clc;

clear all;

Y=[20-23i -10+20i -10+3i

-10+20i 26-52i -16+32i

-10+3i -16+32i 26-35i];

%Conductance & Succeptance Values

for i=1:3;

for j=1:3;

G(i,j)=real(Y(i,j));

B(i,j)=imag(Y(i,j));

end

end

% Given Specification in pu(known)

V1MAG=1.0;

ANG1=0;

V2MAG=1.05;

P2sp=1.0;

P3sp=-0.9;

Q3sp=-0.6;

%Initialization

Iter=0;

Iter\_Max=6;

tol=1;

delANG2=0;

delANG3=0;

delMAG3=0;

%to be determined

ANG2=0;

ANG3=0;

V3MAG=1;

%start Iteration Process

while(Iter<Iter\_Max)

ANG2=ANG2+delANG2;

ANG3=ANG3+delANG3;

V3MAG=V3MAG+delMAG3;

% Creation of Jacobian J

J(1,1)=V2MAG\*(V1MAG\*(B(2,1)\*cos(ANG2-ANG1)-G(2,1)\*sin(ANG2-ANG1))+V3MAG\*(B(2,3)\*cos(ANG2-ANG3)-G(2,3)\*sin(ANG2-ANG3)));

J(1,2)=V2MAG\*V3MAG\*(G(2,3)\*sin(ANG2-ANG3)-B(2,3)\*cos(ANG2-ANG3));

J(1,3)=V2MAG\*(G(2,3)\*cos(ANG2-ANG3)+B(2,3)\*sin(ANG2-ANG3));

J(2,1)=V3MAG\*V2MAG\*(G(3,2)\*sin(ANG3-ANG2)-B(3,2)\*cos(ANG3-ANG2));

J(2,2)=V3MAG\*(V1MAG\*(B(3,1)\*cos(ANG3-ANG1)-G(3,1)\*sin(ANG3-ANG1))+V2MAG\*(B(3,2)\*cos(ANG3-ANG2)-G(3,2)\*sin(ANG3-ANG2)));

J(2,3)=2\*G(3,3)\*V3MAG+V1MAG\*(G(3,1)\*cos(ANG3-ANG1)+B(3,1)\*sin(ANG3-ANG1)+V2MAG\*(G(3,2)\*cos(ANG3-ANG2)+B(3,2)\*sin(ANG3-ANG2)));

J(3,1)=-V3MAG\*V2MAG\*(G(3,2)\*cos(ANG3-ANG2)+B(3,2)\*sin(ANG3-ANG2));

J(3,2)=V3MAG\*(V1MAG\*(B(3,1)\*cos(ANG3-ANG1)+G(3,1)\*sin(ANG3-ANG1))+V2MAG\*(B(3,2)\*cos(ANG3-ANG2)+G(3,2)\*sin(ANG3-ANG2)));

J(3,3)=-2\*B(3,3)\*V3MAG+V1MAG\*(G(3,1)\*cos(ANG3-ANG1)-B(3,1)\*sin(ANG3-ANG1)+V2MAG\*(G(3,2)\*cos(ANG3-ANG2)-B(3,2)\*sin(ANG3-ANG2)));

J = [J(1,1) J(1,2) J(1,3); J(2,1) J(2,2) J(2,3); J(3,1) J(3,2) J(3,3)];

% calculation of updated voltages with angles

V(1) = V1MAG\*exp(1i\*ANG1);

V(2) = V2MAG\*exp(1i\*ANG2);

V(3) = V3MAG\*exp(1i\*ANG3);

V = [V(1); V(2); V(3)];

%Current injections at each bus based on updated voltages and angles

I = Y\*V;

% calculations of P and Q

S(1) = V(1)\*conj(I(1));

S(2) = V(2)\*conj(I(2));

S(3) = V(3)\*conj(I(3));

%Mismatches

Mismatch(1) = P2sp-real(S(2));

Mismatch(2) = P3sp-real(S(3));

Mismatch(3) = Q3sp-imag(S(3));

%calculate the deltaANG and deltaVMAG

del = inv(J).\*Mismatch;

delANG2 = del(1,1);

delANG3 = del(2,1);

delMAG3 = del(3,1);

Iter = Iter + 1;

end

disp('-------Voltage and Delta is given as------');

for i=1:3;

fprintf('V%d=%f\t\t DEL%d=%f\n', i, V(i), i, angle(V(i)));

end

% calculate power flow of transmission lines

disp('--------Power flow is given by--------');

for i=1:3;

for j=1:3;

if(i~=j)

P(i,j)=real(V(i)\*conj(Y(i,j)\*(V(i)-V(j))));

fprintf('P(%d,%d)=%f\t\t', i, j, P(i,j));

Q(i,j)=imag(V(i)\*conj(Y(i,j)\*(V(i)-V(j))));

fprintf('Q(%d,%d)=%f\t\t\n', i, j, Q(i,j));

end

end

end