Exp - 08	Load Flow Solution Using Newton Raphson Method

Aim:

To write the MATLAB coding to find the electrical parameters of various buses in a power system using Newton Raphson method.

Apparatus Required:

SI.No	Apparatus	Specification
1	PC	Dual core, RAM 512 MB 1.2 GHz speed, 80 GB
2	MATLAB	7.5

Theory:

The Newton Raphson method is a power full method of solving non-linear algebric equation. It work is faster and is sure to coverage in most cases. It is indeed the practical method of load flow solution of large power network. In order to solve non-linear equations through this method we need to calculate jacobian matrix. If is obtained by differentiating the function vector f with respect to x and evaluating it at x.

$$F_0+J_0\, X_0=0$$

These sets of linear algebraic equation can be solved effectively by triangularization and block substitution .Iterations are continued till Where i=1,2,3……..n

Its only drawback is large requirement of complex. Computer memory which has been over come through a compact storage scheme convergence can be speeded up by performing the first iteration through Gs method and using the value so obtained as starting Newton Raphson method.

Algorithm:

Step 1: Assume a flat voltage profile 1+j0bfor all buses except the slack bus

Step 2: Assume a suitable value of ε called convergence criterion

Step 3: Set iteration count, k=0 and assumed voltage profile of the buses are denoted as
$V_i^0, V_2^0, ...V_n^0$

Step 4: Set the bus count P = 1

Step 5: Check for slack bus, If it is a slack bus go to step 13 otherwise go to next step

Step 6: Calculate the real and reactive power of bus P using the equation

$$P_p^k = \sum_{q=1}^{n} \left[e_p^k \left(e_p^k G_{pq} + F_q^k B_{pq} \right) + F_q^k \left(F_q^k G_{pq} - e_q^k B_{pq} \right) \right]$$

$$Q_p^k = \sum_{q=1}^{n} \left[F_p^k \left(e_p^k G_{pq} + F_q^k B_{pq} \right) + e_p^k \left(F_q^k G_{pq} - e_q^k B_{pq} \right) \right]$$

Step 7: Calculate the change in real power

$$\Delta P^k = P_{p\,spec} - P_p^k$$

Step 8:Check the generator bus if it is a generator bus go to next step otherwise go to step 12

Step 9:Check for reactive power limit violation of generator bus.for this compare the calculated reactive power $Q_p^k$with specified limits if the limit is violated go to step 11 otherwise go to next step else go to step 13

Step 10: Check for reactive power limit violation ofgenerator buses if the limit is violated go to step 12 else go to next step

Step 11: If the calculated reactive power is within the specified limits then consider this bus as generator bus. Now calculate the voltage residue

$$\left|\Delta V_p^k\right|^2 = \left|V_p\right|^2_{spec} - \left|V_p^k\right|^2 \quad \text{then go to step 12}$$

Step 12:Is reactive power limit is violated then treat this bus as a load bus (i.e) if a $Q_p^k < Q_{p\,mn}$ then $Q_{p\,spc} = Q_{p1\,min}$

Step 13:Calculate change in reactive power for load bus

$$\Delta Q_p^k = \left|Q_{p\,spec}\right| - Q_p^k$$

Step 14: Repeat the step 6 to 13 untill all residues are calculated for this increment the bus count by 1 and go to step 6 until the Bs count is

Step15: Determine largest of the absolute value of residue let this change be ΔE

Step16: Compare ΔE & ε if ΔE < E then goto step 11 if ΔE > E go to next step

Step17: Determine the elements of jacobian matrix by partially differentiating the load flow equation and evaluates using K^{th} iteration values

Step18: Calculate the increments in real and reactive part of voltages Δe_p^k & $\Delta \delta_p^k$ by solving the matrix B=JC

Step19: Calculate new bus voltages

$$e_p^{k+1} = e_p^k + \Delta e_p^k$$

$$\delta_p^{k+1} = \delta_p^k + \Delta \delta_p^k$$

$$\left|V_p^{k+1}\right| = \sqrt{\left(e_p^{k+1}\right)^2 + \left(\delta_p^{k+1}\right)^2} \;\; \& \;\; \delta_p^{k+1} = \tan^{-1}\left(e_p^{k+1} \Big/ \delta_p^{k+1} \right)$$

Therefore

$$V_p^{k+1} = \left|V_p^{k+1}\right| \langle \delta_p^{k+1}$$

Step20: Advance iteration count K=K+1 & go to step 5

Step21: Calculate the line flows and stop the program

Program:

```
clear all;
clc;
n=input('Enter the number of buses');
for i=1:n
    for j=1:n
        fprintf('Enter the Admittance Value Between %d & %d',i,j)
        y(i,j)=input('');
    end
end
yb(n,n)=0;
for i=1:n
    for j=1:n
        if i==j
            for k=1:n
                yb(i,j)=yb(i,j)+y(i,k);
            end
        else
            yb(i,j)=-y(i,j);
        end
    end
end
mag(1)=1.05;
for i=2:n
    mag(i)=1;
end
th(1:n)=0;
for i=1:n
    acp(i)=input('enter real power value:');
end
for i=1:n
    acq(i)=input('enter reactive power value:');
end
my=abs(yb);an=angle(yb);
g=real(yb);b=imag(yb);
yb
mag
th
acp
acq
Pp(n)=0;Qq(n)=0;
for i=2:n
    for j=1:n
        Pp(i)=Pp(i)+mag(i)*my(i,j)*mag(j)*cos(an(i,j)-th(i)+th(j));
        Qq(i)=Qq(i)-mag(i)*my(i,j)*mag(j)*sin(an(i,j)-th(i)+th(j));
    end
end
Pp
Qq
for i=2:n
    for j=2:n
        if i~=j
            j1(i,j)=mag(i)*mag(j)*(g(i,j)*sin(th(i)-th(j))-b(i,j)*cos(th(i)-th(j)));
            j3(i,j)=-mag(i)*mag(j)*(g(i,j)*cos(th(i)-th(j))+b(i,j)*sin(th(i)-th(j)));
```

```
            j2(i,j)=-j3(i,j);
            j4(i,j)=j1(i,j);
        else
            j1(i,j)=-Qq(i)-b(i,j)*(mag(i)^2);
            j2(i,j)=Pp(i)+g(i,j)*(mag(i)^2);
            j3(i,j)=Pp(i)-g(i,j)*(mag(i)^2);
            j4(i,j)=Qq(i)-b(i,j)*(mag(i)^2);
        end
    end
end
ja1(1:n-1,1:n-1)=j1(2:n,2:n);
ja2(1:n-1,1:n-1)=j2(2:n,2:n);
ja3(1:n-1,1:n-1)=j3(2:n,2:n);
ja4(1:n-1,1:n-1)=j4(2:n,2:n);
jacob=[ja1 ja2;ja3 ja4]
delp(1:n-1)=acp(2:n)-Pp(2:n);
delq(1:n-1)=acq(2)-Qq(2);
Char=inv(jacob)*[delp delq]';
Chth(2:n)=Char(1:n-1);
Chmag(2:n)=Char(n:2*n-2);
mag=mag+Chmag;
th=th+Chth;
fprintf('the voltage values for buses');
mag
fprintf('The angle values for buses');
th
```

Result: