

[x]=xlsread('pqinput.xls','A2:A2'); %节点数

[y]=xlsread('pqinput.xls','B2:B2'); %支路数

e=xlsread('pqinput.xls','B4:B4'); %精度

[point]=xlsread('pqinput.xls','D3:H100'); %节点数据100指的是100个节点，可更改

[zhilu]=xlsread('pqinput.xls','J3:R100'); %线路数，100条支路

TYPE=zeros(x,1);

U=zeros(x,1);

a=zeros(x,1);

P=zeros(x,1);

Q=zeros(x,1);

I=zeros(y,1);

J=zeros(y,1);

Rij=zeros(y,1);

Xij=zeros(y,1);

Zij=Rij+j\*Xij;

Y=zeros(x);

G=zeros(x);

B=zeros(x);

B0=zeros(y,1);

RT=zeros(y,1);

XT=zeros(y,1);

ZT=RT+j\*XT;

KT=zeros(y,1);

%------------------------------矩阵赋初值：

TYPE=point(:,1); %节点类型

U=point(:,2); %节点电压

a=point(:,3); %节点相角

P=point(:,4); %节点有功功率

Q=point(:,5); %节点无功功率

I=zhilu(:,1); %起始节点编号

J=zhilu(:,2); %终止节点编号

Rij=zhilu(:,3); %线路电阻

Xij=zhilu(:,4); %线路电抗

Zij=Rij+j\*Xij; %线路阻抗

B0=zhilu(:,5); %线路对地导纳b

RT=zhilu(:,6); %变压器佃租

XT=zhilu(:,7); %变压器电抗

ZT=RT+j\*XT;

KT=zhilu(:,8); %标准变比

W=zhilu(:,9); %折算标志

%------------------------------求节点导纳矩阵Y

for m=1:y

if KT(m)==0

Y(I(m),J(m))=-1/Zij(m);

Y(J(m),I(m))=-1/Zij(m);

else

Y(I(m),J(m))=-1/(KT(m)\*ZT(m)); %默认初始变压器阻抗归算至低压侧，不同线路需要进行统一基准值归算，此处向高压侧折算

Y(J(m),I(m))=-1/(KT(m)\*ZT(m)); %同上

end

end

for m=1:x %节点数

for n=1:y %支路数

if KT(n)==0 %支路标志

if(I(n)==m|J(n)==m) %或

Y(m,m)=Y(m,m)-Y(I(n),J(n))+j\*B0(n)/2;

end

else

if W==0 %起始节点低压侧

if I(n)==m %起始节点

Y(m,m)=Y(m,m)-Y(I(n),J(n))+(KT(n)-1)/KT(n)\*(1/ZT(n)); %折算到高压侧

elseif J(n)==m %终止节点高压侧

Y(m,m)=Y(m,m)-Y(I(n),J(n))+(1-KT(n))/(KT(n)^2)\*(1/ZT(n));

end

else %起始节点高压侧

if I(n)==m %起始节点

Y(m,m)=Y(m,m)-Y(I(n),J(n))+(1-KT(n))/(KT(n)^2)\*(1/ZT(n));

elseif J(n)==m %终止节点

Y(m,m)=Y(m,m)-Y(I(n),J(n))+(KT(n)-1)/KT(n)\*(1/ZT(n));

else Y(m,m)=Y(m,m);

end

end

end

end

end

G=real(Y); %矩阵Y实部组成矩阵

%-----------------------求B'矩阵及其逆矩阵B1

B=imag(Y); %矩阵Y虚部组成矩阵

ph=find(TYPE(:,1)==3); %寻找平衡节点

BB=B;

BB(:,ph)=[]; %平衡节点所在列值变为0

BB(ph,:)=[]; %平衡节点所在行值变为0

B1=BB;

B1=inv(B1); %逆矩阵函数

%-----------------------%求B''及其逆矩阵B2

phpv=find(TYPE(:,1)>1); %寻找平衡节点及PV节点

BB=B;

BB(:,phpv)=[]; %平衡节点及PV节点所在行和列置0

BB(phpv,:)=[];

B2=BB;

B2=inv(B2); %逆矩阵函数

%-------------计算各节点有功功率不平衡量deltaPi

k=0;

kp=1;

kq=1;

while(((kp~=0)||(kq~=0))&&k<=20) %若误差不满足精度要求，最多迭代20次

kp=1;

kq=1;

notph=find(TYPE(:,1)<3); %找出节点不是平衡的节点

deltaPi=zeros(x-1,1);

pq=find(TYPE(:,1)==1); %PQ节点

pqnum=size(B2); %返回一个行向量第一个元素时矩阵的行数第二个元素是矩阵的列数。

pqnum=pqnum(1); %行数

deltaQi=zeros(pqnum,1);

for m=1:(x-1)

sum1=0;

for n=1:x

sum1=sum1+U(notph(m))\*U(n)\*(G(notph(m),n)\*cos(a(notph(m))-a(n))+B(notph(m),n)\*sin(a(notph(m))-a(n)));

end

deltaPi(m)=P(notph(m))-sum1;

end

max1=max(abs(deltaPi)); %节点有功功率误差绝对值最大值

if max1<=e %精度

kp=0;

if kq==0

break

else

for m=1:pqnum

sum2=0;

for n=1:x

sum2=sum2+U(pq(m))\*U(n)\*(G(pq(m),n)\*sin(a(pq(m))-a(n))-B(pq(m),n)\*cos(a(pq(m))-a(n)));

end

deltaQi(m)=Q(pq(m))-sum2;

end

max2=max(abs(deltaQi));

if max2<=e

kq=0;

if kp==0

break

else

k=k+1;

end

else

%将电压矩阵V的平衡节点和PV节点对应的电压值为空，形成新矩阵。

Uq=U;

Uq(phpv)=[];

Upq=Uq;

deltaU=-B2\*(deltaQi./Upq);

for m=1:pqnum

U(pq(m))=U(pq(m))+deltaU(m);

end

kp=1;

k=k+1;

end

end

else

%将电压矩阵U的平衡节点对应的电压值为空，形成新矩阵Up

Up=U;

Up(ph)=[];

Unotph=Up;

deltaa=((-B1\*(deltaPi./Unotph))./Unotph);

for m=1:(x-1)

a(notph(m))=a(notph(m))+deltaa(m);

end

kq=1;

for m=1:pqnum

sum2=0;

for n=1:x

sum2=sum2+U(pq(m))\*U(n)\*(G(pq(m),n)\*sin(a(pq(m))-a(n))-B(pq(m),n)\*cos(a(pq(m))-a(n)));

end

deltaQi(m)=Q(pq(m))-sum2;

end

Uq=U;

Uq(phpv)=[];

Upq=Uq;

deltaU=-B2\*(deltaQi./Upq);

for m=1:pqnum

U(pq(m))=U(pq(m))+deltaU(m);

end

kp=1;

k=k+1;

end

end

sum3=0+j\*0;

for m=1:x

sum3=sum3+conj(Y(ph,m))\*(U(m)\*cos(a(m))-i\*U(m)\*sin(a(m)))

end

%平衡节点复功率

Sph=(U(ph)\*cos(a(ph))+j\*U(ph)\*sin(a(ph)))\*sum3;

%-----------------求线路功率Sij和Sji

Sij=zeros(y,1);

Sji=zeros(y,1);

for m=1:y

if KT(m)==0 Sij(m)=(U(I(m))\*cos(a(I(m)))+i\*U(I(m))\*sin(a(I(m))))\*((U(I(m))\*cos(a(I(m)))-i\*U(I(m))\*sin(a(I(m))))\*(-i\*B0(m)/2)+(U(I(m))\*cos(a(I(m)))-i\*U(I(m))\*sin(a(I(m)))-U(J(m))\*cos(a(J(m)))+i\*U(J(m))\*sin(a(J(m))))\*conj(-Y(I(m),J(m))));

Sji(m)=(U(J(m))\*cos(a(J(m)))+i\*U(J(m))\*sin(a(J(m))))\*((U(J(m))\*cos(a(J(m)))-i\*U(J(m))\*sin(a(J(m))))\*(-i\*B0(m)/2)+(U(J(m))\*cos(a(J(m)))-i\*U(J(m))\*sin(a(J(m)))-U(I(m))\*cos(a(I(m)))+i\*U(I(m))\*sin(a(I(m))))\*conj(-Y(I(m),J(m))));

else

if W==1 Sij(m)=(U(I(m))\*cos(a(I(m)))+i\*U(I(m))\*sin(a(I(m))))\*((U(I(m))\*cos(a(I(m)))-i\*U(I(m))\*sin(a(I(m))))\*((1-KT(m))/(KT(m))^2\*conj(1/ZT(m)))+(U(I(m))\*cos(a(I(m)))-i\*U(I(m))\*sin(a(I(m)))-U(J(m))\*cos(a(J(m)))+i\*U(J(m))\*sin(a(J(m))))\*conj(-Y(I(m),J(m))));

Sji(m)=(U(J(m))\*cos(a(J(m)))+i\*U(J(m))\*sin(a(J(m))))\*((U(J(m))\*cos(a(J(m)))-i\*U(J(m))\*sin(a(J(m))))\*((KT(m)-1)/KT(m)\*conj(1/ZT(m)))+(U(J(m))\*cos(a(J(m)))-i\*U(J(m))\*sin(a(J(m)))-U(I(m))\*cos(a(I(m)))+i\*U(I(m))\*sin(a(I(m))))\*conj(-Y(I(m),J(m))));

else Sij(m)=(U(I(m))\*cos(a(I(m)))+i\*U(I(m))\*sin(a(I(m))))\*((U(I(m))\*cos(a(I(m)))-i\*U(I(m))\*sin(a(I(m))))\*((KT(m)-1)/KT(m)\*conj(1/ZT(m)))+(U(I(m))\*cos(a(I(m)))-i\*U(I(m))\*sin(a(I(m)))-U(J(m))\*cos(a(J(m)))+i\*U(J(m))\*sin(a(J(m))))\*conj(-Y(I(m),J(m))));

Sji(m)=(U(J(m))\*cos(a(J(m)))+i\*U(J(m))\*sin(a(J(m))))\*((U(J(m))\*cos(a(J(m)))-i\*U(J(m))\*sin(a(J(m))))\*((1-KT(m))/(KT(m))^2\*conj(1/ZT(m)))+(U(J(m))\*cos(a(J(m)))-i\*U(J(m))\*sin(a(J(m)))-U(I(m))\*cos(a(I(m)))+i\*U(I(m))\*sin(a(I(m))))\*conj(-Y(I(m),J(m))));

end

end

end

deltaSij=Sij+Sji;

S=zeros(x,1)+i\*zeros(x,1);

for b=1:x

for m=1:y

if I(m)==b

S(b)=S(b)+Sij(m);

else

if J(m)==b

S(b)=S(b)+Sji(m);

else

S(b)=S(b);

end

end

end

end

P=real(S);

Q=imag(S);

sumdeltaS=sum(S);

fid=fopen('pq分解法输出结果.txt','wt');

fprintf(fid,' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*潮流输出结果\*\*\*\*\*\*\*\*\*\*\*\*\*\n');

fprintf(fid,' 迭代次数k为: %d \n',k);

fprintf(fid,' ================================================\n');

fprintf(fid,' 平衡节点%d的复功率Sph为: %f+j\*(%f) \n',ph,real(Sph),imag(Sph));

fprintf(fid,' ================================================\n');

fprintf(fid,' 节点电压U为: \n');

for m=1:x

fprintf(fid,' 第%d个节点电压: %f\n',m,U(m));

end

fprintf(fid,' ================================================\n');

fprintf(fid,' 节点相角a为: \n');

for m=1:x

fprintf(fid,' 第%d个节点相角: %f\n',m,a(m));

end

fprintf(fid,' ================================================\n');

fprintf(fid,' 节点复功率S为: \n');

for m=1:x

fprintf(fid,' 第%d个节点复功率: %f+i\*(%f)\n',m,real(S(m)),imag(S(m)));

end

fprintf(fid,' ================================================\n');

fprintf(fid,' 节点有功功率P为: \n');

for m=1:x

fprintf(fid,' 第%d个节点有功功率: %f\n',m,P(m));

end

fprintf(fid,' ================================================\n');

fprintf(fid,' 节点无功功率Q为: \n');

for m=1:x

fprintf(fid,' 第%d个节点无功功率Q: %f\n',m,Q(m));

end

fprintf(fid,' ================================================\n');

fprintf(fid,' 线路功率Sij和Sji为: \n');

for m=1:y

fprintf(fid,' 节点%d到节点%d的功率为: %f+i\*(%f) \n',I(m),J(m),real(Sij(m)),imag(Sij(m)));

fprintf(fid,' 节点%d到节点%d的功率为: %f+i\*(%f) \n',J(m),I(m),real(Sji(m)),imag(Sji(m)));

end

fprintf(fid,' ================================================\n');

fprintf(fid,' 网络总损耗sumdeltaS为: %f+i\*(%f) \n',real(sumdeltaS),imag(sumdeltaS));

fprintf(fid,' ================================================\n');

fprintf(fid,' 线路功率损耗deltaSij为: \n');

for m=1:y

fprintf(fid,' %d--%d线路的功率损耗为: %f+i\*(%f) \n',I(m),J(m),real(deltaSij(m)),imag(deltaSij(m)));

end

fclose(fid); %关闭文件,程序结束