POWER FLOW SOLUTION BY USING MATLAB

By

SHAHRANDIN BIN ASLI

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme

in Partial Fulfillment of the Requirements

for the Degree

Bachelor of Engineering (Hons)

(Electrical & Electronics Engineering)

Universiti Teknologi Petronas

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

POWER FLOW SOLUTION BY USING MATLAB

by

Shahrandin Bin Asli

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Mr. Zuhairi Baharudin
Lecturer
Electrical & Electronic Engineering
Universiti Teknologi PETRONAS

Zuhairi Baharudin
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

December 2005

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Shahrandin Bin Asli

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Performance of power systems under steady-state operation can be analyzed by studying power flow study. Gauss-Seidel iteration method was used to solve the power flow equations. The power flow programming was made with help of MATLAB software. The voltages, currents, and real and reactive power flows in the power system under a given set of load conditions were determined as well as the line current, line flows and line losses. With all information available, the user of power flow study can determine whether any of the components in a power system will be overloaded by the particular conditions of the study. If the ratings of any components are exceeded, then the conditions of this study are an unacceptable steady state operating condition and something else must be tried. With load studies, the engineers are able to plan ahead the operation of the power system under normal conditions by fulfilling the economic and security requirements.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to Allah for the guidance for me upon completing my final year project.

Many people have been contributing ideas, guidance, supervision, support and technical assistance during the accomplishment of this project entitled 'Power Flow Solution by Using MATLAB'.

Here, I would like to express my utmost gratitude to my project supervisor Mr. Zuhairi Baharudin for spending his time to guide me in the completion of this project and lending me related books that I need in finishing the project. Special appreciation also dedicated to Ir. N Perumal, for helping me in understanding the power flow analysis in the power systems.

My sincere thanks also go to Miss Siti Hawa Bt. Hj Mohd Tahir, lab technologist, electrical and electronics department who has been assisting and providing me with technical support throughout the project.

Last but not least, thank you to my beloved family and friends who have been very supportive from the beginning of the project and all other people who are not mentioned here.

Thank you.

CHAPTER 1

INTRODUCTION

1.1 Background of Study

For this project, the main task is to develop power flow program and perform a study on a power flow analysis. It is very helpful for planning forward the operation of a power system under normal conditions at safe operation and future expansion. The Gauss-Seidel (G-S) Iterative method will be used to solve the power flow equations. Besides, it requires the knowledge and understanding of MATLAB for solving the power flow equations.

1.2 Problem Statement

1.2.1 Problem Identification

This project requires a solving of power flow solution by using MATLAB.

1.2.2 Significant of the Project

By studying and analyzing the power flow analysis, the voltages, currents and power flows in a power system under steady-state conditions can be determined.

1.3 Objective and Scope of Study

1.3.1 Objective

1. To study the important of power flow studies
2. To study power flow solution by using G-S Iteration Method
3. To apply the power flow solution by using MATLAB

1.3.2 Relevancy of the project

The study required for this project aims at obtaining the magnitude and phase angle of the voltage at each bus and the power flowing in each line. This study is useful for analyzing the performance of Power System both in normal operating conditions and under fault conditions. The operation of a Power System can be planned ahead for its safe operation and future growth.

1.3.3 Feasibility of the Project within the Scope and Time Frame

This project is a two-semester project such that the first semester will be mostly on research and study. The MATLAB software will be able to accomplish the requirement to solve the power flow equations. Therefore, this project is feasible to be carried out within the time and scope.

GANTT CHART

No	Activity	Semester JAN 2005													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	List, analyze and choose the final year project topics														
2	First meeting with FYP supervisor														
3	Preparation for the preliminary report														
4	Study and research literature review for power flow analysis -purpose to perform power flow study -study various types of power solution -read, understand and summarize journal from IEEE														
5	Preparation for progress report														
6	Study and understand the Gauss-Seidel iterative method to solve the power flow equation														
7	Done several case studies by running the power_flow program: -study and review power_flow program -analyze the result -study how to develop own program														
8	Preparation for Interim Report														
9	Preparation for FYP presentation														

No	Activity	Semester JULY 2005													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Continue and plan the advancement of previous semester project														
2	Try to visualize the power flow network by GUI application														
3	Study how to configure and build power system network by using Simulation Power Systems														
4	Find all data and specification related to the project requirement														
5	Preparation for the progress report														
6	Review last semester case study and understand the coding														
7	Find all data for IEEE Bus Standard Sample System														
8	Test IEEE 30-Bus Standard Test System														
9	Review and understanding of power flow and the method use in solving power Flow Equations														
10	Prepare for Progress report 2														
11	Review case study which has been done last semester														
12	Sharpen the understanding of power flow analysis and methods for solving the power flow equations														
13	Prepare for Pre-EDX presentation														
14	Prepare for draft report & Final Presentation														
15	Prepare for Final Report														

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction to Power-Flow Studies

Two of the earliest programs to be developed for power system analysis were the fault and load flow (power flow) programs. Both were originally produced in the late 1950s. Many programs in used today are either based on these two types of program or have one or the other embedded in them [1].

The purpose of a power system is to deliver the power that customers require in real time, on demand, within acceptable voltage and frequency limits, and in a reliable and economic manner [2].

A power system engineer must be able to analyze the performance of power systems both in normal operating conditions and under fault (short-circuit) conditions. The analysis of a power system in normal steady-state operation is known as a power-flow study or load-flow study. Power flow studies attempt to determine the voltages, currents, and real and reactive power flows in a power system under a given set of load conditions.

An engineer who understands the operation of a power system under normal conditions will be able to plan ahead for its safe operation and future growth. A good understanding of steady-state conditions allows the engineer to play "what if?' games.

For example, suppose that a new customer intends to open an industrial plant that requires 100 MW of power on the outskirts of some city. Will there be enough power-handling capacity in the system for this new load, or will the additional load cause some components to be overloaded? Will it be necessary to build new transmission lines, or to increase the number of size of installed transformers? Will it be necessary to build new generation capacity?

As another example, suppose that a power system is properly supplying loads, and a transmission line within the system must be taken off line for maintenance. Can the remaining lines in the power system handle the required loads without exceeding their ratings? These sorts of questions are asked on a regular basis.

Furthermore, the central planning departments of power companies try to anticipate the power needs 10 to 20 years into the future, and to simulate systems serving those needs. These studies help them to identify the need for additional transmission lines and generation capability early enough to plan properly for the future.

2.1.1 Basic Technique for Power Flow Studies

A power-flow study is an analysis of the voltages, currents and power flows in a power system under steady-state conditions. In a power flow study, the power systems engineer makes an assumption about either the voltage at bus or the power being supplied to the bus for each bus in the power system and then determines the magnitude and phase angles of the bus voltages, line currents, etc. that result from the assumed combination of voltages and power flows.

The simplest way to perform power-flow calculations is by iteration. We create a bus admittances matrix Y_{bus} for the power system, and make an initial estimate for the voltages at each bus in the system. We can then update the voltage estimate for the voltages at each bus in the system one at a time, based on the estimates for the voltages and power flows at every other bus and the values of the bus admittance matrix. Since the voltage at a given bus depends on the voltages at all of the other busses in the system, and the voltages at the other busses are just estimates, the updated voltage will not be correct. However, it will usually be closer to the correct answer than the original guess. If this process is repeated the voltages at each bus no longer change significantly during iteration, the solution has converged to the correct answer.

The equations used to update the estimates of voltage at a bus differ for different types of busses, as we will describe below. Each bus in a power system is classified as one of three types:

1. *Load bus.* A load bus is a bus at which the real power and reactive power are specified, and for which the bus voltage will be calculated. A load bus is known as PQ bus. All busses that do not have generators attached are load busses.
2. *Generator bus.* A generator bus is bus at which the magnitude of the voltage is kept constant by adjusting the field current of a synchronous generator tied to the bus. At a generator bus, the field current is assumed to be adjusted maintain at a constant terminal voltage, V_T. Generator's prime mover also controls the real power supplied by the generator to the power system. Thus at a generator bus, the magnitude of the bus voltage $|V_i|$ and real power P_i can be controlled and specified. A generator bus is known as PV bus.
3. *Slack bus.* The slack bus is a special generator bus that serves as the reference bus for the power system. Its voltage is assumed to be fixed in both magnitude and phase; most often it is taken to be $1\angle 0°$ pu. The real and reactive power of the slack bus is uncontrolled; it supplies whatever real or reactive power is necessary to make the power flows in the system balance.

These categories generally correspond to the ways in which real power systems operate. Load busses are busses that supply power to loads, and the amount of power supplied will be whatever the loads demand. The voltage on a load bus in a real power system will go up and down with changing loads, unless the power company compensates for these changes with tap-changing under load (TCUL) transformers or switched capacitors. Thus, load busses have specified values of P and Q, while V varies with load conditions.

Real generators operate most efficiently when they are running at full load, so power system try to keep all but one (or a few) generator running at 100 percent capacity, while allowing the remaining generator (called the swing generator) to handle increases and decreases in load demand. Thus, most busses with generators attached to them will be supplying a fixed amount of power (the rated full load power of the generators), and the magnitude of their voltages will be maintained constant by field circuits of the generators. These generator busses have specified values of P and $|V_i|$.

Finally, the controls on the swing generator will be adjusted to maintain a constant power system voltage and frequency, allowing real and reactive power to increase or decrease is required whenever loads change. The bus that the swing generator is connected to is called the slack bus [3].

2.1.2 Existing Techniques

Several techniques enable the study of electric systems and interconnections on digital computers. The various techniques have a common basis: the separation of the ac system solution into two sets of linear equations. The most widely used technique divides the real and reactive equations into independent sets of equations. The first set is the real power equations. The second is the reactive power equations. Some techniques are in fact variants of the methods used to solve sets of simultaneous linear algebraic equations. A popular variant is the GS technique that is related to relaxation techniques for simultaneous equation solution [4].

Basically, there have been many ways for applying numerical techniques to solve the power flow equations. Presently, the following are the leading techniques [5]:

1. **G-S method-** is known as the method of successive displacements. The process of iterations repeated until the change in variable is within the desired accuracy. It can be seen that Gauss-Seidel method needs many iterations to achieve the desired accuracy and there is no guarantee for the convergence.
2. **Newton-Raphson method-** the most widely used method for solving simultaneous nonlinear algebraic equations. Newton's method is a successive approximation procedure based on an initial estimate of the unknown and the use

of Taylor's series expansion. It is found to be efficient and practical for large power systems. The number iterations required to obtain a solution is independent of the system size, but more functional evaluations are required at each iteration.

3. **Fast decoupled power flow method**- requires more iteration than the Newton-Raphson method, but requires considerably less time per iteration and a power flow solution is obtained very rapidly. This technique is very useful in contingency analysis where numerous outages are to be simulated or a power flow solution is required for online control.

(Note: For this project, G-S Iteration method is used)

2.2 Power Flow Analysis Equation

2.2.1 Gauss Seidel Iterative Method

The complexity of obtaining a formal solution for load-flow in a power system arises because of the differences in the type of data specified for the different kind of buses. Although the formulation of sufficient equations is not difficult, the closed form of solution is not practical. Digital solutions of the load-flow problems shall be considered to follow an iterative process by assigning estimated values to the unknown bus voltages and calculating a new value for each bus voltage from the estimated values at the other buses, the real power specified, and the specified reactive power or voltage magnitude. A new set of values for voltage is thus obtained for each bus and used to calculate still another set of bus voltages. Each calculation of a new set of voltages is called iteration. The iterative process is repeated until the changes at each bus are less than a specified minimum value.

First, the solution shall be examined based on expressing the voltage of a bus as a function of the real and reactive power delivered to a bus from generators or supplied to the load connected to the bus, the estimated or previously calculated voltages at the other buses, and the self and mutual admittances of the nodes. The derivation of the fundamental equations starts with a node formulation of the network equations. The equations shall be derived for a four-bus system as example and the general equations will be written later. With the swing bus designated as number 1, computations start with bus 2. If P_2 and Q_2 are scheduled real and reactive power entering the system at bus 2,

Equation 1

$$V_2 I_2^* = P_2 + jQ_2$$

From which I_2 is expressed as

Equation 2

$$I_2 = \frac{P_2 - jQ_2}{V_2^*}$$

And in terms of self and mutual admittances of the nodes, with generators and loads omitted since the current into each node is expressed as in Equation below:

Equation 3

$$\frac{P_2 - jQ_2}{V_2^*} = Y_{21}V_1 + Y_{22}V_2 + Y_{23}V_3 + Y_{24}V_4$$

Solving for V_2 gives

Equation 4

$$V_2 = \frac{1}{Y_{22}}\left[\frac{P_2 - jQ_2}{V_2} - \left(Y_{21}V_1 + Y_{23}V_3 + Y_{24}V_4\right)\right]$$

Equation 4 gives a corrected value for V_2 based upon scheduled P_2 and Q_2 when the values estimated originally are substituted for the voltage expressions on the right side of the equation. The calculated value for V_2 and the estimated value for V_2^* will not agree.

By substituting the conjugate of the calculated value of V_2 for V_2^* in Equation 4 to calculate another value for V_2, agreement would be reached to a good degree of accuracy after several iterations and would be the correct value for V_2 with the estimated voltages and without regard to power at the other buses. This value would not be the solution for V_2 for the specific power flow conditions, however, because the voltages upon which this calculation for V_2 depends are the estimated values of voltage at the other buses and the actual voltages are not yet known. Two successive calculations of V_2 (the second being like the first except for the correction of V_2^*) are recommended at each bus before proceeding to the next one.

As the corrected voltage is found at each bus, it is used in calculating the corrected voltage at the next. The process is repeated at each bus consecutively throughout the network (except at the swing bus) to complete the first iteration. Then the entire process is carried out again and again until the amount of correction voltage at every each bus is less than some predetermined precision index.

Convergence upon erroneous solution may occur if the original voltages are widely different from the correct values. Erroneous convergence is usually avoided if the original values are of reasonable magnitude and do not differ too widely in phase. Any unwanted solution is usually detected easily by inspection of the results since the voltages of the system do not normally have a range in phase wider than 45° and the difference between nearby buses is less than about 10° and often very small.

For a total of N buses the calculated voltage at any bus k where P_k and Q_k are given is

Equation 5

$$V_i = \frac{1}{Y_{ii}}\left[\frac{P_i - jQ_i}{V_i^*} - \sum_{\substack{k=1 \\ k\neq i}}^{N} Y_{ik}V_k\right]$$

Where $k \neq i$. The values for the voltages on the right side of the equation are the most recently calculated values for the corresponding buses (or the estimated voltage if no iteration has yet been made at that particular bus).

Experience with G-S method of solution of power flow problems has shown that an excessive number of iterations are required before the voltage corrections are within an acceptable precision index if the corrected voltage at a bus merely replaces the best previous value as the computations proceed from bus to bus. The number of iterations required is reduced considerably if the correction in voltage at each bus is multiplied by some constant that increases the amount of correction to bring the voltage closer to the value it is approaching.

The multipliers that accomplish this improved convergence are called acceleration factors. The difference between the newly calculated voltage and the best previous voltage at the bus is multiplied by the appropriate acceleration factor to obtain a better correction to be added to the previous value. The acceleration factor for the real component of the correction may differ from that for the imaginary component. For any system, optimum values for acceleration factors exist, and poor choice of factors may result in less rapid convergence or make convergence impossible. An acceleration of 1.6 for both the real and imaginary components is usually a good choice. Studies may be made to determine the best choice for a particular system.

At a bus where voltage magnitude rather than reactive power is specified, the real and imaginary components of the voltage for each iteration are found by computing a value for the reactive power. From Equation 5:

Equation 6

$$P_i - jQ_i = \left(Y_{ii}V_i + \sum_{\substack{k=1 \\ k\neq i}}^{N} Y_{ik}V_k \right) V_i^*$$

Where $k \neq i$. If it is allow that n to equal to k

Equation 7

$$P_i - jQ_i = V_i^* \sum_{k=1}^{N} Y_{ik}V_k$$

Equation 8

$$Q_i = -I_m\left\{V_i^*\sum_{k=1}^{N} Y_{ik}V_k\right\}$$

Where, I_m means "imaginary part of".

Reactive power Q_k is evaluated by Equation 8 for the best previous voltage values at the buses and this value of Q_k is substituted in Equation 5 to find a new V_k. The components of the new V_k are then multiplied by the ratio of the specified constant magnitude of V_k to the magnitude of the V_k found by Equation 5. The result is the corrected complex voltage of the specified magnitude [4].

3.1 Information Derived From Power-Flow Studies

After the bus voltages are calculated at all busses in a power system, a power flow program can be set up to provide alerts if the voltage at any given bus exceeds plus minus 5 percent of the nominal value. The information is important, since power is supposed to be supplied at a constant at voltage level. The high or low voltages indicate spots where some sort of compensatory work is required, either with switched capacitor or with tap changing under load (TCUL) transformer. If the voltage is too low at a bus, additional capacitors can be switched on to the bus and the problem can be resolved to determine the effect of this action on the bus voltage. By solving such problems repeatedly, it is possible to define the amount of switched capacitance needed to ensure that the voltage on a bus remains in tolerance under all load conditions.

In addition, it is possible to determine the net real and reactive power either supplied to or removed from each bus by generators or loads connected to it. To calculate the real and reactive power at a bus, the net current injected is calculated at the bus. The net current injected at a bus will be equal to the sum of all the currents leaving the bus through transmission lines. The current leaving the bus on each transmission line will be equal to the difference between the voltages at either end of the transmission line multiplied by the admittance of the line, so the total current injected at the now will be

Equation 6

$$I_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} Y_{ik}(V_i - V_k)$$

The resulting real and reactive power injected at the bus can be found from equation

Equation 7

$$S_i = -V_i I_i^* = P_i + jQ_i$$

Where, the minus sign takes into account the fact that current is assumed to be injected instead of leaving the node.

Similarly, the power-flow study can show the real and reactive power flowing in every transmission line in the system. The current flow out of a node along a particular transmission line between bus i and bus j can be calculated from the equation

Equation 8

$$I_{ij} = Y_{ij}(V_i - V_j)$$

Where, Y_{ij} is the admittance of the transmission line between those two busses. The resulting real and reactive power can be calculated from the equation

Equation 9

$$S_{ij} = -V_i I_{ij}^* = P_{ij} + jQ_{ij}$$

By comparing the real and reactive power flows at either end of the transmission line, the real and reactive power losses on each line can be determined.

In modern professional power-flow programs, this information is displayed graphically on a computer screen, with real and reactive power flows into and out of each bus and transmission line being shown as arrows. The display uses color to highlight areas where the power system is overloaded. This graphical representation makes it very easy to locate 'hot spots'.

With all of this information available, the user of a power-flow study can determine whether any of the components in a power system will be overloaded by the particular conditions of the study. If the ratings of any components are exceeded, the conditions of this study are an unacceptable steady-state operating condition, and something else must be tried.

Power flow studies help a power system to operate more efficiently. A typical power system has many generators in many different geographical locations and of course the loads on the system are distributed in many different locations. The loads on the power system can be supplied by the generators in many different ways, with the generators at some power stations supplying full power while the generators at other stations serve as swing generators. If power flow studies are performed for different combinations of generation, the engineer can determine which combination produces the minimum transmission losses, and thus provides the power most efficiently. The study of the efficient operation of power systems is called economic dispatch. It is a major sub-discipline within power systems engineering [6].

CHAPTER 3

METHODOLOGY

This project will consist of two major stages which are being used for solving the power flow analysis. There are:

a) Development of the power flow program
b) Application of the power flow program for several case studies

3.1 Power Flow Program Development

The power_flow program is developed by MATLAB software. The power_flow command is made under m-file format.

```
function power_flow(filename)
```

Power_flow function is to perform a power flow analysis. The function of power_flow reads a data set describing a power system from a disk file and performs a power-flow analysis on it. The only argument is the name of the input file. There are three types of lines in the input file:

1. The "SYSTEM" line specifies the name of the power system and the base apparent power of the power system in MVA. Its form is:

```
SYSTEM name baseMVA tol
```

Where
name = The name of the power system
baseMVA = The base MVA of the power system
tol = Voltage tolerance

2. The "BUS" line specifies the characteristics of a bus on the power system. Its form is:

```
BUS name type volts Pgen Qgen Pload Qload Qcap
```

Where
name = The name of the bus
type = The type of the bus, one of PQ, PV, or SL

V_{bus} = The initial voltage guess for PQ busses
The fixed magnitude of voltage PV busses
The fixed magnitude at an angle of 0 deg for SL busses
P_{gen} = Is the real power generation in MW at the bus
Q_{gen} = Is the reactive power generation in MVAR at the bus
P_{load} = Is the real power load in MW at the bus
Q_{load} = Is the reactive power load in MVAR at the bus
Q_{cap} = Is the reactive power of capacitors in MVAR at the bus

3. The "LINE" line specifies the characteristics of a transmission line on the power system. Note that the impedance of the transformers in series with the transmission line should also be lumped into these terms. Its form is:

```
LINE from to Rse Xse Gsh Bsh Rating
```

Where
from = The name of the "from" bus
to = The type of the "to" bus
R_{se} = Per-unit series resistance
X_{se} = Per-unit series reactance
G_{sh} = Per-unit shunt conductance
B_{sh} = Per-unit shunt susceptance
Rating = Max power rating of the line in MVAR
The lines should appear in the order as below:

```
SYSTEM
BUS
LINE
```

The program reads the data from the input file and solves for the voltages at every bus. Then, it generates a report giving the voltages and power flows throughout the system.

To get the power system data
```
[bus, line, system] = read_system(filename);
```

To Build Y_{bus}
```
ybus = build_ybus(bus, line);
```

To Solve for the bus voltages
```
[bus, n_iter] = solve_system(bus, ybus);
```

To Display results
```
report_system(1, bus, line, system, ybus, n_iter);
```

```
==============================================================
function [bus, line, system] = read_system(filename)
```
read_system command use to read a power system from disk.

To check for a legal number of input arguments.

```
msg = nargchk(1,1,nargin);
error(msg);
```

To initialise counters

```
n_system  = 0;        Number of SYSTEM cards
n_bus     = 0;        Number of BUS cards
n_lines   = 0;        Number of LINE cards
n_bad     = 0;        Number of INVALID cards
n_comment = 0;        Number of comment lines
i_line    = 0;        Current line number

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% To Open input file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[fid,message] = fopen(filename,'r');
 FID = -1 for failure.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% To Check for error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if fid < 0
   str = ['ERROR: Can't find system file: ' filename];
   error(str);
else

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % File open OK, so read lines.
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   while feof(fid) == 0

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % Get next line
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      data = fgetl(fid);
      i_line = i_line + 1;

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % Extract keyword
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      maxlen = min( [ 6 length(data) ] );
      keyword = data(1:maxlen);

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % Determine the type of the line
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      if strncmpi(keyword,'SYSTEM',6) == 1

         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
         % This is a SYSTEM card
         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
         n_system = n_system + 1;
```

Get system name

```
blanks = findstr(data,' ');
test = diff(blanks) > 1;
for ii = 1:length(test)
   if test(ii) > 0
      system.name = data(blanks(ii)+1:blanks(ii+1)-1);
      break;
   end
end
```

Get base MVA

```
ii = blanks(ii+1);
temp = sscanf(data(ii:length(data)),'%g');
system.baseMVA = temp(1);
```

Voltage tolerance

```
system.v_tol = temp(2);

elseif strncmpi(keyword,'BUS',3) == 1

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

% This is a BUS card

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n_bus = n_bus + 1;
```

Confirm that a SYSTEM card has preceded this card

```
if n_system == 0
   error (['ERROR: A SYSTEM card must precede any BUS cards!']);
end
```

Get bus name

```
blanks = findstr(data,' ');
test = diff(blanks) > 1;
for ii = 1:length(test)
   if test(ii) > 0
      bus(n_bus).name = data(blanks(ii)+1:blanks(ii+1)-1);
      break;
   end
end
```

Get bus type

```
for ii = ii+1:length(test)
   if test(ii) > 0
      bus(n_bus).type = data(blanks(ii)+1:blanks(ii+1)-1);
      break;
   end
end
```

Get voltage

```
ii = blanks(ii+1);
temp = sscanf(data(ii:length(data)),'%g');
bus(n_bus).Vbus = temp(1);
```

```
   Get power generated, loads, and capactive MVAR
   bus(n_bus).PG = temp(2) / system.baseMVA;
   bus(n_bus).QG = temp(3) / system.baseMVA;
   bus(n_bus).PL = temp(4) / system.baseMVA;
   bus(n_bus).QL = temp(5) / system.baseMVA;
   bus(n_bus).QC = temp(6) / system.baseMVA;

elseif strncmpi(keyword,'LINE',4) == 1

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % This is a LINE card
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   n_lines = n_lines + 1;

   Confirm that a SYSTEM card has preceded this card
   if n_system == 0
      error (['ERROR: A SYSTEM card must precede any LINE cards!']);
   end

   Get "from" bus name
   blanks = findstr(data,' ');
   test = diff(blanks) > 1;
   for ii = 1:length(test)
      if test(ii) > 0
         line(n_lines).from_name = data(blanks(ii)+1:blanks(ii+1)-1);
         break;
      end
   end

   Get "to" bus name
   for ii = ii+1:length(test)
      if test(ii) > 0
         line(n_lines).to_name = data(blanks(ii)+1:blanks(ii+1)-1);
         break;
      end
   end

   Get numeric values
   ii = blanks(ii+1);
   temp = sscanf(data(ii:length(data)),'%g');

   Get values
   line(n_lines).Rse    = temp(1);
   line(n_lines).Xse    = temp(2);
   line(n_lines).Gsh    = temp(3);
   line(n_lines).Bsh    = temp(4);
   line(n_lines).rating = temp(5);

elseif isempty(keyword)

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % This is a null line--do nothing
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %n_comment = n_comment + 1;
```

```
   elseif keyword(1:1) == '%'

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % This is a comment line
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      n_comment = n_comment + 1;

   else

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % This is an invalid line
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      n_bad = n_bad + 1;
      if ischar(data)
         disp(['WARNING: Invalid line ' int2str(i_line) ': ' data]);
      end
   end
end
```

Now, test input data for consistency

```
for ii = 1:n_bus
   bus(ii).n_lines = 0;
end
for ii = 1:n_lines
   line(ii).from = 0;
   line(ii).to = 0;
```

Check for line terminations

```
   for jj = 1:n_bus
      if strcmpi(line(ii).from_name, bus(jj).name)
         bus(jj).n_lines = bus(jj).n_lines + 1;
         line(ii).from   = jj;
      end
      if strcmpi(line(ii).to_name, bus(jj).name)
         bus(jj).n_lines = bus(jj).n_lines + 1;
         line(ii).to     = jj;
      end
   end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now, test input data for consistency.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Check to see that one and only one SYSTEM card was present.

```
error_count = 0;
if n_system == 0
   error_count = error_count + 1;
   disp (['ERROR: No SYSTEM card supplied!']);
end
if n_system > 1
   error_count = error_count + 1;
   disp (['ERROR: Too many SYSTEM cards supplied!']);
```

```
end
```

Check to see that there are no isolated busses.

```
for ii = 1:n_bus
   if bus(ii).n_lines <= 0
      error_count = error_count + 1;
      disp (['ERROR: Isolated bus: ' bus(ii).name]);
   end
end
```

Now, check for lines with invalid bus names.

```
for ii = 1:n_lines
   if line(ii).from <= 0
      error_count = error_count + 1;
      str = ['ERROR: Invalid from bus on line ' num2str(ii) ...
             ': ' line(ii).from_name];
      disp(str);
   end
   if line(ii).to <= 0
      error_count = error_count + 1;
      str = ['ERROR: Invalid  to  bus on line ' num2str(ii) ...
             ': ' line(ii).to_name];
      disp(str);
   end
end
```

Check and see if there was one and only one slack bus.

```
sl_count = 0;
for ii = 1:n_bus
   if bus(ii).type == 'SL'
      sl_count = sl_count + 1;
   end
end
if sl_count == 0
   error_count = error_count + 1;
   disp (['ERROR: No slack bus specified!']);
end
if sl_count > 1
   error_count = error_count + 1;
   disp (['ERROR: Too many slack busses specified!']);
end
```

Check and see if each bus with non-zero generation is either type 'SL' or type 'PV'.

```
for ii = 1:n_bus
   if ~strcmp(bus(ii).type,'SL') & ~strcmp(bus(ii).type,'PV') & ...
      ( (bus(ii).PG ~= 0) | (bus(ii).QG ~= 0) )
      error_count = error_count + 1;
      disp (['ERROR: Generator bus ' int2str(ii) ...
             ' specified as type ' bus(ii).type]);
   end
end
```

If there were errors, abort with error message.

```
if error_count > 0
   disp([int2str(error_count) ' errors total!']);
```

```
   end

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Write out data summary
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   fprintf('Input summary statistics:\n' );
   fprintf('%4d lines in system file\n', i_line );
   fprintf('%4d SYSTEM lines\n', n_system );
   fprintf('%4d BUS lines\n', n_bus );
   fprintf('%4d LINE lines\n', n_lines );

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Close file
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   fclose(fid);

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Abort on errors
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   if error_count > 0
      error(['Job aborted due to input errors.']);
   end
end
```

===

```
function ybus = build_ybus(bus, line)
```

BUILD_YBUS Build the bus admittance matrix from an input array of lines. Function of build_ybus builds a bus admittance matrix from an input structure of lines. Each line stretches between two busses, and it characterized by a per-unit series impedance and a per-unit shunt admittance. Each array element in the "line" array contains the following fields:

from_name	Name of "from" bus
from	Number of "from" bus
to_name	Name of "to" bus
to	Number of "to" bus
R_{se}	Series resistance, pu
X_{se}	Series reactance, pu
G_{sh}	Shunt conductance, pu
B_{sh}	Shunt susceptance, pu
Rating	Rated apparent power (MVA)

Check for a legal number of input arguments.

```
msg = nargchk(2,2,nargin);
error(msg);
```

Get the number of busses included in the system.
```
n_bus = length(bus);
```

Create ybus
```
ybus = zeros(n_bus,n_bus);
```

Now, build the bus admittance matrix
```
for ii = 1:length(line)
```

Get indices
```
  fr = line(ii).from;
  to = line(ii).to;
```

Convert series impedance to a series admittance, and conductance and susceptance to a shunt admittance.
```
  Yse = 1.0 / ( line(ii).Rse + j*line(ii).Xse );
  Ysh = line(ii).Gsh + j*line(ii).Bsh;
```

Diagonal terms
```
  ybus(fr,fr) = ybus(fr,fr) + Yse;
  ybus(to,to) = ybus(to,to) + Yse;
```

Off-diagonal terms
```
  ybus(fr,to) = ybus(fr,to) - Yse;
  ybus(to,fr) = ybus(to,fr) - Yse;
```

Shunt admittance
```
  ybus(fr,fr) = ybus(fr,fr) + Ysh;
  ybus(to,to) = ybus(to,to) + Ysh;
end
```

```
==============================================================
```

```
function [bus, n_iter] = solve_system(bus, ybus)
```

SOLVE_SYSTEM Solve for the bus voltages in the system. Function of solve_system solves for the bus voltage in the power system using the Gauss-Siedel method.

Check for a legal number of input arguments.
```
msg = nargchk(2,2,nargin);
error(msg);
```

Set problem size and initial conditions
```
n_bus = length(bus);
acc_fac = 1.0;
eps = 0.0001;
n_iter = 0;
```

Initialize the real and reactive power supplied to the power system at each bus. Note that the power at the swing bus doesn't matter, and the reactive power at the generator bus will be recomputed dynamically.

```
for ii = 1:n_bus
   bus(ii).P = bus(ii).PG - bus(ii).PL;
   bus(ii).Q = bus(ii).QG - bus(ii).QL + bus(ii).QC;
end
```

Initialize V_{bus}

```
for ii = 1:n_bus
   Vbus(ii) = bus(ii).Vbus;
end
```

Create an infinite loop

```
while (1)
```

Increment the iteration count

```
  n_iter = n_iter + 1;
```

Save old bus voltages for comparison purposes

```
  Vbus_old = Vbus;
```

Calculate the updated bus voltage

```
  for ii = 1:n_bus
```

Skip the swing bus!

```
     if ~strcmpi(bus(ii).type, 'SL')
```

If this is a generator bus, update the reactive power estimate.

```
        if strcmpi(bus(ii).type, 'PV')
           temp = 0;
           for jj = 1:n_bus
              temp = temp + ybus(ii,jj) * Vbus(jj);
           end
           temp = conj(Vbus(ii)) * temp;
           bus(ii).Q = -imag(temp);
        end
```

Calculate updated voltage at bus 'ii'. First, sum up the current contributions at bus 'ii' from all other busses.

```
        temp = 0;
        for jj = 1:n_bus
           if ii ~= jj
              temp = temp - ybus(ii,jj) * Vbus(jj);
           end
        end
```

Add in the current injected at this node

```
        temp = (bus(ii).P - j*bus(ii).Q) / conj(Vbus(ii)) + temp;
```

Get updated estimate of Vbus at 'ii'

```
        Vnew = 1/ybus(ii,ii) * temp;
```

Apply an acceleration factor to the new voltage estimate
```
Vbus(ii) = Vbus_old(ii) + acc_fac * (Vnew - Vbus_old(ii));
```

If this is a generator bus, update the magnitude of the voltage to keep it constant.
```
        if strcmpi(bus(ii).type, 'PV')
           Vbus(ii) = Vbus(ii) * abs(Vbus_old(ii)) / abs(Vbus(ii));
        end
     end
   end
```

Compare the old and new estimate of the voltages. Note that we will compare the real and the imag parts separately, and both must be within tolerances for every bus.
```
   temp = Vbus - Vbus_old;
   if max(abs([real(temp) imag(temp)])) < eps
      break;
   end
end
```

Save the bus voltages in the bus array
```
for ii = 1:n_bus
   bus(ii).Vbus = Vbus(ii);
end
```

```
=================================================================
function report_system(fid, bus, line, system, ybus, n_iter)
```

REPORT_SYSTEM Write a report of power system load flows. Function of report_system writes a load flow report to unit "fid". If this unit is a file, the file must be opened before the function is called.

Check for a legal number of input arguments.
```
msg = nargchk(6,6,nargin);
error(msg);
```

Get number of busses
```
n_bus = length(bus);
```

Reset sums
```
PGtot = 0;
QGtot = 0;
PLtot = 0;
QLtot = 0;
QCtot = 0;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```
% Calculate and display bus and line quantities
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```
Set titles and labels
```
fprintf('\n');
fprintf('                                        Results for Ca');
fprintf('se %s\n', system.name);
fprintf('|===================Bus Information================');
```

```
fprintf('=====================|=======Line Information======|\n');
fprintf('Bus             Bus   Volts / angle  |--Generation--|-');
fprintf('-----Load-----|--Cap--|      To     |----Line Flow---|\n');
fprintf('no.      Name   Type    (pu)    (deg)     (MW)   (MVAR)    ');
fprintf(' (MW)   (MVAR)    (MVAR)        Bus       (MW)     (MVAR)|\n');
fprintf('|================================================');
fprintf('==================================================|\n');
```

Write out bus information

```
for ii = 1:n_bus
```

Write bus number

```
  fprintf('%3d', ii );
```

To write bus name

```
  fprintf(' %10s', bus(ii).name );
```

To write bus type

```
  fprintf('  %2s', bus(ii).type );
```

write pu voltage

```
  [mag, phase] = r2p(bus(ii).Vbus);
  fprintf('   %5.3f/', mag );
  fprintf('%7.2f', phase );
```

If this is a slack bus, update the real and reactive power supplied.

```
  if strcmpi(bus(ii).type, 'SL')
     temp = 0;
     for jj = 1:n_bus
        temp = temp + ybus(ii,jj) * bus(jj).Vbus;
     end
     temp = conj(bus(ii).Vbus) * temp;
     bus(ii).P = real(temp);
     bus(ii).Q = -imag(temp);
  end
```

Write Generation (MW / MVAR)

```
  P = (bus(ii).P + bus(ii).PL) * system.baseMVA;
  Q = (bus(ii).Q + bus(ii).QL - bus(ii).QC) * system.baseMVA;
  fprintf(' %7.2f', P );
  fprintf(' %7.2f', Q );
```

Sum generation

```
  PGtot = PGtot + P;
  QGtot = QGtot + Q;
```

Write Load (MW / MVAR)

```
  fprintf(' %7.2f', bus(ii).PL * system.baseMVA );
  fprintf(' %7.2f', bus(ii).QL * system.baseMVA );
```

Sum loads

```
  PLtot = PLtot + bus(ii).PL * system.baseMVA;
  QLtot = QLtot + bus(ii).QL * system.baseMVA;
```

Write Capacitive MVAR

```
fprintf(' %7.2f', bus(ii).QC * system.baseMVA );
```

Sum capacitive load

```
QCtot = QCtot + bus(ii).QC * system.baseMVA;
```

Calculate the powers flowing out of this bus to other buses

```
count = 0;
for jj = 1:length(line)

   if line(jj).from == ii
```

This line starts at the current bus. To write the "to" bus name

```
      count = count + 1;
      if count > 1
         fprintf(' %84s', line(jj).to_name );
      else
         fprintf(' %10s', line(jj).to_name );
      end
```

Calculate the current and power flow in line

```
      kk = line(jj).to;
      il = (bus(ii).Vbus - bus(kk).Vbus) * ybus(ii,kk);
      pl = bus(ii).Vbus * conj(il);
      P  = -real(pl) * system.baseMVA;
      Q  = -imag(pl) * system.baseMVA;
```

Display power flows in line

```
      fprintf('   %7.2f', P );
      fprintf('   %7.2f\n', Q );

   elseif line(jj).to == ii
```

This line ends at the current bus. To Write the "from" bus name

```
      count = count + 1;
      if count > 1
         fprintf(' %84s', line(jj).from_name );
      else
         fprintf(' %10s', line(jj).from_name );
      end
```

Calculate the current and power flow in line

```
      kk = line(jj).from;
      il = (bus(ii).Vbus - bus(kk).Vbus) * ybus(ii,kk);
      pl = bus(ii).Vbus * conj(il);
      P  = -real(pl) * system.baseMVA;
      Q  = -imag(pl) * system.baseMVA;
```

Display power flows in line

```
      fprintf('   %7.2f', P );
      fprintf('   %7.2f\n', Q );

   end
```

```
   end
end
```

To Write totals

```
fprintf('|=================================================');
fprintf('=================================================|\n');
fprintf('                              Totals        ');
fprintf(' %7.2f', PGtot );
fprintf(' %7.2f', QGtot );
fprintf(' %7.2f', PLtot );
fprintf(' %7.2f', QLtot );
fprintf(' %7.2f\n', QCtot );
fprintf('|=================================================');
fprintf('=================================================|\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

% Calculate and display line losses

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Skip some lines

```
fprintf('\n\n');
```

Set up labels

```
fprintf('                               Line Losses                      \n');
fprintf('|=====================================================|\n');
fprintf('|  Line          From          To      Ploss     Qloss   |\n');
fprintf('|   no.           Bus          Bus      (MW)     (MVAR)  |\n');
fprintf('|=====================================================|\n');
```

Initialize total line loss

```
Pltot = 0;
Qltot = 0;
```

Calculate and write out line losses

```
for ii = 1:length(line);
   jj = line(ii).from;
   kk = line(ii).to;
   il = (bus(jj).Vbus - bus(kk).Vbus) * ybus(jj,kk);
   Pl = abs(il)^2 * line(ii).Rse * system.baseMVA;
   Ql = abs(il)^2 * line(ii).Xse * system.baseMVA;

   Pltot = Pltot + Pl;
   Qltot = Qltot + Ql;
```

Write out lines

```
   fprintf('    %4d',ii);
   fprintf('    %10s',line(ii).from_name);
   fprintf('    %10s',line(ii).to_name);
   fprintf('    %7.2f', Pl );
   fprintf('    %7.2f\n', Ql );
end
```

Write out total line losses

```
fprintf('|=====================================================|\n');
fprintf('                                    Totals:');
fprintf('    %7.2f', Pltot );
```

```
fprintf('   %7.2f\n', Qltot );
fprintf('|=====================================================|\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate and display alerts
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Skip some lines
fprintf('\n\n');

Initialize alert counter
n_alert = 0;

Set up labels
fprintf('                    Alerts                    \n');
fprintf('|=====================================================|\n');

Check for voltages out of range
for ii = 1:length(bus)
   if (abs(bus(ii).Vbus) < (1.0 - system.v_tol) ) | ...
      (abs(bus(ii).Vbus) > (1.0 + system.v_tol) )
      n_alert = n_alert + 1;
      fprintf('ALERT: Voltage on bus %s out of tolerance.\n',bus(ii).name);
   end
end

Check for power lines whose ratings are exceeded.
for ii = 1:length(line);
   jj = line(ii).from;
   kk = line(ii).to;
   il = (bus(jj).Vbus - bus(kk).Vbus) * ybus(jj,kk);
   Sl = abs(bus(jj).Vbus * conj(il)) * system.baseMVA;

Check for line exceeding limit
   if Sl > line(ii).rating
      n_alert = n_alert + 1;
      fprintf('ALERT: Rating on line %d exceeded: %.2f MVA > %.2f MVA.\n', ...
              ii, Sl, line(ii).rating );
   end
end

Write out "none" if not alerts were generated
if n_alert == 0
   fprintf('NONE\n');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display number of iterations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('\nDone in %d iterations.\n', n_iter );

============================================================
function outval = p2r(mag,phase)
```

p2r Convert a complex number in polar format to rectangular. Function p2r converts either a complex number in polar format with the angle in degrees into a complex number in rectangular format.

Define variables:
mag -- Magnitude of number
outval -- Output value in rectangular form
phase -- Phase of number, in degrees

Check for a legal number of input arguments.
```
msg = nargchk(2,2,nargin);
error(msg);
```

Convert number
```
theta = phase * pi / 180;
outval = mag * ( cos(theta) + j*sin(theta) );
```

```
================================================================
function [mag,phase] = r2p(inval)
```
R2P Convert a complex number in rectangular format to polar in degrees. Function R2P converts either a complex number into a complex number in polar format, with the angle in degrees.

Define variables:
inval -- Input value in rectangular form
mag -- Magnitude of number
phase -- Phase of number, in degrees

To Check for a legal number of input arguments.
```
msg = nargchk(1,1,nargin);
error(msg);
```

To Convert number
```
mag = abs(inval);
phase = angle(inval) * 180 / pi;
```
phase = angle(inval) * 180 / pi;

3.2 Application of the power_flow program

After the power_flow program development, the program is then will be run by MATLAB. There are two types of data that should be run together:

a) Power_flow program- is the power flow program that has been developed to solve the power flow equation
b) Input file of the power system- is the power system data load into the program which are consist of power system specification including the bus data and line data

The power_flow program which has been developed should be in the same directory in the MATLAB folder together with the input data and then run the system in command window in the format of: " file_name>space bar>input file" .

There are two case studies has been carried out for this project:

1) Case 1- Test on IEEE 30 bus standard system
2) Case 2- Power System with five busses and six transmission lines

3.2.1 Case 1

IEEE 30-Bus Standard System Test

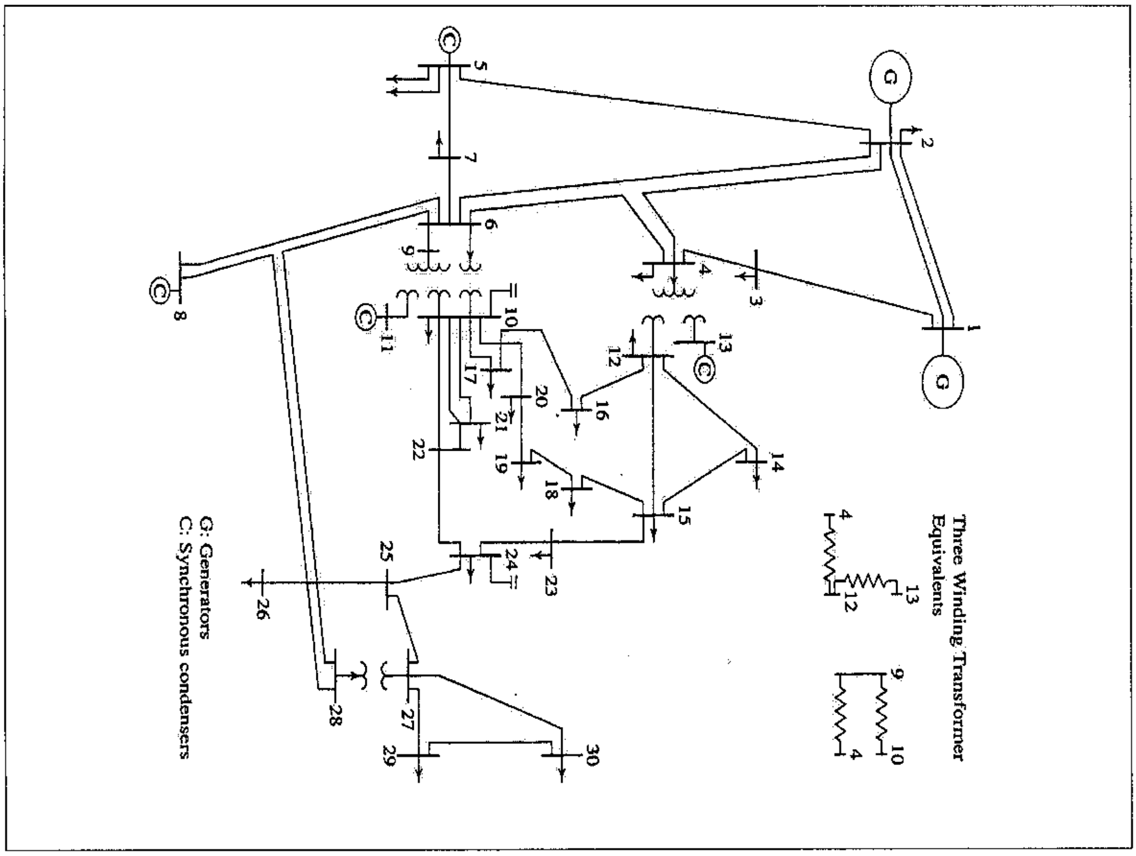

Figure 1: IEEE 30 bus power system network

Table 1: IEEE 30 bus - Bus Data

Bus No	Bus Type	Voltage Mag.	Angle Degree	Generator				Load		Injected MVAR
				MW	MVAR	Qmin	Qmax	MW	MVAR	
1	SL	1.06	0	0	0	0	0	0	0	0
2	PV	1.043	0	40	0	-40	50	21.7	12.7	0
3	PQ	1.0	0	0	0	0	0	2.4	1.2	0
4	PQ	1.06	0	0	0	0	0	7.6	1.6	0
5	PV	1.01	0	0	0	-40	40	94.2	19	0
6	PQ	1.0	0	0	0	0	0	0	0	0
7	PQ	1.0	0	0	0	0	0	22.8	10.9	0
8	PV	1.01	0	0	0	-10	40	30	30	0
9	PQ	1.0	0	0	0	0	0	0	0	0
10	PQ	1.0	0	0	0	0	0	5.8	2	0
11	PV	1.082	0	0	0	-6	24	0	0	19
12	PQ	1.0	0	0	0	0	0	11.2	7.5	0
13	PV	1.071	0	0	0	-6	24	0	0	0
14	PQ	1.0	0	0	0	0	0	6.2	1.6	0
15	PQ	1.0	0	0	0	0	0	8.2	2.5	0
16	PQ	1.0	0	0	0	0	0	3.5	1.8	0
17	PQ	1.0	0	0	0	0	0	9.0	5.8	0
18	PQ	1.0	0	0	0	0	0	3.2	0.9	0
19	PQ	1.0	0	0	0	0	0	9.5	3.4	0
20	PQ	1.0	0	0	0	0	0	2.2	0.7	0
21	PQ	1.0	0	0	0	0	0	17.5	11.2	0
22	PQ	1.0	0	0	0	0	0	0	0	0
23	PQ	1.0	0	0	0	0	0	3.2	1.6	0
24	PQ	1.0	0	0	0	0	0	8.7	6.7	4.3
25	PQ	1.0	0	0	0	0	0	0	0	0
26	PQ	1.0	0	0	0	0	0	3.5	2.3	0
27	PQ	1.0	0	0	0	0	0	0	0	0
28	PQ	1.0	0	0	0	0	0	0	0	0
29	PQ	1.0	0	0	0	0	0	2.4	0.9	0
30	PQ	1.0	0	0	0	0	0	10.6	1.9	0

Table 2: IEEE 30 bus - Line Data

BUS		R	X	½ B	1 for Line code or pu
From bus	To bus	(pu)	(pu)	(pu)	tap setting value
1	2	0.0192	0.0575	0.02640	1
1	3	0.0452	0.1852	0.02040	1
2	4	0.0570	0.1737	0.01840	1
3	4	0.0132	0.0379	0.00420	1
2	5	0.0472	0.1983	0.02090	1
2	6	0.0581	0.1763	0.01870	1
4	6	0.0119	0.0414	0.00450	1
5	7	0.0460	0.1160	0.01020	1
6	7	0.0267	0.0820	0.00850	1
6	8	0.0120	0.0420	0.00450	1
6	9	0	0.2080	0	0.978
6	10	0	0.5560	0	0.969
9	11	0	0.2080	0	1
9	10	0	0.1100	0	1
4	12	0	0.2560	0	0.932
12	13	0	0.1400	0	1
12	14	0.1231	0.2559	0	1
12	15	0.0662	0.1304	0	1
12	16	0.0945	0.1987	0	1
14	15	0.2210	0.1997	0	1
16	17	0.0824	0.1923	0	1
15	18	0.1073	0.2185	0	1
18	19	0.0639	0.1292	0	1
19	20	0.0340	0.0680	0	1
10	20	0.0936	0.2090	0	1
10	17	0.0324	0.0845	0	1
10	21	0.0348	0.0749	0	1
10	22	0.0727	0.1499	0	1
21	22	0.0116	0.0236	0	1
15	23	0.1000	0.2020	0	1
22	24	0.1150	0.1790	0	1
23	24	0.1320	0.2700	0	1

24	25	0.1885	0.3292	0	1
25	26	0.2544	0.3800	0	1
25	27	0.1093	0.2087	0	1
28	27	0	0.3960	0	0.968
27	29	0.2198	0.4153	0	1
27	30	0.3202	0.6027	0	1
29	30	0.2399	0.4533	0	1
8	28	0.0636	0.2000	0.0214	1
6	28	0.0169	0.0599	0.065	1

3.2.2 Case 2

Figure 2 below shows the simple power system with five busses and six transmission lines. The base apparent power of this power system is 100 MVA and the tolerance on each bus is 5%.

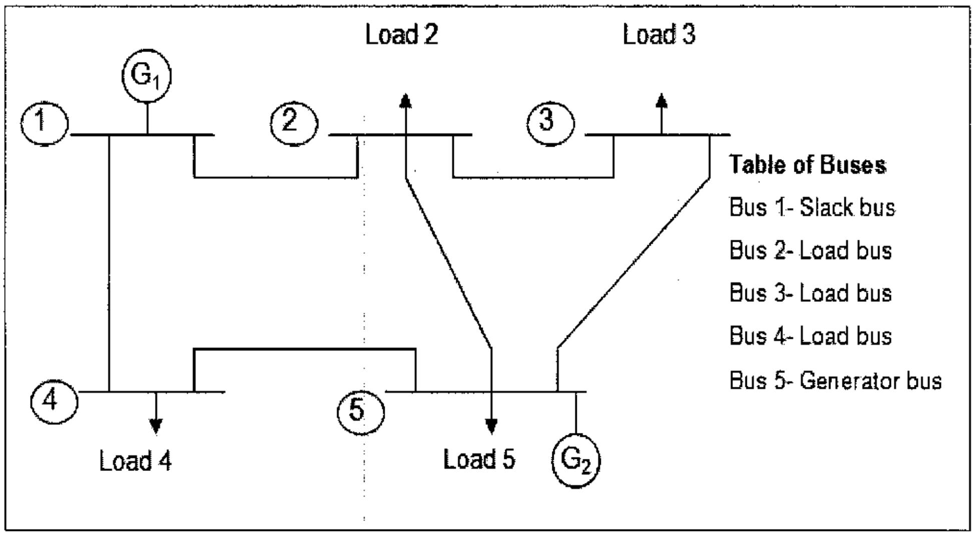

Figure 2: Power system network

Table 3: Bus Data

Bus name	Type	V(pu)	Generation		Loads	
			P(MW)	Q(MVAR)	P(MW)	Q(MVAR)
1	SL	$1\angle 0^o$				
2	PQ	$1\angle 0^o$			60	35
3	PQ	$1\angle 0^o$			70	40
4	PQ	$1\angle 0^o$			80	50
5	PV	$1\angle 0^o$	190		40	30

Table 4: Line Data

Transmission line number	From/to (bus to bus)	Series Impedance Z(pu)	Rated MVA
1	1-2	0.0210 + j0.1250	50
2	1-4	0.0235 + j0.0940	100
3	2-3	0.0250 + j0.1500	50
4	2-5	0.0180 + j0.0730	100
5	3-5	0.0220 + j0.1100	100
6	4-5	0.0190 + j0.0800	100

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results for Case 1

In order to validate the power_flow program which has been developed by MATLAB software, the IEEE 30-bus standard test system and several case studies are carried out.

Bus Data

Results for Case Base_case

Bus Information										Line Information		
Bus		Bus	Volts /	angle	Generation		Load		Cap	To	Line Flow	
no.	Name	Type	(pu)	(deg)	(MW)	(MVAR)	(MW)	(MVAR)	(MVAR)	Bus	(MW)	(MVAR)
1	One	SL	1.060/	0.00	254.66	-12.11	0.00	0.00	0.00	Two	173.59	-18.21
										Three	81.07	6.10
2	Two	PV	1.043/	-5.36	40.00	54.44	21.70	12.70	0.00	One	-168.38	33.80
										Four	44.37	2.57
										Five	82.38	4.03
										Six	60.54	1.51
3	Three	PQ	1.024/	-7.80	0.00	0.00	2.40	1.20	0.00	One	-78.41	4.79
										Four	76.86	-6.18
4	Four	PQ	1.017/	-9.45	0.00	0.00	7.60	1.60	0.00	Two	-43.34	0.58
										Three	-76.11	8.33
										Six	70.57	-6.36
										Twelve	41.83	-4.17
5	Five	PV	1.010/	-14.18	0.00	39.90	94.20	19.00	0.00	Two	-79.43	8.37
										Seven	-14.55	12.61
6	Six	PQ	1.012/	-11.12	0.00	0.00	0.00	0.00	0.00	Two	-58.58	4.43
										Four	-70.00	8.37
										Seven	38.27	-0.24
										Eight	29.79	-3.93
										Nine	27.63	-10.32
										Ten	15.66	-1.35
										Twenty8	18.80	2.73
7	Seven	PQ	1.002/	-12.89	0.00	0.00	22.80	10.90	0.00	Five	14.71	-12.18
										Six	-37.88	1.41
8	Eight	PV	1.010/	-11.85	0.00	35.53	30.00	30.00	0.00	Six	-29.69	4.30
										Twenty8	-0.55	1.63
9	Nine	PQ	1.035/	-14.27	0.00	0.00	0.00	0.00	0.00	Six	-27.63	12.09
										Eleven	0.16	-23.58
										Ten	28.19	11.46

10	Ten	PQ	1.023/ -15.94	0.00	0.00	5.80	2.00	19.00	Six	-15.66	2.69
									Nine	-28.19	-10.51
									Twenty	9.43	3.73
									Seventeen	6.36	4.65
									Twenty1	15.79	10.65
									Twenty2	7.86	4.88
11	Eleven	PV	1.082/ -14.28	0.00	24.66	0.00	0.00	0.00	Nine	-0.16	24.66
12	Twelve	PQ	1.033/ -15.30	0.00	0.00	11.20	7.50	0.00	Four	-41.83	8.55
									Thirteen	0.24	-28.08
									Fourteen	7.65	2.37
									Fifeteen	17.22	6.35
									Sixteen	6.61	2.91
13	Thirteen	PV	1.071/ -15.32	0.00	29.12	0.00	0.00	0.00	Twelve	-0.24	29.12
14	Fourteen	PQ	1.018/ -16.21	0.00	0.00	6.20	1.60	0.00	Twelve	-7.58	-2.21
									Fifeteen	1.39	0.52
15	Fifeteen	PQ	1.014/ -16.30	0.00	0.00	8.20	1.50	0.00	Twelve	-17.01	-5.93
									Fourteen	-1.39	-0.52
									Eighteen	5.51	1.66
									Twenty3	4.85	3.26
16	Sixteen	PQ	1.021/ -15.86	0.00	0.00	3.50	1.80	0.00	Twelve	-6.57	-2.81
									Seventeen	3.17	0.96
17	Seventeen	PQ	1.017/ -16.16	0.00	0.00	9.00	5.80	0.00	Sixteen	-3.16	-0.94
									Ten	-6.34	-4.60
18	Eighteen	PQ	1.005/ -16.88	0.00	0.00	3.20	0.90	0.00	Fifeteen	-5.48	-1.59
									Nineteen	2.51	0.55
19	Nineteen	PQ	1.002/ -17.04	0.00	0.00	9.50	3.40	0.00	Eighteen	-2.51	-0.54
									Twenty	-6.58	-3.11
20	Twenty	PQ	1.007/ -16.85	0.00	0.00	2.20	0.70	0.00	Nineteen	6.59	3.15
									Ten	-9.33	-3.52
21	Twenty1	PQ	1.010/ -16.39	0.00	0.00	17.50	11.20	0.00	Ten	-15.67	-10.40
									Twenty2	-0.34	-1.71
22	Twenty2	PQ	1.010/ -16.40	0.00	0.00	0.00	0.00	0.00	Ten	-7.80	-4.75
									Twenty1	0.34	1.71
									Twenty4	6.30	3.72
23	Twenty3	PQ	1.003/ -16.67	0.00	0.00	3.20	1.60	0.00	Fifeteen	-4.81	-3.19
									Twenty4	1.63	1.58
24	Twenty4	PQ	0.996/ -16.80	0.00	0.00	8.70	6.70	4.30	Twenty2	-6.24	-3.63
									Twenty3	-1.62	-1.57
									Twenty5	-0.95	2.88
25	Twenty5	PQ	0.989/ -16.30	0.00	0.00	0.00	0.00	0.00	Twenty4	0.97	-2.85
									Twenty6	3.60	2.33
									Twenty7	-4.40	0.41
26	Twenty6	PQ	0.971/ -16.76	0.00	0.00	3.50	2.30	0.00	Twenty5	-3.55	-2.26
27	Twenty7	PQ	0.993/ -15.74	0.00	0.00	0.00	0.00	0.00	Twenty5	4.43	-0.37
									Twenty8	-17.67	-2.98
									Twenty9	6.21	1.67
									Thirty	7.14	1.66
28	Twenty8	PQ	1.007/ -11.73	0.00	0.00	0.00	0.00	0.00	Twenty7	17.67	4.27
									Eight	0.56	-1.63
									Six	-18.74	-2.52
29	Twenty9	PQ	0.972/ -17.05	0.00	0.00	2.40	0.90	0.00	Twenty7	-6.12	-1.50
									Thirty	3.74	0.59
30	Thirty	PQ	0.960/ -18.01	0.00	0.00	10.60	1.90	0.00	Twenty7	-6.97	-1.33
									Twenty9	-3.70	-0.52
			Totals	294.66	171.53	283.40	125.20	23.30			

Line Losses

Line Losses

Line no.	From Bus	To Bus	Ploss (MW)	Qloss (MVAR)
1	One	Two	5.21	15.59
2	One	Three	2.66	10.89
3	Two	Four	1.04	3.15
4	Three	Four	0.75	2.15
5	Two	Five	2.95	12.40
6	Two	Six	1.96	5.94
7	Four	Six	0.58	2.01
8	Five	Seven	0.17	0.42
9	Six	Seven	0.38	1.17
10	Six	Eight	0.11	0.37
11	Six	Nine	0.00	1.77
12	Six	Ten	0.00	1.34
13	Nine	Eleven	0.00	1.08
14	Nine	Ten	0.00	0.95
15	Four	Twelve	0.00	4.37
16	Twelve	Thirteen	0.00	1.03
17	Twelve	Fourteen	0.07	0.15
18	Twelve	Fifeteen	0.21	0.41
19	Twelve	Sixteen	0.05	0.10
20	Fourteen	Fifeteen	0.00	0.00
21	Sixteen	Seventeen	0.01	0.02
22	Fifeteen	Eighteen	0.03	0.07
23	Eighteen	Nineteen	0.00	0.01
24	Nineteen	Twenty	0.02	0.04
25	Ten	Twenty	0.09	0.21
26	Ten	Seventeen	0.02	0.05
27	Ten	Twenty1	0.12	0.26
28	Ten	Twenty2	0.06	0.12
29	Twenty1	Twenty2	0.00	0.00
30	Fifeteen	Twenty3	0.03	0.07
31	Twenty2	Twenty4	0.06	0.09
32	Twenty3	Twenty4	0.01	0.01
33	Twenty4	Twenty5	0.02	0.03
34	Twenty5	Twenty6	0.05	0.07
35	Twenty5	Twenty7	0.02	0.04
36	Twenty8	Twenty7	0.00	1.29
37	Twenty7	Twenty9	0.09	0.17
38	Twenty7	Thirty	0.17	0.33
39	Twenty9	Thirty	0.04	0.07
40	Eight	Twenty8	0.00	0.01
41	Six	Twenty8	0.06	0.21
		Totals:	17.03	68.49

Alerts

NONE

Done in 32 iterations.

Comments: The iteration done at 32 iterations. The power_flow program can be executed with IEEE 30 Standard Sample System Test. Therefore, the power_flow program is validated for analyzing others Power Flow System.

4.2 Results for Case 2

1) Find

   a) The voltages at each bus in the power system

   b) The real and reactive power flows in each transmission line

Table 5: Bus Voltages at each bus in the power system, solution for a)

Results for Case 2

Bus no.	Name	Bus Type	Volts / (pu)	angle (deg)	Generation (MW)	Generation (MVAR)	Load (MW)	Load (MVAR)	Cap (MVAR)	To Bus	Line Flow (MW)	Line Flow (MVAR)
1	One	SL	1.000/	0.00	62.55	44.63	0.00	0.00	0.00	Two	26.52	19.76
										Four	36.03	24.88
2	Two	PQ	0.970/	-1.71	0.00	0.00	60.00	35.00	0.00	One	-26.29	-18.39
										Three	15.97	10.10
										Five	-49.52	-26.74
3	Three	PQ	0.951/	-3.04	0.00	0.00	70.00	40.00	0.00	Two	-15.87	-9.54
										Five	-54.07	-30.47
4	Four	PQ	0.969/	-1.66	0.00	0.00	80.00	50.00	0.00	One	-35.58	-23.07
										Five	-44.34	-26.95
5	Five	PV	1.000/	0.14	190.00	123.56	40.00	30.00	0.00	Two	50.13	29.20
										Three	55.01	35.16
										Four	44.88	29.24
			Totals		252.55	168.19	250.00	155.00	0.00			

Table 6: Real and Reactive power flow data, solution for b)

Line Losses

Line no.	From Bus	To Bus	Ploss (MW)	Qloss (MVAR)
1	One	Two	0.23	1.37
2	One	Four	0.45	1.80
3	Two	Three	0.09	0.57
4	Two	Five	0.61	2.46
5	Three	Five	0.94	4.69
6	Four	Five	0.55	2.30
		Totals:	2.86	13.18

Alerts

NONE

Done in 12 iterations.

   c) Are any of the bus voltages out of tolerance in the power system?

   No bus voltages out of tolerance

   d) Are any of the transmission lines overloaded?

   No transmission lines overloaded

2) If transmission line 3 in the previous problem (between busses 2 and 3) is open circuited for maintenance. Find the bus voltages and transmission line powers in the power system with the line removed. Are any of the voltages out of tolerance? Are any of the transmission lines overloaded?

(Note: refer the power system input data in appendices)

Voltage on bus Three is out of tolerance. Rating on line 4 exceeded: 80.58 MVA > 70.00 MVA.

From **Table 3** the voltage bus Three is out of tolerance. The voltage bus tolerance can be calculated by using this formula:

% of Tolerance

$$= \frac{V_{old} - V_{new}}{V_{old}} \times 100$$

$$= \frac{1.0 - 0.943}{1.0} \times 100$$

$$= 6.6\%$$

While, the rating of power flow on each bus can be calculated as below:

Specified bus rating, S_{rated} = x

New bus power, S_{new} = y

From **Table 3**, S_{new} can be computed by the addition of the Real, P and Reactive power, Q for each bus:

$$y = \sqrt{P^2 + Q^2}$$

If y > x, rating of the line is exceeded. For case 1, rating on line 4 (S_{rated}=70MVA) exceeded, since we know the bus 3 is out of tolerance. The rating can be computed:

$$y = \sqrt{(-69.95)^2 + (-40.01)^2} = 80.58MVA$$

80.58MVA > 70MVA .Hence, it can be concluded that the line is overloaded.

Table 7: Bus Voltages at each bus in the power system for no 2)

Results for Case 2

Bus no.	Name	Bus Type	Volts (pu)	angle (deg)	Generation (MW)	Generation (MVAR)	Load (MW)	Load (MVAR)	Cap (MVAR)	To Bus	Line Flow (MW)	Line Flow (MVAR)
1	One	SL	1.000/	0.00	62.90	38.85	0.00	0.00	0.00	Two	23.61	14.66
										Four	39.29	24.19
2	Two	PQ	0.977/	-1.55	0.00	0.00	60.00	35.00	0.00	One	-23.45	-13.69
										Five	-36.46	-21.33
3	Three	PQ	0.934/	-4.40	0.00	0.00	70.00	40.00	0.00	Five	-69.95	-40.01
4	Four	PQ	0.969/	-1.85	0.00	0.00	80.00	50.00	0.00	One	-38.79	-22.19
										Five	-41.14	-27.83
5	Five	PV	1.000/	-0.21	190.00	130.78	40.00	30.00	0.00	Two	36.80	22.69
										Three	71.59	48.20
										Four	41.63	29.93
			Totals		252.90	169.63	250.00	155.00	0.00			

Table 8: Real and Reactive power flow data for no 2)

Line Losses

Line no.	From Bus	To Bus	Ploss (MW)	Qloss (MVAR)
1	One	Two	0.16	0.97
2	One	Four	0.50	2.00
3	Two	Five	0.34	1.36
4	Three	Five	1.64	8.19
5	Four	Five	0.50	2.10
		Totals:	3.14	14.63

Alerts

ALERT: Voltage on bus Three out of tolerance.
ALERT: Rating on line 4 exceeded: 80.58 MVA > 70.00 MVA

Done in 10 iterations.

3) Suppose that a 40-MVAR capacitor bank is added to bus 3 of the power system in previous problem. What happens to the bus voltages in this power system? What happens to the apparent powers of the transmission lines? It this situation better or worse that the one in previous problem no. 2?

(Note: refer the power system data to appendices case 2)

The apparent powers of the transmission lines recover back to the normal operation. The situation is better than the on in previous problem no. 2.

Table 9: Bus Voltages at each bus in the power system for no 3)

Results for Case 2

Bus no.	Name	Bus Type	Volts / (pu)	angle (deg)	Generation (MW)	Generation (MVAR)	Load (MW)	Load (MVAR)	Cap (MVAR)	To Bus	Line Flow (MW)	Line Flow (MVAR)
1	One	SL	1.000/	0.00	62.41	38.94	0.00	0.00	0.00	Two	23.38	14.70
										Four	39.03	24.24
2	Two	PQ	0.977/	-1.53	0.00	0.00	60.00	35.00	0.00	One	-23.22	-13.74
										Five	-36.71	-21.27
3	Three	PQ	0.981/	-4.68	0.00	0.00	70.00	40.00	40.00	Five	-69.95	-0.01
4	Four	PQ	0.969/	-1.83	0.00	0.00	80.00	50.00	0.00	One	-38.53	-22.26
										Five	-41.40	-27.76
5	Five	PV	1.000/	-0.19	190.00	88.09	40.00	30.00	0.00	Two	37.05	22.65
										Three	71.07	5.60
										Four	41.90	29.88
		Totals			252.41	127.03	250.00	155.00	40.00			

Table 10: Real and Reactive power flow data for no 3)

Line Losses

Line no.	From Bus	To Bus	Ploss (MW)	Qloss (MVAR)
1	One	Two	0.16	0.95
2	One	Four	0.50	1.98
3	Two	Five	0.34	1.38
4	Three	Five	1.12	5.59
5	Four	Five	0.50	2.12
		Totals:	2.62	12.02

Alerts

NONE

Done in 10 iterations.

4) Assuming that the power system is restored to its original configuration. A new plant consuming 20MW at 0.95 PF lagging is to be added to bus 4. Will the new load case any problems for the power system? If the new load will cause problems, what solution could be recommended?

(Note: refer the power system data to appendices)

It won't cause any problems when the new load case is added. If the new load cause problem the solution that could be recommended is by adding the capacitor bank at bus 4.

Table 11: Bus Voltages at each bus in the power system for no 4)

Results for Case 2

Bus no.	Name	Bus Type	Volts (pu)	angle (deg)	Generation (MW)	Generation (MVAR)	Load (MW)	Load (MVAR)	Cap (MVAR)	To Bus	Line Flow (MW)	Line Flow (MVAR)
1	One	SL	1.000/	0.00	83.08	46.59	0.00	0.00	0.00	Two	32.01	18.93
										Four	51.07	27.65
2	Two	PQ	0.970/	-2.13	0.00	0.00	60.00	35.00	0.00	One	-31.72	-17.20
										Three	17.15	9.95
										Five	-45.30	-27.77
3	Three	PQ	0.951/	-3.57	0.00	0.00	70.00	40.00	0.00	Two	-17.05	-9.32
										Five	-52.90	-30.69
4	Four	PQ	0.963/	-2.47	0.00	0.00	100.00	56.60	0.00	One	-50.28	-24.48
										Five	-49.65	-32.13
5	Five	PV	1.000/	-0.47	190.00	130.31	40.00	30.00	0.00	Two	45.84	29.96
										Three	53.81	35.24
										Four	50.36	35.15
			Totals		273.08	176.89	270.00	161.60	0.00			

Table 12: Real and Reactive power flow data for no 4)

Line Losses

Line no.	From Bus	To Bus	Ploss (MW)	Qloss (MVAR)
1	One	Two	0.29	1.73
2	One	Four	0.79	3.17
3	Two	Three	0.10	0.63
4	Two	Five	0.54	2.19
5	Three	Five	0.91	4.55
6	Four	Five	0.72	3.02
		Totals:	3.35	15.28

Alerts

NONE

Done in 15 iterations.

4.3 Findings

The G-S method is also known as the method of successive displacements. By applying the G-S algorithms to perform the iteration process, it can be seen that the Gauss-Seidel method needs many iterations to achieve the desired accuracy, and there is no guarantee for the convergence. There are many types of method for solving simultaneous nonlinear algebraic equations. Different method gives different values and performance.

In general, the solution that a nonlinear system of equations converges to depends on the starting point for the analysis. The closer the starts to the correct answer, the more likely the equations are to converge to it. A flat start which all busses assumed to be $1\angle 0°$ is pretty close to the correct answer for most power systems, because power systems are designed to keep voltage reasonably constant for the user. However, sometimes with a flat start, the proper answer fails to converge. In that case, it could possibly on trying different starting conditions to be solved with convergence to the correct answer.

Since, some sets of power system equations and starting conditions can iterate forever without coming to a solution, a good power network analysis program should include a maximum number of iterations, and should not stop running if that number of iterations is reached without converging to a solution.

A major advantage of the G-S iterative method is that it is relatively stable and usually results in the right answer. Unfortunately, the method converges to a solution relatively slowly, requiring a lot of computer time. Fortunately, there are a couple of simple tricks that can be applied to speed up the convergence process. These tricks are described below:

1. *Applying updated bus voltage values immediately.* The first trick is to begin using the new estimates of bus voltage as soon as they are calculated, instead of waiting to the beginning of the next iteration to apply them.

2. *Applying acceleration factors.* Studies have shown that the GS method moves bus voltage estimates slowly toward optimum values, with each step tending to be in the same direction as the previous one. If the difference between the new voltage estimate and the old voltage estimate is multiplied by an acceleration factor and added to the old voltage estimate, then in a single step it can jump closer to the true solution than would possible with the ordinary GS method. Normally, the acceleration factor must be greater than 1.0 but if it is too high (higher than 2.0), it will be unstable and convergence will never occur.

Table 13 below is shown the effect of varying the acceleration factor against the number of iteration for *Case 1*:

Table 13: Effect of varying acceleration factor to the number of iteration

Acceleration Factor	Number of Iteration
1.0	82
1.1	73
1.2	65
1.3	56
1.4	48
1.5	40
1.6	32
1.7	23
1.8	25
1.9	61
2.0	7070

In addition, speed-ups are possible by using more advanced solution techniques such as the Newton-Raphson method.

In the Gauss-Seidel method, the updated values of the variables calculated in the preceding equations are immediately used in the solution of the subsequent equations. At the end of each iteration stage, the calculated values of all variables are tested against the previous values. If all changes in the variables are within the specified accuracy, a solution has converged; otherwise iteration must be performed.

The voltage that was calculated is used in the **power_flow** program by MATLAB. The program calculates the voltages and powers at every bus, plus the real and reactive power flowing on each transmission line. It also provides alerts if any of the bus voltages are out of tolerance, or if any of the transmission lines are overloaded.

The input data for program is placed in an input file, which can be created using any available editor. The file must contain three types of lines; a SYSTEM line to define the system me and base MVA, BUS lines to define the initial voltage estimates and the power flows at each bus, and LINE lines to define the transmission lines connecting the various busses together. It may also contain comment lines. Instead of using the **power_flow** analysis coding, the own program can be also created to solve the power flow equation based on understanding.

4.4 Discussions

Load busses are busses that supply power to loads and the amount of power supplied will be whatever the loads demand. The voltage on a load bus in a real power system will go up and down with changing loads. The load busses have specified values of P and Q, while V varies with load conditions.

Real generators operate most efficiently when they are running at full load, so power systems try to keep all but one of a few generators running at 100 percent capacity, while allowing the remaining generator called the swing generator to handle increases and decreases in load demand. Thus, most busses with generators attached to them will be supplying a fixed amount of power and the magnitude of their voltages will be maintained constant by field circuits of the generators. These generators have specified values of P and $|V_i|$.

Finally, the controls on the swing generator will be adjusted to maintain a constant power system voltage and frequency, allowing real and reactive power to increase or decrease as required whenever loads change. The bus that the swing generator is connected to is called the slack bus.

In reality, there sometimes that the future load conditions for the system are unacceptable, so there is a need to make a recommendation for changes to solve the problem. It can be possibly done by many different solutions, such as adding additional generation and a new bus or adding transmission lines, but these solutions are very expensive. Before embarking on them, we must be ensuring that there is no cheaper way to solve the problem.

One possible way to improve the situation is to add capacitor banks to one or more busses in the power system. Capacitor banks supply reactive power to the system at the point where they are connected, so that the reactive power does not have to be supplied by the generators through the transmission lines. This action both reduces the current flow in the lines and increases the voltage nearby busses.

CHAPTER 5

CONCLUSION

Power flow studies are an important part of power system analysis which necessary for planning, economic scheduling and control of an existing system as well as planning its future expansion.

The analysis of a power system in normal steady-state operation is known as a power flow study. Power flow studies attempt to determine the voltages, currents, and real and reactive power flows in a power system under a given set of load conditions.

The power flow study techniques are based on the admittance bus matrix Y_{bus}. The busses on a power system are classified as load busses, generator busses and the slack bus. Real and reactive power flows are specified at load busses, real power and voltage magnitude are specified at generator busses and both the voltage magnitude and angle are specified at the slack bus which usually $1\angle 0°$ pu.

The voltages in the power system are then solved using G-S iteration method. In this method, the voltage at each bus is calculated from a knowledge power flows and an estimate of all the other bus voltages in the system. As long as initial values for bus voltage are reasonably close to the correct answers, the iteration process will usually converge to a correct answer. However, since the network system equations are fundamentally nonlinear, they can converge to erroneous solutions or even fail to converge at all.

REFERENCES

[1] Editor: Ricard C. Dorf, 'The Electrical Engineering Handbook', CRC Press, IEEE Press, pp. 1486.

[2] Arthur R. Bergen and Vijay Vittal, 'Power Systems Analysis', pp. 323

[3] Stephen J. Chapman, 'Electric Machinery and Power System Fundamentals', McGraw-Hill International Edition 2002, pp. 512-514, 537-538.

[4] 'Wiley Encyclopedia of Electrical and Electronics Engineering', vol. 6, pp.681.

[5] Hadi Saadat, 'Power System Analysis', McGraw Hill, 2^{nd} Edition, pp. 195-247)

[6] William D. Stevenson, Jr., 'Elements of Power System Analysis', Mc Graw Hill, 4th edition, pp. 194-196.

APPENDICES

APPENDIX CASE 1

30 IEEE Bus Standard Test System Data

```
% System data has the form:

%SYSTEM     name        baseMVA    Voltage Tolerance
SYSTEM       1           100              0.1
%
% Bus data has the form:
%BUS name         type  volts  Pgen   Qgen   Pload  Qload  Qcap
BUS One            SL   1.06    0      0      0      0      0
BUS Two            PV   1.043   40     0      21.7   12.7   0
BUS Three          PQ   1.00    0      0      2.4    1.2    0
BUS Four           PQ   1.06    0      0      7.6    1.6    0
BUS Five           PV   1.01    0      0      94.2   19     0
BUS Six            PQ   1.00    0      0      0      0      0
BUS Seven          PQ   1.00    0      0      22.8   10.9   0
BUS Eight          PV   1.01    0      0      30     30     0
BUS Nine           PQ   1.00    0      0      0      0      0
BUS Ten            PQ   1.00    0      0      5.8    2      0
BUS Eleven         PV   1.082   0      0      0      0      19
BUS Twelve         PQ   1.043   0      0      11.2   7.5    0
BUS Thirteen       PV   1.071   0      0      0      0      0
BUS Fourteen       PQ   1.00    0      0      6.2    1.6    0
BUS Fifteen        PQ   1.00    0      0      8.2    2.5    0
BUS Sixteen        PQ   1.00    0      0      3.5    1.8    0
BUS Seventeen      PQ   1.00    0      0      9.0    5.8    0
BUS Eighteen       PQ   1.00    0      0      3.2    0.9    0
BUS Nineteen       PQ   1.00    0      0      9.5    3.4    0
BUS Twenty         PQ   1.00    0      0      2.2    0.7    0
BUS Twenty1        PQ   1.06    0      0      17.5   11.2   0
BUS Twenty2        PQ   1.00    0      0      0      0      0
BUS Twenty3        PQ   1.00    0      0      3.2    1.6    0
BUS Twenty4        PQ   1.00    0      0      8.7    6.7    4.3
BUS Twenty5        PQ   1.00    0      0      0      0      0
BUS Twenty6        PQ   1.06    0      0      3.5    2.3    0
BUS Twenty7        PQ   1.00    0      0      0      0      0
BUS Twenty8        PQ   1.00    0      0      0      0      0
BUS Twenty9        PQ   1.00    0      0      2.4    0.9    0
BUS Thirty         PQ   1.00    0      0      10.6   1.9    0
```

```
% Transmission line data has the form:
%LINE  from       to          Rse     Xse     Gsh     Bsh     Rating(MVA)
LINE   One        Two         0.0192  0.0575  0.000   0.02640     200
LINE   One        Three       0.0452  0.1852  0.000   0.02040     100
LINE   Two        Four        0.0570  0.1737  0.000   0.01840     100
LINE   Three      Four        0.0132  0.0379  0.000   0.00420     100
LINE   Two        Five        0.0472  0.1983  0.000   0.02090     100
LINE   Two        Six         0.0581  0.1763  0.000   0.01870     100
LINE   Four       Six         0.0119  0.0414  0.000   0.00450     100
LINE   Five       Seven       0.0460  0.1160  0.000   0.01020     100
LINE   Six        Seven       0.0267  0.0820  0.000   0.00850     100
LINE   Six        Eight       0.0120  0.0420  0.000   0.00450     100
LINE   Six        Nine        0.0000  0.2080  0.000   0.000       100
LINE   Six        Ten         0.0000  0.5560  0.000   0.000       100
LINE   Nine       Eleven      0.0000  0.2080  0.000   0.000       100
LINE   Nine       Ten         0.0000  0.1100  0.000   0.000       100
LINE   Four       Twelve      0.0000  0.2560  0.000   0.000       100
LINE   Twelve     Thirteen    0.0000  0.1400  0.000   0.000       100
LINE   Twelve     Fourteen    0.1231  0.2559  0.000   0.000       100
LINE   Twelve     Fifteen     0.0662  0.1304  0.000   0.000       100
LINE   Twelve     Sixteen     0.0945  0.1987  0.000   0.000       100
LINE   Fourteen   Fifteen     0.2210  0.1997  0.000   0.000       100
LINE   Sixteen    Seventeen   0.0824  0.1923  0.000   0.000       100
LINE   Fifteen    Eighteen    0.1073  0.2185  0.000   0.000       100
LINE   Eighteen   Nineteen    0.0639  0.1292  0.000   0.000       100
LINE   Nineteen   Twenty      0.0340  0.0680  0.000   0.000       100
LINE   Ten        Twenty      0.0936  0.2090  0.000   0.000       100
LINE   Ten        Seventeen   0.0324  0.0845  0.000   0.000       100
LINE   Ten        Twenty1     0.0348  0.0749  0.000   0.000       100
LINE   Ten        Twenty2     0.0727  0.1499  0.000   0.000       100
LINE   Twenty1    Twenty2     0.0116  0.0236  0.000   0.000       100
LINE   Fifteen    Twenty3     0.1000  0.2020  0.000   0.000       100
LINE   Twenty2    Twenty4     0.1150  0.1790  0.000   0.000       100
LINE   Twenty3    Twenty4     0.1320  0.2700  0.000   0.000       100
LINE   Twenty4    Twenty5     0.1885  0.3292  0.000   0.000       100
LINE   Twenty5    Twenty6     0.2544  0.3800  0.000   0.000       100
LINE   Twenty5    Twenty7     0.1093  0.2087  0.000   0.000       100
LINE   Twenty8    Twenty7     0.0000  0.3960  0.000   0.000       100
LINE   Twenty7    Twenty9     0.2198  0.4153  0.000   0.000       100
LINE   Twenty7    Thirty      0.3202  0.6027  0.000   0.000       100
LINE   Twenty9    Thirty      0.2399  0.4533  0.000   0.000       100
LINE   Eight      Twenty8     0.0636  0.2000  0.000   0.0214      100
LINE   Six        Twenty8     0.0169  0.0599  0.000   0.065       100
```

APPENDIX CASE 2

1) Power System Data

Power System Data

```
% File describing the base case for the power system of
% System data has the form:
%SYSTEM        name       baseMVA    Voltage Tolerance
SYSTEM         2           100               0.05
%
% Bus data has the form:
%BUS name        type  volts   Pgen    Qgen    Pload  Qload  Qcap
BUS One          SL    1.00       0       0       0      0      0
BUS Two          PQ    1.00       0       0      60     35      0
BUS Three        PQ    1.00       0       0      70     40      0
BUS Four         PQ    1.00       0       0     100     63      0
BUS Five         PV    1.00     190       0      40     30      0
%
% Transmission line data has the form:
%LINE   from      to       Rse      Xse      Gsh      Bsh       Rating(MVA)
LINE    One       Two      0.0210   0.1250   0.000    0.000         50
LINE    One       Four     0.0235   0.0940   0.000    0.000        100
LINE    Two       Three    0.0250   0.1500   0.000    0.000         50
LINE    Two       Five     0.0180   0.0730   0.000    0.000        100
LINE    Three     Five     0.0220   0.1100   0.000    0.000         70
LINE    Four      Five     0.0190   0.0800   0.000    0.000        100
```

2) Power System Data

```
% File describing the base case for the power system of
% System data has the form:
%SYSTEM        name       baseMVA    Voltage Tolerance
SYSTEM          2           100             0.05
%
% Bus data has the form:
%BUS name       type  volts   Pgen    Qgen    Pload  Qload  Qcap
BUS One          SL   1.00    0        0       0      0      0
BUS Two          PQ   1.00    0        0      60     35      0
BUS Three        PQ   1.00    0        0      70     40      0
BUS Four         PQ   1.00    0        0     100     63      0
BUS Five         PV   1.00  190        0      40     30      0
```

```
% Transmission line data has the form:
%LINE   from   to         Rse      Xse      Gsh      Bsh      Rating(MVA)
LINE    One      Two     0.0210   0.1250   0.000    0.000         50
LINE    One      Four    0.0235   0.0940   0.000    0.000        100
%LINE    Two     Three   0.0250   0.1500   0.000    0.000         50
%open circuit for maintenance
LINE    Two      Five    0.0180   0.0730   0.000    0.000        100
LINE    Three    Five    0.0220   0.1100   0.000    0.000         70
LINE    Four     Five    0.0190   0.0800   0.000    0.000        100
```

3) Power System Data

```
% File describing the base case for the power system of
% System data has the form:
%SYSTEM       name       baseMVA    Voltage Tolerance
SYSTEM          2          100            0.05
%
% Bus data has the form:
%BUS name      type  volts   Pgen    Qgen    Pload  Qload  Qcap
BUS One        SL    1.00      0       0       0      0      0
BUS Two        PQ    1.00      0       0      60     35      0
BUS Three      PQ    1.00      0       0      70     40     40
%adding 40MVAR at Bus 3
BUS Four       PQ    1.00      0       0     100     63      0
BUS Five       PV    1.00    190       0      40     30      0
%
% Transmission line data has the form:
%LINE   from      to         Rse      Xse      Gsh      Bsh      Rating(MVA)
LINE    One       Two      0.0210   0.1250   0.000    0.000         50
LINE    One       Four     0.0235   0.0940   0.000    0.000        100
%LINE    Two      Three    0.0250   0.1500   0.000    0.000         50
 %open circuit for maintenance
LINE    Two       Five     0.0180   0.0730   0.000    0.000        100
LINE    Three     Five     0.0220   0.1100   0.000    0.000         70
LINE    Four      Five     0.0190   0.0800   0.000    0.000        100
```

4) Power System Data

```
% File describing the base case for the power system of
% System data has the form:
%SYSTEM       name       baseMVA    Voltage Tolerance
SYSTEM          2          100             0.05
%
% Bus data has the form:
%BUS name    type  volts   Pgen    Qgen    Pload  Qload  Qcap
BUS One      SL    1.00      0      0        0      0      0
BUS Two      PQ    1.00      0      0       60     35      0
BUS Three    PQ    1.00      0      0       70     40      0
BUS Four     PQ    1.00      0      0      120     70      0
BUS Five     PV    1.00    190      0       40     30      0
```

```
% Transmission line data has the form:
%LINE   from      to        Rse     Xse     Gsh     Bsh       Rating(MVA)
LINE    One       Two      0.0210  0.1250  0.000   0.000          50
LINE    One       Four     0.0235  0.0940  0.000   0.000         100
LINE    Two       Three    0.0250  0.1500  0.000   0.000          50
LINE    Two       Five     0.0180  0.0730  0.000   0.000         100
LINE    Three     Five     0.0220  0.1100  0.000   0.000          70
LINE    Four      Five     0.0190  0.0800  0.000   0.000         100
```