GNG1106 – Fundamentals of Engineering Computation
Course Project

Chemical Engineering Project
Transient Response of Chemical Reactors

Regan Aubrey 300066276
Timothy Lam 300120566

Date: November 17th 2019

SECTION 1: Problem Identification and Statement

When multiple reactors are connected together, they create what we call a system of reactors. In this project, we will be creating a program that solves a specific system of reactors based on given values from the user. The system consists of 3 reactors which each have constant volumes.

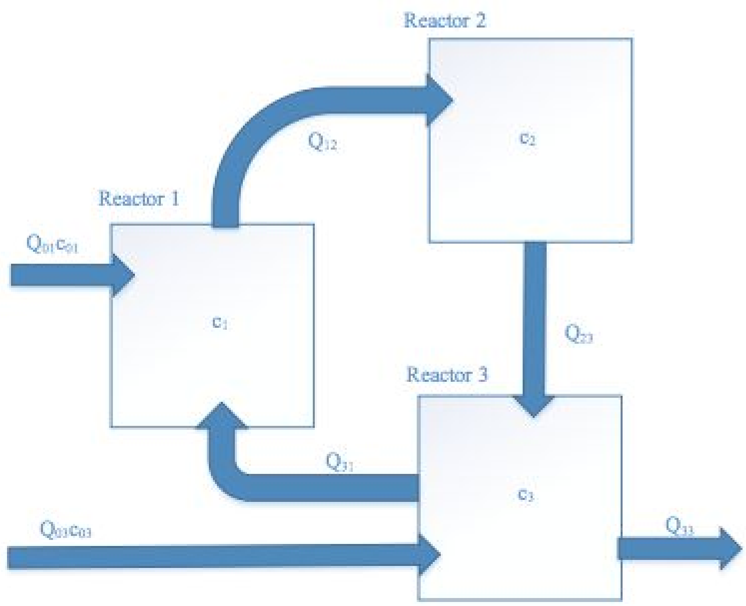


Figure 1. System of coupled reactors

Inside chemical reactors, the input flows can change concentrations. When those concentrations change, it will cause the chemical substance concentrations are in the reactor to change. This change in concentration is known as the transient response of a chemical reactor.

Chemical reactor systems can be monitored using mass balances. Mass Balances must follow the law of conservation of mass. This means that the total accumulation of the system will be whatever is input less whatever is output. Based on this fact it is possible to come up with an equation for the accumulation in a mass balance system:

Accumulation = Input - Output *(eq.1)*

In eq.1, the input is whatever is being put into the system and will be the concentration of the substance being put in, multiplied by the flow rate of the input stream. The output is very similar and can be found by multiplying the concentration of the outflow substance by the flow rate leaving the system. By using the mass balance, it is possible to solve the system by setting the inputs and outputs to actual solvable variables as well as parameters as seen in the equations below.

Input = Qin*c	Output=Qout*c

Qin = Inflow c= concentration *(eq.2)*	Qout = Outflow c= concentration *(eq.3)*

The purpose of this project is to solve a transient system in which the reactants concentrations change within the reactors over time. It is possible to compute these concentration changes by finding the derivative of the concentration in terms of time and multiplying it by the volume of the liquid. This can be done due to the fact that the volume will remain constant throughout the process, as long as we can assume that the input as well as the output flow rates, are equivalent. Thus we will be left with another equation for accumulation.

$$\text{Accumulation} = V(dc/dt) \quad \textit{(eq.4)}$$

These equations can be used to start solving the mass balance system. By setting (eq.1) equal to (eq.4) you will be able to calculate the mass balance for each reactor.

$$V(dc/dt) = Qin*c - Qout*c \quad \textit{(eq.5)}$$

By using *(eq.5)* it is possible to come up with a system of equations for each reactor in *Figure 1.* This can be done by finding the inflows of each reactor as well as the outflows. For example, if we were to look at the first reactor, there are two streams coming in and one stream leaving. The first input stream of reactor 1 can be written as $Q_{01}c_{01}$ and the second input stream (coming from the third reactor) can be written as $Q_{31}c_3$. Since the last stream connected to the first reactor is an output stream it is necessary to negate it when adding it to the equation, to account for the flow leaving the reactor. If this process is followed for each reactor, a series of equations can be created.

Reactor 1: $(dc_1/dt) = (1/V_1)(Q_{01}c_{01} + Q_{31}c_3 - Q_{12}c_1)$ *(eq.6)*
Reactor 2: $(dc_2/dt) = (1/V_2)(Q_{12}c_1 - Q_{23}c_2)$ *(eq.7)*
Reactor 3: $(dc_3/dt) = (1/V_3)(Q_{03}c_{03} + Q_{23}c_2 - Q_{31}c_3 - Q_{33}c_3)$ *(eq.8)*

Figure 3. system of mass balances

A system of equations is necessary in order for the C program to verify that the user input that the inflow is equivalent to the outflow in the system. This can be shown in a similar way to what was done for the mass balance equations above. For the first reactor you set the inflows to positive and the outflows to negative, so you would get a positive flow going into the reactor to begin with, a positive flow coming from the third reactor into the first reactor and a negative flow leaving the first reactor into the second reactor. The system of equations you end up with will all be equal to zero. It is important to note that an overall mass balance system should be included, ie. the streams going into the overall system and the streams leaving the overall system.

Reactor 1: $0 = Q_{01} + Q_{31} - Q_{12}$ *(eq.9)*
Reactor 2: $0 = Q_{12} - Q_{23}$ *(eq.10)*
Reactor 3: $0 = Q_{03} + Q_{23} - Q_{31} - Q_{33}$ *(eq.11)*
Overall Mass Balance: $0 = Q_{01} + Q_{03} - Q_{33}$ *(eq.12)*

Figure 3. A system of equations to verify that the inflow = outflow and the whole system

A few assumptions can be made for the mass balance:

1. The substance's concentration will be uniform everywhere in the reactor (fully mixed at all times).
2. The reactor's volume remains constant, therefore the inflow must be equivalent to the outflow.
3. The substance inside the reactor has the same concentration as the concentration of the outflow.

In order to solve the mass balance and transient concentrations as a substance flows through a system of reactors (figure 1), we will be developing a C program. The C program will take the user's input of the volumes from each reactor (V_1, V_2, and V_3), the flows (Q_{01}, Q_{03}, Q_{12}, Q_{23}, Q_{31}, and Q_{33}), the input concentrations (c_{01} and c_{03}), the initial concentrations at the start time, (t=0, c_{10}, c_{20} and c_{30}), as well as the end time which tells the program how long the response will be studied for.

The software will follow all of the assumptions written above, as well as make sure that every value input by the user is valid by using a function that will check the constraints. The function should include a flagged loop that will be able to confirm the validity of the values. If they meet the conditions given by *(Figure 3),* (flag is TRUE), then it ensures the program that all of the volumes within the reactors are constant. If all of the values input by the user are valid, the program will continue to run, and plot three curves. Each curve demonstrates the concentration change over time for each reactor, as well as the input values by the user in tables on a separate part of the console. When the user inputs the values the program will allow them to save up to five value sets in a file, and can also choose which set they want to use. They can also input an entirely new set of values.

In summary, the program will be made up of multiple functions that will use loops, plot graphs, do calculations and print statements to the console as well as graphs for the user to see and understand.

SECTION 2: Gathering of Information and Input/Output Description

2.1 SubSection 1: Numerical Methods: Using Euler's Method to Determine An Equation Which Calculates the Change in Concentration for The First Reactor

$$C_{1,i} = C_{1,i-1} + \left(\frac{Q_{01}c_{01}+Q_{31}c_{3,i-1}-Q_{12}c_{1,i-1}}{V_1}\right) \Delta t$$

- $C_{1,i}$ is the change in concentration of the first reactor
- $C_{1,i-1}$ is the concentration of the chemical substance at a given time
- Q_{01} represents the flow rate into the first reactor
- c_{01} is the concentration into the first reactor
- Q_{31} represents the flow rate of the first reactor from the third reactor
- $c_{3,\,i-1}$ is the concentration in the third reactor at a given time
- Q_{12} represents the flow rate out of the first reactor
- $c_{1,i-1}$ is the concentration in the first reactor at a given time
- V_1 is the volume of the first reactor
- Δt is the change in time of the substance concentration, in which t>0

2.2 SubSection 2: Numerical Methods: Using Euler's Method to Determine An Equation Which Calculates the Change in Concentration for The Second Reactor

$$C_{2,i} = C_{2,i-1} + \left(\frac{Q_{12}c_{1,i}+Q_{23}c_{2,i-1}}{V_2}\right) \Delta t$$

- $C_{2,i}$ is the change in concentration of the second reactor
- $C_{2,i-1}$ is the concentration of the chemical substance at a given time
- Q_{12} represents the flow rate into the second reactor
- Q_{23} represents the flow rate of the third reactor from the second reactor
- V_2 is the volume of the second reactor
- Δt is the change in time of the substance concentration in which t>0

2.3 SubSection 3: Numerical Methods: Using Euler's Method to Determine An Equation Which Calculates the Change in Concentration for The Third Reactor

$$c_{3,i} = c_{3,i-1} + \left(\frac{Q_{03}c_{03}+Q_{23}c_{2,i-1}-Q_{31}c_{3,i-1}+Q_{33}c_{3,i-1}}{V_3}\right) \Delta t$$

- $c_{3,i}$ is the change in concentration of the third reactor
- $c_{3,i-1}$ is the concentration of the chemical substance at a given time
- Q_{03} represents the flow rate into the third reactor
- c_{03} is the concentration into the third reactor
- Q_{23} represents the flow rate of the third reactor from the second reactor
- $c_{2,\, i-1}$ is the concentration in the second reactor at time t.
- Q_{31} represents the flow rate out of the third reactor from the first reactor
- Q_{33} represents the flow rate out of the third reactor
- V_3 is the volume of the third reactor
- Δt is the change in time of the substance concentration

2.4 Input and Output

Input:

In order for euler's method to determine the change in concentration of the reactors, it would require input values from the user. The user will input the volumes of each reactor V_1, V_2, V_3,the flow rates $Q_{01}, Q_{03}, Q_{12}, Q_{23}, Q_{31}, Q_{33}$, the input concentrations c_{01} and c_{03}, the initial concentrations at time t=0, c_{10}, c_{20}, c_{30}, and the final time that defines the range over which Δt can be determined by subtracting the initial time by the final time provided by the user. With the given inputs by the user, the values can be substituted into the equation to give the resulted output. From this, the change in concentration of the next reactors can be determined by reusing the values of the parameters provided by the user and from the calculations.

Output:

If the values of the input are valid, the output would be 3 plots on a graph. Each curve on the graph will show how the concentration of each reactor will change over time. In addition to the output, the software will display the information of the input values by the user on the console in a table format. Incrementally, the values of the inputs can be saved in a file; up to five sets of values can be stored so that the user can reuse the old parameter values.

SECTION 3: Test Cases and Design

3.1 Test Cases

	Test Cases						
Variables	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6-should not work	Test 7-should not work
Q_{01}	5	2	0	22	1000	5	7
Q_{03}	5	1.5	4	43	1240	-15	3
Q_{12}	15	2.5	14	83	1500	25	7
Q_{23}	15	2.5	14	83	1500	26	8
Q_{31}	10	0.5	14	61	500	20	5
Q_{33}	10	3.5	4	65	2240	20	3
V_1	1	10	5	60	500	0 - should not work	3
V_2	2	15	10	80	1000	-15	7
V_3	3	20	5	60	500	-20	3
c_{01}	1	8	10	80	1000	15	7
c_{03}	3	12	5	60	500	20	3
$c_{1}\square_{0}$	1	8	8	64	550	15	5
$c_{2}\square_{0}$	2	12	14	68	560	17	7
$c_{3}\square_{0}$	3	8	16	72	570	19	9
tf	6	16	18	20	15	-20	10

The test cases used in the above table show a range from small to large numbers. The first five test cases in the table should work and the last two test cases (6 and 7) should be invalid and the program should as the user to input new variables. There are multiple issues for the last two test cases, they do not follow the constraints given by the equations below, there are negative inputs, and the initial volume of one of them is zero which will not work (since you can't divide by zero).

Constraints:

$0 = Q_{01} + Q_{31} - Q_{12}$

$0 = Q_{12} - Q_{23}$

$0 = Q_{03} + Q_{23} - Q_{31} - Q_{33}$

$0 = Q_{01} + Q_{03} - Q_{33}$

$tf > 0$

$V_{1,2 \text{ and } 3} > 0$

3.2 Design

3.2.1 Introduction

Every value input into the program by the user will be stored in the structures. There will be three functions which use the input values (parameters). All of the parameters, other than those from the arrays will be those given by the user. The program will use and call the functions in the main function, in order to calculate the change in concentration of each reactor based on the given values, save the given values to a separate file, and output the values as well as a graph with three curves representing the change in concentration of each reactor. A list of the structures to be used in this program are shown below.

Structures:

REACTORS: v_1, v_2, v_3

CONCENTRATIONS: c_$_{01}$, c_$_{03}$, cr1[ ], cr2[ ], cr3[ ] ,c1_0 , c2_0,c3_0, time_axis[], time_final

FLOW_RATES: Q_$_{01}$, Q_$_{03}$, Q_$_{12}$, Q_$_{23}$, Q_$_{31}$, Q_$_{33}$

Functions that will be used in this program include:

1. void receiveUserInputs(REACTORS *rPtr, FLOW_RATES *fPtr, CONCENTRATIONS *cPtr);
2. int testConstraints(FLOW_RATES *fPtr);
3. void calculateConcentrations(REACTORS *r, FLOW_RATES *f, CONCENTRATIONS *c);
4. void plotTable(CONCENTRATIONS *cPtr);
5. void storeFiles(REACTORS *rPtr, FLOW_RATES *fPtr, CONCENTRATIONS *cPtr);
6. int retrieveFiles(REACTORS *rPtr, FLOW_RATES *fPtr, CONCENTRATIONS *cPtr);
7. double getMinDouble(double [], int);
8. double getMaxDouble(double [], int);

3.2.2 Functions for Interacting with the User

The function used to interact with the user, and to receive the inputs is called by the main function. The function begins by asking the user to input parameter values that will be used to calculate the output (change in concentrations), it then stores those values to their specific structures mentioned in section 3.2.2. The user's inputs will then go through a second function which checks the constraints to make sure the inputs are valid. The first function will only be complete if final time is greater than zero as well as the initial volumes of the reactors.

1. void receiveUserInputs(REACTORS *rPtr, FLOW_RATES *fPtr, CONCENTRATIONS *cPtr)

Parameters in structures:
double startTime = 0
double finalTime
double c_{01}, c_{03}
double cr1[ ], cr2[ ], cr3[ ]
double Q_{01}, Q_{03}, Q_{12}, Q_{23}, Q_{31}, Q_{33}
double V_1, V_2, V_3

Logic: User inputs values and they are stored in the structures. Uses printf and scanf to execute this function.

The second function which interacts with the user is the function which tests the users inputs to ensure that they are all valid. If they are invalid inputs it will ask them to input new values until the constraints are met.

2. double testConstraints(FLOW_RATES *fPtr)

Parameters included in this function:
double Q_{01}, Q_{03}, Q_{12}, Q_{23}, Q_{31}, Q_{33}

Constraints:
$0 = Q_{01} + Q_{31} - Q_{12}$
$0 = Q_{12} - Q_{23}$
$0 = Q_{03} + Q_{23} - Q_{31} - Q_{33}$
$0 = Q_{01} + Q_{03} - Q_{33}$

Logic: Function 2. is called by function 1. and will test the values based on the constraints and will not continue through the program until the values input are valid.

3.2.3 Functions for Calculations

3. void calculateR1Concentration(REACTORS *r, FLOW_RATES *f, CONCENTRATIONS *c)
   Parameters:

   double Q_{01}, Q_{31}, Q_{12}
   double c_{01}, cr1[ ], cr3[ ]
   double startTime = 0, finalTime, V_1

4. void calculateR2Concentration(REACTORS *r, FLOW_RATES *f, CONCENTRATIONS *c)
   Parameters:

   double Q_{12}, Q_{23}
   double cr1[ ], cr2[ ]
   double startTime = 0, finalTime, V_2

5. void calculateR3Concentration(REACTORS *r, FLOW_RATES *f, CONCENTRATIONS *c)
   Parameters:

   double Q_{03}, Q_{31}, Q_{23}, Q_{33}
   double c_{03}, cr2[ ], cr3[ ]
   double startTime = 0, finalTime, V_3

Logic:

The three functions calculate the concentrations of the tree reactors, called by the main function. Each function will take the values stored in the structured and place them into the correction equation when its variable is called through a pointer. These functions will fill in the arrays from the equations.

3.2.4 Functions for Plotting

6. void plotTable(REACTORS *rPtr, FLOW_RATES *fPtr, CONCENTRATIONS *cPtr)

   Parameters:

   int ix;
   double Q_{01}, Q_{03}, Q_{12}, Q_{23}, Q_{31}, Q_{33}
   double c_{01}, c_{03}, cr1[], cr2[], cr3[]
   double V_1, V_2, V_3, finalTime, startTime=0

This function is called by the main function and displays a table of start times vs concentration of the three reactors. The ix variable is an increment and to control the loop of rows distributed. The increment is calculated by dividing finalTime by N-1.

3.2.5 Functions for Files

7. void storeFiles(REACTORS *rPtr, FLOW_RATES *fPtr, CONCENTRATIONS *cPtr)

   Parameters:

   double Q_{01}, Q_{03}, Q_{12}, Q_{23}, Q_{31}, Q_{33}
   double c_{01}, c_{03}, cr1[], cr2[], cr3[]
   double V_1, V_2, V_3
   double finalTime

double startTime = 0
double calculateR1Concentration
double calculateR2Concentration
double calculateR3Concentration

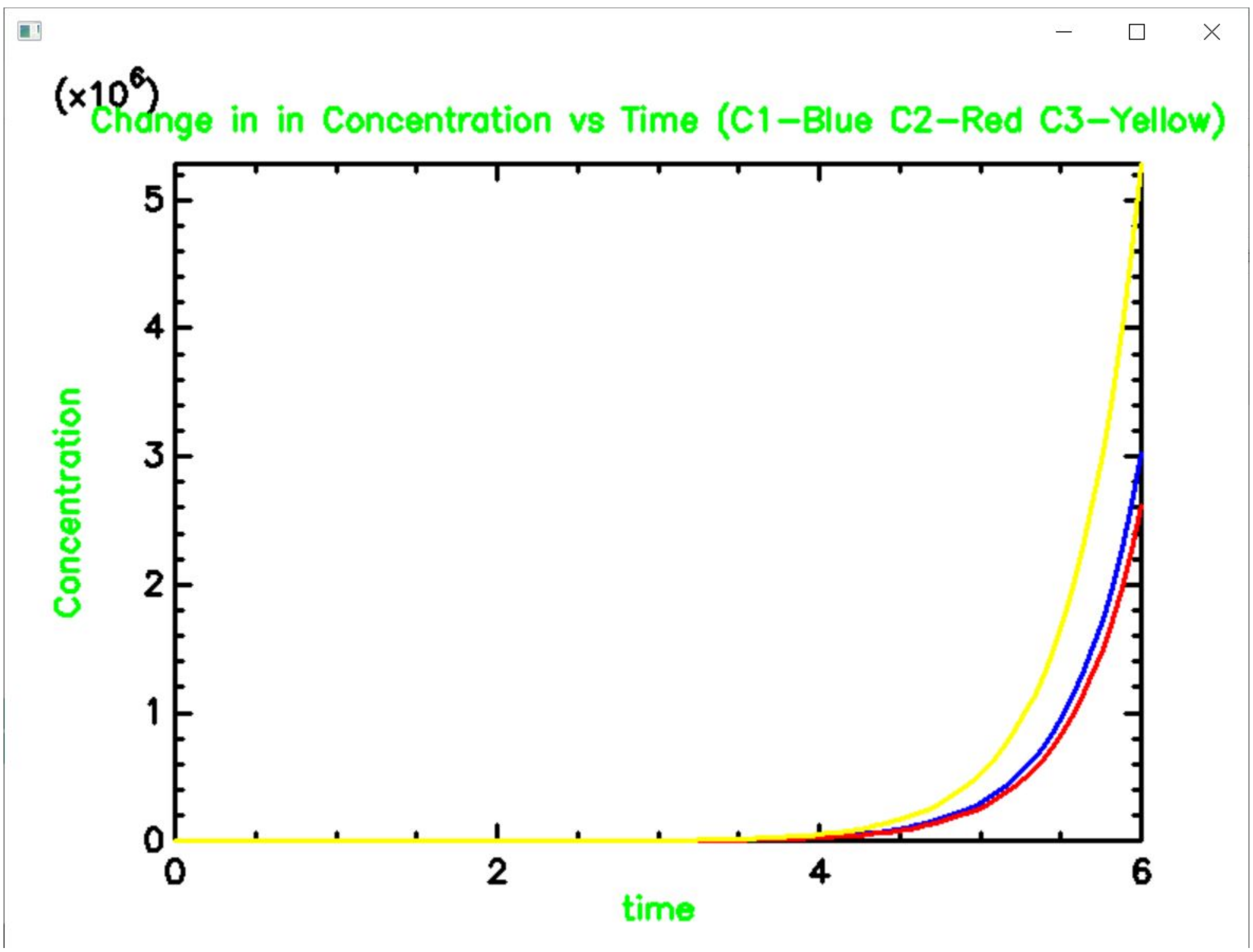
Logic:

The function asks the user if they want to store a data set by prompting the user. If the user chooses to store the data, the data set will be saved to a file. This will allow them to access previous inputs if they are to use the program again.

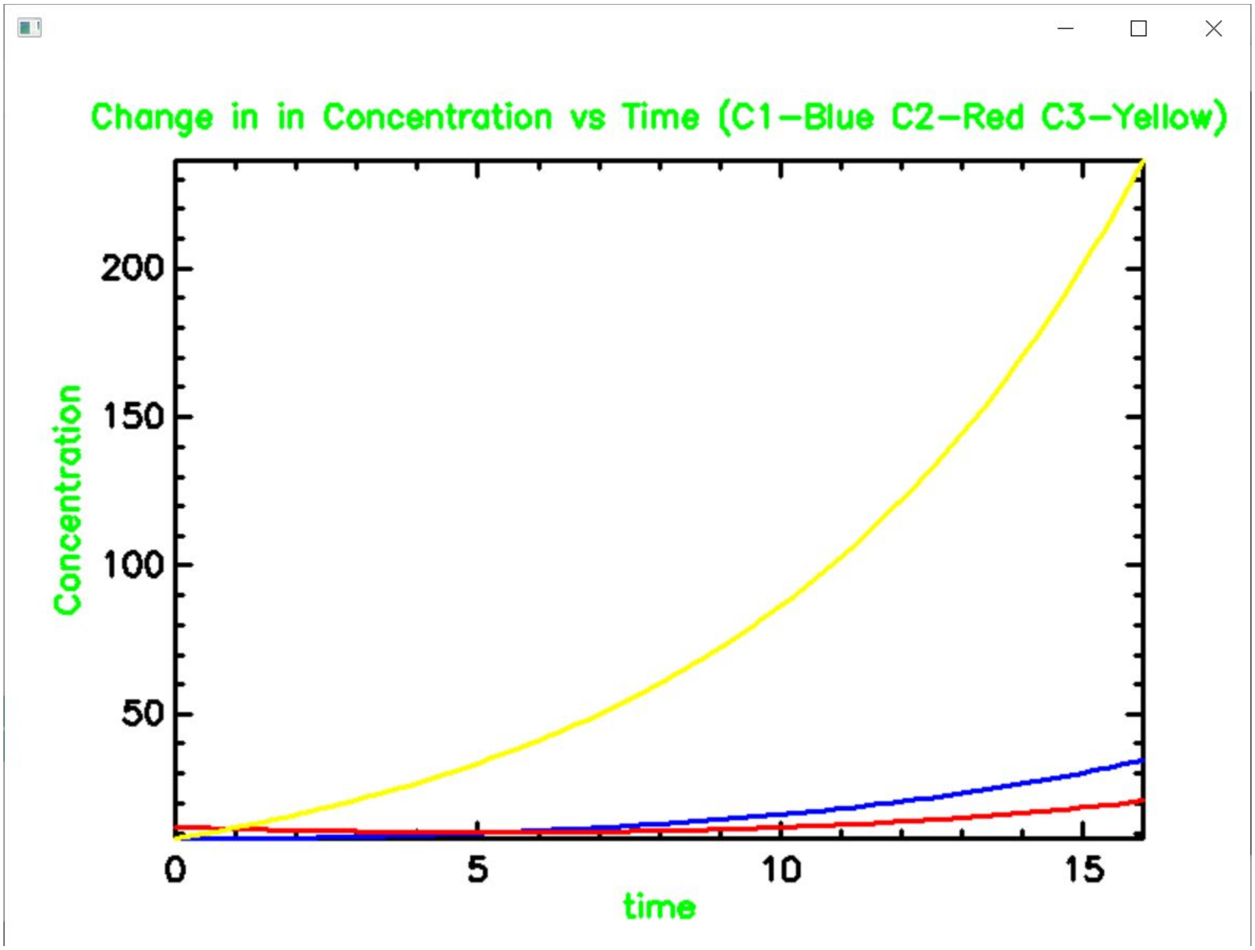
SECTION 4-Implementation

See attached c-file in folders Project_300066276.zip and Project_ 300120566.zip

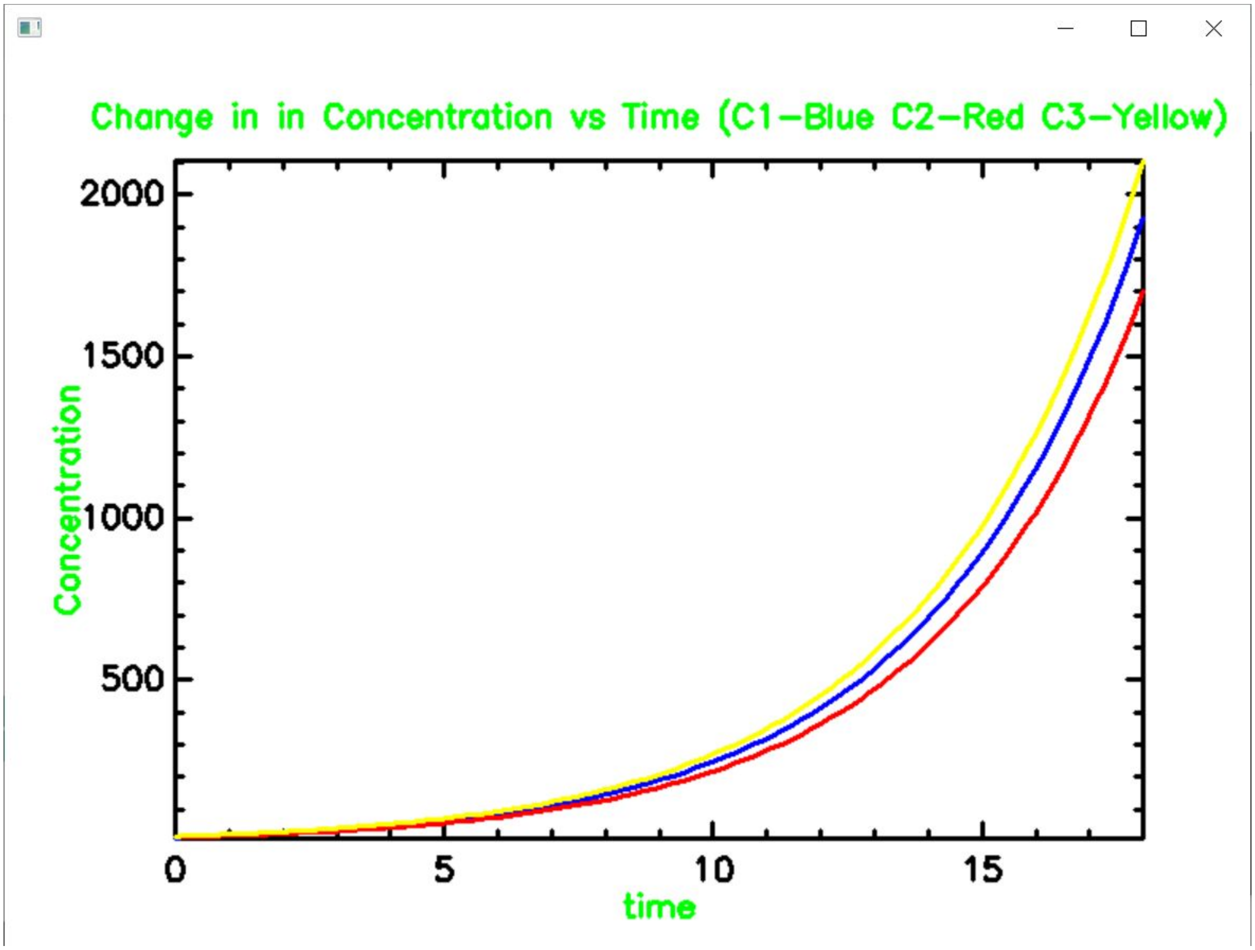
SECTION 5-Software Testing and Verification



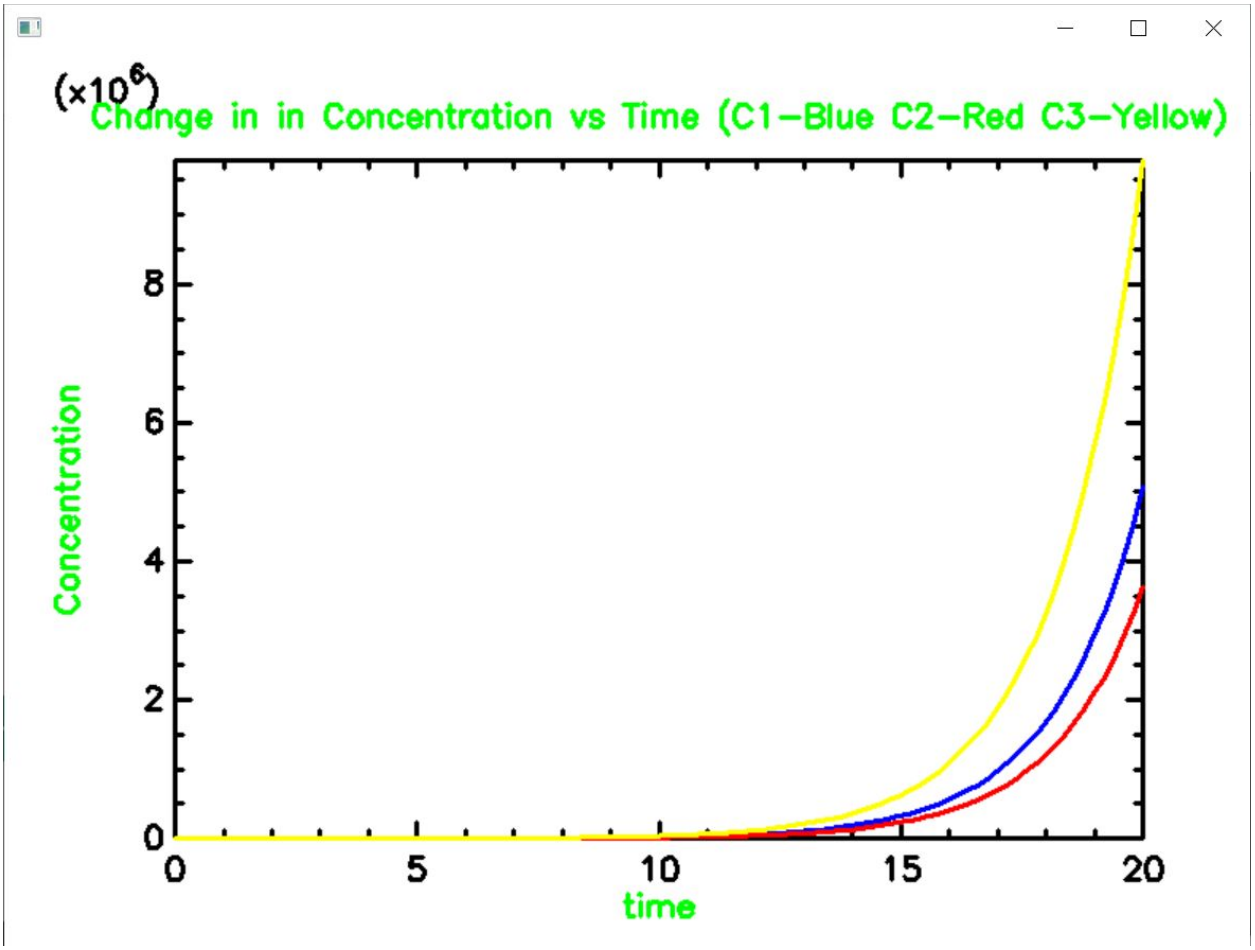
Test 1 Graph


Change in in Concentration vs Time (C1-Blue C2-Red C3-Yellow)
Concentration
200
150
100
50
0
5
10
15
time

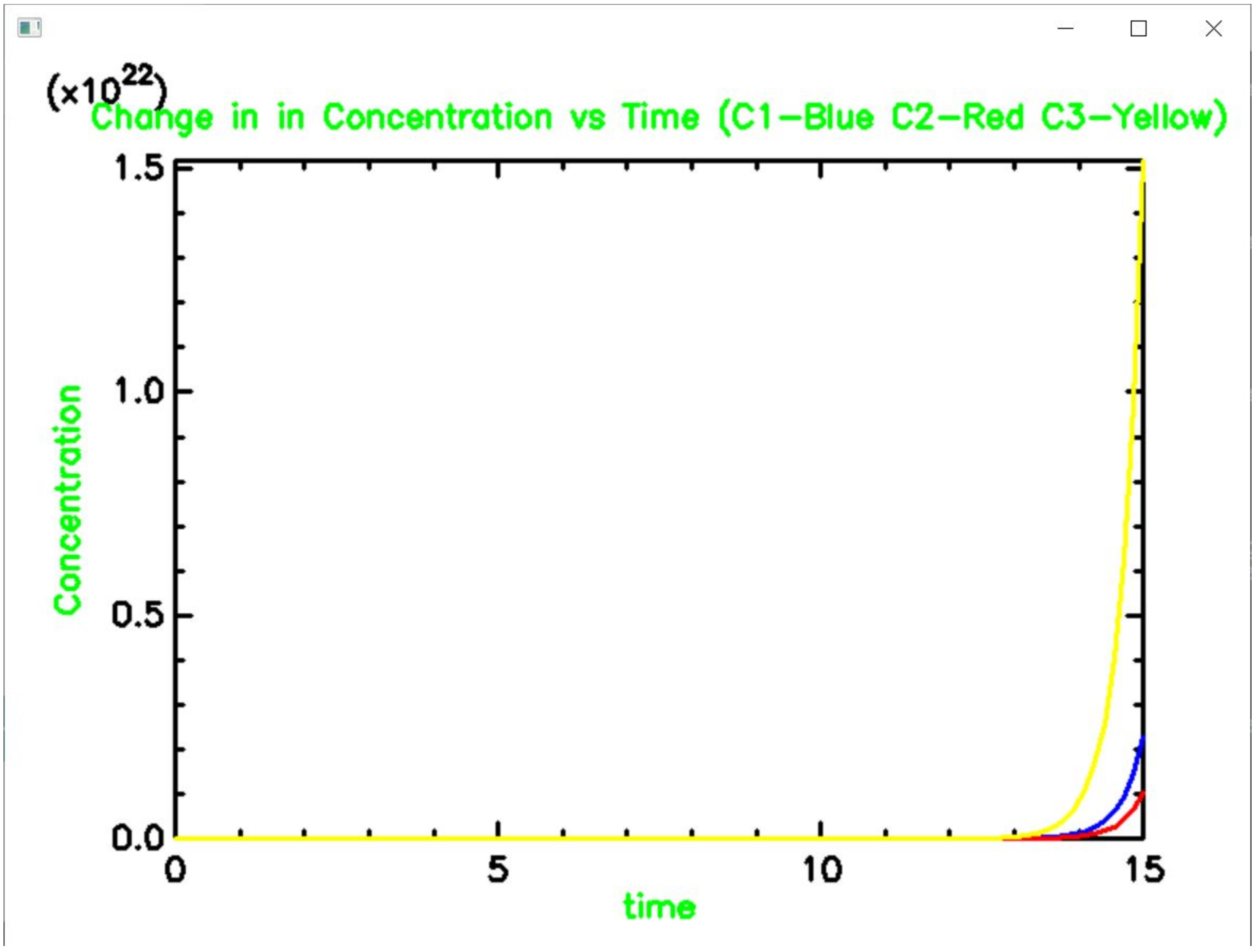
Test 2 Graph


Change in in Concentration vs Time (C1-Blue C2-Red C3-Yellow)
Concentration
2000
1500
1000
500
0
5
10
15
time

Test 3 Graph


(x10^6)
Change in in Concentration vs Time (C1-Blue C2-Red C3-Yellow)
Concentration
time
0
2
4
6
8
0
5
10
15
20

Test 4 Graph


(×10^22)
Change in in Concentration vs Time (C1-Blue C2-Red C3-Yellow)
1.5
1.0
0.5
0.0
Concentration
0
5
10
15
time

Test 5 Graph

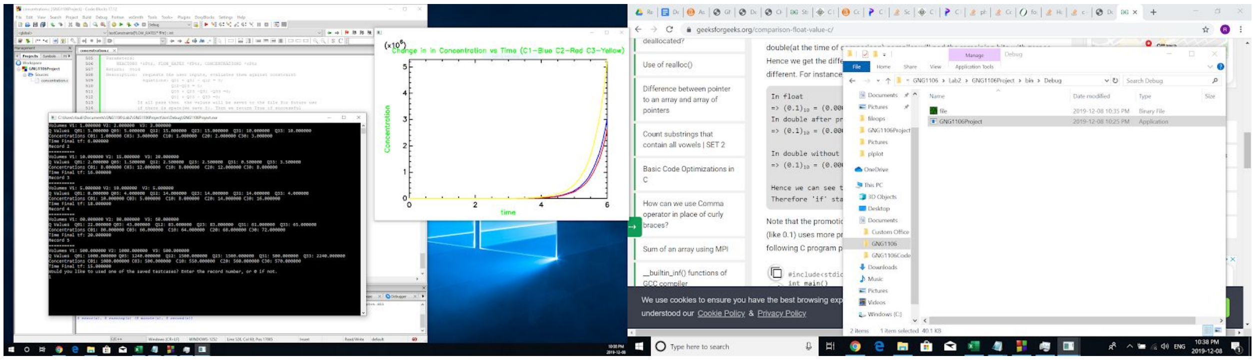
Select C:\Users\rlaub\Documents\GNG1106\Lab2\GNG1106Project\bin\Debug\GNG1106Project.exe

```
Time Final tf: 20.000000
Record 5
==========
Volumes V1: 500.000000 V2: 1000.000000  V3: 500.000000
Q Values  Q01: 1000.000000 Q03: 1240.000000  Q12: 1500.000000  Q23: 1500.000000  Q31: 500.000000  Q33: 2240.000000
Concentrations C01: 1000.000000 C03: 500.000000  C10: 550.000000  C20: 560.000000 C30: 570.000000
Time Final tf: 15.000000
Would you like to used one of the saved testcases? Enter the record number, or 0 if not.
0
Enter C01:
15
Enter C03:
20
Enter C10:
15
Enter C20:
17
Enter C30:
19
Enter the final time tf:
-20
Sorry, time has to be greater than zero
Enter the final time tf:
-20
Sorry, time has to be greater than zero
Enter the final time tf:
20
Enter V1:
0
Sorry, V1 has to be greater than zero
Enter V1:
1
Enter V2:
-15
Sorry, V2 has to be greater than zero
Enter V2:
15
Enter V3:
-20
Sorry, V3 has to be greater than zero
Enter V3:
20
Enter Q01:
5
Enter Q03:
-15
Enter Q12:
25
Enter Q23:
26
Enter Q31:
20
Enter Q33:
20
Sorry, that doesn't satisfy Q12 - Q23 = 0
Sorry, that doesn't satisfy Q03 + Q23 -Q31 -Q33 =0
Sorry, that doesn't satisfy Q01 + Q03 - Q33 =0
Enter C01:
```

Test 6 - Invalid

```
C:\Users\rlaub\Documents\GNG1106\Lab2\GNG1106Project\bin\Debug\GNG1106Project.exe
Time Final tf: 18.000000
Record 4
==========
Volumes V1: 60.000000 V2: 80.000000  V3: 60.000000
Q Values  Q01: 22.000000 Q03: 43.000000  Q12: 83.000000  Q23: 83.000000  Q31: 61.000000  Q33: 65.000000
Concentrations C01: 80.000000 C03: 60.000000  C10: 64.000000  C20: 68.000000 C30: 72.000000
Time Final tf: 20.000000
Record 5
==========
Volumes V1: 500.000000 V2: 1000.000000  V3: 500.000000
Q Values  Q01: 1000.000000 Q03: 1240.000000  Q12: 1500.000000  Q23: 1500.000000  Q31: 500.000000  Q33: 2240.000000
Concentrations C01: 1000.000000 C03: 500.000000  C10: 550.000000  C20: 560.000000 C30: 570.000000
Time Final tf: 15.000000
Would you like to used one of the saved testcases? Enter the record number, or 0 if not.
0
Enter C01:
7
Enter C03:
3
Enter C10:
5
Enter C20:
7
Enter C30:
9
Enter the final time tf:
10
Enter V1:
3
Enter V2:
7
Enter V3:
3
Enter Q01:
7
Enter Q03:
3
Enter Q12:
7
Enter Q23:
8
Enter Q31:
5
Enter Q33:
3
Sorry, that doesn't satisfy Q01 + Q31 - Q12 = 0
Sorry, that doesn't satisfy Q12 - Q23 = 0
Sorry, that doesn't satisfy Q03 + Q23 -Q31 -Q33 =0
Sorry, that doesn't satisfy Q01 + Q03 - Q33 =0
Enter C01:
```

Test 7 - Invalid



Test Using Saved Record