Electrical and Computer Engineering at Lawrence Technological University

Computer Engineering

Electrical Engineering

Adv. Computer Applications In EE

		Biomedical Engineering	Computers	Electronics	Electric Energy Engineering
		Bio-instrumentation	Adv. Digital	Adv. Digital	Intro to Electrical Systems
		Imaging	Computer Architecture	Adv. Electronics	Electric Machines and Drives
		Bio-physics	Computer Elective	Opto-electronics	Energy Elective
			Real-time Operating Systems		Power Electronics
			Embedded Systems		Power Systems
					Process Control (PLCs)
					Alternative Energy (from ME)

The Electric Utility Power Flow Problem

Introduction

The problem of finding all the steady-state voltages, currents, and power (both real and reactive) in an electric utility distribution system is called the power flow (or load flow) problem. The problem is solved using a node-voltage analysis, where the nodes of the system are called buses. Consider the utility distribution system shown in Fig. 1 below. (The size of this system is unrealistically small to make the problem tractable. Also, to make the problem simpler there are no transformers in the system.)

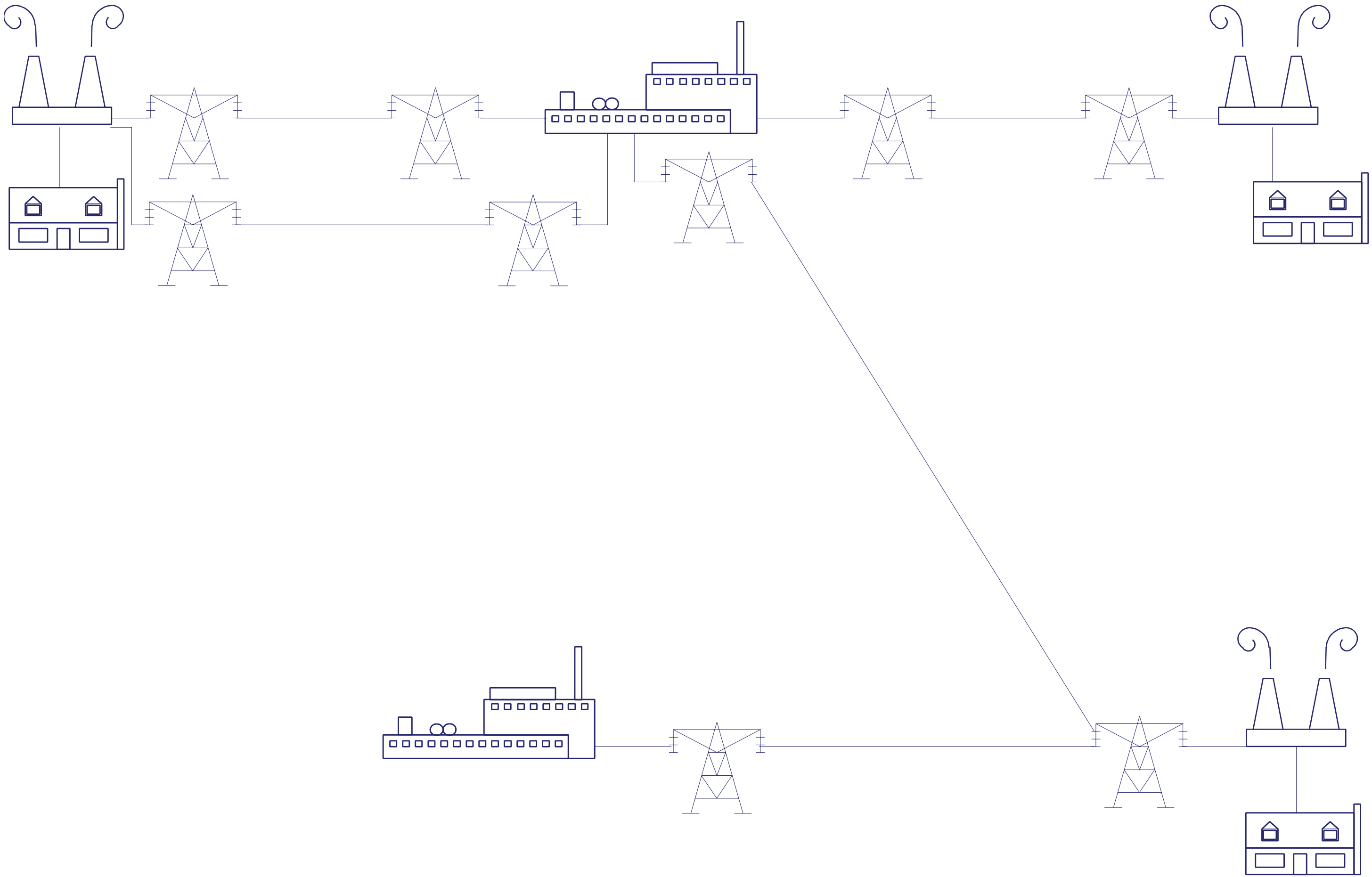


Figure 1 Test System

Since power systems are analyzed by node voltage analysis, where the nodes are called buses, we need to identify the buses. To see where the buses are, compare Fig. 1 to Fig. 2, which is the so-called “one-line“ diagram of the power system shown in Fig.1.

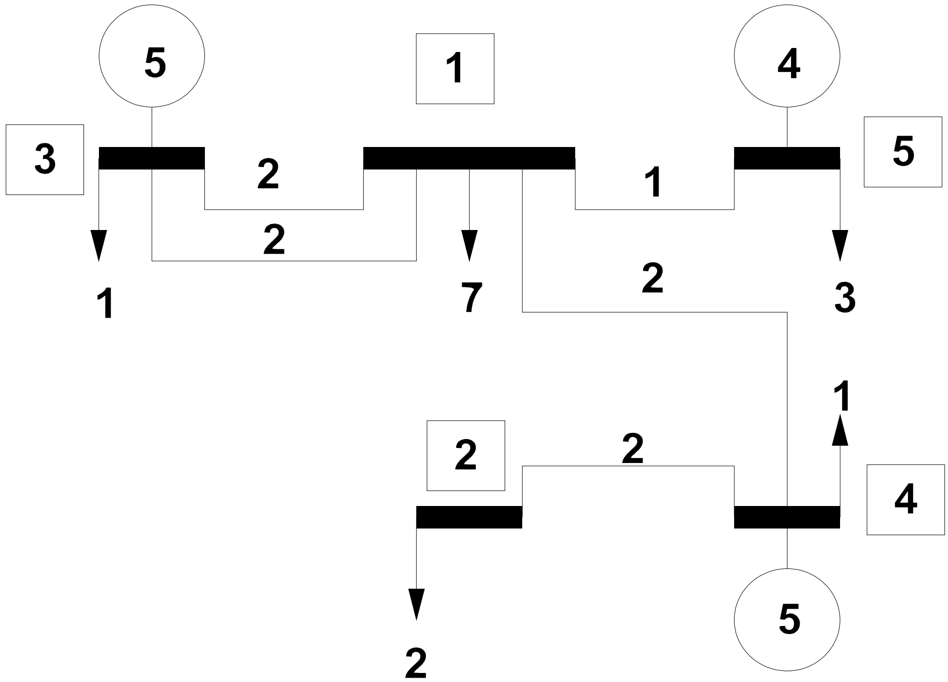


Figure 2 “One-Line” Diagram for Test System

The interpretation of Fig. 2 is as follows. The thick horizontal lines are the buses (constant- voltage-points or nodes), the numbers in the boxes adjacent to the buses are the bus numbers, the circles are the generators, the arrows are load demands, and the thin lines are the transmission lines. The numbers in the circles are the values of the generator powers, the numbers adjacent to the transmission lines are the transmission line powers, and the numbers adjacent to the arrows are the load demands. The generators are referred to by their bus numbers; generator 3 produces 5 units of power, generator 4 produces 5 units of power, and generator 5 produces 4 units of power. Note that in this simplified representation we have assumed that the transmission lines have no resistance and therefore dissipate no power, so that eq(1) below applies to this oversimplified system. The transmission line powers in Fig. 2 are not the powers dissipated in the lines (that power is zero since we have assumed lossless transmission lines), but the power that the transmission lines are carrying. Note for instance that the 5 units of power form the leftmost generator is split between two transmission lines (two units of power each) and one demand (one unit of power). For this system n (the number of busses) is 5, and ℓ (the number of generators) is 3. The symbols n and ℓ are standard throughout the power industry.

If there were no losses in the transmission system (i.e. if $R_{ij} = 0 \ \forall \ i,j$ where i and j are bus numbers) then the real power balance would simply be

$$\sum_{i=1}^{\ell} P_{g_i} = \sum_{k=1}^{n} P_{d_k}$$

where ℓ is the number of generators in the system, and n is the total number of buses in the system. Since the demand (load) powers $P_{d\,i}$ are assumed known, the generator powers could be set arbitrarily as long as (1) is satisfied.

Now the overhead (or underground) transmission lines are modeled as in Fig3.

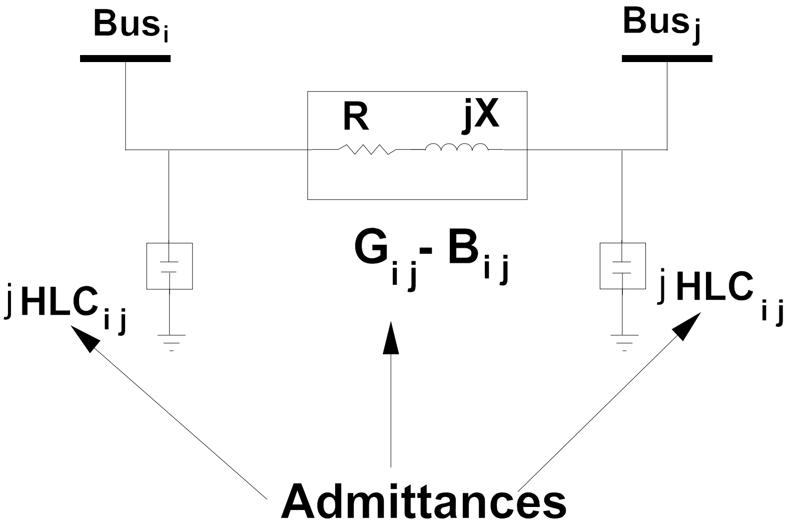


Figure 3 Transmission Line Model

The notion that an overhead transmission line has resistance and inductance should be familiar to most students.

(Recall University Physics II -> $R = \dfrac{\rho l}{A}$ and $L = \dfrac{\mu_0 l}{\pi} \ln\left(\dfrac{d-A}{A}\right)$)

The capacitance shown in Fig. 3 is a consequence of the fact the ground is conductive. We have then, two conductors (the overhead wire is one conductor and the ground is the other) separated by a dielectric (the air between them). This distributed capacitance is lumped together into an ideal capacitor at either end of the transmission line. This is the origin of the term "half line charging capacitance" or HLC in Fig. 3. Similar arguments hold for underground transmission lines.

Loads are represented, not as impedances whose resistance and inductance we know, but as boxes whose real and reactive power dissipation are known. (We shall discuss the notions of real and reactive power shortly.) This is because the utility company cannot easily measure the impedances of its load, but can easily measure the real and reactive power of its loads. In addition there is a wealth of empirical data which indicates that power system loads behave more like constant power devices than like constant impedance devices.

Now since the transmission lines do have finite resistance, there are losses in the system, and further these losses are unknown, since the voltage at some of the buses is unknown. Therefore, only ℓ-1 generators are set to prescribed powers and the other generator (the slack generator) is allowed to assume whatever power is necessary to make up the system losses. For this situation (including real power losses in the system), the real power balance for the system of Fig. 2 is:

$$\sum_{i=1}^{\ell-1} P_{gi} + P_{g\ell} = \sum_{k=1}^{n} P_{dk} + P_{Loss} \qquad \textbf{(1)}$$

where $P_{g\ell}$ is the real power from the slack bus. The slack generator is thus said to "take up the slack" meaning the transmission losses (the $I^2 R$ losses in the resistances of the transmission lines). It should be pointed out that in reality, each generator (including the slack generator) provides some demand (load) power and some loss power. In other words one can't really tell where each generator power goes. However, as a computational aid, one often sets the sum of the non-slack generators equal to the demand powers which makes the slack generator power equal to the system losses.

AC Power (We now digress to discuss the "units of power" in Fig. 2, and the notions of P & Q.)

If we wish to transfer power from a sinusoidal source to a complex load, we use the circuit shown in Fig. 4 below. Note that the load impedance $\mathbf{Z}\angle\theta$ is a complex number, i.e. the load has both a resistive part and a reactive part. In other words we could write $\mathbf{Z\angle\theta = R + jX}$

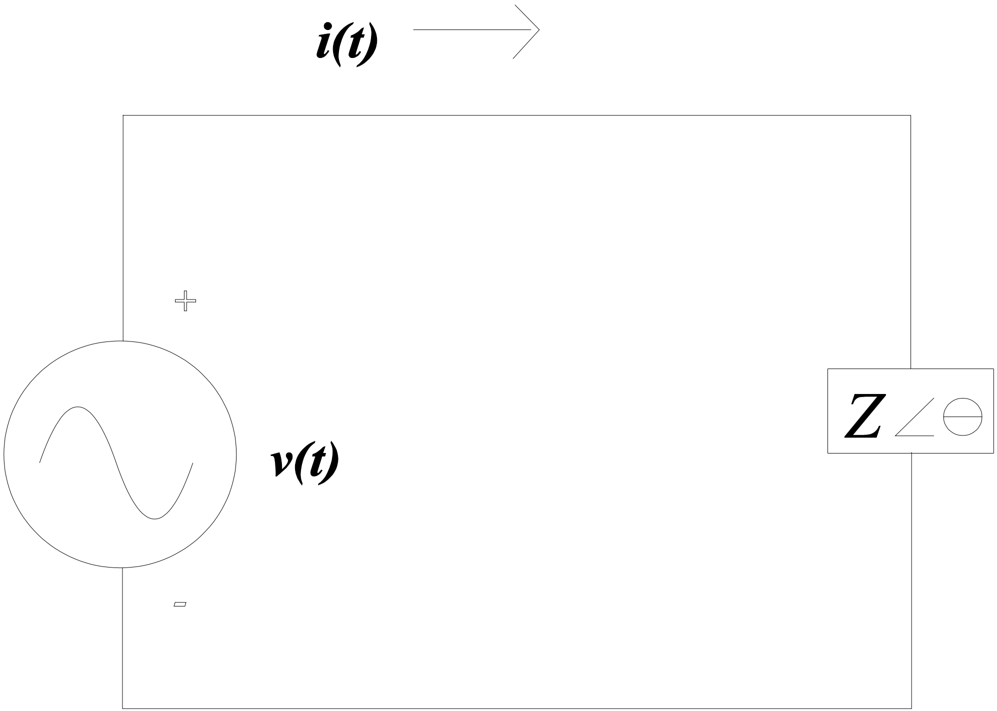


Figure 4. Time Domain

if **$v(t) = V_p \cos(\omega t)$** then we have the phasor diagram shown in Fig. 5 below.

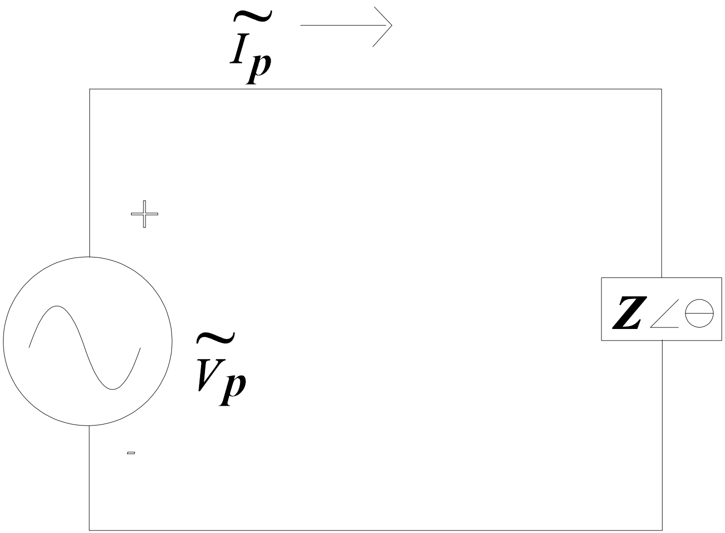


Figure 5 Phasor Diagram

where

$$\tilde{I}_p = \frac{\tilde{V}_p}{\tilde{Z}} = \frac{V_p}{Z}\angle -\theta \qquad\qquad i(t) = \frac{V_p}{Z}\cos(\omega t - \theta) \qquad \textbf{(2)}$$

Now the question is: "If we know the peak values of the voltage and current, and the load impedance, how much average power is delivered to the load?" Since we may write

$$p(t) = v(t) \cdot i(t) = V_p \cos(\omega t) \cdot I_p \cos(\omega t - \theta) \tag{3}$$

and since

$$\cos(a) \cdot \cos(b) = \frac{1}{2} \cdot (\cos(a+b) + \cos(a-b)) \tag{4}$$

we have

$$p(t) = \frac{V_p I_p}{2} (\cos\theta + \cos(2\omega t - \theta)) \tag{5}$$

For example if $\tilde{V} = 110\sqrt{2}\cos(\omega t)$ and $\tilde{I} = 50\sqrt{2}\cos(\omega t)$ as shown in Fig. 6 below, (i.e. the voltage and the current are in phase)

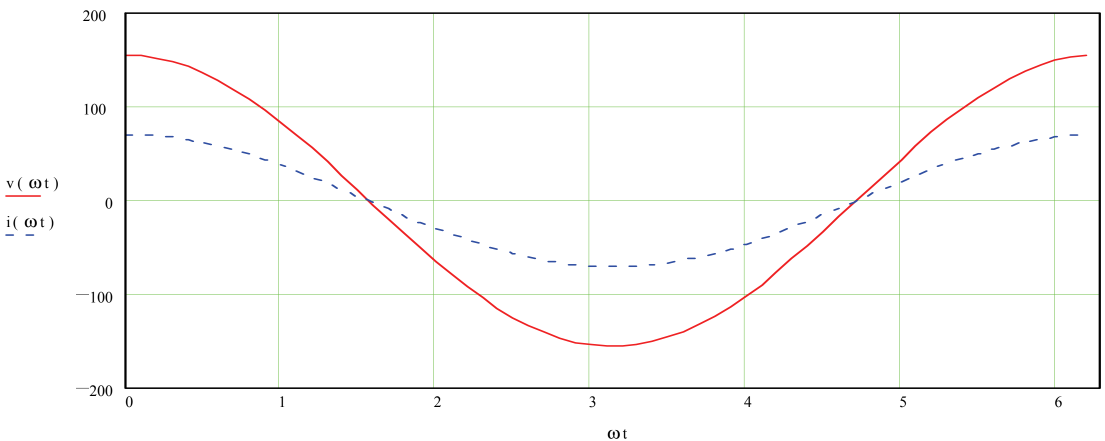


Figure 6 Voltage and Current Functions for θ = 0

then p(ωt) is as shown below in Fig. 7.

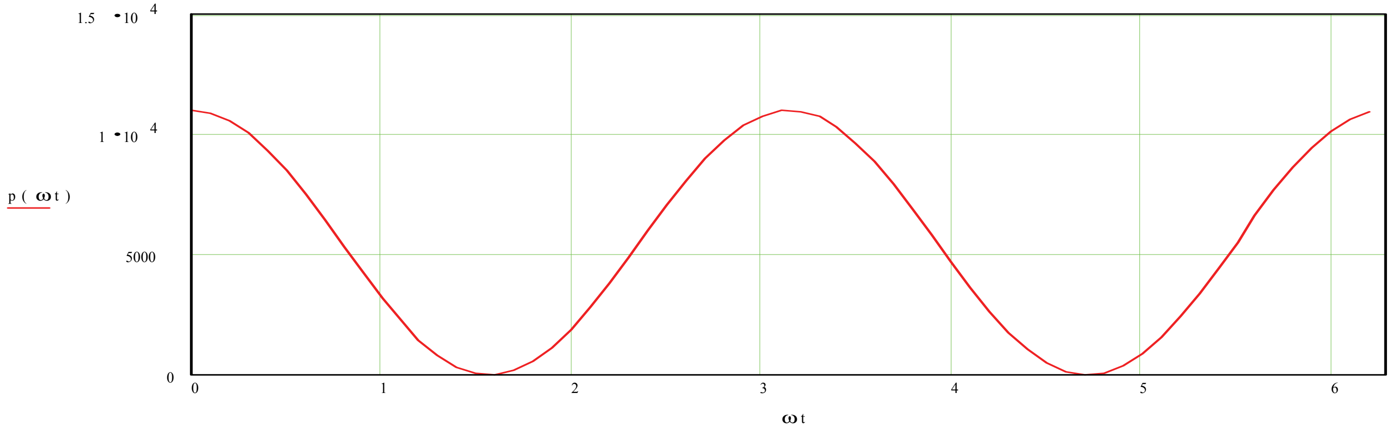


Figure 7 Power function for θ = 0

Notice that in Fig. 7 the frequency of the power as a function of time is twice the frequency of either the voltage or the current functions, and that the power is always positive so long as the current is in phase with the voltage. The average power of the function in Fig. 7 is

$$P_{AVE} = \frac{1}{T}\int_{0}^{2\pi} v(\omega\ t)\ i(\omega\ t)\ d\ \omega t \tag{6}$$

which calculated numerically on a point by point basis in Maple is P_{AVE} = 5.5 kW

If $\tilde{V} = 110\sqrt{2}\cos(\omega t)$ and $\tilde{I} = 50\sqrt{2}\cos(\omega t - 30°)$ as shown in Fig. 8 below,
(i.e. the voltage and current are 30° out of phase)

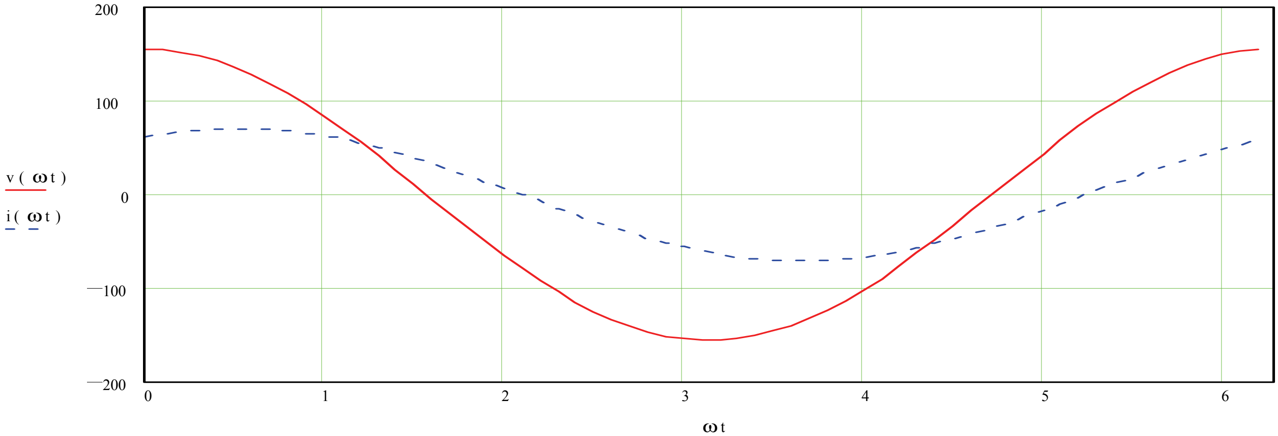


Figure 8 Voltage and Current Functions for θ = 30°

then p(ωt) is as shown below in Fig. 9.

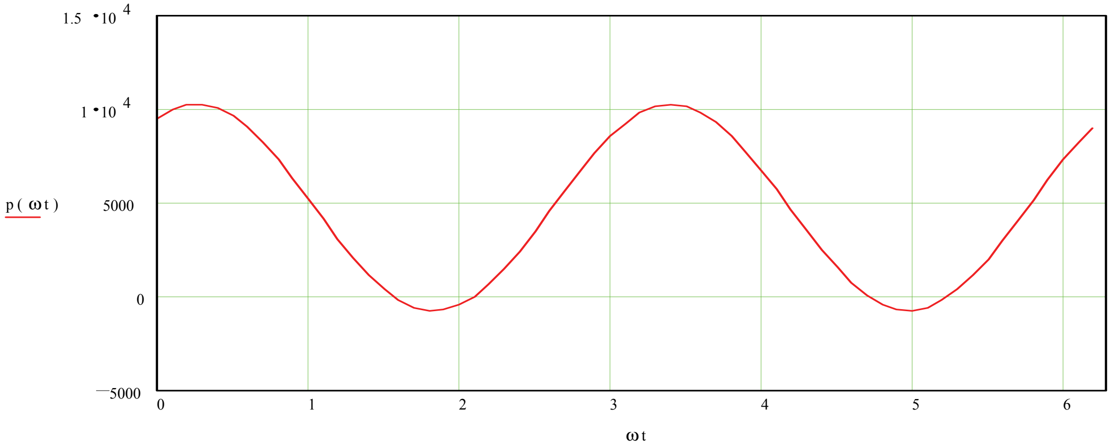


Figure 9 Power Function for $\theta = 30°$

Notice that for the case where the current is not in phase with the voltage, the power is sometimes positive and sometimes negative. The significance of the sign of the power is that when the power is positive, energy is being transferred from the source to the load, and when the power is negative energy is being transferred from the load to the source. The average power of the function of Fig. 9 is $P_{AVE} = 4.8$ kW, which is smaller than the power for the in-phase case, since the power is sometimes negative.

If $\tilde{V} = 110\sqrt{2}\cos(\omega t)$ and $\tilde{I} = 50\sqrt{2}\cos(\omega t - 60°)$ as shown in Fig. 10 below (i.e. the voltage and current are 60° out of phase), then p(ωt) is as shown below in Fig. 11.

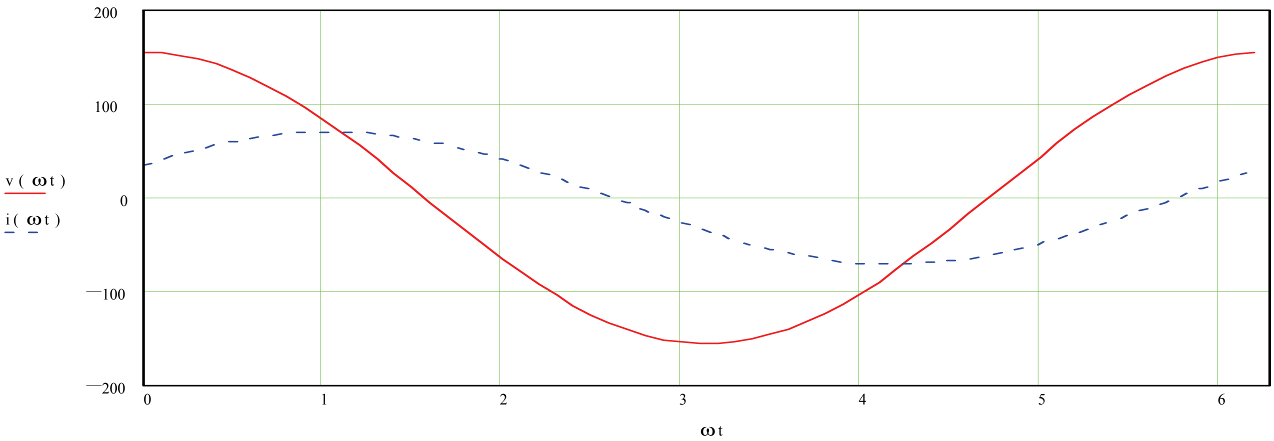


Figure 10 Voltage and Current Functions for θ = 60°

The average power in the function of Fig. 11 is P_{AVE} = 2.8 kW. This is smaller than the value for Fig. 9, since the function of Fig. 11 is negative for a longer part of the period.

Finally if we let $\tilde{V} = 110\sqrt{2}\cos(\omega t)$ and $\tilde{I} = 50\sqrt{2}\cos(\omega t - 90°)$ as shown in Fig. 12 below, (i.e. the voltage and current are 90° out of phase), then p(ωt) is as shown below in Fig. 13.

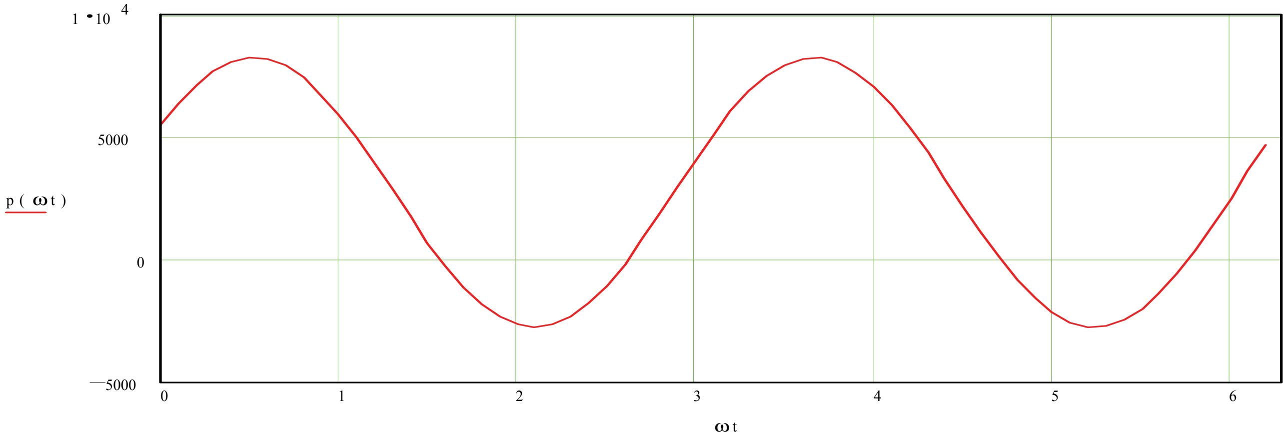


Figure 11 Power Function for $\theta = 60°$

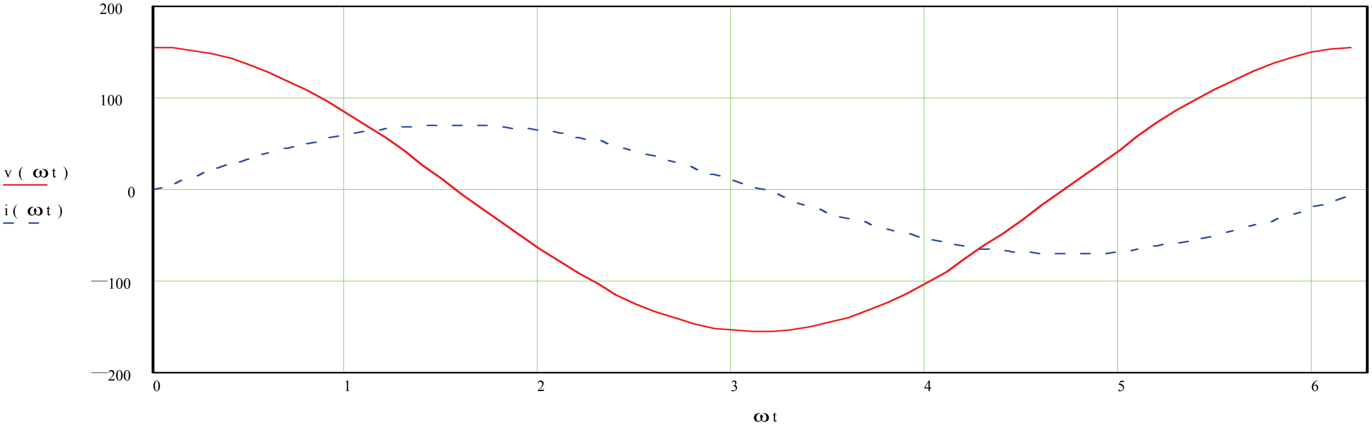


Figure 12 Voltage and Current Functions for $\theta = 90°$

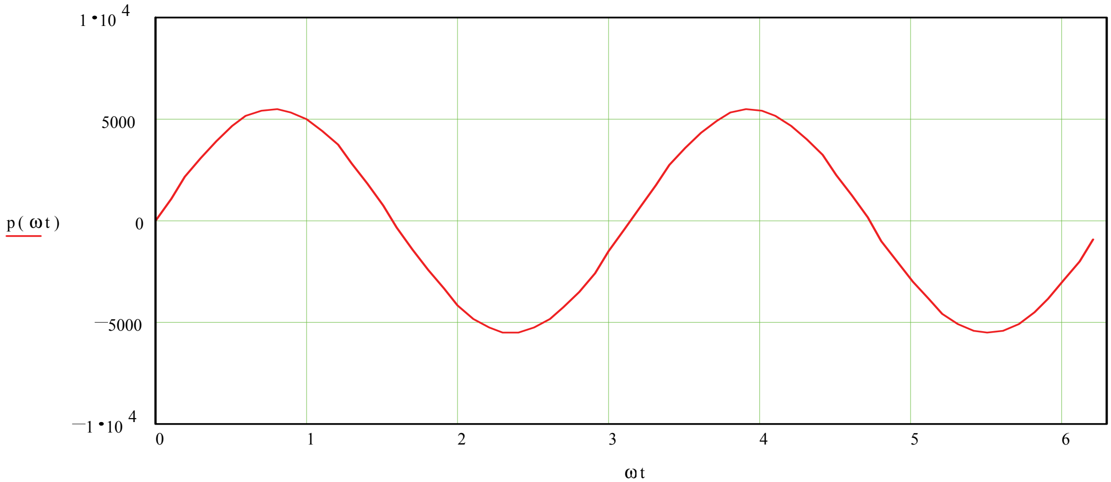


Figure 13 Power Function for θ = 90°

Notice that the average power of the function in Fig. 13 is zero, since the function is positive for the same amount of time that it is negative. We describe the dependence of the average power on θ, the angle between the voltage and the current, by decomposing the power function into two components: the power that flows unidirectionaly from the source to the load, and the power that is circulating back and forth symmetrically between the source and the load. To get expressions for these components, we apply

$$\cos(a \pm b) = \cos(a) \cdot \cos(b) \mp \sin(a) \cdot \sin(b)$$

to (5) [repeated below for convenience]

$$p(t) = \frac{V_p I_p}{2} (\cos\theta + \cos(2\omega t - \theta))$$

from which we obtain

$$p(t) = \frac{V_p I_p}{2} \cdot \left(\cos(\theta) + \cos(\theta) \cdot \cos(2\omega t) - \sin(-\theta) \cdot \sin(2\omega t)\right) \quad \textbf{(7)}$$

which can be rearranged to give

$$p(t) = \frac{V_p I_p}{2} \cos\theta \;(1 + \cos 2\omega t) + \frac{V_p I_p}{2} \sin\theta \sin(2\omega t) \quad \textbf{(8)}$$

If we define
Then

$$P = \frac{V_p I_p}{2} \cos(\theta) \qquad \text{and} \qquad Q = \frac{V_p I_p}{2} \sin(\theta) \quad \textbf{(9)}$$

$$p(t) = P\;(1 + \cos(2\omega t)) + Q\;\sin(2\omega t) \quad \textbf{(10)}$$

and

$$P_{AVE} = \frac{1}{T}\int_0^T p(t)\,dt = \frac{1}{T}\int_0^T P(1 + \cos(2\omega t)) + Q\sin(2\omega t)\,dt = \frac{1}{T}(P)(T) = P \quad \textbf{(11)}$$

P is called the real power (with units of Watts) and represents the AVERAGE power transferred by the source to the load. (There is no such thing as RMS power.) Q is called the reactive power (with units VAR) and represents the amplitude of the power that oscillates back and forth between the source and the load.

We define the complex power $\tilde{S} = P + jQ = S\angle\theta$ **(12)**

and the power factor $P.F. = \cos(\theta)$ where

$$S = \sqrt{P^2 + Q^2} = \frac{V_p \cdot I_p}{2} \qquad \theta = \tan^{-1}\frac{Q}{P} \tag{13}$$

Lagging power factor $\Rightarrow$ Inductive Load, $\theta \rangle$ 0, current lags voltage.

Leading power factor $\Rightarrow$ Capacitive Load, $\theta \langle$ 0, current leads voltage.

Power factor =1 (unity power factor)$\Rightarrow$ Resistive load, θ = 0, current and voltage in phase.

We define impedance in the usual way, noting that the impedance angle and the power angle are the same.

$$\tilde{Z} = \frac{V_p}{I_p} \angle\theta$$

The "Power Triangle" in Fig. 14 to the right, is a convenient way to remember the vector relationship among P, Q, S, and θ, as well as their units. If we express magnitudes in RMS then

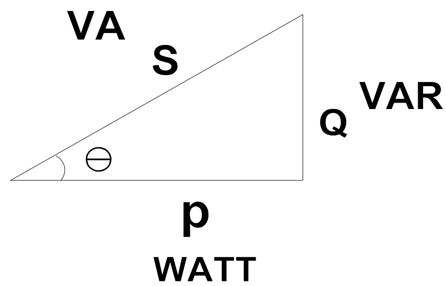


Figure 14 Power Triangle

$$v(t) = V_p \cos(\omega t) \rightarrow \quad V \angle 0 \quad \text{where} \quad V = \frac{V_p}{\sqrt{2}}$$

$$i(t) = I_p \cos(\omega t - \theta) \rightarrow \quad I \angle \text{-}\theta \quad \text{where} \quad I = \frac{I_p}{\sqrt{2}}$$

and $P = V\ I\ cos(\theta)$ $\quad Q = V\ I\ sin(\theta)$ $\quad$ and $\quad S = V\ I$ **(14)**

The fundamental power formula for AC power is $\tilde{S} = \tilde{V}\,\tilde{I}^*$ **(15)**

where the superscript * denotes complex conjugation.

If $\tilde{Z} = Z \angle \theta = R + JX$, then since $\tilde{V} = \tilde{Z}\,\tilde{I}$

we have $P = I^2 R \qquad Q = I^2 X$ or

$$P = \frac{V^2 R}{R^2 + X^2} \qquad Q = \frac{V^2 X}{R^2 + X^2} \qquad \textbf{(16)}$$

Per-Unit System

Now as mentioned above, the units of power are watts. However, in power system analysis, the units are normalized to a common voltage base V_{BASE} and complex power base S_{BASE}. So powers in per-unit would be

$$S_{Per-Unit} = \frac{S}{S_{BASE}} \qquad P_{Per-Unit} = \frac{P}{S_{BASE}} \qquad Q_{Per-Unit} = \frac{Q}{S_{BASE}} \tag{17}$$

voltages in per-unit would be

$$V_{per-unit} = \frac{V}{V_{Base}}$$

We define the current base and the impedance base as

$$I_{BASE} = \frac{S_{BASE}}{V_{BASE}} \qquad Z_{BASE} = \frac{S_{BASE}}{I_{BASE}^2} \tag{18}$$

The per-unit currents are given as

$$I_{Per-Unit} = \frac{I}{I_{BASE}} \tag{19}$$

and per-unit impedances are given as

$$R_{Per-Unit} = \frac{R}{Z_{BASE}} \qquad X_{Per-Unit} = \frac{X}{Z_{BASE}} \tag{20}$$

Thus the "units of power" in Fig. 2 are per-unit watts. Power engineers have been using the per-unit system as a matter of convenience for many years. (Transformers are handled particularly easily in the per-unit system.) However, we shall use the per-unit system for our analysis because the normalization makes the numerical routines much more stable.

Bus Types

A general system bus showing all the possible connections that might be made to it, (not all buses have all these

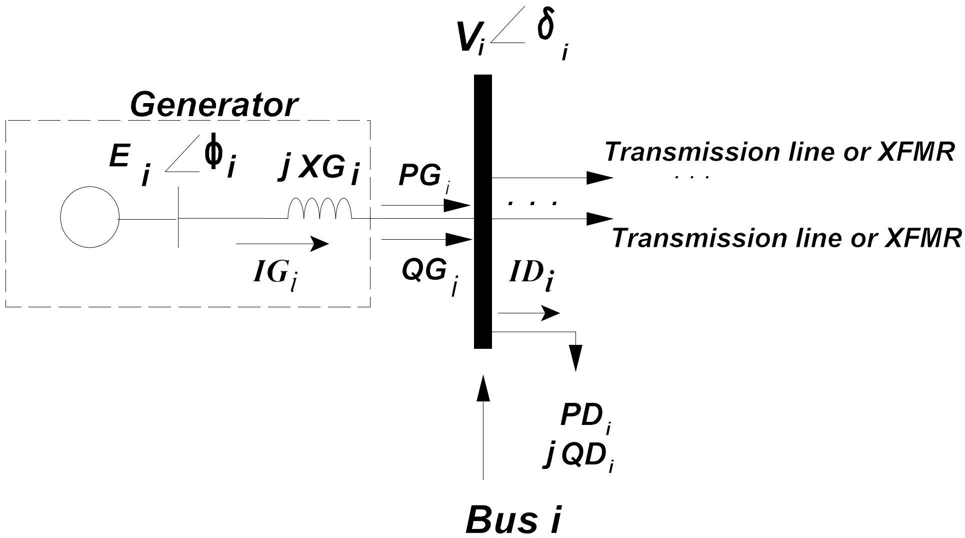


Figure 15 General Bus

connections) is shown in Fig. 15.

The buses can be divided into those that have generators connected to them , and those that do not. The buses that are not connected to generators are called “load buses,” and those that are connected to generators (even if the bus has a load connected to it in addition to the generator) are called “voltage-controlled buses.” These buses are so named because the generator excitation current is automatically and continuously adjusted to keep the magnitude of the bus voltage of the bus to which the generator is connected at some prescribed value. In addition, the steam-valve to the generator turbine is adjusted to keep the real power output at its prescribed value. It is the constant power nature of the loads, and the power (not current) control of the generators that turn what would otherwise be a linear circuit problem into a nonlinear problem that must be analyzed numerically.
Then if bus i is a VC (voltage controlled) bus, V_i and $P_{g\,i}$ are assumed known but $Q_{g\,i}$ (see AC Power above) is not known and depends on the network that is connected to bus i. So for a VC bus V_i and $P_{g\,i}$ are known parameters of the problem and $Q_{g\,i}$ is one of the unknowns for which we are trying to solve; δ_i is the other unknown. (The current is never used as a variable in a power flow problem.) If Bus i is a load bus then $P_{d\,i}$ (the load or demand real power) and $Q_{d\,i}$ are assumed to be known parameters of the system, and V_i and δ_i are unknown variables. Now in any phasor problem, any one angle can be chosen arbitrarily (the value of the reference angle is usually chosen to be zero), and in the power flow problem one of the generator voltages is chosen for this purpose, and the machine that is so chosen is referred to as the slack generator. The bus to which the slack generator is connected is called the slack bus. The reasoning behind the choice of this name is detailed below.
With all the demands $P_{d\,i}$ and $Q_{d\,i}$ specified, the buses, can be divided into three classes depending on the unknowns associated with each particular bus as follows:

Load Buses - these buses are not connected to generators; the unknowns are V_i and δ_i.

VC Buses - these voltage controlled buses are those connected to generators whose real power is specified; the unknowns are $Q_{g\,i}$ and δ_i.

The Slack Bus - one bus is selected as the slack bus and its real power is allowed to take on whatever value is required. The slack bus unknowns are $P_{g\,s}$ and $Q_{g\,s}$; V_s and δ_s are prescribed values at the slack bus.

Bus Numbering

To facilitate the creation of the many summations necessary to solve our problem, we number the buses in a very specific order as shown in Table 1 below.

Table 1. Bus Numbering

Bus 1 through bus $(n - \ell)$ are the load buses.
Bus $(n - \ell) + 1$ through bus $(n - 1)$ are the VC buses
Bus n is the slack bus.

From the definition of the buses and the above numbering conventions, we know that P_1 through P_{n-1} are prescribed, and that $V_{n-\ell+1}$ through V_n are prescribed. The power flow problem unknowns are, therefore, as shown in Table 2 below.

Table 2. Power flow unknowns

$\delta_1 , \ldots , \delta_{n-1}$	$n - 1$ unknowns (all voltage angles except at the slack bus)
$V_1 , \ldots , V_{n-\ell}$	$n-\ell$ unknowns (voltage magnitudes at load buses)
$Q_{n-\ell+1} , \ldots , Qn$	ℓ unknowns (Q_g at VC buses and at slack bus)
P_n	1 unknown (P_g at slack bus)

For the system of Fig. 2, $n = 5$, $\ell = 3$, and according to the numbering scheme of Table 1, buses one and two are load buses, buses three and four are VC buses, and bus five is the slack bus.

Node Voltage Analysis

Consider the system of Fig. 2, where we have replaced the transmission line powers with the transmission line admittances, and have shown the result in Fig. 16 below.

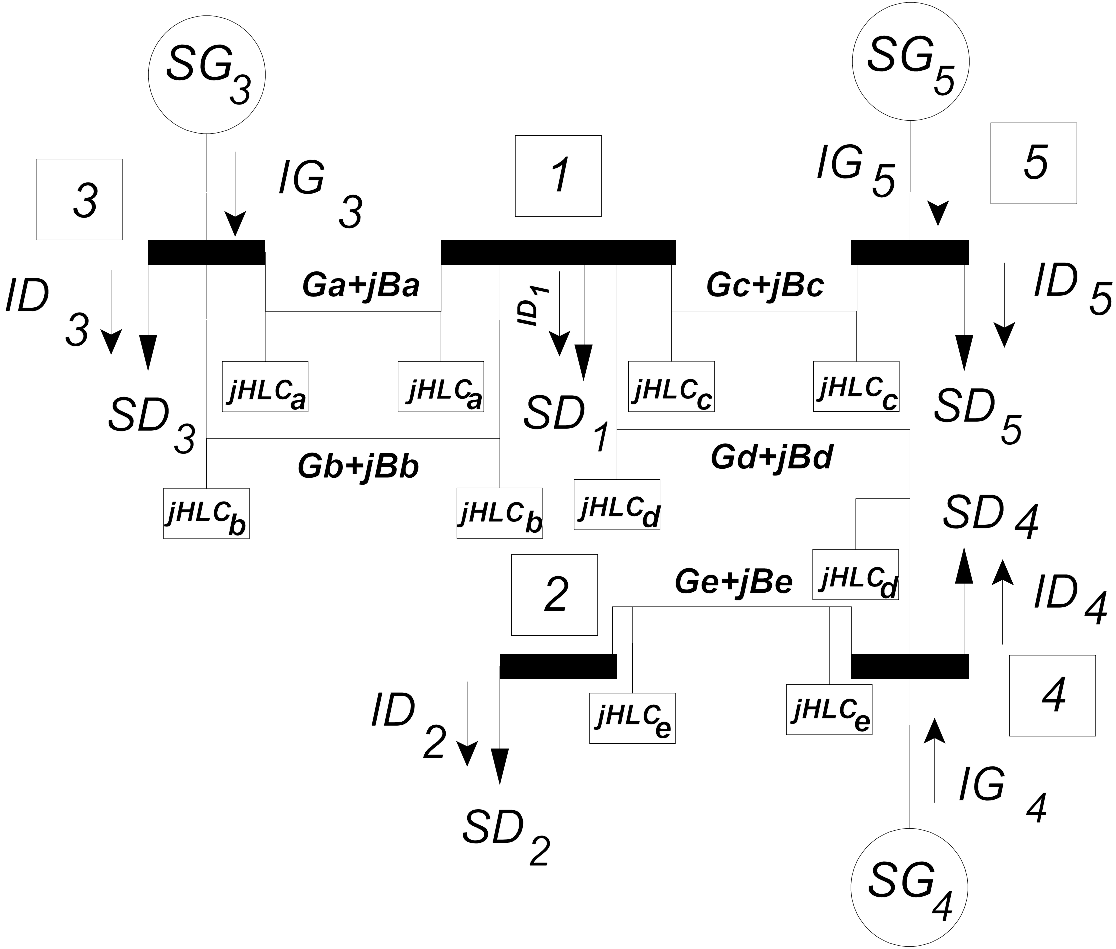


Figure 16 Test System with Transmission Line Admittances

The Node Voltage equations for the system of Fig. 16 are as follows:

$$\tilde{ID}_1 + \left(\tilde{V}_1 - \tilde{V}_3\right)\left(G_a + jB_a\right) + \left(\tilde{V}_1 - \tilde{V}_3\right)\left(G_b + jB_b\right) + \left(\tilde{V}_1 - \tilde{V}_4\right)\left(G_d + jB_d\right) \\ + \left(\tilde{V}_1 - \tilde{V}_5\right)\left(G_c + jB_c\right) + j\tilde{V}_1 HLC_a + j\tilde{V}_1 HLC_b + j\tilde{V}_1 HLC_c + j\tilde{V}_1 HLC_d \\ = 0 \tag{21a}$$

$$\tilde{ID}_2 + \left(\tilde{V}_2 - \tilde{V}_4\right)\left(G_e + jB_e\right) + j\tilde{V}_2 HLC_e = 0 \tag{21b}$$

$$-\tilde{IG}_3 + \tilde{ID}_3 + \left(\tilde{V}_3 - \tilde{V}_1\right)\left(G_a + jB_a\right) + \left(\tilde{V}_3 - \tilde{V}_1\right)\left(G_b + jB_b\right) + j\tilde{V}_3 HLC_a + j\tilde{V}_3 HLC_b = 0 \tag{21c}$$

$$-\tilde{IG}_4 + \tilde{ID}_4 + \left(\tilde{V}_4 - \tilde{V}_2\right)\left(G_e + jB_e\right) + \left(\tilde{V}_4 - \tilde{V}_1\right)\left(G_d + jB_d\right) + j\tilde{V}_4 HLC_d + j\tilde{V}_4 HLC_e = 0 \tag{21d}$$

$$-\tilde{IG}_5 + \tilde{ID}_5 + \left(\tilde{V}_5 - \tilde{V}_1\right)\left(G_c + jB_c\right) + j\tilde{V}_5 HLC_c = 0 \tag{21e}$$

Equations (21) can be rearranged in the usual manner to produce:

$$\tilde{V}_1\left[\left(G_a + jB_a\right) + \left(G_b + jB_b\right) + \left(G_c + jB_c\right) + \left(G_d + jB_d\right) + j\left(HLC_a + HLC_b + HLC_c + HLC_d\right)\right] \\ - \tilde{V}_3\left[\left(G_a + jB_a\right) + \left(G_b + jB_b\right)\right] \\ - \tilde{V}_4\left(G_e + jB_e\right) - \tilde{V}_5\left[\left(G_c + jB_c\right)\right] \\ = 0 - \tilde{ID}_1 \tag{22a}$$

$$\tilde{V}_2\left[\left(G_e + jB_e\right) + jHLC_e\right] - \tilde{V}_4\left[\left(G_e + jB_e\right)\right] = 0 - \tilde{ID}_2 \tag{22b}$$

$$\begin{aligned}\tilde{V}_3\left[\left(G_a+jB_a\right)+\left(G_b+jB_b\right)+j\left(HLC_a+HLC_b\right)\right]& \\ -\tilde{V}_1\left[\left(G_a+jB_a\right)+\left(G_b+jB_b\right)\right]& \\ = I\tilde{G}_3-I\tilde{D}_3&\end{aligned} \tag{22c}$$

$$\begin{aligned}\tilde{V}_4\left[\left(G_e+jB_e\right)+\left(G_d+jB_d\right)+j\left(HLC_e+HLC_d\right)\right]& \\ -\tilde{V}_2\left[\left(G_e+jB_e\right)\right]-\tilde{V}_1\left[\left(G_d+jB_d\right)\right]& \\ = I\tilde{G}_4-I\tilde{D}_4&\end{aligned} \tag{22d}$$

$$\tilde{V}_5\left[\left(G_c+jB_c\right)+jHLC_c\right]-\tilde{V}_1\left[\left(G_c+jB_c\right)\right] = I\tilde{G}_5-I\tilde{D}_5 \tag{22e}$$

Note in equations (22) that the term multiplying each voltage is an admittance (or sum of admittances), so that equations (22) can be written as:

$$\begin{bmatrix} 0-I\tilde{D}_1 \\ 0-I\tilde{D}_2 \\ I\tilde{G}_3-I\tilde{D}_3 \\ I\tilde{G}_4-I\tilde{D}_4 \\ I\tilde{G}_5-I\tilde{D}_5 \end{bmatrix} = \begin{bmatrix} \tilde{Y}_{11} & \tilde{Y}_{12} & \tilde{Y}_{13} & \tilde{Y}_{14} & \tilde{Y}_{15} \\ \tilde{Y}_{21} & \tilde{Y}_{22} & \tilde{Y}_{23} & \tilde{Y}_{24} & \tilde{Y}_{25} \\ \tilde{Y}_{31} & \tilde{Y}_{32} & \tilde{Y}_{33} & \tilde{Y}_{34} & \tilde{Y}_{35} \\ \tilde{Y}_{41} & \tilde{Y}_{42} & \tilde{Y}_{43} & \tilde{Y}_{44} & \tilde{Y}_{45} \\ \tilde{Y}_{51} & \tilde{Y}_{52} & \tilde{Y}_{53} & \tilde{Y}_{54} & \tilde{Y}_{55} \end{bmatrix} \begin{bmatrix} \tilde{V}_1 \\ \tilde{V}_2 \\ \tilde{V}_3 \\ \tilde{V}_4 \\ \tilde{V}_5 \end{bmatrix} \tag{23}$$

Now what makes (23) particularly useful is that the admittance matrix (usually called Y_{BUS}) can be relatively easily constructed. Examination of equations (22) shows that $\tilde{Y}_{11}$ is simply the sum of all the admittances connected to bus 1 (this is why the transmission line HLCs are given in terms of admittances rather than impedances), $\tilde{Y}_{22}$ is the sum of all admittances connected to bus 2, and so on for bus 3, bus 4, and bus 5. In general then, we have

$$\tilde{Y}_{ii} = \sum \text{Admittances on Bus i} \tag{24}$$

Further examination of (22a) reveals that in the first equation, the coefficient of $\tilde{V}_3$ is simply the negative of the sum of the admittances connected between bus 1 and bus 3, the coefficient of $\tilde{V}_4$ is the negative of the admittance connected between bus 1 and bus 4, and the coefficient of $\tilde{V}_5$ is the negative of the admittance connected between bus 1 and bus 5. Examination of (22d) reveals that in the fourth equation, the coefficient of $\tilde{V}_2$ is simply the negative of the admittance connected between bus 4 and bus 2, and the coefficient of $\tilde{V}_5$ is the negative of the admittance connected between bus 4 and bus 5. In general we have

$$\tilde{Y}_{ij} = -\sum \text{Admittances between Bus i and Bus j} \qquad i \neq j \tag{25}$$

We shall discuss the details of constructing Y_{BUS} later on when we describe the typical data files that contain the transmission line data.

The definition of matrix multiplication permits us to rewrite the node voltage equations (23) as

$$\sum_{k=1}^{n} \tilde{Y}_{ik} \tilde{V}_k = \tilde{I}_{gi} - \tilde{I}_{di} \qquad (i = 1, 2, \ldots, n) \tag{26}$$

Note that in (26) $\vec{V}$ is a vector of phasors, where $\tilde{V}_k$ is the kth phasor element of the vector, and $\vec{Y}$. is a matrix of phasors where $\tilde{Y}_{ik}$ is the (i,k)th element of the matrix. Now so far we have a linear problem since all the transmission line admittances are linear. Unfortunately we do not have constant current generators (as in a circuits problem), but constant power generators. To get an expression for the generator currents in terms of the generator powers, we invoke the fundamental power formula (15). The ith generator complex power is given as

$$\tilde{S}_{gi} = P_{gi} + j\, Q_{gi} = \tilde{V}_i \tilde{I}^*_{gi} \tag{27}$$

Note that in (27) $\vec{P}_g$.is a vector of real numbers (the generator powers), and P_{gi} is the ith element of the vector If we take the complex conjugate of both sides of the right-most equality in (27), then divide both sides by $\tilde{V}_i^*$ we have

$$\tilde{I}_{gi} = \frac{P_{gi} - jQ_{gi}}{\tilde{V}_i^*} \tag{28}$$

A similar argument for the ith load (demand) power gives

$$\tilde{I}_{di} = \frac{P_{di} - jQ_{di}}{\tilde{V}_i^*} \tag{29}$$

Substituting (28) and (29) into(26) gives

$$\sum_{k=1}^{n} \tilde{Y}_{ik} \tilde{V}_k = \frac{(P_{gi} - jQ_{gi}) - (P_{di} - jQ_{di})}{\tilde{V}_i^*} \qquad i = 1,\ldots,n \tag{30}$$

If we define

$$P_i = P_{gi} - P_{di} \qquad\qquad Q_i = Q_{gi} - Q_{di} \tag{31}$$

we may substitute (31) into (30), and multiplying both sides by $\tilde{V}_i^*$ we produce

$$\tilde{V}_i^* \sum_{k=1}^{n} \tilde{Y}_{ik} \tilde{V}_k = P_i - jQ_i \qquad i = 1,\ldots,n \tag{32}$$

Since the voltages and admittances can be written in polar form as

$$\tilde{V}_i = |\tilde{V}_i| \angle\delta_i = V_i \angle\delta_i \qquad\qquad \tilde{Y}_{ij} = |\tilde{Y}_{ij}| \angle\theta_{ij} = Y_{ij} \angle\theta_{ij} \tag{33}$$

and recalling the manner in which polar complex numbers multiply

$$A \angle\alpha \cdot B \angle\beta = A \cdot B \angle(\alpha + \beta)$$

we may rewrite (32) as

$$P_i - jQ_i = \sum_{k=1}^{n} \left[V_i \, Y_{ik} \, V_k \angle(-\delta_i + \theta_{ik} + \delta_k) \right] \qquad i = 1,\ldots,n \tag{34}$$

Finally recalling Euler's formula

$$r \angle\theta = r\cos(\theta) + j\, r\sin(\theta)$$

we may write

$$P_i - jQ_i = \sum_{k=1}^{n} \begin{bmatrix} V_i \, Y_{ik} \, V_k \cos(\theta_{ik} - \delta_i + \delta_k) \\ + jV_i \, Y_{ik} \, V_k \sin(\theta_{ik} - \delta_i + \delta_k) \end{bmatrix} \qquad i = 1,\ldots,n \tag{35}$$

Since the real part of the left hand side of a complex equation must equal the real part of the right hand side of the equation, and the imaginary part of the left hand side of a complex equation must equal the imaginary part of the right hand side of the equation, the n complex equations (35) may be written as 2n real equations as follows:

$$\begin{aligned} P_i &= \sum_{k=1}^{n} V_i \, Y_{ik} \, V_k \cos(\theta_{ik} - \delta_i + \delta_k) \\ Q_i &= -\sum_{k=1}^{n} V_i \, Y_{ik} \, V_k \sin(\theta_{ik} - \delta_i + \delta_k) \end{aligned} \qquad i = 1,\ldots,n \tag{36}$$

The 2n equations (36) are called the power flow (or load flow) equations and they contain 2n unknowns: (n-ℓ) load bus voltages, (n-ℓ) load bus angles, ℓ generator reactive powers (Qs),
(ℓ-1) VC bus voltage angles, and the slack generator real power. (See Table 2.) The solutions of (36) (if they exist),

constitute the solution of the power flow problem.

Line Data and the Calculation of Y_{BUS}

The transmission line data is contained in a space delimited ASCII text file with the following format: (there are NLS such lines in the file.)

Table 3. Transmission Line Data File Format

FROM	TO	R	X	HLC
Bus# i	Bus# j	Riji	Xij	HLC
Bus# i	Bus# j	Rij	Xij	HLC
	...			
Bus# i	Bus# j	Rij	Xij	HLC

FROM is the number of the bus where the transmission line originates, TO is the number of the bus where the transmission line terminates, R is the series resistance of the transmission line connected between bus # FROM and bus # TO, X is the series inductive reactance of the transmission line connected between bus # FROM and bus # TO, and HLC is the capacitive suseptance (half line charging capacitive suseptance) of the transmission line as seen at bus # FROM and at bus # TO. The resistances and inductive reactances are given in per-unit ohms and the capacitive suseptance is given in per-unit seimens (mhos). The n by n complex matrix Y_{BUS} is realized as two real n by n matrices : Y and Θ. Y and Θ are constructed by first using (25) and (26) to form n by n real matrix G which contains the real parts of the admittances, and n by n real matrix B which contains the imaginary parts of the admittances. (Recall that the real part of the left hand side of a complex equation is equal to the real part of the right hand side, and the imaginary part of the left hand side of a complex equation is equal to the imaginary part of the right hand side.) Y and Θ are then formed from G and B as follows

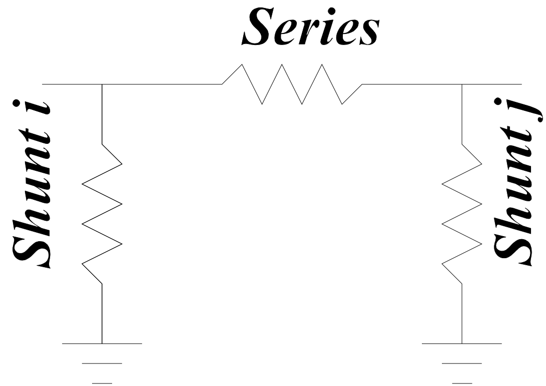
$$Y_{ij} = |G_{ij} + jB_{ij}| \qquad \Theta_{ij} = \angle\left(G_{ij} + jB_{ij}\right) \qquad \begin{matrix} i = 1,\ldots,n \\ j = 1,\ldots,n \end{matrix} \tag{37}$$

The transmission line data is defined as an NLS by NLS matrix LD in Matlab in an m-file we shall call lfdat.m.

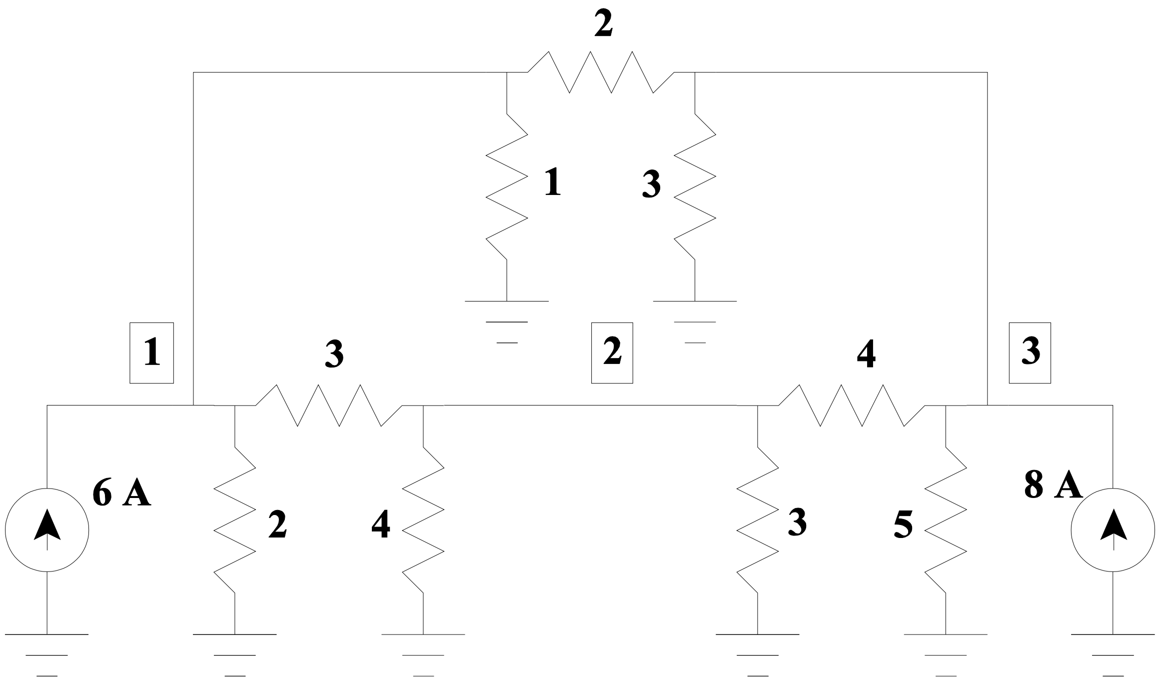
Lab Demo #3 YBUS

Problem 1.

We wish to analyze a structure composed of sub-elements with the following form.



The i and j refer to the nodes to which the device is connected. Suppose we wish to analyze a circuit composed of three such devices and two current sources as shown below.



We begin by writing the Node-Voltage equations as follows. (The boxed numbers are the node numbers.)

$$\frac{V_1}{1} + \frac{V_1}{2} + \frac{V_1 - V_2}{3} + \frac{V_1 - V_3}{2} = 6$$

$$\frac{V_2}{4} + \frac{V_2}{3} + \frac{V_2 - V_1}{3} + \frac{V_2 - V_3}{4} = 0$$

$$\frac{V_3}{5} + \frac{V_3}{3} + \frac{V_3 - V_2}{4} + \frac{V_3 - V_1}{2} = 8$$

We may rewrite the above equations in matrix form as

$$\begin{bmatrix} \frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{2} & -\frac{1}{3} & -\frac{1}{2} \\ -\frac{1}{3} & \frac{1}{4}+\frac{1}{3}+\frac{1}{3}+\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{2} & -\frac{1}{4} & \frac{1}{5}+\frac{1}{3}+\frac{1}{4}+\frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 8 \end{bmatrix}$$

Let us define a matrix Y and vectors V and S so that

$$[Y] \cdot [V] = [S]$$

which when expanded by terms becomes

$$\begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 8 \end{bmatrix}$$

A close examination of the above [Y] matrices (the [Y] matrix is typically called YBUS) shows that the following rules can be formulated for calculating the individual elements of YBUS

$$Y_{ii} = \sum_{k=1}^{n} \text{All Ys connected to Bus i}$$

$$Y_{ij} = -\sum_{k=1}^{n} \text{All Ys connected between Bus i and Bus j}$$

Now the data describing the resistances associated with our three devices is given in a space delimited ASCII table (with one line per device) as

FROM	TO	SERIES RES	SHUNT i RES	SHUNT j RES
1	2	3	2	4
2	3	4	3	5
1	3	2	1	3

Our task is to create the [Y] matrix from the above data table and use [Y] to solve the Node-Voltage equations.

The following Matlab m-file does the job.

```
% File ldem3_1.m
```

```
%
% Set the number of Lines (NLS) to 3 ( There is one line in %
% the data file for each transmission line.)%
%
NLS=3;
%
% Set the number of nodes (n) to 3 %
% It is a coincidence that NLS happens to be the same as n %
% in this particular case.%
%
n=3;
%
% Define the column vector S %
%(S=[6 0 8] would define a row vector which is not what %
% we need.)%
%
S=[6 0 8]';
%
%
% (S = [ 6                    %
%        0                    %
%        8];or %
% S = [ 6; 0; 8] would work also)%
%
%Define the data matrix D%
%
D=[1 2 3 2 4 ;
   2 3 4 3 5 ;
   1 3 2 1 3];
```

```
% Initialize Y - This serves two purposes%
% 1. It sets the storage allocation for Y ahead of time greatly %
% speeding the execution of the m file and %
% 2. It sets all the elements of Y initially to zero %
%
Y=zeros(n);
%
% The following loop calculates Y from D and is the main point %
% of the discussion - this is the technique you will use to %
% calculate YBUS from the Line Data %
%
for k=1:NLS
i=D(k,1);
j=D(k,2);
ser=1/D(k,3);
shunti=1/D(k,4);
shuntj=1/D(k,5);
Y(i,i)=Y(i,i)+ser+shunti;
Y(j,j)=Y(j,j)+ser+shuntj;
Y(i,j)=-ser;
Y(j,i)=-ser;
end;
%
% Solve the Node-Voltage equations by inversion of the Y matrix %
%
V=inv(Y)*S;
%
% Alternatively, the equations can be solved by Gaussian %
% elimination with the following syntax %
VV=Y\S;
```

```
% Gaussian elimination is preferable for large problems like %
% the Load-Flow problem. %
```

The following matlab session shows how to run file ldem3_1, and how to display the results.

```
EDU» who
EDU» ldem3_1
EDU» who

Your variables are:

D          V          i          n          shuntj
NLS        VV         j          ser
S          Y          k          shunti

EDU» Y

Y =

    2.3333   -0.3333   -0.5000
   -0.3333    1.1667   -0.2500
   -0.5000   -0.2500    1.2833

EDU» V

V =

    4.9262
    3.2920
    8.7944
```

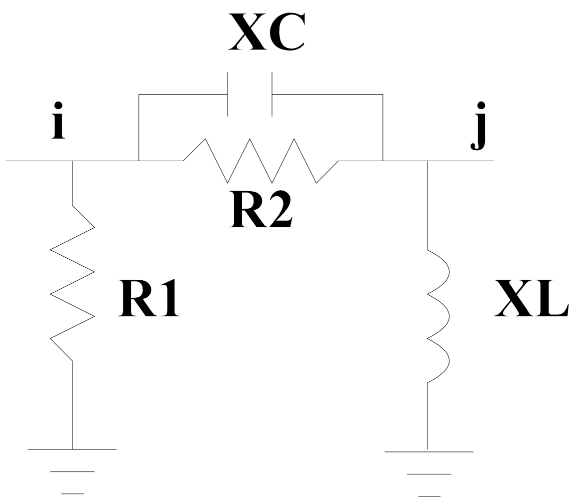
```
EDU» VV

VV =

    4.9262
    3.2920
    8.7944

EDU»
```

Problem 2.
Consider a circuit composed of structures like the following:



The circuit is formed according to the table below.

i	j	R1	R2	XC	XL
1	2	10	16	22	28
1	3	11	17	23	28
1	4	12	18	24	30
2	3	13	19	25	31
2	4	14	20	26	32
3	4	15	21	27	33

where

$$XL = \omega L \qquad\qquad XC = \frac{1}{\omega C}$$

Our task is to find the magnitude and angle of the admittance matrix for this circuit.

Since the impedance of a capacitor is $\frac{XC}{j}$ (a negative number), we have that the admittance of a capacitor is $\frac{j}{XC}$ (a positive number).

For computational purposes we realize the complex matrix $\vec{Y}$ as a real matrix Y consisting of the magnitudes of the admittances in $\vec{Y}$, and a real matrix Theta containing the angles of the admittances in $\vec{Y}$. We construct these matrices by first constructing real matrix G ,which contains the real parts of the admittances in $\vec{Y}$, and a real matrix B, which contains the imaginary parts of the impedances in $\vec{Y}$. Y and Theta are then constructed as in (37).

The matlab file that solves this problem is as follows.

```
%File        Ldem3_2.m
%
n = 4;
NLS = 6;
%
D = [ 1 2 10 16 22 28
      1 3 11 17 23 28
      1 4 12 18 24 30
      2 3 13 19 25 31
      2 4 14 20 26 32
      3 4 15 21 27 33];

p=sqrt(-1);

G     = zeros(n);
B     = zeros(n);
Y     = zeros(n);
Theta = zeros(n);

%

for k = 1:NLS
i =  D(k,1);
j =  D(k,2);
r1 = D(k,3);
r2 = D(k,4);
xc = D(k,5);
xL = D(k,6);
```

```
G(i,i) = G(i,i)+1/r1+1/r2;
G(j,j) = G(j,j)+1/r2;
G(i,j) = G(i,j)-1/r2;   % There might be more than one line    %
G(j,i) = G(j,i)-1/r2;   % connecting node i and node j         %
B(i,i) = B(i,i)+1/xc;
B(j,j) = B(j,j)+1/xc-1/xL;
B(i,j) = B(i,j)-1/xc;
B(j,i) = B(j,i)-1/xc;
end;

Y = abs(G+p*B);
Theta = angle(G+p*B);
```

Matlab session showing file ldem3_2 being run

```
EDU ldem3_2
EDU» G

G =

    0.4511   -0.0625   -0.0588   -0.0556
   -0.0625    0.3135   -0.0526   -0.0500
   -0.0588   -0.0526    0.2257   -0.0476
   -0.0556   -0.0500   -0.0476    0.1532


EDU» B

B =
```

```
    0.1306   -0.0455   -0.0435   -0.0417
   -0.0455    0.0882   -0.0400   -0.0385
   -0.0435   -0.0400    0.0525   -0.0370
   -0.0417   -0.0385   -0.0370    0.0223

EDU» Y

Y =

    0.4696    0.0773    0.0731    0.0694
    0.0773    0.3257    0.0661    0.0631
    0.0731    0.0661    0.2318    0.0603
    0.0694    0.0631    0.0603    0.1548

EDU» Theta


Theta =

    0.2818   -2.5128   -2.5051   -2.4981
   -2.5128    0.2743   -2.4917   -2.4859
   -2.5051   -2.4917    0.2287   -2.4805
   -2.4981   -2.4859   -2.4805    0.1444
```

Assignment #5 YBUS

Four transmission lines of the type described in Fig. 3 of the Electric Utility Power Flow section are connected according to the table below. Find YBUS

From	To	R	X	HLC
1	2	0.01008	0.0504	0.05125
1	3	0.00744	0.0372	0.03875
2	4	0.00744	0.0372	0.03875
3	4	0.01272	0.0636	0.06375

Bus Data

The bus data is contained in a space delimited ASCII text file with the following format: (there are n such lines in the file).

Table 4. Bus Data File Format

BUS	V	δ	Pg	Qg	Pd	Qd
BUS # i	V_i	δ_i	$P_{g\,i}$	$Q_{g\,i}$	$P_{d\,i}$	$Q_{d\,i}$
BUS # i	V_i	δ_i	$P_{g\,i}$	$Q_{g\,i}$	$P_{d\,i}$	$Q_{d\,i}$
		...				
BUS # i	V_i	δ_i	$P_{g\,i}$	$Q_{g\,i}$	$P_{d\,i}$	$Q_{d\,i}$

Bus # is the number of the bus in question, P_g is the scheduled (known) real power of the generator connected to the bus, Q_g is the unknown generator reactive power, P_d is the known demand real power, and Q_d is the known demand reactive power. V and δ are initial estimates for the voltage at the bus. Since we have chosen per-unit variables, the bus voltages are all close to 1 per-unit volts and since the voltage angles typically have relatively small magnitudes, we arbitrarily choose $1 \angle 0°$ as the initial estimate for all voltages. The bus data is defined as n x n matrix BD in Matlab which is used to define vectors V, δ, P and Q, where $P = P_g - P_d$ and $Q = Q_g - Q_d$. (This is all accomplished in lfdat.m)

The Solution of the Equilibrium Equations - Newton's Method

Newton's algorithm solves non-linear equations of the form

$$f(x) = 0$$

according to the following iterative procedure:

$$x^{\nu+1} = x^{\nu} - \frac{f(x^{\nu})}{\left(\dfrac{df(x^{\nu})}{dx^{\nu}}\right)} \tag{38}$$

where $x^{\nu+1}$ is the $(\nu + 1)^{th}$ estimate for x. (Superscript ν is the iteration counter - not a raising to the power ν.) The iterative process is terminated when f(x) is close enough to 0.

The algorithm can be extended to handle sets of non-linear equations of the form

$$\vec{f}(\vec{x}) = 0$$

where $\vec{f}$ is a vector of functions and $\vec{x}$ is a vector of unknowns as follows:

$$\vec{x}^{\nu+1} = \vec{x}^{\nu} - \left[J_{\vec{x}}\left(\vec{f}(\vec{x})\right)\right]^{-1}_{\vec{x} = \vec{x}^{\nu}} \cdot \vec{f}(\vec{x}^{\nu}) \tag{39}$$

where $\left[J_{\vec{x}}\left(\vec{f}(\vec{x})\right)\right]$ denotes the Jacobian matrix as defined below.

(We shall discuss the calculation of the Jacobian in the next section.)

$$\left[J_{\vec{x}}\left(\vec{f}(\vec{x})\right)\right] = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_2(x)}{\partial x_n} \\ & & \cdots & \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \cdots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix} \tag{40}$$

Rearranging (36) we have

$$\begin{aligned} f_{Ri} &= P_i - \sum_{k=1}^{n} V_i Y_{ik} V_k \cos(\theta_{ik} - \delta_i + \delta_k) \\ f_{Ii} &= Q_i + \sum_{k=1}^{n} V_i Y_{ik} V_k \sin(\theta_{ik} - \delta_i + \delta_k) \end{aligned} \qquad i = 1, \ldots, n \tag{41}$$

Combining all 2n equations given by (41) into a single vector equation, we have

$$\vec{f}(\vec{x}) = 0 \tag{42}$$

where

$$\vec{f} = \begin{bmatrix} f_{R1} \\ f_{R2} \\ \ldots \\ f_{Rn} \\ f_{I1} \\ f_{I2} \\ \ldots \\ f_{In} \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \ldots \\ \delta_{n-1} \\ P_n \\ V_1 \\ V_2 \\ \ldots \\ V_{n-\ell} \\ Q_{n-\ell+1} \\ \ldots \\ Q_n \end{bmatrix} \tag{43}$$

Extrinsic and intrinsic variables

Suppose we have a set of four equations in four unknowns as shown below.

Note that the variables x_3 and x_4 occur only in one equation each. Such variables are called extrinsic, whereas the variables that are found in all four equations are called intrinsic variables. The absence of these variables from the last 2 equations means that we do not have to solve these 4 equations simultaneously to obtain values for the four variables. Rather we can solve the last 2 equations in 2 unknowns for x_1 and x_2. Then we use the first equation to solve for x_3 and the second equation to solve for x_4.

$$x_1^2 + x_1 \cdot x_2 + 4 \cdot x_3 = 0$$

$$x_1 \cdot x_4 + \sin(x_2) + x_2 \cdot x_4 = 0$$

$$x_1 \cdot x_2^2 + \cos(x_1) + \sin(x_1 \cdot x_2) = 0$$

$$3 \cdot x_1 + 4 \cdot x_1 \cdot x_2 + \cos(x_2) = 0$$

Examining (41) and (43) we see that P_n , and $Q_{n-\ell+1}$ through Q_n , each appear in only one equation. We can eliminate these variables and the equations in which they appear from the system of equations we must solve simultaneously, since if we solve for $\delta_1, \ldots, \delta_n$ and $V_1, \ldots, V_{n-\ell}$ simultaneously (each of these variables can appear in all of the equations) we can then get P_n from f_{Rn} and $Q_{n-\ell+1}, \ldots, Q_n$ from, $f_{In-\ell+1} \ldots f_{In}$. The unknowns removed in this way are called explicit unknowns since they appear explicitly: i.e. in only one equation each. This removal of some variables also simplifies the calculation of the Jacobian, since derivatives need to be taken with respect to fewer variables. We apply Newton's method to to the reduced system g(y) = 0 $\vec{g}(\vec{y}) = 0$ where

$$\vec{g} = \begin{bmatrix} f_{R1} \\ f_{R2} \\ \ldots \\ f_{Rn-1} \\ f_{I1} \\ f_{I2} \\ \ldots \\ f_{In-\ell} \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \ldots \\ \delta_{n-1} \\ V_1 \\ V_2 \\ \ldots \\ V_{n-\ell} \end{bmatrix} \tag{44}$$

then calculate the explicit unknowns once the bus voltages are known.

The Jacobian matrix for virtually all physical power systems is very sparse (most of the entries in the matrix are zero). This sparsity is a result of the fact that $\vec{Y}$ is sparse since each bus in the system is not connected to every other bus in the system, but to only a few other buses. We can significantly reduce the storage requirements and execution time of the computer analysis by taking advantage of the sparsity of Y_{BUS} and not storing the zero elements. Instead we store only the non-zero elements and their location in the matrix, and work primarily with these non-zero elements. Since the inversion of a sparse matrix does not yield another sparse matrix, we seek a modification of Newton's method that does not involve inverting the Jacobian. This is accomplished by the following rearrangement of (39). (The replacement of x by y and f by g is a result of eliminating the extrinsic variables - the key change for the modification of Newton's method is that we have subtracted $\vec{y}^{\nu}$ from each side of equation (39).)

$$\vec{y}^{\,\nu+1} - \vec{y}^{\,\nu} = -\left[J_{\vec{y}}\left(\vec{g}(\vec{y})\right)\right]^{-1}_{\vec{y}=\vec{y}^{\nu}} \cdot \vec{g}(\vec{y}^{\,\nu}) \qquad \textbf{(45)}$$

Let

$$\Delta y^{\nu} = y^{\nu+1} - y^{\nu} \qquad \textbf{(46)}$$

then multiplying both sides by the Jacobian we have

$$\left[J_{\vec{y}}\left(\vec{g}(\vec{y})\right)\right]_{\vec{y}=\vec{y}^{\nu}} \cdot \Delta y^{\nu} = -\vec{g}(\vec{y}^{\,\nu}) \qquad \textbf{(47)}$$

which is of the form Ax = b and can be solved by Gaussian elimination (See Lab demo #3.) for Δy^{ν}. Then $y^{\nu+1}$ is calculated from (46) and the process is repeated until $\left|\vec{g}(\vec{y})\right| \le \epsilon$ (using the infinity norm), or until the maximum number of allowed iterations is reached. Sparse matrix techniques are automatically used by Matlab whenever the matrices in question are declared sparse. Gausian elimination on Ax = b is invoked in Matlab by x=A\b.

Lab Demo #4: Newton's Method Let us solve the following set of equations directly by Newton's method.

$$6x_1^2 + 3x_1x_2 + 4x_3^2 = 8$$
$$x_1\cos(2x_2) + \sin(x_2) = 5x_3 - 7$$
$$4x_1^2x_2 + 2x_1 + 3x_3^2x_2 = x_2$$

First we write the equations in matrix form as

$$\vec{f}(\vec{x}) = \begin{bmatrix} 6x_1^2 + 3x_1x_2 + 4x_3^2 - 8 \\ x_1\cos(2x_2) + \sin(x_2) - 5x_3 + 7 \\ 4x_1^2x_2 + 2x_1 + 3x_3^2x_2 - x_2 \end{bmatrix} = 0$$

where

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Then we recall the Jacobian matrix of the above equations that we have calculated earlier.

$$J_x\left(\vec{f}(\vec{x})\right) = \begin{bmatrix} 12x_1 + 3x_2 & 3x_1 & 8x_3 \\ \cos(2x_2) & -2x_1\sin(2x_2) + \cos(x_2) & -5 \\ 8x_1x_2 + 2 & 4x_2^2 + 3x_3^2 - 1 & 6x_3x_2 \end{bmatrix}$$

Now when we give Matlab our initial estimate x = [0; 0.5; 1] the implication is that

$$\begin{aligned} x(1) &= x_1 \\ x(2) &= x_2 \\ x(3) &= x_3 \end{aligned}$$

since in Matlab if there exists a vector x then x(i) is the i^{th} component of vector x.
Recall the steps in a Newton's method solution:

1. Make initial estimate of x
2. Calculate f using initial estimate
3. Calculate the Jacobian J using initial estimate
4. Calculate new estimate of x by $x_{new} = x_{old}$ - inv(J) * f
5. Repeat step 1 - 4 until f is close enough to zero

The following Matlab code achieves the desired result.

```
%File   ldemo4.m
   x=[0.0 ; 0.5 ; 1.0];% Initial Estimate%
 max=10;                % Maximum Number of Iterations%
 tol=0.001;             % Error Tolerance%
%
   for k=1:max
   f=[6*x(1)^2+3*x(1)*x(2)+4*x(3)^2-8
      x(1)*cos(2*x(2))+sin(x(2))-5*x(3)+7
      4*x(1)^2*x(2)+2*x(1)+3*x(3)^2*x(2)-x(2)];

   J(1,1)=12*x(1)+3*x(2);
   J(1,2)=3*x(1);
   J(1,3)=8*x(3);
   J(2,1)=cos(2*x(2));
   J(2,2)= -2*x(1)*sin(2*x(2))+cos(x(2));
   J(2,3)= -5;
   J(3,1)=8*x(1)*x(2)+2;
   J(3,2)=4*x(1)^2+3*x(3)^2-1;
   J(3,3)=6*x(3)*x(2);

      x=x-inv(J)*f;
%
     if norm(f,inf)<=tol;
     break;
     end;
   end;
```

Now one of the characteristics of the power flow problem is that, since each bus is connected to only a few nearby

buses, the admittance matrix YBUS is sparse, that is most of its entries are zero. Matlab has built-in facilities for taking advantage of the sparsity by not storing all the zero entries, but only the non-zero entries and the locations in the matrix of the non-zero entries.
Taking advantage of the sparsity of YBUS means that much larger power systems can be analyzed by a computer with a fixed amount of memory. Unfortunately, in general the inverse of a sparse matrix is not sparse. Therefore we would like to implement Newton's method without inverting the Jacobian matrix. This is accomplished by the following rearrangement:

Define	$x_{new} - x_{old}$ = deltax
then	$x_{new} = x_{old}$ - inv(J)*f
becomes	deltax = -inv(J)*f
Multiplying both sides by J gives	J *deltax = -f

The last equation is easily solved for deltax by Gaussian elimination (Recall Lab Demo #3 Prob. #1), then x_{new} can be calculated from the definition of deltax.

The Matlab code for the modified Newton's method is:

```
%File   ldemo4mod.m
   x=[0.0 ; 0.5 ; 1.0];% Initial Estimate%
 max=10;                % Maximum Number of Iterations%
 tol=0.001;             % Error Tolerance%
%
   for k=1:max
%
   f=[6*x(1)^2+3*x(1)*x(2)+4*x(3)^2-8
      x(1)*cos(2*x(2))+sin(x(2))-5*x(3)+7
      4*x(1)^2*x(2)+2*x(1)+3*x(3)^2*x(2)-x(2)];
```

```
%
   J(1,1)=12*x(1)+3*x(2);
   J(1,2)=3*x(1);
   J(1,3)=8*x(3);
   J(2,1)=cos(2*x(2));
   J(2,2)= -2*x(1)*sin(2*x(2))+cos(x(2));
   J(2,3)= -5;
   J(3,1)=8*x(1)*x(2)+2;
   J(3,2)=4*x(1)^2+3*x(3)^2-1;
   J(3,3)=6*x(3)*x(2);
%
 delatx = -J\f;        %These are the only two statements that%
      x = x+deltax;   %need to be changed%
%
     if norm(f,inf)<=tol;
     break;
     end;
%
   end;
```

The following matlab session shows the results of running ldemo4mod.m (Note that whenever you run an m-file that uses Newton's method, you must ALWAYS display the value of the iteration counter.)

```
EDU» who
EDU» ldemo4mod
EDU» who
Your variables are:

J          k          tol
f          max        x          deltax
```

```
EDU» k
k =
       8
EDU» x
x =
     0.0901
    -0.0360
     1.4108
EDU»
```

Assignment #6

Solve the following equations by Newton's method in Matlab

$$x_1 x_2 \cos(x_2 x_3) + x_2^2 \sin(x_1 x_2) + 4x_3 = 17$$
$$x_1^2 x_2^2 x_3 + x_1 x_2 \tan(x_3^2) = 6x_1 x_3$$
$$x_2 asin(3x_1 x_3) - x_1 \sqrt{x_2 x_3^2} = x_3 e^{x_1 x_2}$$

Use the following initial estimate:

$$x_1 = 0.05$$
$$x_2 = 5.0$$
$$x_3 = -1.5$$

Hint: remember that you must have f(x)=0 to start.
Matlab equations can be carried over from line to line by terminating the first part of the line with an ellipis (...).
In matlab the exponential function e^x must be invoked as exp(x). The natural log ln(x) is written log(x) in Matlab.

$$x_1 \cdot x_2 \cdot \cos\left(x_2 \cdot x_3\right) + x_2^2 \cdot \sin\left(x_1 \cdot x_2\right) + 4 \cdot x_3 - 17$$

$$x_1\, x_2 \cos\left(x_2\, x_3\right) + x_2^2 \sin\left(x_1\, x_2\right) + 4\, x_3 - 17 \tag{1}$$

$$\frac{\partial}{\partial x_2}\ \mathbf{(1)}$$

$$x_1 \cos\left(x_2\, x_3\right) - x_1\, x_2 \sin\left(x_2\, x_3\right) x_3 + 2\, x_2 \sin\left(x_1\, x_2\right) + x_2^2 \cos\left(x_1\, x_2\right) x_1 \tag{2}$$

CodeGeneration[*Matlab*]((**1**))

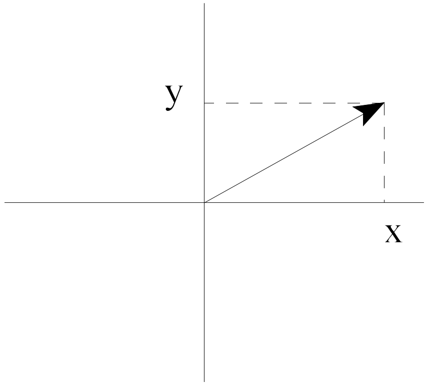
```
cg = x(1) * x(2) * cos(x(2) * x(3)) + x(2) ^ 2 * sin(x(1) * x(2))
+ 0.4e1 * x(3) - 0.17e2;
```

CodeGeneration[*Matlab*]((**2**))

```
cg0 = x(1) * cos(x(2) * x(3)) - x(1) * x(2) * sin(x(2) * x(3)) *
x(3) + 0.2e1 * x(2) * sin(x(1) * x(2)) + x(2) ^ 2 * cos(x(1) * x
(2)) * x(1);
```

Norms

Everyone should be familiar with the notion of the length of a vector given in Cartesian coordinates. For example the length of a vector whose coordinates are (x,y) is given by Pythagoras as $\sqrt{x^2 + y^2}$



The definition is easily extended to vectors in three space as $\sqrt{x^2 + y^2 + z^2}$ The length of an n dimensional vector in n dimensional space $\mathbb{R}^n$ is given as

$$\left(\sum_{k=1}^{n} x_k^2 \right)^{\frac{1}{2}}$$

The notion of length is extended in theoretical numerical analysis to the norm. In general the p-norm of a vector is given as

$$\left(\sum_{k=1}^{n} x_k^p\right)^{\frac{1}{p}}$$

We see that for p=2 the 2-norm is simply the length of a vector in the usual sense. If we take the limit of the p-norm as p approaches infinity we call the result the infinity-norm, and we find that the infinity-norm of a vector is simply the largest component of the vector. This is the reason for using the norm(f,inf) statement to determine when Newton's method has converged. To see that the infinity norm is indeed the largest component of a vector consider the vector

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

where $x_1 > x_2 > x_3$ Now consider $\lim_{p\to\infty}\left(x_1^p + x_2^p + x_3^p\right)$ now if $x_1 > x_2$ then $x_1^p \gg x_2^p$ as p becomes sufficiently large. So as p-> infinity, x_2^p is negligible compared to x_1^p. A similar argument allows us to neglect x_3^p compared to x_1^p. So

$$\lim_{p\to\infty}\left(x_1^p + x_2^p + x_3^p\right) = x_1^p$$

and

$$\lim_{p\to\infty}\left(x_1^p + x_2^p + x_3^p\right)^{\frac{1}{p}} = \left(x_1^p\right)^{\frac{1}{p}} = x_1$$

which is the result we are trying to prove.

Calculation of the Jacobian

Having removed the extrinsic variables (and their equations) the Jacobian matrix is given as

$$J = \begin{bmatrix} \begin{bmatrix} \dfrac{\partial f_{R1}}{\partial \delta_1} & \cdots & \dfrac{\partial f_{R1}}{\partial \delta_{n-1}} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial f_{R(n-1)}}{\partial \delta_1} & \cdots & \dfrac{\partial f_{R(n-1)}}{\partial \delta_{n-1}} \end{bmatrix} & \begin{bmatrix} \dfrac{\partial f_{R1}}{\partial V_1} & \cdots & \dfrac{\partial f_{R1}}{\partial V_{n-\ell}} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial f_{R(n-1)}}{\partial V_1} & \cdots & \dfrac{\partial f_{R(n-1)}}{\partial V_{n-\ell}} \end{bmatrix} \\ \begin{bmatrix} \dfrac{\partial f_{I1}}{\partial \delta_1} & \cdots & \dfrac{\partial f_{I1}}{\partial \delta_{n-1}} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial f_{I(n-\ell)}}{\partial \delta_1} & \cdots & \dfrac{\partial f_{I(n-\ell)}}{\partial \delta_{n-1}} \end{bmatrix} & \begin{bmatrix} \dfrac{\partial f_{I1}}{\partial V_1} & \cdots & \dfrac{\partial f_{I1}}{\partial V_{n-\ell}} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial f_{I(n-\ell)}}{\partial V_1} & \cdots & \dfrac{\partial f_{I(n-\ell)}}{\partial V_{n-\ell}} \end{bmatrix} \end{bmatrix} \tag{48}$$

We partition the Jacobian as follows:

$$J = \begin{bmatrix} [C] & [D] \\ [E] & [F] \end{bmatrix} \tag{49}$$

where [C] is an (n-1) x (n-1) matrix
[D] is an (n-1) x (n-ℓ) matrix
[E] is an (n-ℓ) x (n-1) matrix
[F] is an (n-ℓ) x (n-ℓ) matrix

This partition exposes the four different derivatives that must be taken.

$$
\begin{aligned}
C_{ij} &= \frac{\partial}{\partial \delta_j} f_{Ri} \\
&= \frac{\partial}{\partial \delta_j} \left(P_i - \sum_{k=1}^{n} V_i \, Y_{ik} \, V_k \cos\left(\Theta_{ik} - \delta_i + \delta_k\right) \right) \qquad \begin{array}{l} i = 1,\ldots,(n-1) \\ j = 1,\ldots,(n-1) \end{array} \\
&= 0 - V_i \left(\sum_{k=1}^{n} Y_{ik} \, V_k \frac{\partial}{\partial \delta_j} \cos\left(\Theta_{ik} - \delta_i + \delta_k\right) \right)
\end{aligned} \tag{50}
$$

Now the derivative of the right hand side of the last equality in (50) has a non-zero value only when i = j or when i ≠ j AND k = j. (These are the only conditions under which the expression we are trying to differentiate contains δ_j.) So we have

$$
\begin{aligned}
C_{ij} &= -V_i \sum_{\substack{k=1 \\ k \neq i}}^{n} Y_{ik} \, V_k \sin\left(\Theta_{ik} - \delta_i + \delta_k\right) \qquad & i = j \\
&= V_i \, Y_{ij} \, V_j \sin\left(\Theta_{ij} - \delta_i + \delta_j\right) \qquad & i \neq j
\end{aligned} \tag{51}
$$

for i = 1, ... , n-1 and j = 1, ... , n-1 in C.

$$D_{ij} = \frac{\partial}{\partial V_j}\left(P_i - \sum_{k=1}^{n} V_i Y_{ik} V_k \cos\left(\Theta_{ik} - \delta_i + \delta_k\right)\right)$$

$$= 0 - \sum_{\substack{k=1 \\ k \neq i}}^{n} Y_{ik} V_k \cos\left(\Theta_{ik} - \delta_i + \delta_k\right) \tag{52}$$

$$- \frac{\partial}{\partial V_j}\left(V_j Y_{ij} V_j \cos\left(\Theta_{ii} - \delta_i + \delta_i\right)\right)$$

(The last term in (52) is the k = i =j term that was left out of the summation - see eq (59) .)

$$D_{ij} = - \sum_{\substack{k=1 \\ k \neq i}}^{n} Y_{ik} V_k \cos\left(\Theta_{ik} - \delta_i + \delta_k\right) - 2 V_i Y_{ii} \cos\left(\Theta_{ii}\right) \qquad (i = j)$$

$$D_{ij} = - Y_{ij} V_i \cos\left(\Theta_{ij} - \delta_i + \delta_j\right) \qquad (i \neq j) \tag{53}$$

for i = 1, ... , n-1 and j = 1, ... , n-ℓ in D.

$$E_{ij} = \frac{\partial}{\partial \delta_j} f_{Ii}$$

$$= \frac{\partial}{\partial \delta_j}\left(Q_i + \sum_{k=1}^{n} V_i Y_{ik} V_k \sin\left(\Theta_{ik} - \delta_i + \delta_k\right)\right) \qquad \begin{matrix} i = 1,\ldots,(n-\ell) \\ j = 1,\ldots,(n-1) \end{matrix} \tag{54}$$

$$=0 + V_i \left(\sum_{k=1}^{n} Y_{ik} V_k \frac{\partial}{\partial \delta_j} \sin\left(\Theta_{ik} - \delta_i + \delta_k\right)\right)$$

for i = 1, ... , n-ℓ, and j = 1, ... , n-1 in E.

$$E_{i,j} = -V_i \cdot \sum_{\substack{k=1 \\ k \neq i}}^{n} Y_{i,k} \cdot V_k \cdot \cos\left(\Theta_{i,k} - \delta_i + \delta_k\right) \qquad (i = j) \tag{55}$$

$$V_i \cdot Y_{i,j} \cdot V_j \cdot \cos\left(\Theta_{i,j} - \delta_i + \delta_j\right) \qquad (i \neq j)$$

for i = 1, ... , n-ℓ and j = 1, ... , n-ℓ in F.

$$\begin{aligned} F_{ij} &= \frac{\partial}{\partial V_j}\left(Q_i + \sum_{k=1}^{n} V_i \, Y_{ik} \, V_k \sin\left(\Theta_{ik} - \delta_i + \delta_k\right)\right) \\ &= 0 + \sum_{\substack{k=1 \\ k \neq i}}^{n} Y_{ik} \, V_k \sin\left(\Theta_{ik} - \delta_i + \delta_k\right) \\ &\quad + \frac{\partial}{\partial V_j}\left(V_j \, Y_{ij} \, V_j \sin\left(\Theta_{ii} - \delta_i + \delta_i\right)\right) \end{aligned} \tag{56}$$

$$\begin{aligned} F_{ij} &= \sum_{\substack{k=1 \\ k \neq i}}^{n} Y_{ik} \, V_k \sin\left(\Theta_{ik} - \delta_i + \delta_k\right) + 2 \, V_i \, Y_{ii} \sin\left(\Theta_{ii}\right) \qquad (i = j) \\ F_{ij} &= Y_{ij} \, V_i \sin\left(\Theta_{ij} - \delta_i + \delta_j\right) \qquad (i \neq j) \end{aligned} \tag{57}$$

If we define (n-1) dimensional vector CI and (n-1) dimensional vector DI as

$$CI_i = \sum_{\substack{k=1 \\ k \neq i}}^{n} Y_{ik} \; V_k \sin\left(\Theta_{ik} - \delta_i + \delta_k\right) \qquad (i = 1,\ldots,n-1) \tag{58}$$

$$DI_i = \sum_{\substack{k=1 \\ k \neq i}}^{n} Y_{ik} \; V_k \cos\left(\Theta_{ik} - \delta_i + \delta_k\right) \qquad (i = 1,\ldots,n-1) \tag{59}$$

then the Jacobian equations reduce to

$$\left.\begin{aligned} C_{ij} &= - V_i \; CI_i && i = j \\ C_{ij} &= Y_{ij} \; V_i \; V_j \sin\left(\Theta_{ij} - \delta_i + \delta_j\right) && i \neq j \end{aligned}\right\} \quad \left\{\begin{aligned} i &= 1,\ldots,(n-1) \\ j &= 1,\ldots,(n-1) \end{aligned}\right\} \tag{60}$$

$$\left.\begin{aligned} D_{ij} &= - DI_i - 2 \; Y_{ii} \cos\left(\Theta_{ii}\right) && i = j \\ D_{ij} &= - Y_{ij} \; V_i \cos\left(\Theta_{ij} - \delta_i + \delta_j\right) && i \neq j \end{aligned}\right\} \quad \left\{\begin{aligned} i &= 1,\ldots,(n-1) \\ j &= 1,\ldots,(n-\ell) \end{aligned}\right\} \tag{61}$$

$$\left.\begin{aligned} E_{ij} &= - V_i \; DI_i && i = j \\ E_{ij} &= Y_{ij} \; V_i \; V_j \cos\left(\Theta_{ij} - \delta_i + \delta_j\right) && i \neq j \end{aligned}\right\} \quad \left\{\begin{aligned} i &= 1,\ldots,(n-\ell) \\ j &= 1,\ldots,(n-1) \end{aligned}\right\} \tag{62}$$

$$\left.\begin{aligned} F_{ij} &= CI_i + 2 \; Y_{ii} \sin\left(\Theta_{ii}\right) && i = j \\ F_{ij} &= Y_{ij} \; V_i \sin\left(\Theta_{ij} - \delta_i + \delta_j\right) && i \neq j \end{aligned}\right\} \quad \left\{\begin{aligned} i &= 1,\ldots,(n-\ell) \\ j &= 1,\ldots,(n-\ell) \end{aligned}\right\} \tag{63}$$

Once the matrices C, D, E, and F are constructed, it is a simple matter to build J in Matlab as

J = [C D ; E F]

Further once (n-1) dimensional vector fR and (n-ℓ) dimensional vector fI have been constructed according to (41) it is a simple matter to build g in Matlab as

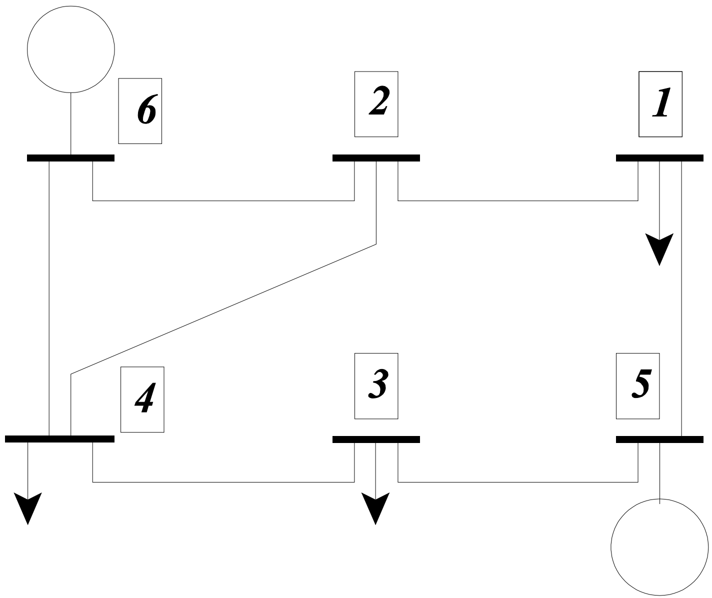
g=[fR; fI].

Matlab Program (This is page 63)
The Matlab program to solve this problem completes the following tasks in the following order.

1. Construct n x n matrices Y and Theta
2. Construct n dimensional vectors V, delta, P, and Q (1 and 2 are done in script file lfdat.m)
3. Construct (2n - ℓ -1) dimensional vectors y, and g
4. Construct (2n - ℓ -1) x (2n - ℓ -1) matrix J
5. Solve (47) for Δy
6. Use (46) to calculate new value for y
7. Evaluate |g(y)| for new value of y
8. If |g(y)| $\leq$ tol stop, if not go back to 3.
9. Calculate the values of the extrinsic variables. (3 - 9 are done in script file lfnewt.m)

Project #4

Solve the power-flow equations for the simplified Ward-Hale 6 bus system shown below. (The transformers have been removed to simplify the problem.) Your program must be completely general: loops should count from 1 to n not 1 to 6 from 1 to NLS not from 1 to 7 etc. Your program must not take advantage of the special form of the data i.e some of the entries are zero.



Use the following transmission line data:

From	To	R	X	HLC
6	4	0.123	0.518	0.0198
6	2	0.080	0.370	0.0141
2	4	0.097	0.407	0.0153
4	3	0.001	0.300	0.0
5	3	0.282	0.640	0.0
5	1	0.723	1.050	0.0
2	1	0.001	0.133	0.0

The system has the following voltage, load power and generator power data:

Bus	V	δ	Pg	Qg	Pd	Qd
1			0.0	0.0	0.52	0.13
2			0.0	0.0	0.0	0.0
3			0.0	0.0	0.282	0.18
4			0.0	0.0	0.47	0.05
5	1.10		0.5		0.0	0.0
6	1.05	0.0			0.0	0.0

The blanks in this data table should be filled in with the initial estimates. For voltage magnitudes (V) the initial estimate is 1.0. For Voltage angles (δ) the initial estimate is 0.0. For each of the two Qg entries, the initial estimate should be

$$\frac{\sum_{k=1}^{n} Qd_k}{\ell}.$$

For the initial estimate of the slack bus power Pg, use

$$\sum_{k=1}^{n} Pd_k - \sum_{k=n-\ell+1}^{k=n-1} Pg_k$$

You must submit hard copies of both the m-file(s) you use and the matlab session that displays the results. Your results must include the value of the iteration counter, all of the bus voltage magnitudes, all of the bus voltage angles, all of the generator Qg values and the slack generator real power. You must also submit a copy of your m-file(s) to the digital drop-box, so that I may verify that they run correctly.

Load-flow hints: to do (1.) & (2.) Of the **Matlab Program**(p63), put (be sure to define n, NLS, and L) in file lfdat.m

Calculate Y and Theta as in assignment 5 then

```
BD= [                          ;
                               ;
                               ;
                               ;
                               ;
                               ;
                                ];
```

```
for i=1:n
    V(i)=BD(i,2);
delta(i)=BD(i,3);
    P(i)=BD(i,4)-BD(i,6);
    Q(i)=BD(i,5)-BD(i,7);
end;
```

In file lfnewt.m:

```
dd=zeros(n-1,1); %Defines an n-1 dimensional column vector.)
vv=zeros(n-L,1);
for i=1:n-1          % Generate y from V and delta %
dd(i)=delta(i);
end;
for i=1:n-L
vv(i)=V(i);
end;

y=[dd; vv]];

for iter=1:max        % Newton's algorithm starts here.%

CI=zeros(n,1);

% from (58) we write the following nested loops %

% Even though we only need n-1 CIs and DIs for the calculation of the %
% Jacobian, we calculate n values of CI and DI for use in calculating the %
% extrinsic variables. %
```

```
for i=1:n
  for k=1:n
    if k~=i   % ~= means "not equal to" %
    CI(i)=CI(i)+Y(i,k)*V(k)*sin( Theta(i,k)-delta(i)+delta(k) );
    else
    end;
  end;
end;

DI stuff from (59)

fR=zeros(n-1,1);
for i=1:n-1
fR(i)=P(i)-V(i)*DI(i)-V(i)*Y(i,i)*V(i)*cos( Theta(i,i) ); %(41)%
end;                     ^
                         |
                         |
% The 3rd term in fR(i) is the k=i term missing from DI(i) %

fI(i) stuff from (41)

g=[fR; fI];
               %   This completes (3.)(p63) %


% Now calculate the Jacobian   (4.)(p63) %
```

```
% from (60)%

C=zeros(n-1,n-1)
for i=1:n-1
  for j=1:n-1
    if i==j
    C(i,j)=-V(i)*CI(i);
    else
    C(i,j)=Y(i,j)*V(i)*V(j)*sin( Theta(i,j)-delta(i)+delta(j) );
    end;
  end;
end;

% D stuff from (61) %

% E stuff from (62) %

% F stuff from (63) %

J=[C D; E F];

% (5.)(p63)    from (47) we write%

deltay = -J\g;

% (6.) (P63) %     From (46) we write

y=y+deltay;
```

```
% now change V and delta to the new values calculated in y %

for i=1:n-1
delta(i)=y(i);
end;

for i=1:n-L
V(i)=y(i+n-1);
end;


% to do (7.) & (8.)  (p63) we write %

if norm(g,inf) < tol
break;
else
end;

end;

% this last end ends the for iter=1:max %



% (9.) (P63) calculate Pn from fR(n) and
% Q(n-L+1) -> Q(n) from fI(n-L+1) -> fI(n) %
```

Assignment #5 Answers

```
EDU» B
B =

  -44.8360   19.0781   25.8478         0
   19.0781  -44.8360         0   25.8478
   25.8478         0  -40.8638   15.1185
         0   25.8478   15.1185  -40.8638

EDU» G
G =

    8.9852   -3.8156   -5.1696         0
   -3.8156    8.9852         0   -5.1696
   -5.1696         0    8.1933   -3.0237
         0   -5.1696   -3.0237    8.1933

EDU» Y
Y =

   45.7274   19.4560   26.3597         0
   19.4560   45.7274         0   26.3597
   26.3597         0   41.6771   15.4179
         0   26.3597   15.4179   41.6771
```

```
EDU» Theta
Theta =

   -1.3730    1.7682    1.7682         0
    1.7682   -1.3730         0    1.7682
    1.7682         0   -1.3729    1.7682
         0    1.7682    1.7682   -1.3729
```

Assignment #6 Answers

```
EDU» k
k =

     6

EDU» x
x =

    0.0646
    7.1919
   -1.6137

EDU»
```

Project #4 Hints and Answers

$$B = \begin{matrix} -8.1644 & 7.5184 & 0 & 0 & 0.6461 & 0 \\ 7.5184 & -12.3959 & 0 & 2.3249 & 0 & 2.5820 \\ 0 & 0 & -4.6418 & 3.3333 & 1.3085 & 0 \\ 0 & 2.3249 & 3.3333 & -7.4506 & 0 & 1.8275 \\ 0.6461 & 0 & 1.3085 & 0 & -1.9545 & 0 \\ 0 & 2.5820 & 0 & 1.8275 & 0 & -4.3756 \end{matrix}$$

$$G = \begin{matrix} 0.5014 & -0.0565 & 0 & 0 & -0.4449 & 0 \\ -0.0565 & 1.1689 & 0 & -0.5541 & 0 & -0.5583 \\ 0 & 0 & 0.5877 & -0.0111 & -0.5765 & 0 \\ 0 & -0.5541 & -0.0111 & 0.9991 & 0 & -0.4339 \\ -0.4449 & 0 & -0.5765 & 0 & 1.0214 & 0 \\ 0 & -0.5583 & 0 & -0.4339 & 0 & 0.9922 \end{matrix}$$

$$Y = \begin{matrix} 8.1798 & 7.5186 & 0 & 0 & 0.7844 & 0 \\ 7.5186 & 12.4509 & 0 & 2.3901 & 0 & 2.6417 \\ 0 & 0 & 4.6788 & 3.3333 & 1.4299 & 0 \\ 0 & 2.3901 & 3.3333 & 7.5173 & 0 & 1.8783 \\ 0.7844 & 0 & 1.4299 & 0 & 2.2053 & 0 \\ 0 & 2.6417 & 0 & 1.8783 & 0 & 4.4866 \end{matrix}$$

$$Theta = \begin{matrix} -1.5095 & 1.5783 & 0 & 0 & 2.1738 & 0 \\ 1.5783 & -1.4768 & 0 & 1.8048 & 0 & 1.7837 \\ 0 & 0 & -1.4449 & 1.5741 & 1.9858 & 0 \\ 0 & 1.8048 & 1.5741 & -1.4375 & 0 & 1.8039 \\ 2.1738 & 0 & 1.9858 & 0 & -1.0892 & 0 \\ 0 & 1.7837 & 0 & 1.8039 & 0 & -1.3478 \end{matrix}$$

After **ONE** iteration (set max =1 for this)

```
CI = [ 8.2290  12.5544    4.7726    7.5771    1.9545    4.4095]
DI = [-0.5459  -1.1968   -0.6453   -1.0208   -1.0214   -0.9922]
(Only 5 values of CI and DI are needed to calculate J, I use the    last
CI and DI values to calculate the intrinsic variables)

C =   -8.2290    7.5184         0         0    0.7107
       7.5184  -12.5544         0    2.3249         0
            0         0   -4.7726    3.3333    1.4393
            0    2.3249    3.3333   -7.5771         0
       0.7107         0    1.4393         0   -2.1500

D =   -0.4569    0.0565         0         0
       0.0565   -1.1410         0    0.5541
            0         0   -0.5300    0.0111
            0    0.5541    0.0111   -0.9775
       0.4893         0    0.6342         0

E =    0.5459   -0.0565         0         0   -0.4893
      -0.0565    1.1968         0   -0.5541         0
            0         0    0.6453   -0.0111   -0.6342
            0   -0.5541   -0.0111    1.0208         0

F =   -8.0998    7.5184         0         0
       7.5184  -12.2374         0    2.3249
            0         0   -4.5109    3.3333
            0    2.3249    3.3333   -7.3241
```

```
y =[-0.1862  -0.1410  -0.1676  -0.1732  -0.0204  0.9587  0.9753
     0.9406   0.9624]'

g =[  -0.4755  0.0279  -0.2243  -0.4483  0.3876  -0.0654  0.1585
      -0.0492  0.0765]'
```

After Newton’s method has converged (tol = 0.001) (set max = 10 for this)

```
iter = 4

y = [  -0.2071  -0.1547  -0.1944  -0.1937  -0.0501   0.9369
        0.9544   0.9204   0.9404]'

g =  1.0e-003 * [  -0.0680  -0.0991  -0.0899  -0.1341  -0.0012
                   -0.0018  -0.0052  -0.0015  -0.0053]'

V = [  0.9370  0.9545  0.9204  0.9405  1.1000  1.0500]'

(The following angles are in degrees)
d = [  -11.8628  -8.8567  -11.1302  -11.0904  -2.8635  0]'

Qg = [  0  0  0  0  0.2406  0.3278]'

Pg = [  0  0  0  0  0.5000  0.8661]'
```