CASE STUDY

nineteen

Option Pricing

case study

OVERVIEW

CS19.1 *Application Overview and Model Development*

There are two main types of options: call options and put options. Call options allow the user to buy a stock at a specified exercise price that is lower than the actual stock price. Put options, on the other hand, allow the user to sell a stock at a specified exercise price that is higher than the actual stock price. For the purposes of this case study, we will consider European options in which the owner can only exercise the option at the specified exercise date. This application provides the user with two options for working with European call and put options. With the first option, the user can compare option prices for a set risk rate and various volatility values. With the second option, the user can determine the optimal number of call and put options to buy of various option prices in order to minimize risk for a desired return.

CS19.1.1 Model Definition and Assumptions

Let's now discuss the calculations and assumptions for both of the user's options. The compare prices option calculates the option price for a given exercise price over various volatility values. The Black Scholes formula, which calculates the option price, assumes that the selected stock prices follow a lognormal random variable. It also assumes that the risk free rate is continuously compounded. A different formula computes the option price for the call options, *C*, and the put options, *P*. Both formulas are complex, so we break them into two parts: d_1 and d_2. The formula uses the current stock price, *S*, the exercise time, *t*, the exercise price, *X*, the risk free rate, *r*, and the annual volatility, *v*. It also uses the cumulative distribution function of the normal distribution, $F_N(\cdot)$.

$$d_1 = \frac{\ln(S/X) + (r + v^2/2)t}{v\sqrt{t}}$$

$$d_2 = d_1 - v\sqrt{t}$$

$$C = S * F_N(d_1) - Xe^{-rt} * F_N(d_2)$$

$$P = S * (F_N(d_1) - 1) - Xe^{-rt} * (F_N(d_2) - 1)$$

We consider three possible exercise prices for the call and put options. The exercise prices for the call options are all greater than the current stock price, and the exercise prices for the put options are all smaller than the current stock price. The option prices are computed for each exercise price of the call and put options. These option prices are recalculated for the new risk rates and for various volatility values in the compare prices option. For the optimization option, they are used as input.

The optimization option seeks to determine the optimal number of options to purchase for each option price. The objective function is to minimize the overall risk, which is the standard deviation of the returns. The total cost should be less than or equal to the available funds displayed in the input cells. The average returns should be greater than or equal to the desired returns. And the desired returns are equal to the average returns found using a set of initial trial values for the decision variables.

Minimize risk = stdev(total returns)

Subject to: total cost ≤ available funds
average(total returns) ≥ desired return

We employ the Solver to determine the solution of this optimization problem. The Solver commands are:

```
SolverOK SetCell:=Range("Risk"), MaxMinVal:=2, ByChange:=Range("CallDV, PutDV")
SolverAdd CellRef:=Range("Returns"), Relation:=3, FormulaText:=Range("DesRet")
SolverAdd CellRef:=Range("TotCost"), Relation:=1, FormulaText:=
Range("CalcStart").Offset(1, 1)
SolverOptions MaxTime:=1000, AssumeNonNeg:=True
SolverSolve UserFinish:=True
```

We use bootstrapping on historical stock prices to estimate the final stock price of a selected stock for 1000 months. For each estimated stock price, the call option revenue and the put option revenue are calculated for each of the three exercise prices using their corresponding option prices. The revenue for the call options is the value of the current price minus the option price. The revenue for the put options is the value of the option price minus the current price. Both revenues are considered to be zero if their values are negative. The total revenues are the sum of the product of each option's revenue and the corresponding decision variable value. The returns are the difference between the total cost of the options bought and the total revenue.

For more details on option pricing, please refer to *Introduction to Probability Models* by Winston and *Data Analysis and Decision Making* by Albright, Winston, and Zappe.

CS19.1.2 Input

The input for this application consists of the following:

- Stock name
- Current stock price
- Current number of shares
- Volatility
- Risk free rate
- Available funds for buying options
- Current date

CS19.1.3 Output

The output for this application consists of the following:

- Exercise prices and corresponding option prices
- Chart of option prices as the volatility varies for a given risk free rate
- Optimal number of options to buy

CS19.2 *Worksheets*

This application requires four worksheets: the welcome sheet, the comparison option sheet, the optimization option sheet, and the calculation sheet. The welcome sheet contains the title, the description of the application, and the "Start" button. (See Figure

CS19.1.) The "Start" button shows the user several forms and then takes him or her to either the comparison option sheet or the optimization options sheet.

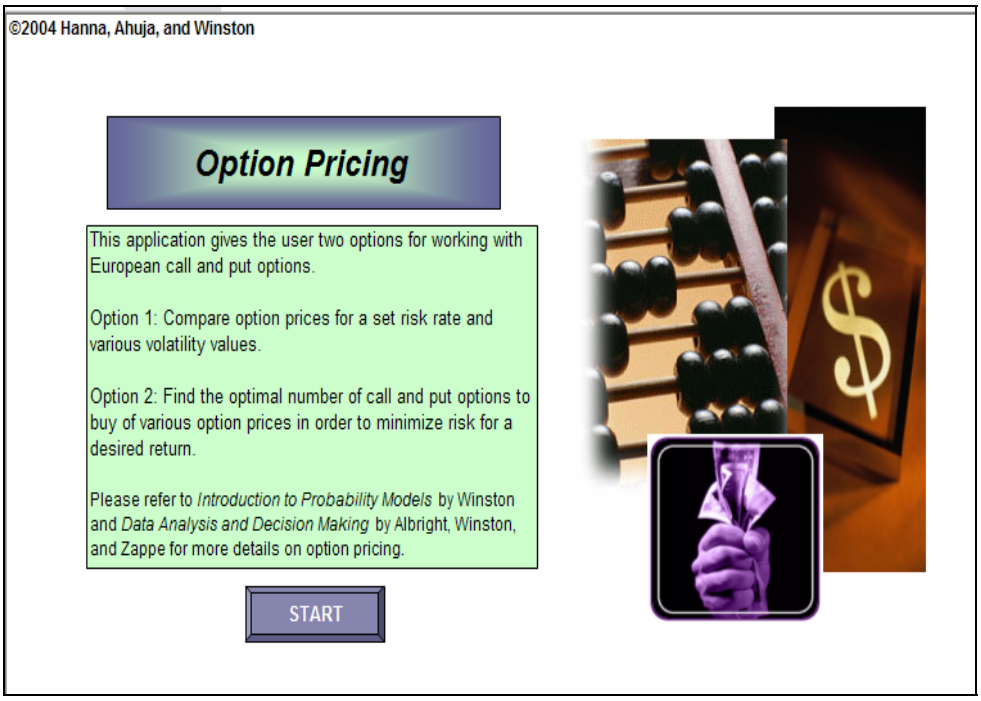


Figure CS19.1 The welcome sheet.

The comparison option sheet allows the user to compare various option prices with varying volatility values. (See Figure CS19.2.) The user's input, which has already been received from the user via a user form, is reported in a table. The initial risk rate appears in a text box; however the user can change this value with the adjacent spin buttons. As this value is changed, the option prices are recalculated using the new risk rate value and various volatility values. These volatility values vary in a range whose center value is the user's initial volatility input value. The option prices are calculated for three exercise prices. There are three exercise prices that are greater than the user's current stock price input for the call options and three exercise prices that are less than the user's current stock price input for the put options. The graphs reflect the option prices of the varying volatility values for each of these exercise prices for both the call and put options.

The optimization option sheet displays the result of an optimization performed that determines the best number of call and put options to buy in order to minimize risk. (See Figure CS19.3.) The user's input appears in a table; below this table are the returns and the risk of the user's investment resulting from the optimization. Then, a table displays the call options and the put options; each table shows the three exercise prices and their corresponding option prices. The sheet also presents the optimal number of each of these options to buy, which was determined by the optimization. The user can press the "View Model" button to view the calculations sheet that performs this optimization.

The calculations sheet contains historical data and formulas used in the optimization. (See Figure CS19.4.) The historical data includes the recorded returns for three different stocks over 60 months. We employ this data to perform bootstrapping on the final stock price for 1000 months.

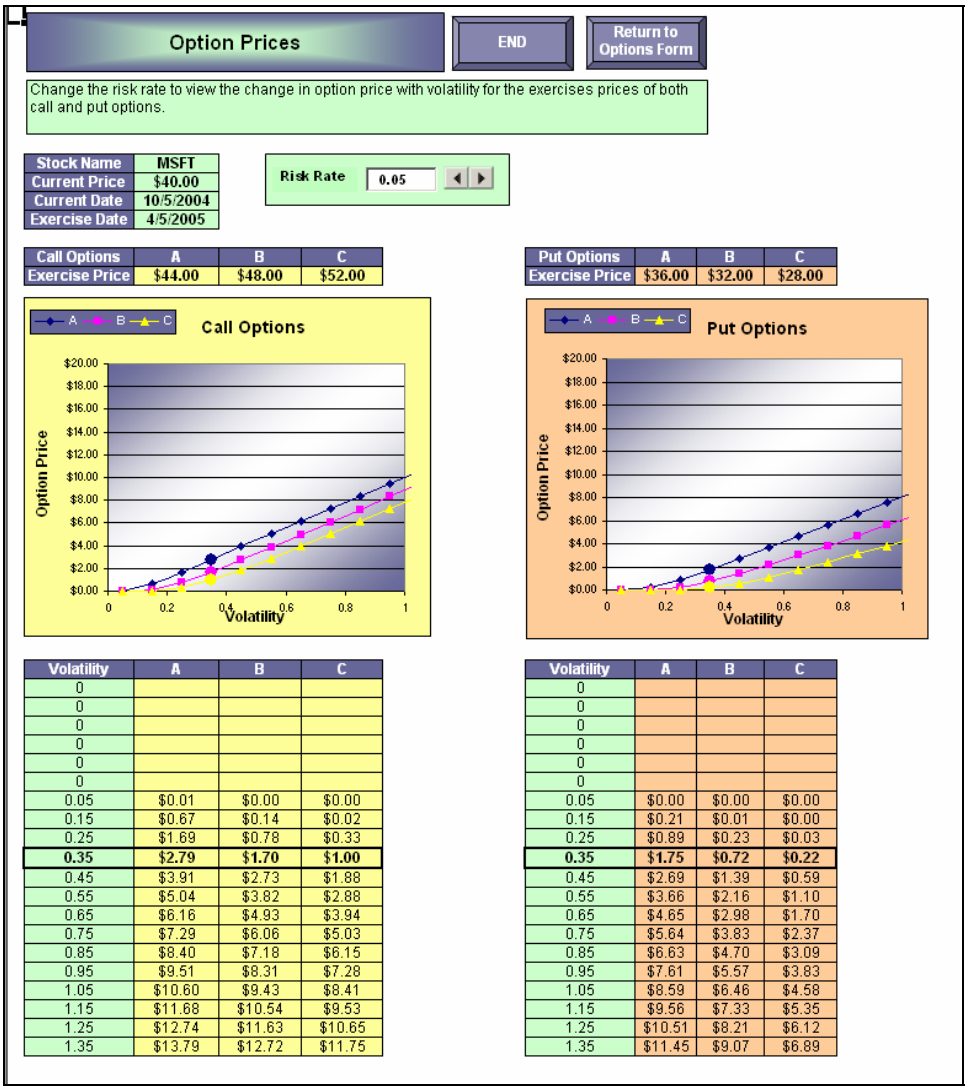
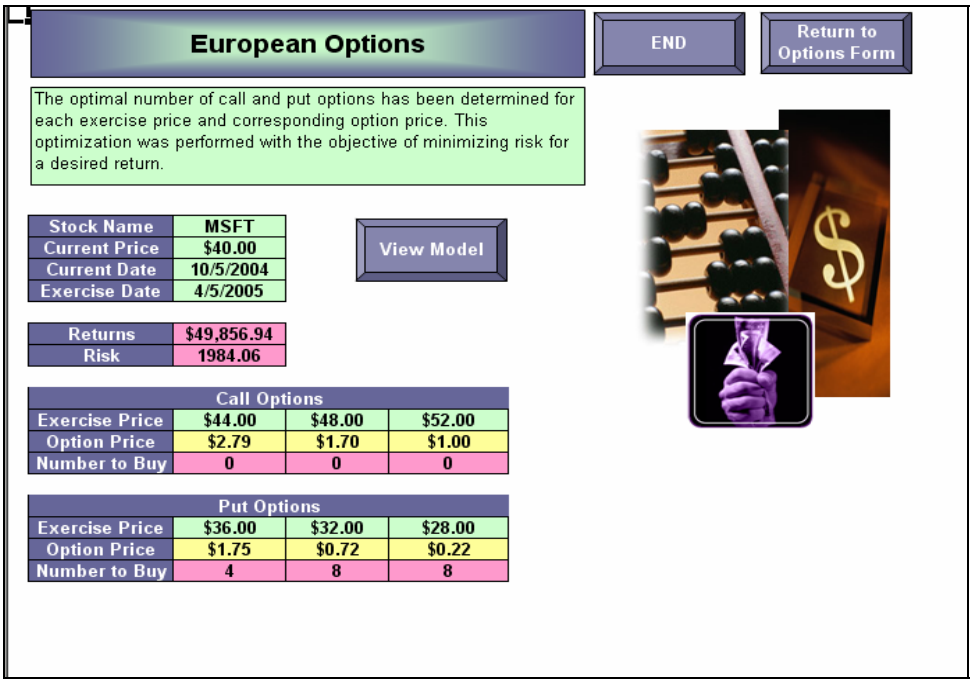


Figure CS19.2 The comparison option sheet.

For each estimated stock price, the call option revenue and put option revenue are calculated for each of the three exercise prices using their corresponding option prices. The total revenues are then calculated using the decision variable cells (the number to buy of each option). Next, the returns are calculated using the total cost of the options bought; this total cost is also calculated using the decision variable cells and the option prices. The total cost should be less than or equal to the available funds displayed in the input cells. The objective function cell is the risk formula cell; the risk is calculated as the standard deviation of the returns. The average returns should be greater than or equal to the desired returns, and the desired returns are equal to the average returns found using a set of initial trial values for the decision variables.



European Options

END

Return to Options Form

The optimal number of call and put options has been determined for each exercise price and corresponding option price. This optimization was performed with the objective of minimizing risk for a desired return.

View Model

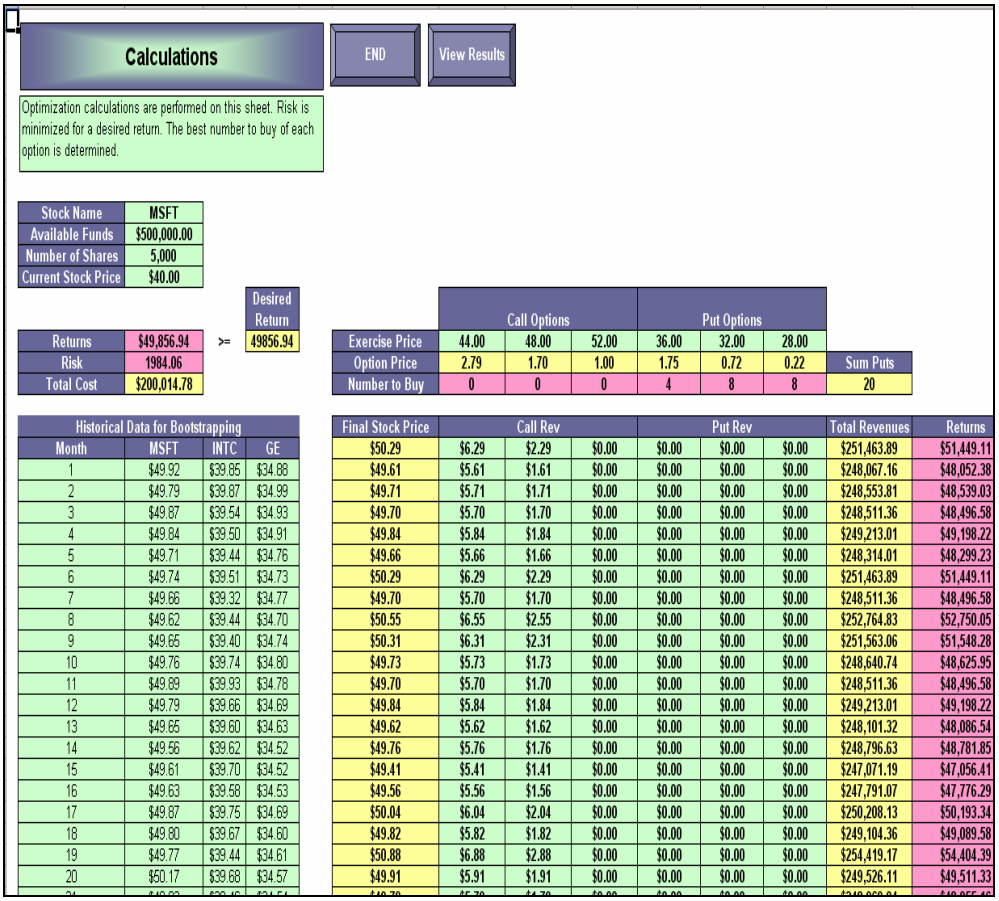
Stock Name	MSFT
Current Price	$40.00
Current Date	10/5/2004
Exercise Date	4/5/2005

Returns	$49,856.94
Risk	1984.06

Call Options			
Exercise Price	$44.00	$48.00	$52.00
Option Price	$2.79	$1.70	$1.00
Number to Buy	0	0	0

Put Options			
Exercise Price	$36.00	$32.00	$28.00
Option Price	$1.75	$0.72	$0.22
Number to Buy	4	8	8

Figure CS19.3 The optimization option sheet.



Calculations

END

View Results

Optimization calculations are performed on this sheet. Risk is minimized for a desired return. The best number to buy of each option is determined.

Stock Name	MSFT
Available Funds	$500,000.00
Number of Shares	5,000
Current Stock Price	$40.00

			Desired Return
Returns	$49,856.94	>=	49856.94
Risk	1984.06		
Total Cost	$200,014.78		

	Call Options			Put Options			
Exercise Price	44.00	48.00	52.00	36.00	32.00	28.00	
Option Price	2.79	1.70	1.00	1.75	0.72	0.22	Sum Puts
Number to Buy	0	0	0	4	8	8	20

Historical Data for Bootstrapping			
Month	MSFT	INTC	GE
1	$49.92	$39.85	$34.88
2	$49.79	$39.87	$34.99
3	$49.87	$39.54	$34.93
4	$49.84	$39.50	$34.91
5	$49.71	$39.44	$34.76
6	$49.74	$39.51	$34.73
7	$49.66	$39.32	$34.77
8	$49.62	$39.44	$34.70
9	$49.65	$39.40	$34.74
10	$49.76	$39.74	$34.80
11	$49.89	$39.93	$34.78
12	$49.79	$39.66	$34.69
13	$49.65	$39.60	$34.63
14	$49.56	$39.62	$34.52
15	$49.61	$39.70	$34.52
16	$49.63	$39.58	$34.53
17	$49.87	$39.75	$34.69
18	$49.80	$39.67	$34.60
19	$49.77	$39.44	$34.61
20	$50.17	$39.68	$34.57

Final Stock Price	Call Rev			Put Rev			Total Revenues	Returns
$50.29	$6.29	$2.29	$0.00	$0.00	$0.00	$0.00	$251,463.89	$51,449.11
$49.61	$5.61	$1.61	$0.00	$0.00	$0.00	$0.00	$248,067.16	$48,052.38
$49.71	$5.71	$1.71	$0.00	$0.00	$0.00	$0.00	$248,553.81	$48,539.03
$49.70	$5.70	$1.70	$0.00	$0.00	$0.00	$0.00	$248,511.36	$48,496.58
$49.84	$5.84	$1.84	$0.00	$0.00	$0.00	$0.00	$249,213.01	$49,198.22
$49.66	$5.66	$1.66	$0.00	$0.00	$0.00	$0.00	$248,314.01	$48,299.23
$50.29	$6.29	$2.29	$0.00	$0.00	$0.00	$0.00	$251,463.89	$51,449.11
$49.70	$5.70	$1.70	$0.00	$0.00	$0.00	$0.00	$248,511.36	$48,496.58
$50.55	$6.55	$2.55	$0.00	$0.00	$0.00	$0.00	$252,764.83	$52,750.05
$50.31	$6.31	$2.31	$0.00	$0.00	$0.00	$0.00	$251,563.06	$51,548.28
$49.73	$5.73	$1.73	$0.00	$0.00	$0.00	$0.00	$248,640.74	$48,625.95
$49.70	$5.70	$1.70	$0.00	$0.00	$0.00	$0.00	$248,511.36	$48,496.58
$49.84	$5.84	$1.84	$0.00	$0.00	$0.00	$0.00	$249,213.01	$49,198.22
$49.62	$5.62	$1.62	$0.00	$0.00	$0.00	$0.00	$248,101.32	$48,086.54
$49.76	$5.76	$1.76	$0.00	$0.00	$0.00	$0.00	$248,796.63	$48,781.85
$49.41	$5.41	$1.41	$0.00	$0.00	$0.00	$0.00	$247,071.19	$47,056.41
$49.56	$5.56	$1.56	$0.00	$0.00	$0.00	$0.00	$247,791.07	$47,776.29
$50.04	$6.04	$2.04	$0.00	$0.00	$0.00	$0.00	$250,208.13	$50,193.34
$49.82	$5.82	$1.82	$0.00	$0.00	$0.00	$0.00	$249,104.36	$49,089.58
$50.88	$6.88	$2.88	$0.00	$0.00	$0.00	$0.00	$254,419.17	$54,404.39
$49.91	$5.91	$1.91	$0.00	$0.00	$0.00	$0.00	$249,526.11	$49,511.33

Figure CS19.4 The calculation sheet.

Summary

Welcome sheet	Includes the application description and the “Run Demo” and “Start” buttons.
Comparison option sheet	Allows the user to compare various option prices with varying volatility values.
Optimization option sheet	Shows the result of an optimization performed that determines the best number of call and put options to buy in order to minimize risk.
Calculation sheet	Contains the historical data and the formulas used in the optimization.

CS19.3 *User Interface*

This application’s user interface requires navigational buttons, two user forms, and controls on the worksheet. The first user form is the options form. (See Figure CS19.5.) This form allows the user to choose between the comparison and optimization options for the application. We include a frame with two option buttons on this form.

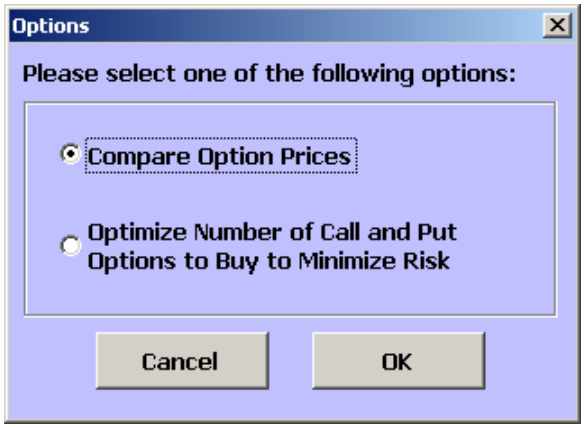


Figure CS19.5 The options form.

The second user form is the input form. (See Figure CS19.6.) We include this form to gather all of the input values from the user. Frames group the textboxes and labels of similar input. In the combo box, the user can select one of the three available stocks.

The compare prices sheet includes a spin button and textbox control to allow the user to change the risk rate and update the option price values. The “End” navigational button brings the user back to the welcome sheet, the “Return to Options Form” button redisplays the options form, and the “View Model” and “View Results” buttons bring the user to the calculations sheet and optimization option sheet, respectively.

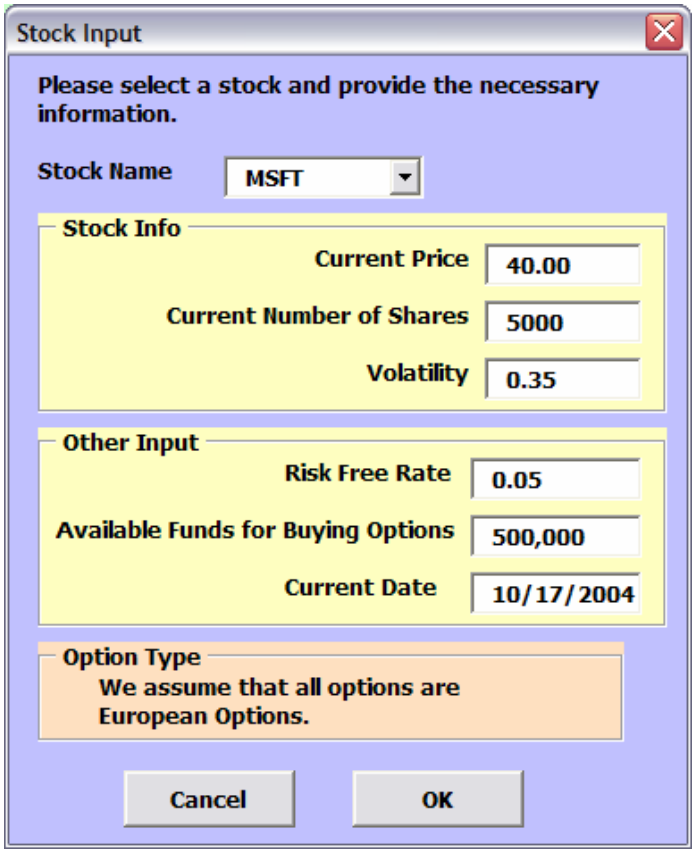


Figure CS19.6 The input form.

Summary

Options form	Allows the user to choose between the compare price and optimization options.
Input form	Prompts the user to enter all the input.
Controls on the worksheet	Include the spin button and the textbox on the comparison sheet; they allow the user to change the risk rate and the update results.
Navigational buttons	Include "Start," "End," "Return to Options Form," "View Model," and "View Results."

CS19.4 *Procedures*

We will now outline the procedures for this application beginning with the initial sub procedures and variable definitions. (See Figure CS19.7.) The *Main* procedure, which is called from the "Start" button, presents the options form.

The procedures for the options form note which option the user has selected. (See Figure CS19.8.) If the user has chosen the compare prices option, then he or she is taken to the comparison sheet, the input form is displayed, and the *ComparePrices* procedure is called. If the user has chosen the optimization option, the calculation sheet and input form are displayed, and then the *Optimize* procedure is called.

```
Option Explicit
Option Base 1

Const NatE = 2.71828182845905
Public i As Integer, CurrPrice As Double, NumShares As Integer, Volatility As Double, _
AvailFunds As Double, RiskRate As Double, CurrDate As Date, Init As Boolean, _
Calls(3, 2) As Double, Puts(3, 2) As Double, ExTime As Double, StockName As String, j As Integer

Sub Main()  'called from Start button and Return to Options buttons
   frmOptions.Show
End Sub
```

Figure CS19.7 The variable definitions and the *Main* procedure.

```
Private Sub cmdCancel_Click()
    Unload Me
    End
End Sub

Private Sub cmdOK_Click()
    If optPrices Then
        Unload Me
        Worksheets("Prices").Visible = True
        Worksheets("Welcome").Visible = False
        Worksheets("Europe").Visible = False

        frmInput.Show
        Call ComparePrices
    ElseIf optOptimize Then
        Unload Me
        Worksheets("Calc").Visible = True
        Worksheets("Welcome").Visible = False
        Worksheets("Prices").Visible = False

        frmInput.Show
        Call Optimize
    End If
End Sub
```

Figure CS19.8 The procedures for the option form.

The procedures for the input form record the user's input. (See Figure CS19.9.) The values from the text boxes are converted to double, integer, and date data types as they are equated to their respective variables. The application employs the input for the current stock price to create the exercise prices for the call and put options.

The *BlkSchols* function procedure, which appears in Figure CS19.10, is called from both the *ComparePrices* and the *Optimize* procedures to compute the option prices with the Black Schols formula. The input for this function is the exercise price, the option type (call or put), and the volatility value. Since the function is a little complicated, we use two variables to break it into parts. The first part uses the current price, the exercise price, the risk rate, the volatility, and the exercise time (6 months = 0.5 years). The second part uses the first part as well as the volatility and exercise time. We then use these parts to compute the option price in the final formulation of the Black Schols equation.

```
Private Sub cmdCancel_Click()
    Worksheest("Welcome").Visible = True
    ActiveSheet.Visible = False
    Unload Me
    End
End Sub

Private Sub cmdOK_Click()
    'convert values to Double and Date data types
    StockName = cmbStocks.Value
    CurrPrice = CDbl(txtPrice.Value)
    NumShares = CInt(txtShares.Value)
    Volatility = CDbl(txtVol.Value)
    AvailFunds = CDbl(txtFunds.Value)
    RiskRate = CDbl(txtRisk.Value)
    CurrDate = CDate(txtDate.Value)

    ExTime = 0.5      'assume 6 month options
    'record variations of current price
    For i = 1 To 3
        Calls(i, 1) = CurrPrice + CurrPrice * 0.1 * i
        Puts(i, 1) = CurrPrice - CurrPrice * 0.1 * i
    Next i
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    cmbStocks.RowSource = "Stocks"
    cmbStocks.Value = cmbStocks.List(0)
    txtDate.Value = Date
End Sub
```

Figure CS19.9 The procedures for the input form.

The final computation varies for call and put options, so we use the option type input passed to the function to determine which calculation to perform. This final calculation uses both parts previously calculated as well as the current price, exercise price, risk rate, and natural logarithm (stored as a constant variable).

```
Function BlkSchols(ExPrice, OptType, Vol)   'Black Scholes function
    Dim Part1 As Double, Part2 As Double
    Part1 = (Log(CurrPrice / ExPrice) + (RiskRate + Vol ^ 2 / 2) * ExTime) / (Vol * Sqr(ExTime))
    Part2 = Part1 - (Vol * Sqr(ExTime))

    If OptType = "C" Then
        BlkSchols = CurrPrice * Application.WorksheetFunction.NormSDist(Part1) - _
            ExPrice * NatE ^ (-RiskRate * ExTime) * Application.WorksheetFunction.NormSDist(Part2)
    ElseIf OptType = "P" Then
        BlkSchols = CurrPrice * (Application.WorksheetFunction.NormSDist(Part1) - 1) - _
            ExPrice * NatE ^ (-RiskRate * ExTime) * (Application.WorksheetFunction.NormSDist(Part2) - 1)
    End If
End Function
```

Figure CS19.10 The *BlkSchols* function procedure.

The *ComparePrices* procedure performs the calculations for the comparison option. (See Figure CS19.11.) It begins by displaying the input values on the compare prices sheet,

and it also outputs the three exercise prices for the call and put options. Next, it initializes the risk rate controls (both the spin button and the text box) to present the risk rate value provided by the user on the input form. It then uses the user's volatility input value to create a table of volatility values for the call and put options. Each table has a range of volatility values of ten more than and ten less than the user's initial value. The procedure then loops over these volatility values and calculates the option price for each exercise price in the call and put options for these volatility values. It performs this calculation by calling the *BlkSchols* function procedure and passing the exercise price, option type, and volatility value. Notice that the option price is only calculated for the positive volatility values.

```
Sub ComparePrices() 'called for compare prices option
    Application.ScreenUpdating = False
    'display input values
    With Range("PricesStart")
        .Offset(0, 1).Value = StockName
        .Offset(1, 1).Value = CurrPrice
        .Offset(2, 1).Value = CurrDate
        .Offset(3, 1).Value = DateAdd("m", ExTime * 12, CurrDate)
    End With
    For i = 1 To 3
        Range("CallStart").Offset(1, i).Value = Calls(i, 1)
        Range("PutStart").Offset(1, i).Value = Puts(i, 1)
    Next i

    'set risk rate control to input value
    Init = True
    Worksheets("Prices").spnRiskRate = RiskRate * 100
    Worksheets("Prices").txtRiskRate = RiskRate

    'create table of volatility values
    For i = 1 To 10
        Range("CallPrices").Offset(i, 0).Value = Volatility - (1 - i / 10)
        Range("PutPrices").Offset(i, 0).Value = Volatility - (1 - i / 10)
    Next i
    For i = 11 To 20
        Range("CallPrices").Offset(i, 0).Value = Volatility + (i / 10 - 1)
        Range("PutPrices").Offset(i, 0).Value = Volatility + (i / 10 - 1)
    Next i

    'calculate corresponding option prices for positive volatility values in table
    For i = 1 To 20
        If Range("CallPrices").Offset(i, 0).Value > 0 Then
            For j = 1 To 3
                Range("CallPrices").Offset(i, j).Value = _
                    BlkSchols(Calls(j, 1), "C", Range("CallPrices").Offset(i, 0).Value)
                Range("PutPrices").Offset(i, j).Value = _
                    BlkSchols(Puts(j, 1), "P", Range("PutPrices").Offset(i, 0).Value)
            Next j
        Else
            For j = 1 To 3
                Range("CallPrices").Offset(i, j).Value = "--"
                Range("PutPrices").Offset(i, j).Value = "--"
            Next j
        End If
    Next i
    Application.ScreenUpdating = True
End Sub
```

Figure CS19.11 The *ComparePrices* procedure.

The application calls the *Update* procedure when the risk rate controls are changed. The *Change* event procedure for the spin button and the *LostFocus* event procedure for the text box update the control values and then call the *Update* procedure. (See Figure CS19.12.) This procedure recalculates the option prices for all the volatility values in the call and put option tables using the new risk rate value provided by the user. (See Figure CS19.13.) The axis scales on the charts are also updated.

```
Private Sub spnRiskRate_Change()
    If Init Then
        Init = False
        Exit Sub
    End If
    txtRiskRate.Value = spnRiskRate.Value / 100
    RiskRate = txtRiskRate.Value
    Call Update
End Sub

Private Sub txtRiskRate_LostFocus()
    spnRiskRate.Value = txtRiskRate.Value * 100
    RiskRate = txtRiskRate.Value
    Call Update
End Sub
```

Figure CS19.12 The risk rate control event procedures.

```
Sub Update()    'update option prices for new risk rate
    Application.ScreenUpdating = False
    CurrPrice = Range("PricesStart").Offset(1, 1).Value
    ExTime = 0.5

    'recalculate option prices for all volatility values using new risk rate
    For i = 1 To 20
        If Range("CallPrices").Offset(i, 0).Value > 0 Then
            For j = 1 To 3
                Range("CallPrices").Offset(i, j).Value = _
                    BlkSchols(Range("CallStart").Offset(1, j).Value, "C", Range("CallPrices").Offset(i, 0).Value)
                Range("PutPrices").Offset(i, j).Value = _
                    BlkSchols(Range("PutStart").Offset(1, j).Value, "P", Range("PutPrices").Offset(i, 0).Value)
            Next j
        End If
    Next i

    'set chart axes
    ActiveSheet.ChartObjects(1).Activate
    ActiveChart.Axes(xlCategory).MaximumScale = Volatility + 1
    ActiveChart.Axes(xlValue).MaximumScale = CurrPrice / 2
    ActiveSheet.ChartObjects(2).Activate
    ActiveChart.Axes(xlCategory).MaximumScale = Volatility + 1
    ActiveChart.Axes(xlValue).MaximumScale = CurrPrice / 2
    Range("A1").Select
    Application.ScreenUpdating = True
End Sub
```

Figure CS19.13 The *Update* procedure.

The *Optimize* procedure performs the calculations needed for the optimization option. (See Figure CS19.14.) The procedure first calculates and stores the option prices for the

call and put options' three exercise prices by calling the *BlkSchols* function procedure. It then calls the *RecordInput* and *DoCalc* procedures, which perform the main optimization calculations. The *Optimize* procedure then updates the summary report on the optimization results sheet and takes the user to this sheet.

```
Sub Optimize()  'called for optimization option
    Application.ScreenUpdating = False
    'calculate option prices for given market prices and volatility
    For i = 1 To 3
        Calls(i, 2) = BlkSchols(Calls(i, 1), "C", Volatility)
        Puts(i, 2) = BlkSchols(Puts(i, 1), "P", Volatility)
    Next i

    Call RecordInput
    Call DoCalc

    'update report sheet
    With Range("EurStart")
        .Offset(0, 1).Value = StockName
        .Offset(1, 1).Value = CurrPrice
        .Offset(2, 1).Value = CurrDate
        .Offset(3, 1).Value = DateAdd("m", ExTime * 12, CurrDate)
    End With
    Range("Returns, Risk").Copy
    Range("EurResults").Offset(0, 1).PasteSpecial xlPasteValues
    Range(Range("CallCalc"), Range("CallCalc").Offset(2, 2)).Copy
    Range("EurCalls").Offset(0, 1).PasteSpecial xlPasteValues
    Range(Range("PutCalc"), Range("PutCalc").Offset(2, 2)).Copy
    Range("EurPuts").Offset(0, 1).PasteSpecial xlPasteValues

    Worksheets("Europe").Visible = True
    Worksheets("Calc").Visible = False
    Range("A1").Select
    Application.CutCopyMode = False
    Application.ScreenUpdating = True
End Sub
```

Figure CS19.14 The *Optimize* procedure.

The *RecordInput* procedure records the user's input on the calculation sheet. (See Figure CS19.15.) The input values from the input form as well as the exercise prices and option prices are then recorded.

```
Sub RecordInput()   'record input to model sheet
    With Range("CalcStart")
        .Offset(0, 1).Value = StockName
        .Offset(1, 1).Value = AvailFunds
        .Offset(2, 1).Value = NumShares
        .Offset(3, 1).Value = CurrPrice
    End With
    For i = 1 To 3
        With Range("CallCalc")
            .Offset(0, i - 1).Value = Calls(i, 1)
            .Offset(1, i - 1).Value = Calls(i, 2)
        End With
        With Range("PutCalc")
            .Offset(0, i - 1).Value = Puts(i, 1)
            .Offset(1, i - 1).Value = Puts(i, 2)
        End With
    Next i
End Sub
```

Figure CS19.15 The *RecordInput* procedure.

The *DoCalc* procedure runs the Solver to find the optimization results. (See Figure CS19.16.) Even though most of the formulas for the optimization have already been inserted in the cells on the worksheet during preparation, the bootstrapping formulas are inserted into the future stock value column using the stock selected by the user. The trial decision variable values then determine the desired returns for the user. Finally, the application prepares and runs the Solver, which determines the optimal solution.

```
Sub DoCalc()    'run Solver on model sheet
    'perform bootstrapping of historical data for selected stock
    Dim StockNum As Integer
    StockNum = Application.WorksheetFunction.Match(Range("StockName"), Range("Stocks"), 0) + 1
    For i = 1 To 1000
        Range("FinStock").Cells(i).Formula = "=VLOOKUP(" & Int(Rnd() * 59 + 1) & ",HistData," & StockNum & ")"
    Next i

    'scenario analysis
    Range("CallDV, PutDV").ClearContents
    Range("TrialVal").Copy
    Range("CallDV").PasteSpecial xlPasteValues
    Range("DesRet").Value = Range("Returns").Value

    'run solver for optimal solution
    Worksheets("Calc").Activate
    SolverReset
    SolverOK SetCell:=Range("Risk"), MaxMinVal:=2, ByChange:=Range("CallDV, PutDV")
    SolverAdd CellRef:=Range("Returns"), Relation:=3, FormulaText:=Range("DesRet")
    SolverAdd CellRef:=Range("TotCost"), Relation:=1, FormulaText:=Range("CalcStart").Offset(1, 1)
    SolverAdd CellRef:=Range("SumPuts"), Relation:=1, FormulaText:=Range("CalcStart").Offset(2, 1)
    SolverOptions MaxTime:=1000, AssumeNonNeg:=True
    SolverSolve UserFinish:=True
End Sub
```

Figure CS19.16 The *DoCalc* procedure.

The navigational procedures are for the “End,” “View Model,” and “View Results” buttons. (See Figure CS19.17.)

```
Sub EndProg()
    Worksheets("Welcome").Visible = True
    ActiveSheet.Visible = False
End Sub

Sub ViewModel()
    Worksheets("Calc").Visible = True
    ActiveSheet.Visible = False
End Sub

Sub ViewResults()
    Worksheets("Europe").Visible = True
    ActiveSheet.Visible = False
End Sub
```

Figure CS19.17 The navigational procedures.


Summary

Main	Initializes the application and takes the user to the input sheet.
Options form procedures	Record whether the user has selected the compare prices or the optimization option.
Input form procedures	Record all of the input from the user.
BlkSchols function procedure	Uses the Black Schols equation to calculate the option prices for the given exercise price, the option type (call or put), and the volatility value.
ComparePrices	Calculates the option prices for various volatility values and the given risk rate value.
Optimize	Finds the optimal number of call and put options to buy such that the overall risk is minimized for a desired return.
RecordInput	Records all of the input to the calculation sheet.
DoCalc	Prepares the formulas and runs the Solver for the optimization results.
Navigational procedures	For the “End,” “View Model,” and “View Results” buttons.

CS19.5 *Re-solve Options*

The user can re-solve this application for both options. For the compare prices option, the user can change the risk rate with either the spin button or the text box control. When either of these control values is changed, the option prices for the volatility values are recalculated according to the new risk rate value. The user can also press the “Return to Options Form” button to enter new values in the input form by selecting the compare prices option again. For the optimization option, the user can press the “Return to Options Form” button to enter new value in the input form by selecting the optimization option again.


Summary

Compare Prices Option	Uses the risk rate controls to change the risk rate and to calculate the updated option price values; uses the “Return to Options Form” button to enter new values into the input form.
Optimization Option	Uses the “Return to Options Form” button to enter new values into the input form.

CS19.6 *Summary*

- This application provides the user with two options for working with European call and put options: compare option prices for a set risk rate and various volatility values; and find the optimal number of call and put options of various option prices to buy in order to minimize risk for a desired return.
- This application requires four worksheets: the welcome sheet, the compare prices sheet, the optimization option sheet, and the calculation sheet.

- The user interface requires navigational buttons, controls on the worksheet, and two user forms.
- Several procedures for this application perform the compare prices calculations and optimization calculations.
- The user can re-solve the application by changing the risk rate controls for the compare prices option or by selecting the “Return to Options Form” button to enter new values in the input form for both options.

CS19.7 *Extensions*

- Include American options as part of all the calculations. Let the user choose between European and American options as part of the input form.
- Allow the user to vary the exercise time of the options; in other words, do not always assume a six-month option. Add this feature to the input form.
- Allow the user to enter a desired return for the optimization option. Then provide the option to minimize risk or maximize return before performing the optimization.