

Vector (Waveform) to Matrix Conversion

Occasionally you may need to convert between a vector form of data and a matrix form of the same data values. For example, you may have a vector of 16 data values stored in a waveform named `sixteenVals` that you want to treat as a matrix of 8 rows and 2 columns.

Though the Redimension operation normally doesn't move data from one dimension to another, in the special case of converting to or from a 1D wave Redimension will leave the data in place while changing the dimensionality of the wave. You can use the command:

```
Make/O/N=16 sixteenVals          // 1D
Redimension/N=(8,2) sixteenVals  // Now 2D, no data lost
```

to accomplish the conversion. When redimensioning from a 1D wave, columns are filled first, then layers, followed by chunks. Redimensioning from a multidimensional wave to a 1D wave doesn't lose data, either.

Matrix to Matrix Conversion

To convert a matrix from one matrix form to another, don't directly redimension it to the desired form. For instance, if you have a 6x6 matrix wave, and you would like it to be 3x12, you might try:

```
Make/O/N=(6,6) thirtySixVals      // 2D
Redimension/N=(3,12) thirtySixVals // This loses the last three rows
```

But Igor will first shrink the number of rows to 3, discarding the data for the last three rows, and then add 6 columns of zeroes.

The simplest way to work around this is to convert the matrix to a 1D vector, and then convert it to the new matrix form:

```
Make/O/N=(6,6) thirtySixVals      // 2D
Redimension/N=36 thirtySixVals     // 1D vector preserves the data
Redimension/N=(3,12) thirtySixVals // Data preserved
```

Multidimensional Decimation

You can create reduced-size representations of 2D and 3D waves.

The ImageInterpolate operation with the pixelate keyword can create a reduced-size representation of a 2D wave, or of the layers of a 3D wave, by averaging blocks of rows and columns. Use ImageInterpolate/PXSZ={*nx,ny*} pixelate where *nx* and *ny* are the number of rows and columns respectively to average.

The ImageInterpolate operation with the pixelate keyword can also create a reduced-size representation of a 3D wave by averaging 3D subsections. Use ImageInterpolate/PXSZ={*nx,ny,nz*} pixelate where *nx*, *ny*, and *nz* are the number of rows, columns, and layers respectively to average.

The ReduceMatrixSize function, provided by the "Reduce Matrix Size.ipf" WaveMetrics procedure file which was added in Igor Pro 9.00, creates a reduced-size representation of a 2D wave, or of the layers of a 3D wave, by sampling rows and columns or by averaging blocks of rows and columns. It uses a wave assignment for sampling and ImageInterpolate pixelate for averaging and provides a convenient wrapper for both techniques. To use ReduceMatrixSize, activate it as a global procedure file or #include it using:

```
#include <Reduce Matrix Size>
```

See **Decimation** on page III-135 for a discussion of decimating 1D waves.

Multidimensional Fourier Transform

Igor's FFT and IFFT routines are mixed-radix and multidimensional. Mixed-radix means you do not need a power of two number of data points (or dimension size).