## Vibrational Model for Investigation of Hydration Effects on Flexibility

Hydration adds in abrasive design as it allows for the control of process parameters (by allowing the polishing designer to utilize the effects of viscoelasticity).

In addition to the multi-layered nature of the multicon abrasive (gelatin-SiC-diamond), an incredible variety of effects can be acheived with minimal changes to the process design. The below model is an attempt at characterizing the effect of hydrating the abrasive to different levels, firstly to prove that the addition of hydration aids in reducing contact stress and thus enables ductile regime polishing conditions to occur at higher than usual velocities, and secondly: to create a series of relations and inputs for further and more in-depth contact mechanics analysis to occur.

Figure 1 below shows the model used at further analysis, where m represents the mass of the abrasive, F represents the force applied due to impinging velocity, c represents the damping due to hydration (a desired output of this research), k represents the stiffness of the abrasive system, and x represents the deformation of the abrasive. The fixed ground is assumed as the workpiece of the material (which in this case would be a flat SLS produced Ti-6Al-4V component).

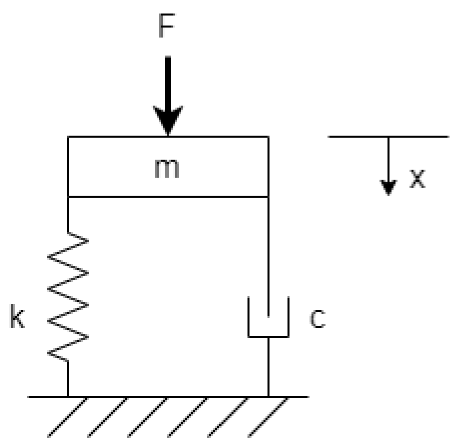



Figure 1: Classic Externally Forced Damped Vibrational Model

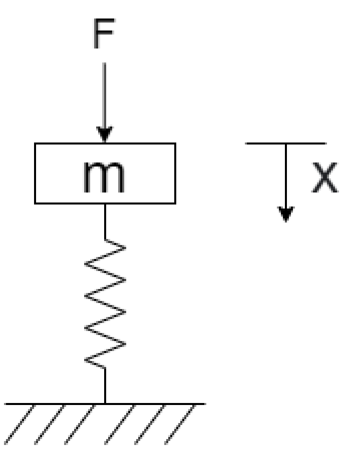



Figure 2: Externally Forced Undamped Vibrational Model

Figure 2 above shows the first vibrational model used in this research (where the abrasive is assumed to not be hydrated at all). This allows for a slightly simpler solution which can then be modified to include hydration factors.


Figure 3: Key

Another method of spring-damper modelling is that of the ever present viscoelastic models. Many are available but the most fitting model applicable to this scenario is that of Kelvin-Voigts (see Figure 4 below). Besides for its applicability this model is NOT used here as a vibrational model is more suitable.

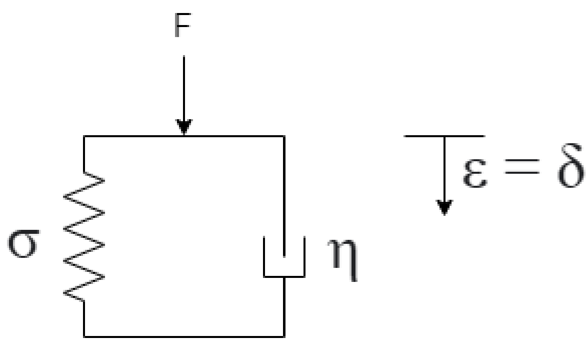



Figure 4: Kelvin Voigt Model

The Kelvin-Voigt model is characterised by the equation below (where strain is equivalent in the damper and the spring but the total stress is the sum of the stress experienced in the spring and the stress experienced in the damper).

$$\sigma(t) = E\varepsilon(t) + \eta \frac{d\varepsilon(t)}{\mathrm{dt}}$$

This is then solved to:

$$\varepsilon(t) = \frac{\sigma_0}{E}\left(1 - e^{\frac{-t}{\tau_R}}\right)$$

where:

$$\tau_R = \frac{\eta}{E}$$

Other information applicable to this model (later on) is:

$$\sigma = \left(\frac{2E\gamma_s}{\pi a}\right)^{\frac{1}{2}}$$

$\tau_{\text{theo}} = \left[\frac{2G}{(1-2\nu)}\right] e^{1-2\pi\frac{\omega}{b}}$ where $\frac{\omega}{b} = \frac{1}{1-\nu}$ for ductile materials

Knowing the average size of an abrasive (1.1mm) as well as the density of gelatin (680kg/m^3), we can calculate the mass of abrasive at various hydration levels (as follows):

```
vl = (4/3)*pi*(1.0025*10^-3)^3;
mass0 = vl*680;
```

Material Properties:

```
%E = ((0.97/(43.2*1000)) + (0.0001/(1100*10^9)) +(0.0029/(330*10^9)))^-1;
Egelatin = 43.2*10^3;
vgel = 0.5;
Esic = 330*10^9;
vsic = 0.14;
Ediamond = 1100*10^9;
vdiamond = 0.148;
Ewater = 0.9*10^3;
vwater = 0.5;

pgelatin = 680;
pwater = 997;
psic = 3020;
pdiamond = 3500;
```

The force balance diagram for the abrasive contact

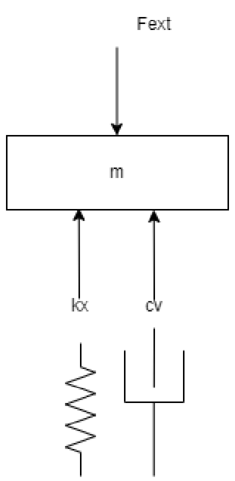



Figure 5: Force Balance on Figure 1

The force balance is characterized by:

$$m\ddot{x} + c\dot{x} + \mathrm{kx} = F_{\mathrm{ext}}$$

This can be simplified to:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = \frac{F_{\mathrm{ext}}}{m}$$

where: $\omega_n = \sqrt{\frac{k}{m}} = \sqrt{\frac{E}{m}}$ and $\zeta = \frac{c}{2m\omega_n}$ and the external force can be described by $F_{\mathrm{ext}} = F_0 \sin(\Omega t)$

Where all variables are as stated above (except t which is the time)

Ω can be described as the period of contact (where force is 0 as contact begins, building up to the largest force F0 before the workpiece applies an opposite force equal in magnitude (-F0) to remove the abrasive from the surface). See Figure 6 below:

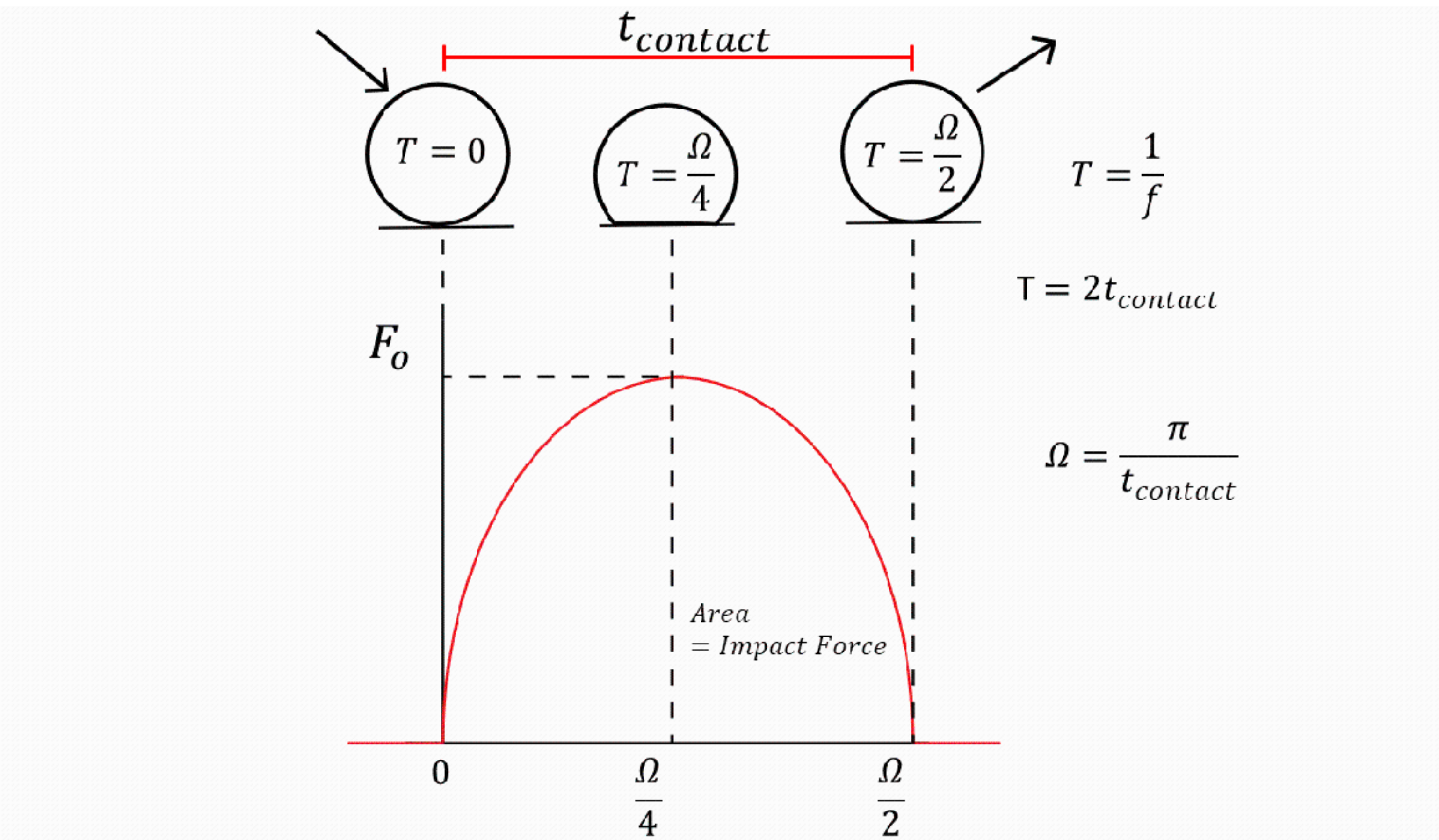



Figure 6: Period of Contact Force

The period of contact force can thus simply be described as half the inverse of contact time:

$$\Omega = \frac{\pi}{t_{\mathrm{contact}}}$$

An interesting paper by Roberts et al. shows the measurement of contact time in short duration sports ball impacts (which is very similar in nature to the impact of an assumed spherical abrasive with a harder workpiece). A golfball is assumed hit by a Titanium golf head and this makes the study even more relevant to this proposed model. Hocknell showed that a reasonable estimated of impact duration (contact time) could be made using the following formula derived from Hertz Law (adapted by Goldsmith):

$$\tau = 4.53\left[\frac{m_B(\delta_A+\delta_B)}{\sqrt{(R_B v_{\text{impinging}})}}\right]^{\frac{2}{5}}$$

where $\delta_A = \dfrac{1-\nu_A^2}{\pi E_A}$ and $\delta_B = \dfrac{1-\nu_B^2}{\pi E_B}$

where the subscript B denotes the abrasive and A denotes the workpiece.

m stands for mass, R stands for radius of abrasive, vimpinging is as above, $\nu$ is the Poisson ratio of each respective material and E is the elastic modulus of each respective material.

If we use the combined elastic modulus for a layered composite (which is described by:

$E_{\text{mix}} = \left(\dfrac{\mu_s}{E_s}+\dfrac{\mu_a}{E_a}\right)^{-1}$ where $\mu_n = \dfrac{h_n}{2h_{\text{substrate}}+h_{\text{outerlayer}}}$ and n will be either the substrate or outerlayer for s or a respectivelly.)

In our case: gelatin has an elastic modulus of 43.2kPa and a Poisson's ratio of 0.5 as well as a radius of 2.005mm/2 = 1.0025mm. Mass will vary for each hydration level (hydration is a direct function of mass)

radius is then calculated by:

$$r = \sqrt[3]{\frac{3m}{4\rho\pi}}$$

```
R0 = 0.5*10^-3;
mass0 = pgelatin*((4/3)*pi*(R0^3));

row10 = ((0.9/pgelatin) + (0.1/pwater))^-1;
row30 = ((0.7/pgelatin) + (0.3/pwater))^-1;
row50 = ((0.5/pgelatin) + (0.5/pwater))^-1;

mass10 = mass0*1.1;
mass30 = mass10*1.3;
mass50 = mass10*1.5;
ma = [mass0 mass0*1.1 mass10*1.3 mass10*1.5]
```

```
ma = 1×4
10^-6 ×
    0.3560    0.3917    0.5091    0.5875
```

```
ma2 = ma.*10^6
```

```
ma2 = 1×4
    0.3560    0.3917    0.5091    0.5875
```

```
R10 = ((3*mass10)/(4*row10*pi))^(1/3);
R30 = ((3*mass30)/(4*row30*pi))^(1/3);
```

```
R50 = ((3*mass50)/(4*row50*pi))^(1/3);

Rall = [R0 R10 R30 R50]
```

```
Rall = 1×4
10^-3 ×
    0.5000    0.5106    0.5448    0.5577
```

```
va = transpose(repmat([6.28 15 31.4 45 60],4,1));
vo = [6.28 15 31.4 45 60];

KEi = 0.5*mass0.*(vo.^2)
```

```
KEi = 1×5
10^-3 ×
    0.0070    0.0401    0.1755    0.3605    0.6409
```

```
KEi2 = KEi*10^3;
v10b = sqrt((KEi)./(0.5*mass10))
```

```
v10b = 1×5
    5.9877   14.3019   29.9387   42.9058   57.2078
```

```
v30b = sqrt((KEi)./(0.5*mass30))
```

```
v30b = 1×5
    5.2516   12.5436   26.2580   37.6309   50.1745
```

```
v50b = sqrt((KEi)./(0.5*mass50))
```

```
v50b = 1×5
    4.8890   11.6775   24.4449   35.0325   46.7099
```

```
p10 = 0.1;
p30 = 0.3;
p50 = 0.5;

pgel = 0.97;
psic = 0.029;
pdiam = 0.001;
```

The combined abrasive elastic modulus will then be given by:

$$E = \left(\frac{H_{\%}}{E_{\text{water}}} + \frac{(0.97 - H_{\%})}{E_{\text{gelatin}}} + \frac{(0.029)}{E_{\text{SiC}}} + \frac{0.001}{E_{\text{diamond}}}\right)^{-1}$$

which varies as a function of hydration level

```
E0 = ((0*(1/Ewater)) + ((pgel-0)*(1/Egelatin)) + (psic*(1/Esic)) + (pdiam*(1/Ediamond)))^-1;
E10 = ((p10*(1/Ewater)) + ((pgel-p10)*(1/Egelatin)) + (psic*(1/Esic)) + (pdiam*(1/Ediamond)))^-
E30 = ((p30*(1/Ewater)) + ((pgel-p30)*(1/Egelatin)) + (psic*(1/Esic)) + (pdiam*(1/Ediamond)))^-
E50 = ((p50*(1/Ewater)) + ((pgel-p50)*(1/Egelatin)) + (psic*(1/Esic)) + (pdiam*(1/Ediamond)))^-
```

Razali measured static modulus at different hydrations as:

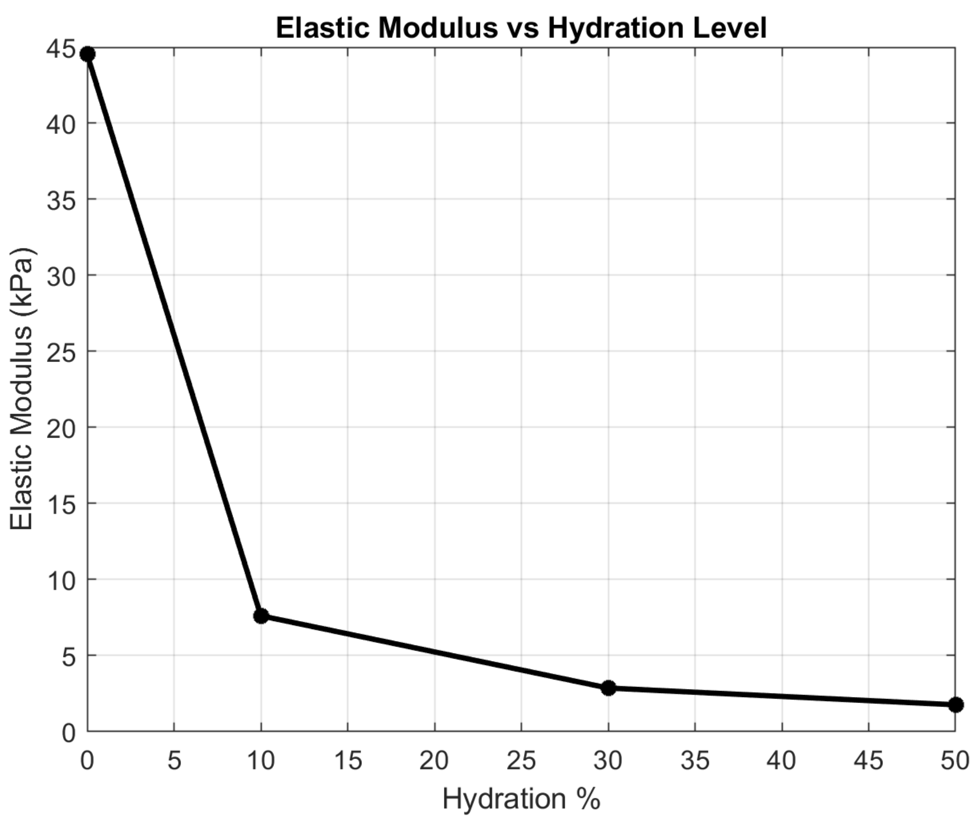
```
%E0 = 43.2*10^3;
%E10  = 7.8*10^3;
%E30 = (7.8*10^3)/10;
%E50 = (7.8*10^3)/15;

Eall = [E0 E10 E30 E50]
```

```
Eall = 1×4
10^4 ×
    4.4536    0.7619    0.2867    0.1765
```

```
eal = Eall*10^-3
```

```
eal = 1×4
   44.5361    7.6190    2.8666    1.7654
```

```
H = [0 10 30 50];
plot(H,Eall.*10^-3, '-*k','LineWidth',2.0);

grid on
xlabel('Hydration %')
ylabel('Elastic Modulus')

title('Elastic Modulus vs Hydration Level')
ylabel('Elastic Modulus (kPa)')
```



Poisson's ratio varies similarly to Elastic modulus but due to gelatin and water having the same values of Poisson's ratio, no change occurs.

```
pois0 = ((0*(1/vwater)) + ((pgel-0)*(1/vgel)) + (psic*(1/vsic)) + (pdiam*(1/vdiamond)))^-1;
pois10 = ((p10*(1/vwater)) + ((pgel-p10)*(1/vgel)) + (psic*(1/vsic)) + (pdiam*(1/vdiamond)))^-1
pois30 = ((p30*(1/vwater)) + ((pgel-p30)*(1/vgel)) + (psic*(1/vsic)) + (pdiam*(1/vdiamond)))^-1
pois50 = ((p50*(1/vwater)) + ((pgel-p50)*(1/vgel)) + (psic*(1/vsic)) + (pdiam*(1/vdiamond)))^-1
```

Using the notion of critical values (as in my previous derivation):

Effective elastic modulus: $E_{\text{combined}} = \dfrac{2}{\left(\dfrac{1-\upsilon_{\text{Ti6Al4V}}{}^{2}}{E_{\text{Ti6Al4V}}} + \dfrac{1-\upsilon_{\text{abrasive}}{}^{2}}{E_{\text{abrasive}}}\right)}$

Critical Yield Stress Coefficient: $C_{\text{abr}} = 1.295e^{0.736\upsilon_{\text{abr}}}$

We can then find critical deflection of spherical contact, critical spherical contact force and critical velocity for each condition of wetness and velocity:

```
Cconst = 1.295*exp(0.736*pois0);
Eallmatcom = ((1-0.4643^2)./Eall) + ((1-0.342^2)/(113.8*10^9));
Eb = 2./(Eallmatcom)
```

```
Eb = 1×4
10^5 ×
    1.1355    0.1943    0.0731    0.0450
```

```
xc = (4/3)*(Rall./Eb).^2;
zc = (Cconst*pi*(210*10^3)/2)^3;
Pc = xc.*zc;

wc = Rall.*((pi*(210*10^3)*Cconst)./(2*Eb)).^2;

vc = sqrt((4*wc.*Pc)./(5.*ma));


ma2 = (repmat(ma,5,1))
```

```
ma2 = 5×4
10^-6 ×
    0.3560    0.3917    0.5091    0.5875
    0.3560    0.3917    0.5091    0.5875
    0.3560    0.3917    0.5091    0.5875
    0.3560    0.3917    0.5091    0.5875
    0.3560    0.3917    0.5091    0.5875
```

```
va = transpose([vo;v10b;v30b;v50b])
```

```
va = 5×4
    6.2800    5.9877    5.2516    4.8890
   15.0000   14.3019   12.5436   11.6775
   31.4000   29.9387   26.2580   24.4449
```

```
   45.0000   42.9058   37.6309   35.0325
   60.0000   57.2078   50.1745   46.7099
```

```
wv = ((5*(va.^2).*ma2.*(wc.^(3/2)))./(Pc)).^(2/5)
```

```
wv = 5×4
    0.0008    0.0017    0.0025    0.0030
    0.0017    0.0034    0.0050    0.0060
    0.0031    0.0062    0.0090    0.0109
    0.0041    0.0082    0.0120    0.0145
    0.0051    0.0104    0.0151    0.0183
```

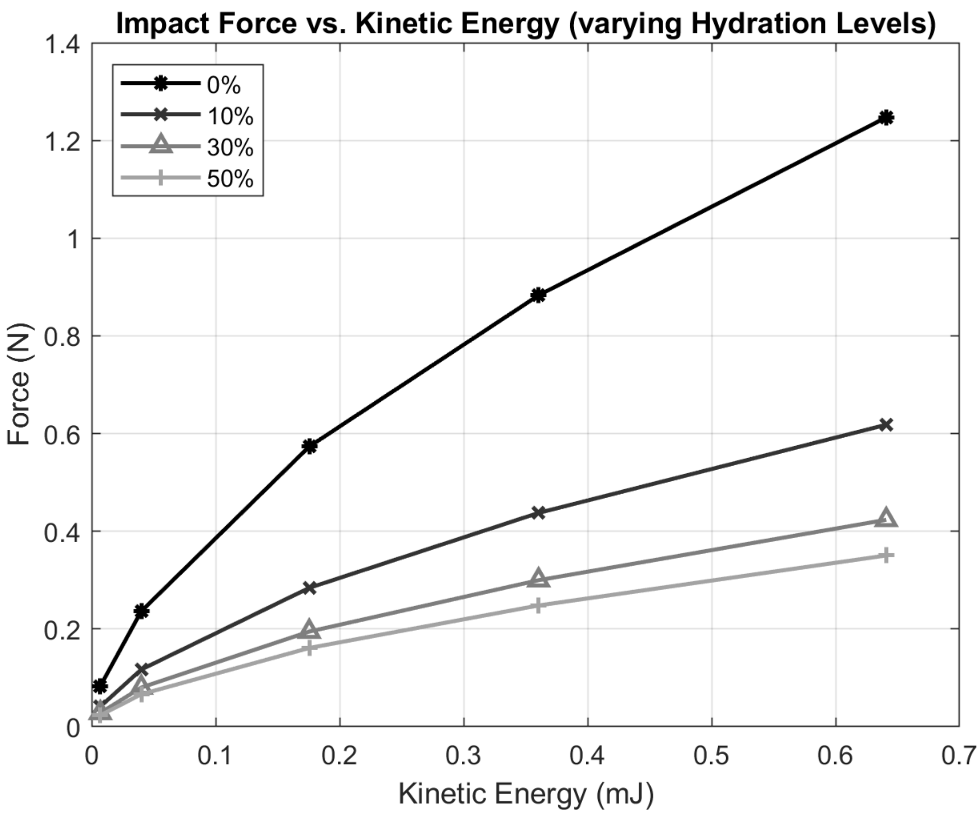
```
Fimp = Pc.*((wv./wc).^(3/2));
F0 = transpose(Fimp(:,1));
F10 = transpose(Fimp(:,2));
F30 = transpose(Fimp(:,3));
F50 = transpose(Fimp(:,4));

ac = (pi^3)*((Rall.*Cconst*(210*10^3)).^2)./(2*Eb)
```

```
ac = 1×4
    4.9998   30.4790   92.2197  156.9180
```

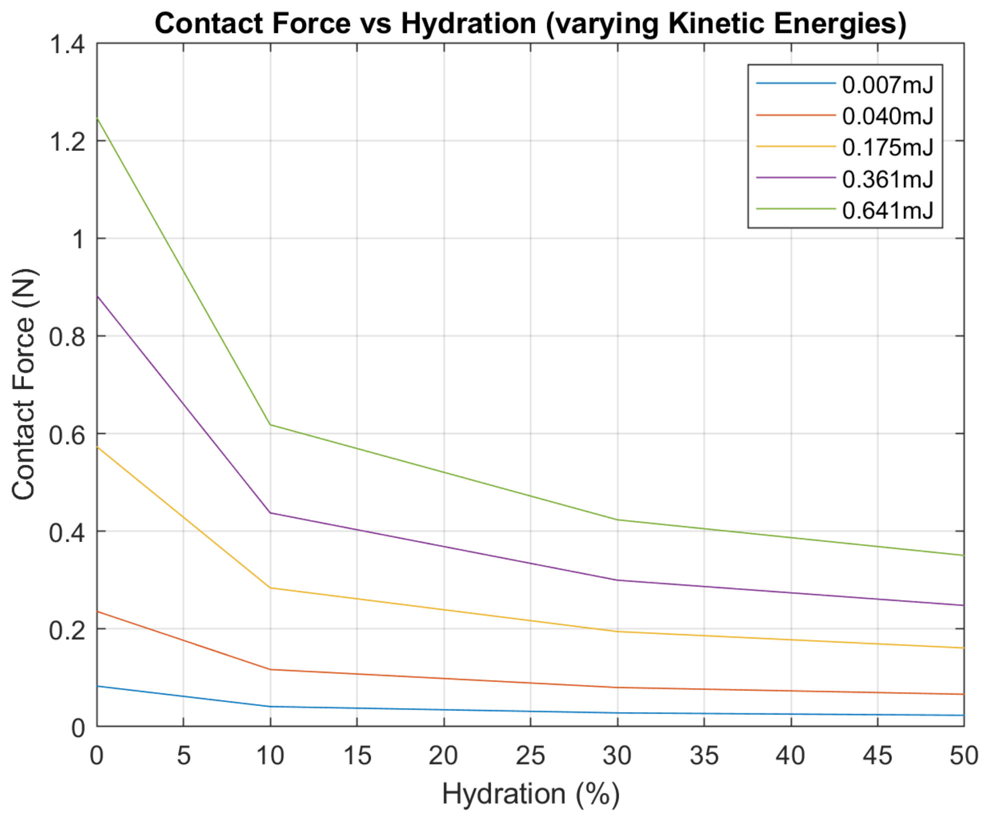
```
Csall = ((1.5*Fimp)./ac)
```

```
Csall = 5×4
    0.0249    0.0020    0.0005    0.0002
    0.0709    0.0058    0.0013    0.0006
    0.1720    0.0140    0.0032    0.0015
    0.2649    0.0215    0.0049    0.0024
    0.3742    0.0304    0.0069    0.0034
```

```
plot(KEi2, F0, '-*k','LineWidth',1.5);
hold on;
plot(KEi2, F10, '-x','LineWidth',1.5,'Color','#333333');
hold on;
plot(KEi2, F30, '-^','LineWidth',1.5,'Color','#7E7E7E');
hold on;
plot(KEi2, F50, '-+','LineWidth',1.5,'Color','#A3A3A3');
hold off;
grid on;
title('Impact Force vs. Kinetic Energy (varying Hydration Levels)');
legend('0%','10%','30%','50%', 'Location', 'NorthWest');
xlabel('Kinetic Energy (mJ)');
ylabel('Force (N)');
```




```
forceall = [F0; F10; F30; F50];
forcest = transpose(forceall);
Hforce = [0 10 30 50];

plot(Hforce, forcest(1,:));
hold on;
plot(Hforce, forcest(2,:));
hold on;
plot(Hforce, forcest(3,:));
hold on;
plot(Hforce, forcest(4,:));
hold on;
plot(Hforce, forcest(5,:));

title('Contact Force vs Hydration (varying Kinetic Energies)');
xlabel('Hydration (%)');
ylabel('Contact Force (N)');

legend('0.007mJ','0.040mJ','0.175mJ','0.361mJ','0.641mJ', 'Location','northeast');
hold off

grid on
```



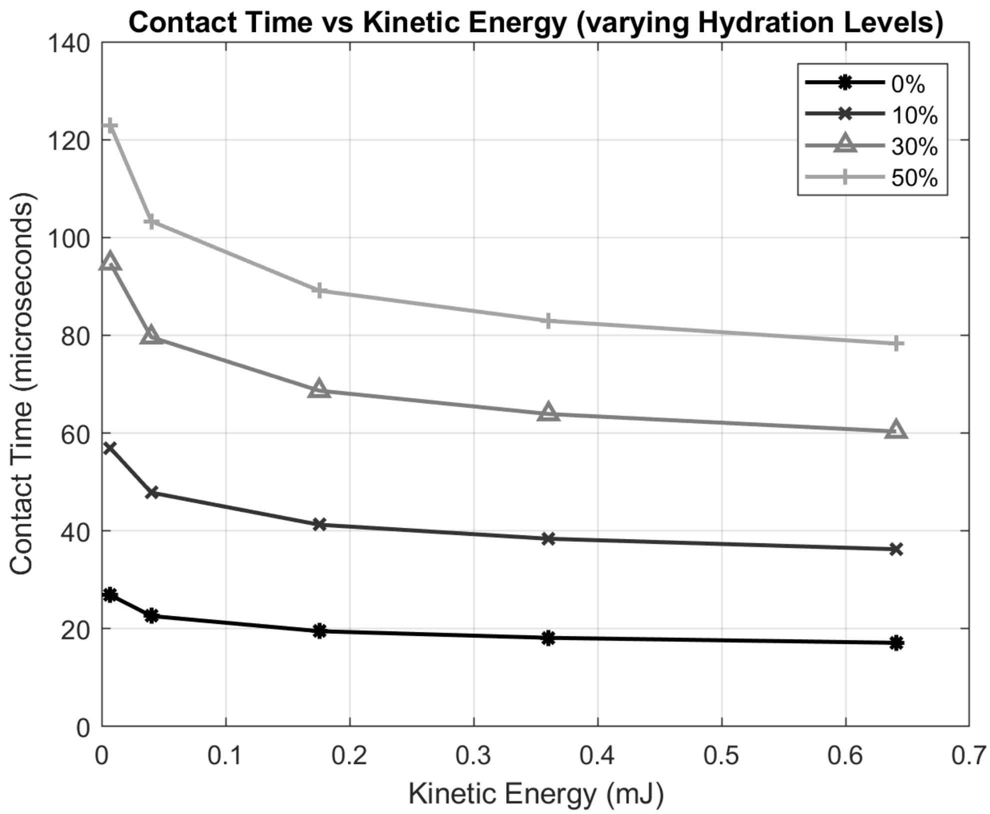

$$\sigma = \frac{1.5F_o}{C.A.} = \frac{1.5F_o}{\pi(2\delta R - \delta^2)}$$

```
v0 = vo
```

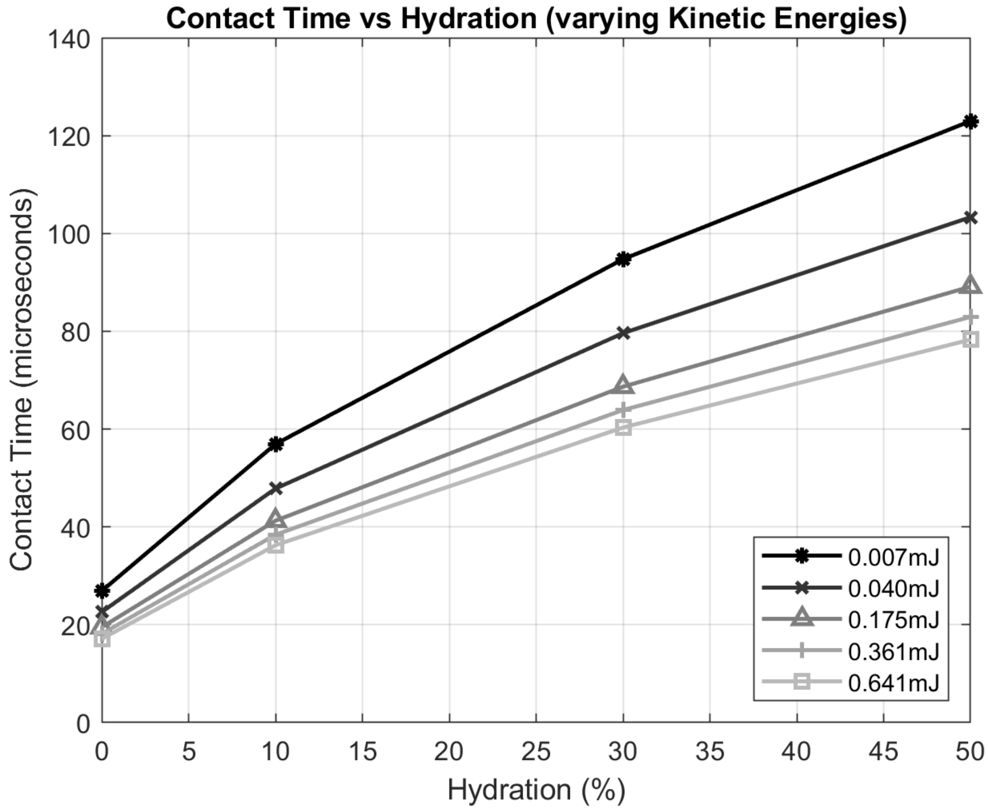
```
v0 = 1×5
    6.2800   15.0000   31.4000   45.0000   60.0000
```

```
t0 = (mass0*(v0))./F0;

t10 = (mass10*(v10b))./F10;

t30 = (mass30*(v30b))./F30;

t50 = (mass50*(v50b))./F50;

plot(KEi2, t0.*10^6, '-*k','LineWidth',1.5);
hold on;
plot(KEi2, t10.*10^6, '-x','LineWidth',1.5,'Color','#333333');
hold on;
plot(KEi2, t30.*10^6, '-^','LineWidth',1.5,'Color','#7E7E7E');
hold on;
plot(KEi2, t50.*10^6, '-+','LineWidth',1.5,'Color','#A3A3A3');
legend('0%','10%','30%','50%');
xlabel('Kinetic Energy (mJ)');
ylabel('Contact Time (microseconds)');
```

```
grid on;
hold off;
title('Contact Time vs Kinetic Energy (varying Hydration Levels)');
```




```
timeall = [t0; t10; t30; t50];
timest = transpose(timeall);
Htime = [0 10 30 50];

plot(Htime, timest(1,:).*10^6, '-*k','LineWidth',1.5);
hold on;
plot(Htime, timest(2,:).*10^6, '-x','LineWidth',1.5,'Color','#333333');
hold on;
plot(Htime, timest(3,:).*10^6, '-^','LineWidth',1.5,'Color','#7E7E7E');
hold on;
plot(Htime, timest(4,:).*10^6, '-+','LineWidth',1.5,'Color','#A3A3A3');
hold on;
plot(Htime, timest(5,:).*10^6, '-s','LineWidth',1.5,'Color','#B9B9B9');

title('Contact Time vs Hydration (varying Kinetic Energies)');
xlabel('Hydration (%)');
ylabel('Contact Time (microseconds)');

legend('0.007mJ','0.040mJ','0.175mJ','0.361mJ','0.641mJ', 'Location','southeast');
hold off

grid on
```




Now that we have contact forces and contact times for various wetness levels and impinging speeds we can move onto the vibrational analysis:

A note:

Using geometry we can find deformation to contact radius and area (assuming circular contact):

$$a = \sqrt{2\delta R - \delta^2}$$
$$C.A. = \pi a^2$$

The maximum value of x is the maximum deformation occuring:

$$x = \delta$$

From Contact Mechanics we can get the Contact Stress (for a sphere interacting with a plane) as:

$$\sigma = \frac{1.5F_o}{C.A.} = \frac{1.5F_o}{\pi(2\delta R - \delta^2)}$$

For an undamped solution (zeta = 0):

$$A = \frac{\frac{F_o}{k}}{1 - r^2}$$

$$\phi = 0$$

$$x = \mathrm{Asin}(\Omega t)$$

find the period of force contact by:

```
omega0 = pi./(t0);
```

while the natural frequency is stated by:

```
mabr = mass0;
omegan = sqrt(E0/mabr);
```

The ratio is then calculated by:

```
r = omega0./omegan;
```

We can then get the maximum displacement at various contact velocities by (sin90 = 1), therefore we can omit the sin term for maximum displacement):

```
H0 = 0;
v0 = [6.28 15 31.4 45 60];

siu = vo/omegan;

asq = 2*siu*Rall(1) - siu.^2;
asp = asq*pi;
cs0 = (1.5*F0)./asp;

cs0MPa = cs0*10^-6
```

```
cs0MPa = 1×5
    2.2754    2.7779    3.3841    3.7968    4.2271
```

Now that we have the undamped case completed, we need to calculate the displacements (and subsequently the contact areas and contact stresses) for varying hydration levels. This is slightly more complex in nature than the undamped solution as we need to find damping ratios.

This shows that the natural frequency changes slightly as per:

```
wn10 = sqrt(E10/mass10)
```

```
wn10 = 1.3948e+05
```

```
ccrit10 = mass10*(wn10);
wn30 = sqrt(E30/mass30)
```

```
wn30 = 7.5035e+04
```

```
ccrit30 = mass30*(wn30)
```

```
ccrit30 = 0.0382
```

```
wn50 = sqrt(E50/mass50)
```

```
wn50 = 5.4819e+04
```

```
ccrit50 = mass50*(wn50)
```

```
ccrit50 = 0.0322
```

zeta then changes as per (each term must be multiplied by c here):

```
zeta10c = 1/(2*mass10*wn10);
zeta30c = 1/(2*mass30*wn30);
zeta50c = 1/(2*mass50*wn50);
```

Note that $\zeta = \sqrt{1 - \left(\frac{\Omega}{\omega_n}\right)^2}$

we must first calculate contact times for each wetness by:

```
t10 = transpose(t10);

t30 = transpose(t30);

t50 = transpose(t50);
```

we can then get omega (contact period) by:

```
omega10 = pi./t10;
omega30 = pi./t30;
omega50 = pi./t50;
```

we can then get omega ratio by:

```
omrat10 = omega10./wn10
```

```
omrat10 = 5×1
    0.3956
    0.4709
    0.5458
    0.5866
    0.6213
```

```
omrat30 = omega30./wn30
```

```
omrat30 = 5×1
    0.4419
    0.5260
    0.6097
    0.6552
    0.6940
```

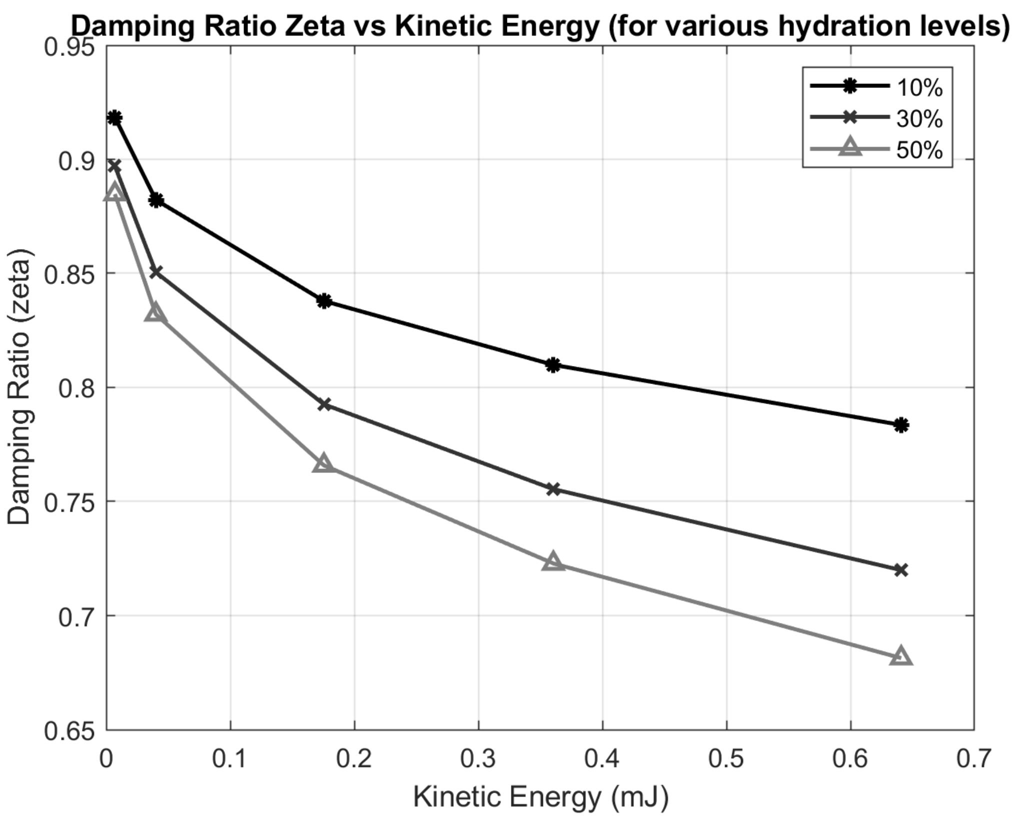
```
omrat50 = omega50./wn50
```

```
omrat50 = 5×1
```

```
0.4661
0.5547
0.6430
0.6910
0.7319
```

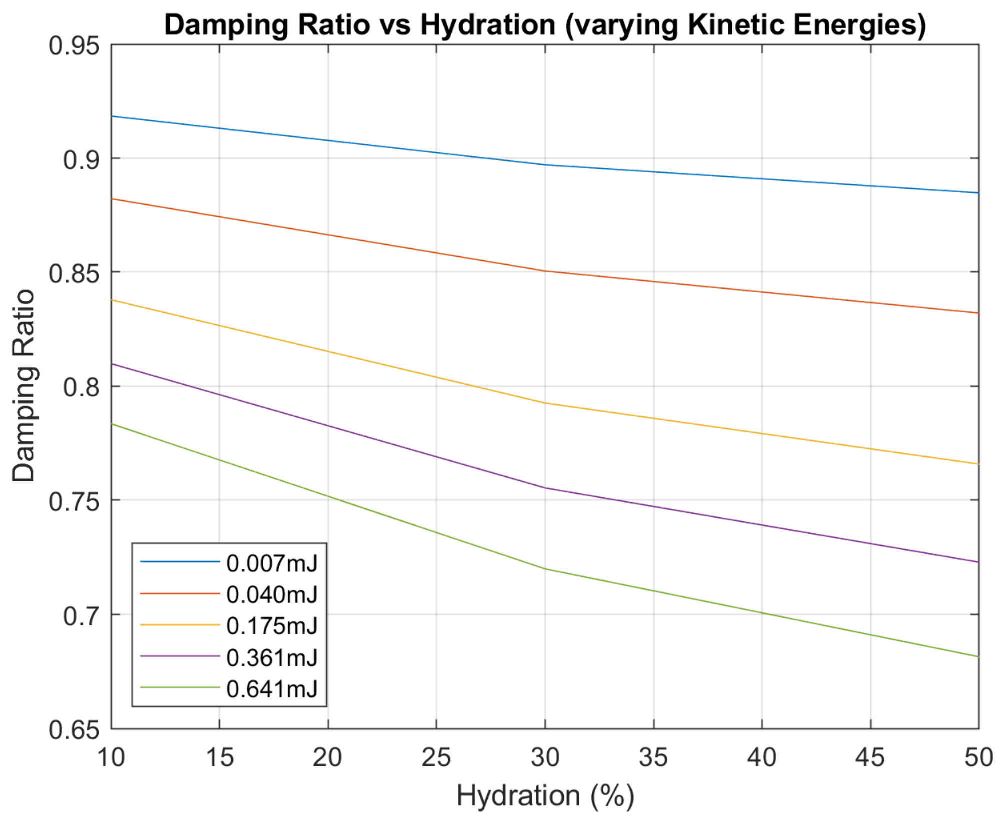
we can then get actual zeta by:

```
zeta10 = sqrt((1 - omrat10.^2));
zeta30 = sqrt((1 - omrat30.^2));
zeta50 = sqrt((1 - omrat50.^2));

plot(KEi2,zeta10, '-*k','LineWidth',1.5);
hold on;
plot(KEi2,zeta30, '-x','LineWidth',1.5,'Color','#333333');
hold on;
plot(KEi2,zeta50, '-^','LineWidth',1.5,'Color','#7E7E7E');
title('Damping Ratio Zeta vs Kinetic Energy (for various hydration levels)');
xlabel('Kinetic Energy (mJ)');
ylabel('Damping Ratio (zeta)');
legend('10%','30%','50%','Location','northeast');
hold off;
grid on;
```




```
zetaall = [zeta10, zeta30, zeta50];
zetast = transpose(zetaall);
Hzeta = [10 30 50];
```

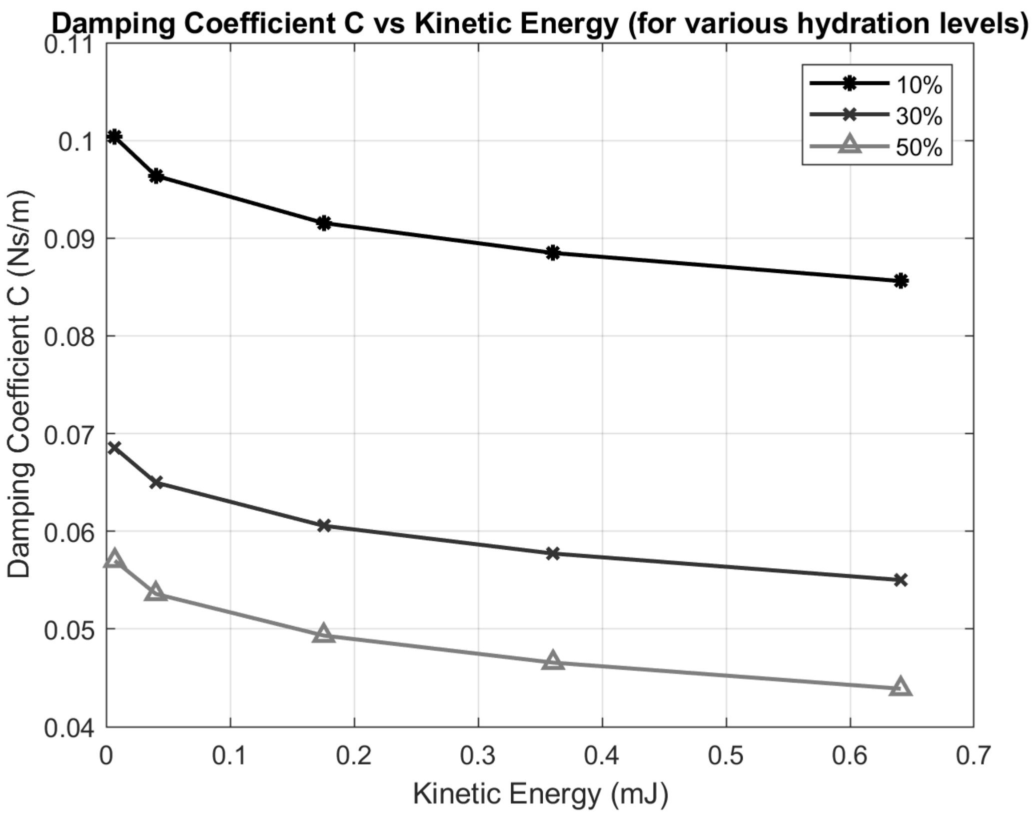
```
plot(Hzeta, zetaall(1,:));
hold on;
plot(Hzeta, zetaall(2,:));
hold on;
plot(Hzeta, zetaall(3,:));
hold on;
plot(Hzeta, zetaall(4,:));
hold on;
plot(Hzeta, zetaall(5,:));

title('Damping Ratio vs Hydration (varying Kinetic Energies)');
xlabel('Hydration (%)');
ylabel('Damping Ratio');

legend('0.007mJ','0.040mJ','0.175mJ','0.361mJ','0.641mJ', 'Location','southwest');
hold off

grid on
```




to find c:

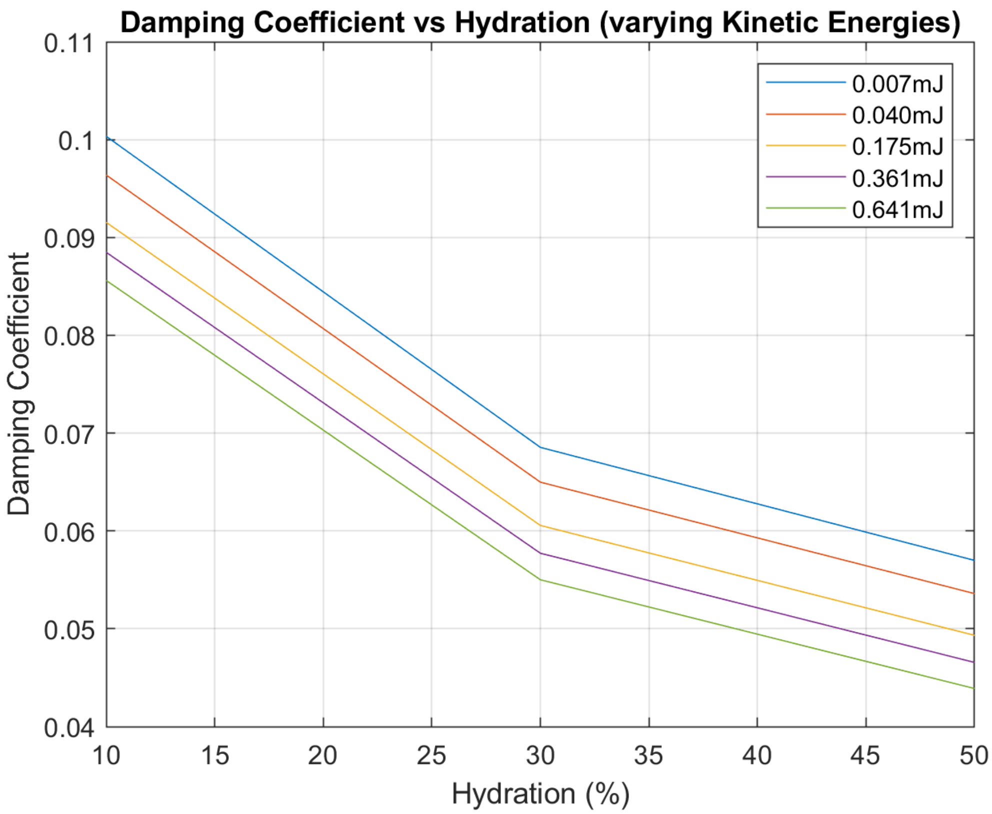
```
c10 = zeta10./zeta10c;
c30 = zeta30./zeta30c;
c50 = zeta50./zeta50c;
plot(KEi2,c10, '-*k','LineWidth',1.5);
hold on;
plot(KEi2,c30, '-x','LineWidth',1.5,'Color','#333333');
hold on;
plot(KEi2,c50, '-^','LineWidth',1.5,'Color','#7E7E7E');
```

```
title('Damping Coefficient C vs Kinetic Energy (for various hydration levels)');
xlabel('Kinetic Energy (mJ)');
ylabel('Damping Coefficient C (Ns/m)');
legend('10%','30%','50%','Location','northeast');
hold off;
grid on;
```




```
dampall = [c10, c30, c50];
dampst = transpose(dampall);
Hdamp = [10 30 50];

plot(Hdamp, dampall(1,:));
hold on;
plot(Hdamp, dampall(2,:));
hold on;
plot(Hdamp, dampall(3,:));
hold on;
plot(Hdamp, dampall(4,:));
hold on;
plot(Hdamp, dampall(5,:));

title('Damping Coefficient vs Hydration (varying Kinetic Energies)');
xlabel('Hydration (%)');
ylabel('Damping Coefficient');

legend('0.007mJ','0.040mJ','0.175mJ','0.361mJ','0.641mJ', 'Location','northeast');
```

```
hold off

grid on
```




We should now attempt to get contact stresses for each hydration

```
H10 = 10;
H30 = 30;
H50 = 50;

A10 = v10b./transpose(omega10);
A30 = v30b./transpose(omega30);
A50 = v50b./transpose(omega50);

ta10 = omrat10./zeta10
```

```
ta10 = 5×1
    0.4308
    0.5337
    0.6514
    0.7243
    0.7929
```

```
tb10 = (atan(ta10))./omega10
```

```
tb10 = 5×1
10^-5 ×
    0.7371
    0.7465
    0.7584
```

```
    0.7662
    0.7736
```

```
x10ab = (transpose(v10b)./omega10).*exp(-zeta10.*wn10.*tb10).*sin(omega10.*tb10)
```

```
x10ab = 5×1
10^-3 ×
    0.0167
    0.0409
    0.0885
    0.1295
    0.1761
```

```
R = Rall;

asq10 = 2*x10ab.*R(2) - x10ab.^2;
asp10 = asq10*pi;
cs10 = (1.5*F10)./transpose(asp10);

cs10MPa = abs(cs10*10^-6);

ta30 = omrat30./zeta30
```

```
ta30 = 5×1
    0.4926
    0.6185
    0.7693
    0.8674
    0.9641
```

```
tb30 = (atan(ta30))./omega30
```

```
tb30 = 5×1
10^-4 ×
    0.1380
    0.1403
    0.1433
    0.1453
    0.1473
```

```
x30ab = (transpose(v30b)./omega30).*exp(-zeta30.*wn30.*tb30).*sin(omega30.*tb30)
```

```
x30ab = 5×1
10^-3 ×
    0.0276
    0.0683
    0.1492
    0.2201
    0.3018
```

```
asq30 = 2*x30ab.*R(3) - x30ab.^2;
asp30 = asq30*pi;
cs30 = (1.5*F30)./transpose(asp30);

cs30MPa = abs(cs30*10^-6);
```

```
ta50 = omrat50./zeta50
```

```
ta50 = 5×1
    0.5268
    0.6667
    0.8396
    0.9560
    1.0742
```

```
tb50 = (atan(ta50))./omega50
```

tb50 = 5×1

$10^{-4}$ ×

```
    0.1898
    0.1934
    0.1981
    0.2014
    0.2047
```

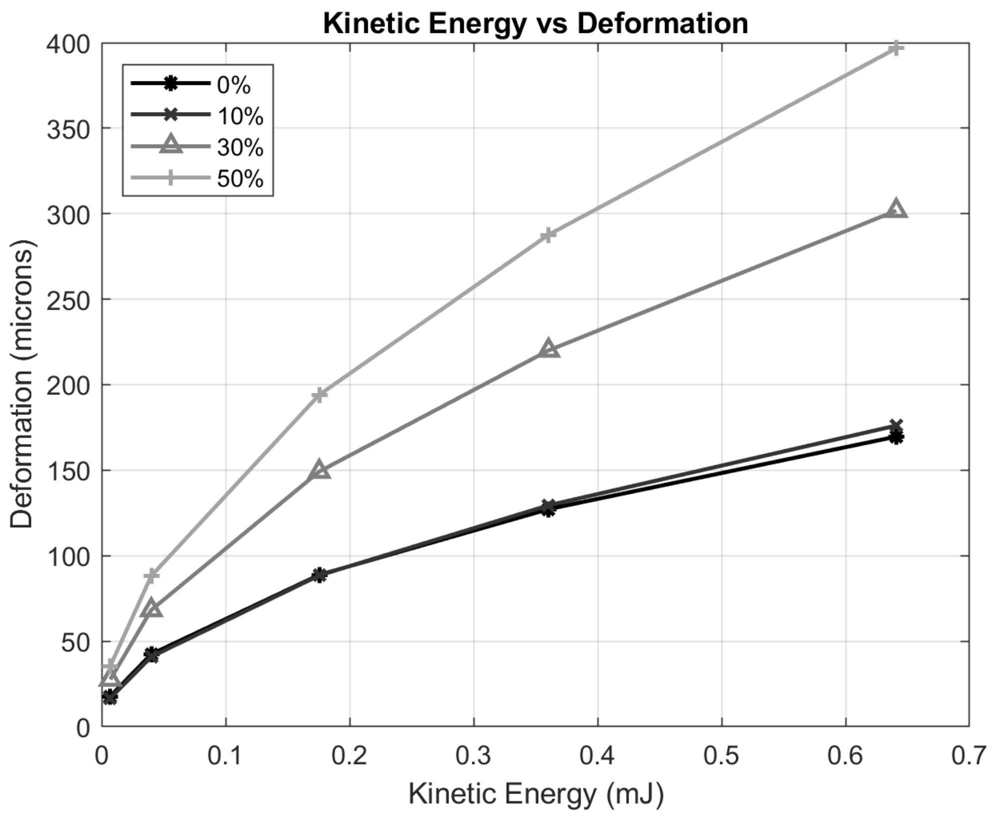
```
x50ab = (transpose(v50b)./omega50).*exp(-zeta50.*wn50.*tb50).*sin(omega50.*tb50)
```

x50ab = 5×1

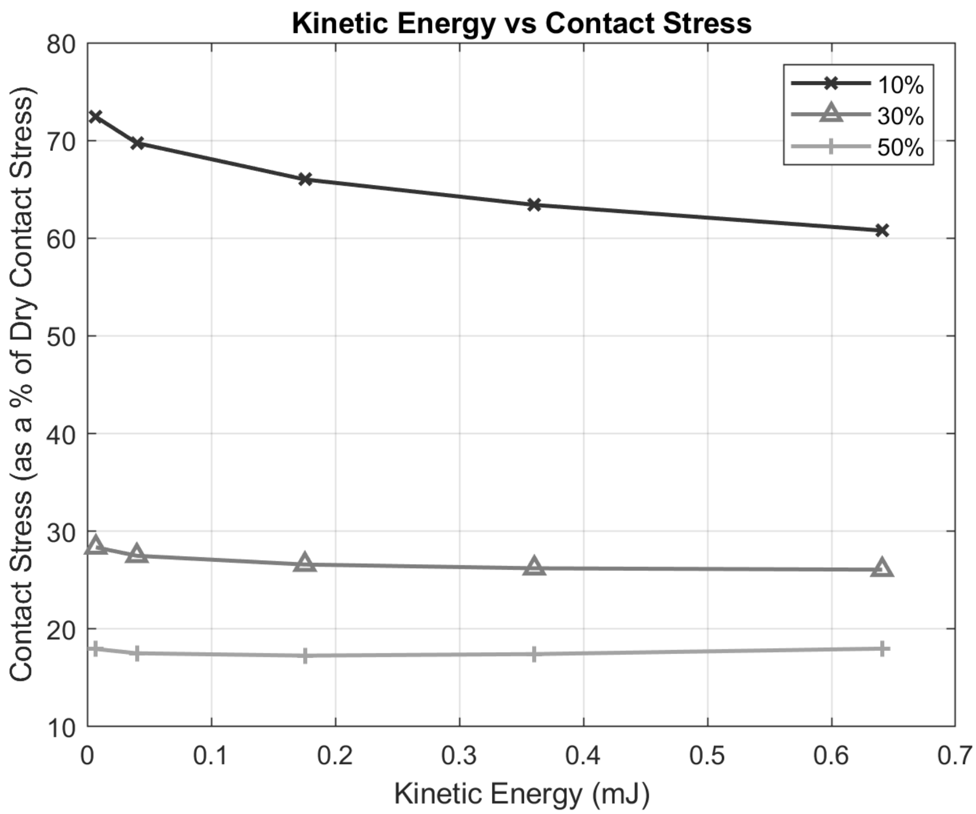
$10^{-3}$ ×

```
    0.0355
    0.0882
    0.1941
    0.2877
    0.3967
```

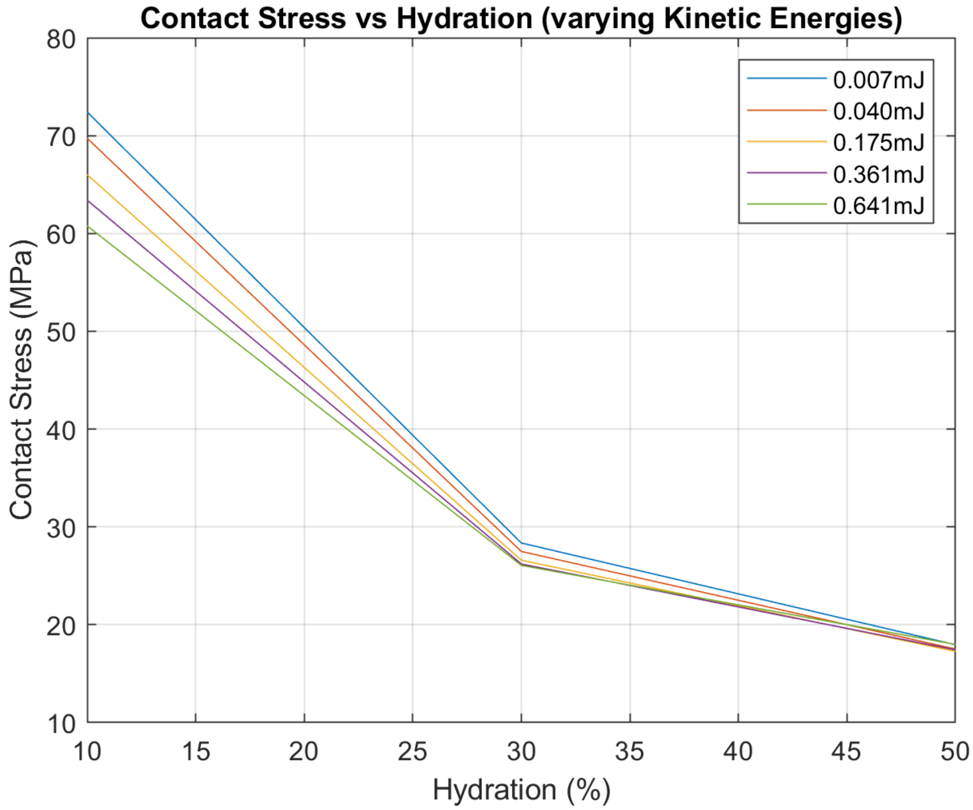
```
asq50 = 2*x50ab.*R(4) - x50ab.^2;
asp50 = asq50*pi;

cs50 = (1.5*F50)./transpose(asp50);

cs50MPa = cs50*10^-6;

plot(KEi2, siu.*10^6, '-*k','LineWidth',1.5);
hold on;
plot(KEi2, x10ab.*10^6, '-x','LineWidth',1.5,'Color','#333333');
hold on;
plot(KEi2, x30ab.*10^6, '-^','LineWidth',1.5,'Color','#7E7E7E');
hold on;
plot(KEi2, x50ab.*10^6, '-+','LineWidth',1.5,'Color','#A3A3A3');
%ylim([0,25])


grid on;
title('Kinetic Energy vs Deformation');
xlabel('Kinetic Energy (mJ)');
ylabel('Deformation (microns)');
legend('0%','10%','30%','50%','Location','northwest');

hold off;
```




```
plot(KEi2, (cs10MPa./cs0GPa).*100, '-x','LineWidth',1.5,'Color','#333333');
hold on;
plot(KEi2, (cs30MPa./cs0GPa).*100, '-^','LineWidth',1.5,'Color','#7E7E7E');
hold on;
plot(KEi2, (cs50MPa./cs0GPa).*100, '-+','LineWidth',1.5,'Color','#A3A3A3');
%ylim([0,25])


grid on;
title('Kinetic Energy vs Contact Stress');
xlabel('Kinetic Energy (mJ)');
ylabel('Contact Stress (as a % of Dry Contact Stress)');
legend('10%','30%','50%','Location','northeast');

hold off;
```




```
csall = [cs0GPa./cs0GPa; cs10MPa./cs0GPa; cs30MPa./cs0GPa; cs50MPa./cs0GPa];
cst = transpose(csall);
H = [10 30 50];

plot(H, cst(1,2:4).*100);
hold on;
plot(H, cst(2,2:4).*100);
hold on;
plot(H, cst(3,2:4).*100);
hold on;
plot(H, cst(4,2:4).*100);
hold on;
plot(H, cst(5,2:4).*100);

title('Contact Stress vs Hydration (varying Kinetic Energies)');
xlabel('Hydration (%)');
ylabel('Contact Stress (MPa)');

legend('0.007mJ','0.040mJ','0.175mJ','0.361mJ','0.641mJ', 'Location','northeast');

grid on;
hold off;
```


Contact Stress vs Hydration (varying Kinetic Energies)
0.007mJ
0.040mJ
0.175mJ
0.361mJ
0.641mJ
Contact Stress (MPa)
Hydration (%)
10
20
30
40
50
60
70
80
10
15
20
25
30
35
40
45
50