PHYC 500: Introduction to LabView

M.P. Hasselbeck, University of New Mexico

Exercise 1 (v 1.3)

Setup
The user interface of LabView is highly customizable. How this is done is a personal preference. For the seminar, it is suggested to implement the following:

From the main menu, select Tools: Options: Block Diagram. Uncheck "Place front panel terminals as icons" and click OK. This saves space on block diagram.

Open a blank VI. Go to the Front Panel, right-click on the open space, scroll down to "Change Visible Palettes", and check Modern, Classic, and Silver. Click OK. These Front Panel palettes give the user many options to develop the desired appearance of the user interface. Go to the Block Diagram of this VI. Right-click in the blank space, scroll down, click "Change Visible Palettes". Select Programming, Measurement I/O, Mathematics, Signal Processing, Control Design & Simulation, User Libraries, and Select a VI... If this option isn't visible, pin the palette open (upper left hand corner) and select Customize.

Finding roots of 2^nd order polynomial

The 2^nd order equation

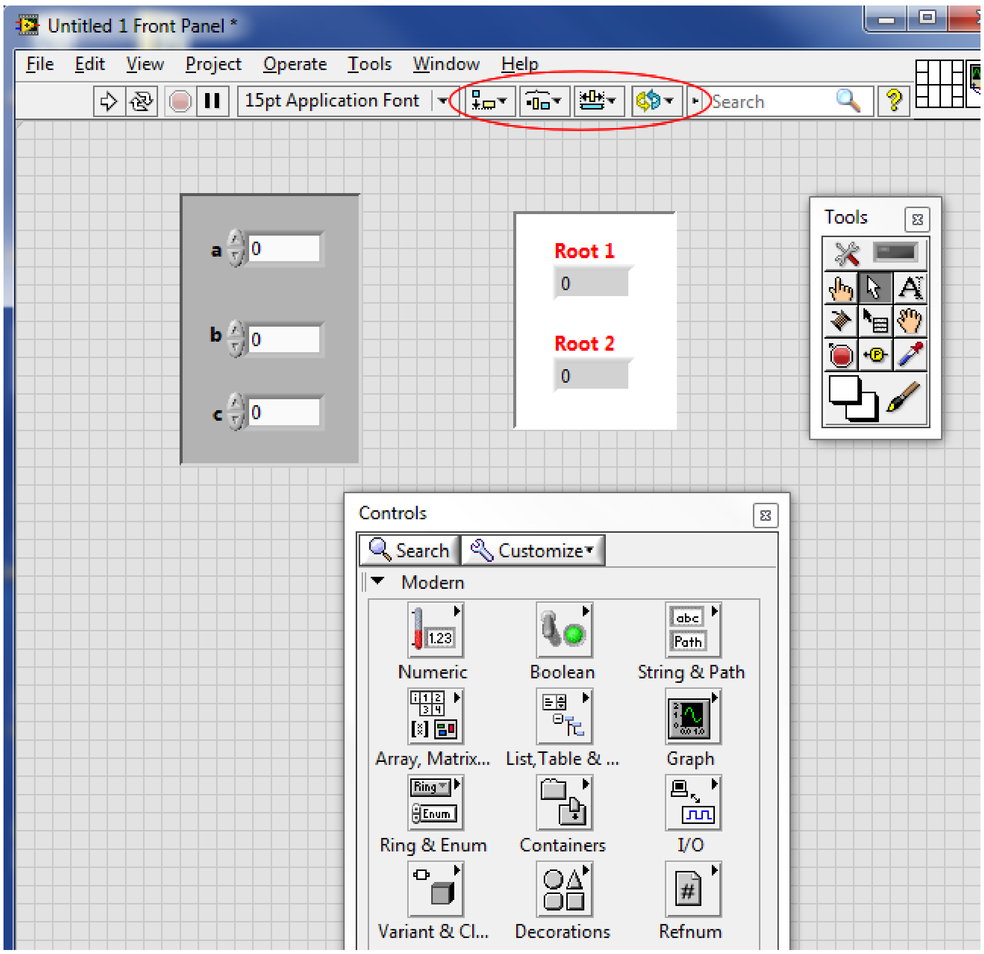
$$ax^2 + bx + c = 0$$

has two roots given by the quadratic formula

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

In this exercise, a VI will be written to find these roots. Go to the Front Panel of your blank VI. Select View: Tools Palette. The top button in the palette enables/disables automatic tool selection. For a new user, it's probably a good idea to disable this function. Select the arrow tool. Right click anywhere on the blank Front Panel to open the Controls palette. Pin this window open at its upper left corner. Select and place three numeric controls corresponding to the polynomial coefficients and label them a, b, and c. Next right-click and place two numerical indicators on the panel.

One does not have to accept the default appearance of Front Panel objects. In the example layout below, the labels **a, b,** and **c** have been placed to the left of the controls and their size and style (**bold** font) have been changed. The label can be moved and re-positioned independently of its control, but when the control is selected the label moves along with it.

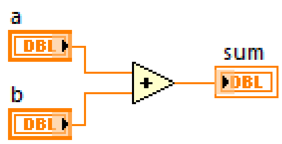


The two root indicators have been changed to **bold red**. These modifications are done with the Arrow and Edit Text tools. Keyboard shortcuts for increasing/decreasing font size are CTRL = and CTRL -, respectively.

The input coefficients and output values are grouped together in recessed boxes that can be found in Decorations. The buttons in the top menu bar (circled below) allow you to position and align these objects. The recessed boxes will have to be moved to the back of the controls and indicators to see them.

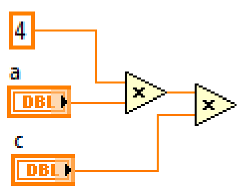
The box holding the two roots has been colored white using the paintbrush (Set Color) in the Tools Palette. Making these cosmetic modifications is not necessary for the VI to operate, but it can help a user better understand the organization and function of your

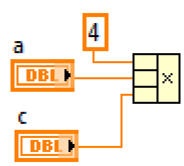
program.
Go to the Block Diagram and identify the five terminals (three inputs, two outputs). These icons are orange double-precision floating point, but notice the different color shading for inputs and outputs. Setup the quadratic formula by right clicking and opening the Numeric palette. Pin it open at the upper-left corner of the window. You will see icons representing the various mathematical functions including x^2, $\sqrt{x}$, and $(-x)$. Use the Connect Wire tool to implement the formula above; this is a left-click and drag operation to connect the various ports on the terminal. As an example, adding $a + b = sum$ would look as follows:



When working with LabView, it's a good idea to have the Context Help enabled. This is found in the Help menu, clicking the question mark button on the menu bar, or by typing CTRL-H. When you scroll over an icon or object, a Help window will provide useful specific information. LabView also uses "tip strips", which can display information when the mouse is hovered over a control or indicator of a running VI.

The multiplication operation *4ac* is implemented with two multiplication steps as shown in the diagram below (left).





To create the constant *4*, right-click on an unwired input terminal, then Create: Constant. The default value is *0*; simply change it to *4*. It is cleaner, however, to use the compound arithmetic function (right). Select it from the Numeric palette and place it on the block diagram. Drag down the icon to produce a third input terminal. The default operation is compound addition, so right-click on the icon, Change Mode: Multiply.

Construct a VI to find the two roots of the quadratic formula. You should first write code to find one root, then add the required operations to find the second. The latter task will be easier because most of the needed calculations will be already in place. You can use copy-paste to duplicate code, but it's often easier to select it (arrow tool), then drag the highlighted code while holding down the CTRL key.

LabView is constantly monitoring for errors while you are building a VI. If one or more problems exist, the run button (arrow) on the upper-left menu bar will appear broken. If you click on the broken arrow, a window will open listing all the existing errors. When the execution arrow is solid, the VI will run. Choose three coefficients (*a,b,c*) and generate the two roots. If both roots are shown as NaN, the VI has attempted to take the square-root of a negative number. Choose coefficients so both roots are real.

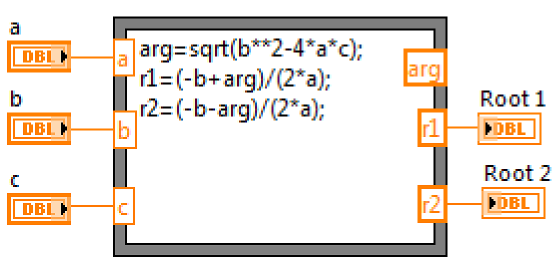
Go to the Block Diagram and click Highlight Execution (light bulb icon). Run the VI again. This slows down the operation so that you can see the flow of information.

Another useful tool is the Block Diagram Cleanup that is found on the menu bar:

When the wiring and icon arrangement becomes cluttered and unreadable, LabView can automatically tidy the layout. Use the select tool to highlight the desired region to cleanup, then press the button shown. If you are not happy with the results, the operation can be undone with CTRL-Z. Save this VI as it will be used again in this exercise.

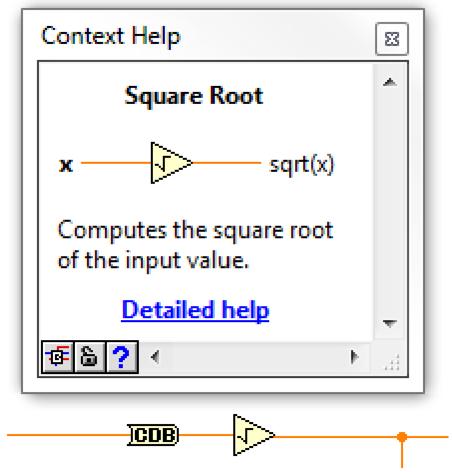
Formula node

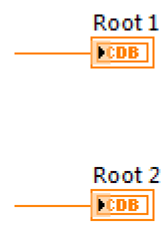
Users familiar with C programming may find the Formula Node useful. Instead of using graphical icons, calculations can be performed with text-based code. Go to the Block Diagram, right-click and select Structures: Formula Node. A cursor will appear; use this to draw a programming box. Right-click on the left edge of the box and create three inputs; label these a, b, and c, then connect the corresponding three numeric controls. Add two outputs on the right edge of the box and wire these to the two root indicators. How you implement the code is personal preference, but each calculated value must have a corresponding output. In the example below, the result of the square-root operation is an intermediate result with an unwired output terminal labeled arg. The Formula node cannot handle complex numbers.



Complex roots

The programs constructed so far cannot handle complex roots. Open the saved VI. The Highlight Execution mode in the Block Diagram will allow you to identify the source of the difficulty; in this case it is the square-root function. By clicking on "Detailed help" in the Context Help window, you will learn that the square-root function will output NaN (not a number) for negative input *unless* it is wired to accept a complex number. This is implemented by converting the floating point input to complex double-precision. Right-click on the input wire, Insert: Numeric Palette: Conversion: To Double Precision Complex. This will produce a modified diagram shown below (left).





To display the output as a complex number, the numerical indicators (Root 1 and Root 2) must also be changed. Right-click on each icon, Representation: Complex Double (CDB) as shown above, right. Now run the VI with coefficients that produce complex roots. You will probably have to re-size the Front Panel indicators to see the real and imaginary components.

LabView defaults to six-digit display precision. Change this to three digits by right-clicking on each of the two root indicators and selecting Display Format (also found in Properties). Save a copy of your VI.

Create a SubVI

Use the arrow tool and drag a box around the entire block diagram. Select Edit: Create SubVI. Double click on the new icon to open it. You will see a Front Panel that matches the VI you just created. In the upper right corner of the window, there will be two icons showing the terminals (left) and display artwork (right). Right-click on the terminal icon, select Patterns, and notice the variety of designs available. Unused terminals can be left unwired, so there is no need to change the pattern. The icon design is also customizable and you can optionally edit this. Save the SubVI using a name of your choice.

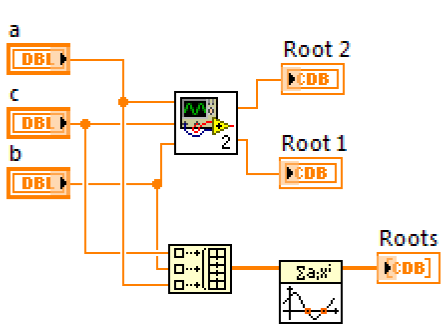
Go back to the original VI. You should see a greyed-out icon of the SubVI. Right-click on the icon and click “Relink to SubVI”. Verify that the VI works with the SubVI in place. Close the original VI but do not save it.

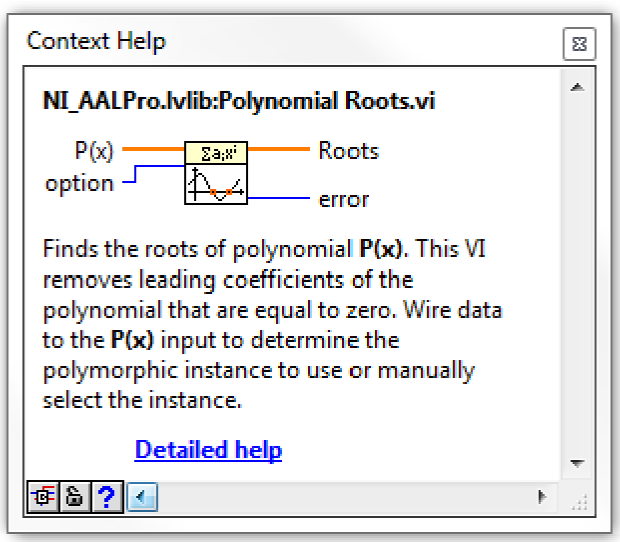
Open up a new blank VI. In the Block Diagram, right-click and highlight “Select a VI”. A dialog will open; navigate to the SubVI you just made, select it, and place it in the diagram. Select the Connect Wire tool. Right click on each of the three input terminals (left side of icon) and Create: Control. Use the same tool and right-click on the two output terminals (right side) and Create: Indicator. You should have three inputs (a,b,c) and two outputs (Roots 1 and 2). This VI will find the complex roots of a specified quadratic equation using the code contained entirely in the SubVI.

Built-in root finder

LabView has many powerful built-in functions that can eliminate the need to write code from scratch. One of these functions is in the Mathematics palette and will find the roots of an nth-order polynomial. Navigate to Mathematics: Polynomial: Polynomial Roots and place it in the block diagram. Alternatively, use LabView's Quick Drop function (simultaneously hold down CTRL and the spacebar) to open up a search window and type in the keywords.

The Help window shows that this VI requires the coefficients in the form of an array. In the block diagram, select Array: Build Array. Use the arrow tool to drag the array icon so that it displays three terminals. Wire coefficient *c* to the top, *b* to the middle, and *a* to the bottom terminal. Wire the output of the Build Array function to the input terminal *P(x)* of the polynomial VI. Note that the orange wire thickness has increased, which indicates the presence of an array. Right-click on the output terminal of this function (Roots) and create an indicator. The Block Diagram should look similar to the following:





On the Front Panel, you may have to re-size this indicator to show the two elements of the output array. Run the VI and confirm that the two calculations agree.

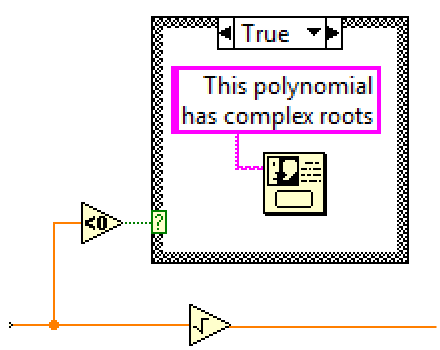
Comparison and Case Structure

Open the saved version of the quadratic root finder VI (Alternatively, open the SubVI you created and save it under a new name). The VI will be modified to warn the user before it attempts to take the square-root of a negative number, perhaps because complex roots are unphysical in the application.

Go to the Block Diagram. Right-click and select "Comparison: Less than 0?" and place it on the diagram. Wire its input to the input of the square-root function. The output of the comparison operation is a Boolean value, i.e. TRUE or FALSE. This output will be used to address a Case Structure.

Right-click on the Block Diagram, Structures: Case Structure. This selects a square, dotted cursor. Use the left mouse button and draw a Case Structure box. Next, use the Operate Value tool (depicted as a vertically pointing finger) to scroll between the True and False cases. Select the True case. Wire the output of the comparison function to the Case Structure input terminal, which is shown as a green question mark [?]. You will see that the connection wire is a green dotted line, indicating the presence of Boolean data. If the value is True, that means there is a negative number being sent to the square-root function and the VI should alert the user. This can be done in a variety of ways; a warning dialog will be used here.

Right-click on the Block Diagram and select Dialog & User Interface: One Button Dialog. Place this icon inside the Case Structure. The negative input situation must be conveyed to the user with a message; use the Connect Wire tool, right-click on the terminal labeled message, and create a constant. An empty pink box should appear. This corresponds to string data, which holds text. Use the Edit Text tool to type in an appropriate alert message. An example of how this might look is shown below.



If the data passed to the square-root operation is not negative, the comparison operation will return a Boolean False output. In this case, no warning is needed so the False case of the Case Structure is left completely blank. If you run the VI with Highlight Execution enabled, you can observe the Boolean logic operation.

Save your working VI for the next exercise.