Massachusetts Institute of Technology

Department of Electrical Engineering and Computer Science

6.691 Seminar in Advanced Electric Power Systems

Problem Set 3 Solutions April 9, 2006

Load Flow I used the decoupled Newton-Raphson method for this, but anything that you could get running should do. The purpose of this problem was simply to make use of the routine you spent so much effort getting up and running. You might recall the voltages and line currents resulting from the load flow, with everything operating normally, were:

```
Bus  Voltages  are:
Bus     Per-Unit   Angle  (Radians)   (Degrees)
  1    1.0000         0.0000          0.00
  2    1.0020         0.0072          0.41
  3    0.9896        -0.0160         -0.91
  4    0.9545        -0.0918         -5.26
  5    0.9548        -0.0909         -5.21
  6    0.9574        -0.0876         -5.02
  7    0.9516        -0.0972         -5.57
  8    0.9640        -0.0732         -4.19
  9    0.9657        -0.0603         -3.46
 10    0.9073        -0.1325         -7.59
 11    0.9745        -0.0505         -2.90
 12    0.9749        -0.0463         -2.65
 13    0.8981        -0.1556         -8.91
 14    0.9951        -0.0083         -0.47
 15    0.9609        -0.0734         -4.21
 16    0.9547        -0.0859         -4.92
 17    0.9582        -0.0746         -4.28
Line  Current  Flows
Line     Amperes
  1    42.8
  2    192.1
  3    88.6
  4    370.1
  5    358.8
  6    215.6
  7    281.7
  8    367.8
  9    213.7
 10    12.9
 11    136.1
 12    46.4
```

```
13   151.4
14   349.7
15   211.4
16   335.7
17   246.3
18   85.6
19   211.5
20   362.8
21   105.5
22   90.6
```

Note that the voltage on Bus 10 is a bit low (if you want voltage to be within 5% of nominal). But the other voltages are all good. Now we try some bad things:

1. Loss of the transformer between buses 9 and 17. To do this we simply set the elements of the node incidence matrix to zero, taking that branch out of the network. the resulting voltages and currents are:

```
Bus  Voltages  are:
Bus     Per-Unit   Angle  (Radians)   (Degrees)
  1    1.0000         0.0000           0.00
  2    0.9908         0.0028           0.16
  3    0.9682        -0.0260          -1.49
  4    0.9429        -0.0969          -5.55
  5    0.9383        -0.1001          -5.73
  6    0.9394        -0.0977          -5.60
  7    0.9309        -0.1095          -6.27
  8    0.9477        -0.0819          -4.69
  9    0.9648        -0.0558          -3.20
 10    0.6952        -0.3647         -20.90
 11    0.9633        -0.0560          -3.21
 12    0.9589        -0.0542          -3.11
 13    0.7405        -0.3150         -18.05
 14    0.9890        -0.0108          -0.62
 15    0.9309        -0.0920          -5.27
 16    0.9090        -0.1234          -7.07
 17    0.6952        -0.3647         -20.90
Line  Current  Flows
Line    Amperes
  1   68.4
  2   226.4
  3   71.2
  4   372.2
  5   341.4
  6   300.0
  7   315.1
```

8	397.1
9	216.0
10	79.0
11	233.1
12	27.4
13	131.0
14	338.1
15	139.9
16	389.2
17	0.0
18	216.9
19	567.5
20	367.6
21	0.0
22	243.2

As expected, the current in lines 21 and 17 both go to zero. Line 19 is quit a bit more heavily loaded, and the voltages on buses 10 and 13 are substantially below normal. It appears this transformer is necessary for normal system operation.

2. An increase in the load on bus 10 from 15 to 35 MW,

Bus Voltages are:

Bus	Per-Unit	Angle (Radians)	(Degrees)
1	1.0000	0.0000	0.00
2	0.9992	0.0025	0.14
3	0.9851	-0.0242	-1.39
4	0.9501	-0.1003	-5.75
5	0.9506	-0.0991	-5.68
6	0.9531	-0.0959	-5.50
7	0.9470	-0.1061	-6.08
8	0.9600	-0.0807	-4.63
9	0.9613	-0.0694	-3.97
10	0.8867	-0.1907	-10.92
11	0.9716	-0.0556	-3.19
12	0.9711	-0.0532	-3.05
13	0.8821	-0.1972	-11.30
14	0.9936	-0.0109	-0.62
15	0.9556	-0.0840	-4.81
16	0.9485	-0.1006	-5.76
17	0.9522	-0.0922	-5.28

Line Current Flows

Line	Amperes
1	56.1
2	211.7
3	75.5
4	371.4

```
  5   409.9
  6   211.6
  7   301.8
  8   381.8
  9   219.7
 10    18.0
 11    153.1
 12    44.0
 13    144.5
 14    348.7
 15    196.1
 16    347.3
 17    371.1
 18    27.7
 19    272.3
 20    369.5
 21    159.0
 22    116.7
```

Nothing really bad here, except for a somewhat lower voltage on bus 10.

3. A combination of the previous two conditions. As it turns out, the system won't handle this combination. Below is what happens with the largest loading on Bus 10 I could find: 23.5 MW. For larger attempted loading the voltage on Bus 10 collapses.

```
Bus  Voltages  are:
Bus     Per-Unit   Angle  (Radians)   (Degrees)
  1    1.0000         0.0000           0.00
  2    0.9781        -0.0004          -0.02
  3    0.9450        -0.0327          -1.87
  4    0.9267        -0.1044          -5.98
  5    0.9189        -0.1087          -6.23
  6    0.9187        -0.1067          -6.11
  7    0.9078        -0.1199          -6.87
  8    0.9289        -0.0896          -5.13
  9    0.9555        -0.0595          -3.41
 10    0.5314        -0.5882         -33.70
 11    0.9504        -0.0608          -3.48
 12    0.9407        -0.0607          -3.48
 13    0.6051        -0.4521         -25.90
 14    0.9821        -0.0127          -0.73
 15    0.8998        -0.1050          -6.02
 16    0.8636        -0.1491          -8.54
 17    0.5314        -0.5882         -33.70
Line  Current  Flows
Line     Amperes
  1    97.6
```

2	264.9
3	66.5
4	376.7
5	383.7
6	341.7
7	353.5
8	426.4
9	224.2
10	125.7
11	304.9
12	21.3
13	121.7
14	336.0
15	134.6
16	429.8
17	0.0
18	401.2
19	833.4
20	378.6
21	0.0
22	357.2

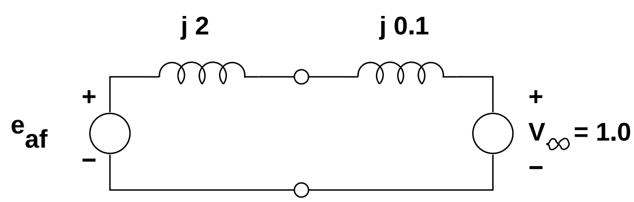

Figure 1: Per-Unit Equivalent: Generator connection to system

Voltage This one turns out to be a bit more difficult than one would think. I found it expedient to use a kind of decoupled Newton-Raphson method. Noting that:

$$p = \frac{VV_\infty}{x_\ell}\sin\theta$$
$$q = \frac{VV_\infty\cos\theta - V_\infty^2}{x_\ell}$$

we can, for some set of V and θ, compute values of p and q. The trick is to get values of V and θ that result in the right values of p and q. We can generate corrections to V and θ by noting that:

$$\frac{dp}{d\theta} = \frac{VV_\infty}{x_\ell}\cos\theta$$
$$\frac{dq}{dV} = \frac{V_\infty}{x_\ell}\cos\theta$$

Then it is straightforward to start with some guess and do the corrections until an arbitrary error is reached. If the objective for real and reactive power are p_r and q_r, at each step:

$$V_{\text{new}} = V_{\text{old}} - \frac{q - qv}{\frac{dq}{dV}}$$

$$\theta_{\text{new}} = \theta_{\text{old}} - \frac{p-pv}{\frac{dp}{d\theta}}$$

This is implemented in an appended Matlab script. Terminal voltage is shown in Figure 2.

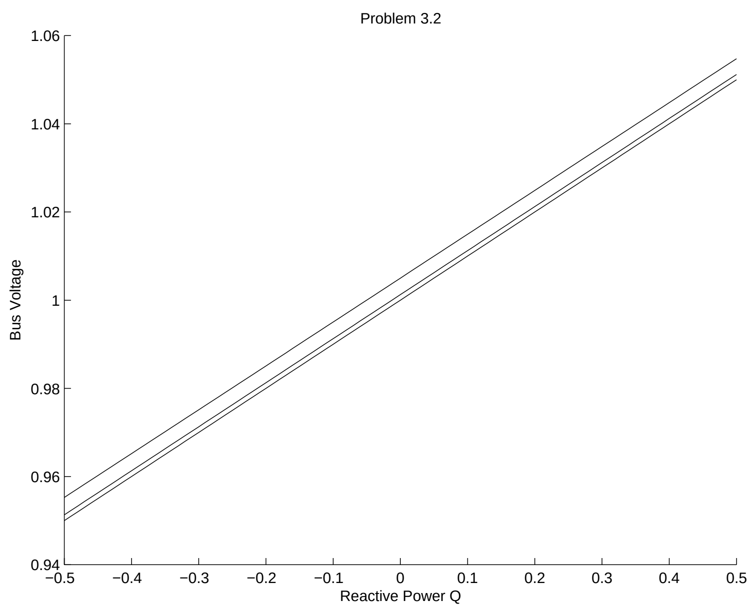

Figure 2: Terminal Voltage

To find internal voltage, note that, if we hold terminal voltage to be 'real', we find current by:

$$i = \frac{p - jq}{v}$$

Then the (complex) value of internal voltage is:

$$e_{af} = v + j * xd * i$$

This is shown in Figure 3. Note that the curve corresponding to zero real power has a kink in it, corresponding to a zero crossing. This machine would not operate stably at large negative reactive power, so not all of these points are realistic.

Equivalent Impedance This one, too, turns out to be a bit trickier than it looks at first. Note that current into the 'swing bus' (Bus 1) is:

$$i_1 = y_{1,1}v_1 + y_{2,2}v_2 + ...$$

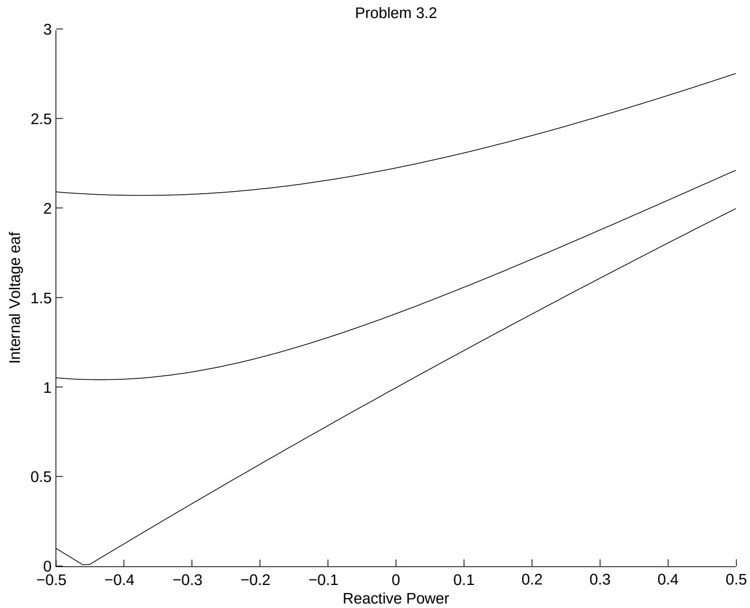

Figure 3: Terminal Voltage

The thevenin equivalent voltage is the voltage that would exist at that bus when the *current* is zero, assuming all other voltages are the same as in the actual situation. Thus, if we know all of the other voltages, the thevenin equivalent voltage is:

$$v_{\text{thev}} = -\frac{1}{y_{1,1}} \sum_{k=2}^{N} y_{1,k} v_k$$

Of course the thevenin equivalent admittance is just $y_{1,1}$. There are other ways of calculating it, including noting that the current times driving point impedance (reciprocal of admittance) is also the difference between bus voltage and thevenin equivalent voltage. The calculations are shown in an appended script and the results are:

```
Swing  Bus
Real  Power  =  1.54894    Reactive  Power  =  0.583641
Current  at  Bus  1  =  1.54894  +  j  -0.583641
Driving  Point  Admittance  at  Bus  1  =  6.36863  +  j  -36.0464
Thevenin  Equivalent  Voltage  =  0.976937  +  j  -0.038896
Check:  Real  Power  =  1.54894   Reactive  Power  =  0.583641
```

```
% 6.691 Problem Set 3, Problem 2

V_infty = 1.0;          % basic data
xl = .1;
xd = 2;
pv = [0 .5 1.0];         % stated range
qv = -.5:.01:.5;
V = zeros(length(pv), length(qv));
D = zeros(length(pv), length(qv));

tol = 1e-8;

for i = 1:length(pv)
    pr = pv(i);
    v = 1;                  % we start here
    delt = 0;
    for k = 1:length(qv)
        qr = qv(k);
        not_done = 1;
        while not_done == 1
            dpdd = v*V_infty*cos(delt)/xl;
            dqdv = V_infty*cos(delt)/xl;
            p = v*V_infty*sin(delt)/xl;
            q = (v*V_infty*cos(delt)-V_infty^2)/xl;
            E = (p-pr)^2 + (q-qr)^2;
            if E < tol
                not_done = 0;
                break
            end
            delt = delt - (p-pr)/dpdd;
            v = v - (q-qr)/dqdv;
            %pause
        end                 % end of while loop
        V(i, k) = v;
        D(i, k) = delt;
    end                     % end of q loop
end                         % end of p loop

figure(1)
clf
hold on
for i = 1:length(pv)
    plot(qv, V(i, :))
end
hold off
```

```
title('Problem 3.2')
ylabel('Bus Voltage')
xlabel('Reactive Power Q')

% now let's just check to see if this is right
Pchk = zeros(length(pv), length(qv));
Qchk = zeros(length(pv), length(qv));
for i = 1:length(pv)
    for k = 1:length(qv)
        Pchk(i, k) = V(i, k)*V_infty*sin(D(i, k))/xl;
        Qchk(i, k) = (V(i, k)*V_infty*cos(D(i, k))-V_infty^2)/xl;
    end
end

figure(2)
clf
hold on
for i = 1:length(pv)
    plot(Qchk(i, :), Pchk(i, :))
end
hold off
title('Problem 3.2')
ylabel('Real Power')
xlabel('Reactive Power')

% finally, get eaf
eaf = zeros(length(pv), length(qv));
for i = 1:length(pv)
    for k = 1:length(qv)
        I = (pv(i) -j*qv(k))/V(i, k);
        eaf(i, k) = abs(V(i, k) + j*I*xd);
    end
end

figure(3)
clf
hold on
for i = 1:length(pv)
    plot(qv, eaf(i, :))
end
hold off
title('Problem 3.2')
ylabel('Internal Voltage eaf')
xlabel('Reactive Power')
```

```
% Solution to Problem Set 2
% Newton-Raphson Solution..Decoupled.
% Augmented to get bus 1 equivalents

tol=.000001;

% first, here are the line impedances
Z_l = [3.629+j*20.53 4.718+j*26.7 3.085+j*17.47 2.774+j*15.66 3.085+j*17.47...
       2.411+j*13.69 2.514+j*14.18 2.618+j*14.78...
       1.996+j*8.17 1.529+j*6.30 1.089+j*4.46 1.97+j*8.09 3.551+j*20.09...
       3.551+j*20.09 1.886+j*10.63 3.003+j*17.16 3.433+j*11.49 3.033+j*10.15...
       4.462+j*15.54 1.270+j*7.13];
% and here are the connections for each line
C_l = [1 14;
       1 11;
       14 2;
       11 2;
       1 9;
       9 4;
       11 5;
       2 12;
       5 8;
       5 4;
       5 7;
       5 6;
       12 3;
       6 3;
       7 15;
       3 15;
       17 10;
       10 13;
       13 16;
       8 12;
       9 17;
       15 16];

Pb = 100e6;    % we are going to use this base power
Vb1 = 161e3;   % and these base voltages
Vb2 = 69e3;

Zb1 = Vb1^2/Pb;
Zb2 = Vb2^2/Pb;
Ib1 = Pb/(sqrt(3)*Vb1);
Ib2 = Pb/(sqrt(3)*Vb2);
```

```
z_l = zeros(size(Z_l));
% unfortunately we have to cobble together the impedance vector
z_l(1:16) = Z_l(1:16) ./ Zb1;
z_l(20) = Z_l(20) / Zb1;
z_l(17:19) = Z_l(17:19) ./ Zb2;

% the last two lines are transformers
z_line = [z_l j*.08/1.5 j*.08/1.5];

fprintf('Line Impedances \n')
fprintf('Buses          Ohms                  Per-Unit\n')
for i = 1:length(Z_l)
fprintf('%3.0f %3.0f %10.3f + j %10.3f %10.4f + j %10.4f\n',...
         C_l(i, 1), C_l(i, 2), real(Z_l(i)), imag(Z_l(i)),...
         real(z_l(i)), imag(z_l(i)))
end

Nl = length(z_line);        % number of lines
Nb = 17;                    % number of buses
% Now construct the node-incidence matrix:
NI = zeros(Nb, Nl);         % to start: now we fill it in
for i = 1:Nl
    if C_l(i, 1) ~=0,
        NI(C_l(i, 1), i) = 1;
        NI(C_l(i, 2), i) = -1;
    end
end

y_line = zeros(Nl);
% now the line admittance matrix is:
for i = 1:Nl
y_line(i, i) = 1 / z_line(i);
end

% and the bus admittance matrix is:
y_bus = NI * y_line * NI';

% Here are the bus power flows:

S_bus = [2.2+j*.7;     % but note we are going to ignore this one
         2.2+j*.7;
         2.2+j*.7;
         -.6-j*.1;
         -1-j*.3;
         -.8-j*.15;
```

```
        -.9-j*.2;
        -.4-j*.05;
        -.1-j*.05;
        -.15-j*.1;
        -.75-j*.15;
        -.4-j*.15;
        -.3-j*.1;
        -.35-j*.1;
        -.1;
        0;  0];

G  =  real(y_bus);
B  =  imag(y_bus);

P_r  =  real(S_bus);
Q_r  =  imag(S_bus);

%  now  here  is  an  initial  guess  about  voltages:
V  =  ones(Nb,  1);
th  =  zeros(Nb,  1);

%  now  we  go  into  a  loop
n_iter  =  0;
not_done  =  1;
while  (not_done  ==  1)
P  =  zeros(Nb,  1);
Q  =  zeros(Nb,  1);
for  i  =  1:Nb,
    for  k  =  1:Nb,
        P(i)  =  P(i)  +  V(i)*V(k)*(G(i,  k)*cos(th(i)-th(k))+B(i,k)*sin(th(i)-th(k)));
        Q(i)  =  Q(i)  +  V(i)*V(k)*(G(i,  k)*sin(th(i)-th(k))-B(i,k)*cos(th(i)-th(k)));
    end
end

X  =  [th(2:Nb);  V(2:Nb)];                   %  to  be  found
J11  =  zeros(Nb-1);               %  components  of  the  Jacobian
J12  =  zeros(Nb-1);
J21  =  zeros(Nb-1);
J22  =  zeros(Nb-1);

for  i  =  2:Nb
 for  k  =  2:Nb
  ii=  i-1;
  kk  =  k-1;
  if  k  ~=  i
```

```
   J11(ii, kk) = V(i)*V(k)*(G(i,k)*sin(th(i)-th(k))-B(i,k)*cos(th(i)-th(k)));
   J12(ii, kk) = V(i)*(G(i,k)*cos(th(i)-th(k))+ B(i,k)*sin(th(i)-th(k)));
   J21(ii, kk) = -V(i)*V(k)*(G(i,k)*cos(th(i)-th(k))+B(i,k)*sin(th(i)-th(k)));
   J22(ii, kk) = V(i)*(G(i,k)*sin(th(i)-th(k))-B(i,k)*cos(th(i)-th(k)));
  else
   J11(ii, ii) = -V(i)^2*B(i,i)-Q(i);
   J12(ii, ii) = P(i)/V(i)+V(i)*G(i,i);
   J21(ii, ii) = P(i)-V(i)^2*G(i,i);
   J22(ii, ii) = Q(i)/V(i)-V(i)*B(i,i);
%   pause
  end  % end of if k~= i
 end   % end of index k loop
end    % end of index i loop: Jacobian is constructed

J = [J11 zeros(Nb-1);zeros(Nb-1) J22];

% now find real and reactive power
PE = P-P_r;
QE = Q-Q_r;
E = [PE(2:Nb);QE(2:Nb)];
SE = sum(E .^2);         % this is absolute, per-unit error
if SE < tol
  not_done = 0;
end

X = X - inv(J)*E;
th =[0;X(1:Nb-1)];
V = [1; X(Nb:2*Nb-2)];

n_iter = n_iter + 1;

fprintf('Error = %g\n', SE);
%pause
end

fprintf('That took %10.0f iterations\n', n_iter)

% now generate per-unit line flows

v_bus = V .* exp(j .* th);
v_line = NI' * v_bus;
i_line = y_line * v_line;

I_line = zeros(size(i_line));
I_line(1:16) = Ib1 .* i_line(1:16);
```

```
I_line(17:19) = Ib2 .* i_line(17:19);
I_line(20:22) = Ib1 .* i_line(20:22);

% now generate a report:

fprintf('Bus Voltages are:\n')
fprintf('Bus    Per-Unit   Angle  (Radians)    (Degrees)\n');
for i = 1:Nb
  fprintf('%3.0f    %3.4f     %8.4f         %6.2f\n',i, V(i), th(i), (180/pi)*th(i));
end

fprintf('Line Current Flows\n')
fprintf('Line    Amperes\n')
for i = 1:22
  fprintf('%3.0f    %3.1f\n',i, abs(I_line(i)));
end

% now we are working on Problem Set 3: We want to get the thevenin
% equivalent impedance seen from unit 1

i_sb = i_line(1) + i_line(2) + i_line(5);  % this is current into bus 1
p_sb = real(i_sb);
q_sb = - imag(i_sb);
y_thev = y_bus(1, 1)
v_thev = 1 - i_sb/y_thev;
va = conj((1-v_thev)*y_thev);

fprintf('\nSwing Bus\n')
fprintf('Real Power = %g   Reactive Power = %g\n', p_sb, q_sb)
fprintf('Current at Bus 1 = %g + j %g\n', real(i_sb), imag(i_sb))
fprintf('Driving Point Admittance at Bus 1 = %g + j %g\n', real(y_thev), imag(y_thev))
fprintf('Thevenin Equivalent Voltage = %g + j %g\n', real(v_thev), imag(v_thev));
fprintf('Check: Real Power = %g  Reactive Power = %g\n', real(va), imag(va));

% here is yet another way of doing the calculation
V_e = v_bus(2:length(v_bus));
Y = y_bus(1, 2:length(y_bus(1,:)));
V_thev = -sum(V_e .* Y.') / y_thev

% now check

VA = conj((1 - V_thev) * y_thev)
```