1. Define vectors
    a. `v = [1 2 3 4 5]`
    b. `a = v*3`
    c. `b = v.^2`
    d. `c = v.^v`

2. Find a limit

MATLAB is unable to work with factorials beyond 170 according to the documentation. I can successfully calculate far larger factorials in Python and Mathematica. MATLAB's factorial function is not working with such large numbers. I do not know or understand why.

```
clear
format long
for n = 100
    a = [0:n];
    b(1:n+1) = 3;
    c = b.^a;
    d(1:n+1) = factorial(a);
    % I used this method while figuring out what happened with the shorter
one.
    e = c./d;
    % This is the more succinct version.
    f = (b.^a)./factorial(a);
    sum(e)
    sum(f)
    exp(3)
end
```

3. Solve a system of linear equations

```
% x, y, z, c
fprintf('Given these three linear systems of equations...')
A = [4 3 1;5 6 2;2 5 -4]
B = [-8;10;9]
fprintf('These are the coefficients in order of x, y, z')
A\B
```

>> Hw2p3

Given these three linear systems of equations...

A =

4 3 1

5 6 2

2 5 -4

B =

-8

10

9

These are the coefficients in order of x, y, z

ans =

-8.6667

7.8235

3.1961

4. Bacteria growth

a) By Hand

b)

```
clear
format short

disp('N=N_0e^kt where N_0=1 and double time=3')
k = log(2)/3;
t = 0;
N = 1;
t = 0:2:24;
N = N*exp(k*t)
```

>> HW2

N=N_0e^kt where N_0=1 and double time=3

N =

Columns 1 through 8

1.0000 1.5874 2.5198 4.0000 6.3496 10.0794 16.0000 25.3984

Columns 9 through 13

40.3175 64.0000 101.5937 161.2699 256.0000

5. Radioactive decay

a) By Hand

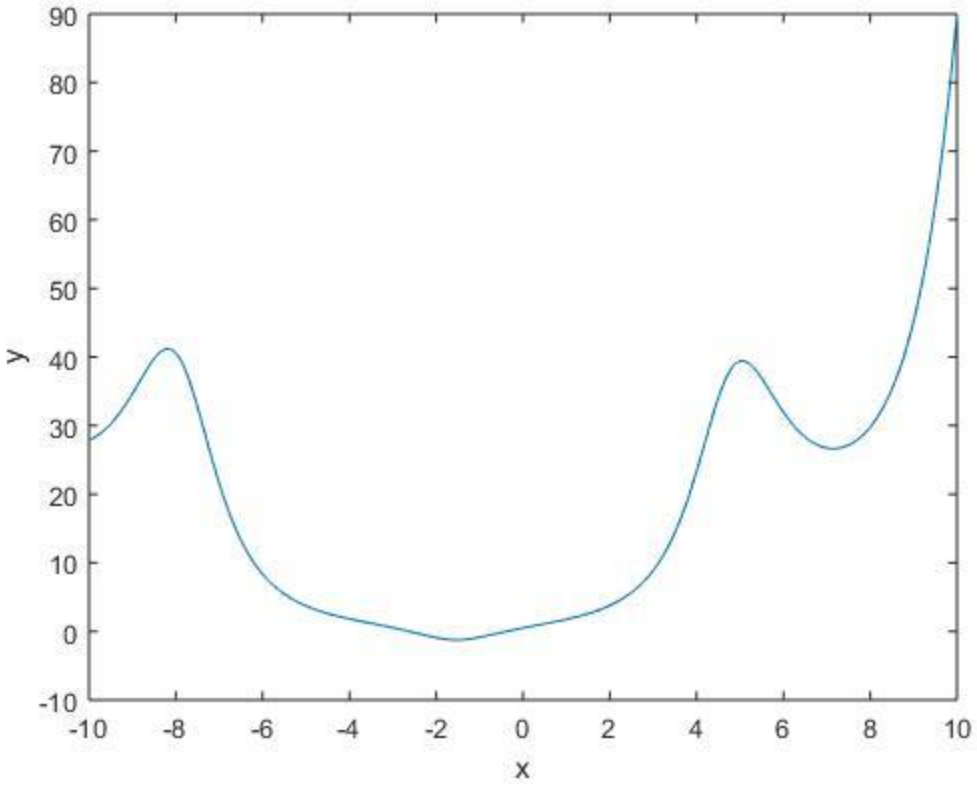
b)

```
% Using N_0 = 10
k = log(1/20)/11;
t = 0:4:48
N0 = 10;
N = N0*exp(k*t);
disp('Ratio')
N/N0
```

```
>> HW2

t =

    0    4    8   12   16   20   24   28   32   36   40   44   48

Ratio

ans =

  Columns 1 through 8

    1.0000    0.3364    0.1132    0.0381    0.0128    0.0043    0.0015    0.0005

  Columns 9 through 13

    0.0002    0.0001    0.0000    0.0000    0.0000
```

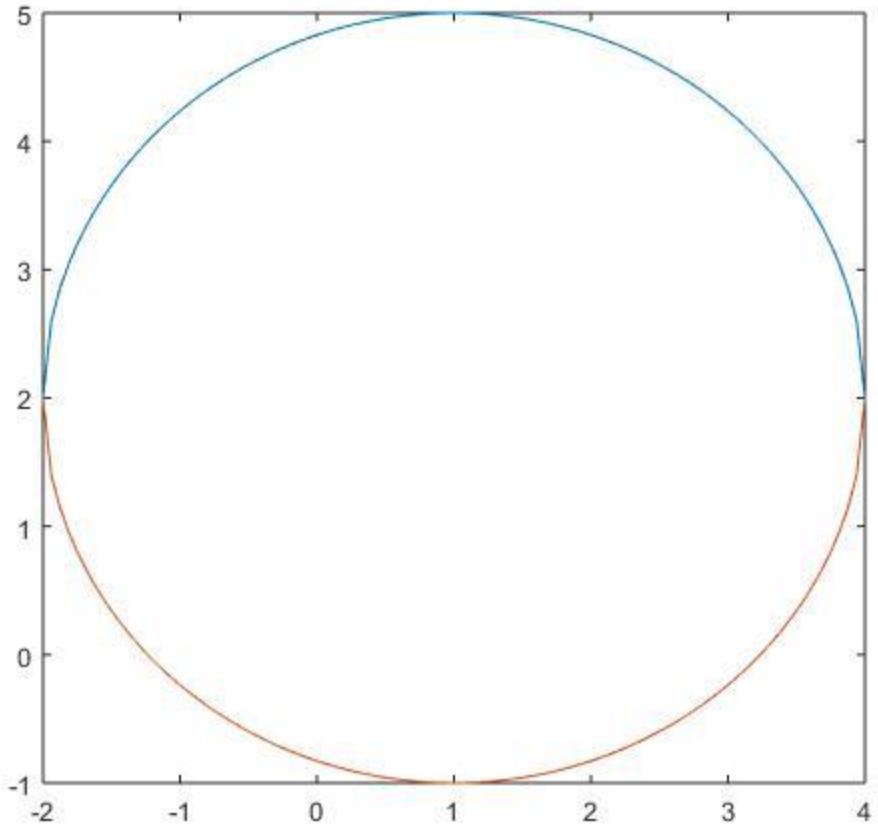
6. Plot a function

```
x = -10:0.01:10;
y = (x.^2 + 3.*x +1)./(2+sin(x));
plot (x,y), xlabel('x'), ylabel('y')
```



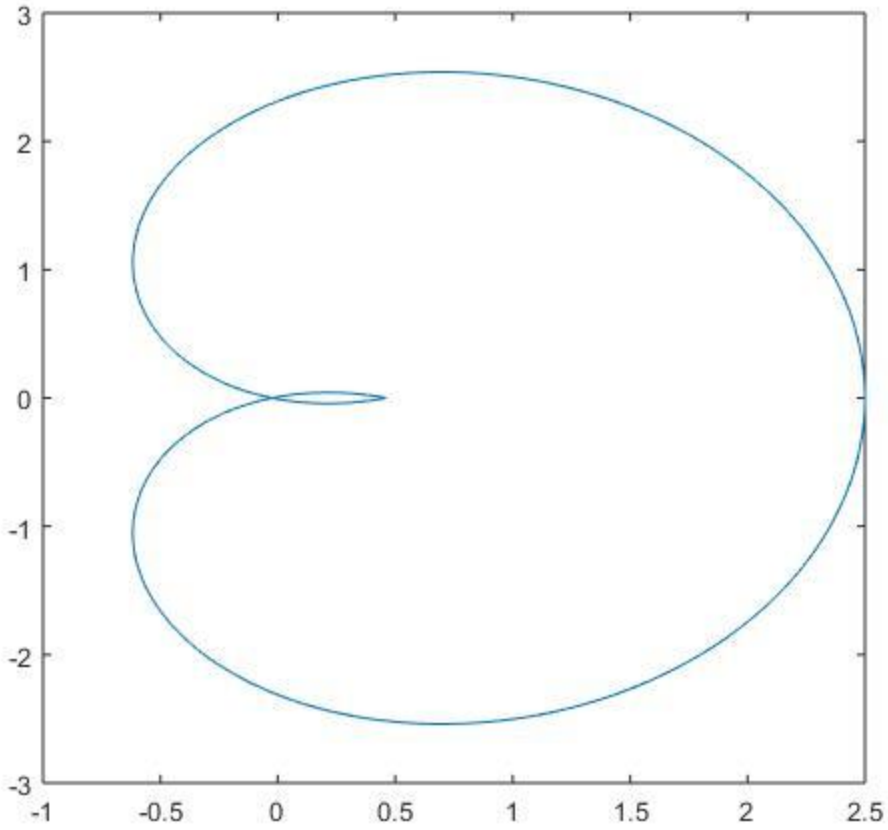
7. Plot a circle

```
a = 1;
b = 2;
r = 3;
x = linspace(a-r,a+r,100);
y1 = b + sqrt(r.^2-(x-a).^2);
y2 = b - sqrt(r.^2-(x-a).^2);
plot(x,y1,x,y2)
```



8. Plot a graph

```
t = -pi:0.01:pi;
x = (2.5-0.3.*t.^2).*cos(t);
y = (3.3-0.4.*t.^2).*sin(t);
plot(x,y)
```



9. Taylor series

```
x = -5:0.01:5;
e3 = 1 + x + (x.^2)/factorial(2);
e5 = 1 + x + (x.^2)/factorial(2)+(x.^3)/factorial(3)+(x.^4)/factorial(4);
e7 = 1 + x +
(x.^2)/factorial(2)+(x.^3)/factorial(3)+(x.^4)/factorial(4)+(x.^5)/factorial(
5)+(x.^6)/factorial(6);

subplot(2,2,1)
plot(x,e3)
title('Taylor Approximation with 3 Terms')
legend('3 terms, e^x')
xlabel('x')
ylabel('y')

subplot(2,2,2)
plot(x,e5)
title('Taylor Approximation with 5 Terms')
legend('5 terms, e^x')
xlabel('x')
ylabel('y')
```

```
subplot(2,2,3)
plot(x,e7)
title('Taylor Approximation with 7 Terms')
legend('5 terms, e^x')
xlabel('x')
ylabel('y')

subplot(2,2,4)
plot(x,exp(x))
title('e^x from MATLAB')
legend('Computed, e^x')
xlabel('x')
ylabel('y')
```