G C

RCarson10

Abstract

This project implements a potential solution for each of the system of Linear differential equations

Benchmark – Project 4 – Modeling with Systems of 1st Order Differential Equations

By Gaj Carson

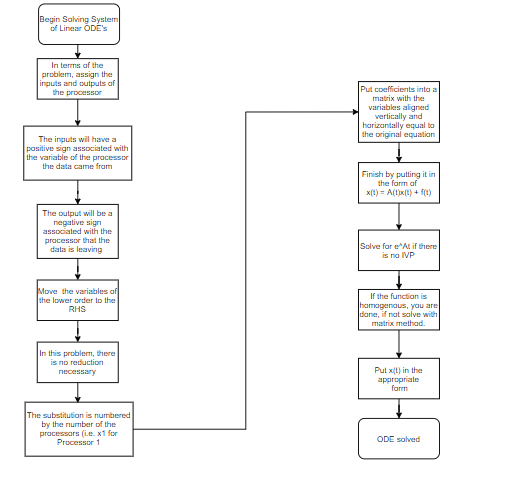
# **Abstract**

A linear first-order system of two differential equations is of the form (or can be put in the form)

The coefficients , , , and the forcing functions , are prescribed functions of the independent variable t on the interval of interest, and and are dependent variables - the unknowns. If both and are zero, then (1) is homogeneous; otherwise, it is nonhomogeneous. The double subscript notation for the coefficients is as follows: The first subscript indicates the equation in which the coefficient occurs, and the second indicates the unknown that it multiplies. For instance, is in the second equation (i.e., the differential equation) and it multiplies the first unknown .

**Flowchart**

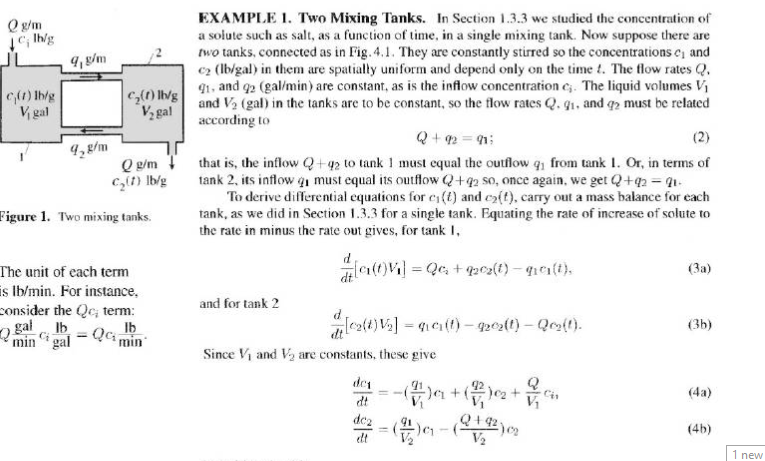
The flowchart below explains the approach I took to solving both parts of the benchmark.



**Hand Calculations Part 1**

The equation I derived from the three processors is:

The methodology I used was basically following the mixture example in Chapter 4 of our ODE book.



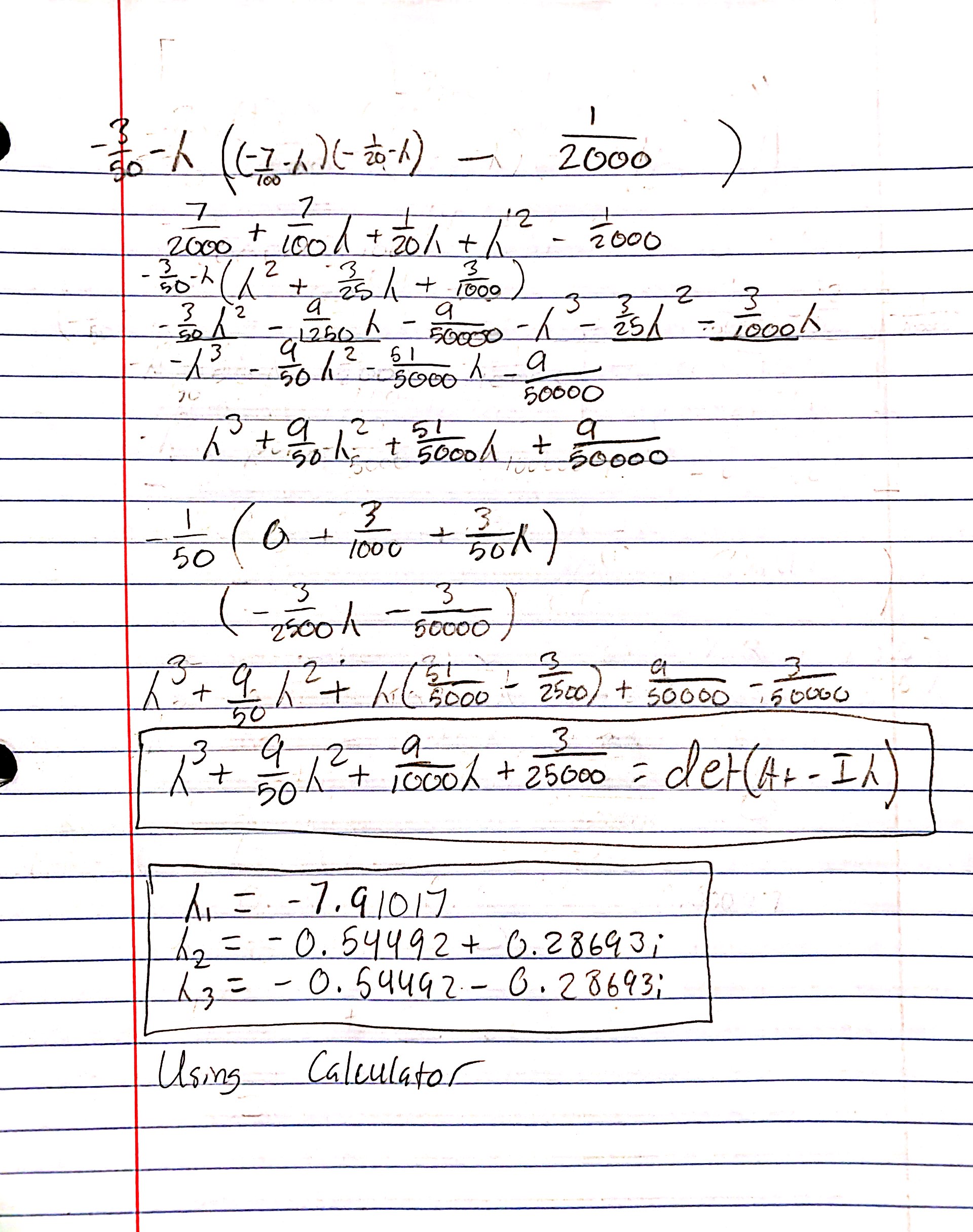
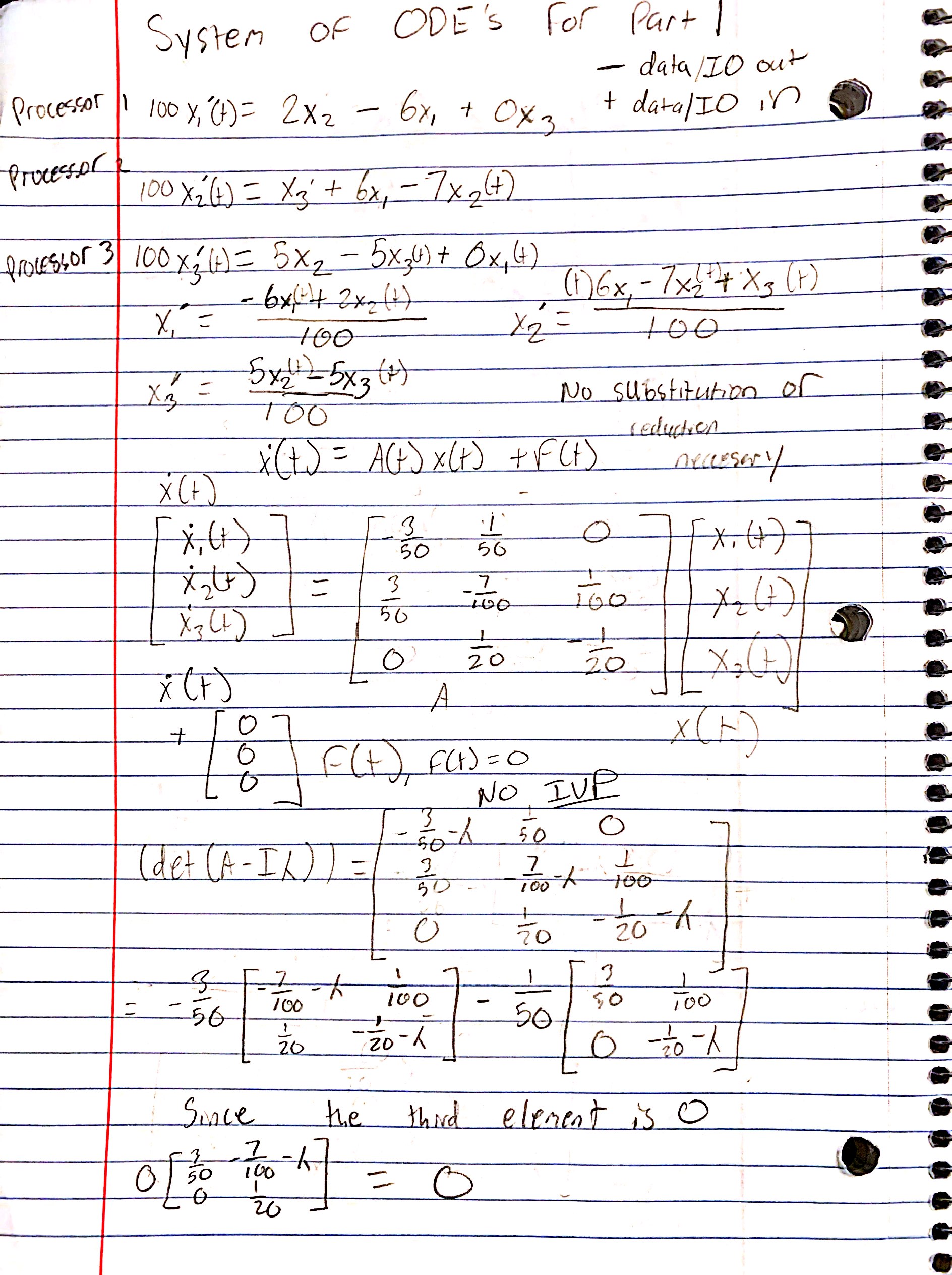
I solved the system by deriving a first order ODE for each processor and then transformed the system into the form of

The data into the processor is a positive coefficient multiplied by the variable representing which processor the data came from. The data out of the processor is a negative coefficient multiplied by the variable representing the processor of which the data is leaving which ends up being the total data leaving the processor. Following the example in the book, each processor has 100 MB of data which is equivalent to the V constant in the mixture example and each processor has that coefficient attached to the DE. Since the equation is directly derived into three Linear First Order differential equations, there is no need for the reduction step when converting the system into the form of

. = \* +

Since , the zero vector is added thus just making the answer . Lastly, the substitutions where also derived into the construction of the equations, the data from/leaving Processor 1, the data from/leaving Processor 2, and the data from/leaving Processor 3. Since I/O input in the beginning contains no data yet, it is not factored into the input of the first processor since we are solving for I/O data and not I/O. The eigenvalues end up being complex numbers and this same methodology is implemented in part two.

My calculations are demonstrated below in the screenshots.

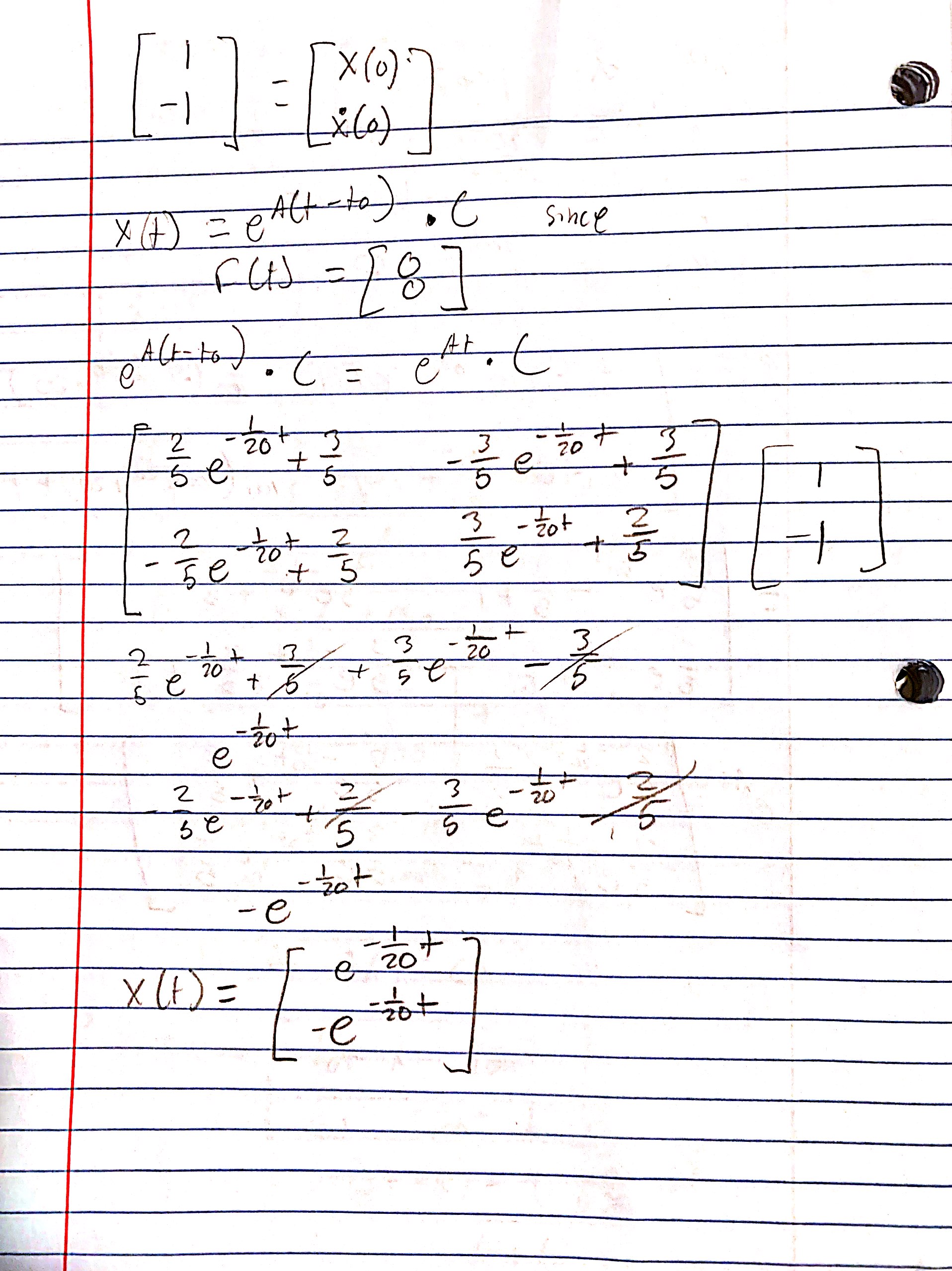
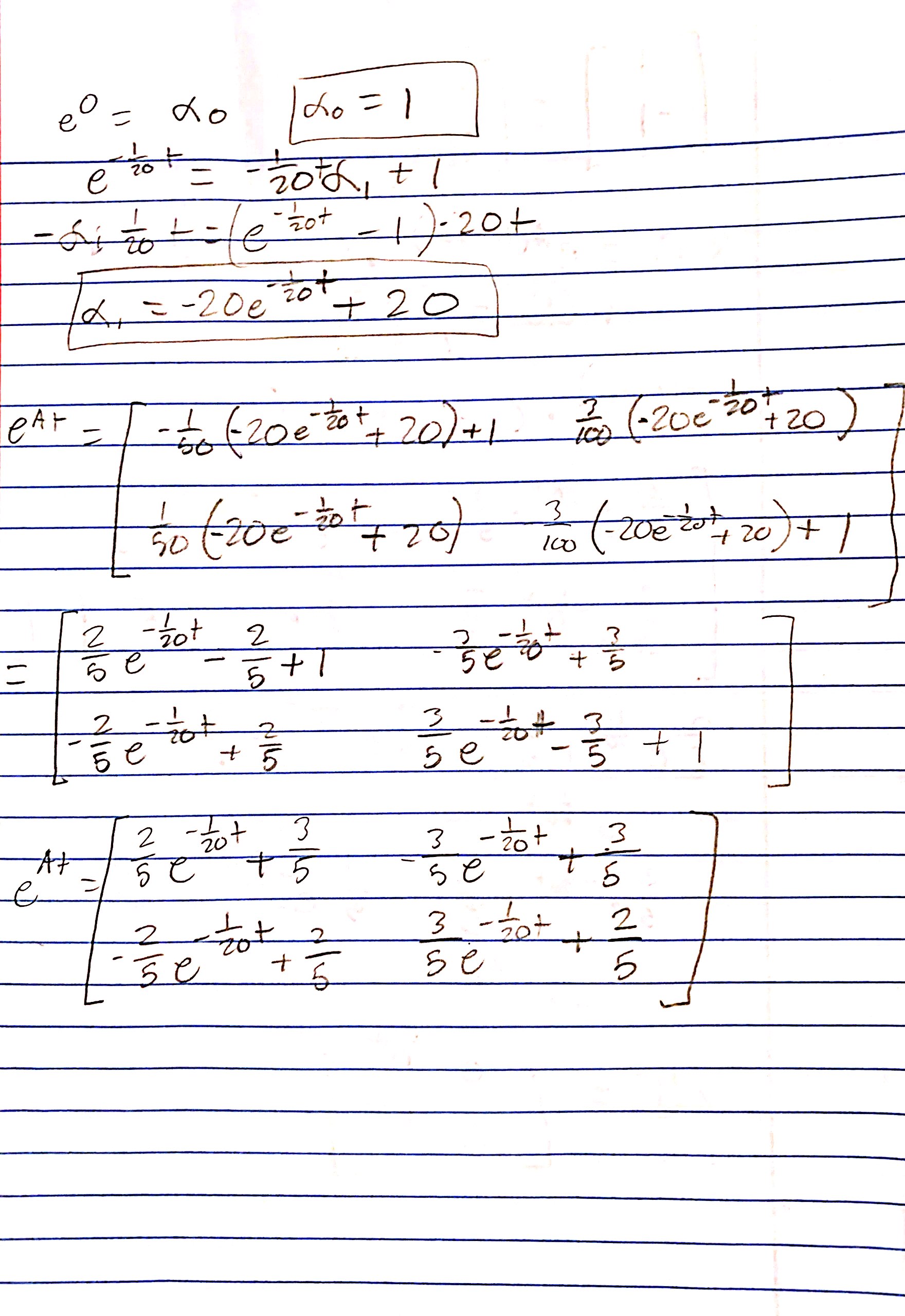
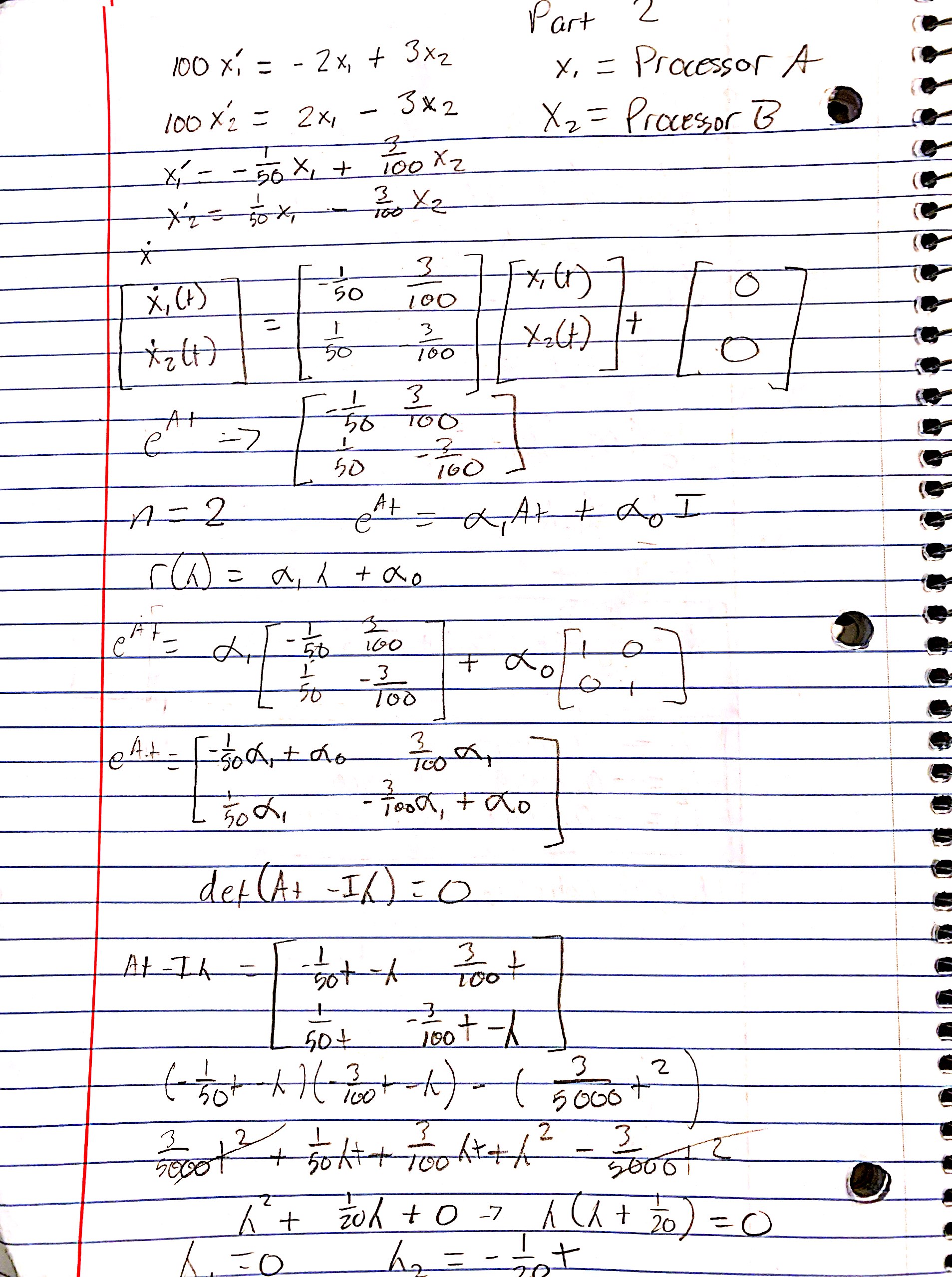


**Hand Calculations Part 2**

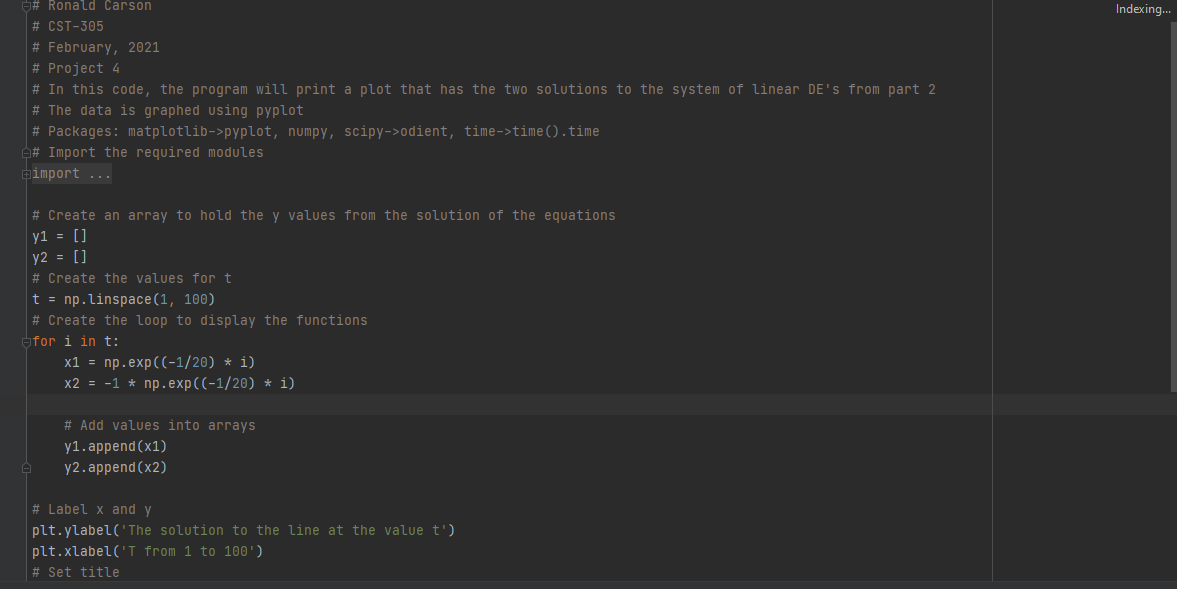
The way this part was calculated is the same as in Part 1. The equation I derived from the two processors is:

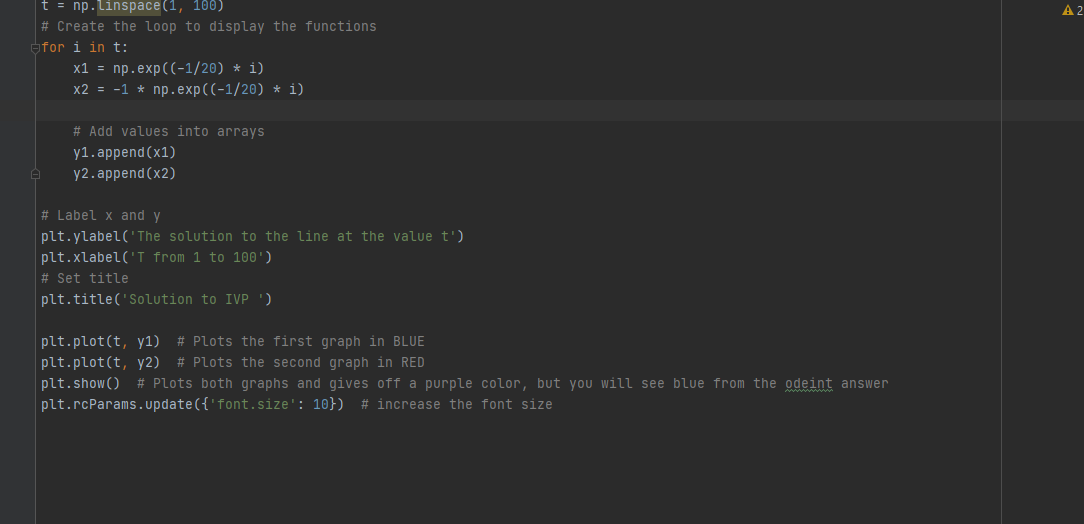
The hand calculations are below followed by the code and the graph that was created from the code. Next, here is the and the gained from the matrix method since and the DE is homogenous.

The answer to the initial value problem is:



### Code Output And Code





#### Graph Output for Part 2

## 

## References

1. Ordinary differential equations. (2012). In *Ordinary Differential Equations*. Erscheinungsort nicht ermittelbar: John Wiley & Sons.