Lecture 24

SOLUTION OF DISCRETE ALGEBRAIC SYSTEMS

24.1 INTRODUCTION

Application of FDM, FVM or FEM leads to a system of algebraic equations which may be linear or non-linear depending on the problem. For non-linear systems, we have to use an iterative procedure which can be

- Newton-Raphson type method: rate of convergence is very fast provided initial guess is selected properly, OR
- A global method (e.g. Picard iteration): guarantees converged solution starting from an arbitrary initial guess, but rate of convergence is often very slow.
- A globally convergent method which employs a mix of global and Newton's method (Press et al., 2002).

Most of these methods require solution of a linear system at each iteration. Thus, we require an efficient linear equation solver for linear as well as non-linear problems, and hence, we concentrate on solution of linear systems in this and next two lectures.

Choice of a particular solution scheme must account for

- Sparse nature of the algebraic system obtained from FD/FV/FE discretization
- Nature of the linear system (whether it's symmetric or not)

Before proceeding further, let us have a brief look at methods for non-linear equations. There are numerous methods available in the literature for solution of nonlinear algebraic equations (see Press et al., 2002). We discuss two simple, but widely used, approaches in this lecture.

24.2 SEQUENTIAL (PICARD) ITERATION

The simplest approach to solve a nonlinear equation $f(x)=0$ is to recast it in the form

$$x = g(x) \tag{24.1}$$

Then, starting from an initial guess x_0, successive iterates can be computed as

$$x_{k+1} = g(x_k), \quad k = 0,1,2,.... \tag{24.2}$$

Iterations are continued till a converged solution is obtained. Similar procedure can also be extended to a system of coupled non-linear equations

$$\mathbf{F(x) = 0} \tag{24.3}$$

which can be re-cast as

$$\mathbf{x = G(x)} \tag{24.4}$$

Permitting computation of improved iterates starting from an initial guess $\boldsymbol{x}^0$ as

$$x_i^{k+1} = G_i(\mathbf{x}^k), \quad k = 0,1,2,.... \tag{24.5}$$

Another iterative approach can be formulated by re-casting the system of equations (24.3) as a linearized system given by

$$\mathbf{A(x)x = b(x)} \tag{24.6}$$

in which system matrix **A** and load vector **b** are functions of unknown vector **x**. In this case, starting from the initial guess $\boldsymbol{x}^0$, improved iterates are computed by solving a linear system of equation at each iteration, i.e.

$$\mathbf{A}(\mathbf{x}^k)\mathbf{x}^{k+1} = \mathbf{b}(\mathbf{x}^k), \quad k = 0,1,2,..... \tag{24.7}$$

In the preceding equation, system matrix **A** load vector **b** are evaluated using the current known iterate of vector **x**. This

24.3 NEWTON-RAPHSON METHOD

Newton-Raphson method is based on Taylor series expansion of function $f(x)$ around a guessed value, i.e.

$$f(x) = f(x_k) + (x - x_k) f'(x_k) + \frac{(x - x_k)^2}{2!} f''(x_k) + ... \tag{24.8}$$

In Eq. (24.8), superscript primes indicate derivatives. If we retain only first two terms on RHS, then an improved guess can be estimated as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0,1,2,.... \tag{24.9}$$

The preceding approach can be easily generalized for a coupled nonlinear system (24.3). For a small increment $\delta\mathbf{x}$, function F_i can be expanded in Taylor series as

$$F_i(\mathbf{x} + \delta\mathbf{x}) = F_i(\mathbf{x}) + \sum_{j=1}^{n} \frac{\partial F_i}{\partial x_j} \delta x_j + O(\delta\mathbf{x}^2) \tag{24.10}$$

By neglecting higher order terms and setting $F_i(\mathbf{x} + \delta\mathbf{x}) = 0$, we get

$$\sum_{j=1}^{n} \frac{\partial F_i}{\partial x_j} \delta x_j = -F_i(\mathbf{x}) \ \text{ or } \ J_{ij}\delta x_j = -F_i(\mathbf{x}), \quad J_{ij} = \frac{\partial F_i}{\partial x_j} \tag{24.11}$$

In the preceding equation, **J** is called the *Jacobian* matrix. For a given iterate $\mathbf{x}^k$, we first compute (i) the function values and (ii) the Jacobian, and then solve Eq. (24.11) to get the correction vector $\delta\mathbf{x}^k$. Thus, new iterate is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \delta\mathbf{x}^k, \qquad \delta\mathbf{x}^k = -\mathbf{J}^{-1}(\mathbf{x}^k)\mathbf{F}(\mathbf{x}^k) \tag{24.12}$$

Newton iteration converges very rapidly if the initial guess is close to the solution. Its convergence not guaranteed starting from an arbitrary guess. To get over this problem, a globally convergent algorithm based on line searches and back-tracking in conjunction with Newton-Raphson iteration is usually preferred (see Press et al., 2002 for further details of this algorithm).

REFERENCES/FURTHER READING

Ferziger, J. H. And Perić, M. (2003). *Computational Methods for Fluid Dynamics*. Springer.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (2002). *Numerical Recipes in C++*. Cambridge University Press, Cambridge.