Augustus Crosby
Lab Wednesday 10:45
Boerner

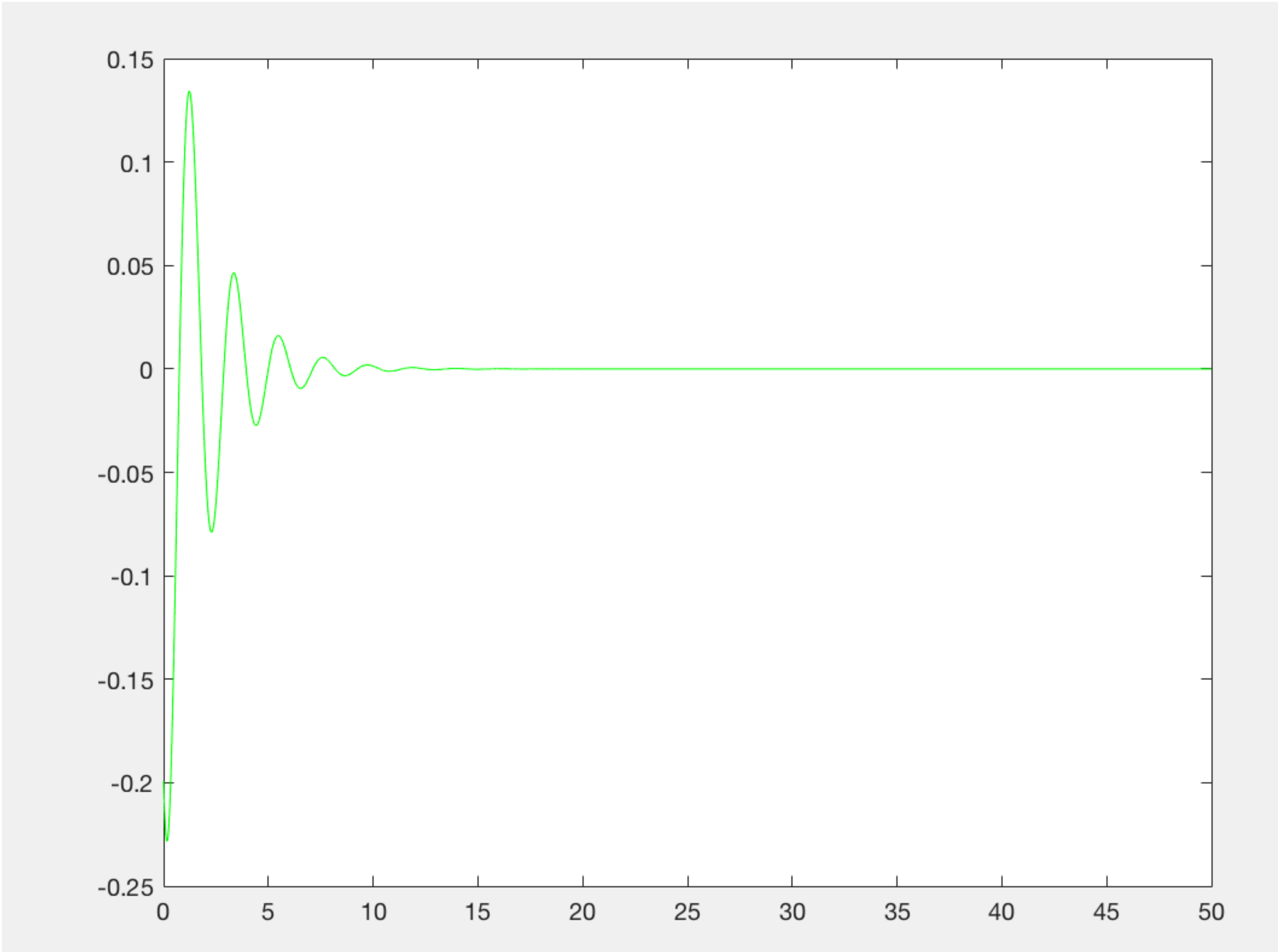
1) a) The period of the forced oscillation is 2.618. We find this by 2*pi/omega in the LAB06ex1.m file. It's clear that when we graph the function the period is about 2.6 also. The numerical value of the angle alpha in L6.4 appears to be the arctan(c(omega)/((omega_o)^2-(omega)^2)) which is .6375. Modulation of that is the same value, .6375.

b)
```
function LAB06ex1
clc
omega0 = 3; c = 1; omega = 2.4;
param = [omega0,c,omega];
t0 = 0; y0 = 0; v0 = 0; Y0 = [y0;v0]; tf = 50;
options = odeset('AbsTol',1e-10,'relTol',1e-10);
[t,Y] = ode45(@f,[t0,tf],Y0,options,param);
y = Y(:,1); v = Y(:,2);
figure(1)
plot(t,y,'b-'); ylabel('y'); grid on;
t1 = 25; i = find(t>t1);
C = (max(Y(i,1))-min(Y(i,1)))/2;
disp(['computed amplitude of forced oscillation = ' num2str(C)]);
Ctheory = 1/sqrt((omega0^2-omega^2)^2+(c*omega)^2);
disp(['theoretical amplitude = ' num2str(Ctheory)]);
if omega0 > omega
    alpha = atan(c*omega/((omega0)^2-(omega)^2))
else
    alpha = pi+ atan(c*omega/((omega0)^2-(omega)^2))
end
T=2*pi/omega
hold on
figure(2);
yc=y-(Ctheory*cos(omega*t-alpha));
plot(t,yc,'g-')
end

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

function dYdt = f(t,Y,param)
y = Y(1); v = Y(2);
omega0 = param(1); c = param(2); omega = param(3);
dYdt = [ v ; cos(omega*t)-omega0^2*y-c*v ];
end
```

This does look like an exponentially decreasing function in this case. I believe this is because the particular solution is driving the equation and that the function eventually goes to zero.



2) a)

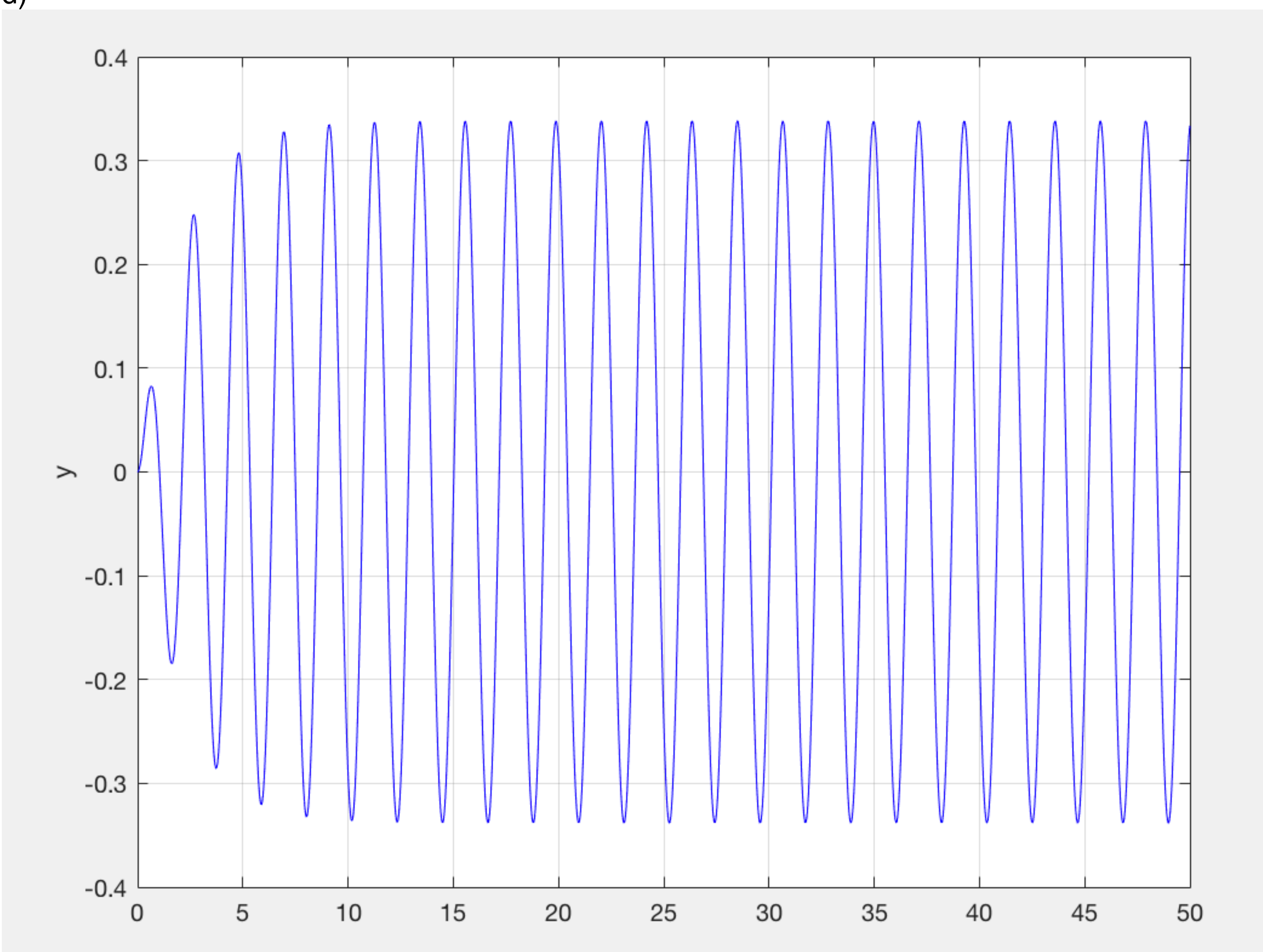
```
function LAB06ex2
omega0 = 3; c = 1;
OMEGA = 2:0.02:4;
C = zeros(size(OMEGA));
Ctheory = zeros(size(OMEGA));
t0 = 0; y0 = 0; v0 = 0; Y0 = [y0;v0]; tf = 50; t1 = 25;
for k = 1:length(OMEGA)
    omega = OMEGA(k);
    param = [omega0,c,omega];
    [t,Y] = ode45(@f,[t0,tf],Y0,[],param);
    i = find(t>t1);
    C(k) = (max(Y(i,1))-min(Y(i,1)))/2;
    Ctheory(k) =  1/sqrt((omega0^2-omega^2)^2+(c*omega)^2);
% FILL-IN
end
figure(2)
plot(OMEGA, C, 'b-', OMEGA, Ctheory, 'ro');
grid on;
```

```
xlabel('\omega'); ylabel('C');
%--------------------------------------------------------
function dYdt = f(t,Y,param)
y = Y(1); v = Y(2);
omega0 = param(1); c = param(2); omega = param(3);
dYdt = [ v ; cos(omega*t)-omega0^2*y-c*v ];
```

b)
It looks like the max is about at omega = 2.9, and C is about =.335.

c)
I took the derivative of C with respect to omega and set that equal to zero.
The equation ended up being (17w-2w^3)/(w^4-17w^2+81)^(3/2)=0
I got that omega = 2.915

d)

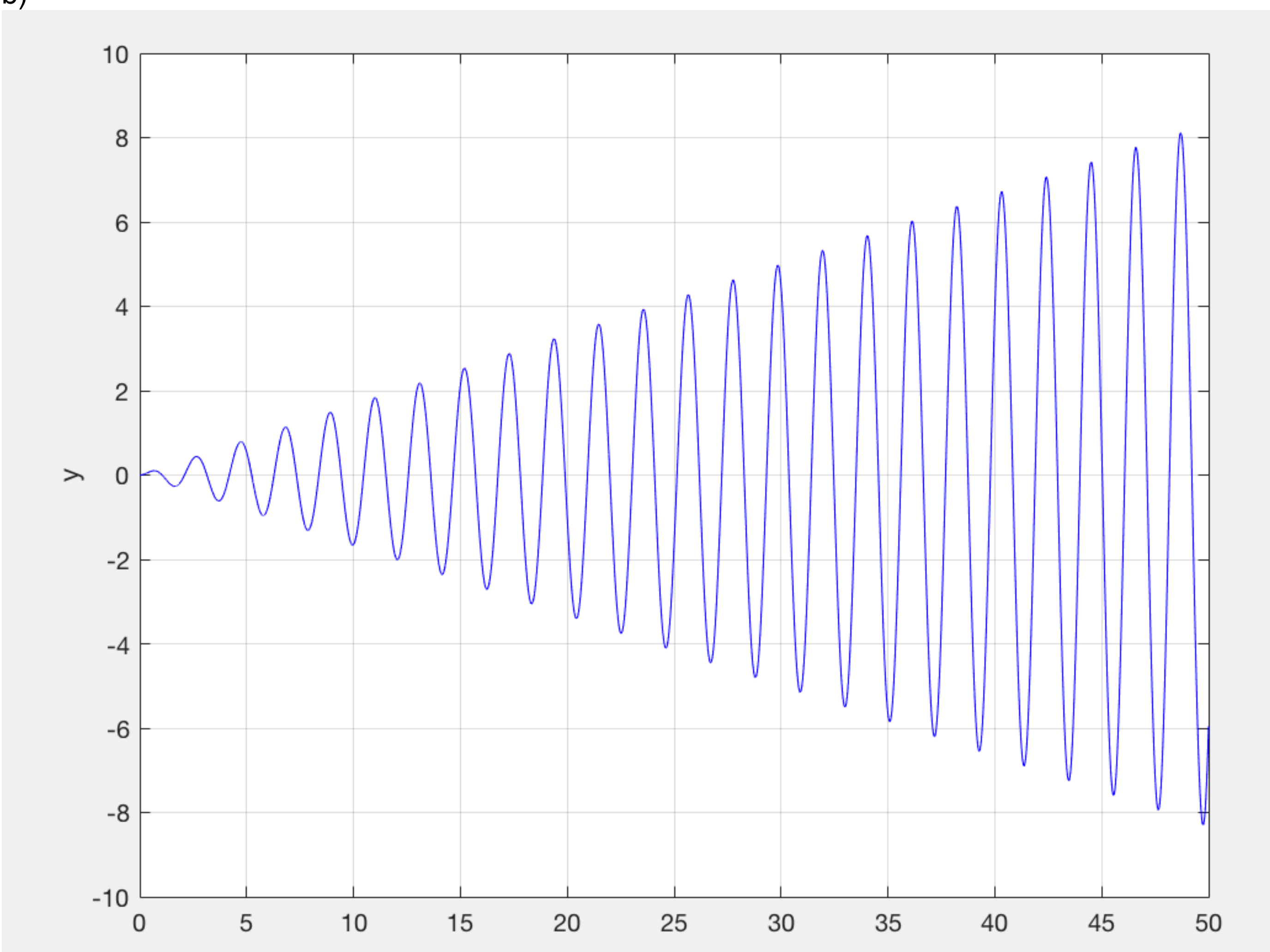


The amplitude of this graph is greater than in the last value of omega. This is expected because this is the highest value that omega can cause, as we just found that. In other words, any other value of omega will give a smaller amplitude than the one found here.

e) These results would not change based on initial conditions because y0 and v0 aren't found in the equations. I can confirm this by changing the y0 and v0 values in either example. These results are only changed if omega, omega0 or c were changed.

3) a) With c = 0, the max amplitude of C appears to be 8.2 or so. The omega that has this value is 3, and it is equal to omega0. This of course makes sense when looking at the equation.

b)



It appears that the amplitude does not stay constant and will never reach a max. The theoretical amplitude was infinity.

4) a)

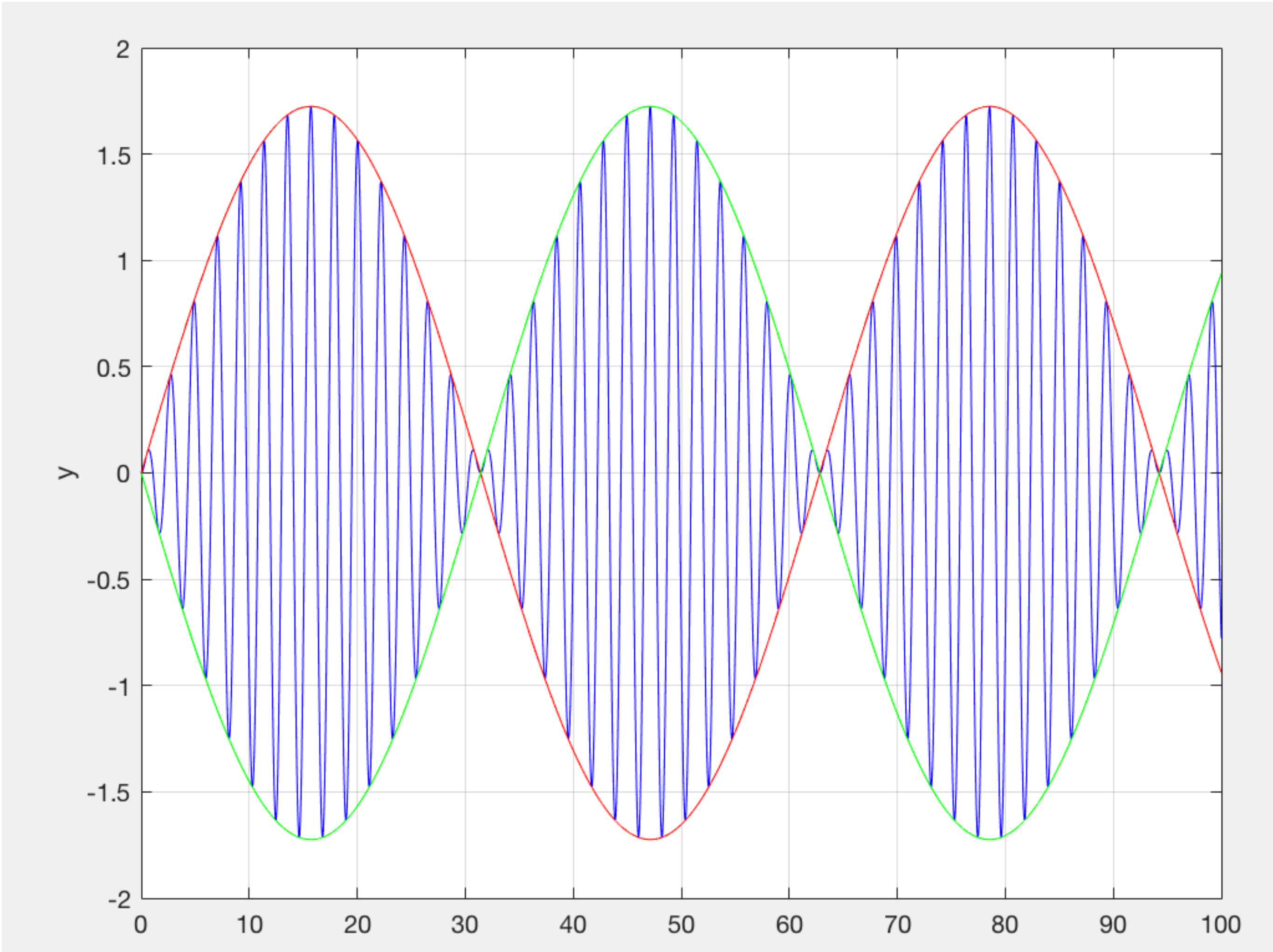
```
function LAB06ex1
clc
omega0 = 3; c = 0; omega = 2.8;
param = [omega0,c,omega];
t0 = 0; y0 = 0; v0 = 0; Y0 = [y0;v0]; tf = 100;
options = odeset('AbsTol',1e-10,'relTol',1e-10);
[t,Y] = ode45(@f,[t0,tf],Y0,options,param);
y = Y(:,1); v = Y(:,2);
figure(1)
C = 1/abs((omega0)^2-omega^2);
A = 2*C*sin(.5*(omega0-omega)*t);
plot(t,y,'b-',t,A,'r-',t,-A,'g-'); ylabel('y'); grid on;
t1 = 25; i = find(t>t1);
disp(['computed amplitude of forced oscillation = ' num2str(C)]);
Ctheory = 1/sqrt((omega0^2-omega^2)^2+(c*omega)^2);
disp(['theoretical amplitude = ' num2str(Ctheory)]);
if omega0 > omega
    alpha = atan(c*omega/((omega0)^2-(omega)^2))
else
    alpha = pi+ atan(c*omega/((omega0)^2-(omega)^2))
end
T=2*pi/omega
end

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

function dYdt = f(t,Y,param)
y = Y(1); v = Y(2);
omega0 = param(1); c = param(2); omega = param(3);
dYdt = [ v ; cos(omega*t)-omega0^2*y-c*v ];
end
```

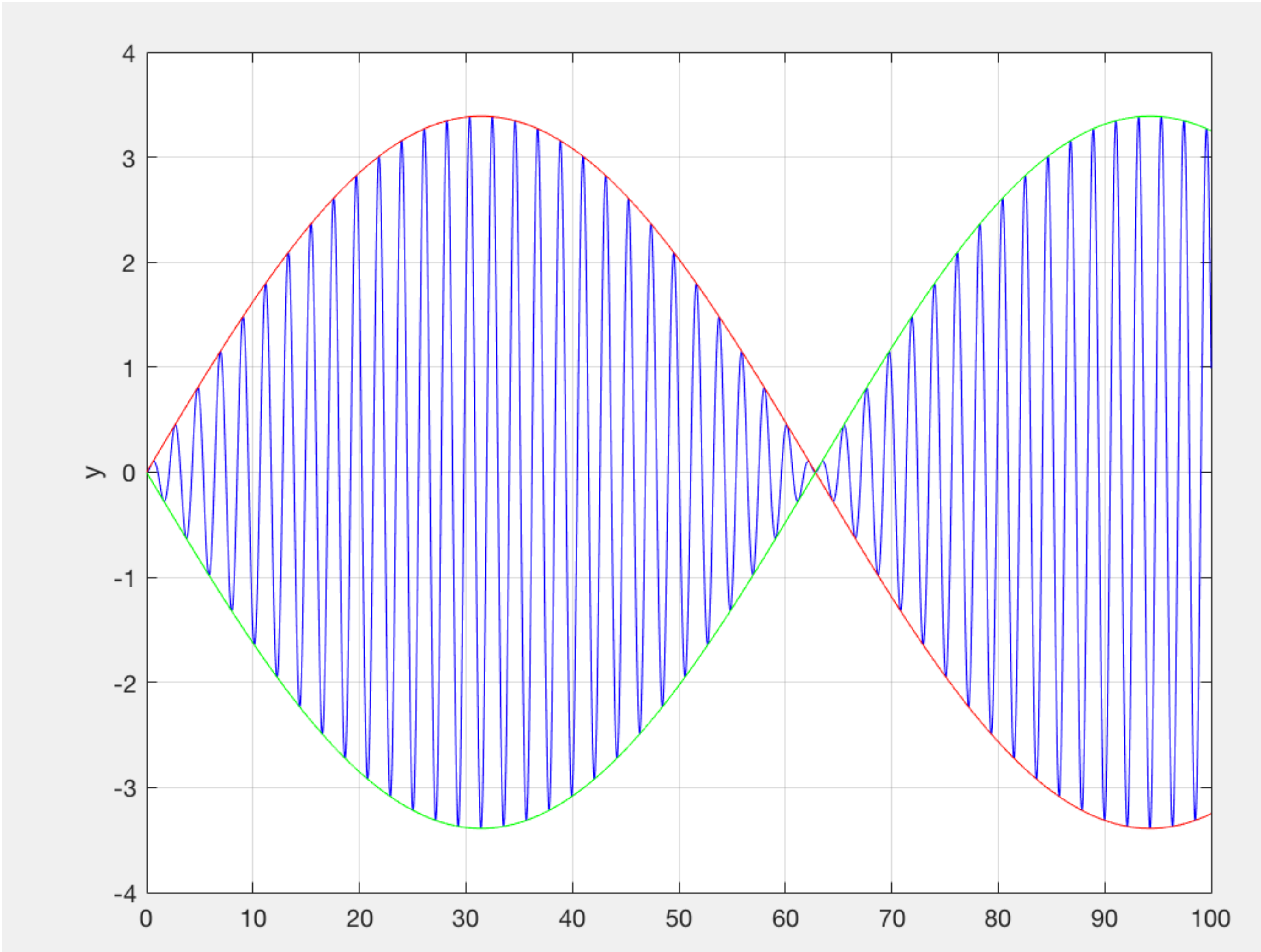
b)

I believe the period would be equal to T= 2*pi/(.5*(omega0+omega)) = 2.1666. Looking at the graph, 2.166 appears to be the period.



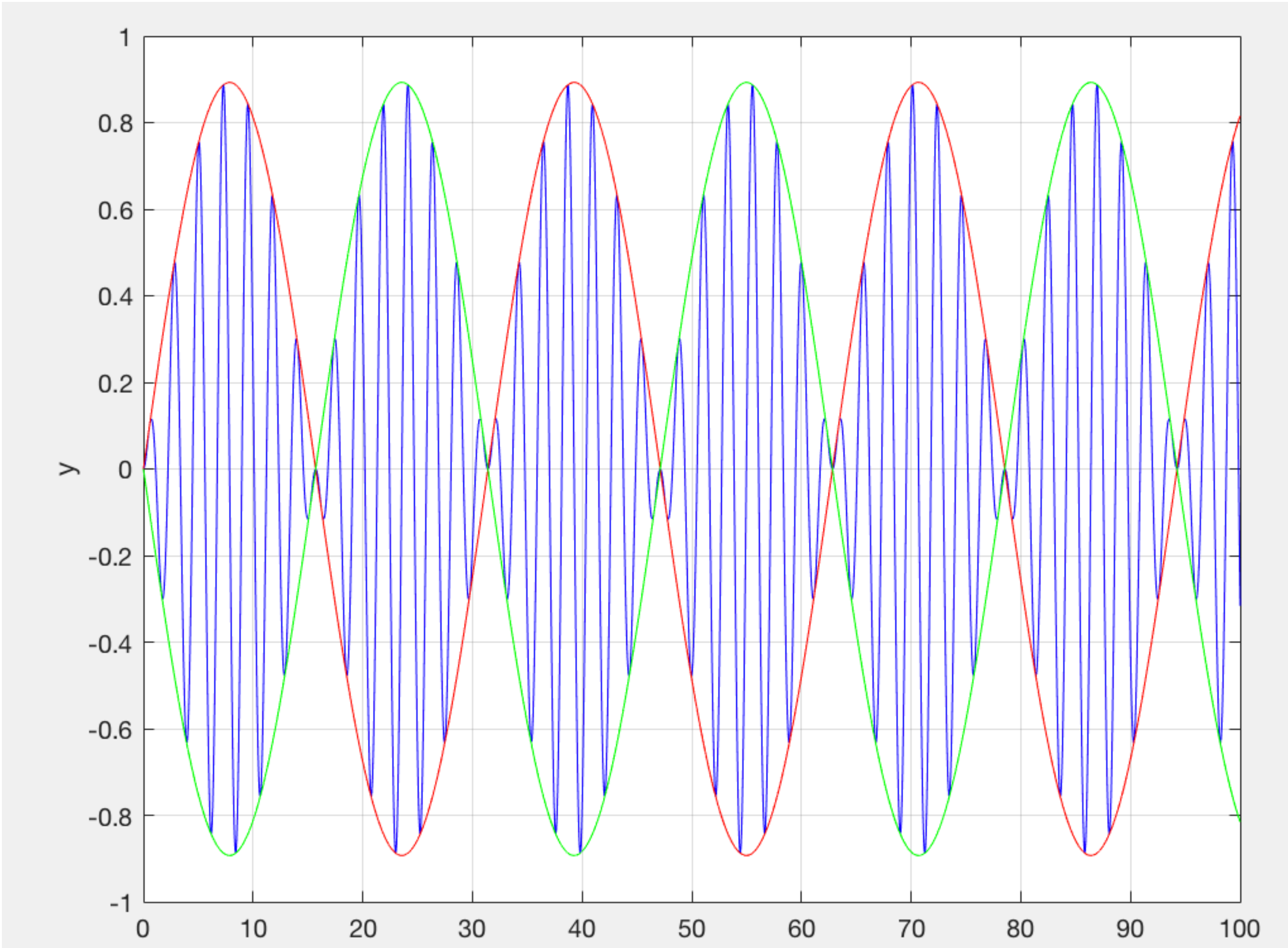
c) Looking at the graph, it appears to be a bit greater than 30. The length of the beats is equal to 2*pi/abs(omega0-omega) and that is determined to be 31.4159, which is definitely what it appears to be visually

d)
When omega = 2.9:



The fast oscillation is equal to 2.1299.
And the length of the beats are: 62.83.

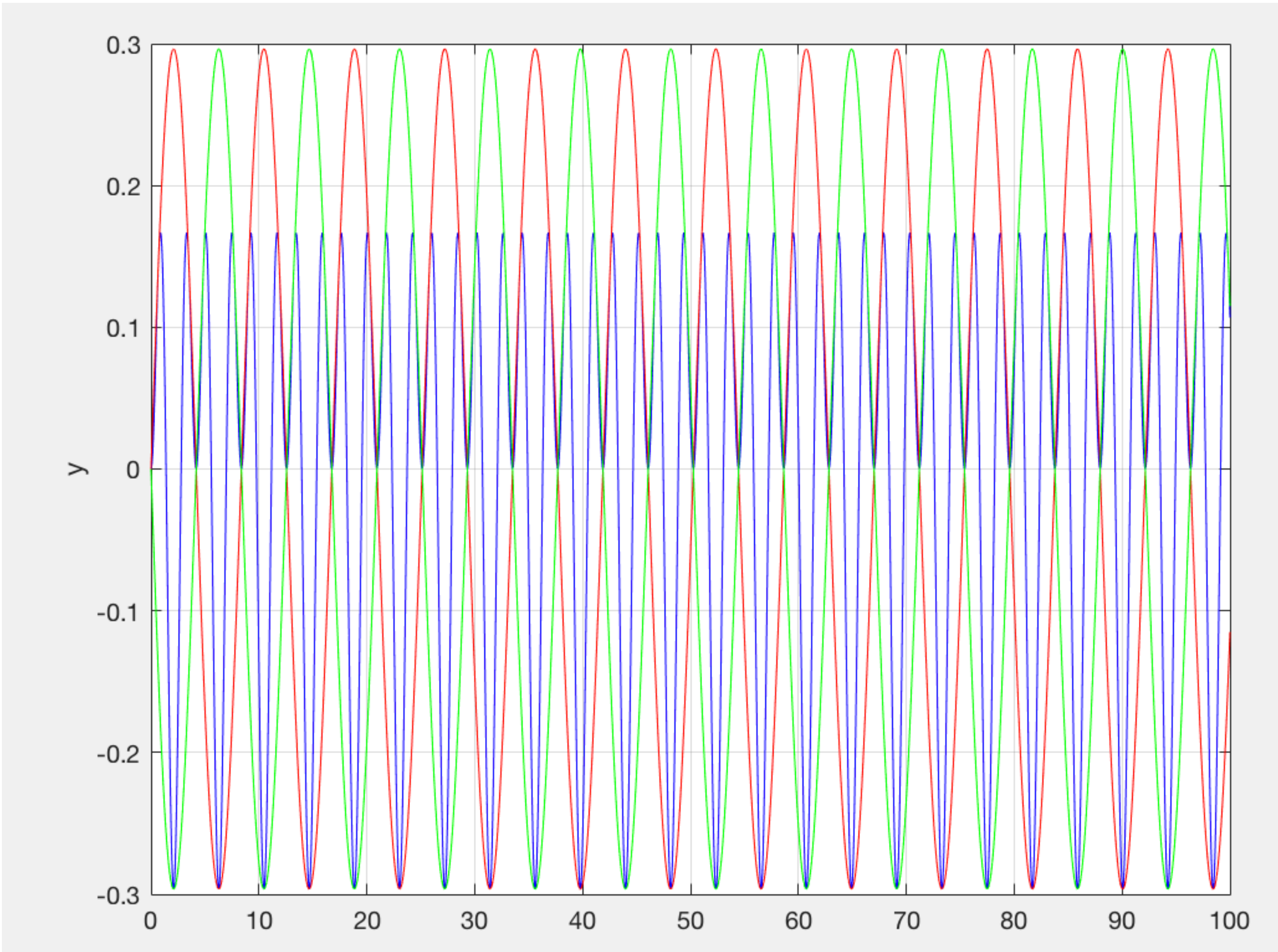
When omega = 2.6:



The fast oscillation is equal to 2.2440.
And the length of the beats are: 15.7080.

The change is that as omega gets farther away from omega0 the fast oscillation becomes greater and the lengths become shorter while getting closer the fast oscillation become smaller and the lengths are longer. This makes logical sense just by thinking about dividing the two omegas.

e) Does it technically exist? I suppose so as the math I'm putting in works and to note, alpha = 0. But the thing is that it doesn't look like a "beats phenomenon" looking at the graph, as there isn't many fast oscillations in the length of each beat:



0.3
0.2
0.1
0
-0.1
-0.2
-0.3
y
0
10
20
30
40
50
60
70
80
90
100