ROF – 005

```
%Starting Code

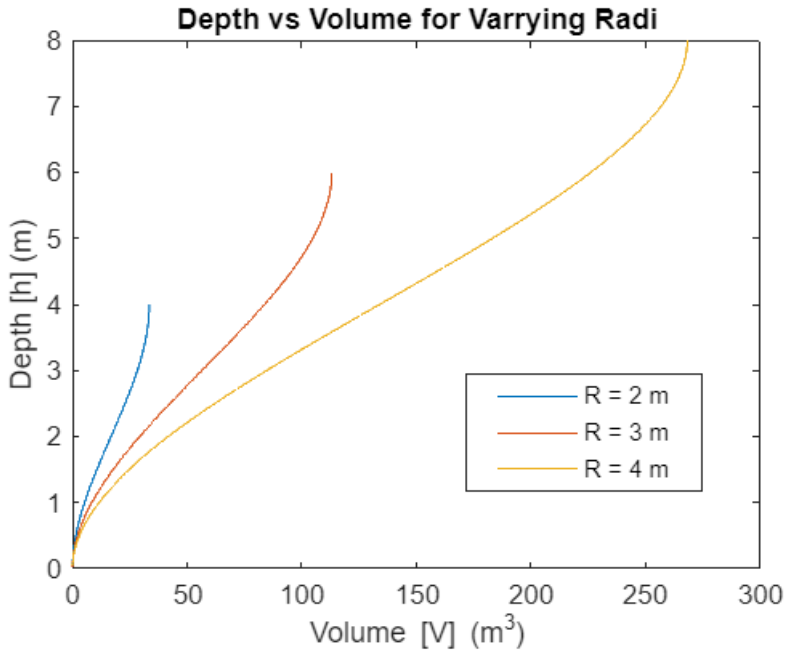
clear
close all
clc
%Part a
%Initial guess must be greater than 0 and less than the h max
%Since the water height must be within the container h max = D
%D in this case represents diameter or 2r
%Governing Equn
% V = pih^2(3r -h)/3
%Define Given Variables
V = 30;
r = 3;
%Using bisection evaluate for h
function h = bis(r, V, tolerance)
  h_lower_bound = 0;
  h_higher_bound = 2 * r;
  volume_f = @(h) (pi * h^2 * (3*r - h) / 3) - V;


  while (h_higher_bound - h_lower_bound) / 2 > tolerance
    h_intermediate_est = (h_lower_bound + h_higher_bound) / 2;
    f_mid = volume_f(h_intermediate_est);
    if volume_f(h_lower_bound) * f_mid < 0
      h_higher_bound = h_intermediate_est;
    else
      h_lower_bound = h_intermediate_est;
    end
  end
  h = (h_lower_bound + h_higher_bound) / 2;
end
tolerance = 0.0001;
h = bis(r,V,tolerance);
fprintf('Answer for part b displayed below: \n h = %g',h)
%Iterate with a vector value of r as defined in part c
R = [2, 3, 4];
%Specify the number of v points
num_V_points = 1/tolerance;
figure;
% Loop through each R value and calculate h for the range of V values.
for i = 1:length(R)
% Calculate maximum possible volume.
max_V = (4/3)*pi*(R(i))^3;
V_values = linspace(0, max_V, num_V_points);
%Improve running speed by preallocating
  h_values = zeros(1, num_V_points);
```

```
  for j = 1:num_V_points
    h_values(j) = bis(R(i), V_values(j), tolerance);
  end
  % Plot h vs V for the current R value
%Do not plot the first point as it causes issues with R = 4
  plot(V_values(2:end), h_values(2:end),LineStyle="-");
  hold on;
end
hold off;
xlabel('Volume [V] (m^3)');
ylabel('Depth [h] (m)');
title('Depth vs Volume for Varrying Radi');
legend(["R = 2 m", "R = 3 m", "R = 4 m"], "Position", [0.5716 0.2461 0.2680,
0.1791])
%Part d
%The depth is observed to increase with volume (as expected) with
%asymptotes at the V = 0 and V = Max values. This existance of a max V
%value is also what creates the lack of data for beyone V(max) on R =2 and
%R=3 lines. This same lack of info would be observed on R=4 if the graph
```

```
%bounds were different.
```

```
Answer for part b displayed below:
 h = 2.02689
```



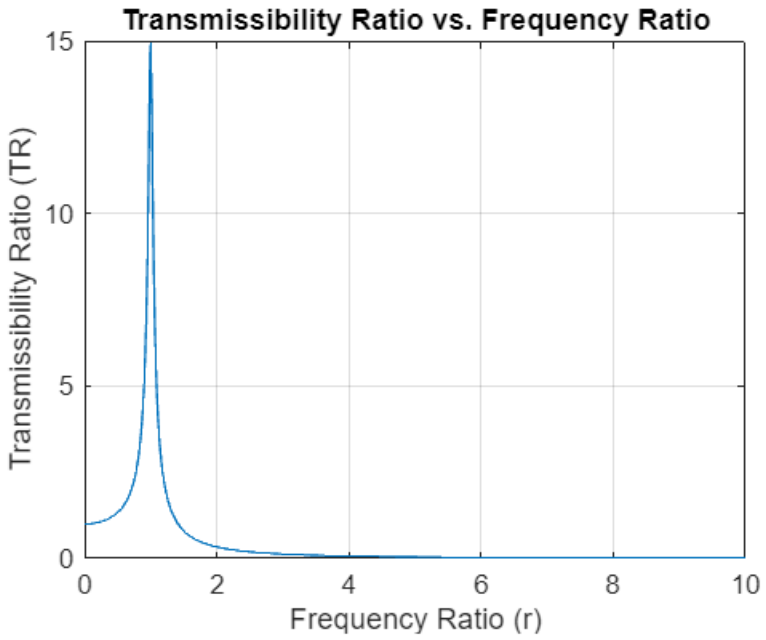
ROF – 006

```
%Starting Code

clear
close all
clc
% Given values
m = 100;
c = 30;
k = 2000;
% Calculate damping ratio
zeta = c / (2 * sqrt(k * m));
% Display the result
fprintf('Answer to part A Below: \nDamping ratio (ζ)= %g', zeta);
```

```
%Part B
fprintf('Answer to part B Below: \nζ & r are dimensionless')
%Part C
% set r range
r = linspace(0, 10, 1000);
% Calculate TR
TR = sqrt((1 + (2 * zeta * r).^2) ./ ((1 - r.^2).^2 + (2 * zeta * r).^2));
% Plot TR vs r
figure;
plot(r, TR);
xlabel('Frequency Ratio (r)');
ylabel('Transmissibility Ratio (TR)');
title('Transmissibility Ratio vs. Frequency Ratio');
grid on;
%Part D
X = 0.015;
Y = 0.02;
Value_needed = X/Y;
%Find r with quadratic formula
a = Value_needed^2;
b = 4*Value_needed^2*zeta^2 - 2*Value_needed^2 - 4*zeta^2;
c = Value_needed^2 - 1;
r_new_pos = (-b + sqrt(b^2 -4*a*c))/(2*a);
r_new_neg = (-b - sqrt(b^2 -4*a*c))/(2*a);
%note that r = w/wn wn = sqrt k/m
wn = sqrt(k/m);
w_min_rpos = r_new_pos*wn;
w_min_rneg = r_new_neg*wn;
fprintf('Answer Below for part D: \nAssuming that negative w values are allwoed\n
w_min = %g\nIf negative values are invalid\n w_min = %g',w_min_rneg,w_min_rpos)
```

```
Answer to part A Below:
Damping ratio (ζ)= 0.033541
Answer to part B Below:
ζ & r are dimensionless
```



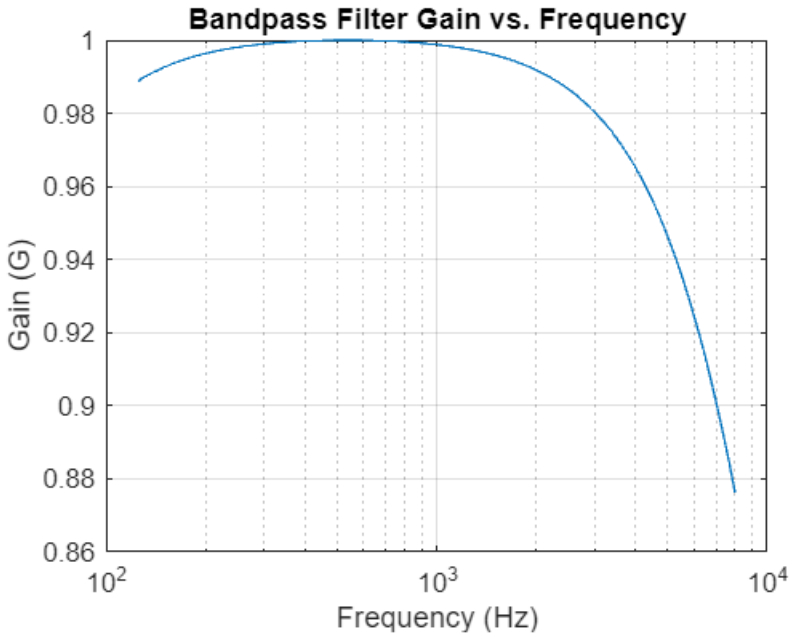

```
Answer Below for part D:
Assuming that negative w values are allwoed
 w_min = -1.48876
If negative values are invalid
 w_min = 10.4487
```

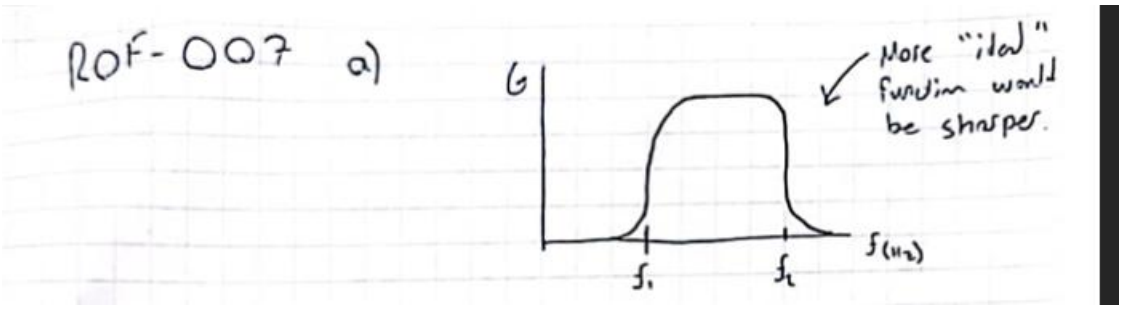
ROF – 007

```
%Starting Code
clear
close all
clc
%Part B


% Given values
C = 8e-6;
L = 11e-3;
R = 1000;
% Frequency range (logarithmic scale)
f = logspace(log10(125), log10(8000), 400)
% Gain function
function G = gain(f, R, L, C)
    G = abs(R*2*pi*f*C ./ sqrt((1 - (2*pi*f).^2 * L * C).^2 + (2*pi*f * R *
C).^2));
```

```
end
%Part B
G = gain(f, R, L, C);
%Plot
figure;
semilogx(f, G);
xlabel('Frequency (Hz)');
ylabel('Gain (G)');
title('Bandpass Filter Gain vs. Frequency');
grid on;
%Part C
%Estimating from looks the f1 value appears to be at aproximatly 125 and
%the f2 value looks to be at aproximatly 2000
[gmaxvalue gmaxidx] = max(G);
gmax_frequency = f(gmaxidx);
gtarget = gmaxvalue*(1/sqrt(2))
f1 = find(G <= gtarget,1, 'first')
f2 = find(G <= gtarget,1, 'last')
%Why is this not working
```





b) The function graphed in matlab and shown above is far more curved than the one I sketched and as such represents a less idealized more real-world realistic version of the graph sketched above in ROF – 007 a. Continuing analysis will be put in comments in the code which will be seen above with screenshots of outputs found below this comment.

