CHAPTER - IV

SOLUTIONS OF SYSTEMS OF NON LINEAR EQUATIONS

The problem of finding the solution of a set of non linear equations is much more difficult than for linear equations. Moreover, there are no good and general methods for solving systems of more than one non- linear equations.

Consider non- linear problem, where we want to solve simultaneously the system

$$\begin{aligned} f_1(x_1, x_2, ..., x_n) &= 0 \\ f_2(x_1, x_2, ..., x_n) &= 0 \\ &...... \\ f_n(x_1, x_2, ..., x_n) &= 0 \end{aligned} \qquad (4.1)$$

The most commonly used methods of solution are the iterative methods and Newton – Raphson method.

i. Iterative method

Fixed -Point iteration for Functions of Several Variables

Where each function f_i of Eqn (4.1) can be thought of as mapping a vector x = $(x_1,x_2,x_3, , ,x_n)^t$ of the n-dimensional space R^n into the real line R. This system of n nonlinear equations in n unknowns can also be represented by defining a function F mapping R^n into R^n as

$$\mathbf{F}(x_1, x_2, \ldots, x_n) = (f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), \ldots, f_n(x_1, x_2, \ldots, x_n))^t.$$

If vector notation is used to represent the variables x_1,x_2 x_n, system (4.1) assumes the form

$$\mathbf{F(x) = 0.} \qquad (4.2)$$

The functions $f_1,f_2,... ,f_n$ are the coordinate functions of **F.**

The following procedure can be used for solving non- linear problem by iterative method.

i. the i^{th} equation is solved for x_i, the system is changed into the fixed-point problem

ii. Using the initial guess x_2, x_3 … solve for x_1

iii. Using the values of x_1 from step *ii* and x_3 ,x_4, … solve for x_2

iv. Using the value of x_1 from step *ii* and that of x_2 from step *iii* solve for x_3

v. Repeat this procedure until the required accuracy is achieved. /interval is smaller than some specified tolerance/

Example 1: solve the system of non linear equation by iterative method

$$3x_1 - \cos(x_2x_3) - \frac{1}{2} = 0,$$
$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0,$$
$$e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

If the i^th^ equation is solved for x_i, the system is changed into the fixed-point problem

$$x_1 = \frac{1}{3}\cos(x_2x_3) + \frac{1}{6},$$
$$x_2 = \frac{1}{9}\sqrt{x_1^2 + \sin x_3 + 1.06} - 0.1, \qquad (4.3)$$
$$x_3 = -\frac{1}{20}e^{-x_1x_2} - \frac{10\pi - 3}{60}.$$

Let $\mathbf{G} : \mathbb{R}^3 \to \mathbb{R}^3$ be defined by $\mathbf{G}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), g_3(\mathbf{x}))^t$, where

$$g_1(x_1, x_2, x_3) = \frac{1}{3}\cos(x_2x_3) + \frac{1}{6},$$
$$g_2(x_1, x_2, x_3) = \frac{1}{9}\sqrt{x_1^2 + \sin x_3 + 1.06} - 0.1,$$
$$g_3(x_1, x_2, x_3) = -\frac{1}{20}e^{-x_1x_2} - \frac{10\pi - 3}{60}.$$

To approximate the fixed point p, we choose $x(0) = (0.1, 0.1, -0.1)^t$. The sequence of vectors generated by

$$x_1^{(k)} = \frac{1}{3}\cos x_2^{(k-1)}x_3^{(k-1)} + \frac{1}{6},$$
$$x_2^{(k)} = \frac{1}{9}\sqrt{\left(x_1^{(k-1)}\right)^2 + \sin x_3^{(k-1)} + 1.06} - 0.1,$$
$$x_3^{(k)} = -\frac{1}{20}e^{-x_1^{(k-1)}x_2^{(k-1)}} - \frac{10\pi - 3}{60}$$

converges to the unique solution of (4.3). The results in Table 4.1 were generated until; the error $/X^{(k)} - X^{(k-1)}/ < 10^{-5}$

Table 4.1

k	$X_1^{(k)}$	$X_2^{(k)}$	$X_3^{(k)}$	$/X^{(k)} - X^{(k-1)}/$
0	0.1	0.1	-0.1	
1	0.49998333	0.00944115	-0.52310127	0.423
2	0.49999593	0.00002557	-0.52336331	9.4×10^{-3}
3	0.5000000	0.00001234	-0.52359814	2.3×10^{-4}
4	0.5000000	0.00000003	-0.52359847	1.2×10^{-5}
5	0.5000000	0.00000002	-0.52359877	3.1×10^{-7}

One way to accelerate convergence of the fixed-point iteration is to use the latest estimates $x_1^{(k)}, \ldots, x_{i-1}^{(k)}$ instead of $x_1^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$ to compute $x_i^{(k)}$, as in the **Gauss-Seidel method** for linear systems. The component equations then become

$$x_1^{(k)} = \frac{1}{3}\cos\left(x_2^{(k-1)}x_3^{(k-1)}\right) + \frac{1}{6},$$

$$x_2^{(k)} = \frac{1}{9}\sqrt{\left(x_1^{(k)}\right)^2 + \sin x_3^{(k-1)} + 1.06} - 0.1,$$

$$x_3^{(k)} = -\frac{1}{20}e^{-x_1^{(k)}x_2^{(k)}} - \frac{10\pi - 3}{60}.$$

So the convergence was indeed accelerated for this problem by using **the Gauss-Seidel method**. However, this method does not always accelerate the convergence.

Table 4.2

k	$X_1^{(k)}$	$X_2^{(k)}$	$X_3^{(k)}$	$/X^{(k)} - X^{(k-1)}/$
0	0.1	0.1	-0.10000000	
1	0.49998333	0.02222979	-0.52304613	0.423
2	0.49997747	0.00002815	-0.52359807	2.2×10^{-2}
3	0.5000000	0.00000004	-0.52359877	2.8×10^{-5}
4	0.5000000	0.00000000	-0.52359877	3.8×10^{-8}

ii. Newton – Raphson method

The problem in Example 1 of the previous section is transformed into a convergent fixed- point problem by algebraically solving the three equations for the three variables x_1, x_2, and x_3. It is, however, rather unusual for this technique to be successful. In this section, we consider an algorithmic procedure to perform the transformation in a more general situation.

One of the methods available for solving such a system /non linear system of equations/ is based on a multi-dimensional version of Newton–Raphson method. The method consists of linearizing the non-linear systems at each iteration step and solving system of linear equations.

To construct the algorithm that led to an appropriate fixed-point method in the one- dimensional case, we found a function ϕ with the property that $g(x) = x-\phi(x)f(x)$ gives quadratic convergence to the fixed point p of the function g(x). From this condition Newton's method evolved by choosing $\phi(x)= 1/f'(x)$, assuming that $f'(x)\neq 0$. Using a similar approach in the n-dimensional case involves a matrix

$$A(\mathbf{x}) = \begin{bmatrix} a_{11}(\mathbf{x}) & a_{12}(\mathbf{x}) & \cdots & a_{1n}(\mathbf{x}) \\ a_{21}(\mathbf{x}) & a_{22}(\mathbf{x}) & \cdots & a_{2n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ a_{n1}(\mathbf{x}) & a_{n2}(\mathbf{x}) & \cdots & a_{nn}(\mathbf{x}) \end{bmatrix},$$

where each of the entries $a_{ij}(x)$ is a function from R^n into R. This requires that A(x) be found so that $G(x) = X - A(x)^{-1}F(x)$ gives quadratic convergence to the solution of **F(x) =** 0, assuming that A(x) is nonsingular at the fixed point p of G.

Defining the matrix $J(\mathbf{x})$ by

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix},$$

An appropriate choice for A (x) is, consequently, A (x) = J (x) since this satisfies condition $\partial g_i(\mathbf{p})/\partial x_k = 0$, for each $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, n$. The function G is defined by $G(x) = X - J(x)^{-1}F(x)$, and the functional iteration procedure evolves from selecting $x^{(0)}$ and generating, for k > 1.

$$\mathbf{X^{(k)} = G(x^{(k-1)}) = x^{(k-1)} - J(x^{(k-1)})^{-1}F(x^{(k-1)})}$$

This is called **Newton's method for non-linear systems**, and it is generally expected to give quadratic convergence, provided that a sufficiently accurate starting value is known and $J(p)^{-1}$ exists. The matrix J(x) is called the **Jacobian matrix** and has a number of applications in analysis.

$$\begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix} = \begin{bmatrix} x_1^{(k-1)} \\ x_2^{(k-1)} \\ x_3^{(k-1)} \end{bmatrix} + \begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix},$$

where

$$\begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix} = -\left(J\left(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)}\right)\right)^{-1} \mathbf{F}\left(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)}\right)$$

The disadvantages in Newton's method are:-

- It is often not simple to evaluate the partial derivatives
- For most of the problems, it is not always easy to find initial guesses that will lead to a solution.
- Need to solve system of linear equations at each iterative step.

Example 2: Solve the nonlinear system by Newton's method

$$3x_1 - \cos(x_2x_3) - \frac{1}{2} = 0,$$
$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0,$$
$$e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

Newton's method will be used to obtain this approximation using the initial approximation is $x^{(0)}=(0.1,0.1,-0.1)$ and

$$\mathbf{F}(x_1, x_2, x_3) = (f_1(x_1, x_2, x_3), f_2(x_1, x_2, x_3), f_3(x_1, x_2, x_3))^t,$$

where

$$f_1(x_1, x_2, x_3) = 3x_1 - \cos(x_2x_3) - \frac{1}{2},$$
$$f_2(x_1, x_2, x_3) = x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06,$$

and

$$f_3(x_1, x_2, x_3) = e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3}.$$

The Jacobian matrix $J(\mathbf{x})$ for this system is

$$J(x_1, x_2, x_3) = \begin{bmatrix} 3 & x_3 \sin x_2x_3 & x_2 \sin x_2x_3 \\ 2x_1 & -162(x_2 + 0.1) & \cos x_3 \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{bmatrix}$$

Thus, at the k^{th} step, the linear system $J(x^{(k-1)})y^{(k-1)} = -F(x^{(k-1)})$ must be solved.

$$\begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix} = -\left(J\left(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)}\right)\right)^{-1} \mathbf{F}\left(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)}\right)$$

$$\begin{bmatrix} 3 & x_3 \sin x_2x_3 & x_2 \sin x_2x_3 \\ 2x_1 & -162(x_2+0.1) & \cos x_3 \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = -\begin{bmatrix} 3x_1 - \cos x_2x_3 - \frac{1}{2} \\ x_1^2 - 81\left(x_2+0.1\right)^2 + \sin x_3 + 1.06 \\ e^{-x_1x_2} + 20x_3 + \frac{10\pi-3}{3} \end{bmatrix} \quad *$$

Taking $x_1^{(0)}=0.1$, $x_2^{(0)}=0.1$and $x_3^{(0)}=-0.1$ as initial approximation, equation (*) reduces to

$$\begin{vmatrix} 3 & 9.99\text{x } 10^{-4} & -9.99\text{x } 10^{-4} \\ 0.2 & -32.4 & 0.995 \\ -0.099 & -0.099 & 20 \end{vmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{vmatrix} 1.19995 \\ 2.2698 \\ -8.46 \end{vmatrix}$$

Solving this linear system of equation, results in $y_1=0.39987$, $y_2=-\ 0.0805331$ and $y_3=-0.42121$. Use this results, the next approximation for the unknown variables can be calculated as

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{bmatrix} = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{bmatrix} + \begin{bmatrix} y_1^{(0)} \\ y_2^{(0)} \\ y_3^{(0)} \end{bmatrix}$$

$X_1^{(1)} = x_1^{(0)} + y_1 = 0.1+0.39987 = \underline{\underline{0.49987}}$

$X_2^{(1)} = x_2^{(0)} + y_2 = 0.1-0.0805331 = \underline{\underline{0.019466}}$ and

$X_3^{(1)} = x_3^{(0)} + y_3 = -0.1-0.42121 = -\underline{\underline{0.521520}}$

Now substituting the calculated values of $X_1^{(1)}$, $X_2^{(1)}$ and $X_3^{(1)}$ in eqn (*),we construct a new linear system and solve it to obtain the new correction factor y_1, y_2, y_3 and then calculate the new approximations $X_1^{(2)}$, $X_2^{(2)}$ and $X_3^{(2)}$. This process is continued until convergent is achieved.

The results using this iterative procedure are shown in table 4.3

<u>Table 4.3</u>

k	$X_1^{(k)}$	$X_2^{(k)}$	$X_3^{(k)}$	$/X^{(k)} - X^{(k-1)}/$
0	0.1	0.1	-0.1	
1	0.49987	0.01946686	-0.52152047	0.422
2	0.50004593	0.00158859	-0.52355711	1.79×10^{-2}
3	0.50000034	0.00001244	-0.52359845	1.58×10^{-3}
4	0.50000000	0.00000000	-0.52359877	1.24×10^{-5}
5	0.50000000	0.00000000	-0.52359877	0

The example illustrates that Newton's method can converge very rapidly once an approximation is obtained that is near the true solution.