6.061 / 6.690 Introduction to Electric Power Systems
Spring 2007

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.061 Introduction to Power Systems
Class Notes Chapter 5
Introduction To Load Flow [*]

J.L. Kirtley Jr.

1 Introduction

Even though electric power networks are composed of components which are (or can be approximated to be) *linear*, electric power flow, real and reactive, is a *nonlinear* quantity. The calculation of load flow in a network is the solution to a set of nonlinear equations. The purpose of this note is to describe how network load flows may be calculated.

This is only an elementary treatment of this problem: there is still quite a bit of activity in the professional literature concerning load flow algorithms. The reason for this is that electric utility networks are often quite large, having thousands of buses, so that the amount of computational effort required for a solution is substantial. A lot of effort goes into doing the calculation efficiently. This discussion, and the little computer program at the end of this note, uses the crudest possible algorithm for this purpose. However, for the relatively simple problems we will be doing, it should work just fine.

2 Power Flow

Power flow in a network is determined by the voltage at each bus of the network and the impedances of the lines between buses. Power flow into and out of each of the buses that are network terminals is the sum of power flows of all of the lines connected to that bus. The load flow problem consists of finding the set of voltages: magnitude and angle, which, together with the network impedances, produces the load flows that are known to be correct at the system terminals. To start, we view the power system as being a collection of *buses*, connected together by *lines*. At each of the buses, which we may regard as *nodes*, we may connect equipment which will supply power to or remove power from the system. (Note: in speaking of *power* here, we are really referring to *complex* power, with both real and reactive components). If we have made a connection to a given system node (say with a generator), the complex power flow *into* the network at node k is:

$$\mathbf{S}_k = P_k + jQ_k = \mathbf{V}_k \mathbf{I}_k^* \qquad (1)$$

3 Bus Admittance

Now, if the network itself is linear, interconnections between buses and between buses and ground can all be summarized in a multiport *bus impedance matrix* or its inverse, the *bus admittance matrix*. As it turns out, the admittance matrix is easy to formulate.

The network consists of a number N_b of buses and another number N_ℓ of lines. Each of the lines will have some (generally complex) impedance Z. We form the *line admittance matrix* by placing the admittance (reciprocal of impedance) of each line on the appropriate spot on the main diagonal of an $N_\ell \times N_\ell$ matrix:

$$\mathbf{Y}_\ell = \begin{bmatrix} \frac{1}{\mathbf{Z_1}} & 0 & 0 & \cdots \\ 0 & \frac{1}{\mathbf{Z_2}} & 0 & \cdots \\ 0 & 0 & \frac{1}{\mathbf{Z_3}} & \cdots \\ \vdots & \vdots & & \ddots \end{bmatrix} \tag{2}$$

Interconnections between buses is described by the *bus incidence matrix.* This matrix, which has N_ℓ columns and N_b rows, has two entries for each line, corresponding to the buses at each end. A "direction" should be established for each line, and the entry for that line, at location (n_b, n_ℓ) in the *node incidencd matrix*, is a 1 for the "sending" end and a -1 at the "receiving" end. Actually, it is not important which end is which. The bus incidence matrix for the network described by Figure 1 below is:

$$\underline{\underline{NI}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

It is not difficult to show that the *bus admittance* matrix is given by the easily computed expression:

$$\underline{\underline{\mathbf{Y}}} = \underline{\underline{\mathbf{NI}}} \quad \underline{\underline{\mathbf{Y}_\ell}} \quad \underline{\underline{\mathbf{NI}'}} \tag{3}$$

The elements of the *bus admittance matrix*, the self– and mutual– admittances, are all of the following form:

$$\mathbf{Y}_{jk} = \frac{\mathbf{I}_k}{\mathbf{V}_j} \tag{4}$$

with all other voltages equal to zero.

Thus an alternative way to estimate the bus admittance matrix is to:

- Assume that all nodes (buses) are shorted to ground,
- Assume that one node is unshorted and connected to a voltage source,
- Calculate all node currents resulting from that one source.
- Do this for each node.

We may observe:

- Reciprocity holds:
$$\mathbf{Y}_{jk} = Y_{kj} \tag{5}$$
- Driving point admittance $\mathbf{Y}_{kk}$ is just the sum of all admittances of lines connected to bus k, including any fixed impedances connected from that bus to ground.
- Mutual admittance $\mathbf{Y}_{jk}$ is *minus* the sum of the admittances of all lines connected *directly* between buses j and k. Usually there is only one such line.

Network currents are then given by:

$$\underline{\mathbf{I}} = \underline{\underline{\mathbf{Y}}} \quad \underline{\mathbf{V}} \tag{6}$$

Where $\underline{\mathbf{I}}$ is the vector of bus currents (that is, those currents *entering* the network at its buses. $\underline{\mathbf{V}}$ represents the bus voltages and $\underline{\underline{Y}}$ is the *bus admittance matrix.* We will have more to say about estimating the bus admittance matrix in another section. For the moment, note that an individual bus current is given by:

$$\mathbf{I}_k = \sum_{j=1}^{N} \mathbf{Y}_{jk}\mathbf{V}_j \tag{7}$$

where N is the number of buses in the network. Then complex power flow at a node is:

$$\mathbf{S}_k = \mathbf{V}_k \sum_{j=1}^{N} \mathbf{Y}_{jk}^{*}\mathbf{V}_j^{*} \tag{8}$$

Now, the typical load flow problem involves buses with different constraints. It is possible to specify six quantities at each bus: voltage magnitude and angle, current magnitude and angle, real and reactive power. These are, of course, inter–related so that any two of these are specified by the other four, and the network itself provides two more constraints. Thus it is necessary to, in setting up a load flow problem, specify two of these six quantities. Typical combinations are:

- **Generator Bus:** Real power and terminal voltage magnitude are specified.
- **Load Bus:** Real and reactive power are specified.
- **Fixed Impedance:** A fixed, linear impedance connected to a bus constrains the relationship between voltage and current. Because it constrains both magnitude and angle, such an impedance constitutes two constraints.
- **Infinite Bus:** This is a voltage source, of constant magnitude and phase angle.

The load flow problem consists of solving [8] as constrained by the terminal relationships.

One bus in a load flow problem is assigned to be the “slack bus” or “swing bus”. This bus, which is taken to be an “infinite bus”, since it does not have real nor reactive power constrained, accommodates real power dissipated and reactive power stored in network lines. This bus is necessary because these losses are not known *a priori.* Further, one phase angle needs to be specified, to serve as an origin for all of the others.

4 Gauss–Seidel Iterative Technique

This is one of many techniques for solving the nonlinear load flow problem. It should be pointed out that this solution technique, while straightforward to use and easy to understand, has a tendency to use a lot of computation, particularly in working large problems. It is also quite capable of converging on incorrect solutions (that is a problem with nonlinear systems). As with other iterative techniques, it is often difficult to tell when the correct solution has been reached. Despite these shortcomings, Gauss–Seidel can be used to get a good feel for load flow problems without excessive numerical analysis baggage.

Suppose we have an initial estimate (ok: guess) for network voltages. We may partition [8] as:

$$\mathbf{S}_k = \mathbf{V}_k \sum_{j \neq k} \mathbf{Y}_{jk}^* \mathbf{V}_j^* + \mathbf{V}_k \mathbf{Y}_{kk}^* \mathbf{V}_k^* \tag{9}$$

Noting that $\mathbf{S}_k = P_k + jQ_k$, we can solve for $\mathbf{V}_k^*$ and, taking the complex conjugate of that, we have an expression for $\mathbf{V}_k$ in terms of all of the voltages of the network, P_k and Q_k:

$$\mathbf{V}_k = \frac{1}{\mathbf{Y}_{kk}} \left(\frac{P_k - jQ_k}{\mathbf{V}_k^*} - \sum_{j \neq k} \mathbf{Y}_{jk} \mathbf{V}_j \right) \tag{10}$$

Expression [10] is a better estimate of $\mathbf{V}_k$ than we started with. The solution to the set of nonlinear equations consists of carrying out this expression, repeatedly, for all of the buses of the network.

An iterative procedure in which a correction for each of the voltages of the network is computed in one step, and the corrections applied all at once is called *Gaussian Iteration.* If, on the other hand, the improved variables are used immediately, the procedure is called *Gauss–Seidel Iteration.*

Note that [10] uses as its constraints P and Q for the bus in question. Thus it is useable directly for *load* type buses. For other types of bus constraints, modifications are required. We consider only two of many possible sets of constraints.

For *generator* buses, usually the *real* power and terminal voltage *magnitude* are specified. At each time step it is necessary to come out with a terminal voltage of specified magnitude: voltage phase angle and reactive power Q are the unknowns. One way of handling this situation is to:

1. Generate an *estimate* for reactive power Q, then

2. Use [10] to generate an estimate for terminal voltage, and finally,

3. Holding voltage phase angle constant, adjust magnitude to the constraint.

At any point in the iteration, reactive power is:

$$Q_k = \mathrm{Im} \left\{ \mathbf{V}_k \sum_{j=1}^{N} \mathbf{Y}_{jk}^* \mathbf{V}_j^* \right\} \tag{11}$$

It should be noted that there are other ways of doing this calculation. Generally they are more work to set up but often converge more quickly. Newton's method and variations are good examples.

For buses loaded by constant impedance, it is sufficient to lump the load impedance into the network. That is, the *load admittance* goes directly in parallel with the *driving point admittance* at the node in question.

These three bus constraint types, generator, load and constant impedance are sufficient for handling most problems of practical importance.

5 Example: Simple–Minded Program

Attached to this note is a MATLAB script which will set up carry out the Gauss–Seidel procedure for networks with the simple constraints described here. The script is self-explanatory and carries out the load flow described by the simple example below.

Note that, as with many nonlinear equation solvers, success sometimes requires having an initial guess for the solution which is reasonably close to the final solution.

6 Example

Consider the system shown in Figure 1. This simple system has five buses (numbered 1 through 5) and four lines. Two of the buses are connected to generators, two to loads and bus 5 is the "swing bus", represented as an "infinite bus", or voltage supply.

For the purpose of this excercise, assume that the line impedances are:

$$\begin{aligned} \mathbf{Z}_0 &= .05 + j.1 \\ \mathbf{Z}_1 &= .05 + j.05 \\ \mathbf{Z}_2 &= .15 + j.2 \\ \mathbf{Z}_3 &= .04 + j.12 \end{aligned} \tag{12}$$

We also specify real power and voltage magnitude for the generators and real and reactive power for the loads:

- Bus 1: Real power is 1, voltage is 1.05 per–unit
- Bus 2: Real power is 1, voltage is 1.00 per–unit
- Bus 3: Real power is -.9 per–unit, reactive power is 0.
- Bus 4: Real power is -1, reactive power is -.2 per–unit.

Note that load power is taken to be negative, for this simple–minded program assumes all power is measured *into* the network.

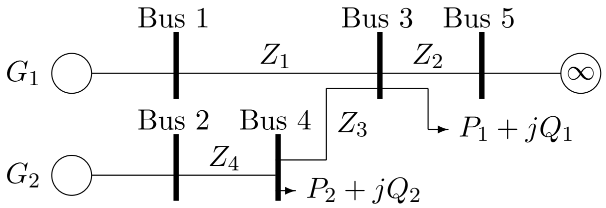

Figure 1: Sample System

```
% Simple-Minded Load Flow Example
% First, impedances
Z1=.05+j*.1;
Z2=.05+j*.05;
Z3=.15+j*.2;
Z4=.04+j*.12;
% This is the node-incidence Matrix
NI=[1 0 0 0;0 0 0 1;-1 1 1 0;0 0 -1 -1;0 -1 0 0];
% This is the vector of "known" voltage magnitudes
VNM = [1.05 1 0 0 1]’;
% And the vector of known voltage angles
VNA = [0 0 0 0 0]’;
% and this is the "key" to which are actually known
KNM = [1 1 0 0 1]’;
KNA = [0 0 0 0 1]’;
% and which are to be manipulated by the system
KUM = 1 - KNM;
KUA = 1 - KNA;
% Here are the known loads (positive is INTO network
% Use zeros for unknowns
P=[1 1 -.9 -1 0]’;
Q=[0 0 0 -.2 0]’;
% and here are the corresponding vectors to indicate
% which elements should be checked in error checking
PC = [1 1 1 1 0]’;
QC = [0 0 1 1 0]’;
Check = KNM + KNA + PC + QC;
% Unknown P and Q vectors
PU = 1 - PC;
QU = 1 - QC;
fprintf(’Here is the line admittance matrix:\n’);
Y=[1/Z1 0 0 0;0 1/Z2  0 0;0 0 1/Z3 0;0 0 0 1/Z4]
% Construct Node-Admittance Matrix
fprintf(’And here is the bus admittance matrix\n’)
YN=NI*Y*NI’
% Now: here are some starting voltage magnitudes and angles
VM = [1.05 1 .993 .949 1]’;
VA = [.0965 .146 .00713 .0261 0]’;
% Here starts a loop
Error = 1;
Tol=1e-10;
N = length(VNM);
 % Construct a candidate voltage from what we have so far
VMAG = VNM .* KNM + VM .* KUM;
VANG = VNA .* KNA + VA .* KUA;
```

```
V = VMAG .* exp(j .* VANG);
% and calculate power to start
 I = (YN*V);
 PI = real(V .* conj(I));
 QI = imag(V .* conj(I));
%pause
while(Error>Tol);
 for i=1:N,             % Run through all of the buses
                        % What we do depends on what bus!
   if (KUM(i) == 1) & (KUA(i) == 1),    % don't know voltage magnitude or angle
      pvc= (P(i)-j*Q(i))/conj(V(i));
      for n=1:N,
        if n ~=i, pvc = pvc - (YN(i,n) * V(n)); end
      end
      V(i) = pvc/YN(i,i);
   elseif (KUM(i) == 0) & (KUA(i) == 1), % know magnitude but not angle
      % first must generate an estimate for Q
      Qn = imag(V(i) * conj(YN(i,:)*V));
      pvc= (P(i)-j*Qn)/conj(V(i));
      for n=1:N,
        if n ~=i, pvc = pvc - (YN(i,n) * V(n));  end
      end
      pv=pvc/YN(i,i);
      V(i) = VM(i) * exp(j*angle(pv));
   end          % probably should have more cases
 end            % one shot through voltage list: check error
 % Now calculate currents indicated by this voltage expression
 I = (YN*V);
 % For error checking purposes, compute indicated power
 PI = real(V .* conj(I));
 QI = imag(V .* conj(I));
 % Now we find out how close we are to desired conditions
 PERR = (P-PI) .* PC;
 QERR = (Q-QI) .* QC;
Error = sum(abs(PERR) .^2 + abs(QERR) .^2);
end
fprintf('Here are the voltages\n')
V
fprintf('Real Power\n')
P
fprintf('Reactive Power\n')
Q
```