Power Systems Laboratory (EE3P006)

EXPERIMENT-4

Load Flow Studies using Gauss Seidel Method

Shorya Sharma
19EE01017

Aim of the Experiment:

Steady State analysis of given Power System using Gauss-Seidel method of LoadFlow study.

CODE

```
clc
clear
close all

%           from_bus to_bus     line_admittance   shunt_admittance
linedata = [ 1          2        0.0000+0.1000j   0.000j;
             6          2        0.0175+0.0628j   0.030j;
             6          5        0.0777+0.2013j   0.025j;
             2          5        0.0573+0.1580j   0.020j;
             2          4        0.0607+0.1710j   0.020j;
             2          3        0.0431+0.1400j   0.015j;
             5          4        0.0110+0.0280j   0.010j;
             4          3        0.0810+0.2065j   0.025j;
             6          7        0.0000+0.1000j   0.010j];


frombus     = linedata(:,1);
tobus       = linedata(:,2);
y           = 1./linedata(:,3);  % Line admittance.
b           = linedata(:,4);     % Shunt admittance.
numbuses    = max([max(frombus) max(tobus)]); % Number of buses.
numbranches = length(frombus);   % Nnumber of branches.

%creating Ybus matrix
Ybus = zeros(numbuses,numbuses);

% Off Diagonal Elements.
for i=1:numbranches
    Ybus(frombus(i),tobus(i)) = -1*y(i);
    Ybus(tobus(i),frombus(i)) = -1*y(i);

end

% Diagonal Element.
for i=1:numbuses
    for j=1:numbranches
        if ((frombus(j) == i) || (tobus(j) == i))
            Ybus(i,i) = Ybus(i,i) + y(j) + b(j)./2;
        end
    end
end

Ybus;


% Entering the Bus Data.
Base_MVA = 100;
% Bus Type, 1=Slack, 2=PQ, 3=PV.
          %Bus no, bus type, voltage, angle, Pi,Qi, qmax, qmin
busdata = [1  1  1.06  0   0      0      1.05  -10;
            2  2  1.00  0   0      0      0       0;
            3  2  1.00  0  -0.2   -0.1    0       0;
            4  2  1.00  0  -0.4   -0.05   0       0;
            5  2  1.00  0  -0.45  -0.1    0       0;
            6  2  1.00  0  -0.6   -0.1    0       0;
            7  3  1.00  0   0.4    0     10.0   -10];
%disp("Initial");
busdata;
Nbus = max(busdata(:,1));
V       = (busdata(:,3).*exp(1i*busdata(:,4)))';
V1      = [1.06 1.00 1.00 1.00 1.00 1.00 1.00];
err     = 1;
k = 0;
while err > 0.000001
for i=1:Nbus
    if busdata(i,2)==2
        V(i)=((busdata(i,5)-1i*busdata(i,6))/conj(V(i)) - (sum(Ybus(i,:).*V)) + Ybus(i,i)*V(i))/Ybus(i,i);
        busdata(i,3)=abs(V(i));
        busdata(i,4)=angle(V(i));

    elseif busdata(i,2)==3
        busdata(i,6) = -imag( conj(V(i)) * (sum(Ybus(i,:).*V)) );
        if busdata(i,6) > busdata(i,7)
```

```
            busdata(i,6) = busdata(i,7);
            busdata(i,2) = 2;
            V(i) = ( (busdata(i,5)-1i*busdata(i,6))/conj(V(i)) - (sum(ybus(i,:).*V)) + Ybus(i,i)*V(i) )/Ybus(i,i);
            busdata(i,3) = abs(V(i));
            busdata(i,4) = angle(V(i));
        elseif busdata(i,6) < busdata(i,8)
            busdata(i,6) = busdata(i,8);
            busdata(i,2) = 2;
            V(i) = ( (busdata(i,5)-1i*busdata(i,6))/conj(V(i)) - (sum(Ybus(i,:).*V)) + Ybus(i,i)*V(i) )/Ybus(i,i);
            busdata(i,3) = abs(V(i));
            busdata(i,4) = angle(V(i));
        else
            busdata(i,4) = angle(( (busdata(i,5)-1i*busdata(i,6))/conj(V(i)) - (sum(Ybus(i,:).*V)) + Ybus(i,i)*V(i)
)/Ybus(i,i));
            V(i) = (busdata(i,3).*exp(1i*busdata(i,4)))';
        end
    end
end

err = max(abs(V-V1));
V1 = V;
k = k+1;
end
disp("Final");
disp(busdata);
disp("Error : "+err);
disp("No of Iterations : "+k);

V_mag = abs(V);
Y_mag = abs(Ybus);
delta = angle(V);
theta = angle(Ybus);

% Computation of Slack Bus Power:

P_G =zeros(size(Ybus));
Q_G =zeros(size(Ybus));

for i=1:Nbus
    for k=1:Nbus
        P_G(i,k) =  (Y_mag(i,k)*V_mag(i)*V_mag(k)*cos(theta(i,k)-delta(i)+delta(k)));
        Q_G(i,k) = - (Y_mag(i,k)*V_mag(i)*V_mag(k)*sin(theta(i,k)-delta(i)+delta(k)));

    end
end
P_G;
Q_G;

P_G=sum(P_G,2);
Q_G=sum(Q_G,2);

P_G=P_G*Base_MVA;
disp(P_G);
Q_G=Q_G*Base_MVA;
disp(Q_G);

% Computation of Line Flows:
%when i!=k
P = zeros(Nbus,Nbus);
Q = zeros(Nbus,Nbus);
for i=1:Nbus
    for k=1:Nbus
         if i~=k
            P(i,k)= -(V_mag(i)^2)*Y_mag(i,k)*cos(theta(i,k))+ (Y_mag(i,k)*V_mag(i)*V_mag(k)*cos(theta(i,k)-
delta(i)+delta(k)));
            Q(i,k)=(V_mag(i)^2)*Y_mag(i,k)*sin(theta(i,k))- (Y_mag(i,k)*V_mag(i)*V_mag(k)*sin(theta(i,k)-
delta(i)+delta(k)));

         end
    end
end

% Computation of Losses
P_Loss = zeros(Nbus,Nbus);
Q_Loss = zeros(Nbus,Nbus);
for i=1:Nbus
    for k=i:Nbus
        P_Loss(i,k)= P(i,k)+ P(k,i);
        Q_Loss(i,k)= Q(i,k)+ Q(k,i);
    end
end
%converting pu to value
P_Loss=P_Loss*Base_MVA;
Q_Loss=Q_Loss*Base_MVA;
disp(P_Loss);
disp(Q_Loss);
```

```
%calculating total losses by adding losses of all the buses
Total_P_Loss= sum(P_Loss(:));
Total_Q_Loss= sum(Q_Loss(:));
disp(Total_P_Loss);
disp(Total_Q_Loss);
```

MATLAB OUTPUT

```
Final
    1.0000    1.0000    1.0600         0         0         0    1.0500  -10.0000
    2.0000    2.0000    1.0199   -0.0713         0         0         0         0
    3.0000    2.0000    0.9972   -0.1031   -0.2000   -0.1000         0         0
    4.0000    2.0000    0.9930   -0.1147   -0.4000   -0.0500         0         0
    5.0000    2.0000    0.9929   -0.1124   -0.4500   -0.1000         0         0
    6.0000    2.0000    1.0090   -0.0643   -0.6000   -0.1000         0         0
    7.0000    3.0000    1.0000   -0.0249    0.4000   -0.0548   10.0000  -10.0000

Error : 8.6152e-07
No of Iterations : 57
   76.9697
   -0.0017
  -20.0003
  -40.0023
  -45.0002
  -59.9998
   89.8589

   45.2224
   -0.0012
  -10.0002
   -5.0013
  -10.0001
  -10.0000
   -5.4773
```

```
         0         0         0         0         0         0         0
         0         0    0.3101    0.4862    0.4959    0.0700         0
         0         0         0    0.0250         0         0         0
         0         0         0         0    0.0062         0         0
         0         0         0         0         0    0.4307         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0

         0    7.0927         0         0         0         0         0
         0         0    1.0073    1.3698    1.3675    0.2513         0
         0         0         0    0.0638         0         0         0
         0         0         0         0    0.0158         0         0
         0         0         0         0         0    1.1159         0
         0         0         0         0         0         0    8.0994
         0         0         0         0         0         0         0

    1.8243

   20.3835
```

DISCUSSION

1. We first input the data i.e., read the voltage at the slack bus, Real and reactivepower at PQ buses, real power and voltage at PV bus along with the desired limits of reactive power (Q_{max} and Q_{min}).
2. From the given values of line parameters, we develop the bus admittance matrix.
3. We initialize the voltage, set convergence criteria and set iteration count to zero.
4. For PV type bus, we calculate Reactive power for the next iteration for say bus i i.e., Q_i^k . If this value satisfies the specified limits criterion i.e.

   $$Q_i^k > Q_{min}$$
   $$Q_i^k < Q_{max}$$

   , then we move to the next iteration otherwise we make $Q_i = Q_{min}$ or Q_{max} depending on which condition is not satisfied at treating it as a PQ bus. (Calculate δ and proceed for calculation as a PQ bus)
5. For the PQ bus, we calculate voltage magnitude and phase using static load flow equations. If the calculated values are within acceptable error limits, we proceed to check the convergence criterion. If not met we perform one more iteration otherwise we evaluate line flow, line loss and power flow.
6. When all the buses are evaluated, we stop the process.

CONCLUSION

Gauss-Seidel Method which is based on modified gauss-iteration method, is used to solve the linear system equations. In this method we first assume an initial approximation then keep on improving the approximation until a reasonable solution is obtained. Although gauss method normally converges to a solution but convergence on an erroneous solution might happen if the original voltages are taken widely different from correct values.

Any unwanted solutions can be easily detected by inspection of results since the voltages of the system generally have a range phase wider than 45^0 and the difference between nearby busses is usually very small.

We assume the voltage for all the buses except the slack bus where the voltage magnitude and phase angle are specified and remain fixed. Normally, we set i.e., assume the voltage magnitude and phase angle of these buses equal to that of the slack bus and work in a per unit system. The process is continued till:

$$V_i^{k+1} - V_i^k < \varepsilon \quad \text{for all buses}$$

The number of iterations is reduced considerably if the correction in voltage at each bus is multiplied by an acceleration factor which helps the value to converge faster. Generally, an acceleration factor value of 1.6 is suitable for convergence in lessiterations.

Thus, the process is continued till the mod of the bus voltage obtained at the current iteration less the value of bus voltage at the previous iteration is smaller than a chosen very small number and in this way we obtain the solution, in magnitude and phase angle of voltage.