

Dynamic Hedging Strategies

In this article, the authors use the Black-Scholes option pricing model to simulate hedging strategies for portfolios of derivatives and other assets.

by Simon Benninga and Zvi Wiener

A *dynamic hedging strategy* typically involves two positions:

- A static position in a security or a commitment by a firm. For example:
 - A financial institution has written a call on a stock or a portfolio; this call expires some time in the future and is held by a counterparty which has the choice of either exercising it or not.
 - A U.S. firm has committed itself to sell 1000 widgets at some defined time in the future; the receipts will be in German marks.
- An offsetting position in a financial contract. Typically, this counter-balancing position is adjusted when market conditions change; hence the name *dynamic hedging strategy*:

To hedge its written call, the issuing firm decides to buy shares of the underlying stock or portfolio. The number of shares purchased at time t will depend on the price of the underlying stock at t and on the amount of time remaining until the expiration of the call. Another way of viewing this is that the amount of stock held against the call position depends on the probability that the option will be exercised.

To hedge its anticipated receipt of German marks, the firm might enter into a forward contract. This would be a static hedge position. On the other hand, it might buy a put option on Deutschmarks (DM), which would give it the right to sell DM for a given dollar price; this is still a static hedge position, though somewhat more sophisticated. A third possibility would be to *dynamically simulate* a put on the DM position, replicating it by holding amounts of dollars and DM (these amounts changing with market conditions).

In this article we consider some simple dynamic hedging strategies and show how to simulate them in *Mathematica*. Our discussion will lead us naturally to the consideration of Value-at-Risk (VaR), a measure of possible future losses which has become increasingly popular.

1. A SIMPLE EXAMPLE

To start off, consider the following example, which we have adapted from Hull (1997): A financial institution has sold a European call option for \$300,000. The call is written on 100,000 shares of a non-dividend paying stock with the following parameters:

Current stock price = \$49

Strike price = X = \$50

Stock volatility = 20%

Risk-free interest rate r = 5%.

Option time to maturity T = 20 weeks

Stock expected return = 13%

The Black-Scholes price of this option is slightly over \$240,000 and can be calculated using the *Mathematica* program defined in our previous article:

```
In[1]:= Clear[snormal, d1, d2, bsCall];
snormal[x_] :=
  1/2*Erf[x/Sqrt[2.]] + 0.5;
d1[s_, x_, sigma_, T_, r_] :=
  (Log[s/x] + (r + sigma^2/2)*T)/
  (sigma*Sqrt[T]);
d2[s_, x_, sigma_, T_, r_] :=
  d1[s, x, sigma, T, r] - sigma*Sqrt[T];
bsCall[s_, x_, sigma_, T_, r_] :=
  s*snormal[d1[s, x, sigma, T, r]] -
  x*Exp[-(r*T)]*
  snormal[d2[s, x, sigma, T, r]];
bsPut[s_, x_, sigma_, T_, r_] :=
  bsCall[s, x, sigma, T, r] +
  x*Exp[-(r*T)] - s;
bsCall[49, 50, 0.2, 20/52, 0.05]*100000.
```

Out[1]= 240053.

It follows that the financial institution has earned approximately \$60,000 from writing the call. However, unless the financial institution *hedges* its obligation in order to offset the effects of price changes in the future, it could stand to lose much money at the call's expiration. For example, if the price of the stock at the option's expiration date is \$60, and if the call is not hedged, then the financial institution will lose \$1,000,000 (=100,000*(60-50)) at the option's expiration.

As we shall show, however, this possible future loss can

be offset by a judicious hedging strategy. In the succeeding sections we consider a number of hedging strategies.

2. A STOP-LOSS STRATEGY

One strategy which could be followed is to purchase the shares only if the calls are in the money. This strategy looks something like the following: Initially, the bank sells the calls for \$300,000. Since the current share price of \$49 is less than the call exercise price of \$50, the calls are out of the money. The bank then invests the \$300,000 in the 5% bonds for one week (the current hedging period). At the end of the week there are two possibilities:

- If current stock price is $< \$50$, reinvest the proceeds from the \$300,000 (now worth $\$300,000 \cdot \text{Exp}[0.05/20] = \$300,751$) into bonds, letting them earn a further 5% annually for another week.
- If the current stock price is $> \$50$, buy 100,000 shares, fully covering the call. Pay for these shares by selling the \$300,751 of bonds and borrowing the rest of the money (at 5% annually).

At the end of the second week:

- If the current stock price is $> \$50$, continue to hold onto the 100,000 shares, refinancing the net debt for another week at 5%.
- If current stock price is $< \$50$, sell all the shares and put the proceeds into bonds earning 5% annually.

We can write a *Mathematica* program which simulates this strategy:

```
Needs["Statistics`Master`"]
Clear[stopLoss]
numPer=20; (* number of periods *)
length = 1/52; (*length of each period *)
initPrice = 49; (* initial price of stock *)
sigma = 0.2;r = 0.05;mu = 0.15;
numOptions = 100; (* number of options written *)
strike = 50;
(* financial institution gets
  for writing calls *)
receivedForCalls = 300;

stopLoss[nn_]:=
Module[{price,stockPosition},
If[nn!=0, SeedRandom[nn],SeedRandom[]];
price[0]=initPrice;
price[t_]:=price[t]=
  price[t-1]*
  Exp[(mu-sigma^2/2)*length +
  sigma*Sqrt[length]*
  Random[NormalDistribution[0,1] ] ]/N;
stockPosition[t_]:= If[price[t]<strike,0,100];
debt[0]=stockPosition[0]-receivedForCalls;
debt[t_]:=debt[t-1]*Exp[r*length]+
  (stockPosition[t]-stockPosition[t-1])*price[t];
```

```
debt[numPer]=debt[numPer-1]*Exp[r*length] -
  stockPosition[numPer-1]*price[numPer] +
  numOptions*Max[price[numPer] - strike,0];
debt[numPer]
];
```

In this simple program: `price[t]` simulates a series of random prices, starting with `initPrice`. `stockPosition` is a function which is either 100 (if the current stock price is greater than the strike price) or 0. `debt[]` is a function which is negative if the financial institution has money invested in riskless bonds and positive if it has borrowings (so that at the terminal date negative debt means profits for the firm).

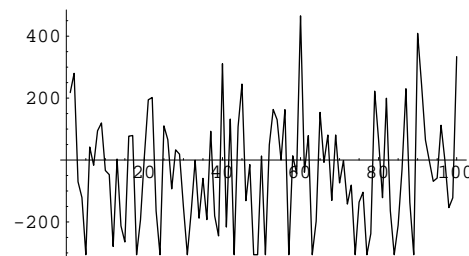
The program returns the terminal debt of the firm `debt[numPer]`. The parameter `nn` of the `stopLoss` function is used to set the random seed either to be fixed (when `nn` is different from zero) or to be chosen by the program. If we wish to be able to repeat the same results in multiple runs we use `nn` different from zero. Running this simulation 100 times produces the following result:

```
In[2]:= (* number of periods *)
numPer=20;
(*length of each period *)
length = 1/52;
(* initial price of stock *)
initPrice = 49;
sigma = 0.2;
r = 0.05;
mu = 0.15;
(* number of options written *)
numOptions = 100;
strike = 50;
(* financial institution gets
  for writing calls *)
receivedForCalls = 300;
tbl=Table[stopLoss[n],{n,1,100}];
{Mean[tbl],StandardDeviation[tbl]}

Out[2]= {-38.7918, 183.951}
```

This means that—for the particular series of random numbers generated with random seeds 1,...,100 and with weekly rebalancing of the position—the average gain of the bank is \$38,792 (with standard deviation of \$183,951). It is interesting to plot the results of all 100 of these simulations:

```
In[3]:= ListPlot[tbl, PlotJoined -> True,
  PlotRange -> All];
```



The results of the simulation are, of course, sensitive to the random numbers generated. For example, if we replace `tbl=Table[stopLoss[n+30],{n,1,100}]` by

tbl=Table[stopLoss[n+30],{n,1,100}], we get as output {-50.4517,183.548}.

3. A SIMPLE EXAMPLE OF DELTA HEDGING

In this section, we hedge this position by using a *delta hedge* (in Section 4 we shall explain why this hedge is better than the stop-loss strategy illustrated in the previous section). In a delta hedge, we hedge the call position by purchasing $\frac{\partial C}{\partial S}$ shares, where C is the call price. In the next section we shall illustrate a dynamic delta hedge, but in this chapter we concentrate on a static hedge. To hedge the position, we buy $\frac{\partial C}{\partial S} = 52,160$ shares. The hedge ratio was calculated using *Mathematica*:

```
In[4]:= Clear[delta]
delta[s_, x_, sigma_, T_, r_] :=
  snormal[d1[s, x, sigma, T, r]];
delta[49, 50, 0.2, 20/52, 0.05]*100000
```

Out[4]= 52160.5

Note that we have used an analytical equation for **delta**, instead of using the derivative: `delta[s_, x_, T_, r_, sigma_] := D[bsCall[s1,x,T,r,sigma], s1] /. s1 -> s`. The derivative takes much longer to compute. We prove this analytic formula below. The cost of purchasing these shares is $\partial C/\partial S * 49 = 52,160 * 49 = \$2,555,863$ (note the use of `AccountingForm` to format the numbers):

```
In[5]:= delta[49, 50, 0.2, 20/52, 0.05]*100000*49
AccountingForm[%, NumberPadding ->
  {"", ""}, DigitBlock -> 3]
2.55586 × 106
"2,555,863."
```

We borrow the cost of purchasing these shares. If, after 1 week, the price of the share has increased to \$52, then the call written by the financial institution is worth \$415,897:

```
In[6]:= bsCall[52,50,0.2,19/52,0.05] * 100000//
AccountingForm
Out[6]= "415897."
```

We can now do a cost-benefit analysis, assuming that the position is completely closed out (i.e., that the writer pays out the increase in the call's value to the purchaser, sells his shares, and repays the loan with interest):

- The increase in the call's value represents a *loss* for the call writer of \$115,897 = \$415,897 - \$300,000.
- On the other hand the 52,160 shares purchased by the call writer are now worth 52,160*52 = \$2,712,320, which is a gain of 52,160*(52-\$49) = \$156,480, which partially offsets this loss.
- Finally, the call writer must repay the loan.

The net gain or loss to the writer after one week is the sum of these:

Payment to buyer of call	-\$115,897
Sale of shares	\$2,712,320
Repayment of loan	-\$2,558,322
Total	\$38,101

Analytic Delta

The analytic expression for the delta of an option (its sensitivity to the changes in the underlying price) is well known. We use *Mathematica* to derive this formula. First we wish to explain why the naive `D[bsCall[s, x, T, r, sigma], s]` does not work. In the definition of `bsCall` we have used two numeric values, 2. and 0.5. As soon as a numeric value appears the whole function is numeric and not analytic, and it cannot be simplified. A slight variation on the original definition allow us to differentiate `bsCall` and `bsPut`:

```
In[7]:= Clear[snormal,d1,d2,bsCall];
snormal[x_] := Erf[x/Sqrt[2]]/2 + 1/2;
d1[s_, x_, sigma_, T_, r_] :=
  (Log[s/x]+(r+sigma^2/2)*T)/(sigma*Sqrt[T]);
d2[s_, x_, sigma_, T_, r_] :=
  d1[s, x, sigma, T, r]-sigma*Sqrt[T];
bsCall[s_, x_, sigma_, T_, r_] :=
  s*snormal[d1[s,x,sigma,T,r]]-
  x*Exp[-r*T]*snormal[d2[s,x,sigma,T,r]];
bsPut[s_,x_,sigma_,T_,r_] :=
  bsCall[s,x,sigma,T,r] + x*Exp[-r*T] - s;
```

We can now differentiate the option pricing function analytically:

```
In[8]:= delta =
  D[bsCall[s, x, sigma, T, r], s] //Simplify
Out[8]= 1/2 (1 + Erf[ (0.247487 T + 3.53553 Log[s/x]) / Sqrt[T] ] )
```

We can compare this expression with $N(d_1)$ to see that they are the same:

```
In[9]:= delta2 = snormal[d1[s, x, sigma, T, r]]//Simplify
Out[9]= 1/2 (1 + Erf[ (3.53553 (0.07 T + Log[s/x]) ) / Sqrt[T] ] )
```

While this result is an impressive illustration of *Mathematica's* power (especially to those of us who have struggled with taking the derivative of the Black-Scholes option pricing function!), the analytic expression for `bsCall` is not user friendly; for example, `bsCall[49, 50, 2/10, 20/52, 5/100]` returns a long analytic expression and not a number (this could be fixed by appending `//N` to the expression below or by making at least one of the parameters a value, for example, by writing `50.` instead of `50` or `0.2` instead of `2/10`):

```
In[10]:= bsCall[49,50.,2/10,20/52,5/100]
Out[10]= 2.40053
```

Both analytic and numeric approaches are useful in some situations, but one should distinguish between them.

4. DYNAMIC DELTA HEDGING: A SLIGHTLY MORE COMPLICATED EXAMPLE

We can make the above example more realistic by assuming that the hedge position is adjusted in each of the 20 weeks. At time $t = 0, 1, \dots, 20$:

$$\text{Number of shares sold } N_t = \partial C_t / \partial S_t * 100,000$$

PROGRAM 1:

```

numPer = 20; length = 1/52.; initPrice = 49; sigma = 0.2;
r = 0.05; mu = 0.15; numOptions = 100; strike = 50;
receivedForCalls = 300; Clear[simulation]
simulation[nn_, length_] :=
simulation[nn, length] =
Module[{path, price, hedge, stockPosition, priceOfStocks,
numStockBought, paidForStock, debt, numPer},
numPer = 20/length; path :=
Module[{}, If[nn != 0, SeedRandom[nn], SeedRandom[]];
Table[Random[NormalDistribution[0, 1]], {numPer}]];
price =
FoldList[#1*Exp[(mu - sigma^2/2)*length +
sigma*Sqrt[length]*#2] &, initPrice, path];
hedge =
Table[delta[price[[i]], strike, sigma,
length*(numPer - i + 1), r], {i, numPer}];
stockPosition = Join[numOptions*hedge, {0}];
priceOfStocks = stockPosition*price;
numStockBought =
Join[{stockPosition[[1]]},
Table[stockPosition[[i + 1]] - stockPosition[[i]],
{i, numPer}]]; paidForStock = numStockBought*price;
debt = {paidForStock[[1]] - receivedForCalls};
Do[AppendTo[debt, debt[[i - 1]]*Exp[r*length] +
paidForStock[[i]], {i, 2, numPer + 1}];
debt[[numPer + 1]] =
debt[[numPer + 1]] +
numOptions*Max[price[[numPer + 1]] - strike, 0];
N[{Last[debt], path, price, hedge, stockPosition,
priceOfStocks, numStockBought, paidForStock, debt}]];

```

Net loan outstanding = $L_t = L_{t-1} * e^{5\%/52} + (N_t - N_{t-1}) S_t$ (i.e., if new shares are bought, they are purchased with borrowed funds, and if shares are sold, the proceeds pay down the loan).

Terminal profitability of the position at $t = 20$: $-100,000 * \text{Max}(S_{20} - X, 0) - L_t$

Program 1 simulates this position.

Explanation of simulation[nn_]

The module simulation[nn_] is more complicated than the module stopLoss[nn_] of Section 1. The complications allow us to illustrate some additional features of *Mathematica*. In the rest of this subsection we explain this program. Recall that any Module has the syntax;

```

Module[{local variables},
lines of code separated by semicolons;
last line of code is returned,
provided there is no semicolon]

```

In our program this works out as follows: {path, price, hedge, stockPosition, priceOfStocks, numStockBought, paidForStock, debt} are the local variables of simulation[nn_]. path is itself a module, which creates numPer normally-distributed random variables:

```

path:=Module[{}, If[nn!=0,
SeedRandom[nn],SeedRandom[]];
Table[Random[NormalDistribution[0,1]], {numPer}]];

```

If nn = 0, then path chooses a random seed stochastically (actually from the computer's internal clock); if, however nn ≠ 0, then path chooses SeedRandom[nn]. In either case, path goes on to create numPer standard normal variables.

```

price=FoldList[(#1*Exp[(mu-sigma^2/2) *
length+sigma*Sqrt[length]*#2])&,initPrice,path]

```

price creates a list of random prices. This list has numPer + 1 prices, and the first price is initPrice. Each succeeding price is given by multiplying the current price by a lognormally distributed random factor corresponding to market moves. The corrected drift (mu-sigma^2/2) appears due to the Ito lemma.

```

hedge= Table[delta[price[[i]],strike,
length*(numPer-i+1),r,sigma], {i,numPer}]

```

gives the list of deltas for the random prices generated. To calculate the `debt` at each point in time, we start off with the initial amount of debt initially taken on:

```
debt = {paidForStock[[1]] - receivedForCalls}
```

At each time period `i`, we take the `debt[[i-1]]`, multiply by one-plus the interest `Exp[r*length]`, and add the cost of the stock bought at `i`:

```
Do[AppendTo[debt,debt[[i-1]]*
  Exp[r*length] (+ paidForStock[[i]]
  ],{i,2,numPer+1}]
```

Note that `debt` can be either positive or negative (negative debt means money invested at the risk-free interest rate). The effectiveness of the hedge can be measured by the amount of debt at the end of the hedging plus the cost of honoring the calls written:

```
debt[[numPer+1]] =
  debt[[numPer+1]] + numOptions*
  Max[price[[numPer+1]] - strike, 0]
```

Running the hedge simulation

The output of `simulation[n]` is a vector of all the results generated, but what really interests us is the first element of this vector, `Last[debt]`, which gives the amount of debt left at the end of 20 weeks. To generate only this amount, we run `simulation[n][[1]]`. Running this 50 times generates a table of random outcomes:

```
In[11]:= tbl = Table[simulation[i, 1/20.][[1]],
  {i, aftercomma50}]
Out[11]= {7969.18, 7994.77, 7968.05, 7992.79, 7927.93,
  7972.82, 7960.95, 7938.29, 7992.04, 7945.55,
  7979.41, 7978.91, 7959.63, 7965.54, 7972.74,
  7995.79, 7978.17, 7997.79, 7980.57, 7978.44,
  7971.64, 7948.41, 7973.88, 8013.82, 7991.98,
  8006.59, 7968.67, 7977.05, 7979.57, 8054.86,
  7970.76, 7965.6, 8007.75, 7972.41, 7971.62,
  7983.68, 7999.05, 7987.44, 7943.52, 8019.04,
  7956.5, 8000.2, 8009.4, 7972.85, 7988.62,
  7992.61, 7991.6, 8004.89, 7948.11, 7958.88}
```

As the length of the periods in the simulation gets shorter, the hedge gets progressively better:

```
In[12]:= m = 50
Table[
  {Mean[Table[simulation[i, 1/n][[1]],
  {i, m}], StandardDeviation[
  Table[simulation[i, 1/n][[1]],
  {i, m}]]}, {n, 1, 50, 10}]] //
  MatrixForm
```

```
Out[12]= 
$$\begin{pmatrix} 8052.76 & 131.256 \\ 7981.18 & 34.6874 \\ 7979.17 & 22.5188 \\ 7978.12 & 17.6612 \\ 7978.15 & 16.3685 \end{pmatrix}$$

```

5. CONCLUSION

We have shown several simple examples of how to model dynamic and static hedging strategies. A deeper explanation of related issues can be found in the references below. In a real world one can not completely hedge many types of risks. However one can estimate the realistic losses under normal scenarios using the models we have described. This is useful for risk management. After running a long sequence of Monte Carlo simulations one can estimate for example a lower quantile of the value distribution of a portfolio. This is one of the most popular risk measuring units,

Value-at-Risk (VaR). A strict definition and ways to calculate it will be discussed later in this series.

REFERENCES

- BENNINGA, SIMON. *Financial Modeling*. MIT Press, Cambridge, Massachusetts (1997).
- DERMAN EMANUEL, DENIZ ERGENER, IRAJ KANI, Forever Hedged, *RISK*, 1994, vol. 7, no. 9, p. 141–145.
- DERMAN EMANUEL, DENIZ ERGENER, IRAJ KANI, Static Options Replication, *The Journal of Derivatives*, 1995, vol. 2, no. 4, p. 78–95.
- EDIRISINGHE CHANAKA, NAIK VASANTTILAK, UPPAL RAMAN, Optimal Replication of Options with Transactions Costs and Trading Restrictions, *J. of Financial and Quantitative Analysis*, vol. 28, no.1, March 1993, p. 117–138.
- HULL, J., *Options, Futures, and Other Derivatives*, third edition, Prentice Hall, Englewood Cliffs, New Jersey, 1997.
- LELAND HAYNE E., Option Pricing and Replication with Transactions Costs, *The Journal of Finance*, vol. 40, no.5, December 1985.

ABOUT THE AUTHORS

The authors acknowledge grants from Wolfram Research, the Krueger and Eshkol Centers at the Hebrew University, and the Israeli Academy of Science. Wiener's research has benefited from a grant from the Israel Foundations Trustees and the Alon Fellowship. Simon Benninga is professor of finance at Tel-Aviv University (Israel) and the Wharton School of the University of Pennsylvania. He is the author of *Financial Modeling* (MIT Press, 1997) and of *Corporate Finance: A Valuation Approach* (with Oded Sarig, McGraw-Hill, 1997); he is also the editor of the *European Finance Review*.

Simon Benninga
 Faculty of Management
 Tel-Aviv University, Tel-Aviv, Israel
 benninga@post.tau.ac.il
<http://finance.wharton.upenn.edu/~benninga>

Zvi Wiener is assistant professor of finance at the business school of the Hebrew University of Jerusalem. His finance research concentrates on the pricing of derivative securities, value-at-risk, computational finance and stochastic dominance. He wrote this article while visiting at the Olin School of Business at Washington University in St. Louis.

Zvi Wiener
 Finance Department, Business School
 Hebrew University, Jerusalem, Israel
 mswiener@pluto.mssc.huji.ac.il
<http://pluto.mssc.huji.ac.il/~mswiener/zvi.html>

ELECTRONIC SUBSCRIPTIONS

Included in the distribution for each electronic subscription is the file `hedgeStrat.nb` containing *Mathematica* code for the material described in this article.