## *3.1Introduction*

The optimal power flow (OPF) was first introduced by Carpentier in 1962. The goal of OPF is to find the optimal settings of a given power system network that optimize the system objective functions such as total generation cost, system loss, bus voltage deviation, emission of generating units, number of control actions, and load shedding while satisfying its power flow equations, system security, and equipment operating limits. Different control variables, some of which are generators real power outputs and voltages, transformer tap changing settings, phase shifters, switched capacitors, and reactors, are manipulated to achieve an optimal network setting based on the problem formulation.

According to the selected objective functions, and constraints, there are different mathematical formulations for the OPF problem. They can be broadly classified as follows:

1. Linear problem in which objectives and constraints are given in linear forms with continuous control variables
2. Nonlinear problem where either objectives or constraints or both combined are nonlinear with continuous control variables
3. Mixed - integer linear problems when control variables are both discrete and continuous

Various techniques were developed to solve the OPF problem. The algorithms may be classified into three groups:

1. Conventional optimization methods,
2. Intelligence search methods, and
3. Non-quantity approach to address uncertainties in objectives and constraints [8].

## 3.2 Conventional Optimization Method:

Traditionally, conventional methods are used to effectively solve OPF. The application of these methods had been an area of active research in the recent past. The ***conventional methods*** are based on mathematical programming approaches and used to solve different size of OPF problems. To meet the requirements of different objective functions, types of application and nature of constraints, the popular conventional methods is further sub divided into the following:

(a) Gradient Method
(b) Newton-Raphson Method
(c) Linear Programming Method
(d) Quadratic Programming Method
(e) Interior Point Method

Even though, excellent advancements have been made in classical methods, they suffer with the following disadvantages: In most cases, mathematical formulations have to be simplified to get the solutions because of the extremely limited capability to solve real-world large-scale power system problems. They are weak in handling qualitative constraints. They have poor convergence, may get stuck at local optimum, they can find only a single optimized solution in a single simulation run, they become too slow if number of variables are large and they are computationally expensive for solution of a large system.

For this thesis *Newton-Raphson* method was implemented as a *conventional method* on the IEEE 39 New England test system to find the optimal power flow.

### 3.2.1 Newton – Raphson Method:

In the area of Power systems, Newton’s method is well known for solution of Power Flow. It has been the standard solution algorithm for the power flow problem for a long time The Newton approach is a flexible formulation that can be adopted to develop different OPF algorithms suited to the requirements of different applications. Although the Newton approach exists as a concept entirely apart from any specific method of implementation, it would not be possible to develop practical OPF programs without employing special Sparsity techniques. The concept and the techniques together comprise the given approach. Other Newton-based approaches are possible.

Newton's method is a very powerful solution algorithm because of its rapid convergence near the solution. This property is especially useful for power system applications because an initial guess near the solution is easily attained. System voltages will be near rated system values, generator outputs can be estimated from historical data, and transformer tap ratios will be near 1.0 p.u.

### *3.2.1.1 Newton-Raphson Solution Algorithm:*

Let us consider a *N*-bus power system having *NG* number of thermal power generators. Then the aim of optimal power flow problem is to minimize the cost of thermal power generation,

$$F_{c_{Total}} = \sum_{i=1}^{NG} F_{c_i} = \sum_{i=1}^{NG} \alpha_i (P_{g_i})^2 + \beta_i P_{g_i} + \gamma_i \text{ unit of cost/hr} \quad (3.1)$$

Subjected to

(i) Active power balance in the network

$$P_i(|V|, \delta) - P_{g_i} + P_{load} = 0 \text{ for i=1,2,3,....,N} \quad (3.2)$$

Where $P_i$ = active power injection at i-th bus and is a function of |V| and δ. For load buses [i.e for i= (NG+1),(NG+2),…(N)], $P_{g_i}$ =0;

(ii) Reactive power balance in the network

$$Q_i(V, \delta) - Q_{g_i} + Q_{load} = 0 \text{ for i=(NG+1),(NG+2),....,N} \quad (3.3)$$

Where $Q_i$ = reactive power injection at i-th bus and also a function of |V| and δ. $Q_{g_i}$ =reactive power generation at *i-th* bus;

*(iii)* Security related constrains *(also called soft constrain)*.these constrain are discussed in chapter two in equations (2.6) to (2.8).

The constraint minimization problem can be transformed into unconstrained one by augment the load flow constraints into objective function. The additional variables are known as the *Lagrange Multiplier Functions* or *Incremental Cost Function* in power system optimization. The *Lagrangian Function* then becomes

$$\mathrm{L}(\mathrm{P}_g . |V|, \delta) = \sum_{i=1}^{NG} F_c(P_{g_i}) + \sum_{i=1}^{N} \lambda_{Pi}[P_i(|V|,\delta) - P_{g_i} - P_{load_i}] + \sum_{i=NG+1}^{N} \lambda_{qi}[Q_i(|V|,\delta) - Q_{g_i} - Q_{load_i}] \tag{3.4}$$

The optimization Problem is solved, only if the following equation satisfied,

$$\frac{\partial L}{\partial P_{g_i}} = \frac{\partial F}{\partial P_{g_i}} - \lambda_{p_i} \text{ for i=1,2,3,...,NG} \tag{3.5}$$

$$\frac{\partial L}{\partial \delta_i} = \sum_{k=1}^{N}\left[\lambda_{p_k} \frac{\partial P_k}{\partial \delta i}\right] + \sum_{k=NG+1}^{N}\left[\lambda_{q_k} \frac{\partial Q_k}{\partial \delta_i}\right] \text{ for i=2,3,...,NG} \tag{3.6}$$

From equation (3.4) we can write the following

$$\frac{\partial L}{\partial \lambda_{p_i}} = P_i\left(|V|,\delta\right) - P_{g_i} + P_{load_i} \text{for i=1,2,3,...,N} \tag{3.7}$$

And

$$\frac{\partial L}{\partial |V_i|} = \sum_{k=1}^{N}\left[\lambda_{p_k} \frac{\partial P_k}{\partial |V_i|}\right] + \sum_{k=NG+1}^{N}\left[\lambda_{q_k} \frac{\partial Q_k}{\partial |V_i|}\right] \text{ for i=NG+1,...,N} \tag{3.8}$$

Further to this

$$\frac{\partial L}{\partial \lambda_{q_i}} = Q_i\left(|V|,\delta\right) - Q_{g_i} + Q_{load_i} \text{for i=NG+1,...,N} \tag{3.9}$$

Any small variation in control variables about their initial values is obtained by forming differential as given below:

$$\sum_{k=1}^{NG} \frac{\partial^2 L}{\partial P_{g_i} \partial P_{g_k}} \Delta P_{g_k} + \sum_{k=2}^{N} \frac{\partial^2 L}{\partial P_{g_i} \partial \delta_k} \Delta \delta_k + \sum_{k=1}^{N} \frac{\partial^2 L}{\partial P_{g_i} \partial \lambda_{p_k}} \Delta \lambda_{p_k} + \sum_{k=NG+1}^{N} \frac{\partial^2 L}{\partial P_{g_i} \partial \lambda_{q_k}} \Delta \lambda_{q_k} + \sum_{k=NG+1}^{N} \frac{\partial^2 L}{\partial P_{g_i} \partial |V_k|} \Delta |V_k| = -\frac{\partial L}{\partial P_{g_i}}$$

$$\text{For i=1,2,3,…,NG} \tag{3.10}$$

$$\sum_{k=1}^{NG}\frac{\partial^2 L}{\partial\delta_i\partial P_{g_k}}\Delta P_{g_k}+\sum_{k=2}^{N}\frac{\partial^2 L}{\partial\delta_i\partial\delta_k}\Delta\delta_k+\sum_{k=1}^{N}\frac{\partial^2 L}{\partial\delta_i\partial\lambda_{p_k}}\Delta\lambda_{p_k}+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial\delta_i\partial\lambda_{q_k}}\Delta\lambda_{q_k}+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial\delta_i\partial|V_k|}\Delta|V_k|=-\frac{\partial L}{\partial\delta_i}$$

For i=1,2,3,…,N (3.11)

$$\sum_{k=1}^{NG}\frac{\partial^2 L}{\partial\lambda_{p_i}\partial P_{g_k}}\Delta P_{g_k}+\sum_{k=2}^{N}\frac{\partial^2 L}{\partial\lambda_{p_i}\partial\delta_k}\Delta\delta_k+\sum_{k=1}^{N}\frac{\partial^2 L}{\partial\lambda_{p_i}\partial\lambda_{p_k}}\Delta\lambda_{p_k}+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial\lambda_{p_i}\partial\lambda_{q_k}}\Delta\lambda_{q_k}+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial\lambda_{p_i}\partial|V_k|}\Delta|V_k|=-\frac{\partial L}{\partial\lambda_{p_i}}$$

For i=1,2,3,…,N (3.12)

$$\sum_{k=1}^{NG}\frac{\partial^2 L}{\partial|V_i|\partial P_{g_k}}\Delta P_{g_k}+\sum_{k=2}^{N}\frac{\partial^2 L}{\partial|V_i|\partial\delta_k}\Delta\delta_k+\sum_{k=1}^{N}\frac{\partial^2 L}{\partial|V_i|\partial\lambda_{p_k}}\Delta\lambda_{p_k}+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial|V_i|\partial\lambda_{q_k}}\Delta\lambda_{q_k}+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial|V_i|\partial|V_k|}\Delta|V_k|=-\frac{\partial L}{\partial|V_i|}$$

For i=NG+1,…,N (3.13)

$$\sum_{k=1}^{NG}\frac{\partial^2 L}{\partial\lambda_{q_k}\partial P_{g_k}}\Delta P_{g_k}+\sum_{k=2}^{N}\frac{\partial^2 L}{\partial\lambda_{q_k}\partial\delta_k}\Delta\delta_k+\sum_{k=1}^{N}\frac{\partial^2 L}{\partial\lambda_{q_k}\partial\lambda_{p_k}}\Delta\lambda_{p_k}+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial\lambda_{q_k}\partial\lambda_{q_k}}\Delta\lambda_{q_k}+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial\lambda_{q_k}\partial|V_k|}\Delta|V_k|=-\frac{\partial L}{\partial\lambda_{q_k}}$$

For i=NG+1,…,N (3.14)

Let us now differentiate equations (3.5) to (3.9) with respect to control variables ($P_{gi}$, $\delta_i$, $\lambda_{pi}$ , $\lambda_{qi}$ and $V_i$) to get second order partial derivative required for equations (3.5) to (3.9) as follow:

$$\frac{\partial^2 L}{\partial P^2_{g_i}}=\frac{\partial^2 F_{c_i}}{\partial P^2_{g_i}}=2a_i$$ for $i$=1,2,3,…,NG (3.15)

$$\frac{\partial^2 L}{\partial P_{g_i}\partial P_{g_k}}=0$$ for $i$=1,2,3,…,NG;k=1,2,3,…,NG but i≠k (3.16)

$$\frac{\partial^2 L}{\partial P_{g_i}\partial\delta_k}=\frac{\partial^2 L}{\partial\delta_k\partial P_{g_i}}=0$$ for $i$=1,2,3,…,NG;k=1,2,3,…,N (3.17)

$$\frac{\partial^2 L}{\partial P_{g_i}\partial\lambda_{p_i}}=\frac{\partial^2 L}{\partial\lambda_{p_i}\partial P_{g_i}}=-1$$ for $i$=1,2,3,…,NG (3.18)

$$\frac{\partial^2 L}{\partial P_{g_i}\partial\lambda_{p_k}}=\frac{\partial^2 L}{\partial\lambda_{p_k}\partial P_{g_i}}=0$$ for $i$=1,2,3,…,NG;k=1,2,3,…,NG but i≠k (3.19)

$$\frac{\partial^2 L}{\partial P_{g_i}\partial\lambda_{q_k}}=\frac{\partial^2 L}{\partial\lambda_{q_k}\partial P_{g_i}}=0$$ for $i$=1,2,3,…,NG;k=1,2,3,…,N (3.20)

$$\frac{\partial^2 L}{\partial P_{g_i}\partial|V_k|}=\frac{\partial^2 L}{\partial|V_k|\partial P_{g_i}}=0$$ for $i$=1,2,3,…,NG;k=1,2,3,…,N (3.21)

Similarly, second order partial derivation required for equation (3.10) are obtained by differentiating equation (3.5) with respect to control variables , and are as follow:

$$\frac{\partial L}{\partial \delta_i} = \left[ \lambda_{p_1} \frac{\partial P_1}{\partial \delta_i} + \lambda_{p_2} \frac{\partial P_2}{\partial \delta_i} + \ldots.. + \lambda_{p_N} \frac{\partial P_N}{\partial \delta_i} \right] + \left[ \lambda_{q_{NG+1}} \frac{\partial Q_{NG+1}}{\partial \delta_i} + \lambda_{q_{NG+2}} \frac{\partial Q_{NG+2}}{\partial \delta_i} + \ldots.. + \lambda_{q_N} \frac{\partial Q_N}{\partial \delta_i} \right]$$

Differentiating both sides with respect to $\delta_k$, we get

$$\frac{\partial^2 L}{\partial \delta_i \partial \delta_k} = \left[ \lambda_{p_1} \frac{\partial^2 P_1}{\partial \delta_i \partial \delta_k} + \lambda_{p_2} \frac{\partial^2 P_2}{\partial \delta_i \partial \delta_{ki}} + \ldots.. + \lambda_{p_N} \frac{\partial^2 P_N}{\partial \delta_i \partial \delta_k} \right] + \left[ \lambda_{q_{NG+1}} \frac{\partial^2 Q_{NG+1}}{\partial \delta_i \partial \delta_k} + \lambda_{q_{NG+2}} \frac{\partial^2 Q_{NG+2}}{\partial \delta_i \partial \delta_k} + \ldots.. + \lambda_{q_N} \frac{\partial^2 Q_N}{\partial \delta_i \partial \delta_k} \right]$$

$$\therefore \quad \frac{\partial^2 L}{\partial \delta_i \partial \delta_k} = \sum_{r=1}^{N} \lambda_{p_r} \frac{\partial^2 P_r}{\partial \delta_i \partial \delta_k} + \sum_{r=NG+1}^{N} \lambda_{q_r} \frac{\partial Q_r}{\partial \delta_i \partial \delta_k} \qquad \text{for } i=2,3,..,N;\ k=2,3,..,N \quad (3.22)$$

$$\frac{\partial^2 L}{\partial \delta_i \partial \lambda_{p_k}} = \frac{\partial P_k}{\partial \delta_i} \qquad \text{for } i=2,3,..,N;\ k=2,3,..,N \quad (3.23)$$

$$\frac{\partial^2 L}{\partial \delta_i \partial \lambda_{q_k}} = \frac{\partial Q_k}{\partial \delta_i} \qquad \text{for } i=2,3,..,N;\ k=NG+1,..,N \quad (3.24)$$

$$\frac{\partial^2 L}{\partial \delta_i \partial |V_k|} = \sum_{r=1}^{N} \lambda_{p_r} \frac{\partial^2 P_r}{\partial \delta_i \partial |V_k|} + \sum_{r=NG+1}^{N} \lambda_{q_r} \frac{\partial^2 Q_r}{\partial \delta_i \partial |V_k|} \qquad \text{for } i=2,3,..,N;\ k=NG+1,..,N \quad (3.25)$$

Next, second order partial derivatives required for equation (3.14) are obtained by differentiating equation (3.7) With respect to control variables:

$$\frac{\partial^2 L}{\partial \lambda_{p_i} \partial \delta_k} = \frac{\partial P_i}{\partial \delta_k} \qquad \text{for } i=2,3,..,N;\ k=2,3,..,N \quad (3.26)$$

$$\frac{\partial^2 L}{\partial \lambda_{p_i} \partial \lambda_{p_k}} = 0 \qquad \text{for } i=1,2,3,..,N;\ k=1,2,3,..,N \quad (3.27)$$

$$\frac{\partial^2 L}{\partial \lambda_{p_i} \partial \lambda_{q_k}} = 0 \qquad \text{for } i=1,2,3,..,N;\ k=NG+1,..N \quad (3.28)$$

$$\frac{\partial^2 L}{\partial \lambda_{p_i} \partial |V_k|} = \frac{\partial P_i}{\partial |V_k|} \qquad \text{for } i=1,2,3,..,N;\ k=NG+1,..N \quad (3.29)$$

Also, second order partial derivatives required for equation (3.13) are obtained be differentiating equation (3.8) with respect to control variables, and as follow:

$$\frac{\partial^2 L}{\partial |V_i| \partial \delta_k} = \sum_{r=1}^{N} \lambda_{p_r} \frac{\partial^2 P_r}{\partial |V_i| \partial \delta_k} + \sum_{r=NG+1}^{N} \lambda_{q_r} \frac{\partial^2 Q_r}{\partial |V_i| \partial \delta_k} \qquad \text{for } i=NG+1,..,N;\ k=2,3,..,N \quad (3.30)$$

$$\frac{\partial^2 L}{\partial |V_i| \partial \lambda_{p_k}} = \frac{\partial P_k}{\partial |V_i|}$$ for $i$=NG+1,..,N; k=1,2,3,..,N (3.31)

$$\frac{\partial^2 L}{\partial |V_i| \partial \lambda_{q_k}} = \frac{\partial Q_k}{\partial |V_i|}$$ for $i$=NG+1,..,N; k=NG+1,..,N (3.32)

$$\frac{\partial^2 L}{\partial |V_i| \partial |V_k|} = \sum_{r=1}^{N} \lambda_{p_r} \frac{\partial^2 P_r}{\partial |V_i| \partial |V_k|} + \sum_{r=NG+1}^{N} \lambda_{q_r} \frac{\partial^2 Q_r}{\partial |V_i| \partial |V_k|}$$

for $i$=NG+1,..,N; k=NG+1,..,N (3.33)

Second order partial derivatives required for equation (3.14) are obtained by differentiating equation (3.9) with respect to control variables and are as follow:

$$\frac{\partial^2 L}{\partial \lambda_{q_i} \partial \delta_k} = \frac{\partial Q_i}{\partial \delta_k}$$ for $i$=NG+1,..,N; k=2,3,..,N (3.34)

$$\frac{\partial^2 L}{\partial \lambda_{q_i} \partial \lambda_{p_k}} = 0$$ for $i$=NG+1,..,N; k=1,2,3,..,N (3.35)

$$\frac{\partial^2 L}{\partial \lambda_{q_i} \partial \lambda_{q_k}} = 0$$ for $i$=NG+1,..,N; k=NG+1,..,N (3.36)

$$\frac{\partial^2 L}{\partial \lambda_{q_k} \partial |V_k|} = \frac{\partial Q_k}{\partial |V_k|}$$ for $i$=NG+1,..,N; k=NG+1,..,N (3.37)

Equations (3.10) to (3.14) can be rewritten as:

$$\frac{\partial^2 L}{\partial P_{g_i}^2} \Delta P_{g_i} + \frac{\partial^2 L}{\partial P_{g_i} \partial \lambda_{p_i}} \Delta \lambda_{p_i} = -\frac{\partial L}{\partial P_{g_i}}$$ for $i$=1,2,3,…,NG (3.38)

$$\sum_{k=2}^{N} \frac{\partial^2 L}{\partial \delta_i \partial \delta_k} \Delta \delta_k + \sum_{k=1}^{N} \frac{\partial^2 L}{\partial \delta_i \partial \lambda_{p_k}} \Delta \lambda_{p_k} + \sum_{k=NG+1}^{N} \frac{\partial^2 L}{\partial \delta_i \partial \lambda_{q_k}} \Delta \lambda_{q_k} + \sum_{k=NG+1}^{N} \frac{\partial^2 L}{\partial \delta_i \partial |V_k|} \Delta |V_k| = -\frac{\partial L}{\partial \delta_i}$$

for $i$=1,2,3,…,NG (3.39)

$$\frac{\partial^2 L}{\partial \lambda_{pi} \partial P_{gi}} \Delta P_{gi} + \sum_{k=2}^{N} \frac{\partial^2 L}{\partial \lambda_{pi} \partial \delta_k} \Delta \delta_k + \sum_{k=NG+1}^{N} \frac{\partial^2 L}{\partial \lambda_{pi} \partial |V_k|} \Delta |V_k| = -\frac{\partial L}{\partial \lambda_{pi}}$$

for $i$=1,2,3,…,NG (3.40)

$$\sum_{k=2}^{N} \frac{\partial^2 L}{\partial |V_i| \partial \delta_k} \Delta \delta_k + \sum_{k=1}^{N} \frac{\partial^2 L}{\partial |V_i| \partial \lambda_{p_k}} \Delta \lambda_{p_k} + \sum_{k=NG+1}^{N} \frac{\partial^2 L}{\partial |V_i| \partial \lambda_{q_k}} \Delta \lambda_{q_k} + \sum_{k=NG+1}^{N} \frac{\partial^2 L}{\partial |V_i| \partial |V_k|} \Delta |V_k| = -\frac{\partial L}{\partial |V_i|}$$

for $i$=NG+1,…,NG (3.41)

$$\sum_{k=2}^{N}\frac{\partial^2 L}{\partial\lambda_{q_i}\partial\delta_k}\Delta\delta_k+\sum_{k=NG+1}^{N}\frac{\partial^2 L}{\partial\lambda_{q_i}\partial|V_k|}\Delta|V_k|=-\frac{\partial L}{\partial\lambda_{q_i}} \qquad \text{for } i=NG+1,\ldots,NG$$

(3.42)

Equation (3.10) to(3.14) can be written as follow:

$$\begin{bmatrix}
\frac{\partial^2 L}{\partial P_{g_i}\partial P_{g_k}} & 0 & \frac{\partial^2 L}{\partial P_{g_i}\partial\lambda_{p_k}} & 0 & 0\\
0 & \frac{\partial^2 L}{\partial\delta_i\partial\delta_k} & \frac{\partial^2 L}{\partial\delta_i\partial\lambda_{p_k}} & \frac{\partial^2 L}{\partial\delta_i\partial\lambda_{q_k}} & \frac{\partial^2 L}{\partial\delta_i\partial|V_k|}\\
\frac{\partial^2 L}{\partial\lambda_{p_i}\partial P_{g_k}} & \frac{\partial^2 L}{\partial\lambda_{p_i}\partial\delta_k} & 0 & 0 & \frac{\partial^2 L}{\partial\lambda_{p_i}\partial|V_k|}\\
0 & \frac{\partial^2 L}{\partial|V_i|\partial\delta_k} & \frac{\partial^2 L}{\partial|V_i|\partial\lambda_{p_k}} & \frac{\partial^2 L}{\partial|V_i|\partial\lambda_{q_k}} & \frac{\partial^2 L}{\partial|V_i|\partial|V_k|}\\
0 & \frac{\partial^2 L}{\partial\lambda_{q_i}\partial\delta_k} & 0 & 0 & \frac{\partial^2 L}{\partial\lambda_{q_i}\partial|V_k|}
\end{bmatrix}
\begin{bmatrix}\Delta P_{g_i}\\ \Delta\delta_i\\ \Delta\lambda_{p_i}\\ \Delta\lambda_{q_i}\\ \Delta|V_i|\end{bmatrix}
=
\begin{bmatrix}-\frac{\partial L}{\partial P_{g_i}}\\ -\frac{\partial L}{\partial\delta_i}\\ -\frac{\partial L}{\partial\lambda_{p_i}}\\ -\frac{\partial L}{\partial|V_i|}\\ -\frac{\partial L}{\partial\lambda_{q_i}}\end{bmatrix}$$

Or,

$$\begin{bmatrix}
H_{P_g P_g} & 0 & H_{P_g\lambda_P} & 0 & 0\\
0 & H_{\delta\delta} & H_{\delta\lambda_P} & H_{\delta\lambda_q} & H_{\delta|V|}\\
H_{\lambda_p P_g} & H_{\lambda_p\delta} & 0 & 0 & H_{\lambda_p|V|}\\
0 & H_{|V|\delta} & H_{|V|\lambda_P} & H_{|V|\lambda_q} & H_{|V||V|}\\
0 & H_{\lambda_P\delta} & 0 & 0 & H_{\lambda_P|V|}
\end{bmatrix}
\begin{bmatrix}\Delta P_{g_i}\\ \Delta\delta_i\\ \Delta\lambda_{p_i}\\ \Delta\lambda_{q_i}\\ \Delta|V_i|\end{bmatrix}
=
\begin{bmatrix}J_{P_{g_i}}\\ J_{\delta_i}\\ J_{\lambda_{p_i}}\\ J_{|V_i|}\\ J_{\lambda_{q_i}}\end{bmatrix} \qquad (3.43)$$

Where H & J are called *Hesseian* and *Jacobian Matrices* , respectively.

The flow-chart for solution of optimal power flow problem using *Newton – Raphson* method is shown in Figure 3.1

Start

Read bus data of the system and form [$Y_{bus}$] matrix

Read ,

1. Tolerance allowed, ε and maximum number of iteration allowed, *KPMAX*
2. Total No. of generators, NG, total Number of buses N
3. $\alpha_i$ , $\beta_i$, $\gamma_i$, for each generator, i.e for i=1,2,3,…,NG

Read the following for each bus (i.e. for i=1,2,3,…,NG

Initial bus voltage magnitude, initial load angel; Real Load; Reactive Load

Total real load $P_{load} = \sum_{i=1}^{N} P_{load_i}$

Separate real and imaginary parts of the admittance matrix Y

$G_{ij} = \mathrm{Re}\,al(Y_{ij}), B_{ij} = \mathrm{Im}(Y_{ij})$ for i=1,2,3,…,N & j=1,2,3,..,N

Fix generator bus voltage magnitude and angle for KPMAX number of iteration,

$V_i^j = V_i^0, and\ \delta_i^j = \delta_i^0$ for i=1,2,3,…,NG & j=1,2,3,..,KPMAX

Compute initial of $\lambda_{pi}, \lambda_{pi}{}^0 = \left[ P_{Load_T} + \left( \sum_{i=1}^{NG} \frac{\beta_i}{2\alpha_i} \right) \Big/ \left( \sum_{i=1}^{NG} \frac{1}{2\alpha_i} \right)^{-1} \right]$ for all generators

A

A

Compute initial real power generation $P_{gi}^{0} = (\lambda^{0} - \beta_i) / 2\alpha_i$ for i=1,2,..,NG

Set initial values of the following:

1. $\lambda_{qi}^{0} = 0$ for all generators i=1,2,3,...,NG
2. $V_i^{0} = 0$ and $\delta_i^{j} = 0$ for all load buses generators i=(NG+ 1),..,N

Set initial real and reactive power generation at all load buses zero

Set iteration kp=0

B

Compute the elements of Jacobian matrix [j]

Compute the element of Hessian matrix [H]

Solve (3.5) to obtain $\Delta P_{gi}^{kp}, \Delta\delta_i^{kp}, \Delta\lambda_{pi}^{kp}, \Delta\lambda_{qi}^{kp}, \Delta|V_i^{kp}|$

Update the control variables:

$$P_{gi}^{kp+1} = P_{gi}^{kp} + \Delta P_{gi}^{kp}$$
$$\delta_i^{kp+1} = \delta_i^{kp} + \Delta\delta_i^{kp}$$
$$\lambda_{pi}^{kp+1} = \lambda_{pi}^{kp} + \Delta\lambda_{pi}^{kp}$$
$$\lambda_{qi}^{kp+1} = \lambda_{qi}^{kp} + \Delta\lambda_{qi}^{kp},$$
$$|V_i^{kp+1}| = |V_i^{kp}| + \Delta|V_i^{kp}|$$

C

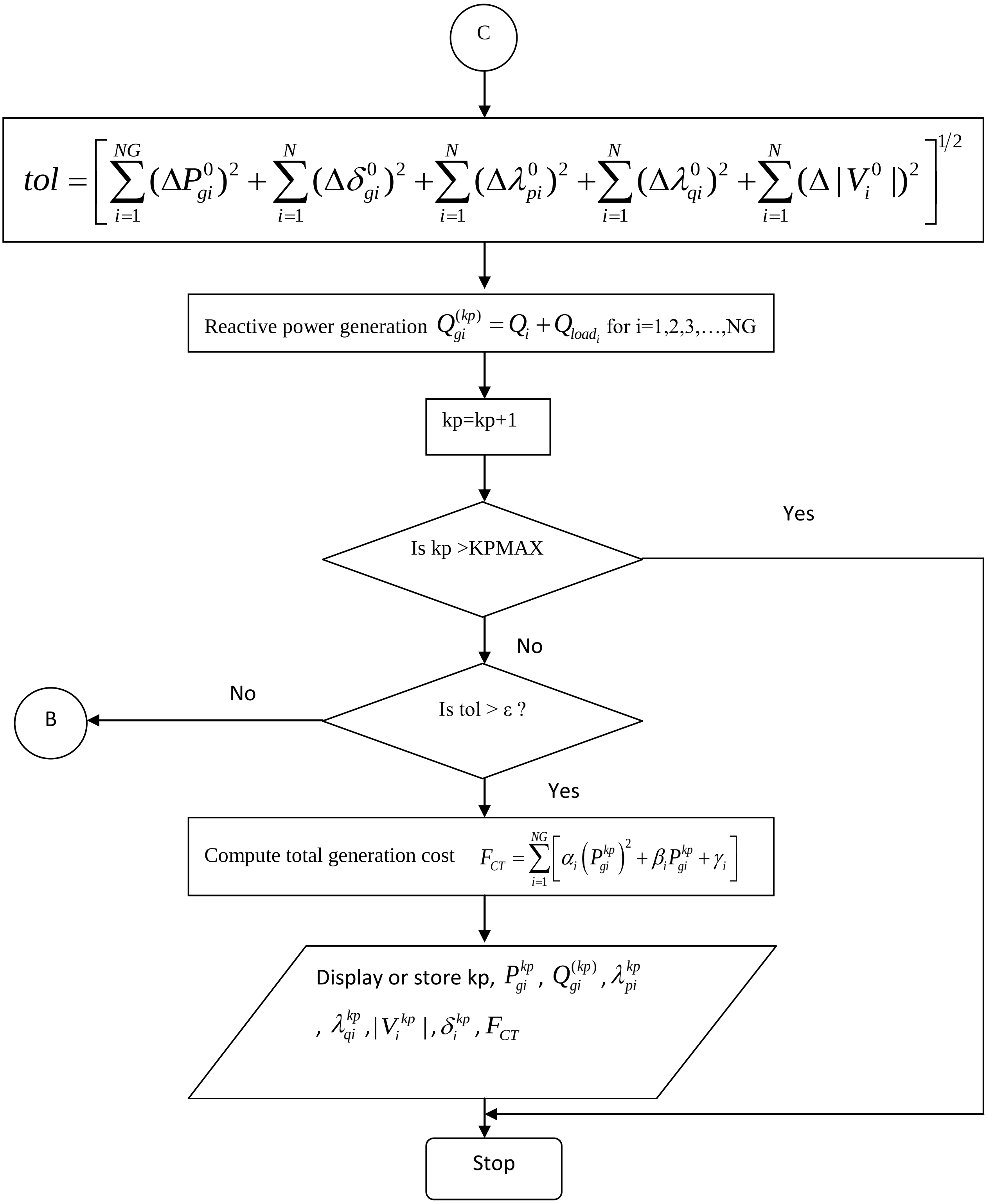


Figure 3.1 Flow-chart To Find Optimal Power Flow Solution Using Newton Raphson Method.

## 3.3 Intelligent Methods

To overcome the limitations and deficiencies in analytical methods, ***intelligent methods*** based on ***Artificial Intelligence*** (AI) techniques have been developed in the recent past. These methods can be classified or divided into the following,

a) Artificial Neural Networks (ANN)
b) Genetic Algorithms (GA)
c) Particle Swarm Optimization (PSO)
d) Ant Colony Algorithm

The major advantage of the intelligent methods is that they are relatively versatile for handling various qualitative constraints. These methods can find multiple optimal solutions in single simulation run. So they are quite suitable in solving multi objective optimization problems. In most cases, they can find the global optimum solution. The main advantages of intelligent methods are: Possesses learning ability, fast, appropriate for non-linear modeling, etc. whereas, large dimensionality and the choice of training methodology are some disadvantages of intelligent methods.

For the intelligent method *Particle Swarm Optimization* was considered to find the optimal power flow for IEEE 39 New England test system for comparison with the conventional method.

### 3.3.1 Particle Swarm Optimization:

Particle swarm optimization (PSO) is a population based evolutionary computation technique inspired from the social behaviors of bird flocking or fish schooling. Since its invention in 1995 by Kennedy and Eberhart, PSO has become one of the most popular methods applied to various optimization problems due to its simplicity and capability to find near optimal solutions. In conventional PSO, a population of particles moves in the search space of a problem to approach the global optimum. Figure 3.2 and Figure 3.4 shows the nature of PSO method:

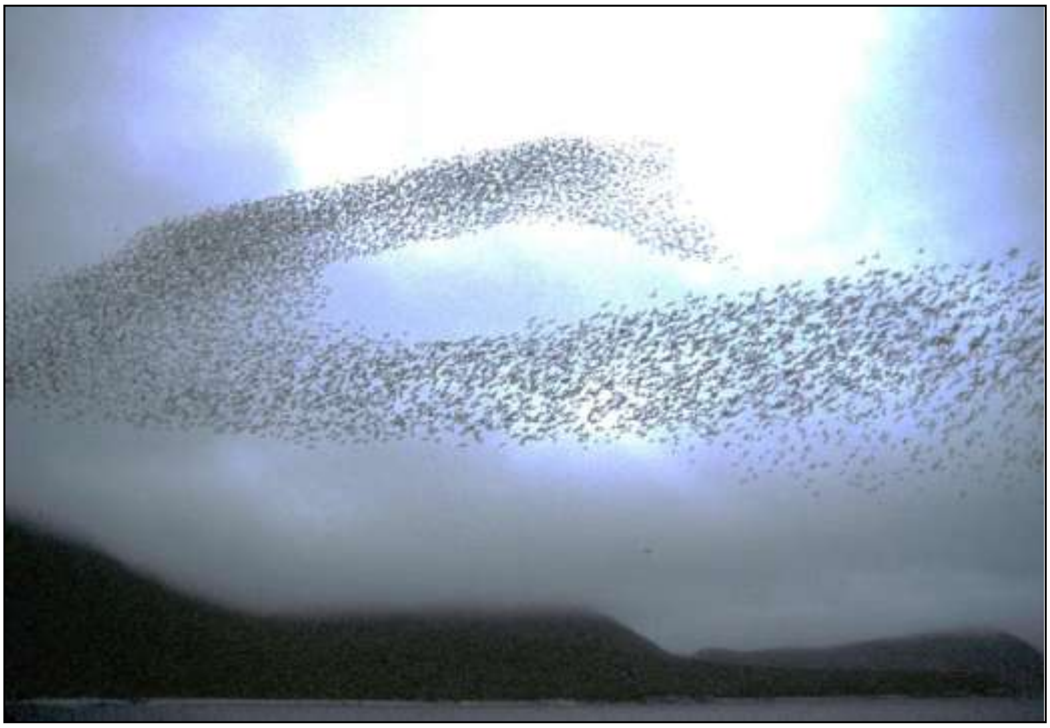

Figure 3.2: Example on the flock of bird in nature

Figure 3.3: example of school of fish in nature

The movement of each particle in the population is determined via its location and velocity. During the movement, the velocity of each particle is changed over time and its position is updated accordingly [4].

Consider an *n*-dimensional optimization problem:

Min $f(p)$

where $p = [p1, p2, \ldots, pn]$ is a vector of variables.

For implementation to the problem, the position and velocity vectors of particle $d$ are represented by

$p_d = [p_{1d}, p_{2d}, \ldots, p_{nd}]$ and

$v_d = [v_{1d}, v_{2d}, \ldots, v_{nd}]$, respectively,

Where $d = 1,\ldots, NP$ and $NP$ is the number of particles.

The best previous position of particle $d$ is based on the valuation of the fitness function represented by $pbest_d = [p_{1d}, p_{2d}, \ldots, p_{nd}]$ and the best particle among all particles represented by *gbest*. The velocity and position of each particle in the next iteration ($k$+1) for fitness function evaluation are calculated as follows:

$$v_{id}^{(k+1)} = w^{(k+1)} * v_{id}^{k} + C_1 * rand_1 * (pbest_{id}^{(k)} - p_{id}^{(k)}) + C_2 * rand_2 * (gbest_i^{(k)} - p_{id}^{(k)}) \tag{3.6}$$

$$p_{id}^{(k+1)} = p_{id}^{(k)} + v_{id}^{(k+1)} \tag{3.7}$$

Where $w$ is the inertia weight factor, $C_1$ and $C_2$ are cognitive and social parameters, respectively, and $rand_1$ and $rand_2$ are random values in [0, 1].

In conventional PSO, the inertia weight factor and cognitive and social parameters are constants. Position and velocity of each particle have their own limits. Regarding position limits, the lower and upper bounds are defined by the limits of variables represented by the particle's position. However, the velocity limits for the particles can be defined by the user. Generally, the solution quality of PSO is sensitive to cognitive and social parameters and velocity limits for particles.

The inertia weight factor linearly declines from its maximum to the minimum value as the number of iterations increases from 0 to *ITmax*. The inertia weight factor at iteration $k$ is updated as follows:

$$w^{(k)} = w_{\max} - (w_{\max} - w_{\min}) \frac{k}{IT_{\max}} \tag{3.8}$$

Where $w_{max}$ and $w_{min}$ are maximum and minimum weight factor, respectively, and *ITmax* is the maximum number of iterations.

### *3.3.1.1 Application of PSO Method to Economic Load Dispatch:*

Steps of Implementation:

1. Initialize the Fitness Function, i.e. Total cost function from the individual cost function of the various generating stations.
2. Initialize the PSO parameters Population size, $C_1$, $C_2$, $W_{max}$, $W_{min}$, error gradient etc.
3. Input the Fuel cost Functions, MW limits of the generating stations along with the B-coefficient matrix and the total power demand.
4. at the first steps of the execution of the program a large No. (equal to the population size) of vectors of active power satisfying the MW limits are randomly allocated.
5. For each vector of active power the value of the fitness function is calculated. All values obtained in iteration are compared to obtain Pbest. At each iteration all values of the whole population till then are compared to obtain the Gbest. At each step these values are updated.
6. At each step error gradient is checked and the value of Gbest is plotted till it comes within the pre-specified range.
7. This final value of Gbest is the minimum cost and the active power vector represents the economic load dispatch solution.

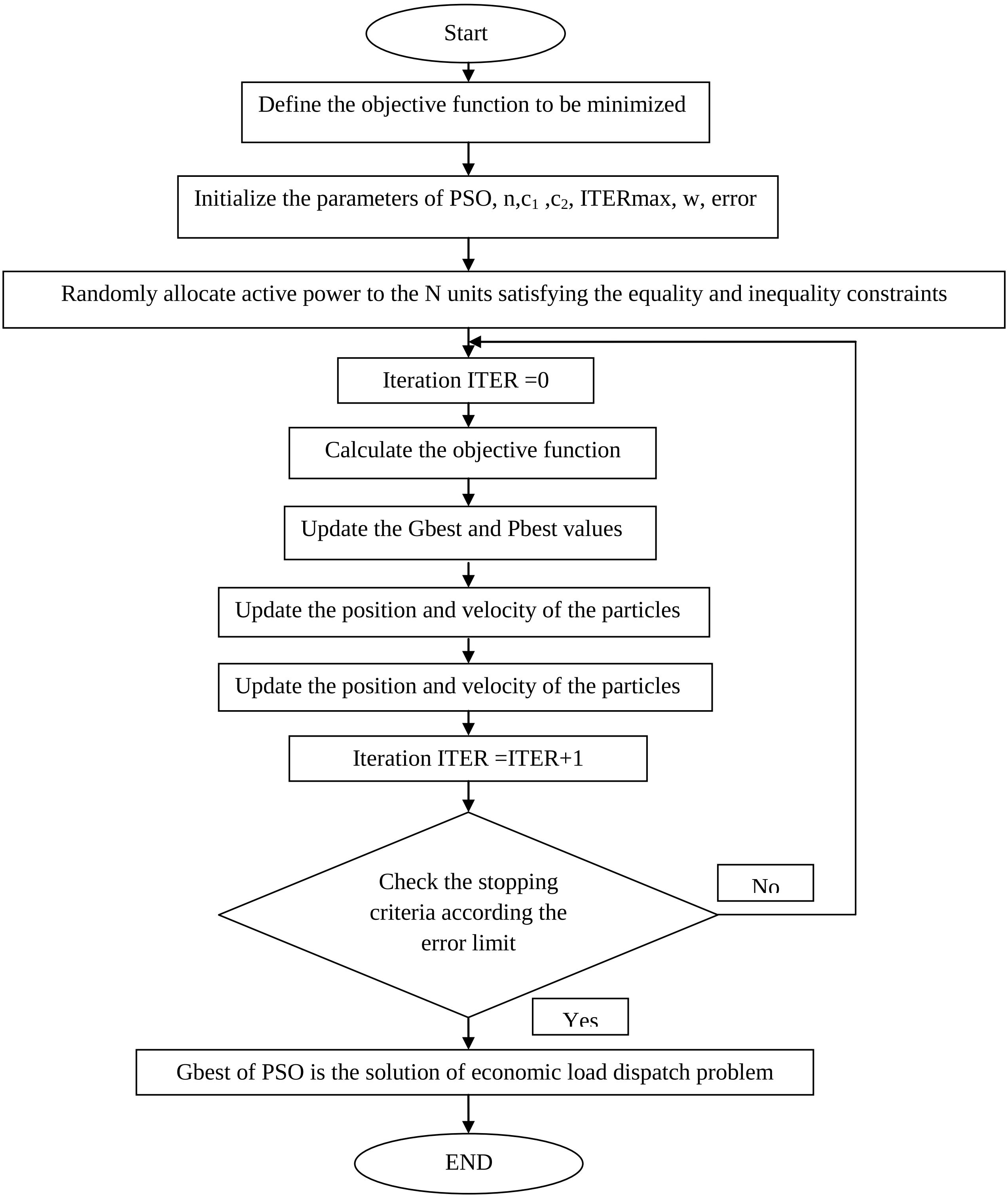


Figure 3.4 Flow-chart To Find Optimal Power Flow Solution Using Particle Swarm Optimization Method.

### *3.3.1.2 The Advantages and Disadvantages of Using PSO*

- ***Advantages***

  1. PSO is easy to implement the coding.
  2. PSO is able to produce high quality solutions by using less time.
  3. PSO is less sensitive to the objective function compared to conventional mathematical methods.
  4. PSO has less negative impact toward the solutions.
  5. PSO is less divergence.
  6. PSO has less parameter to control.

- ***Disadvantages***

  1. PSO need a longer computation time compared to the mathematical methods.
  2. PSO need more iteration than the classical method.