MODULE – 3

LOAD FLOW ANALYSIS (continued)

SYLLABUS

Load Flow Analysis (continued): Newton-Raphson Method Derivation in Polar Form, Fast Decoupled Load Flow Method, Flow Charts of LFS Methods, Comparison of Load Flow Methods, Illustrative Examples.

3.1 Newton-Raphson (RP) Method Derivation in Polar Form

- The Newton–Raphson (NR) method is a powerful method of solving non-linear algebraic equations.
- It works faster, and is sure to converge in most cases as compared to the Gauss–Siedel (GS) method. (For its convergence properties see Appendix I).
- It is indeed the practical method of load flow solution of large power networks. Its only drawback is the large requirement of computer memory, which can be overcome through a compact storage scheme. Convergence can be considerably speeded up by performing the first iteration through the GS method, and using the values so obtained for solving the NR iterations.
- We consider a set of n nonlinear algebraic equations

$$\boldsymbol{f_i(x_1, x_2, \dots, x_n) = 0 \qquad Where\ i = 1, 2, \dots, n\ ---------- (1)}$$

We assume initial values of unknowns as $x_1^0, x_2^0, \dots, x_n^0$. Let $\Delta x_1^0, \Delta x_2^0, \dots, \Delta x_n^0$ be the corrections to be found out, which on being added to the initial values, give the actual solution. Therefore,

$$\boldsymbol{f_i\left(x_1^0, x_2^0, \dots, x_n^0, \Delta x_n^0\right) = 0 \qquad Where\ i = 1, 2, \dots, n\ ---------- (2)}$$

Expanding these equations around the initial values by Taylor series, we have

$$\boldsymbol{f_i^0\left(x_1^0, x_2^0, \dots, x_n^0\right) + \left[\left(\frac{\partial f_i}{\partial x_1}\right)^0 \Delta x_1^0 + \cdots + \left(\frac{\partial f_i}{\partial x_n}\right)^0 \Delta x_n^0\right] + higher\ order\ terms = 0 ----(3)}$$

Where $\left(\frac{\partial f_i}{\partial x_1}\right)^0, \dots, \left(\frac{\partial f_i}{\partial x_n}\right)^0$ are the derivatives of f_i w.r.t $x_1, x_2, \dots, x_n$ evaluated at $x_1^1, \dots, x_n^0$

Neglecting the higher order terms, Eqn. (3) can be written in matrix form as

$$\begin{bmatrix} f_1^0 \\ \vdots \\ f_n^0 \end{bmatrix} + \begin{bmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)^0 & \cdots & \left(\frac{\partial f_1}{\partial x_n}\right)^0 \\ \vdots & & \vdots \\ \left(\frac{\partial f_n}{\partial x_1}\right)^0 & \cdots & \left(\frac{\partial f_n}{\partial x_n}\right)^0 \end{bmatrix} \begin{bmatrix} \Delta x_1^0 \\ \vdots \\ \Delta x_n^0 \end{bmatrix} \cong \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} --------(4)$$

Or in vector matrix form

$$\boldsymbol{f^0 + J^0 \Delta x^0 = 0 ----- (5)}$$

Where J^0 is the Jacobian matrix evaluated at x^0.

In compact notation,

$$\boldsymbol{J^0} = \left[\frac{\partial \boldsymbol{f(x)}}{\partial \boldsymbol{x}}\right]^{\boldsymbol{0}} ----(\boldsymbol{6})$$

In Eqn. (6), Δx^0 is the vector of approximate correction. This can be written in the form

$$\Delta \boldsymbol{x^0} = (-\boldsymbol{J^0})^{-1}\, \boldsymbol{f^0} \quad ------(7)$$

Thus, Δx^0 can be evaluated by calculating inverse of J^0. But, in practice, we do not evaluate the inverse matrix, as it is computationally expensive and not really needed. Also, the inverse has to be found for every iteration. We write Eqn. (6) in the form

$$\boldsymbol{J^0}\, \Delta \boldsymbol{x^0} \cong -\boldsymbol{f^0} ----(\boldsymbol{8})$$

These, being a set of linear algebraic equations, can be solved for Dx0 efficiently by triangularisation and back substitution. Updated values of x are then

$$\boldsymbol{x^1} = \boldsymbol{x^0} + \Delta \boldsymbol{x^0}$$

In general, for the (r + 1)th iteration

$$[\boldsymbol{J}(\boldsymbol{x^r})]\, \Delta \boldsymbol{x^r} = -\boldsymbol{f}(\boldsymbol{x^r}) \quad \boldsymbol{or} \quad [-\boldsymbol{J}(\boldsymbol{x^r})]\, \Delta \boldsymbol{x^r} = \boldsymbol{f}(\boldsymbol{x^r}) ------(\boldsymbol{9})$$
$$\boldsymbol{or} \quad (-\boldsymbol{J^r})\;\; \Delta \boldsymbol{x^r} = \boldsymbol{f^r}$$

and

$$\boldsymbol{x^{(r+1)}} = \boldsymbol{x^r} + \Delta \boldsymbol{x^r} \quad --------(\boldsymbol{10})$$

Iterations are continued till Eqn. (1) is satisfied to any desired accuracy, i.e.,

$$|\boldsymbol{f_i}\,(\boldsymbol{x^r})| < \;\boldsymbol{\varepsilon}\;\;(\boldsymbol{a\ specified\ value}), \quad \boldsymbol{i = 1, 2, \ldots, n}$$

Thus, each iteration involves the evaluation of $\boldsymbol{f}(\boldsymbol{x^r})$, $\boldsymbol{J}(\boldsymbol{x^r})$ and the correction $\Delta \boldsymbol{x^r}$**.** Therefore, time taken for each iteration by NR method is more compared to the GS method, but the method converges in only a few iterations and the total computation time is much less than by the Gauss Siedel (GS) method.

3.1.1 NR Algorithm for Load Flow Solution

We first consider the presence of PQ buses only apart from a slack bus. From Eqn. (6.28), for an ith bus,

$$P_i = \sum_{k=1}^{n} |V_i||V_k||Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i) = P_i(|V|, \boldsymbol{\delta})$$

------ (11.a)

$$Q_i = -\sum_{k=1}^{n} |V_i||V_k||Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i) = Q_i(|V|, \boldsymbol{\delta})$$

------------- (11.b)

i.e. both real and reactive powers are functions of (| V |, δ),

Where $\quad |\boldsymbol{V}| = (|V_1|, \ldots, |V_n|^T)\ \boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)^T$

We write

$$P_i(|V|, \boldsymbol{\delta}) = P_i(\boldsymbol{x})$$

$$Q_i(|V|, \boldsymbol{\delta}) = Q_i(\boldsymbol{x})$$

Where $\boldsymbol{x} = \begin{bmatrix} \delta \\ |V| \end{bmatrix}$

Let P*i* (scheduled) and Q*i* (scheduled) be the scheduled powers at the load buses. In the course of iteration x should tend to that value which makes

$$\mathbf{P}_i - \mathbf{P}_i(x) = 0 \text{ and } \mathbf{Q}_i - \mathbf{Q}_i(x) = 0 \quad \text{---------- (12.a)}$$

Writing Eq. (6.60a) for all load buses, we get its matrix form

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{P}(\text{scheduled}) - \boldsymbol{P}(\boldsymbol{x}) \\ \boldsymbol{Q}(\text{scheduled}) - \boldsymbol{Q}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \Delta\boldsymbol{P}(\boldsymbol{x}) \\ \Delta\boldsymbol{Q}(\boldsymbol{x}) \end{bmatrix} \cong 0 \quad \text{------- (12.b)}$$

At the slack bus (bus number 1), P_1 and Q_1 are unspecified. Therefore, the values $P_1(x)$ and $Q_1(x)$ do not enter into Eqn. (12.a), and hence (12.b). Thus, x is a 2(n – 1) vector (n – 1 load buses), with each element function of (n – 1)

Variables given by the vector, $\boldsymbol{x} = \begin{bmatrix} \delta \\ |V| \end{bmatrix}$

From Eqn. (9), we can write

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \Delta\boldsymbol{P}(\boldsymbol{x}) \\ \Delta\boldsymbol{Q}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} -\boldsymbol{J}_{11}(\boldsymbol{x}) & -\boldsymbol{J}_{12}(\boldsymbol{x}) \\ -\boldsymbol{J}_{21}(\boldsymbol{x}) & -\boldsymbol{J}_{22}(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta|V| \end{bmatrix} \quad \text{------- (13.a)}$$

Where $\Delta\delta = (\Delta\delta_2, \ldots, \Delta\delta_n)^T$

$$\Delta|V| = (\Delta|V_2|, \ldots, \Delta|V_n|)^T$$

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{bmatrix} -\boldsymbol{J}_{11}(\boldsymbol{x}) & -\boldsymbol{J}_{12}(\boldsymbol{x}) \\ -\boldsymbol{J}_{21}(\boldsymbol{x}) & -\boldsymbol{J}_{22}(\boldsymbol{x}) \end{bmatrix} \quad \text{------(13.b)}$$

$J(x)$ is the Jacobian matrix, Each J_{11}, J_{12}, J_{21}, J_{22} are (n – 1) X (n – 1) matrices. It follows from Eqn. (4) and Eqn. (12.b) that

$$-\boldsymbol{J}_{11}(\boldsymbol{x}) = \frac{\partial \boldsymbol{P}(\boldsymbol{x})}{\partial \delta}$$

$$-\boldsymbol{J}_{12}(\boldsymbol{x}) = \frac{\partial \boldsymbol{P}(\boldsymbol{x})}{\partial |V|} \quad \text{------------------ (14)}$$

$$-\boldsymbol{J}_{21}(\boldsymbol{x}) = \frac{\partial \boldsymbol{Q}(\boldsymbol{x})}{\partial \delta}$$

$$-\boldsymbol{J}_{22}(\boldsymbol{x}) = \frac{\partial \boldsymbol{Q}(\boldsymbol{x})}{\partial |V|}$$

The elements of $-\boldsymbol{J}_{11}, -\boldsymbol{J}_{12}, -\boldsymbol{J}_{21}, -\boldsymbol{J}_{22}$ are $\dfrac{\partial \boldsymbol{P}_i(\boldsymbol{x})}{\partial \delta_k}, \dfrac{\partial \boldsymbol{P}_i(\boldsymbol{x})}{\partial |V_k|}, \dfrac{\partial \boldsymbol{Q}_i(\boldsymbol{x})}{\partial \delta_k},$

$\dfrac{\partial \boldsymbol{Q}_i(\boldsymbol{x})}{\partial |V_k|}$, where $i = 2, .., n$; $k = 2, \ldots, n$.

From Eqns. 11(a) and (b) we have,

$$\frac{\partial \boldsymbol{P}_i(\boldsymbol{x})}{\partial \delta_k} = -|V_i||V_k||Y_{ik}|\sin(\theta_{ik} + \delta_k - \delta_i) \quad (i \neq k)$$

$$= \sum_{\substack{k=1 \\ k \neq i}}^{n} |V_i||V_k||Y_{ik}|\sin(\theta_{ik} + \delta_k - \delta_i)) \quad (i = k)$$

-------- (15.a)

$$\frac{\partial \boldsymbol{P}_i(\boldsymbol{x})}{\delta |V_k|} = |V_i||Y_{ik}|\cos(\theta_{ik} + \delta_k - \delta_i) \quad (i \neq k)$$

$$= 2|V_i||Y_{ii}|\cos\theta_{ii} + \sum_{\substack{k=1 \\ k \neq i}}^{n} (|V_k||Y_{ik}|\cos(\theta_{ik} + \delta_k - \delta_i)) \qquad (i = k)$$

------ (15.b)

$$\frac{\partial \boldsymbol{Q}_i(\boldsymbol{x})}{\partial \delta_k} = |V_i||Y_{ik}||V_k|\cos(\theta_{ik} + \delta_k - \delta_i) \quad (i \neq \mathrm{k})$$

$$= -\sum_{\substack{k=1 \\ k \neq i}}^{n} (|V_i||V_k||Y_{ik}|\cos(\theta_{ik} + \delta_k - \delta_i)) \quad (i = k)$$

------------ (15.c)

$$\frac{\partial \boldsymbol{Q}_i(\boldsymbol{x})}{\partial |V_k|} = |V_i||Y_{ik}|\sin(\theta_{ik} + \delta_k - \delta_i) \quad (i \neq k)$$

$$= 2|V_i||Y_{ii}|\sin\theta_{ii} + \sum_{\substack{k=1 \\ k \neq i}}^{n} (|V_k||Y_{ik}|\sin(\theta_{ik} + \delta_k - \delta_i)) \quad (i = k)$$

-------- (15.d)

An important observation can be made with respect to the elements of Jacobian matrix. If there is no connection between ith and kth bus, then $Y_{ik} = 0$, and from Eqn. (15), the elements of the Jacobian matrix corresponding to ith and kth buses are zero. Hence, like Y_{BUS} matrix, the Jacobian matrix is also sparse. There are computational techniques to take advantage of this sparsity in the computer solution.

The Jacobian matrix in Eqn. (13) is rearranged for obtaining the approximate correction vectors and can be written as:

$$\begin{array}{c} i\text{ th} \\ \text{PQ} \\ \text{bus} \end{array} \left\{ \begin{array}{|c|} \hline \\ \hline \Delta P_i \\ \hline \Delta Q_i \\ \hline \\ \hline \end{array} \right. = \begin{array}{c} i\text{ th} \\ \text{bus} \end{array} \begin{array}{c} m\text{th bus} \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & H_{im} & N_{im} & \\ \hline & J_{im} & L_{im} & \\ \hline & & & \\ \hline \end{array} \end{array} \begin{array}{|c|} \hline \\ \hline \Delta\delta_m \\ \hline \Delta|V_m| \\ \hline \\ \hline \end{array} \begin{array}{c} m\text{th} \\ \text{PQ} \\ \text{bus} \end{array}$$

-------- (16)

i = 2, ..., n and m = 2, ..., n, where

$$H_{im} = \frac{\partial P_i}{\partial \delta_m}$$

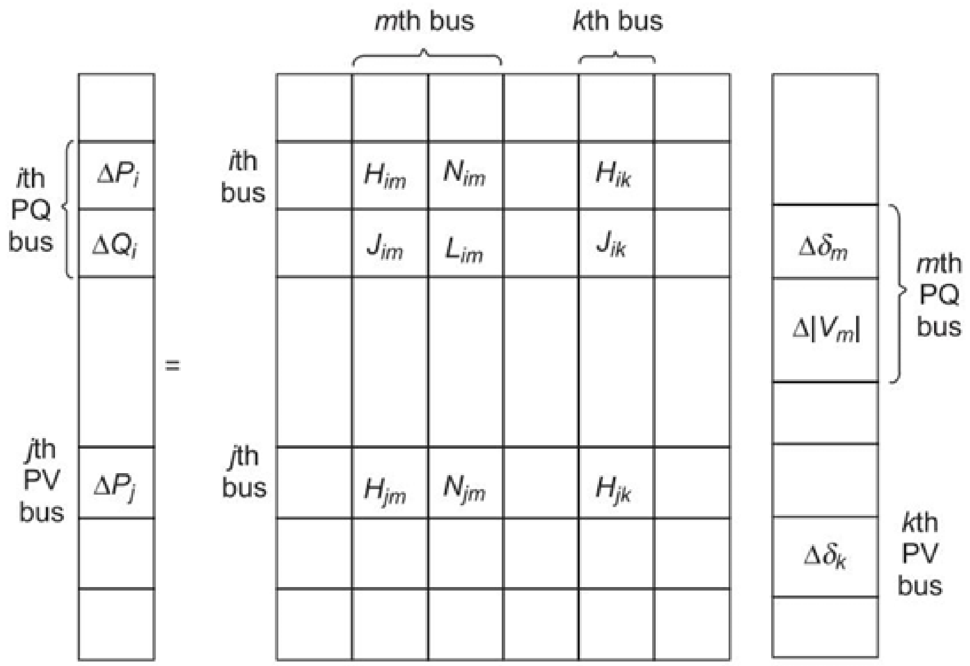
$$N_{im} = \frac{\partial P_i}{\partial |V_m|} \quad \text{-------- (16.a)}$$

$$J_{im} = \frac{\partial Q_i}{\partial \delta_m}$$

$$L_{im} = \frac{\partial Q_i}{\partial |V_m|} \quad \text{---------(16.b)}$$

It is to be immediately observed that the Jacobian elements corresponding to the ith bus residuals and mth bus corrections are a 2 X 2 matrix enclosed in the box in Eqn. (16), where i and m are both PQ buses.

Consider now the presence of PV buses. Since Qi is not specified and |Vi| is fixed for a PV bus, ΔQi does not enter on the LHS of Eqn. (16), and Δ|Vi| (= 0) does not enter on the RHS of Eqn. (16). Let ith and mth buses be PQ buses and jth and kth buses be PV buses. Then we have



-------- (17)

It is convenient for numerical solution to normalize the voltage corrections as (Δ|Vm| / |Vm|) , as a consequence of which, the corresponding Jacobian elements become

$$N_{im} = \frac{\partial P_i}{\partial |V_m|} \; |V_m|$$

$$L_{im} = \frac{\partial Q_i}{\partial |V_m|} \; |V_m| \quad \text{--------- (18)}$$

Expressions for elements of the Jacobian (in normalized form) of load flow Eqn. (12.b) are derived in Appendix D and are given below:

Case 1 $(m \neq i)$

$$H_{im} = L_{im} = a_m f_i - b_m e_i$$

$$N_{im} = -J_{im} = a_m e_i + b_m f_i \quad \text{----------- (19)}$$

where $Y_{im} = G_{im} + JB_{im}$

$$V_i = e_i + jf_i$$

$$a_m + jb_m = (G_{im} + jB_{im})(e_m + jf_m)$$

Case 2 $m = i$

$$H_{ii} = -Q_i - B_{ii} |V_i|^2$$

$$N_{ii} = P_i + G_{ii} |V_i|^2 \quad \text{----------- (20)}$$

$$J_{ii} = P_i - G_{ii} |V_i|^2$$

$$L_{ii} = Q_i - B_{ii} |V_i|^2$$

If buses *i* and *m* are not connected,
Hence from Eqn. (19) we can write

$$H_{im} = H_{mi} = 0$$

$$N_{im} = N_{mi} = 0$$

$$J_{im} = J_{mi} = 0$$

$$L_{im} = L_{mi} = 0 \quad \text{------------------ (21)}$$

Thus the Jacobian is as sparse as the YBUS matrix.

Formation of Eqn. (17) of the NR method is best illustrated by a problem. Figure 3.1 shows a five bus power network with bus types indicated therein. The matrix equation for determining vector of corrections from the vector of residuals follows very soon.

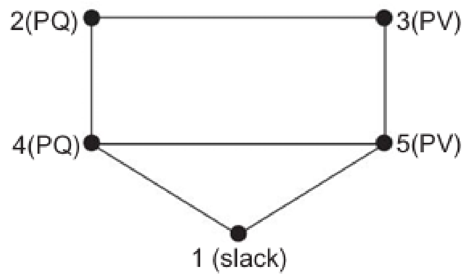


Fig. 3.1 Sample Five-Bus Network

Corresponding to a particular vector of variables (δ_2, $|V_2|$, δ_3, δ_4, ΔQ_2, $|V_4|$, δ_5)^T the vector of residuals is (ΔP_2, ΔQ_2, ΔP_3, ΔP_4, ΔQ_4, ΔP_5)^T and the Jacobian (6 X 6 in this example) are computed.

bus no. ↓ / bus no. →			2		3	4		5	
2	ΔP_2		H_{22}	N_{22}	H_{23}	H_{24}	N_{24}		$\Delta\delta_2$
	ΔQ_2		J_{22}	L_{22}	J_{23}	J_{24}	L_{24}		$\Delta\lvert V_2\rvert / \lvert V_2\rvert$
3	ΔP_3	=	H_{32}	N_{32}	H_{33}			H_{35}	$\Delta\delta_3$
4	ΔP_4		H_{42}	N_{42}		H_{44}	N_{44}	H_{45}	$\Delta\delta_4$
	ΔQ_4		J_{42}	L_{42}		J_{44}	L_{44}	J_{45}	$\Delta\lvert V_4\rvert / \lvert V_4\rvert$
5	ΔP_5				H_{53}	H_{54}	N_{54}	H_{55}	$\Delta\delta_5$

Residuals ↓ Jacobian (evaluated at trial values of variables) ↓ Corrections in variables ↓

-------- (22)

Equation (22) is then solved by triangularization and back substitution procedure to obtain the vector of corrections ($\Delta\delta_2$, $\Delta\ |V_2|\ /\ |V_2|$, $\Delta\delta_3$, $\Delta\delta_4$, $\Delta\ |V_4|\ /\ |V_4|$, $\Delta\delta_5$)^T. Corrections are then added to update the vector of variables.

3.1.2 Iterative Algorithm

The iterative algorithm for the solution of the load flow problem by the NR method is as follows:

1. With the voltage and angle at the slack bus fixed at $V_1\ \angle\delta_1$ (= 1 $\angle 0^o$), we assume $|V|$, $\angle\delta$ at all PQ buses and δ at all PV buses. In the absence of any information flat voltage start is recommended.

2. In the rth iteration, we have

$$P_i^r = \sum_{k=1}^{n} |V_i|^r\,|V_k|^r\,|Y_{ik}|\cos\left(\theta_{ik} + \delta_k^r - \delta_i^r\right) \quad \text{-------- (23.a)}$$

$$Q_i^r = -\sum_{k=1}^{n} |V_i|^r\,|V_k|^r\,|Y_{ik}|\sin\left(\theta_{ik} + \delta_k^r - \delta_i^r\right) \quad \text{-------- (23.b)}$$

$$\text{Let} \qquad e_i^r = |V_i|^r \cos\delta_i^r,\ f_i^r = |V_i|^r \sin\delta_i^r \quad \text{------- (24)}$$

$$G_{ik} = |Y_{ik}|\cos\theta_{ik},\ B_{ik} = |Y_{ik}|\sin\theta_{ik} \quad \text{-------(25)}$$

Using Eqns. (24) and (25) in Eqns. 23(a) and (b), we have

$$P_i^r = \sum_{k=1}^{n} \left(e_i^r \left(e_k^r G_{ik} - f_k^r B_{ik}\right) + f_i^r\left(f_k^r G_{ik} + e_k^r B_{ik}\right)\right)$$

----------- (26.a)

$$Q_i^r = \sum_{k=1}^{n} \left(f_i^r \left(e_k^r G_{ik} - f_k^r B_{ik}\right) - e_i^r\left(f_k^r G_{ik} + e_k^r B_{ik}\right)\right),\ i = 2, \ldots, n$$

----------- (26.b)

We next compute

$$\Delta P_i^r = P_i\ (\text{scheduled}) - P_i^r \quad \text{for PV and PQ buses}$$ **----------- (27.a)**

$$\Delta Q_i^r = Q_i\ (\text{scheduled}) - Q_i^r \quad \text{for PQ buses}$$ **------------ (27.b)**

If all values of ΔP_i^r and ΔQ_i^r are less than the prescribed tolerance, we stop the iteration, calculate P_1 and Q_1 and print the entire solution, including line flows.

3. If the convergence criterion is not satisfied, we evaluate the Jacobian elements using Eqns. (19) and (20).

4. We solve Eqn. (17) for correction of voltage magnitudes $\Delta|V|^r$ and angle $\Delta\delta^r$.

5. Next we update voltage magnitudes and angles,

$$|\boldsymbol{V}|^{(r+1)} = |\boldsymbol{V}|^r + |\Delta \boldsymbol{V}|^r$$

$$\boldsymbol{\delta}^{(r+1)} = \boldsymbol{\delta}^r + \Delta\boldsymbol{\delta}^r$$

Then we return to step 2.

It is to be noted that:

(a) In step 2, if there are limits on the controllable Q sources at PV buses, Q is computed each time using Eqn. (26.b), and if it violates the limits, it is made equal to the limiting value and the corresponding PV bus is made PQ bus in that iteration. If in subsequent computation, Q comes with the prescribed limits, the bus is switched back to a PV bus. Thus if $|Q_i|r \geq Q_{i\,max}$, then we let $|Q_i| = Q_{i\,max}$ and we have from Eqn. (17)

$$\Delta Q_i^r = Q_i - Q_i^r = \sum_{k=2}^{m} \left(J_{ik}^r \Delta\delta_k^r + L_{ik}^r \Delta |V_k|^r\right) + \sum_{k=m+1}^{n} J_{ik}^r \Delta\delta_k^r$$ **-------- (28)**

In the RHS of Eqn. (28), all Ddks and D |Vk|s are known except D | Vi |r ◊ Jiks and Liks are calculated from Eqns. (19) and (20). Hence, from Eqn. (28)

$$\Delta |V_i|^r = \frac{1}{L_{ii}^r}\left[\Delta Q_i^r - \sum_{\substack{k=2\\k\neq i}}^{n}\left(J_{ik}\Delta\delta_k^r + L_{ik}^r \Delta |V_k|^r\right) - \sum_{\substack{k=m+1\\k\neq i}}^{n}\left(J_{ik}^r\Delta\delta_k^r\right) - J_{ii}^r\Delta\delta_i^r\right]$$ -----(29)

After computing $\Delta\,|Vi|^r$, the new value of |Vi| (scheduled) for the PV bus (now a PQ bus due to violation of the limits on Q_i^r) is

$$|Vi| = |Vi|\ (\text{scheduled}) + \Delta\,|Vi|^r \text{ ------------- } (30)$$

With this scheduled value of |Vi|, the bus is restored to PV bus, and the next iteration is continued. The above procedure is repeated till Q_i^r lies within its limits.

(b) Similarly, if there are voltage limits on a PQ bus, and if any of these limits is violated, the corresponding PQ bus is made a PV bus in that iteration with voltage fixed at the limiting value.

3.2 Fast Decoupled Load Flow (FDLF) Method

The Jacobian of the decoupled Newton load flow can be made constant in value, based on physically justifiable assumptions. Hence the triangularization has to be done only once per solution.

The Fast Decoupled Load Flow (FDLF) was developed by B. Stott in 1974 in the process of further simplifications and assumptions.

The assumptions which are valid in normal power system operation are made as follows:

$$\boldsymbol{cos\delta_{ik} \cong 1 - - - - - (31)}$$

$$\boldsymbol{sin\delta_{ik} \cong 0 - - - - - (32)}$$

$$\boldsymbol{G_{ij}\ sin\delta_{ik} \ll B_{ik};\quad and \quad Q_i \ll B_{ii}\ |V_i|^2}$$

With these assumptions, the entries of the [H] and [L] submatrices become considerably simplified, and are given by

$$\boldsymbol{H_{ik} = L_{ik} = -|V_i|\ |V_k|\ B_{ik} \qquad i \neq k - - - - - - (33.a)}$$

$$\boldsymbol{H_{ii} = L_{ii} = -B_{ii}\ |V_i|^2 \qquad i = k - - - - - - (33.b)}$$

Matrices [H] and [L] are square matrices with dimensions (n – 1) and (m – 1), respectively, (m – 1 = number of PQ buses, and n – 1 = number of PQ and PV buses).

Equations 6.87(a and b)

$$\Delta P_i = \sum_{k=2}^{n} H_{ik}\,\Delta\delta_k \qquad i = 2, \ldots, n$$

$$\Delta Q_i = \sum_{k=2}^{m} L_{ik}\frac{\Delta|V_k|}{|V_k|} \quad i = 2, \ldots, n$$

can now be written as (after substituting the values of H_{ik}, L_{ik}, H_{ii}, L_{ii} from Eqns. 33(a and b)

$$\frac{\Delta P_i}{|V_i|} = -\sum_{k=2}^{n} |V_k| B_{ik} \Delta\delta_k \quad i = 2, \ldots, n$$

$$\frac{\Delta Q_i}{|V_i|} = -\sum_{k=2}^{n} B_{ik} \Delta |V_k| \quad i = 2, \ldots, m$$

-------- (34.a & b)

Setting $|V_k|$ = 1 pu on the R.H.S. of Eqn. (34.a), we get

$$\frac{\Delta P_i}{|V_i|} = \sum_{k=2}^{n} [-B_{ik}] \Delta\delta_k \quad i = 2, \ldots, n$$

$$\frac{\Delta Q_i}{|V_i|} = \sum_{k=2}^{n} [-B_{ik}] \Delta |V_k| \quad i = 2, \ldots, m$$

or in matrix form, after writing for all *i* 's, we get

$$\left[\frac{\Delta \boldsymbol{P}}{|\boldsymbol{V}|}\right]_{(n-1)\times 1} = [\boldsymbol{B}']_{(n-1)\times(n-1)} [\Delta\boldsymbol{\delta}]_{(n-1)\times 1}$$

$$\left[\frac{\Delta \boldsymbol{Q}}{|\boldsymbol{V}|}\right]_{(m-1)\times 1} = [\boldsymbol{B}'']_{(m-1)\times(m-1)} [\Delta |\boldsymbol{V}|]_{(m-1)\times 1}$$

-------- (35.a & b)

Where B' and B" are matrices of elements $-B_{ik}$ 297 (i = 2, ..., n and k = 2, ..., n) and $-B_{ik}$ (i = 2, ..., m and k = 2, ..., n).

Further simplification of the FDLF algorithm is achieved by:

1. Omitting the elements of [B'] that predominantly affect reactive power flows, i.e., shunt reactances and transformer off-nominal in-phase taps.
2. Omitting from [B"] the angle shifting effect of phase shifter (that which predominantly affects real power flow).
3. Ignoring the series resistance in calculating the elements of [B'], which then becomes the DC approximation of the power flow matrix.

After these simplifications Eqns. 35(a and b) are rewritten as

$$[\Delta \boldsymbol{P}/|\boldsymbol{V}|] = [\boldsymbol{B}'] [\Delta\boldsymbol{\delta}]$$

$$[\Delta \boldsymbol{Q}/|\boldsymbol{V}|] = [\boldsymbol{B}''] [\Delta |\boldsymbol{V}|]$$

-------- (36.a & b)

In Eqns. 36(a and b), both (B') and (B") are real, sparse and have the structures of [H] and [L], respectively. Since they contain only admittances, they are constant and need to be triangularized (or inverted) only once at the beginning of load flow analysis. If the phase shifters are not present, both [B'] and [B"] are always symmetrical, and their constant sparse upper triangular factors are calculated and stored only once at the beginning of the solution.

Equations 36(a and b) are solved alternatively, always employing the most recent voltage values. One iteration implies one solution for [Δd], to update [δ], and then one solution for [Δ |V|], to update [|V|] to be called 1–δ and 1 – V iteration. Separate convergence tests are applied for the real and reactive power mismatches as follows:

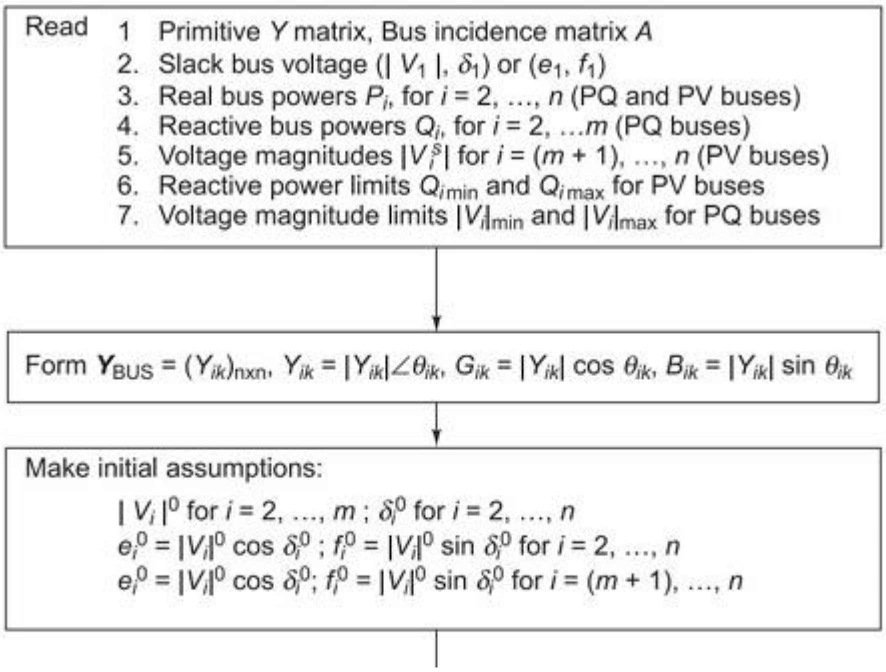
$$\max\,[\Delta \boldsymbol{P}] \le \varepsilon_P;\ \text{and}\ \max\,[\Delta \boldsymbol{Q}] \le \varepsilon_Q$$

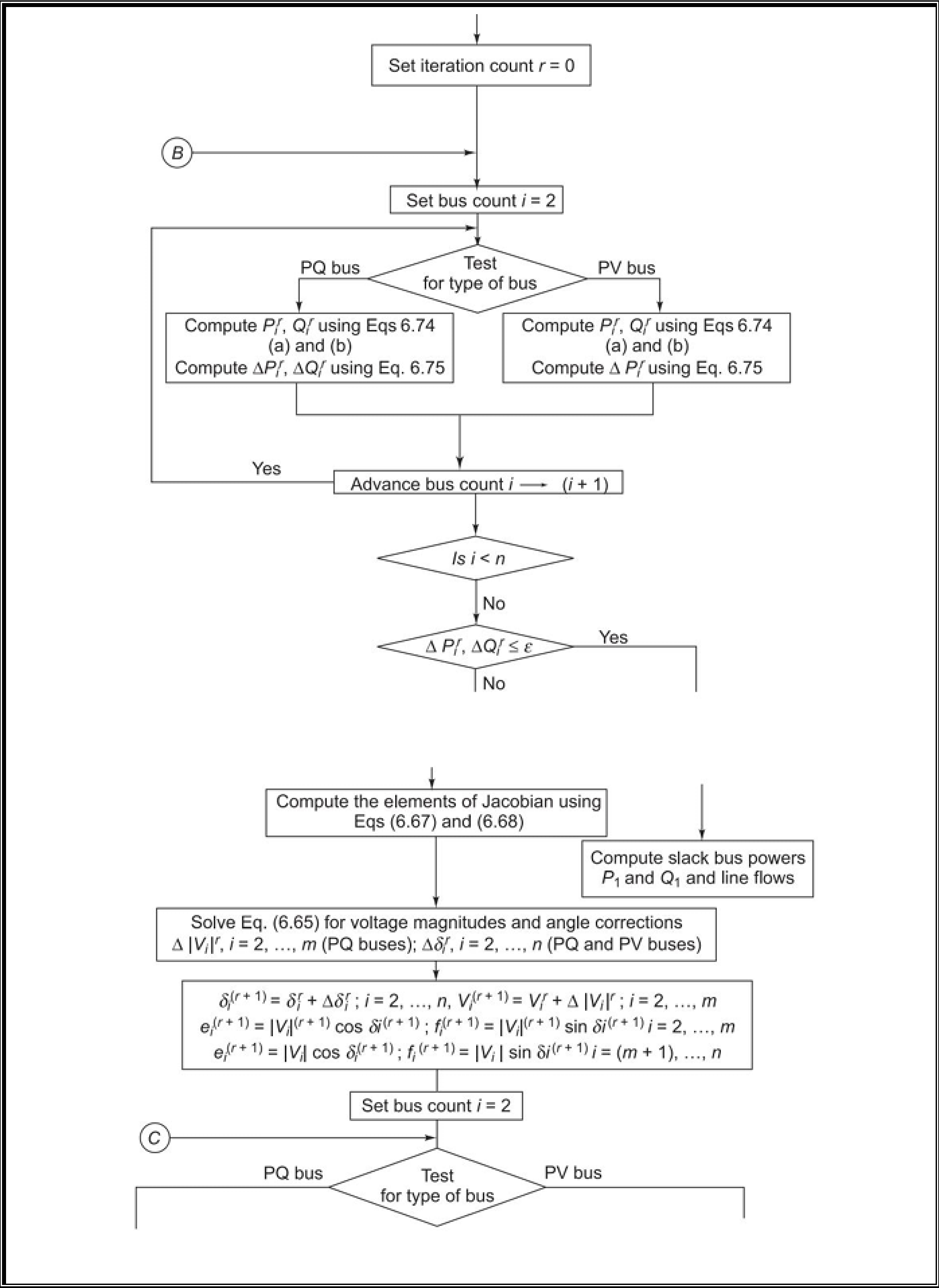
Where ε_P and ε_Q are the tolerances.

In FDLF method usually two to five iterations are required for practical accuracies. The method is more reliable than the formal NR method. The speed for iterations is about five times that of the formal NR method or about two thirds that of the GS method. Storage requirements are about 60 percent of the formal NR method, but slightly more than the DLF method.

3.3 Flow Charts of Load Flow Studies (LFS) Methods

The detailed flow chart for load flow solution by NR method is given in **Fig. 3.2**




Set iteration count $r = 0$
B
Set bus count $i = 2$
Test for type of bus
PQ bus
PV bus
Compute P_i^r, Q_i^r using Eqs 6.74 (a) and (b)
Compute ΔP_i^r, ΔQ_i^r using Eq. 6.75
Compute P_i^r, Q_i^r using Eqs 6.74 (a) and (b)
Compute ΔP_i^r using Eq. 6.75
Yes
Advance bus count $i \longrightarrow (i + 1)$
Is $i < n$
No
$\Delta P_i^r, \Delta Q_i^r \le \varepsilon$
Yes
No
Compute the elements of Jacobian using Eqs (6.67) and (6.68)
Compute slack bus powers P_1 and Q_1 and line flows
Solve Eq. (6.65) for voltage magnitudes and angle corrections $\Delta |V_i|^r$, $i = 2, \ldots, m$ (PQ buses); $\Delta\delta_i^r$, $i = 2, \ldots, n$ (PQ and PV buses)
$\delta_i^{(r+1)} = \delta_i^r + \Delta\delta_i^r$; $i = 2, \ldots, n$, $V_i^{(r+1)} = V_i^r + \Delta |V_i|^r$; $i = 2, \ldots, m$
$e_i^{(r+1)} = |V_i|^{(r+1)} \cos \delta i^{(r+1)}$; $f_i^{(r+1)} = |V_i|^{(r+1)} \sin \delta i^{(r+1)}$ $i = 2, \ldots, m$
$e_i^{(r+1)} = |V_i| \cos \delta_i^{(r+1)}$; $f_i^{(r+1)} = |V_i| \sin \delta i^{(r+1)}$ $i = (m + 1), \ldots, n$
Set bus count $i = 2$
C
Test for type of bus
PQ bus
PV bus

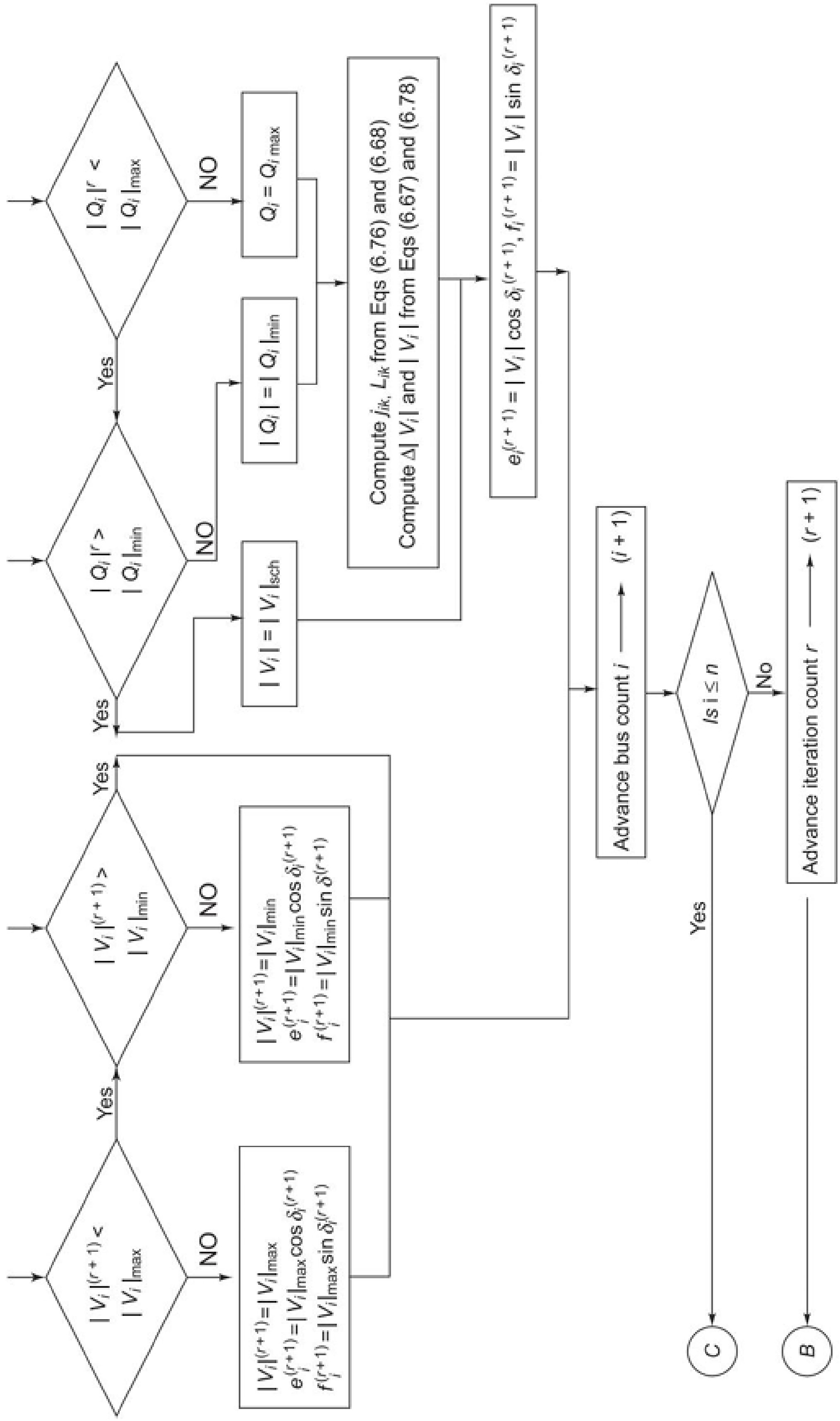
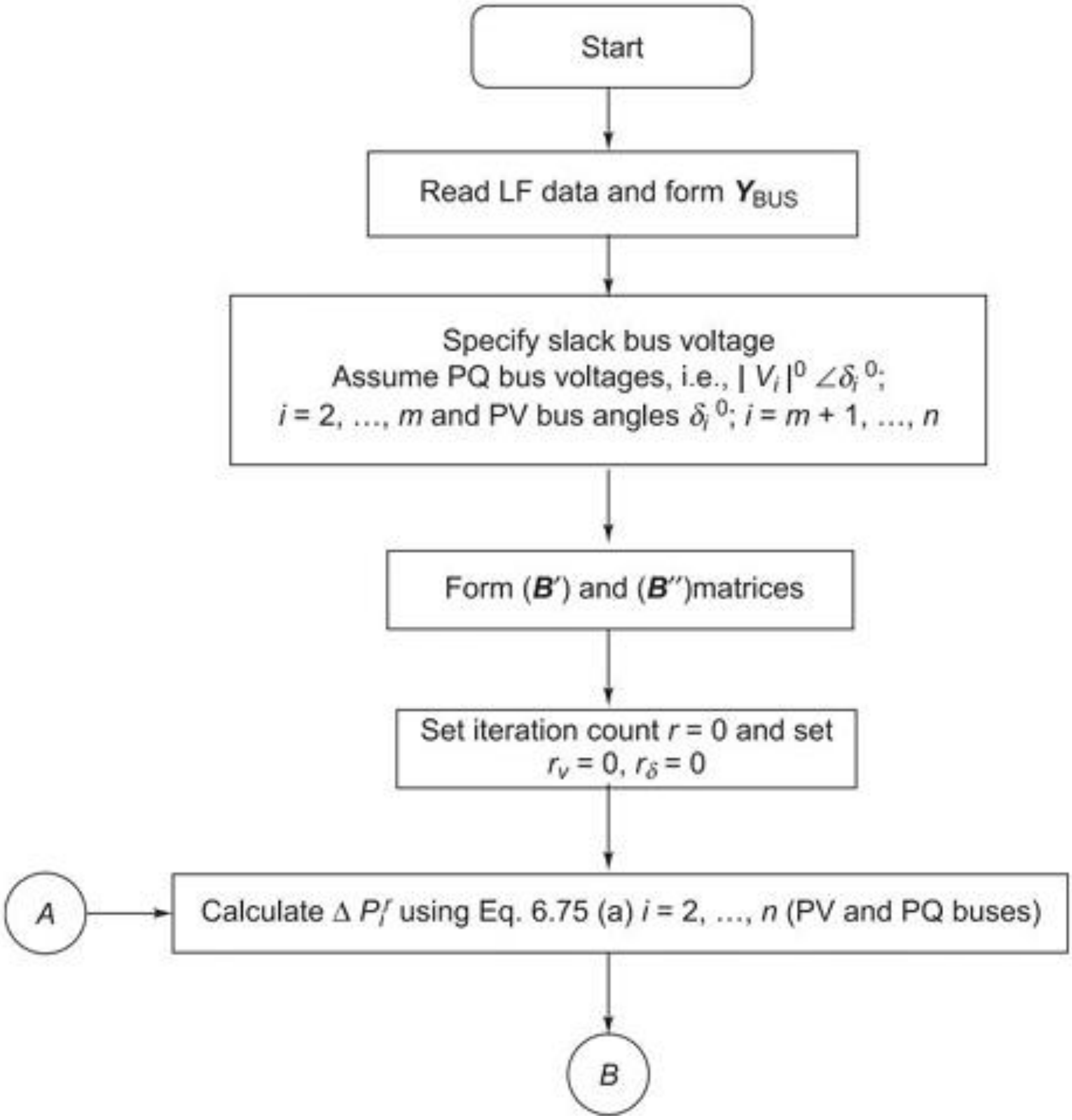


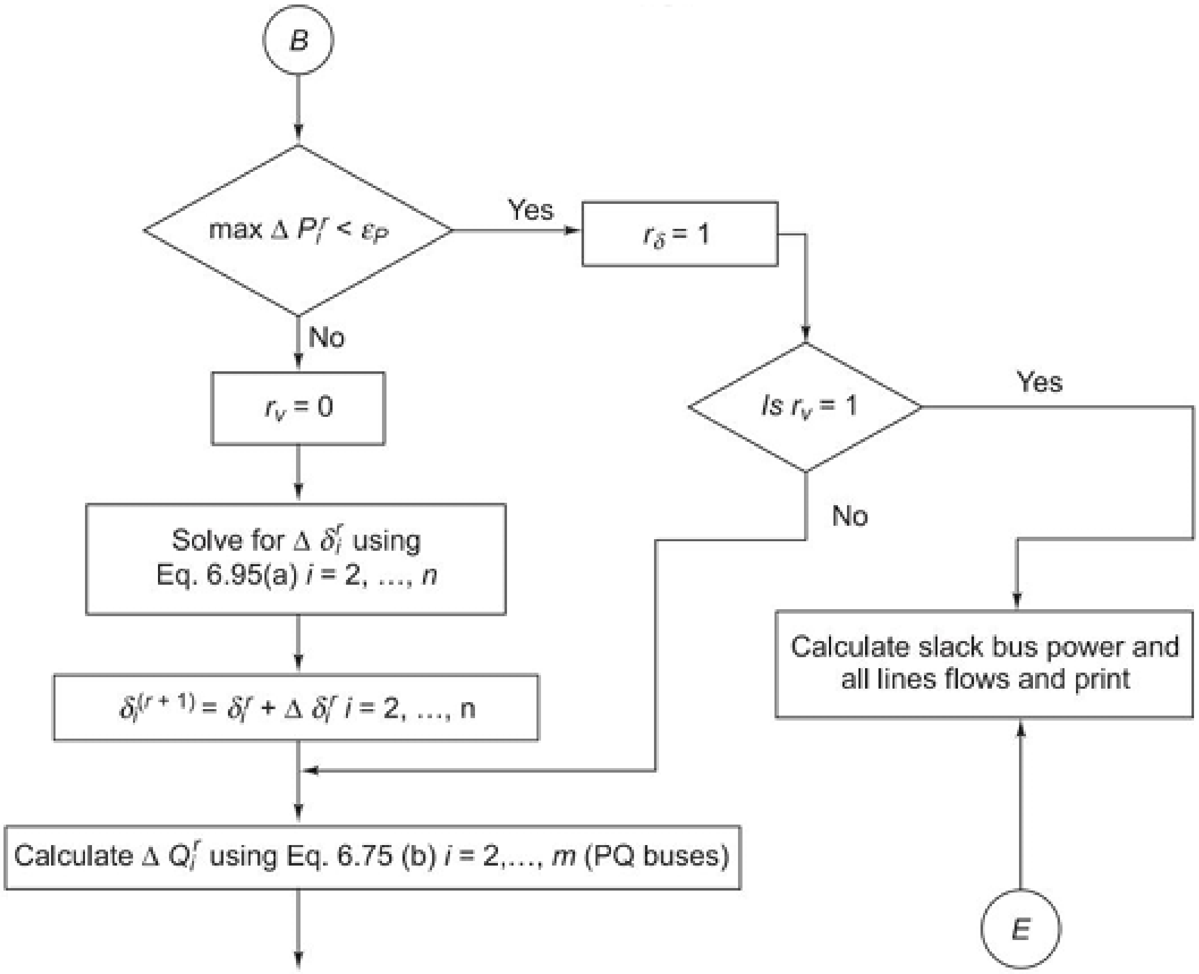
Fig. 3.2 Flow Chart for Load Flow Solution by Newton–Raphson Iterative Method using YBUS

A flow chart giving FDLF algorithm is presented in **Fig. 3.3**



Note: **Eqn. 6.75 (a):**

$$\Delta P_i^r = P_i\,(scheduled) - P_i^r \; for\ PV\ and\ PQ\ buses$$



Note: **Eqn. 6.95 (a):**

$$\left[\frac{\Delta P}{|V|}\right] = [B'] \, [\Delta\delta]$$

Eqn. 6.75 (b):

$$\Delta Q_i^r = Q_i \, (scheduled) - Q_i^r \; for \; PQ \; buses$$

Eqn. 6.95 (b):

$$\left[\frac{\Delta Q}{|V|}\right] = [B''] \, [\Delta|V|]$$

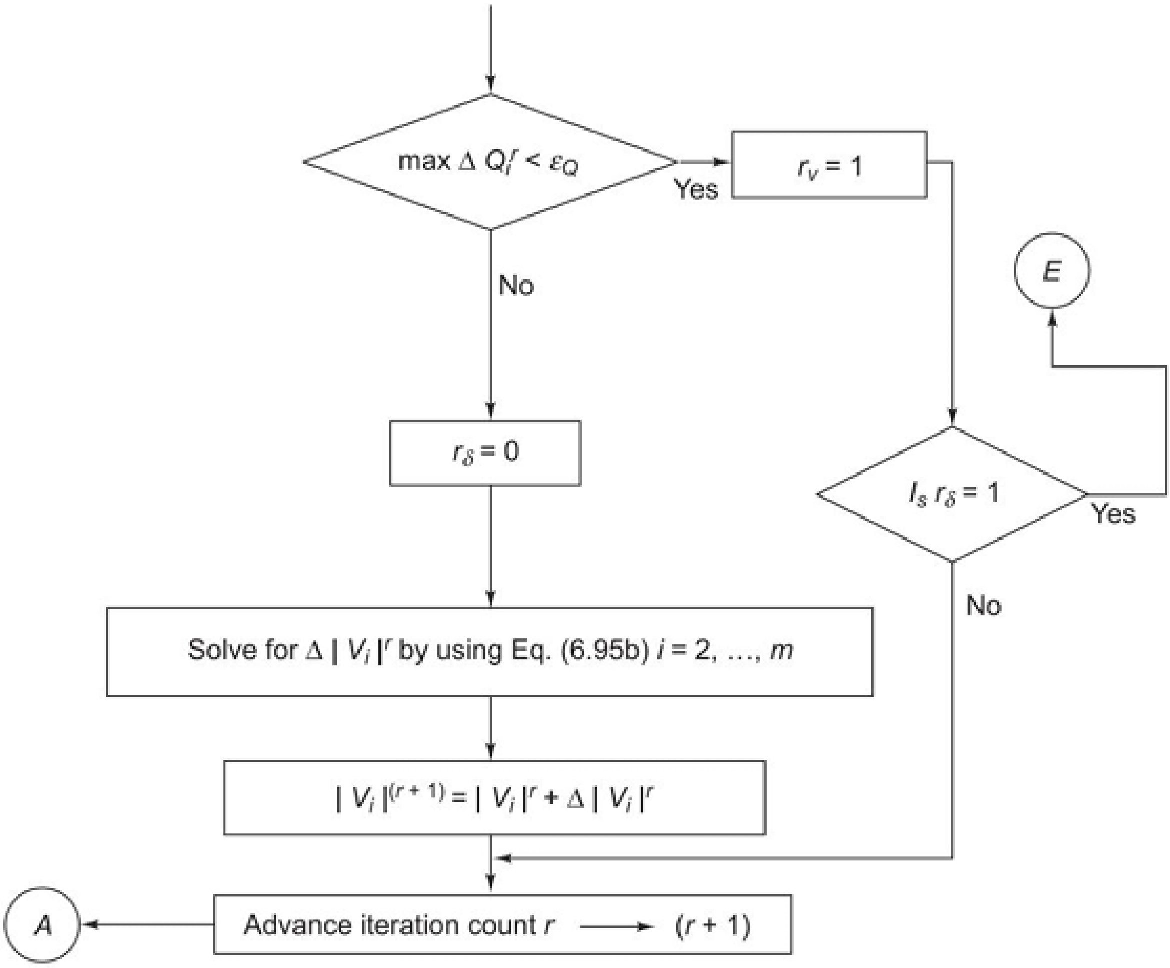


Fig. 3.3 Flow Chart for FDLF Algorithm

3.4 Comparison of Load Flow Methods

Sl. No	Parameter of Comparison	Gauss Siedel (GS) Method	Newton Raphson (NR) Method	Fast Decoupled Load Flow (FDLF) Method
1.	Coordinates	Works well with rectangular coordinates	Polar coordinates preferred as rectangular coordinates more memory	Polar coordinates
2.	Arithmetic Operations	Least in number to be completed one iteration	Elements of Jacobian to be calculated in each iteration	Less than Newton Raphson Method
3.	Time	Requires less time per iteration but increases with in number of buses	Time / Iteration is 7times of GS Method and increases with increase in number of buses	Less time when compared to NR Method and GS Method
4.	Convergence	Linear convergence	Quadratic convergence	Geometric convergence
5.	Number of iterations	Larger number, increases with increase in buses	Very less (3 to 5 only) for large systems and is practically constant	Only 2 to 5 iterations for practical accuracies

6.	Slack Bus Selection	Choice of slack bus affects convergence adversely	Sensitivity to this is minimal	Moderate
7.	Accuracy	Less accurate	More accurate	Moderate
8.	Memory	Less memory because of sparsity of matrix	Large memory even with compact storage scheme	Only 60% of memory when compared to NR Method
9.	Usage / Application	Small size systems	Large systems, ill-conditioned problems, optimal load flow studies	Optimization studies multiple load flow studies, contingency evaluation for security assessment and
10.	Programming Logic	Easy	Very difficult	Moderate
11.	Reliability	Reliable only for small systems	Reliable even for large systems	More reliable than NR Method

3.5 Illustrative Examples

Example-1: Consider the three-bus system of Fig. 3.4. Each of the three lines has a series impedance of 0.02 + j0.08 pu and a total shunt admittance of j0.02 pu. The specified quantities at the buses are tabulated below:

Bus	*Real load demand* P_D	*Reactive load demand* Q_D	*Real power generation* P_G	*Reactive Power generation* Q_G	*Voltage specification*
1	2.0	1.0	Unspecified	Unspecified	$V_1 = 1.04 + j0$ (Slack bus)
2	0.0	0.0	0.5	1.0	Unspecified (PQ bus)
3	1.5	0.6	0.0	$Q_{G3} = ?$	$V_3 = 1.04$ (PV bus)

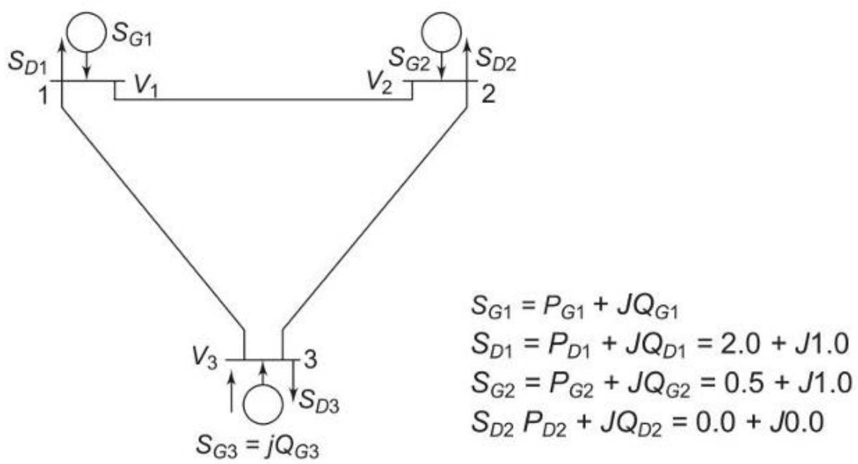


Fig.3.4

Controllable reactive power source is available at bus 3 with the constraint, $0 \leq Q_{G3} \leq 1.5$ pu Find the load flow solution using the NR method. Use a tolerance of 0.01 for power mismatch.

Solution:

Using the nominal-π model for transmission lines, Y_{BUS} for the given system is obtained as follows. For each line

$$y_{\text{series}} = \frac{1}{0.02 + J\,0.08} = 2.941 - j11.764 = 12.13 \angle -75.96^\circ$$

$$\text{Each off-diagonal term} = -2.941 + j11.764$$

$$\text{Each self term} = 2((2.941 - j11.764) + j0.01)$$

$$= 5.882 - j23.528 = 24.23 \angle -75.95^\circ$$

$$\boldsymbol{Y}_{\text{BUS}} = \begin{bmatrix} 24.23\angle{-75.95^\circ} & 12.13\angle 104.04^\circ & 12.13\angle 104.04^\circ \\ 12.13\angle 104.04^\circ & 24.23\angle -75.95^\circ & 12.13\angle 104.04^\circ \\ 12.13\angle 104.04^\circ & 12.13\angle 104.04^\circ & 24.23\angle -75.95^\circ \end{bmatrix}$$

To start iteration choose $V_2^0 = 1 + j0 \; and \; \delta_3^0 = 0$

From power flow equations (a & b),

$$P_2 = [|V_2||V_1||Y_{21}|\cos(\theta_{21} + \delta_1 - \delta_2) + |V_2|^2 |Y_{22}|\cos\theta_{22}$$
$$+ |V_2||V_3||Y_{23}|\cos(\theta_{23} + \delta_3 - \delta_2)]$$
$$P_3 = [|V_3||V_1||Y_{31}|\cos(\theta_{31} + \delta_1 - \delta_3)$$
$$+ |V_3||V_2||Y_{32}|\cos(\theta_{32} + \delta_2 - \delta_3) + |V_3|^2 |Y_{33}|\cos\theta_{33}]$$
$$Q_2 = [-|V_2||V_1||Y_{21}|\sin(\theta_{21} + \delta_1 - \delta_2) - |V_2|^2 |Y_{22}|\sin\theta_{22}$$
$$-|V_2||V_3||Y_{23}|\sin(\theta_{23} + \delta_3 - \delta_2)]$$

Substituting given and assumed values of different quantities, we get the values of powers as

$$P_2^0 = -0.23\, pu; \qquad P_3^0 = 0.12\, pu; \qquad Q_2^0 = \text{-0.96 pu}$$

Power residuals as per Eqn. (12b) are

$$\Delta P_2^0 = P_2\,(specified) - P_2^0\,(calculated) = 0.5 - (-0.23) = 0.73$$

$$P_3^0 = P_3\,(specified) - P_3^0\,(calculated) = -1.5 - (-0.12) = -1.62$$

$$\Delta Q_2^0 = Q_2\,(specified) - Q_2^0\,(calculated) = 1 - (-0.96) = 1.96$$

The changes in variables at the end of the first iteration are obtained as follows:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} \partial P_2/\partial\delta_2 & \partial P_2/\partial\delta_3 & \partial P_2/\partial|V_2| \\ \partial P_3/\partial\delta_2 & \partial P_3/\partial\delta_3 & \partial P_3/\partial|V_2| \\ \partial Q_2/\partial\delta_2 & \partial Q_2/\partial\delta_3 & \partial Q_2/\partial|V_2| \end{bmatrix} = \begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta|V_2| \end{bmatrix}$$

Jacobian elements can be evaluated by differentiating the expressions given above for P_2, P_3, Q_2 with respect to δ_2, δ_3 and $|V_2|$ and substituting the given and assumed values at the start of iteration. The changes in variables are obtained as

$$\begin{bmatrix} \Delta\delta_2^1 \\ \Delta\delta_3^1 \\ \Delta|V_2|^1 \end{bmatrix} = \begin{bmatrix} 24.47 & -12.23 & 5.64 \\ -12.23 & 24.95 & -3.05 \\ -6.11 & 3.05 & 22.54 \end{bmatrix}^{-1} \begin{bmatrix} 0.73 \\ -1.62 \\ 1.96 \end{bmatrix} = \begin{bmatrix} -0.0230 \\ -0.0654 \\ 0.0890 \end{bmatrix}$$

$$\begin{bmatrix} \delta_2^1 \\ \delta_3^1 \\ |V_2|^1 \end{bmatrix} = \begin{bmatrix} \delta_2^0 \\ \delta_3^0 \\ |V_2|^0 \end{bmatrix} + \begin{bmatrix} \Delta\delta_2^1 \\ \Delta\delta_3^1 \\ \Delta|V_2|^1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.0230 \\ -0.0654 \\ +0.0890 \end{bmatrix} = \begin{bmatrix} -0.0230 \\ -0.0654 \\ 1.0890 \end{bmatrix}$$

We can now calculate, using power flow Eqn. (b),

$$Q_3^1 = 0.4677; \qquad Q_{G3}^1 = Q_3^1 + Q_{D3} = 0.4677 + 0.6 = 1.0677$$

which is within limits.

If the same problem is solved using a digital computer, the solution converges in three iterations. The final results are given below:

$$V_2 = 1.081\angle -0.024\, rad; \qquad V_3 = 1.04\angle -0.0655\, rad$$

$$Q_{G3} = -0.15 + 0.6 = 0.45\ (within\ limits)$$

$$S_1 = 1.031 + j(-0.791); \qquad S_2 = 0.5 + j1.00; \qquad S_3 = -1.5 - j0.15$$

Transmission loss = 0.031 pu

Line Flows: The following matrix shows the real part of line flows.

$$\begin{bmatrix} 0.0 & 0.191312E00 & 0.839861E00 \\ -0.184229E00 & 0.0 & 0.684697E00 \\ -0.826213E00 & -0.673847E00 & 0.0 \end{bmatrix}$$

The following matrix shows the imaginary part of line flows.

$$\begin{bmatrix} 0.0 & -0.599464E00 & -0.191782E00 \\ -0.605274E00 & 0.0 & 0.396045E00 \\ 0.224742E00 & -0.375165E00 & 0.0 \end{bmatrix}$$

Example-2: In a two bus power system network shown in fig. 3.5, bus-1 is a slack bus with $V_1 = 1\angle 0^o$ pu and bus-2 is a load bus with P_2 = 10MW; Q_2 = 50MVAr. The line impedance is (0.12 +j 0.16) pu on a base of 100MVA. Using NR method of load flow solution, compute the elements of the Jacobian matrix. (VTU June/July 2024 QP)

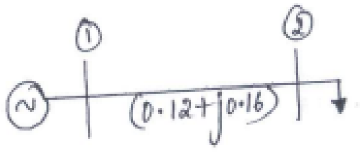


Fig. 3.5

Solution:

Given Data:

- **Bus 1**: Slack Bus, $V_1 = 1\angle 0^o$
- **Bus 2**: Load Bus with P_2 =100 MW, Q_2 = 50MVAr
- **Base Power** = 100 MVA
- **Line Impedance** = Z = 0.12 + j0.16 pu
- **Number of Buses** = 2

Step 1: Convert the given load to per unit

$$P_2^{pu} = \frac{100}{100} = 1.0\,pu, \quad Q_2^{pu} = \frac{50}{100} = 0.5\,pu$$

Step 2: Calculate Admittance (Y_{BUS})

$$Z = 0.12 + j0.16$$

$$Y = \frac{1}{Z} = \frac{1}{0.12 + j0.16}$$

$$|Z| = \sqrt{0.12^2 + 0.16^2} = 0.2, \quad \theta = \tan^{-1}\left(\frac{0.16}{0.12}\right) = 53.13^\circ$$

$$Y = \frac{1}{0.2}\angle - 53.13^\circ = 5\angle - 53.13^\circ = 3 - j4\,pu$$

Thus,

$$Y_{12} = Y_{21} = -(3 - j4), \quad Y_{11} = Y_{22} = 3 - j4$$

Ybus Matrix:

$$Y_{bus} = \begin{bmatrix} 3-j4 & -3+j4 \\ -3+j4 & 3-j4 \end{bmatrix}$$

Step 3: Power Equations at Bus 2

$$P_2 = V_2 \sum_{k=1}^{2} V_k [G_{2k}\cos(\delta_2-\delta_k) + B_{2k}\sin(\delta_2-\delta_k)]$$

$$Q_2 = V_2 \sum_{k=1}^{2} V_k [G_{2k}\sin(\delta_2-\delta_k) - B_{2k}\cos(\delta_2-\delta_k)]$$

Since bus-1 is slack:

- $V_1 = 1\angle 0^\circ$, $V_2 = V_2\angle\delta_2$.

Step 4: Jacobian Matrix Elements

The Jacobian matrix for 1 PQ bus (Bus 2) is:

$$\begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial V_2} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial V_2} \end{bmatrix}$$

Partial Derivatives:

1. $\frac{\partial P_2}{\partial \delta_2}$:

$$= V_1V_2[-G_{21}\sin(\delta_2-\delta_1) + B_{21}\cos(\delta_2-\delta_1)]$$
$$+2V_2^2B_{22}$$

2. $\frac{\partial P_2}{\partial V_2}$:

$$= V_1[G_{21}\cos(\delta_2-\delta_1) + B_{21}\sin(\delta_2-\delta_1)]$$
$$+2V_2G_{22}$$

3. $\frac{\partial Q_2}{\partial \delta_2}$:

$$= V_1 V_2 [G_{21} \cos(\delta_2 - \delta_1) + B_{21} \sin(\delta_2 - \delta_1)]$$

4. $\frac{\partial Q_2}{\partial V_2}$:

$$= -V_1 [G_{21} \sin(\delta_2 - \delta_1) - B_{21} \cos(\delta_2 - \delta_1)]$$

$$-2V_2 B_{22}$$

Step 5: Simplification at Initial Guess

At the initial iteration (flat start):

- $V_2 = 1$, $\delta_2 = 0$, $\delta_1 = 0$, $\cos(0) = 1$, $\sin(0) = 0$

Values:

- $G_{21} = -3$, $B_{21} = 4$, $G_{22} = 3$, $B_{22} = -4$, $V_1 = 1$

Final Jacobian Elements:

- $\frac{\partial P_2}{\partial \delta_2} = V_1 V_2 [0 + 4(1)] = 4$
- $\frac{\partial P_2}{\partial V_2} = V_1 [-3(1) + 0] + 2(1)(3) = -3 + 6 = 3$
- $\frac{\partial Q_2}{\partial \delta_2} = V_1 V_2 [-3(1) + 0] = -3$
- $\frac{\partial Q_2}{\partial V_2} = -V_1 [0 - 4(1)] - 2(1)(-4) = 4 + 8 = 12$

Final Jacobian Matrix:

$$J = \begin{bmatrix} 4 & 3 \\ -3 & 12 \end{bmatrix}$$