

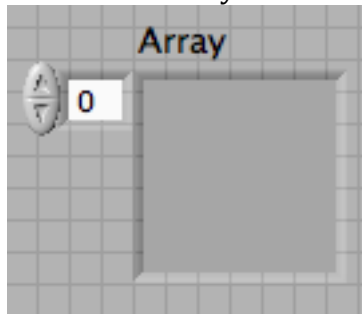
### Lab 3 FYSS 385

The goal of this lab is to study how arrays and clusters are presented in LabVIEW and how the data can be represented as charts, plots and using these. As the topic involves skills that are very much hands-on; it is presented as a lab exercise.

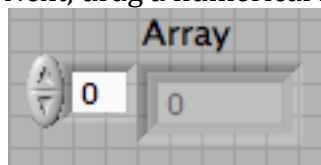
#### 1. Creating array indicators and constants

Arrays are  $n$ -dimensional data structures where the data is all of the same type. (strings, numbers, Boolean variables, etc but not a mixture of these.) Waveforms and graphs contain a large number of coordinate points and hence it is useful to use arrays to help perform the repetitive calculations involved in handling these.

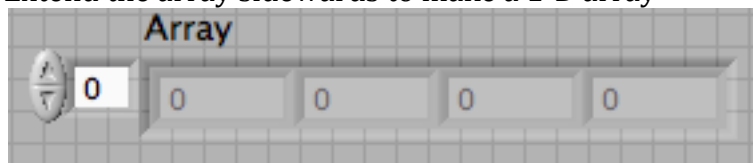
-First make an array constant. Open a VI, go to the control panel and choose add array from the controls panel. Note the array is empty.



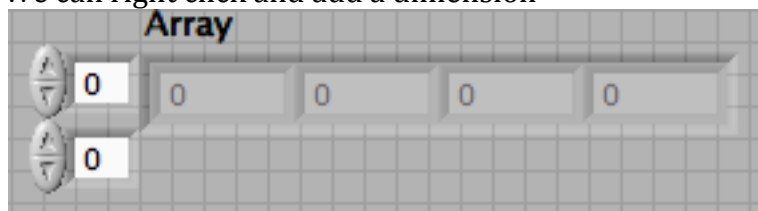
- Next, drag a numerical indicator into the array.



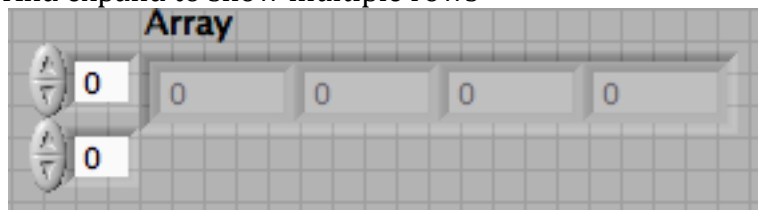
- Extend the array sideways to make a 1-D array



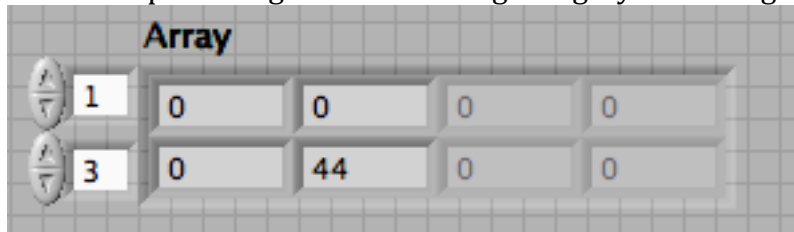
- We can right click and add a dimension



- And expand to show multiple rows

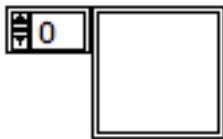


- Values can be set using the pointer tool and in. (eg. 44 in 1,1 in this case- note preceding elements are lighter grey indicating they are set.

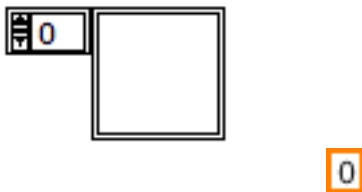


- You can also initialise using an array constant in the Block Diagram.

Place a Array constant (Array and cluster sub-palette) on the block diagram



Drag a numeric constant into the array constant



Then it changes form:



- Then expand and add a array indicator:

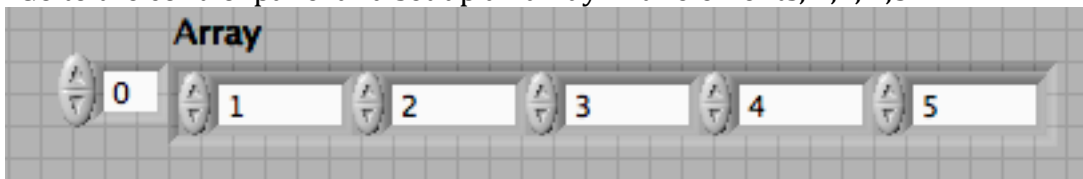


- Initialise the elements to some different values.

2. Make a VI using arrays that you enter a 2x2 array that is converted to a matrix and the inverse matrices and is calculated and displayed. Make a check and multiply with the original matrix and verify you get the unity matrix.

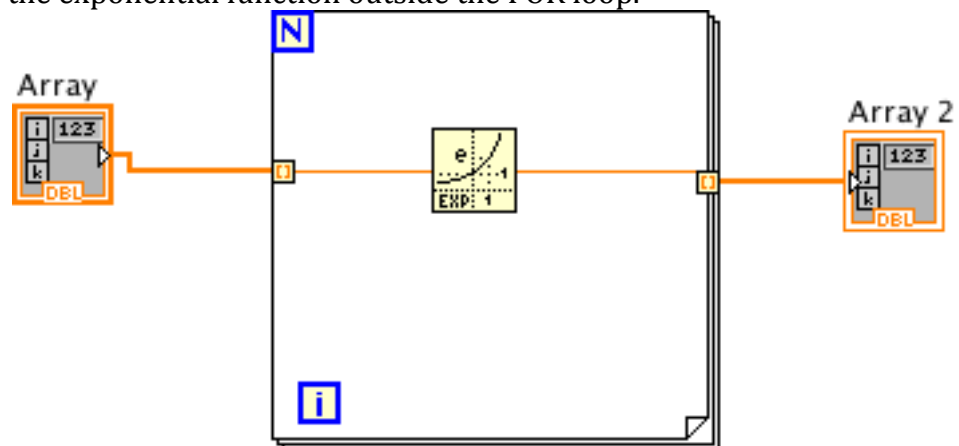
3. A powerful feature of LabVIEW is auto-indexed arrays. This means a FOR or WHILE structure can be used to perform operations on arrays.

- Go to the control panel and set up an array with elements, 1,2,...,5

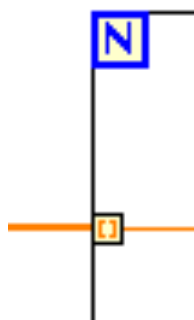


- Next go to the Block Diagram and set up a FOR loop containing a exponential function.

- Wire the array to the exponential function and create an indicator for the exponential function outside the FOR loop.

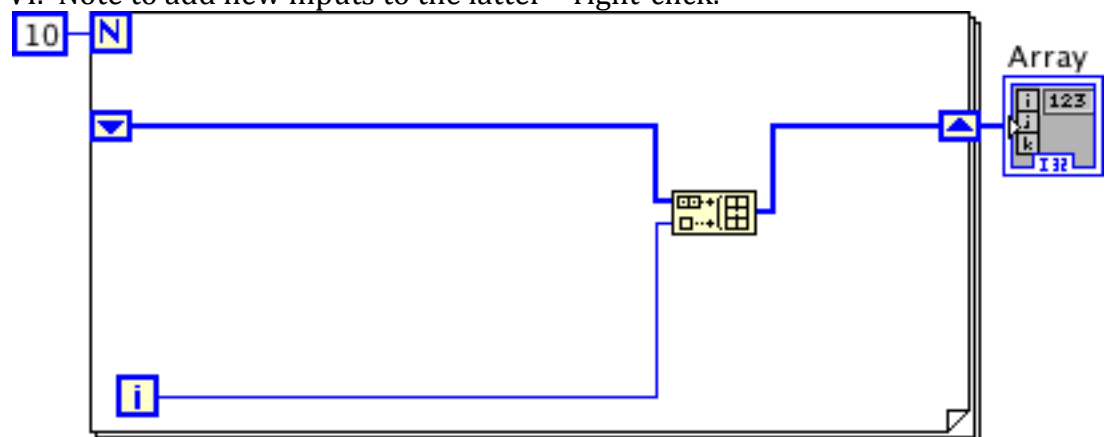


- Notice the shape of the terminal on the FOR loop and that the line changes from 1D array to scalar width when we go from outside to inside the loop structure.

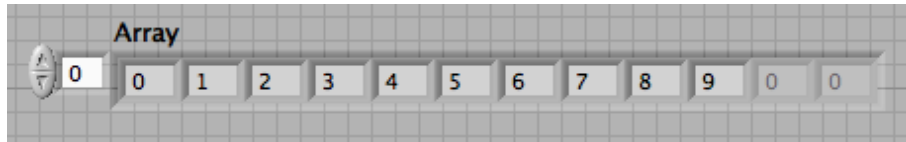


This denotes the array is auto-indexing and automatically converts from a array to a scalar and visa versa. This can be switched off by right-clicking the terminal on the FOR or While structure

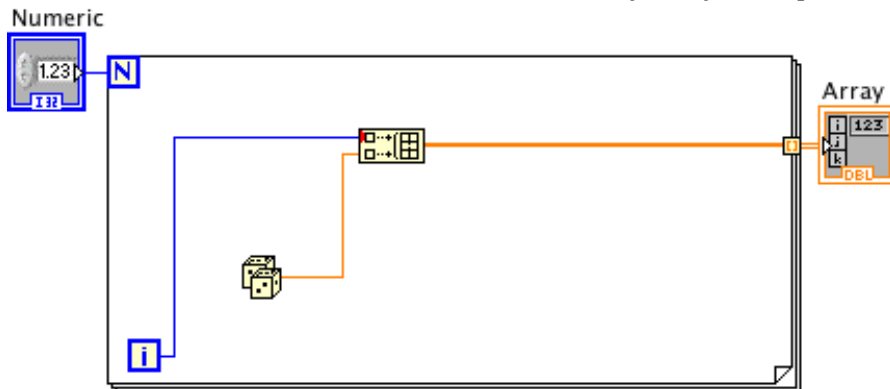
- Another way to build arrays is to use a shift register and Build array VI. Note to add new inputs to the latter – right-click.



-Run this VI



- Lets use the two methods to make an  $2 \times n$  array of  $x,y$  data points.



- Use this to make an array of 1000  $x,y$  data points where  $y$  is a random number.

-Save the VI, it will be used later.

#### 4. Clusters

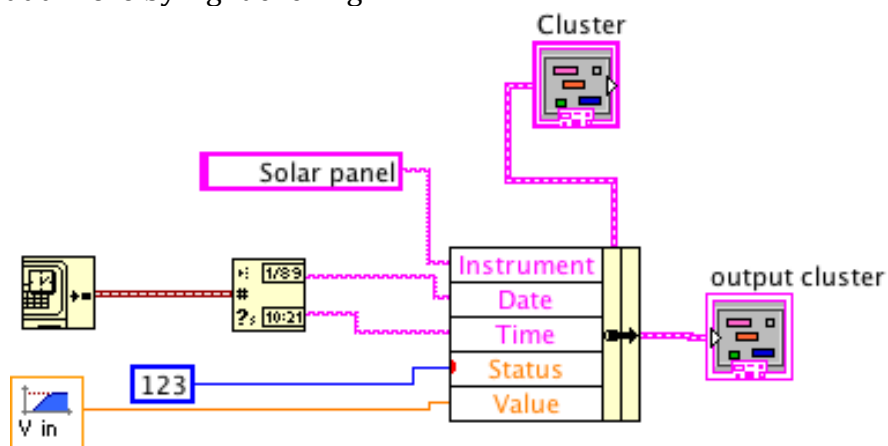
In this section we explore clusters, which are similar to arrays, but with the important difference that the elements can be of different data types. They are useful for arranging data in standard.

- Consider a VI that logs instrument name, date, time, instrument status code and data value. Make first a cluster control using the same technique as for the array. This should contain the following variables.

Signal	Type
Instrument	String
Date	String
Time	String
Status	Number
Value	Number



- We use the “Bundle by name function”. It has by default one input we add more by right clicking.



- Notice in the control that some default values have been set. Set the run mode to debug and run the VI. Notice what happens.
- Using the unbundle by name a particular parameter can be extracted, or using bundle by name a particular variable can be set.
- Notice the time stamp VI gets the time in seconds from 12:00 a.m., Friday, January 1, 1904, in the current time zone. (This was last century’s earliest day in a year that was not a leap year.) This is converted to two strings one with the date and the other the time.

- Try to put a probe from the Tools palette on the cluster and observe its structure. This is useful to understand unknown structures
- 

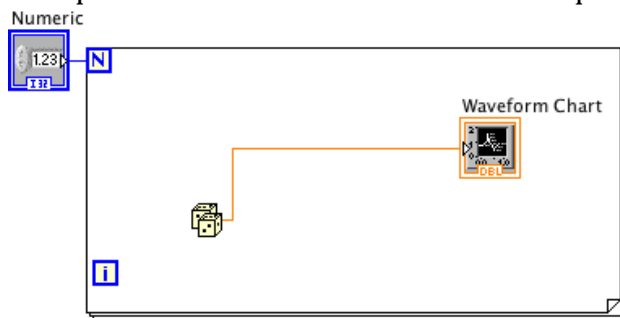
#### 4. Waveform graphs and charts

Data in the form of long tables of numbers are not very to understand for humans. A common way to overcome this problem is to present the data visually as a 2- or 3- or even 4-dimensional plot. LabVIEW provides powerful data rendering capabilities.

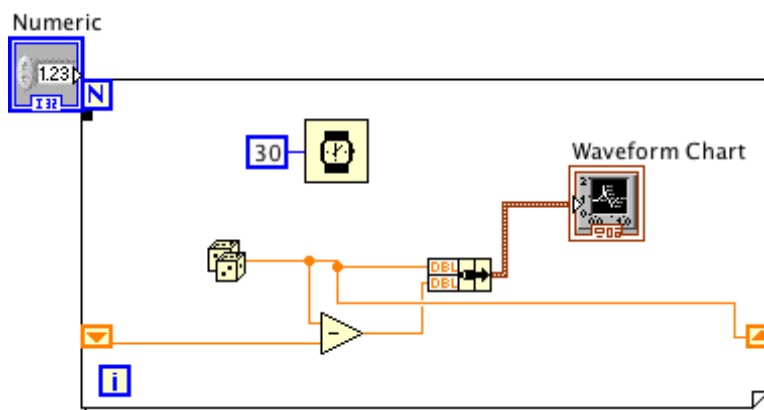
Data can be represented using indicators depending on if it is equally spaced in, or unequally spaced

	<b>Absissa spacing</b>	<b>Plot type</b>
Time	Equally spaced (eg. Time sampled)	Waveform chart
Time	Unequally spaced	X-Y plot

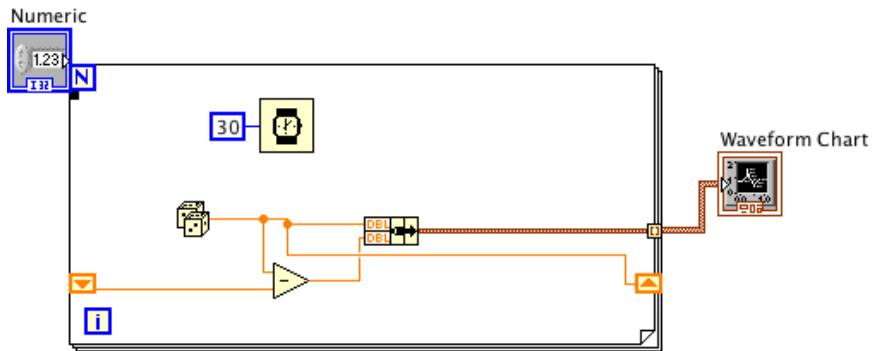
The simplest is the chart where the data is equally spaced in the abscissa.



We can plot multiple charts by bundling the two ordinate values into a cluster.

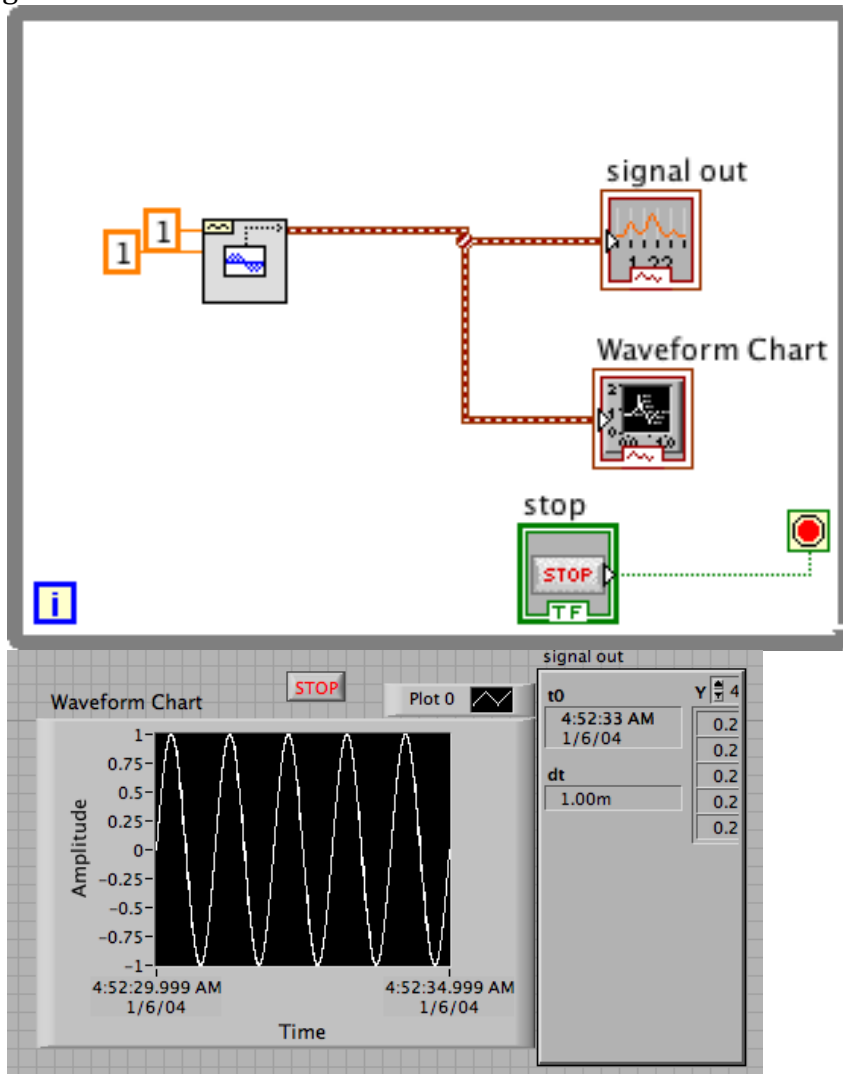


Notice that when the VI is run the data is plotted dynamically. We can also move the waveform chart out of the loop.



In this case the data is first plotted when the loop has completed.

-There is a special cluster called the waveform cluster or waveform data type that is accepted as input by waveform chart. In this case we use the sine waveform generator in the Signal Processing function palette to generate a sine wave in a waveform data cluster.



- What does the waveform data type cluster contain?

## 5. X-Y plotting

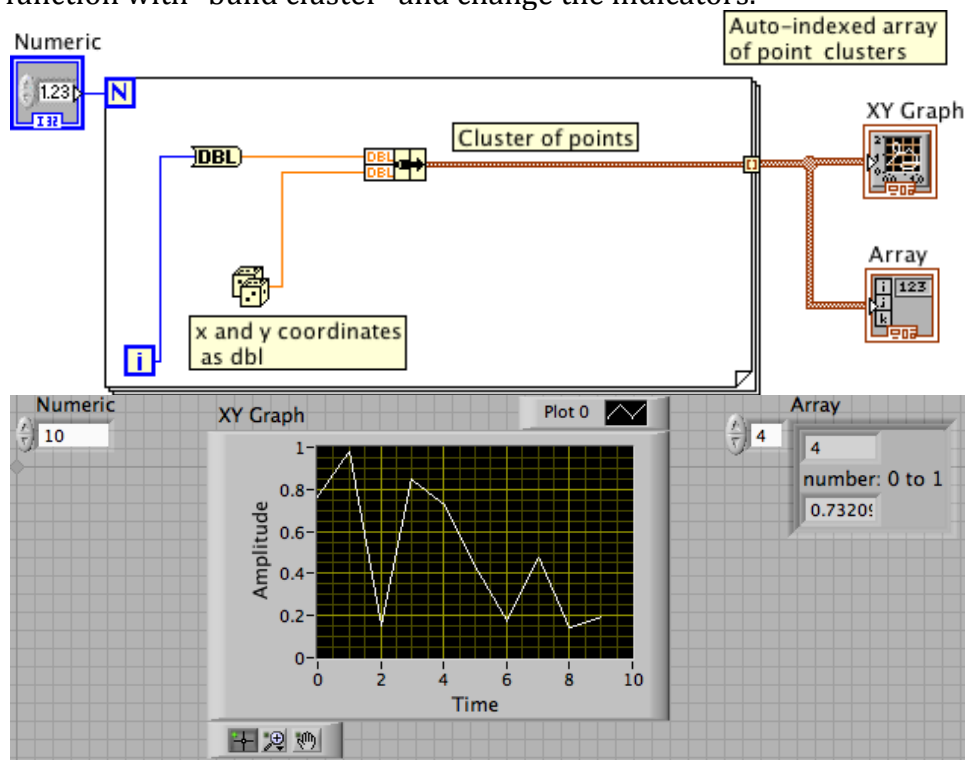
A very common situation is to make a X-Y plot of data collected from a measurement. The X-Y plot VI accepts data in several forms.

**Single plot:** (i) cluster containing an x-array and a y-array, (ii) array of points where each point is a cluster of x and y-values.

**Multiple plots:** (iii) Arrays of clusters of (arrays of clusters of x,y points). i.e. arrays of clusters containing arrays for single plots.

Unlike charts, X-Y plots can represent data where the spacing in the x-direction are not uniformly spaced.

-Open the random array example from above. Replace the “build array” function with “build cluster” and change the indicators.

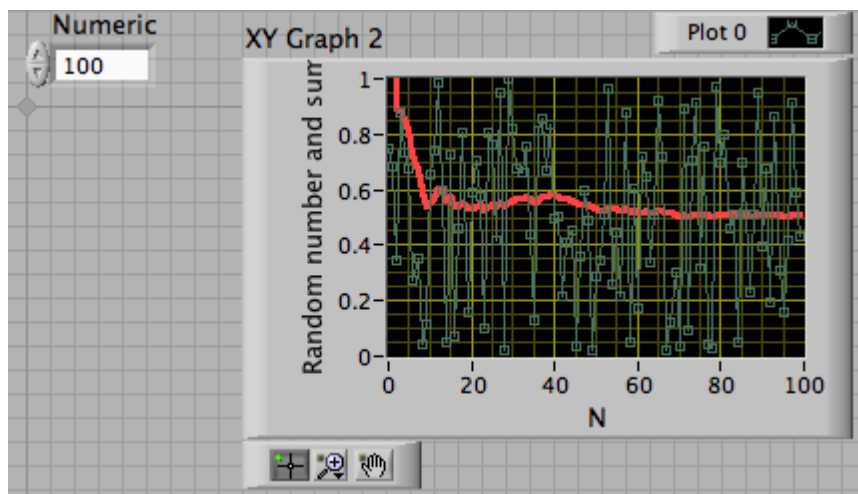
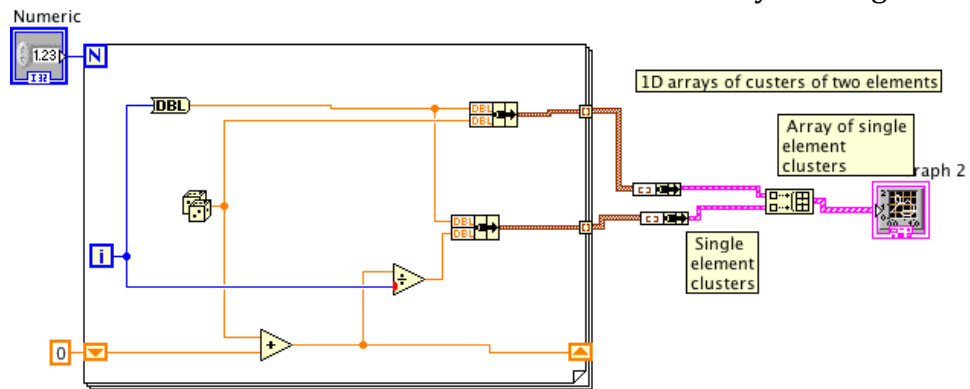


-Right click on the graph and make the “Graph palette” visible. Explore their function. You can also change other plotting variables, such as the scales and the format.

- Often we want to plot several things on the same X-Y graph. One way to do that is to bundle each plot which is an array of points (where each point is a cluster of x,y coordinates). As an example calculate 100 random



numbers and the mean value of the random numbers by running average.



If you have time investigate using the help functions how you can change the plot characteristics from within LabVIEW instead of right clicking.

### Report

No detailed step by step report is needed. Instead you should address the following bigger questions:

1. What is the difference between an array and a matrix?
2. What is the advantage of bundling variables and arrays into a cluster?
3. When should indicators be placed in loops and when not?