Determination of Power Flow in PQ 5 Bus System

A Project report submitted in partial fulfillment
of the requirements for the degree of B. Tech in Electrical Engineering

by

MAYUKH MUKHERJEE (11701614026)
SOMSURYA SENGUPTA (11701614045)
SOHOM OJHA (11701614044)

Under the supervision of

Mr. Dipankar Santra
Associate Professor
Department of Electrical Engineering

श्रमम् बिना न किमपि साध्यम्

Department of Electrical Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015, WEST BENGAL Maulana Abul Kalam Azad University of Technology (MAKAUT)

CERTIFICATE

To whom it may concern

This is to certify that the project work entitled **Determination of Power Flow in PQ 5 Bus System** is the bonafide work carried out by **MAYUKH MUKHERJEE (11701614026), SOMSURYA SENGUPTA(11701614045), SOHOM OJHA(11701614044)**the students of B.Tech in the Department of Electrical Engineering, RCC Institute of Information Technology (RCCIIT), Canal South Road, Beliaghata, Kolkata-700015, affiliated to Maulana Abul Kalam Azad University of Technology (MAKAUT), West Bengal, India, during the academic year 2017-18, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Electrical Engineering.

____

(Dipankar Santra)

Assistant Professor

Department of Electrical Engineering

RCC Institute of Information Technology

Countersigned by

____

(Dr. Alok Kole)

HOD, Electrical Engineering Dept

RCC Institute of Information Technology

____

(External Examiner)

ACKNOWLEDGEMENT

It is our great fortune that we have got opportunity to carry out this project work under the supervision of **Mr. Dipankar Santra** in the Department of Electrical Engineering, RCC Institute of Information Technology (RCCIIT), Canal South Road, Beliaghata, Kolkata-700015, affiliated to Maulana Abul Kalam Azad University of Technology (**MAKAUT**), West Bengal, India. We express our sincere thanks and deepest sense of gratitude to our guide for his constant support, unparalleled guidance and limitless encouragement.

We would also like to convey our gratitude to all the faculty members and staffs of the Department of Electrical Engineering, RCCIIT for their whole-hearted cooperation to make this work turn into reality.

We are very thankful to our Department and to the authority of RCCIIT for providing all kinds of infrastructural facility towards the research work.

Thanks to the fellow members of our group for working as a team.

MAYUKH MUKHERJEE (11701614026)

SOMSURYA SENGUPTA (11701614045)

SOHOM OJHA (11701614044)

To

The Head of the Department
Department of Electrical Engineering
RCC Institute of Information Technology
Canal South Rd. Beliagahata, Kolkata-700015

Respected Sir,

In accordance with the requirements of the degree of Bachelor of Technology in the Department of Electrical Engineering, RCC Institute of Information Technology, we present the following thesis entitled "**DETERMINATION OF POWER FLOW IN PQ 5 BUS SYSTEM**". This work was performed under the valuable guidance of Mr. Dipankar Santra, Assistant Professor in the Dept. of Electrical Engineering.

Yours Sincerely,

MAYUKH MUKHERJEE (11701614026)

SOMSURYA SENGUPTA (11701614045)

SOHOM OJHA (11701614044)

Contents

List of Figures

List of Tables:

Abbreviations &acronym

indx	the number of iterations
v	bus voltages in Cartesian form
abs(v)	magnitude of bus voltages
angle(v)/d2r	angle of bus voltage in degree
preal	real power in MW
preac	reactive power in MVAr
pwr	power flow in the various line segments
qwr	reactive power flow in the various line segments
q	reactive power entering or leaving a bus
pl	real power losses in various line segments
ql	reactive drops in various line segments
del	tolerance
pcal	real power calculated
qcal	reactive power calculated

ngn =number of generators

genbus = generator bus

number

PGEN = scheduled active power contributed by the generator

QGEN = scheduled reactive power contributed by the generator

QMAX = generator reactive power

upper limit QMIN = generator reactive

power lower limit ngn = 2 ;

ABSTRACT

The goal of this thesis is to do a performance analysis on numerical methods including Newton-Raphson method, Gauss-Seidel method for a load flow run to achieve less run time. Unlike the proposed method, the traditional load flow analysis uses only one numerical method at a time.
These algorithms perform all the computation for finding the bus voltage angles and magnitudes, real and reactive powers for the given generation and load values, while keeping track of the proximity to convergence of a solution. This work focuses on finding the most effective algorithm.
The convergence time is compared among the methods. The proposed method is implemented on a 5-bus system and 9 bus system with different contingencies and the solutions obtained are verified with MATLAB a commercial software for load flow analysis.

Key Words: Power flow, Gauss-Seidel method, Newton-Raphson method, convergence time.

CHAPTER
1

(INTRODUCTION TO LOAD FLOW)

Introduction

Concept of Power Flow:

The power flow analysis is a very important tool in power system analysis. Power flow studies are routinely used in planning, control, and operations of existing electric power systems as well as planning for future expansion. The successful operation of power systems depends upon knowing the effects of adding interconnections, adding new loads, connecting new generators or connecting new transmission line before it is installed. 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.

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, 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 can be analytically determined. Due to the nonlinear nature of the problem, numerical methods are employed to obtain solution that is within an acceptable tolerance.

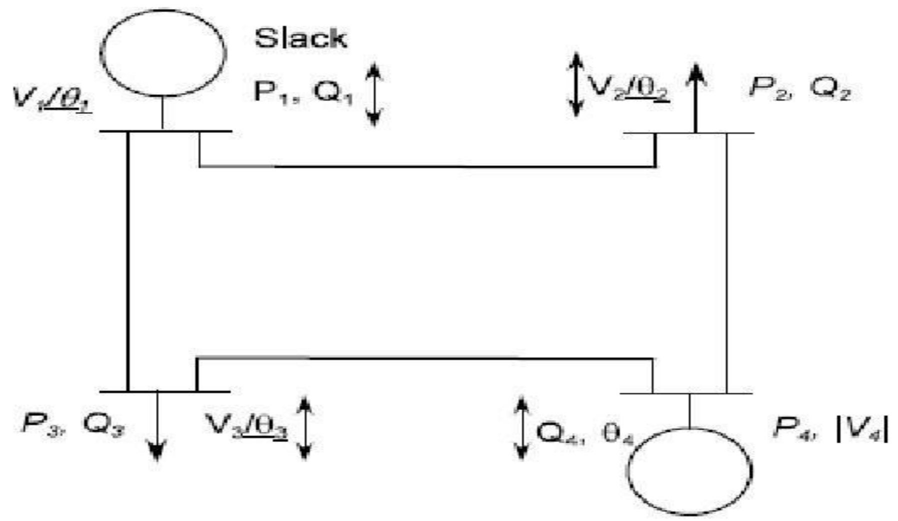


Figure 1 PQ bus system

Load flow studies are one of the most important aspects of power system planning and operation. The load flow gives us the sinusoidal steady state of the entire system – voltages, real and reactive power generated and absorbed and line losses. Since the load is a static quantity and it is the power that flows through transmission lines, the purists prefer to call this Power Flow studies rather than load flow studies. We shall however stick to the original nomenclature of load flow.

Through the load flow studies, we can obtain the voltage magnitudes and angles at each bus in the steady state. This is rather important as the magnitudes of the bus voltages are required to be held within a specified limit. Once the bus voltage magnitudes and their angles are computed using the load flow, the real and reactive power flow through each line can be computed. Also based on the difference between power flow in the sending and receiving ends, the losses in a particular line can also be computed. Furthermore, from the line flow we can also determine the over and under load conditions.

The steady state power and reactive powers supplied by a bus in a power network are expressed in terms of nonlinear algebraic equations. We therefore would require iterative methods for solving these equations. In this chapter we shall discuss two of the load flow methods. We shall also delineate how to interpret the load flow results.

OBJECTIVES OF LOAD FLOW

- Power flow analysis is very important in planning stages of new networks or addition to existing ones like adding new generator sites, meeting increase load demand and locating new transmission sites.

- The load flow solution gives the nodal voltages and phase angles and hence the power injection at all the buses and power flows through interconnecting power channels.
- It is helpful in determining the best location as well as optimal capacity of proposed generating station, substation and new lines.
- It determines the voltage of the buses. The voltage level at the certain buses must be kept within the closed tolerances.
- System transmission loss minimizes.
- Economic system operation with respect to fuel cost to generate all the power needed
- The line flows can be known. The line should not be overloaded, it means, we should not operate the close to their stability or thermal limits.

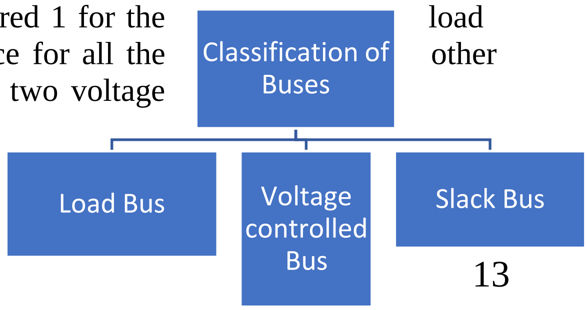
BUS CLASSIFICATION

For load flow studies it is assumed that the loads are constant and they are defined by their real and reactive power consumption. It is further assumed that the generator terminal voltages are tightly regulated and therefore are constant. The main objective of the load flow is to find the voltage magnitude of each bus and its angle when the powers generated and loads are pre-specified. To facilitate this, we classify the different buses of the power system as listed below.

1. Load Buses: In these buses no generators are connected and hence the generated real power P_{Gi} and reactive power Q_{Gi} are taken as zero. The load drawn by these buses are defined by real power $-P_{Li}$ and reactive power $-Q_{Li}$ in which the negative sign accommodates for the power flowing out of the bus. This is why these buses are sometimes referred to as P-Q bus. The objective of the load flow is to find the bus voltage magnitude $|V_i|$ and its angle δ_i.

2. Voltage Controlled Buses: These are the buses where generators are connected. Therefore, the power generation in such buses is controlled through a prime mover while the terminal voltage is controlled through the generator excitation. Keeping the input power constant through turbine-governor control and keeping the bus voltage constant using automatic voltage regulator, we can specify constant P_{Gi} and $|V_i|$ for these buses. This is why such buses are also referred to as P-V buses. It is to be noted that the reactive power supplied by the generator Q_{Gi} depends on the system configuration and cannot be specified in advance. Furthermore, we have to find the unknown angle δ_i of the bus voltage.

3. Slack or Swing Bus: Usually this bus is numbered 1 for the load flow studies. This bus sets the angular reference for all the other buses. Since it is the angle difference between two voltage sources that dictates the real and reactive power flow between them, the particular angle of the slack bus is not important. However, it



sets the reference against which angles of all the other bus voltages are measured. For this reason, the angle of this bus is usually chosen as 0°. Furthermore, it is assumed that the magnitude of the voltage of this bus is known.

Figure 2 Bus classification

Now consider a typical load flow problem in which all the load demands are known. Even if the generation matches the sum total of these demands

Figure 3

Type of Bus	Known Quantities	Quantities to be specified

exactly, the mismatch between generation and load will persist because of the line I^2R losses. Since the I^2R loss of a line depends on the line current which, in turn, depends on the magnitudes and angles of voltages of the two buses connected to the line, it is rather difficult to estimate the loss without calculating the voltages and angles. For this reason a generator bus is usually chosen as the slack bus without specifying its real power. It is assumed that the generator connected to this bus will supply the balance of the real power required and the line losses.

OVERVIEW

The power flow algorithm written in this thesis is based on the Gauss-Seidel method and Newton-Raphson method.
A software program has been developed. The program gives the power flow solution for a given problem as well as
computes the complete voltage angle and magnitude information for each bus in a power system for specified load and generator real power and voltage conditions.

Generator bus	P, Q	|V|, d
Load bus	P, |V|	Q, d
Slack bus	|V|, d	P, Q

Power flow analysis is the backbone of power system analysis and design. They are necessary for planning, operation, economic scheduling and exchange of power between utilities. Power flow analysis is required for many other analyses such as transient stability, optimal power flow and contingency studies. The principal information of power flow analysis is to find the magnitude and phase angle of voltage at each bus and the real and reactive power flowing in each transmission lines.

Power flow analysis is an importance tool involving numerical analysis applied to a power system. In this analysis, iterative techniques are used due to there no known analytical method to solve the problem. This resulted nonlinear set of equations or called power flow equations are generated. To finish this analysis there are methods of mathematical calculations which consist plenty of step depend on the size of system. This process is difficult and takes much time to perform by hand. By develop a toolbox for power flow analysis surely will help the analysis become easier.

Power flow analysis software can help users to calculate the power flow problem. Over the past decade, a few versions of educational software packages using advanced programming languages, such as C, C++, Pascal, or FORTRAN have been developed for power engineering curriculums. These choose an integrated study platform with support of database and GUI functions.

Power flow analysis software develops by the author use MATLAB software. MATLAB as a high-performance language for technical computation integrates calculation, visualization and programming in an easy-to-use environment, thus becomes a standard instructional tool for introductory and advanced courses in mathematics, engineering and science in the university environment. Most of the students are familiar with it.

MATLAB is viewed by many users not only as a high-performance language for technical computing but also as a convenient environment for building graphical user interfaces (GUI). Data visualization and GUI design in MATLAB are based on the Handle Graphics System in which the objects organized in a Graphics Object Hierarchy can be manipulated by various high and low-level commands. If using MATLAB7 the GUI design more flexible and versatile, they also increase the complexity of the Handle Graphics System and require some effort to adapt to.

1.5 ORGANISATION OF THESIS

The thesis is organised into seven chapters including the chapter of introduction. Each chapter is different from the other and is described along with the necessary theory required to comprehend it.

Chapter 2 deals with the literature reviews. From this chapter we can see before our project who else works on this topic and how our project is different.

Chapter 3 deals with load flow studies or power flow concept, explained with necessary equations for power flow.

Chapter 4 deals with conventional methods of numerical methods to solve load flow problem.

Chapter 5 shows the MATLAB programs or software coding using the conventional methods.

Chapter 6 focuses on the comparison between gauss-seidel and newton Raphson method based on results.

Chapter 7 concludes about the load flow performance by the two conventional methods from the results obtained from MATLAB 2016a.

Chapter 8 references

Appendix A & B software coding and datasheets are listed here.

CHAPTER

2

(Literature Review)

LITERATURE REVIEW

Power flow analysis came into existence in the early 20th century. There were many research works done on the power flow analysis. In the beginning, the main aim of the power flow analysis was to find the solution irrespective of time.

Over the last 20 years, efforts have been expended in the research and development on the numerical techniques.

Before the invention of digital computers, the load flow solutions were obtained using the network analysis.

In the year 1956 the first practical automatic digital solution was found. The early generation computers were built with less memory storage, the Y-Bus matrix iterative method was well suitable for these computers. Although performance was satisfactory on many power flow problems, the time taken to convergence was very slow and sometimes they never converged.

In order to overcome the difficulties of this method a new method was developed based on the Z-Bus matrix.

This new method converges more reliably compared to the Y-Bus matrix method, but it requires more memory storage when solving large problems. During this time, the iterative methods were showing very powerful convergence properties but were difficult in terms of computation. In the mid 1960's major changes in the power system came with the development of very efficient sparsity programmed ordered elimination by Tinny.

CHAPTER 3

(Load Flow Studies)

REAL AND REACTIVE POWER INJECTED IN A BUS

For the formulation of the real and reactive power entering a bus, we need to define the following quantities. Let the voltage at the i^{th} bus be denoted by

$$V_i = |V_i|\angle\delta_i = |V_i|(\cos\delta_i + j\sin\delta_i) \tag{3.1}$$

Also let us define the self-admittance at bus-i as

$$Y_{ii} = |Y_{ii}|\angle\theta_{ii} = |Y_{ii}|(\cos\theta_{ii} + j\sin\theta_{ii}) = G_{ii} + jB_{ii} \tag{3.2}$$

Similarly, the mutual admittance between the buses i and j can be written as

$$Y_{ij} = |Y_{ij}|\angle\theta_{ij} = |Y_{ij}|(\cos\theta_{ij} + j\sin\theta_{ij}) = G_{ij} + jB_{ij} \tag{3.3}$$

Let the power system contains a total number of *n* buses. The current injected at bus-*i* is given as

$$\begin{aligned} I_i &= Y_{i1}V_1 + Y_{i2}V_2 + \cdots + Y_{in}V_n \\ &= \sum_{k=1}^{n} Y_{ik}V_k \end{aligned} \tag{3.4}$$

It is to be noted we shall assume the current entering a bus to be positive and that leaving the bus to be negative. As a consequence, the power and reactive power entering a bus will also be assumed to be positive. The complex power at bus-*i* is then given by

$$\begin{aligned} P_i - jQ_i &= V_i^* I_i = V_i^* \sum_{k=1}^{n} Y_{ik}V_k \\ &= |V_i|(\cos\delta_i - j\sin\delta_i)\sum_{k=1}^{n} |Y_{ik}V_k|(\cos\theta_{ik} + j\sin\theta_{ik})(\cos\delta_k + j\sin\delta_k) \\ &= \sum_{k=1}^{n} |Y_{ik}V_iV_k|(\cos\delta_i - j\sin\delta_i)(\cos\theta_{ik} + j\sin\theta_{ik})(\cos\delta_k + j\sin\delta_k) \end{aligned} \tag{3.5}$$

$$\begin{aligned} &(\cos\delta_i - j\sin\delta_i)(\cos\theta_{ik} + j\sin\theta_{ik})(\cos\delta_k + j\sin\delta_k) \\ &\quad = (\cos\delta_i - j\sin\delta_i)[\cos(\theta_{ik} + \delta_k) + j\sin(\theta_{ik} + \delta_k)] \\ &\quad = \cos(\theta_{ik} + \delta_k - \delta_i) + j\sin(\theta_{ik} + \delta_k - \delta_i) \end{aligned}$$

Therefore, substituting in (3.5) we get the real and reactive power as

$$P_i = \sum_{k=1}^{n} |Y_{ik}V_iV_k| \cos(\theta_{ik} + \delta_k - \delta_i) \tag{3.6}$$

$$Q_i = -\sum_{k=1}^{n} |Y_{ik}V_iV_k| \sin(\theta_{ik} + \delta_k - \delta_i) \tag{3.7}$$

PREPARATION OF DATA FOR LOAD FLOW

Let real and reactive power generated at bus-*i* be denoted by P_{Gi} and Q_{Gi} respectively. Also let us denote the real and reactive power consumed at the i^{th} bus by P_{Li} and Q_{Li} respectively. Then the net real power injected in bus-*i* is

$$P_{i,inj} = P_{Gi} - P_{Li} \tag{3.8}$$

Let the injected power calculated by the load flow program be $P_{i,calc}$. Then the mismatch between the actual injected and calculated values is given by

$$\Delta P_i = P_{i,inj} - P_{i,calc} = P_{Gi} - P_{Li} - P_{i,calc} \tag{3.9}$$

In a similar way the mismatch between the reactive power injected and calculated values is given by

$$\Delta Q_i = Q_{i,inj} - Q_{i,calc} = Q_{Gi} - Q_{Li} - Q_{i,calc} \tag{3.10}$$

The purpose of the load flow is to minimize the above two mismatches. It is to be noted that (3.6) and (3.7) are used for the calculation of real and reactive power in (3.9) and (3.10). However, since the magnitudes of all the voltages and their angles are not known a priori, an iterative procedure must be used to estimate the bus voltages and their angles in order to calculate the mismatches. It is expected that mismatches ΔP_i and ΔQ_i reduce with each iteration and the load flow is said to have converged when the mismatches of all the buses become less than a very small number.

For the load flow studies, we shall consider the system of Fig. 3.2, which has 2 generator and 3 load buses. We define bus-1 as the slack bus while taking bus-5 as the P-V bus. Buses 2, 3 and 4 are P-Q buses. The line impedances and the line charging admittances are given in Datasheet. Based on this data the Y_{bus} matrix is given in Table 3. It is to be noted here that the sources and their internal impedances are not considered while forming the Y_{bus} matrix for load flow studies which deal only with the bus voltages.

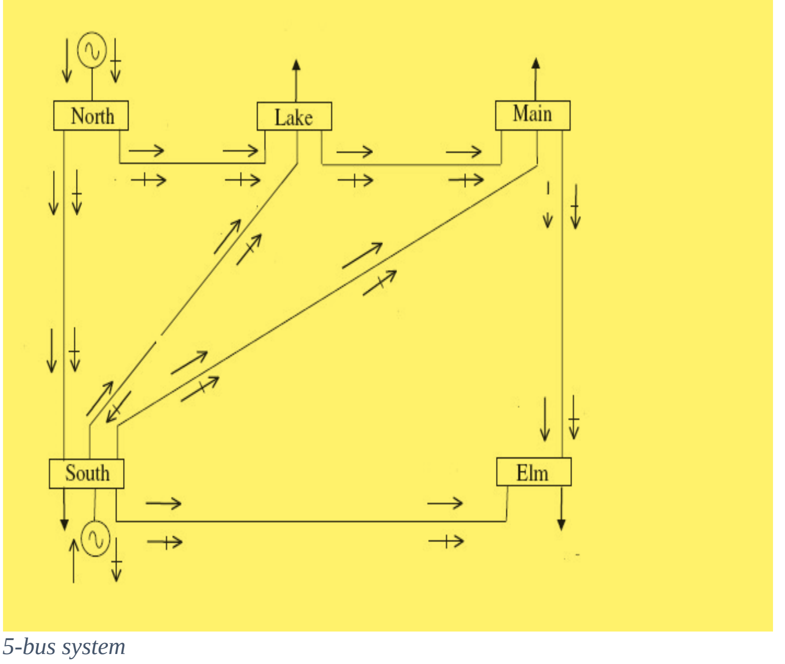


Figure 4 5-bus system

CHAPTER 4

Gauss-Seidel Method

Gauss-Seidel method is also known as the method of successive displacements.

To illustrate the technique, consider the solution of the nonlinear equation given by

$$F(x)=0 \quad (1)$$

Above function is rearrange and writes as

$$x=g(x) \quad (2)$$

If $x=(k)$ is an initial estimate of the variable x, the following iterative sequence is formed

$$X^{(k+1)}= g(x^{(k)}) \quad (3)$$

A solution is obtained when the difference between the absolute value of the successive iteration is less than a specified accuracy, i.e.,

$$| x^{(+k1)}- x^{(k)}|\leq \varepsilon \quad (4)$$

Where ε is the desire accuracy

The process is repeated until the change in variable is within the desired accuracy. So the Gauss-Seidel method needs much iteration to achieve the desired accuracy, and there is no guarantee for the convergence.

LOAD FLOW BY GAUSS-SEIDEL METHOD

The basic power flow equations (3.6) and (3.7) are nonlinear. In an n-bus power system, let the number of P-Q buses be n_p and the number of P-V (generator) buses be n_g such that $n = n_p + n_g + 1$. Both voltage magnitudes and angles of the P-Q buses and voltage angles of the P-V buses are unknown making a total number of $2n_p + n_g$ quantities to be determined. Amongst the known quantities are $2n_p$ numbers of real and reactive powers of the P-Q buses, $2n_g$ numbers of real powers and voltage magnitudes of the P-V buses and voltage magnitude and angle of the slack bus. Therefore, there are sufficient numbers of known quantities to obtain a solution of the load flow problem. However, it is rather difficult to obtain a set of closed form equations from (3.6) and (3.7). We therefore have to resort to obtain iterative solutions of the load flow problem.

At the beginning of an iterative method, a set of values for the unknown quantities are chosen. These are then updated at each iteration. The process continues till errors between all the known and actual quantities reduce below a pre-specified value. In the Gauss-Seidel load flow we denote the initial voltage of the i^{th} bus by $V_i^{(0)}$, $i = 2, \ldots, n$. This should read as the voltage of the i^{th} bus at the 0^{th} iteration, or initial guess. Similarly this voltage after the first iteration will be denoted by $V_i^{(1)}$. In this Gauss-Seidel load flow the load buses and voltage controlled, buses are treated differently. However, in both these types of buses we use the complex power equation given in (3.5) for updating the voltages. Knowing the real and reactive power injected at any bus we can expand (3.5) as

$$P_{i,inj} - jQ_{i,inj} = V_i^* \sum_{k=1}^{n} Y_{ik}V_k = V_i^*\left[Y_{i1}V_1 + Y_{i2}V_2 + \cdots + Y_{ii}V_i + \cdots + Y_{in}V_n\right] \quad (4.1)$$

We can rewrite (4.11) as

$$V_i = \frac{1}{Y_{ii}}\left[\frac{P_{i,inj} - jQ_{i,inj}}{V_i^*} - Y_{i1}V_1 - Y_{i2}V_2 - \cdots - Y_{in}V_n\right] \tag{4.2}$$

In this fashion the voltages of all the buses are updated. We shall outline this procedure with the help of the system of Fig. 4.1, with the system data given in Tables 4.1 to 4.3. It is to be noted that the real and reactive powers are given respectively in MW and MVAr. However, they are converted into per unit quantities where a base of 100 MVA is chosen.

Updating Load Bus Voltages

Let us start the procedure with bus-2. Since this is load bus, both the real and reactive power into this bus is known. We can therefore write from (4.12)

$$V_2^{(1)} = \frac{1}{Y_{22}}\left[\frac{P_{2,inj} - jQ_{2,inj}}{V_2^{*(0)}} - Y_{21}V_1 - Y_{23}V_3^{(0)} - Y_{24}V_4^{(0)} - Y_{25}V_5^{(0)}\right] \tag{4.3}$$

From the data given in Table 4.3 we can write

$$V_2^{(1)} = \frac{1}{Y_{22}}\left[\frac{-0.96 + j0.62}{1} - 1.05Y_{21} - Y_{23} - Y_{24} - 1.02Y_{25}\right]$$

It is to be noted that since the real and reactive power is drawn from this bus, both these quantities appear in the above equation with a negative sign. With the values of the Y_{bus} elements given in Table 4.2 we get $V_2^{(1)} = 0.9927\angle -2.5959°$.

The first iteration voltage of bus-3 is given by

$$V_3^{(1)} = \frac{1}{Y_{33}}\left[\frac{P_{3,inj} - jQ_{3,inj}}{V_3^{*(0)}} - Y_{31}V_1 - Y_{32}V_2^{(1)} - Y_{34}V_4^{(0)} - Y_{35}V_5^{(0)}\right] \tag{4.4}$$

Updating P-V Bus Voltages

It can be seen from Table 4.3 that even though the real power is specified for the P-V bus-5, its reactive power is unknown. Therefore, to update the voltage of this bus, we must first estimate the reactive power of this bus. Note from Fig. 4.11 that

$$Q_{i,inj} = -\text{Im}\left[V_i^* \sum_{k=1}^{n} Y_{ik}V_k\right] = -\text{Im}\left[V_i^*\left\{Y_{i1}V_1 + Y_{i2}V_2 + \cdots + Y_{ii}V_i + \cdots + Y_{in}V_n\right\}\right] \tag{4.5}$$

And hence we can write the k^{th} iteration values as

$$Q_{i,inj}{}^{(k)} = -\operatorname{Im}\left[V_i^{*(k-1)}\left\{Y_{i1}V_1 + Y_{i2}V_2{}^{(k)} + \cdots + Y_{ii}V_i^{(k-1)} + \cdots + Y_{in}V_n{}^{(k-1)}\right\}\right] \tag{4.6}$$

For the system of Fig. 4.1 we have

$$Q_{5,inj}{}^{(1)} = -\operatorname{Im}\left[V_1^{*(0)}\left\{Y_{51}V_1 + Y_{52}V_2{}^{(1)} + Y_{53}V_3{}^{(1)} + Y_{54}V_4{}^{(1)} + Y_{55}V_5{}^{(0)}\right\}\right] \tag{4.7}$$

This is computed as 0.0899 per unit. Once the reactive power is estimated, the bus-5 voltage is updated as

$$V_5^{(1)} = \frac{1}{Y_{55}}\left[\frac{P_{5,inj} - jQ_{5,inj}{}^{(1)}}{V_5^{*(0)}} - Y_{51}V_1 - Y_{52}V_2{}^{(1)} - Y_{53}V_3{}^{(1)} - Y_{54}V_4{}^{(0)}\right] \tag{4.8}$$

It is to be noted that even though the power generation in bus-5 is 48 MW, there is a local load that is consuming half that amount. Therefore the net power injected by this bus is 24 MW and consequently the injected power $P_{5,inj}$ in this case is taken as 0.24 per unit. The voltage is calculated as $V_4^{(1)} = 1.0169\angle -0.8894°$. Unfortunately, however the magnitude of the voltage obtained above is not equal to the magnitude given. We must therefore force this voltage magnitude to be equal to that specified. This is accomplished by

$$V_{5,corr}{}^{(1)} = \left|V_5\right| \times \frac{V_5^{(1)}}{\left|V_5^{(1)}\right|} \tag{4.9}$$

This will fix the voltage magnitude to be 1.02 per unit while retaining the phase of $-0.8894°$. The corrected voltage is used in the next iteration.

Convergence of the Algorithm

As can be seen from Table 4.3 that a total number of 4 real and 3 reactive powers are known to us. We must then calculate each of these from (4.6) and (4.7) using the values of the voltage magnitudes and their angle obtained after each iteration. The power mismatches are then calculated from (4.9) and (4.10). The process is assumed to have converged when each of ΔP_2, ΔP_3, ΔP_4, ΔP_5, ΔQ_2, ΔQ_3 and ΔQ_4 is below a small pre-specified value. At this point the process is terminated.

Sometimes to accelerate computation in the P-Q buses the voltages obtained from (4.2) is multiplied by a constant. The voltage update of bus-*i* is then given by

$$V_{i,acc}{}^{(k)} = (1-\lambda)V_{i,acc}{}^{(k-1)} + \lambda V_i^{(k)} = V_{i,acc}{}^{(k-1)} + \lambda\left\{V_i^{(k)} - V_{i,acc}{}^{(k-1)}\right\} \tag{4.10}$$

where λ is a constant that is known as the *acceleration factor*. The value of λ has to be below 2.0 for the convergence to occur. Table 4.4 lists the values of the bus voltages after the 1st iteration and number of iterations required for the algorithm to converge for different values of λ. It can be seen that the algorithm converges in the least number of iterations when λ is

1.4 and the maximum number of iterations are required when λ is 2. In fact the algorithm will start to diverge if larger values of acceleration factor are chosen. The system data after the convergence of the algorithm will be discussed later.

Forming Y_{bus} Matrix

This is a function that can be called by various programs. The function can be invoked by the statement

[yb,ych]=ybus;

where ‘yb’ and ‘ych’ are respectively the Y_{bus} matrix and a matrix containing the line charging admittances. It is assumed that the system data of Table are given in matrix form and the matrix that contains line impedances is ‘zz’, while ‘ych’ contains the line charging information. This program is stored in the file ybus.m.

Ybus matrix so formed for the 5 bus system.

Figure 5 Y BUS MATRIX

	1	2	3	4	5
1	2.6923 – j13.4115	– 1.9231 + j9.6154	0	0	– 0.7692 + j3.8462
2	– 1.9231 + j9.6154	3.6538 – j18.1942	– 0.9615 + j4.8077	0	– 0.7692 + j3.8462
3	0	– 0.9615 + [illegible]	[illegible]2115 – j11.0027	– 0.7692 + j3.8462	– 0.4808 + j2.4038
4	0	0	– 0.7692 + j3.8462	1.1538 – j5.6742	– 0.3846 + j1.9231
5	– 0.7692 + j3.8462	– 0.7692 + j3.8462	– 0.4808 + j2.4038	– 0.3846 + j1.9231	2.4038 – j11.8942

FLOWCHART

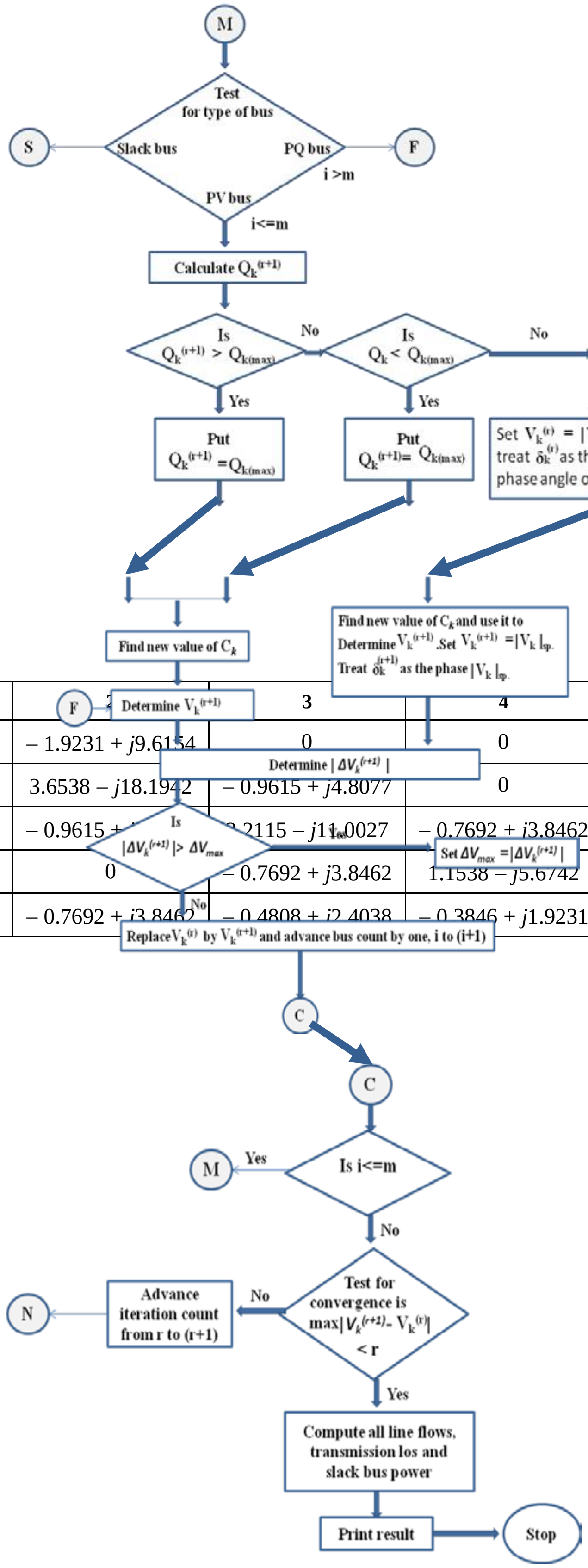


Fig 5 Flow chart of Gauss-Seidel Method

Q_i.

ALGORITHM OF GAUSS-SEIDEL

- Step 1: Form the nodal admittance matrix (Y_{ij}).
- Step 2: Choose a tolerance valuee.
- Step 3: Assume the initial voltage values to be 1pu and 0degree except for the slack bus.
- Step 4: Start iteration for bus i =1 with count 0.
- Step 5: Check for the slack bus if for i=slack bus or PV bus update the value of $Q_{i.}$
- Step 6: Calculate the new bus voltage V_i from the load flow equation.

- Step 7: Find the difference in the voltages
- $$dV_i^{k+1} = V_i^{k+1} - V_i^k$$
- Step 8: The new calculated value of the bus voltage is updated in the old bus voltage value and is used for the calculations at the next bus.
- Step 9: Go for the next bus and repeat the steps 5 to 7 until a new set of values of bus voltages are obtained for all the buses.
- Step 10: Continue the iteration from 5 to 9 until the value of dV_i^k at all the buses is within the chosen tolerance value

$$dV_i^{k+1} < e$$

where k gives the number of iterations.

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 of n variables $x_1, x_2, \ldots, x_n$. Let these equations be given by

$$\begin{aligned} f_1(x_1, \cdots, x_n) &= \eta_1 \\ f_2(x_1, \cdots, x_n) &= \eta_2 \\ &\vdots \\ f_n(x_1, \cdots, x_n) &= \eta_n \end{aligned} \qquad (4.11)$$

where $f_1, \ldots, f_n$ are functions of the variables $x_1, x_2, \ldots, x_n$. We can then define another set of functions $g_1, \ldots, g_n$ as given below

$$\begin{aligned} g_1(x_1, \cdots, x_n) &= f_1(x_1, \cdots, x_n) - \eta_1 = 0 \\ g_2(x_1, \cdots, x_n) &= f_2(x_1, \cdots, x_n) - \eta_2 = 0 \\ &\vdots \\ g_n(x_1, \cdots, x_n) &= f_n(x_1, \cdots, x_n) - \eta_n = 0 \end{aligned} \qquad (4.12)$$

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

$$\begin{aligned} x_1^* &= x_1^{(0)} + \Delta x_1^{(0)} \\ x_2^* &= x_2^{(0)} + \Delta x_2^{(0)} \\ &\vdots \\ x_n^* &= x_n^{(0)} + \Delta x_n^{(0)} \end{aligned} \qquad (4.13)$$

The functions in (4.23) then can be written in terms of the variables given in (4.24) as

$$g_k(x_1^*, \cdots, x_n^*) = g_k\left(x_1^{(0)} + \Delta x_1^{(0)}, \cdots, x_n^{(0)} + \Delta x_n^{(0)}\right), \quad k = 1, \ldots, n \qquad (4.14)$$

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, $k = 1, \ldots, n$ is given as

$$g_k\left(x_1^*, \cdots, x_n^*\right) = g_k\left(x_1^{(0)}, \cdots, x_n^{(0)}\right) + \Delta x_1^{(0)} \left.\frac{\partial g_k}{\partial x_1}\right|^{(0)} + \Delta x_2^{(0)} \left.\frac{\partial g_k}{\partial x_2}\right|^{(0)} + \cdots + \Delta x_n^{(0)} \left.\frac{\partial g_k}{\partial x_n}\right|^{(0)} \qquad (4.15)$$

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

Equation (4.15) can be written in vector-matrix form as

$$\begin{bmatrix} \partial g_1/\partial x_1 & \partial g_1/\partial x_2 & \cdots & \partial g_1/\partial x_n \\ \partial g_2/\partial x_1 & \partial g_2/\partial x_2 & \cdots & \partial g_2/\partial x_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial g_n/\partial x_1 & \partial g_n/\partial x_2 & \cdots & \partial g_n/\partial x_n \end{bmatrix}^{(0)} \begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \\ \vdots \\ \Delta x_n^{(0)} \end{bmatrix} = \begin{bmatrix} 0 - g_1\left(x_1^{(0)}, \cdots, x_n^{(0)}\right) \\ 0 - g_2\left(x_1^{(0)}, \cdots, x_n^{(0)}\right) \\ \vdots \\ 0 - g_n\left(x_1^{(0)}, \cdots, x_n^{(0)}\right) \end{bmatrix} \qquad (4.16)$$

The square matrix of partial derivatives is called the Jacobian matrix J with $J^{(0)}$ indicating that the matrix is evaluated for the initial values of $x_2^{(0)}$, …, $x_n^{(0)}$. We can then write the solution of (4.16) as

$$\begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \\ \vdots \\ \Delta x_n^{(0)} \end{bmatrix} = \left[J^{(0)}\right]^{-1} \begin{bmatrix} \Delta g_1^{(0)} \\ \Delta g_2^{(0)} \\ \vdots \\ \Delta g_n^{(0)} \end{bmatrix} \qquad (4.17)$$

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

$$\begin{aligned} x_1^{(1)} &= x_1^{(0)} + \Delta x_1^{(0)} \\ x_2^{(1)} &= x_2^{(0)} + \Delta x_2^{(0)} \\ &\vdots \\ x_n^{(1)} &= x_n^{(0)} + \Delta x_n^{(0)} \end{aligned} \qquad (4.18)$$

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 (4.18) 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.

Example 1: Let us consider the following set of nonlinear equations

$$g_1(x_1, x_2, x_3) = x_1^2 - x_2^2 + x_3^2 - 11 = 0$$
$$g_2(x_1, x_2, x_3) = x_1 x_2 + x_2^2 - 3x_3 - 3 = 0$$
$$g_3(x_1, x_2, x_3) = x_1 - x_1 x_3 + x_2 x_3 - 6 = 0$$

The Jacobian matrix is then given by

$$J = \begin{bmatrix} 2x_1 & -2x_2 & 2x_3 \\ x_2 & x_1 + 2x_2 & -3 \\ 1 - x_3 & x_3 & -x_1 + x_2 \end{bmatrix}$$

The initial values to start the Newton-Raphson procedure must be carefully chosen. For example if we choose $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$ then all the elements of the 1st row will be zero making the matrix $J^{(0)}$ singular. In our procedure let us choose $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 1$. The Jacobian matrix is then given by

$$J^{(0)} = \begin{bmatrix} 2 & -2 & 2 \\ 1 & 3 & -3 \\ 0 & 1 & 0 \end{bmatrix}$$

Also the mismatches are given by

$$\begin{bmatrix} \Delta g_1^{(0)} \\ \Delta g_2^{(0)} \\ \Delta g_3^{(0)} \end{bmatrix} = \begin{bmatrix} 0 - g_1^{(0)} \\ 0 - g_2^{(0)} \\ 0 - g_3^{(0)} \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \\ 5 \end{bmatrix}$$

Consequently the corrections and updates calculated respectively from (4.17) and (4.19) are

$$\begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \\ \Delta x_3^{(0)} \end{bmatrix} = \begin{bmatrix} 4.75 \\ 5.00 \\ 5.25 \end{bmatrix} \text{ and } \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{bmatrix} = \begin{bmatrix} 5.75 \\ 6.00 \\ 6.25 \end{bmatrix}$$

The process converges in 7 iterations with the values of

$$x_1 = 2, \; x_2 = 3 \text{ and } x_3 = 4$$

LOAD FLOW BY NEWTON-RAPHSON METHOD

Let us assume that an n-bus power system contains a total number of n_p 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. We shall further use the mismatch equations of ΔP_i and ΔQ_i given in (4.9) and (4.10) 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 (4.27). For the load flow problem, this equation is of the form

$$J\begin{bmatrix} \Delta\delta_2 \\ \vdots \\ \Delta\delta_n \\ \dfrac{\Delta|V_2|}{|V_2|} \\ \vdots \\ \dfrac{\Delta|V_{1+n_p}|}{|V_{1+n_p}|} \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_n \\ \Delta Q_2 \\ \vdots \\ \Delta Q_{1+n_p} \end{bmatrix} \tag{4.20}$$

where the Jacobian matrix is divided into submatrices as

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \tag{4.21}$$

It can be seen that the size of the Jacobian matrix is $(n + n_p - 1) \times (n + n_p - 1)$. For example for the 5-bus problem of Fig. 4 this matrix will be of the size (7×7). 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 submatrices are

$$J_{11} = \begin{bmatrix} \dfrac{\partial P_2}{\partial \delta_2} & \cdots & \dfrac{\partial P_2}{\partial \delta_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial P_n}{\partial \delta_2} & \cdots & \dfrac{\partial P_n}{\partial \delta_n} \end{bmatrix} \tag{4.22}$$

$$J_{12} = \begin{bmatrix} |V_2|\dfrac{\partial P_2}{\partial |V_2|} & \cdots & |V_{1+n_p}|\dfrac{\partial P_2}{\partial |V_{1+n_p}|} \\ \vdots & \ddots & \vdots \\ |V_2|\dfrac{\partial P_n}{\partial |V_2|} & \cdots & |V_{1+n_p}|\dfrac{\partial P_n}{\partial |V_{1+n_p}|} \end{bmatrix} \tag{4.23}$$

$$J_{21} = \begin{bmatrix} \dfrac{\partial Q_2}{\partial \delta_2} & \cdots & \dfrac{\partial Q_2}{\partial \delta_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial Q_{1+n_p}}{\partial \delta_2} & \cdots & \dfrac{\partial Q_{1+n_p}}{\partial \delta_n} \end{bmatrix} \tag{4.24}$$

$$J_{22} = \begin{bmatrix} |V_2|\frac{\partial Q_2}{\partial |V_2|} & \cdots & |V_{1+n_p}|\frac{\partial Q_2}{\partial |V_{1+n_p}|} \\ \vdots & \ddots & \vdots \\ |V_2|\frac{\partial Q_{1+n_p}}{\partial |V_2|} & \cdots & |V_{1+n_p}|\frac{\partial Q_{1+n_p}}{\partial |V_{1+n_p}|} \end{bmatrix} \tag{4.25}$$

4.6.2 Formation of the Jacobian Matrix

We shall now discuss the formation of the submatrices of the Jacobian matrix. To do that we shall use the real and reactive power equations of . Let us rewrite them with the help of (2) as

$$P_i = |V_i|^2 G_{ii} + \sum_{\substack{k=1 \\ k \neq i}}^{n} |Y_{ik} V_i V_k| \cos(\theta_{ik} + \delta_k - \delta_i) \tag{4.26}$$

$$Q_i = -|V_i|^2 B_{ii} - \sum_{\substack{k=1 \\ k \neq i}}^{n} |Y_{ik} V_i V_k| \sin(\theta_{ik} + \delta_k - \delta_i) \tag{4.27}$$

A. Formation of J_{11}

Let us define J_{11} as

$$J_{11} = \begin{bmatrix} L_{22} & \cdots & L_{2n} \\ \vdots & \ddots & \vdots \\ L_{n2} & \cdots & L_{nn} \end{bmatrix} \tag{4.28}$$

It can be seen from (4.32) that M_{ik}'s are the partial derivatives of P_i with respect to δ_k. The derivative P_i (4.38) with respect to k for $i \neq k$ is given by

$$L_{ik} = \frac{\partial P_i}{\partial \delta_k} = -|Y_{ik} V_i V_k| \sin(\theta_{ik} + \delta_k - \delta_i), \quad i \neq k \tag{4.29}$$

Similarly, the derivative P_i with respect to k for $i = k$ is given by

,

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

Comparing the above equation with we can write

$$L_{ii} = \frac{\partial P_i}{\partial \delta_i} = -Q_i - |V_i|^2 B_{ii} \tag{4.30}$$

B. Formation of J_{21}

Let us define J_{21} as

$$J_{21} = \begin{bmatrix} M_{22} & \cdots & M_{2n} \\ \vdots & \ddots & \vdots \\ M_{n_p 2} & \cdots & M_{n_p n} \end{bmatrix} \tag{4.31}$$

From (4.34) it is evident that the elements of J_{21} are the partial derivative of Q with respect to δ. From (4.39) we can write

$$M_{ik} = \frac{\partial Q_i}{\partial \delta_k} = -|Y_{ik}V_iV_k|\cos(\theta_{ik} + \delta_k - \delta_i), \quad i \neq k \tag{4.32}$$

Similarly for $i = k$ we have

$$M_{ii} = \frac{\partial Q_i}{\partial \delta_i} = \sum_{\substack{k=1 \\ k \neq i}}^{n} |Y_{ik}V_iV_k|\cos(\theta_{ik} + \delta_k - \delta_i) = P_i - |V_i|^2 G_{ii} \tag{4.33}$$

The last equality of (4.35) is evident from (4.28).

C. Formation of J_{12}

Let us define J_{12} as

$$J_{12} = \begin{bmatrix} N_{22} & \cdots & N_{2n_p} \\ \vdots & \ddots & \vdots \\ N_{n2} & \cdots & N_{nn_p} \end{bmatrix} \tag{4.34}$$

As evident from (4.23), the elements of J_{21} involve the derivatives of real power P with respect to magnitude of bus voltage $|V|$. For $i \neq k$, we can write from (4.28)

$$N_{ik} = |V_k|\frac{\partial P_i}{\partial |V_k|} = |Y_{ik}V_iV_k|\cos(\theta_{ik} + \delta_k - \delta_i) = -M_{ik} \quad i \neq k \tag{4.35}$$

For $i = k$ we have

$$\begin{aligned} N_{ii} = |V_i|\frac{\partial P_i}{\partial |V_i|} &= |V_i|\left[2|V_i|G_{ii} + \sum_{\substack{k=1 \\ k \neq i}}^{n} |Y_{ik}V_k|\cos(\theta_{ik} + \delta_k - \delta_i)\right] \\ &= 2|V_i|^2 G_{ii} + \sum_{\substack{k=1 \\ k \neq i}}^{n} |Y_{ik}V_iV_k|\cos(\theta_{ik} + \delta_k - \delta_i) = 2|V_i|^2 G_{ii} + M_{ii} \end{aligned} \tag{4.36}$$

D. Formation of J_{22}

For the formation of J_{22} let us define

$$J_{22} = \begin{bmatrix} O_{22} & \cdots & O_{2n_p} \\ \vdots & \ddots & \vdots \\ O_{n_p 2} & \cdots & O_{n_p n_p} \end{bmatrix} \tag{4.37}$$

For $i \neq k$ we can write from (4.39)

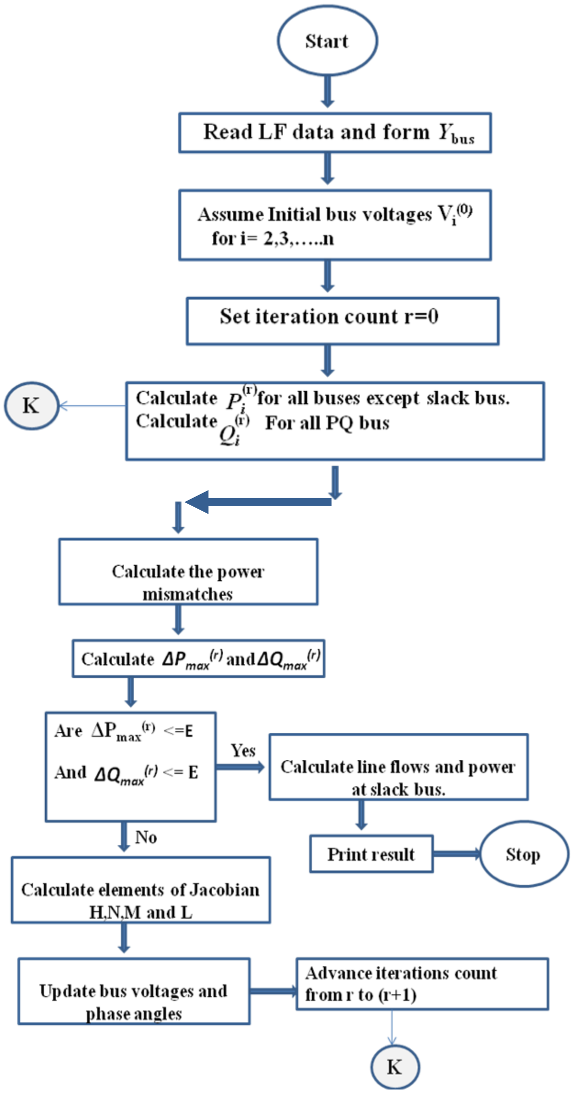
$$O_{ik} = |V_i| \frac{\partial Q_i}{\partial |V_k|} = -|V_i||Y_{ik} V_i V_k| \sin(\theta_{ik} + \delta_k - \delta_i) = L_{ik}, \quad i \neq k \tag{4.38}$$

Finally, for $i = k$ we have

$$\begin{aligned} O_{ii} = |V_i| \frac{\partial Q_i}{\partial |V_k|} &= |V_i| \left[-2|V_i| B_{ii} - \sum_{\substack{k=1 \\ k \neq i}}^{n} |Y_{ik} V_k| \sin(\theta_{ik} + \delta_k - \delta_i) \right] \\ &= -2|V_i|^2 B_{ii} - \sum_{\substack{k=1 \\ k \neq i}}^{n} |Y_{ik} V_i V_k| \sin(\theta_{ik} + \delta_k - \delta_i) = -2|V_i|^2 B_{ii} - L_{ii} \end{aligned} \tag{4.39}$$

We therefore see that once the submatrices J_{11} and J_{21} are computed, the formation of the submatrices J_{12} and J_{22} is fairly straightforward. For large system this will result in considerable saving in the computation time.

FLOWCHART



Load Flow Algorithm

The Newton-Raphson procedure is as follows:

Step-1: Choose the initial values of the voltage magnitudes $|V|^{(0)}$ of all n_p load buses and $n - 1$ angles $\delta^{(0)}$ of the voltages of all the buses except the slack bus.

Step-2: Use the estimated $|V|^{(0)}$ and $\delta^{(0)}$ to calculate a total $n - 1$ number of injected real power $P_{calc}^{(0)}$ and equal number of real power mismatch $\Delta P^{(0)}$.

Step-3: Use the estimated $|V|^{(0)}$ and $\delta^{(0)}$ to calculate a total n_p number of injected reactive power $Q_{calc}^{(0)}$ and equal number of reactive power mismatch $\Delta Q^{(0)}$.

Step-3: Use the estimated $|V|^{(0)}$ and $\delta^{(0)}$ to formulate the Jacobian matrix $J^{(0)}$.

Step-4: Solve (4.20) for $\Delta\delta^{(0)}$ and $\Delta|V|^{(0)} \div |V|^{(0)}$.

Step-5: Obtain the updates from

$$\delta^{(1)} = \delta^{(0)} + \Delta\delta^{(0)} \tag{4.40}$$

$$|V|^{(1)} = |V|^{(0)}\left[1 + \frac{\Delta|V|^{(0)}}{|V|^{(0)}}\right] \tag{4.41}$$

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 (4.26) and (4.27).

4.6.3 Solution of Newton-Raphson Load Flow

The Newton-Raphson load flow program is tested on the system of Fig. 4.1 with the system data and initial conditions given in datasheets we can write

$$L_{23}^{(0)} = -\left|Y_{23}V_2^{(0)}V_3^{(0)}\right|\sin\left(\theta_{23} + \delta_3 - \delta_2\right) = -|Y_{23}|\sin\theta_{23} = -B_{23} = -4.8077$$

Similarly, we have

$$Q_2^{(0)} = -\left|V_2^{(0)}\right|^2 B_{22} - \sum_{\substack{k=1\\k\neq 2}}^{n}\left|Y_{2k}V_2^{(0)}V_k^{(0)}\right|\sin\left(\theta_{2k} + \delta_k - \delta_2\right)$$
$$= -B_{22} - 1.05B_{21} - B_{23} - B_{24} - 1.02B_{25} = -0.6327$$

Hence, we get

$$L_{22}^{(0)} = -Q_2^{(0)} - \left|V_2^{(0)}\right|^2 B_{22} = -0.6327 - B_{22} = 18.8269$$

In a similar way the rest of the components of the matrix $J_{11}^{(0)}$ are calculated. This matrix is given by

$$J_{11}^{(0)} = \begin{bmatrix} 18.8269 & -4.8077 & 0 & -3.9231 \\ -4.8077 & 11.1058 & -3.8462 & -2.4519 \\ 0 & -3.8462 & 5.8077 & -1.9615 \\ -3.9231 & -2.4519 & -1.9615 & 12.4558 \end{bmatrix}$$

For forming the off- diagonal elements of J_{21} we note from (4.44) that

$$M_{23}^{(0)} = -\left|Y_{23}V_2^{(0)}V_3^{(0)}\right|\cos\left(\theta_{23} + \delta_2 - \delta_3\right) = -G_{23} = 0.9615$$

Also, the real power injected at bus-2 is calculated as

$$P_2^{(0)} = \left|V_2^{(0)}\right|^2 G_{22} + \sum_{\substack{k=1 \\ k\neq 2}}^{n} \left|Y_{2k} V_2^{(0)} V_k^{(0)}\right| \cos\left(\theta_{2k} + \delta_k - \delta_2\right)$$

$$= G_{22} + 1.05G_{21} + G_{23} + G_{24} + 1.02G_{25} = -0.1115$$

we have

$$M_{22} = P_2^{(0)} - \left|V_2^{(0)}\right|^2 G_{22} = -3.7654$$

Similarly, the rest of the elements of the matrix J_{21} are calculated. This matrix is then given as

$$J_{21}^{(0)} = \begin{bmatrix} -3.7654 & 0.9615 & 0 & 0.7846 \\ 0.9615 & -2.2212 & 0.7692 & 0.4904 \\ 0 & 0.7692 & -1.1615 & 0.3923 \end{bmatrix}$$

For calculating the off- diagonal elements of the matrix J_{12} we note from (4.47) that they are negative of the off- diagonal elements of J_{21}. However, the size of J_{21} is (3×4) while the size of J_{12} is (4×3). Therefore, to avoid this discrepancy we first compute a matrix M that is given by

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}$$

The elements of the above matrix are computed in accordance with (4.44) and (4.45). We can then define

$$J_{21} = M(1:3, 1:4) \text{ and } J_{12} = -M(1:4, 1:3)$$

Furthermore, the diagonal elements of J_{12} are overwritten in accordance with (4.48). This matrix is then given by

$$J_{12}^{(0)} = \begin{bmatrix} 3.5423 & -0.9615 & 0 \\ -0.9615 & 2.2019 & -0.7692 \\ 0 & -0.7692 & 1.1462 \\ 0.7846 & -0.4904 & -0.3923 \end{bmatrix}$$

Finally, it can be noticed that $J_{22} = J_{11}(1:3, 1:3)$. However, the diagonal elements of J_{22} are then overwritten in accordance with (4.51). This gives the following matrix

$$J_{22}^{(0)} = \begin{bmatrix} 17.5615 & -4.8077 & 0 \\ -4.8077 & 10.8996 & -3.8462 \\ 0 & -3.8462 & 5.5408 \end{bmatrix}$$

From the initial conditions the power and reactive power are computed as

$$P_{calc}^{(0)} = [-0.1115 \quad -0.0096 \quad -0.0077 \quad -0.0098]^T$$
$$Q_{calc}^{(0)} = [-0.6327 \quad -0.1031 \quad -0.1335]^T$$

Consequently, the mismatches are found to be

$$\Delta P^{(0)} = [-0.8485 \quad -0.3404 \quad -0.1523 \quad 0.2302]^T$$
$$\Delta Q^{(0)} = [0.0127 \quad -0.0369 \quad 0.0535]^T$$

Then the updates at the end of the first iteration are given as

$$\begin{bmatrix} \delta_2^{(0)} \\ \delta_3^{(0)} \\ \delta_3^{(0)} \\ \delta_4^{(0)} \end{bmatrix} = \begin{bmatrix} -4.91 \\ -6.95 \\ -7.19 \\ -3.09 \end{bmatrix} \text{deg} \qquad \begin{bmatrix} |V_2|^{(0)} \\ |V_3|^{(0)} \\ |V_4|^{(0)} \end{bmatrix} = \begin{bmatrix} 0.9864 \\ 0.9817 \\ 0.9913 \end{bmatrix}$$

The load flow converges in 7 iterations when all the power and reactive power mismatches are below 10^{-6}.

CHAPTER 5

MATLAB PROGRAM

GAUSS-SEIDEL

```
clear all
d2r=pi/180; w=100*pi;
% The Y  bus matrix is
[ybus]=ybus;
g=real(ybus); b=imag(ybus);
% The given parameters and initial conditions are
p=[0;-0.96;-0.35;-0.16;0.24];
q=[0;-0.62;-0.14;-0.08;-0.35];
mv=[1.05;1;1;1;1.02];
th=[0;0;0;0;0];
v=[mv(1);mv(2);mv(3);mv(4);mv(5)];
acc=input('Enter the acceleration constant: ');
del=1;
indx=0;
% The Gauss-Seidel iterations starts here
while del>1e-4
% P-Q buses
  for i=2:4
    tmp1=(p(i)-j*q(i))/conj(v(i));
    tmp2=0;
    for k=1:5
      if (i==k)
        tmp2=tmp2+0;
      else
        tmp2=tmp2+ybus(i,k)*v(k);
      end
    end
    vt=(tmp1-tmp2)/ybus(i,i);
    v(i)=v(i)+acc*(vt-v(i));
 end
% P-V bus
    q5=0;
    for i=1:5
      q5=q5+ybus(5,i)*v(i);
    end
    q5=-imag(conj(v(5))*q5);
    tmp1=(p(5)-j*q5)/conj(v(5));
    tmp2=0;
    for k=1:4
```

```
      tmp2=tmp2+ybus(5,k)*v(k);
    end
    vt=(tmp1-tmp2)/ybus(5,5);
    v(5)=abs(v(5))*vt/abs(vt);
% Calculate P and Q
    for i=1:5
      sm=0;
      for k=1:5
        sm=sm+ybus(i,k)*v(k);
      end
      s(i)=conj(v(i))*sm;
    end

% The mismatch
    delp=p-real(s)';
    delq=q+imag(s)';
    delpq=[delp(2:5);delq(2:4)];
    del=max(abs(delpq));
    indx=indx+1;
    if indx==1
      pause
    end
 end
```

Newton Raphson

```
% THIS IS THE NEWTON-RAPHSON POWER FLOW PROGRAM

clear all

d2r=pi/180;w=100*pi;

% The Ybus matrix is

[ybus]=ybus;
g=real(ybus);b=imag(ybus);
% The given parameters and initial conditions are
p=[0;-0.96;-0.35;-0.16;0.24];
q=[0;-0.62;-0.14;-0.08;-0.35];
mv=[1.05;1;1;1;1.02];
th=[0;0;0;0;0];
del=1;indx=0;
% The Newton-Raphson iterations starts here
while del>1e-4
  for i=1:5
    temp=0;
    for k=1:5
      temp=temp+mv(i)*mv(k)*(g(i,k)-j*b(i,k))*exp(j*(th(i)-th(k)));
```

```
    end
    pcal(i)=real(temp);qcal(i)=imag(temp);
  end
% The mismatches
  delp=p-pcal';
  delq=q-qcal';
% The Jacobian matrix
   for i=1:4
     ii=i+1;
     for k=1:4
       kk=k+1;
       j11(i,k)=mv(ii)*mv(kk)*(g(ii,kk)*sin(th(ii)-th(kk))-b(ii,kk)*cos(th(ii)-th(kk)));
    end
    j11(i,i)=-qcal(ii)-b(ii,ii)*mv(ii)^2;
  end
 for i=1:4
    ii=i+1;
     for k=1:4
       kk=k+1;
       j211(i,k)=-mv(ii)*mv(kk)*(g(ii,kk)*cos(th(ii)-th(kk))-b(ii,kk)*sin(th(ii)-th(kk)));
    end
j211(i,i)=pcal(ii)-g(ii,ii)*mv(ii)^2;
  end
  j21=j211(1:3,1:4);
  j12=-j211(1:4,1:3);
  for i=1:3
    j12(i,i)=pcal(i+1)+g(i+1,i+1)*mv(i+1)^2;
  end
  j22=j11(1:3,1:3);
  for i=1:3
    j22(i,i)=qcal(i+1)-b(i+1,i+1)*mv(i+1)^2;
  end
  jacob=[j11 j12;j21 j22];
  delpq=[delp(2:5);delq(2:4)];
  corr=inv(jacob)*delpq;
  th=th+[0;corr(1:4)];

mv=mv+[0;mv(2:4).*corr(5:7);0];
  del=max(abs(delpq));
  indx=indx+1;
end
preal=(pcal+[0 0 0 0 0.24])*100;
preac=(qcal+[0 0 0 0 0.11])*100;
% Power flow calculations
for i=1:5
  v(i)=mv(i)*exp(j*th(i));
end
```

```
for i=1:4
  for k=i+1:5
    if (ybus(i,k)==0)
      s(i,k)=0;s(k,i)=0;
      c(i,k)=0;c(k,i)=0;
      q(i,k)=0;q(k,i)=0;
      cur(i,k)=0;cur(k,i)=0;
    else
cu=-(v(i)-v(k))*ybus(i,k);
      s(i,k)=-v(i)*cu'*100;
      s(k,i)=v(k)*cu'*100;
      c(i,k)=100*abs(ybus(i,k))*abs(v(i))^2;
      c(k,i)=100*abs(ybus(k,i))*abs(v(k))^2;
      cur(i,k)=cu;cur(k,i)=-cur(i,k);
    end
  end
end
pwr=real(s);
qwr=imag(s);
q=qwr-c;
% Power loss
ilin=abs(cur);
for i=1:4
  for k=i+1:5
    if (ybus(i,k)==0)
      pl(i,k)=0;pl(k,i)=0;
      ql(i,k)=0;ql(k,i)=0;
    else
      z=-1/ybus(i,k);
      r=real(z);
      x=imag(z);
      pl(i,k)=100*r*ilin(i,k)^2;pl(k,i)=pl(i,k);
      ql(i,k)=100*x*ilin(i,k)^2;ql(k,i)=ql(i,k);
    end
  end
end
```

Comparison between two methods:

```
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*0.020;
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 48];
QsG=[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;
else
```

```
if 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]
t=t+abs(V(N)*Y(k,N))*cos(theta(k,N)+del(N)-del(k));
end
 t1=2*abs(V(k)*Y(k,n))*cos(theta(k,n))+t;
JPV(k-1,n-2)=t1; t=0;
Else
if k~=n
JPV(k-1,n-2)=abs(V(k)*Y(k,n))*cos(theta(k,n)- del(k)+del(n));
 end
end
if k>=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;
else
if 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(N)*Y(k,N))*sin(theta(k,N)- del(k)+del(N));
```

```
 end
t3=-2*abs(V(k)*Y(k,n))*sin(theta(k,n))-t2;
 t2=0;
JQV(k-2,n-2)=t3;
 Else
if k~=n
JQV(k-2,n-2)=-abs(V(k)*Y(k,n))*sin(theta(k,n)+del(n)- del(k));
 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 Del(kk+1)=jac_inv(kk);
else
if kk>3 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));
P3sch=(PsG(3)-PsL(3));
```

```
P4sch=(PsG(4)-PsL(4));
Q2sch=(QsG(2)-QsL(2));
Q3sch=(QsG(3)-QsL(3));
Q4sch=(QsG(4)-QsL(4));
V2=(1/Y(2,2))*((P2sch-1i*Q2sch)/V(2)-(Y(2,1)*V(1))- (Y(2,3)*V(3))-(Y(2,4)*V(4))-
(Y(2,5)*V(5)));
V3=(1/Y(3,3))*((P3sch-1i*Q3sch)/V(3)-(Y(3,1)*V(1))- (Y(3,2)*V(2))-(Y(3,4)*V(4))-
(Y(3,5)*V(5)));
V4=(1/Y(4,4))*((P4sch-1i*Q4sch)/V(4)-(Y(4,1)*V(1))- (Y(4,2)*V(2))-(Y(4,3)*V(3))-
(Y(4,5)*V(5)));
%% Updating reactive power on PV bus
Q=- 1i*[V(5)*(Y(5,1)*V(1)+Y(5,2)*V2+Y(5,3)*V3+Y(5,4)*V4+Y(5,5)*V( 5))];
V(2:4)=[V2 V3 V4];
V=abs(V);
k=1:5;
figure plot(k,V1,'r','linewidth',2,'marker','o','markersize',10)
hold on
grid on
plot(k,V,'k','linewidth',2,'marker','o','markersize',10)
xlabel('Number of Buses','fontsize',10,'fontweight','bold')
ylabel('Voltage','fontsize',10,'fontweight','bold')
title('Graph between Voltage vs Number of buses','fontsize',10,'fontweight','bold')
legend('Newton Raphson','Gauss seidel')
```

CHAPTER 6

COMPARISON BETWEEN POWER FLOW SOLUTION METHODS:

Figure 6

GAUSS-SEIDAL	NEWTON RAPHSON
1. The variables are expressed in rectangular coordinates. 2. Computation time per iteration is less. 3. It has linear convergence characteristics. 4. The number of iterations required for convergence increase with size of the system. 5. The choice of slack bus is critical.	1. The variables are expressed in polar coordinates. 2.Computation time per iteration is more 3. It has quadratic convergence characteristics. 4.The number of iterations are independent of the size of the system. 5.The choice of slack bus is arbitrary.

CHAPTER 7

(Conclusion, Result & Future scope)

RESULTS

Gauss-Seidel Iteration Output:

```
ybus =
Columns 1 through 4
   2.6923 -13.4115i   -1.9231 + 9.6154i       0                    0
  -1.9231 + 9.6154i   3.6538 -18.1942i  -0.9615 + 4.8077i       0
       0                  -0.9615 + 4.8077i   2.2115 -11.0027i  -
0.7692 + 3.8462i
       0                          0              -0.7692 + 3.8462i
1.1538 - 5.6742i
  -0.7692 + 3.8462i  -0.7692 + 3.8462i  -0.4808 + 2.4038i  -
0.3846 + 1.9231i
Column 5
  -0.7692 + 3.8462i
  -0.7692 + 3.8462i
  -0.4808 + 2.4038i
  -0.3846 + 1.9231i
   2.4038 -11.8942i

Enter the acceleration constant: 1.6
GS LOAD FLOW CONVERGES IN ITERATIONS
indx = 24
FINAL VOLTAGE MAGNITUDES ARE:
    1.0500    0.9826    0.9777    0.9876    1.0200
FINAL ANGLES IN DEGREE ARE:-
```

0 -5.0124 -7.1322 -7.3705 -3.2014
THE REAL POWERS IN EACH BUS IN MW ARE
126.5955 -95.9999 -35.0000 -16.0000 48.0000
THE REACTIVE POWERS IN EACH BUS IN MVar ARE:
57.1094 -62.0001 -14.0000 -8.0000 15.5861

Newton Raphson Output

ybus =

Columns 1 through 4

2.6923 -13.4115i -1.9231 + 9.6154i 0 0
-1.9231 + 9.6154i 3.6538 -18.1942i -0.9615 + 4.8077i 0
0 -0.9615 + 4.8077i 2.2115 -11.0027i -
0.7692 + 3.8462i
0 0 -0.7692 + 3.8462i
1.1538 - 5.6742i
-0.7692 + 3.8462i -0.7692 + 3.8462i -0.4808 + 2.4038i -
0.3846 + 1.9231i

Column 5

-0.7692 + 3.8462i
-0.7692 + 3.8462i
-0.4808 + 2.4038i
-0.3846 + 1.9231i
2.4038 -11.8942i

Bus number	Voltage(pu)	Angle(degree)	Real Power(MW)	Reactive (MVar)
1	0.982	0	524.674	237.2981

2	1.0152	-0.1542	0.0148	-0.001
3	0.9956	-0.0983	0.046	-0.0159
4	0.9572	-0.1528	-321.97	-2.4013
5	0.9234	-0.1778	-499.98	-184.0008

NEWTON RAPHSON DATA AFTER ITERATION

table 2

COMPARISON OF PERFORMANCE

5-BUS SYSTEM	Gaus-Seidel Method	Newton-Raphson Method
Initial voltage and angle	1 p.u. and 0 degree	1 p.u. and 0 degre
Assumed Tolerance Value	0.0001	0.0001
Number of Iterations	24	9
Convergence Time	0.621	0.551
Maximum Mismatch	10.24	8.21

TABLE 3

λ	Bus voltages (per unit) after 1st iteration				No of iterations for convergence
	V_2	V_3	V_4	V_5	
1	$0.9927\angle -2.6°$	$0.9883\angle -2.83°$	$0.9968\angle -3.48°$	$1.02\angle -0.89°$	28
2	$0.9874\angle -5.22°$	$0.9766\angle -8.04°$	$0.9918\angle -14.02°$	$1.02\angle -4.39°$	860
1.8	$0.9883\angle -4.7°$	$0.9785\angle -6.8°$	$0.9903\angle -11.12°$	$1.02\angle -3.52°$	54
1.6	$0.9893\angle -4.17°$	$0.9807\angle -5.67°$	$0.9909\angle -8.65°$	$1.02\angle -2.74°$	24
1.4	$0.9903\angle -3.64°$	$0.9831\angle -4.62°$	$0.9926\angle -6.57°$	$1.02\angle -2.05°$	14
1.2	$0.9915\angle -3.11°$	$0.9857\angle -3.68°$	$0.9947\angle -4.87°$	$1.02\angle -1.43°$	19

TABLE 4 DATA FOR DIFFERENT ACCELERATION CONSTANTS

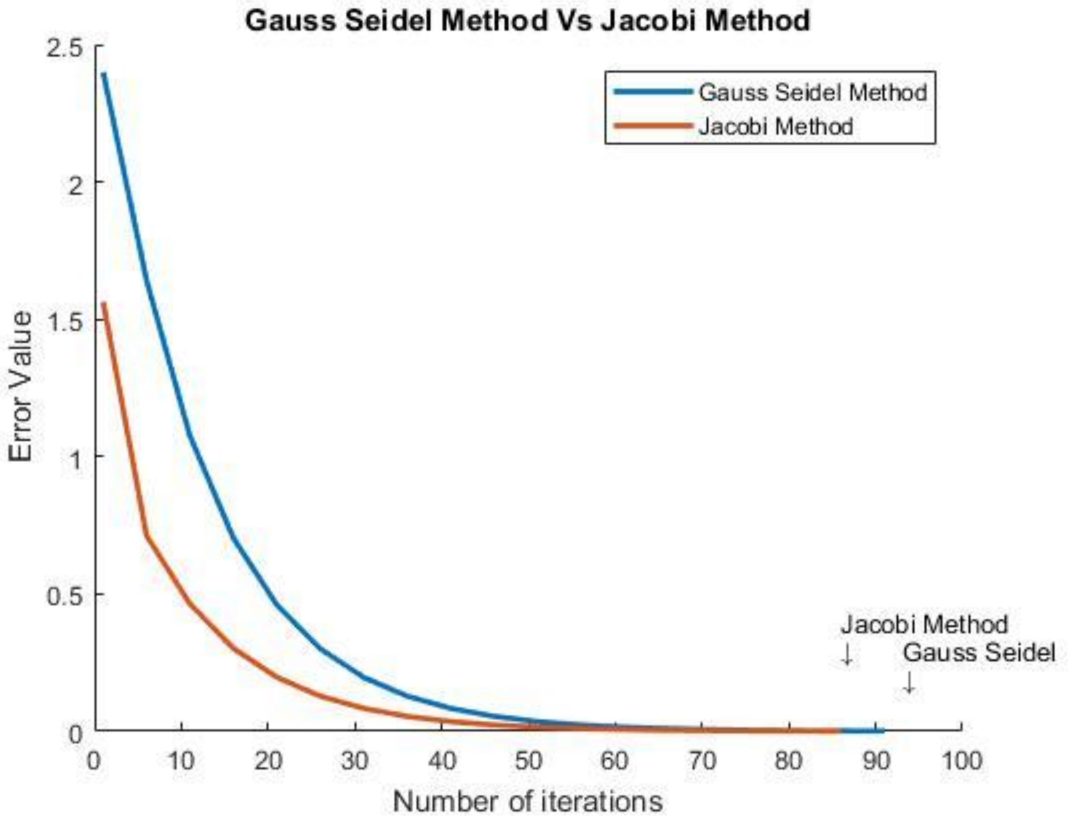


Figure 6CONVERGENCE CURVE

Bus no.	Bus voltage		Power generated		Load	
	Magnitude (pu)	Angle (deg)	P (MW)	Q (MVAr)	P (MW)	P (MVAr)
1	1.05	0	126.60	57.11	0	0
2	0.9826	– 5.0124	0	0	96	62
3	0.9777	– 7.1322	0	0	35	14
4	0.9876	– 7.3705	0	0	16	8
5	1.02	– 3.2014	48	15.59	24	11

Table 5

INITIAL DATA

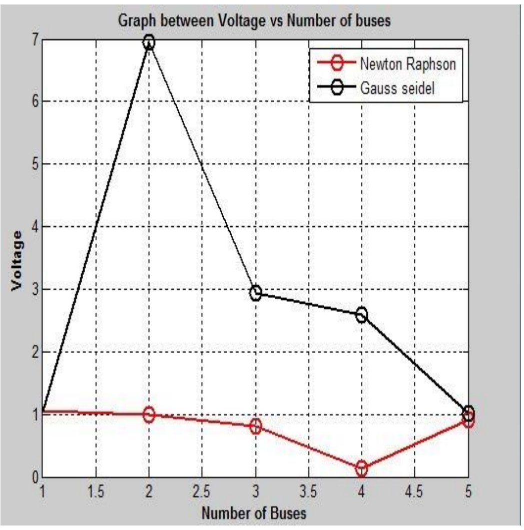


Figure 7VOLTAGE VS BUSES

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.

FUTURE SCOPE

This project concentrates on MATLAB programming to enable the users to calculate power flow problem. MATLAB is used to program the power flow solution and Graphical User Interface (GUI) use to help a user easy to use.

To achieve all the project’s objectives, the developer must have fulfilled all the scope below:

i. Studies MATLAB programming and MATLAB GUI
ii. Identify appropriate command for MATLAB M-files
iii. Build MATLAB program for the power flow analysis using M-files iv. Run simulation of power flow analysis using MATLAB for small, medium and large-scale system.

CHAPTER 8

(REFERENCES)

[1] John Grainger; William Stevenson's classic, Elements of Power System Analysis; Science/Engineering/Math; 1 ed.,pp.783., January 1, 1994.

[2] Elgerd, O.L. (2012) Electric Energy Systems Theory: An Introduction. 2nd Edition, Mc-Graw-Hill.

[3] Kothari, I.J. and Nagrath, D.P. (2007) Modern Power System Analysis. 3rd Edition, New York.

[4] Keyhani, A., Abur, A. and Hao, S. (1989) Evaluation of Power Flow Techniques for Personal Computers. *IEEE Transactions on Power Systems*, **4**, 817-826.

[5] Hale, H.W. and Goodrich, R.W. (1959) Digital Computation or Power Flow—Some New Aspects. Power Apparatus and Systems, Part III. *Transactions of the American Institute of Electrical Engineers*, **78**, 919-923.

[6] Sato, N. and Tinney, W.F. (1963) Techniques for Exploiting the Sparsity or the Network Admittance Matrix. *IEEE Transactions on Power Apparatus and Systems*, **82**, 944-950.

[7] Aroop, B., Satyajit, B. and Sanjib, H. (2014) Power Flow Analysis on IEEE 57 bus System Using Mathlab. *International Journal of Engineering Research & Technology* (*IJERT*), **3**.

[8] Milano, F. (2009) Continuous Newton's Method for Power Flow Analysis. *IEEE Transactions on Power Systems*, **24**, 50-57.

[9] Grainger, J.J. and Stevenson, W.D. (1994) Power System Analysis. McGraw-Hill, New York.

[10] Tinney, W.F. and Hart, C.E. (1967) Power Flow Solution by Newton's Method. *IEEE Transactions on Power Apparatus and Systems*, **PAS-86**, 1449-1460.

[11] Bhakti, N. and Rajani, N. (2014) Steady State Analysis of IEEE-6 Bus System Using PSAT Power Tool Box. *International Journal of Engineering Science and Innovation Technology* (*IJESIT*), **3**.

[12] Hadi, S. (2010) Power System Analysis. 3rd Edition, PSA Publishing, North York.

[13] Kabisama, H.W. Electrical Power Engineering. McGraw-Hill, New York.

[14] Gilbert, G.M., Bouchard, D.E. and Chikhani, A.Y. (1998) A Comparison of Load Flow Analysis Using Dist Flow, Gauss-Seidel, and Optimal Load Flow Algorithms. *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, Waterloo, Ontario, 24-28 May 1998, 850-853.

[15] Glover, J.D. and Sarma, M.S. (2002) Power System Analysis and Design. 3rd Edition, Brooks/Cole, Pacific Grove.

[16] Stott, B. and Alsac, O. (1974) Fast Decoupled Load Flow. *IEEE Transactions on Power Apparatus and Systems*, **PAS-93**, 859-869.

[17] Stott, B. (1974) Review of Load-Flow Calculation Methods. *Proceedings of the IEEE*, **62**, 916-929.

[18] Adejumobi, I.A., *et al*. (2014) Numerical Methods in Load Flow Analysis: An Application to Nigeria Grid System. *International Journal of Electrical and Electronics Engineering* (*IJEEE*), **3**.

APPENDIX A

SOFTWARE

MATLAB (*matrix laboratory*) is a multi-paradigm numerical computing environment. A proprietary programming languagedeveloped by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

VERSION: 2016a

Structures

MATLAB has structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations, etc.). Unfortunately, MATLAB JIT does not support MATLAB structures, therefore just a simple bundling of various variables into a structure will come at a cost.

Functions

When creating a MATLAB function, the name of the file should match the name of the first function in the file. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores. Functions are often case sensitive.

Function handles

MATLAB supports elements of lambda calculas by introducing function handles, or function references, which are implemented either in .m files or anonymous/nested functions.

Classes and object-oriented programming

MATLAB supports object-oriented programming including classes, inheritance, virtual dispatch, packages, pass-by-value semantics, and pass-by-reference semantics. However, the syntax and calling conventions are significantly different from other languages. MATLAB has value classes and reference classes, depending on whether the class has *handle*as a super-class (for reference classes) or not (for value classes).

MATLAB GUI

A graphical user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth. The GUI should behave in an understandable and predictable manner, so that a user knows what to expect when he or she performs an action. For example, when a mouse click occurs on pushbutton, the GUI should initiate the action described on the label of the button. This chapter introduces the basic elements of the MATLAB GUIs.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

APPENDIX B

DATASHEET

BUS DATA

Line (bus to bus)	Impedance	Line charging (*Y*/2)
1-2	0.02 + *j*0.10	*j*0.030
1-5	0.05 + *j*0.25	*j*0.020
2-3	0.04 + *j*0.20	*j*0.025
2-5	0.05 + *j*0.25	*j*0.020
3-4	0.05 + *j*0.25	*j*0.020
3-5	0.08 + *j*0.40	*j*0.010
4-5	0.10 + *j*0.50	*j*0.075

Table 6 BUS DATA