Lesson 1

1. What is the difference between a numerical method and a numerical algorithm?

Given the mathematical description of a problem, a numerical method lays down the broad approach to be adopted to solve the problem numerically. Once the numerical method has been chosen, there are a wide variety of algorithms than can be chosen to do a step by step implementation of the numerical method. For instance, one may use the Finite Element **Method** to solve a particular boundary value problem in structural dynamics. However the algorithm to be used may be widely different: one can use an unconditionally stable **implicit** algorithm or a conditionally stable **explicit** algorithm for instance to advance the solutions in time.

2. When will a simple iterative scheme, where the iterates are defined by $x_{n+1} = F(x_n)$, to find the root of the one-dimensional equation $F(x) = 0$ converge?

If the derivative of the function $F(x)$ is strictly less than zero for all x in the neighbourhood of the starting iterates x_0 and x_1, and this neighbourhood includes the root, we can be sure of convergence. On the other hand, if the magnitude of the slope is greater than one near the root the iterate will converge to the root only in exceptional cases, no matter how close to the root one chooses the starting value of the iteration x_0

3. Briefly describe the Newton Raphson Method.

The Newton Raphson method, consists of an iterative scheme where at each successive iteration, we approximate the function $y=f(x)$ by its tangent at that point, and use that to find the next iterate:

$$f(x_{n+1}) = 0 \approx f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$$
$$\therefore x_{n+1} = x_n + \frac{f(x_n)}{f'(x_n)}$$

The next iterate x_{n+1} is the point where the linear approximation to the function $y=f(x)$ given by the straight line passing through (x_n, $f(x_n)$) and with slope $f'(x_n)$ intersects the X axis. The approximation of $y=f(x)$ by its tangent at the point (x_n, $f(x_n)$) is equivalent to replacing the function with the first degree terms in its Taylor series about $x = x_n$.

4. When is a numerical algorithm unstable?

An algorithm is unstable when small perturbations in the input data or in the intermediate computations can lead to gross errors, i.e. totally destroy the result of the computations. Perturbations in the input data can occur, e.g., due to minor errors in recording experimental results. Perturbations in the intermediate computations can occur due to numerical roundoff resulting from finite precision arithmetic.
The choice of algorithm is crucial in determining whether a numerical method will give stable results. Certain algorithms are inherently unstable, others are stable for a certain

range of input data, while still others are unconditionally stable, i.e. they are stable for all ranges of input data.

5. What is meant by the “accuracy” of a numerical algorithm?

In addition to giving stable results, it is also desirable that the results of a numerical algorithm be accurate. The errors which measure the deviation of the results obtained using a numerical algorithm from the exact or ‘true’ solution of a problem must be as small as possible.