POSOCO - ERLDC

14, Golf Club Road, Tollygunge, KOLKATA-700 033

INTERNSHIP PROJECT REPORT ON

"Load flow analysis by Newton-Rapshon method using Matlab programming"

Intership period: ***08.05.2017 to 31.05.2017***

Prepared by

Name	Roll No.	Department	Semester	College
RAJSEKHAR GANGULI	510614003	ELECTRICAL ENGINEERING	6th	IIEST,SHIBPUR

ACKNOWLEDGEMENT

Internship at ERLDC, Kolkata has been a golden opportunity to me for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through, who guided me, taught me & blessed me with profound knowledge.

I wish to express my indebted gratitude and special thanks to Mr**. Surajit Banerjee** who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path, allowing me to learn the basics about this esteemed organization.

This project would not be possible without the continuous encouragement and helpful advice of **Mr. Saibal Ghosh**, who also gave me the idea of this project "Load flow analysis by Newton-Rapshon method using Matlab programming".

I am also obliged to **Mr. G.K.Kundu** and other members for their cooperation during the period of internship.

CONTENTS

Load Flow Analysis and its uses

A load flow study should be performed during the planning design stages of a power system and when evaluating changes to an existing system. A load flow study calculates the voltage drop on each feeder, the voltage at each bus, and the power flow and losses in all branch and feeder circuits. Load flow studies determine if system voltages remain within specified limits under normal or emergency operating conditions, and whether equipment such as transformers and conductors are overloaded.

Load flow studies are commonly used to:

- Optimize component or circuit loading
- Develop practical bus voltage profiles
- Identify real and reactive power flow
- Minimize kW and kVAR losses
- Develop equipment specification guidelines
- Identify proper transformer tap settings

The most important information obtained from the load flow analysis is the voltage profile of the system. If voltage varies greatly over the system, large reactive flows will result. This, in turn, will lead to increased real power losses and, in extreme cases, an increased likelihood of voltage collapse. When a particular bus has an unacceptably low voltage, the usual practice is to install capacitor banks in order to provide reactive compensation to the load. Load flow studies are used to determine how much reactive compensation should be applied at a bus, to bring its voltage up to an appropriate level. If new lines (or additional transformers) are to be installed, to reinforce the system, a power flow study will show how it will relieve overloads on adjacent lines. An inefficient or unbalanced load can also cause unpredictable behavior in your localized power grid, increasing the risk of equipment damage and unplanned outages.

Load Flow Problem Formulation

The net complex power injection into a bus is defined as Sk=Sgk-Sdk.Now an expression for this quantity in terms of network voltages and admittances is to be derived. Sk may be expressed as: Sk=VkIk*
The current injection into any bus k may be expressed as :

$$I_k = \sum_{j=1}^{N} Y_{kj} V_j$$

where the Ykj terms are admittance matrix elements. So,

$$S_k = V_k \left(\sum_{j=1}^{N} Y_{kj} V_j \right)^* = V_k \sum_{j=1}^{N} Y_{kj}^* V_j^*$$

[Vk=|Vk|∠θk]

Gkj and Bkj are defined as the real and imaginary parts of the admittance matrix element Ykj, respectively, so that Ykj=Gkj+jBkj. Now,

$$S_k = V_k \sum_{j=1}^{N} Y_{kj}^* V_j^* = |V_k|\angle\theta_k \sum_{j=1}^{N} (G_{kj} + jB_{kj})^* \left(|V_j|\angle\theta_j\right)^* = |V_k|\angle\theta_k \sum_{j=1}^{N} (G_{kj} - jB_{kj})\left(|V_j|\angle -\theta_j\right)$$

$$= \sum_{j=1}^{N} |V_k|\angle\theta_k \left(|V_j|\angle -\theta_j\right)(G_{kj} - jB_{kj}) = \sum_{j=1}^{N} \left(|V_k||V_j|\angle(\theta_k - \theta_j)\right)(G_{kj} - jB_{kj})$$

or,

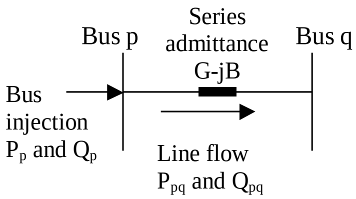
$$S_k = \sum_{j=1}^{N} \left(|V_k||V_j|\angle(\theta_k - \theta_j)\right)(G_{kj} - jB_{kj})$$

$$= \sum_{j=1}^{N} |V_k||V_j| \left(\cos(\theta_k - \theta_j) + j\sin(\theta_k - \theta_j)\right)(G_{kj} - jB_{kj})$$

As, Sk=Pk+jQk, it can be written that,

$$P_k = \sum_{j=1}^{N} |V_k||V_j| \left(G_{kj}\cos(\theta_k - \theta_j) + B_{kj}\sin(\theta_k - \theta_j)\right)$$

$$Q_k = \sum_{j=1}^{N} |V_k||V_j| \left(G_{kj}\sin(\theta_k - \theta_j) - B_{kj}\cos(\theta_k - \theta_j)\right)$$



Bus p Connected to Only Bus q

$$P_p = |V_p|^2 G_{pp} + |V_p||V_q| G_{pq}\cos(\theta_p - \theta_q) + |V_p||V_q| B_{pq}\sin(\theta_p - \theta_q)$$
$$Q_p = -|V_p|^2 B_{pp} + |V_p||V_q| G_{pq}\sin(\theta_p - \theta_q) - |V_p||V_q| B_{pq}\cos(\theta_p - \theta_q)$$

$$P_p = |V_p|^2 G - |V_p||V_q| G\cos(\theta_p - \theta_q) + |V_p||V_q| B\sin(\theta_p - \theta_q)$$
$$Q_p = |V_p|^2 B - |V_p||V_q| B\cos(\theta_p - \theta_q) - |V_p||V_q| G\sin(\theta_p - \theta_q)$$

In general,

$$P_k = \sum_{j=1}^{N} |V_k||V_j| \left(G_{kj}\cos(\theta_k - \theta_j) + B_{kj}\sin(\theta_k - \theta_j)\right), \qquad k = 2,\dots,N$$
$$Q_k = \sum_{j=1}^{N} |V_k||V_j| \left(G_{kj}\sin(\theta_k - \theta_j) - B_{kj}\cos(\theta_k - \theta_j)\right), \qquad k = N_G + 1,\dots,N$$

Here,

a. The angles for the voltage phasors at all buses except the slack bus (it is 0° at the slack bus), i.e., θ_k, k=2,…,N
b. The magnitudes for the voltage phasors at all type PQ buses, i.e., $|V_k|$, k=N_G+1, …, N

Solution of a Set of Nonlinear Equations by Newton-Raphson Method

In this section we shall discuss the solution of a set of nonlinear equations through Newton-Raphson method. Let us consider that we have a set of *n* nonlinear equations of a total number

$$f_1(x_1,\cdots,x_n)=\eta_1$$
$$f_2(x_1,\cdots,x_n)=\eta_2$$
$$\vdots$$
$$f_n(x_1,\cdots,x_n)=\eta_n$$

of *n* variables x_1 , x_2 , ... , x_n. where f_1, ... , f_n are functions of the variables x_1 , x_2 , ... , x_n.

We can then define another set of functions g_1 , ... , g_n as given below,

$$g_1(x_1,\cdots,x_n)=f_1(x_1,\cdots,x_n)-\eta_1=0$$
$$g_2(x_1,\cdots,x_n)=f_2(x_1,\cdots,x_n)-\eta_2=0$$
$$\vdots$$
$$g_n(x_1,\cdots,x_n)=f_n(x_1,\cdots,x_n)-\eta_n=0$$

Let us assume that the initial estimates of the *n* variables are $x_1^{(0)}$, $x_2^{(0)}$, ..., $x_n^{(0)}$. Let us add corrections $\Delta x_1^{(0)}$, $\Delta x_2^{(0)}$, ... , $\Delta x_n^{(0)}$ to these variables such that we get the correct solution of these variables defined by

$$x_1^* = x_1^{(0)}+\Delta x_1^{(0)}$$
$$x_2^* = x_2^{(0)}+\Delta x_2^{(0)}$$
$$\vdots$$
$$x_n^* = x_n^{(0)}+\Delta x_n^{(0)}$$

So, $g_k\left(x_1^*,\cdots,x_n^*\right)=g_k\left(x_1^{(0)}+\Delta x_1^{(0)},\cdots,x_n^{(0)}+\Delta x_n^{(0)}\right)=0$, where $k=1,\ldots,n$

We can then expand the above equation in Taylor 's series around the nominal values of $x_1^{(0)}$, $x_2^{(0)}$, ... , $x_n^{(0)}$. Neglecting the second and higher order terms of the series, the expansion of g_k , k = 1, ... , *n* is given as

$$g_k\left(x_1^*,\cdots,x_n^*\right)=g_k\left(x_1^{(0)},\cdots,x_n^{(0)}\right)+\Delta x_1^{(0)}\left.\frac{\partial g_k}{\partial x_1}\right|^{(0)}+\Delta x_2^{(0)}\left.\frac{\partial g_k}{\partial x_2}\right|^{(0)}+\cdots+\Delta x_n^{(0)}\left.\frac{\partial g_k}{\partial x_n}\right|^{(0)}$$

where $\left.\partial g_k/\partial x_i\right|^{(0)}$ is the partial derivative of g_k evaluated at $x_2^{(1)}$, ... , $x_n^{(1)}$.

In vector-matrix form,

$$\begin{bmatrix} \partial g_1/\partial x_1 & \partial g_1/\partial x_2 & \cdots & \partial g_1/\partial x_n \\ \partial g_2/\partial x_1 & \partial g_2/\partial x_2 & \cdots & \partial g_2/\partial x_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial g_n/\partial x_1 & \partial g_n/\partial x_2 & \cdots & \partial g_n/\partial x_n \end{bmatrix}^{(0)} \begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \\ \vdots \\ \Delta x_n^{(0)} \end{bmatrix} = \begin{bmatrix} 0-g_1\left(x_1^{(0)},\cdots,x_n^{(0)}\right) \\ 0-g_2\left(x_1^{(0)},\cdots,x_n^{(0)}\right) \\ \vdots \\ 0-g_n\left(x_1^{(0)},\cdots,x_n^{(0)}\right) \end{bmatrix}$$

Or,

$$\begin{bmatrix}\Delta x_1^{(0)}\\ \Delta x_2^{(0)}\\ \vdots \\ \Delta x_n^{(0)}\end{bmatrix} = \left[J^{(0)}\right]^{-1}\begin{bmatrix}\Delta g_1^{(0)}\\ \Delta g_2^{(0)}\\ \vdots \\ \Delta g_n^{(0)}\end{bmatrix}; \qquad \Delta g_k^{(0)} = g_k\left(x_1^*,\cdots,x_n^*\right) - g_k\left(x_1^{(0)},\cdots,x_n^{(0)}\right), \text{ k= 1, ...n}$$

Since the Taylor 's series is truncated by neglecting the 2nd and higher order terms, we cannot expect to find the correct solution at the end of first iteration. We shall then have

$$\begin{aligned} x_1^{(1)} &= x_1^{(0)} + \Delta x_1^{(0)} \\ x_2^{(1)} &= x_2^{(0)} + \Delta x_2^{(0)} \\ &\vdots \\ x_n^{(1)} &= x_n^{(0)} + \Delta x_n^{(0)} \end{aligned}$$

These are then used to find $J^{(1)}$ and $\Delta g_k^{(1)}$, $k = 1, \ldots, n$. We can then find $\Delta x_2^{(1)}$, ... , $\Delta x_n^{(1)}$ from an equation like (4.28) and subsequently calculate $x_2^{(1)}$, ... , $x_n^{(1)}$. The process continues till Δg_k , $k = 1,\ldots, n$ becomes less than a small quantity.

$$\Delta g_k^{(0)} = 0 - g_k\left(x_1^{(0)},\cdots,x_n^{(0)}\right)$$

Load Flow By Newton-Raphson Method

❖ *Load Flow Algorithm*

Assume that an n -bus power system contains a total n_p number of P-Q buses while the number of P-V (generator) buses be n_g such that $n = n_p + n_g + 1$. Bus-1 is assumed to be the slack bus. Also further use the mismatch equations of ΔP_i and ΔQ_i given in and respectively. The approach to Newton-Raphson load flow is similar to that of solving a system of nonlinear equations using the **Newton-Raphson method**: At each iteration we have to form a Jacobian matrix and solve for the corrections from an equation of the type given in.For the load flow problem, this equation is of the form

$$J\begin{bmatrix} \Delta\delta_2 \\ \vdots \\ \Delta\delta_n \\ \dfrac{\Delta|V_2|}{|V_2|} \\ \vdots \\ \dfrac{\Delta|V_{1+n_p}|}{|V_{1+n_p}|} \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_n \\ \Delta Q_2 \\ \vdots \\ \Delta Q_{1+n_p} \end{bmatrix}$$

where the Jacobian matrix is divided into submatrices as,

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

The size of the Jacobian matrix is (n + np − 1) x (n + np −1).
The dimensions of the submatrices are as follows:

J_{11}: $(n - 1) \times (n - 1)$, J_{12}: $(n - 1) \times n_p$, J_{21}: $n_p \times (n - 1)$ and J_{22}: $n_p \times n_p$

The submatrics are

$$J_{11} = \begin{bmatrix} \dfrac{\partial P_2}{\partial \delta_2} & \cdots & \dfrac{\partial P_2}{\partial \delta_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial P_n}{\partial \delta_2} & \cdots & \dfrac{\partial P_n}{\partial \delta_n} \end{bmatrix}$$

$$J_{12} = \begin{bmatrix} |V_2|\dfrac{\partial P_2}{\partial |V_2|} & \cdots & |V_{1+n_p}|\dfrac{\partial P_2}{\partial |V_{1+n_p}|} \\ \vdots & \ddots & \vdots \\ |V_2|\dfrac{\partial P_n}{\partial |V_2|} & \cdots & |V_{1+n_p}|\dfrac{\partial P_n}{\partial |V_{1+n_p}|} \end{bmatrix}$$

$$J_{21} = \begin{bmatrix} \frac{\partial Q_2}{\partial \delta_2} & \cdots & \frac{\partial Q_2}{\partial \delta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial Q_{1+n_o}}{\partial \delta_2} & \cdots & \frac{\partial Q_{1+n_o}}{\partial \delta_n} \end{bmatrix} \qquad J_{22} = \begin{bmatrix} |V_2|\frac{\partial Q_2}{\partial |V_2|} & \cdots & |V_{1+n_o}|\frac{\partial Q_2}{\partial |V_{1+n_o}|} \\ \vdots & \ddots & \vdots \\ |V_2|\frac{\partial Q_{1+n_o}}{\partial |V_2|} & \cdots & |V_{1+n_o}|\frac{\partial Q_{1+n_o}}{\partial |V_{1+n_o}|} \end{bmatrix}$$

❖ ***Load Flow Algorithm***

The Newton-Raphson procedure is as follows:

Step-1: Choose the initial values of the voltage magnitudes |V| (0) of all np load buses and n − 1 angles δ (0) of the voltages of all the buses except the slack bus.

Step-2: Use the estimated |V|(0) and δ (0) to calculate a total n − 1 number of injected real power Pcalc(0) and equal number of real power mismatch ΔP (0) .

Step-3: Use the estimated |V| (0) and δ (0) to calculate a total np number of injected reactive power Qcalc(0) and equal number of reactive power mismatch ΔQ (0) and to formulate the Jacobian matrix J (0) for δ (0) and Δ |V| (0) ÷ |V| (0).

Step-4 : Obtain the updates from

$$\delta^{(1)} = \delta^{(0)} + \Delta\delta^{(0)}$$

Step-5: Solve

$$|V|^{(1)} = |V|^{(0)}\left[1 + \frac{\Delta|V|^{(0)}}{|V|^{(0)}}\right]$$

Step-6: Check if all the mismatches are below a small number. Terminate the process if yes. Otherwise go back to step-1 to start the next iteration with the updates given by the last two equations.

❖ ***Formation of Jacobian Matrics***

The real and reactive power equations for n-bus system ->

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

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

A. Formation of J_{11} --> Let us define J_{11} as,

$$J_{11} = \begin{bmatrix} L_{22} & \cdots & L_{2n} \\ \vdots & \ddots & \vdots \\ L_{n2} & \cdots & L_{nn} \end{bmatrix}$$

$$L_{ik} = \frac{\partial P_i}{\partial \delta_k} = -|Y_{ik} V_i V_k| \sin(\theta_{ik} + \delta_k - \delta_i), \quad i \neq k$$

For i=k,

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

So,

$$L_{ii} = \frac{\partial P_i}{\partial \delta_i} = -Q_i - |V_i|^2 B_{ii}$$

B. Formation of J_{21} --> Let us define J_{21} as,

$$J_{21} = \begin{bmatrix} M_{22} & \cdots & M_{2n} \\ \vdots & \ddots & \vdots \\ M_{n_o 2} & \cdots & M_{n_o n} \end{bmatrix}$$

$$M_{ik} = \frac{\partial Q_i}{\partial \delta_k} = -|Y_{ik} V_i V_k| \cos(\theta_{ik} + \delta_k - \delta_i), \quad i \neq k$$

For i=k,

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

C. Formation of J_{12} --> Let us define J_{12} as,

$$J_{12} = \begin{bmatrix} N_{22} & \cdots & N_{2n_o} \\ \vdots & \ddots & \vdots \\ N_{n2} & \cdots & N_{nn_o} \end{bmatrix}$$

$$N_{ik} = |V_k|\frac{\partial P_i}{\partial |V_k|} = |Y_{ik}V_iV_k|\cos(\theta_{ik} + \delta_k - \delta_i) = -M_{ik} \quad i \neq k$$

For i=k,

$$N_{ii} = |V_i|\frac{\partial P_i}{\partial |V_i|} = |V_i|\left[2|V_i|G_{ii} + \sum_{\substack{k=1 \\ k\neq i}}^{n}|Y_{ik}V_k|\cos(\theta_{ik} + \delta_k - \delta_i)\right]$$

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

D. Formation of J_{22} --> For the formation of J_{22} let us define :

$$J_{22} = \begin{bmatrix} O_{22} & \cdots & O_{2n_o} \\ \vdots & \ddots & \vdots \\ O_{n_o2} & \cdots & O_{n_on_o} \end{bmatrix}$$

$$O_{ik} = |V_i|\frac{\partial Q_i}{\partial |V_k|} = -|V_i||Y_{ik}V_iV_k|\sin(\theta_{ik} + \delta_k - \delta_i) = L_{ik}, \quad i \neq k$$

Again, for i=k we have—

$$O_{ii} = |V_i|\frac{\partial Q_i}{\partial |V_k|} = |V_i|\left[-2|V_i|B_{ii} - \sum_{\substack{k=1 \\ k\neq i}}^{n}|Y_{ik}V_k|\sin(\theta_{ik} + \delta_k - \delta_i)\right]$$

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

It can be seen, that once the submatrices J11 and J21 are computed, the formation of the submatrices J12 and J22 is fairly straightforward. For large system this will result in considerable saving in the computation time.

MATLAB PROGRAM FOR LOAD FLOW ANALYSIS BY NEWTON-RAPSHON METHOD

CODE -->

```
function [Ybus,theta] = LFNRq(nbus, nlns, linedata, sh, mbus, pfbd)

%INITIALISATION OF YBUS & CALCULATION OF YBUS MATRIX INCLUDING CHARGING

%SUCEPTANCE &TRANSFORMER TAP SETTING:

n1= linedata(:, 1) ; nr =linedata(:, 2) ; R= linedata(:, 3) ;

X= linedata(:, 4) ;Bc= j*linedata(:, 5) ; a= linedata(:, 6) ;

nbr=length(linedata(:, 1)) ; nbus= max(max(n1), max(nr)) ;

Z= R+ j*X ; y= ones(ndr, 1)./Z ;

Ybus= zeros (nbus, nbus) ;

         for k=1 : nbr

                    Ybus(n1(k), nr(k)) = Ybus(n1(k), nr(k)) – y(k)*a(k) ;

                     Ybus(nr(k), n1(k)) = ybus(n1(k), nr(k)) ;

         end

         for n=1 : nbus

              if sh(n) > 0

                  Ybus(n, n) =Ybus(n, n) +  j*sh(n) ;

              else

              end

        for k=1 : nbr

              if n1(k) == n

                 Ybus(n, n) = Ybus(n, n) y(k) =Bc(k) ;

              elseif  nr(k) == n

        Ybus(n, n) = Ybus(n, n) + y(k)*(a(k)^2) + Bc(k)   ;
```

```
                end
            end
 end
 Ybus ;
theta =angle(Ybus) ;
Ybus = abs (Ybus) ;
%INITIALISATION    PROCESS…………..
Vmag (l) = pfbd (l, 3) ;
delta (l) = pfbd (l, 4) ;
for  k= 2 : nbus
delta (k) = pfbd (k, 4) ;
Vmag(k) = pfbd (k, 3) ;
 end
itr =0 ;  count =0 ;
while count < 2*(nbus_l)_mbus ;
  itr = itr + l ;
% Q LIMIT CHECKING
for k=2 : nbus
sum = 0 ;
   if pfbd (k, 2) == 2
     for m=1 : nbus
       sum = sum + Ybus (k, m) * abs(Vmag (m))*sin(theta(k, m) – delta (k)+ delta
(m)) ;
    end
     Ql = - (abs (Vmag (k)) * sum) ;
     Q = Q1 + pfbd (k, 6) – pfbd (k, 11) ;
```

```
     if Q > pfbd (k, 9)
      Q = pfbd (k, 9) ;
     pfbd (k, 2) = 0 ;
     mbus= mbus – l ;
     end
   end
end
%CALCULATION  OF JACOBIAN MATRIX
JJ = zeros 9(2*(nbus – l) – mbus) , (2*(nbus – l) – mbus )) ;
X = 0 ;
for I =2 : nbus
y = 0 ;
    for m= 2: nbus
      sum = 0 ;
         if m == i ;
            for k= 1: nbus
               if k ~= m
              sum = sum + Ybus (I, k)*Vmag(k)*sin(theta(I, k) – delta (i) + delta (k) ;
              end
        end
        JJ( i- l + x, m – l + y) + Vmag(i)*sum ;
      if  pfbd( m, 2)  == 0
        y = y+ l ;
      else
     end
 else
```

```
  JJ (i- l + x, m –l +y ) = - Vmag(i)*Ybus( i,m)*Vmag(m)*sin(theta(I, m)- delta (i)
+ delta (m) ;
   if  pfbd(m, 2) == 0
   y = y + 1 ;
    else
    end
  end
end
    if  pfbd (i, 2) ==0
     x = x + 1 ;
    else
    end
 end
 I = 0 ; x= 0 ;
 for  i= 2: nbus
 z= 0; I = I + 1 ; J= 0 ;
    for m =2 : nbus
      if pfbd (m, 2) == 0
       sum1= 0 ; J = J + 2 ;
          if I == m
            for k= 1: nbus
                  if  k~= m
                         sum1= sum1 + Ybus (I, k)*Vmag(k)*cos(theta(I, k) – delta (i)
+ delta (k) ) ;
                   end
                end
```

```
  JJ (I + x, J + z) = 2*Vmag(i)*Ybus(i, j)*cos(theta(i, j)) + sum1 ;
  else
     JJ (I + x, J + z) = Vmag(i)*Ybus(i, m)*cos(theta(i, m) – delta(i) + delta(m) ;
  end
 else
 z = z + l ;
 end
end
  if pfbd(I, 2) == 0
  x = x + l ;
  else
  end
end
Il = 0 ; x= 0 ; z = 0 ;
for i=2: nbus
   yl= 0 ;
     if (pfbd(i, 2) == 0
         x = x + l ;     Il = Il + l ;    J = 0 ;
   for m= 2 : nbus
      sum1 = 0 ; J= J + l ;
              if I ==m
                     for k= 1: nbus
                         if k ~=m
                                 sum1 = sum1+ Ybus(i, k)*Vmag(k)*cos(theta (i, k) –
delta(i) + delta(k)) ;
                        end
```

```
                end
JJ(Il + x + z, J+ y1) = Vmag(i)*suml ;
if pfbd(m, 2) ==0
  y1= y1 + 1 ;
 else
 end
else
JJ(Il +x + z, J+y1) = -Vmag(i)*Ybus(i, m)*Vmag(m)*cos(theta(i, m)- delta(i)+
delta(m)) ;
if pfbd(m, 2) == 0
    y1 = y1 +1 ;
    else
   end
 end
end
else
    z= z + l ;
 end
end
I2 = 0 ; x= 0 ;
    for i = 2:nbus
            if pfbd(i, 2) == 0
                y= 0 ; I2= I2 + 2 ; J= 0 ;
            for m= 2: nbus
                if pfbd(m, 2) == 0
             sum1= 0 ; J= J+2 :
```

```
if I == m
          for k= 1: nbus
                   if k~=m
                       sum1 = sum1 + Ybus(i, k)*Vmag(k)*sin(theta(i, k)- delta(i) +
delta(k)) ;
  end
end

JJ(I2 = x, J+ y)= -2*Vmag(i)*Ybus(i,i)*sin(theta(i, i))- sum1 ;
else
JJ(I2 = x, J+ y)= -Vmag(i)*Ybus(i,m)*sin(theta(i, m)-delta(i) + delta(m)) ;
end
else
 y= y+ 1 ;
   end
  end
else
x = x + 1 ;
  end
end
JJ ;
JJ1= inv(JJ) ;
% CALCULATION   OF   de1P   &   de1Q   FROM   GIVEN   DATA :
    ddpq= zeros((2*(nbus- 1) – mbus) , 1) ;
x= 0 ;
for k= 2: nbus
```

```
  sum = 0 ;
   if pfbd(k, 2) == 0
        for m=1: nbus
              sum = sum + Ybus(k, m)*Vmag(m)*cos(theta(k, m) – delta(k) +
delta(m)) ;
  end
 sum1= Vmag(k)*sum ;
 ddpq(k- 1 +x)= (pfbd(k, 7)- pfbd(k, 5)) – sum1 ;
  sum =0 ;
for m= 1:nbus
       sum = sum + Ybus(k, m)*Vmag(m)*sin(theta(k, m) – delta(k) + delta(m)) ;
    end
sum2= -Vmag(k)*sum ;
ddpq(k+ x)= (pfbd(k, 8)+ pfbd(k, 11)- pfbd(k, 6))- sum2 ;
x= x+1 ;
else
     for m=1:nbus
           sum = sum + Ybus(k, m)*Vmag(m)*cos(theta(k, m) – delta(k) +
delta(m)) ;
      end
             sum1= Vmag(k)*sum ;
             ddpq(k-1+x)= (pfbd(k, 7) – pfbd(k, 5)) –sum1 ;
       end
end
ddpq ;
ddv= JJ1*ddpq ;
```

```
% CALCULATION      OF      NEW      VOLTAGE    &    ANGLE   :

x= 0 ;

for k = 1: nbus-1

n= k+ 1 ;

delta(n)= delta(n) + ddv((n+ x) ;

deltal(n)= (delta(n)*180)/pi ;

if pfbd(n, 2)==0

Vmag(n)= abs(Vmag(n)+ ddv(n+x)) ;   x=x+1 ;

else

end

end

%CHECKING   OF   POWER   MISMATCH   AT   BUSSES :

    for k=1: size(ddpq)

          if abs(ddpq(k))<=0.0001

               count = count + 1 ;

          end

  end

end

fprintf(‘ No. of Iterations= %g \n\n’,itr)

mbus ;

hdng= [‘ Bus No.          Voltage Mag.          Angle in Rad.        Angle in Deg. ‘ ]

disp(hdng)

for n=1: nbus
  fprintf(‘ %5g’, n), fprintf(‘%10.5f’,Vmag(n)), fprintf(‘%12.5f’, delta(n)),
fprintf(‘%12.5f\n’,deltal(n)),

end
```

INPUT DATA-->

```
% INPUT (POWER   FLOW   SOLUTION   BY   N-R   METHOD)

nbus= 6 ;

n1ns= 8 ;

linedata=   [ 1   3   0.04   0.3   0.01   1
              2   3   0.03   0.2   0.01   1
              3   4   0.03   0.2   0.01   1
              3   5   0.03   0.2   0.01   1
              4   5   0.03   0.15  0.01   1
              5   6   0.05   0.3   0.01   1
              1   2   0.0    0.4   0.00   1.05
              2   6   0.0    0.2   0.00   1.02  ] ;

%FROM_BUS,TO_BUS,R,X,B/2,OFF_NOMINAL_TAP_RATIO

sh= [0; 0; 0; 0; 0] ;

mbus= 1 ;      % Number of P-V Bus

pfbd=   [ 1   1   1.05   0   0      0      0      0      0   0   0
          2   0   1.00   0   1.00   0.10   0.00   0.00   0   0   0
          3   0   1.00   0   0.00   0.00   1.50   0.75   0   0   0
          4   2   1.02   0   0.45   0.20   0.00   0.00   2   0   0
          5   0   1.00   0   0.40   0.25   0.00   0.00   0   0   0
          6   0   1.00   0   0.30   0.10   0.00   0.00   0   0   0  ] ;

%BUS_NO.,GEN_BUS,VOLTAGE,ANGLE,P_DELI,Q_DELI,P_GEN,Q_GEN

[Ybus, theta]= LFNRq(nbus, n1ns, linedata, sh, mbus, pfbd) ;
```

OUTPUT -->

% OUTPUT (VOLTAGE & ANGLE VALUES)

No. of Iterations= 4

Bus No.	Voltage Mag.	Angle in Rad.	Angle in Deg.
1	1.05000	0.00000	0.00000
2	0.97876	- 0.17967	- 10.29431
3	1.03572	- 0.05549	- 3.17909
4	0.97256	- 0.14756	- 8.45432
5	0.96650	- 0.15373	- 8.80812
6	0.94676	- 0.20745	- 11.88597

REFERENCES

- https://en.wikipedia.org/wiki/Power-flow_study
- http://www.avotraining.com/arc-flash/load-flow-studies
- http://iitb.vlab.co.in/?sub=44&brch=141&sim=258&cnt=496
- www.nptel.ac.in
- https://www.researchgate.net
- "A Course in Power System" by J.B.Gupta
- "Power System Engineering" by Nagrath & Kothari

THANK YOU