8-25 Write a program that will perform a numerical approximation of the derivative with respect to time of an array of experimental data and plot both functions as two separate subplots in a column format.

INPUTS: A row vector y consisting of an arbitrary number of input data points.

The time step **delt**

OUTPUTS: Upper plot of y as a function of time t

Lower plot of the derivative **z** as a function of time t.

Appropriate labeling should be provided for both curves.

Hint 1: Since the length of the function is arbitrary, you may prefer to have the program determine its length rather than having to enter it as a separate value. Consider the simple command **n=size(x)**. The quantity **n** in general will be a row vector having two values in the following order: (1) the number of rows in **x** and (2) the number of columns in **x**. Since **x** will be a row vector, the number of columns will be **n(1,2)**, or since there will be only one row, MATLAB will recognize **n(2)**.

Hint 2: To suppress the printing on the screen of input data values that you have entered, place a semicolon to the right of the right-hand parenthesis at the end of an **Input** statement.

(a) Run the program, and when polled for the input row vector, type **0:100** and enter it. This will generate a 101-point array with values from 0 to 100 and, in effect, create a linearly increasing input function (a **ramp**). When polled for **delt**, enter a value of 0.01.

(b) Run the program, and when polled for the input row vector, type **randn(1,101)** and enter it. This will generate a 101-point row vector with a random pattern of numbers having a gaussian normal distribution (to be studied later). Statistically, the mean value of the array is 0 and the standard deviation is 1. Again, for **delt**, enter a value of 0.01.

8-26. Write a program that will perform a **first-order** numerical approximation of the running integral with respect to time of an array of experimental data and plot both functions as two separate subplots in a column format.

INPUTS: A row matrix y consisting of an arbitrary number of input data points.

The time step **delt**

OUTPUTS: Upper plot of y as a function of time t

Lower plot of the derivative **z** as a function of time **t**.

Appropriate labeling should be provided for both curves. See the **Hints** in Problem 8-25.

(a) Run the program for the same conditions of part (a) of Problem 8-25.

(b) Run the program for the same conditions of part (b) of Problem 8-25

(c) Compare the results for the random input in Problem 8-25 and 8-26 and comment on the relative size and behavior of the derivative versus the integral.

4-3 MULTIPLE PLOTS ON THE SAME GRAPH

So far we have dealt with only one curve on a given plot. However, MATLAB can generate multiple curves on the same scale whenever it is convenient and reasonable.

The most common situation is where there is one independent variable and two or more dependent variables, each of which is to be plotted as a function of the one independent variable. For multiple plots of this nature to make the most sense, the dependent variables should be of the same order of magnitude. For example, a plot of Bill Gates' income versus time on the same graph as the author's income versus time wouldn't make good sense since one curve would completely obscure the other one. (You can guess which one!)

We will illustrate the command with a hypothetical situation of three dependent variables y_1, y_2, and y_3, each of which is a function of a single independent variable x. We will assume that the vectors for the four variables have the same length. The plot command associated with this situation is as follows:

```
>> plot(x, y1, x, y2, x, y3)                                    (4-31)
```

Note that it is necessary to show x preceding each of the dependent variables.

The preceding situation is based on the same independent variable for all of the curves. Actually, it is possible to have more than one independent variable and the vector sizes in that case can be different. The only requirement is that the vector sizes of a given independent and dependent variable pair should be the same.

To illustrate different independent variables, assume that x1 and x2 are two independent vectors and y1 and y2 are two dependent vectors. Assume that x1 and y1 have the same size and assume that x2 and y2 have the same size. The following command will allow both curves on the same plot:

```
>> plot(x1, y1, x2, y2)                                         (4-32)
```

An alternate way to put more than one curve on a given set of axes is through the **hold on** command. Assuming that one or more curves have already been placed on the axes, but you anticipate adding one or more later, the following command can be used:

```
>> hold on                                                      (4-33)
```

You may then add one or more additional curves. After the curves are complete, the figure option may be opened for new curves by the command

```
>> hold off                                                     (4-34)
```

Labeling Different Curves

An obvious question that arises when there is more one curve is the process of labeling for identification. Actually, MATLAB has numerous options for creating different types of labels, including the use of different symbols on different curves and the use of different types of curves and shading. Once you become proficient in the basic operations of MATLAB, you will probably want to investigate some of the different forms. However, we don't want to overwhelm the reader with too many fine details at this point, so we will limit the development here to one particular form, the **gtext** command. Others will be considered at various points in the text.

The **gtext** command allows the user to specify wording that can be placed on the figure with crosshairs. The format is as follows: