The values in the yw wave correspond to a row of the table in the example above. That is:

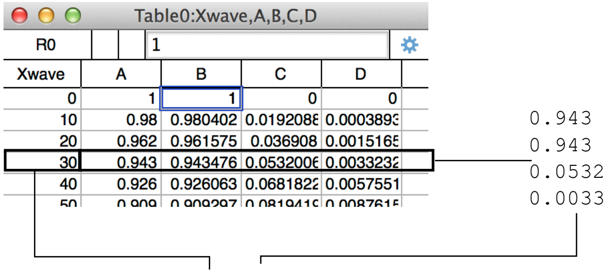


Table0:Xwave,A,B,C,D

R0 | 1

Xwave	A	B	C	D
0	1	1	0	0
10	0.98	0.980402	0.0192088	0.0003893
20	0.962	0.961575	0.036908	0.0015165
30	0.943	0.943476	0.0532006	0.0033232
40	0.926	0.926063	0.0681822	0.0057551
50	0.909	0.909297	0.0819419	0.0087615

```
Function D(pw, xx, yw, dydx)
    Wave pw      // parameter wave (input)
    Variable xx // x value at which to calculate derivatives
    Wave yw      // wave containing y[i] (input)
    Wave dydx    // wave to receive dy[i]/dx (output)

    dydx[0] = <expression for one derivative>
    dydx[1] = <expression for next derivative>
    <etc.>

    return 0
End
```

The wave yw contains the present value, or estimated value, of Y[*i*] at X =xx. You may need this value to calculate the derivatives.

Your derivative function is called many times during the course of a solution, and it will be called at values of X that do not correspond to X values in the final solution. The reason for this is two-fold: First, the solution method steps from one value of X to another using estimates of the derivatives at several intermediate X values. Second, the spacing between X values that you want may be larger than can be calculated accurately, and Igor may need to find the solution at intermediate values. These intermediate values are not reported to you unless you call the **IntegrateODE** operation (see page V-452) in free-run mode.

Because the derivative function is called at intermediate X values, the yw wave is not the same wave as the Y wave you create and pass to IntegrateODE. Note that one row of your Y wave, or one value from each Y wave, corresponds to the elements of the one-dimensional yw wave that is passed in to your derivative function. While the illustration implies that values from your Y wave are passed to the derivative function, in fact the values in the yw wave passed into the derivative function correspond to whatever Y values the integrator needs at the moment. The correspondence to your Y wave or waves is only conceptual.

You should be aware that, with the exception of the parameter wave (pw above) the waves are not waves that exist in your Igor experiment. Do not try to resize them with InsertPoints/DeletePoints and don't do anything to them with the Redimension operation. The yw wave is input-only; altering it will not change anything. The dydx wave is output-only; the only thing you should do with it is to assign appropriate derivative (right-hand-side) values.

Some examples are presented in the following sections.

A First-Order Equation

Let's say you want a numerical solution to a simple first-order differential equation:

$$\frac{dy}{dx} = -ay$$

First you need to create a function that calculates the derivative. Enter the following in the procedure window:

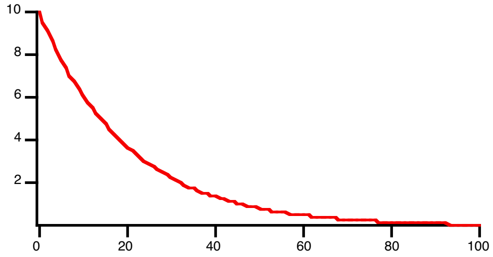
```
Function FirstOrder(pw, xx, yw, dydx)
   Wave pw        // pw[0] contains the value of the constant a
   Variable xx    // not actually used in this example
   Wave yw        // has just one element- there is just one equation
   Wave dydx      // has just one element- there is just one equation
   // There's only one equation, so only one expression here.
   // The constant a in the equation is passed in pw[0]
   dydx[0] = -pw[0]*yw[0]

   return 0
End
```

Paste the function into the procedure window and then execute these commands:

```
Make/D/O/N=101 YY     // wave to receive results
YY[0] = 10            // initial condition- y0=10
Display YY            // make a graph
Make/D/O PP={0.05}    // set constant a to 0.05
IntegrateODE FirstOrder, PP, YY
```

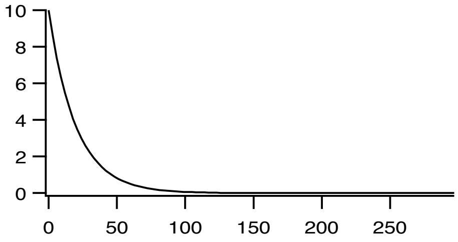
This results in the following graph with the expected exponential decay:



The IntegrateODE command shown in the example is the simplest you can use. It names the derivative function, `FirstOrder`, a parameter wave, `PP`, and a results wave, `YY`.

Because the IntegrateODE command does not explicitly set the X values, the output results are calculated according to the X scaling of the results wave `YY`. You can change the spacing of the X values by changing the X scaling of `YY`:

```
SetScale/P x 0,3,YY         // now the results will be at an x interval of 3
IntegrateODE FirstOrder, PP, YY
```



The same thing can be achieved by using your specified x0 and deltax with the /X flag:

```
IntegrateODE/X={0,3} FirstOrder, PP, YY
```

We presume that you have your own reasons for using the /X={x0, deltax} form. Note that when you do this, it doesn't use the X scaling of your Y wave. If you graph the Y wave the values on the X axis may not match the X values used during the calculation.

Finally, you don't have to use a constant spacing in X if you provide an X wave. You might want to do this to get closely-spaced values only where the solution changes rapidly. For instance:

```
Make/D/O/N=101 XX     // same length as YY
XX = exp(p/20)        // X values get farther apart as X increases
Display YY vs XX      // make an XY graph
ModifyGraph mode=2    // plot with dots so you can see the points
IntegrateODE/X=XX FirstOrder, PP, YY
```