```
%------------------------------------------

%ChE613A (The Structure and Rheology of Complex fluids)
%Assignment 05 (due date: 27/09/2021)
%Student name: Keerthi Vasan M (Roll No.: 21102023}

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


%------------------------------------------
%Extracted data

f_prime=[0.0031 0.0046 0.0068 0.0099 0.0145 0.0216 0.0314 0.0462 0.0681 0.1003 0.1478 0.2122 0.3127 0.4607 0.6744 1.0000  1.4544 2.1568];
G_prime=[1.0834 1.4177 1.7969 2.2774 2.8134 3.4534 4.0792 4.8804 5.6189 6.3868 7.1673 7.9407 8.6857 9.5005 14.8752 11.5872 13.1708 15.9610];
f_doublePgiven=[0.0021 0.0031 0.0046 0.0067 0.0099 0.0146 0.0213 0.0316 0.0459 0.0677 0.0990 0.1469 0.2150 0.3167 0.46067 0.6787 0.9936 1.4544];
G_doublePgiven=[1.1330 1.3469 1.6012 1.8435 2.0955 2.3515 2.5887 2.7955 2.9236 3.0382 3.0774 3.0382 2.9995 2.9424 2.8497 2.7423 2.6389 2.6389];
%(f_prime,G_prime) and (f_doublePgiven,G_doublePgiven) were (f,G') and (f,G'') data respectively
%------------------------------------------


%------------------------------------------
%Calculation of relaxation times of the 'N' choosen Maxwell modes

N=6;
%N is number of modes in the model
%Note: If N is changed then do the corresponding change in the 'fun' statement that is used for the 'lsqcurvefit' curve fitting
Omega_prime=f_prime*2*3.14;
%Omega_prime is the angular frequency computed using the 'f' data of given (f,G') data set
n=length(Omega_prime);
%n is the number of data points present in the given (f vs G',G'') plot
omega_doublePgiven=f_doublePgiven*2*3.14;
%Omega_doublePgiven is the angular frequency computed using the 'f' data of given (f,G'') data set
Tau_max=power(Omega_prime(1),-1);
Tau_min=power(Omega_prime(n),-1);
%[Tau_min,Tau_max] is the the range within which all relaxation times will lie
%Tau_min=1/Omega_prime_max and Tau_max=1/Omega_prime_min
%Omega_prime_max = Omega_prime(n) and Omega_prime_min = Omega_prime(1)
for i=1:N
    Tau(i)=Tau_min*power((Tau_max/Tau_min),((i-1)/(N-1)));
end
%Tau is the relaxtion time of a Maxwell mode, here Tau is a array containing relaxtion times of all 'N' modes choose
%------------------------------------------


%------------------------------------------
%Calculation of relaxation modulus (g) of the choosen 'N' modes using the extracted storage modulus (G')
for i=1:N
    x0(i)=1;%x0 is array containing the initial points for the variables that 'fun' accepts
    for j=1:n
        abscissa(i,j)=power((Omega_prime(j)*Tau(i)),2)/(1+power((Omega_prime(j)*Tau(i)),2));
    end
end
```

```matlab
%abscissa is a array of (N*n) dimension and it is the input data (xdata) for the 'lsqcurvefit'
options = optimoptions('lsqcurvefit','StepTolerance',exp(pi/2));
%Above statement fixes the termination tolerence on 'abscissa' as 'exp(pi/2)
lb = [];
ub = [];
%'lb and 'ub' are vectors of lower and upper bounds respectively
fun=@(x,abscissa)(x(1)*abscissa(1,:))+(x(2)*abscissa(2,:))+(x(3)*abscissa(3,:))+(x(4)*abscissa(4,:))+(x(5)*abscissa(5,:))+(x(6)*abscissa(6,:));
%Note: If N is changed then do the corresponding change in the 'fun' statement that is used for the 'lsqcurvefit' curve fitting
g=lsqcurvefit(fun,x0,abscissa,G_prime,lb,ub,options);
%g is a array containing relaxation modulus of all the 'N' modes choosen.

%Now, we are calculating the relaxation modulus (G(t)) at very time 't'
G=0;
t=1:1000;
for i=1:N
    G=G+(g(i)*exp(-t/Tau(i)) );
end
%------------------------------------------



%------------------------------------------
%Verifying the results by using relaxation modulus (g) obtained to calculate loss modulus (G'')
sum=0;
for i=1:n
    for j=1:N
        sum=sum+((g(j)*Omega_prime(i)*Tau(j))/(1+power((Omega_prime(i)*Tau(j)),2)));
    end
    G_doubleprime(i)=sum;%G_doubleprime is the calculated loss modulus (G'')
    sum=0;
    error(i)=abs(G_doublePgiven(i)-G_doubleprime(i))/G_doublePgiven(i);
end
Avg_error=0;
for i=1:n
    Avg_error=Avg_error+error(i);
end
Avg_error=Avg_error/n;
%------------------------------------------



%------------------------------------------
%Display of the results obtained so far
fprintf('-------------------------------------------------------------')
fprintf('\nMode no.\tRelaxation modulus\tRelaxation time (s)')
for i=1:N
    fprintf('\n   %i\t\t     %.4f\t\t     %.4f',i,g(i),Tau(i))
end
fprintf('\n-------------------------------------------------------------')

G_dis="G=";
Eta_dis="Eta=";
for i=1:N
    if i==N
        G_dis=G_dis+"{"+num2str(g(i))+"*exp(-t/"+num2str(Tau(i))+")}";
        Eta_dis=Eta_dis+"{"+num2str(g(i)*Tau(i))+"*exp(-t/"+num2str(Tau(i))+")}";
```

```
    else
        G_dis=G_dis+"{"+num2str(g(i))+"*exp(-t/"+num2str(Tau(i))+")} + ";
        Eta_dis=Eta_dis+"{"+num2str(g(i)*Tau(i))+"*exp(-t/"+num2str(Tau(i))+")} + ";
    end
end
fprintf('\nFor N = %i, Average error in the G"(ω) calculated is %.4f %%',N,Avg_error*100)
fprintf('\nRelaxation modulus is %s',G_dis)
fprintf('\nShear  viscosity is %s',Eta_dis)

figure(1);
loglog(f_doublePgiven,G_doublePgiven,'o',f_prime,G_doubleprime,'.')
title('Comparison between G" Vs f plot for calculated and extracted data ')
xlabel('f (Hz)')
legend('G" given','G" calculated')


figure(2);
plot(t,G);
title('Plot of relaxation modulus G(t) with respect to time')
xlabel('Time (s)')
ylabel('G(t)')
%-------------------------------------------
```