LISTING PROGRAM *PROVIBE*

```
clear all;
clc;

%Read Data From Excel
arhx=xlsread('SI.xls','INPUT DATA','A5');
arhy=xlsread('SI.xls','INPUT DATA','B5');
tkt=xlsread('SI.xls','INPUT DATA','C5:C25');
bcex=xlsread('SI.xls','INPUT DATA','D5:D25');
hcex=xlsread('SI.xls','INPUT DATA','E5:E25');
tcex=xlsread('SI.xls','INPUT DATA','F5:F25');
bcin=xlsread('SI.xls','INPUT DATA','G5:G25');
hcin=xlsread('SI.xls','INPUT DATA','H5:H25');
tcin=xlsread('SI.xls','INPUT DATA','I5:I25');
bb=xlsread('SI.xls','INPUT DATA','J5:J25');
hb=xlsread('SI.xls','INPUT DATA','K5:K25');
lb=xlsread('SI.xls','INPUT DATA','L5:L25');
tp=xlsread('SI.xls','INPUT DATA','M5');
q=xlsread('SI.xls','INPUT DATA','N5:N25');
fc=xlsread('SI.xls','INPUT DATA','O5');
dia=xlsread('SI.xls','INPUT DATA','P5');
tbi=xlsread('SI.xls','INPUT DATA','Q5');
Tp=xlsread('SI.xls','INPUT DATA','R5');
W=xlsread('SI.xls','INPUT DATA','S5');
GF=xlsread('SI.xls','INPUT DATA','T5');
EQ=xlsread('SI.xls','INPUT DATA','U5');
scl=xlsread('SI.xls','INPUT DATA','V5');
alpha=xlsread('SI.xls','INPUT DATA','W5');
r=xlsread('SI.xls','INPUT DATA','X5');
tmax=xlsread('SI.xls','INPUT DATA','Y5:Y25');
j=xlsread('SI.xls','INPUT DATA','Z5');
gravity=980; %(cm/dt^2)

dim=length(tkt);

%Earthquake Data
GA1=xlsread('EQ.xls','Bucharest','B1:B800');
GA2=xlsread('EQ.xls','El Centro','B1:B3931');
GA3=xlsread('EQ.xls','Koyna','B1:B1000');
GA4=xlsread('EQ.xls','Kobe','B1:B2199');
GA5=xlsread('EQ.xls','Corinth','B1:B3910');
GA6=xlsread('EQ.xls','Loma Prieta','B1:B4000');
GA7=xlsread('EQ.xls','Duzce Turki','B1:B4803');
GA8=xlsread('EQ.xls','Duzce Elnashay','B1:B3000');
GA9=xlsread('EQ.xls','Elnashay','B1:B3997');
GA10=xlsread('EQ.xls','Manjil','B1:B3985');
GA11=xlsread('EQ.xls','Santacruz','B1:B1850');
GA12=xlsread('EQ.xls','Gilroy','B1:B1149');
GA13=xlsread('EQ.xls','Rendah','B1:B3998');
GA14=xlsread('EQ.xls','Menengah','B1:B3999');
GA15=xlsread('EQ.xls','Tinggi','B1:B5101');
```

```
if EQ==1
    GA=GA1;
elseif EQ==2
    GA=GA2;
elseif EQ==3
    GA=GA3;
elseif EQ==4
    GA=GA4;
elseif EQ==5
    GA=GA5;
elseif EQ==6
    GA=GA6;
elseif EQ==7
    GA=GA7;
elseif EQ==8
    GA=GA8;
elseif EQ==9
    GA=GA9;
elseif EQ==10
    GA=GA10;
elseif EQ==11
    GA=GA11;
elseif EQ==12
    GA=GA12;
elseif EQ==13
    GA=GA13;
elseif EQ==14
    GA=GA14;
elseif EQ==15
    GA=GA15;
end

% gh and ah Parameter
for i=1:dim
    if scl==1
        gh=0.04;
        ah=1;
    elseif scl==2
        gh=0.07;
        ah=1;
    elseif scl==3
        gh=0.12;
        ah=0.95;
    elseif scl==4
        gh=0.20;
        ah=0.87;
    elseif scl==5
        gh=0.38;
        ah=0.75;
    elseif scl==6
        gh=1.10;
        ah=0.97;
    end
end

%Kekakuan
```

```
Icex=zeros(dim,1);     %inersia kolom exterior
Icin=zeros(dim,1);     %inersia kolom intrior
ijcex=zeros(dim,1);
ijcin=zeros(dim,1);
kj=zeros(dim,1);       %kekakuan shear building
a1=zeros(dim,1);
a2=zeros(dim,1);
y1=zeros(dim,1);
y2=zeros(dim,1);
ybar=zeros(dim,1);
Ib1=zeros(dim,1);
Ib2=zeros(dim,1);
Ib=zeros(dim,1);       %inersia balok T
eilcex=zeros(dim,1);
eilcin=zeros(dim,1);
eilb=zeros(dim,1);
kcex=zeros(dim,1);
kcin=zeros(dim,1);
kb=zeros(dim,1);
k1cex=zeros(dim,1);
k1cin=zeros(dim,1);
Cmcex=zeros(dim,1);
Cmcin=zeros(dim,1);
Kb=zeros(dim,1);
G=GF/9.8;
A=0.25*pi*dia^2;
E=4700*10.2*(fc.^0.5);  %(Kg/cm^2)
ncex=2;
ncin=max(arhx,arhy)-1;
nb=max(arhx,arhy);
for i=1:dim
    if j==1
        if i<2
            %kekakuan base isolation/isolasi dasar
            Kb(i)=G*A/tbi; %(Kg/cm)
        else
            Icex(i)=(1/12)*bcex(i)*(hcex(i)^3);
%(cm^4)
            Icin(i)=(1/12)*bcin(i)*(hcin(i)^3);
%(cm^4)
            ijcex(i)=(12*E*Icex(i))/(tcex(i)^3);
%(Kg/cm)
            ijcin(i)=(12*E*Icex(i))/(tcin(i)^3);
%(Kg/cm)
            kj(i)=(ncex*ijcex(i))+(ncin*ijcin(i));
%(Kg/cm)
            %mencari inersia balok T
            a1(i)=bb(i)*(hb(i)-tp);
%(cm^2)
            a2(i)=tp*(8*tp);
%(cm^2)
            y1(i)=((hb(i)-tp)/2);
%(cm)
            y2(i)=hb(i)-(tp/2);
%(cm)
```

```
            ybar(i)=((a1(i)*y1(i))+(a2(i)*y2(i)))/(a1(i)+a2(i));
%(cm)
            Ib1(i)=((1/12)*bb(i)*((hb(i)-tp)^3))+(a1(i)*(ybar(i)-
y1(i))^2);
            Ib2(i)=((1/12)*(8*tp)*(tp^3))+(a2(i)*(ybar(i)-
y2(i))^2);   %(cm^4)
            Ib(i)=Ib1(i)+Ib2(i);
%(cm^4)
            eilcex(i)=(E*Icex(i))/tcex(i);
%(cm^3)
            eilcin(i)=(E*Icin(i))/tcin(i);
%(cm^3)
            eilb(i)=(E*Ib(i))/lb(i);
%(cm^3)
            kcex(i)=eilcex(i)/eilcex(i);
            kcin(i)=eilcin(i)/eilcex(i);
            kb(i)=eilb(i)/eilcex(i);
            Cmcex(i)=(kb(i)+kb(i))/(kb(i)+kb(i)+4*kcex(i));

Cmcin(i)=(2*(kb(i)+kb(i)))/(2*(kb(i)+kb(i))+4*kcin(i));

Kb(i)=(ncex*Cmcex(i)*ijcex(i))+(ncin*Cmcin(i)*ijcin(i));
%(Kg/cm)
        end
    elseif j==2
        Icex(i)=(1/12)*bcex(i)*(hcex(i)^3);
%(cm^4)
        Icin(i)=(1/12)*bcin(i)*(hcin(i)^3);
%(cm^4)
        ijcex(i)=(12*E*Icex(i))/(tcex(i)^3);
%(Kg/cm)
        ijcin(i)=(12*E*Icex(i))/(tcin(i)^3);
%(Kg/cm)
        kj(i)=(ncex*ijcex(i))+(ncin*ijcin(i));
%(Kg/cm)
        %mencari inersia balok T
        a1(i)=bb(i)*(hb(i)-tp);
%(cm^2)
        a2(i)=tp*(8*tp);
%(cm^2)
        y1(i)=((hb(i)-tp)/2);
%(cm)
        y2(i)=hb(i)-(tp/2);
%(cm)
        ybar(i)=((a1(i)*y1(i))+(a2(i)*y2(i)))/(a1(i)+a2(i));
%(cm)
        Ib1(i)=((1/12)*bb(i)*((hb(i)-tp)^3))+(a1(i)*(ybar(i)-
y1(i))^2);
        Ib2(i)=((1/12)*(8*tp)*(tp^3))+(a2(i)*(ybar(i)-y2(i))^2);
%(cm^4)
        Ib(i)=Ib1(i)+Ib2(i);
%(cm^4)
        eilcex(i)=(E*Icex(i))/tcex(i);
%(cm^3)
        eilcin(i)=(E*Icin(i))/tcin(i);
%(cm^3)
```

```
        eilb(i)=(E*Ib(i))/lb(i);
%(cm^3)
        kcex(i)=eilcex(i)/eilcex(i);
        kcin(i)=eilcin(i)/eilcex(i);
        kb(i)=eilb(i)/eilcex(i);
        if i<2
            Cmcex(i)=(kb(i)+0.5*kcex(i))/(kb(i)+2*kcex(i));

Cmcin(i)=(kb(i)+kb(i)+0.5*kcin(i))/(kb(i)+kb(i)+2*kcin(i));
        else
            Cmcex(i)=(kb(i)+kb(i))/(kb(i)+kb(i)+4*kcex(i));

Cmcin(i)=(2*(kb(i)+kb(i)))/(2*(kb(i)+kb(i))+4*kcin(i));
        end
        Kb(i)=(ncex*Cmcex(i)*ijcex(i))+(ncin*Cmcin(i)*ijcin(i));
%(Kg/cm)
    end
end

%Massa
m=(q.*lb.*nb.*10)/gravity; %(Kg.dt^2/cm)

%Write Data To Excel
xlswrite('SI.xls',Icex,'KEKAKUAN & MASSA','B4');
xlswrite('SI.xls',Icin,'KEKAKUAN & MASSA','C4');
xlswrite('SI.xls',Ib,'KEKAKUAN & MASSA','D4');
xlswrite('SI.xls',E,'KEKAKUAN & MASSA','E4');
xlswrite('SI.xls',kj,'KEKAKUAN & MASSA','F4');
xlswrite('SI.xls',eilcex,'KEKAKUAN & MASSA','G4');
xlswrite('SI.xls',eilcin,'KEKAKUAN & MASSA','H4');
xlswrite('SI.xls',eilb,'KEKAKUAN & MASSA','I4');
xlswrite('SI.xls',kcex,'KEKAKUAN & MASSA','J4');
xlswrite('SI.xls',kcin,'KEKAKUAN & MASSA','K4');
xlswrite('SI.xls',kb,'KEKAKUAN & MASSA','L4');
xlswrite('SI.xls',Cmcex,'KEKAKUAN & MASSA','M4');
xlswrite('SI.xls',Cmcin,'KEKAKUAN & MASSA','N4');
xlswrite('SI.xls',Kb,'KEKAKUAN & MASSA','O4');
xlswrite('SI.xls',m,'KEKAKUAN & MASSA','P4');

%Differential Equation Of M.D.O.F Structure

%Assemble Mass Matrix
M=diag(m);

%Assemble Stifness Matrix
kc=zeros(1,dim-1);
for i=1:(dim-1)
    kc(i)=(Kb(i)+Kb(i+1));
end
kd=[kc Kb(end)];
k0=diag(kd);
ke=zeros(1,dim-1);
for i=2:dim
    ke(i)=-Kb(i);
end
```

```
ke=ke(2:end);
k1=diag(ke,1);
k2=diag(ke,-1);
K=k0+k1+k2;

%EigenProblen
um=min(m);
uk=min(Kb);
Mu=M/um;
Ku=K/uk;
[eigvec,eigval]=eig(Ku,Mu);
lambda=diag(eigval);
N1=ones(1,dim);
N2=eigvec(1,:);
N3=N1./N2;

%Mode Shape
mode=eigvec*diag(N3);

%Mode Participation Factor
Pn=sum(M*mode);
Pn=Pn';
Mn=sum(M*mode.^2);
Mn=Mn';
rn=Pn./Mn;
Sigmarn=sum(rn);

%Angular Frequency
omega=lambda.^0.5*(uk/um)^0.5;

%Natural Periode
T=2.*pi./omega;

%Natural Frequency
f=1./T;

%Write Data To Excel
xlswrite('SI.xls',K,'MATRIX KEKAKUAN','B3');
xlswrite('SI.xls',M,'MATRIX MASSA','B3');
xlswrite('SI.xls',lambda,'MODE SHAPE','A3');
xlswrite('SI.xls',omega,'MODE SHAPE','B3');
xlswrite('SI.xls',T,'MODE SHAPE','C3');
xlswrite('SI.xls',f,'MODE SHAPE','D3');
xlswrite('SI.xls',mode,'MODE SHAPE','F3');
xlswrite('SI.xls',Sigmarn,'PF','F3');

%Numerical Solution by Beta Newmark Direct Integration Method
NEQ=length(GA);

for iter=1:(NEQ-1)
    if j==1
        if iter<2
            kf=Kb(:,1);
            kf=kf';
            for i=1:(dim-1)
```

```
            kc(i)=(kf(i)+kf(i+1));
        end
        kd=[kc kf(end)];
        k0=diag(kd);
        for i=2:dim
            ke(i)=-kf(i);
        end
        ke=ke(2:end);
        k1=diag(ke,1);
        k2=diag(ke,-1);
        K=k0+k1+k2;
    else
        kf=Kb(:,1);
        kz=kp(:,iter-1);
        kf=kf';
        kz=kz';
        for i=1:(dim-1)
            if i==1
                kc(i)=(kz(i)+kf(i+1));
            else
                kc(i)=(kf(i)+kf(i+1));
            end
        end
        kd=[kc kf(end)];
        k0=diag(kd);
        for i=2:dim
            ke(i)=-kf(i);
        end
        ke=ke(2:end);
        k1=diag(ke,1);
        k2=diag(ke,-1);
        K=k0+k1+k2;
    end
elseif j==2
    kf=Kb(:,1);
    kf=kf';
    for i=1:(dim-1)
        kc(i)=(kf(i)+kf(i+1));
    end
    kd=[kc kf(end)];
    k0=diag(kd);
    for i=2:dim
        ke(i)=-kf(i);
    end
    ke=ke(2:end);
    k1=diag(ke,1);
    k2=diag(ke,-1);
    K=k0+k1+k2;
end

%Arrange Damping Matrix
dr=0.05;
Bdamp=2*(dr*omega(3)-dr*omega(1))/(omega(3)^2-omega(1)^2);
Adamp=2*dr*omega(3)-Bdamp*omega(3)^2;
C=Adamp*M+Bdamp*K;
```

```
    %Effective Stiffness
    dt=0.01;
    gamma=0.5;
    beta=0.25;
    Ks=K+gamma/(beta*dt)*C+1/(beta*dt^2)*M;

    %"a" Constant Matrix
    ca=1/(beta*dt)*M+gamma/beta*C;

    %"b" Constant Matrix
    cb=1/(2*beta)*M+(gamma/(2*beta)-1)*dt*C;

    if iter<2
        y0=zeros(dim,1);
        v0=zeros(dim,1);
        a0=zeros(dim,1);
        dp(:,iter)=(GA(iter+1)-GA(iter))*diag(M)+ca*v0+cb*a0;
        dy(:,iter)=(Ks)\dp(:,iter);
        dv(:,iter)=gamma/(beta*dt)*dy(:,iter)-gamma/beta*v0+(1-
gamma/(2*beta))*dt*a0;
        da(:,iter)=1/(beta*dt^2)*dy(:,iter)-1/(beta*dt)*v0-
1/(2*beta)*a0;
        y(:,iter)=y0+dy(:,iter);
        v(:,iter)=v0+dv(:,iter);
        a(:,iter)=a0+da(:,iter);
        ynet(:,iter)=y(:,iter)-y(1,iter);
        if j==1
            F(:,iter)=K*ynet(:,iter);
        elseif j==2
            F(:,iter)=K*y(:,iter);
        end
    else
        dp(:,iter)=(GA(iter+1)-GA(iter))*diag(M)+ca*v(:,iter-
1)+cb*a(:,iter-1);
        dy(:,iter)=(Ks)\dp(:,iter);
        dv(:,iter)=gamma/(beta*dt)*dy(:,iter)-gamma/beta*v(:,iter-
1)+(1-gamma/(2*beta))*dt*a(:,iter-1);
        da(:,iter)=1/(beta*dt^2)*dy(:,iter)-1/(beta*dt)*v(:,iter-
1)-1/(2*beta)*a(:,iter-1);
        y(:,iter)=y(:,iter-1)+dy(:,iter);
        v(:,iter)=v(:,iter-1)+dv(:,iter);
        a(:,iter)=a(:,iter-1)+da(:,iter);
        ynet(:,iter)=y(:,iter)-y(1,iter);
        if j==1
            F(:,iter)=K*ynet(:,iter);
        elseif j==2
            F(:,iter)=K*y(:,iter);
        end
    end

    %Shear strain every level
    if iter<2
        rg(:,iter)=((y(:,iter)-y0)./tcex);
    else
        rg(:,iter)=((y(:,iter)-y(:,iter-1))./tcex)+rg(:,iter-1);
    end
```

```
    %Shear Modulus Every Step
    Gp(:,iter)=1./(1+ah.*(abs(rg(:,iter)).^ah)./gh).*G(:,1);

    if i<2
        GGo=1;
    else
        GGo(:,iter)=Gp(:,iter)./G;
    end

    %Stiffness Every Step
    kp(:,iter)=1./(1+ah.*(abs(rg(:,iter)).^ah)./gh).*Kb(:,1);
end

%tmax=(0.1*W)/A

%Maximum Shear Strain
rgmax=(tmax./G(:,1));

rg_a=zeros(dim,NEQ-1);
rg_pos=zeros(dim,NEQ-1);
rg_neg=zeros(dim,NEQ-1);
rg_cek=zeros(dim,NEQ-1);

for i=1:(NEQ-1)
    for tingkat=1:dim
        if i>1
            if v(tingkat,i)*v(tingkat,i-1)<0 && v(tingkat,i)>0
                rg_pos(tingkat,i)=rg(tingkat,i-1);
            elseif v(tingkat,i)*v(tingkat,i-1)<0 && v(tingkat,i)<0
                rg_pos(tingkat,i)=rg_pos(tingkat,i-1);
            elseif v(tingkat,i)*v(tingkat,i-1)>0
                rg_pos(tingkat,i)=rg_pos(tingkat,i-1);
            end
        end
    end
end

for i=1:(NEQ-1)
    for tingkat=1:dim
        if i>1
            if v(tingkat,i)*v(tingkat,i-1)<0 && v(tingkat,i)<0
                rg_neg(tingkat,i)=rg(tingkat,i-1);
            elseif v(tingkat,i)*v(tingkat,i-1)<0 && v(tingkat,i)>0
                rg_neg(tingkat,i)=rg_neg(tingkat,i-1);
            elseif v(tingkat,i)*v(tingkat,i-1)>0
                rg_neg(tingkat,i)=rg_neg(tingkat,i-1);
            end
        end
    end
end

for i=1:(NEQ-1)
    for tingkat=1:dim
        if i>1
```

```
            if v(tingkat,i)*v(tingkat,i-1)<0
                rg_cek(tingkat,i)=rg(tingkat,i-1);
            else
                rg_cek(tingkat,i)=rg_cek(tingkat,i-1);
            end
        end
    end
end

for i=1:(NEQ-1)
    for tingkat=1:dim
        if i>2
            vkec(tingkat,i)=v(tingkat,i)*v(tingkat,i-1);
        end
        if i<2
            cpem(tingkat,i)=rg(tingkat,i)./(rgmax(tingkat));
        elseif i>1 && i<80
            if v(tingkat,i)*v(tingkat,i-1)<0
                rg_a(tingkat,i)=rg(tingkat,i-1);
            else
                rg_a(tingkat,i)=rg_a(tingkat,i-1);
            end
            cpem(tingkat,i)=(rg(tingkat,i)-
rg_a(tingkat,1))./(2*rgmax(tingkat));
        elseif i>=80
            if v(tingkat,i)*v(tingkat,i-1)<0
                if abs(rg(tingkat,i-1)-rg_cek(tingkat,i-1))*100 >
5
                    rg_a(tingkat,i)=rg(tingkat,i-1);
                else
                    if v(tingkat,i)>0
                        rg_a(tingkat,i)=rg_pos(tingkat,i-1);
                    else
                        rg_a(tingkat,i)=rg_neg(tingkat,i-1);
                    end
                end
            else
                rg_a(tingkat,i)=rg_a(tingkat,i-1);
            end
            cpem(tingkat,i)=(rg(tingkat,i)-
rg_a(tingkat,i)).(2*rgmax(tingkat));
        end
    end
end

%Transpose Matrix
y=y';
v=v';
a=a';
ynet=ynet';
F=F';
Gp=Gp';
rgA=rg';
GGo=GGo';
cpem=cpem';
```

```
%Write Data To Excel
xlswrite('SI.xls',y,'DISP','B3');
xlswrite('SI.xls',v,'VELO','B3');
xlswrite('SI.xls',a,'ACC','B3');
xlswrite('SI.xls',ynet,'DISPNET','B3');
xlswrite('SI.xls',F,'GAYA HORIZONTAL','B3');
xlswrite('SI.xls',Gp,'MOD','B3');
xlswrite('SI.xls',rgA,'STRAIN','B3');
xlswrite('SI.xls',GGo,'GGO','B3');
xlswrite('SI.xls',GA,'REKAP ALL','B3');

%Newton Raphson Method
%menghitung x-pemisalan setiap tingkat
clc;
hasil=[];
iter=0;
nData=length(GA)-1;
for Z=1:(nData*length(tkt))
    iter=iter+1;
    const=cpem(Z);
    if r==1
        x=[alpha -const];
    elseif r==2
        x=[alpha 1 -const];
    elseif r==3
        x=[alpha 0 1 -const];
    elseif r==4
        x=[alpha 0 0 1 -const];
    elseif r==5
        x=[alpha 0 0 0 1 -const];
    elseif r==6
        x=[alpha 0 0 0 0 1 -const];
    elseif r==7
        x=[alpha 0 0 0 0 0 1 -const];
    elseif r==8
        x=[alpha 0 0 0 0 0 0 1 -const];
    elseif r==9
        x=[alpha 0 0 0 0 0 0 0 1 -const];
    elseif r==10
        x=[alpha 0 0 0 0 0 0 0 0 1 -const];
    end
    rx=roots(x);
    xnew=min(rx);
    hasil(iter,1)=xnew;
end
xpem=hasil;
Xpem=reshape(xpem,nData,length(tkt));

tg_a=zeros(nData,length(tkt));
tg_pos=zeros(nData,length(tkt));
tg_neg=zeros(nData,length(tkt));

%menghitung tegangan geser setiap tingkat
for i=1:nData
    for tingkat=1:length(tkt)
        if i<2
```

```
            tg(i,tingkat)=tmax(tingkat).*Xpem(i,tingkat);
        elseif i>1 && i<80
            if v(i,tingkat)*v(i-1,tingkat)<0
                if v(i,tingkat)>0
                    tg_a(i,tingkat)=tg(i-1,tingkat);
                    tg_pos(i,tingkat)=tg(i-1,tingkat);
                    tg_neg(i,tingkat)=tg_neg(i-1,tingkat);
                else
                    tg_a(i,tingkat)=tg(i-1,tingkat);
                    tg_pos(i,tingkat)=tg_pos(i-1,tingkat);
                    tg_neg(i,tingkat)=tg(i-1,tingkat);
                end
            else
                tg_a(i,tingkat)=tg_a(i-1,tingkat);
                tg_pos(i,tingkat)=tg_pos(i-1,tingkat);
                tg_neg(i,tingkat)=tg_neg(i-1,tingkat);
            end

tg(i,tingkat)=2*tmax(tingkat).*Xpem(i,tingkat)+tg_a(i,tingkat);
        elseif i>=80
            if v(i,tingkat)*v(i-1,tingkat)<0
                if abs(rg(tingkat,i-1)-rg_cek(tingkat,i-1))*100 >
5
                    if v(i,tingkat)>0
                        tg_a(i,tingkat)=tg(i-1,tingkat);
                        tg_pos(i,tingkat)=tg(i-1,tingkat);
                        tg_neg(i,tingkat)=tg_neg(i-1,tingkat);
                    else
                        tg_a(i,tingkat)=tg(i-1,tingkat);
                        tg_pos(i,tingkat)=tg_pos(i-1,tingkat);
                        tg_neg(i,tingkat)=tg(i-1,tingkat);
                    end
                else
                    if v(i,tingkat)>0
                        tg_a(i,tingkat)=tg_pos(i-1,tingkat);
                        tg_pos(i,tingkat)=tg(i-1,tingkat);
                        tg_neg(i,tingkat)=tg_neg(i-1,tingkat);
                    else
                        tg_a(i,tingkat)=tg_neg(i-1,tingkat);
                        tg_pos(i,tingkat)=tg_pos(i-1,tingkat);
                        tg_neg(i,tingkat)=tg(i-1,tingkat);
                    end
                end
            else
                tg_a(i,tingkat)=tg_a(i-1,tingkat);
                tg_pos(i,tingkat)=tg_pos(i-1,tingkat);
                tg_neg(i,tingkat)=tg_neg(i-1,tingkat);
            end

tg(i,tingkat)=2*tmax(tingkat).*Xpem(i,tingkat)+tg_a(i,tingkat);
        end
    end
end

%Write Data To Excel
xlswrite('SI.xls',tg,'STRESS','B3');
```