



# **NEAR EAST UNIVERSITY**

## **Faculty of Engineering**

### **Department of Electrical and Electronic Engineering**

#### **TRANSMISSION LINE MODELING AND FORMATION OF YBUS FOR A POWER SYSTEM**

##### **Graduation Project EE – 400**

**Student: Serhat Başer (20030789)**

**Supervisor: Assoc. Prof. Murat Fahrioğlu**

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT</b>	i
<b>ABSTRACT</b>	ii
<b>TABLE OF CONTENTS</b>	iii
<b>INTRODUCTION</b>	v
<b>1. CHAPTER 1 : Transmission Line Modeling</b>	1
1.1 Overview Of Transmission Line Modeling	2
1.2 Distributed Transmission Line Parameter	2
1.3 Lumped Transmission Line Parameters	4
1.4 Lumped Parameter Transmission Line Models	7
<b>2. CHAPTER 2: Admittance Matrix</b>	12
2.1 Introduction	12
2.2 Bus Admittance Matrix	15
<b>3. CHAPTER 3: Ybus Formation (USING MATLAB)</b>	17
3.1 Matlab Codes Of Ybus	18
3.1.1 calc_ybus.m	18
3.1.2 bldybus.m	18
3.1.3 pfolwjac.m	19
3.1.4 pfsolve.m	19
3.1.5 pfmiss.m	20
3.1.6 readcf.m	21
3.1.7 scanint.m	24
3.1.8 scanreal.m	24
3.2 IEEE 30 BUS SYSTEM ( GRAPH · DATA · CODE · OUTPUT )	25
3.2.1 IEEE 30 Bus System Data	26
3.2.2 Matlab Codes For Ybus	27
3.2.3 VOLTAGE - BUS GRAPH for IEEE 30 BUS SYSTEM	28
3.2.4 Ybus For IEEE 30 Bus System	28

3.3 HOW TO RUN (Screenshut)	30
<b>4. CONCLUSION</b>	34
<b>5. REFERENCES</b>	35
<b>6. APPENDIX A</b>	36
<b>7. APPENDIX B</b>	43

## INTRODUCTION

Chapter one explains transmission lines and types. Transmission line modeling and related theory was shown.

Chapter two is about admittance matrix and its formulas.

In the last chapter some toolboxes were used to calculate the Ybus in Matlab. These toolboxes were explained with detail.

A case study of a 30 bus IEEE system was analysed and the results are shown to verify the main program works correctly.

# CHAPTER 1

## TRANSMISSION LINE MODELING

# TRANSMISSION LINES

## INTRODUCTION:

Transmission lines carry electric energy from one point to another in an electric power system. They can carry alternating current or direct current or a system can be a combination of both. Also, electric current can be carried by either overhead or underground lines. The main characteristics that distinguish transmission lines from distribution lines are that they are operated at relatively high voltages, they transmit large quantities of power and they transmit the power over large distances.

### The types of transmission lines are:

**Overhead Transmission Lines :**They carry 3-phase current. The voltages vary according to the particular grid system they belong to. Transmission voltages vary from 69 kv up to 765 kv. The following are examples of different overhead transmission line structures in use today. The DC voltage transmission tower has lines in pairs rather than in threes (for 3-phase current) as in AC voltage lines. One line is the positive current line and the other is the negative current line.

**Subtransmission Lines :**Carry voltages reduced from the major transmission line system. Typically, 34.5 kv to 69 kv, this power is sent to regional distribution substations. Sometimes the subtransmission voltage is tapped along the way for use in industrial or large commercial operations. Some utilities categorize these as transmission lines.

**Underground Transmission Lines:** Are more common in populated areas. They may be buried with no protection, or placed in conduit, trenches, or tunnels.

## 1. POWER TRANSMISSION LINE MODELING

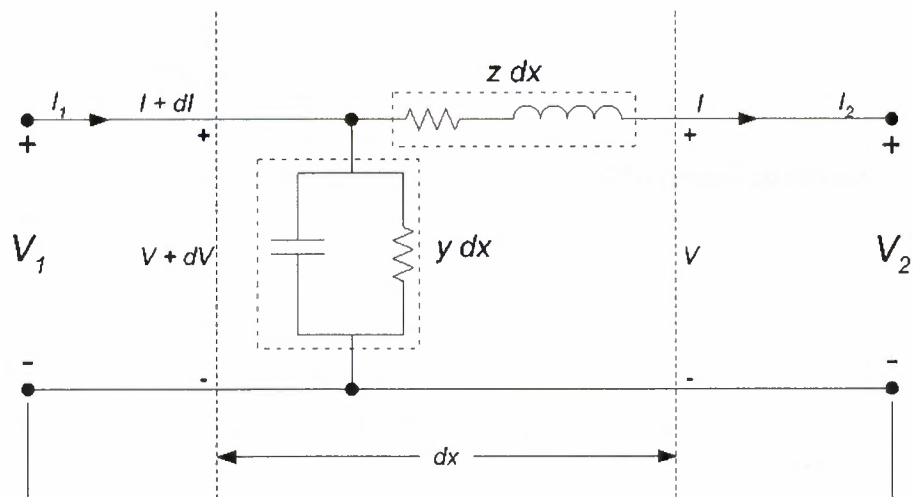
### 1.1 Overview Of Transmission Line Modeling

The transmission line transmits electrical power from one end of the line, sending end, to another, receiving end. A common method of analyzing this behavior is through parameterization and modeling of the transmission lines with passive components. The passive components used in this modeling are resistors, capacitors and inductors. The quantity of these parameters depends mainly on the conductors used in the lines and the physical or geometrical configuration of the lines. The conductors themselves will have certain characteristics such as resistance and reactance both in series from sending to receiving ends of the line and shunt from the line to electrical ground associated with them. In addition, there is inherently mutual inductance, or coupling, of the lines with respect to each other as they are bundled together or placed in close proximity to one another in a multi-phase system. This can all be taken into account through proper analysis and parameterization of the transmission lines.



## 1.2 Distributed Transmission Line Parameters

The distributed transmission line model gains its name from 'distributing' the parameters of the line equally through the line. The parameters are quantified per unit length and are additive in nature as the length of the line increases. This is effective for wave propagation analysis along the line as well as at the terminals of the line. Specifically for a power transmission line the voltage and current behavior along the line is of interest. Power transmitted through the line as well as losses can be observed and determined from these quantities. A diagram for a segment of a distributed transmission line is shown in Figure 1. The line model segment consists of a series impedance and a shunt admittance. The overall distributed transmission line model consists of a summation of these line segments.



**Figure 1 :** Distributed Parameter Transmission Line Model

The series impedance of the line is represented by a resistance in series with an inductance. This is quantified per unit length as follows:

$$z = r + j\omega l$$

The series resistance  $r$  and the series inductance  $l$  are quantified respectively as ohms per unit length and henrys per unit length. The term represents the operational angular frequency of the AC power system in radians per second. The resulting impedance of the inductor is therefore dependant on the operational frequency of the system. In this thesis lower case letters will be used for distributed parameters and upper case letters for lumped parameters. The shunt elements of the line model consist of a capacitance and a resistance in parallel with each other. The admittance of this shunt element is quantified by:

$$y = g + j\omega c$$

The capacitor admittance is also dependant on the system operational

frequency.

As shown in Figure 1, the length of the whole line is denoted by  $l$  and the length of the differential segment by  $dx$ . The series impedance across a segment is  $zdx$  and the shunt segment  $ydx$ .  $V_1$  and  $I_1$  denote the voltage and current at the sending end,  $x = 0$ , and  $V_2$  and  $I_2$  at the receiving end,  $x = l$ . A voltage drop  $dV$  appears across the series

element of the line and a current  $dI$  flows through the shunt element. Two first-order differential equations quantify the voltage drop and current loss of the line.

$$dv/dx = -zI \quad dI/dx = -yV$$

This same behavior can also be represented by a single second-order linear equation in terms of either current or voltage:

$$\frac{d^2V}{dx^2} = yzV = \gamma^2V$$

$$\frac{d^2I}{dx^2} = yzI = \gamma^2I$$

The term  $\gamma$  is called the propagation constant. Solving the differential equations yields the following solutions :

$$V = V_2 \cosh(\gamma x) + Z_c I_2 \sinh(\gamma x)$$

$$I = I_2 \cosh(\gamma x) + \frac{V_2}{Z_c} \sinh(\gamma x)$$

These equations yield the voltage and current at any point along the transmission line designated by  $x$ . The term  $Z_c$  is referred to as the characteristic impedance of the line and is computed by:

$$Z_c = \sqrt{z/y}$$

The terminal voltages and currents are usually of more interest and can be computed by setting  $x = l$  :

$$V_1 = V_2 \cosh(\gamma l) + Z_c I_2 \sinh(\gamma l)$$

$$I_1 = I_2 \cosh(\gamma l) + \frac{V_2}{Z_c} \sinh(\gamma l)$$

The lumped parameter model is a simplification of the distributed model and was developed to model only the terminal relationships at the sending end and receiving end of the transmission lines. It was derived but contains no information of voltage and current as they propagate down the line. Voltage and current information at any point



along the line can be useful but where it is not required the simplified lumped models are adequate.

### 1.3 Lumped Transmission Line Parameters

Traditionally in power flow analysis the voltages and currents at the line terminals are the values of more interest as compared to points along the transmission line. Historically the wave propagation on the lines is neglected and a simplified lumped parameter model is implemented for power flow. To develop the lumped model equation can be put in the following form [1]:

$$V_1 = AV_2 + BI_2$$

$$I_1 = CV_2 + DI_2$$

where:

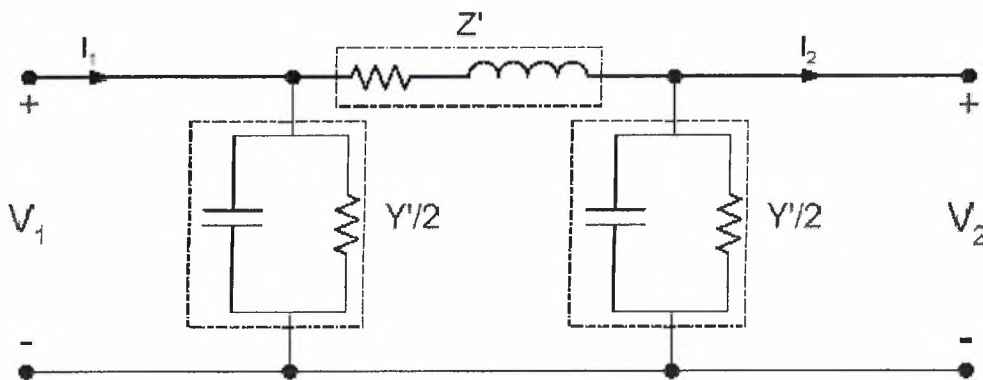
$$A = \cosh(\gamma l)$$

$$B = Z_c \sinh(\gamma l)$$

$$C = \frac{1}{Z_c} \sinh(\gamma l)$$

$$D = \cosh(\gamma l)$$

The lumped transmission line model holds the appropriate terminal behavior in equation with lumped circuit elements. A u-equivalent circuit for this purpose is shown in Figure 2.



**Figure 2:**  
Lumped Parameter  $\Pi$ -Equivalent Transmission Line

Passive elements model the series and shunt elements in this model. The effects of the distribution of impedance across the line are lumped together in quantities of ohms.

The formulation here allows the quantification of parameters in a lumped  $\pi$ -equivalent circuit with respect to the length, characteristic impedance and the propagation constant of the line. Resistors, capacitors and inductors for the lumped model are sized by. From there circuit analysis on the resultant RLC circuit will maintain the appropriate terminal behavior. In a similar manner a T equivalent circuit could be constructed with two series impedance elements and one shunt impedance element. This is simply a transformation of the  $\pi$  model which holds the same terminal relationship. For shorter length transmission lines this derivation may be simplified.

$$\frac{Y'}{2} = \frac{1}{Z_e} \tanh\left(\frac{\gamma l}{2}\right)$$

$$Z' = Z_e \sinh(\gamma l)$$

$$D = 1 + \frac{2}{Z' Y'}$$

$$C = Y' \left(1 + \frac{4}{Z' Y'}\right)$$

$$B = Z'$$

$$A = 1 + \frac{2}{Z' Y'}$$

Comparing the quantification of the A, B, C, and D parameters for the lumped equivalent circuit.

$$\begin{aligned} I_1 &= \frac{Y'}{2} V_1 + \frac{Y'}{2} V_2 + I_2 \\ V_1 &= V_2 + Z' \left( I_2 + \frac{Y'}{2} I_2 \right) \\ &= \left( 1 + \frac{Z' Y'}{2} \right) V_2 + Z' I_2 \\ &= Y' \left( 1 + \frac{4}{Z' Y'} \right) V_2 + \left( 1 + \frac{2}{Z' Y'} \right) I_2 \end{aligned}$$

They are related to the length of the line but are not in terms of unit length. By appropriately picking the values of  $Z'$  and  $Y'/2$  the same terminal behavior of the distributed line is held. Utilization of Kirchhoff's voltage (KVL) and current (KCL) laws to the lumped parameter circuit in Figure 2 yield the following results [1]:

For lines that are not particularly long  $\gamma l \ll 1$ . If this is the case small angle approximations can be made to simplify the computation without substantial loss in computational accuracy. The next section deals with this simplification along with the omission of certain passive elements to simplify the lumped model even further.

## 1.4 Lumped Parameter Transmission Line Models

The prior sections introduced both distributed and lumped parameter transmission line modeling. Specifically the lumped parameter model was a  $\pi$ -equivalent circuit which incorporates shunt resistive and capacitive elements along with series resistive and inductive elements. Certain simplifying assumptions can be made to this lumped equivalent circuit based on the length of the transmission line. Four different line models are presented here based on the transmission line length. As the lines become shorter certain parameters have a minimal effect on the terminal voltages and currents of the line and can then be neglected.

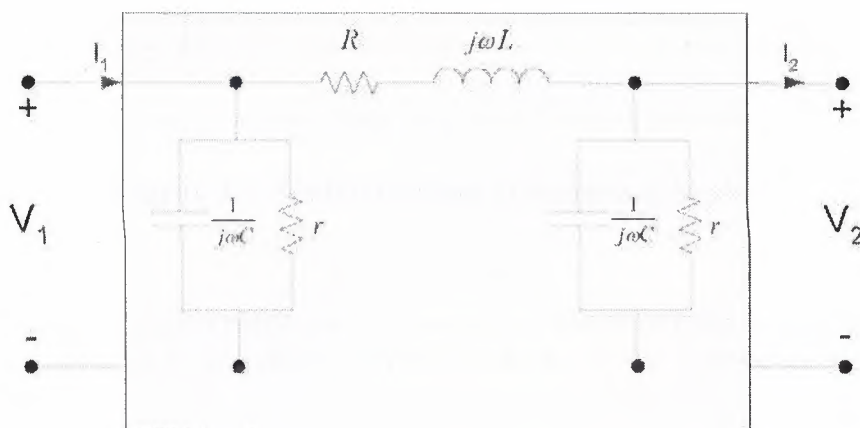
For a long transmission line ( $>150$  miles) no approximations should be made.

The circuit elements for a  $\pi$ -equivalent circuit should be computed. For this  $\pi$ -equivalent circuit the characteristic impedance and propagation constant are computed as follows:

$$\begin{aligned}\gamma &= \sqrt{yz} = \sqrt{(g + j\omega c)(r + j\omega l)} \\ &= \sqrt{gr - \omega^2 cl + j(\omega cr + \omega gl)}\end{aligned}$$

$$Z_c = \sqrt{\frac{z}{y}} = \sqrt{\frac{r + j\omega l}{g + j\omega c}}$$

With  $\gamma$  and  $Z_c$  the complex values of  $Z'/2$  and  $Y'/2$  are computed. The RLC circuit elements are then sized for the equivalent circuit in Figure 3. This representation will be referred to as the long transmission line model.



**Figure 3:** Long Transmission Line Model

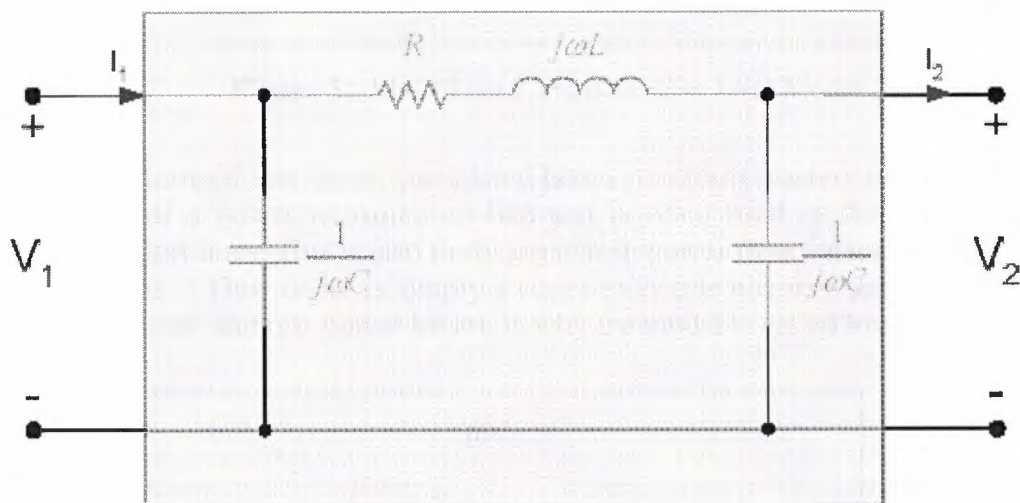
Where:

$$Z' = R + j\omega L$$

$$\frac{Y'}{2} = \frac{1}{r} + j\omega C$$

Note that “r” is a lumped parameter. It is in lower case here to differentiate it from the series resistive element. This shunt resistive element can be neglected if a transmission line is less than 150 miles in length. This is defined as a medium length transmission line.

The resistive shunt current flow on a transmission line is usually very small and is proportional to the terminal voltage magnitude. This element’s contribution is only significant in longer high voltage transmission lines. For medium length lines this component is neglected with little discrepancy in calculated results. If the shunt resistance of the line is neglected the lumped equivalent circuit will take the form in Figure 4.



**Figure 4:** Medium Length Transmission Model

Further approximations are also made for this model. No longer are the propagation constant and characteristic impedances used to calculate the RLC elements in the lumped circuit. The following approximations are made:

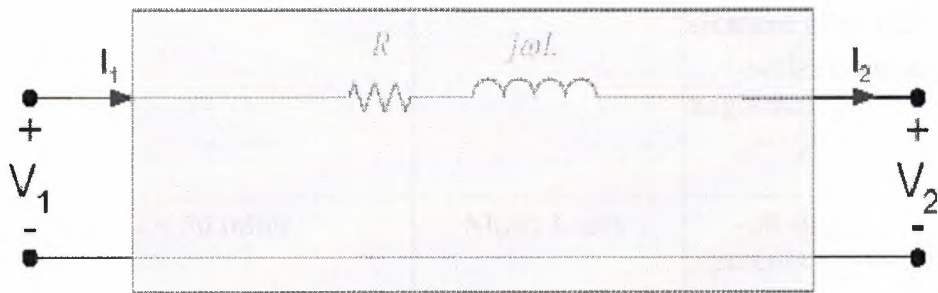
$$\frac{Y'}{2} \approx \frac{Y}{2} = y l = -j\omega C$$

$$Z' \approx Z = z l = R + j\omega L$$



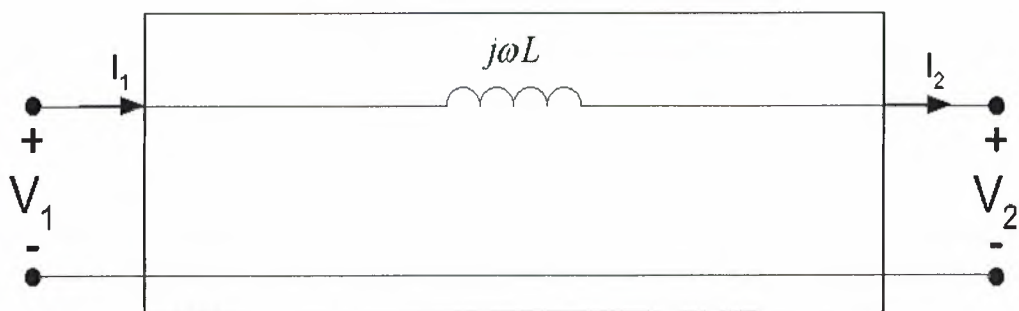
This is simpler than the long line model computation. The elements in this model can be quantified by the line length and parameters in per unit length. No hyperbolic functions are necessary. Next further approximations are made for short transmission line. A short transmission line is defined as a line less than 50 miles in length.

For short transmission lines all the shunt elements are neglected. The argument for this is that the shunt charging due to the shunt capacitance is negligible for short transmission lines and eliminating them from the modeling results in little inaccuracy. Two short transmission line models are represented here. One is referred to as a short lossy transmission line and the other short lossless transmission line. The lossy model incorporates both the series resistor and inductor components shown in Figure 5. The term lossy is formulated from the presence of  $I^2R$  real power losses in the model.



**Figure 5:** Short Lossy Transmission Line Model

The fourth and final lumped line model considered here is the short lossless line model. This is the simplest representation of a power transmission line and is often used in distribution system line modeling and for very short lines. This is also fairly popular in power flow solvers in order to simplify and speed up calculations. This model is simply a series inductive element shown in Figure 6. The absence of the series resistor, and real power losses, is why the model is called lossless.



**Figure 6:** Short Lossless Transmission Line Model

The inductor in this model is sized by the following equation:  $Z = j\omega L$

Four lumped line models have been presented, specifically long, medium, short lossy and short lossless models. Care must be taken when to incorporate which model in calculations to prevent large errors in the results. Guidelines have been given in [1] with respect to the transmission line length. This is summarized in Table 1 and each line type is labeled by a letter. Intuition indicates that the short lossless line model should only be used in lines much less than 50 miles in length.

**Table 1:** Details on Lumped Line Models

Line Model	Transmission Line Length	Appropriate Lumped Model	Assumptions Made
<b>A</b>	$l \ll 50$ miles	Short Lossless	-all shunt element neglected -series resistance neglected $Z \approx Z$
<b>B</b>	$l < 50$ miles	Short Lossy	-all shunt elements neglected $Z \approx Z$
<b>C</b>	$50 \leq l < 150$ miles	Medium Line	-shunt resistor neglected $\frac{Y'}{2} \approx \frac{Y}{2}$ $Z' \approx Z$
<b>D</b>	$l \geq 150$ miles	Long Line	None

The four lumped line models presented here are the basis for the analog transmission line modeling for power flow. Analog circuit equivalents to the four models derived are developed and tested for the analog power flow emulation.

## CHAPTER 2

# ADMITTANCE MATRIX

## Admittance Matrix

### 2.1 Introduction

Current injections at a bus are analogous to power injections. The student may have already been introduced to them in the form of current sources at a node. Current injections may be either positive (into the bus) or negative (out of the bus). Unlike current flowing through a branch (and thus is a branch quantity), a current injection is a nodal quantity. The admittance matrix, a fundamental network analysis tool that we shall use heavily, relates current injections at a bus to the bus voltages. Thus, the admittance matrix relates nodal quantities. We motivate these ideas by introducing a simple example.

Figure 1 shows a network represented in a hybrid fashion using one-line diagram representation for the nodes (buses 1-4) and circuit representation for the branches connecting the nodes and the branches to ground. The branches connecting the nodes represent lines. The branches to ground represent any shunt elements at the buses, including the charging capacitance at either end of the line. All branches are denoted with their admittance values  $y_{ij}$  for a branch connecting bus  $i$  to bus  $j$  and  $y_i$  for a shunt element at bus  $i$ . The current injections at each bus  $i$  are denoted by  $I_i$ .

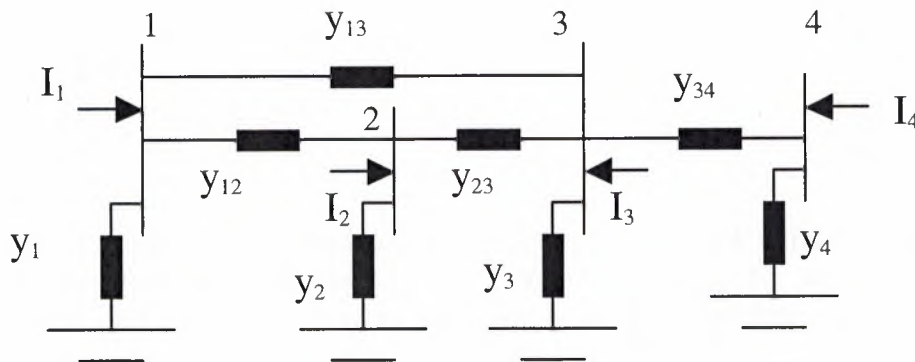


Figure: 7

Kirchoff's Current Law (KCL) requires that each of the current injections be equal to the sum of the currents flowing out of the bus and into the lines connecting the bus to other buses, or to the ground. Therefore, recalling Ohm's Law,  $I=V/z=Vy$ , the current injected into bus 1 may be written as:

$$I_1=(V_1-V_2)y_{12} + (V_1-V_3)y_{13} + V_1y_1 \quad (1)$$

To be complete, we may also consider that bus 1 is "connected" to bus 4 through an infinite impedance, which implies that the corresponding admittance  $y_{14}$  is zero. The advantage to doing this is that it allows us to consider that bus 1 could be connected to any bus in the network. Then, we have:

$$I_1 = (V_1 - V_2)y_{12} + (V_1 - V_3)y_{13} + (V_1 - V_4)y_{14} + V_1 y_1$$

(2)

Note that the current contribution of the term containing  $y_{14}$  is zero since  $y_{14}$  is zero. Rearranging eq. (2), we have:

$$I_1 = V_1(y_1 + y_{12} + y_{13} + y_{14}) + V_2(-y_{12}) + V_3(-y_{13}) + V_4(-y_{14})$$

(3)

Similarly, we may develop the current injections at buses 2, 3, and 4 as:

$$I_2 = V_1(-y_{21}) + V_2(y_2 + y_{21} + y_{23} + y_{24}) + V_3(-y_{23}) + V_4(-y_{24})$$

(4)

$$I_3 = V_1(-y_{31}) + V_2(-y_{32}) + V_3(y_3 + y_{31} + y_{32} + y_{34}) + V_4(-y_{34})$$

(5)

$$I_4 = V_1(-y_{41}) + V_2(-y_{42}) + V_3(-y_{34}) + V_4(y_4 + y_{41} + y_{42} + y_{43})$$

(6)

where we recognize that the admittance of the circuit from bus  $k$  to bus  $i$  is the same as the admittance from bus  $i$  to bus  $k$ , i.e.,  $y_{ki} = y_{ik}$ . From eqs. (3-6), we see that the current injections are linear functions of the nodal voltages. Therefore, we may write these equations in a more compact form using matrices according to:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} y_1 + y_{12} + y_{13} + y_{14} & -y_{12} & -y_{13} & -y_{14} \\ -y_{21} & y_2 + y_{21} + y_{23} + y_{24} & -y_{23} & -y_{24} \\ -y_{31} & -y_{32} & y_3 + y_{31} + y_{32} + y_{34} & -y_{34} \\ -y_{41} & -y_{42} & -y_{43} & y_4 + y_{41} + y_{42} + y_{43} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$$

(7)

The matrix containing the network admittances in eq. (7) is the admittance matrix, also known as the Y-bus, and denoted as:



$$\underline{Y} = \begin{bmatrix} y_1 + y_{12} + y_{13} + y_{14} & -y_{12} & -y_{13} & -y_{14} \\ -y_{21} & y_2 + y_{21} + y_{23} + y_{24} & -y_{23} & -y_{24} \\ -y_{31} & -y_{32} & y_3 + y_{31} + y_{32} + y_{34} & -y_{34} \\ -y_{41} & -y_{42} & -y_{43} & y_4 + y_{41} + y_{42} + y_{43} \end{bmatrix}$$

(8)

Denoting the element in row  $i$ , column  $j$ , as  $Y_{ij}$ , we rewrite eq. (8) as:

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

(9)

where the terms  $Y_{ij}$  are not admittances but rather elements of the admittance matrix. Therefore, eq. (7) becomes:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$$

(10)

By defining the vectors  $V$  and  $I$ , we may write eq. (10) in compact form according to:

$$\underline{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}, \quad \underline{I} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix}$$

(11)

$$I = Y V$$

We make several observations about the admittance matrix given in eqs. (10) and (11). These observations hold true for any linear network of any size.

The matrix is symmetric, i.e.,  $Y_{ij}=Y_{ji}$ .

A diagonal element  $Y_{ii}$  is obtained as the sum of admittances for all branches

$$Y_{ii} = y_i + \sum_{k=1, k \neq i}^N y_{ik}$$

connected to bus i, including the shunt branch, i.e.,

where we emphasize once again that  $y_{ik}$  is non-zero only when there exists a physical connection between buses i and k.

The off-diagonal elements are the negative of the admittances connecting buses i and j, i.e.,  $Y_{ij}=-y_{ji}$ .

## 2.2 Bus Admittance Matrix

The first step is to number all the nodes of the system from 0 to n. Node 0 is the datum or reference node (or ground node).

replace all generators by equivalent current sources in parallel with an admittance.

Replace all lines, transformers, loads to equivalent admittances whenever possible. Knowing a load in MVA but not knowing the operating voltage, makes it impossible to change the load to an admittance. The rule for this is simple:  $y=1/z$  where

y and z are generally complex numbers.

4. The bus admittance matrix Y is then formed by inspection as follows (this is similar to what we learned in circuit theory):

$y_{ii}$  = sum of admittances connected to node i and

$y_{ij} = -\text{sum of admittances connected from node i to node j}$

The current vector is next found from the sources connected to nodes 0 to n. If no source is connected, the injected current would be 0.

The equations which result are called the node-voltage equations and are given the "bus" subscript in power studies thus:

$$\begin{matrix} I \\ \text{bus} \end{matrix} = \begin{matrix} Y \\ \text{bus} \end{matrix} \begin{matrix} V \\ \text{bus} \end{matrix}$$

The inverse of these equations results in the set  $V_{\text{bus}} = Y_{\text{bus}}^{-1} I_{\text{bus}}$ . It is emphasized that the matrix  $Y_{\text{bus}}$  is not the same as the matrix Z which results from solving a circuit using mesh equations. To clearly show this difference we define  $Z_{\text{bus}} = Y_{\text{bus}}^{-1}$ . It is noted that the matrix  $Y_{\text{bus}}$  is the same as the Y matrix obtained from circuit theory. Thus  $Y_{\text{bus}} = Y$  however  $Z_{\text{bus}} \neq Z$ .

These observations enable us to formulate the admittance matrix very quickly from the network based on visual inspection. The following example will clarify. The most common way to represent such a system is to use the node-voltage method. Given the voltages of generators at all generator nodes, and knowing all impedances of machines and loads, one can solve for all the currents in the typical node voltage analysis methods using Kirchhoff's current law.

First the generators are replaced by equivalent current sources and the node

equations are written in the form

$I = YV$  where  $I$  is the injected current vector,  $Y$  is the admittance matrix and  $V$  is the node voltage vector. These equations are easy to write by inspection of the circuit.

The problem is not so simple in real power circuits and systems. Usually in a power system the complex power may be known at load nodes, and sometimes on generator nodes only the real power and voltage are known. Thus not enough variables are known to solve an equation of the form  $I = YV$ . In fact, since the power is a nonlinear function of the current and voltage, the solution of the resulting equations (while it may exist) is not easy! In fact there is no known analytical method to find the solution.

As a result iterative techniques are used to find the solution (voltages, currents, etc.). The nonlinear set of equations which are generated are called power flow equations. The solution of such equations results in a power flow study or load flow analysis. Such studies are the backbone of power system studies, for analysis, design, control, and economic operation of the power system. They are also essential for transient analysis of the system.

**Example:** Consider the network given in Fig. 8, where the numbers indicate admittances.

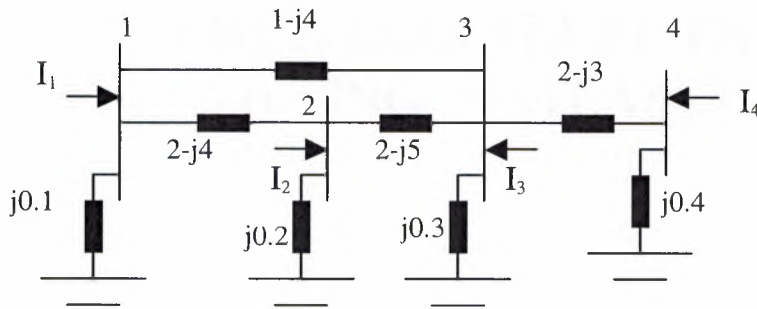


Figure: 8

The admittance matrix is given by inspection as:

$$\underline{Y} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} = \begin{bmatrix} 3 - j7.9 & -2 + j4 & -1 + j4 & 0 \\ -2 + j4 & 4 - j8.8 & -2 + j5 & 0 \\ -1 + j4 & -2 + j5 & 5 - j11.7 & -2 + j3 \\ 0 & 0 & -2 + j3 & 2 - j2.6 \end{bmatrix}$$

## **CHAPTER 3**

### **YBUS FORMATION (USING MATLAB)**



## 3.1 Matlab codes of Ybus

### 3.1.1 calc\_ybus.m

```
clear;
out=readcf;
out.Ybus=bldybus(out);      % Build the Y-bus matrix
out.snet = sparse(out.Machine.BusRef, 1,
out.Machine.MW+j*out.Machine.MVAR, out.Bus.n, 1)...
-sparse(out.Load.BusRef, 1 ,out.Load.MW+j*out.Load.MVAR,
out.Bus.n, 1);

out=pfsolve(out);
YBUS= zeros(size(out.Ybus));
YBUS(:, :) = out.Ybus(:, :);
comet (out.Bus.Voltages); figure(gcf);
```

### 3.1.2 bldybus.m

```
function Ybus=bldybus(S)
% bldybus.m Build the y-bus matrices
Ybus=sparse(S.Bus.n,S.Bus.n);
YL=[];

if isfield(S.Bus,'G'),
    YL=S.Bus.G;
end

if isfield(S.Bus,'B'),
    YL=YL+i*S.Bus.B;
end

if ~isempty(YL),
    Ybus=Ybus+diag(sparse(YL));
end

n=S.Bus.n;
Y11=(1./S.Branch.Z+(i*S.Branch.B)/2).*S.Branch.Status;
Ybus=Ybus+sparse(S.Branch.From,S.Branch.From,Y11,n,n);
Y12=(-1./(S.Branch.TAP.*S.Branch.Z)).*S.Branch.Status;
Ybus=Ybus+sparse(S.Branch.From,S.Branch.To,Y12,n,n);
Y21=(-
1./(conj(S.Branch.TAP).*S.Branch.Z)).*S.Branch.Status;
Ybus=Ybus+sparse(S.Branch.To,S.Branch.From,Y21,n,n);
Y22=(1./((abs(S.Branch.TAP).^2).*S.Branch.Z)+(i*S.Branch.B
)/2).*S.Branch.Status;
Ybus=Ybus+sparse(S.Branch.To,S.Branch.To,Y22,n,n);

if isfield(S.Branch,'YI'),
    Ybus=Ybus+sparse(S.Branch.From,S.Branch.From,S.Branch
.YI,n,n);
end
```



```

if isfield(S.Branch,'YJ'),
    Ybus=Ybus+sparse(S.Branch.To,S.Branch.To,S.Branch.YJ,
n,n);
end

if isfield(S,'Shunt'),
    Ybus=Ybus+sparse(S.Shunt.I,S.Shunt.I,i*S.Shunt.BINIT,
n,n);
end;

```

### 3.1.3 pflowjac.m

```

function [dSdd,dSdv] = pflowjac(Y,vb)
% Construct the power flow Jacobian matrix in complex
form
% Usage: [dSdd,dSdv] = pflowjac(Y,vb)
% Arguments: complex bus admittance matrix, Y;
%            vector of complex bus voltage phasors, vb.
% Returns   : Two full, complex n by n matrices of partial
%            derivatives;
%            dSdd = partial of complex power w.r.t. delta;
%            dSdv = partial of complex power w.r.t. voltage
magnitude.
% Comments   : We return ALL the partial derivatives of
power
%            absorbed at EVERY bus with respect to EVERY
%            voltage magnitude and phase angle.
% The operation of this routine is transparent if one
% recognizes that:
%  $S = \text{diag}(vb) * \text{conj}(ib) = \text{diag}(\text{conj}(ib)) * vb;$ 
% hence:
%  $dS/d(\text{delta}) =$ 
%  $\text{diag}(vb) * \text{conj}(d(ib)/d(\text{delta})) + \text{diag}(\text{conj}(ib)) * d(vb)/d(\text{delta})$ 
%  $dS/d(vmag) =$ 
%  $\text{diag}(vb) * \text{conj}(d(ib)/d(vmag)) + \text{diag}(\text{conj}(ib)) * d(vb)/d(vmag)$ 
%
ib=Y*vb;
dSdd=j*diag(conj(ib)).*vb)-
j*diag(vb)*conj(Y)*diag(conj(vb));
dSdv=diag(conj(ib)).*(vb./abs(vb))+diag(vb)*conj(Y)*diag(c
onj(vb)./abs(vb));

```

### 3.1.4 pfsolve.m

```

function S=pfsolve(S)
% This routine solves the power flow problem using a
% Newton- Raphson iteration with full Jacobian update at
every
% iteration.

```



```

% Compute initial mismatch
MAXITER=15;    n=S.Bus.n;
vbus=S.Bus.Voltages;

% pick off largest component of relevant mismatch
S.mismatch = Inf;

% Newton-Raphson Iteration
S.itcnt = 0;
while S.mismatch > .0001
    fullmiss = pfmisss(S);

    rmiss=[real(fullmiss(S.PVlist));real(fullmiss(S.PQlist));.
    ..
            imag(fullmiss(S.PQlist))];
    S.mismatch = max(abs(rmiss));
    [dsdd, dsdv] = pflowjac(S.Ybus,vbus);

% dsdd and dsdv are composed of all partial derivatives
% rjac is a selection from these
    rjac = [ ...
            real(dsdd(S.PVlist,S.PVlist))
real(dsdd(S.PVlist,S.PQlist)) ...
            real(dsdv(S.PVlist,S.PQlist)); ...
            real(dsdd(S.PQlist,S.PVlist))
real(dsdd(S.PQlist,S.PQlist)) ...
            real(dsdv(S.PQlist,S.PQlist)); ...
            imag(dsdd(S.PQlist,S.PVlist))
imag(dsdd(S.PQlist,S.PQlist)) ...
            imag(dsdv(S.PQlist,S.PQlist)) ];
    x=[angle(vbus(S.PVlist))*pi/180;
        angle(vbus(S.PQlist))*pi/180;
        abs(vbus(S.PQlist))];
    dx = -rjac\rmiss; % Here is the actual update
    del=angle(vbus);
    vmag=abs(vbus);
    n1=length(S.PVlist); n2=n1+length(S.PQlist);
    n3=length(dx);
    del(S.PVlist)=del(S.PVlist)+dx(1:n1);
    del(S.PQlist)=del(S.PQlist)+dx((n1+1):n2);
    vmag(S.PQlist)=vmag(S.PQlist)+dx((n2+1):n3);
    vbus=vmag.*exp(j*del);
    S.Bus.Voltages=vbus;
    S.itcnt=S.itcnt+1;
    if (S.itcnt > MAXITER) | ((S.itcnt>3)&(S.mismatch>100)),
        S.itcnt=Inf;
        break;
    end;
end
if (S.itcnt<=MAXITER),
    S.Bus.Voltages=vbus;
end

```

### 3.1.5 pfmiss.m

```
function nmiss = pfmiss(S)
% Form the vector of complex power mismatches.
% Usage: nmiss = pfmiss(Y,vb,s_net_inject)
% Arguments: complex bus admittance matrix, Y;
%            vector of complex bus voltage phasors, vb;
%            vector of complex bus power demand (net),
%            S.snet; (note that s_net_inject will be
%            negative at buses demanding power)
% Returns:   a full complex n-vector of power mismatches.
%            Power LEAVING the bus is positive
% Comments:  Note that we return ALL mismatches at EVERY
bus.
ib=S.Ybus*S.Bus.Voltages;
nmiss = S.Bus.Voltages.*conj(ib)-S.snet;
```

### 3.1.6 readcf.m

```
function S=readcf(fname)
% Read the common format file and create a data dictionary
% structure
if nargin<1, [fname,pname]=uigetfile('*.cf'); fname=[pname
fname]; end
S.Misc.BaseMVA=100;
fcf=fopen(fname,'r'); % Open common format file
s=fgetl(fcf);
% Find the start of bus data
while strcmp(s(1:min(3,length(s))), 'BUS')~=1,
s=fgetl(fcf);
end
s=fgetl(fcf);
while s(1)=='%', s=fgetl(fcf); end;
nL=0; ng=0; n=0;
S.PQlist=[]; S.SlackList=[]; S.BlackList=[]; S.PVlist=[];
S.Bus.Voltages=sparse(n,1);
S.Interch.I=[];
S.Interch.areaCount=zeros(100,1);
while strcmp(s(1:4), '-999')~=1,
n=n+1;
Pg=scanreal(s,59,67);
Qg=scanreal(s,68,75);
Pd=scanreal(s,41,49);
Qd=scanreal(s,50,58);
S.Bus.busType(n,1)=sscanf(s(26),'%d');
if (s(26)=='1') | (s(26)=='2') | (s(26)=='3')
if s(26)=='3', S.SlackList=[S.SlackList n];
else, S.PVlist=[S.PVlist n];
end;
else
S.PQlist=[S.PQlist n];
end;
```

```

bus=scanint(s,1,5);
S.Bus.Number(n,1)=bus;
S.Bus.Name{n}=s(6:17);
S.newBus(bus)=n;
S.Machine.newGen(bus,1)=0;
if (Pg~=0) | (Qg~=0) | (s(26)=='3') | (s(26)=='2'),
    ng=ng+1;
    S.Machine.newGen(bus,1)=ng;
    S.Machine.BusRef(ng,1)=bus;      % Generation bus number
    S.Machine.MW(ng,1)=Pg/S.Misc.BaseMVA; % Active Power
    Generation (actual)
    S.Machine.MVAR(ng,1)=Qg/S.Misc.BaseMVA; % Reactive
    Power Generation
    S.Machine.Status(ng,1)=1;
end
if (Pd~=0) | (Qd~=0),
    nL=nL+1;
    newLoad(bus,1)=nL;
    S.Load.BusRef(nL,1)=bus;          % Load bus number
    S.Load.MW(nL,1)=Pd/S.Misc.BaseMVA; % Net active power
    load
    S.Load.MVAR(nL,1)=Qd/S.Misc.BaseMVA; % Net reactive power
    load
    S.Load.Status(nL,1)=1;
end;
S.Bus.Generation(n,1)=(Pg+j*Qg)/S.Misc.BaseMVA;
S.Bus.Load(n,1)=(Pd+j*Qd)/S.Misc.BaseMVA;
S.Bus.Voltages(n,1)=scanreal(s,28,33,1)* ...
    exp(sqrt(-
1)*scanreal(s,34,40)*pi/180);
S.Bus.area(n,1)=sscanf(s(19:20),'%d'); % This is
numeric, +1
S.Bus.Zone(n,:)=s(21:23);
S.Bus.KV(n,1)=scanreal(s,77,83);
if S.Machine.newGen(bus),
    S.Machine.MinOperatingVolt(ng,1)=scanreal(s,85,90);
    S.Machine.MaxOperatingVolt(ng,1)=scanreal(s,85,90);

S.Machine.MaxQOutput(ng,1)=scanreal(s,91,98)/S.Misc.BaseMV
A;

S.Machine.MinQOutput(ng,1)=scanreal(s,99,106)/S.Misc.BaseM
VA;
    S.Machine.ControlledBusRef(ng,1)=scanreal(s,124,128);
end
S.Bus.G(n,1)=scanreal(s,107,114);
S.Bus.B(n,1)=scanreal(s,115,122);
s=fgetl(fcf);
while s(1)=='%', s=fgetl(fcf); end;
end;
minArea=min(S.Bus.area);
dArea=1-minArea;

```



```

S.Interch.areaCount=sparse(S.Bus.area+dArea,1,1);

while ~strcmp(s(1:3),'BRA'), s=fgetl(fcf); end
s=fgetl(fcf);
while s(1)=='%', s=fgetl(fcf); end;
nn=0;
while ~strcmp(s(1:4),'-999'),
    nn=nn+1;
    S.Branch.To(nn,1)=S.newBus(scanint(s,1,5));
    S.Branch.From(nn,1)=S.newBus(scanint(s,6,10));
    R=scanreal(s,20,29); % Resistance
    X=scanreal(s,30,40); % Reactance
    S.Branch.Z(nn,1)=R+j*X;
    S.Branch.B(nn,1)=scanreal(s,41,49); % Line Charging

S.Branch.RateValue(nn,1)=scanreal(s,51,55)/S.Misc.BaseMVA;
S.Branch.Type(nn,1)=scanint(s,19,19); % Type
if S.Branch.Type(nn,1), % If dealing with a transformer
    tap=scanreal(s,77,82);
    alpha=scanreal(s,84,90)*pi/180;
    S.Branch.TAP(nn,1)=tap.*exp(j*alpha);
else
    S.Branch.TAP(nn,1)=1;
end;
s=fgetl(fcf); % Next line
while s(1)=='%', s=fgetl(fcf); end; % Skip comments
end
S.Branch.YI=zeros(nn,1);
S.Branch.YJ=zeros(nn,1);
S.Branch.Status=ones(nn,1);
kzero=find(S.Branch.RateValue==0); % No zero ratings
permitted
S.Branch.RateValue(kzero)=Inf;

while ~strcmp(s(1:3),'INT'),
    s=fgetl(fcf);
end
s=fgetl(fcf);
while s(1)=='%', s=fgetl(fcf); end;
nInterch=0;
while ~strcmp(s(1:2),'-9'),
    nInterch=nInterch+1;
    S.Interch.I(nInterch,1)=scanint(s,1,2);
    S.Interch.ControlGenRef(nInterch,1)=scanint(s,3,7);

S.Interch.SchedNetMW(nInterch,1)=scanreal(s,21,29)/S.Misc.
BaseMVA;
s=fgetl(fcf);
while s(1)=='%', s=fgetl(fcf); end;
end
fclose(fcf);

```



```

S.Machine.BusRef=S.newBus(S.Machine.BusRef)'; %Use
internal b.n
S.Load.BusRef=S.newBus(S.Load.BusRef)'; %Use internal bus
num.s
ky=find(S.Machine.ControlledBusRef); % Use internal bus
numbers
kx=S.Machine.ControlledBusRef(ky);
S.Machine.ControlledBusRef(ky)=S.newBus(kx);
S.Bus.n=length(S.Bus.Voltages);
S.Branch.nn=length(S.Branch.From);
S.Bus.area=S.Bus.area+dArea;

```

### 3.1.7 scanint.m

```

function i=scanint(s,beg,last,default),
% Function to read an integer from a designated field
if nargin<4, default=0; end;
i=default;
[dummy,ns]=size(s);
if ns<beg, return; end;
if last<beg, return; end;
last=min(last,ns);
if isspace(s(beg:last)), return; end;
i=sscanf(s(beg:last),'%d');
return;

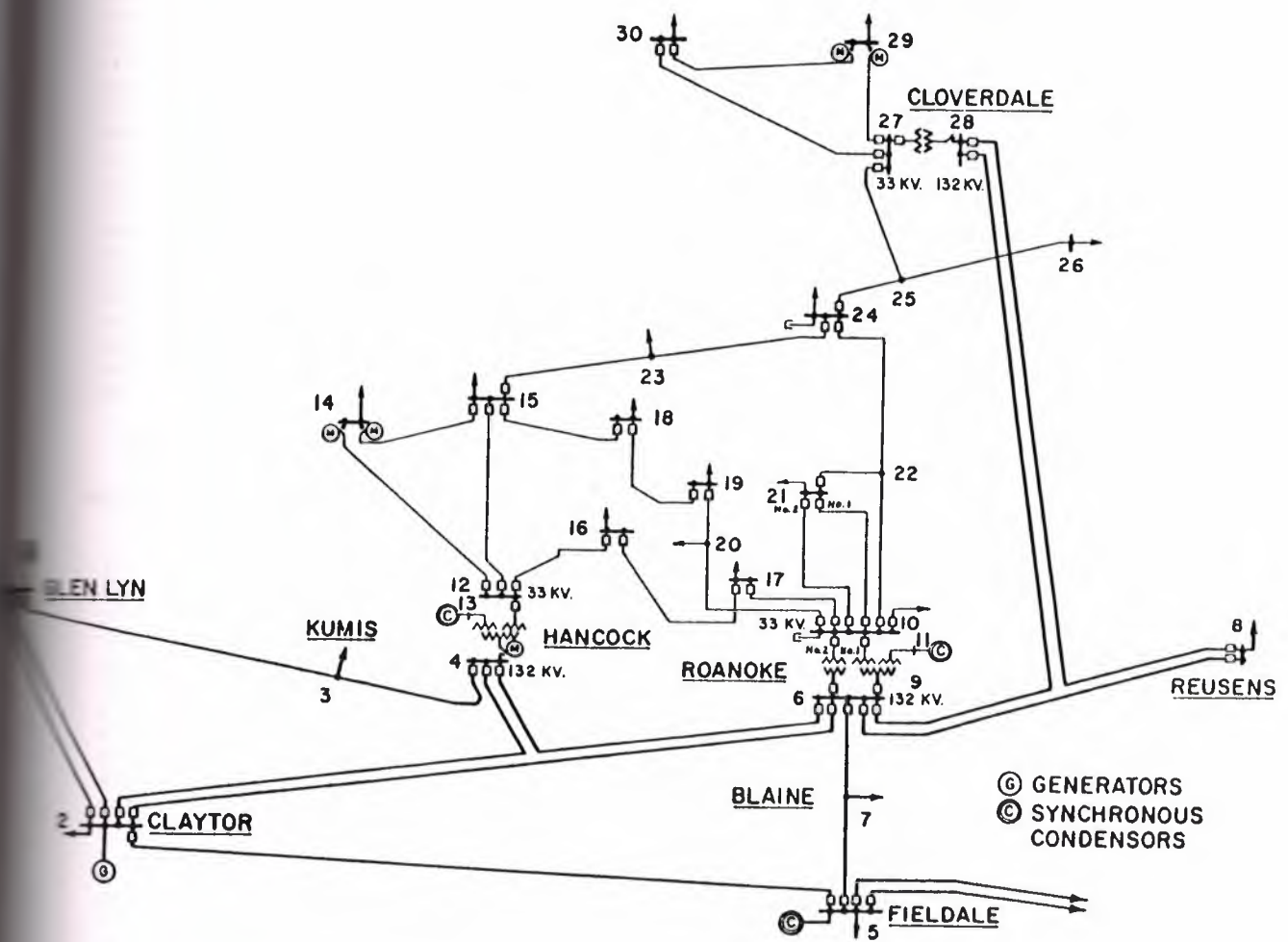
```

### 3.1.8 scanreal.m

```

function i=scanint(s,beg,last,default),
% Function to read an integer from a designated field
if nargin<4, default=0; end;
i=default;
[dummy,ns]=size(s);
if ns<beg, return; end;
if last<beg, return; end;
last=min(last,ns);
if isspace(s(beg:last)), return; end;
i=sscanf(s(beg:last),'%d');
return;

```



### 3.2.1 IEEE 30 BUS SYSTEM DATA

08/20/93 UW ARCHIVE 100.0 1961 W IEEE 30 Bus Test Case														
BUS DATA FOLLOWS					30 ITEMS									
1 Glen Lyn	132	1	1	3	1.060	0.0	0.0	0.0	260.2	-16.1	132.0	1.060	0.0	0.0
2 Claytor	132	1	1	2	1.043	-5.48	21.7	12.7	40.0	50.0	132.0	1.045	50.0	-40.0
3 Kumis	132	1	1	0	1.021	-7.96	2.4	1.2	0.0	0.0	132.0	0.0	0.0	0.0
4 Hancock	132	1	1	0	1.012	-9.62	7.6	1.6	0.0	0.0	132.0	0.0	0.0	0.0
5 Fieldale	132	1	1	2	1.010	-14.37	94.2	19.0	0.0	37.0	132.0	1.010	40.0	-40.0
6 Roanoke	132	1	1	0	1.010	-11.34	0.0	0.0	0.0	0.0	132.0	0.0	0.0	0.0
7 Blaine	132	1	1	0	1.002	-13.12	22.8	10.9	0.0	0.0	132.0	0.0	0.0	0.0
8 Reusens	132	1	1	2	1.010	-12.10	30.0	30.0	0.0	37.3	132.0	1.010	40.0	-10.0
9 Roanoke	1.0	1	1	0	1.051	-14.38	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
10 Roanoke	33	1	1	0	1.045	-15.97	5.8	2.0	0.0	0.0	33.0	0.0	0.0	0.19
11 Roanoke	11	1	1	2	1.082	-14.39	0.0	0.0	0.0	16.2	11.0	1.082	24.0	-6.0
12 Hancock	33	1	1	0	1.057	-15.24	11.2	7.5	0.0	0.0	33.0	0.0	0.0	0.0
13 Hancock	11	1	1	2	1.071	-15.24	0.0	0.0	0.0	10.6	11.0	1.071	24.0	-6.0
14 Bus 14	33	1	1	0	1.042	-16.13	6.2	1.6	0.0	0.0	33.0	0.0	0.0	0.0
15 Bus 15	33	1	1	0	1.038	-16.22	8.2	2.5	0.0	0.0	33.0	0.0	0.0	0.0
16 Bus 16	33	1	1	0	1.045	-15.83	3.5	1.8	0.0	0.0	33.0	0.0	0.0	0.0
17 Bus 17	33	1	1	0	1.040	-16.14	9.0	5.8	0.0	0.0	33.0	0.0	0.0	0.0
18 Bus 18	33	1	1	0	1.028	-16.82	3.2	0.9	0.0	0.0	33.0	0.0	0.0	0.0
19 Bus 19	33	1	1	0	1.026	-17.00	9.5	3.4	0.0	0.0	33.0	0.0	0.0	0.0
20 Bus 20	33	1	1	0	1.030	-16.80	2.2	0.7	0.0	0.0	33.0	0.0	0.0	0.0
21 Bus 21	33	1	1	0	1.033	-16.42	17.5	11.2	0.0	0.0	33.0	0.0	0.0	0.0
22 Bus 22	33	1	1	0	1.033	-16.41	0.0	0.0	0.0	0.0	33.0	0.0	0.0	0.0
23 Bus 23	33	1	1	0	1.027	-16.61	3.2	1.6	0.0	0.0	33.0	0.0	0.0	0.0
24 Bus 24	33	1	1	0	1.021	-16.78	8.7	6.7	0.0	0.0	33.0	0.0	0.0	0.043
25 Bus 25	33	1	1	0	1.017	-16.35	0.0	0.0	0.0	0.0	33.0	0.0	0.0	0.0
26 Bus 26	33	1	1	0	1.000	-16.77	3.5	2.3	0.0	0.0	33.0	0.0	0.0	0.0
27 Cloverdle	33	1	1	0	1.023	-15.82	0.0	0.0	0.0	0.0	33.0	0.0	0.0	0.0
28 Cloverdle	132	1	1	0	1.007	-11.97	0.0	0.0	0.0	0.0	132.0	0.0	0.0	0.0
29 Bus 29	33	1	1	0	1.003	-17.06	2.4	0.9	0.0	0.0	33.0	0.0	0.0	0.0
30 Bus 30	33	1	1	0	0.992	-17.94	10.6	1.9	0.0	0.0	33.0	0.0	0.0	0.0

-999

BRANCH DATA FOLLOWS					41 ITEMS									
1 2	1	1	1	0	0.0192	0.0575	0.0528	0	0	0	0.0	0.0	0.0	0.0
1 3	1	1	1	0	0.0452	0.1652	0.0408	0	0	0	0.0	0.0	0.0	0.0
2 4	1	1	1	0	0.0570	0.1737	0.0368	0	0	0	0.0	0.0	0.0	0.0
3 4	1	1	1	0	0.0132	0.0379	0.0084	0	0	0	0.0	0.0	0.0	0.0
2 5	1	1	1	0	0.0472	0.1983	0.0418	0	0	0	0.0	0.0	0.0	0.0
2 6	1	1	1	0	0.0581	0.1763	0.0374	0	0	0	0.0	0.0	0.0	0.0
4 6	1	1	1	0	0.0119	0.0414	0.0090	0	0	0	0.0	0.0	0.0	0.0
5 7	1	1	1	0	0.0460	0.1160	0.0204	0	0	0	0.0	0.0	0.0	0.0
6 7	1	1	1	0	0.0267	0.0820	0.0170	0	0	0	0.0	0.0	0.0	0.0
6 8	1	1	1	0	0.0120	0.0420	0.0090	0	0	0	0.0	0.0	0.0	0.0
6 9	1	1	1	0	0.0	0.2080	0.0	0	0	0	0.978	0.0	0.0	0.0
6 10	1	1	1	0	0.0	0.5560	0.0	0	0	0	0.969	0.0	0.0	0.0
9 11	1	1	1	0	0.0	0.2080	0.0	0	0	0	0.0	0.0	0.0	0.0
9 10	1	1	1	0	0.0	0.1100	0.0	0	0	0	0.0	0.0	0.0	0.0
4 12	1	1	1	0	0.0	0.2560	0.0	0	0	0	0.932	0.0	0.0	0.0
12 13	1	1	1	0	0.0	0.1400	0.0	0	0	0	0.0	0.0	0.0	0.0
12 14	1	1	1	0	0.1231	0.2559	0.0	0	0	0	0.0	0.0	0.0	0.0
12 15	1	1	1	0	0.0662	0.1304	0.0	0	0	0	0.0	0.0	0.0	0.0
12 16	1	1	1	0	0.0945	0.1987	0.0	0	0	0	0.0	0.0	0.0	0.0
14 15	1	1	1	0	0.2210	0.1997	0.0	0	0	0	0.0	0.0	0.0	0.0
16 17	1	1	1	0	0.0524	0.1923	0.0	0	0	0	0.0	0.0	0.0	0.0
15 18	1	1	1	0	0.1073	0.2185	0.0	0	0	0	0.0	0.0	0.0	0.0
18 19	1	1	1	0	0.0639	0.1292	0.0	0	0	0	0.0	0.0	0.0	0.0
19 20	1	1	1	0	0.0340	0.0680	0.0	0	0	0	0.0	0.0	0.0	0.0
10 20	1	1	1	0	0.0936	0.2090	0.0	0	0	0	0.0	0.0	0.0	0.0
10 17	1	1	1	0	0.0324	0.0845	0.0	0	0	0	0.0	0.0	0.0	0.0
10 21	1	1	1	0	0.0348	0.0749	0.0	0	0	0	0.0	0.0	0.0	0.0
10 22	1	1	1	0	0.0727	0.1499	0.0	0	0	0	0.0	0.0	0.0	0.0
21 22	1	1	1	0	0.0116	0.0236	0.0	0	0	0	0.0	0.0	0.0	0.0
15 23	1	1	1	0	0.1000	0.2020	0.0	0	0	0	0.0	0.0	0.0	0.0
22 24	1	1	1	0	0.1150	0.1790	0.0	0	0	0	0.0	0.0	0.0	0.0
23 24	1	1	1	0	0.1320	0.2700	0.0	0	0	0	0.0	0.0	0.0	0.0
24 25	1	1	1	0	0.1885	0.3292	0.0	0	0	0	0.0	0.0	0.0	0.0
25 26	1	1	1	0	0.2544	0.3800	0.0	0	0	0	0.0	0.0	0.0	0.0
25 27	1	1	1	0	0.1093	0.2087	0.0	0	0	0	0.0	0.0	0.0	0.0
28 27	1	1	1	0	0.0	0.3960	0.0	0	0	0	0.968	0.0	0.0	0.0

```

27 29 1 1 1 0 0.2198 0.4153 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
27 30 1 1 1 0 0.3202 0.6027 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
29 30 1 1 1 0 0.2399 0.4533 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
8 28 1 1 1 0 0.0636 0.2000 0.0428 0 0 0 0 0 0.0 0.0 0.0 0.0
6 28 1 1 1 0 0.0169 0.0599 0.0130 0 0 0 0 0 0.0 0.0 0.0 0.0
-999
LOSS ZONES FOLLOWS 1 ITEMS
1 IEEE 30 BUS
-99
INTERCHANGE DATA FOLLOWS 1 ITEMS
-9
1 2 Claytor 132 0.0 999.99 IEEE30 IEEE 30 Bus Test Case
TIE LINES FOLLOWS 0 ITEMS
-999
END OF DATA

```

### 3.2.2 MATLAB CODES FOR Y-BUS

```

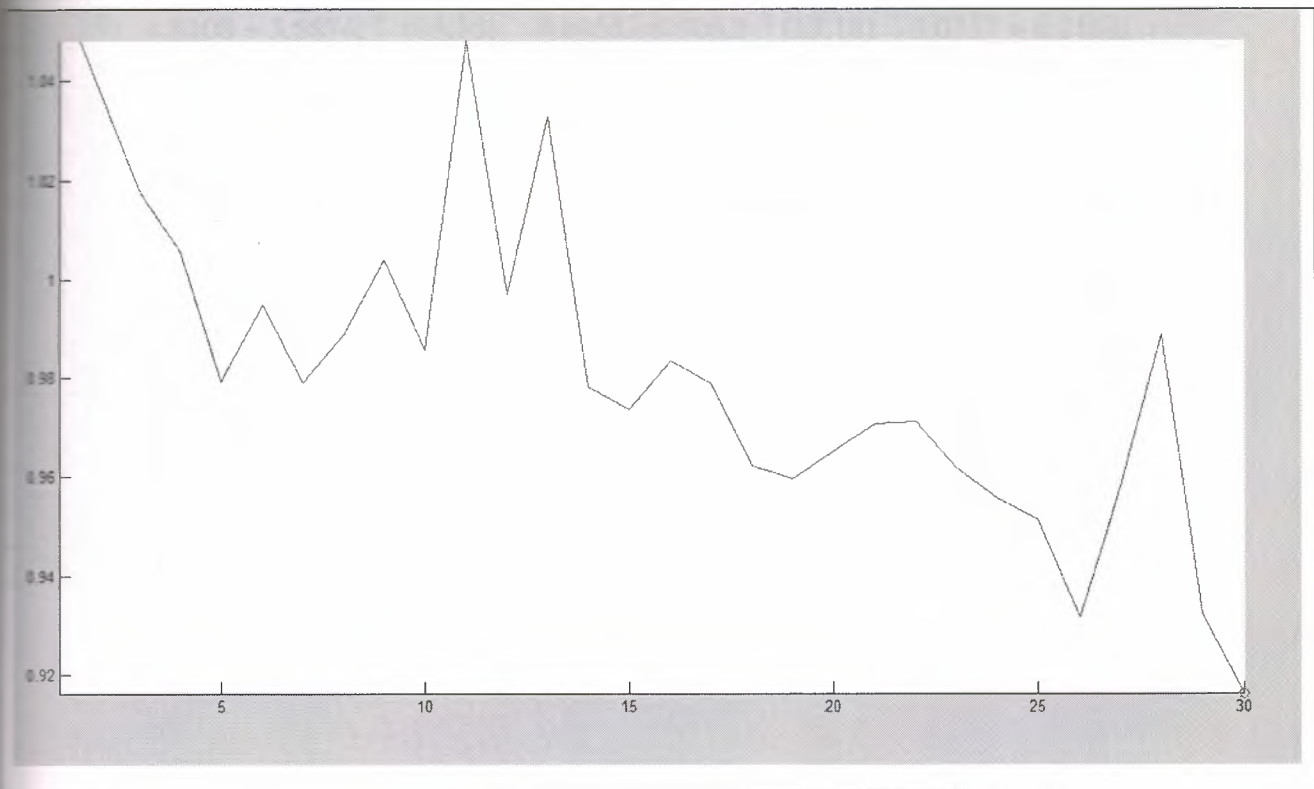
out=readcf('G:\son\data\ieee30bus.cf'); % Read data from IEEE Common Data
Formatted File
out.Ybus=bldybus(out); % Build the Y-bus matrix
out.snet=
sparse(out.Machine.BusRef,1,out.Machine.MW+j*out.Machine.MVAR,out.Bus.n,1)...
-sparse(out.Load.BusRef,1,out.Load.MW+j*out.Load.MVAR,out.Bus.n,1);
out=pfsolve(out);
YBUS= zeros(size(out.Ybus));
YBUS(:, :) = out.Ybus(:, :);
comet (out.Bus.Voltages); figure(gcf); % Plot Graph

```



### 3.2.3

### VOLTAGE - BUS GRAPH for IEEE 30 BUS SYSTEM



### 3.2.4

### Ybus for IEEE 30 Bus System

		(1,1)	6.7655 -21.2316i	(2,1)	-5.2246+15.6467i
(3,1)	-1.5409 + 5.6317i	(1,2)	-5.2246+15.6467i	(2,2)	9.7523 -30.6487i
(4,2)	-1.7055 + 5.1974i	(5,2)	-1.1360 + 4.7725i	(6,2)	-1.6861 + 5.1165i
(1,3)	-1.5409 + 5.6317i	(3,3)	9.7363 -29.1379i	(4,3)	-8.1954+23.5309i
(2,4)	-1.7055 + 5.1974i	(3,4)	-8.1954+23.5309i	(4,4)	16.3141 54.9186i
(6,4)	-6.4131+22.3112i	(12,4)	0 + 3.9063i	(2,5)	-1.1360 + 4.7725i
(5,5)	4.0900 -12.1906i	(7,5)	-2.9540 + 7.4493i	(2,6)	-1.6861 + 5.1165i
(4,6)	-6.4131+22.3112i	(6,6)	22.3416- 82.4935i	(7,6)	-3.5902+11.0261i
(8,6)	-6.2893+22.0126i	(9,6)	0 + 4.8077i	(10,6)	0 + 1.7986i
(28,6)	-4.3628 +15.4636i	(5,7)	-2.9540 + 7.4493i	(6,7)	-3.5902+11.0261i
(7,7)	6.5442 -18.4567i	(6,8)	-6.2893+22.0126i	(8,8)	7.7333 -26.5275i
(28,8)	-1.4440 +4.5408i	(6,9)	0 + 4.8077i	(9,9)	0 -18.7063i
(10,9)	0 + 9.0909i	(11,9)	0 + 4.8077i	(6,10)	0 + 1.7986i
(9,10)	0 + 9.0909i	(10,10)	13.4620 -41.3838i	(17,10)	-3.9560 +10.3174i
(20,10)	-1.7848+3.9854i	(21,10)	-5.1019+10.9807i	(22,10)	-2.6193 + 5.4008i
(9,11)	0 + 4.8077i	(11,11)	0 - 4.8077i	(4,12)	0 + 3.9063i
(12,12)	6.5740 -24.4242i	(13,12)	0 + 7.1429i	(14,12)	-1.5266 + 3.1734i
(15,12)	-3.0954 + 6.0973i	(16,12)	-1.9520 + 4.1044i	(12,13)	0 + 7.1429i
(13,13)	0 - 7.1429i	(12,14)	-1.5266 + 3.1734i	(14,14)	4.0175 - 5.4243i
(15,14)	-2.4910 + 2.2509i	(12,15)	-3.0954 +6.0973i	(14,15)	-2.4910 + 2.2509i

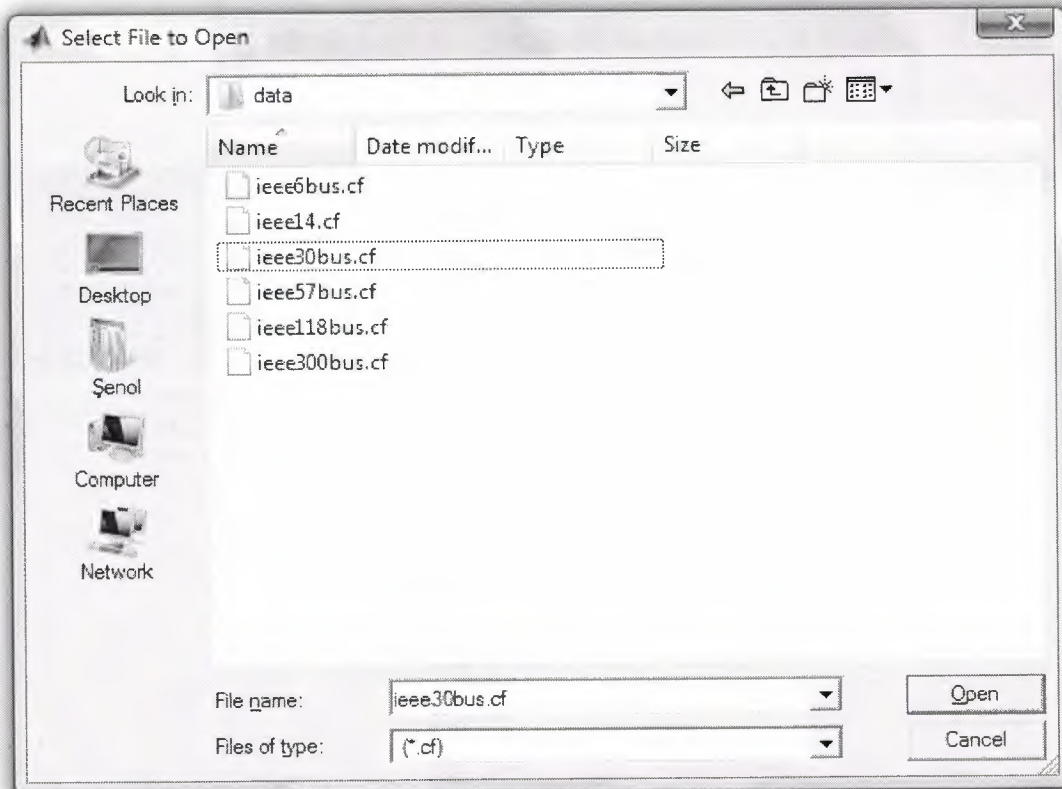


(15,15) 9.3655 -16.0116i	(18,15) -1.8108 + 3.6874i	(23,15) -1.9683 + 3.9761i
(12,16) -1.9520 + 4.1044i	(16,16) 3.2711 - 8.9451i	(17,16) -1.3191 + 4.8408i
(10,17) -3.9560 +10.3174i	(16,17) -1.3191 + 4.8408i	(17,17) 5.2751 -15.1582i
(15,18) -1.8108 + 3.6874i	(18,18) 4.8865 - 9.9062i	(19,18) -3.0757 + 6.2188i
(18,19) -3.0757 + 6.2188i	(19,19) 8.9580 -17.9835i	(20,19) -5.8824 +11.7647i
(10,20) -1.7848 + 3.9854i	(19,20) -5.8824 +11.7647i	(20,20) 7.6672 -15.7501i
(10,21) -5.1019 +10.9807i	(21,21) 21.8765 -45.1084i	(22,21) -16.7746+34.1277i
(10,22) -2.6193 + 5.4008i	(21,22) -16.7746+34.1277i	(22,22) 21.9345 -43.4829i
(24,22) -2.5405 + 3.9544i	(15,23) -1.9683 + 3.9761i	(23,23) 3.4298 - 6.9653i
(24,23) -1.4614 + 2.9892i	(22,24) -2.5405 + 3.9544i	(23,24) -1.4614 + 2.9892i
(24,24) 5.3118 - 9.1883i	(25,24) -1.3099 + 2.2876i	(24,25) -1.3099 + 2.2876i
(25,25) 4.4957 - 7.8650i	(26,25) -1.2165 + 1.8171i	(27,25) -1.9693 + 3.7602i
(25,26) -1.2165 + 1.8171i	(26,26) 1.2165 - 1.8171i	(25,27) -1.9693 + 3.7602i
(27,27) 3.6523 - 9.4604i	(28,27) 0 + 2.5253i	(29,27) -0.9955 + 1.8810i
(30,27) -0.6875 + 1.2940i	(6,28) -4.3628 +15.4636i	(8,28) -1.4440 + 4.5408i
(27,28) 0 + 2.5253i	(28,28) 5.8068 -22.5017i	(27,29) -0.9955 + 1.8810i
(29,29) 1.9076 - 3.6044i	(30,29) -0.9121 + 1.7234i	(27,30) -0.6875 + 1.2940i
(29,30) -0.9121 + 1.7234i	(30,30) 1.5995 - 3.0173i	

### 3.3

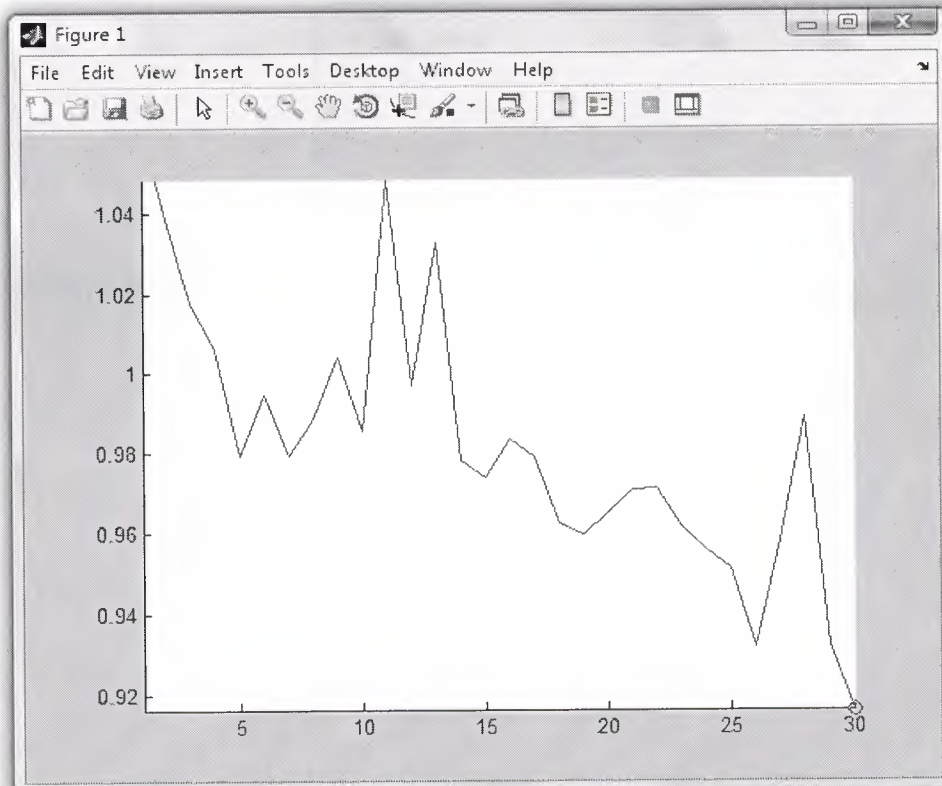
## **How To Run** (screenshot)

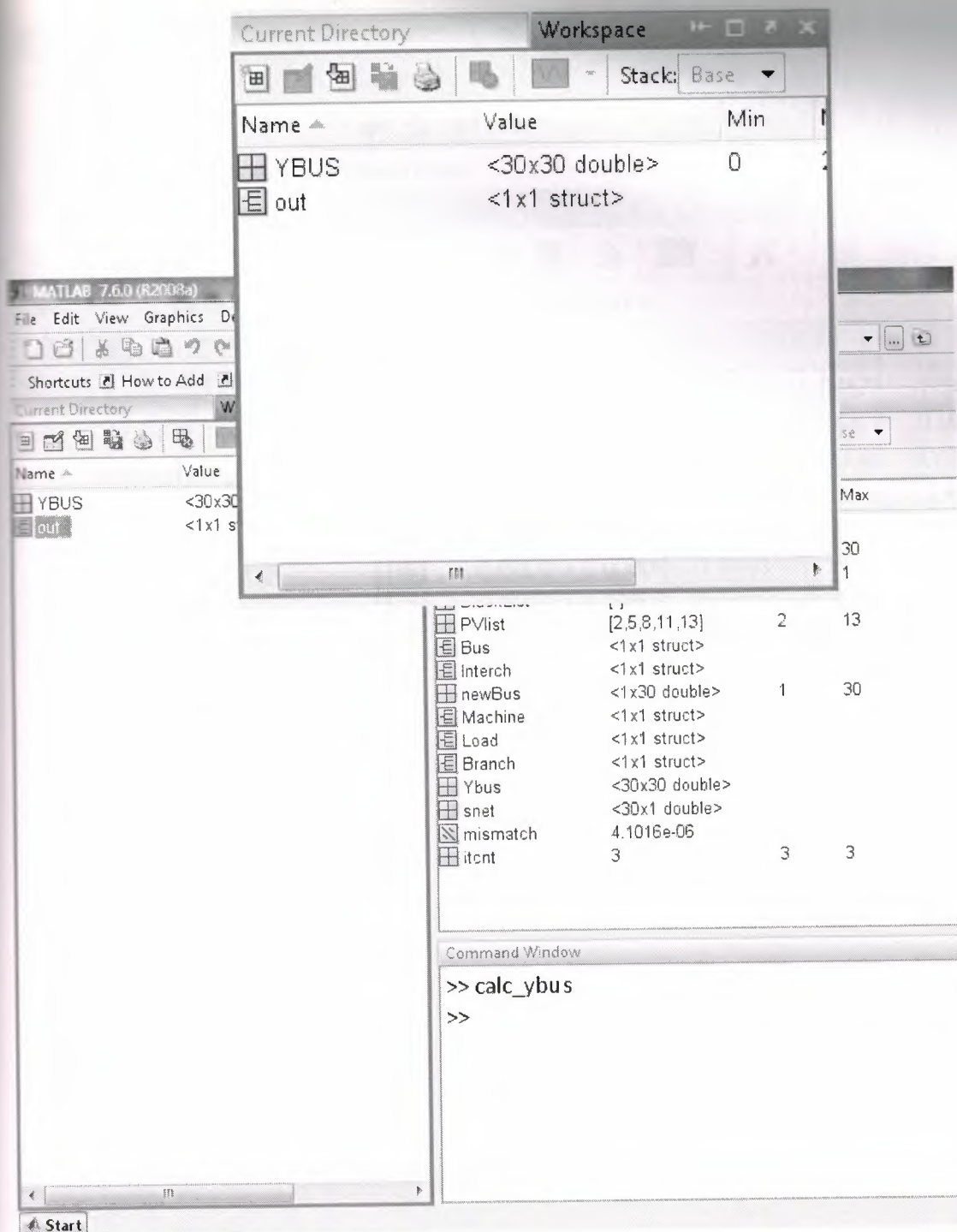
&gt;&gt; calc\_ybus



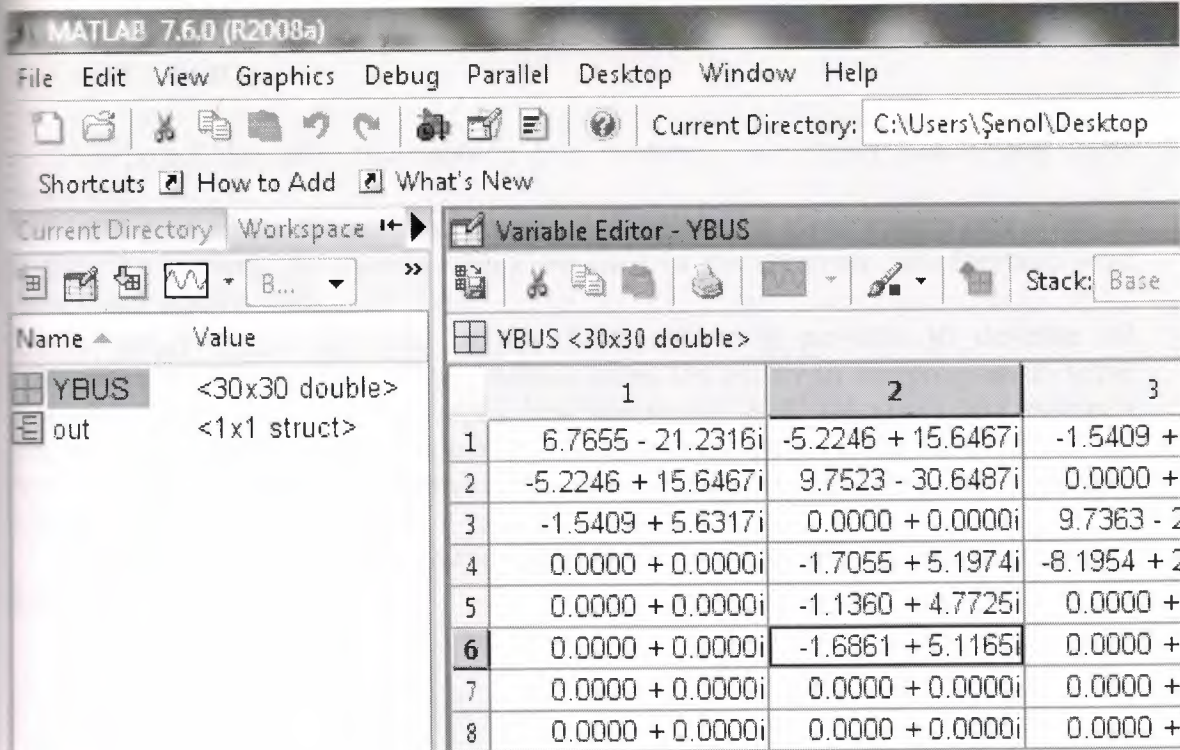
&gt;&gt; calc\_ybus

&gt;&gt;











#### 4. CONCLUSION

This project describes transmission line modeling, admittance matrix and Ybus formation.

The main goal of the Project is to prepare and calculate a given system of a Ybus in matlab. As systems, the system which are used in the common data formats, were chosen and used of IEEE.

The program was prepared in this database and it provides to describe all information in this database. That database gives the ability to the program to solve more than one system. The program's best feature of IEEE calculates any system's Ybus to describes the format of common data. Disadvantages of using this program that can not calculate other data formats. But this program can calculate all of the 'ieee common data format'.

This program can be developed to calculate all systems' ybus by adding other data formats in the future.

## 5. REFERENCES

- [1] R. Fried, R. S. Cherkaoui, and C. C. Enz, "Low-Power CMOS, Analog Transient-Stability-Simulator for a Two-Machine Power System," presented at ISCAS, Hong-Kong, 1997.
- [2] H. Doi, M. Goto, T. Kawai, S. Yokokawa, and T. Suzuki, "Advanced Power-System Analog Simulator," in IEEE Transactions on Power Systems, vol. 5, 1990, pp. 962-968.
- [3] P. G. McLaren, P. Forsyth, A. Perks, and P. R. Bishop, "New Simulation Tools for Power Systems," presented at Transmission and Distribution Conference and Exposition IEEE/PES, 2001.
- [4] X. G. Wang, D. A. Woodford, R. Kuffel, and R. Wierckx, "A real-time transmission line model for a digital TNA," in IEEE Transactions on Power Delivery, vol. 11, 1996, pp. 1092-1097.
- [5] R. C. Durie and C. Pottle, "An Extensible Real-Time Digital Transient Network Analyzer," in IEEE Transactions on Power Systems, vol. 8, 1993, pp. 84-89.
- [6] R. Fried, R. S. Cherkaoui, C. C. Enz, A. Germond, and E. A. Vittoz, "Approaches for analog VLSI simulation of the transient stability of large power networks," IEEE Transactions on Circuits and Systems I-Fundamental Theory and Applications, vol. 46, pp. 1249-1263, 1999
- [7] J. J. Grainger and J. William D. Stevenson, Power System Analysis: McGraw-Hill, 1994.
- [8] A. R. Bergen and V. Vittal, Power System Analysis, 2nd ed: Prentice-Hall, 2000.
- [9] K. H. LaCommare and J. H. Eto, "Understanding the Cost of Power Interruptions to U.S. Electricity Consumers," in Energy Analysis Department. Berkeley: University of California Berkeley, 2004.
- [10] IEEE Committee Report, "Common format for exchange of solved load flow data," IEEE Transaction on Power Apparatus and Systems, Volume PAS-92, Number 6 Pages 1916-1925

**6.**

## **APPENDIX A**

### **IEEE(6-14-57) Bus System Configuration and Ybus**

## Ybus for IEEE 6 Bus System

	1	2	3	4	5	6
1	$4.0063 - 11.7829i$	$-2.0000 + 4.0000i$	$0.0000 + 0.0000i$	$-1.1765 + 4.7059i$	$-0.8299 + 3.1120i$	$0.0000 + 0.0000i$
2	$-2.0000 + 4.0000i$	$9.3283 - 23.2480i$	$-0.7692 + 3.8462i$	$-4.0000 + 8.0000i$	$-1.0000 + 3.0000i$	$-1.5590 + 4.4543i$
3	$0.0000 + 0.0000i$	$-0.7692 + 3.8462i$	$4.1557 - 16.5998i$	$0.0000 + 0.0000i$	$-1.4634 + 3.1707i$	$-1.9231 + 9.6154i$
4	$-1.1765 + 4.7059i$	$-4.0000 + 8.0000i$	$0.0000 + 0.0000i$	$6.1765 - 14.6709i$	$-1.0000 + 2.0000i$	$0.0000 + 0.0000i$
5	$-0.8299 + 3.1120i$	$-1.0000 + 3.0000i$	$-1.4634 + 3.1707i$	$-1.0000 + 2.0000i$	$5.2933 - 14.2103i$	$-1.0000 + 3.0000i$
6	$0.0000 + 0.0000i$	$-1.5590 + 4.4543i$	$-1.9231 + 9.6154i$	$0.0000 + 0.0000i$	$-1.0000 + 3.0000i$	$4.4821 - 17.0372i$

## Ybus for IEEE 14 Bus System

(1,1)	6.0250 -19.4471i	(2,1)	-4.9991 +15.2631i
(5,1)	-1.0259 + 4.2350i	(1,2)	-4.9991 +15.2631i
(2,2)	9.5213 -30.2721i	(3,2)	-1.1350 + 4.7819i
(4,2)	-1.6860 + 5.1158i	(5,2)	-1.7011 + 5.1939i
(2,3)	-1.1350 + 4.7819i	(3,3)	3.1210 - 9.8224i
(4,3)	-1.9860 + 5.0688i	(2,4)	-1.6860 + 5.1158i
(3,4)	-1.9860 + 5.0688i	(4,4)	10.5130 -38.3197i
(5,4)	-6.8410 +21.5786i	(7,4)	0 + 4.7819i
(9,4)	0 + 1.7980i	(1,5)	-1.0259 + 4.2350i
(2,5)	-1.7011 + 5.1939i	(4,5)	-6.8410 +21.5786i
(5,5)	9.5680 -34.9335i	(6,5)	0 + 3.9679i
(5,6)	0 + 3.9679i	(6,6)	6.5799 -17.3407i
(11,6)	-1.9550 + 4.0941i	(12,6)	-1.5260 + 3.1760i
(13,6)	-3.0989 + 6.1028i	(4,7)	0 + 4.7819i
(7,7)	0 -19.5490i	(8,7)	0 + 5.6770i
(9,7)	0 + 9.0901i	(7,8)	0 + 5.6770i
(8,8)	0 - 5.6770i	(4,9)	0 + 1.7980i
(7,9)	0 + 9.0901i	(9,9)	5.3261 -24.0925i
(10,9)	-3.9020 +10.3654i	(14,9)	-1.4240 + 3.0291i
(9,10)	-3.9020 +10.3654i	(10,10)	5.7829 -14.7683i
(11,10)	-1.8809 + 4.4029i	(6,11)	-1.9550 + 4.0941i
(10,11)	-1.8809 + 4.4029i	(11,11)	3.8359 - 8.4970i
(6,12)	-1.5260 + 3.1760i	(12,12)	4.0150 - 5.4279i
(13,12)	-2.4890 + 2.2520i	(6,13)	-3.0989 + 6.1028i
(12,13)	-2.4890 + 2.2520i	(13,13)	6.7249 -10.6697i
(14,13)	-1.1370 + 2.3150i	(9,14)	-1.4240 + 3.0291i
(13,14)	-1.1370 + 2.3150i	(14,14)	2.5610 - 5.3440i



## Ybus for IEEE 57 Bus System

(1,1)	14.7682 -56.7180i	(2,1)	-9.7316 +32.8296i	(15,1)	-2.0703 +10.5841i	(16,1)	-1.0203 + 4.6295i
(17,1)	-1.9460 + 8.8304i	(1,2)	-9.7316 +32.8296i	(2,2)	13.4047 -43.2011i	(3,2)	-3.6731 +10.4770i
(2,3)	-3.6731 +10.4770i	(3,3)	16.5925 -52.6285i	(4,3)	-7.6451 +24.9829i	(15,3)	-5.2744 +17.2557i
(3,4)	-7.6451 +24.9829i	(4,4)	12.3855 -41.4802i	(5,4)	-2.9301 + 6.1884i	(6,4)	-1.8103 + 6.2308i
(18,4)	0 + 4.1274i	(4,5)	-2.9301 + 6.1884i	(5,5)	8.9450 -18.9361i	(6,5)	-6.0149 +12.7668i
(4,6)	-1.8103 + 6.2308i	(5,6)	-6.0149 +12.7668i	(6,6)	10.7672 -33.9442i	(7,6)	-1.8512 + 9.4409i
(8,6)	-1.0908 + 5.5666i	(6,7)	-1.8512 + 9.4409i	(7,7)	4.4924 -38.3789i	(8,7)	-2.6413 +13.5293i
(29,7)	0 +15.4321i	(6,8)	-1.0908 + 5.5666i	(7,8)	-2.6413 +13.5293i	(8,8)	7.4704 -38.1044i
(9,8)	-3.7383 +19.0691i	(8,9)	-3.7383 +19.0691i	(9,9)	10.7445 -52.7496i	(10,9)	-1.2486 + 5.6815i
(11,9)	-3.2838 +10.7934i	(12,9)	-0.7103 + 3.2338i	(13,9)	-1.7634 + 5.7923i	(55,9)	0 + 8.2988i
(9,10)	-1.2486 + 5.6815i	(10,10)	2.9080 -27.2478i	(12,10)	-1.6593 + 7.5597i	(51,10)	0 +14.0449i
(9,11)	-3.2838 +10.7934i	(11,11)	7.0922 -31.1451i	(13,11)	-3.8084 +12.5010i	(41,11)	0 + 1.3351i
(43,11)	0 + 6.5359i	(9,12)	-0.7103 + 3.2338i	(10,12)	-1.6593 + 7.5597i	(12,12)	10.9825 -43.4810i
(13,12)	-4.8359 +15.7573i	(16,12)	-2.5960 +11.7254i	(17,12)	-1.1809 + 5.3247i	(9,13)	-1.7634 + 5.7923i
(11,13)	-3.8084 +12.5010i	(12,13)	-4.8359 +15.7573i	(13,13)	20.0729 -70.8010i	(14,13)	-6.4146 +21.0905i
(15,13)	-3.2507 +10.5012i	(49,13)	0 + 5.2356i	(13,14)	-6.4146 +21.0905i	(14,14)	11.6209 -51.3370i
(15,14)	-5.2063 +16.6540i	(46,14)	0 +13.6054i	(1,15)	-2.0703 +10.5841i	(3,15)	-5.2744 +17.2557i
(13,15)	-3.2507 +10.5012i	(14,15)	-5.2063 +16.6540i	(15,15)	15.8016 -64.4964i	(45,15)	0 + 9.5969i
(1,16)	-1.0203 + 4.6295i	(12,16)	-2.5960 +11.7254i	(16,16)	3.6163 -16.3168i	(1,17)	-1.9460 + 8.8304i
(12,17)	-1.1809 + 5.3247i	(17,17)	3.1269 -14.1170i	(4,18)	0 + 4.1274i	(18,18)	0.6762 - 5.0322i
(19,18)	-0.6762 + 1.0048i	(18,19)	-0.6762 + 1.0048i	(19,19)	1.7304 - 2.6215i	(20,19)	-1.0542 + 1.6167i
(19,20)	-1.0542 + 1.6167i	(20,20)	1.0542 - 2.9042i	(21,20)	0 + 1.2875i	(20,21)	0 + 1.2875i
(21,21)	3.8522 - 7.4112i	(22,21)	-3.8522 + 6.1237i	(21,22)	-3.8522 + 6.1237i	(22,22)	49.4366 -76.1290i
(23,22)	-30.0866 +46.1936i	(38,22)	-15.4977 +23.8116i	(22,23)	-30.0866 +46.1936i	(23,23)	31.8698 -48.9394i
(24,23)	-1.7832 + 2.7500i	(23,24)	-1.7832 + 2.7500i	(24,24)	1.7832 -25.5464i	(25,24)	0 + 1.6590i
(26,24)	0 +21.1416i	(24,25)	0 + 1.6590i	(25,25)	2.2870 - 5.0221i	(30,25)	-2.2870 + 3.4220i
(24,26)	0 +21.1416i	(26,26)	1.7985 -23.9103i	(27,26)	-1.7985 + 2.7687i	(26,27)	-1.7985 + 2.7687i
(27,27)	6.5817 -10.1523i	(28,27)	-4.7831 + 7.3837i	(27,28)	-4.7831 + 7.3837i	(28,28)	12.8325 -18.6875i
(29,28)	-8.0494 +11.3038i	(7,29)	0 +15.4321i	(28,29)	-8.0494 +11.3038i	(29,29)	10.6354 -30.0894i
(52,29)	-2.5860 + 3.3535i	(25,30)	-2.2870 + 3.4220i	(30,30)	3.2098 - 4.8288i	(31,30)	-0.9228 + 1.4068i
(30,31)	-0.9228 + 1.4068i	(31,31)	1.5358 - 2.3197i	(32,31)	-0.6130 + 0.9129i	(31,32)	-0.6130 + 0.9129i
(32,32)	14.4517 -14.6712i	(33,32)	-13.8387 +12.7090i	(34,32)	0 + 1.0493i	(32,33)	-13.8387 +12.7090i
(33,33)	13.8387 -12.7090i	(32,34)	0 + 1.0493i	(34,34)	5.9172 - 9.9235i	(35,34)	-5.9172 + 8.8757i
(34,35)	-5.9172 + 8.8757i	(35,35)	15.0029 -20.2200i	(36,35)	-9.0857 +11.3466i	(35,36)	-9.0857 +11.3466i
(36,36)	32.1521 -43.3019i	(37,36)	-13.2993 +16.7847i	(40,36)	-9.7670 +15.1714i	(36,37)	-13.2993 +16.7847i
(37,37)	29.7189 -42.6596i	(38,37)	-4.5149 + 6.9978i	(39,37)	-11.9046 +18.8781i	(22,38)	-15.4977 +23.8116i
(37,38)	-4.5149 + 6.9978i	(38,38)	38.8460 -63.1400i	(44,38)	-6.7881 +13.7406i	(48,38)	-9.4641 +14.6208i
(49,38)	-2.5811 + 3.9727i	(37,39)	-11.9046 +18.8781i	(39,39)	11.9046 -19.6161i	(57,39)	0 + 0.7380i
(36,40)	-9.7670 +15.1714i	(40,40)	9.7670 -16.0083i	(56,40)	0 + 0.8368i	(11,41)	0 + 1.3351i
(41,41)	2.1521 - 6.7773i	(42,41)	-1.2414 + 2.1109i	(43,41)	0 + 2.4272i	(56,41)	-0.9107 + 0.9041i
(41,42)	-1.2414 + 2.1109i	(42,42)	2.4879 - 4.1875i	(56,42)	-1.2465 + 2.0766i	(11,43)	0 + 6.5359i
(41,43)	0 + 2.4272i	(43,43)	0 - 8.9631i	(38,44)	-6.7881 +13.7406i	(44,44)	10.0180 -20.1664i
(45,44)	-3.2299 + 6.4288i	(15,45)	0 + 9.5969i	(44,45)	-3.2299 + 6.4288i	(45,45)	3.2299 -16.0237i
(14,46)	0 +13.6054i	(46,46)	4.4634 -26.8000i	(47,46)	-4.4634 +13.1962i	(46,47)	-4.4634 +13.1962i
(47,47)	25.2841 -39.8497i	(48,47)	-20.8207 +26.6551i	(38,48)	-9.4641 +14.6208i	(47,48)	-20.8207 +26.6551i
(48,48)	33.8192 -46.7403i	(49,48)	-3.5344 + 5.4669i	(13,49)	0 + 5.2356i	(38,49)	-2.5811 + 3.9727i
(48,49)	-3.5344 + 5.4669i	(49,49)	9.6287 -20.2853i	(50,49)	-3.5132 + 5.6140i	(49,50)	-3.5132 + 5.6140i
(50,50)	5.5632 - 8.8680i	(51,50)	-2.0500 + 3.2540i	(10,51)	0 +14.0449i	(50,51)	-2.0500 + 3.2540i
(51,51)	2.0500 -17.2989i	(29,52)	-2.5860 + 3.3535i	(52,52)	7.5056 - 9.7064i	(53,52)	-4.9196 + 6.3529i
(52,53)	-4.9196 + 6.3529i	(53,53)	7.0275 - 8.8939i	(54,53)	-2.1079 + 2.6040i	(53,54)	-2.1079 + 2.6040i
(54,54)	4.2383 - 5.3900i	(55,54)	-2.1304 + 2.7860i	(9,55)	0 + 8.2988i	(54,55)	-2.1304 + 2.7860i
(55,55)	2.1304 -11.0847i	(40,56)	0 + 0.8368i	(41,56)	-0.9107 + 0.9041i	(42,56)	-1.2465 + 2.0766i
(56,56)	3.9350 - 6.4740i	(57,56)	-1.7778 + 2.6564i	(39,57)	0 + 0.7380i	(56,57)	-1.7778 + 2.6564i
(57,57)	1.7778 - 3.3944i						

**7.**

## **APPENDIX B**

### **CD**

(Ybus Program-  
Systems Configurations, Datas,Outputs)