Washington State University
EE 521 Analysis of Power System
Fall 2021
Project 5
Optimal Power Flow

Name: Chufeng Sun
Professor: Dr. Anjan Bose
12/15/2021

Contents

Overview

Many power system applications, such as the power flow, offer only a snapshot of the system operation. Usually, the system planner or operator is interested in the effect that making adjustments to the system parameters will have on the power flow through lines or system losses. Rather than making the adjustments in a random fashion, the system planner will attempt to optimize the adjustments according to some objective costs, reservoir water levels, or system losses, among others. The optimal power flow problem is to formulate the power flow problem to find system voltages and generated powers within the framework of the objective function. In this project, OPF is applied in the IEEE 14-bus system to minimize the cost.

Economic Dispatch

For 14 buses system, the generator costs are:

$$F_1(P_1) = 0.004P_1^2 + 8P_1 \qquad (1)$$

$$F_2(P_2) = 0.0048P_2^2 + 6.4P_1 \qquad (2)$$

Object function:

$$f1 = \frac{\partial \zeta}{\partial \lambda} = g(x, u) = 0 \qquad (3)$$

Also,

$$L = f - \lambda(P1 + P2 - 259) \qquad (4)$$

For the optimal power flow

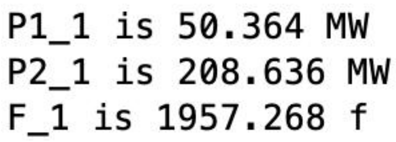
$$g = \begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} \qquad (5)$$

$$x = \begin{bmatrix} \delta_i \\ V_i \end{bmatrix} \qquad (6)$$

$$\frac{\partial \zeta}{\partial u} = \frac{\partial f}{\partial u} - [\frac{\partial g}{\partial u}]^T \lambda = \frac{\partial f}{\partial u} - [\frac{\partial g}{\partial u}]^T [\frac{\partial g}{\partial x}]^{T-1} [\frac{\partial f}{\partial x}] \qquad (7)$$

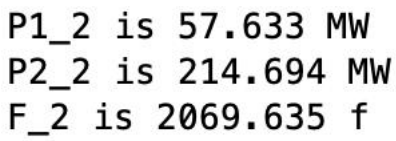
Result

Solve the economic dispatch for P(total)=259 MW



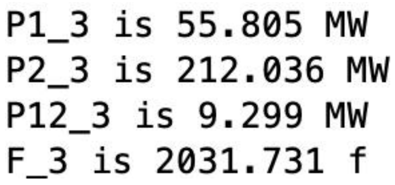
```
P1_1 is 50.364 MW
P2_1 is 208.636 MW
F_1 is 1957.268 f
```

From base PF, use P(total)=P1+P2 to solve economic dispatch



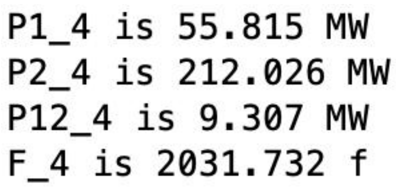
```
P1_2 is 57.633 MW
P2_2 is 214.694 MW
F_2 is 2069.635 f
```

Solve OPF with u=p2



```
P1_3 is 55.805 MW
P2_3 is 212.036 MW
P12_3 is 9.299 MW
F_3 is 2031.731 f
```

Solve OPF with line constraint P12 ≤ 5MW



```
P1_4 is 55.815 MW
P2_4 is 212.026 MW
P12_4 is 9.307 MW
F_4 is 2031.732 f
```

Code

```
clc
clear all
%   Brahcn      Bus      To bus   R              X                Line                Rtio
%                                                                 Charging B
linedata=[    1         2          0.01938    0.05917      0.0528              1
                    1         5          0.05403    0.22304      0.0492              1
                    2         3          0.04699    0.19797      0.0438              1
                    2         4          0.05811    0.17632      0.0340              1
                    2         5          0.05695    0.17388      0.0346              1
                    3         4          0.06701    0.17103      0.0128              1
                    4         5          0.01335    0.04211      0.0                 1
                    4         7          0.0        0.20912      0.0             0.978
                    4         9          0.0        0.55618      0.0             0.969
                    5         6          0.0        0.25202      0.0             0.932
                    6        11          0.09498    0.19890      0.0                 1
                    6        12          0.12291    0.25581      0.0                 1
                    6        13          0.06615    0.13027      0.0                 1
                    7         8          0.0        0.17615      0.0                 1
                    7         9          0.0        0.11001      0.0                 1
                    9        10          0.03181    0.08450      0.0                 1
                    9        14          0.12711    0.27038      0.0                 1
                   10        11          0.08205    0.19207      0.0                 1
                   12        13          0.22092    0.19988      0.0                 1
                   13        14          0.17093    0.34802      0.0                 1];

            numberline=length(linedata(:,1))
            a=linedata(:,1);                % Number of Buses
            b=linedata(:,2);                % Number of to Buses
            R=linedata(:,3);                 % Get the resistance
            X=linedata(:,4);                 % Get the Reactance
            B_Charging=i*linedata(:,5)/2;              % Get B/2
            T=linedata(:,6);                    % Get the ratio of transformer
            Ti=T*i;


      %        |   1     |   2   |   3       |     4           |       5       |       6        |     7    |        8        |      9      |
10           |       11      |       12      |       13     |       14    |      15     |
        %         |    Bi    | Type|  Fin-V     | FinAng-Deg |     PL-MW     | QL-MVAR     |     PGen  | QGen-MVAR |
BaseKV | DesiredVolts | MaxMVAR | MinMVAR | ShuntG |     ShuntB| Remote |

node     = [    1           3         1.060             0.0              0.0             0.0           232.4          -16.9            0.0
1.060       0.0       0.0      0.0     0.0        0;
            2     2     1.045      -4.98      21.7      12.7      40.0      42.4      0.0
1.045       50.0     -40.0      0.0     0.0        0;
            3     2     1.010     -12.72      94.2      19.0       0.0      23.4      0.0
1.010       40.0       0.0      0.0     0.0        0;
            4     0     1.019     -10.33      47.8      -3.9       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.0        0;
            5     0     1.020      -8.78       7.6       1.6       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.0        0;
            6     2     1.070     -14.22      11.2       7.5       0.0      12.2      0.0
1.070       24.0      -6.0      0.0     0.0        0;
            7     0     1.062     -13.37       0.0       0.0       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.0        0;
            8     2     1.090     -13.36       0.0       0.0       0.0      17.4      0.0
1.090       24.0      -6.0      0.0     0.0        0;
            9     0     1.056     -14.94      29.5      16.6       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.19       0;
           10     0     1.051     -15.10       9.0       5.8       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.0        0;
           11     0     1.057     -14.79       3.5       1.8       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.0        0;
           12     0     1.055     -15.07       6.1       1.6       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.0        0;
           13     0     1.050     -15.16      13.5       5.8       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.0        0;
           14     0     1.036     -16.04      14.9       5.0       0.0       0.0      0.0
0.0         0.0        0.0      0.0     0.0        0];

      numberbus=max(node(:,1));                % Get the number of the node
      V=node(:,10);
      V(~V)=1;                                 % The initial voltage from 1
      Vangle=zeros(numberbus,1);               % The initial angle of voltage from 0
      PG0=node(:,7)/100;
      P0=(node(:,7)-node(:,5))/100;            % Get P of each bus
      Q0=(node(:,8)-node(:,6))/100;            % Get Q of each bus
      C=node(:,13)+i*node(:,14);               % Get the Shunt

      PVbus=node(:,2)==2;                      % Define 2 is PV bus
      PQbus=node(:,2)==0;                      % Define 0 is PQ bus
      numberPVbus=sum(PVbus);                  % Define The number of PV bus
      numberPQbus=sum(PQbus);                  % Define the number of PQ bus
      PQ=find(node(:,2)==0|node(:,2)==1);      % Find the order of PQ from the form
```

```
    Z=R+i*X;              % Define the impedance

    Y=(1./Z);             % Define the resistance




    %=====================off Diagonal for Y bus==============================
    Ybus=zeros(14,14);          % Create a new empty Y matrix bus
    for k=1:numberline
      Ybus(linedata(k,1),linedata(k,2))=Ybus(linedata(k,1),linedata(k,2))-Y(k)/T(k);
      Ybus(linedata(k,2),linedata(k,1))=Ybus(linedata(k,1),linedata(k,2));
    end

    %===================Diagonal for Y bus=============================

   for m=1:numberbus
        for n=1:numberline
            if m==linedata(n,1);
                Ybus(m,m)=Ybus(m,m)+Y(n)/(T(n)^2)+B_Charging(n);
            elseif m==linedata(n,2);
                Ybus(m,m)=Ybus(m,m)+Y(n)+B_Charging(n);
            end
        end
        Ybus(m, m) = Ybus(m, m) + C(m);
   end





  Yabs=abs(Ybus);        % Get the real of Y bus
  Yangle=angle(Ybus);        % Get the angle of Y bus
  Gi=real(Ybus);                        % Get the real of Y bus
  Bi=imag(Ybus);                    % Get the imag of Y bus

%% 1st part
  PTotal=259;
  PP=[0.008,0,-1;0 0.0096 -1;1 1 0];
  L=[-8;-6.4;PTotal];
  c=inv(PP)*L;
```

```
    P1_1=c(1);
    P2_1=c(2);
    F_1 = 0.004 * P1_1^2 + 8 * P1_1 + 0.0048 * P2_1^2 + 6.4 * P2_1;




    %% 2nd part . - Solve economic dispatch

    %%%%%%%%%% Get the initial P and Q Value %%%%%%%%%%%%%%%%%%%%%%%%%%
      Pi=zeros(numberbus,1);
      Qi=zeros(numberbus,1);
      for m=1:numberbus
          for n=1:numberbus
              Pi(m)=Pi(m)+Yabs(m,n)*V(m)*V(n)*cos(Vangle(m)-Vangle(n)-Yangle(m,n));
              Qi(m)=Qi(m)+Yabs(m,n)*V(m)*V(n)*sin(Vangle(m)-Vangle(n)-Yangle(m,n));

          end

      end

      dP=P0-Pi;
      dQ=Q0-Qi;
      dm=[dP(2:end);dQ(PQ)];    % Get the first time mismatches


      Tol=1
      Iter=0;

while max(abs(Tol))>1e-2 && (Iter<10)



      %%%%%%%% Jacobian %%%%%%%%
      %%%%%%%% J11 %%%%%%%%%%
      for m=1:(numberPQbus+numberPVbus)
          for n=1:(numberPQbus+numberPVbus)
      if m==n
          J11(m,n)=-Qi(m+1)-Bi(m+1,n+1)*V(m+1)^2;
      else
          J11(m,n)=V(m+1)*V(n+1)*Yabs(m+1,n+1)*sin(Vangle(m+1)-Vangle(n+1)-Yangle(m+1,n+1));
      end
          end
      end
```

```
%%%%%%%% J12 %%%%%%%%%%
for m=2:(numberPQbus+numberPVbus+1)
    for n=2:numberPQbus+1
        k=PQ(n-1);
        if k==m
    J12(m-1,n-1)=Pi(k)/V(k)+Gi(m,k)*V(k);
else
    J12(m-1,n-1)=V(m)*Yabs(m,k)*cos(Vangle(m)-Vangle(k)-Yangle(m,k));
end
    end
end


%%%%%%%% J21 %%%%%%%%%%
for m=2:numberPQbus+1
    k=PQ(m-1)
    for n=2:(numberPQbus+numberPVbus+1)
        if n==k
    J21(m-1,n-1)=Pi(k)-Gi(k,n)*V(k)^2;
else
    J21(m-1,n-1)=-1*V(k)*V(n)*Yabs(k,n)*cos(Vangle(k)-Vangle(n)-Yangle(k,n));
end
    end
end


%%%%%%%% J22 %%%%%%%%%%
for m=2:numberPQbus+1
    k1=PQ(m-1)
    for n=2:numberPQbus+1
        k2=PQ(n-1)
        if k1==k2
    J22(m-1,n-1)=Qi(k1)/V(k1)-Bi(k1,k2)*V(k1);
else
    J22(m-1,n-1)=V(k1)*Yabs(k1,k2)*sin(Vangle(k1)-Vangle(k2)-Yangle(k1,k2));
end
    end
end

J=[J11 J12; J21 J22];
```

```
        X0=[Vangle(2:end);V(PQ)];    % Create a vector that made of Vangle and V
        X=X0+inv(J)*(dm);

        Vangle0=X(1:numberbus-1);    % Get the Vangle after the calculation
        V0=X(numberbus:end);          % Get the V after the calculation




        Vangle=[0;Vangle0];          % Get the Vagnle excepting slack bus
        V=node(:,10);                 % Get the V from the node data
        V(~V)=V0;                     % Use the V0 to instead of 0



        %%%%%%    Get a new P and Q after the calculation %%%%%%%%%%%%
         Pi=zeros(numberbus,1);
         Qi=zeros(numberbus,1);
        for m=1:numberbus
            for n=1:numberbus
                Pi(m)=Pi(m)+Yabs(m,n)*V(m)*V(n)*cos(Vangle(m)-Vangle(n)-Yangle(m,n));
                Qi(m)=Qi(m)+Yabs(m,n)*V(m)*V(n)*sin(Vangle(m)-Vangle(n)-Yangle(m,n));

            end

        end




        dP=P0-Pi;
        dQ=Q0-Qi;
        dm=[dP(2:end);dQ(PQ)];    % Mismatches

        Tol=dm;
        Iter=Iter+1;



end

      PG=Pi+node(:,5)/100;
      PTotal2=sum(PG(1:2)) * 100;
      PP2 = [0.008,0,-1;0 0.0096 -1;1 1 0];
      L2 = [-8;-6.4;PTotal2];
      P = inv(PP2) * L2;
      P1_2 = P(1);
```

```
P2_2 = P(2);
F_2 = 0.004 * (P1_2)^2 + 8 * P1_2 + 0.0048 * (P2_2)^2 + 6.4 * P2_2;


%% Solve OPF with u=P2
P1=P1_2;
P2=P2_2/100;
P12=40;

lu= 1e3;


  V=node(:,10);
  V(~V)=1;                                  % The initial voltage from 1
  Vangle=zeros(numberbus,1);                % The initial angle of voltage from 0

 while abs(lu) > 1e-4



         if P2 > 0
           PG0(2) = P2;
         end

         P0=PG0-node(:,5)/100;

         %%%%%%%%%% Get the initial P and Q Value %%%%%%%%%%%%%%%%%%%%%%%%%
  Pi=zeros(numberbus,1);
  Qi=zeros(numberbus,1);
  for m=1:numberbus
      for n=1:numberbus
          Pi(m)=Pi(m)+Yabs(m,n)*V(m)*V(n)*cos(Vangle(m)-Vangle(n)-Yangle(m,n));
          Qi(m)=Qi(m)+Yabs(m,n)*V(m)*V(n)*sin(Vangle(m)-Vangle(n)-Yangle(m,n));

      end

  end

  dP=P0-Pi;
  dQ=Q0-Qi;
  dm=[dP(2:end);dQ(PQ)];    % Get the first time mismatches
```

```
          Tol=1;
          Iter=0;
while max(abs(Tol))>1e-2 && (Iter<10)
    %%%%%%%% Jacobian %%%%%%%%
    %%%%%%%% J11 %%%%%%%%%%
    for m=1:(numberPQbus+numberPVbus)
        for n=1:(numberPQbus+numberPVbus)
    if m==n
        J11(m,n)=-Qi(m+1)-Bi(m+1,n+1)*V(m+1)^2;
    else
        J11(m,n)=V(m+1)*V(n+1)*Yabs(m+1,n+1)*sin(Vangle(m+1)-Vangle(n+1)-Yangle(m+1,n+1));
    end
        end
    end


    %%%%%%%% J12 %%%%%%%%%%
    for m=2:(numberPQbus+numberPVbus+1)
        for n=2:numberPQbus+1
            k=PQ(n-1);
            if k==m
        J12(m-1,n-1)=Pi(k)/V(k)+Gi(m,k)*V(k);
    else
        J12(m-1,n-1)=V(m)*Yabs(m,k)*cos(Vangle(m)-Vangle(k)-Yangle(m,k));
    end
        end
    end


    %%%%%%%% J21 %%%%%%%%%%
    for m=2:numberPQbus+1
        k=PQ(m-1);
        for n=2:(numberPQbus+numberPVbus+1)
            if n==k
        J21(m-1,n-1)=Pi(k)-Gi(k,n)*V(k)^2;
    else
        J21(m-1,n-1)=-1*V(k)*V(n)*Yabs(k,n)*cos(Vangle(k)-Vangle(n)-Yangle(k,n));
    end
        end
    end


    %%%%%%%% J22 %%%%%%%%%%
    for m=2:numberPQbus+1
        k1=PQ(m-1);
        for n=2:numberPQbus+1
            k2=PQ(n-1);
            if k1==k2
        J22(m-1,n-1)=Qi(k1)/V(k1)-Bi(k1,k2)*V(k1);
    else
        J22(m-1,n-1)=V(k1)*Yabs(k1,k2)*sin(Vangle(k1)-Vangle(k2)-Yangle(k1,k2));
    end
        end
    end

    J=[J11 J12; J21 J22];



    X0=[Vangle(2:end);V(PQ)];    % Create a vector that made of Vangle and V
    X=X0+inv(J)*(dm);

    Vangle0=X(1:numberbus-1);    % Get the Vangle after the calculation
    V0=X(numberbus:end);         % Get the V after the calculation



    Vangle=[0;Vangle0];          % Get the Vagnle excepting slack bus
    V=node(:,10);                % Get the V from the node data
    V(~V)=V0;                    % Use the V0 to instead of 0


    %%%%%%    Get a new P and Q after the calculation %%%%%%%%%%%%%
      Pi=zeros(numberbus,1);
      Qi=zeros(numberbus,1);
    for m=1:numberbus
        for n=1:numberbus
            Pi(m)=Pi(m)+Yabs(m,n)*V(m)*V(n)*cos(Vangle(m)-Vangle(n)-Yangle(m,n));
            Qi(m)=Qi(m)+Yabs(m,n)*V(m)*V(n)*sin(Vangle(m)-Vangle(n)-Yangle(m,n));

        end

    end



    dP=P0-Pi;
    dQ=Q0-Qi;
```

```
      dm=[dP(2:end);dQ(PQ)];    % Mismatches

      Tol=dm;
      Iter=Iter+1;

  end


    PG=Pi+node(:,5)/100;


    if P12 == 0
    P12_max = 0;
    else
    P12_max = P12;
    end



    dgx =-J;
    dgu = [1, zeros(1, 21)];
    dfu = 0.0096 * P2 * 100 + 6.4;
    dP1x = [];

    for i = 2:numberbus
    dP1x = [dP1x; Yabs(1,i) * V(1) * V(i) * sin(Vangle(1) - Vangle(i) - Yangle(1,i))];
    end

    for j = PQ'
    dP1x = [dP1x; Yabs(1,j) * V(1) * cos(Vangle(1) - Vangle(j) - Yangle(1,j))];
    end



    P12_tem = (Yabs(1,2) * V(1) * V(2) * cos(Vangle(1) - Vangle(2) - Yangle(1,2)) - Yabs(1,2) * V(1)^2 * cos(Yangle(1,2))) * 100;

    if abs(P12) > P12_max

    dP12_x = [Yabs(1,2) * V(1) * V(2) * sin(Vangle(1) - Vangle(2) - Yangle(1,2)) zeros(1,21)]';
    else
    dP12_x = 0;
    end

    dfx = (0.008 * P1 + 8) * dP1x + 2 * (P12_tem - P12_max) * dP12_x;
    dlu = (dfu - dgu * inv(dgx') * dfx);
    P1 = PG(1) * 100;
    P2 = P2 -    dlu/100;
    lu=dlu;



     end

     P1_3=P1;
     P2_3 = P2 * 100;
     P12_3 = (Yabs(1,2) * V(1) * V(2) * cos(Vangle(1) - Vangle(2) - Yangle(1,2)) - Yabs(1,2)* V(1)^2 * cos(Yangle(1,2))) * 100;
     F_3 = 0.004 * P1^2 + 8 * P1 + 0.0048 * P2_3^2 + 6.4 * P2_3;



     %% Solve OPF with line constraint %%

     P2 = P2/100;
     P12 = 5;

     lu= 1e3;


      V=node(:,10);
      V(~V)=1;                                  % The initial voltage from 1
      Vangle=zeros(numberbus,1);                % The initial angle of voltage from 0


      while abs(lu) > 1e-4



             if P2 > 0
              PG0(2) = P2;
             end

             P0=PG0-node(:,5)/100;

             %%%%%%%%%% Get the initial P and Q Value %%%%%%%%%%%%%%%%%%%%%%%%%%
      Pi=zeros(numberbus,1);
      Qi=zeros(numberbus,1);
      for m=1:numberbus
          for n=1:numberbus
              Pi(m)=Pi(m)+Yabs(m,n)*V(m)*V(n)*cos(Vangle(m)-Vangle(n)-Yangle(m,n));
              Qi(m)=Qi(m)+Yabs(m,n)*V(m)*V(n)*sin(Vangle(m)-Vangle(n)-Yangle(m,n));

          end

      end

      dP=P0-Pi;
      dQ=Q0-Qi;
      dm=[dP(2:end);dQ(PQ)];    % Get the first time mismatches



            Tol=1;
            Iter=0;
  while max(abs(Tol))>1e-2 && (Iter<10)
      %%%%%%%% Jacobian %%%%%%%%
      %%%%%%%% J11 %%%%%%%%%%%
      for m=1:(numberPQbus+numberPVbus)
          for n=1:(numberPQbus+numberPVbus)
      if m==n
          J11(m,n)=-Qi(m+1)-Bi(m+1,n+1)*V(m+1)^2;
      else
          J11(m,n)=V(m+1)*V(n+1)*Yabs(m+1,n+1)*sin(Vangle(m+1)-Vangle(n+1)-Yangle(m+1,n+1));
      end
          end
      end



      %%%%%%%% J12 %%%%%%%%%%%
      for m=2:(numberPQbus+numberPVbus+1)
          for n=2:numberPQbus+1
              k=PQ(n-1);
              if k==m
          J12(m-1,n-1)=Pi(k)/V(k)+Gi(m,k)*V(k);
      else
          J12(m-1,n-1)=V(m)*Yabs(m,k)*cos(Vangle(m)-Vangle(k)-Yangle(m,k));
      end
      end
      end



      %%%%%%%% J21 %%%%%%%%%%%
```

```
for m=2:numberPQbus+1
    k=PQ(m-1);
    for n=2:(numberPQbus+numberPVbus+1)
        if n==k
    J21(m-1,n-1)=Pi(k)-Gi(k,n)*V(k)^2;
else
    J21(m-1,n-1)=-1*V(k)*V(n)*Yabs(k,n)*cos(Vangle(k)-Vangle(n)-Yangle(k,n));
end
    end
end



%%%%%%%% J22 %%%%%%%%%%
for m=2:numberPQbus+1
    k1=PQ(m-1);
    for n=2:numberPQbus+1
        k2=PQ(n-1);
        if k1==k2
    J22(m-1,n-1)=Qi(k1)/V(k1)-Bi(k1,k2)*V(k1);
else
    J22(m-1,n-1)=V(k1)*Yabs(k1,k2)*sin(Vangle(k1)-Vangle(k2)-Yangle(k1,k2));
end
    end
end

J=[J11 J12; J21 J22];




X0=[Vangle(2:end);V(PQ)];    % Create a vector that made of Vangle and V
X=X0+inv(J)*(dm);

Vangle0=X(1:numberbus-1);    % Get the Vangle after the calculation
V0=X(numberbus:end);          % Get the V after the calculation





Vangle=[0;Vangle0];          % Get the Vagnle excepting slack bus
V=node(:,10);                % Get the V from the node data
V(~V)=V0;                    % Use the V0 to instead of 0



%%%%%%    Get a new P and Q after the calculation %%%%%%%%%%%%%
  Pi=zeros(numberbus,1);
```

```
     Qi=zeros(numberbus,1);
    for m=1:numberbus
        for n=1:numberbus
            Pi(m)=Pi(m)+Yabs(m,n)*V(m)*V(n)*cos(Vangle(m)-Vangle(n)-Yangle(m,n));
            Qi(m)=Qi(m)+Yabs(m,n)*V(m)*V(n)*sin(Vangle(m)-Vangle(n)-Yangle(m,n));

        end

    end



    dP=P0-Pi;
    dQ=Q0-Qi;
    dm=[dP(2:end);dQ(PQ)];    % Mismatches

    Tol=dm;
    Iter=Iter+1;

end


  PG=Pi+node(:,5)/100;


  if P12 == 0
  P12_max = 0;
  else
  P12_max = P12;
  end


  dgx =-J;
  dgu = [1, zeros(1, 21)];
  dfu = 0.0096 * P2 * 100 + 6.4;
  dP1x = [];

  for i = 2:numberbus
  dP1x = [dP1x; Yabs(1,i) * V(1) * V(i) * sin(Vangle(1) - Vangle(i) - Yangle(1,i))];
  end

  for j = PQ'
  dP1x = [dP1x; Yabs(1,j) * V(1) * cos(Vangle(1) - Vangle(j) - Yangle(1,j))];
  end
```

```
    P12_tem = (Yabs(1,2) * V(1) * V(2) * cos(Vangle(1) - Vangle(2) - Yangle(1,2)) - Yabs(1,2) * V(1)^2 * cos(Yangle(1,2))) * 100;

    if abs(P12) > P12_max

    dP12_x = [Yabs(1,2) * V(1) * V(2) * sin(Vangle(1) - Vangle(2) - Yangle(1,2)) zeros(1,21)]';
    else
    dP12_x = 0;
    end

    dfx = (0.008 * P1 + 8) * dP1x + 2 * (P12_tem - P12_max) * dP12_x;
    dlu = (dfu - dgu * inv(dgx') * dfx);

    P1 = PG(1) * 100;
    P2 = P2 -    dlu/100;
    lu=dlu;



     end

     P1_4=P1;
     P2_4 = P2 * 100;
     P12_4 = (Yabs(1,2) * V(1) * V(2) * cos(Vangle(1) - Vangle(2) - Yangle(1,2)) - Yabs(1,2)* V(1)^2 * cos(Yangle(1,2))) * 100;
     F_4 = 0.004 * P1^2 + 8 * P1 + 0.0048 * P2_4^2 + 6.4 * P2_4;



     fprintf('P1_1 is %.3f MW\n', P1_1);
     fprintf('P2_1 is %.3f MW\n', P2_1);
     fprintf('F_1 is %.3f f\n', F_1);

     fprintf('P1_2 is %.3f MW\n', P1_2);
     fprintf('P2_2 is %.3f MW\n', P2_2);
     fprintf('F_2 is %.3f f\n', F_2);

     fprintf('P1_3 is %.3f MW\n', P1_3);
     fprintf('P2_3 is %.3f MW\n', P2_3);
     fprintf('P12_3 is %.3f MW\n', P12_3);
     fprintf('F_3 is %.3f f\n', F_3);
```

```
fprintf('P1_4 is %.3f MW\n', P1_4);
fprintf('P2_4 is %.3f MW\n', P2_4);
fprintf('P12_4 is %.3f MW\n', P12_4);
fprintf('F_4 is %.3f f\n', F_4);
```