% IEEE test system tested with FACTS devicec(UPFC) using NR method

clc

clear

basemva = 100;

accuracy = 0.0001;

maxiter = 100;

r=0.01;

gammad=95;

gamma=(pi/180)\*gammad;

p=6;

q=9;

if p==q

fprintf('ERROR!!\n\nUPFC cannot be connected to the same bus\n')

%if the UPFC is added to the same bus i.e which is an error, then this message is printed

fprintf('Hence the power flow solution is calculated for without UPFC\n\n\nPress Enter to continue\n')

pause

end

% Bus data is to be given here

% Bus Bus |V| Ang ---Load--- ---Gen--- Gen Mvar Injected

% No. code p.u. Deg MW Mvar MW Mvar Min Max Mvar

busdata=[

1 1 1.06 0.0 0.0 0.0 124.2960 0.0 0 0 0

2 2 1.043 0.0 21.70 12.7 73.0390 0.0 -40 50 0

3 0 1.0 0.0 2.4 1.2 0.0 0.0 0 0 0

4 0 1.06 0.0 7.6 1.6 0.0 0.0 0 0 0

5 2 1.01 0.0 94.2 19.0 45.1985 0.0 -40 40 0

6 0 1.0 0.0 0.0 0.0 0.0 0.0 0 0 0

7 0 1.0 0.0 22.8 10.9 0.0 0.0 0 0 0

8 2 1.01 0.0 30.0 30.0 12.3877 0.0 -30 40 0

9 0 1.0 0.0 0.0 0.0 0.0 0.0 0 0 0

10 0 1.0 0.0 5.8 2.0 0.0 0.0 0 0 19

11 2 1.082 0.0 0.0 0.0 10.0000 0.0 -6 24 0

12 0 1.0 0 11.2 7.5 0 0 0 0 0

13 2 1.071 0 0 0.0 18.6398 0 -6 24 0

14 0 1 0 6.2 1.6 0 0 0 0 0

15 0 1 0 8.2 2.5 0 0 0 0 0

16 0 1 0 3.5 1.8 0 0 0 0 0

17 0 1 0 9.0 5.8 0 0 0 0 0

18 0 1 0 3.2 0.9 0 0 0 0 0

19 0 1 0 9.5 3.4 0 0 0 0 0

20 0 1 0 2.2 0.7 0 0 0 0 0

21 0 1 0 17.5 11.2 0 0 0 0 0

22 0 1 0 0 0.0 0 0 0 0 0

23 0 1 0 3.2 1.6 0 0 0 0 0

24 0 1 0 8.7 6.7 0 0 0 0 4.3

25 0 1 0 0 0.0 0 0 0 0 0

26 0 1 0 3.5 2.3 0 0 0 0 0

27 0 1 0 0 0.0 0 0 0 0 0

28 0 1 0 0 0.0 0 0 0 0 0

29 0 1 0 2.4 0.9 0 0 0 0 0

30 0 1 0 10.6 1.9 0 0 0 0 0

];

% | From | To | R | X | B/2 | X'mer |

% | Bus | Bus | pu | pu | pu | TAP (a) |

linedata =[

1 2 0.0192 0.0575 0.02640 1

1 3 0.0452 0.1852 0.02040 1

2 4 0.0570 0.1737 0.01840 1

3 4 0.0132 0.0379 0.00420 1

2 5 0.0472 0.1983 0.02090 1

2 6 0.0581 0.1763 0.01870 1

4 6 0.0119 0.0414 0.00450 1

5 7 0.0460 0.1160 0.01020 1

6 7 0.0267 0.0820 0.00850 1

6 8 0.0120 0.0420 0.00450 1

6 9 0.0 0.2080 0.0 0.978

6 10 0 .5560 0 0.969

9 11 0 .2080 0 1

9 10 0 .1100 0 1

4 12 0 .2560 0 0.932

12 13 0 .1400 0 1

12 14 .1231 .2559 0 1

12 15 .0662 .1304 0 1

12 16 .0945 .1987 0 1

14 15 .2210 .1997 0 1

16 17 .0824 .1923 0 1

15 18 .1073 .2185 0 1

18 19 .0639 .1292 0 1

19 20 .0340 .0680 0 1

10 20 .0936 .2090 0 1

10 17 .0324 .0845 0 1

10 21 .0348 .0749 0 1

10 22 .0727 .1499 0 1

21 22 .0116 .0236 0 1

15 23 .1000 .2020 0 1

22 24 .1150 .1790 0 1

23 24 .1320 .2700 0 1

24 25 .1885 .3292 0 1

25 26 .2544 .3800 0 1

25 27 .1093 .2087 0 1

28 27 0 .3960 0 0.968

27 29 .2198 .4153 0 1

27 30 .3202 .6027 0 1

29 30 .2399 .4533 0 1

8 28 .0636 .2000 0.0214 1

6 28 .0169 .0599 0.065 1

];

G(1)=0; G(2)=0; G(3)=1;

% This program obtains the Bus Admittance Matrix for power flow solution

j=sqrt(-1);

i = sqrt(-1);

nl = linedata(:,1);

nr = linedata(:,2);

R = linedata(:,3);

X = linedata(:,4);

Bc = j\*linedata(:,5);

a = linedata(:, 6);

nbr=length(linedata(:,1));

nbus = max(max(nl), max(nr));

Z = R + j\*X ;

y= ones(nbr,1)./Z; %branch admittance;

% Here RR & XX takes the right value accordingly when UPFC is inserted

% between p and q bus, RR and XX corresponds to the transmission line

%resistance and reactance between p and q bus., i.e.,where the UPFC is

%embedded in the power system

%Here starts the logic

RR=0; XX=0;

for k=1:nbr

if nl(k)==p & nr(k)==q

RR=linedata(k,3);

XX=linedata(k,4);

elseif nr(k)==p & nl(k)==q

RR=linedata(k,3);

XX=linedata(k,4);

else end

end

%Here Ends the logic

for n = 1:nbr

if a(n) <= 0

a(n) = 1;

else end

ZZ=(RR^2+XX^2);

Zi=1/(ZZ);

W=XX\*cos(gamma)-RR\*sin(gamma);

% initialize Ybus to zero

Ybus=zeros(nbus,nbus);

% formation of the off diagonal elements

for k=1:nbr

Ybus(nl(k),nr(k))=Ybus(nl(k),nr(k))-y(k)/a(k);

Ybus(nr(k),nl(k))=Ybus(nl(k),nr(k));

end

end

% formation of the diagonal elements

for n=1:nbus

for k=1:nbr

if nl(k)==n

Ybus(n,n) = Ybus(n,n)+y(k)/(a(k)^2) + Bc(k);

elseif nr(k)==n

Ybus(n,n) = Ybus(n,n)+y(k) +Bc(k);

else, end

end

end

clear Pgg

% Power flow solution by Newton-Raphson method

ns=0;

ng=0;

Vm=0;

delta=0;

yload=0;

deltad=0;

nbus = length(busdata(:,1));

%=========================================================================

for k=1:nbus

kb(k)=busdata(k,2);

end

ii=0;

jj=0;

ij=0;

t=1;

for z=1:nbus

Pq(z)=0;

end

for k=1:nbus

pq(k)=0;

end

for k=1:nbus

if kb(k)==0

PQ=k;

Pq(k)=PQ;

end

end

for z=1:nbus

if Pq(z)~=0

pq(t)=Pq(z);

t=t+1;

else

end

end

%--------------------------------------------------------------------------

%-------Modifying jacobian matrix

%i.e for taking ll,lk,lm,nn

if (kb(q)==0)

for k=1:nbus

ii=ii+1;

if pq(k)==q

jj=ii;

end

end

end

ii=0;

if (kb(p)==0)

for k=1:nbus

ii=ii+1;

if pq(k)==p

ij=ii;

end

end

end

ij;

jj;

%--------------------------------------------------------------------------

for k=1:nbus

n=busdata(k,1);

kb(n)=busdata(k,2);

Vm(n)=busdata(k,3);

delta(n)=busdata(k, 4);

Pd(n)=(1.5)\*busdata(k,5);

Qd(n)=(1.5)\*busdata(k,6);

Pg(n)=busdata(k,7);

Qg(n) = busdata(k,8);

Qmin(n)=busdata(k, 9);

Qmax(n)=busdata(k, 10);

Qsh(n)=busdata(k, 11);

if Vm(n) <= 0

Vm(n) = 1.0;

V(n) = 1 + j\*0;

else

delta(n) = pi/180\*delta(n);

V(n) = Vm(n)\*(cos(delta(n)) + j\*sin(delta(n)));

P(n)=(Pg(n)-Pd(n))/basemva;

Q(n)=(Qg(n)-Qd(n)+ Qsh(n))/basemva;

S(n) = P(n) + j\*Q(n);

end

end

for k=1:nbus

if kb(k) == 1

ns = ns+1;

else, end

if kb(k) == 2

ng = ng+1;

else, end

ngs(k) = ng;

nss(k) = ns;

end

Ym=abs(Ybus);

t = angle(Ybus);

m=2\*nbus-ng-2\*ns;

maxerror = 1;

converge=1;

iter = 0;

% Start of iterations

clear A DC J DX

while maxerror >= accuracy & iter <= maxiter % Test for max. power mismatch

for i=1:m

for k=1:m

A(i,k)=0; %Initializing Jacobian matrix;

end

end

iter = iter+1;

for n=1:nbus

nn=n-nss(n);

lm=nbus+n-ngs(n)-nss(n)-ns;

J11=0;

J22=0;

J33=0;

J44=0;

% hii=(-1.02\*r\*Vm(p)\*Vm(q)\*Zi)\*((XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))));

% hij=(1.02\*r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q)));

% hji=(r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q)));

% hjj=(r\*Vm(p)\*Vm(q)\*Zi)\*(RR\*sin(delta(p)+gamma-delta(q))-XX\*cos(delta(p)+gamma-delta(q)));

% nii=(1.02\*r\*Vm(q)\*Zi)\*(RR\*cos(delta(p)+gamma-delta(q))-XX\*sin(delta(p)+gamma-delta(q)))-(2.04\*RR\*r^2\*Vm(p)\*Zi)+(0.04\*r\*Vm(p)\*XX\*sin(gamma)\*Zi)-(4.04\*r\*RR\*Vm(p)\*cos(gamma)\*Zi);

% nij=(-1.02\*r\*Vm(p)\*Zi)\*(XX\*sin(delta(p)+gamma-delta(q))-RR\*cos(delta(p)+gamma-delta(q)));

% nji=(r\*Vm(q)\*Zi)\*(RR\*cos(delta(p)+gamma-delta(q))+XX\*sin(delta(p)+gamma-delta(q)));

% njj=(r\*Vm(p)\*Zi)\*(RR\*cos(delta(p)+gamma-delta(q))+XX\*sin(delta(p)+gamma-delta(q)));

% jii=0;

% jij=0;

% jji=(-r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*sin(delta(p)+gamma-delta(q))+RR\*cos(delta(p)+gamma-delta(q)));

% jjj=(r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*sin(delta(p)+gamma-delta(q))+RR\*cos(delta(p)+gamma-delta(q)));

% lii=(-2\*r\*Vm(p)\*Zi)\*(XX\*cos(gamma)-RR\*sin(gamma));

% lij=0;

% lji=(r\*Vm(q)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q)));

% ljj=(r\*Vm(p)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q))) ;

%hii=(G(1)\*Zi)\*[r\*Vm(p)^3\*Vm(q)]\*[XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q)));

hii=[G(1)\*[[-r\*Vm(p)^3\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]]+[G(2)\*[[-r\*Vm(p)^2\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]]+[G(3)\*[[-r\*Vm(p)\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]];

%hij=(1.02\*r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q)));

hij=[G(1)\*[[r\*Vm(p)^3\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]]+[G(2)\*[[r\*Vm(p)^2\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]]+[G(3)\*[[r\*Vm(p)\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]];

% hji=(r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q)));

hji=[G(1)\*[[r\*Vm(p)^3\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q))]]]+[G(2)\*[[r\*Vm(p)^2\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q))]]]+[G(3)\*[[r\*Vm(p)\*Vm(q)\*Zi]\*[XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q))]]];

%hjj=(r\*Vm(p)\*Vm(q)\*Zi)\*(RR\*sin(delta(p)+gamma-delta(q))-XX\*cos(delta(p)+gamma-delta(q)));

hjj=[G(1)\*[[r\*Vm(p)^3\*Vm(q)\*Zi]\*[-XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]]+[G(2)\*[[r\*Vm(p)^2\*Vm(q)\*Zi]\*[-XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]]+[G(3)\*[[r\*Vm(p)\*Vm(q)\*Zi]\*[-XX\*cos(delta(p)+gamma-delta(q))+RR\*sin(delta(p)+gamma-delta(q))]]];

%nii=(1.02\*r\*Vm(q)\*Zi)\*(RR\*cos(delta(p)+gamma-delta(q))-XX\*sin(delta(p)+gamma-delta(q)))-(2.04\*RR\*r^2\*Vm(p)\*Zi)+(0.04\*r\*Vm(p)\*XX\*sin(gamma)\*Zi)-(4.04\*r\*RR\*Vm(p)\*cos(gamma)\*Zi);

nii=[G(1)\*[[-8\*r\*Vm(p)^3\*RR\*(1/(RR+XX))\*cos(gamma)]+[4\*r\*Vm(p)^3\*Zi\*XX\*sin(gamma)]-[4\*RR\*r^2\*Vm(p)^3\*Zi]-[3\*r\*Vm(p)^2\*Vm(q)\*Zi\*XX\*sin(delta(p)+gamma-delta(q))]+[3\*r\*Vm(p)^2\*Vm(q)\*Zi\*RR\*cos(delta(p)+gamma-delta(q))]]]+[G(2)\*[-6\*r\*Vm(p)^2\*RR\*(1/(RR+XX))\*cos(gamma)]+[3\*r\*Vm(p)^2\*Zi\*XX\*sin(gamma)]-[3\*RR\*r^2\*Zi\*Vm(p)^2]-[2\*r\*Vm(p)\*Vm(q)\*Zi\*XX\*sin(delta(p)+gamma-delta(q))]+[2\*r\*Vm(p)\*Vm(q)\*Zi\*RR\*cos(delta(p)+gamma-delta(q))]]+[G(3)\*[-4\*r\*Vm(p)\*RR\*(1/(RR+XX))\*cos(gamma)]+[2\*r\*Vm(p)\*Zi\*XX\*sin(gamma)]-[2\*RR\*r^2\*Zi\*Vm(p)]-[r\*Vm(q)\*Zi\*XX\*sin(delta(p)+gamma-delta(q))]+[r\*Vm(q)\*Zi\*RR\*cos(delta(p)+gamma-delta(q))]];

% nij=(-1.02\*r\*Vm(p)\*Zi)\*(XX\*sin(delta(p)+gamma-delta(q))-RR\*cos(delta(p)+gamma-delta(q)));

nij=(r\*Zi)\*[G(1)\*[(Vm(p)^3\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)^3\*cos(delta(p)+gamma-delta(q)))]+[G(2)\*[(Vm(p)^2\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)^2\*cos(delta(p)+gamma-delta(q)))]]+[G(3)\*[(Vm(p)\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)\*cos(delta(p)+gamma-delta(q)))]]];

%nji=(r\*Vm(q)\*Zi)\*(RR\*cos(delta(p)+gamma-delta(q))+XX\*sin(delta(p)+gamma-delta(q)));

nji=(r\*Vm(q)\*Zi)\*[[G(1)\*[(3\*Vm(p)^2\*[XX\*sin(delta(p)+gamma-delta(q))+RR\*cos(delta(p)+gamma-delta(q))])]]+[G(2)\*[(2\*Vm(p)\*[XX\*sin(delta(p)+gamma-delta(q))+RR\*cos(delta(p)+gamma-delta(q))])]]+[G(3)\*[XX\*sin(delta(p)+gamma-delta(q))+RR\*cos(delta(p)+gamma-delta(q))]]];

%njj=(r\*Vm(p)\*Zi)\*(RR\*cos(delta(p)+gamma-delta(q))+XX\*sin(delta(p)+gamma-delta(q)));

njj=(r\*Zi)\*[G(1)\*[(Vm(p)^3\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)^3\*cos(delta(p)+gamma-delta(q)))]+[G(2)\*[(Vm(p)^2\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)^2\*cos(delta(p)+gamma-delta(q)))]]+[G(3)\*[(Vm(p)\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)\*cos(delta(p)+gamma-delta(q)))]]];

jii=0;

jij=0;

% jji=(-r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*sin(delta(p)+gamma-delta(q))+RR\*cos(delta(p)+gamma-delta(q)));

jji=(r\*Vm(q)\*Zi)\*[G(1)\*[(-Vm(p)^3\*XX\*sin(delta(p)+gamma-delta(q))-RR\*Vm(p)^3\*cos(delta(p)+gamma-delta(q)))]+[G(2)\*[(-Vm(p)^2\*XX\*sin(delta(p)+gamma-delta(q))-RR\*Vm(p)^2\*cos(delta(p)+gamma-delta(q)))]]+[G(3)\*[(-Vm(p)\*XX\*sin(delta(p)+gamma-delta(q))-RR\*Vm(p)\*cos(delta(p)+gamma-delta(q)))]]];

%jjj=(r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*sin(delta(p)+gamma-delta(q))+RR\*cos(delta(p)+gamma-delta(q)));

jjj=(r\*Vm(q)\*Zi)\*[[G(1)\*[(Vm(p)^3\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)^3\*cos(delta(p)+gamma-delta(q)))]+[G(2)\*[(Vm(p)^2\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)^2\*cos(delta(p)+gamma-delta(q)))]]+[G(3)\*[(Vm(p)\*XX\*sin(delta(p)+gamma-delta(q))+RR\*Vm(p)\*cos(delta(p)+gamma-delta(q)))]]]];

% lii=(-2\*r\*Vm(p)\*Zi)\*(XX\*cos(gamma)-RR\*sin(gamma));

lii=(r\*Vm(q)\*Zi)\*([G(1)\*[(-3\*Vm(p)^2\*XX\*cos(gamma))+(3\*Vm(p)^2\*RR\*sin(gamma))]]+[G(2)\*[(-2\*Vm(p)\*XX\*cos(gamma))+(2\*RR\*Vm(p)\*sin(gamma))]]+[G(3)\*[(-XX\*cos(gamma))+RR\*sin(gamma)]]);

%lij=0;

lij=(r\*Zi)\*([G(1)\*[(-Vm(p)^3\*XX\*cos(gamma))+(Vm(p)^3\*RR\*sin(gamma))]]+[G(2)\*[(-Vm(p)^2\*XX\*cos(gamma))+(RR\*Vm(p)^2\*sin(gamma))]]+[G(3)\*[(-Vm(p)\*XX\*cos(gamma))+(Vm(p)\*RR\*sin(gamma))]]);

%lji=(r\*Vm(q)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q)));

lji=(r\*Vm(q)\*Zi)\*[G(1)\*[(3\*Vm(p)^2\*XX\*cos(delta(p)+gamma-delta(q))-3\*RR\*Vm(p)^2\*sin(delta(p)+gamma-delta(q)))]+[G(2)\*[(2\*Vm(p)\*XX\*cos(delta(p)+gamma-delta(q))-2\*RR\*Vm(p)\*sin(delta(p)+gamma-delta(q)))]]+[G(3)\*[XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q))]]];

%ljj=(r\*Vm(p)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q))) ;

ljj=(r\*Zi)\*(G(1)\*[(Vm(p)^3\*XX\*cos(delta(p)+gamma-delta(q)))-(RR\*Vm(p)^3\*sin(delta(p)+gamma-delta(q)))]+[G(2)\*[(Vm(p)^2\*XX\*cos(delta(p)+gamma-delta(q)))-(RR\*Vm(p)^2\*sin(delta(p)+gamma-delta(q)))]]+[G(3)\*[(Vm(p)\*XX\*cos(delta(p)+gamma-delta(q)))-(Vm(p)\*RR\*sin(delta(p)+gamma-delta(q)))]]);

for i=1:nbr

if nl(i) == n | nr(i) == n

if nl(i) == n, l = nr(i); end

if nr(i) == n, l = nl(i); end

J11=J11+ Vm(n)\*Vm(l)\*Ym(n,l)\*sin(t(n,l)- delta(n) + delta(l));

J33=J33+ Vm(n)\*Vm(l)\*Ym(n,l)\*cos(t(n,l)- delta(n) + delta(l));

if kb(n)~=1

J22=J22+ Vm(l)\*Ym(n,l)\*cos(t(n,l)- delta(n) + delta(l));

J44=J44+ Vm(l)\*Ym(n,l)\*sin(t(n,l)- delta(n) + delta(l));

else, end

if kb(n) ~= 1 & kb(l) ~=1

lk = nbus+l-ngs(l)-nss(l)-ns;

ll = l -nss(l);

% off diagonalelements of J1

A(nn, ll) =-Vm(n)\*Vm(l)\*Ym(n,l)\*sin(t(n,l)- delta(n) + delta(l));

if kb(l) == 0 % off diagonal elements of J2

A(nn, lk) =Vm(n)\*Ym(n,l)\*cos(t(n,l)- delta(n) + delta(l));

end

if kb(n) == 0 % off diagonal elements of J3

A(lm, ll) =-Vm(n)\*Vm(l)\*Ym(n,l)\*cos(t(n,l)- delta(n)+delta(l));

end

if kb(n) == 0 & kb(l) == 0 % off diagonal elements of J4

A(lm, lk) =-Vm(n)\*Ym(n,l)\*sin(t(n,l)- delta(n) + delta(l));

end

%modifying Jacobian elements starts here

if kb(p)==2 & kb(q)==2 %if UPFC is connected between PV-PV bus

%if UPFC is connected between pv-pv bus,only H-matrix gets affected

% modifying the off-diagonal elements of H-matrix starts here

if nn==(p-1) & ll==(q-1)

A(nn,ll)=A(nn,ll)+hij;

A(ll,nn)=A(ll,nn)+hji;

end

%modifying the off-diagonal elements of H-matrix ends here

elseif (kb(p)==2 & kb(q)==0) | (kb(p)==0 & kb(q)==2) % if UPFC is connected between PV-PQ bus or PQ-PV bus

%if upfc is connected between pv-pq bus, some of the elements in

%H,N,J,L matrices does not get affected

%modifying the off-diagonal elements of H-matrix starts here

if nn==(p-1) & ll==(q-1)

A(nn,ll)=A(nn,ll)+hij;

A(ll,nn)=A(ll,nn)+hji;

end

%modifying the off-diagonal elements of H-matrix ends here

%modifying the off-diagonal elements of N-matrix starts here

if nn==(p-1) & lk==(nbus-1+jj)

A(nn,lk)=A(nn,lk)+nij;

end

%modifying the off-diagonal elements of N-matrix ends here

%modifying the off-diagonal elements of J-matrix starts here

if lm==(nbus-1+jj) & ll==(p-1)

A(lm,ll)=A(lm,ll)+jji;

end

%modifying the off-diagonal elements of J-matrix ends here

%ends modifying the off-diagonal elements when upfc is connected

%between pv-pq bus & pq-pv bus as well

elseif kb(p)==2 & kb(q)==2 %if UPFC is connected between PQ-PQ bus

%modifying the off-diagonal elements of H-matrix starts here

if nn==(p-1) & ll==(q-1)

A(nn,ll)=A(nn,ll)+hij;

A(ll,nn)=A(ll,nn)+hji;

end

%modifying the off-diagonal elements of H-matrix ends here

%modifying the off-diagonal elements of N-matrix starts here

if nn==(p-1) & lk==(nbus-1+jj)

A(nn,lk)=A(nn,lk)+nij;

end

if nn==(q-1) & lk==(nbus-1+ij)

A(nn,lk)=A(nn,lk)+nji;

end

%modifying the off-diagonal elements of N-matrix ends here

%modifying the off-diagonal elements of J-matrix starts here

if lm==(nbus-1+ij) & ll==(q-1)

A(lm,ll)=A(lm,ll)+jij;

end

if lm==(nbus-1+jj) & ll==(p-1)

A(lm,ll)=A(lm,ll)+jji;

end

%modifying the off-diagonal elements of L-matrix starts here

if lm==(nbus-1+ij) & lk==(nbus-1+jj)

A(lm,lk)=A(lm,lk)+lij;

end

if lm==(nbus-1+jj) & lk==(nbus-1+ij)

A(lm,lk)=A(lm,lk)+lji;

end

%modifying the off-diagonal elements of L-matrix ends here

else end

else end

else , end

end

% Qaupfc=(-r\*Vm(p)^2)\*Zi\*W;

% Pbupfc=(r\*Vm(p)\*Vm(q)\*Zi)\*(RR\*cos(delta(p)+gamma-delta(q))+XX\*sin(delta(p)+gamma-delta(q)));

% Qbupfc=(r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*cos(delta(p)+gamma-delta(q))-RR\*sin(delta(p)+gamma-delta(q)));

% Paupfc= (-1.02\*r\*Vm(p)\*Vm(q)\*Zi)\*(XX\*sin(delta(p)+gamma-delta(q))-RR\*cos(delta(p)+gamma-delta(q)))+(-1.02\*RR\*r^2\*Vm(p)^2\*Zi)+(0.02\*r\*Vm(p)^2\*XX\*sin(gamma)\*Zi)+(-2.02\*RR\*r\*Vm(p)^2\*cos(gamma)\*Zi);

Paupfc=[G(1)\*((-2\*r\*Vm(p)^4\*RR\*cos(gamma)\*(1/(RR+XX)))+(r\*Vm(p)^4\*XX\*sin(gamma)\*Zi)-(RR\*r^2\*Vm(p)^4\*Zi)-(r\*Vm(p)^3\*Vm(q)\*Zi\*XX\*sin(delta(p)+gamma-delta(q)))+(r\*Vm(p)^3\*Vm(q)\*Zi\*RR\*cos(delta(p)+gamma-delta(q))))]+[G(2)\*((-2\*r\*Vm(p)^3\*RR\*cos(gamma)\*(1/(RR+XX)))+(r\*Vm(p)^3\*XX\*sin(gamma)\*Zi)-(RR\*r^2\*Vm(p)^3\*Zi)-(r\*Vm(p)^2\*Vm(q)\*Zi\*XX\*sin(delta(p)+gamma-delta(q)))+(r\*Vm(p)^2\*Vm(q)\*Zi\*RR\*cos(delta(p)+gamma-delta(q))))]+[G(3)\*((-2\*r\*Vm(p)^2\*RR\*cos(gamma)\*(1/(RR+XX)))+(r\*Vm(p)^2\*XX\*sin(gamma)\*Zi)-(RR\*r^2\*Vm(p)^2\*Zi)-(r\*Vm(p)\*Vm(q)\*Zi\*XX\*sin(delta(p)+gamma-delta(q)))-(r\*Vm(p)\*Vm(q)\*Zi\*RR\*cos(delta(p)+gamma-delta(q))))];

Qaupfc=(-r\*Vm(q)\*Zi)\*([G(1)\*(Vm(p)^3\*XX\*cos(gamma)-Vm(p)^3\*RR\*sin(gamma))]+[G(2)\*(Vm(p)^2\*XX\*cos(gamma)-Vm(p)^2\*RR\*sin(gamma))]+[G(3)\*(Vm(p)\*XX\*cos(gamma)-Vm(p)\*RR\*sin(gamma))]);

Pbupfc=(r\*Vm(q)\*Zi)\*([G(1)\*(Vm(p)^3\*RR\*cos(delta(p)+gamma-delta(q))+Vm(p)^3\*XX\*sin(delta(p)+gamma-delta(q)))]+[G(2)\*(Vm(p)^2\*RR\*cos(delta(p)+gamma-delta(q))+Vm(p)^2\*XX\*sin(delta(p)+gamma-delta(q)))]+[G(3)\*(Vm(p)\*RR\*cos(delta(p)+gamma-delta(q))+Vm(p)\*XX\*sin(delta(p)+gamma-delta(q)))]);

Qbupfc=(r\*Vm(q)\*Zi)\*([G(1)\*(Vm(p)^3\*XX\*cos(delta(p)+gamma-delta(q))-Vm(p)^3\*RR\*sin(delta(p)+gamma-delta(q)))]+[G(2)\*(Vm(p)^2\*XX\*cos(delta(p)+gamma-delta(q))-Vm(p)^2\*RR\*sin(delta(p)+gamma-delta(q)))]+[G(3)\*(Vm(p)\*XX\*cos(delta(p)+gamma-delta(q))+Vm(p)\*RR\*sin(delta(p)+gamma-delta(q)))]);

Pk = Vm(n)^2\*Ym(n,n)\*cos(t(n,n))+J33;

Qk = -Vm(n)^2\*Ym(n,n)\*sin(t(n,n))-J11;

if kb(n) == 1

P(n)=Pk;

Q(n) = Qk;

end % Swing bus P

if kb(n) == 2

Q(n)=Qk;

if Qmax(n) ~= 0

Qgc = Q(n)\*basemva + Qd(n) - Qsh(n);

if iter <= 7 % Between the 2th & 6th iterations

if iter > 2 % the Mvar of generator buses are

if Qgc < Qmin(n), % tested. If not within limits Vm(n)

Vm(n) = Vm(n) + 0.01; % is changed in steps of 0.01 pu to

elseif Qgc > Qmax(n), % bring the generator Mvar within

Vm(n) = Vm(n) - 0.01;end % the specified limits.

else, end

else,end

else,end

end

if kb(n) ~= 1

A(nn,nn) = J11; %diagonal elements of J1

DC(nn)=P(n)-Pk;

%modifying the daigonal elements of H-matrix starts here

%This modification of H-matrix doesnot depend whether upfc is

%connected between what type of bus.

if nn==(p-1)

A(nn,nn)=A(nn,nn)+hii;

end

if nn==(q-1)

A(nn,nn)=A(nn,nn)+hjj;

end

%modification of diagonal elements of H-matrix ends here

end

if kb(n)~=1

if n==p

DC(nn)=DC(nn)+Paupfc;

elseif n==q

DC(nn)=DC(nn)+Pbupfc;

end

end

if kb(n) == 0

A(nn,lm) = 2\*Vm(n)\*Ym(n,n)\*cos(t(n,n))+J22; %diagonal elements of J2

A(lm,nn)= J33; %diagonal elements of J3

A(lm,lm) =-2\*Vm(n)\*Ym(n,n)\*sin(t(n,n))-J44; %diagonal of elements of J4

if (kb(p)==2 & kb(q)==0) | (kb(p)==0 & kb(q)==2)%if UPFC is connected between PV-PQ bus

%modifying the diagonal elements of N-matrix starts here

if nn==(q-1) & lm==(nbus-1+jj)

A(nn,lm)=A(nn,lm)+njj;

end

%modifying the diagonal elements of N-matrix ends here

%modifying the diagonal elements of J-matrix starts here

if lm==(nbus-1+jj) & nn==(q-1)

A(lm,nn)=A(lm,nn)+jjj;

end

%modifying the diagonal elements of J-matrix ends here

%modifying the diagonal elements of L-matrix starts here

if lm==(nbus-1+jj)

A(lm,lm)=A(lm,lm)+ljj;

end

%modifying the diagonal elements of L-matrix ends here

elseif kb(p)==0 & kb(q)==0 %if UPFC is connected between PQ-PQ bus

%modifying the diagonal elements of N-matrix starts here

if nn==(p-1) & lm==(nbus-1+ij)

A(nn,lm)=A(nn,lm)+nii;

end

if nn==(q-1) & lm==(nbus-1+jj)

A(nn,lm)=A(nn,lm)+njj;

end

%modifying the diagonal elements of N-matrix ends here

%modifying the diagonal elements of J-matrix starts here

if lm==(nbus-1+ij) & nn==(p-1)

A(lm,nn)=A(lm,nn)+jii;

end

if lm==(nbus-1+jj) & nn==(q-1)

A(lm,nn)=A(lm,nn)+jjj;

end

%modifying the diagonal elements of J-matrix ends here

%modifying the diagonal elements of L-matrix starts here

if lm==(nbus-1+ij)

A(lm,lm)=A(lm,lm)+lii;

end

if lm==(nbus-1+jj)

A(lm,lm)=A(lm,lm)+ljj;

end

%modifying the diagonal elements of L-matrix ends here

else end

if n==p

DC(lm)=Q(n)+Qaupfc-Qk;

elseif n==q

DC(lm)=Q(n)+Qbupfc-Qk;

else

DC(lm)=Q(n)-Qk;

end

end

end

DX=A\DC';

for n=1:nbus

nn=n-nss(n);

lm=nbus+n-ngs(n)-nss(n)-ns;

if kb(n) ~= 1

delta(n) = delta(n)+DX(nn); end

if kb(n) == 0

Vm(n)=Vm(n)+DX(lm); end

end

maxerror=max(abs(DC));

if iter == maxiter & maxerror > accuracy

fprintf('\nWARNING: Iterative solution did not converged after ')

fprintf('%g', iter), fprintf(' iterations.\n\n')

fprintf('Press Enter to terminate the iterations and print the results \n')

converge = 0; pause, else, end

end

if converge ~= 1

tech= (' ITERATIVE SOLUTION DID NOT CONVERGE'); else,

tech=(' Power Flow Solution by Newton-Raphson Method');

end

V = Vm.\*cos(delta)+j\*Vm.\*sin(delta);

deltad=180/pi\*delta;

i=sqrt(-1);

k=0;

for n = 1:nbus

if kb(n) == 1

k=k+1;

S(n)= P(n)+j\*Q(n);

Pg(n) = P(n)\*basemva + Pd(n);

Qg(n) = Q(n)\*basemva + Qd(n) - Qsh(n);

Pgg(k)=Pg(n);

Qgg(k)=Qg(n);

elseif kb(n) ==2

k=k+1;

S(n)=P(n)+j\*Q(n);

Qg(n) = Q(n)\*basemva + Qd(n) - Qsh(n);

Pgg(k)=Pg(n);

Qgg(k)=Qg(n);

end

yload(n) = (Pd(n)- j\*Qd(n)+j\*Qsh(n))/(basemva\*Vm(n)^2);

end

busdata(:,3)=Vm'; busdata(:,4)=deltad';

Pgt = sum(Pg); Qgt = sum(Qg); Pdt = sum(Pd); Qdt = sum(Qd); Qsht = sum(Qsh);

%This program is used in conjunction with lfgauss or lf Newton

%for the computation of line flow and line losses.

SLT = 0;

fprintf('\n')

fprintf(' Line Flow and Losses \n\n')

fprintf(' --Line-- Power at bus & line flow --Line loss-- Transformer\n')

fprintf(' from to MW Mvar MVA MW Mvar tap\n')

for n = 1:nbus

busprt = 0;

n;

for L = 1:nbr

L;

if busprt == 0

fprintf(' \n'), fprintf('%6g', n), fprintf(' %9.3f', P(n)\*basemva)

fprintf('%9.3f', Q(n)\*basemva), fprintf('%9.3f\n', abs(S(n)\*basemva))

busprt = 1;

else, end

if nl(L)==n

k = nr(L);

nl(L);

if (nl(L)==p & k==q) & r~=0

In=((V(p)+r\*V(p)\*(cos(gamma)+j\*sin(gamma)))-a(L)\*V(q))\*y(L)/a(L)^2+Bc(L)/a(L)^2\*((V(p)+r\*V(p)\*(cos(gamma)+j\*sin(gamma))));

Ik=(V(q)-(V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma)))/a(L))\*y(L)+Bc(L)\*V(q);

Snk=(V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma)))\*conj(In)\*basemva;

Skn=V(q)\*conj(Ik)\*basemva;

SL=Snk+Skn;

SLT=SLT+SL;

elseif (nl(L)==q & k==p) & r~=0

In=(V(q)-a(L)\*(V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma))))\*y(L)/a(L)^2+Bc(L)/a(L)^2\*V(q);

Ik=((V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma)))-V(q)/a(L))\*y(L)+Bc(L)\*(V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma)));

Snk=V(n)\*conj(In)\*basemva;

Skn=(V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma)))\*conj(Ik)\*basemva;

SL=Snk+Skn;

SLT=SLT+SL;

else

In = (V(n) - a(L)\*V(k))\*y(L)/a(L)^2 + Bc(L)/a(L)^2\*V(n);

Ik = (V(k) - V(n)/a(L))\*y(L) + Bc(L)\*V(k);

Snk = V(n)\*conj(In)\*basemva;

Skn = V(k)\*conj(Ik)\*basemva;

SL = Snk + Skn;

SLT = SLT + SL;

end

elseif nr(L)==n

k = nl(L);

nr(L);

if (nr(L)==q & k==p) & r~=0

In=(V(q)-((V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma)))/a(L)))\*y(L)+Bc(L)\*V(q);

Ik=((V(p)+r\*V(p)\*(cos(gamma)+j\*sin(gamma)))-a(L)\*V(q))\*y(L)/a(L)^2+Bc(L)/a(L)^2\*(V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma)));

Snk=V(q)\*conj(In)\*basemva;

Skn=(V(p)+V(p)\*r\*(cos(gamma)+j\*sin(gamma)))\*conj(Ik)\*basemva;

SL=Snk+Skn;

SLT=SLT+SL;

elseif (nr(L)==p & k==q) & r~=0

In=((V(p)+V(p)\*r\*(cos(pi+gamma)+j\*sin(pi+gamma)))-V(q)/a(L))\*y(L)+Bc(L)\*(V(p)+V(p)\*r\*(cos(pi+gamma)+j\*sin(pi+gamma)));

Ik=(V(q)-a(L)\*(V(p)+r\*V(p)\*(cos(pi+gamma)+j\*sin(pi+gamma))))\*y(L)/a(L)^2+Bc(L)/a(L)^2\*V(q);

Snk=(V(p)+V(p)\*r\*(cos(pi+gamma)+j\*sin(pi+gamma)))\*conj(In)\*basemva;

Skn=V(q)\*conj(Ik)\*basemva;

SL=Snk+Skn;

SLT=SLT+SL;

else

In = (V(n) - V(k)/a(L))\*y(L) + Bc(L)\*V(n);

Ik = (V(k) - a(L)\*V(n))\*y(L)/a(L)^2 + Bc(L)/a(L)^2\*V(k);

Snk = V(n)\*conj(In)\*basemva;

Skn = V(k)\*conj(Ik)\*basemva;

SL = Snk + Skn;

SLT = SLT + SL;

end

else, end

if nl(L)==n | nr(L)==n

fprintf('%12g', k),

fprintf('%9.3f', real(Snk)), fprintf('%9.3f', imag(Snk))

fprintf('%9.3f', abs(Snk)),

fprintf('%9.3f', real(SL)),

if nl(L) ==n & a(L) ~= 1

fprintf('%9.3f', imag(SL)), fprintf('%9.3f\n', a(L))

else, fprintf('%9.3f\n', imag(SL))

end

else, end

end

end

SLT = SLT/2;

fprintf(' \n'), fprintf(' Total loss ')

fprintf('%9.3f', real(SLT)), fprintf('%9.3f\n', imag(SLT))

clear Ik In SL SLT Skn Snk

%clear A DC DX J11 J22 J33 J44 Qk delta lk ll lm

%clear A DC DX J11 J22 J33 Qk delta lk ll lm

% Prints the power flow solution on the screen

% This program prints the power flow solution in a tabulated form

% on the screen.

disp(tech)

fprintf(' Maximum Power Mismatch = %g \n', maxerror)

fprintf(' No. of Iterations = %g \n\n', iter)

head =[' Bus Voltage Angle ------Load------ ---Generation--- Injected'

' No. Mag. Degree MW Mvar MW Mvar Mvar '

' '];

disp(head)

for n=1:nbus

fprintf(' %5g', n), fprintf(' %7.3f', Vm(n)),

fprintf(' %8.3f', deltad(n)), fprintf(' %9.3f', Pd(n)),

fprintf(' %9.3f', Qd(n)), fprintf(' %9.3f', Pg(n)),

fprintf(' %9.3f ', Qg(n)), fprintf(' %8.3f\n', Qsh(n))

end

fprintf(' \n'), fprintf(' Total ')

fprintf(' %9.3f', Pdt), fprintf(' %9.3f', Qdt),

fprintf(' %9.3f', Pgt), fprintf(' %9.3f', Qgt), fprintf(' %9.3f\n\n', Qsht)

% data=[

% 0.0038 2 0 50 200

% 0.0175 1.75 0 20 80

% 0.0625 1 0 15 50

% 0.0083 3.25 0 10 35

% 0.025 3 0 10 30

% 0.025 3 0 12 40

% ];

data=[

0 2 0.0038 50 200

0 1.75 0.0175 20 80

0 1 0.0625 15 50

0 3.25 0.0083 10 35

0 3 0.025 10 30

0 3 0.025 12 40

];

Pg(Pg==0)=[];

N = length(data(:,1)); % no.of generators

C=zeros(N,1); total\_Gen=0;

for i=1:N

C(i)=(data(i,1)\*(Pg(i)^2)+data(i,2)\*Pg(i)+data(i,3));

end

fprintf('Generator no. Pg Fuel cost\n')

for i=1:length(Pg)

fprintf('%d %f %f\n',i,Pg(i),C(i))

end

Tot\_gen = sum(Pg);

fprintf('The total generation and fuel cost:\n',Tot\_gen)

TotCost=sum(C);

fprintf('%f\t%f\n',Tot\_gen,TotCost)