```
    yw = yWave(xw[p]) + pw[0]
End
```

Note that all the waves created in this function are double-precision. If you don't use the /D flag with the Make operation, Igor makes single-precision waves. Curve fitting computations can be very sensitive to floating-point roundoff errors. We have solved many user problems by simply making intermediate waves double precision.

None of the computations involve the wave yw. That was done so that the computations could be done at finer resolution than the resolution of the fitted data. By making a separate wave, it is not necessary to modify yw.

The actual return values are picked out of yWave and stored in yw using the special X-scale-based indexing available for one-dimensional waves in a wave assignment. This allows any arbitrary X values in the xw input wave, so long as the intermediate computations are done over the full X range included in xw.

Multivariate All-At-Once Fitting Functions

A multivariate all-at-once fitting function accepts more than one independent variable. To make one, simply add additional waves after the xw wave:

```
Function MultivariateAllAtOnce(pw, yw, xw1, xw2) : FitFunc
    WAVE pw, yw, xw1, xw2

    yw = <expression involving pw and all the xw waves>
End
```

This example accepts two independent variables, xw1 and xw2. If you have more, add additional X wave parameters.

All the X waves are 1D waves with the same number of points as the Y wave, yw, even if your input data is in the form of a matrix wave. If you use **FuncFitMD** to fit an N row by M column matrix, Igor unrolls the matrix to form a 1D Y wave containing NxM points. This is passed to your function as the Y wave parameter. The X wave parameters are 1D waves containing NxM points with values that repeat for each column.

If you know that the input data fall into a fully-formed matrix wave, you may be able to temporarily re-fold the yw wave into a matrix using the Redimension operation:

```
Function MultivariateAllAtOnce(pw, yw, xw1, xw2) : FitFunc
    WAVE pw, yw, xw1, xw2

    Redimension/N=(rows, columns) yw
    MatrixOP yw = <matrix math expression>
    Redimension/N=(rows*columns) yw

    yw = <expression involving pw and all the xw waves>
End
```

Use this technique with great caution, though. In order to know the appropriate rows and columns for the Redimension command, you must make assumptions about the size of the original data set. If there are missing values (NaNs) in the original input matrix, these values will be missing from the Y wave, resulting in fewer points than you would expect.

If you use this technique, you will not be able to use the auto-destination feature (the /D flag with no destination wave) because the auto-destination wave is pretty much guaranteed to be a different size than you expect.

Similarly to the second example all-at-once function above, you can create an intermediate matrix wave and then use wave assignment to get values from the matrix to assign to the yw wave. It may be tricky to figure out how to extract the correct values from your intermediate matrix.