Synopsis

Chemical Reactor

Abstract

Dynamic system are those systems in which the state or the variables vary continuously with time and can be generally be described by means of differential equation.

In a chemical reactor when two substances A and B are brought together they produce the third chemical substance C. it will be known that if 1gm of substance A combined with the 1gm of substance B to produce 2gm of substance C. Furthermore the rate of formation C is the proportional to the product of amounts of A and B present.

In additional to forward reaction there is also a backward reaction decomposing c back into A and B. The rate of decomposition of C is proportion to the amount of C present in the mixer. In other words at anytime if A, B and C present respectively then their rates of increases are described by the differential equation.

Objective of the Project

The major objective of this software is to make the user understand how a typical process of Chemical reaction works.

Another objective is to simulate the affect of reactant concentration and reaction time on the reaction process and on the product concentration generated.

Hence the objectives of the project could be classified as given below:

1. To simulate how a process of chemical reaction work.
2. Easy to use and understand.
3. To make it user friendly.

Scope of the Project

The project has been developed keeping in view the requirements of the end user. The utility has been developed for a dynamic environment in a structured manner to enable later modifications easily.

Our project is concerned with chemical reactions and their simulation. For this we have to understand about some basic characteristics reactions like how they proceed, affect of time, concentration and some constants on them etc. Hence some information is provided about these basic aspects regarding a chemical reaction.

Hardware & Software Requirements

Hardware Configuration

❖ Processor	Pentium -3 750MHZ
❖ Hard disk	20GB
❖ RAM	256MB
❖ VGA	16bit
❖ Keyboard	108 keys

Software Requirements

❖ Language	C with graphics
❖ Operating system	Windows 98 and above

Chapter 1

Introduction

1.1 Specification Definition of System Type

Chemical reaction is a process in which one or more substances, the reactants, are converted to one or more different substances, the products. Substances are either chemical elements or compounds. A chemical reaction rearranges the constituent atoms of the reactants to create different substances as products.

Chemical reactions are an integral part of technology, of culture, and indeed of life itself. Burning fuels, smelting iron, making glass and pottery, brewing beer, and making wine and cheese are among many examples of activities incorporating chemical reactions that have been known and used for thousands of years. Chemical reactions abound in the geology of Earth, in the atmosphere and oceans, and in a vast array of complicated processes that occur in all living systems.

A chemical reaction is a process that is usually characterized by a chemical change in which the starting materials (reactants) are different from the products. Chemical reactions tend to involve the motion of electrons, leading to the formation and breaking of chemical bonds. There are several different types of chemical reactions and more than one way of classifying them. Some types of chemical reactions are:

Direct Combination or Synthesis Reaction

In a synthesis reaction two or more chemical species combine to form a more complex product.

$A + B \rightarrow AB$

The combination of iron and sulphurs form iron (II) sulphide is an example of a synthesis reaction:

$8\ Fe + S_8 \rightarrow 8\ FeS$

Chemical Decomposition or Analysis Reaction

In a decomposition reaction a compound is broken into smaller chemical species.

$AB \rightarrow A + B$

The electrolysis of water into oxygen and hydrogen gas is an example of a decomposition reaction:

$2\ H_2O \rightarrow 2\ H_2 + O_2$

1.2 Problem Statement

The problem statement of this software is to make the user understand how a typical process of Chemical reaction works.

Another objective is to simulate the affect of reactant concentration and reaction time on the reaction process and on the product concentration generated. Hence the objectives of the project could be classified as given below:

1. To simulate how a process of chemical reaction work.
2. Easy to use and understand.
3. To make it user friendly.

1.3 Specification Definition of Model Type

In a chemical reactor when two substances A and B are brought together they produce the third chemical substance C. it will be known that if 1gm of substance A combined with the 1gm of substance B to produce 2gm of substance C. Furthermore the rate of formation C is the proportional to the product of amounts of A and B present.

In additional to forward reaction there is also a backward reaction decomposing c back into A and B. The rate of decomposition of C is proportion to the amount of C present in the mixer. In other words at anytime if A, B and C present respectively then their rates of increases are described by the differential equation.

The project has been developed keeping in view the requirements of the end user. The utility has been developed for a dynamic environment in a structured manner to enable later modifications easily.

Our project is concerned with chemical reactions and their simulation. For this we have to understand about some basic characteristics reactions like how they proceed, affect of time, concentration and some constants on them etc. Hence some information is provided about these basic aspects regarding a chemical reaction.

.

Chapter 2

Literature Survey

2.1 Brief Description of the Application

Continuous dynamic systems are those systems in which the state or the variables vary continuously with time and can be generally described by means of differential equation. In a chemical substance c. it is well known that if 1gm of A combines with 1gm of B to produce 2 gm of C.

Furthermore the rate of formation of c is proportional to the product of the amounts of A and B present. In additional to forward reaction there is also a backward reaction decomposition C is proportional to the amount of C present in the mixer. In other words at any time if A, B and C present respectively then their rates of increases are described by the following three differential equation.

$$da/dt = k2c - k1ab$$

$$db/dt = k2c - k1ab$$

$$dc/dt = 2k1ab - 2k2c$$

Where k1 and k2 are the rate constants. And these constants will vary with temperature and pressure, but we do not allow the temperature or pressure of the reaction to vary given the values of the constants k1 and k2 and the initial quantities of the chemical A and B and C=0. Let us determine how much of C has been produced reactions is important in many industrial processes.

A straight forward method of simulating this system is to start at time zero and increment time in small steps of Δt. We assume that the quantities of chemical remain un altered during each step and only change instantaneously at the end of the step. Thus the quality A(B or C) at the end of one such step is given in terms of the quality at the beginning of the step as

$$A(t+\Delta t) = A(t)$$

If Δt is sufficiently small eq(4) is a reasonable representation. Identical equations can be written for b(t+Δt). Suppose we wish to simulate the system for a period T. we will divide this period T in to a large number of N of small periods Δt.

i.e;

$$T = N, \Delta T$$

At time zero we know a(0), b(0),c(0). From these initial values k1 and k2 we compute the amount of chemicals at time Δt as,

$A(\Delta t) = a(0)+[k2,c(0) – k1, .a(0).b(0)]\ \Delta t$

$B(\Delta t) =b(0)+[k2.c(0) – k1, .a(0).b(0)]\ \Delta t$

$C(\Delta t) = c(0)+[2k1.a(0).b(0) – 2k2,c(0)]\ \Delta t$

Using these values we calculate the next stage of the system i.e: at time 2t as

$A(2\Delta t) = a(\Delta t)+[k2,c(\Delta t) – k1, .a(\Delta t).b(\Delta t)]\ \Delta t$

$B(2\Delta t) =b(\Delta t)+[k2.c(\Delta t) – k1, .a(\Delta t).b(\Delta t)]\ \Delta t$

$C(2\Delta t) = c(\Delta t)+[2k1.a(\Delta t).b(\Delta t) – 2k2,c(\Delta t)]\ \Delta t$

Using the state of the system at Δt we determine its state at 3 Δt, and so on. We continue in this vein, moving time forward by Δt and determining the state of system from the previous state, for N steps at the end of which we have the desired results.

2.2 Introduction of Simulation & Modelling

Simulation is a representation of reality through the use of a model or other device which will react in the same manner as reality under a given set of conditions. A model is a representation of our system for the purpose of studying the functioning of the system. A simulation technique is the imitation of the operation of a real world process or system over time. Whether done by hand or on a computer, simulation generates artificial systems so that we can study them and draw important conclusions with respect to the real system. The behavior of a system as it evolves over time is studied by developing a simulation model. This model usually takes the form of a set of assumptions concerning the operation of the system. These assumptions are expressed in mathematical, logical and symbolic relationships between entities or objects of interest in the system.

Simulation can also be used to study systems in the design state, before such systems are built. Thus, simulation modeling can be used both as an analysis tool predicting the effect of changes to the existing systems, and design tool to predict the performance of new system under varying sets of circumstances. From the simulation, data are collected as if real system is observed. This simulation generated data is used to estimate the measures of performance of the system. Thus representation of reality, which may be either in physical form or in a mathematical equations form, is called SIMULATION.

"Before we proceed further, it becomes necessary to define the term simulation in suitable forms:"

Definition 1:

"Simulation is a representation of reality through the use of a model or other device which will react in the same manner as reality under a given set of conditions".

Definition 2:

"Simulation is the use of a system model that has designed the characteristics of reality in order to produce the essence of the actual operation".

Types of Simulation: - Simulation is of two types

1. ANALOGUE OR ENVIRONMENTAL SIMULATION:

The simulation of reality in the physical form is called as analogue simulation

2. COMPUTER OR SYSTEM SIMULATION:

The mathematical model is developed for the system, for this model a computer program is developed. Using a super-fast electronic computer solves this program. Such a simulation is called system simulation. There are two types of system simulation namely,

a) CONTINUOUS SYSTEM SIMULATION

   A continuous system is one which the predominant activities of the system cause smooth changes in the attributes of the system entities. When such a system is modelled mathematically, the variables of the model representing the attributes are controlled functions.

   Ex: Single server queuing system.

b) DISCRETE SYSTEM SIMULATION

Discrete system simulation refers to the modelling of a system as it evolves over time by a representation in which the state variable changes instantaneously at separate points in time. These points in time are the ones at which an event occurs, where a event can be defined as an instantaneous occurrence that may change the state of the system.

Ex: Pure pursuit problem

Process of Simulation Study

Simulation is very general method of studying problem therefore formal procedure can be given for showing how a simulation study will proceed. There is not even a simple way of deciding whether to simulate or not or if simulation is used, whether to use a continuous steps involved in the progress of simulation a study are illustrated by the flowchart.

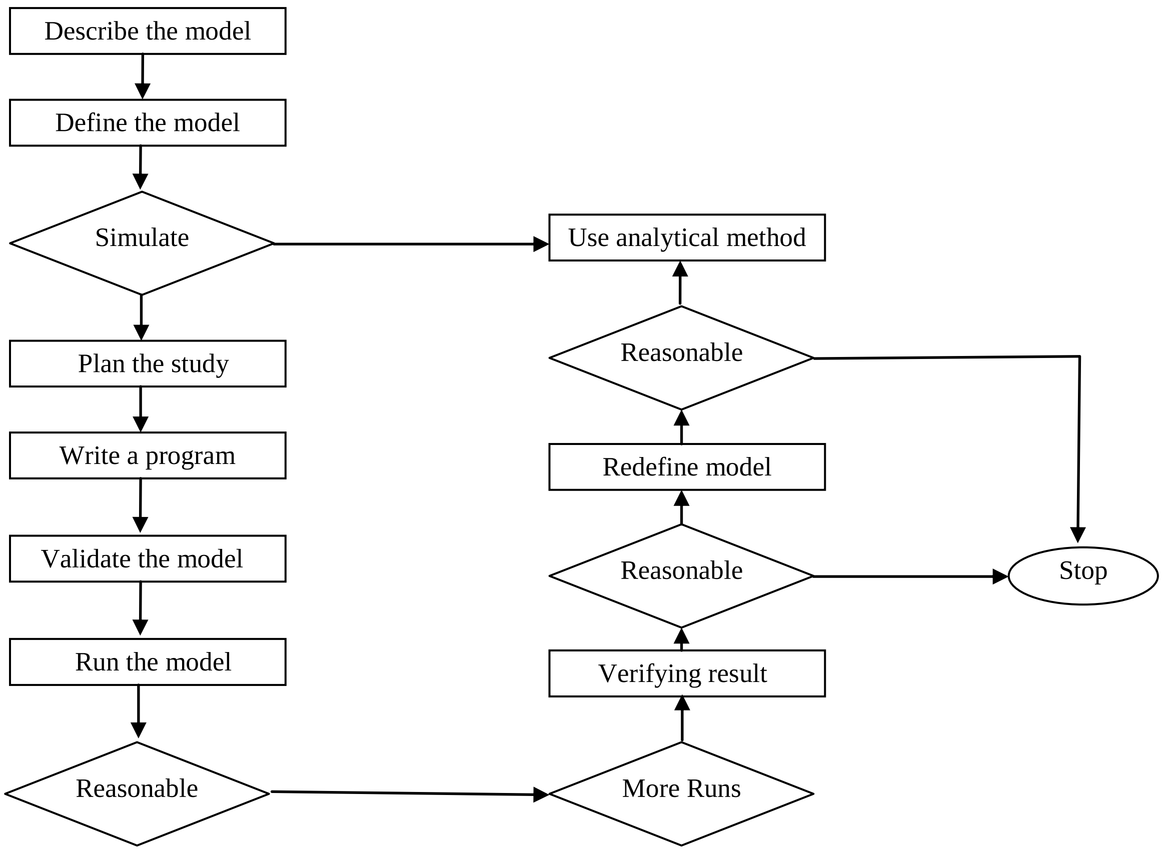


Figure: 3.1 Simulation Study

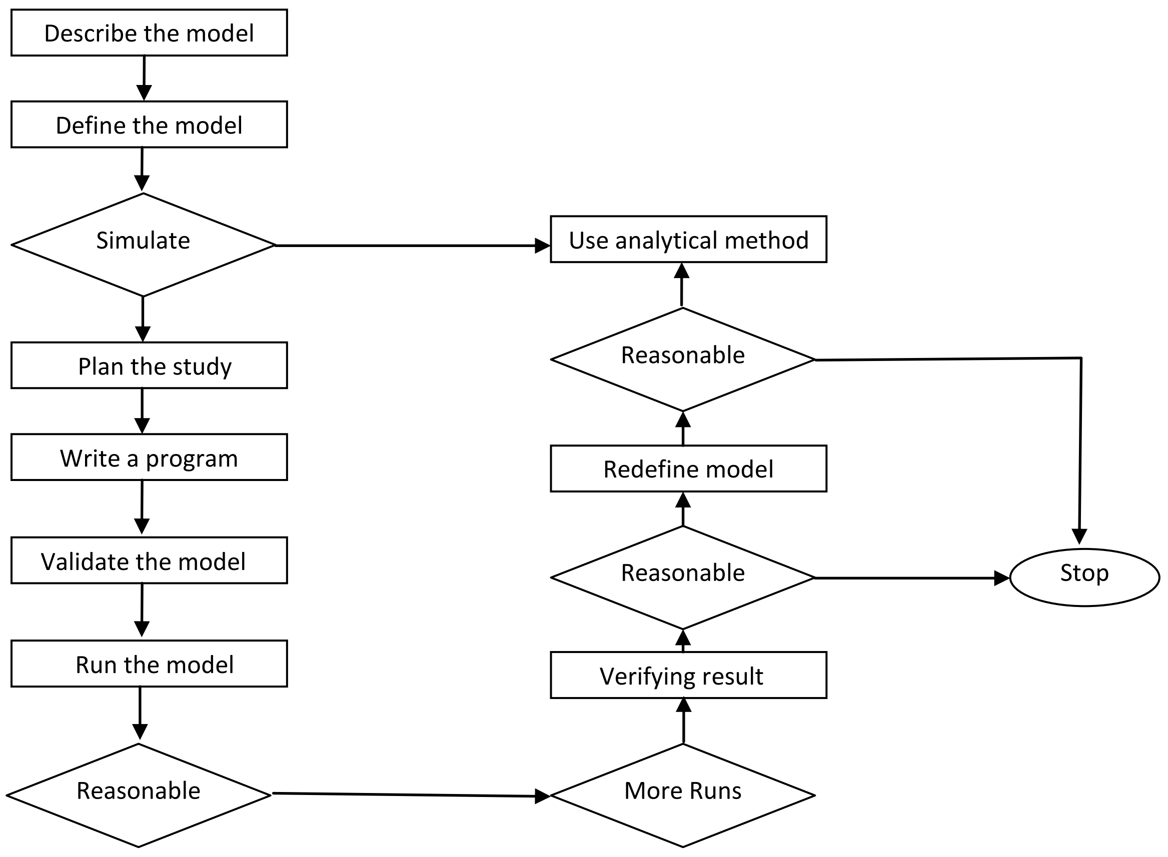


Figure: 3.1 Simulation Study

Models have to be made for the systems which, we desire to study

a) **Deterministic models**

In these models, input and output variables are not permitted to be random variables and the models are described by exact functional relationship.

b) **Stochastic models**

In these models, at least one of the variables or functional relationship is given by probability functions.

c) **Static models**

These models do not take variables time into consideration.

d) **Dynamic models**

These models deal with time varying interactions.

When Simulation is Appropriate

All of us in our daily lives encounter problems. Some problems although having mathematical nature are too complex to be solved. Sometimes the system itself cannot be developed properly. Simulation provides an alternative now, which is cheap and fast and fills the gap between exact analysis and the physical requirement.

- Simulation is used in science and engineering research when developing physical models like network analyzers, river basin models, air fact simulators and laboratory experiments.
- Simulation is used in very important fields like biology (the science of life), medicine, economics and mathematical areas where large problems involving thousands of variables are solved.
- Simulation is used by business executives as there are many problems faced by management which cannot be solved by even advanced concepts like operational research, dynamic programming and research tools.

Simulation models designed for training, allow learning without cost that is experiment study of the model and which is as good as studying the real system.

When Simulation is Not Appropriate

Simulation cannot be used always but when the need arises and simulation is the only resort only then the procedure should be undertaken.

- Simulation should not be used when the problem can be tackled by commonsense.

- Simulation must never be used if the problem can be solved analytically.
- Direct experiments performed must not be simulated.
- If the costs exceed savings then simulation is majority inappropriate.
- Simulation must not be performed if there are no proper resources or time.

Advantages of Simulation

- Simulation helps to learn about a real system, without having the system at all.
- Many managerial decision making problems are too complex to be solved by mathematical programming.
- In the real system, the changes we want to study may take place too slowly or too fast to fast be observed conveniently.
- Computer simulation can compress the performance of a system over years in few minutes of computer running time. Conversely, in systems like nuclear reactors where millions of events take place per second, simulation can expand the time to required level.
- Through simulation, management can foresee the difficulties and bottlenecks, which may come up due to the introduction of new machines, equipments and processes. It thus eliminates the need of costly trial and error method to accommodate the changing environment to the real situation.
- The operating personnel and non-technical mangers can easily understand simulation being relatively free from mathematics. This helps in getting the proposed plans accepted and implemented. Simulation technique is easier to use than the mathematical models, and can be used for wide range of situations.
- Extensive computer software packages are available, making it very convenient to use fairly sophisticated simulation models.
- Simulation is a very good tool of training and has advantageously been used for training the operating and management stuff in the operation of complex systems.

Limitations of Simulation

- Simulation does not produce optimum results. When the model deals with uncertainties, the results of simulation are only reliable estimates subject to statistical errors.

- Quantification of the variables is another difficulty. In a number of situations, it is no possible to quantity all the variables that affect the behaviour of the system.
- In large and complex problems, the large number of variables, and the interrelationships between them make the problem very unwieldy.
- Simulation is by no means a cheap method of analysis. Even small simulations take considerable amount of computer time. In a number of situations, simulation is comparatively costlier and time consuming.
- Other important limitation stem from too much tendency to rely on the simulation models. This results in applications of the technique to some simple situations, which can more appropriately be handled by other techniques of mathematical programming.

Chapter 3

System Specification & Design

3.1 Modules Features

Simulations in chemical reaction engineering are used for different reasons during the investigation and development of a reaction process or system. In the initial stages, they are used to dissect and understand the process or system. By setting up a model and studying the results from the simulations, engineers and scientists achieve the understanding and intuition required for further innovation. Once a process is well understood, modelling and simulations are used to optimize and control the process' variables and parameters. These 'virtual' experiments are run to adapt the process to different operating conditions.

Another possible use for modelling is to simulate scenarios that may be difficult to investigate experimentally. One such example of this is to improve safety, such as when an uncontrolled release of chemicals occurs during an accident. Simulations are used to develop precautions for preventing or containing the impact from these hypothetical accidents. In all these cases, modelling and simulations provide value for money by reducing the need for large numbers of experiments or building prototypes, while, potentially, granting alternate and better insights into a process or design.

Modelling Mass and Energy Balances and Chemical Reactions

A plate reactor where chemical reactions occur throughout and reacting chemicals are introduced at two points in the reactor.

Perfect for All Unit Operations in Chemical & Process Industries

Optimizing chemical reactors, filtration equipment, mixers, and other processes is made easy with the Chemical Reaction Engineering Module. It contains the tools for you to simulate material transport and heat transfer together with arbitrary chemical kinetics in all types of environments - gases, fluid flow, porous media, on surfaces, and within solid phases - or combinations of all of these. This makes it perfect for all facets of the chemical and process industries and even within environmental engineering where the "process unit" or "chemical reactor" is the environment surrounding you.

Convection & Diffusion with Arbitrary Chemical Kinetics

The Chemical Reaction Engineering Module contains intuitive user interfaces for you to define material transport in dilute and concentrated solutions or mixtures through convection, diffusion, and ionic migration of an arbitrary number of chemical species. These are easily connected to definitions of reversible, irreversible, and equilibrium reaction kinetics that can be described by the Arrhenius equation, or any arbitrary rate law, where the effects of concentration and temperature on the kinetics can be included. The interface for defining chemical reactions is straightforward as chemical formulas and equations are entered essentially as you would write them on paper. COMSOL sets up the appropriate reaction expressions using the mass action law, which you can alter or override with your own kinetic expressions. The stoichiometry in your reaction formulas is used to automatically define mass and energy balances, whether they are homogeneous or heterogeneous, occurring in bulk or on surfaces.

Complete Transport Phenomena

Tools for calculating thermodynamics properties, including from external sources, are included in the Chemical Reaction Engineering Module so as to augment the coupling of heat transport and enthalpy balances to your material transport and chemical reactions. User interfaces for defining momentum transport are also available for you to consider the complete description of your process' transport phenomena. This includes laminar and porous media flow described by the Navier-Stokes equation, Darcy's Law, and the Brinkman Equations. By coupling the CFD Moduleor Heat Transfer Moduleto you’re modelling, you are also able to incorporate turbulent flow, multiphase flow, and non isothermal flow, as well as radiation heat transfer.

An Integral Part of Optimizing Your Chemical Reaction Processes

The Chemical Reaction Engineering Module is applicable to engineers and scientists working within the chemical, process, electric power, pharmaceutical, polymer, and food industries where material transport and chemical reaction are integral to the process you are working with. It provides tools to study all facets of these applications: from test tube studies in a lab, to an overhaul of a chemical reactor in the middle of a plant. Your chemical kinetics can be intrinsically simulated in controlled environments to accurately describe your chemical kinetics using built-in features for parameter estimation and comparison to experimental data. From here, the Chemical Reaction Engineering Module provides a number of pre-defined reactor types for more involved studies:

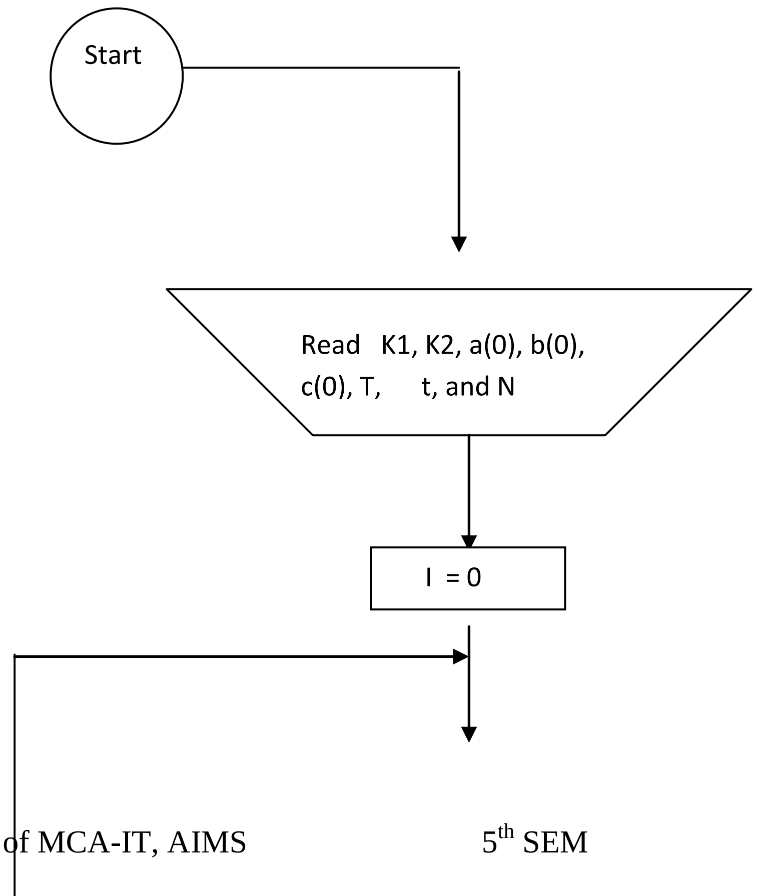
Batch and Semi batch Reactors

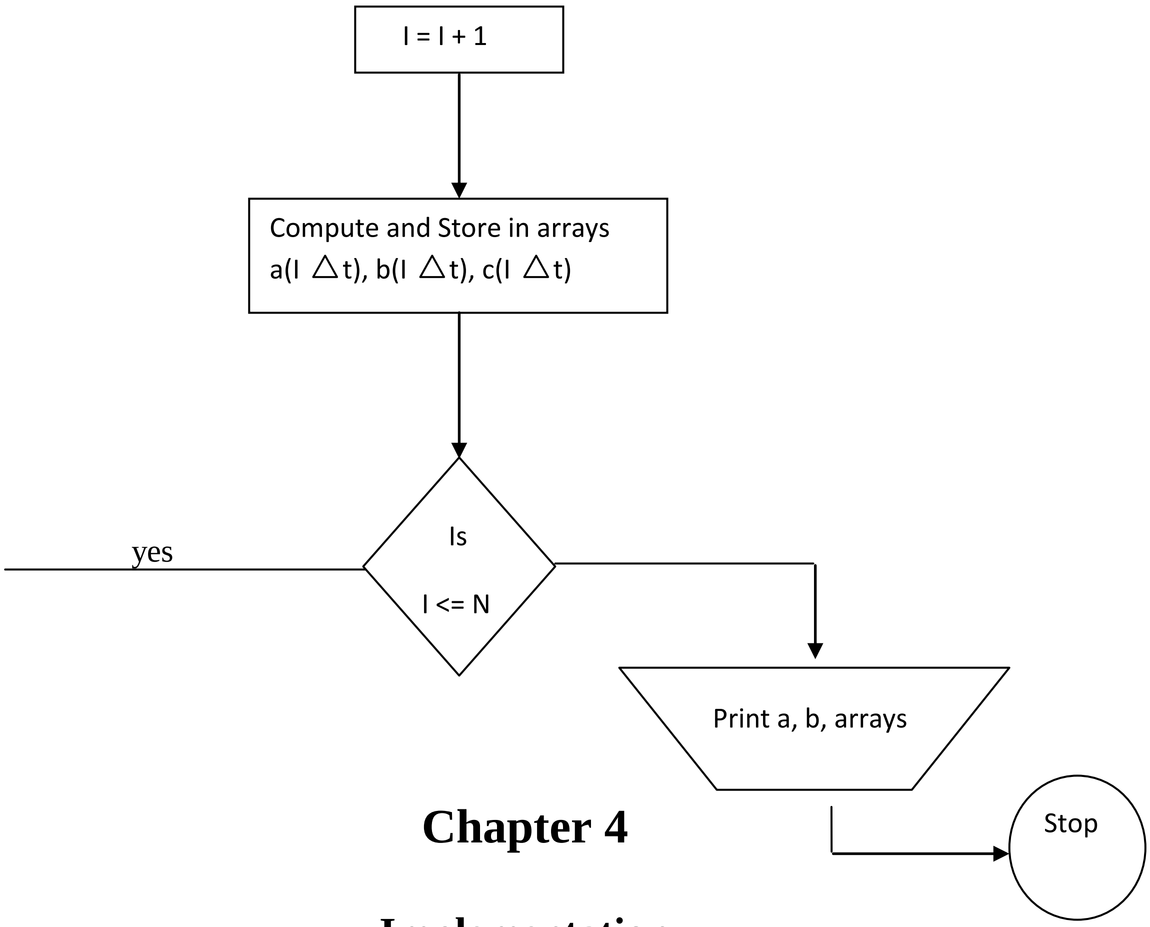
Continuous Stirred Tank Reactors (CSTR)

Plug Flow Reactors

These are all supplied with appropriate definitions for constant masses or volumes, as well as isothermal, non isothermal and adiabatic conditions. Perfect for incorporating your optimized kinetics in a process environment, these simple models allow for increased understanding of your system, and let you simulate a myriad of different operating conditions. With all the knowledge you gain from this, your next step is to optimize your unit's design and fine-tune you’re operating conditions through a full 2D axis symmetric or 3D model. The Generate Space-Dependent Model feature can be used to fully incorporate your system's mass and energy balances together with fluid flow and chemical rate of reactions.

3.2 Flowchart





Chapter 4

Implementation

4.1 List of Important Functions

The library functions used in our program have been arranged in alphabetical order as below. All these functions have been prototyped in the file graphics.h. Hence this file has to be included whenever I wish to use these functions in any program.

Circle

Purpose: Draws a circle with specified center and radius.

Syntax: circle (int x, int y, int rad);

cleardevice

Purpose: Clears the previous view.

Syntax: clear viewport();

close graph

Purpose: Shuts down the graphics system and de-allocates all memory allocated by the graphics system.

Syntax: void close graph();

flood fill

Purpose: Flood fills the bounded region.

Syntax: void flood fill (int x, int y, int border);

Comments: (x,y) is a ‘seed’ point within the enclosed area to be filled. The area bounded by the color border is flooded with the current fill pattern and fill color.

getbkcolor

Purpose: Returns the current drawing color.

Syntax: int get color();

Comments: The drawing color is the value to which pixels are set when lines, rectangles, etc, are drawn.

getmaxx

Purpose: Returns maximum x screen coordinate in the current mode.

Syntax: int getmaxx();

getmaxy

Purpose: Returns maximum y screen coordinate in the current mode.

Syntax: int getmaxy()

initgraph

Purpose: Initializes the graphics system.

Syntax: void initgraph (int *gd, int *gm, char *path);

*gd is an integer that specifies the graphics driver to be used.

*gm is an integer that specifies the initial graphics mode.

line

Purpose: Draws a line between two specified points.

Syntax: void line (int x1, int y1, int x2, int y2)

outtextxy

Purpose: Displays a string at a specified location.

Syntax: void outtextxy(int x, int y, char *str);

rectangle

Purpose: Draws a rectangle.

Syntax: void far rectangle(int left, int top, int right, int bottom);

setbkcolor

Purpose: Sets the current background color using the palette.

Syntax: void setbkcolor(int color);

set color

Purpose: Sets the current drawing color using the palette.

Syntax: void set color(int color);

setfillstyle

Purpose: Sets the fill pattern and color .

Syntax: void setfillstyle(int pattern, int color);

setline style

Purpose: Sets the current line width and style

Syntax: void setline style(int linestyle, unsigned upattern, int thickness);

settextjustify

Purpose: Sets text justification for graphics functions.

Syntax: void settextjustify(int horiz, int vert);

Settextstyle

Purpose: Sets the current characteristics for graphics output.

Syntax: void settextstyle(int font, int direction, int charsize);

set viewport

Purpose: Sets the current viewport for graphics output.

Syntax: void viewport (int left, int top, int right, int clip);

Source Code

```
#include<conio.h>

#include<graphics.h>

#include<stdio.h>

#include<dos.h>

#include<math.h>

#include<stdlib.h>


union REGS i,o;

void *a;

int top,bottom,windows_flag=0;
```

```
float product_a,product_b,reaction_time;



void first();

void welcome();

void demo();

void fill();

void exitscr();


void first_window();

void borders();

void mouse_moving();

void table1(float,float,float);

void reaction();

void graphb(float,float);

void grapha(float,float,float);

void flowchart();

void help();


        //INTERRUPTS FOR MOUSE

                //RESET MOUSE

initmouse()

{
        i.x.ax=0;

        int86(0x33,&i,&o);

        return(o.x.ax);

}

                //SHOW MOUSE POINTER

void showmouseptr()

{

        i.x.ax=1;

        int86(0x33,&i,&o);

}

                //TO GET MOUSE POSITION ANB BUTTON STATUS

void getmousepos(int *button,int *x,int *y)

{

        i.x.ax=3;

        int86(0x33,&i,&o);

        *button=o.x.bx;

        *x=o.x.cx;

        *y=o.x.dx;

}

                //TO HIDE MOUSE POINTER

void hidemouseptr()

{

        i.x.ax=2;
        int86(0x33,&i,&o);
}


void first()
{
        int j;
        setcolor(15);
        outtextxy(200,70,"START WITH THE NAME OF GOD");
        line(200,80,405,80);
        settextstyle(7,0,8);
        setcolor(3);
//      getch();
        delay(1200);
        for(int i=0;i<7;i++)
        {
//              for(int k=0;k<14;k++)
//              {
//                      setcolor(k);
                        outtextxy(60+i,100+i," WEL COME");
                        delay(100);
//              }
        }
        setcolor(9);
```

```
settextstyle(2,0,6);

outtextxy(220,220,"TO THE SOFTWARE FOR");



settextstyle(1,0,2);

setcolor(9);

outtextxy(290,410,"Press Any Key To Continue....");



while(!kbhit())

{

        setcolor(2);

        settextstyle(1,0,2);

        outtextxy(150,250,"SIMULATION OF CHEMICAL REACTOR");

        sleep(1);

        setcolor(4);

        settextstyle(1,0,2);

        outtextxy(150,250,"SIMULATION OF CHEMICAL REACTOR");

        delay(300);

        sleep(1);

        setcolor(3);

        settextstyle(1,0,2);

        outtextxy(150,250,"SIMULATION OF CHEMICAL REACTOR");

        delay(350);

}
```

```
        getch();
        cleardevice();
}

void welcome()
{
        int y=0;
        int i;

        for(i=0;i<getmaxx();i++)
        {
                y=32;
                while(y<=400)
                {
                        putpixel(random(i),random(i),DARKGRAY);
                        y++;
                }
        }
        for(i=0;i<getmaxx();i++)
        {
                y=0;
                while(y<=60)
                {
                        putpixel(random(i),random(y),LIGHTGRAY);

                        y++;

                }

        }

        setcolor(RED);

        setbkcolor(BLACK);

        settextstyle(6, HORIZ_DIR, 4);

        outtextxy(223, 0, "Chemical Reactor");

        sleep(1);

        setcolor(WHITE);

        setbkcolor(BLACK);

        settextstyle(6, HORIZ_DIR, 4);

        outtextxy(221, 2, "Chemical Reactor");



        delay(500);

        setcolor(14);

        settextstyle(11, HORIZ_DIR, 1);

        outtextxy(150, 107,"TITLE          :");

        outtextxy(150, 137,"COURSE         :");

        outtextxy(150, 167,"COLLEGE        :");

        outtextxy(150, 197,"SEMESTER       :");

        outtextxy(150, 227,"LANGUAGE USED :");
        outtextxy(150, 257,"PROGRAMMED BY :");


        int m=100;

        setcolor(11);

        settextstyle(11, HORIZ_DIR, 1);

        outtextxy(300, 107, "CHEMICAL REACTOR");

        delay(500-m);

        outtextxy(300, 137, "Master Of Computer Application");

        delay(500-m);

        outtextxy(300, 167,"Acharya Institute of Management & Science ");

        delay(500-m);

        outtextxy(300, 197,"Fifth");

        delay(500-m);

        outtextxy(300, 227,"Turbo C");

        delay(500-m);

        outtextxy(300, 280,"Amritha Vimalkumar      (11SKSCA005)");

        delay(500-m);

        outtextxy(300, 310,"Anuja Kumari                (11SKSCA006)");

        delay(500-m);

        outtextxy(300, 340,"Mudasir Wani               (11SKSCA0023)");

        delay(500-m);

        for(i=0;i<getmaxx();i++)

        {
        			y=440;
        			while(y<=480)
        			{
        				putpixel(random(i),y,DARKGRAY);
        				y++;
        			}
        		}
        		delay(500-m);
        		setcolor(WHITE);
        		settextstyle(11, HORIZ_DIR, 1);
        		outtextxy(400, 460,"Press any key to continue...");
        		while(!kbhit())
        		{
        			for(i=0;i<getmaxx();i++)
        			{
        				putpixel(i,60,random(13));
        				putpixel(i,440,random(15));
        			}
        		}
        		getch();
        //		cleardevice();
        void first_window()
        {
```

```
//	int windows_flag=0;

	cleardevice();

	borders();

	settextstyle(8,0,2);

	outtextxy(250,142,"Demo");

	outtextxy(250,172,"Simulate");

	outtextxy(250,202,"Flowchart");

	outtextxy(250,232,"Help");

	outtextxy(250,263,"Exit");

	setfillstyle(1,2);

	bar(250,20,400,40);

	int size =imagesize(250,20,400,40);

	free(a);

	a=malloc(size);

	getimage(250,20,400,40,a);

	setfillstyle(1,0);

	bar(250,20,400,40);

	putimage1(250,150,a);

	top=250;

	bottom=150;

	showmouseptr();

}
```

```
void get_input()
{
//	int product_a,product_b,time;
	clrscr();
	printf("\n\n\tenter the value of product A:----->");
	scanf("%f",&product_a);
	printf("\n\n\tenter the value of product B:----->");
	scanf("%f",&product_b);
	printf("\n\n\tenter the reaction time:---------->");
	scanf("%f",&reaction_time);
}
void mouse_moving()
{
		int x,y,button;
		while(!kbhit())
		{
			showmouseptr();
			getmousepos(&button,&x,&y);
			if(x>=250&&x<=400&&y>=150&&y<=170)
			{
				hidemouseptr();
				clear(top,bottom,a);
				bottom=150;
```

```
                putimage1(top,bottom,a);

                showmouseptr();

                delay(80);

                if(button==1)

                {

                        hidemouseptr();

                        demo();

                        fill();

                        reaction();

                        table1(100.00,50.00,5.00);

                        showmouseptr();

                }

        }

        else if(x>=250&&x<=400&&y>=180&&y<=200)

        {

                hidemouseptr();

                clear(top,bottom,a);

                bottom=180;

                putimage1(top,bottom,a);

                showmouseptr();

                delay(80);

                if(button==1)

                {
```

```
                                        hidemouseptr();
                                        get_input();
//                                      demo();
//                                      fill();
//                                      reaction();
                                        table1(product_a,product_b,reaction_time);
                                        first_window();
                                        showmouseptr();
                                }
                        }
                        else if(x>=250&&x<=400&&y>=210&&y<=230)
                        {
                                hidemouseptr();
                                clear(top,bottom,a);
                                bottom=210;
                                putimage1(top,bottom,a);
                                showmouseptr();
                                delay(80);
                                if(button==1)
                                {
                                        hidemouseptr();
                                        flowchart();
                                        showmouseptr();
```

```
        }
}
else if(x>=250&&x<=400&&y>=240&&y<=260)
{
        hidemouseptr();
        clear(top,bottom,a);
        bottom=240;
        putimage1(top,bottom,a);
        showmouseptr();
        delay(80);
        if(button==1)
        {
                hidemouseptr();
                help();
                showmouseptr();
        }
}
else if(x>=250&&x<=400&&y>=270&&y<=290)
{
        hidemouseptr();
        clear(top,bottom,a);
        bottom=270;
        putimage1(top,bottom,a);
```

```
                    showmouseptr();
                    delay(80);
                    if(button==1)
                    {
                        hidemouseptr();
                        exitscr();
                        showmouseptr();
                    }

                }
/*              gotoxy(10,20);
                printf("%d",button);
                gotoxy(50,20);
                printf("%d",x);
                gotoxy(80,20);
                printf("%d",y);*/
            }
}



void demo()
{
        int maxx,maxy,j,k;

        float x1,x2,y1,y2;

        char str[40];

        float a[52],b[52],c[52],t,i;

        char mesg[200];

        cleardevice();

  int x;

  setbkcolor(9);

  setcolor(1);

  settextstyle(2,HORIZ_DIR,0);

  rectangle(200,300,410,315);

  setfillstyle(SOLID_FILL,12);

  outtextxy(200,280,"Loading .......");

        for(x=202;x<408;x=x+8)

         {

           bar(x,302,x+4,313);

           delay(150);

         }

  cleardevice();

  setbkcolor(9);

  setcolor(1);

  settextstyle(2,HORIZ_DIR,0);

  rectangle(200,300,410,315);
```

```
setfillstyle(SOLID_FILL,12);

outtextxy(200,280,"Loading .......");

        for(x=202;x<408;x=x+8)

         {

           bar(x,302,x+4,313);

           delay(150);              }
```

Chapter 5

Testing

In a software development project, errors can be injected at any stage during the development. Testing performs a very critical role for quality and for ensuring the reliability of software. During testing, the program to be tested is executed with set of test cases, and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to. Due to its approach, dynamic testing can only ascertain the presence of error in the program the exact nature of the error is not usually decided by testing. Testing forms the first step in determining the errors in the program. Clearly the success of testing in revealing errors in programs depends critically on the test cases.

Testing is usually relied upon to detect the faults that occur during any phase of the software development cycle, in addition to the faults that introduced during the coding phase itself. For this, different levels of testing are used which perform different tasks and aim to test different aspects of the system. The basic levels of testing are unit testing, integration testing, system and acceptance testing. The different levels of testing attempt to detect different types of faults.

Client Needs	Acceptance Testing
Requirements	System Testing

Design	Integration Test
Code	Unit Testing

5.1 Output Analysis

For various test cases the output analysis of the project is given below. While giving different input values of the Product A and Product B the output is Product C. The output Product C depends upon the time and input given. The test cases are given below in the table.

Test Cases

Menu selected	Input 1 (product A qty)	Input 2 (product B qty)	Output (product C qty)	Test Result
1	Nil	Nil		Satisfactory
2	10	23		Satisfactory
2	100	150		Satisfactory
3	Nil	Nil	flow chart	Satisfactory

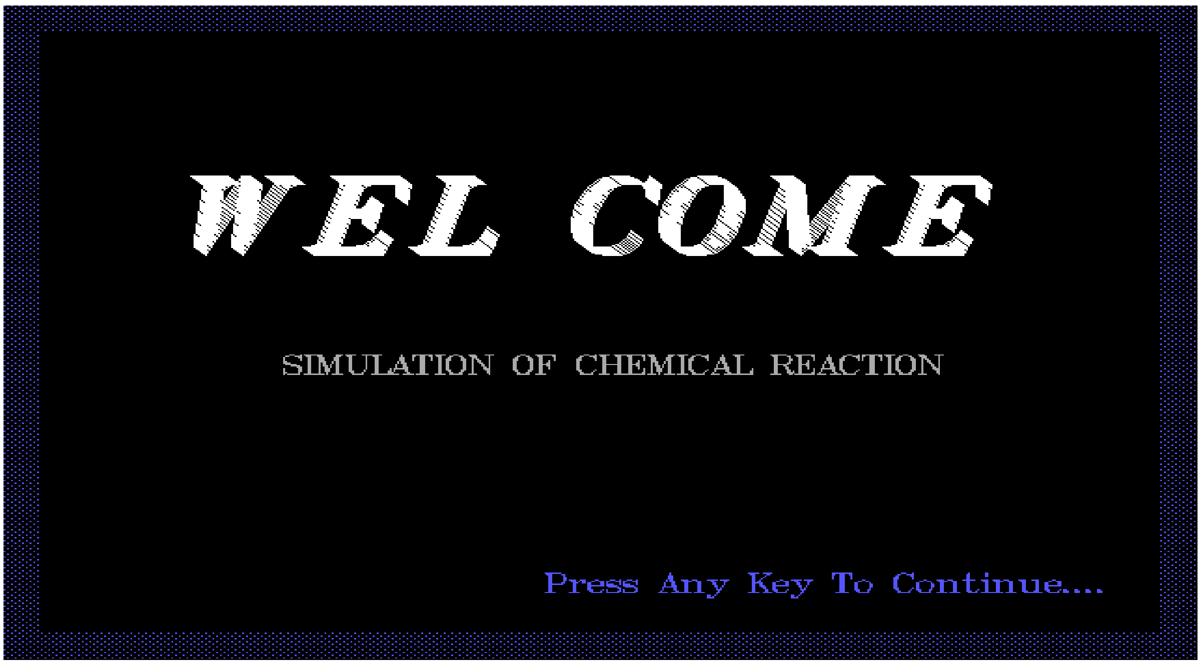
5.2 Test Report

- In menu when we select first option a demo of the project starts.
- The selection of second option demonstrates the simulation of the project by entering the value of products A and B and time limit respectively, it shows the process of chemical reaction with respect to time and the value of product C obtained.
- The third option provides the necessary help to execute the project.
- The final option is exit and it takes back to the program.

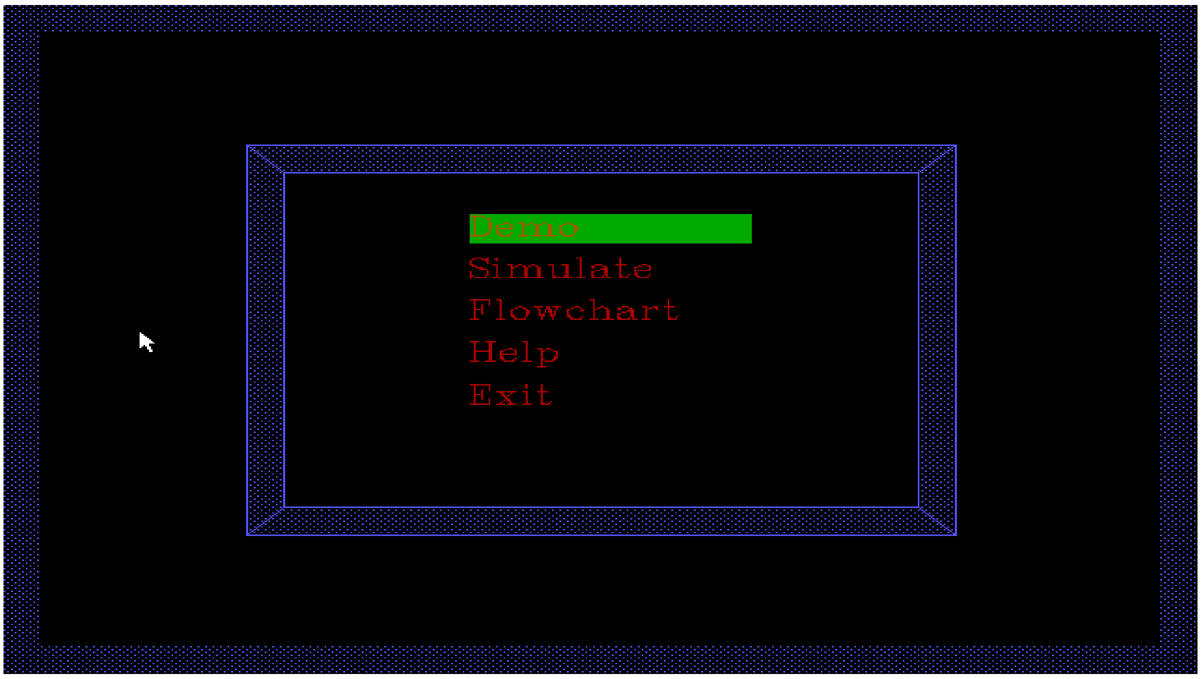
Chapter 6

Screen Layout

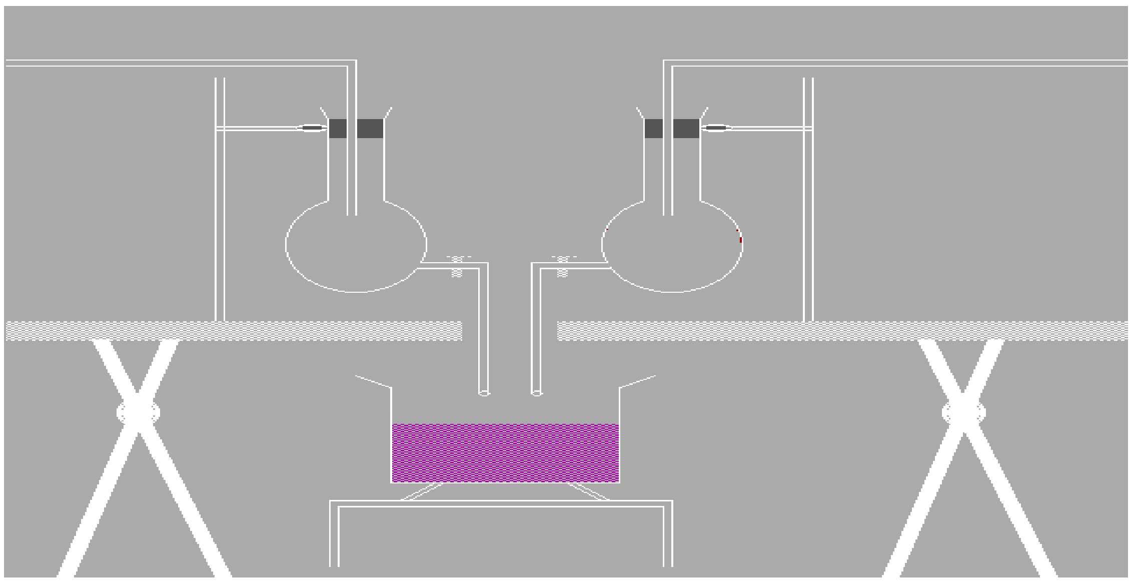
Homepage

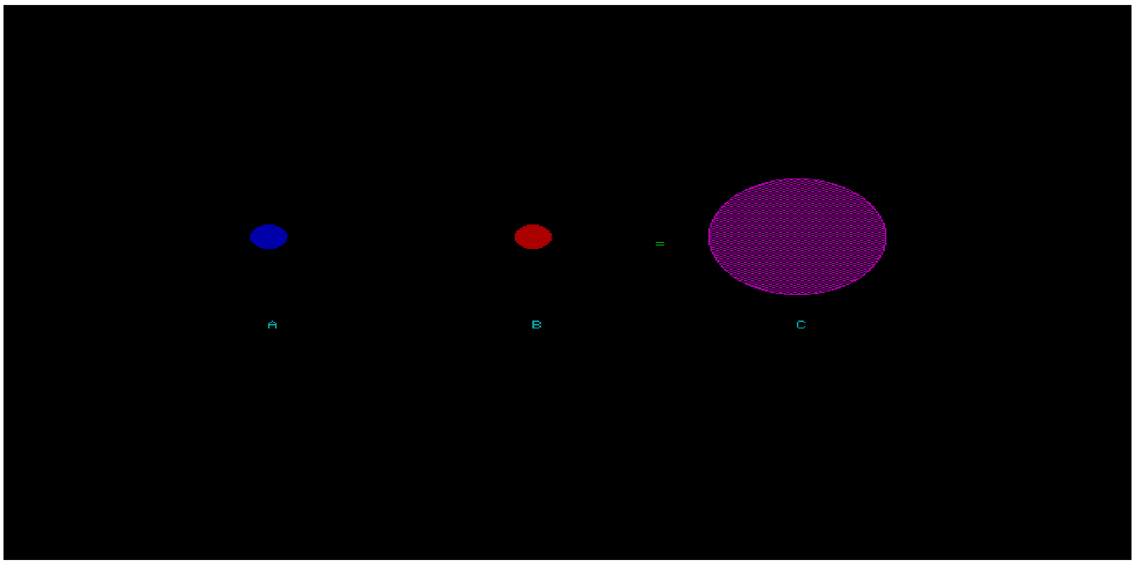

WEL COME
SIMULATION OF CHEMICAL REACTION
Press Any Key To Continue....

Main Menu



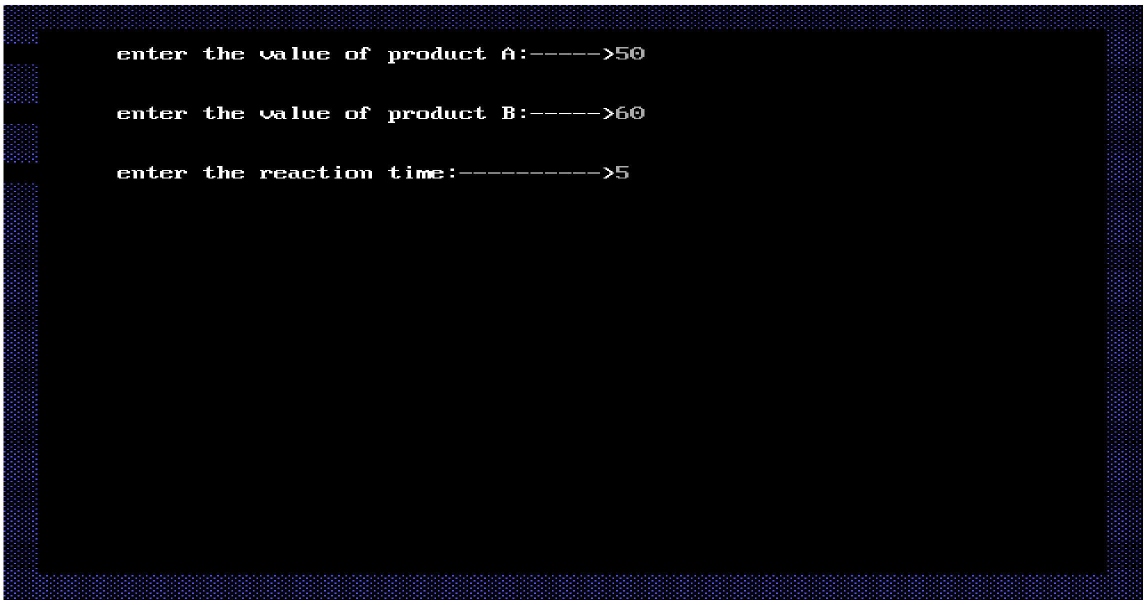
Demo





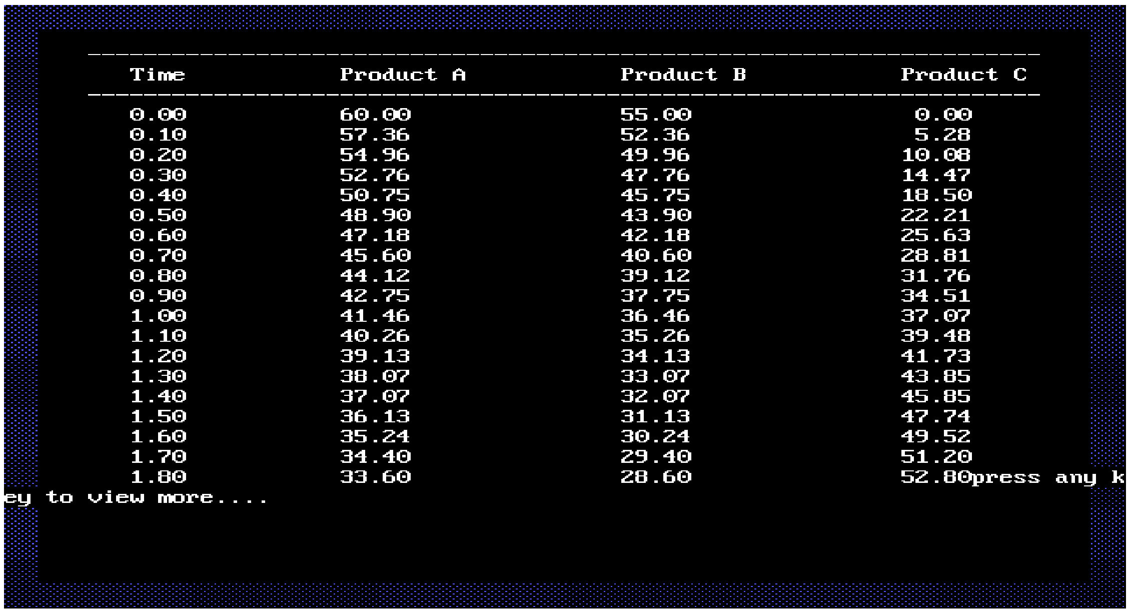
Simulation

Input



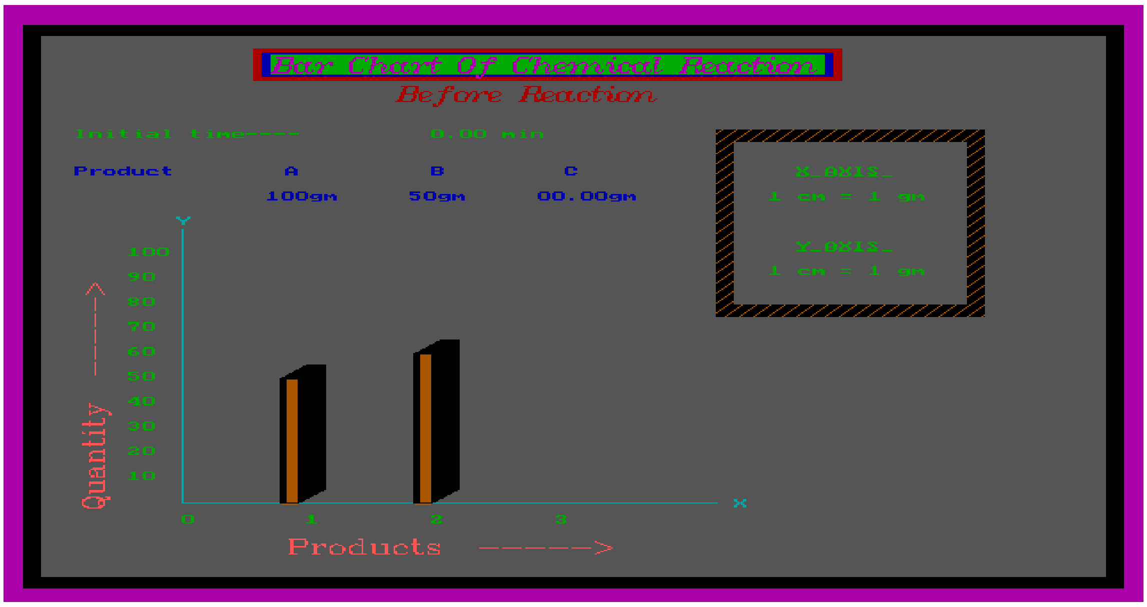
```
        enter the value of product A:----->50

        enter the value of product B:----->60

        enter the reaction time:---------->5
```

Output

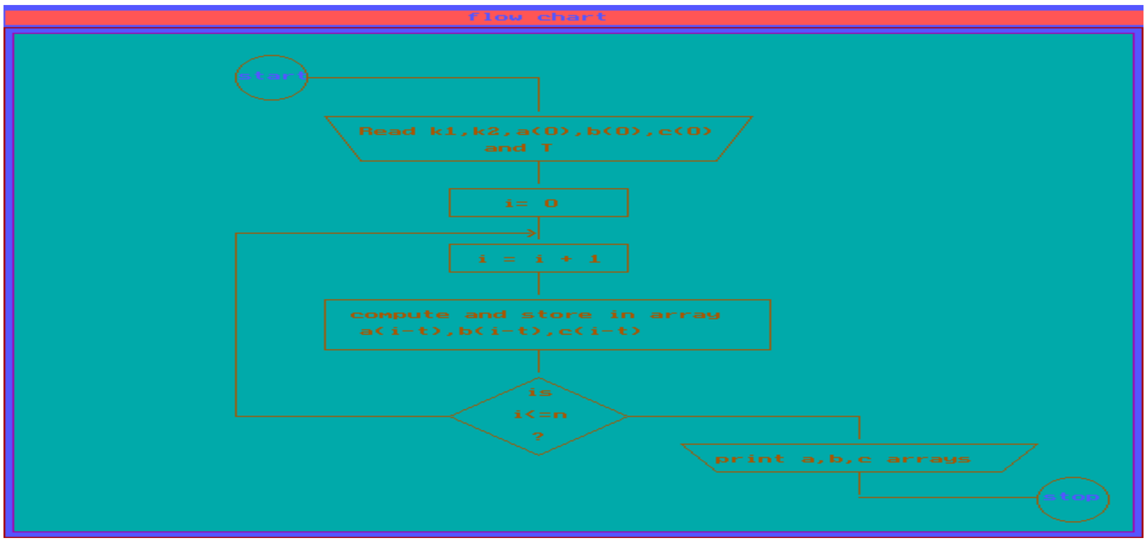


```
    -------------------------------------------------------------------
       Time            Product A            Product B           Product C
    -------------------------------------------------------------------
       0.00            60.00                55.00                0.00
       0.10            57.36                52.36                5.28
       0.20            54.96                49.96               10.08
       0.30            52.76                47.76               14.47
       0.40            50.75                45.75               18.50
       0.50            48.90                43.90               22.21
       0.60            47.18                42.18               25.63
       0.70            45.60                40.60               28.81
       0.80            44.12                39.12               31.76
       0.90            42.75                37.75               34.51
       1.00            41.46                36.46               37.07
       1.10            40.26                35.26               39.48
       1.20            39.13                34.13               41.73
       1.30            38.07                33.07               43.85
       1.40            37.07                32.07               45.85
       1.50            36.13                31.13               47.74
       1.60            35.24                30.24               49.52
       1.70            34.40                29.40               51.20
       1.80            33.60                28.60               52.80press any k
ey to view more....
```

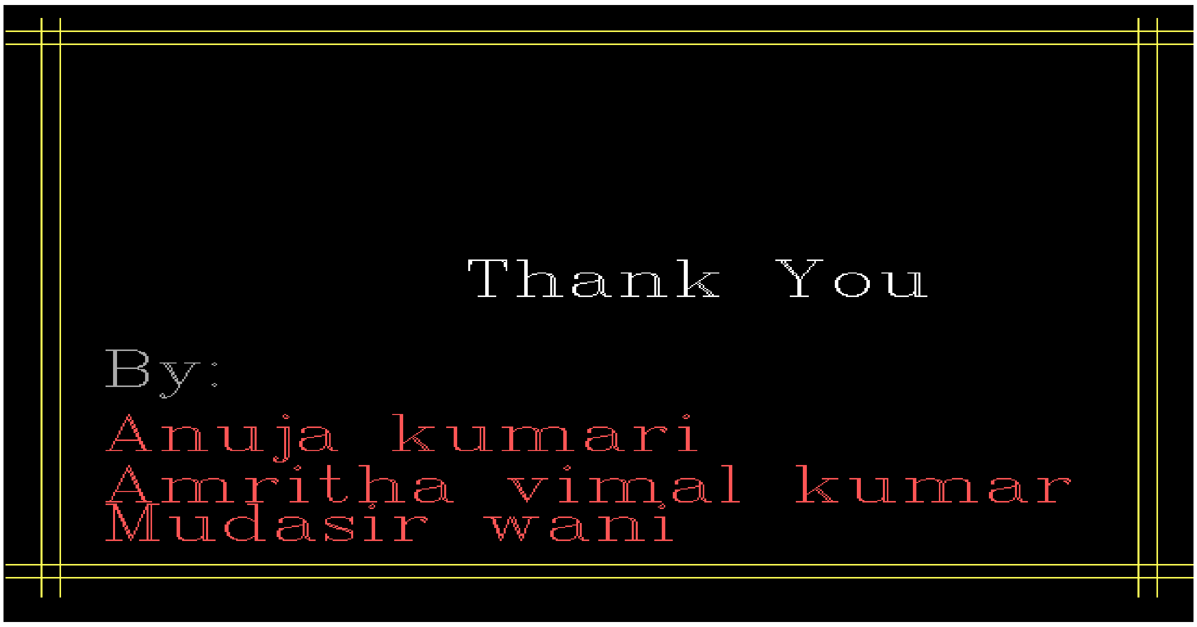
Bar Chart



Flow Chart



Exit



Thank You

By:

Anuja kumari

Amritha vimal kumar

Mudasir wani

Chapter 7

Conclusion

The simulation of **“Chemical Reactor”** allows us to understand the activities that occur during chemical reaction. It is fast and relatively inexpensive method of doing an experiment on the computer. Using the simulated model it is easy to find the problems that occur in the system, and the solution for that problem.

In real system, to find out the problem that occur in the system and to find out the solution to those problems, will take a long and it is very expensive. So using simulation we can do the above tasks in a low cost. So simulation of **"Chemical Reactor"**

- Saves time.
- Saves money.
- Easy to understand the system.
- Easy to find problems and solutions to those problems.

BIBLIOGRAPHY

We have consulted and drawn our knowledge from the following reference material. This proved to be of immense help.

1) System simulation 2nd edition by Gordon

2) Introduction to simulation by Payer

3) Graphics under ‘C’ by Yeshwanth kanetkar

4) http//:www.google.com