

Creating Interactive Workbooks Using MS Excel

Sarah Mabrouk, Framingham State College

Overview: MS Excel provides powerful calculation, statistical, graphing, and general data analysis and organizational tools for education, research, and business. When most people think of MS Excel, they think of spreadsheets, statistics, curve fitting, and general graphing of data. When people think of scrollbars, buttons, pop-ups, and general computer manipulatives, they think of Java or JavaScript applets. Since most educators do not know Java and JavaScript, they believe that they cannot create interactive workbooks that can be used for demonstrations, student assignments, and concept exploration by students. While one can create web pages that can be used to display movies created using a series of images, in general, the user must watch the preprogrammed sequence, unable to change or to control aspects of the individual variables. However, MS Excel can be used to create interactive workbooks that allow the user to control various aspects of information displayed in data, graphics, and even problems to be examined using the same type of scrollbars and buttons that one would expect in Java or JavaScript applets. The advantages to using MS Excel include the fact that, once downloaded, students can use the workbook without the Internet, students can easily download additional copies of the file if the workbook becomes corrupted by "play", MS Excel is familiar and available software for most faculty and students, and no plug-ins are needed to use the files.

Imagine a file or web page that uses simple scrollbars to allow students

- to explore and to compare graphs of lines, $ax + by = c$, as they change the values of a, b, and c,

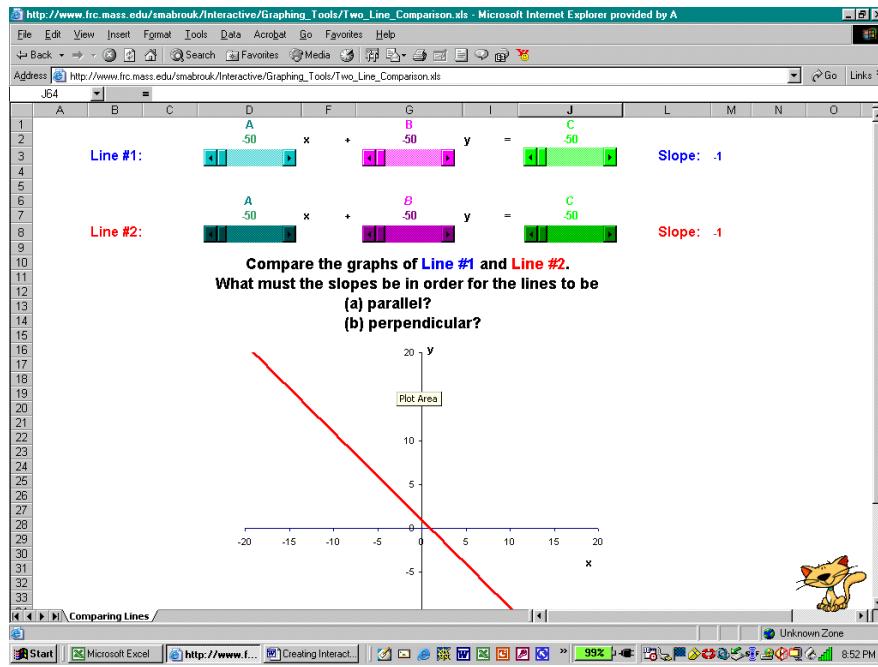


Figure 1

http://www.framingham.edu/faculty/smabrouk/Interactive/Graphing_Tools/Two_Line_Comparison.xls

- to examine the graphs of lines $ax + by = c$ as they change the values of a, b, and c, http://www.framingham.edu/faculty/smabrouk/Interactive/Graphing_Tools/Stndrd_Frm_of_Line_Explrtn.xls;
- to compare the graph of a linear function $y = ax + b$ to the graph of $y = x$ and the graph of the quadratic function $y = ax^2 + b$ with $y = x^2$ as they change the values of a and b, http://www.framingham.edu/faculty/smabrouk/Interactive/Graphing_Tools/Lin_Quad_Fn_Comparison.xls;
- to examine the graphs of linear $y = ax + b$, quadratic $y = ax^2 + bx + c$, and cubic functions $y = ax^3 + bx^2 + cx + d$ as they change the various coefficient values, a, b, c, and d;

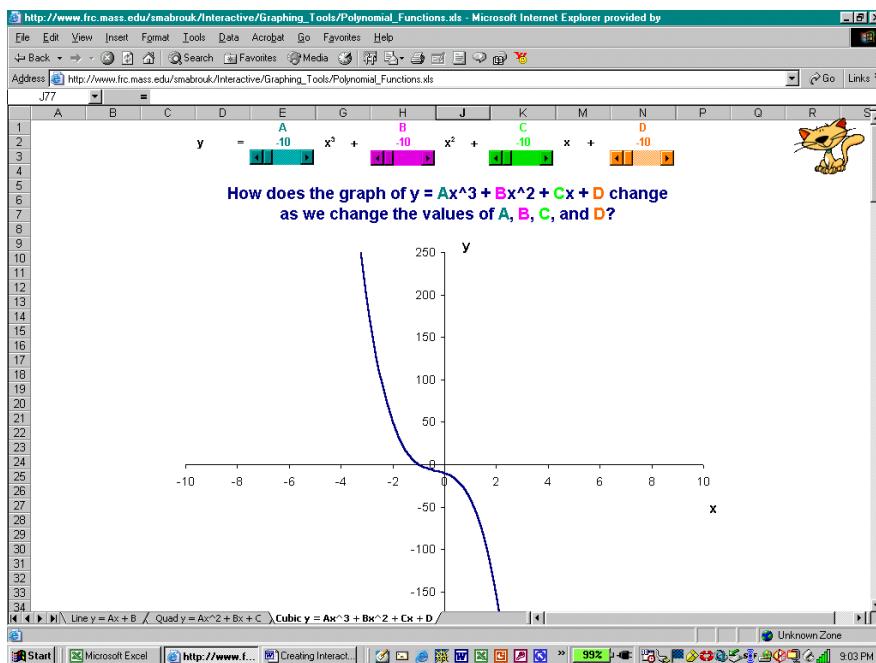


Figure 2

http://www.framingham.edu/faculty/smabrouk/Interactive/Graphing_Tools/Polynomial_Functions.xls

- to examine the graphs of quadratic functions $y = a(x - h)^2 + k$ as they change the values of the coefficient a and the values of the coordinates of the vertex, h and k ,

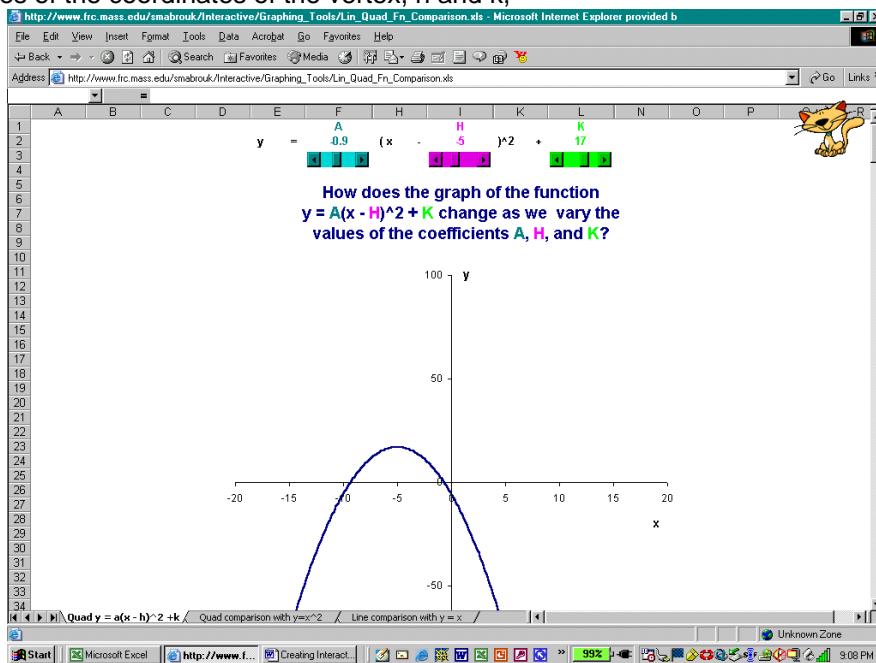


Figure 3

http://www.framingham.edu/faculty/smabrouk/Interactive/Graphing_Tools/Lin_Quad_Fn_Comparison.xls

- to examine the graphs of exponential functions $y = a^x$ and logarithmic functions $y = \log_a x$ as they change the value of the base a ,
http://www.framingham.edu/faculty/smabrouk/Interactive/Graphing_Tools/Exp_Log_Comparison.xls; and
- to compare the graphs of exponential functions $y = a^x$ and logarithmic functions $y = \log_a x$ on the same axes as they change the value of the base a .

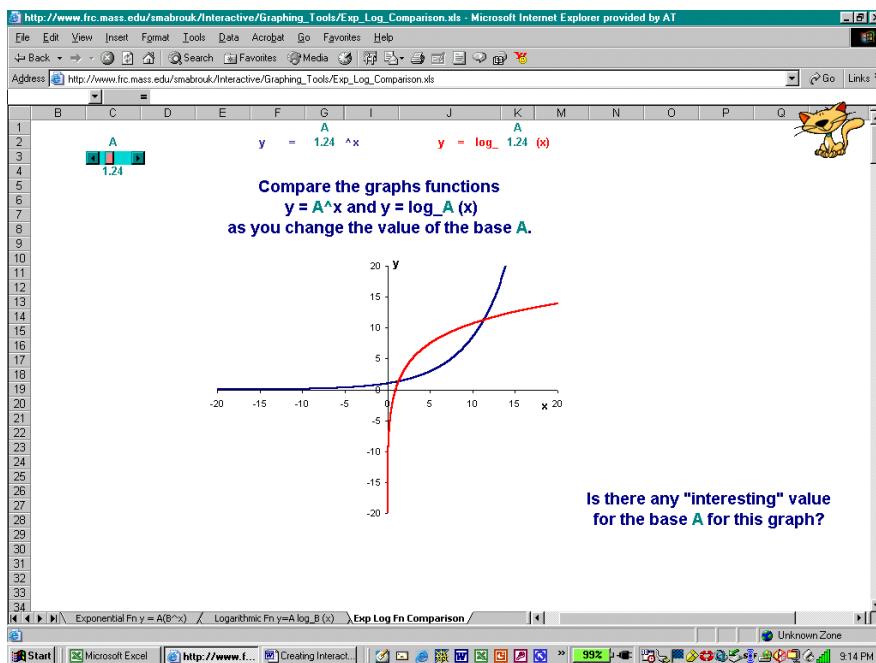


Figure 4
http://www.framingham.edu/faculty/smabrouk/Interactive/Graphing_Tools/Exp_Log_Comparison.xls

- to explore series and sequences and series,
http://www.framingham.edu/faculty/smabrouk/Interactive/Sequences_Series_Tools/Sequence_And_Series_Exploration_1_over_n.xls;
- to explore percentages,
http://www.framingham.edu/faculty/smabrouk/Interactive/General/Percentage_Investigation.xls;
- to explore applications of quadratic functions such as revenue and projectile motion,
http://www.framingham.edu/faculty/smabrouk/Interactive/Quadratic_Fns/Monthly_Rental_Revenue.xls and
http://www.framingham.edu/faculty/smabrouk/Interactive/Quadratic_Fns/Human_Cannonball.xls;
- to explore how changing one point affects linear regression,
http://www.framingham.edu/faculty/smabrouk/Interactive/Statistics_Tools/Linear_Regression.xls;
- to explore the affects of outliers in normal, skewed, and multimodal distributions,
http://www.framingham.edu/faculty/smabrouk/Interactive/Statistics_Tools/Outlier_Investigation.xls; and
- to explore how the mean and median are affected by changing one data value,
http://www.framingham.edu/faculty/smabrouk/Interactive/Statistics_Tools/Mean_Median.xls.

All of these and many more are possible! I created and have used the above-described interactive workbooks using MS Excel as well as others that allow the user to explore such topics as the normal distribution, intersections of lines, and trigonometric functions. Students find these interactive workbooks easy to use and actually *fun* to use. Such workbooks are easy and fun to create.

These interactive workbooks can be used for demonstrations as well as for assignments, and one can design workbooks that include questions, graphs, equations, and explanations with the interactive aspects provided by buttons and scrollbars. By creating handouts that contain explicit instructions on various scrollbar values to explore, for example, one can use the same interactive workbooks that one would use for in-class demonstrations for stand-alone assignments and guided learning activities. Incorporating such instructions or questions in the workbook to provide the user with appropriate direction, the interactive workbooks can be used for projects/assignments. Examples of project workbooks, Percentage Investigation, Monthly Rental Revenue, and Human Cannonball, can be viewed at <http://www.framingham.edu/faculty/smabrouk/Interactive/index.htm>.

To make interactive workbooks using MS Excel, all you need to know are the basics of how to use MS Excel for calculations and graphing. To this, you simply add the Control Toolbox, one of the built-in toolbars. To display the Control Toolbox, you can use the Customize dialog box displayed in Figure 5, access through Customize on the Tools menu, or Control Toolbox on the Toolbars sub-menu accessed through the View menu displayed in Figure 6.

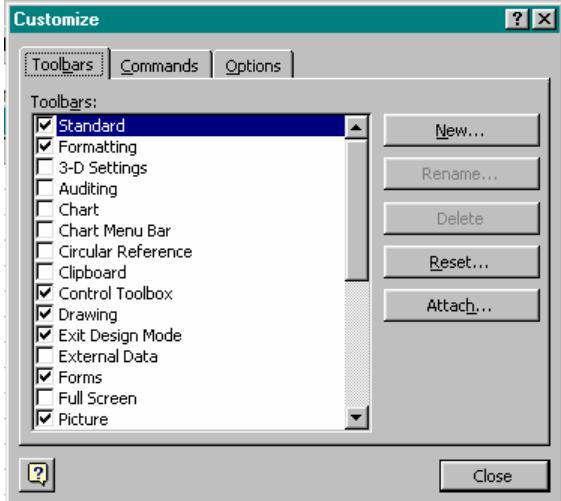


Figure 5: The Customize Dialog Box

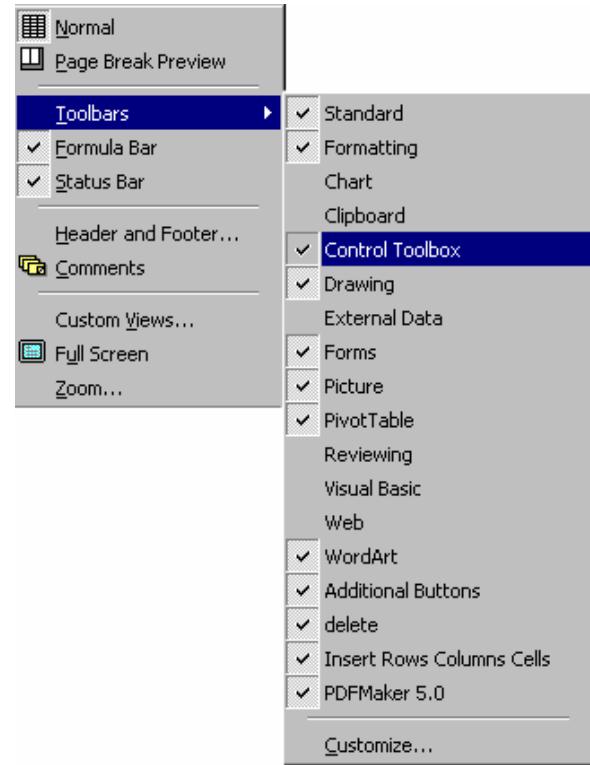


Figure 6: The Toolbars submenu access through the View menu.

Once you have added the Control Toolbar,



Figure 7: The Control Toolbox Toolbar

you can move it to any position on your screen or place it at the top of your screen with any other toolbars that you ordinarily use. The first button on the toolbar allows you to open and close the design mode. To create and to edit the components on the Control Toolbox Toolbar, you must be in design mode. The second button allow you to view the properties of any selected components on the page of the workbook while the third button allows you to edit the code associated with the components. The next buttons are used to create the check boxes, text boxes, command buttons, option buttons, list boxes, combo boxes, spin buttons, scroll bars, labels, images, and more controls, respectively.

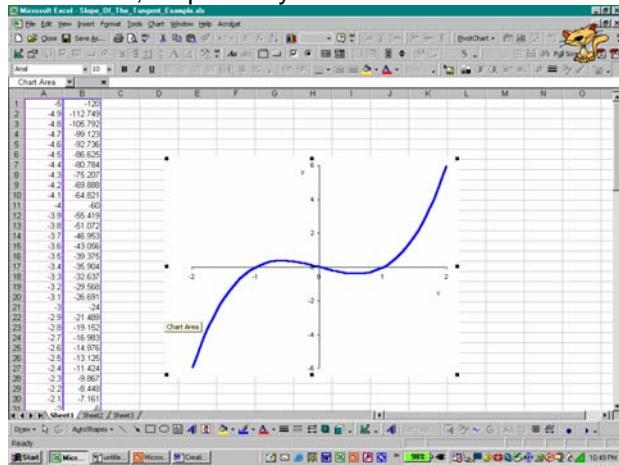


Figure 8: The graph and data for a slope of the tangent line analysis for the function $y = x^3 - x$.

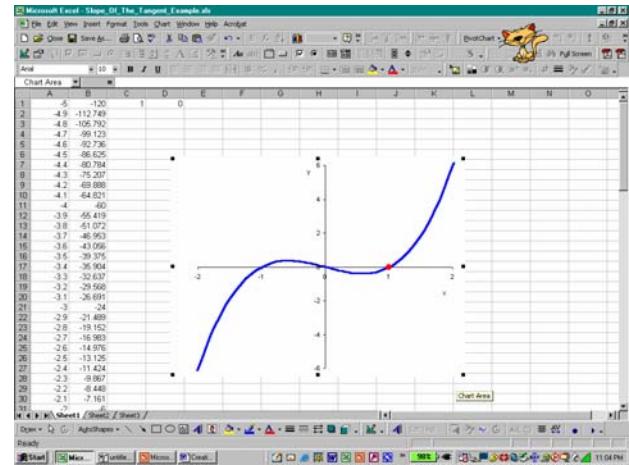


Figure 9: The point (1, 0) added to the graph.

So, let us create an interactive workbook that will allow students to explore the slope of the tangent line to a cubic function, the traditional changing of the secant line into a tangent line. To do this, we first must create the graph of the function for our analysis. Suppose we use $y = x^3 - x$ on the interval $[-2, 2]$. We create the graph in

the same way in which we would create any other MS Excel graph as displayed in Figure 8. Suppose we examine the slope of the tangent line to the graph of the function at $x = 1$ and that we allow the students to explore the slope of the secant line as the second point varies as x changes from -2 to 2 in the interval, of course not allowing x to be 1 . After adding the point $(1, 0)$ to the graph, we can add a scrollbar that we will use to move the second point along the curve. To do this, we must first enter design mode by left clicking the design mode button on the Control Toolbox Toolbar. Then, we can select to create a scrollbar by left-clicking the scrollbar button.

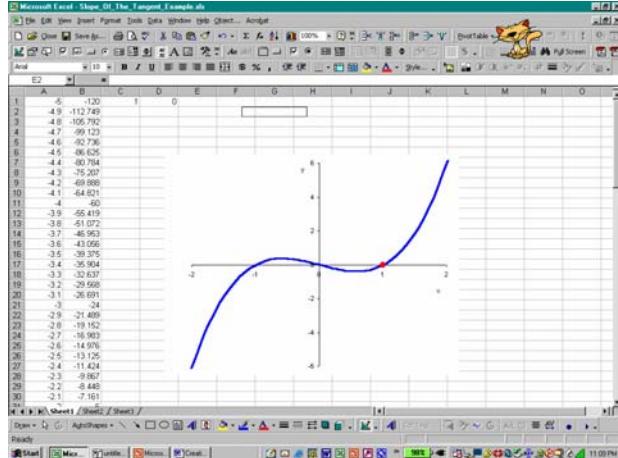


Figure 10: Creating the scrollbar by tracing it on the page.

Then, we can trace our scrollbar on the page to the desired height either horizontally or vertically, Figure 10. When we release the mouse after tracing the scrollbar, we create a gray scrollbar on the page. We can then change the properties of the scrollbar including the color, size, and orientation as desired.

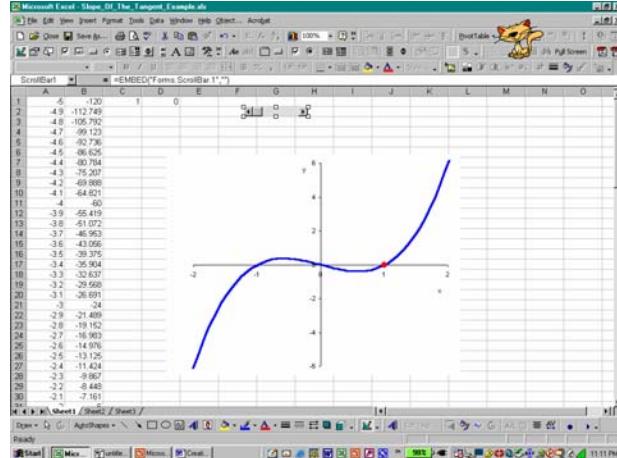


Figure 11: The scrollbar.

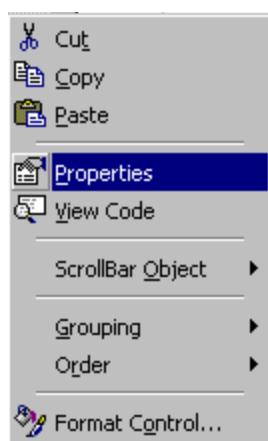


Figure 12: The Properties entry on the pop-up menu displayed by right-clicking the scrollbar.

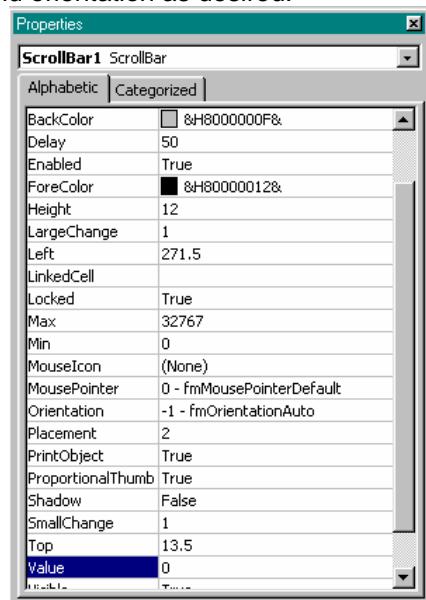


Figure 13: The Properties dialog box for the scrollbar.

To change the properties for the scrollbar, we can left-click the Properties button on the Control Toolbox toolbar or right click the scrollbar and select the Properties entry on the pop up menu, Figure 12. Either will open the Properties dialog box for the scrollbar, Figure 13. We can edit the properties using the alphabetical or the categorical listing of the properties, Figures 13 and 14. The categorized listing is easier to use since similar properties are grouped together, for example, appearance elements such as the BackColor, ForeColor, and Orientation are listed together. Here, we will use the categorized listing. To change any of these values, we will either left-click to select the property and enter the desire information or we will left-click to open the pull-down menu and select the desired entry. To finalize the information as we enter it, we press the enter key; this is important for information that we directly enter in this dialog box since we must finalize the information that we enter before we edit another entry.

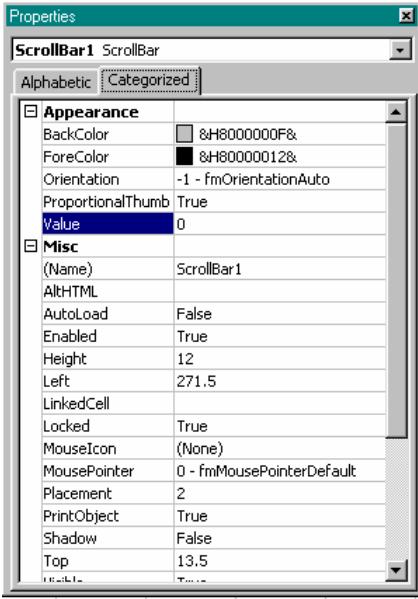


Figure 14: The Categorized tab of the Properties dialog box. Left-clicking the back color and the fore color entries allows us to select the colors for the scrollbar. If you would like to select colors other than shades of gray then use the Pallet tab as displayed in Figure 16.

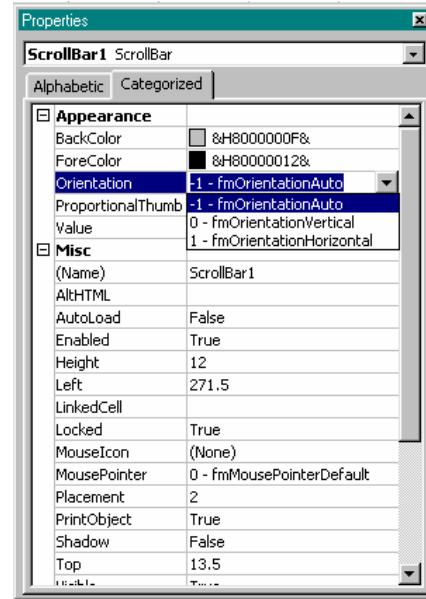


Figure 15: Changing the orientation for the scrollbar.

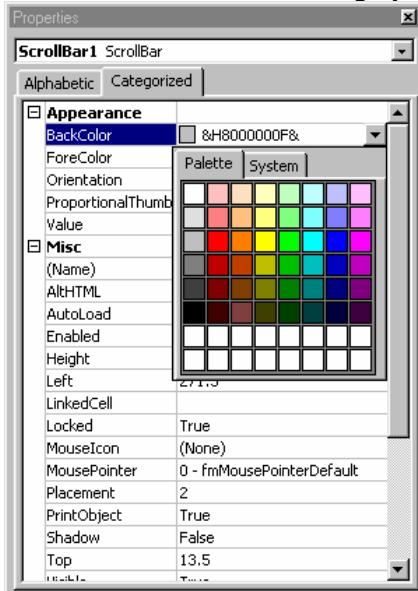


Figure 16: Changing the back color for the scrollbar using the pallet tab.

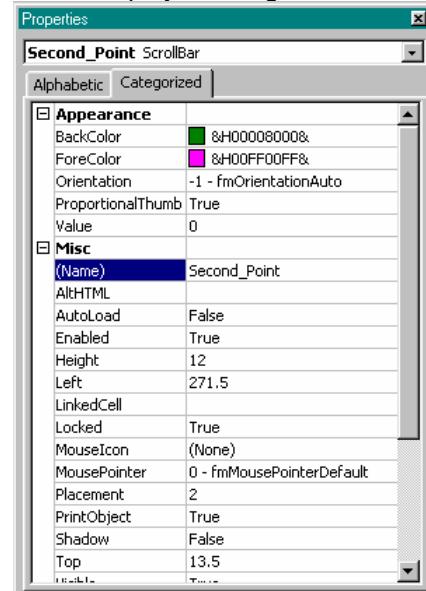


Figure 17: Naming the scrollbar.

After selecting these colors, we can name the scrollbar using the Name entry, Figure 17; we left-click and enter the desired name. While it is not necessary to name the objects as you create them, you will find it helpful if you have multiple scrollbars, for example, on a page. After naming the scrollbar, the new name will appear at the top of the Properties dialog box.

Next we set the maximum, minimum, and change values for the scrollbar. These must be positive numbers with, of course, the maximum larger than the minimum, and the change value must be a natural number. Here, we select a change of 1, a minimum of 0 and a maximum of 40, Figure 18. We will change these to the desired values using one of the cells. We must also choose a linked cell into which to put the value generated by the scrollbar. Since I prefer to display the values in our interval near the scrollbar, we will use cell E1 as our linked cell. If you would like to change the pointer, you can do that by selecting the mouse pointer menu. In addition to the listed pointers, you can select a pointer using the last entry on the menu; for this, you must select a pointer from the available Windows pointers or an image that you have saved on your hard drive.

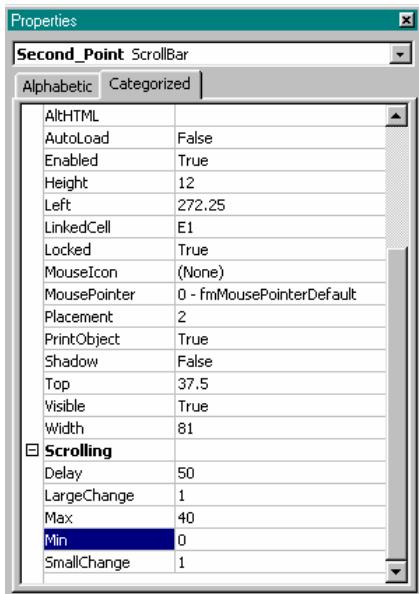


Figure 18: Setting the linked cell, the maximum value, and the minimum value for the scrollbar.

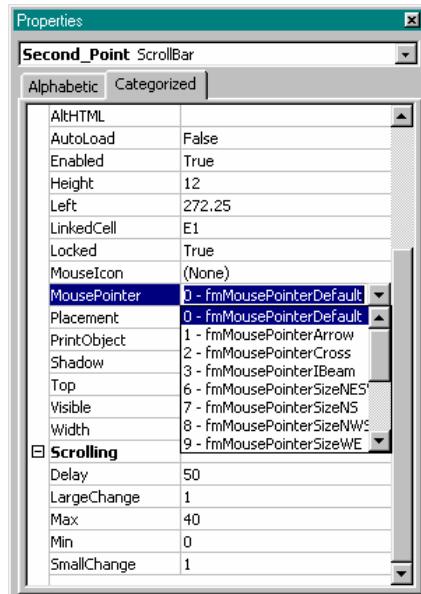


Figure 19: Changing the pointer.

When we have finished setting the desired properties, we exit design mode by left-clicking the design mode button on the toolbar. Once we have done this, we can examine the values of our scrollbar, Figure 20, checking to see that they vary from 0 to 40 in steps of 1 as set. Then, by entering a formula, here $(e1 - 20)/10$, to allow the values to change from -2 to 2 in steps of 0.1. We can use similar formulas to allow the values to change using other increments as desired.

E	F	G	H
31			

Figure 20: The scrollbar produces values from 0 to 40.

So, as we see in Figure 22, the values in cell G2 vary from -2 to 2 as desired. Once we have the scrollbar values set, we can format and color the displayed number and use it to create the slope, Figure 23, as well as data with which to graph the secant line.

E	F	G	H
29			

Figure 22: Values from -2 to 2.

$= (e1 - 20) / 10$				
C	D	E	F	G
1	0	27		

Figure 21: Creating the values that range from -2 to 2.

In Figure 24, we set up the data for the secant line. As with any graph, you can choose how much data to use. Here, we will graph the scrollbar from $x = -2.3$ to $x = 2.3$ so that it will be easy to see on the graph for all values.

To help students to learn and to question, it can be helpful to “cause trouble”. What do I mean by “cause trouble”? Well, include scrollbar values that may be meaningless, for example, values of the base of a logarithm that are impossible such as 0. In this case, if we would like to “cause trouble” for the students, we can create the data and allow the division by zero in the slope that will occur when the scrollbar value is 1. One advantage to doing this is that it allows for a lot of discussion and *explain* questions if you use the workbook as part of an assignment. The disadvantage is that without guidance, students will accept the graph of the tangent line that MS Excel produces without question. If you choose to display this line, I recommend that you post questions on the page that will guide the students and help them to understand that the graph of this secant line to be incorrect. So, for this scenario, every value produced by the scrollbar will have a corresponding secant line in the graph.

C	D
1	0
$=((g2^3 - g2) - 0) / (g2 - 1)$	

Figure 23: Creating the slope of the tangent line in cell C3.

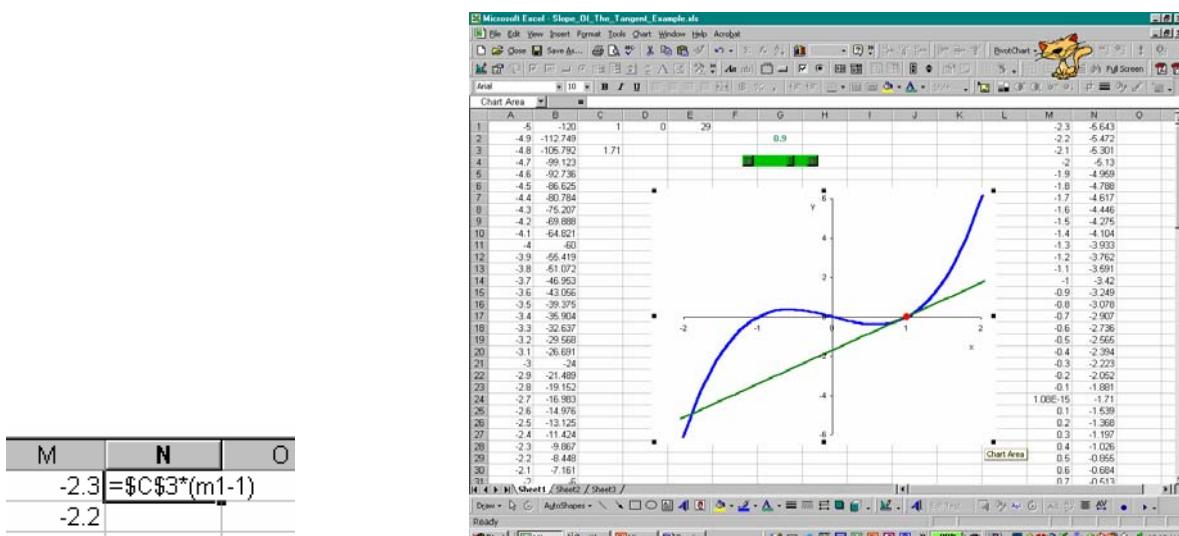


Figure 24: Create the data to graph the secant line.

If you choose to eliminate the erroneous secant line, you can do this quite easily using an If function when you enter the formula to generated the y-coordinate of the secant line data as displayed in Figures 26 and 27. It is important to remember to use appropriate notation, \$'s, to keep the scrollbar value and the slope value from changing when you use the automatic fill on the spreadsheet.

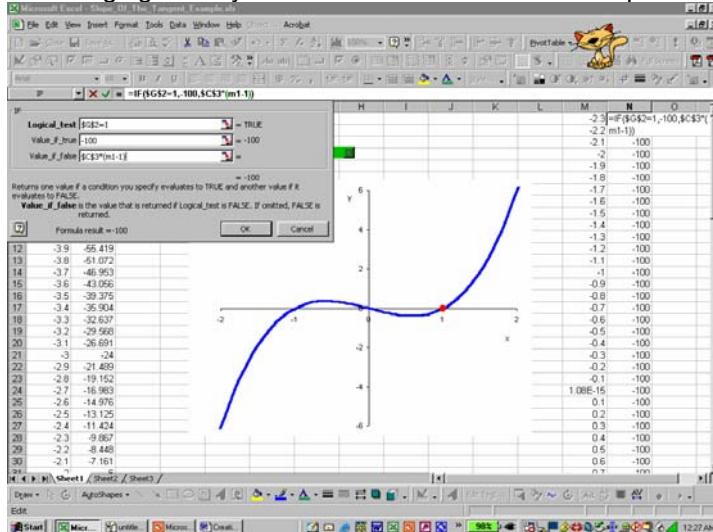


Figure 26: Using an If function to eliminate the graph when the scrollbar value is 1.

Setting the true-value in the If function allows us to prevent the graph from being displayed when the scrollbar value is 1. So, when the scrollbar value is 1, no graph will be displayed for the secant line as displayed in Figure 28 while all other secant line graphs will be displayed, Figure 29.

This too has its downside since students may not question why or even notice that there is no secant line displayed when the scrollbar value is 1. Again, a few well-designed questions will help a great deal and guide the user to do the appropriate thinking and analysis. If you choose to include elements to “cause trouble” in your workbooks, it is important to design these questions carefully so that the user will be able to understand the question(s) and to understand the concepts and mathematics necessary to answer the question(s).

With the scrollbar complete, we can hide the graph data, and eliminate the grid. What we need to do is make the background and the font colors match; other ways in which to hide the data will be mentioned in the later discussion. Since I prefer a white background, we will simply make the font for the graph data be white and then we only need to turn off the grid.

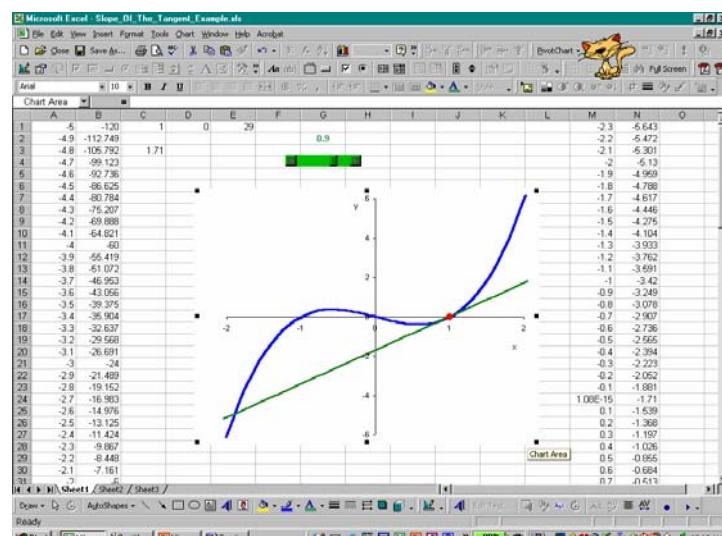


Figure 25: The secant line.

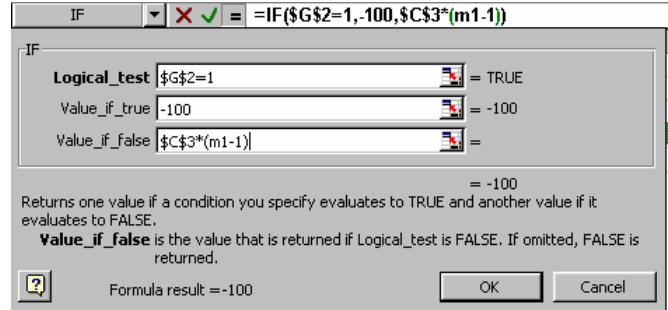


Figure 27: The If function.

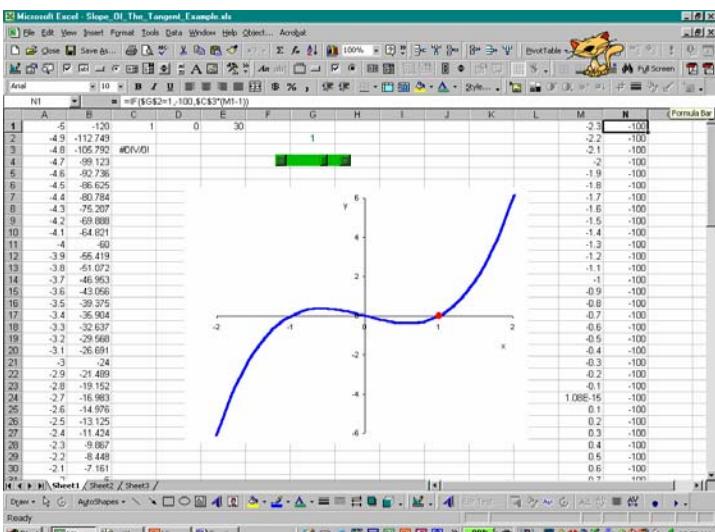


Figure 28: The graph of the secant line disappears when the scrollbar value is 1.

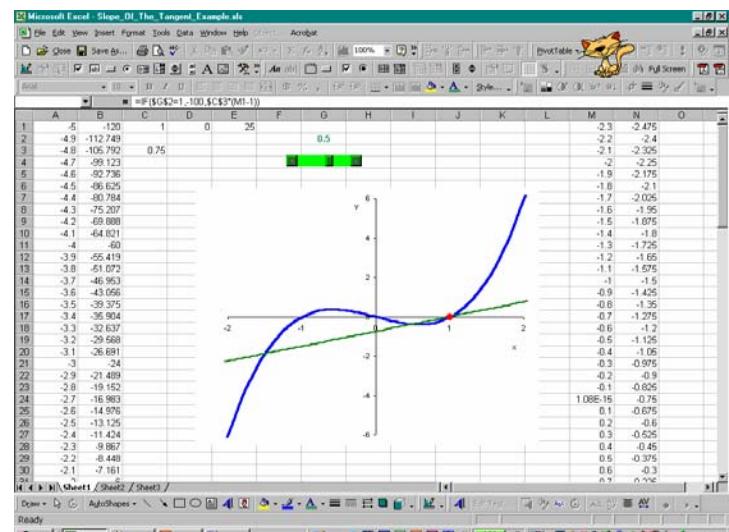


Figure 29: The graph of the secant line for scrollbar values other than 1.

So, as is Figure 30 and Figure 31, we select the desired columns and then change the color of the font.

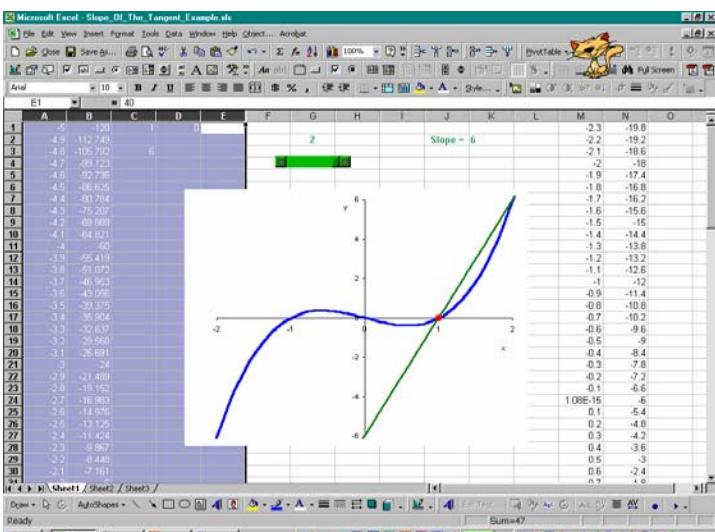


Figure 30: Hiding the data for the function.

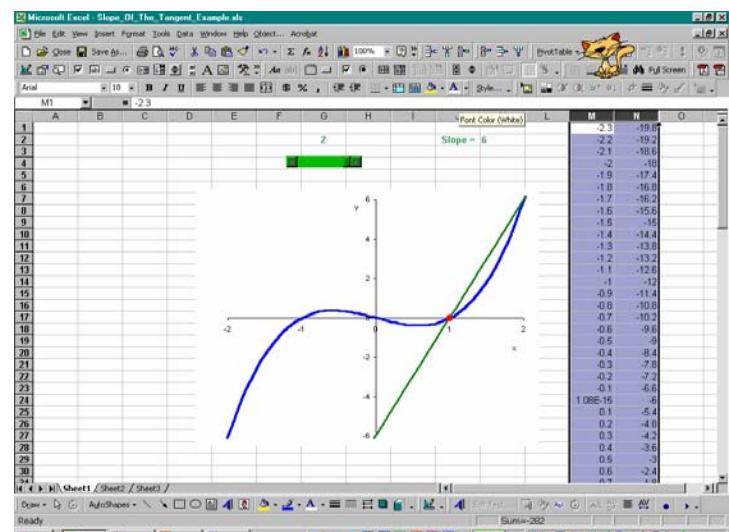


Figure 31: Hiding the data for the secant line.

After changing the font color for the data and turning off the grid, we have can name the page, and if we are not going to use the other pages in the workbook, delete the extra pages.

We can add other objects such as check boxes, text boxes, command buttons, option buttons, list boxes, combo boxes, toggle buttons, and spin buttons and pages with multiple activities to our workbook to create guided learning experiences for students as well as workbooks that can act as stand-alone assignments and projects.



check boxes text boxes command buttons option buttons



list boxes combo boxes toggle buttons spin buttons

For now, ...

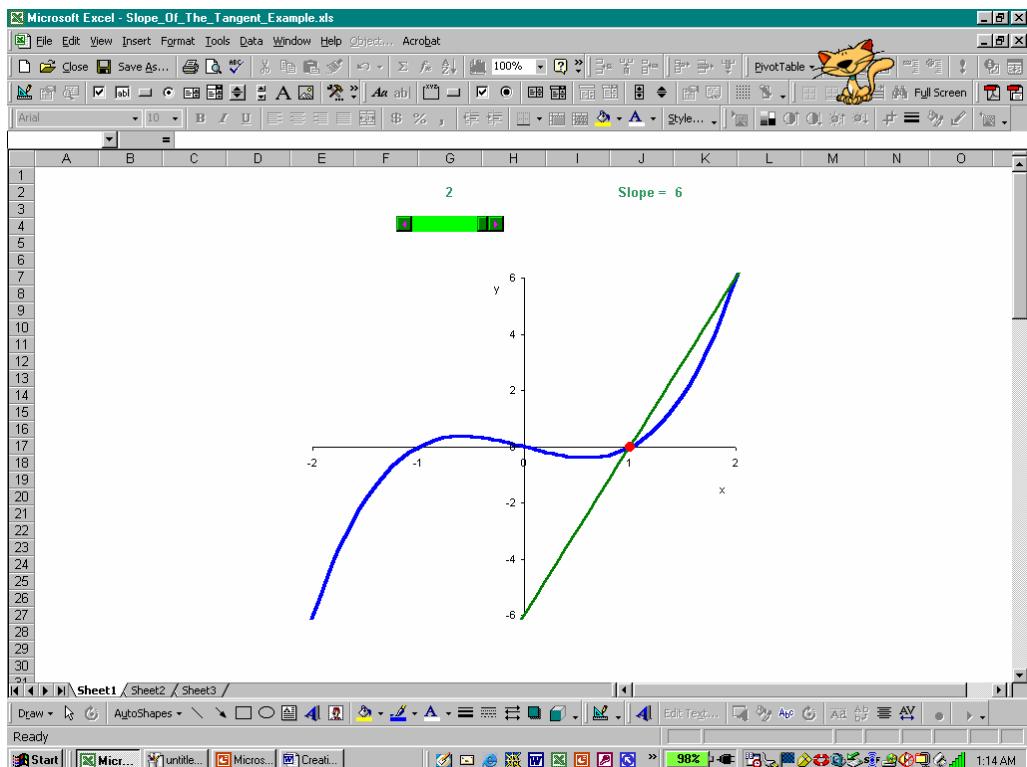
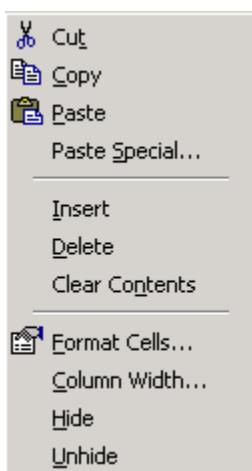


Figure 32: Turn off the grid to make the background white.

We see our completed workbook in Figure 34. While we can hide the data for the graph by changing the font color to match the background, we can also simply hide the columns that contain the data as well. A very simple way in which to hide the data is to position the data out of view. What does that mean? Well, we can insert columns on the worksheet so that the data that we want to hide is not close to the worksheet elements that we want to display in the MS Excel window. The downside to doing this is that we do not know the Screen Area settings on the computer on which the user is accessing the workbook. If one designs a workbook using a computer with Screen Area settings of 800 pixels by 600 pixels then one viewing this workbook using Screen Area settings of 1024 pixels by 768 pixels may be able to view data that is suppose to be out of "view".



Hiding the columns can be beneficial if you plan to show the data since you will be able to unhide the columns easily. To hide columns in MS Excel, simply select the columns by left-clicking-holding-and dragging the cursor until the desired columns are highlighted and then right-click this section to display the pop-up menu displayed in Figure 33. To hide the selected columns, we select the Hide option on the menu. When we want to unhide the columns, we select the columns around the hidden columns and then right-click this section to display the same menu on which we would now select the Unhide option.

In general, I find it best to hide the data by making the data the same color as the background since hiding columns can affect graphs and objects in the same region as the hidden columns. In addition, except for when I am showing others how to create interactive workbooks, I rarely show the underlying data.

It is always a nice finishing touch to rename the sheet in a meaningful way and to

Figure 33: The pop-up menu displayed when you right-click a column in MS Excel

delete any sheets that are not used in the workbook; unused worksheets can be a distraction to the user. Naming the worksheets helps the user understand the components of the workbook and helps the user to navigate the workbook; it is important to put the worksheets in a useful, logical order as well. If you create a workbook for which the user needs special instructions, it can be helpful to provide information/instructions for the use of the workbook. Using one sheet, the sheet

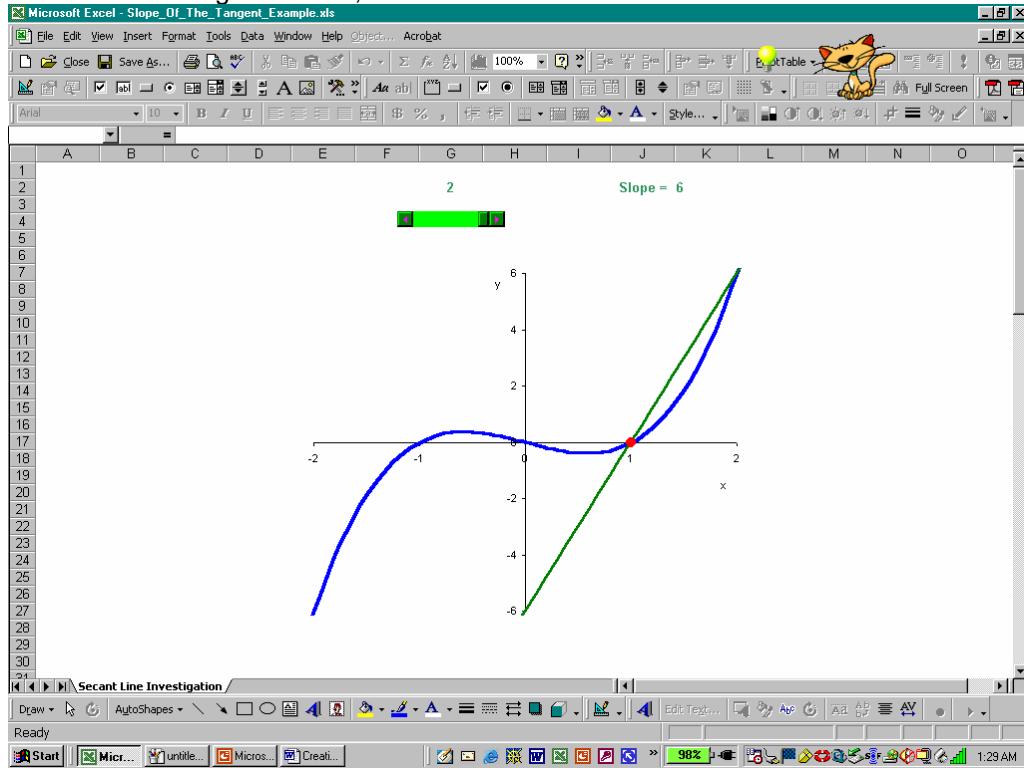


Figure 34: The workbook with one page.

to which the workbook first opens, as an instruction page will make the workbook more user-friendly. Finally, before you save and close the workbook, it is very important to make sure that you have set the scrollbar to the first value that you would like the user to see and select the page to which you would like the workbook to open. With this done, the workbook can be saved and closed, and then, the workbook is ready for use.

Now that we have created our first workbook with one element, a scrollbar, let us create a second workbook that contains three scrollbars, a pop-up text box, and a command button. Suppose you would like to create an interactive workbook that will allow students to explore the standard form of a quadratic function. Since the standard form for a quadratic function is $y = a(x - h)^2 + k$, we need a scrollbar for a , h , and k . As before, we

"turn on" the Design mode by left-clicking the Design Mode button on the Control Toolbox Toolbar, and then

left-click the Scrollbar button. We select the location, direction, and dimensions for the scrollbar by tracing the desired scrollbar on the worksheet; we release to generate the scrollbar. If you would like to generate three scrollbars of the same size then it is best either to copy the scrollbar twice more onto the worksheet or to set the dimensions of the other scrollbars to match the first when you are editing the scrollbar properties. For this scrollbar, Figure 35, we will create the first scrollbar, edit its properties, and then generate the second and third scrollbars by copying the first scrollbar and editing the properties of the new scrollbars.

To set the properties for the scrollbar, we can right-click the scrollbar and select Properties on the pop-up menu displayed in Figure 36 or left-click the Properties button, , on the Control Toolbox Toolbar. If you left-click the Properties toolbar button without selecting the scrollbar first, the Properties dialog box will display the properties of the worksheet rather than the properties of the scrollbar. This is not a problem since we can view the scrollbar properties using the pull-down menu at the top of the dialog box; these are displayed in Figures 37 and 38.

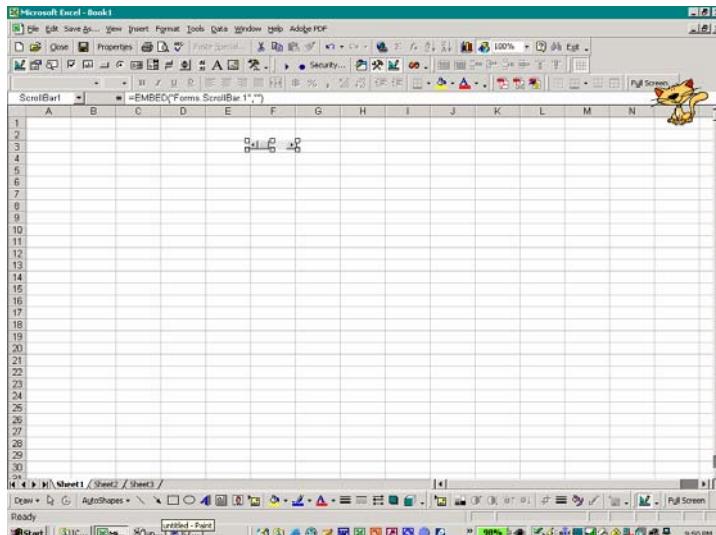


Figure 35: The newly generated scrollbar for which the properties must be set.

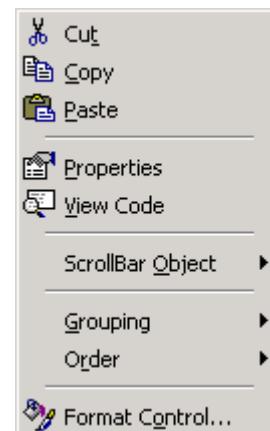


Figure 36: Pop-up menu displayed when you right-click the scrollbar.

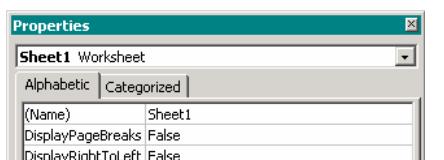


Figure 37: The properties dialog box displays the properties for the worksheet if the worksheet was selected.

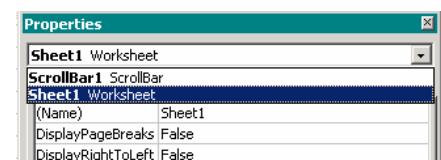


Figure 38: To display the properties for the scrollbar, we use the pull-down menu at the top of the properties dialog box.

Now, we set the colors, the scrollbar name, if desired, the linked cell, and the maximum, minimum, and change values for the scrollbar. We can select a mouse icon for the scrollbar as well as add visual effects such as a shadow.

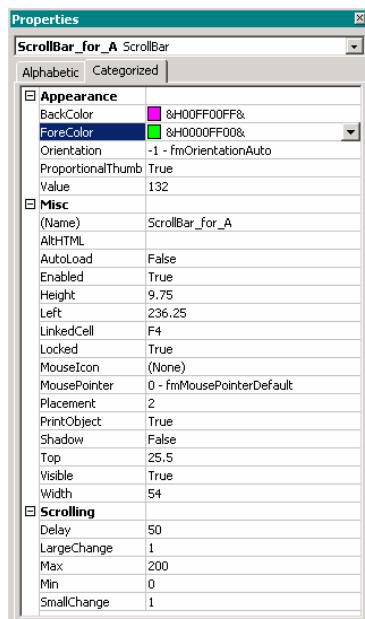


Figure 39: The properties for ScrollBar_for_A.

Appropriately, displaying a minimum value of 0 and a maximum value of 200 for each in the linked cell as displayed in Figure 40.

Here, we have set the background color to magenta, the foreground color to be a lime green, named the scrollbar ScrollBar_for_A, chosen cell F4 as the LinkedCell, set the maximum value for the scrollbar to be 200, the minimum value to be 0, and the increment as 1. When we copy this scrollbar, we must change the LinkedCell and, if desired, the maximum, minimum, and increment amounts for the scrollbar. For these scrollbars, although I will select different colors, I will use the same maximum, minimum, and increment values.

After creating the scrollbars and editing the properties for each, it is important to test them. To do this, we must exit Design mode by left-clicking the depressed

Design Mode toolbar button, . Then, using the bar or the arrows in the scrollbar, we see that the scrollbar works



Figure 40: The tested scrollbars.

Since the maximum and minimum values for the scrollbar must be between 0 and 32,767, inclusive, we must use a formula to generate negative values, if desired. Since I would like these scrollbars to vary from -10 to 10 by tenths, I used the formula “=(J4-100)/10” in cell J2 and similar formulas in cells F2 and H2; this is one of the many possible formulas. Now, testing the scrollbars, I find that they vary, as desired, from -10 to 10 in increments of 0.1.

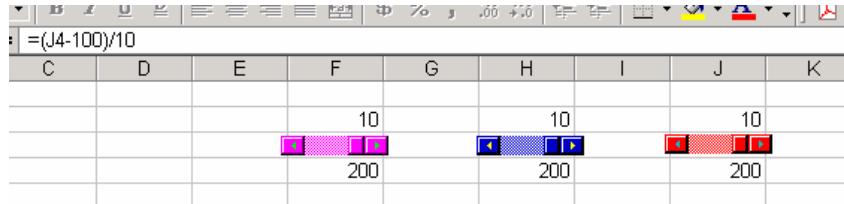


Figure 41: The scrollbars and the formula entered in cell J2.

Since we have created and set the scrollbars for the function coefficients A, H, and K, we are ready to create the data and the graph. We create the data so that the function the coefficient values for A, H, and K in cells F2, H2, and J2, respectively. Since the input values will be entered in column A, we use the formula “=\$F\$2*(A1-\$H\$2)^2+\$J\$2” in cell B1; we must use the dollar signs (\$) around the column names for the cells containing the function coefficients since we want to fix these cell locations in the formula.

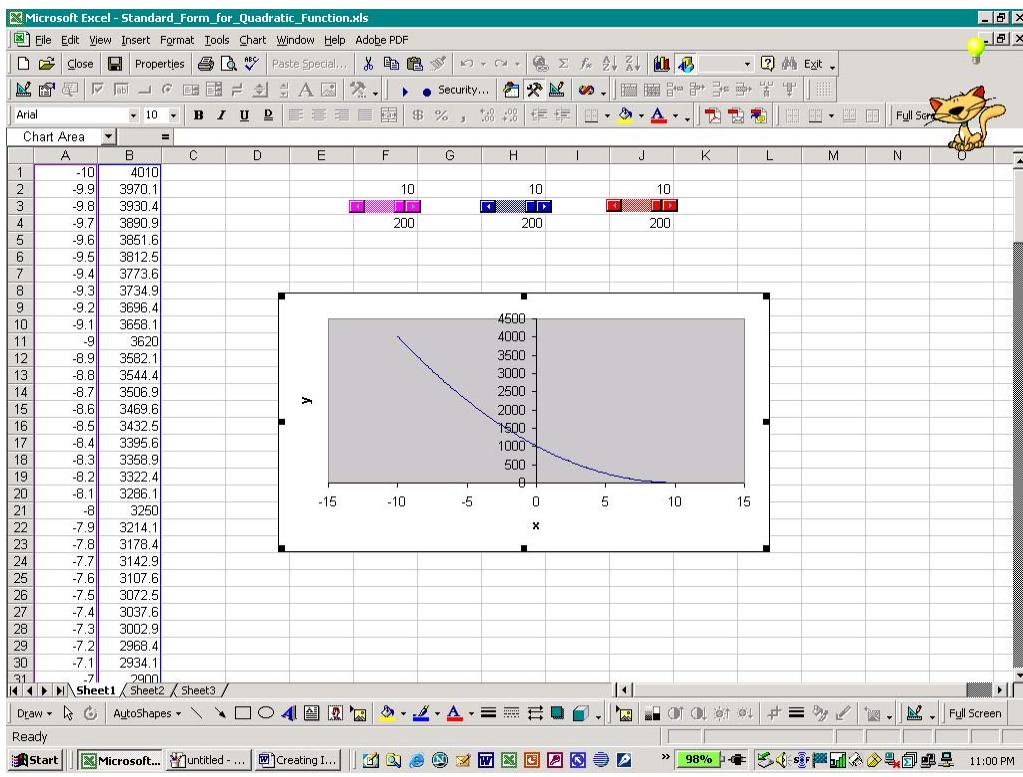


Figure 42: The worksheet with data and graph.

In Figure 42, we see the data and the graph for the data in the worksheet. Here, the data changes in increments of one-tenth. While some like to allow the axes to change as the user changes the parameter values, I prefer to fix the graph window to allow the user a more “continuous” view. It is important to note that if the graph window is not set then the graph window will change to accommodate the changing data values; this can be very distracting for the user.

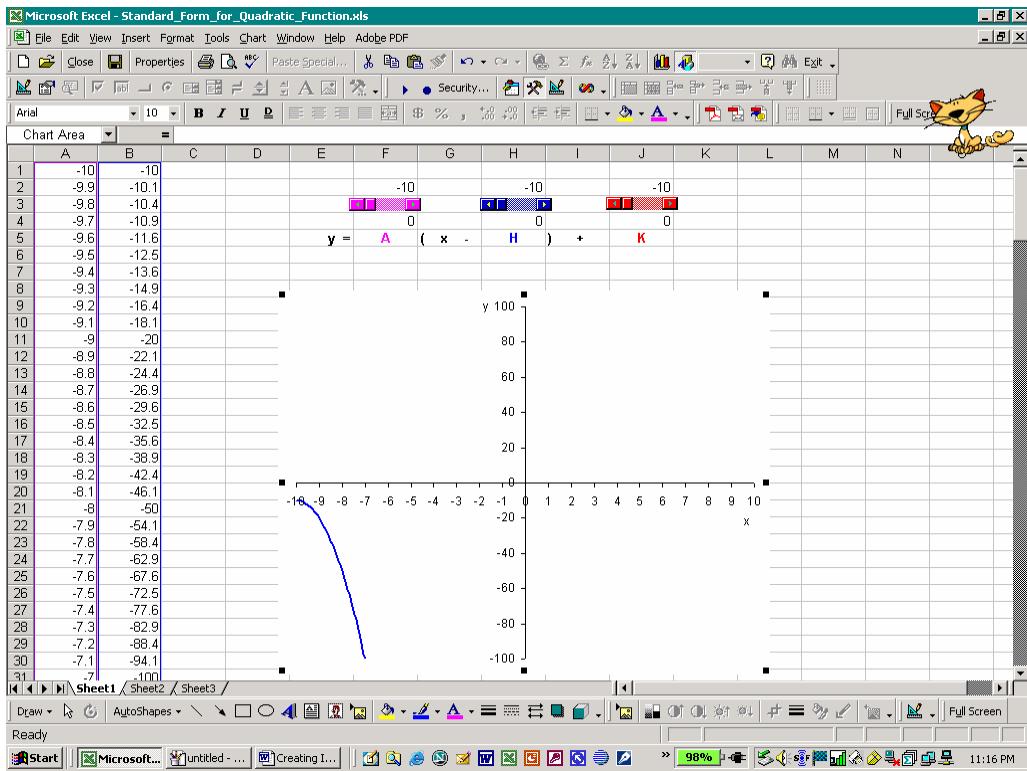


Figure 43: The worksheet with modified graph and function equation.

Figure 43 displays the worksheet with the modified graph: the graph window set to display inputs between -10 and 10 , inclusive, and function outputs between -100 and 100 , inclusive, the graph background has been set as white, the curve weight and color have been changed and a color-coded equation has been added near the scrollbars.

We are now ready to add the command button that will generate the pop-up text box. To create the command button, we enter Design Mode and left-click the Command Button toolbar button, , on the Control Toolbox Toolbar. We trace the button in the desired location as we traced the size of the scrollbar before. We will select text for the button and set the properties as we did for the scrollbars. Here, we will create a command button that can be used to provide help. The caption of the button has been set as “Do you need help?”.

The size of the button, like the size of the scrollbar, can be set/adjusted using the properties dialog box. The dimensions of the button can be adjusted using the size tab in the Format Control dialog box that can be accessed by right-clicking the button and selecting the Format Control option on the pop up menu as well. Using the CommandButton Object option on this pop up menu, we can select Edit to edit the text directly on the button. It is important to note that double-clicking the button opens the Microsoft Visual Basic editor. The Microsoft Visual Basic editor allows one to edit the Visual Basic code for the button as well as to create/enter a macro that can be run by left-clicking the button; Design Mode must be turned off in order to run the buttons and scrollbars on the sheet.

To generate the pop-up text box, we must create a macro. To create a Macro, we determine what we would like to generate and then “record” the creation of this element. Then, we can attach this macro to the button so that when the user left-clicks the button the text box will appear on the screen. Before beginning the macro, it is important to make sure that all the elements are in their desired locations; I changed the position of the Help_Button slightly from that displayed in Figure 44.

With all the elements in their desired locations, we can begin the macro. We exit the Design Mode and select Macro on the Tools menu; the Tools menu is displayed in Figure 45. On the Macro submenu, we select Record New Macro. In the Record Macro dialog box, we name the macro and click Ok to begin.

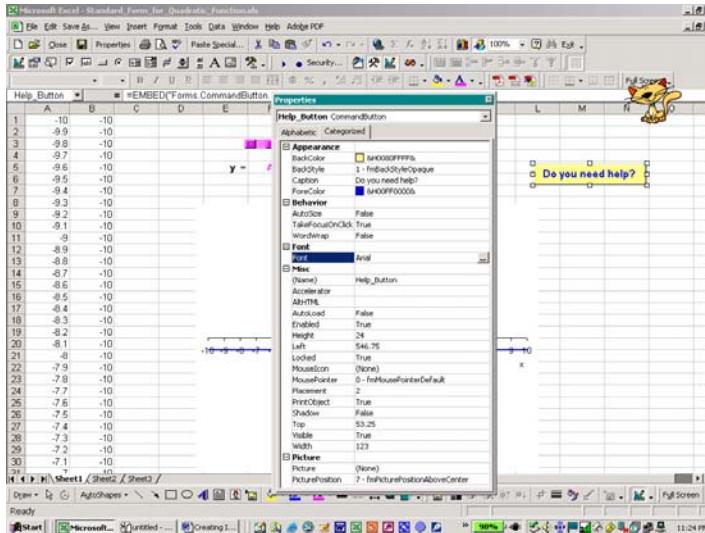


Figure 44: The Properties for the Help_Button.

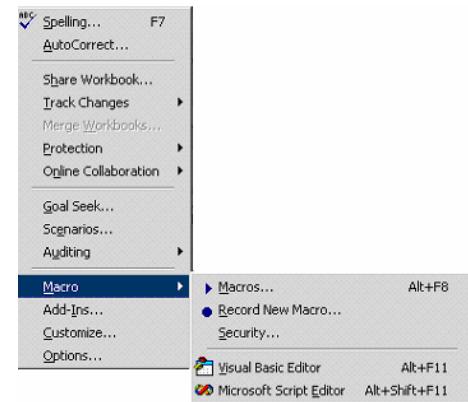


Figure 45: The Tools menu.

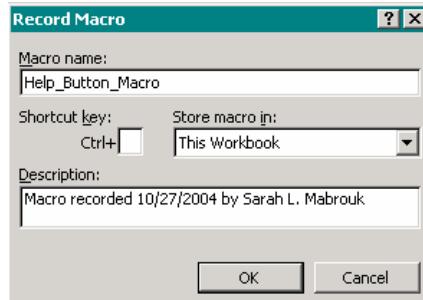


Figure 46: the Record Macro dialog box.

When we finish generating the macro, we click the stop button, . We use the text box button, to generate a text box; as with other elements on the sheet, we left-click and drag to generate the box of the desired size and shape. Then with the box placed, we add the desired text, in this case instructions on how to use the scrollbars to change the function coefficients, and select the font and font color for the text; right-click

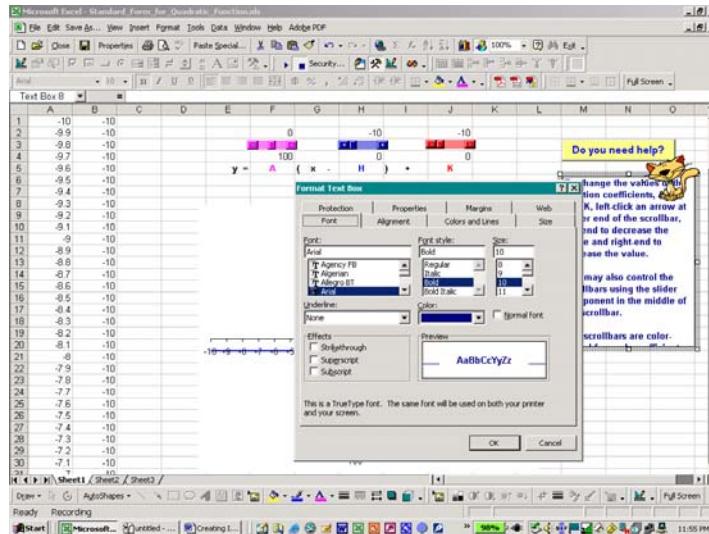


Figure 47: The Format Text Box dialog box.

the text box edge, and select Format Text Box change the text box properties in the Format Text Box dialog box. Here, I have set the text box colors to be the same as the Help_Button colors.

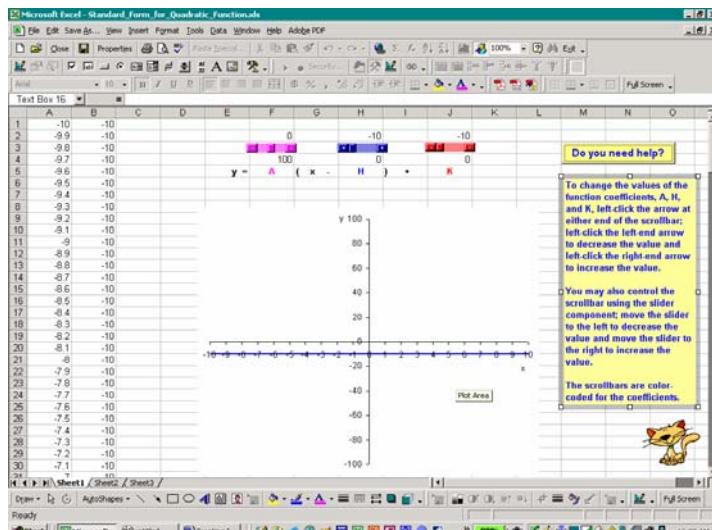


Figure 48: The new text box.

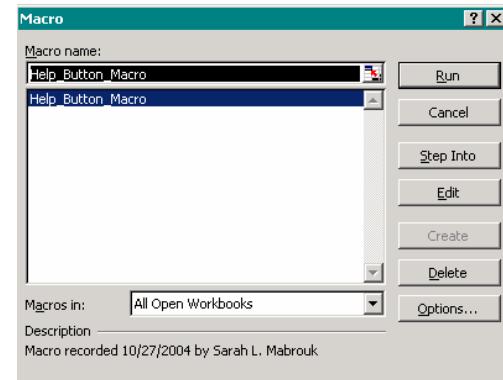


Figure 49: The Macro dialog box.

Now that the text box is in place. I will stop recording the macro by left-clicking the Stop button. To view the macro code and to attach this macro to the button, we select Macro on the Macro submenu of the Tools menu and then left-click the Edit button. All we need to do is copy the Visual Basic code for the Macro into the corresponding location for the Help_Button in the MS Visual Basic Editor. If the Help_Button window is not open in the Visual Basic editor, you can enter Design Mode and double-left-click the Help_Button to open it. We copy all the code between "Sub Help_Button_Macro()" and "End Sub" and copy this code between the corresponding beginning and end code for the button, "Private Sub Help_Button_Click()" and "End Sub", respectively.

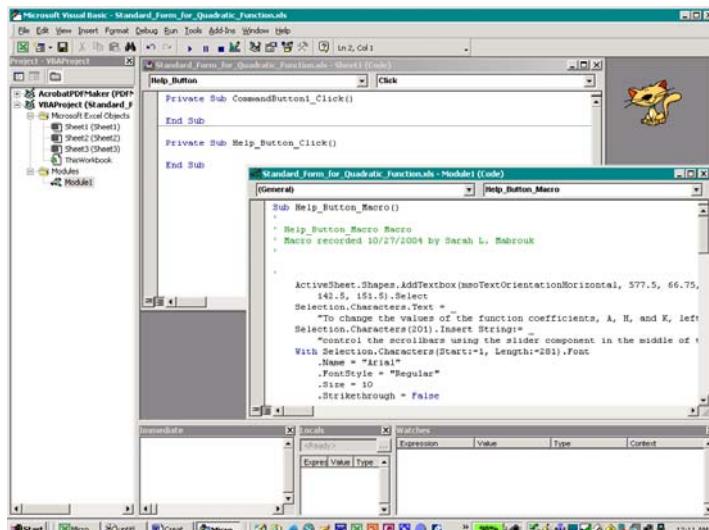


Figure 50: The MS Visual Basic Editor.

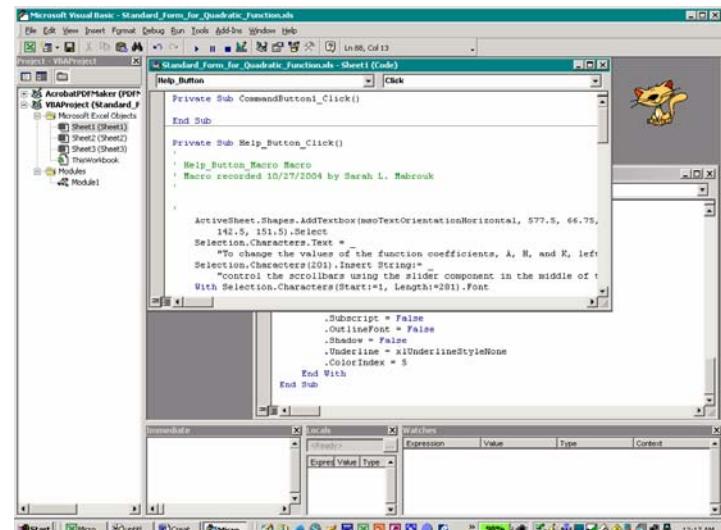


Figure 51: The Help_Button window in the MS Visual Basic Editor.

After pasting the macro code into place, we can close the Visual Basic editor and test the Help-Button. In this case, when we left-click the Help-Button, the text box will appear and remain selected within the worksheet. Leaving the text box selected provides added highlighting for the text. Adding a Close-Box is an added touch that the user may appreciate.

Right-clicking the button while in Design Mode, you may use the Format Control option on the pop up menu to adjust the height, width, and rotation of the button as well as to set positioning options relative to the cells on the worksheet, to set the button as a print object on the sheet, and to include alternate text for web search engines. The same pop up menu provides options for grouping and ordering elements on the sheet. The grouping and ordering options can be helpful if you would like to create a group of buttons such as a group of option buttons that can be set to work together.

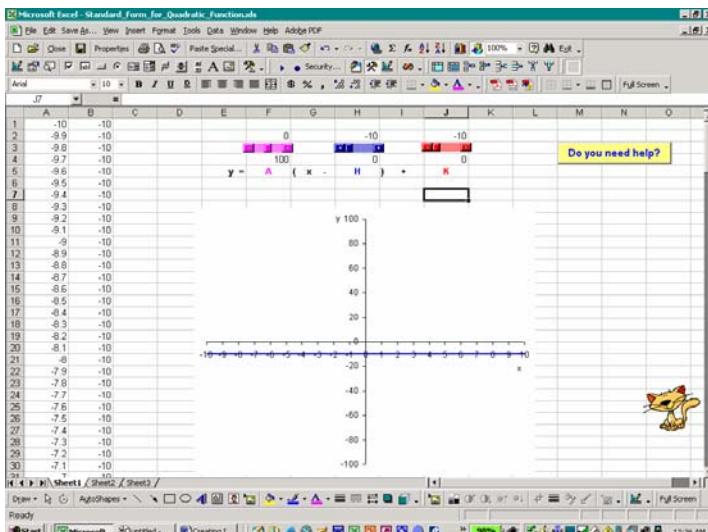


Figure 53: Left-clicking the Help-Button produces ...

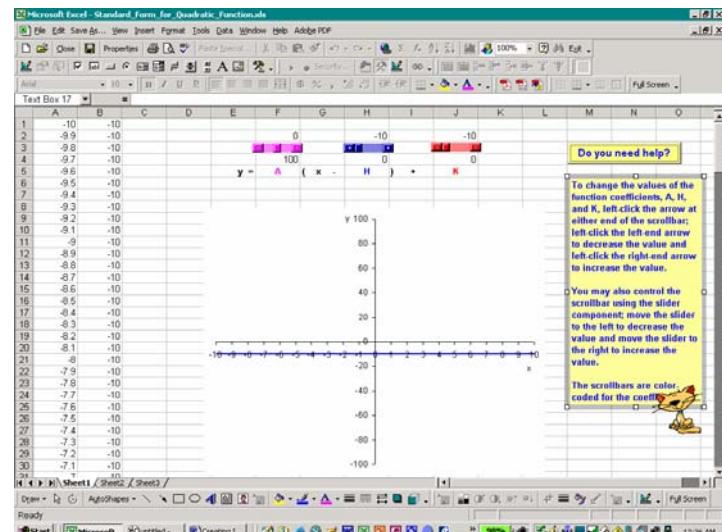


Figure 54: The text box generated using the Help_Button.

Final touches for the workbook include hiding the data, here using a white font to match the background, deleting the text box, deleting the extra sheets, naming this sheet, and color-coding the displayed values of the coefficients, hiding the values in the linked cells using a white font, eliminating the grid, and setting the initial values for the scrollbars.

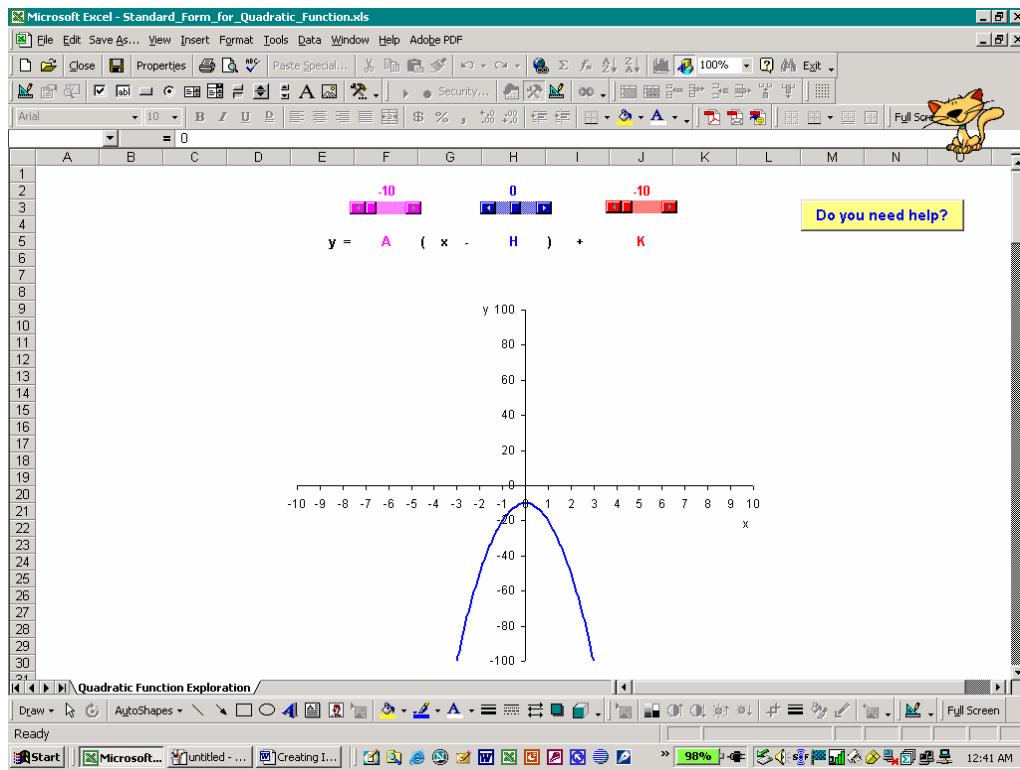


Figure 55: The finished interactive workbook.

It is important to remember that we must be in Design Mode in order to add buttons, scrollbars, and other elements to a worksheet. In addition, we must be in Design Mode in order to edit any of these elements. To

check that the Design Mode is active for a workbook, we examine the Design Mode button, , on the Control Toolbox Toolbar: if the Design Mode button is highlighted then the Design Mode is turned on for the workbook. Before we publish the workbook on a web site or send the workbook to folks for use, it is very important to make sure that the Design Mode for the workbook has been turned off; again, if the Design Mode button on the Control Toolbox Toolbar is highlighted then Design Mode is turned on for the workbook.

The other elements on the Control Toolbox Toolbar such as the Option Button, , the Spin Button, , and the List, Combo, and Toggle Buttons, , are created in a similar manner, and macros can be recorded for and attached to some of these elements as well.

Individual Option Buttons, , may have only one of two values, "True" or "False". If the Option Button is selected, indicated by the black dot at the center of the button, then the value of the button is "True" otherwise the value of the button is "False". Option Buttons be grouped to create multiple options for questions, only one of which can be "True" at one time. Once a group of Option Buttons has been created, the value for the buttons that have not been selected will be set to "False" once an individual button has been selected; only the value of the button that is selected will be set to be "True". To create a group of Option Buttons, we would create the individual Option Buttons and, after setting the properties for each, we group the buttons using the Grouping option on the right-click pop up menu. To group the buttons, we left-click the first button and then either shift and left-click or press Ctrl and left-click each additional button to select all the buttons for the group: we right-click the group of selected buttons and then select the Group option on the Grouping submenu on the pop up menu to group the buttons.

Unlike Option Buttons, Checkboxes, , cannot be grouped to work together. While we can group Checkboxes so that they can be moved on the worksheet to maintain their position relative to each other, Checkboxes work individually, producing a linked cell value of "True" when the box is checked and a linked cell value of "False" when the box is not checked. Checkboxes can be used any time one wants to generate values of "True" or "False". Again, however, unlike the Option Buttons, Checkboxes cannot be grouped to create a group of Checkboxes that work together, only one of which can be "True" at a time; this can only be done by creating a group of Option Buttons.

While the Textbox, , can be a useful element to include in an interactive workbook, it does have an important drawback, how the user enters the text. If the user does not enter the text that we anticipate then this text will not be considered to be "correct" by our conditional test. That is, if the user does not enter the correct spelling or anticipated terminology then the conditional test statement that we would use in our examination of the entered text would consider the text to be incorrect and produce a value of "False" for the conditional test. If we want to minimize the possibility that incorrect spellings or unanticipated terminology is used, we can provide the user with a list of words to enter in the textbox as well as a caution about incorrect spellings. To eliminate the need for this, we could simply use a group of Option Buttons.

The Forms Toolbar, , has many elements similar to those on the Control Toolbox Toolbar. Unlike the elements on the Control Toolbox Toolbar, those on the Form Toolbar are not ActiveX Controls. ActiveX Controls such as the Scrollbars, Command Buttons, Option Buttons, and Checkboxes that we have discussed provide the user with options or allow the user to run macros or scripts that automate a task. The macros for the controls can be written in Microsoft Visual Basic for Applications (VBA) or scripts in the Microsoft Script Editor. While macros can be added to elements on the Forms Toolbar, these macros can only be recorded and then attached to the toolbar elements: we cannot write or modify the macro code in VBA for elements on the Forms Toolbar.

For the elements on the Forms Toolbar, we do not have control over the various aspects of the elements such as the color of the element or the mouse icon that will be displayed when the cursor is near the element as we do with the ActiveX Controls on the Control Toolbox Toolbar. For example, the buttons and scrollbar elements on the Forms Toolbar will be basic gray in color and the default mouse icon that is set on the user's PC will be displayed when the cursor is near the element. We cannot view or change the properties of the elements on the Forms Toolbar. We set the basic information for these controls using the Format Control option on the right-click pop up menu associated with each element. The Format Control dialog box for the scrollbar on the Forms Toolbar is displayed in Figure 56. The size and orientation of the scrollbar can be set using the Size tab in the Format Control dialog box or by right-clicking the control to select it and then using the handles (the white circles

or black squares depending on the version of MS Office) at the corners to drag the scrollbar to the desired size and orientation. The values for the scrollbar, minimum value, maximum value, incremental change, and the page change, can be set using the Control tab; the incremental change is the amount by which the scrollbar value will change as we left-click either black end arrow, and the Page Change is increment by which the scrollbar value will change if we left-click the scrollbar between the two black end arrows.

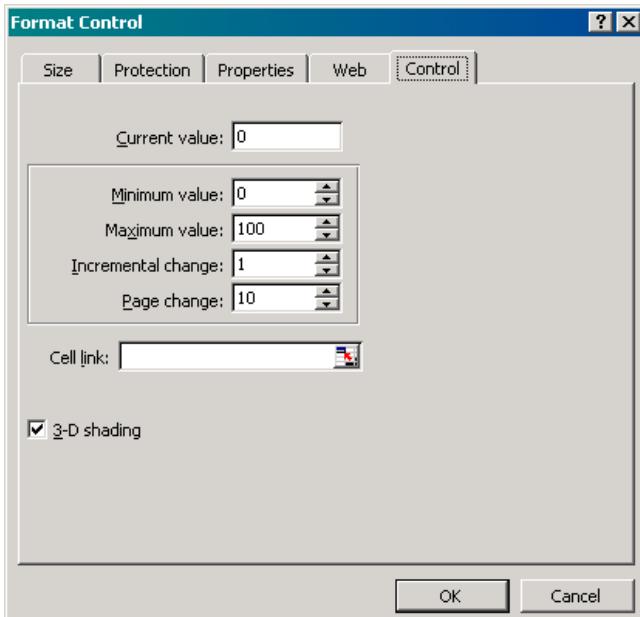


Figure 56: The Format Control dialog box for the Scrollbar on the Forms Toolbar

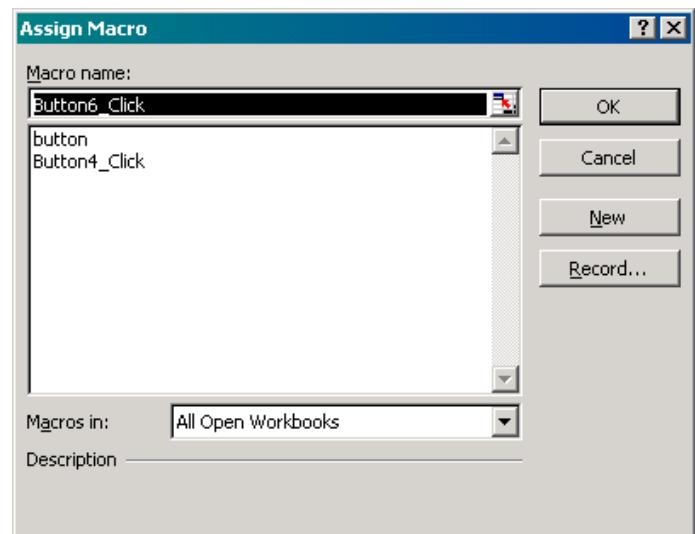


Figure 57: The Assign Macro dialog box for the Button on the Forms Toolbar

Unlike the Command Buttons on the Control Toolbox Toolbar, the macro to be assigned to the Button on the Forms Toolbar must be assigned when the button is created. When the button is created, the Assign Macro dialog box, Figure 57, will automatically open and we can select either a macro that has already been created or we can create a new macro for the button.

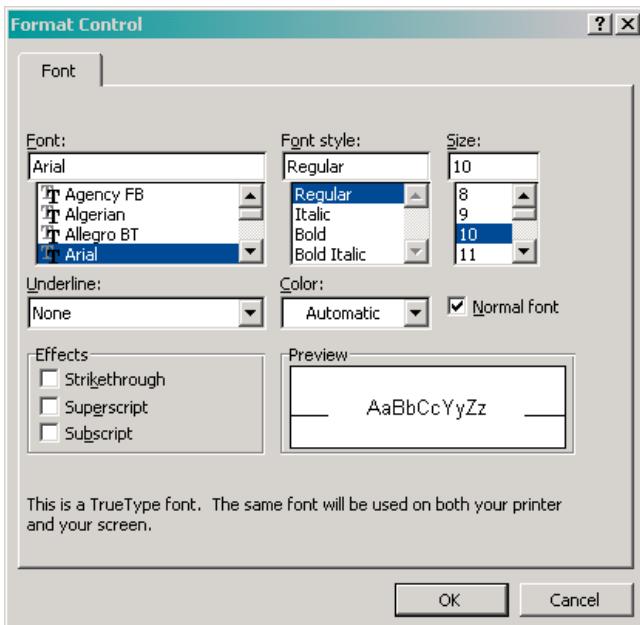


Figure 58: The Format Control dialog box for the Button (if no macro is assigned) on the Forms Toolbar

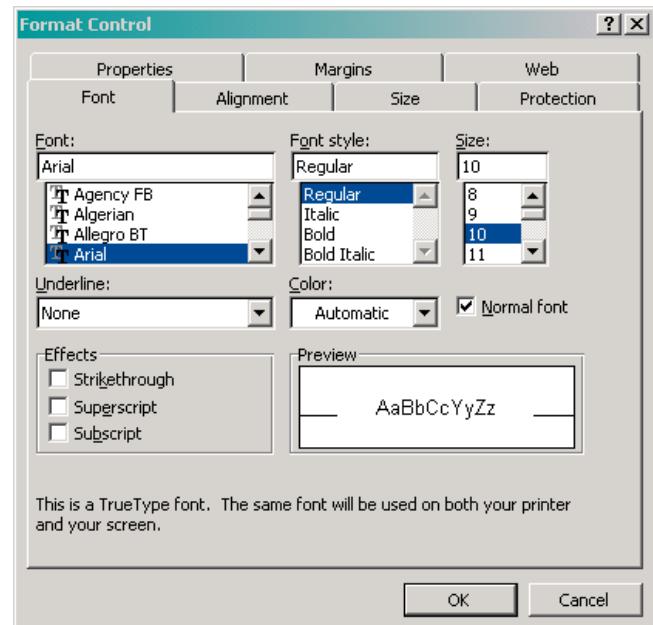


Figure 59: The Format Control dialog box for the Button (if a macro is assigned) on the Forms Toolbar

If we do not assign a macro to the button when we create it, the Format Control dialog box, Figure 58, associated with the button will appear different than the one associated with the button if we do assign a macro to the button, Figure 59.

Design elements such as color aside, the biggest disadvantage to the elements on the Forms Toolbar is that unless the workbook has password protection to lock the elements, right-clicking these elements allows the user to select the elements and then Cntrl-X or the Delete key will allow the user to delete these elements. If the user is not familiar with Design Mode or VBA, the user cannot delete the ActiveX Controls that can be created using the Control Toolbox Toolbar.

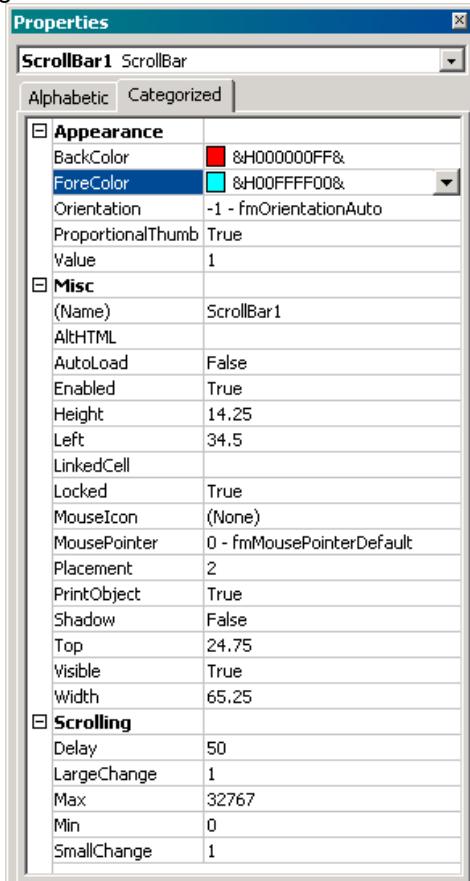


Figure 60: The Properties dialog box for an ActiveX scrollbar

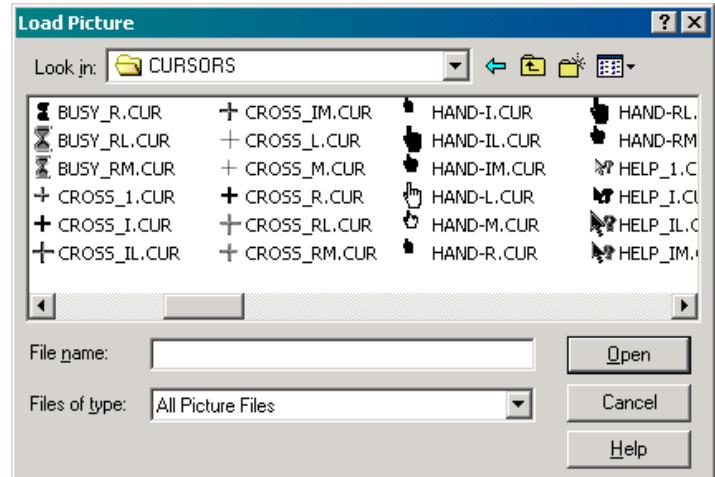


Figure 61: The Load Picture dialog box for selecting the MouseIcon

One of the nicest features to customize for ActiveX controls is the MousePointer. We customize this aspect of an ActiveX control using the MousePointer and Mouselcon options in the Properties dialog box associated with the control; the Properties dialog box for a scrollbar is displayed in Figure 60. Left-clicking the MousePointer option generates a down arrow and left-clicking this down arrow opens a pull-down menu, the last option on which allows us to select a custom icon. Once we have set the control to use a custom mouse pointer, we left-click the Mouselcon option in the Properties dialog box to open the Load Picture, Figure 61, dialog box. Navigating to the desired folder using the Up One Level button or the Look In pull-down menu, we can select the folder that has the desired picture. If we navigate to the Windows cursors file then we can select one of the Windows cursors or we can design our own; we can use any picture file as a mouse icon.

I hope that you found this handout helpful. I have tried to provide enough information to help you to get started without your feeling overwhelmed. The best way in which to learn how to create interactive workbooks, 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.