

Spreadsheets Make EXCELlent Tools for Exploring Mathematics: Buttons, Macros, and Visual Basic

Sarah L. Mabrouk
Framingham State College

Suppose you would like to create a tool that will allow students to explore the graphs of the function linear function $y = Ax + B$ and the quadratic function $y = Ax^2 + B$, using a button to change the function type. Using the buttons available on the Forms toolbar or the Control Toolbox toolbar, we can do exactly that. The advantages of creating such a tool using MS Excel include availability of the software, ease of use, and ease of creation.

The Control Toolbox Toolbar contains ActiveX Controls, check and text boxes, clickable buttons, option buttons, scrollbars and spin buttons, text and pictures, and combo boxes, providing extended design options and additional control options as well as limiting user modifications to the Design Mode.



Figure 1 The Control Toolbox Toolbar

The Forms Toolbar contains similar tools but without color options for versions of MS Excel 2003 and earlier and without the extended control options. In addition, for worksheets without protection, the user can select and delete Forms Toolbar elements using a right-click.



Figure 2 The Forms Toolbar

For this discussion, we will discuss the elements on the Control Toolbox Toolbar.

- **Option Buttons**

Suppose we have created a basic tool that allows the user to graph the linear function $y = Ax + B$ and we would like to use option buttons to switch between the linear function $y = Ax + B$ and the quadratic function $y = Ax^2 + B$. We can use a coupled pair of option buttons to switch between function types and the IF function to change the data and the displayed function equation.

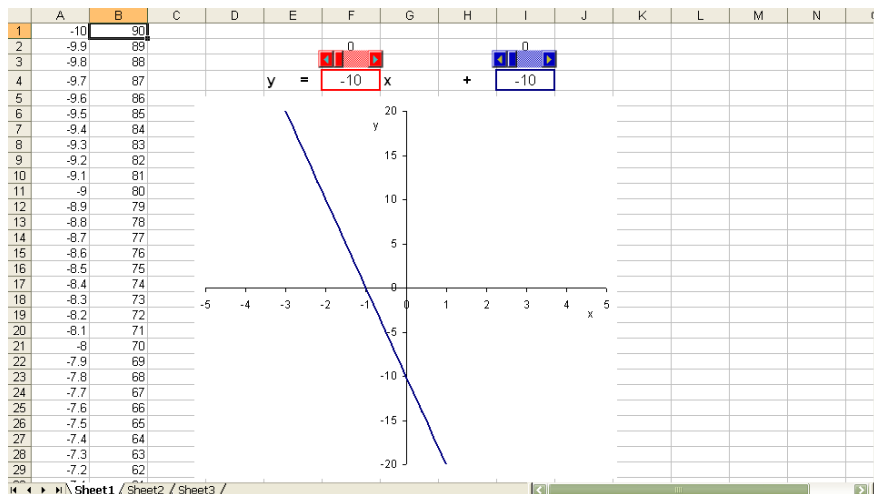




Figure 3 Basic tool with scrollbars for coefficients A and B.

Let us begin by adding the option buttons. Option buttons, , operate on a True/False basis: the button deposits the value TRUE into a linked cell if the button is selected and the value FALSE into the linked cell if the statement is false. Grouping buttons allows them to operate as a unit with only one button “turned on” at a time. To select the Option Button on the Control Toolbox Toolbar, we can either first left-click the Design Mode button, , to turn on the Design Mode or simply left-click the Option Button on the Control Toolbox Toolbar. Then, we left-click, hold and drag the cursor on the worksheet to draw the Option Button to the desired size. Releasing the left mouse button,

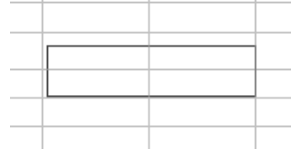


Figure 4 The trace of the Option Button on the worksheet.

will produce the initial Option Button. The button will initially be named OptionButton1. The selected option button, indicated by the circular handles in Figure 5 below, can be resized by moving the cursor to one of these handles where the cursor will change to a double-headed arrow, left-clicking, holding and dragging the option button the desired size. It is best to use one of the handles at the corner since this will allow you to modify the length and the width of the button simultaneously.

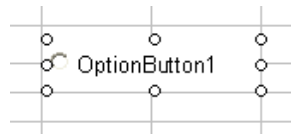



Figure 5 The selected Option Button after it is traced on the worksheet.

In addition to changing the button size by visually, we can modify the size of the button while we set the button properties.

To set the properties for the Option Button, we can right-click the button and select Properties on the pop-up menu, or select the Properties button, , on the Control Toolbox Toolbar.

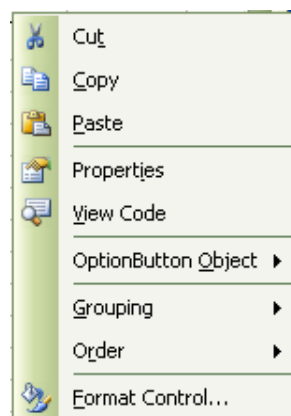


Figure 6 The pop-up menu displayed when we right-click OptionButton1 on the worksheet.

Using the Properties dialog box, Figure 7, we can set the properties of the control with the options grouped alphabetically or by category; we will use the Categorized tab to set the options. There are two alignment options, left- alignment, with the text on the left and the button on the right, and right-alignment, with the text on the right and the button on the left; here, we select right-alignment. The BackColor and ForeColor options are used to set the background color for the button and the text color for the button. To activate the drop-down color menu, we left-click the region to the right of BackColor or ForeColor or simply enter the desired color code between two ampersands, &. Left-clicking the region to the right of

the option, next to the color code, opens the color menus, the System colors, Figure 8, are the base grays, white and black, and the Palette colors, Figure 9, provide other hues.

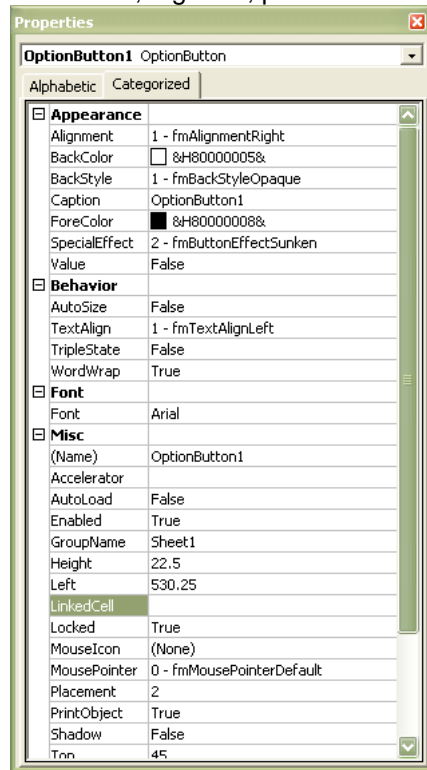


Figure 7 The Properties dialog box.

There are two special effects available, the flat button and the sunken button, as well as two back styles, transparent and opaque.

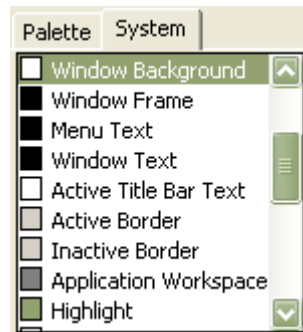


Figure 8 The System colors.

We use the caption to change the text displayed on the button. For this button, we change the caption to Linear. Autosize is set to False since we set the size of the button. The TextAlign option can be used to align the text on the button to the left, right, or center. For longer captions, the WordWrap can be set. The font for the caption is set using the Font option; in Figure 10, we set the font for the caption as 12pt Arial Bold.

The (Name) property is very important if you are going to use multiple controls on a worksheet or within a workbook. Setting the name of the control by type and even including the sheet name allows you to move easily among the various controls when you are working in the Visual Basic editor. Since this worksheet will contain two option buttons, we will name this option button OptionButton_Lin; we will name the other option button OptionButton_Quad. We use the underscore since spaces cannot be used in the names for controls.

The height and the width for the button can be fine-tuned using the Height and Width options, respectively, in the Misc category.

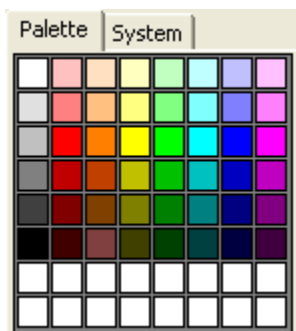


Figure 9 The Palette colors.

The LinkedCell is the cell into which the TRUE or FALSE value for the selected or unselected button will be deposited. Here, we use cell L1 as the linked cell; we will use cell L2 as the linked cell for the Quadratic option button.

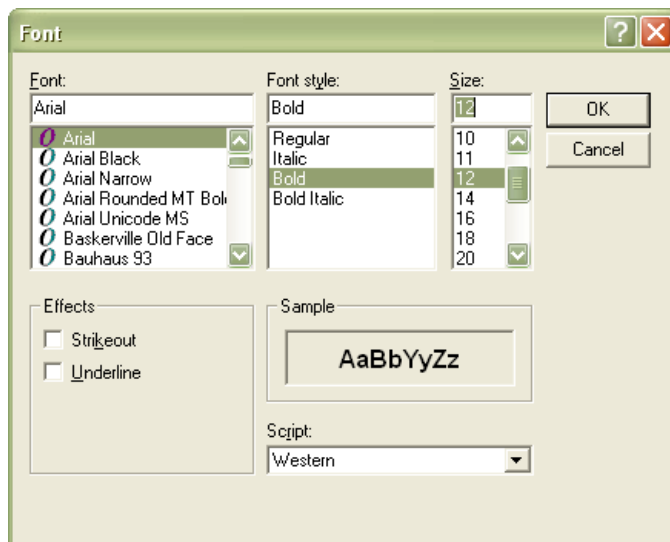




Figure 10 The Font dialog box.

The MouseIcon and MousePointer options can be used to select the cursor icon that will be displayed when the cursor is moved near the button. Left-clicking the section to the right of the MousePointer prompt, opens the MousePointer pull-down menu. We can select one of the mouse pointers listed or select the last option “99 – fmMousePointerCustom” to select our own cursor from the Windows cursors or any cursors that we might have available. Left-clicking the region containing (none) to the right of MouseIcon, produces a button, , and left-clicking the button allows us to navigate to a folder containing cursors. The Windows folder on the C drive (other name if you have multiple drives or have named your hard drive) contains a Cursors folder from which you can select a variety of Windows cursors.

The Shadow option can be used to display a shadowed button. The Visible option can be used to make the button visible or invisible when the Design Mode has been turned off, and the PrintObject option can be used to make the button printable or unprintable when the worksheet is printed. The Picture option can be used to display a picture on the button in addition to or instead of text; the PicturePosition sets the position of the picture on the button.

With the properties for the button set, we close the Properties dialog box and turn off the Design mode by left-clicking the Design Mode button, . Left-clicking the button then enters the value TRUE in cell L1. The option button is turned on once the button is clicked and the button cannot produce the value FALSE unless it is grouped with at least one other button.

L
TRUE




Figure 11 Left-clicking OptionButton_Lin.

L	M
TRUE	
FALSE	

Linear
 Quadratic

Figure 12 The two Option Buttons.

We create the Quadratic button in the same manner in which we created the Linear button. Turning on the Design Mode and selecting the Linear button, we can also copy the Linear button to create the Quadratic button with similar properties. We must use a unique linked cell; here, we will use cell L2.

Having created the second option button, Figure 12, we see that they work together with only one button being selected at one time and either the value of L1 or L2 being TRUE while the other's value is FALSE. Grouping the buttons allows us to move them together as a unit on the sheet.

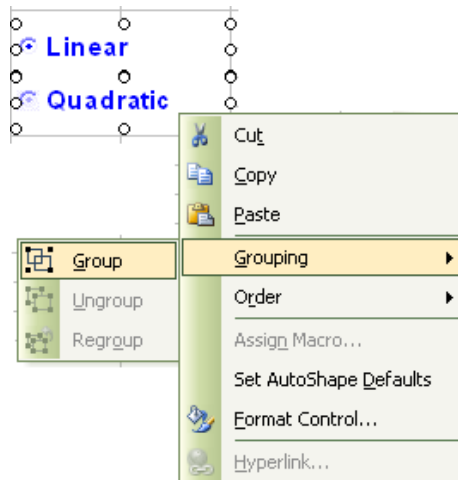


Figure 13 Grouping the buttons.

Turning on the Design Mode, left-clicking one option button and simultaneously pressing the Cntrl button and left-clicking the other option button to select both (indicated by handles around each button) and then right-clicking the selected region, we select the Grouping and Group options on the pop-up menu, Figure 13, to group the buttons; if we later choose to add buttons or to modify the buttons, we can use the Ungroup or Regroup options.

It is important to mention that if we want to modify the buttons in any way, including their position on the worksheet, we must turn on the Design Mode. This feature of the ActiveX controls on the Control Toolbox Toolbar is very helpful since the user cannot alter or delete the buttons unless (s)he know about the Design Mode. That is, students cannot accidentally delete the various control elements on the worksheet.

Since the buttons work as a unit, we can now use an IF function to change the function that is being graphed from the linear function to the quadratic function, referencing only one of these cells; here, we will reference cell L1. We use “=IF(\$L\$1=TRUE,\$F\$4*A1+\$I\$4,\$F\$4*A1^2+\$I\$4)” in cell B1 and drag this

formula to fill the cells used to generate the y-values for the graph. Since we want to examine L1 as well as use the values of the coefficients stored in cells F4 and I4, the locations for the values for the coefficients A and B, respectively, we must be careful to fix these cells in the formula.

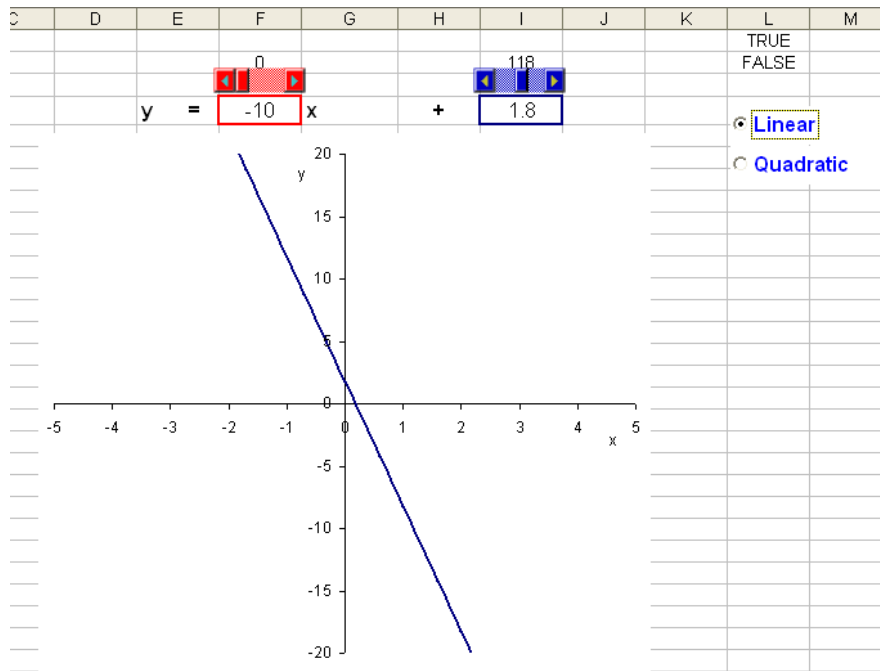


Figure 14 The linear function displayed.

For the function equation, we use the formula “=IF(L1=TRUE,"x","x^2”)”; note the use of the quotation marks around the x and x^2 in the formula and that we do not need to fix the cell for L1 since we are not copying this formula to other cells.

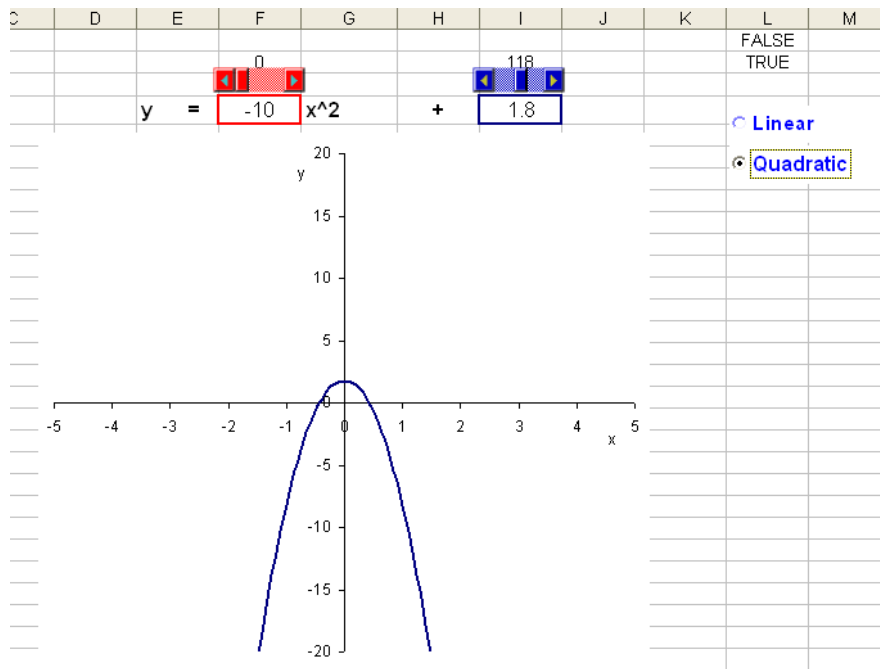



Figure 15 The quadratic function displayed.

Figure 14 and Figure 15 display the nearly completed tool linear and quadratic functions displayed, respectively. To complete the tool, all we need to do is hide any information that we do not want to display, for example the TRUE or FALSE values in cells L1 and L2 and the values for the linked cells for the scrollbars, F2 and I2 as well as the data that is graphed in columns A and B, in the window for the tool as well as provide any desired instructions to the user.

The values in columns A and B and cells F2, I2, L1, and L2 can be easily hidden by making them the same color as the background of the sheet, here, white. Then, we can turn off the grid using either the Toggle Grid button on the Forms Toolbar, , or unchecking the Gridlines box in the Window options section of the View tab in the Options dialog box accessed using the Options entry on the Tools menu.

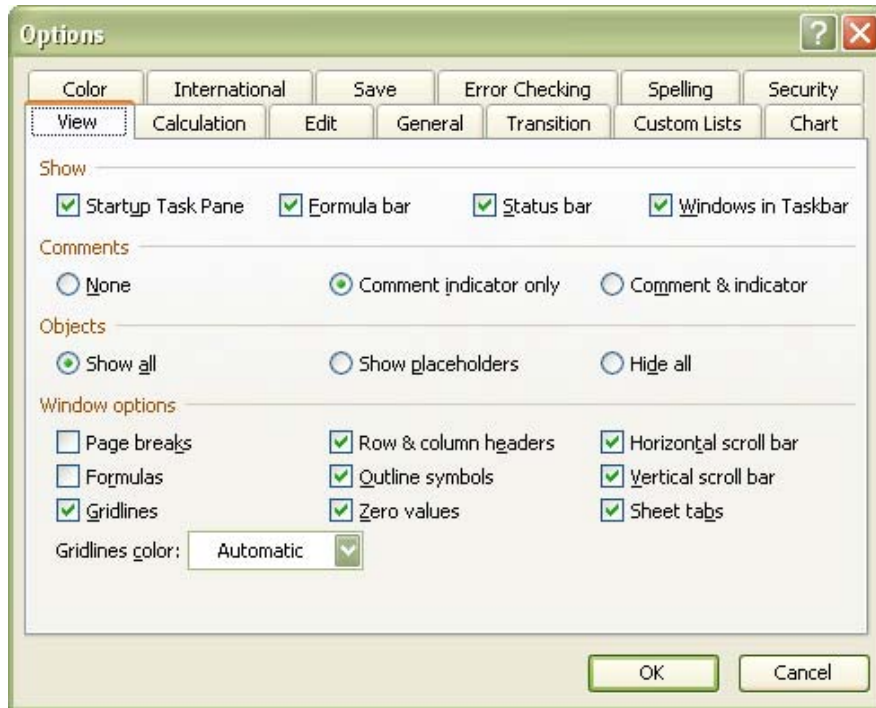


Figure 16 The Options dialog box.

Finally, we delete any sheets that we do not want to keep in the workbook as well as name the sheet(s) that we keep in the workbook, producing our completed tool in Figure 17 below.

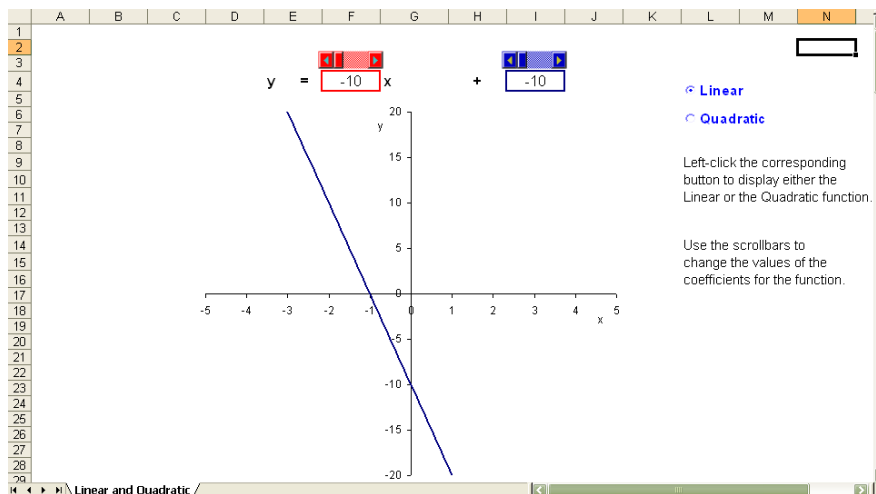


Figure 17 The completed tool.

If we use MS Excel 2007, there are some differences. A major difference is the location for the Forms elements and the ActiveX controls: these are displayed on the Developer tab on the ribbon. The Developer tab must be turned on from within the Excel Options.

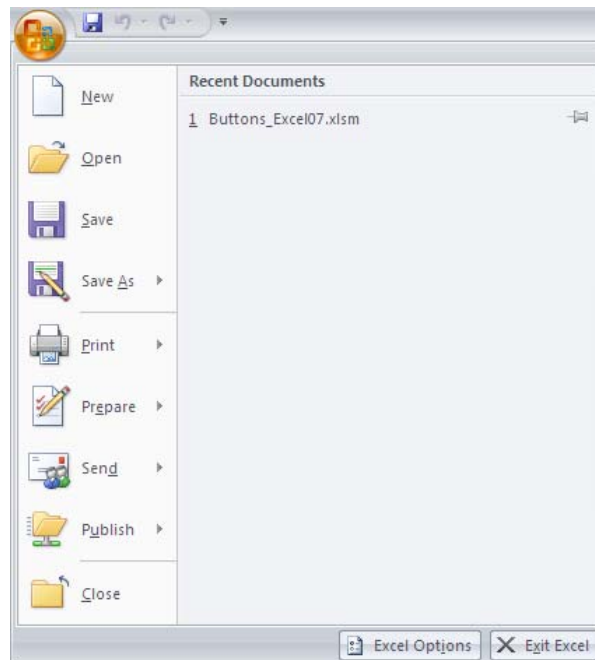



Figure 18 The menu displayed when upon left-clicking the Office Button.

We left-click the Office Button, , to open the main menu and then left-click the Excel Options button.

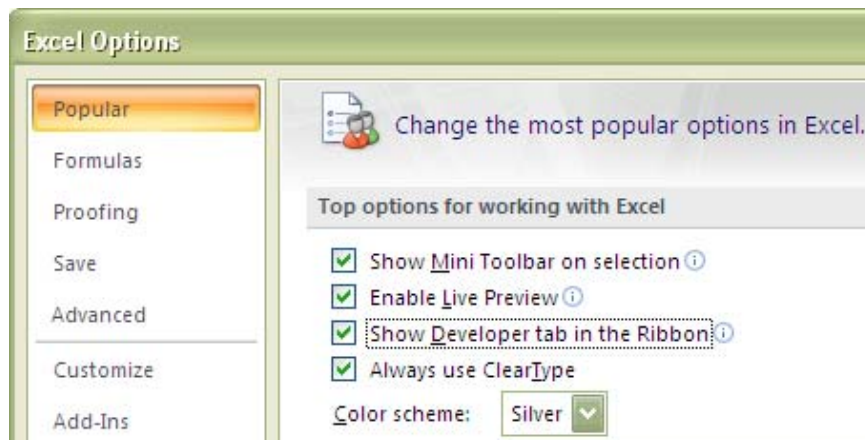


Figure 19 The Popular tab of the Excel Options dialog box.

On the Popular tab of the Excel Options dialog box, we check the “Show Developer tab in the Ribbon” option and left-click the Ok button; then, Developer tab is available.

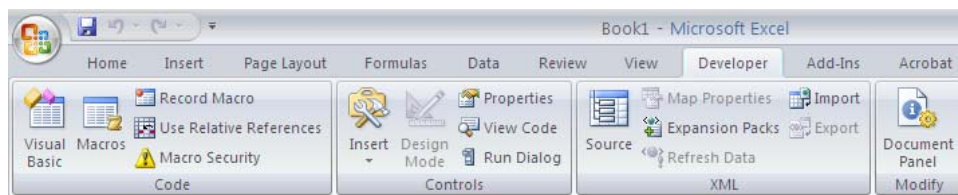


Figure 20 The options on the Developer tab.

The Developer tab provides groups for macros, whether recorded on the sheet or entered using Visual Basic and the controls, both Forms elements and ActiveX controls.

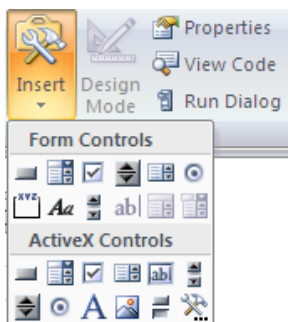


Figure 21 The Forms and ActiveX controls accessed using the Insert button.

Left-clicking the Insert button opens the menu from which we can enter both Forms Controls and ActiveX controls. As before, we left-click the control and trace it to the desired size in the desired location on the worksheet. As before, left-clicking an ActiveX Control opens the Design Mode.

To set the Properties for ActiveX Controls we may either right-click the control and select the Properties option on the pop-up menu or we may left-click the Properties button on the ribbon.

Other than needing initially to turn on the Developer tab on the ribbon and the minor location changes, the controls are set-up and modified in the same as for earlier versions of MS Excel.

- **Command Button with a Macro**

Now let us make a similar tool that uses a Command Button with a macro. We can record the macro and then attach the macro to the button. We create the command button and modify its properties in a similar manner to that in which we created the Options Buttons. In this case as well, we will need one button for the Linear function and one Button for the Quadratic Function.

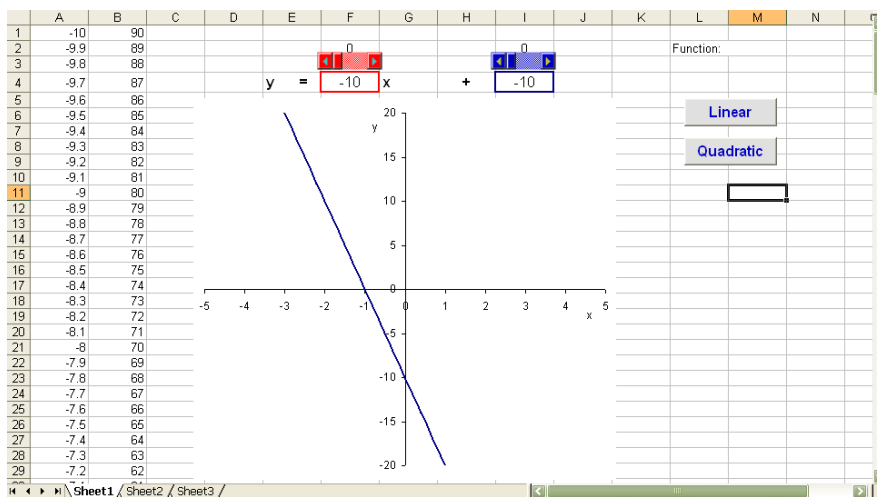


Figure 22 Two Command Buttons on the worksheet.

With the Command buttons created, all we need to do is create the macros and attach them to the buttons. Since the current formula is for a linear function, we will create a macro for the Quadratic button first, changing the formula for the y-value in cell B1 from “= \$F\$4*A1+\$I\$4” to “= \$F\$4*A1^2+\$I\$4” and dragging/copying this formula from cell B1 to each cell up to and including B201 as well as changing the “x” in cell G4 to “x^2” for the function formula.

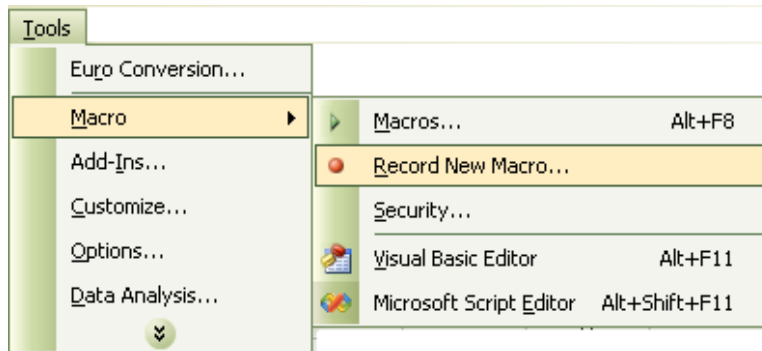


Figure 23 The Macros menu on the Tools menu.

To create this macro, we can record actions that we perform on the worksheet. To do this, we use the Record a New Macro option on the Macro option on the Tools menu for MS Excel 2003 and earlier, Figure 23 below, or the Record Macro option on the Developer tab for MS Excel 2007.

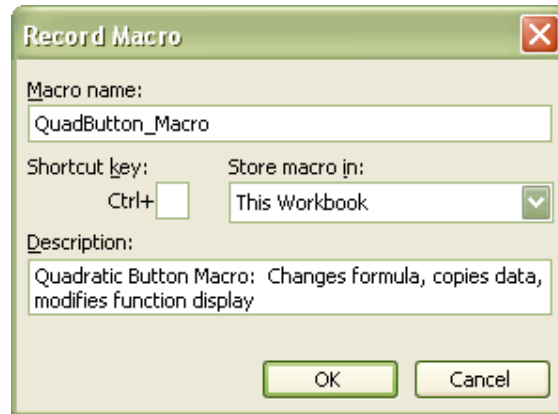


Figure 24 The Record Macro dialog box.

For either version, the Record Macro dialog box is opened: we name the macro, provide a description, if desired, and provide a location in which to store the macro. Since we want to use this macro only with this tool, we select “This Workbook” as the location in which to store the macro.




Upon left-clicking OK, we begin recording the macro, being careful not to make errors; errors can be removed with some knowledge of Visual Basic. Once we have performed all the desired operations, we left-click the Stop button, , on the Stop Recording toolbar, Figure 25.



Figure 25 The Stop Recording toolbar.

With the macro recorded, we can open the Macros dialog box, to test the macro; we left-click the Run button. Upon finding that the macro runs properly, we can use the Edit option in the Macro dialog box, Figure 26, to enter the Visual Basic editor, Figure 27, and copy/paste the macro to the Quadratic Button.

To attach this macro to the Quadratic button, we turn on the Design Mode, select the Quadratic button, and either double left-click or left-click the View Code button, , for MS Excel 2003 and earlier versions, and , for MS Excel 2007.

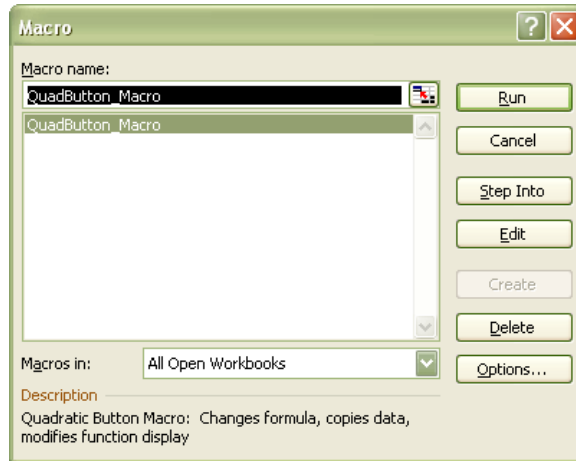


Figure 26 The Macros dialog box.

In the Visual Basic editor, we copy the text of the macro, starting with “Range(“B1”), Select” and ending with “Range(“M3”), Select”, Figure 27,

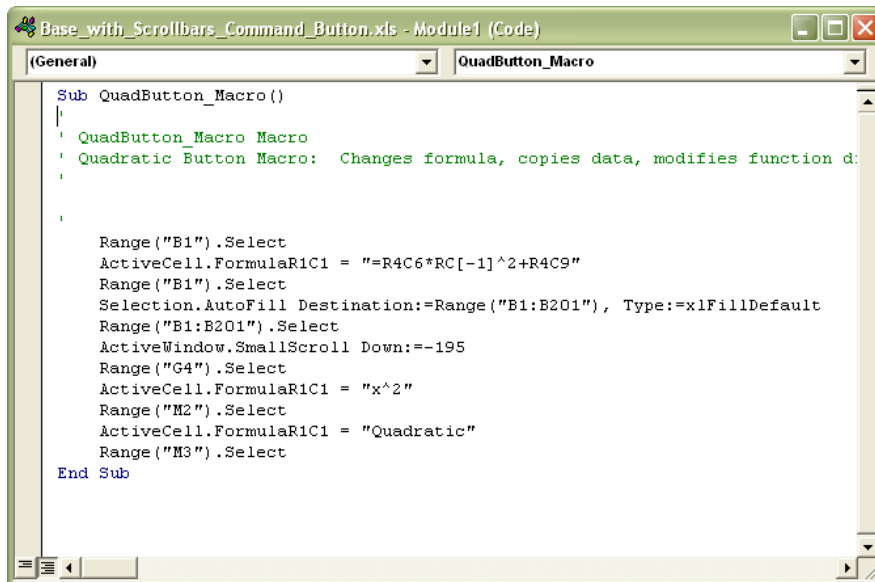


Figure 27 The QuadButton_Macro displayed in the Visual Basic editor.

and paste it between the "Private Sub CommandButton_Quad_Click()" and "End Sub" in the CommandButton_Quad window; this illustrates the importance of how we name the buttons.

We can also edit this macro, removing the “^2” in “ActiveCell.FormulaR1C1 = "=R4C6*RC[-1]^2+R4C9"" as well as in “ActiveCell.FormulaR1C1 = "x^2"" and changing the “Quadratic” in “ActiveCell.FormulaR1C1 = "Quadratic"" to “Linear”, in order to create the macro for the Linear button. We can use the scrollbar to display the CommandButton_Lin base code or use the pull-down menu at the top of the window.

After entering the code, we simply close the Visual Basic editor and turn off the Design Mode. The tool is now complete, and we can use the buttons to change the function type from Linear to Quadratic.

As before, all we need to do to finish the tool, is to hide the data values such as the x-values and y-values in columns A and B as well as the linked cells for the scrollbars, F2 and I2, provide any desired instructions for the user, name the worksheet, and delete unused worksheets.

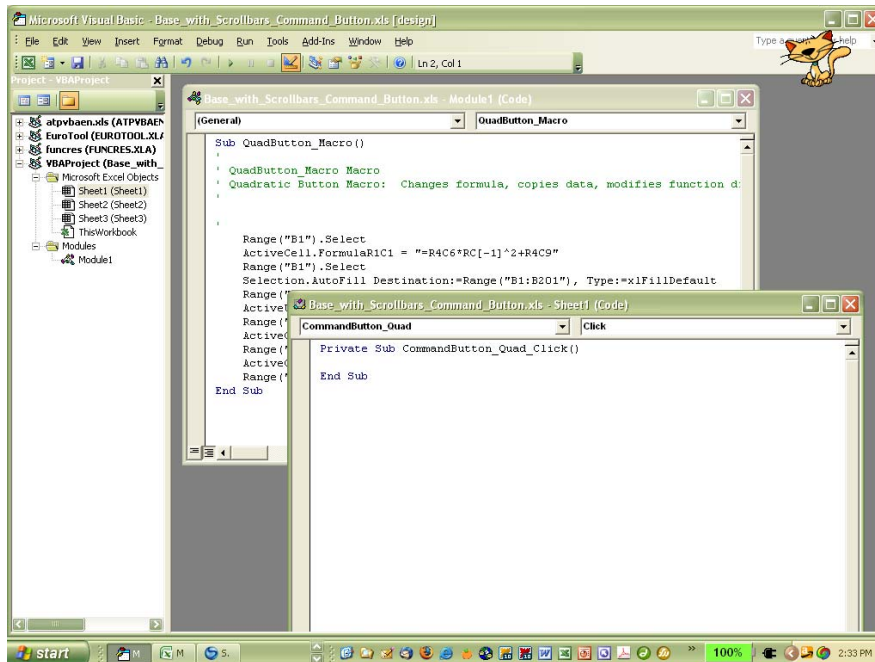


Figure 28 The Visual Basic editor.

With that, our second tool is complete.

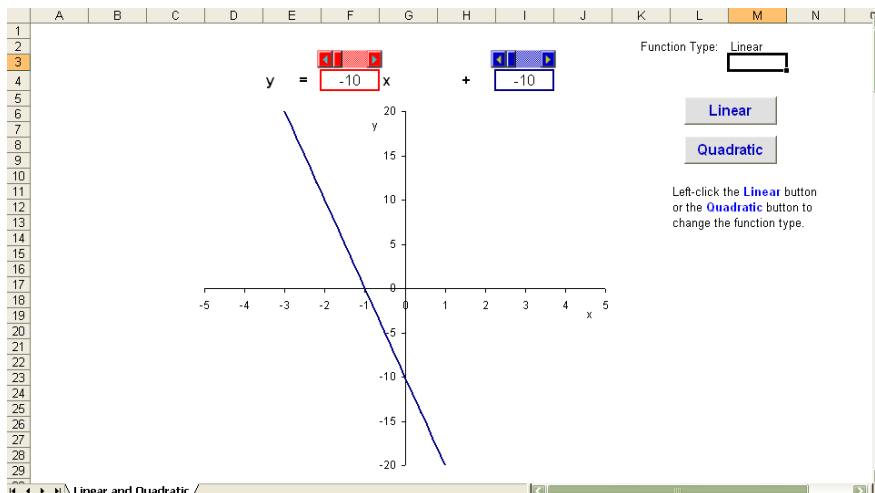


Figure 29 The completed tool.

Recording macros and examining the code, is a great way in which to begin to learn Visual Basic.



I hope that you found this handout helpful. I tried to provide enough information to help you to get started without making you feeling overwhelmed. The best way in which to learn how to create interactive tools, as with learning to use any software, is to explore the keywords using the help facility and to “play” with it, adding elements to a workbook and exploring and changing their properties. As you explore/experiment with creating interactive workbooks, remember to save the workbooks that you create so that you can refer to them in the future – we learn through trial and error, and we can learn a lot from our mistakes.

Please feel free to contact me, smabrouk@frc.mass.edu, if you have any questions – I am always glad to help.