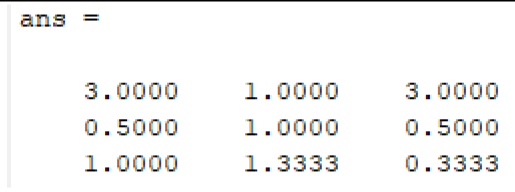
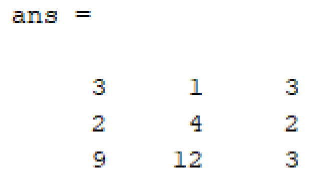
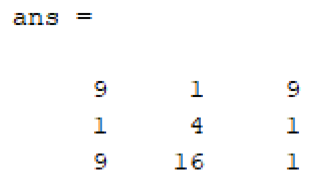
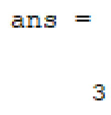
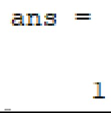
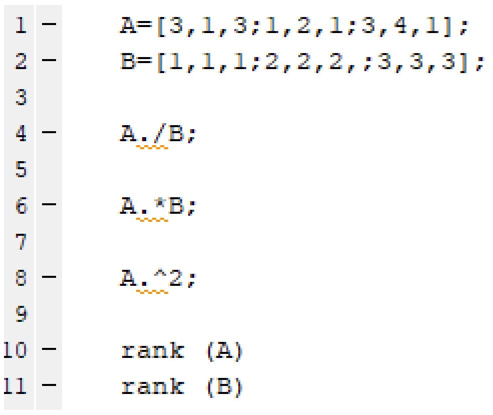
Abdurrafay Khan 1005215517

Lab #5

Exercises: Matrix Basic Operations

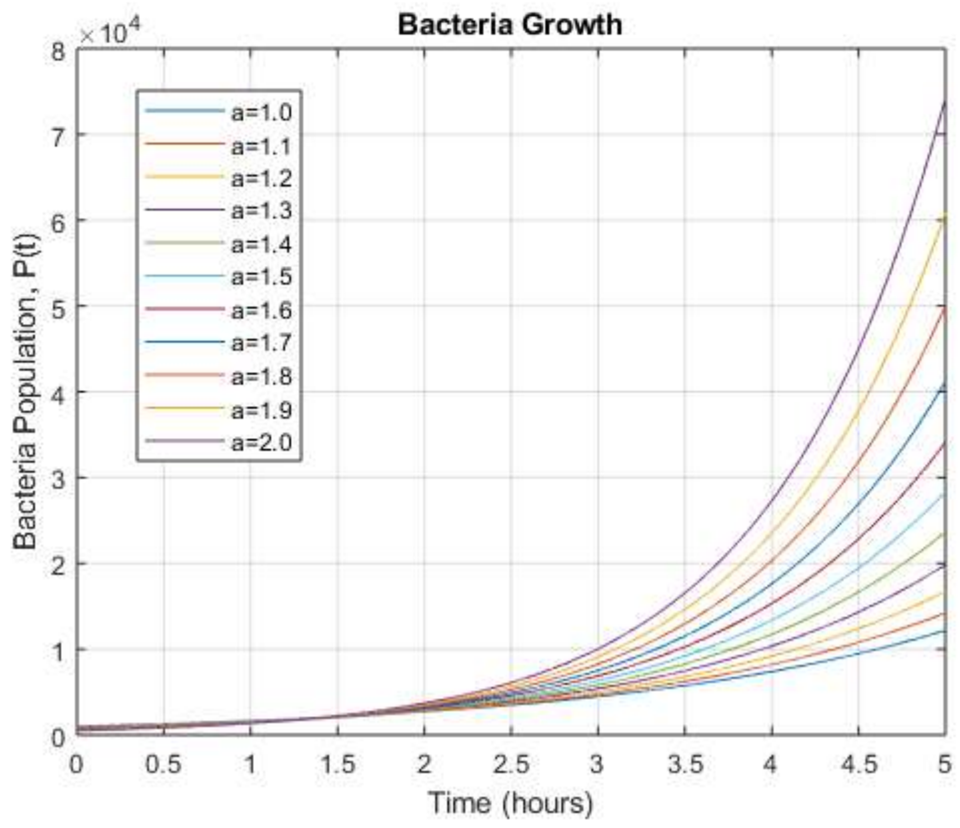
Expression	Result
A./B	 ans = 3.0000 1.0000 3.0000 0.5000 1.0000 0.5000 1.0000 1.3333 0.3333
A.*B	 ans = 3 1 3 2 4 2 9 12 3
A.^2	 ans = 9 1 9 1 4 1 9 16 1
rank(A)	 ans = 3
rank(B)	 ans = 1

The table above was filled out by running the following code:

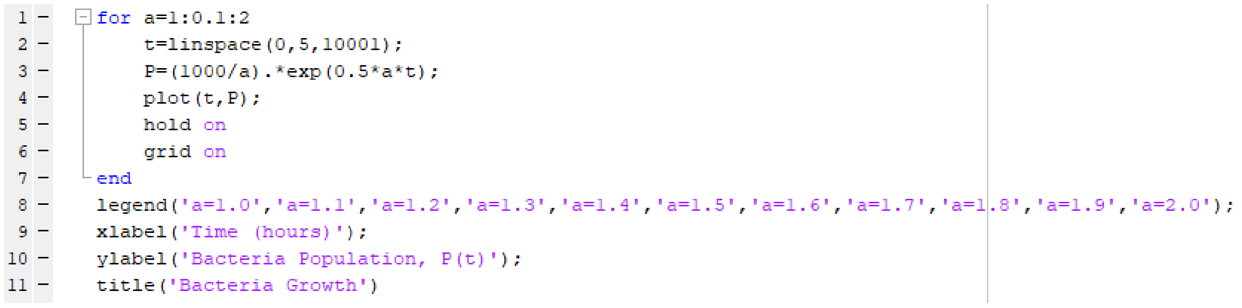


```
A=[3,1,3;1,2,1;3,4,1];
B=[1,1,1;2,2,2,;3,3,3];

A./B;

A.*B;

A.^2;

rank (A)
rank (B)
```

Exercise: Bacteria Growth and Interpretation



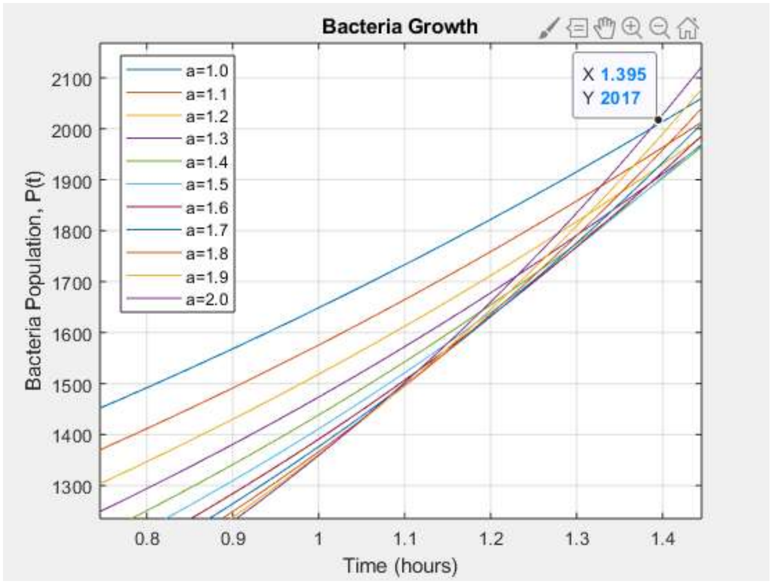
The graph above was made through the following code:



```
for a=1:0.1:2
    t=linspace(0,5,10001);
    P=(1000/a).*exp(0.5*a*t);
    plot(t,P);
    hold on
    grid on
end
legend('a=1.0','a=1.1','a=1.2','a=1.3','a=1.4','a=1.5','a=1.6','a=1.7','a=1.8','a=1.9','a=2.0');
xlabel('Time (hours)');
ylabel('Bacteria Population, P(t)');
title('Bacteria Growth')
```

1. The lowest line (blue line) belongs to a=1.0 and the highest and steepest line (purple line) belongs to a=2.0. This is the case because as exponents increase, so does the output. In the case of this graph, as ‘a’ increases, the graph gets steeper, hence the purple line is associated with a=2.0 and the blue line is associated with a=1.0

2. From the previous question, we already determined that a=2.0 is the largest as time progresses. Based on the graph below, a=2.0 becomes the largest graph at around t=1.4hrs



3. Below are the doubling Time for the population of 1000 to become 2000 based on the graph, for the 'a' values of:

   a=1.0: about 1.386 hrs
   a=1.5: about 0.924 hrs
   a=2.0: about 0.692 hrs

   These values are extremely close to the values I have determined on my calculator. The relationship between doubling time and the 'a' value is that as 'a' increases, the doubling time decreases.

4. The similarities in doing matrix math by hand and via Matlab is that the user must input Matrices and operations that work for the matrices inputted. For example, matrix addition would not be possible if the user does not create matrices of the same size, whether it be by hand calculations or via Matlab. The difference between the two methods is how matrices are written. When doing matrices by hand, the user writes the matrix consisting of rows and columns surrounded by square brackets. When using Matlab, the entire matrix is written in one line of code where each row in the matrix is separated by a semi-colon.

   Since Matlab performs matrix operations much quicker than a human would do by manual calculation, using Matlab would be much more appropriate, especially in terms of larger matrices like 6x6 matrices and greater. Working with large matrices by hand also increases the chance for error so Matlab would help. Performing operations as well as finding the rank of a matrix are done quickly using Matlab so it would be much more efficient to use at times. An engineer should use 'for' loops when there is a function or task they want repeated for a certain number of times. This is a very effective tool in Matlab to avoid wasting time writing tedious repetitive code. One should be careful to note how long they want their loop to repeat for and set their given condition with caution and proper thought prior ahead.