Prof. Nuno Alves (nalves@wne.edu), College of Engineering

Lecture Notes #9 - 14/February/2012

Arrays (pages 288-313)

An array is a collection of data. We build an array by arranging data into rows and columns. To access data in an array we need to know the unique row-column 'address' of a particular element in the array. Examples of 1D arrays are row or column lists. Examples of 2D arrays are things like spreadsheets or the screen on your PC. Each pixel has a unique memory address telling it how to 'light up'. Your eyes use arrays of light-sensitive cells to sense the outside world. We can create arrays of higher dimensions but can only visualize arrays up to three dimensions.

We describe array elements as subscripted variables. For 1D arrays we a single subscript: A_n. For 2D arrays we use a double subscript: $A_{n,m}$, where the subscript pair represents (row, column) coordinates. For a 3D array we use a triple subscript: $A_{m,n,p}$, where the subscript triplet represents(row, column, plane) coordinates.

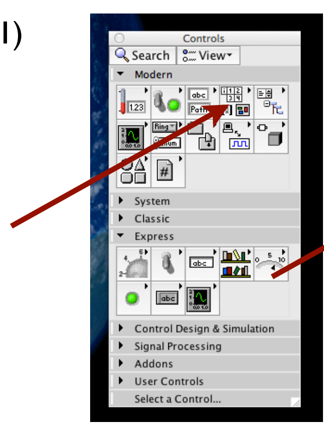
In LabVIEW we can build arrays in either the Front Panel or the Block Diagram. We also have a number of pallet objects that allow us to manipulate whole or partial arrays and to perform array (or matrix) arithmetic.

Very important: All elements in an array have the same data type. This is not the same in clusters.

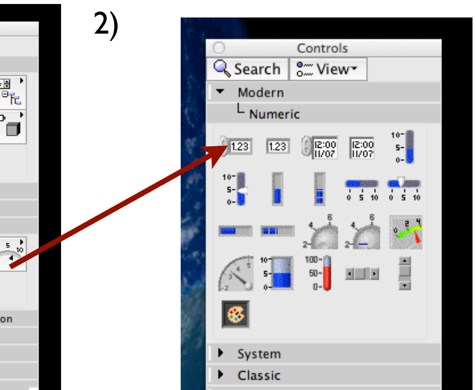
The three steps to build arrays

1) Placing an Array Shell.
2) Dropping a control or indicator object into the shell.
3) Expanding the composite object into the appropriate dimensions desired through click-and-drag operations. Note that expansion is a function of the dimensionality chosen.

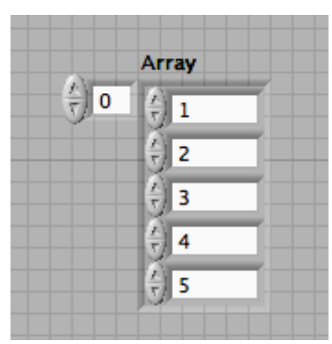
1)

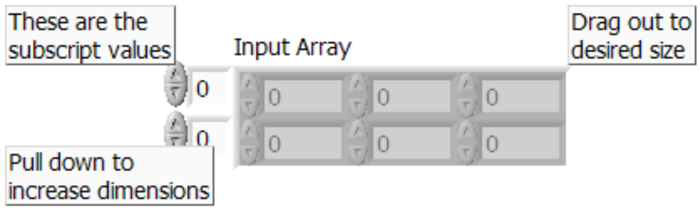


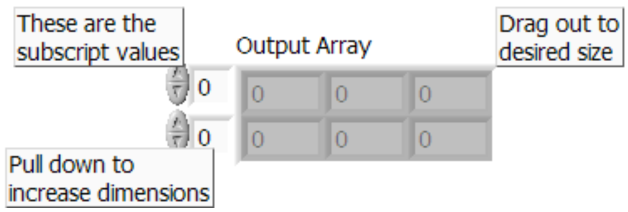
2)



3)

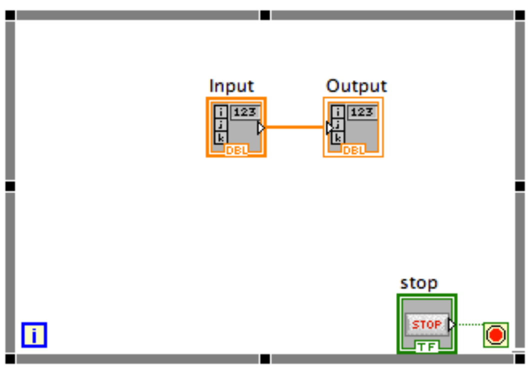


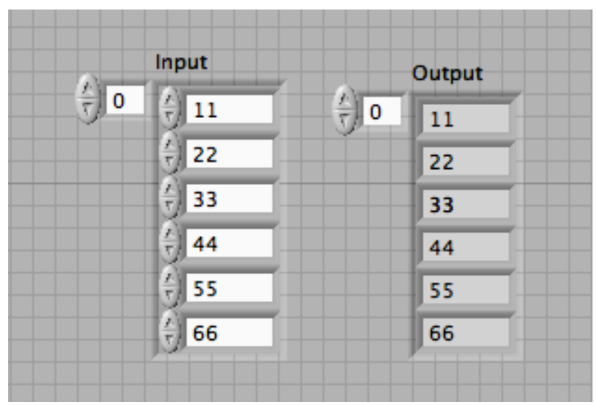




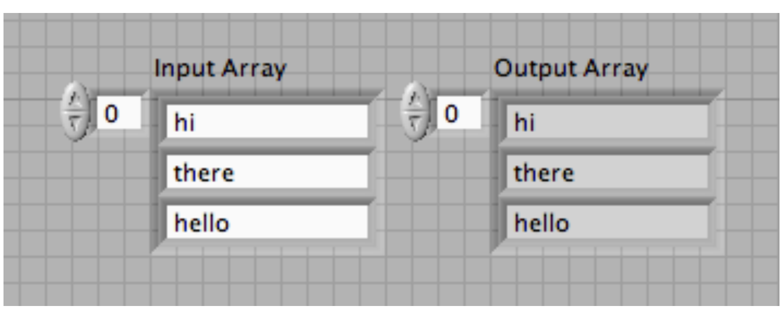
In class exercise: numerical control

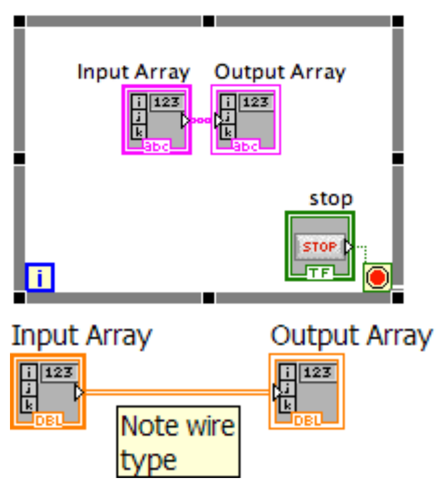
Put different values into the Input array, click the run button and see the data transferred to the Output array. Shrink the data viewing windows to one and show how the index variables work the data in the single viewing window.





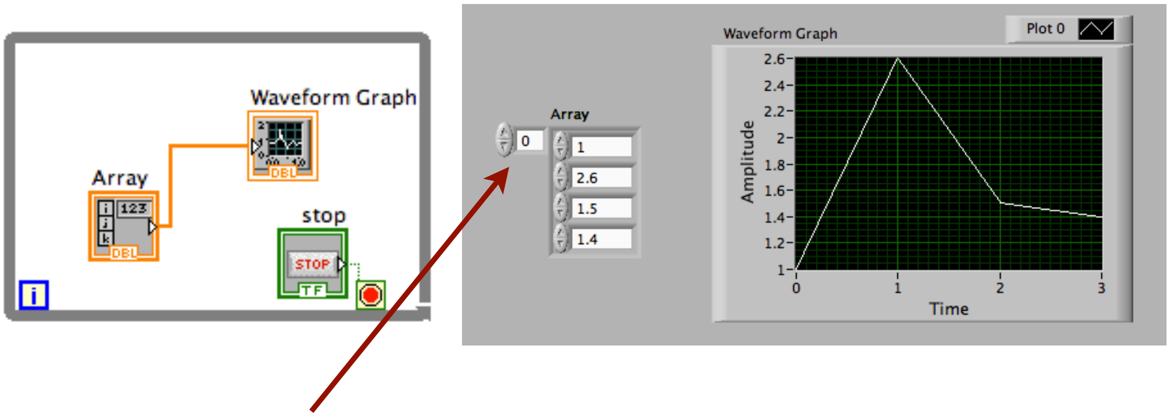
In class exercise: string control



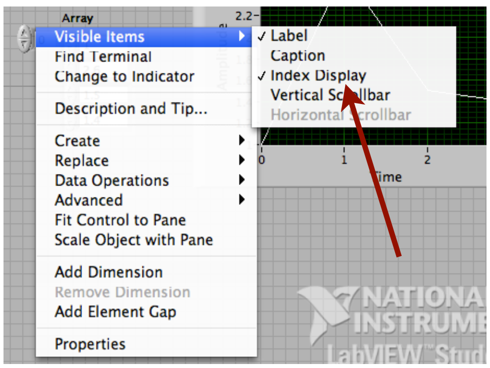


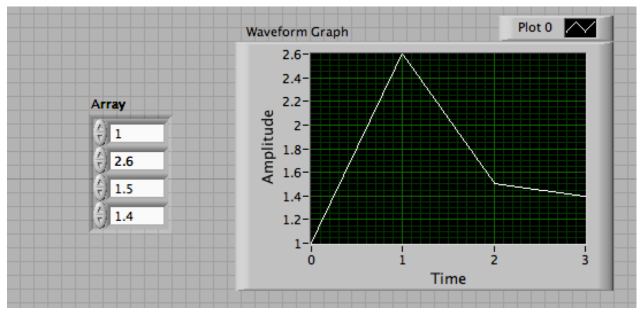
Note that all windows get the same size in the group.

In class exercise: display array on graph



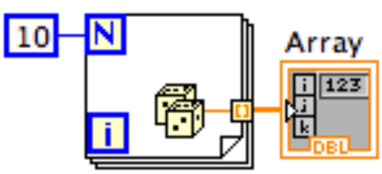
We can remove the index indicator in the array... by right clicking on the array.

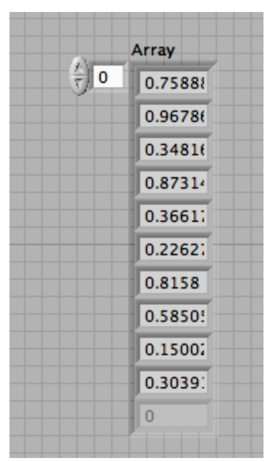




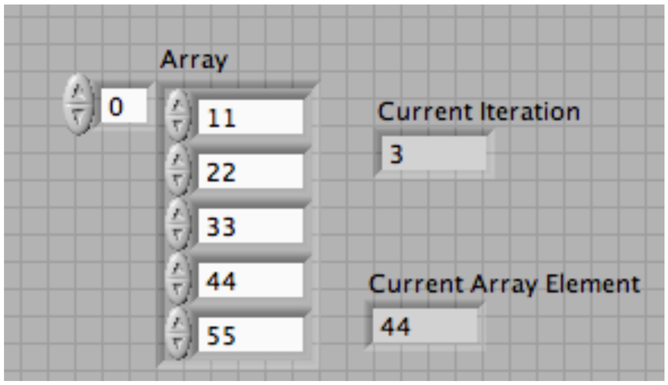
Create arrays with loops: pages 293-297

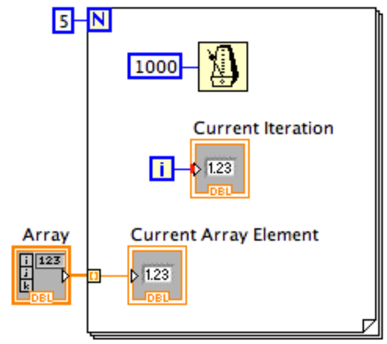
We can generate arrays by generating a random number inside a loop and enabling the auto-indexing.





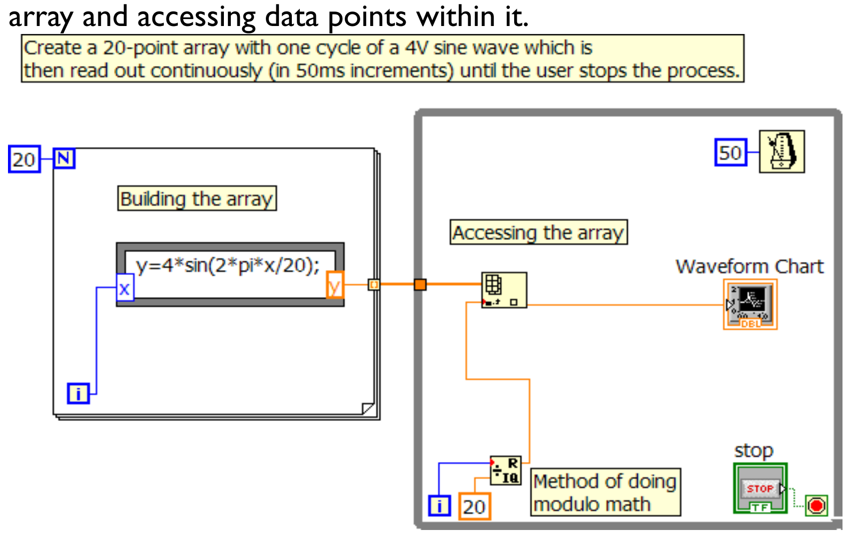
We can also pass arrays into a loop one element at a time or the entire array at once.



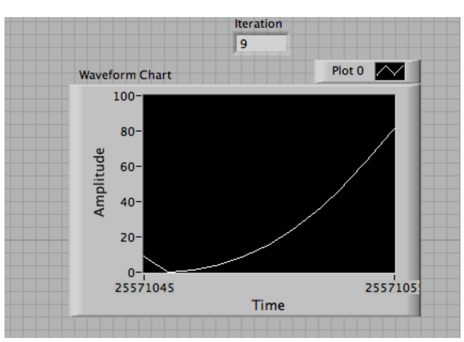


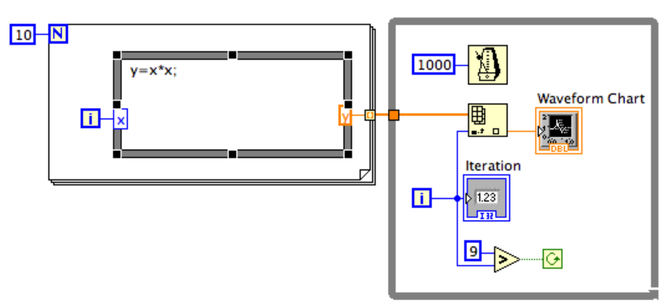
In class exercise: creating a data array with a formula node

Creating a data array and accessing data points within it.



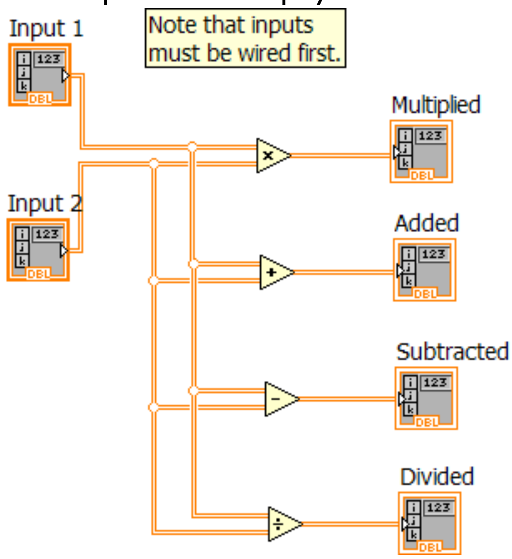
Re-create the following virtual instrument that will create an array of 10 elements in which each element is the square of its index. Display them in a waveform chart.





In class exercise: multiply, subtract, add and divide data arrays

Design a VI which accepts two user arrays and performs addition, subtraction, multiplication, and division on them. The input and output data is displayed on the front panel.



What are the effects of changing array sizes?

Homework due next class:

• Read pages 313-331 in text.

• For each of the following questions submit your virtual instrument (VI) files to manhattan. Homework submitted after 8am on Feb. 16th will not be graded.

• **Question #1** (Not on book) - Create a VI that produces and displays a continuous sine wave with a peak amplitude of 3 (until the user stops the process) to which a random number between +1 and -1 is added to each sine data point as created (this is adding noise to a signal). The curve should have 20 data points per period. The process should produce one noisy data point every 50 milliseconds. Hint: Remember the Formula Node? Additionally, the VI should also display a 'filtered' version of the noisy sine wave which is found by averaging three or four past values of the wave as each new wave data point is produced. Display the noisy wave and the filtered wave on separate charts (or a dual-trace plot). What are the apparent differences between a filtered wave that uses three-point averaging and a filtered wave using four-point averaging?

• **Question #2** (P6.2 on page 342) - Create a VI that calculates the dot product of two *n*-dimensional vectors. Double check your math by comparing your calculation using arrays and math functions with the results of the **Dot Product.vi** that can be found in the Functions >> Mathematics >> Linear algebra palette.

• **Question #3** (P6.3 on page 342) - Create a VI that calculates the cross product of two 3-dimensional vectors.

• **Question #4** (P6.6 on page 343) - Build a VI that generates and plots 500 random numbers on a waveform graph indicator. Compute the average of the random numbers and display the result on the front panel. Use the Mathematics >> Probability and Statistics palette to compute the average of the random numbers.