GNG1106 – Fundamentals of Engineering Computation
Course Project

Calculating Concentration of Reactors in a System

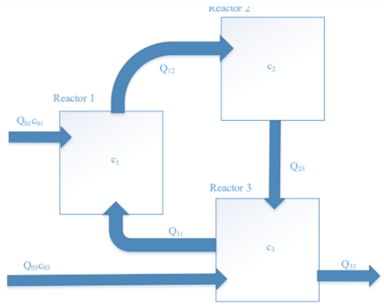
Sam Gilmore – 300050165
Alois Clerc - 300070936

Date: November 5th 2018

1 Problem Identification and Statement

Objective

Develop Software that allows the user to study the transient response (how a concentration of a substance in a reactor changes over time) of the three reactors shown in Figure 2.



(Figure 1)

The reactors are completely mixed reactors, meaning the concentration of a substance is the same in the entire reactor. Additionally, each reactor contains a constant volume given by v1(reactor 1), v2 (reactor 2), v3 (reactor 3). To solve the problem using software, it is necessary to determine what data will be necessary for the program to solve for the concentration in each reactor over time. Test cases are an effective tool for understanding what information is necessary for this task. When the required input and output are known, functions can be developed to prompt and receive the information from the user. A separate function can check that appropriate data has been saved to the program variables. For example, the function getPositiveValue is useful tool for ensuring returned values are greater than zero. A numerical method can be used to calculate the transient response from a variety of inputs. From the project guidelines, Euler's method is provided as an effective numerical method to solve for the system's transient response. With the calculated concentrations for each reactor with respect to time, plplot is an helpful programming tool and can be used to plot the calculated concentrations for each reactor and show the user the concentrations of each reactor based on elapsed time of the reaction given.

2 Gathering of Information and Input/Output Description

Review

The study of chemical reactors is an important part of Chemical Engineering. A big focus of chemical engineering in this field is with the calculating mass balancing of the input and output of a system. For example, if we wish to quantify the rate of mass flow through a reactor; the flow rate is the product of the flow rate Q (m · min)^3 and substance concentration c (Mg · m) for each pipe. Completing the project from a computation standpoint we will have to focus on calculating the mass balance by expressing inputs and outputs in terms of variables and parameters.

Equations

Equations [1] – [3]

Equations [1], [2], and [3] below are the introductory equations for understanding the accumulation of mass in a reactor. Equation [1] gives a very general description of how the accumulation of mass in a reactor is simply the input masses subtracted by the output masses. Equation [2] takes this theory a step further by showing how the volume and derivative of concentration with respect to time are also data that can be computed to solve for accumulation of mass in a reactor. While each of these equations are useful tools for understanding accumulation, they will not be used to solve for the change in concentration in all three reactors.

Equations [3] - [6]
Equation [3] – [6] are similar to the first three equations in the sense that they show a generalization of calculating the change in concentration for the programmer's understanding but will not be used to compute concentration in the final program. Equations [7] – [10] express the flow rate constants necessary to keep the volumes in each reactor constant. These equations will be used in the program input function to ensure the inputted values by the user follow the flow rate constraints and keep the volumes in each reactor constant.

Equations [11] – [13]
Equations [11] and [12] express the general form of Euler's differential equation so that it can be understood by the programmer and implemented into the program to solve for the change in concentration over time. Equation [13] does exactly this and demonstrates how Euler's method should be manipulated to solve for concentration.

General Accumulation Equation: $\text{Accumulation} = \text{inputs} - \text{outputs}$ [1] Description: This is the general expression for accumulation of mass in the reactors. Parameters: Input (unit not specified), Output (unit not specified).
Accumulation of Mass in the Reactor Equation: $\text{Accumulation} = V\frac{dc}{dt}$ [2] Description: Expression for accumulation of mass in the reactor is the constant volume multiplied by the derivative of c with respect to t. The equation represents how the accumulation of a chemical substance changes with respect to change in concentration. Return: Accumulation of mass in the reactor. Parameters: *V* is constant volume, *c* is the reactor concentration.
Change in Concentration Equation: $V\frac{dc}{dt} = Qc_{in} - Qc_0$ [3] Description: Can be used to calculate how the concentration of a substance changes over time. Return: Change of concentration over time. Parameters: *V* is constant volume, *c* is the reactor concentration, Qc_0 is initial concentration at (time t = 0), Qc_{in} is the flow concentration (input flow).

Transition of Concentration in Three Reactors

Equation(s):

(a) $\frac{dc_1}{dt} = \frac{1}{V_1}(Q_{01}c_{01} + Q_{31}c_3 - Q_{12}c_1)$ [4]

(b) $\frac{dc_2}{dt} = \frac{1}{V_2}(Q_{12}c_1 - Q_{23}c_2)$ [5]

(c) $\frac{dc_3}{dt} = \frac{1}{V_3}(Q_{03}c_{03} + Q_{23}c_2 - Q_{31}c_3 - Q_{33}c_3)$ [6]

Description:

Given volumes and concentration for each reactor, the change in concentration in each reactor can be calculated.

Return:

Change in Concentration in the Reactors.

Parameters:

V is the volume in the reactor, $Q_x c_y$ is the flow concentration.

Flow Rate Constraints

Equation(s):

(a) $Q_{01} + Q_{31} - Q_{12} = 0$ [7]

(b) $Q_{12} - Q_{23} = 0$ [8]

(c) $Q_{01} + Q_{23} - Q_{31} - Q_{33} = 0$ [9]

(d) $Q_{01} + Q_{03} - Q_{33} = 0$ [10]

Description:

In order for flow rates to remain constant, flow rates must respect the given constraints.

Parameters:

The flow rates through the channels connecting the reactors

Return:

n/a

Euler's Method Difference Equation

Equation:

$\frac{dx}{dt}_{t=i} = \frac{x_i - x_{i-1}}{\Delta t} = f(x_{i-1})$ [11]

$x_i = x_{i-1} + f(x_{i-1})\Delta t$ [12]

Description:

General format for Euler's Method Difference Equation.

Difference Equation for Change in Reactor One

Equation:

$c_{1,i} = c_{1,i-1} + \frac{Q_{01}c_{01} + Q_{31}c_{3,i-1} - Q_{12}c_{1,i-1}}{V_1}$ [13]

Description:

Equation [13] represents the difference equation for change in reactor 1 concentration and can be manipulated for other reactors. E.g. The concentration ($c_{3,i-1}$) represents the concentration in reactor 3 at time (ti-1). Note: Two other difference equations must be developed from the differential equations 3b and 3c to determine the concentrations c2 and c3. Thus for given values of c1,0, c2,0, and c3,0 at time t0, it is possible to compute all values of c1,i, c2,i, c3,i as time is incremented by Δt (which gives the values of t1, t2, t3, …).

Parameters:

$c_{1,i}$ and $c_{1,i-1}$ represent concentrations of the chemical substance in the reactor at times (t=i) , and (ti-1) respectively, $Q_x c_y$ is the flow concentration, *V* volume in the reactor.

2.4 Input and Output

Input:

Data:

i) Volumes in each reactor: *v1, v2, v3.*

ii) The flows: $Q_{01}, Q_{03}, Q_{12}, Q_{23}, Q_{31}, Q_{33}$.

iii) The input concentrations: c_{01}, c_{03}.

iv) Initial concentrations at time t: 0, $c_{1,0}$, $c_{2,0}$, $c_{3,0}$

v) Final time that that defines range of trial (i.e. between t = 0 and t = t_f.

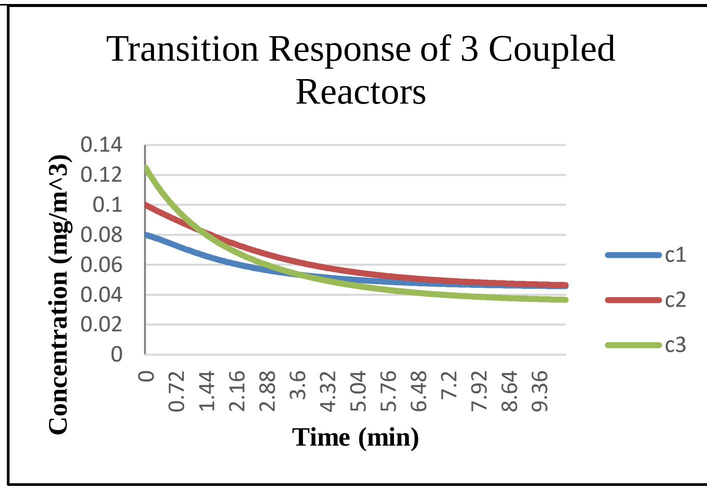
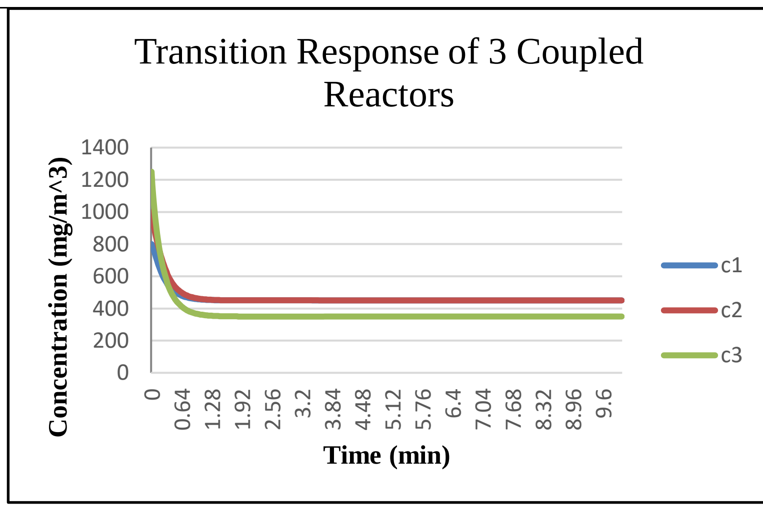
	Data	Variable	Data Type	Type	Restrictions
1	Volumes in each reactor	v1 (reactor 1), v2 (reactor 2), v3 (reactor 3)	Double	Array of Structure REACTOR	Cannot be a negative value
2	Flow through channels	Q_{01} = Q01, Q_{03} = Q03, Q_{12} = Q12, Q_{23} = Q23, Q_{31}= Q31, Q_{33} = Q33	Double	Array of Structure REACTOR	Must respect flow rate constraints
3	Input concentrations	c_{01} = put1 c_{03} = put2	Double	Array of Structure REACTOR	Cannot be a negative value
4	Initial concentrations at time(s)	$c_{1,0}$ = in1, $c_{2,0}$ = in2, $c_{3,0}$ =in3	Double	Array of Structure REACTOR	Cannot be a negative value
5	Final time for range of trial	t_f = tfinal	Double	Variable of Structure REACTOR	Cannot be a negative value
6	Time step for Euler method	deltat	double	Variable of Structure REACTOR	

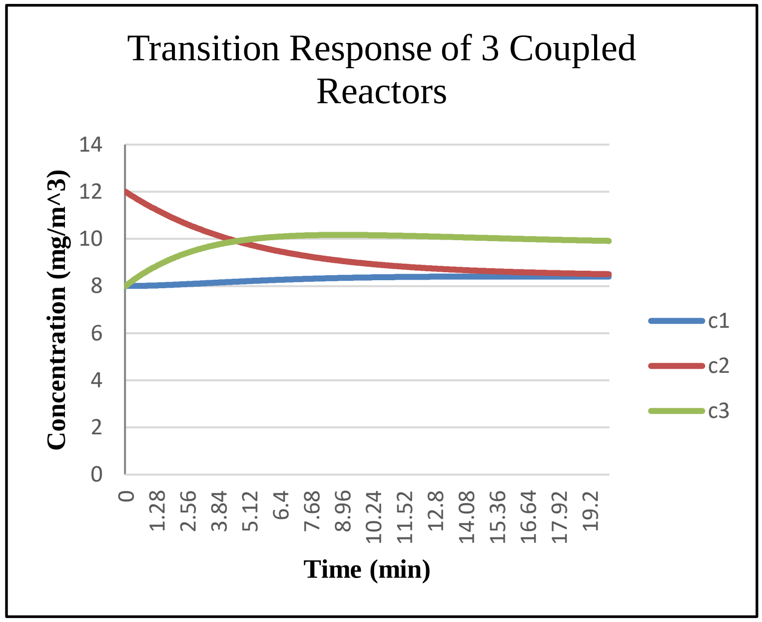
Output:

	Data	Content	Display Method
1	User Input Data	All variables listed in the input table above	Print Statement to Console
2	Prompt user if they want to use a previous save	“To open a previous save enter the save slot number(1-5). For a new save press 0.”	Print Statement to Console
3	Prompt user if they want to save their data	“To save your file enter a save slot number(1-5). To quit without saving press 0.”	Print Statement to Console
4	Concentration Graphs	Three curve expressing the concentration of all three reactors over time on one graph	Matlab program

3 Test Cases and Design

3.1 Test Cases

Test Case	Test Values	Objective	Expected Output
1	Reactor Volumes 1, 2, 3: **0.10, 0.20, 0.30** Channel Flows 01, 31, 12, 23, 31, 33: **0.10, 0.05, 0.15, 0.15, 0.20, 0.10** Input Concentrations 01, 03: **0.05, 0.02** Initial Concentrations 1, 2, 3: **0.08, 0.10, 0.125** Final time: **10.0** Time Step: **0.72**	Testing correct, small values	
2	Reactor Volumes 1, 2, 3: **120.5, 150.8, 340.4** Channel Flows 01, 12, 23, 03, 31, 33: **1000, 500, 1500, 1500, 2000, 1000** Input Concentrations 01, 03: **500, 200** Initial Concentrations 1, 2, 3: **800, 1000, 1200**	Testing correct, large values	

	Final time: **10** Time Step: **0.64**		
3	Reactor Volumes 1, 2, 3: **10, 15, 20** Channel Flows 01, 12, 23, 03, 31, 33: **2, 0.5, 2.5, 2.5, 4, 2** Input Concentrations 01, 03: **8, 12** Initial Concentrations 1, 2, 3: **8, 12, 8** Final time: **20** Time Step: **0.6**	Testing increasing concentration	
4	Reactor Volumes 1, 2, 3: **-120.5, -150.8, -340.4** Channel Flows 01, 12, 23, 03, 31, 33: **-10, -15, -15, -10, -5, -20** Input Concentrations 01, 03: **-500, -200** Initial Concentrations 1, 2, 3: **-800, -1000, -1200** Final time: **-10**	Testing incorrect negative values	"The value must be greater than zero" – this message will display after each negative value in the test case is inputted and will repeatedly display for each until a positive value is entered.
5	Reactor Volumes 1, 2, 3: **120.5, 150.8, 340.4**	Testing incorrect values not following	"Please input flow rates that follow volume constraints, refer to equation 4 of project guidelines and try again." – this message should repeatedly display until proper values are entered.

	Channel Flows 01, 12, 23, 03, 31, 33: **1, 2, 3, 4, 5, 6** Input Concentrations 01, 03: **500, 200** Initial Concentrations 1, 2, 3: **800, 1000, 1200** Final time: **10**	channel flow constants	
6	Reactor Volumes 1, 2, 3: **0, 0, 0** Channel Flows 01, 12, 23, 03, 31, 33: **1000, 1500, 1500, 1000, 500, 2000** Input Concentrations 01, 03: **0, 0** Initial Concentrations 1, 2, 3: **0, 0, 0** Final time: **0**	Testing incorrect 0 values	"The value must be greater than zero" – this message will display after each zero value in the test case is inputted and will repeatedly display for each until a positive value is entered.

3.2 Design

3.2.1 Introduction

The purpose of the program is to plot the concentration over time of three reactors. The program requires variables containing values from the user and a numerical method to solve for the changes in concentration in each reactor. The brain function of they program is the main function and will call all the necessary functions to get to the end result of plotting the concentrations over time. The main function starts by asking if the user would like to load a previously calculated set of data. If the user says no, and wants to make a new calculation, the program will define a structure of type REACTOR. The main function will call on function getUserInput to receive the user input values and implement function getPositiveValue to make sure all given values are greater than zero. The main function will then call function calculateConcentration to calculate changes in concentration for each reactor stored in the structure REACTOR. Finally, a plotting function plotGraph be called by main to plot the concentrations of the three reactors over time. Initially, if a previous set of data is chosen, only the plotting function will be called to plot a saved set of data. In closing, the user will be asked if they would like to save their data in one of five data slots.

3.2.2 Functions for Interacting with the User

Functions in this subsection are designed to interact with the user by collecting and storing data provided by the user in the appropriate variables. The functions getUserInput and getPositiveValue, work in unison to prompt the user for the required data, use algorithms to check the given number is greater than zero and if necessary; inform the user the input is incorrect and repeat the process if requirements are not meant. Finally, the function will store the given value in the appropriate variable.

i) getUserInput (REACTOR *)

Parameters:

- rPtr - reference to REACTOR structure variable.
- vol - array for reactor volumes
- flow - array for flow rate through channels
- inputC - array for input concentrations
- initialC - array for initial concentrations at time(s)
- tfinal - reaction end time

Return Value: Type Void

No values are returned from getUserInput. The function uses a pointer to store the appropriate variable values.

Logic/Algorithm:

This function utilizes pointers and a prompt string to give inputted values to the appropriate variable. getUserInput calls on the function getPositiveValue to ensure the inputted values are greater than zero for the volume, flow rates, input concentration, initial concentration and final time. The function additionally uses a do while loop with a nested if statement to ensure the given flow rates follow the constraints given in equation 4 of the project guidelines. The loop guarantees that the flow rates keep the volume constant throughout all reactors during the reaction.

ii) getPositiveValue(char *)

Parameters:

- value: stores the value passed by pointer from function getUserInput.

Return Value: Type Double

Values that are greater than zero are returned to the appropriate variable in function getUserInput.

Logic/Algorithm:

getPositiveValue uses a do while loop in addition to an if statement to check if the given number is greater than zero. If the number is less than or equal to zero, the function informs the user that the input is incorrect and restarts the input.

3.2.3 Functions for Calculations

The function in this subsection use user inputted values and the Euler equation for each reactor to calculate the concentration over a range of time specified by the user. The function calculateConcentration fills 3 arrays located in the REACTOR structure with concentration values of specific increments of time in order to acquire tables of values that can be plotted into a curve.

iv) calculateConcentrationEuler (REACTOR *)

Parameters:

- rPtr - reference to REACTOR pointer structure variable.
- i – integer for the increment of the for loop
- f_t – variable of type double for storing the concentrations during the calculations

Return Value: Type Void
No values are returned
Logic/Algorithm:
The calculateConcentrationEuler function receives the values obtained in the getUserInput function through a pointer to a structure REACTOR. A for loop is used to fill the time array with the previously found increments of time. For each of these time values, the Euler function in order to place a concentration value for all three reactors in the concentration arrays in the place that corresponds with the time array.

3.2.4 Functions for Plotting

Introduction

vi) plotGraph(REACTOR *rPtr)

Parameters:
- rPtr - reference to REACTOR structure variable.
- vol - array for reactor volumes
- flow - array for flow rate through channels
- inputC - array for input concentrations
- initialC - array for initial concentrations at time(s)
- tfinal - reaction end time

Return Value: Type Void
Function plots 3 different plots on graph.
Logic/Algorithm:
Function plotGraph first calculates the maximum concentration to initialize the size of the graph. plPlot commands are required to initialize the plPlot page and set the device to wingcc, a codeblocks compiler.

vi) getMin(double *array, int n)

Parameters:
- array – reference to an array with double values
- n – number of elements in the array

Return Value: Type Double
Minimum value stored in the array
Logic/Algorithm:
Function getMin uses a for-loop to check all elements in the array (n) and when it finds a value lower than variable ‘min’ it will set the value equal to ‘min’. The function will follow this process for the duration of the loop until ‘min’ has the value for the minimum value in the array.

vii) getMax(double *array, int n)

Parameters:
- array – reference to an array with double values
- n – number of elements in the array

Return Value: Type Double
Maximum value stored in the array
Logic/Algorithm:

Function getMax uses a for-loop to check all elements in the array (n) and when it finds a value greater than variable ‘max’ it will set the value equal to ‘max’. The function will follow this process for the duration of the loop until ‘max’ has the value for the maximum value in the array.

3.2.5 Functions for Files

The functions in this subsection complete two purposes. The getFileValue function is the first function called by the main and gives the user the option to utilize previously saved data in their program and to complete the program using this pre-established data set. The function saveFile is the last function called by the main after the plots have been created. It asks the user if they would like to save their dataset to one of 5 data slots which are saved to a text file.

viii) getFileValue

Logic/Algorithm:
First, the function creates a file pointer, and opens the file for reading at the created file pointer using the fopen(SAVEFILE, “r”) command(with SAVEFILE as a constant containing the name of the file). The function checks if the file exists, and if it doesn’t the program ends. If it does, it utilizes the fread function with the to read the data from the corresponding lines of the corresponding save into the values stored in the structure. The function verifies which reactors are filled using an integer boolean in the structure variable and notifies the user which functions contain data. The function asks the user if they would like to load a file in slots 1-5. If the user selects 0, the function ends. If the user selects a number of 1-5 the function will check if those slots are empty, and if they aren’t it will return the REACTOR structure of the chosen slot.
Parameters:

- ptr – reference to REACTOR structure variable
- saves[5] – array of REACTOR structures

Return Value: Type Int
Returns either the chosen reactor values in the form of a structure REACTOR or it returns a blank reactor structure to be modified. Also returns an integer that states whether a save file has been loaded or not

ix) saveFile

Logic/Algorithm:
The function verifies which reactors in the saves[5] array are filled using an integer boolean in the structure variable and notifies the user which functions contain data. It then asks the user if they would like to save their data into slots 1-5. If the user selects 0, the function ends. If the user selects a number of 1-5 the function will copy the values of the user given structure given by a pointer to the chosen reactor in the saves[5] array. Then, it will open the file for writing at the created file pointer using the fopen(SAVEFILE, “w”) command(with SAVEFILE as a constant containing the name of the file) and utilize the fwrite function to overwrite the file with all of the save information stored in the saves[5] array.
Parameters:

- rPtr - reference to REACTOR structure variable.
- saves[5] – array of REACTOR structures

Return Value: Type Void
No values are returned by this function

4 Implementation

See the C program in file ***Deliverable3CompleteCode.c***

5 Software Testing and Verification

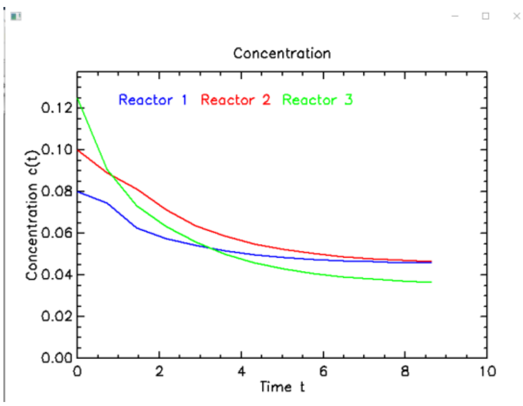
Software Testing: Executed Test Case Output Results

5.1 Valid User Input Cases

Test Case 1:

Console Input:
Reactor Volumes 1, 2, 3: **0.10, 0.20, 0.30**
Channel Flows 01, 31, 12, 23, 31, 33: **0.10, 0.05, 0.15, 0.15, 0.20, 0.10**
Input Concentrations 01, 03: **0.05, 0.02**
Initial Concentrations 1, 2, 3: **0.08, 0.10, 0.125**
Final time: **10.0**
Time Step: **0.72**

Graph Output:
Console Output: N/A

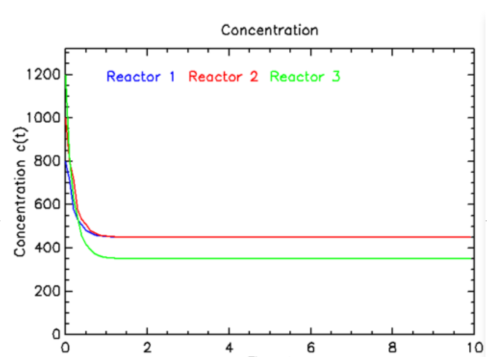


File Input: Test case saved in slot 1

Test Case 2:

Console Input:
Reactor Volumes 1, 2, 3: **120.5, 150.8, 340.4**
Channel Flows 01, 12, 23, 03, 31, 33: **1000, 500, 1500, 1500, 2000, 1000**
Input Concentrations 01, 03: **500, 200**
Initial Concentrations 1, 2, 3: **800, 1000, 1200**
Final time: **10**
Time Step: **0.64**

Graph Output:
Console Output: N/A

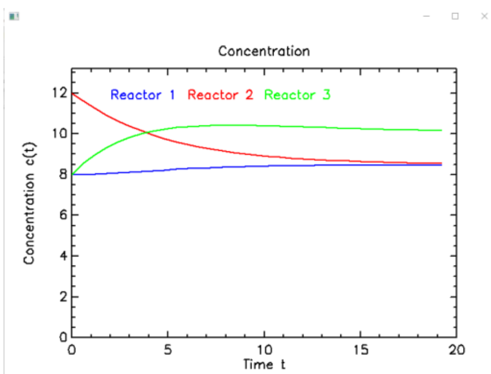


File Input: Test case saved in slot 2

Test Case 3:

Console Input:
Reactor Volumes 1, 2, 3: **10, 15, 20**
Channel Flows 01, 12, 23, 03, 31, 33: **2, 0.5, 2.5, 2.5, 4, 2**
Input Concentrations 01, 03: **8, 12**
Initial Concentrations 1, 2, 3: **8, 12, 8**
Final time: **20**
Time Step: **0.6**

Graph Output:
Console Output: N/A

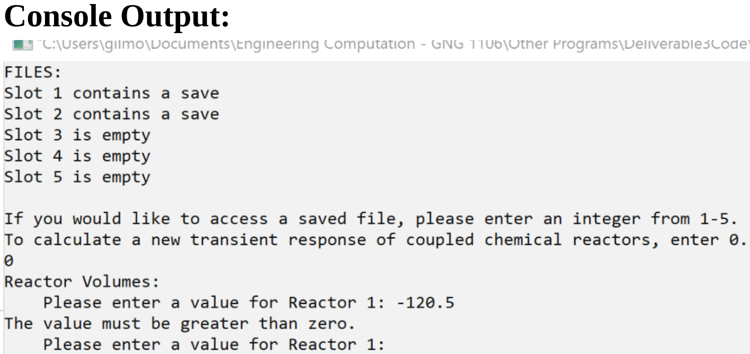


File Input: Test case saved in slot 3

5.2 Invalid User Input Cases
Test Case 4:

Console Input:
Reactor Volumes 1, 2, 3: **-120.5, -150.8, -340.4**
Channel Flows 01, 12, 23, 03, 31, 33: **-10, -15, -15, -10, -5, -20**
Input Concentrations 01, 03: **-500, -200**
Initial Concentrations 1, 2, 3: **-800, -1000, -1200**
Final time: **-10**

Graph Output: N/A
Console Output:



File Input: N/A

Test Case 5:

Console Input:
Reactor Volumes 1, 2, 3: **120.5, 150.8, 340.4**
Channel Flows 01, 12, 23, 03, 31, 33: **1, 2, 3, 4, 5, 6**
Input Concentrations 01, 03: **500, 200**
Initial Concentrations 1, 2, 3: **800, 1000, 1200**
Final time: **10**

Graph Output: N/A
Console Output:

```
FILES:
Slot 1 contains a save
Slot 2 contains a save
Slot 3 is empty
Slot 4 is empty
Slot 5 is empty

If you would like to access a saved file, please enter an integer from 1-5.
To calculate a new transient response of coupled chemical reactors, enter 0.
0
Reactor Volumes:
    Please enter a value for Reactor 1: 120.5
    Please enter a value for Reactor 2: 150.8
    Please enter a value for Reactor 3: 340.4
Flow Rates:
    Please enter a value for Channel Q-01: 1
    Please enter a value for Channel Q-31: 2
    Please enter a value for Channel Q-12: 3
    Please enter a value for Channel Q-23: 4
Channel Q-23's flow rate must be equal to Channel Q-12's flow rate to follow constraints. Please try again.
    Please enter a value for Channel Q-23:
```

File Input: N/A

Test Case 6:

Console Input:
Reactor Volumes 1, 2, 3: **0, 0, 0**
Channel Flows 01, 12, 23, 03, 31, 33: **1000, 1500, 1500, 1000, 500, 2000**
Input Concentrations 01, 03: **0, 0**
Initial Concentrations 1, 2, 3: **0, 0, 0**
Final time: **0**

Graph Output: N/A
Console Output:

```
FILES:
Slot 1 contains a save
Slot 2 contains a save
Slot 3 is empty
Slot 4 is empty
Slot 5 is empty

If you would like to access a saved file, please enter an integer from 1-5.
To calculate a new transient response of coupled chemical reactors, enter 0.
0
Reactor Volumes:
    Please enter a value for Reactor 1: 0
The value must be greater than zero.
    Please enter a value for Reactor 1: 0
The value must be greater than zero.
    Please enter a value for Reactor 1: 0
The value must be greater than zero.
    Please enter a value for Reactor 1:
```

File Input: N/A