# Iteration II – the REAL DO Loop

In this exercise you will use the DO statement with a REAL variable as the counter to simulate an object falling from an aircraft as a function of time.

## 1. Problem:

An object (such as a bag of flour, or an army field ration) is to be dropped from an airplane, with the hope of hitting a target on the ground. You are to write a program which will compute the position of the bag as it falls, using a DO loop to step over short periods of time.

We will neglect air resistance for now (we will include it in a later program). If the altitude from which the object is released is $A$ (measured in feet, since those are the units used by aircraft altimeters) then the altitude at $t$ seconds after the release is given by the formula

$$y = A - \tfrac{1}{2}\,(32.17)\,t^2\,,$$

where 32.17 ft/sec$^2$ is the acceleration due to gravity. The forward position of the object $t$ seconds after the release is

$$x = x_0 + vt$$

where $x_0$ is the position when the object is released, and $v$ is the forward velocity of the aircraft, measured in feet per second. For simplicity we can assume $x_0 = 0$ in what follows.

Your program should loop over a time interval of up to 30 seconds, with a step size of 0.1 seconds or less, and determine the $x$ and $y$ positions of the object at each time step. The loop should exit early if the altitude of the object becomes zero or less than zero (meaning that the object has struck the ground).

## 2. Input:

Your program should first read the altitude of the aircraft when the object is dropped, in feet.

Next it should read the airspeed of the aircraft when the object is dropped, in knots (nautical miles per hour) since those are the units used by most aircraft airspeed indicators. Your program should convert this airspeed into the velocity $v$, in feet per second. To do this it will help you to know that there are 6076 feet in a nautical mile.

Your program should assume that the initial horizontal position at the time of the drop, $x_0$, is zero.

Do not forget to echo the values of all variables you read in.

### 3. Output:

At each time step (each time through the DO loop) you will want to print the time, followed by the position of the object (horizontal first, then vertical), all on the same line. Since there will be a lot of data you will want to use a `FORMAT` statement to make a nice table where all of the columns line up.

Your output will also look much better and be easier to read if you print headings at the top of each column of data.

If your program gets all the way to 30.0 seconds and the object has not yet struck the ground then it should print a warning message saying that the calculation did not finish. Do not print this warning if the object reaches the ground, but do not go over 30.0 seconds.

### 4. Welcome to the World Wide Web:

The World Wide Web was created by physicists at CERN, the European Center for Particle Physics, to share data and documentation among large numbers of collaborating physicists spread out over the entire world. One of the reasons the web is so successful is that it is extremely easy to put a file on the web.

All you have to do to have a file served out to the world on the web is put the file in a particular directory on a machine which is running the web server software. The computer used for this course, `noether.vassar.edu`, is a web server, and it has been configured so that each user can create web pages simply by putting them in a directory called “`www`” directly under their home directory. Thus Matthew Vassar could share a copy of his program for this exercise by creating the `www` directory and putting a copy of the file in that directory, like so:

```
% cd
% mkdir www
% cp mavassar08.f www
```

This file can then be viewed with any web browser simply by giving it the web address (called a URL – Uniform Resource Locator):

```
http://noether.vassar.edu/~mavassar/mavassar08.f
```

Note that the “`~mavassar`” refers to any file in the `www` directory of that particular user, but “www” is not included in the URL. As another example, the file referenced by the URL http://noether.vassar.edu/~myers/fortran/hello.f is in the directory www/fortran/ under the home directory of user “myers”.

On other comptuers the directory shared to the web might have a different name, such as web or public_html, but the idea is the same. Any file you put in that directory is published to the world. You should keep in mind that files in any subdirectory of this directory are also available to the world. There are ways to limit access, but in general you should be careful to only put files in your web area that you really want to be public.

### 5. Optional Improvements:

You do not have to make these improvements to your program, but you can do so if you really want to make it work nicely:

- Using a counter variable to count the number of lines printed, you can print a new column heading for each page of output. A standard page of computer printout has 66 lines on it. You may (or may not) be interested in what the MOD function (for modulo arithmetic) does. You can force a skip to a new page of output by printing the character ^L (control-L), which can be created by the function call CHAR(12). That is, the following line causes a skip to a new page:

  ```
  PRINT *, CHAR(12)
  ```

- You can, if you wish, create a www directory and put a copy of your program file in it so that the instructory (or anybody else) can view it on the web. Be sure to mention that you have done so to your instructor when you turn in the exercise.

  You can also put your typescript file in your web area instead of printing it out to turn it in. But you need to distinguish it from other typescript files, so rename it to be a “log” file. Remember that the ‘mv‘ command in Unix is used to rename files. For example

  ```
  %  mv typescript mavassar08.log
  %  mv  mavassar08.log www
  ```

### 6. Reading:

You should read about the different data types in Fortran, at least about the difference between INTEGER and REAL variables. You should also read about how to use a REAL variable as the counter in a DO loop.

To produce a nice organized table of output data you should read about the uses of the FORMAT statement.