NAME:- AYUSH JAIN

ROLL NO:- 121601005

SUBJECT:- POWER SYSTEMS PROJECT

Load Flow Analysis by Gauss Seidel Method

Aim:- To perform load flow analysis by Gauss Seidel method and to determine the unknown bus voltages.

Theory:- Load flow analysis is the most important and essential approach to investigating problems in power system operating and planning. Based on a specified generating state and transmission network structure, load flow analysis solves the steady operation state with node voltages and branch power flow in the power system.

Load flow studies determine if system voltages remain within specified limits under normal or emergency operating conditions, and whether equipment such as transformers and conductors are overloaded. Load flow studies are commonly used to: Optimize component or circuit loading. Develop practical bus voltage profiles.

MATLAB CODE:-

```
clear all;
%% Base Values
V_base = 230e3;
MVA_base = 100;
alpha = 1.6;        % Accelarating Factor

%% Take line data as the input from the user.

%           |From|To |    R     |    X     |    G     |    B     |  Total   |  Y/2   |
%           |Bus |Bus|    pu    |    pu    |    pu    |    pu    | charging|  pu    |
linedata = [ 1    2   0.01008  0.05040  3.81563 -19.0781   10.25   0.05125;
             1    3   0.00744  0.03720  5.16956 -25.8478    7.75   0.03875;
             2    4   0.00744  0.03720  5.16956 -25.8478    7.75   0.03875;
             3    4   0.01272  0.06360  3.02371 -15.1185   12.75   0.06375;];
```

```
%% From the line data given, generate the Ybus matrix.
init_bus = linedata(:,1);       % From bus number
final_bus = linedata(:,2);      % To bus number
r = linedata(:,3);              % Resistance, R
x = linedata(:,4);              % Reactance, X
g = linedata(:,5);              % Conductance, G
b = linedata(:,6);              % Susceptance, B
s = linedata(:,8);              % Shunt or Ground Admittance
z = r + 1i*x;                   % Impedance
y = g + 1i*b;                   % Admittance
s = 1i*s;                       % Shunt from bus to the ground

tot_buses = max(max(init_bus),max(final_bus));    % total no. of buses
tot_branches = length(init_bus);                  % no. of branches
Ybus = zeros(tot_buses,tot_branches);             % Initialising YBus
sbus = zeros(tot_buses,tot_branches);

for a = 1:tot_buses
    Ybus(init_bus(a),final_bus(a)) = -y(a);
    Ybus(final_bus(a),init_bus(a)) = -y(a);
end

for b = 1:tot_buses
    sbus(init_bus(b),final_bus(b)) = s(b);
    sbus(final_bus(b),init_bus(b)) = s(b);
end

for b = 1:tot_buses
    for c = 1:tot_buses
        if c ~= b
            Ybus(b,b) = Ybus(b,b) - Ybus(b,c) + sbus(b,c);
        end
    end
end

%% Take bus data as the input from the user.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note:-    Bus Types       -      Symbol
%           Slack Bus       -        1
%         Generator Bus     -        2
%           Load Bus        -        3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%                    |  Generation |      Load      |
%          |Bus|Type|P(MW)|Q(MVAR)|P(MW)| Q(MVAR) | V,pu |delta|Qmin|Qmax|
busdata = [  1   1     0      0      50    30.99    1.00   0.0    0   0.0 ;
             2   3     0      0     170   105.35    1.00   0.0    0   0.0 ;
             3   3     0      0     200   123.94    1.00   0.0    0   0.0 ;
             4   2    318     0      80    49.58    1.02   0.0   0.1  0.2;];
```

```
%% Compute reactive power 'Q' for the voltage controlled buses.
bus_no = busdata(:,1);      % Bus Number
bus_type = busdata(:,2);    % Bus Type
GenMW = busdata(:,3);       % Active Power Generated
GenMVAR = busdata(:,4);     % Reactive Power Generated
LoadMW = busdata(:,5);      % Active Power Demanded
LoadMVAR = busdata(:,6);    % Reactive Power Demanded
V = busdata(:,7);           % Initial Bus Voltages
del = busdata(:,8);         % Initial Bus Angles
Qmin = busdata(:,9);        % Minimum limit on the reactive power
Qmax = busdata(:,10);       % Maximum limit on the reactive power

V_orig = V.*cos(del) + 1i*V.*sin(del);
V_new = V_orig;
P = (GenMW - LoadMW)/MVA_base;        % Pi = PGi - PLi, Active Power at i'th
bus.
Q = (GenMVAR - LoadMVAR)/MVA_base;    % Qi = QGi - QLi, Reactive Power at
i'th bus.

%% Start the iterations.
% The iteration will continue until the difference between two consecutive
% voltage values becomes less than 0.00001.

iter = 1;
disp("Per unit voltage of buses after each iteration are:")
while(V_orig - V_new < 0.00001)
for idx = 1:tot_buses
    temp1 = 0;
    % Computing new voltages for all the load buses.
    if bus_type(idx) == 3
        for b = 1:tot_buses
            if b ~= idx
                temp1 = temp1 + Ybus(idx,b)*V_new(b);
            end
        end
        Vidx_new = ((P(idx)-1i*Q(idx))/V_new(idx) - temp1)/Ybus(idx,idx);
        Vidx_new_acc = (1-alpha)*V_new(idx) + alpha*Vidx_new;
        V_new(idx) = Vidx_new_acc;
    end

    % Computing Q values for all the voltage controlled buses.
    for a = 1:tot_buses
        temp = 0;
        if bus_type(a) == 2
            for b = 1:tot_buses
                temp = temp + V_new(a)*Ybus(a,b)*V_new(b);
            end
            Q(a) = -imag(temp);
            if Q(a) < Qmin(a)
```

```
                Q(a) = Qmin(a);
            end
            if Q(a) > Qmax(a)
               Q(a) = Qmax(a);
            end
        end
    end

    % Computing new voltages for the voltage controlled buses.
    if bus_type(idx) == 2
        for b = 1:tot_buses
            if b ~= idx
                temp1 = temp1 + Ybus(idx,b)*V_new(b);
            end
        end
        Vidx_new = ((P(idx)-1i*Q(idx))/V_new(idx) - temp1)/Ybus(idx,idx);
        V_temp = Vidx_new*V_new(idx)/abs(Vidx_new);
        V_new(idx) = V_temp;
    end
end

disp(['Iteration no: ',num2str(iter)])
disp(V_orig)

if max(V_orig - V_new) > 0.00001
    V_orig = V_new;
end
iter = iter + 1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Note:- This code is valid for any number of buses. For an illustration a 4-bus example is considered here.

References:- Power System Analysis by J.J.Grainger.