CHAPTER 6

POWER FLOW ANALYSIS

6.1 INTRODUCTION

In the previous chapters, modeling of the major components of an electric power system was discussed. This chapter deals with the steady-state analysis of an interconnected power system during normal operation. The system is assumed to be operating under balanced condition and is represented by a single-phase network. The network contains hundreds of nodes and branches with impedances specified in per unit on a common MVA base.

Network equations can be formulated systematically in a variety of forms. However, the node-voltage method, which is the most suitable form for many power system analyses, is commonly used. The formulation of the network equations in the nodal admittance form results in complex linear simultaneous algebraic equations in terms of node currents. When node currents are specified, the set of linear equations can be solved for the node voltages. However, in a power system, powers are known rather than currents. Thus, the resulting equations in terms of power, known as the *power flow equation*, become nonlinear and must be solved by iterative techniques. Power flow studies, commonly referred to as *load flow*, are the backbone of power system analysis and design. They are necessary for planning, operation, economic scheduling and exchange of power between utilities. In addition, power flow analysis is required for many other analyses such as transient stability and contingency studies.

In this chapter, the bus admittance matrix of the node-voltage equation is formulated, and a *MATLAB* function named **ybus** is developed for the systematic formation of the bus admittance matrix. Next, two commonly used iterative techniques, namely Gauss-Seidel and Newton-Raphson methods for the solution of nonlinear algebraic equations, are discussed. These techniques are employed in the solution of power flow problems. Three programs **lfgauss, lfnewton**, and **decouple** are developed for the solution of power flow problems by Gauss-Seidel, Newton-Raphson, and the fast decoupled power flow, respectively.

6.2 BUS ADMITTANCE MATRIX

In order to obtain the node-voltage equations, consider the simple power system shown in Figure 6.1 where impedances are expressed in per unit on a common MVA base and for simplicity resistances are neglected. Since the nodal solution is based upon Kirchhoff's current law, impedances are converted to admittance, i.e.,

$$y_{ij} = \frac{1}{z_{ij}} = \frac{1}{r_{ij} + jx_{ij}}$$

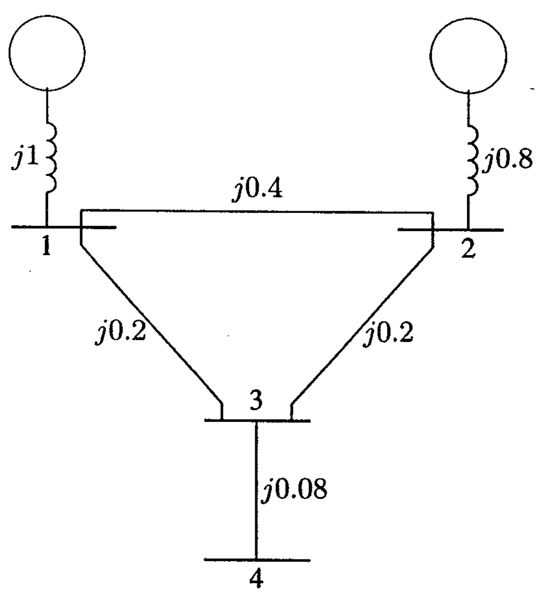

FIGURE 6.1
The impedance diagram of a simple system.

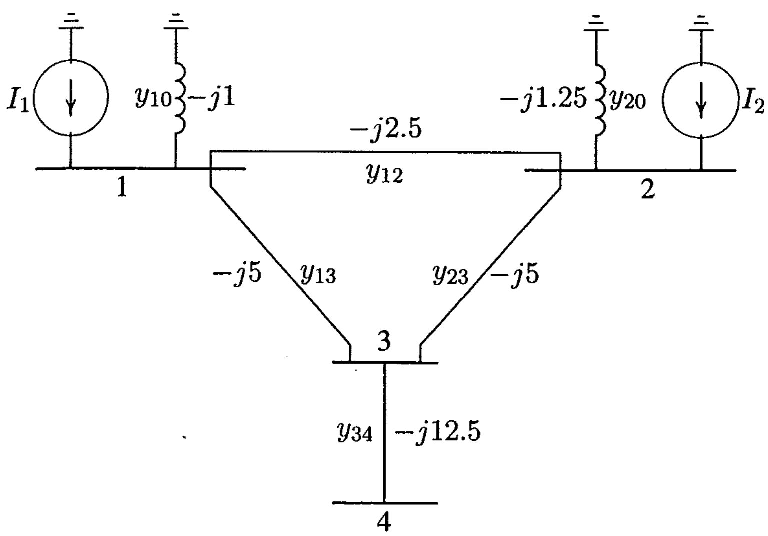

FIGURE 6.2
The admittance diagram for system of Figure 6.1.

The circuit has been redrawn in Figure 6.2 in terms of admittances and transformation to current sources. Node 0 (which is normally ground) is taken as reference. Applying KCL to the independent nodes 1 through 4 results in

$$
\begin{aligned}
I_1 &= y_{10}V_1 + y_{12}(V_1 - V_2) + y_{13}(V_1 - V_3)\\
I_2 &= y_{20}V_2 + y_{12}(V_2 - V_1) + y_{23}(V_2 - V_3)\\
0 &= y_{23}(V_3 - V_2) + y_{13}(V_3 - V_1) + y_{34}(V_3 - V_4)\\
0 &= y_{34}(V_4 - V_3)
\end{aligned}
$$

Rearranging these equations yields

$$
\begin{aligned}
I_1 &= (y_{10} + y_{12} + y_{13})V_1 - y_{12}V_2 - y_{13}V_3\\
I_2 &= -y_{12}V_1 + (y_{20} + y_{12} + y_{23})V_2 - y_{23}V_3\\
0 &= -y_{13}V_1 - y_{23}V_2 + (y_{13} + y_{23} + y_{34})V_3 - y_{34}V_4\\
0 &= -y_{34}V_3 + y_{34}V_4
\end{aligned}
$$

We introduce the following admittances

$$
\begin{aligned}
Y_{11} &= y_{10} + y_{12} + y_{13}\\
Y_{22} &= y_{20} + y_{12} + y_{23}
\end{aligned}
$$

$$Y_{33} = y_{13} + y_{23} + y_{34}$$
$$Y_{44} = y_{34}$$
$$Y_{12} = Y_{21} = -y_{12}$$
$$Y_{13} = Y_{31} = -y_{13}$$
$$Y_{23} = Y_{32} = -y_{23}$$
$$Y_{34} = Y_{43} = -y_{34}$$

The node equation reduces to

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

In the above network, since there is no connection between bus 1 and 4, $Y_{14} = Y_{41} = 0$; similarly $Y_{24} = Y_{42} = 0$.

Extending the above relation to an n bus system, the node-voltage equation in matrix form is

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_i \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1i} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2i} & \cdots & Y_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ Y_{i1} & Y_{i2} & \cdots & Y_{ii} & \cdots & Y_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{ni} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_i \\ \vdots \\ V_n \end{bmatrix} \tag{6.1}$$

or

$$\mathbf{I}_{bus} = \mathbf{Y}_{bus}\,\mathbf{V}_{bus} \tag{6.2}$$

where $\mathbf{I}_{bus}$ is the vector of the injected bus currents (i.e., external current sources). The current is positive when flowing towards the bus, and it is negative if flowing away from the bus. $\mathbf{V}_{bus}$ is the vector of bus voltages measured from the reference node (i.e., node voltages). $\mathbf{Y}_{bus}$ is known as the *bus admittance matrix.* The diagonal element of each node is the sum of admittances connected to it. It is known as the *self-admittance* or *driving point admittance*, i.e.,

$$Y_{ii} = \sum_{j=0}^{n} y_{ij} \qquad j \neq i \tag{6.3}$$

The off-diagonal element is equal to the negative of the admittance between the nodes. It is known as the *mutual admittance* or *transfer admittance*, i.e.,

$$Y_{ij} = Y_{ji} = -y_{ij} \tag{6.4}$$

When the bus currents are known, (6.2) can be solved for the n bus voltages.

$$\mathbf{V}_{bus} = \mathbf{Y}_{bus}^{-1}\,\mathbf{I}_{bus} \tag{6.5}$$

The inverse of the bus admittance matrix is known as the *bus impedance matrix* Z_{bus}. The admittance matrix obtained with one of the buses as reference is nonsingular. Otherwise the nodal matrix is singular.

Inspection of the bus admittance matrix reveals that the matrix is symmetric along the leading diagonal, and we need to store the upper triangular nodal admittance matrix only. In a typical power system network, each bus is connected to only a few nearby buses. Consequently, many off-diagonal elements are zero. Such a matrix is called *sparse*, and efficient numerical techniques can be applied to compute its inverse. By means of an appropriately ordered triangular decomposition, the inverse of a sparse matrix can be expressed as a product of sparse matrix factors, thereby giving an advantage in computational speed, storage and reduction of round-off errors. However, Z_{bus}, which is required for short-circuit analysis, can be obtained directly by the method of *building algorithm* without the need for matrix inversion. This technique is discussed in Chapter 9.

Based on (6.3) and (6.4), the bus admittance matrix for the network in Figure 6.2 obtained by inspection is

$$Y_{bus} = \begin{bmatrix} -j8.50 & j2.50 & j5.00 & 0 \\ j2.50 & -j8.75 & j5.00 & 0 \\ j5.00 & j5.00 & -j22.50 & j12.50 \\ 0 & 0 & j12.50 & -j12.50 \end{bmatrix}.$$

A function called **Y = ybus(zdata)** is written for the formation of the bus admittance matrix. **zdata** is the line data input and contains four columns. The first two columns are the line bus numbers and the remaining columns contain the line resistance and reactance in per unit. The function returns the bus admittance matrix. The algorithm for the bus admittance program is very simple and basic to power system programming. Therefore, it is presented here for the reader to study and understand the method of solution. In the program, the line impedances are first converted to admittances. Y is then initialized to zero. In the first loop, the line data is searched, and the off-diagonal elements are entered. Finally, in a nested loop, line data is searched to find the elements connected to a bus, and the diagonal elements are thus formed.

The following is a program for building the bus admittance matrix:

```
function[Y] = ybus(zdata)
nl=zdata(:,1); nr=zdata(:,2); R=zdata(:,3); X=zdata(:,4);
nbr=length(zdata(:,1)); nbus = max(max(nl), max(nr));
Z = R + j*X;                                %branch impedance
```

27	1.026	-15.912	0.000	0.000	0.000	0.000	0.00
28	1.011	-12.057	0.000	0.000	0.000	0.000	0.00
29	1.006	-17.136	2.400	0.900	0.000	0.000	0.00
30	0.995	-18.014	10.600	1.900	0.000	0.000	0.00
Total			283.400	126.200	300.998	125.145	23.30

The output of the **lineflow** is the same as the line flow output of Example 6.9 with the power mismatch as dictated by the fast decoupled method.

PROBLEMS

6.1. A power system network is shown in Figure 6.17. The generators at buses 1 and 2 are represented by their equivalent current sources with their reactances in per unit on a 100-MVA base. The lines are represented by π model where series reactances and shunt reactances are also expressed in per unit on a 100 MVA base. The loads at buses 3 and 4 are expressed in MW and Mvar.

(a) Assuming a voltage magnitude of 1.0 per unit at buses 3 and 4, convert the loads to per unit impedances. Convert network impedances to admittances and obtain the bus admittance matrix by inspection.

(b) Use the function **Y = ybus(zdata)** to obtain the bus admittance matrix. The function argument **zdata** is a matrix containing the line bus numbers, resistance and reactance. (See Example 6.1.)

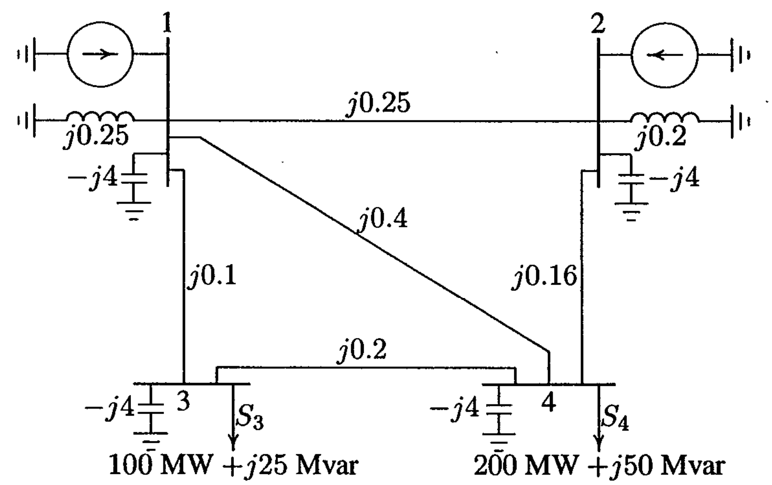

FIGURE 6.17
One-line diagram for Problem 6.1.

6.2. A power system network is shown in Figure 6.18. The values marked are impedances in per unit on a base of 100 MVA. The currents entering buses 1 and 2 are

$$I_1 = 1.38 - j2.72 \text{ pu}$$
$$I_2 = 0.69 - j1.36 \text{ pu}$$

(a) Determine the bus admittance matrix by inspection.
(b) Use the function **Y = ybus(zdata)** to obtain the bus admittance matrix. The function argument **zdata** is a matrix containing the line bus numbers, resistance and reactance. (See Example 6.1.) Write the necessary *MATLAB* commands to obtain the bus voltages.

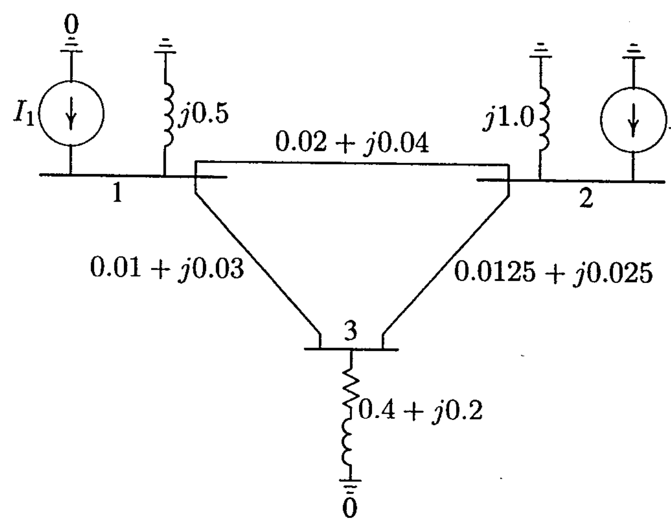

FIGURE 6.18
One-line diagram for Problem 6.2.

6.3. Use Gauss-Seidel method to find the solution of the following equations

$$x_1 + x_1 x_2 = 10$$
$$x_1 + x_2 = 6$$

with the following initial estimates
(a) $x_1^{(0)} = 1$ and $x_2^{(0)} = 1$
(b) $x_1^{(0)} = 1$ and $x_2^{(0)} = 2$
Continue the iterations until $|\Delta x_1^{(k)}|$ and $|\Delta x_2^{(k)}|$ are less than 0.001.