09-723 Proximal probe techniques. Name_____
Homework #6.

Due by 6:30 PM, Thursday, October 14, 2004

Download the DMTForceCurveExplorer.zip file which contains all necessary Matlab files by clicking on it and choosing to save it on your computer (e.g. on your desktop). Open the archive on your computer and extract it to some folder. Make sure that a path is set in Matlab to this folder. To set a path, choose the set path option in the file menu of Matlab. Click the Add with subfolders button and locate the folder to be added to the path in the pop-up window and click OK. Then click Save and Close.

I will be out of town from Friday night (October 8) until late on October 17. I do not know if I will have e-mail access while I am away.

justinl@andrew.cmu.edu
phone: 8-9175

DMTForceCurveExplorer (Some assembly required)

The DMTForceCurveExplorer simulates simple AFM force curve experiments using the DMT potential.

All necessary parameters are entered through the provided Graphical User Interface (GUI), which also controls the execution of the Simulink model, which you will need to construct in Simulink, with some minor adjustments to your model from Homework#4. Save the new model as **ForceCurveModelDMT.mdl**. Do not save over your old model. Save the model in the folder that you extracted the contents of **DMTForceCurve.zip** into. You should have already set a path to this folder. Make sure that in your model you call the variables as follows:

Mass: `Mtip` (As promised in previous homework, this is now the mass of the tip)
Spring constant: `Kspring`
Damping Coefficient: `Bdamp`
The simulation step (Fixed-step size): `SimStep`
The total simulation time (i.e. Stop Time)= `SimTime`
The initial velocity = `Zprime` (initial condition for the first integrator)
The initial position = `Zstart` (initial condition for the second integrator and the drive offset)
The position trajectory = `Ztraj`
The time = `time`
The driving force amplitude = `Fo`
The drive frequency (operating frequency in Hz) = `Foper`
The phase = `Phi`
The potential energy = Penergy
The kinetic energy = Kenergy
The damping power = Bpower
The damping energy = Benergy
The drive power = Dpower
The drive energy = Denergy
The ramp stop distance = Zend
The ramp position = RampPos
The Hamaker constant = Atip
The tip radius = Rtip
Sigma = sigma
The Lennard Jones force = FLJ
The ramping rate = RampRate
Indent constant = IndentConst
aDMT = aDMT

If these names are not used, the GUI will not work.
All other variables will be defined through the GUI.
Steps in construction of your model:

I. Make sure your model uses a fixed step.

In the **Simulation** menu, select **Configuration Parameters. . .** and set **Type**: to **Fixed-Step**, and **Solver**: to **ode4 (Runge-Kutta)**, and **Fixed-Step Size** to **SimStep.**

II. Adding the DMT Force to the model (Replaces Lennard-Jones Force)

Place two **Fcn** block (found under User-Defined Functions) in your model. Use the z signal as the input to both **Fcn** blocks. In the Expression box of the 1st **Fcn** block, type exactly:

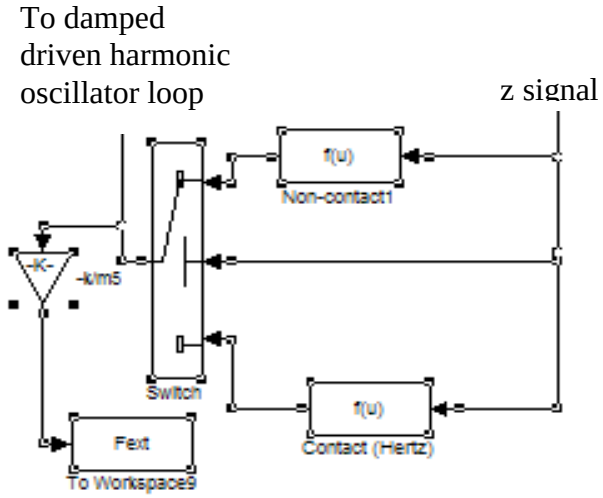
(-((Atip*Rtip)/6)*pow(u,-2))/Mtip

In the Expression box of the 2nd **Fcn** block, type exactly:

(IndentConst*pow(abs((aDMT-u)),(3/2)))/Mtip+(-((Atip*Rtip)/6)*pow(aDMT,-2))/Mtip

Add a **Switch** (found under **Signal Routing**) to the model. The output of the 1st **Fcn** is the input for the top input signal of the **Switch.** The output of the 2nd **Fcn** is the input for the bottom input signal of the **Switch.** The middle input of the **Switch** is the z signal. For the **Switch,** set the **Criteria for passing first input** to **u2> Threshold.** Set the **Threshold** to **aDMT.** Send the signal to the **Add** block of the damped driven harmonic oscillator loop.

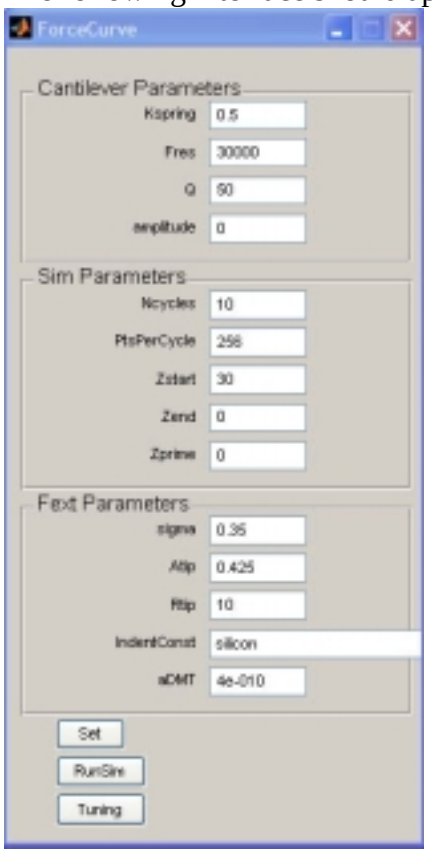
Also, send the signal from this **Switch** through a **Gain** that multiplies it my **Mtip** to the workspace as **Fext** using a **To Workspace** block.



Before running the model you need to provide the values of other parameters using the provided GUI. To use the GUI provided, run `ForceCurveDMT` from command line.

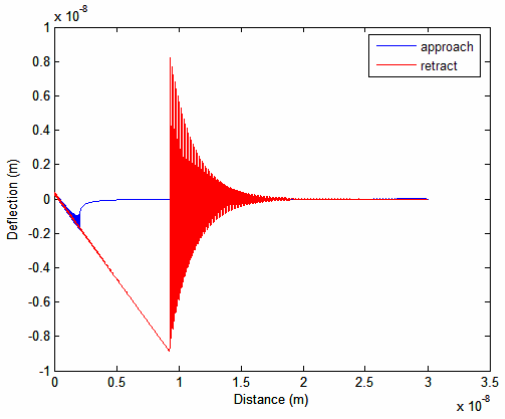
>> `ForeCurveDMT`

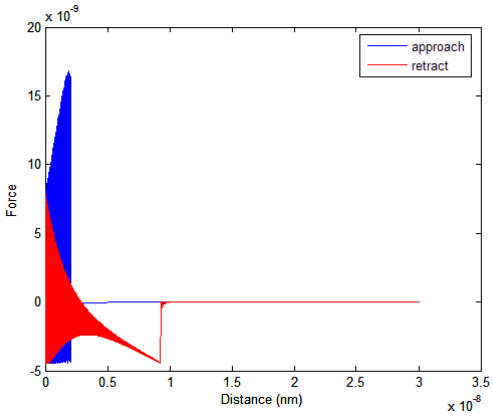
The following interface should appear:



Fres – define your desired resonance frequency in Hz, the GUI will use this and Kspring to determine Mtip, etc.
Q – the quality factor of the cantilever.
amplitude – the desired free amplitude of the cantilever in tapping mode (set it to zero for contact mode force curves)
Ncycles – this is scaled in Q cycles now, i.e. the number of cycles is Ncycles times Q.
Zstart – defines the starting and ending distance from the surface (in nm). For tapping mode, the distance is measured from the bottom of the oscillation
Zend – define force curve depth (in nm). For tapping mode, the distance is measured from the bottom of the oscillation.
IndentConst – select the sample material, and the GUI will generate the proper indentation constant.
Set – sets parameters, needs to be run before simulations.
Run Sim – run the sim
Tuning – opens the cantilever tuning menu. In tapping mode, the cantilever will need to be tuned before running a simulation.

Click the **Set** button and then the **RunSim** button and the following plots should appear (this may take a few minutes as there are several calculations being performed):





There are no tapping mode questions in this homework, but in an effort to work ahead due to my prolonged absence, this GUI is equipped to do some basic tapping mode functions. You don't need to worry about these parameters or functions for now. If you

are curious about these simulations, I have included some instructions on how to set the tapping mode parameters of the model at the end of this homework.

For a nice summary of the history of the development of some of these models, pictures of adhesion related deformations, and a summary of some theoretical issues, follow the link provided:

http://www.clarkson.edu/projects/crcd/me537/downloads/Rimai_1.pdf

Part I: Contact Mode Force Curve with DMT force

1.1

Set **Kspring** = 1, **Fres** = 30000, **Q** = 50, **amplitude** = 0, **Ncycles =** 10, **PtsPerCycle** = 256, **Zstart** = 10, **Zend** = -10, **Zprime** = 0, **sigma** = 0.35, **Atip** = 1e-10, **Rtip** = 10, and **aDMT** = 4e-10.

Notice that **Atip** is very small (very near to zero), indicating that we are operating under conditions when long range interactions are screened (e.g., in non-electrolyte solution). Compare force curves for **IndentConst** set to **silicon** and for a **soft polymer.** Plot them on the same plot and determine the shape of the tip used in the model from these force curves. Set **Atip** back to 0.425 and run force curves for both **silicon** and **soft polymer** again. Plot these on the same plot. What needs to be done to perform the same analysis for determining tip shape under these conditions?

Part II: Contact Pressures

2.1

The contact pressure of the indented surface, *P(r,x),* is given by :

$$P = \frac{3kr}{2\pi R}\sqrt{1-x^2} - \frac{\sqrt{\frac{3kW_{1,2}}{2\pi r}}}{\sqrt{1-x^2}}$$

Where

$$k = \frac{4}{3\pi}\frac{1}{\kappa_{eff}}$$

and

$$\kappa_{eff} = \kappa_1 + \kappa_2$$

and

$$\kappa_n = \frac{1-\nu_n^2}{\pi E_n}$$ where n can be 1 and 2.

$W_{1,2}$ is the work of adhesion, r is the contact radius and R is the effective radius, which becomes the radius of the tip when we consider a surface with infinite radius. $x = \frac{r}{r_o}$, where r_o is the radius of contact in the absence of external force, and is equal to

$$r_o = \left(\frac{6\pi R^2 W_{1,2}}{k} \right)^{1/3}$$

Any material is expected to become fully plastic under the pressure exceeding

$$P \approx 3\,\sigma_y$$

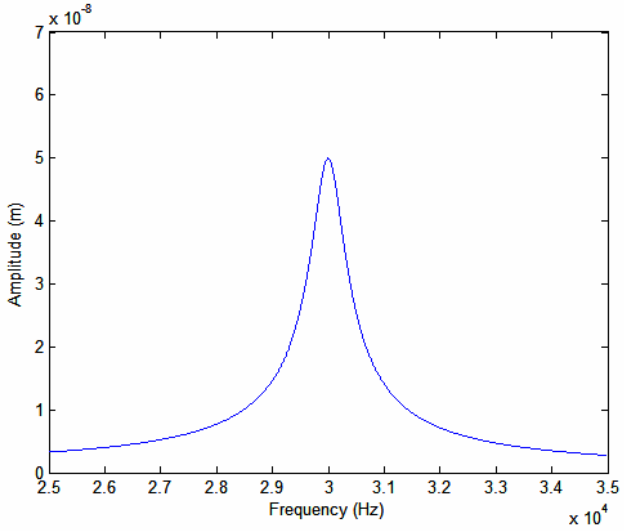
where σ_y is the yield strength. Silicon tip with the radius R=20 nm is in contact with the flat surface of polystyrene (σ_y = 40 MPa). What is the load F above which the sample will exhibit plastic deformation? Show the answer for the JKR contact (assume that $W_{1,2}$ = 7.0 mJ/m^2).

Appendix: basic tapping mode simulations

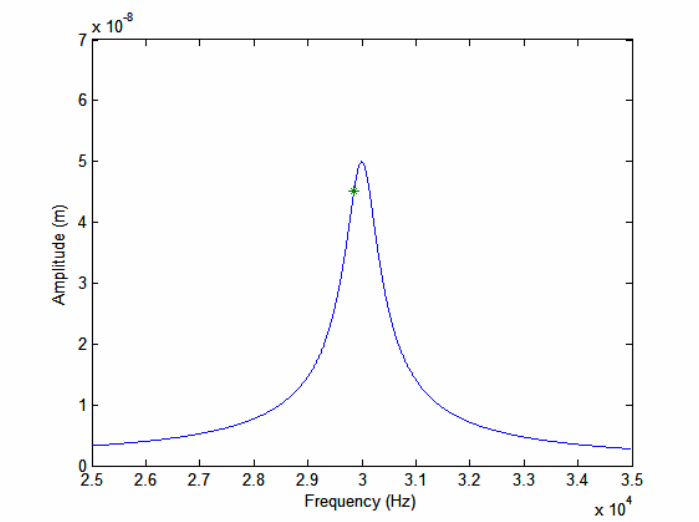
For tapping mode simulations, you will need to tune your cantilever. Change the **amplitude** to a positive value (try 50 to get the exact same plots shown later) and click the **Set** button. Now press the **Tuning** button. The following GUI should appear:

SweepStart – defines the starting frequency (in Hz) for plotting the amplitude resonance of the cantilever.
SweepWidth – defines the sweep size in Hz for plotting the amplitude resonance of the cantilever.
ResPeak – plots the free amplitude of a cantilever as a function of frequency (Hz)
Tune – prompts for the percent off resonance desired for the operating frequency and if the user wishes to operate above or below resonace.

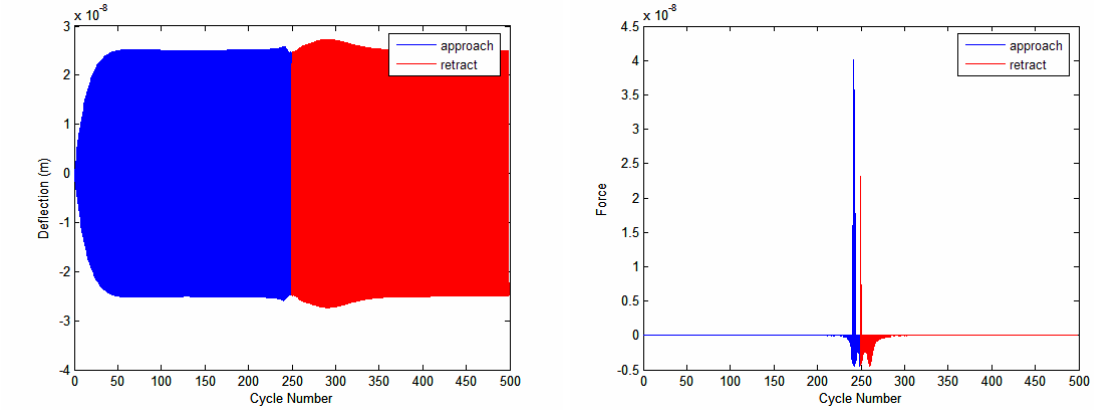
Click the **ResPeak** button. The following plot should appear:



Now, click the **Tune** botton. The first prompt will ask for the "% off resonance". This is to decide what percentage off resonance you wish to operate. Enter **90** and click **OK**. The second prompt will ask "below(1) or above (2)?". This is to determine if you wish to operate above or below resonance, i.e. enter a 1 for below or a 2 for above. Enter **1** and click **OK**. (Note: "% off resonance" is defined as the frequency that produces the free amplitude corresponding a percentage of the free amplitude at resonance. For example, if the free amplitude at resonance is 50 nm, 90% off resonance defines the frequency that produces an amplitude of 40 nm with the same drive amplitude) The following plot should appear:



The green '*' indicates where on the resonance peak you are operating. The **Fo** is automatically adjusted to maintain the value of **amplitude** that was entered in the first GUI. Click **RunSim** on the first GUI, and the following plots should appear:



Notice that the approach and retract or no longer plotted on top of each other. This is to make it easier to see them separately. Also, the x axis is scaled in Cycle Number, which is the number of cycles in the simulation (**Ncycles** * **Q**). If you want to run the sim again, you will need to press the **Set** button and then the **Tune** button (entering the same values) again. If you don't change any of the **Cantilever Parameters**: **Kspring**, **Fres**, **Q**, and **amplitude**, you will not need to rerun **ResPeak**. If you did change them, you will need to rerun **Respeak**, and depending on what you changed, you may need to adjust **SweepStart** and **SweepWidth**. If you change any parameter on the main screen, you will need to press **Set** before running a new simulation. If **amplitude** is zero, you do not need to tune the cantilever.