Chapter 1

Examples of problems involving ordinary differential equations: Projectile motion, Harmonic motion, and celestial mechanics.

Radioactive decay:
Nu(t) is the number of uranium nuclei that are present in the sample at time t.

$$\frac{dN_U}{dt} = -\frac{N_U}{\tau} \qquad N_U = N_U(0)\, e^{-t/\tau}$$

Were tau is the "time constant", Nu(0) is the number of nuclei at t = 0.
Tau is also the mean lifetime of a nucleus.

A numerical approach:
1) Taylor expansion for Nu:

$$N_U(\Delta t) = N_U(0) + \frac{dN_U}{dt}\Delta t + \frac{1}{2}\frac{d^2N_U}{dt^2}(\Delta t)^2 + \cdots$$

If Δt is small, then

$$N_U(\Delta t) \approx N_U(0) + \frac{dN_U}{dt}\Delta t$$

2) Definition of derivative:

$$\frac{dN_U}{dt} \equiv \lim_{\Delta t \to 0} \frac{N_U(t+\Delta t) - N_U(t)}{\Delta t} \approx \frac{N_U(t+\Delta t) - N_U(t)}{\Delta t}$$

And again if Δt is small, then

$$N_U(t+\Delta t) \approx N_U(t) + \frac{dN_U}{dt}\Delta t$$

From the first formula, we can substitute values to get:

$$N_U(t+\Delta t) \approx N_U(t) - \frac{N_U(t)}{\tau}\Delta t$$

Therefore, we can estimate the value of Nu, given a certain Nu(t).
This approach is called the Euler Method.

Design and construction of a working program: codes and pseudocodes:
Always start with an outline of how the problem is to be solved and what variables or parameters will be needed.
Structure of the program:

1) Declare the necessary variables
2) Initialize all variables and parameters
3) Do the calculation
4) Store the results

The subroutine *initialize* sets the initial values of the variables, *calculate* uses the Euler method to do the computation, and *store* puts the results into a file for later use (such as a graphical display).

Itinitialize:
(1) prompt for and initialize Nu(t), tau, and Δt.
(2) set initial value of time, t(0).
(3) set number of time steps for calculation.

Calculate:
(1) For each time step *i* (beginning with *i* = 1), calculate Nu and t at step *i* + 1:
- Nu(t_(i+1)) = Nu(t_i) – (Nu(t_i) / tau)dt
- t_(i+1) = t_i + Δt
- repeat for n – 1 time steps

Stores:
It writes the result to a file, so it can then be read from this file, in order to plot the results, or use them in a subsequent calculation.

This program calculates how Nu varies with time and puts the results as numbers into a file. A really important part is understanding the results, and this can be done by plotting them in a graphical form.

Testing your program:
- Have an idea of what the output should be
- Compare the values with the exact result (if there are)
- Check that it gives the same result with different step sizes

Programming guidelines and philosophy:

- *Program structure.* Use subroutines to organize the major tasks and make the program more readable and understandable. The main program for the decay problem was basically an outline of the program; we recommend this style for all programs. Use subroutines and functions to perform any jobs that take more than a few lines of code, or that are required repeatedly.
- *Use descriptive names.* Choose the names of variables and subroutines ac- cording to the problem at hand. Descriptive names make a program easier to understand, as they act as built-in comment statements.
- *Use comment statements.* Include comment statements to explain program logic and describe variables. A short subroutine that uses descriptive variable names should not need a large number of comment statements.
- *Sacrifice (almost) everything for clarity.* It is often tempting to write a crit- ical piece of code in a very compact or terse manner in the misguided belief that this will make the program run faster. It is almost always better to take a few more lines, or a few more variables, to do a job, if it makes the code more understandable.
- *Take time to make graphical output as clear as possible.* Think carefully about what quantities to plot and in what manner. The axes should be labeled clearly (including units, where appropriate), and parameter values given directly on the graph.