

- Original data is untouched.
- Can display magnitude in decibels.
- Optional phase display in degrees or radians.
- Optional 1D phase unwrapping.
- Resolution enhancement.
- Supports non-power-of-two data with optional windowing.

Use `#include <FTMagPhase>` in your procedure file to access these functions.

### FTMagPhaseThreshold Functions

The FTMagPhaseThreshold functions are the same as the FTMagPhase procedures, but with an extra feature:

- Phase values for low-amplitude signals may be ignored.

Use `#include <FTMagPhaseThreshold>` in your procedure file to access these functions.

### DFTMagPhase Functions

The DFTMagPhase functions are similar to the FTMagPhase procedures, except that the slower Discrete Fourier Transform is used to perform the calculations:

- User-selectable frequency start and end.
- User-selectable number of frequency bands.

The procedures also include the DFTAtOneFrequency procedure, which computes the amplitude and phase at a single user-selectable frequency.

Use `#include <DFTMagPhase>` in your procedure file to access these functions.

### CmplxToMagPhase Functions

The CmplxToMagPhase functions convert a complex wave, presumably the result of an FFT, into separate magnitude and phase waves. It has many of the features of FTMagPhase, but doesn't do the FFT.

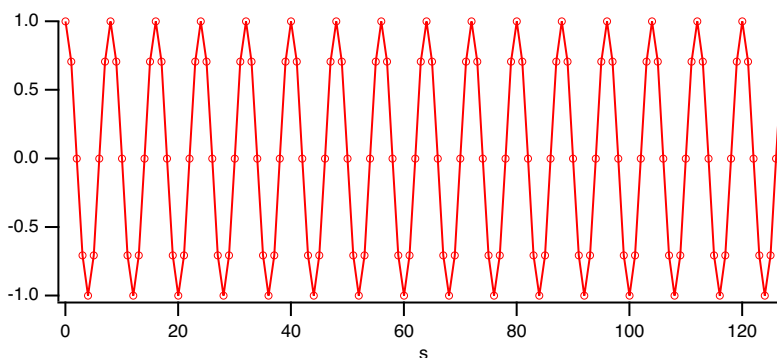
Use `#include <CmplxToMagPhase>` in your procedure file to access these functions.

## Spectral Windowing

The FFT computation makes an assumption that the input data repeats over and over. This is important if the initial value and final value of your data are not the same. A simple example of the consequences of this repeating data assumption follows.

Suppose that your data is a sampled cosine wave containing 16 complete cycles:

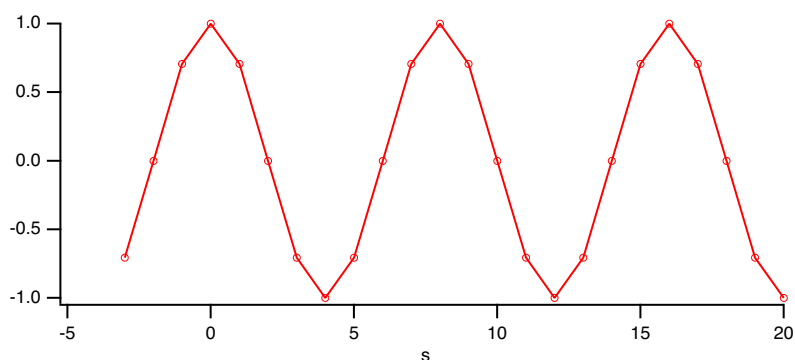
```
Make/O/N=128 cosWave=cos(2*pi*p*16/128)
SetScale/P x, 0, 1, "s", cosWave
Display cosWave
ModifyGraph mode=4,marker=8
```



## Chapter III-9 — Signal Processing

Notice that if you copied the last several points of `cosWave` to the front, they would match up perfectly with the first several points of `cosWave`. In fact, let's do that with the **Rotate** operation (see page V-810):

```
Rotate 3,cosWave          // wrap last three values to front of wave
SetAxis bottom,-5,20      // look more closely there
```

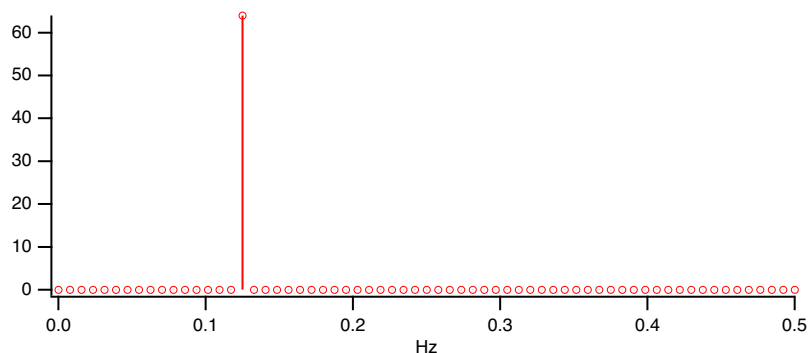


The rotated points appear at  $x=-3$ ,  $-2$ , and  $-1$ . This indicates that there is no discontinuity as far as the FFT is concerned.

Because of the absence of discontinuity, the FFT magnitude result matches the ideal expectation:

Ideal FFT amplitude = cosine amplitude \* number of points/2 =  $1 * 128 / 2 = 64$

```
FFT /OUT=3 /DEST=cosWaveF cosWave
Display cosWaveF
ModifyGraph mode=8, marker=8
```



Notice that all other FFT magnitudes are zero. Now let us change the data so that there are 16.5 cosine cycles:

```
Make/O/N=128 cosWave = cos(2*pi*p*16.5/128)
SetScale/P x, 0, 1, "s", cosWave
SetAxis/A
```

