6000 Level Term Project
Biomedical Fluid Mechanics

Samuel Mayo
Submitted December 15, 2016

Questions:

1.) To produce plots of the input impedance, MATLAB was used due to its powerful numerical computing capabilities as well as its ease of use in terms of plotting and formatting. The first step was to represent all of the relevant parameters as a structure so that the user could quickly and easily manipulate said parameters using dot notation to see the effect that those parameters have on the final plots. Next, all of the constituent equations and expressions were represented as functions to allow for quick implementation of the final algorithms. All of the scripts, functions, and parameters are available in the appendix. Once all of the functions were created, they were assembled into several higher-order functions that received all of the parameters as well as the vector of frequencies to plot against. A vector of frequencies from 0 Hz to 10 Hz was created and passed into the higher-order function *ZvsF()*, which adds the input impedances from both vessels in parallel and produced the plots in Figure 1.

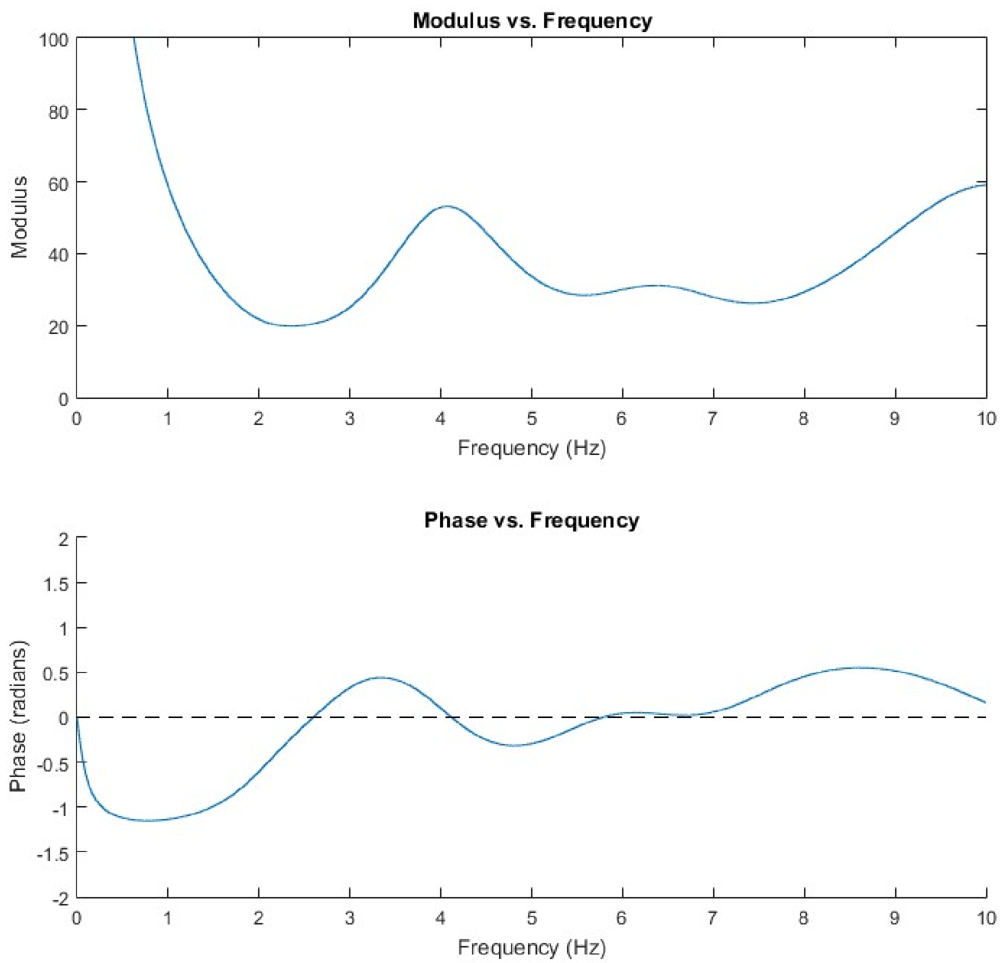

Figure 1: The modulus and phase of the input impedance vs. the frequency of flow oscillation.

This numerical model closely resembles the empirically derived plots provided in the problem statement. The plots provided were of the ascending aorta and the plots created in MATLAB were based off of a model of a bifurcation, so the small differences between the two results can be attributed to the difference in location and vessel geometry.

2.) The elastic modulus of the first tube was varied between 9e6 and 5e6 dyn/cm^2, the elastic modulus of the second tube was varied between 8e6 and 4e6 dyn/cm_2, and the elastic moduli of the two tubes were varied together between 9e6 and 5e6 dyn/cm^2, with the results plotted below. When the elastic modulus of the first tube was increased, it caused a spike in the impedance modulus and a shift to the right of the curve of the impedance phase as seen in Figure 2. This makes sense, since an increase in the stiffness of the vessel will increase the impedance, since the vessel wall is no longer elastic enough to handle the increased frequency of flow oscillation. Interestingly, when the elastic modulus of the second tube is increased, the impedance modulus smooths out but increases overall as seen in Figure 3. When the elastic moduli of both tubes are increased in tandem, the resulting impedance modulus curves resemble the addition of the curves from the independent elastic modulus variations, as shown in Figure 4. This is an interesting consequence of the model and should be followed up with empirical data.

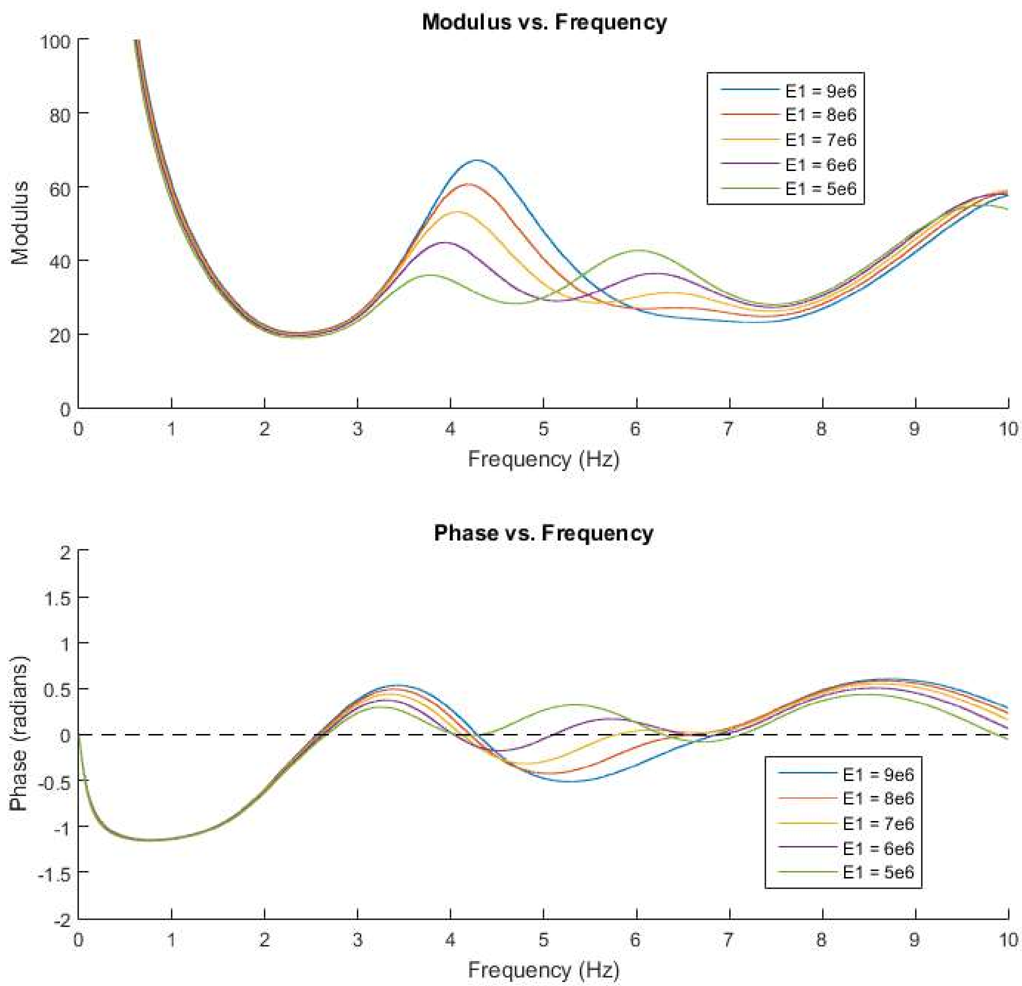

Figure 2: The modulus and phase of the input impedance with varying elastic modulus E1.

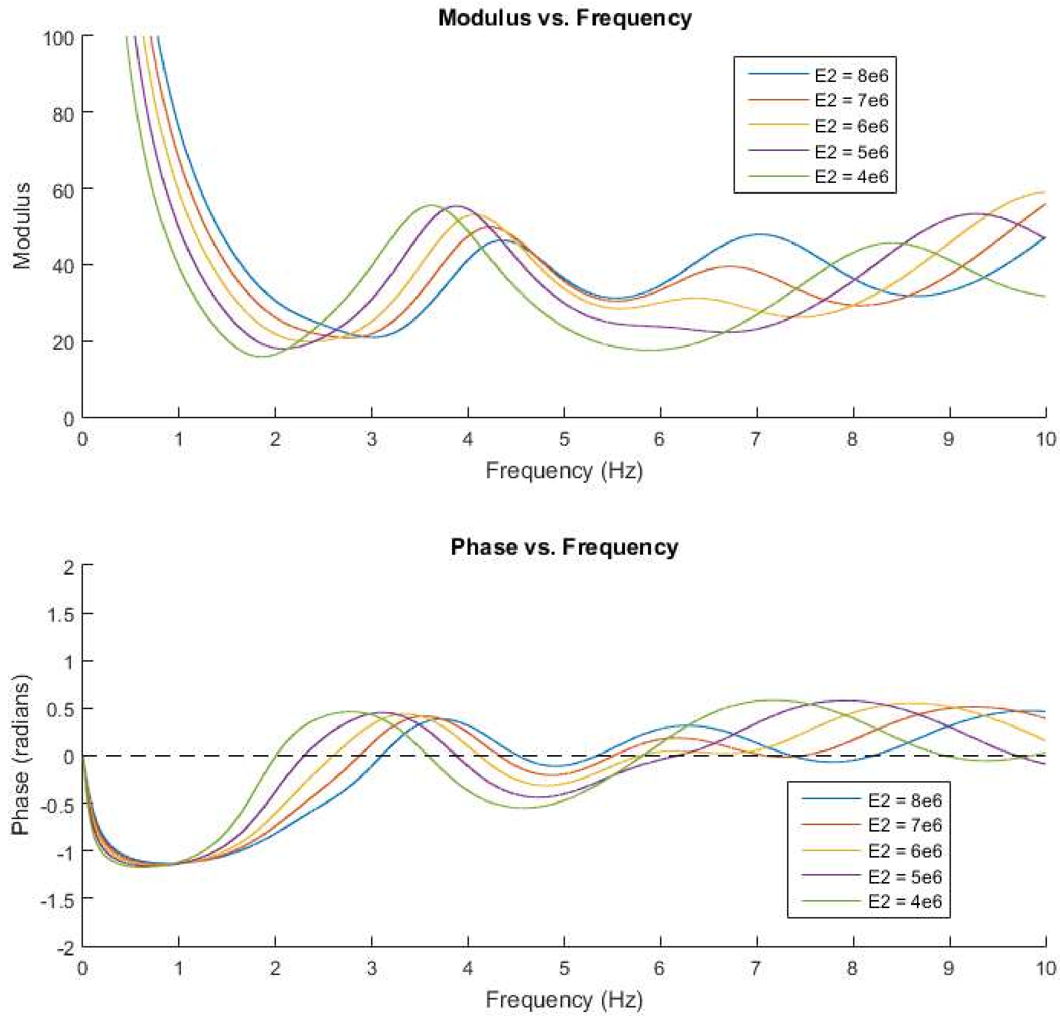

Figure 3: The modulus and phase of the input impedance with varying elastic modulus E2.

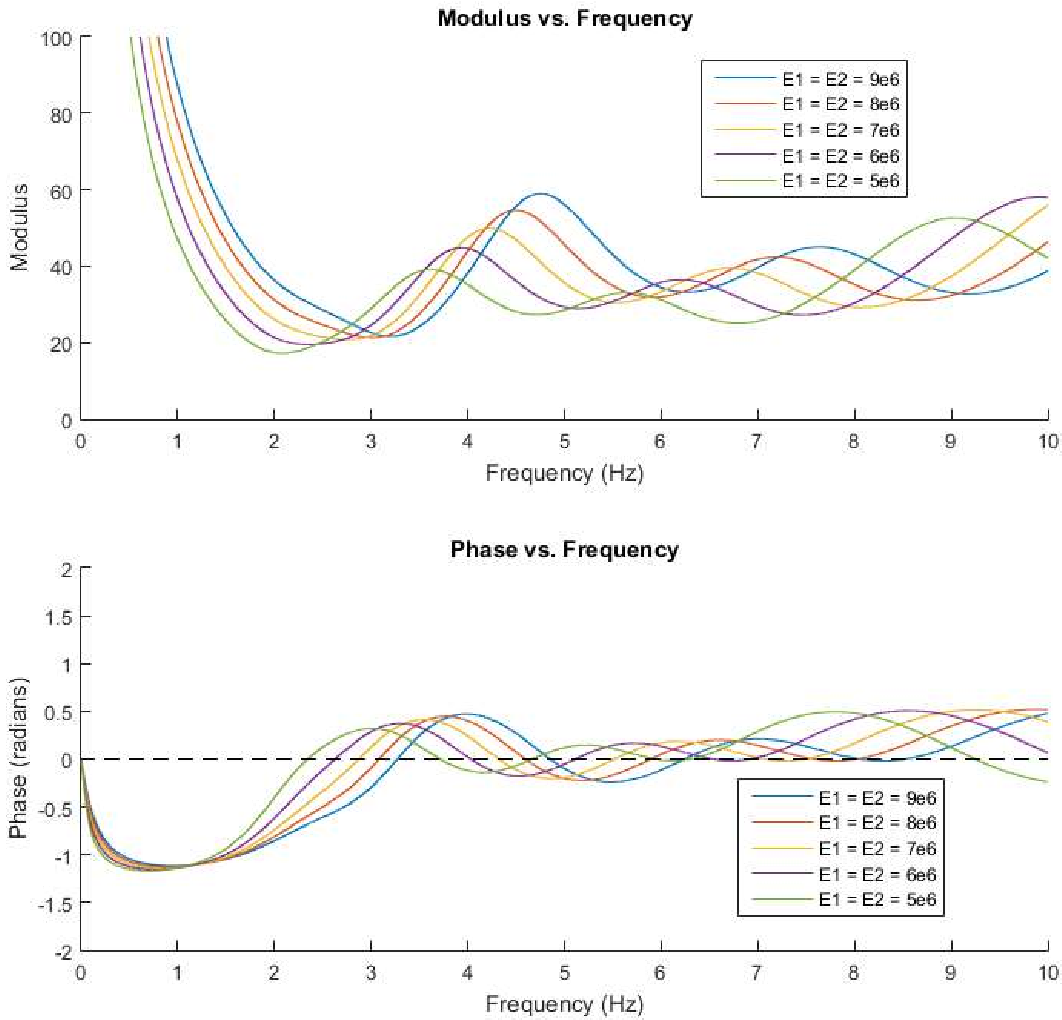

Figure 4: The modulus and phase of the input impedance with varying elastic moduli E1 and E2.

3.) The viscous component of the wall viscosity is represented by the phase angle *phi0* in the MATLAB model. When the vessel wall is purely elastic and without viscosity, this angle is 0° or 0 radians. When the vessel wall is very viscous, this angle approaches 180° or π radians. The input impedance was plotted using the normal angle of 12° and the experimental values of 0° and 180° with the resulting plots shown in Figure 5. When the vessel wall was modeled as purely elastic, this caused an increase in the spikes and troughs in both the impedance modulus and the impedance phase. With a purely elastic vessel wall, the differences in impedance between different frequencies would become greater, as is shown in the model. When the vessel wall was modeled as purely viscous, the impedance modulus and impedance phase both smoothed out to steady state values. With a purely viscous vessel wall, an increase in the frequency of flow oscillation would not produce an appreciable difference in the impedance, as is reflected in the model. It is also interesting to note that the impedance phase leveled out to around π radians, or 180°, which is maximum phase and corresponds to a large impedance phase, which is intuitive.

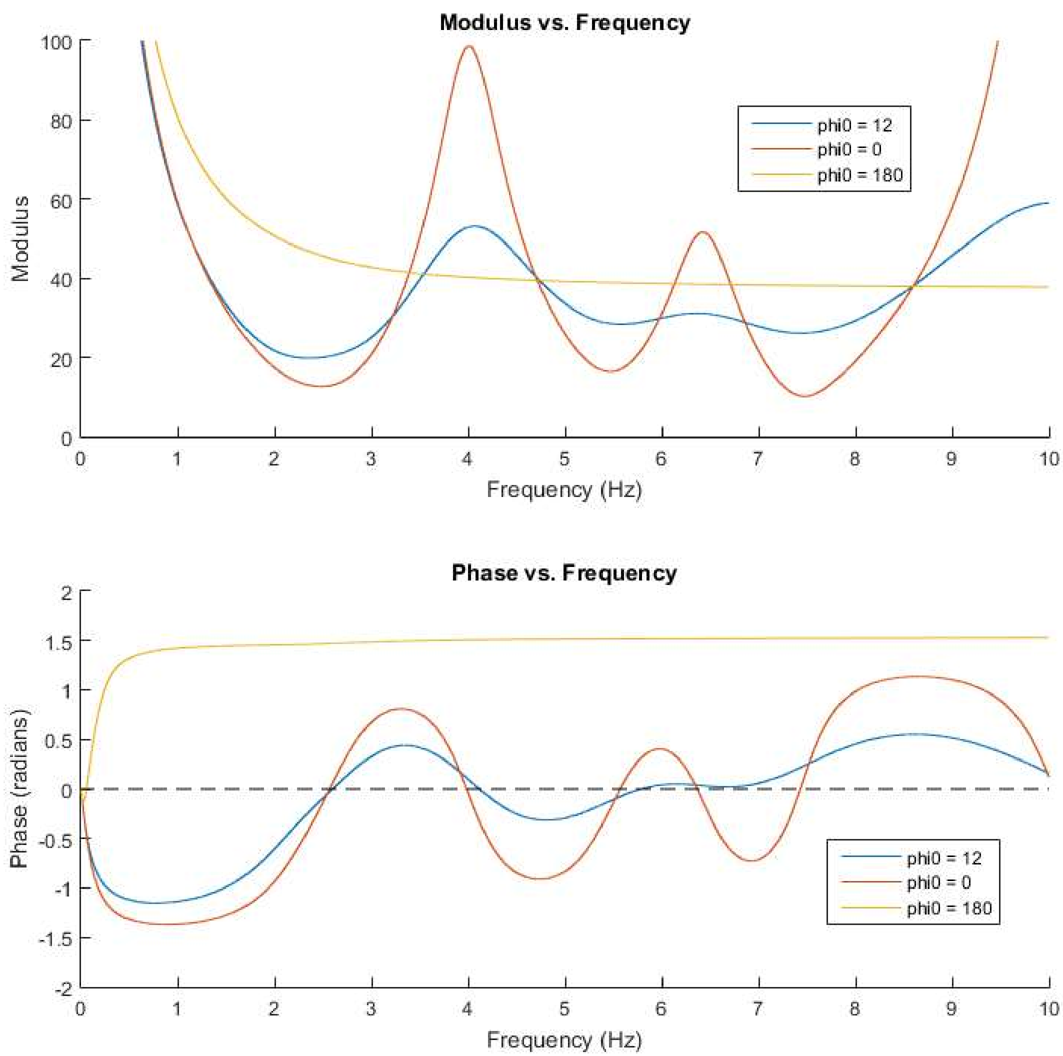

Figure 5: The modulus and phase of the input impedance with varying viscous component phi0.

4.) To model a hypertensive patient, the elastic moduli of both tubes were increased by a factor of two and the peripheral resistance was increased by 25%. The input impedance was plotted with these parameters and compared to the normal parameters in the plots shown in Figure 6. The hypertensive parameters induced a rightwards shift of both the impedance modulus and impedance phase curves and a spike in the impedance modulus. In a hypertensive patient, an increase in the input impedance is to be expected due to the increase in vessel wall stiffness and peripheral resistance, but this rightwards shift of the impedance curves is interesting. This would imply that the hypertensive patient's vessels would have their highest impedances at a different flow oscillation frequency than a normal patient. It follows that this shift in the response curve would alter the amount of impedance observed at a given physiologically relevant flow oscillation frequency and could mean that a hypertensive patient will experience a higher impedance even though their flow oscillation frequency remains unchanged.

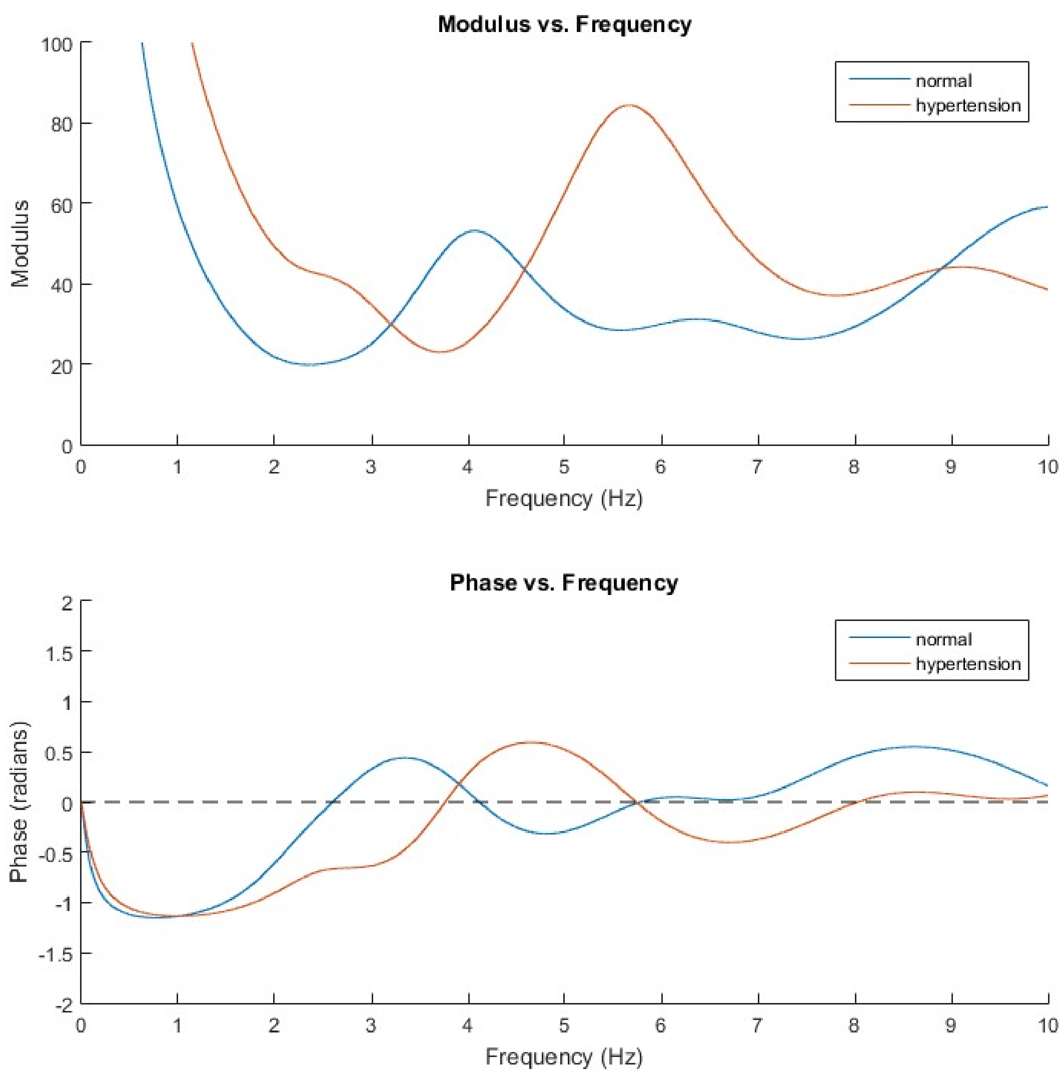

Figure 6: The modulus and phase of the input impedance of a normal patient compared with a hypertensive patient with increased elastic moduli E1 and E2 and peripheral resistance R.

5.) To plot the input impedance at the anatomical locations A (entrance), B (1/3 of the lower tube), C (2/3 of the lower tube), and D (terminal), only the second tube was modeled and the length parameter *l2* was varied between 38, 25.33, 12.67, and 0, respectively. The plots with those parameters are shown in Figure 7. For points A, B, and C, the model outputs were reflective of the plots provided in the problem statement, but the model output for point D was very different from the plot provided in the problem statement. This is an interesting limitation of the model and implies that this particular model is not valid for the terminal impedance. As the location moves down the second tube, the impedance modulus curve approaches a steady state and so does the impedance phase. This is similar to the steady state approach seen when the wall viscosity *phi0* was at maximum 180°. This implies that the farther down the tube, the less the flow oscillation contributes to the input impedance. At locations distal to the input of the flow, the energy from these oscillations will decay and will affect the impedance to a lesser degree and will approach the impedance of a purely viscous vessel wall. This is interesting, as it informs the design of vascular implants distal to the input location in terms of material properties and implant geometry, especially as those relate to their response to flow oscillations.

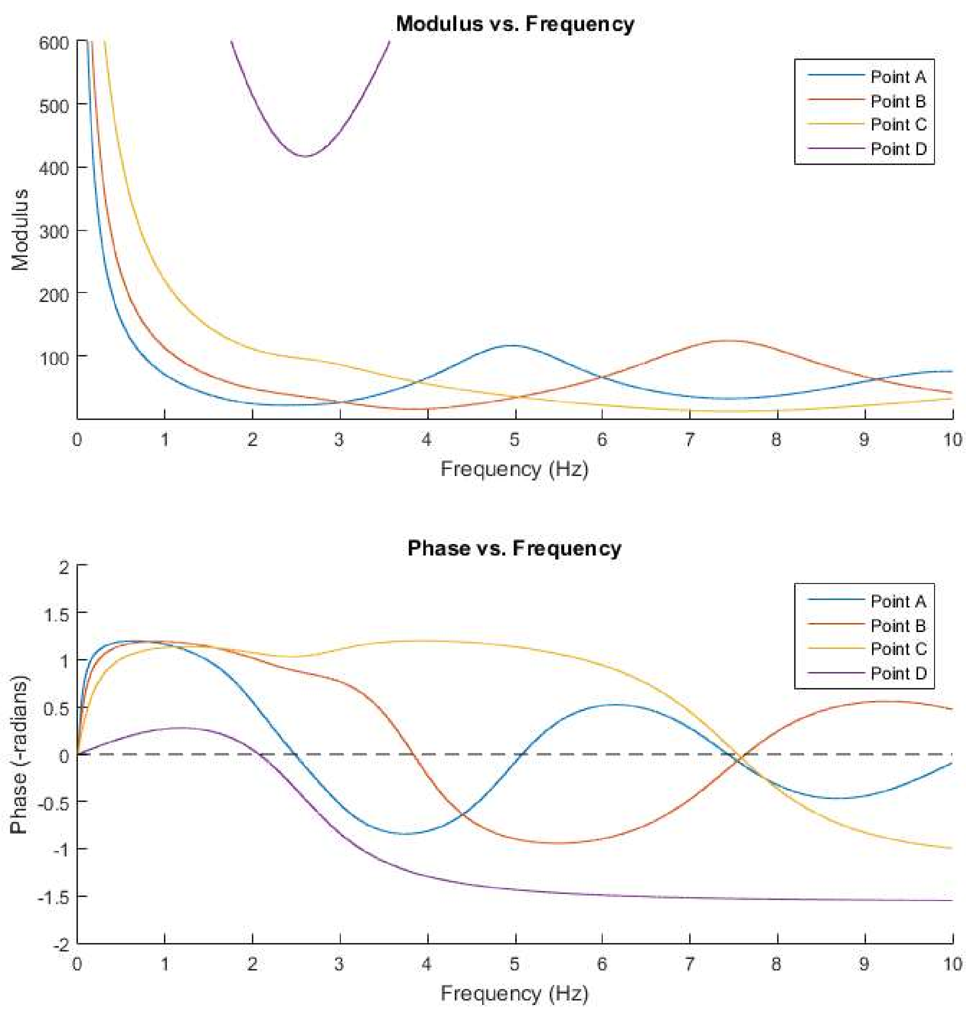

Figure 7: The modulus and phase of the input impedance of the second tube at locations increasingly distal from the input location.

Code Appendix:

Question 01

```
%% Input Impedance
f = 0:0.001:10;

parameters = load('question01parameters.mat');

[ZinCReal,ZinCImag] = ZvsF(f,parameters);

figure

subplot(2,1,1)
plot(f,ZinCReal)
ylim([0 100])
title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')

subplot(2,1,2)
hold on
plot(f,ZinCImag)
plot(f,zeros([1,length(f)]),'k--')
ylim([-2 2])
title('Phase vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Phase (radians)')
```

Question 02

```
f = 0:0.001:10;
parameters = load('question01parameters.mat');

%% Tube 1
E1tests = [ 9e6 8e6 7e6 6e6 5e6 ];
ZinCReal = cell(length(E1tests),1);
ZinCImag = cell(length(E1tests),1);
legendCell = cell(length(E1tests),1);

for k = 1:length(E1tests);
    parameters.E1 = E1tests(k);
    [ZinCReal{k},ZinCImag{k}] = ZvsF(f,parameters);
end

figure
subplot(2,1,1)
ylim([1 100])
hold on

for k = 1:length(E1tests);
    plot(f,ZinCReal{k})
    legendStrTemp = strcat(['E1 = ' sprintf('%.*e',1,E1tests(k)) ' ']);
    legendCell{k} = legendStrTemp;
end

title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')
legend(legendCell)

subplot(2,1,2)
ylim([-2 2])
hold on

for k = 1:length(E1tests);
    plot(f,ZinCImag{k})
end
plot(f,zeros([1,length(f)]),'k--')

title('Phase vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Phase (radians)')
legend(legendCell)

parameters.E1 = 7e6;

%% Tube 2
E2tests = [ 8e6 7e6 6e6 5e6 4e6 ];
ZinCReal = cell(length(E2tests),1);
ZinCImag = cell(length(E2tests),1);
legendCell = cell(length(E2tests),1);
```

```
for k = 1:length(E2tests);
    parameters.E2 = E2tests(k);
    [ZinCReal{k},ZinCImag{k}] = ZvsF(f,parameters);
end

figure
subplot(2,1,1)
ylim([1 100])
hold on

for k = 1:length(E2tests);
    plot(f,ZinCReal{k})
    legendStrTemp = strcat(['E2 = ' sprintf('%.*e',1,E2tests(k)) ' ']);
    legendCell{k} = legendStrTemp;
end

title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')
legend(legendCell)

subplot(2,1,2)
ylim([-2 2])
hold on

for k = 1:length(E2tests);
    plot(f,ZinCImag{k})
end
plot(f,zeros([1,length(f)]),'k--')

title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')
legend(legendCell)

parameters.E2 = 6e6;

%% Both Tubes
E1tests = [ 9e6 8e6 7e6 6e6 5e6 ];
E2tests = [ 9e6 8e6 7e6 6e6 5e6 ];
ZinCReal = cell(length(E1tests),1);
ZinCImag = cell(length(E1tests),1);
legendCell = cell(length(E1tests),1);

for k = 1:length(E1tests);
    parameters.E1 = E1tests(k);
    parameters.E2 = E2tests(k);
    [ZinCReal{k},ZinCImag{k}] = ZvsF(f,parameters);
end

figure
subplot(2,1,1)
ylim([1 100])
hold on
```

```
for k = 1:length(E1tests);
    plot(f,ZinCReal{k})
    legendStrTemp = strcat(['E1 = ' sprintf('%.*e',1,E1tests(k)) ...
    ', E2 = ' sprintf('%.*e',1,E2tests(k))]);
    legendCell{k} = legendStrTemp;
end

title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')
legend(legendCell)

subplot(2,1,2)
ylim([-2 2])
hold on

for k = 1:length(E1tests);
    plot(f,ZinCImag{k})
end
plot(f,zeros([1,length(f)]),'k--')

title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')
legend(legendCell)

parameters.E1 = 7e6;
parameters.E2 = 6e6;
```

Question 03

```
f = 0:0.001:10;
parameters = load('question01parameters.mat');

%% Changing phi0
parameters.phi0 = (12/360)*2*pi;
[ZinCReal01,ZinCImag01] = ZvsF(f,parameters);
parameters.phi0 = 0;
[ZinCReal02,ZinCImag02] = ZvsF(f,parameters);
parameters.phi0 = pi;
[ZinCReal03,ZinCImag03] = ZvsF(f,parameters);

figure
subplot(2,1,1)
hold on
plot(f,ZinCReal01)
plot(f,ZinCReal02)
plot(f,ZinCReal03)
ylim([0 100])
title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')
legend('phi0 = 12','phi0 = 0','phi0 = 180')

subplot(2,1,2)
hold on
plot(f,ZinCImag01)
plot(f,ZinCImag02)
plot(f,ZinCImag03)
plot(f,zeros([1,length(f)]),'k--')
ylim([-2 2])
title('Phase vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Phase (radians)')
legend('phi0 = 12','phi0 = 0','phi0 = 180')
```

Question 04

```
f = 0:0.001:10;
parameters = load('question01parameters.mat');

%% Changing phi0
parameters.E1 = 7e6;
parameters.E2 = 6e6;
parameters.R = 12e3;
[ZinCReal01,ZinCImag01] = ZvsF(f,parameters);
parameters.E1 = 14e6;
parameters.E2 = 12e6;
parameters.R = 16e3;
[ZinCReal02,ZinCImag02] = ZvsF(f,parameters);

figure
subplot(2,1,1)
hold on
plot(f,ZinCReal01)
plot(f,ZinCReal02)
ylim([0 100])
title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')
legend('normal','hypertension')

subplot(2,1,2)
hold on
plot(f,ZinCImag01)
plot(f,ZinCImag02)
plot(f,zeros([1,length(f)]),'k--')
ylim([-2 2])
title('Phase vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Phase (radians)')
legend('normal','hypertension')
```

Question 05

```
f = 0:0.001:10;
parameters = load('question01parameters.mat');

%% Modeling Different Locations on Tube 2
l2tests = [38,25.33,12.67,0];
Zin2Real = cell(length(l2tests),1);
Zin2Imag = cell(length(l2tests),1);

for k = 1:length(l2tests);
    parameters.l2 = l2tests(k);
    [Zin2Real{k},Zin2Imag{k}] = ZvsFtube2(f,parameters);
end

figure
subplot(2,1,1)
ylim([1 600])
hold on

for k = 1:length(l2tests);
    plot(f,Zin2Real{k})
end

title('Modulus vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Modulus')
legend('Point A','Point B','Point C','Point D')

subplot(2,1,2)
ylim([-2 2])
hold on

for k = 1:length(l2tests);
    plot(f,-Zin2Imag{k})
end
plot(f,zeros([1,length(f)]),'k--')

title('Phase vs. Frequency')
xlabel('Frequency (Hz)')
ylabel('Phase (-radians)')
legend('Point A','Point B','Point C','Point D')
```

Function and Parameter List

angFreq.m
```
function [ omega ] = angFreq(f)

omega = 2*pi*f;

end
```

charImpedance.m
```
function [ Zc ] = charImpedance(rho,C0,r,sigma,F10,phi)
j = sqrt(-1);

num = rho*C0;
den = pi*r^2*sqrt(1-sigma^2);
coeff = (1-F10)^(-1/2)*exp(j*phi/2);

Zc = (num/den)*coeff;

end
```

combinedInputImpedance.m
```
function [ ZinC ] = combinedInputImpedance(Zin1,Zin2)

ZinC = 1/((1/Zin1)+(1/Zin2));

end
```

function_F10.m
```
function [ F10 ] = function_F10( alpha )
%FUNCTION_F10 Summary of this function goes here
%   Detailed explanation goes here
j = sqrt(-1);

J0 = besselj(0,alpha*(j)^(3/2));
J1 = besselj(1,alpha*(j)^(3/2));

F10 = (2*J1)/((alpha*(j)^(3/2))*J0);

end
```

inputImpedance.m
```
function [ Zin ] = inputImpedance(Zc,bigGamma,littleGamma,l)

num = 1 + bigGamma*exp(-2*littleGamma*l);
den = 1 - bigGamma*exp(-2*littleGamma*l);

Zin = Zc*(num/den);

end
```

inputImpedanceBig.m

```
function [ Zin ] =
inputImpedanceBig(omega,rho,mu,E,h,r,l,sigma,phi0,k,R,L,C,Rc)
%%INPUTIMPEDANCEBIG receives all relevant constants and variables and
%%outputs the input impedance of a vessel
j = sqrt(-1);

phi = wallVisc(phi0,k,omega);
C0 = moensKorteweg(E,h,rho,r);
Zt = terminalImpedance(Rc,omega,L,R,C);
alpha = womersley(r,rho,omega,mu);
F10 = function_F10(alpha);
Zc = charImpedance(rho,C0,r,sigma,F10,phi);
littleGamma = propCoeff(omega,C0,sigma,F10,phi);
bigGamma = termRefCoeff(Zt,Zc);

Zin = inputImpedance(Zc,bigGamma,littleGamma,l);

end
```

moensKorteweg.m

```
function [ C0 ] = moensKorteweg(E,h,rho,r)
%MOENSKORTEWEG outputs c0
%
C0 = sqrt((E*h)/(2*rho*r));

end
```

propCoeff.m

```
function [ littleGamma ] = propCoeff(omega,C0,sigma,F10,phi)
j = sqrt(-1);

a = j*omega/C0;
b = (1-sigma^2)/(1-F10);

littleGamma = a*sqrt(b)*exp(-j*phi/2);

end
```

terminalImpedance.m

```
function [ Zt ] = terminalImpedance(Rc,omega,L,R,C)
j = sqrt(-1);

Zt = Rc + j*omega*L + (R/(1 + j*omega*R*C));

end
```

termRefCoeff.m
```
function [ bigGamma ] = termRefCoeff(Zt,Zc)

num = Zt - Zc;
den = Zt + Zc;

bigGamma = num/den;

end
```

wallVisc.m
```
function [ phi ] = wallVisc(phi0,k,omega)

phi = phi0*(1-exp(-k*omega));

end
```

womersley.m
```
function [ alpha ] = womersley(r,rho,omega,mu)

alpha = r*sqrt((rho*omega)/(mu));

end
```

ZvsF.m
```
function [ ZinCReal,ZinCImag ] = ZvsF( f,param )
%ZVSF receives a vector of frequencies and a parameters structure and
%outputs the real and imaginary components of the impedances.
%   Detailed explanation goes here

scaleFactor = 0.1;
omega = angFreq(f);
len_om = length(omega);
i = 1;
Zin1 = zeros(0,len_om);
Zin2 = zeros(0,len_om);
ZinC = zeros(0,len_om);
ZinCReal = zeros(0,len_om);
ZinCImag = zeros(0,len_om);

while i < (len_om+1)
    Zin1(i) = inputImpedanceBig(omega(i),param.rho,param.visc,param.E1, ...
    param.h1,param.r1,param.l1,param.sigma,param.phi0,param.k,param.R, ...
    param.L,param.C1,param.Rc1);
    Zin2(i) = inputImpedanceBig(omega(i),param.rho,param.visc,param.E2, ...
    param.h2,param.r2,param.l2,param.sigma,param.phi0,param.k,param.R, ...
    param.L,param.C2,param.Rc2);
    ZinC(i) = combinedInputImpedance(Zin1(i),Zin2(i));
    ZinCReal(i) = abs(ZinC(i))*scaleFactor;
    ZinCImag(i) = angle(ZinC(i));
    i = i + 1;
end

end
```

ZvsFtube2.m
```
function [ ZinCReal,ZinCImag ] = ZvsFtube2( f,param )
%ZVSFTUBE2 receives a vector of frequencies and a parameters structure and
%outputs the real and imaginary components of the impedances.
%   Detailed explanation goes here

scaleFactor = 0.1;
omega = angFreq(f);
len_om = length(omega);
i = 1;
Zin2 = zeros(0,len_om);
ZinCReal = zeros(0,len_om);
ZinCImag = zeros(0,len_om);

while i < (len_om+1)
    Zin2(i) = inputImpedanceBig(omega(i),param.rho,param.visc,param.E2, ...
    param.h2,param.r2,param.l2,param.sigma,param.phi0,param.k,param.R, ...
    param.L,param.C2,param.Rc2);
    ZinCReal(i) = abs(Zin2(i))*scaleFactor;
    ZinCImag(i) = angle(Zin2(i));
    i = i + 1;
end
end
```

```
parameters =

        j: 0.0000 + 1.0000i
      rho: 1.0500
       E1: 7000000
       E2: 6000000
       h1: 0.0150
       h2: 0.0200
       r1: 0.3500
       r2: 0.5000
       l1: 20
       l2: 38
    sigma: 0.5000
     phi0: 0.2094
        k: 2
        R: 12000
        L: 450
       C1: 5.0000e-06
       C2: 8.0000e-06
      Rc1: 260
      Rc2: 240
     visc: 0.0400
```