecture outline

- Last week we learned how to simulate Black Scholes paths
- We applied this to risk neutral pricing
- Today we will use our simulation to test how delta hedging works in practice.
- In the second half of the lecture we learn about gamma hedging and indifference pricing.

Why trade in options at all?

- Speculator
- Hedger
- Banker

Why trade in options at all?

- Speculator: Trades in options because they have an opinion on the market and wish to increase *leverage*.
- Hedger: Trades in options because they wish to minimize the risk of their existing position.
- Banker: Wishes to provide a service at a cost. They *manufacture* the options for speculators and hedgers to buy.

The 99%-profit-price

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 non-standard. It is my invention.
- It is a simplified version of the "indifference price" that we will discuss later.
- It depends 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.

Shopping around

- A stock follows the Black-Scholes model 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?

No hedging

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.

Stop Loss Strategy

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.

Delta Hedging Strategy

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.

Expected conclusions

- You will perform the calculations for cases A and B as homework.
- We will perform the calculation for case C this lecture.
- It turns out that ... charges the lowest price.
- Why is Jancis (the speculator) happy?
- Why is ... happy?

Solution strategy

Our strategy to answer the question is simple:

- Begin by assuming the trader charges \$0.
- Simulate a large number of price paths in the . . . measure.
- Follow the recipe given in the strategy to compute the cashflows at every time for every price path.
- Compute the final bank balance - i.e. the trader's profit for every price path.
- Compute the first percentile of the profit across the price paths, x.
- $-e^{-rT}x$ is an unbiased estimator of the 99%-profit-price is

Proof

This is one of those so-obvious-you-might-struggle-to-prove-it statements. So here is as formal proof to justify the claim that it is obvious.

- Suppose that you charge $-e^{-rT}x$ instead of 0 and put this in the bank.
- Your bank balance at each time t will be $-e^{r(t-T)}x$ higher than it was the last time.
- Therefore your final bank balance in the worst one percent of cases will be $-x + x = 0$.

Step 1, Simulate price paths

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

- We wrote the code for this last week, so we reuse it.
- We simulate paths in the $\mathbb{P}$ measure because we wish to simulate real world behaviour.

Generate paths

```
% Generate random price paths according to the black scholes model
% from time 0 to time T. There should be nSteps in the path and
% nPaths different paths
function [ S, times ] = generateBSPaths( ...
    T, S0, mu, sigma,nPaths, nSteps  )

dt = T/nSteps;
logS0 = log( S0);
eps = randn( nPaths, nSteps );
dlogS = (mu-0.5*sigma^2)*dt + sigma*sqrt(dt)*eps;
logS = logS0 + cumsum( dlogS, 2);
S = exp(logS);
times = dt:dt:T;

end
```

Revision

- What does `cumsum` mean?
- What process does the log of the stock price follow?
- How do we translate this into a difference equation?

Naming our variables in formulae

In our mathematical formulae we will write

- 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 use descriptive names in code.

Cashflows at time 0

- The customer pays P.
- The trader purchases Δ_0 stocks.
- The bank balance is then

$$b_0 = P - \Delta_0 S_0$$

Step 2, compute the bank balance at time 0

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

- We are assuming here that our `blackScholesCallPrice` function returns two values, the call price and the delta.
- The ~ simply means that we have chosen to ignore the first returned value, namely the call price.

Black Scholes formulae

```
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
```

Remarks

- Our pricing function also computes the Greeks for convenience
- The code is all vectorized, so we can find the Greeks for an entire vector of price paths at once.

Cashflows at time point t.

- The money in the bank earns interest
- The trader purchases $\Delta_t - \Delta_{t-1}$ stocks
- The bank balance is then:

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

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

Step 2, compute the bank balance at intermediate times

```
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
```

- In our mathematics we have b_t and b_{t-1} in code we have just `bankBalance`. Reusing the same variable saves a little memory.
- In our mathematics we have Δ_t and Δ_{t-1} the variables `newStockQuantity` and `stockQuantity` are useful when one generalizes to strategies other than delta hedging.

Cashflows at time point n.

- The money in the bank earns interest
- The trader sells Δ_{n-1} shares.
- The trader fulfils the call option contract at cost

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

- The bank balance is then:

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

Step 3, compute the final bank balance

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

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

finalBankBalance = bankBalance;
```

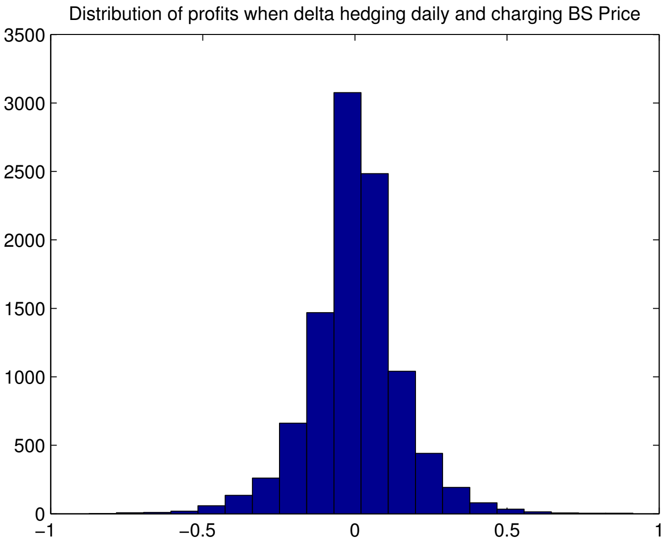
Wrap it in a function

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

% ... code above ...

end
```

Exercises

★ Using the code given, compute the 99%-profit-price for the delta hedging strategy.

★ Suppose that the trader had charged the Black–Scholes price, plot a histogram of the trader's final profit


Distribution of profits when delta hedging daily and charging BS Price
3500
3000
2500
2000
1500
1000
500
0
−1
−0.5
0
0.5
1

Theory

If a trader:

- Charges the Black Scholes Price
- Performs the delta hedge trading strategy at n discrete times

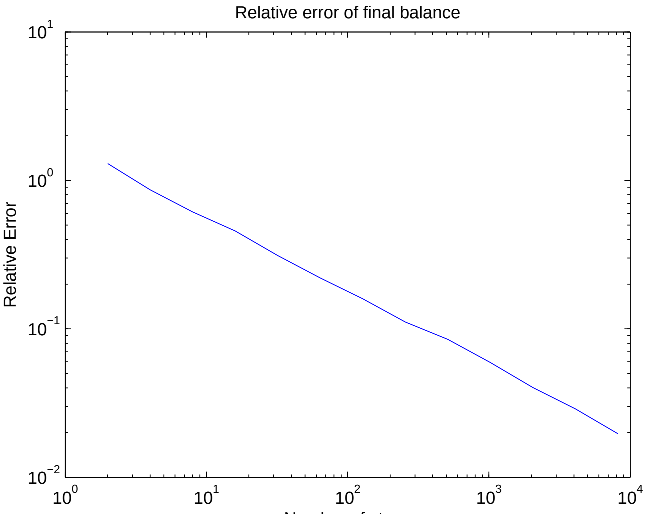
Then

- The expected profit should tend to zero (it does at $O(\delta t)$)
- The standard deviation of the profits should tend to zero (it does at $O(\delta t^{\frac{1}{2}})$)

Probably all you have ever seen proved is that if you delta hedge in continuous time, then you will exactly break even.
If the discrete time result wasn't true, however, then the continuous time theory would be worthless.

Rate of convergence

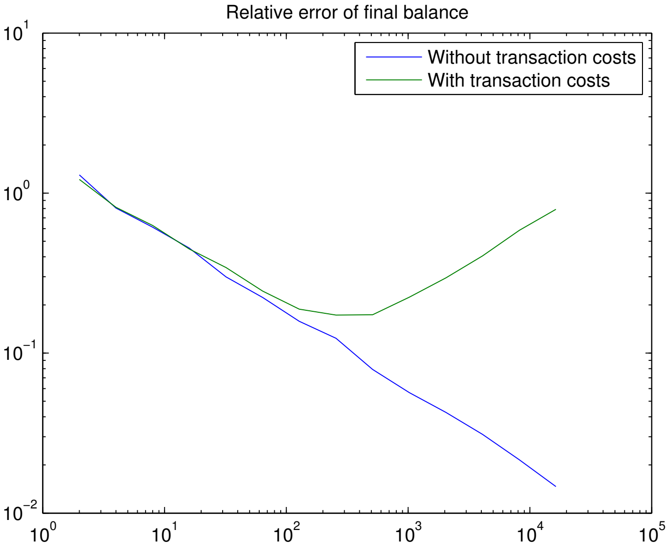
Log-log plot of relative error against n.



Interpretation

- We define the error of the strategy to be the profit or loss in each particular case.
- The root mean squared error gives an overall measure of the error over all the scenarios.
- We define the relative error to be the root mean squared error divided by the initial option price. This is useful because it allows you to see whether the error is relatively large or small.
- It appears from the graph that the relative error is of order $n^{-\frac{1}{2}}$.
- We'll be able to prove this when we discuss the Euler method for approximating SDE's.

With 0.1% transaction costs



- A standard way to model transaction costs is to consider the bid ask spread.
- We model the ask price as following the Black Scholes process S_t above.
- We 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.

Conclusions

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

Gamma Hedging

Gamma Hedging

- You work for a bank and wish to write an exotic option
- You wish to hedge to reduce your risk
- You know that delta hedging isn't a good enough strategy to allow you to offer a market price.
- You could trade in other options on the stock to reduce your risk further!
- Other people buy options to reduce risk, so why shouldn't you.
- We've just shown it is apparently cheaper to buy options on the market to reduce risk than to simply delta hedge, so actually delta hedging is an irrational strategy!

Intuition

- 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.

Gamma Hedging Strategy

- Write an exotic option or a portfolio of options.
- 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.

Constant trading of options

- In practice a trader constantly buys and sells options. They are matching buyers and sellers and so making a profit on the bid-ask spread.
- If customers want unusual strike prices or maturities or exotic options, they cannot be exactly matched. In this case the trader can still approximately offset trades, but they will be left with a portfolio that needs to be hedged.
- This can be delta hedged using the stock alone, or gamma hedged using an exchange traded option.
- Note that the trader can rehedge an entire portfolio with just a couple of trades. They achieve an economy of scale over an individual trying to re-hedge a single option that allows them to worry less about transaction costs.

Objective

- We would like to simulate gamma hedging to see how well it works
- We are only interested in getting a feel for the strategy so we assume for simplicity that:
  - Our “exotic” option is simply a call option with strike $K1$ and maturity T.
  - We hedge using a call option with strike $K2$ and maturity T. The market price of this option is assumed to come from the Black Scholes formula.
- A real trader is hedging a constantly changing portfolio of options, so our trading simulation isn't utterly realistic.
- It does allow us to see that gamma hedging is more effective than delta hedging alone.

Gamma Hedging Symbols

- At time point j as before 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.
- Stock price is S_j.
- Our bank balance is b_j
- Our stock holding is q^S_j
- Our holding in option 1 is $q^1_j = -1$ since we have written the option.
- Our holding in option 2 is q^2_j.

Computing the new quantities

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

$$q_j^S + \Delta_j^1 q_j^1 + \Delta_j^2 q_j^2$$

The gamma of our portfolio is:

$$\Gamma_j^1 q_j^1 + \Gamma_j^2 q_j^2$$

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

$$q_j^S - \Delta_j^1 + \Delta_j^2 q_j^2 = 0$$

$$-\Gamma_j^1 + \Gamma_j^2 q_j^2 = 0$$

We deduce that

$$q_j^2 = \frac{\Gamma_j^1}{\Gamma_j^2}$$

$$q_j^S = \Delta_j^1 - \Delta_j^2 q_j^2$$

Problem 1

- Note that we have the formula

$$q_j^2 = \frac{\Gamma_j^1}{\Gamma_j^2}$$

- If Γ_j^2 becomes very small, as will occur 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.
- For this reason we modify our strategy to be to try to choose q_j^2 using the above formula, but cap the value at 100 or -100 to avoid numerical errors.

Problem 2

- The gamma is the second derivative of the price
- At maturity, the payoff is not differentiable
- For this reason the gamma can sometimes tend to infinity near maturity. This too can lead to numerical errors.
- When simulating gamma hedging, therefore, we stop the simulation a little before maturity and calculate the "market price" of the portfolio using the Black Scholes Formula.
- We introduce a variable `sellTime` to indicate when we sell our portfolio.

Objective of the simulation

- If we were paid the Black Scholes price, we expect the final market price at the sell time to be near zero.
- We wish to generate log log plot of the relative error of this market price.

Initializing the portfolio

```
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;
```

Simulating till the sell time

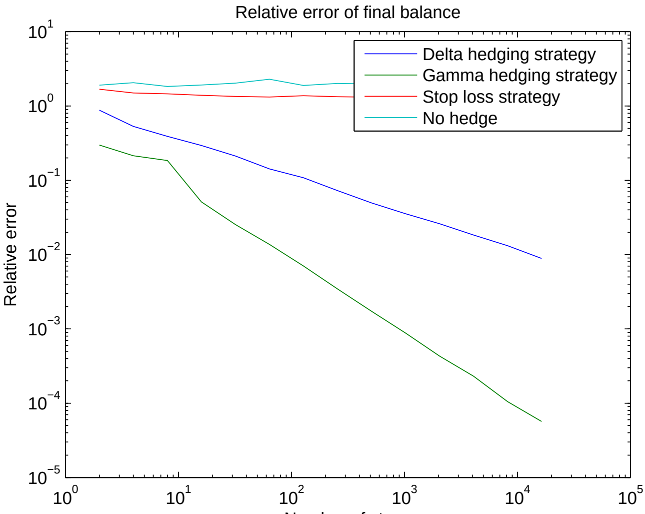
```
for t=1:nSteps

    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
```

Comments

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

Results

Log-log plot of relative error against n.



Conclusions

- 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.
- The rate of convergence of the gamma hedging strategy appears to be order n^{-1}

Indifference pricing

Utility functions

- 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.
- Utility functions are usually increasing because most people agree profit is better than loss.
- Most people are risk averse, they value making money less highly than they hate losing it. For this reason, utility functions are usually concave functions.

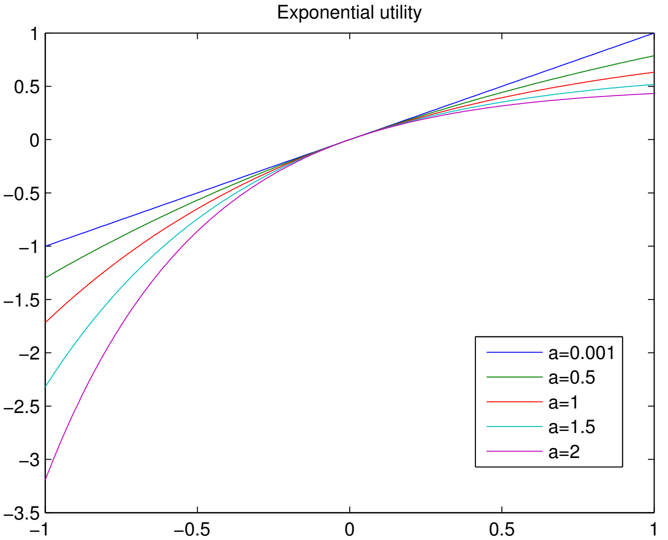
Example

The exponential utility function is:

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

The parameter a measures risk aversion.

Exponential utility



Indifference pricing

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.

Remarks

- 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.

Computing the indifference price

Assumptions

- In our example, let us suppose the trader uses a delta hedging strategy.
- We assume they have zero assets before the trade.
- We assume the utility function is the exponential utility function.

Calculation:

- 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:

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

Solving equations using fzero

- MATLAB comes with a function called `fsolve` to solve equations numerically.
- 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.

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

Computing the indifference price

We wish to solve the following equation for the price P, given random payoffs p

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

```
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 );
```

Exercises

Worksheet 5