

The Power Flow Problem

James D. McCalley, Iowa State University

T7.0 Introduction

The *power flow problem* is a very well known problem in the field of power systems engineering, where voltage magnitudes and angles for one set of buses are desired, given that voltage magnitudes and power levels for another set of buses are known and that a model of the network configuration (unit commitment and circuit topology) is available. A *power flow solution procedure* is a numerical method that is employed to solve the power flow problem. A *power flow program* is a computer code that implements a power flow solution procedure. The *power flow solution* contains the voltages and angles at all buses, and from this information, we may compute the real and reactive generation and load levels at all buses and the real and reactive flows across all circuits. The above terminology is often used with the word “load” substituted for “power,” i.e., load flow problem, load flow solution procedure, load flow program, and load flow solution. However, the former terminology is preferred as one normally does not think of “load” as something that “flows.”

The power flow problem was originally motivated within planning environments where engineers considered different network configurations necessary to serve an expected future load. Later, it became an operational problem as operators and operating engineers were required to monitor the real-time status of the network in terms of voltage magnitudes and circuit flows. Today, the power flow problem is widely recognized as a fundamental problem for power system analysis, and there are many advanced, commercial power flow programs to address it. Most of these programs are capable of solving the power flow program for tens of thousands of interconnected buses. Engineers that understand the power flow problem, its formulation, and corresponding solution procedures are in high demand, particularly if they also have experience with commercial grade power flow programs.

The power flow problem is fundamentally a network analysis problem, and as such, the study of it provides insight into solutions for similar problems that occur in other areas of electrical engineering. For example, integrated circuit designers also encounter network analysis problems, although of significantly smaller physical size, are quite similar otherwise to the power flow problem. For example, references [1,2] are well-known network analysis texts in VLSI design that also provide good insight into the numerical analysis needed by the power flow program designer. Similarly, there are numerous classical power system engineering texts, [3-11] are a representative sample, that provide advanced network analysis methods applicable to VLSI design and analysis problems.

Section T7.1 identifies a feature of power generators important to the power flow problem – real and reactive power limits. Section T7.2 defines some additional terminology necessary to understand the power flow problem and its solution procedure. Section T7.3 introduces the so-called network “Y-bus,” otherwise known more generally as the network admittance matrix. Section T7.4 develops the power flow equations, building from module T1 where equations for real and reactive power flow across a transmission line were introduced. Section T7.5 provides an analytical statement of the power flow problem. Section T7.6 uses a simple example to introduce the Newton-Raphson algorithm for solving systems of non-linear algebraic equations. Section T7.7 illustrates application of the Newton-Raphson algorithm to the power flow problem. Section T7.8 provides an overview of several interesting and advanced attributes of the problem. Section T7.9 summarizes basic power flow input and output quantities and provides an example associated with a commercial power flow program.

T7.1 Generator Reactive Limits

It is well known that generators have maximum and minimum real power capabilities. In addition, they also have maximum and minimum reactive power capabilities. The maximum reactive power capability

corresponds to the maximum reactive power that the generator may produce when operating with a lagging power factor. The minimum reactive power capability corresponds to the maximum reactive power the generator may absorb when operating with a leading power factor. These limitations are a function of the real power output of the generator, that is, as the real power increases, the reactive power limitations move closer to zero. The solid curve in Figure T7.1 is a typical *generator capability* curve, which shows the lagging and leading reactive limitations (the ordinate) as real power is varied (the abscissa). Most power flow programs model the generator reactive capabilities by assuming a somewhat conservative value for P_{\max} (perhaps 95% of the actual value), and then *fixing* the reactive limits Q_{\max} (for the lagging limit) and Q_{\min} (for the leading limit) according to the dotted lines shown in Fig. T2.1.

Approximate Q-limits of each generator using dotted box.

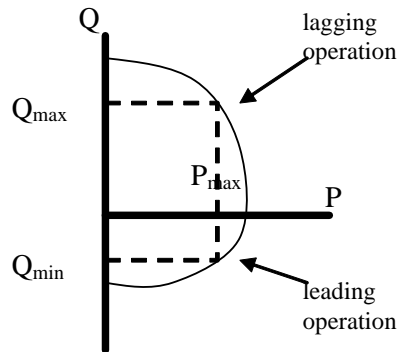


Fig T7.1: Generator Capability Curve and Approximate Reactive Limits

T7.2 Terminology

Bulk high voltage transmission systems are always comprised of three phase circuits. However, under balanced conditions (the currents in all three phases are equal in magnitude and phase separated by 120°), we may analyze the three phase system using a per-phase equivalent circuit consisting of the a-phase and the neutral conductor. Per-unitization of a per-phase equivalent of a three phase, balanced system results in the per-unit circuit. It is the per-unitized, per-phase equivalent circuit of the power system that we use to formulate and solve the power flow problem. For the remainder of this module, we will assume that all quantities are in per-unit. The reader unfamiliar with per-phase equivalent circuits or the per-unit system should refer to modules B3 and B4, respectively.

For pf analysis, we use per-phase circuits in the per-unit system.

One-line diagram: per-phase circuit diagram without neutral.

It is convenient to represent power system networks using the so-called one-line diagram, which can be thought of as the circuit diagram of the per-phase equivalent, but without the neutral conductor (module B3 also provides additional background on the one-line diagram). Figure T7.2 illustrates the one-line diagram of a small transmission system.

Fig. T7.2 illustrates several important elements of the power flow problem. First, one notices we may categorize each bus depending on whether generation and/or load is connected to it. Specifically, a bus may have generation only (buses B1, B2, and B3), load only (buses B5, B7, and B9), or neither generation or load (buses B4, B6, and B8). In addition, a bus may have both generation and load, although none of the buses in Fig. 1 fall into this category. This categorization, which focuses on the load and generation, leads us to define the term “bus injection” or more simply, “injection.” We will use this term frequently, and the student is advised to carefully note its meaning, given and discussed in the following paragraph.

An *injection* is the power, either real or reactive, that is being injected into or withdrawn from a bus by an element having its other terminal (in the per-phase equivalent circuit) connected to ground. Such an element would be either a generator or a load. We define a positive injection as one where power is flowing from the element into the bus (i.e., into the network); a negative injection is then when power is flowing from the bus (i.e., from the network) into the element. Generators normally have positive real power injections, although they may also be assigned negative real power injections, in which case they are

operating as a motor. Generators may have either positive or negative reactive power injections: positive if the generator is operating lagging and delivering reactive power to the bus, negative if the generator is operating leading and absorbing reactive power from the bus, and zero if the generator is operating at unity power factor. Loads normally have negative real and reactive power injections, although they may also be assigned positive real power injections in the case of very special modeling needs. Figure T7.3 (a) and (b) illustrate the two most common possibilities. Figure T7.3 (c) illustrates that we must compute a net injection as the algebraic sum when a bus has both load and generation; in this case, the net injection for both real and reactive power is positive (into the bus). Thus, the net real power injection is $P_k = P_{gk} - P_{dk}$, and the net reactive power injection is $Q_k = Q_{gk} - Q_{dk}$. We may also refer to the net complex power injection as $S_k = S_{gk} - S_{dk}$, where $S_k = P_k + jQ_k$.

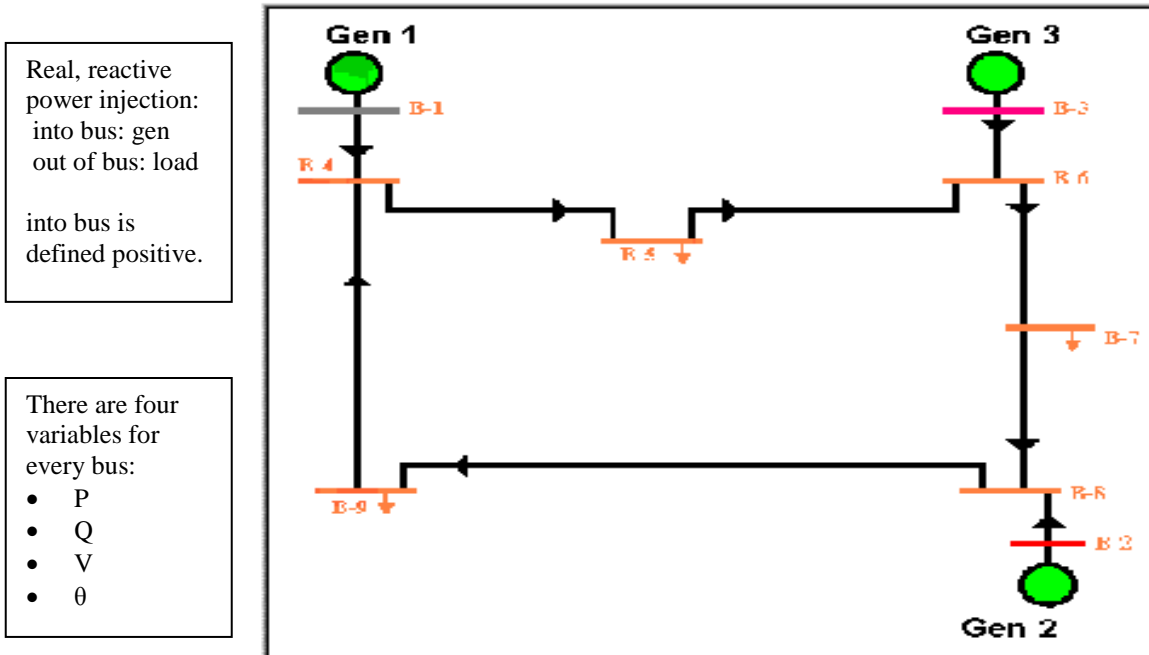


Figure T7.2: Single Line Diagram for Simple Power System

Real, reactive power injection:
 into bus: gen
 out of bus: load

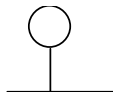
into bus is defined positive.

There are four variables for every bus:

- P
- Q
- V
- θ

$$P_k = 100$$

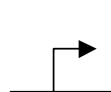
$$Q_k = 30$$



(a)

$$P_k = -40$$

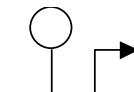
$$Q_k = -20$$



(b)

$$P_k = 100 + (-40) = 60$$

$$Q_k = 30 + (-20) = 10$$



(c)

Three bus types:

1. PV
2. PQ
3. Swing ($P\theta$)

Losses are a function of flows. We do not know flows *a priori*. So one gen-bus has injection as a function of the solution.

$$P_1 = P_{Losses}(solution) - \sum_{k=2,N} P_k$$

$$Q_1 = Q_{Losses}(solution) - \sum_{k=2,N} Q_k$$

Fig T7.3: Illustration of (a) positive injection, (b) negative injection, and (c) net injection

Although it is physically appealing to categorize buses based on the generation/load mix connected to it, we need to be more precise in order to analytically formulate the power flow problem. For proper analytical formulation, it is appropriate to categorize the buses according to what information is known about them before we solve the power flow problem. For each bus, there are four possible variables that characterize the buses electrical condition. Let us consider an arbitrary bus numbered k . The four variables are real and reactive power injection, P_k and Q_k , respectively, and voltage magnitude and angle, $|V_k|$ and θ_k , respectively. From this perspective, there are three basic types of buses. We refer to the first two types using terminology that remind us of the known variables.

- *PV Buses:* For type PV buses, we know P_k and $|V_k|$ but not Q_k or θ_k . These buses fall under the category of *voltage-controlled buses* because of the ability to specify (and therefore to know) the voltage magnitude of this bus. Most generator buses fall into this category, independent of whether it also has load; exceptions are buses that have reactive power injection at either the generator's upper limit (Q_{\max}) or its lower limit (Q_{\min}), and (2) the system swing bus (we describe the swing bus below). There are also special cases where a non-generator bus (i.e., either a bus with load or a bus with neither generation or load) may be classified as type PV, and some examples of these special cases are buses having switched shunt capacitors or static var systems (SVCs). We will not address these special cases in this module. In Fig. T7.2, buses B2 and B3 are type PV. The real power injections of the type PV buses are chosen according to the system dispatch corresponding to the modeled loading conditions. The voltage magnitudes of the type PV buses are chosen according to the expected terminal voltage settings, sometimes called the generator "set points," of the units.
- *PQ Buses:* For type PQ buses, we know P_k and Q_k but not $|V_k|$ or θ_k . All load buses fall into this category, including buses that have not either load or generation. In Fig. T7.2, buses B4-B9 are all type PQ. The real power injections of the type PQ buses are chosen according to the loading conditions being modeled. The reactive power injections of the type PQ buses are chosen according to the expected power factor of the load.

The third type of bus is referred to as the *swing bus*. Two other common terms for this bus are *slack bus* and *reference bus*. There is only one swing bus, and it can be designated by the engineer to be any generator bus in the system. For the swing bus, we know $|V|$ and θ . The fact that we know θ is the reason why it is sometimes called the reference bus. Physically, there is nothing special about the swing bus; in fact, it is a mathematical artifact of the solution procedure. At this point in our treatment of the power flow problem, it is most appropriate to understand this last statement in the following way. The generation must supply both the load and the losses on the circuits. Before solving the power flow problem, we will know all injections at PQ buses, but we will not know what the losses will be as losses are a function of the flows on the circuits which are yet to be computed. So we may set the real power injections for, at most, all but one of the generators. The one generator for which we do not set the real power injection is the one modeled at the swing bus. Thus, this generator "swings" to compensate for the network losses, or, one may say that it "takes up the slack." Therefore, rather than call this generator a $|V|\theta$ bus (as the above naming convention would have it), we choose the terminology "swing" or "slack" as it helps us to better remember its function. The voltage magnitude of the swing bus is chosen to correspond to the typical voltage setting of this generator. The voltage angle may be designated to be any angle, but normally it is designated as 0° .

Check the swing bus after a solution to ensure it is within physical limits.

A word of caution about the swing bus is in order. Because the real power injection of the swing bus is not set by the engineer but rather is an output of the power flow solution, it can take on mathematically tractable but physically impossible values. Therefore, the engineer must always check the swing bus generation level following a solution to ensure that it is within the physical limitations of the generator.

T7.3 The Admittance Matrix

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 T7.4 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 .

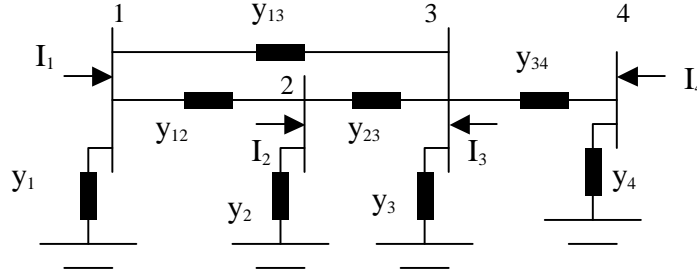


Fig. T7.4: Network for Motivating Admittance Matrix

Express current injections I_k using KCL. To be general, assume a fully connected network.

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:

$$I_1=(V_1-V_2)y_{12} + (V_1-V_3)y_{13} + V_1y_1 \tag{T7.1}$$

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:

$$I_1=(V_1-V_2)y_{12} + (V_1-V_3)y_{13} + (V_1-V_4)y_{14} + V_1y_1 \tag{T7.2}$$

Note that the current contribution of the term containing y_{14} is zero since y_{14} is zero. Rearranging eq. T7.2, we have:

$$I_1= V_1(y_1 + y_{12} + y_{13} + y_{14}) + V_2(-y_{12})+ V_3(-y_{13}) + V_4(-y_{14}) \tag{T7.3}$$

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

$$\begin{aligned} I_2 &= V_1(-y_{21}) + V_2(y_2 + y_{21} + y_{23} + y_{24}) + V_3(-y_{23}) + V_4(-y_{24}) \\ I_3 &= V_1(-y_{31})+ V_2(-y_{32}) + V_3(y_3 + y_{31} + y_{32} + y_{34}) + V_4(-y_{34}) \\ I_4 &= V_1(-y_{41})+ V_2(-y_{42}) + V_3(-y_{34})+ V_4(y_4 + y_{41} + y_{42} + y_{43}) \end{aligned} \tag{T7.4}$$

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. (T7.3) and (T7.4), 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:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} y_1 + y_{12} + y_{13} + y_{14} & -y_{12} & -y_{13} & -y_{14} \\ -y_{21} & y_2 + y_{21} + y_{23} + y_{24} & -y_{23} & -y_{24} \\ -y_{31} & -y_{32} & y_3 + y_{31} + y_{32} + y_{34} & -y_{34} \\ -y_{41} & -y_{42} & -y_{43} & y_4 + y_{41} + y_{42} + y_{43} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \tag{T7.5}$$

The matrix containing the network admittances in eq. (T7.5) is the admittance matrix, also known as the Y-bus, and denoted as:

$$\underline{Y} = \begin{bmatrix} y_1 + y_{12} + y_{13} + y_{14} & -y_{12} & -y_{13} & -y_{14} \\ -y_{21} & y_2 + y_{21} + y_{23} + y_{24} & -y_{23} & -y_{24} \\ -y_{31} & -y_{32} & y_3 + y_{31} + y_{32} + y_{34} & -y_{34} \\ -y_{41} & -y_{42} & -y_{43} & y_4 + y_{41} + y_{42} + y_{43} \end{bmatrix} \quad (\text{T7.6})$$

Observe the symmetry of the Y-bus.

Denoting the element in row i , column j , as Y_{ij} , we rewrite eq. (T7.6) as:

$$\underline{Y} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \quad (\text{T7.7})$$

where the terms Y_{ij} are not admittances but rather elements of the admittance matrix. Therefore, eq. (T7.6) becomes:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \quad (\text{T7.8})$$

By using eq. (T7.7) and (T7.8), and defining the vectors \underline{V} and \underline{I} , we may write eq. (T7.8) in compact form according to:

$$\underline{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}, \quad \underline{I} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} \quad \Rightarrow \quad \underline{I} = \underline{Y}\underline{V} \quad (\text{T7.9})$$

We make several observations about the admittance matrix given in eqs. (T7.6) and (T7.7). These observations hold true for any linear network of any size.

1. The matrix is symmetric, i.e., $Y_{ij}=Y_{ji}$.
 2. A diagonal element Y_{ii} is obtained as the sum of admittances for all branches connected to bus i , including the shunt branch, i.e., $Y_{ii} = y_i + \sum_{k=1, k \neq i}^N y_{ik}$, where we emphasize once again that y_{ik} is non-zero only when there exists a physical connection between buses i and k .
 3. The off-diagonal elements are the negative of the admittances connecting buses i and j , i.e., $Y_{ij}=-y_{ji}$.
- These observations enable us to formulate the admittance matrix very quickly from the network based on visual inspection. The following example will clarify.

Example T7.1

Consider the network given in Fig. T7.5, where the numbers indicate admittances.

Y-bus formulation rules.

The shunt elements all have positive susceptance, and must therefore be capacitive.

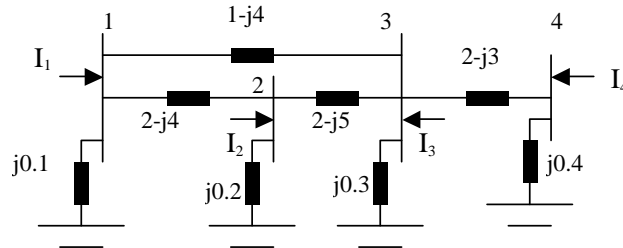


Fig. T7.5: Circuit for Example T7.1

The elements of Y-bus are designated as $Y_{kj} = G_{kj} + jB_{kj}$. Observe $B_{kj} < 0$ for diagonal elements; $B_{kj} > 0$ otherwise.

The admittance matrix is given by inspection as:

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

T7.4 The power flow equations

We have defined the net complex power injection into a bus, in Section T7.2, as $S_k = S_{gk} - S_{dk}$. In this section, we desire to derive an expression for this quantity in terms of network voltages and admittances. We begin by reminding the reader that all quantities are assumed to be in per unit, so we may utilize single-phase power relations. Drawing on the familiar relation for complex power, we may express S_k as:

$$S_k = V_k I_k^* \tag{T7.10}$$

From eq. (T7.8), we see that the current injection into any bus k may be expressed as

$$I_k = \sum_{j=1}^N Y_{kj} V_j \tag{T7.11}$$

where, again, we emphasize that the Y_{kj} terms are admittance matrix elements and not admittances. Substitution of eq. (T7.11) into eq. (T7.10) yields:

The most important step in deriving pf eqts: substitute Y-bus relation into $S = VI^*$.

$$S_k = V_k \left(\sum_{j=1}^N Y_{kj} V_j \right)^* = V_k \sum_{j=1}^N Y_{kj}^* V_j^* \tag{T7.12}$$

Recall that V_k is a phasor, having magnitude and angle, so that $V_k = |V_k| \angle \theta_k$. Also, Y_{kj} , being a function of admittances, is therefore generally complex, and we define G_{kj} and B_{kj} as the real and imaginary parts of the admittance matrix element Y_{kj} , respectively, so that $Y_{kj} = G_{kj} + jB_{kj}$. Then we may rewrite eq. (T7.12) as

$$\begin{aligned} S_k &= V_k \sum_{j=1}^N Y_{kj}^* V_j^* = |V_k| \angle \theta_k \sum_{j=1}^N (G_{kj} + jB_{kj})^* (|V_j| \angle \theta_j)^* = |V_k| \angle \theta_k \sum_{j=1}^N (G_{kj} - jB_{kj}) (|V_j| \angle -\theta_j) \\ &= \sum_{j=1}^N |V_k| \angle \theta_k (|V_j| \angle -\theta_j) (G_{kj} - jB_{kj}) = \sum_{j=1}^N (|V_k| |V_j| \angle (\theta_k - \theta_j)) (G_{kj} - jB_{kj}) \end{aligned} \tag{T7.13}$$

Recall, from the Euler relation, that a phasor may be expressed as complex function of sinusoids, i.e., $V=|V|\angle\theta=|V|\{\cos\theta+j\sin\theta\}$, we may rewrite eq. (T7.13) as

$$\begin{aligned} S_k &= \sum_{j=1}^N \left(|V_k| |V_j| \angle(\theta_k - \theta_j) \right) (G_{kj} - jB_{kj}) \\ &= \sum_{j=1}^N |V_k| |V_j| \left(\cos(\theta_k - \theta_j) + j \sin(\theta_k - \theta_j) \right) (G_{kj} - jB_{kj}) \end{aligned} \quad (\text{T7.14})$$

If we now perform the algebraic multiplication of the two terms inside the parentheses of eq. (T7.14), and then collect real and imaginary parts, and recall that $S_k=P_k+jQ_k$, we can express eq. (T7.14) as two equations, one for the real part, P_k , and one for the imaginary part, Q_k , according to:

The pf equations! We will be very wise to ensure that the left-hand-side is always known.

$$\begin{aligned} P_k &= \sum_{j=1}^N |V_k| |V_j| \left(G_{kj} \cos(\theta_k - \theta_j) + B_{kj} \sin(\theta_k - \theta_j) \right) \\ Q_k &= \sum_{j=1}^N |V_k| |V_j| \left(G_{kj} \sin(\theta_k - \theta_j) - B_{kj} \cos(\theta_k - \theta_j) \right) \end{aligned} \quad (\text{T7.15})$$

The two equations of (T7.15) are called the power flow equations, and they form the fundamental building block from which we attack the power flow problem.

It is interesting to consider the case of eqs. (T7.15) if bus k , relabeled as bus p , is only connected to one other bus, let's say bus q . Then the bus p injection is the same as the flow into the line pq . The situation is illustrated in Fig. T7.6.

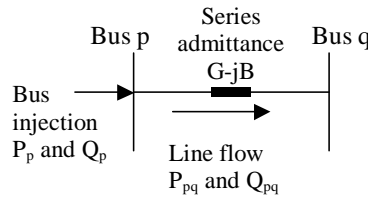


Fig. T7.6: Bus p Connected to Only Bus q

For the situation illustrated in Fig. T7.6, eqs. (T7.15) become:

$$\begin{aligned} P_p &= |V_p|^2 G_{pp} + |V_p| |V_q| G_{pq} \cos(\theta_p - \theta_q) + |V_p| |V_q| B_{pq} \sin(\theta_p - \theta_q) \\ Q_p &= -|V_p|^2 B_{pp} + |V_p| |V_q| G_{pq} \sin(\theta_p - \theta_q) - |V_p| |V_q| B_{pq} \cos(\theta_p - \theta_q) \end{aligned} \quad (\text{T7.16})$$

If the line pq admittance is $y=G-jB^1$, as shown in Fig. T7.6, then $G_{pq}=-G$ and $B_{pq}=B$ (see eq. T7.6). If there is no bus p shunt reactance or line charging, then $G_{pp}=G$ and $B_{pp}=B$. Under these conditions, eqs. (T7.16) become:

$$\begin{aligned} P_p &= |V_p|^2 G - |V_p| |V_q| G \cos(\theta_p - \theta_q) + |V_p| |V_q| B \sin(\theta_p - \theta_q) \\ Q_p &= |V_p|^2 B - |V_p| |V_q| G \sin(\theta_p - \theta_q) - |V_p| |V_q| B \cos(\theta_p - \theta_q) \end{aligned} \quad (\text{T7.17})$$

¹ We have defined the line pq admittance to be $y=G-jB$, instead of $y=G+jB$. The reason for defining in this way is because, since the line admittance y always represents inductive susceptance, the imaginary part of y must be negative; therefore the definition used here requires that B to be a positive number.

If we simply rearrange the order of the terms in the reactive equation, then we have:

$$\begin{aligned} P_p &= |V_p|^2 G - |V_p| |V_q| G \cos(\theta_p - \theta_q) + |V_p| |V_q| B \sin(\theta_p - \theta_q) \\ Q_p &= |V_p|^2 B - |V_p| |V_q| B \cos(\theta_p - \theta_q) - |V_p| |V_q| G \sin(\theta_p - \theta_q) \end{aligned} \quad (T7.18)$$

T7.5 Analytic statement of the power flow problem

#3 is LHS of N-1 real pf equations.

#4 is LHS of N-N_G reactive pf equations.

Since these LHS quantities are known, inclusion of the corresponding pf equation does not add an unknown.

Consider a power system network having N buses, N_G of which are voltage-regulating generators. One of these must be the swing bus. Thus there are N_G-1 type PV buses, and N-N_G type PQ buses. We assume that the swing bus is numbered bus 1, the type PV buses are numbered 2, ..., N_G, and the type PQ buses are numbered N_G+1, ..., N (this assumption on numbering is not necessary, but it makes the following development notationally convenient). It is typical that we know, in advance, the following information about the network (implying that it is input data to the problem):

1. The admittances of all series and shunt elements (implying that we can obtain the Y-bus),
2. The voltage magnitudes V_k, k=1, ..., N_G, at all N_G generator buses,
3. The real power injection of all buses except the swing bus, P_k, k=2, ..., N
4. The reactive power injection of all type PQ buses, Q_k, k=N_G+1, ..., N

Statements 3 and 4 indicate power flow equations for which we know the injections, i.e., the values of the left-hand side of eqs. (T7.15). These equations are very valuable because they have one less unknown than equations for which we do not know the left-hand-side. The number of these equations for which we know the left-hand-side can be determined by adding the number of buses for which we know the real power injection (statement 3 above) to the number of buses for which we know the reactive power injection (statement 4 above). This is (N-1)+(N-N_G)=2N-N_G-1. We repeat the power flow equations here, but this time, we denote the appropriate number to the right.

$$\begin{aligned} P_k &= \sum_{j=1}^N |V_k| |V_j| (G_{kj} \cos(\theta_k - \theta_j) + B_{kj} \sin(\theta_k - \theta_j)), \quad k = 2, \dots, N \\ Q_k &= \sum_{j=1}^N |V_k| |V_j| (G_{kj} \sin(\theta_k - \theta_j) - B_{kj} \cos(\theta_k - \theta_j)), \quad k = N_G + 1, \dots, N \end{aligned} \quad (T7.19)$$

We are trying to find the following information about the network:

- a. The angles for the voltage phasors at all buses except the swing bus (it is 0° at the swing bus), i.e., θ_k, k=2, ..., N
- b. The magnitudes for the voltage phasors at all type PQ buses, i.e., |V_k|, k=N_G+1, ..., N

Statements a and b imply that we have N-1 angle unknowns and N-N_G voltage magnitude unknowns, for a total number of unknowns of (N-1)+(N-N_G)=2N-N_G-1. Referring to the power flow equations, eq. (T7.19), we see that there are no other unknowns on the right-hand side besides voltage magnitudes and angles (the real and imaginary parts of the admittance values, G_{kj} and B_{kj}, are known, based on statement 1 above).

Number of equations= number of unknowns!

Thus we see that the number of equations having known left-hand side (injections) is the same as the number of unknown voltage magnitudes and angles. Therefore it is possible to solve the system of 2N-N_G-1 equations for the 2N-N_G-1 unknowns. However, we note from eq. (T7.19) that these equations are not linear, i.e., they are nonlinear equations. This nonlinearity comes from the fact that we have terms containing products of some of the unknowns and also terms containing trigonometric functions of some of the unknowns. Because of these nonlinearities, we are not able to put them directly into the familiar matrix form of "Ax=b" (where A is a matrix, x is the vector of unknowns, and b is a vector of constants) to obtain their solution. We must therefore resort to some other methods that are applicable for solving nonlinear equations. We describe such a method in Section T7.6. Before doing that, however, it may be helpful to more crisply formulate the exact problem that we want to solve.

Let's first define the vector of unknown variables. This we do in two steps. First, define the vector of unknown angles $\underline{\theta}$ (an underline beneath the variable means it is a vector or a matrix) and the vector of unknown voltage magnitudes $|\underline{V}|$.

$$\underline{\theta} = \begin{bmatrix} \theta_2 \\ \theta_3 \\ \vdots \\ \theta_N \end{bmatrix}, \quad |\underline{V}| = \begin{bmatrix} |V_{N_G+1}| \\ |V_{N_G+2}| \\ \vdots \\ |V_N| \end{bmatrix} \quad (T7.20)$$

Second, define the vector \underline{x} as the composite vector of unknown angles and voltage magnitudes.

Solution vector.

$$\underline{x} = \begin{bmatrix} \underline{\theta} \\ |\underline{V}| \end{bmatrix} = \begin{bmatrix} \theta_2 \\ \theta_3 \\ \vdots \\ \theta_N \\ |V_{N_G+1}| \\ |V_{N_G+2}| \\ \vdots \\ |V_N| \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \\ x_{N+1} \\ \vdots \\ x_{2N-N_G-1} \end{bmatrix} \quad (T7.21)$$

With this notation, we see that the right-hand sides of eqs. (T7.19) depend on the elements of the unknown vector \underline{x} . Expressing this dependence more explicitly, we rewrite eqs. (T7.19) as

Compact notation for PF eqs. LHS is a number; RHS is a function.

$$\begin{aligned} P_k &= P_k(\underline{x}), & k &= 2, \dots, N \\ Q_k &= Q_k(\underline{x}), & k &= N_G + 1, \dots, N \end{aligned} \quad (T7.22)$$

In eqs. (T7.22), P_k and Q_k are the specified injections (known constants) while the right-hand sides are functions of the elements in the unknown vector \underline{x} . Bringing the left-hand side over to the right-hand side, we have that

Put in form $f(\underline{x})=0$.

$$\begin{aligned} P_k(\underline{x}) - P_k &= 0, & k &= 2, \dots, N \\ Q_k(\underline{x}) - Q_k &= 0, & k &= N_G + 1, \dots, N \end{aligned} \quad (T7.23)$$

We now define a vector-valued function $\underline{f}(\underline{x})$ as:

Write out all equations.

$$\underline{f}(\underline{x}) = \begin{bmatrix} f_1(\underline{x}) \\ \vdots \\ f_{N-1}(\underline{x}) \\ \hline f_N(\underline{x}) \\ \vdots \\ f_{2N-N_G-1}(\underline{x}) \end{bmatrix} = \begin{bmatrix} P_2(\underline{x}) - P_2 \\ \vdots \\ P_N(\underline{x}) - P_N \\ \hline Q_{N_G+1}(\underline{x}) - Q_{N_G+1} \\ \vdots \\ Q_N(\underline{x}) - Q_N \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_N \\ \hline \Delta Q_{N_G+1} \\ \vdots \\ \Delta Q_N \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix} = \underline{0} \quad (T7.24)$$

Equation (T7.24) is in the form of $\underline{f}(\underline{x})=\underline{0}$, where $\underline{f}(\underline{x})$ is a vector-valued function and $\underline{0}$ is a vector of zeros; both $\underline{f}(\underline{x})$ and $\underline{0}$ are of dimension $(2N-N_G-1) \times 1$, which is also the dimension of the vector of unknowns, \underline{x} .

The mismatch vector.

We have also introduced nomenclature representing the *mismatch vector* in eq. (T7.24), as the vector of ΔP_k 's and ΔQ_k 's. This vector is used during the solution algorithm, which is iterative, to identify how good the solution is corresponding to any particular iteration. In the next section, we introduce this solution algorithm, which can be used to solve this kind of system of equations. The method is called the Newton-Raphson method.

Iterative methods required to solve this nonlinear algebraic system of equations.

T7.6 The Newton-Raphson Solution Procedure

There are two basic methods for solving the power flow problem: Gauss-Siedel (GS) and Newton-Raphson (NR). Both of these methods are *iterative root finding schemes*.

The GS and NR methods are often classified as root finding schemes because they are geared towards solving equations like $f(x)=0$ (or $\underline{f}(\underline{x})=0$). The solution to such an equation, call it x^* (or \underline{x}^*), is clearly a root of the function $f(x)$ (or $\underline{f}(\underline{x})$).

Method is guess, check, update, check, update, check, ...

The methods are called iterative because they require a series of successive approximations to the solutions. The procedure is generally as follows. First, guess a solution. Unless we are very fortunate, the guess will be, of course, wrong. So we determine an update to the “old” solution that moves to a “new” solution with the intention that the “new” solution is closer to the correct solution than was the “old” solution. A key aspect to this type of procedure is the way we obtain the update. If we can guarantee that the update is always improving the solution, such that the “new” solution is in fact always closer to the correct solution than the “old” solution, then such a procedure can be guaranteed to work if only we are willing to compute enough updates, i.e., if only we are willing to iterate enough times.

Initial guess is very important. If you guess poorly, you may get wrong solution or no solution.

Commercial grade power flow programs may make several different solutions procedures available, but almost all such programs will have available, minimally, the NR method. It is fair to say that the NR method has become the de-facto industry standard. The main reason for this is that the *convergence properties* of the NR scheme are very desirable when the initial, guessed solution is quite good, i.e., when it is chosen close to the correct solution. In the power flow problem, it is usually possible to make a good initial guess regarding the solution. One reason for this is that often, we may actually know the solution of a particular set of conditions because we have already gone through the solution procedure, and we want to resolve for a set of conditions that are almost the same as the previous ones, e.g., maybe remove one circuit or change the load level a little. In this case, we may utilize the previous solution as the initial guessed solution for the new conditions. This is sometimes referred to as a “hot” start. But even if we do not have a previous solution, we still may do very well with our guess. The reason for this is that the power flow problem is always solved with all quantities in per-unit. Because of the way we choose per-unit voltage bases, the per-unit voltages for all buses, under any reasonably normal condition, will be close to 1.0 per-unit. Of course, this tells us nothing about the angles, but it *is* something, and often it is enough to simply guess that all voltages are 1.0 per-unit and all angles are 0 degrees. This is sometimes called a “flat” start.

Divergence indicates bad initial guess or no solution. No solution may be due to stressed condition or bad data.

But what are “convergence properties” of a root finding method? There are basically two of them. One is *whether* the method will converge. The second one is *how fast* the method will converge. For NR, whether the method will converge depends on two things: how close the guessed solution is to the correct solution and the nature of the function close to the correct solution. If the guessed solution is close, and if the function is reasonably “smooth” close to the correct solution, then the NR will converge. Not only that, but it will converge *quadratically*. Quadratic convergence means that each iteration increases the accuracy of the solution by two decimal places. For example, if the correct solution for a particular problem is 0.123456789, and we guess 0.100000000, then the first iteration will yield 0.123xxxxxx, the second iteration will yield 0.12345xxxx, the third iteration will yield 0.1234567xx, and the correct solution will be obtained exactly on the fourth iteration.

In this module, we will not discuss the GS method, but the interested reader may find information about it in many texts on power systems analysis or in books on numerical methods. We will introduce the NR method with a simple illustration, obtained from [3].

Example T7.2

Consider the scalar function $f(x)=x^2-5x+4$. This function may be easily factored to find the roots as $x^*=4,1$.

Let us now illustrate how the NR method finds one of these roots. We first need the derivative: $f'(x)=2x-5$. Assume we are bad guessers, and try an initial guess of $x^{(0)}=6$. The following provides the first two iterations:

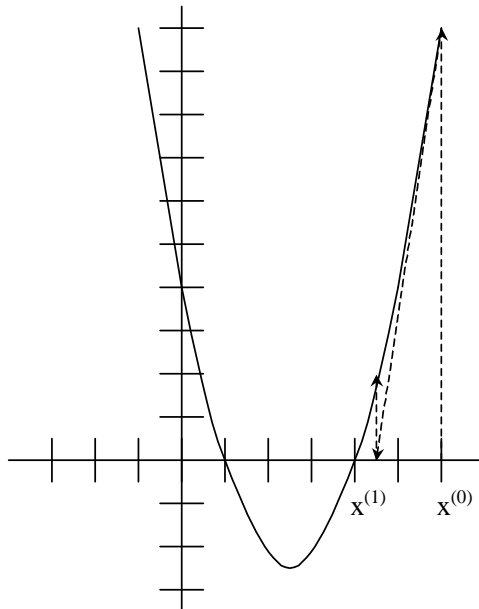
1. $f(x^{(0)})=f(6)=6^2-5(6)+4=10$
 2. $f'(x^{(0)})=f'(6)=2(6)-5=7$
 3. $\Delta x^{(0)} = -f(x^{(0)})/f'(x^{(0)}) = -10/7 = -1.429$
 4. $x^{(1)} = x^{(0)} + \Delta x^{(0)} = 6 + (-1.429) = 4.571$
-
1. $f(x^{(1)})=f(4.571)=2.03904$
 2. $f'(x^{(1)})=f'(4.571)=4.142$
 3. $\Delta x^{(1)} = -f(x^{(1)})/f'(x^{(1)}) = -2.03904/4.142 = -0.492284$
 4. $x^{(2)} = x^{(1)} + \Delta x^{(1)} = 4.571 + (-0.492284) = 4.0787$

One more iteration yields $x^{(3)}=4.002$. Note that by the third iteration, as it is getting very close to the correct solution, the algorithm has almost obtained quadratic convergence. Fig. T7.7 illustrates how the first solution $x^{(1)}$ is found from the initial guessed solution $x^{(0)}$ during the first iteration of this algorithm.

The NR algorithm is not smart enough to know which root you want, rather, it generally finds the closest root. This is another reason for making a good initial guess in regards to the solution. Fortunately, in the case of the power flow problem, alternative solutions are typically “far away” from initial guesses that have near-unity bus voltage magnitudes. On the other hand, it is possible for the solution to diverge, i.e., not to converge at all. This may occur if there is simply no solution, which is a case that engineers encounter frequently when studying highly stressed loading conditions served by weak transmission systems. It also might occur if the initial guessed solution is too far away from the correct solution. For this reason, “flat” starts encounter solution divergence more frequently than “hot” starts.

Numerical illustration of N-R procedure for a single variable root finding problem.

Graphical illustration of NR iteration 1 for a single variable root finding problem.



If your initial guess is 2.5, the derivative is 0, and the algorithm fails (corresponds to matrix singularity in the multi-dimensional case). If your initial guess is >2.5, you get the solution 4. If your initial guess is <2.5, you get the solution 1.

Fig. T7.7: Illustration of the first iteration of the Newton-Raphson algorithm

Next, we develop the NR update formula. We begin with the scalar case, where the update formula may be easily inferred from Example T7.2.

Newton Raphson for the Scalar Case:

Assume that we have guessed a solution $x^{(0)}$ to the problem $f(x)=0$. Then $f(x^{(0)})\neq 0$ because $x^{(0)}$ is just a guess. But there must be some $\Delta x^{(0)}$ which will make $f(x^{(0)} + \Delta x^{(0)})=0$. One way to study this problem is to expand the function $f(x)$ in a Taylor series, as follows:

TSE for
scalar case.

$$f(x^{(0)} + \Delta x^{(0)}) = f(x^{(0)}) + f'(x^{(0)})\Delta x^{(0)} + \frac{1}{2}f''(x^{(0)})(\Delta x^{(0)})^2 + \dots = 0 \quad (T7.25)$$

If the guess is a good one, then $\Delta x^{(0)}$ will be small, and if this is true, then $(\Delta x^{(0)})^2$ will be very small, and any higher order terms (h.o.t.) in eq. (T7.25), which will contain $\Delta x^{(0)}$ raised to even higher powers, will be infinitesimal. As a result, it is reasonable to approximate eq. (T7.25) as

Approximation
depends on a
good guess.

$$f(x^{(0)} + \Delta x^{(0)}) \approx f(x^{(0)}) + f'(x^{(0)})\Delta x^{(0)} = 0 \quad (T7.26)$$

Taking $f(x^{(0)})$ to the right hand side, we have

$$f'(x^{(0)})\Delta x^{(0)} = -f(x^{(0)}) \quad (T7.27)$$

We may easily solve eq. (T7.27) for $\Delta x^{(0)}$ according to:

Correction formula.

$$\Delta x^{(0)} = -\{f'(x^{(0)})\}^{-1} f(x^{(0)}) \quad (T7.28)$$

Because $f'(x^{(0)})$ in eq. (T7.28) is scalar, its inverse is very easily evaluated using simple division so that:

$$\Delta x^{(0)} = \frac{-f(x^{(0)})}{f'(x^{(0)})} \quad (T.29)$$

Equation (T7.28) provides the basis for the update formula to be used in the first iteration of the scalar NR method. This update formula is:

$$x^{(1)} = x^{(0)} + \Delta x^{(0)} = x^{(0)} + \frac{-f(x^{(0)})}{f'(x^{(0)})} \quad (T7.30)$$

and from eq. (T7.28), we may infer the update formula for any particular iteration as:

Update formula.

$$x^{(j+1)} = x^{(j)} + \Delta x^{(j)} = x^{(j)} + \frac{-f(x^{(j)})}{f'(x^{(j)})} \quad (T7.31)$$

Next we develop the update formula for the case where we have n equations and n unknowns. We call this the multidimensional case.

Newton Raphson for the Multidimensional Case:

Assume we have n nonlinear algebraic equations and n unknowns characterized by $\underline{f}(\underline{x})=0$, and that we have guessed a solution $\underline{x}^{(0)}$. Then $\underline{f}(\underline{x}^{(0)})\neq 0$ because $\underline{x}^{(0)}$ is just a guess. But there must be some $\Delta \underline{x}^{(0)}$ which will make $\underline{f}(\underline{x}^{(0)} + \Delta \underline{x}^{(0)})=0$. Again, we expand the function $\underline{f}(\underline{x})$ in a Taylor series, as follows:

$$\begin{aligned}
 f_1(\underline{x}^{(0)} + \Delta \underline{x}^{(0)}) &= f_1(\underline{x}^{(0)}) + f_1'(\underline{x}^{(0)})\Delta \underline{x}^{(0)} + \frac{1}{2}f_1''(\underline{x}^{(0)})(\Delta \underline{x}^{(0)})^2 + \dots = 0 \\
 f_2(\underline{x}^{(0)} + \Delta \underline{x}^{(0)}) &= f_2(\underline{x}^{(0)}) + f_2'(\underline{x}^{(0)})\Delta \underline{x}^{(0)} + \frac{1}{2}f_2''(\underline{x}^{(0)})(\Delta \underline{x}^{(0)})^2 + \dots = 0 \\
 &\vdots \\
 f_n(\underline{x}^{(0)} + \Delta \underline{x}^{(0)}) &= f_n(\underline{x}^{(0)}) + f_n'(\underline{x}^{(0)})\Delta \underline{x}^{(0)} + \frac{1}{2}f_n''(\underline{x}^{(0)})(\Delta \underline{x}^{(0)})^2 + \dots = 0
 \end{aligned}
 \tag{T7.32}$$

Equations (T7.32) may be written more compactly as

$$\underline{f}(\underline{x}^{(0)} + \Delta \underline{x}^{(0)}) = \underline{f}(\underline{x}^{(0)}) + \underline{f}'(\underline{x}^{(0)})\Delta \underline{x}^{(0)} + \frac{1}{2}\underline{f}''(\underline{x}^{(0)})(\Delta \underline{x}^{(0)})^2 + \dots = \underline{0}
 \tag{T7.33}$$

Assuming the guess is a good one such that $\Delta \underline{x}^{(0)}$ is small, then the higher order terms are also small and we can write

What is $\underline{f}'(\underline{x})$?

$$\underline{f}(\underline{x}^{(0)} + \Delta \underline{x}^{(0)}) = \underline{f}(\underline{x}^{(0)}) + \underline{f}'(\underline{x}^{(0)})\Delta \underline{x}^{(0)} = \underline{0}
 \tag{T7.34}$$

One reasonable question to ask at this point is: “Just what is $\underline{f}'(\underline{x}^{(0)})$?” That is, what is the derivative of a vector-valued function of a vector? Since we have n functions and n variables, we could compute a derivative for each individual function with respect to each individual unknown, like $\partial f_k(\underline{x})/\partial x_j$, which gives the derivative of the k^{th} function with respect to the j^{th} unknown. Thus, there will be a number of such derivatives equal to the product of the number of functions by the number of unknowns, in this case, $n \times n$. Thus, it is convenient to store all of these derivatives in a matrix. This matrix has become quite well-known as the *Jacobian* matrix, and it is often denoted using the letter \underline{J} . But how should the $n \times n$ derivatives be stored in this matrix \underline{J} ?

The rows of \underline{J} should be ordered in the same order as the functions, that is, the k^{th} row should contain the derivatives of the k^{th} functions. In eq. (T7.34), since the product $\underline{f}'(\underline{x}^{(0)}) \Delta \underline{x}^{(0)}$ must provide a correction to the function $\underline{f}(\underline{x}^{(0)} + \Delta \underline{x}^{(0)})$, i.e., since $\Delta \underline{f}(\underline{x}^{(0)}) = \underline{f}'(\underline{x}^{(0)}) \Delta \underline{x}^{(0)}$, it must be the case that any row of the matrix \underline{J} must be ordered so that the term in the j^{th} column contains a derivative with respect to the j^{th} unknown of the vector \underline{x} .

The reasoning in the last paragraph suggests that we write the Jacobian matrix as:

The Jacobian matrix...

$$\underline{J} = \begin{bmatrix} \frac{\partial f_1(\underline{x}^{(0)})}{\partial x_1} & \frac{\partial f_1(\underline{x}^{(0)})}{\partial x_2} & \dots & \frac{\partial f_1(\underline{x}^{(0)})}{\partial x_n} \\ \frac{\partial f_2(\underline{x}^{(0)})}{\partial x_1} & \frac{\partial f_2(\underline{x}^{(0)})}{\partial x_2} & \dots & \frac{\partial f_2(\underline{x}^{(0)})}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n(\underline{x}^{(0)})}{\partial x_1} & \frac{\partial f_n(\underline{x}^{(0)})}{\partial x_2} & \dots & \frac{\partial f_n(\underline{x}^{(0)})}{\partial x_n} \end{bmatrix}
 \tag{T7.35}$$

In eq. (T7.34), taking $\underline{f}(\underline{x}^{(0)})$ to the right hand side, we have

$$\underline{f}'(\underline{x}^{(0)})\Delta \underline{x}^{(0)} = -\underline{f}(\underline{x}^{(0)})
 \tag{T7.36}$$

or, in terms of the Jacobian matrix \underline{J} , we have:

The correction formula for first iteration.

$$\underline{J}\underline{\Delta x}^{(0)} = -\underline{f}(\underline{x}^{(0)}) \tag{T7.37}$$

Solving eq. (T7.37) for $\underline{\Delta x}^{(0)}$, we have:

$$\underline{\Delta x}^{(0)} = -\{\underline{f}'(\underline{x}^{(0)})\}^{-1} \underline{f}(\underline{x}^{(0)}) = -\underline{J}^{-1} \underline{f}(\underline{x}^{(0)}) \tag{T7.38}$$

Equation (T7.38) provides the basis for the update formula to be used in the first iteration of the multi-dimensional case. This update formula is:

$$\underline{x}^{(1)} = \underline{x}^{(0)} + \underline{\Delta x}^{(0)} = \underline{x}^{(0)} - \underline{J}^{-1} \underline{f}(\underline{x}^{(0)}) \tag{T7.39}$$

and from eq. (T7.39), we may infer the update formula for any particular iteration as:

The update formula.

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \underline{\Delta x}^{(i)} = \underline{x}^{(i)} - \underline{J}^{-1} \underline{f}(\underline{x}^{(i)}) \tag{T7.40}$$

For problems of relatively small dimension, where the inverse of the Jacobian is easily obtainable, eq. (T7.40) is an appropriate update formula. In general, however, it is a good rule, in programming, to always avoid matrix inversion if at all possible, because for high-dimension problems, as is usually the case for large scale power networks, matrix inversion is *very* time consuming. We always want to avoid matrix inversion if possible, and it usually is.

To see how to avoid matrix inversion, we will state the update formula a little differently. To do this, we write eq. (T7.40) as

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \underline{\Delta x}^{(i)} \tag{T7.41}$$

where $\underline{\Delta x}^{(i)}$ is found from

$$-\underline{J}\underline{\Delta x}^{(i)} = \underline{f}(\underline{x}^{(i)}) \tag{T7.42a}$$

Equation (T7.42a) is a very simple relation. Observing that \underline{J} is just a constant $n \times n$ matrix, $\underline{\Delta x}^{(i)}$ is an $n \times 1$ vector of unknowns, and $\underline{f}(\underline{x}^{(i)})$ is an $n \times 1$ vector of knowns, we see that eq. (T7.42) is just the linear matrix equation

$$\underline{A} \underline{z} = \underline{b} \tag{T7.42b}$$

There are a very many methods of solving (T7.42b). We will cover this topic later in these notes. First, however, let's illustrate the Newton-Raphson procedure for a multi-dimensional case. We will use the simplest multi-dimensional case we can, a two-variable problem.

Example T7.3

Solve the following two equations algebraically and using NR:

$$\begin{aligned} 2x_1^2 + x_1x_2 - x_1 - 2 &= 0, \\ x_1^2 - x_2 &= 0 \end{aligned}$$

The steps for the algebraic solution are to first solve both equations for x_2 , resulting in $x_2 = (-2x_1^2 + x_1 + 2)/x_1$ and $x_2 = x_1^2$. Equating these two expressions for x_2 , and manipulating, results in a cubic $x_1^3 + 2x_1^2 - x_1 - 2 = 0$. This expression may be factored as: $(x_1 - 1)(x_1 + 1)(x_1 + 2) = 0$, and we see that the solutions to the cubic in x_1 are 1, -1 and -2. Plugging these values for x_1 back into either expression for x_2 yields, respectively, 1, 1, and 4, and therefore there are three solutions to the original problem; they are: $(x_1, x_2) = (1, 1), (-1, 1), (-2, 4)$.

But to avoid matrix inversion, we write the update formula like this → where the correction is computed from → this

A system of linear equations!

Now let's solve this same problem using NR?

Define the functions $f_1(x_1, x_2) = 2x_1^2 + x_1x_2 - x_1 - 2$ and $f_2(x_1, x_2) = x_1^2 - x_2$. Then the Jacobian matrix is:

$$\underline{J} = \begin{bmatrix} \frac{\partial f_1(x_1, x_2)}{\partial x_1} & \frac{\partial f_1(x_1, x_2)}{\partial x_2} \\ \frac{\partial f_2(x_1, x_2)}{\partial x_1} & \frac{\partial f_2(x_1, x_2)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 + x_2 - 1 & x_1 \\ 2x_1 & -1 \end{bmatrix}$$

Let's act like we do not know the solution and guess at $(x_1^{(0)}, x_2^{(0)}) = (0.9, 1.1)$. Then the Jacobian \underline{J} , evaluated at this guessed solution, is

$$\underline{J}(x_1^{(0)}, x_2^{(0)}) = \begin{bmatrix} 2x_1 + x_2 - 1 & x_1 \\ 2x_1 & -1 \end{bmatrix}_{\underline{x}^{(0)}} = \begin{bmatrix} 1.9 & 0.9 \\ 1.8 & -1 \end{bmatrix}$$

Inverting the Jacobian results in:

$$\underline{J}^{-1} = \begin{bmatrix} 1.9 & 0.9 \\ 1.8 & -1 \end{bmatrix}^{-1} = \frac{1}{-3.52} \begin{bmatrix} -1 & -0.9 \\ -1.8 & 1.9 \end{bmatrix} = \begin{bmatrix} 0.28409 & 0.25568 \\ 0.51136 & -0.53977 \end{bmatrix}$$

We also need to evaluate:

$$\underline{f}(\underline{x}^{(0)}) = \begin{bmatrix} f_1(\underline{x}^{(0)}) \\ f_2(\underline{x}^{(0)}) \end{bmatrix} = \begin{bmatrix} 2x_1 + x_1x_2 - x_1 - 2 \\ x_1^2 - x_2 \end{bmatrix}_{\underline{x}^{(0)}=(0.9,1.1)} = \begin{bmatrix} -0.11 \\ -0.29 \end{bmatrix}$$

We can now update the solution using eq. (T7.40), as

$$\begin{aligned} \underline{x}^{(1)} &= \underline{x}^{(0)} + \Delta \underline{x}^{(0)} = \underline{x}^{(0)} - \underline{J}^{-1} \underline{f}(\underline{x}^{(0)}) \\ &= \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0.9 \\ 1.1 \end{bmatrix} - \begin{bmatrix} 0.28409 & 0.25568 \\ 0.51136 & -0.53977 \end{bmatrix} \begin{bmatrix} -0.11 \\ -0.29 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 1.1 \end{bmatrix} - \begin{bmatrix} -0.105397 \\ 0.100284 \end{bmatrix} = \begin{bmatrix} 1.005397 \\ 0.999716 \end{bmatrix} \end{aligned}$$

We see that the first update results in a solution that is very close to the actual solution of (1,1). This good performance is due to the fact that we made a good initial guess. The student should repeat the above procedure, but try starting from other points, e.g., (-0.9,1.1), (-1.9,4.1), and (0,1.1), using two iterations each time. Writing a simple program will greatly reduce the effort.

One of these two stopping criterion may be applied.

Type 1: $\Delta \underline{x}$
Type 2: $\underline{f}(\underline{x})$

Type 2 is "power mismatch" and gives indication of where convergence problems may lie.

In general, of course, we usually need to iterate several times in order to obtain a satisfactory solution. How many times is enough? The NR algorithm must employ a *stopping criterion* in order to determine when the solution is satisfactory. There are two ways to do this.

- Type 1 stopping criterion: Test the maximum change in the solution elements from one iteration to the next, and if this maximum change is smaller than a certain predefined tolerance, then stop. This means to compare the maximum absolute value of elements in $\Delta \underline{x}$ against a small number, call it ϵ_1 . In example (T7.3), $\Delta \underline{x} = [-0.105397, 0.100284]^T$, so the maximum absolute value of elements in $\Delta \underline{x}$ is 0.105397. If we had $\epsilon_1 = 0.15$, we could stop. But if we had $\epsilon_1 = 0.05$, we would need to continue to the next iteration.
- Type 2 stopping criterion: Test the maximum value in the function elements of the most current iteration $\underline{f}(\underline{x})$, and if this maximum value of elements in $\underline{f}(\underline{x})$ is smaller than a certain predefined tolerance, then stop. This means to compare the maximum absolute value of elements in $\underline{f}(\underline{x})$ against a small number, call it ϵ_1 . In example (T7.3), $\underline{f}(\underline{x}) = [-0.11, -0.29]^T$, so the maximum absolute value of elements in $\underline{f}(\underline{x})$ is 0.29. If we had $\epsilon_1 = 0.3$, we could stop. But if we had $\epsilon_1 = 0.2$, we would need to continue to the next iteration. This is the most common stopping criterion for power flow solutions, and the value of each element in the function is referred to as the "power mismatch" for the bus

corresponding to the function. For type PQ buses, we test both real and reactive power mismatches. For type PV buses, we test only real power mismatches.

T7.7 Application of NR to Power Flow Solution

Let's revisit the power flow problem outlined in Section T7.5, in light of the NR solution procedure described in Section T7.6. We desire to solve eq. (T7.24), with the vector of unknowns are given by eq. (T7.21) and the functions are in the form of eq. (T7.19). These equations are repeated here for convenience:

Mismatch vector.

$$\underline{f}(\underline{x}) = \begin{bmatrix} f_1(\underline{x}) \\ \vdots \\ f_{N-1}(\underline{x}) \\ \hline f_N(\underline{x}) \\ \vdots \\ f_{2N-N_G-1}(\underline{x}) \end{bmatrix} = \begin{bmatrix} P_2(\underline{x}) - P_2 \\ \vdots \\ P_N(\underline{x}) - P_N \\ \hline Q_{N_G+1}(\underline{x}) - Q_{N_G+1} \\ \vdots \\ Q_N(\underline{x}) - Q_N \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_N \\ \hline \Delta Q_{N_G+1} \\ \vdots \\ \Delta Q_N \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \underline{0} \quad (\text{T7.24})$$

Solution vector.

$$\underline{x} = \begin{bmatrix} \underline{\theta} \\ \underline{V} \end{bmatrix} = \begin{bmatrix} \theta_2 \\ \theta_3 \\ \vdots \\ \theta_N \\ |V_{N_G+1}| \\ |V_{N_G+2}| \\ \vdots \\ |V_N| \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \\ x_{N+1} \\ \vdots \\ x_{2N-N_G-1} \end{bmatrix} \quad (\text{T7.21})$$

PF equations to solve.

$$P_k = \sum_{j=1}^N |V_k| |V_j| (G_{kj} \cos(\theta_k - \theta_j) + B_{kj} \sin(\theta_k - \theta_j)), \quad k = 2, \dots, N \quad (\text{T7.19})$$

$$Q_k = \sum_{j=1}^N |V_k| |V_j| (G_{kj} \sin(\theta_k - \theta_j) - B_{kj} \cos(\theta_k - \theta_j)), \quad k = N_G + 1, \dots, N$$

The solution update formula is given by eq. (T7.40), repeated here for convenience:

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \Delta \underline{x}^{(i)} = \underline{x}^{(i)} - \underline{J}^{-1} \underline{f}(\underline{x}^{(i)}) \quad (\text{T7.40})$$

Clearly, an essential step in applying NR to the power flow problem is to enable calculation of the Jacobian elements, given for the general case by eq. (T7.35) as

Jacobian.
n×n differentiations?

Observation:
Only 2 kinds of equations (P, Q)
Only 2 kinds of unknowns (|V|, θ)

$$\underline{J} = \begin{bmatrix} \frac{\partial f_1(\underline{x}^{(0)})}{\partial x_1} & \frac{\partial f_1(\underline{x}^{(0)})}{\partial x_2} & \dots & \frac{\partial f_1(\underline{x}^{(0)})}{\partial x_n} \\ \frac{\partial f_2(\underline{x}^{(0)})}{\partial x_1} & \frac{\partial f_2(\underline{x}^{(0)})}{\partial x_2} & \dots & \frac{\partial f_2(\underline{x}^{(0)})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(\underline{x}^{(0)})}{\partial x_1} & \frac{\partial f_n(\underline{x}^{(0)})}{\partial x_2} & \dots & \frac{\partial f_n(\underline{x}^{(0)})}{\partial x_n} \end{bmatrix} \quad (T7.35)$$

Evaluation of these elements is facilitated by the recognition, from eq. (T7.24), that there are only two kinds of equations (real power equations and reactive power equations), and from eq. (T7.21), that there are only two kinds of unknowns (voltage angle unknowns and voltage magnitude unknowns). Therefore, there are only four basic types of derivatives in the Jacobian. We denote four sub-matrices corresponding to these four basic types of derivatives as $\underline{J}^{P\theta}$, $\underline{J}^{Q\theta}$, \underline{J}^{PV} , \underline{J}^{QV} , where the first superscript indicates the type of equation we differentiate, and the second superscript indicates the unknown with respect to which we differentiate. Therefore,

Study dimensionality of each sub-matrix.

Compact expression of Jacobian using above observation.

Still n×n differentiations, but are differentiations of same form?

$$\tilde{\underline{J}}^{(2N-1-N_G) \times (2N-1-N_G)} = \begin{bmatrix} \overbrace{\underline{J}^{P\theta}}^{(N-1) \times (N-1)} & \overbrace{\underline{J}^{PV}}^{(N-1) \times (N-N_G)} \\ \overbrace{\underline{J}^{Q\theta}}^{(N-N_G) \times (N-1)} & \overbrace{\underline{J}^{QV}}^{(N-N_G) \times (N-N_G)} \end{bmatrix} \quad (T7.43)$$

The numbers above each sub-matrix in eq. (T7.43) indicate its dimensions, which can be inferred by identifying the number of equations of that type (the number of rows of the sub-matrix) and the number of unknowns of that type (the number of columns of the sub-matrix). We may then identify an individual element of each sub-matrix as:

Subscripts j, k are not row & column number due to (a) removing pf eq & θ for bus 1; (b) stacking Q eqs & V-variables on top of P eqs & θ variables.

Subscripts refer to bus numbers, not matrix location. Software needs to track relation between bus numbers & matrix location

$$J_{jk}^{P\theta} = \frac{\partial P_j}{\partial \theta_k} \quad J_{jk}^{Q\theta} = \frac{\partial Q_j}{\partial \theta_k} \quad J_{jk}^{PV} = \frac{\partial P_j}{\partial |V_k|} \quad J_{jk}^{QV} = \frac{\partial Q_j}{\partial |V_k|} \quad (T7.44)$$

Notationally, observe that the element $J_{jk}^{P\theta}$ is *not* the element in row j, column k of the submatrix $\underline{J}^{P\theta}$, rather it is the derivative of the real power injection equation for bus j with respect to the angle of bus k. Since the swing bus is numbered 1, the Jacobian matrix will have $J_{22}^{P\theta}$ as the element in row 1, column 1. The situation is similar for the other submatrices.

The update equation (T7.42a) is repeated here for convenience:

$$-\underline{J}\Delta\underline{x}^{(i)} = \underline{f}(\underline{x}^{(i)}) \quad (T7.42a)$$

Multiplying both sides by -1, we obtain

$$\underline{J}\Delta\underline{x}^{(i)} = -\underline{f}(\underline{x}^{(i)}) \quad (T7.42c)$$

Using (T7.21), (T7.24), and (T7.43) we can write (T7.42c) as

$$\begin{bmatrix} \underline{J}^{P\theta} & \underline{J}^{PV} \\ \underline{J}^{Q\theta} & \underline{J}^{QV} \end{bmatrix} \begin{bmatrix} \Delta\underline{\theta} \\ \Delta\underline{|V|} \end{bmatrix} = -\begin{bmatrix} \Delta\underline{P} \\ \Delta\underline{Q} \end{bmatrix} \quad (T7.42d)$$

From (T7.42d) we observe that

$$\begin{aligned} \underline{J}^{P\theta} \Delta \underline{\theta} + \underline{J}^{PV} \Delta \underline{V} &= -\Delta \underline{P} \\ \underline{J}^{Q\theta} \Delta \underline{\theta} + \underline{J}^{QV} \Delta \underline{V} &= -\Delta \underline{Q} \end{aligned} \tag{T7.42e}$$

To get the needed derivatives, it is helpful to more explicitly write out the functions of eq. (T7.24). They are:

Mismatch vector.

$$\underline{f}(\underline{x}) = \begin{bmatrix} P_2(\underline{x}) - P_2 \\ \vdots \\ P_N(\underline{x}) - P_N \\ \hline Q_{N_G+1}(\underline{x}) - Q_{N_G+1} \\ \vdots \\ Q_N(\underline{x}) - Q_N \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^N |V_2| |V_j| (G_{2j} \cos(\theta_2 - \theta_j) + B_{2j} \sin(\theta_2 - \theta_j)) - P_2 \\ \vdots \\ \sum_{j=1}^N |V_N| |V_j| (G_{Nj} \cos(\theta_N - \theta_j) + B_{Nj} \sin(\theta_N - \theta_j)) - P_N \\ \hline \sum_{j=1}^N |V_{N_G+1}| |V_j| (G_{N_G+1,j} \sin(\theta_{N_G+1} - \theta_j) + B_{N_G+1,j} \sin(\theta_{N_G+1} - \theta_j)) - P_{N_G+1} \\ \vdots \\ \sum_{j=1}^N |V_N| |V_j| (G_{N,j} \sin(\theta_N - \theta_j) + B_{N,j} \sin(\theta_N - \theta_j)) - P_N \end{bmatrix} \tag{T7.45}$$

So each of the four sub-matrices of eq. (T7.43) has elements given by the expressions of eq. (T7.44), respectively. These expressions are evaluated by taking the appropriate derivatives of the functions in eq. (T7.45). One might think that this represents a formidable problem, since, based on eq. (T7.43), we have $(2N-1-N_G) \times (2N-1-N_G)$ elements in the Jacobian and therefore the same number of derivatives to evaluate. A typical power flow model for a US control area might have 5000 nodes ($N=5000$) and 1000 generators ($N_G=1000$), resulting in a 9898×9898 Jacobian matrix containing 97,970,404 elements, with each element requiring a differentiation of a function like those represented in eq. (T7.45). For a power flow model having 50000 nodes and 5000 generators, the dimension is 94998×94998 , giving 9,024,600,000 elements.

Fortunately, all of the derivatives can be expressed by one of just a few differentiations. At first glance, one might think that there would be four differentiations, one for each sub-matrix. However, for each sub-matrix, the off-diagonal terms, with $j \neq k$, are expressed differently than the diagonal terms, with $j=k$. Therefore, there are eight differentiations to perform. The student should attempt to obtain a few of these expressions. In doing so, the following tips are helpful.

- Before differentiating, it is helpful to pull out from the summation the term that corresponds to the bus injection being computed.
- When differentiating a sum of terms with respect to a particular unknown, the resulting derivative will be non-zero only for those terms in which the unknown appears.
- When differentiating with respect to the angles, the chain rule must be properly applied to account for the derivatives of the trigonometric functions and the arguments of those trigonometric functions.
- Each of the functions appear in the form of $f(\underline{x})=g(\underline{x})-A$. Because A is a constant (represented by P_2, \dots, P_N and Q_{N_G+1}, \dots, Q_N in eq. (T7.45)), it has no effect on the resulting derivatives.

The resulting expressions are given below.

$$J_{jk}^{P\theta} = \frac{\partial P_j(\underline{x})}{\partial \theta_k} = |V_j| |V_k| (G_{jk} \sin(\theta_j - \theta_k) - B_{jk} \cos(\theta_j - \theta_k)) \tag{T7.47}$$

$$J_{jj}^{P\theta} = \frac{\partial P_j(\underline{x})}{\partial \theta_j} = -Q_j(\underline{x}) - B_{jj} |V_j|^2 \tag{T7.48}$$

$$J_{jk}^{Q\theta} = \frac{\partial Q_j(\underline{x})}{\partial \theta_k} = -|V_j| |V_k| (G_{jk} \cos(\theta_j - \theta_k) + B_{jk} \sin(\theta_j - \theta_k)) \tag{T7.49}$$

98 million differentiations for a 5000 bus model.
9 billion differentiations for a 50000 bus model.

8 basic derivative expressions: for each of 4 submatrices, we need diagonal & off-diagonal expressions.

Observe T7.47 & T7.53 are similar. These are off-diagonal elements in $J^{P\theta}$ & $J^{Q\theta}$, bolded below.

$$\begin{bmatrix} \underline{J}^{P\theta} & \underline{J}^{PV} \\ \underline{J}^{Q\theta} & \underline{J}^{QV} \end{bmatrix} \begin{bmatrix} \Delta \underline{\theta} \\ \Delta \underline{V} \end{bmatrix} = - \begin{bmatrix} \Delta \underline{P} \\ \Delta \underline{Q} \end{bmatrix}$$

How many elements is this? Can we do something smart here?

$$J_{jj}^{Q\theta} = \frac{\partial Q_j(\underline{x})}{\partial \theta_k} = P_j(\underline{x}) - G_{jj}|V_j|^2 \tag{T7.50}$$

$$J_{jk}^{PV} = \frac{\partial P_j(\underline{x})}{\partial |V_k|} = |V_j| (G_{jk} \cos(\theta_j - \theta_k) + B_{jk} \sin(\theta_j - \theta_k)) \tag{T7.51}$$

$$J_{jj}^{PV} = \frac{\partial P_j(\underline{x})}{\partial |V_j|} = \frac{P_j(\underline{x})}{|V_j|} + G_{jj}|V_j| \tag{T7.52}$$

$$J_{jk}^{QV} = \frac{\partial Q_j(\underline{x})}{\partial |V_k|} = |V_j| (G_{jk} \sin(\theta_j - \theta_k) - B_{jk} \cos(\theta_j - \theta_k)) \tag{T7.53}$$

$$J_{jj}^{QV} = \frac{\partial Q_j(\underline{x})}{\partial |V_j|} = \frac{Q_j(\underline{x})}{|V_j|} - B_{jj}|V_j| \tag{T7.54}$$

Observe T7.49 & T7.51 are similar. These are off-diagonal elements in $J^{Q\theta}$ & J^{PV} , bolded below.

$$\begin{bmatrix} J^{P\theta} & J^{PV} \\ J^{Q\theta} & J^{QV} \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = - \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$$

How many elements is this? Can we do something smart here?

We are now in a position to provide the algorithm for using NR to solve the power flow problem. Before doing so, it is helpful to more explicitly define the mismatch vector, from eq. (T7.24) or (T7.45) as:

$$\underline{f}(\underline{x}) = \begin{bmatrix} f_1(\underline{x}) \\ \vdots \\ f_{N-1}(\underline{x}) \\ \hline f_N(\underline{x}) \\ \vdots \\ f_{2N-N_g-1}(\underline{x}) \end{bmatrix} = \begin{bmatrix} P_2(\underline{x}) - P_2 \\ \vdots \\ P_N(\underline{x}) - P_N \\ \hline Q_{N_g+1}(\underline{x}) - Q_{N_g+1} \\ \vdots \\ Q_N(\underline{x}) - Q_N \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_N \\ \hline \Delta Q_{N_g+1} \\ \vdots \\ \Delta Q_N \end{bmatrix} = \begin{bmatrix} \Delta P \\ \hline \Delta Q \end{bmatrix} = \underline{0} \tag{T7.53}$$

The NR algorithm, for application to the power flow problem, is:

1. Specify:
 - All admittance data
 - P_d and Q_d for all buses
 - P_g and $|V|$ for all PV buses
 - $|V|$ for swing bus, with $\theta=0^\circ$
2. Let the iteration counter $j=1$. Use one of the following to guess the initial solution.
 - Flat Start: $V_k=1.0 \angle 0^\circ$ for all buses.
 - Hot Start: Use the solution to a previously solved case for this network.
3. Compute the mismatch vector for $\underline{x}^{(j)}$, denoted as $\underline{f}(\underline{x})$ in eq. (T7.24) and eq. (T7.45). In what follows, we denote elements of the mismatch vector as ΔP_k and ΔQ_k corresponding to the real and reactive power mismatch, respectively, for the k^{th} bus (which would not be the k^{th} element of the mismatch vector for two reasons: one reason pertains to the swing bus and the other reason to the fact that for type PQ buses, there are two equations per bus and not one – see boxed comments next to eq. T7.44). This computation will also result in all necessary calculated real and reactive power injections.
4. Perform the following stopping criterion tests:
 - If $|\Delta P_k| < \epsilon_p$ for all type PQ and PV buses and
 - If $|\Delta Q_k| < \epsilon_Q$ for all type PQ buses,
 - Then go to step 6
 - Otherwise, go to step 5.
5. Find an improved solution as follows:
 - Evaluate the Jacobian \underline{J} at $\underline{x}^{(j)}$. Denote this Jacobian as $\underline{J}^{(j)}$
 - Solve for $\Delta \underline{x}^{(j)}$ from:

Flat start/hot start: common industry terms

Compute mismatch vector.

Check stopping criterion

Factorization: method to solve linear simultaneous equations. We will look a little more at this issue.

$$\underline{J}^{(j)} \Delta \underline{x}^{(j)} = - \begin{bmatrix} \underline{\Delta P} \\ \text{-----} \\ \underline{\Delta Q} \end{bmatrix} \quad \text{or} \quad \Delta \underline{x}^{(j)} = - [\underline{J}^{(j)}]^{-1} \begin{bmatrix} \underline{\Delta P} \\ \text{-----} \\ \underline{\Delta Q} \end{bmatrix}$$

where we must use **factorization** with the left equation if the system is large, but if the system is not large, we may use the right hand equation.

- Compute the updated solution vector as $\underline{x}^{(j+1)} = \underline{x}^{(j)} + \Delta \underline{x}^{(j)}$.
 - Return to step 3 with $j=j+1$.
6. Stop.

The above algorithm is applicable as long as all PV buses remain within their reactive limits. To account for generator reactive limits, we must modify the algorithm so that, **at each iteration**, we check to ensure PV bus reactive generation is within its limits (see Section T7.1 regarding modeling of reactive limits). In this case, steps 1-4 remain exactly as given above, but we need a new step 5 and 6, as follows:

5. Check reactive limits for all generator buses as follows:

a. For all type PV buses, perform the following test:

- If $Q_{gk} > Q_{gk,max}$, then
 → $Q_{gk} = Q_{gk,max}$ and CHANGE bus k to a type PQ bus (see step 6a)
- If $Q_{gk} < Q_{gk,min}$, then
 → $Q_{gk} = Q_{gk,min}$ and CHANGE bus k to a type PQ bus (see step 6b)

b. For all type PQ generator buses, perform the following test:

- If $Q_{gk} = Q_{gk,max}$ and $|V_k| > |V_{k,set}|$ or if $Q_{gk} = Q_{gk,min}$ and $|V_k| < |V_{k,set}|$, then
 → CHANGE this bus back to a type PV bus (see step 6b)

6. If there were no CHANGES in Step 5, then stop. If there were one or more CHANGES in step 5, then modify the solution vector and the mismatch vector as follows:

a. For each CHANGE made in step 5-a (changing a PV bus to a PQ bus):

- $N_G = N_G - 1$
- Include the variable ΔV_k to the vector $\Delta \underline{x}$ and the variable V_k to the vector \underline{x} .
- Include the reactive equation corresponding to bus k to the vector $\underline{f}(\underline{x})$.
- Modify the Jacobian by including a column to \underline{J}^{PV} and including a row to \underline{J}^{Q0} and \underline{J}^{QV} .

b. For each CHANGE made in Step 5-b (changing a PQ gen bus back to a PV bus):

- $N_G = N_G + 1$
- Remove the variable ΔV_k to the vector $\Delta \underline{x}$ and the variable V_k from the vector \underline{x} .
- Remove the reactive equation corresponding to bus k from the vector $\underline{f}(\underline{x})$.
- Modify the Jacobian by removing a column to \underline{J}^{PV} and removing a row from \underline{J}^{Q0} and \underline{J}^{QV} .

After modifications have been made for all CHANGES, go back to Step 4.

When the algorithm stops, then all line flows may be computed using

$$S_{jk} = V_j I_{jk}^* = V_j [V_j - V_k]^* y_{jk}$$

Example T7.4 [5] (used with permission of V. Vittal)

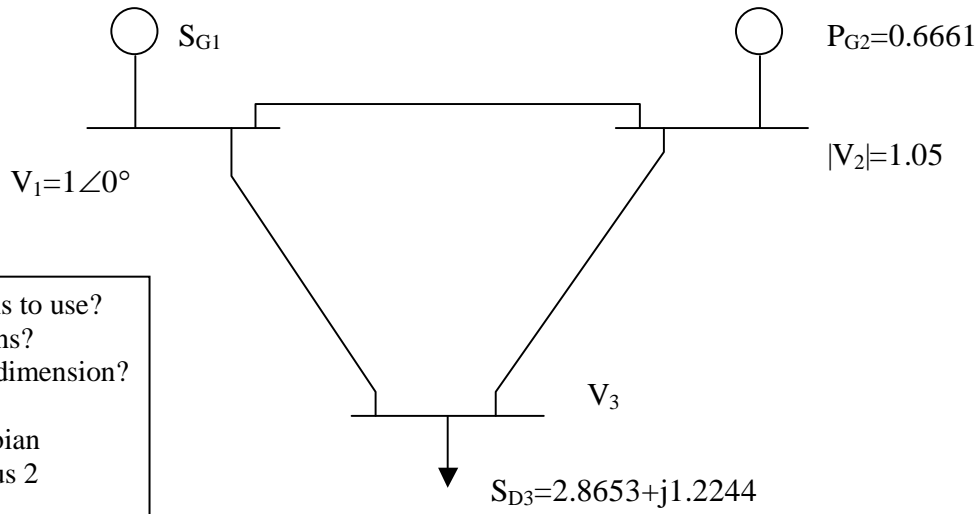
Find θ_2 , $|V_3|$, θ_3 , S_{G1} , and Q_{G2} for the system shown in Fig. T7.8. In the transmission elements are capacitors with an admittance $y_c = j0.01$, while all the series elements are inductors with an impedance of $z_L = j0.1$.

Another way is to let algorithm converge, check limits, make changes, & repeat. But checking every iteration provides increased convergence robustness.

Is there a PQ gen-bus, which was a PV bus, that should be now be PV again?

Is there a PV bus that should be PQ?

Does this equation account for shunt elements or line charging? Should it?



What are equations to use?
 What are unknowns?
 What is Jacobian dimension?

 What would Jacobian dimension be if bus 2 changed to PQ?

Fig. T7.8: Three Bus System for Example T7.4

Solution: The admittance matrix for the system shown in Fig. E10.6 is given by

$$\underline{Y} = \begin{bmatrix} -j19.98 & j10 & j10 \\ j10 & -j19.98 & j10 \\ j10 & j10 & -j19.98 \end{bmatrix}$$

Observe: (1) no real parts due to neglect of resistance; (2) diagonals are negative (inductive admittance), (3) off-diagonals are positive (negated inductive admittance)

Bus 1 is the swing bus. Bus 2 is a PV bus. Bus 3 is a PQ bus. We use the NR method in the solution. The unknown variables are θ_2 , θ_3 , and $|V_3|$. Thus, we will need three equations, and the Jacobian is a 3 x 3 matrix.

We first write eq. (T7.45) for the case at hand, putting in the known values of $|V_1|$, $|V_2|$, θ_1 , and the B_{ij} 's. Note that since we have neglected line resistance in the problem statement, all G_{ij} 's are zero.

$$\begin{aligned} P_2(\mathbf{x}) &= |V_2||V_1|B_{21}\sin(\theta_2 - \theta_1) + |V_2||V_3|B_{23}\sin(\theta_2 - \theta_3) \\ &= 10.5\sin\theta_2 + 10.5|V_3|\sin(\theta_2 - \theta_3) \end{aligned} \tag{T7.54a}$$

$$\begin{aligned} P_3(\mathbf{x}) &= |V_3||V_1|B_{31}\sin(\theta_3 - \theta_1) + |V_3||V_2|B_{32}\sin(\theta_3 - \theta_2) \\ &= 10.0|V_3|\sin\theta_3 + 10.5|V_3|\sin(\theta_3 - \theta_2) \end{aligned} \tag{T7.54b}$$

The equation for $Q_2(\underline{x})$ will not help since we do not know the reactive injection for bus 2, and its inclusion would bring in the reactive injection on the left-hand side as an additional unknown. But this loss of an equation is compensated by the fact that we know $|V_2|$ (and this will always be the case for a type PV bus). So we do not need to write the equation for $Q_2(\underline{x})$. Yet, because bus 3 is a type PQ bus, we do know its reactive injection, and so we will know the left hand side of the reactive power flow equation. This is fortunate, since we do not know $|V_3|$ (and this will always be the situation for a type PQ bus).

$$\begin{aligned}
Q_3(\mathbf{x}) &= -\left[|V_3||V_1|B_{31}\cos(\theta_3 - \theta_1) + |V_3||V_2|B_{32}\cos(\theta_3 - \theta_2) + |V_3|^2 B_{33}\right] \\
&= -\left[10|V_3|\cos\theta_3 + 10.5|V_3|\cos(\theta_3 - \theta_2) - 19.9|V_3|^2\right]
\end{aligned} \tag{T7.54c}$$

The update vector and Jacobian matrix is:

$$\Delta \underline{\mathbf{x}} = \begin{bmatrix} \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta|V_3| \end{bmatrix} \quad \underline{\mathbf{J}} = \begin{bmatrix} \frac{\partial P_2}{\partial \theta_2} & \frac{\partial P_2}{\partial \theta_3} & \frac{\partial P_2}{\partial |V_3|} \\ \frac{\partial P_3}{\partial \theta_2} & \frac{\partial P_3}{\partial \theta_3} & \frac{\partial P_3}{\partial |V_3|} \\ \frac{\partial Q_3}{\partial \theta_2} & \frac{\partial Q_3}{\partial \theta_3} & \frac{\partial Q_3}{\partial |V_3|} \end{bmatrix}$$

We obtain the various partial derivatives for the Jacobian from eqs. (T7.54a,b,c):

$$\frac{\partial P_2}{\partial \theta_2} = |V_2||V_1|B_{21}\cos(\theta_2 - \theta_1) + |V_2||V_3|B_{23}\cos(\theta_2 - \theta_3) = 10.5\cos\theta_2 + 10.5|V_3|\cos(\theta_2 - \theta_3)$$

$$\frac{\partial P_2}{\partial \theta_3} = -|V_2||V_3|B_{23}\cos(\theta_2 - \theta_3) = -10.5|V_3|\cos(\theta_2 - \theta_3)$$

$$\frac{\partial P_2}{\partial |V_3|} = |V_2|B_{23}\sin(\theta_2 - \theta_3) = 10.5\sin(\theta_2 - \theta_3)$$

$$\frac{\partial P_3}{\partial \theta_2} = -10.5|V_3|\cos(\theta_3 - \theta_2)$$

$$\frac{\partial P_3}{\partial \theta_3} = 10.0|V_3|\cos\theta_3 + 10.5|V_3|\cos(\theta_3 - \theta_2)$$

$$\frac{\partial P_3}{\partial |V_3|} = 10\sin\theta_3 + 10.5\sin(\theta_3 - \theta_2)$$

$$\frac{\partial Q_3}{\partial \theta_2} = -10|V_3||V_2|\sin(\theta_3 - \theta_2) = -10.5|V_3|\sin(\theta_3 - \theta_2)$$

$$\frac{\partial Q_3}{\partial \theta_3} = 10|V_3|\sin\theta_3 + 10|V_3||V_2|\sin(\theta_3 - \theta_2) = 10|V_3|\sin\theta_3 + 10.5|V_3|\sin(\theta_3 - \theta_2)$$

$$\frac{\partial Q_3}{\partial |V_3|} = -[|V_1|B_{31}\cos(\theta_{31}) + |V_2|B_{32}\cos(\theta_{32}) + |V_3|B_{33}] - |V_3|B_{33}$$

$$= -[10\cos\theta_3 + 10.5\cos(\theta_3 - \theta_2) - 39.96|V_3|]$$

Injections come from one-line diagram.

We are ready to start iterating using (T7.40). We note that the injections, to be used on the left hand side of eqs. (T7.54a,b,c) are $P_2 = P_{G2} = 0.6661$, $P_3 = -P_{D3} = -2.8653$, and $Q_3 = -Q_{D3} = -1.2244$; these quantities remain constant through the entire iterative process. We use a flat start; therefore our initial guess is $\theta_2 = \theta_3 = 0^\circ$ and $|V_3| = 1.0$. Using eqs. (T7.53) and (T7.54a,b,c) we get:

$$\underline{f}(\underline{x}^{(0)}) = \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix}^{(0)} = \begin{bmatrix} P_2(\underline{x}^{(0)}) \\ P_3(\underline{x}^{(0)}) \\ Q_3(\underline{x}^{(0)}) \end{bmatrix} - \begin{bmatrix} P_2 \\ P_3 \\ Q_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -0.52 \end{bmatrix} - \begin{bmatrix} 0.6661 \\ -2.8653 \\ -1.2244 \end{bmatrix} = \begin{bmatrix} -0.6661 \\ 2.8653 \\ 0.7044 \end{bmatrix}$$

PU mismatch vector. Observe mismatch is very large.

As expected for a flat start, the mismatch is large. Next we calculate the Jacobian matrix:

\underline{J}^{PV} and \underline{J}^{Q0} are both filled with zeros, because with zero resistance, they have only sin terms.

$$\underline{J} = \begin{bmatrix} 21 & -10.5 & 0 \\ -10.5 & 20.5 & 0 \\ 0 & 0 & 19.46 \end{bmatrix} \tag{T7.55}$$

Note that the Jacobian sub-matrices \underline{J}^{PV} and \underline{J}^{Q0} are both filled with zeros. This is because when resistance is neglected, these derivatives depend on sin terms, and because this is the first iteration of a flat start, all angles are zero and therefore the sin terms are all zero.

As mentioned, commercial power flow programs normally use LU factorization to obtain the update. In this case, however, because of the low dimensionality, we may invert the Jacobian. Taking advantage of the block diagonal structure, we have:

$$(\underline{J})^{-1} = \begin{bmatrix} \begin{bmatrix} 21 & -10.5 \\ -10.5 & 20.5 \end{bmatrix}^{-1} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & [19.46]^{-1} \end{bmatrix} = \begin{bmatrix} 0.0640 & 0.0328 & 0 \\ 0.0328 & 0.0656 & 0 \\ 0 & 0 & 0.0514 \end{bmatrix}$$

Now we compute:

$$\Delta \underline{x}^{(j)} = \begin{bmatrix} \Delta \theta_2 \\ \Delta \theta_3 \\ \Delta |V_3| \end{bmatrix} = -[\underline{J}]^{-1} \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = - \begin{bmatrix} 0.0640 & 0.0328 & 0 \\ 0.0328 & 0.0656 & 0 \\ 0 & 0 & 0.0514 \end{bmatrix} \begin{bmatrix} -0.6661 \\ 2.8653 \\ 0.7044 \end{bmatrix} = \begin{bmatrix} -0.0513 \\ -0.1660 \\ -0.0362 \end{bmatrix}$$

Here we inverted \underline{J} but for larger systems, you should always use factorization.

Angular measures are in radians.

The elements of the update vector corresponding to angles are in radians. We can easily convert them to degrees:

$$\begin{bmatrix} \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta|V_3| \end{bmatrix} = \begin{bmatrix} -0.0513 \text{ rad} \\ -0.1660 \text{ rad} \\ -0.0362 \end{bmatrix} = \begin{bmatrix} -2.9396^\circ \\ -9.5139^\circ \\ -0.0362 \end{bmatrix}$$

We now find $\underline{x}^{(1)}$ as follows:

$$\underline{x}^{(1)} = \underline{x}^{(0)} + \Delta\underline{x}^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -2.9396^\circ \\ -9.5139^\circ \\ -0.0362 \end{bmatrix} = \begin{bmatrix} -2.9396^\circ \\ -9.5139^\circ \\ 0.9638 \end{bmatrix}$$

We note that the exact solution is $\begin{bmatrix} -3.00^\circ \\ -10.01^\circ \\ 0.9499 \end{bmatrix}$, so this is pretty good progress for one iteration!

We proceed to the next iteration using the new values $\theta_2^{(1)} = -2.9396^\circ$, $\theta_3^{(1)} = -9.5139^\circ$, and $|V_3|^{(1)} = 0.9638$. Substituting in eq. (T7.54a), we get $P_2(\underline{x}^{(1)}) = 0.6202$, and thus $\Delta P_2^{(1)} = 0.6202 - 0.6661 = -0.0459$. Similarly, using eqs. (T7.54b) and (T7.54c), we get the updated mismatch vector:

$$\begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} -0.0463 \\ 0.1145 \\ 0.2251 \end{bmatrix}$$

Note that, in just one iteration, the mismatch vector has been reduced by a factor of about 10. Calculating \underline{J} using the updated values of the variables, we find that

$$\underline{J} = \begin{bmatrix} 20.5396 & -10.0534 & 1.2017 \\ -10.0534 & 19.5589 & -2.8541 \\ 1.1582 & -2.7508 & 18.2199 \end{bmatrix}$$

Now the \underline{J}^{PV} and \underline{J}^{Q0} terms are non-zero since angles are no longer all zero. But it is interesting that this Jacobian differs very little from initial one, below.

$$\underline{J} = \begin{bmatrix} 21 & -10.5 & 0 \\ -10.5 & 20.5 & 0 \\ 0 & 0 & 19.46 \end{bmatrix}$$

The matrix should be compared with \underline{J} from the previous iteration. It has not changed much. The elements in the off-diagonal matrices \underline{J}^{PV} and \underline{J}^{Q0} are no longer zero, but their elements are small compared to the elements in the diagonal matrices \underline{J}^{P0} and \underline{J}^{QV} . The diagonal matrices themselves have not changed much. It is also important to note that the upper left-hand Jacobian submatrix (\underline{J}^{P0}) is symmetric. This fact allows for a significant savings in storage when dealing with large systems.

The updated inverse is

$$(\underline{J})^{-1} = \begin{bmatrix} 0.0651 & 0.0336 & 0.0010 \\ 0.0336 & 0.0696 & 0.0087 \\ 0.0009 & 0.0084 & 0.0561 \end{bmatrix}$$

Comparing this inverted Jacobian with that of the last iteration, we do not see much change. Using the same procedure as before to calculate the update vector, we obtain

$$\underline{x}^{(2)} = \begin{bmatrix} \theta_2 \\ \theta_3 \\ |V_3| \end{bmatrix} = \begin{bmatrix} -3.0023^\circ \\ -9.9924^\circ \\ 0.9502 \end{bmatrix}$$

This is very close to the correct answer. The largest error is only about 0.08%. Of course in the usual problem we do not know the answer and we would continue into the next iteration. We would stop the iterations when the mismatch vector satisfies the required tolerance. We would find:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix}^2 = \begin{bmatrix} -0.0019 \\ 0.0023 \\ -0.0031 \end{bmatrix}$$

The mismatch has been reduced from that of the last iteration by a factor of 100 and is small enough. On that basis we can stop here. So we stop with the values $\theta_2 = -3.0023^\circ$, $\theta_3 = -9.9924^\circ$, and $|V_3|=0.9502$. It remains to calculate the real and reactive power generation at the swing bus (bus 1) and the reactive power generation at the PV bus (bus 2) using the calculated values of θ_2 , θ_3 , and $|V_3|$.

$$\begin{aligned} P_{G1} = P_1 &= |V_1||V_2|B_{12}\sin(\theta_1 - \theta_2) + |V_1||V_3|B_{13}\sin(\theta_1 - \theta_3) \\ &= 10.5\sin 3.0023^\circ + 9.502\sin 9.9924^\circ = 2.1987 \end{aligned}$$

P_{G1} and Q_{G1} are swing bus generation, a function of the solution.

$$\begin{aligned} Q_{G1} = Q_1 &= -\left[|V_1||V_2|B_{12}\cos(\theta_1 - \theta_2) + |V_1||V_3|B_{13}\cos(\theta_1 - \theta_3) + |V_1|^2 B_{11}\right] \\ &= -\left[10.5\cos 3.0023^\circ + 9.502\cos 9.9924^\circ - 19.98\right] \\ &= 0.1365 \end{aligned}$$

Q_{G2} is reactive generation at PV bus #2, also a function of the solution.

$$\begin{aligned} Q_{G2} = Q_2 &= -\left[|V_2||V_1|B_{21}\cos(\theta_2 - \theta_1) + |V_2||V_3|B_{23}\cos(\theta_2 - \theta_3) + |V_2|^2 B_{22}\right] \\ &= -\left[10.5\cos(-3.0023^\circ) + 9.977\cos(6.9901^\circ) - 22.028\right] \\ &= 1.6395 \end{aligned}$$

This completes the example.

T7.8 Advanced issues associated with power flow

Some more advanced issues in relation to the power flow problem are listed below.

- Jacobian elements as sensitivities and what they tell you about relations between real or reactive power injection and voltage magnitude or angle.
- Sparsity:
 - Sparse characteristic of Jacobian
 - Storage implications
 - Optimal ordering
- Different types of power flow formulations/algorithms:
 - Divide voltage magnitude part of update vector to reduce the Jacobian storage requirements
 - Fast decoupled power flow
 - Governor power flow
 - DC power flow

- Gauss-Seidel
- Some advanced modeling issues:
 - Transformers: regulating transformers
 - Area interchange
 - Switched shunt capacitors

T7.9 Input/Output and Commercial Programs

There are a number of very high quality commercial power flow programs on the market today, some of which include those developed by the Electric Power Research Institute (EPRI), Power Technologies Incorporated (PTI), Operation Technology, Inc., and EDSA. Most of today's commercial software packages are menu-driven from a Windows environment. A predecessor of one of these was interactive from the command line, and we use it here because its scroll-down recording serves well to illustrate some basic features of most commercial programs. The following lists of just a few of these features. The item number in this list is denoted in the illustration that follows, using large 20 font numbers, to help identify the corresponding parts in the session.

1. Solving the power flow
2. Limit checks – overloads
3. Limit checks – undervoltages
4. Reporting the solution – flows (denoted below as “bus flows”)
5. Reporting the solution – generator operation information (denoted below as plant data)

A one-line diagram for the system on which this session was run is shown in Figure (T7.9). This system is a test system developed by a subcommittee of the IEEE Power Engineering Society (PES) [12]. It is referred to as the IEEE RTS'96, and it consists of 24 buses, 11 of which are voltage controlled buses; thus, 13 are type PQ at the beginning of a solution procedure.

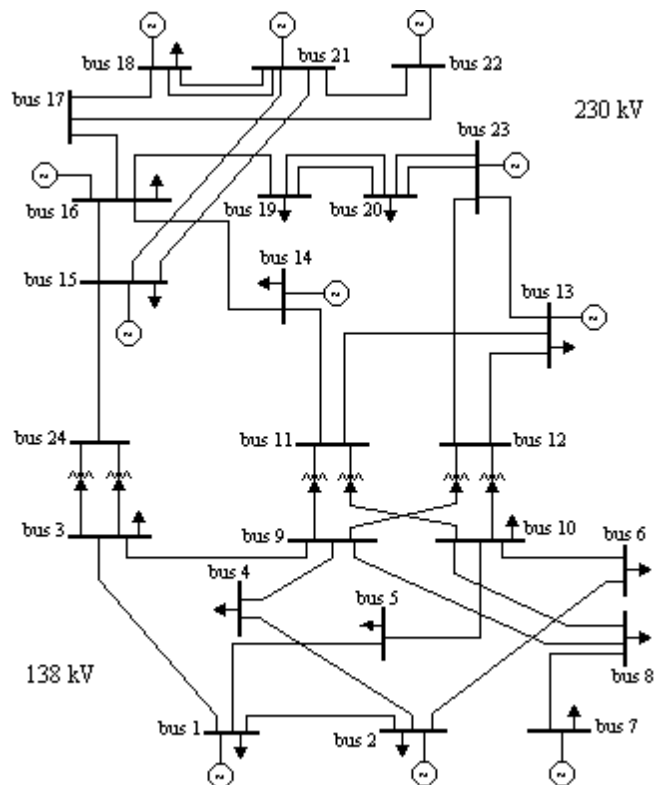


Figure T7.9 – The IEEE Reliability Test System [12]

1.

```

MAIN MENU
=====
1 LOAD THE PSF                8 DATA VERIFICATIO
2 IMPORTING UTILITIES         9 LIMIT CHECKS
3 INPUT DATA PROCESSING     10 SENSITIVITY FACTORS
4 STUDY PREPARATION UTILITIES 11 EXPORTING UTILITIES
5 POWER FLOW SOLUTION        12 CLOSE THE PSF
6 NETWORK REDUCTION          13 QUIT
7 POWER FLOW SOLUTION REPORTING

ENTER MENU CHOICE: 5

*** POWER FLOW SOLUTION ***
=====
1 AUTOMATIC SOLUTION
2 FAST DECOUPLED (XB)
3 NEWTON RAPHSON
4 FAST DECOUPLED (BX)
5 LOCALIZED SOLUTION (FD XB)
6 DC POWER FLOW SOLUTION
7 POWER FLOW SOLUTION OPTIONS

ITEM #, EXIT 3
*** START OF NEWTON RAPHSON ITERATIONS ***
ITERATION : 1  NEWTON RAPHSON                UNSOLVED ABSOLUTE ERROR
----- BUS WITH LARGEST ERROR ----- BUSES  SUMMATION
P(P.U. MW)    0.000375 ( 30 BUS 3138. ) 0      0.0017
V(RADIANS)    0.000179 ( 70 BUS 7138. ) 8      0.0016
Q(P.U. MVAR)  0.001873 ( 120 BUS 12230. ) 0      0.0078
V(P.U. KV)    0.000210 ( 30 BUS 3138. ) 3      0.0009
ITERATION : 2  NEWTON RAPHSON                UNSOLVED ABSOLUTE ERROR
----- BUS WITH LARGEST ERROR ----- BUSES  SUMMATION
P(P.U. MW)    0.000006 ( 160 BUS 16230. ) 0      0.0000
Q(P.U. MVAR)  -0.000008 ( 170 BUS 17230. ) 0      0.0000
*** POWER FLOW SOLUTION IS REACHED IN 2 ITERATIONS ***
    
```

2.

```

MAIN MENU
=====
1 LOAD THE PSF                8 DATA VERIFICATION
2 IMPORTING UTILITIES         9 LIMIT CHECKS
3 INPUT DATA PROCESSING     10 SENSITIVITY FACTORS
4 STUDY PREPARATION UTILITIES 11 EXPORTING UTILITIES
5 POWER FLOW SOLUTION        12 CLOSE THE PSF
6 NETWORK REDUCTION          13 QUIT
7 POWER FLOW SOLUTION REPORTING

ENTER MENU CHOICE: 9

** LOAD FLOW SOLUTION LIMIT CHECK **
=====
1 OVERLOADED LINES OR TRANSFORMERS
2 CONTROL PARAMETERS VIOLATION
3 BUS VOLTAGE OUTSIDE SPECIFIED BAND
4 LINES ACROSS WHICH THE ANGLE EXCEEDS THE SPECIFIED VALUE
5 CHECK FOR ALL LIMITS AND VIOLATIONS
6 SORT OPTIONS

ITEM #, SPECIFY SUBSYSTEM, EXIT 1
ALL LINES TRANSFORMERS: A
*** LOAD FLOW SOLUTION LIMIT CHECK ***
1 OVERLOADED BRANCHES          RATING GROUP = 1  % LOADING = 100.0
<----- FROM BUS -----> <----- TO BUS ----->      MVA    MVA PERCENT
NUMBER    NAME      NUMBER    NAME      CKT    LOADING RATING LOADING
=====  =====
60 BUS    6138.      100 BUS  10138.    1      175.4  175.0  100.22
    
```

3.

```

** LOAD FLOW SOLUTION LIMIT CHECK **
=====
1 OVERLOADED LINES OR TRANSFORMERS
2 CONTROL PARAMETERS VIOLATION
3 BUS VOLTAGE OUTSIDE SPECIFIED BAND
4 LINES ACROSS WHICH THE ANGLE EXCEEDS THE SPECIFIED VALUE
5 CHECK FOR ALL LIMITS AND VIOLATIONS
6 SORT OPTIONS

ITEM #, SPECIFY SUBSYSTEM, EXIT 3
    
```

HI VOLTAGE LIMIT LO VOLTAGE LIMIT: L
 ENTER VALUE IN PU: .99

*** LOAD FLOW SOLUTION LIMIT CHECK ***
 4 BUSES WITH V < 0.990

```
<----- BUS ----->          <-- VOLTAGE -->
NUMBER      NAME          TYPE BASE KV   MAG    ANGLE  AREA ZONE   OWNER ID
=====
   30 BUS    3138.         1     138.00  0.9868  -26.02  10    1  BLANK
   40 BUS    4138.         1     138.00  0.9888  -20.27  10    1  BLANK
   90 BUS    9138.         1     138.00  0.9824  -15.83  10    1  BLANK
  120 BUS   12230.         1     230.00  0.9876   -4.73  10    1  BLANK
```

UP DOWN SCREEN DUMP LIST TO FILE CHANGE RATING EXIT : e

4.

MAIN MENU

=====

- 1 LOAD THE PSF
- 2 IMPORTING UTILITIES
- 3 INPUT DATA PROCESSING
- 4 STUDY PREPARATION UTILITIES
- 5 POWER FLOW SOLUTION
- 6 NETWORK REDUCTION
- 7 POWER FLOW SOLUTION REPORTING
- 8 DATA VERIFICATION
- 9 LIMIT CHECKS
- 10 SENSITIVITY FACTORS
- 11 EXPORTING UTILITIES
- 12 CLOSE THE PSF
- 13 QUIT

ENTER MENU CHOICE: 7

*** LOAD FLOW SOLUTION REPORTING ***

=====

- 1 MISMATCH SUMMARY
- 2 SUBSYSTEM SUMMARY
- 3 BUS SHUNT DATA
- 4 PLANT DATA
- 5 MACHINE DATA
- 6 AREA INTERCHANGE DATA
- 7 TIE LINE FLOWS
- 8 BUS FLOWS
- 9 DC CONVERTERS
- 10 DC FLOWS
- 11 ULTC/PS SUMMARY
- 12 SERIES COMPENSATORS
- 13 POWER FLOW SUMMARY
- 14 SECTIONALIZED BRANCHES
- 15 STATIC TAP CHANGERS / PHASE SHIFTERS
- 16 THREE WINDING TRANSFORMERS
- 17 INTERFACE FLOWS
- 18 VOLTAGE PROFILE
- 19 SORT OPTIONS

ITEM #, SPECIFY SUBSYSTEM, EXIT 8

SUBSYSTEM BUSES PICK BUSES EXIT: p

ENTER BUS LIST FILE NAME OR BUSES TO INCLUDE OR EXCLUDE

> 60

*** SOLUTION REPORTING - BUS FLOWS ***

BUS: 60 BUS 6138.

VOLTAGE : 1.0017 PU (138.2 kV) -20.67

```
BUS NUM      NAME          AREA  CKT      MW      MVAR      MVA      TAP
=====
LOAD                                136.00   28.00   138.85
SW SHUNT                                0.00   100.33   100.33
TO 100 BUS   10138.         10  1      -128.63 -107.64   167.73
TO 20 BUS    2138.         10  1       -7.37  -20.69    21.96
```

UP DOWN NEXT BUS PREVIOUS BUS SCREEN DUMP LIST TO FILE EXIT : e

SUBSYSTEM BUSES PICK BUSES EXIT: p

ENTER BUS LIST FILE NAME OR BUSES TO INCLUDE OR EXCLUDE

> 100

*** SOLUTION REPORTING - BUS FLOWS ***

BUS: 100 BUS 10138.

VOLTAGE : 1.0132 PU (139.8 kV) -16.15

```
BUS NUM      NAME          AREA  CKT      MW      MVAR      MVA      TAP
=====
LOAD                                195.00   40.00   199.06
TO 120 BUS   12230.         10  1     -234.48  60.92   242.27  1.0000LK
TO 110 BUS   11230.         10  1     -235.99  52.21   241.70  1.0000LK
TO 80 BUS    8138.         10  1       69.77   -3.74    69.87
TO 60 BUS    6138.         10  1      130.95 -131.80   185.80
TO 50 BUS    5138.         10  1       74.76  -17.59    76.80
```

UP DOWN NEXT BUS PREVIOUS BUS SCREEN DUMP LIST TO FILE EXIT : e

5.

*** LOAD FLOW SOLUTION REPORTING ***

=====

- 1 MISMATCH SUMMARY
- 2 SUBSYSTEM SUMMARY
- 3 BUS SHUNT DATA
- 11 ULTC/PS SUMMARY
- 12 SERIES COMPENSATORS
- 13 POWER FLOW SUMMARY

```

4 PLANT DATA
5 MACHINE DATA
6 AREA INTERCHANGE DATA
7 TIE LINE FLOWS
8 BUS FLOWS
9 DC CONVERTERS
10 DC FLOWS
14 SECTIONALIZED BRANCHES
15 STATIC TAP CHANGERS / PHASE SHIFTERS
16 THREE WINDING TRANSFORMERS
17 INTERFACE FLOWS
18 VOLTAGE PROFILE
19 SORT OPTIONS

```

ITEM #, SPECIFY SUBSYSTEM, EXIT 4

*** SOLUTION REPORTING - PLANT DATA ***

```

=====
1 ALL PLANTS
2 ON LINE PLANTS
3 PLANTS AT VAR LIMIT WITH UNEQUAL VAR LIMITS
4 PLANTS WITH UNSCHEDULED REACTIVE POWER
5 PLANTS CONTROLLING A REMOTE BUS
6 PLANTS WITH RESERVE REACTIVE POWER

```

ITEM #, EXIT 1

*** SOLUTION REPORTING ***

10 PLANTS

| PLANT NUM | BUS NAME | MACHINES | | | MW | MVAR | QMAX | QMIN | |
|--------------|-------------|----------|-----|-----|----|--------|--------|--------|---------|
| | | TYPE | I/S | O/S | | | | | |
| R | 10 BUS | 1138. | 2 | 4 | 0 | 134.70 | 41.53 | 280.00 | -50.00 |
| R | 20 BUS | 2138. | 2 | 4 | 0 | 187.00 | 30.50 | 140.00 | -100.00 |
| R | 70 BUS | 7138. | 2 | 3 | 0 | 165.00 | 91.63 | 280.00 | 0.00 |
| R | 130 BUS | 13230. | 3 | 3 | 0 | 337.53 | 170.64 | 540.00 | -300.00 |
| R | 150 BUS | 15230. | 2 | 6 | 0 | 185.00 | 40.36 | 310.00 | -55.00 |
| R | 160 BUS | 16230. | 2 | 1 | 0 | 155.00 | 106.41 | 280.00 | -55.00 |
| R | 180 BUS | 18230. | 2 | 1 | 0 | 400.00 | 100.73 | 200.00 | -90.00 |
| R | 210 BUS | 21230. | 2 | 1 | 0 | 400.00 | 25.25 | 200.00 | -90.00 |
| R | 220 BUS | 22230. | 2 | 6 | 0 | 300.00 | -80.55 | 696.00 | -560.00 |
| R | 230 BUS | 23230. | 2 | 3 | 0 | 660.00 | 33.34 | 310.00 | -125.00 |

UP DOWN OTHER SCREEN SCREEN DUMP LIST TO FILE EXIT : e

References

- [1] J. Vlach and K. Singhal, "Computer Methods for Circuit Analysis and Design," Kluwer Academic Publications, 1993.
- [2] L. Chua and P. Lin, "Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques," Prentice Hall, 1975.
- [3] O. Elgerd, "Electric Energy Systems Theory," McGraw-Hill, New York, 1982.
- [4] J. Grainger and W. Stevenson, "Power System Analysis," McGraw-Hill, New York, 1994.
- [5] A. Bergen and V. Vittal, "Power Systems Analysis," second edition, Prentice Hall, Upper Saddle River, New Jersey, 2000.
- [6] C. Gross, "Power System Analysis," John Wiley & Sons, New York, 1979.
- [7] B. Weedy, "Electric Power Systems," third edition, John Wiley & Sons, London, 1979.
- [8] H. Saadat, "Power System Analysis," McGraw-Hill, New York, 1999.
- [9] G. Stagg and A. El-Abiad, "Computer Methods in Power System Analysis," McGraw-Hill, New York, 1968.
- [10] G. Heydt, "Computer Analysis Methods for Power Systems," Macmillan Publishing, New York, 1986.
- [11] V. Del Toro, "Electric Power Systems," Prentice-Hall, Englewood Cliffs, New Jersey, 1992.
- [12] The reliability test system task force of the application of probability methods subcommittee, "The IEEE reliability test system - 1996", *IEEE Trans. Power Systems*, Vol. 14, No. 3, pp. 1010-1020, 1999.