```
Function userFunction2(inY)
   Variable inY
   NVAR globalY=globalY
   globalY=inY
   NVAR globalXmin=globalXmin
   NVAR globalXmax=globalXmax
   return Integrate1D(userFunction1,globalXmin,globalXmax,1)
End
```

Finding Function Roots

The **FindRoots** operation finds roots or zeros of a nonlinear function, a system of nonlinear functions, or of a polynomial with real coefficients.

Here we discuss how the operation works, and give some examples. The discussion falls naturally into three sections:

- Polynomial roots
- Roots of 1D nonlinear functions
- Roots of systems of multidimensional nonlinear functions

Igor's FindRoots operation finds function zeroes. Naturally, you can find other solutions as well. If you have a function f(x) and you want to find the X values that result in f(x) = 1, you would find roots of the function g(x) = f(x)-1. The FindRoots operation provides the /Z flag to make this more convenient.

A related problem is to find places in a curve defined by data points where the data pass through zero or another value. In this case, you don't have an analytical expression of the function. For this, use either the **FindLevel** operation (see page V-242) or the **FindLevels** operation (see page V-244); applications of these operations are discussed under **Level Detection** on page III-287.

Roots of Polynomials with Real Coefficients

The FindRoots operation can find all the complex roots of a polynomial with real coefficients. As an example, we will find the roots of

$x^4 - 3.75x^2 - 1.25x + 1.5$

We just happen to know that this polynomial can be factored as (x+1)(x-2)(x+1.5)(x-0.5) so we already know what the roots are. But let's use Igor to do the work.

First, we need to make a wave with the polynomial coefficients. The wave must have N+1 points, where N is the degree of the polynomial. Point zero is the coefficient for the constant term, the last point is the coefficient for the highest-order term:

```
Make/D/O PolyCoefs = {1.5, -1.25, -3.75, 0, 1}
```

This wave can be used with the **poly** function to generate polynomial values. For instance:

```
Make/D/O PWave                // a wave with 128 points
SetScale/I x -2.5,2.5,PWave   // give it an X range of (-2.5, 2.5)
PWave = Poly(PolyCoefs, x)    // fill it with polynomial values

Display PWave                 // and make a graph of it
ModifyGraph zero(left)=1      // add a zero line to show the roots
```

These commands make the following graph: