```
startCond = str2num(getenv('SGE_TASK_ID'))
endCond = startCond + 15

if(ispc==1)
    foldersource='C:\Users\ctc\Documents\GitHub\Chapter3\';
elseif(ismac==1)
    foldersource='/Users/Shaun/Documents/GitHub/Chapter3/';
elseif(isunix==1)
    foldersource='/mnt/HA/groups/nieburGrp/Shaun/Chapter3/';
end

load(strcat(foldersource,'GrPrSpline.mat'))
load(strcat(foldersource,'ReNuSpline.mat'))
load(strcat(foldersource,'NuReSpline.mat'))
%% Load conductor info
conductorData=importfileAB(strcat(foldersource,'conductorData.csv'));
[conductorCount,~]=size(conductorData);
conductorData.ResistanceACLowdegc=conductorData.ResistanceDCLowdegc;
conductorData.ResistanceACLowdegcMeter=...
    conductorData.ResistanceACLowdegc./...
    conductorData.MetersperResistanceInterval;
conductorData.ResistanceACHighdegcMeter=...
    conductorData.ResistanceACHighdegc./...
    conductorData.MetersperResistanceInterval;
conductorData.convergenceOrder=realmax.*ones(conductorCount,1);
conductorData.simulated=zeros(conductorCount,1);
conductorData.Cmax=zeros(conductorCount,1);
conductorData.Cmin=zeros(conductorCount,1);
conductorData.convergeCurrent = zeros(conductorCount,1);
%% Setup weather data
epsilons=0.9;
H=0;
phi=90*pi/180;
maxpsol=1050;
alphas=0.9;
spacer=10;
searchIncrement=0.001;
weatherPermutationCount=(spacer+1)^4;

psols=zeros(weatherPermutationCount,1);
winds=zeros(weatherPermutationCount,1);
ambtemps=zeros(weatherPermutationCount,1);
currents=zeros(weatherPermutationCount,1);

counter=0;
for imagnitude=0.005:(0.05)/(spacer*2):0.055
    for psol=0:maxpsol/spacer:maxpsol
        for ambtemp=-33:98/spacer:65
            for Vw=0:10/spacer:10
                counter=counter+1;
                psols(counter)=psol*alphas;
                winds(counter)=Vw;
                ambtemps(counter)=ambtemp;
                currents(counter)=imagnitude;
            end
        end
```

```
    end
end

%% Run conductor simulation
if endCond>conductorCount
    endCond=conductorCount;
end
parfor c=startCond:endCond
    disp(c)
    if(conductorData(c,:).polymodels==""||conductorData(c,:).simulated==1)
        continue;
    end
    cdata=conductorData(c,:);
    delta=zeros(weatherPermutationCount,1);
    delta1=zeros(weatherPermutationCount,1);
    cs=zeros(weatherPermutationCount,1);

    maxcurrent=ceil(cdata.AllowableAmpacity);
    diam=cdata.DiamCompleteCable*0.0254;
    beta=(cdata.ResistanceACHighdegcMeter-...
        cdata.ResistanceACLowdegcMeter)/(cdata.HighTemp-cdata.LowTemp);
    alpha=cdata.ResistanceACHighdegcMeter-beta*cdata.HighTemp;
    polymodel=str2func(conductorData(c,:).polymodels);
    for counter=1:weatherPermutationCount
        if(currents(counter)<=conductorData(c,:).convergeCurrent+0.0001)
            continue;
        end
        GuessTc=GetGuessTemp(currents(counter)*maxcurrent,...
            ambtemps(counter),diam,phi,winds(counter),alpha,beta,...
            epsilons,psols(counter),polymodel);
        [rootT,~,~,~,~,~,~] = GetTempNewton(currents(counter)*...
            maxcurrent,ambtemps(counter),H,diam,phi,winds(counter),...
            alpha,beta,epsilons,psols(counter),f,ff,ffinv,polymodel);
        [convergenceOrder] = GetTempNewtonGetCC(currents(counter)*...
            maxcurrent,ambtemps(counter),H,diam,phi,winds(counter),...
            alpha,beta,epsilons,psols(counter),f,ff,ffinv,polymodel,rootT);

        if(convergenceOrder<conductorData(c,:).convergenceOrder)
            conductorData(c,:).convergenceOrder=convergenceOrder;
        end
        topend=max(rootT,GuessTc);
        bottomend=min(rootT,GuessTc);
        temps=(bottomend-10:searchIncrement:topend+10)';

        [searchCount,~]=size(temps);
        tempSearch=zeros(searchCount,3);
        tempSearch(:,1)=temps;
        [Tc,I2R,I2Rprime,Prad,PradPrime,PradPrimePrime,Pcon,PconPrime,...
            PconPrimePrime,~,~,~] =GetTempNewtonFirstIteration2(...
            currents(counter)*maxcurrent,ambtemps(counter),H,diam,phi,...
            winds(counter),alpha,beta,epsilons,psols(counter),...
            tempSearch(:,1),f,ff,ffinv);

        h=I2R+psols(counter)*diam*alphas-Pcon-Prad;
        hprime=I2Rprime-PconPrime-PradPrime;
        hprimeprime=-1*PconPrimePrime-PradPrimePrime;
```

```
        tempSearch(:,2)=Tc;
        tempSearch(:,3)=abs((h.*hprimeprime)./(hprime.^2));
        rerun=1;
        reruncounter=0;

        while(rerun)
            rerun=0;
            reruncounter=reruncounter+1;
            if(reruncounter>5000)
                msg='error condition!';
                error(msg)
            end
            searchRes=tempSearch(tempSearch(:,1)>=bottomend-...
                delta(counter,1) & tempSearch(:,1)<=...
                topend+delta1(counter,1),:);
            [row,~]=size(searchRes);
            if(row>1)
                if(max(searchRes(:,2))>topend+delta1(counter,1))
                    delta1(counter,1)=0.05+max(searchRes(:,2))-topend;
                    rerun=1;
                end
                if(min(searchRes(:,2))<bottomend-delta(counter,1))
                    delta(counter,1)=0.05+bottomend-min(searchRes(:,2));
                    rerun=1;
                end
            end
        end
        if(row>1)
            cs(counter)=max(searchRes(:,3));
            if(cs(counter)>1 && ...
                    currents(counter)>conductorData(c,:).convergeCurrent)
                conductorData(c,:).convergeCurrent=currents(counter);
                disp(currents(counter))
            end
            if(cs(counter)>1 && (rootT-ambtemps(counter))>...
                    conductorData(c,:).lowestRise)
                conductorData(c,:).lowestRise=(rootT-ambtemps(counter));
                disp((rootT-ambtemps(counter)))
            end
        end
    end
    conductorData(c,:).Cmax=max(cs);
    conductorData(c,:).Cmin=min(cs(cs~=0));
    conductorData(c,:).simulated=1;
end
conductorData = conductorData(startCond:endCond,:);
conductorData.Index = (startCond:endCond)';

save(strcat(foldersource,num2str(startCond),'matlab'))
```