Comparison between Load Flow Analysis Methods in Power System using MATLAB

Kriti Singhal

Abstract—Now these days load flow is a very important and fundamental tool for the analysis of any power systems and in the operations as well as planning stages. Certain applications, particularly in distribution automation and optimization of a power system, require repeated load flow solutions. In these applications it is very important to solve the load flow problem as efficiently as possible. Since the invention and widespread use of digital computers and many methods for solving the load flow problem have been developed. Most of the methods have "grown" up around transmission systems and, over the years, variations of the Newton method such as the fast decoupled method; have become the most widely used. Some of the methods based on the general meshed topology of a typical transmission system are also applicable to distribution systems which typically have a radial or tree structure. Specifically, we will compare the proposed method to the standard Newton Method, and the implicit Zbus Gauss method.

Our goal is to develop a formulation and solutions algorithm for solving load flow in large 3-phase unbalanced systems which exploits the radial topological structure to reduce the number of equations and unknowns and the numerical structure by using Gauss-Seidel Method, Newton Raphson Method and Fast Decoupled Method.


1 INTRODUCTION

Load flow study also known as power flow study, is an important tool involving numerical analysis applied to a power system. A power-flow study usually uses simplified notation such as a one-line diagram and per-unit system, and focuses on various forms of AC power (i.e.: voltages, voltage angles, real power and reactive power). Load-flow studies are performed to determine the steady-state operation of an electric power system. It calculates the voltage drop on each feeder, the voltage at each bus, and the power flow in all branch and feeder circuits. Determine if system voltages remain within specified limits under various contingency conditions, and whether equipment such as transformers and conductors are overloaded. It is used to identify the need for additional generation, capacitive, or inductive support, or the placement of capacitors and/or reactors to maintain system voltages within specified limits. Losses in each branch and total system power losses are also calculated. It is Necessary for planning, economic scheduling, and control of an existing system as well as planning its future expansion.

There are two popular numerical methods for solving the power-flow equations. These are the Gauss-Seidel (G-S) and the Newton-Raphson (N-R) Methods (Grainger and Stevenson, 1994; Elgend, 1982; Glover and Sharma, 1994). The N-R method is superior to the G-S method because it exhibits faster convergence characteristics. However, the N-R method suffers from the disadvantages that a “flat start” is not always possible since the solution at the beginning can oscillate without converging towards the solution. In order to avoid this problem, the load-flow solution is often started with a G-S algorithm followed by the N-R algorithm after a few iterations. There is also an approximate but faster method for the load-flow solution. It is a variation of the N-R method, called the faster-decoupled method, which was introduced.

2 POWER FLOW PROBLEM FORMULATION

The goal of a power flow study is to obtain complete voltage angle and magnitude information for each bus in a power system for specified load and generator real power and voltage conditions. Once this information is known, real and reactive power flow on each branch as well as generator reactive power output can be analytically determined.

The solution to the power flow problem begins with identifying the known and unknown variables in the system. The known and unknown variables are dependent on the type of bus. A bus without any generators connected to it is called a Load Bus. A bus with at least one generator connected to it is called a Generator Bus. The exception is one arbitrarily-selected bus that has a generator. This bus is referred to as the slack bus.

In the power flow problem, if the real power and reactive power at each Load Bus are known. For this reason, Load Buses are also known as PQ Buses. For Generator Buses, it is assumed that the real power generated P and the voltage magnitude |V| is known. For the Slack Bus, it is assumed that the voltage magnitude |V| and voltage phase Θ are known. Therefore, for each Load Bus, the voltage magnitude and angle are unknown and must be solved for; for each Generator Bus, the voltage angle must be solved for; there are no variables that must be solved for the Slack Bus. In a system with N buses and R generators, there are then
In order to solve for the above equation, the possible equations to use are power balance equations, which can be written for real and reactive power for each bus. The real power balance equation is:

\[ \sum_{k=0}^{n} y_{ik} v_i v_k + y_{ik} \sum_{k=0}^{n} y_{ik} v_i v_k = P_i \]

Where, \( P_i \) is the net power injected at bus \( i \), \( y_{ik} \) is the imaginary part of the element in the bus admittance matrix, \( Y_{bus} \) corresponding to the \( i \)th row and \( k \)th column, \( \theta_{ik} \) is the imaginary part of the element.

The reactive power balance equation is:

\[ \sum_{k=0}^{n} y_{ik} v_i v_k - y_{ik} \sum_{k=0}^{n} y_{ik} v_i v_k = Q_i \]

Where, \( Q_i \) is the net reactive power injected at bus \( i \), and \( \theta_{ik} = \delta_i - \delta_k \).

Equations included are the real and reactive power balance equations for each Load Bus and the real power balance equation for each Generator Bus. Only the real power balance equation is written for a Generator Bus because the net reactive power injected is not assumed to be known and therefore including the reactive power balance equation would result in an additional unknown variable. For similar reasons, there are no equations written for the Slack Bus.

Fig. 1- Power System Analysis.

3 POWER FLOW EQUATION

Applying KCL to this bus results in:

\[ I_i = y_{i0} v_i + y_{i1} (v_{i1} - v_1) + y_{i2} (v_{i2} - v_2) + ... + y_{in} (v_{in} - v_n) \]

Representing eq. (1) in summation form;

\[ I_i = V_i \sum_{j=0}^{n} y_{ij} - \sum_{j=0}^{n} y_{ij} V_j \quad j \neq i \]

The complex power at its \( i \)th bus is;

\[ P_i + jQ_i = V_i I_i \]

This is for reactive power.

\[ I_i = \frac{P_i - jQ_i}{V_i} \]

This is for active power.

Substituting for \( I_i \) in (2) yields;

\[ \frac{P_i - jQ_i}{V_i} = V_i \sum_{j=0}^{n} y_{ij} - \sum_{j=0}^{n} y_{ij} V_j \quad j \neq i \]

Equation (5) is an algebraic non linear equation which must be solved by iterative techniques. We use Methods like Gauss-Seidel, Newton Raphson and Fast Decoupled Method for further solution[2].

4 GAUSS-SEIDEL METHOD

The Gauss–Seidel method, also known as the Liebmann method or the method of successive displacement, is an iterative method used to solve a linear system of equations. It is named after the German mathematicians Carl Friedrich Gauss and Philipp Ludwig von Seidel, and is similar to the Jacobi method.

4.1 For P-Q bus

Equation (5) is solved for \( V_i \) solved iteratively;

\[ V_i^{(k+1)} = \frac{P_i^{(k)} + jQ_i^{(k)} + \sum_{j=0}^{n} y_{ij} V_j^{(k)}}{\sum_{j=0}^{n} y_{ij}} \quad j \neq i \]

For Generator buses or P-V bus (where real and reactive powers are injected), \( P_i^{(k)} \) and \( Q_i^{(k)} \) have positive values. Load buses or P-Q bus (real and reactive powers flow away from the bus), \( P_i^{(k)} \) and \( Q_i^{(k)} \) have negative values.

Eq (5) can be solved for \( P_i \) and \( Q_i \):

\[ P_j^{(k+1)} = R \left[ v_j^{(k)} \left( v_j^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{j=0}^{n} y_{ij} v_j^{(k)} \right) \right] \quad j \neq i \]

\[ Q_j^{(k+1)} = -I m \left[ v_j^{(k)} \left( v_j^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{j=0}^{n} y_{ij} v_j^{(k)} \right) \right] \quad j \neq i \]

The power flow equation is usually expressed in terms of the elements of the bus admittance matrix, \( Y_{bus} \), shown by upper case letter, are \( Y_i = -y_{ii} \), and the diagonal elements are \( Y_i = \sum y_{ii} \).

Hence eq (6), (7) and (8) can be written as;
Equation (5) is solved until $V_{i}^{k+1} - V_{i}^{k} < 0.0001$.

### 4.2 For P-V bus
Firstly $Q_{i}^{k+1}$ is calculated from eq (11);

$$Q_{i}^{(k+1)} = -\text{Im} \left( V_{i}^{(k)} Y_{ii}^{(k)} + \sum_{j=1, j \neq i}^{n} Y_{ij}^{(k)} V_{j}^{(k)} \right) \quad j \neq i$$

For a P-V bus the upper limit $Q_{\text{max}}$ and lower limit $Q_{\text{min}}$ of $Q$ to hold the generators vars within limits is also given. That is $Q_{\text{min}} < Q_{i} < Q_{\text{max}}$. Therefore, the calculated $Q_{i}^{(k+1)}$ is checked for limits of $Q_{i}$, that is

$$Q_{\text{min}} < Q_{i}^{(k+1)} < Q_{\text{max}}$$

If $Q_{i}^{(k+1)}$ is within its limits then calculate new value of $C_{k}$ and $V_{i}^{(k+1)}$ from eq (6) and treat the ith bus as P-Q bus. The computation is then continued as in case of a P-Q bus.[2]

### 4.3 Flow Diagram

![Flow Diagram](image-url)

Fig 3- Flow chart of Gauss-Seidel Method[3].
5 NEWTON RAPHSON METHOD

If you’ve ever tried to find a root of a complicated function algebraically, you may have had some difficulty. Using some basic concepts of calculus, we have ways of numerically evaluating roots of complicated functions. Commonly, we use the Newton-Raphson method. This iterative process follows a set guideline to approximate one root, considering the function, its derivative, and an initial x-value.

Power flow equations formulated in polar form. For the system in Fig 2, Eqn (2) can be written in terms of bus admittance matrix as:[2]

\[ I_i = \sum_{j=1}^{n} Y_{ij} V_j \]  

Expressing in polar form;

\[ I_i = \sum_{j=1}^{n} |V_j| \angle \theta_{ij} + \delta_j \]  

Substituting for I, from Eqn (21) in Eqn (4);

\[ P_i - jQ_i = |V_i| \angle - \delta_i \sum_{j=1}^{n} |V_j| \angle \theta_{ij} + \delta_j \]  

Separating the real and imaginary parts;

\[ P_i = \sum_{j=1}^{n} |V_j||V_j| \cos(\theta_{ij} - \delta_i + \delta_j) \]  

\[ Q_i = -\sum_{j=1}^{n} |V_j||V_j| \sin(\theta_{ij} - \delta_i + \delta_j) \]  

\( \delta_j \) is phase angle.

Expanding Eqn (15) & (16) in Taylor’s series about the initial estimate neglecting high order terms we get;

\[
\begin{bmatrix}
\Delta P_{i}^{(k)} \\
\Delta Q_{i}^{(k)}
\end{bmatrix}
= 
\begin{bmatrix}
J_{11} & J_{12} \\
J_{21} & J_{22}
\end{bmatrix}
\begin{bmatrix}
\Delta \delta \\
\Delta \delta
\end{bmatrix}
\]

\[ \Delta Q_{i}^{(k)} = \Delta Q_{1}^{(k)} - \Delta Q_{2}^{(k)} \]  

Continue until scheduled errors \( \Delta P_{i}^{(k)} \) and \( \Delta Q_{i}^{(k)} \) for all load buses are within a specified tolerance.

5.1 Figures and Tables

![Flow chart of Newton Raphson Method][3]

6 FAST DECOUPLED METHOD

Fast decoupled load flow method is a variation on Newton-Raphson that exploits the approximate decoupling of active and reactive flows in well-behaved power networks, and additionally fixes the value of the Jacobian during the iteration in order to avoid costly matrix decompositions.

In this power transmission lines have high X/R ratio. Real power changes are less sensitive to voltage magnitude changes and are most sensitive to changes in phase angle \( \Delta \delta \). Similarly, reactive power changes are less sensitive to changes in angle and are mainly dependent on changes in voltage magnitude. Therefore the Jacobian matrix in Equation (17) can be written as;

\[
\begin{bmatrix}
\Delta P_{i}^{(k)} \\
\Delta Q_{i}^{(k)}
\end{bmatrix}
= 
\begin{bmatrix}

\frac{\partial P_i^{(k)}}{\partial \delta_j} \\
\frac{\partial Q_i^{(k)}}{\partial \delta_j}
\end{bmatrix}
\begin{bmatrix}
\Delta P_{i}^{(k)} \\
\Delta Q_{i}^{(k)}
\end{bmatrix}
\]

\[ \Delta P_i^{(k)} = P_i^{sch} - P_i^{(k)} \]
\[ \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & 0 \\ 0 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix} \]  
(22)

\[ \Delta P = J_1 \Delta \delta = \begin{bmatrix} \frac{\partial P}{\partial \delta} \end{bmatrix} \Delta \delta \]  
(23)

\[ \Delta Q = J_4 \Delta |V| = \begin{bmatrix} \frac{\partial Q}{\partial |V|} \end{bmatrix} \Delta |V| \]  
(24)

The diagonal elements of J1 given by Eqn.(18) is written as;

\[ \delta \Delta \delta \begin{bmatrix} \delta \delta \delta \begin{bmatrix} \delta \delta \delta \end{bmatrix} \end{bmatrix} \]  
(22)

\[ \delta \delta \delta \begin{bmatrix} \delta \delta \delta \end{bmatrix} \]  
(23)

\[ \delta \delta \delta \begin{bmatrix} \delta \delta \delta \end{bmatrix} \]  
(24)

The diagonal elements of J1 given by Eqn.(18) is written as;

\[ \sum_{j=1}^{n} \left| V'_i \right| \left| V'_j \right| \cos(\theta_{ij} - \delta_i + \delta_j) - \left| V'_i \right| \left| V'_j \right| \sin \theta_{ij} \]

Replacing the first term with –Qi from (19)

\[ \frac{\partial P_i}{\partial \delta_i} = -Q_i \left| V'_i \right|^2 |\theta_{ii}| \sin \theta_{ii} - B_o \]

B_o = sum of susceptances of the entire elements incident to bus i. In a typical power system, B_o ≫ Q_i. Therefore we may neglect Q_i.

Furthermore, [Vi]² = [Vi] Ultimately;

\[ \frac{\partial P_i}{\partial \delta_i} = - \left| V'_i \right| B_o \]  
(25)

In equation (19) assuming \( \delta_{ii} - \delta_i + \delta_j \approx \delta_{ii} \), the off diagonal elements of J1 becomes,

\[ \frac{\partial P_i}{\partial \delta_j} = - \left| V'_i \right| \left| V'_j \right| B_o \]

Assuming [Vi] ≈ 1 we get;

\[ \frac{\partial P_i}{\partial \delta_j} = - \left| V'_j \right| B_o \]  
(26)

Similarly we can simplify the diagonal and off-diagonal elements of J4 as,

\[ \frac{\partial Q_i}{\partial |V'}_i |} = - \left| V'_i \right| B_o \]

With these assumptions, equations (25) & (26) can be written in the following form,

\[ \Delta P \left| V'_i \right| = -B' \Delta \delta \]  
(27)

\[ \Delta Q \left| V'_i \right| = -B'' \Delta |V'_i| \]  
(28)

B’ and B’’ are the imaginary part of the bus admittance matrix \( Y_{bus} \). Since the elements of the matrix are constant, need to be triangularized and inverted only once at the beginning of the iteration[2].

7 COADING IMPLEMENTATION

clc
clear all
close all
x=[0.02+1i*0.10 0.05+1i*0.25 0.04+1i*0.20 0.05+1i*0.25 0.05+1i*0.25 0.08+1i*0.40 0.10+1i*0.50];
X=1./x;
Y=zeros(5);
Y(1,2)=X(1)+1i*0.030;
Y(1,5)=X(2)+1i*0.020;
Y(2,3)=X(3)+1i*0.025;
Y(2,5)=X(4)+1i*0.020;
Y(3,4)=X(5)+1i*0.010;
Y(3,5)=X(6)+1i*0.010;
Y(4,5)=X(7)+1i*0.075;
Y(1,1)=Y(1,2)+Y(1,5);
Y(1,3)=0;
Y(1,4)=0;
Y(2,1)=Y(1,2);
Y(2,2)=Y(1,2)+Y(2,3);
Y(2,4)=0;
Y(3,1)=0;
Y(3,2)=Y(2,3);
Y(3,3)=Y(2,3)+Y(3,4)+Y(3,5);
Y(4,1)=0;
Y(4,2)=0;
Y(4,3)=Y(3,4);
Y(4,4)=Y(3,4)+Y(4,5);
Y(5,1)=Y(1,5);
Y(5,2)=Y(2,5);
Y(5,3)=Y(3,5);
Y(5,4)=Y(4,5);
Y(5,5)=Y(1,5)+Y(2,5)+Y(3,5)+Y(4,5);
[r,c]=size(Y);
Y1=Y;
for i=1:r
    for j=1:c
        if i~=j
            Y(i,j)=-Y(i,j);
        end
    end
end
theta=angle(Y);
del_tolerance=1;
V=[1.05 1 1 1 1.02];
del=[0 0 0 0 0];
sp=0;
sq=0;
for k=1:5
    for n=1:5
        sp=sp+abs(V(k)*V(n)*Y(k,n))*cos(theta(k,n)+del(n)-del(k));
    end
    P(k)=sp;
P(5)=24;
end
figure('color',[0 .5 1],'menubar','none')
plot(1:5,P ,'color','r','marker','.','markeredgecolor','k','linewidth',2)
hold on
for k=1:5
    for n=1:5
        sq=(sq+abs(V(k)*V(n)*Y(k,n))*sin(theta(k,n)+del(n)-del(k)))
    end
    Q(k)=sq;
    Q(5)=11;
end
plot(1:5,Q,'color','k','marker','.','markeredgecolor','r','linewidth',2)
grid on
xlabel('Number of buses');
ylabel('Powers')
legend('Real Power','Reactive power','location','best')

%Load and Generator
PsL=[0 96 35 16 24];
QsL=[0 62 14 8 11];
PsG=[NaN 0 0 0 NaN];
P2sch=(PsG(2)-PsL(2))/100;
P3sch=(PsG(3)-PsL(3))/100;
P4sch=(PsG(4)-PsL(4))/100;
P5sch=(PsG(5)-PsL(5))/100;
Q2sch=(QsG(2)-QsL(2))/100;
Q3sch=(QsG(3)-QsL(3))/100;
Q4sch=(QsG(4)-QsL(4))/100;

%mismatching in power calculations
DelP2=P2sch-P(2);
DelP3=P3sch-P(3);
DelP4=P4sch-P(4);
DelP5=P5sch-P(5);
DelQ2=Q2sch-Q(2);
DelQ3=Q3sch-Q(3);
DelQ4=Q4sch-Q(4);
final_vector=[DelP2 DelP3 DelP4 DelP5 DelQ2 DelQ3 DelQ4];

%% formation of Jacobian
s=0;
idx=0;
t=0;
t1=0;
t2=0;
for k=2:5
    for n=2:5
        if k==n
            for N=[1 k+1:5]
                s=s+abs(V(k)*V(N)*Y(k,N))*sin(theta(k,N)+del(N)-
                del(k));
            end
            JPD(k-1,n-1)=s;
            s=0;
        elseif k~=n
            JPD(k-1,n-1)=-abs(V(k)*V(n)*Y(k,n))*sin(theta(k,n)-
                del(k)+del(n));
        end
        if n>=3
            if k==n
                for N=[1 k+1:5]
                    t1=t1+abs(V(k)*V(N)*Y(k,N))*cos(theta(k,N)+del(N)-
                    del(k));
                end
                JQD(k-2,n-1)=t1;
                t1=0;
            elseif k~n
                JQD(k-2,n-1)=-abs(V(k)*V(n)*Y(k,n))*cos(theta(k,n)-
                    del(k)+del(n));
            end
        end
        if k>=3 && n>=3
            if k==n
                for N=[1 k+1:5]
                    t2=t2+abs(V(k)*V(N)*Y(k,N))*cos(theta(k,N)-
                    del(k)+del(n));
                end
                JQV(k-2,n-2)=t2;
                t2=0;
            elseif k~n
                JQV(k-2,n-2)=-abs(V(k)*V(n)*Y(k,n))*sin(theta(k,n)+
                    del(k)+del(n));
            end
        end
        if k>=3 && n>=3
            if k==n
                for N=[1 k+1:5]
                    t3=t3+abs(V(k)*V(N)*Y(k,N))*sin(theta(k,N)-
                    del(k)+del(n));
                end
                JQV(k-2,n-2)=t3;
                t3=0;
            elseif k~n
                JQV(k-2,n-2)=-abs(V(k)*V(n)*Y(k,n))*sin(theta(k,n)+
                    del(k)+del(n));
            end
        end
    end
    end
    final_jacobian=[JPD JPV;JQD JQV];
    jac_inv=inv(final_jacobian)*final_vector';
    count=2;
    for kk=1:length(jac_inv)
        if kk<=4
            Delv(kk)=jac_inv(kk);
        elseif kk>4
            count=count+1;
            Delv(count)=jac_inv(kk);
        end
    end
    del_tolerance=max(abs(Del-del));
    del=Del+del;
    V1=Delv+V;
    f=find(V1>2);
    f1=find(V1<1);
    V1(f)=rand(size(f));
    V1(f1)=rand(size(f1));

%% Gauss seidel method
P2sch=(PsG(2)-PsL(2))/100;
P3sch=(PsG(3)-PsL(3))/100;
P4sch=(PsG(4)-PsL(4))/100;
Q2sch=(QsG(2)-QsL(2))/100;
Q3sch=(QsG(3)-QsL(3))/100;
Q4sch=(QsG(4)-QsL(4))/100;
V2=(1/4)*(P2sch+1i*Q2sch)/(V2-(Y(2,1)*V1)-
    (Y(2,2)*V(3))-Y(2,4)*V(4)-(Y(2,5)*V(5)));
V3=(1/4)*(P3sch+1i*Q3sch)/(V3-(Y(3,1)*V1)-
    (Y(3,2)*V(2))-Y(3,4)*V(4)-(Y(3,5)*V(5)));
V4=(1/4)*(P4sch+1i*Q4sch)/(V4-(Y(4,1)*V1)-
    (Y(4,2)*V(2))-Y(4,3)*V(3)-(Y(4,5)*V(5)));

%% Updating reactive power on PV bus
...
8 RESULT

8.1 Comparison between Real power and Reactive Power.

8.2 Comparison between Newton Raphson and Gauss-Seidel.

9 CONCLUSION

Analyses, designing and comparison between different load flow system solving techniques i.e. Gauss-Seidel Method, Newton-Raphson Method in Power System using MATLAB has been successfully done and observed the desired result. In Gauss-Seidel, rate of convergence is slow. It can be easily program and the number of iterations increases directly with the number of buses in the system and in Newton-Raphson, the convergence is very fast and the number of iterations is independent of the size of the system, solution to a high accuracy is obtained. The NR Method convergence is not sensitive to the choice of slack bus. Although a large number of load flow methods are available in literature it has been observed that only the Newton-Raphson and the Fast Decoupled load-flow methods are most popular. The fast decoupled load flow is definitely superior to the Newton-Raphson Method from the point of view of speed and storage.

10 END SECTIONS

10.1 Acknowledgment

It is my proud privilege to express my sincere thanks to my supervisor, Mr. Sanjiv Kumar, Assistant Professor, Department of Electrical & Electronics Engineering, for valuable guidance and constant encouragement throughout the duration of the dissertation. My sincere thanks goes to my another guide Mr. Sajid Ali, Incharge Project & Seminar, EN, for his valuable suggestions and encouragement throughout the duration of my work and also giving an opportunity to work on MATLAB. I am also thankful to my fellow classmates and friends for their cooperation and moral support during my stay. Finally, my heartiest gratitude goes to my Parents, Brother,
Sister, etc., for their encouraging support as well as financial support throughout my studies which allowed me to concentrate on my work. They have always been a great role model and lead me to greater success in pursuit of higher knowledge.

10.2 References