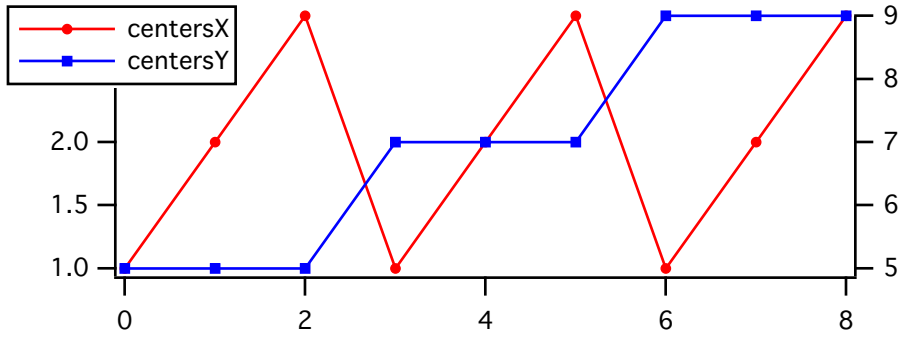

To display this as an image, we transform the data so that the Z wave becomes a 2D matrix representing pixel values and the X and Y waves describe the centers of the rows and columns of pixels:

```
Redimension/N=(3,3) zData
Make/O/N=3 xCenterLocs = centersX[p]       // 1, 2, 3
Make/O/N=3 yCenterLocs = centersY[p*3]     // 5, 7, 9
```

We now have data as described under **Plotting a 2D Z Wave With 1D X and Y Center Data** on page II-389.

Plotting 1D X, Y and Z Waves With Non-Gridded XY Data

In this case you have 1D X, Y and Z waves of equal length that define a set of points in XYZ space. The X and Y waves do not constitute a grid, so the method of the previous section will not work.

A 2D scatter plot is a good way to graphically represent such data:

```
Make/O/N=20 xWave=enoise(4),yWave=enoise(5),zWave=enoise(6)  // Random points
Display yWave vs xWave
ModifyGraph mode=3,marker=19
ModifyGraph zColor(yWave)={zWave,*,*,Rainbow,0}
```

Although the data does not represent a proper image, you may want to display it as an image instead of a scatter plot. You can use the **ImageFromXYZ** operation to create a matrix wave corresponding to your XYZ data. The matrix wave can then be plotted as a simple image plot.

You can also Voronoi interpolation to create a matrix wave from the XYZ data:

```
Concatenate/O {xWave,yWave,zWave}, tripletWave
ImageInterpolate/S={-5,0.1,5,-5,0.1,5} voronoi tripletWave
AppendImage M_InterpolatedImage
```

Note that the algorithm for Voronoi interpolation is computationally expensive so it may not be practical for very large waves. See also **Loess** on page V-515 and **ImageInterpolate** on page V-382 kriging as alternative approaches for generating a smooth surface from unordered scatter data.

Additional options for displaying this type of data as a 3D surface are described under "Scatter Plots" in the "Visualization.ihf" help file and in the video tutorial "Creating a Surface Plot from Scatter Data" at http://www.youtube.com/watch?v=kggo0B43n_c.

Image Orientation

By default, the AppendImage operation draws increasing Y values (matrix column indices) upward, and increasing X (matrix row indices) to the right. Most image formats expect Y to increase downward. As a result, if you create an image plot using

```
Display; AppendImage <image wave>
```