SIMULATION OF CONTINUOUS STIRRED TANK REACTORS (CSTR'S) USING ORTHOGONAL FUNCTIONS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

Master of Technology
In
Chemical Engineering
Department of Chemical Engineering

By

KAMESWARI MANI PRIYANKA NEMANI

Roll No: 213CH1125

Under the Guidance of

Prof. Madhusree Kundu

Department of Chemical Engineering

National Institute Technology, Rourkela-769008

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

CERTIFICATE

This is to certify that the project entitled "SIMULATION OF CONTINUOUS STIRRED TANK REACTORS (CSTR's) USING ORTHOGONAL FUNCTIONS**"** submitted by **Kameswari Mani Priyanka Nemani** (213CH1125) in partial fulfilment of the requirements for the award of Master of Technology degree in Chemical Engineering, Department of Chemical Engineering at National Institute of Technology, Rourkela is an authentic work carried out by her under my supervision and guidance.

To the best of my knowledge the matter embodied in this thesis has not been submitted to any other university/Institute for the award of any Degree.

Date:

Place: Rourkela

Prof. Madhusree Kundu

Department of Chemical Engineering
NIT Rourkela

ACKNOWLEDGEMENT

On the submission of my thesis entitled “SIMULATION OF CONTINUOUS STIRRED TANK REACTORS (CSTR’s) USING ORTHOGONAL FUNCTIONS” I would like to express my most sincere gratitude to **Dr. Madhusree Kundu**, my professor and thesis advisor, for her continuous guidance, encouragement and patience during the course of my project .Her direct contributions helped me in the completion of the thesis. I would like to thank the head of the department Prof.P.Rath for supporting me with all the necessary equipment and for giving such a supportive advisor.

And also thank D.Seshu Kumar, Research Scholar for helping me appreciate the underlying mathematical aspects of this project and Sujeevan Kumar Agir, Research scholar. Thank you to all my professors and technicians in the Department of Chemical Engineering.

I would like to thank all others who have consistently encouraged and gave me moral support, without whose help it would be difficult to finish this project.

I would like to thank my parents and friends for their consistent support throughout.

ABSTRACT

Over the centuries, several numerical methods have been developed to approximate the solution of mathematical problems that are difficult to be solved by analytical methods. These numerical techniques succeeded in attaining a solution that is close enough to the exact solution with minimum errors and maximum stability. However, there may be the development of several other numerical methods which can be robust and efficient than the existing methods.My proposed research work is about the application of one such method-Orthogonal functions. Orthogonal functions can be broadly classified in to three families; namely, the piecewise constant, polynomial, and sine-cosine family. Walsh function and block pulse function belong to the piecewise constant family. So far orthogonal functions have been used in the optimal control, solving integro-differential equations, trajectory problems and so on. However, orthogonal functions have not been applied to chemical systems and processes. Hence my work is emphasised on simulating reactors using orthogonal functions; mainly block pulse functions and triangular functions.

The continuous stirring tank reactors (CSTR's) are widely used in the chemical industries. Hence the reactions in a CSTR are modelled by a set of differential equations which are discretised to a set of algebraic equations by orthogonal functions. Previously many numerical methods such as Runge-Kutta method, Euler method have successfully converted the set of differential equations into a set of algebraic equations. But the orthogonality of the functions has never been used for discretisation. Here orthogonal functions simulate chemical reactors using the principle of orthogonality (two functions are said to be orthogonal if the dot product of the approximating vectors is zero). Block-pulse functions have been used to obtain the dynamics of concentration and temperature of the continuous stirring tank reactors (CSTR's). Further a recurrence relationship developed using block-pulse functions and triangular functions have been used in solving linear and non-linear system of differential equations. The major importance of orthogonal functions lies in its application to optimal control to systems. A recursive algorithm developed using block pulse functions has been applied to a linear control problem to determine the states and optimality criterion.

Keywords: Orthogonal functions, Block pulse functions, Triangular functions, system identification and optimal control.

TABLE OF CONTENTS

Abstract 1

Table of contents 2

List of tables 4

List of figures 5

CHAPTER-1 INTRODUCTION TO NUMERICAL TECHNIQUES

1.1 Numerical Analysis 9

1.1.1 Numerical methods for ODE's 10

1.1.2 Euler Method 11

1.2 Orthogonal Functions 12

1.2.1 Block Pulse Functions 13

1.2.2 Triangular Functions 18

1.2.3 Optimal Control with OF 18

1.3 Objective 18

1.4 Scope 18

1.5 Thesis Organization 19

CHAPTER-2 LITERATURE SURVEY

2.1 Literature Survey 21

CHAPTER-3 SIMULATION OF LINEAR AND NON-LINEAR SYSTEMS USING OPERATIONAL MATRICES OF BLOCK PULSE FUNCTIONS

3.1 Linear Systems 26

3.2 Simulation of Linear Systems 27

3.2.1 Results and Discussions 47

3.3 Non-Linear Systems 48

3.3.1 Differences between Linear and Non Linear Systems 49
3.3.2 Definition of Linear and Non-Linear Systems 49
3.3.3 Non-Linear Differential Equations 51

3.4 Simulation of Non-Linear Systems 52

3.4.1 Results and Discussions 56

CHAPTER-4 SIMULATION OF LINEAR AND NON-LINEAR SYSTEMS VIA ORTHOGONAL FUNCTIONS BY RECURRENCE RELATION

4.1 Recurrence relation using Triangular Functions for Linear Systems 58

4.2 Simulation of Linear systems via Recurrence Relation 61

4.2.1 Results and Discussions 67

4.3 Simulation of Non-Linear systems via Recurrence Relation 67

4.3.1 Results and Discussions 73

CHAPTER-5 OPTIMAL CONTROL OF CSTR'S USING ORTHOGONAL FUNCTIONS

5.1 Optimal Control of Systems 75

5.1.1 Analysis of Linear Control Systems Incorporating Observers 77

5.1.2 Results and Discussions 81

CHAPTER-6 CONCLUSION AND FUTURE SCOPE

6.1 Conclusion 83

6.2 Future Scope 83

REFERENCES 84

LIST OF TABLES

Table No.	Name of the table	Page No.
3.1	Parameter values of series reaction in CSTR	28
3.2	Parameter values of reversible reaction in CSTR	33
3.3	Parameter values of irreversible reaction in CSTR	38
3.4	Parameter values of a Jacketed Heater	43
4.1	Result Comparison Table for simulation of Non-isothermal CSTR using Recurrence relation	64
4.2	Result Comparison Table for simulation of Non-isothermal CSTR using Recurrence relation	66

LIST OF FIGURES

Fig No.	Name of the figure	Page No.
3.1	Concentration C_A, C_R vs. time and for n=100, T=5sec	29
3.2	Concentration C_A, C_R vs. time and for n=100, T=10sec	29
3.3	Concentration C_A, C_R vs. time and for n=100, T=20sec	30
3.4	Concentration C_A, C_R vs. time and for n=500, T=5sec	30
3.5	Concentration C_A, C_R vs. time and for n=500, T=10sec	31
3.6	Concentration C_A, C_R vs. time and for n=500, T=20sec	31
3.7	Concentration C_A, C_R vs. time and for n=100, T=10sec	33
3.8	Concentration C_A, C_R vs. time and for n=100, T=20sec	34
3.9	Concentration C_A, C_R vs. time and for n=100, T=40sec	34
3.10	Concentration C_A, C_R vs. time and for n=500, T=10sec	35
3.11	Concentration C_A, C_R vs. time and for n=500, T=20sec	35
3.12	Concentration C_A, C_R vs. time and for n=500, T=40sec	36
3.13	Concentration C_A, C_R vs. time and for n=100, T=10sec	38
3.14	Concentration C_A, C_R vs. time and for n=100, T=20sec	39
3.15	Concentration C_A, C_R vs. time and for n=100, T=40sec	39

3.16	Concentration C_A, C_R vs. time and for n=500, T=10sec	40
3.17	Concentration C_A, C_R vs. time and for n=500, T=20sec	40
3.18	Concentration C_A, C_R vs. time and for n=500, T=40sec	41
3.19	Temperature T, T_j vs. time and for n=100, T=10sec	44
3.20	Temperature T, T_j vs. time and for n=100, T=20sec	44
3.21	Temperature T, T_j vs. time and for n=100, T=40sec	45
3.22	Temperature T, T_j vs. time and for n=500, T=10sec	45
3.23	Temperature T, T_j vs. time and for n=500, T=20sec	46
3.24	Temperature T, T_j vs. time and for n=500, T=40sec	46
3.25	Concentration C_G, C_X and C_I vs. time and for n=100, T=100sec	54
3.26	Concentration C_G, C_X and C_I vs. time and for n=500, T=500sec	55
3.27	Concentration C_G, C_X and C_I vs. time and for n=1000, T=1000sec	55
4.1	Concentration C_A, C_P vs. time and for n=10, T=10sec	62
4.2	Concentration C_A, C_P vs. time and for n=50, T=50sec	63
4.3	Concentration C_A, C_P vs. time and for n=100, T=100sec	63

4.4	Concentration C_A, C_P vs. time and for n=100, T=10sec	65
4.5	Concentration C_A, C_P vs. time and for n=100, T=20sec	65
4.6	Concentration C_A, C_P vs. time and for n=100, T=30sec	66
4.7	Concentration C_A, T and T_j vs. time and for m=2, T=50sec	69
4.8	Concentration C_A, T and T_j vs. time and for m=5, T=50sec	70
4.9	Concentration C_A, T and T_j vs. time and for m=2, T=100sec	70
4.10	Concentration C_A, T and T_j vs. time and for m=1, T=500sec	71
4.11	Concentration C_G, C_X and C_I vs. time and for m=2, T=10sec	72
4.12	Concentration C_G, C_X and C_I vs. time and for m=2, T=20sec	72
4.13	Concentration C_G, C_X and C_I vs. time and for m=2, T=20sec	73
5.1	Error versus time, t=10sec,m=4	79
5.2	State Variables versus time, t=10sec,m=4	80
5.3	State Variables versus times, t=10sec, m=9	80
5.4	State Variables versus time, t=10sec, m=9	81

CHAPTER-1

INTRODUCTION TO NOVEL NUMERICAL TECHNIQUES APPLIED IN PROCESS SIMULATION AND CONTROL

1.1 NUMERICAL ANALYSIS

Numerical analysis is useful in practical mathematical calculations obtaining approximate solutions with a reasonable bound of errors. The Babylonian tablet that was the earliest mathematical writings approximated $\sqrt{(2)}$ the length of the diagonal of a unit square. Numerical analysis is applicable to all fields of engineering and physical sciences. They were used in solving System of Equations, Eigenvalues and Singular value problems, Numerical Integration, Ordinary and Partial Differential Equations. The overall goal of the numerical analysis is the design and analysis of techniques to approximate solutions of difficult problems accurately. A numerical method not only develops methods but also analyses them by three central concepts: convergence, stability, order. It works with a wide variety of problems such as:

- It makes weather prediction feasible.
- Computing the trajectory of a spacecraft requires an accurate numerical solution of a system of Ordinary Differential equations (ODE's).
- Car companies can improve the crash safety of their vehicles using computer simulations of car crashes, which involves solving partial differential equations.
- Insurance companies use numerical analysis for actuarial analysis.

Computation of solution to a problem in finite number of steps is possible by direct methods giving precise answers. Examples of direct methods include Gaussian elimination, QR factorization for solving linear equations and simplex method for linear programming. Indirect methods take infinite number of steps to produce output to a problem. It starts with an initial guess forming successive approximation and converges to the exact solution with a limit. It also includes a convergence test to decide the accuracy of the solution once it has been obtained. Examples of such methods are Newton-Raphson method, Bisection method and Jacobi iteration. Iterative methods otherwise called as indirect methods are more common than the direct methods.

The important part of the numerical analysis deals with a study of errors, which comprises of round-off errors, truncations errors, discretization errors. Round-off errors arise due to non-representation of all real numbers exactly on a machine with finite memory. Truncation errors occur due to termination of an iterative method or a mathematical procedure, which results in

approximate solution differing from the exact solution. Discretization error is caused when the discretized problem solution does not coincide with that of the continuous problem.

The other important part of the numerical analysis is stability, i.e. a procedure is said to be numerically stable if the error does not grow to a large extent during calculation, which means problem should be well conditioned. When the problem was well conditioned, the solution changes by a small amount for a small change introduced in the input data. However, an ill-conditioned problem exhibits significant deviation for small changes in data. Thus, the art of numerical analysis is to find a stable algorithm for solving a well-defined mathematical problem.

Most of the chemical reactions we come across in the industry are systems to which certain inputs are given and outputs are expected either in terms of mass compositions, temperatures, and pressures. We deal with non-ideal systems in which inputs get accumulated within the reacting system as time proceeds. Thus, unlike the ideal system for which rate of input is equal to rate of output, non-ideal systems have accumulation term equal to the difference between the inputs and outputs. When a reaction occurs, reaction rate term gets included. All these terms put together in mass balance equation and applying limits gives differential equations. Therefore, industrial reacting systems precisely have differential equations to be modelled. In order to analyse the real reactor behaviour, the proposed methods are to be applied to these model equations (set of differential equation).Thus the proposed work focuses on simulating differential equations within a reactor (CSTR).

1.1.1 Numerical Methods for Ordinary Differential Equations

Many differential equations cannot be solved by symbolic computation and hence practically needs to be approximated to the exact solution. Thus, algorithms which can handle ordinary differential equations are developed and used. Ordinary differential equations occur in many disciplines such as in physics, biology, chemistry, economics and engineering. To obtain the solution of some partial differential equations, they are converted to ODE's and then solved.

Numerical methods for initial value problems (IVP's) differ from that of boundary values problems (BVP's) which require a different set of tools. In a BVP, value or component of the unknown variable is defined at more than one point unlike in the IVP, which defines an unknown variable at the initial point of the system. Hence, BVP requires separate methods.

For solving first order initial value problems, methods are categorized into two types: linear multistep methods and Runge-Kutta methods.

A further classification includes implicit and explicit methods. Implicit linear multistep methods are the Adams-Moulton method and backward differentiation method (BDF) whereas implicit Runge-Kutta methods include diagonally RK method, singly diagonally implicit RK method and Gauss-Radau method. Explicit linear multistep methods are Adam-Bashforth method and any Runge-Kutta method with lower diagonal Butcher tableau is explicit. The rule of thumb is that for stiff differential equations implicit schemes must be used and for non-stiff differential equations explicit methods can be used.

1.1.2 Euler Method

Euler method is an SN-order numerical procedure for solving ODE of IVP. It is regarded as the primary explicit method for numerical integration of ODE's and is the simplest of all Runge-Kutta methods. Because it is a first-order method, the local error is proportional to the square of step size while the global error is proportional to step size. Euler method forms the basis for all sophisticated methods.

Forward Euler Method:-The forward Euler method is an explicit method which means a new value $\boldsymbol{y_{n+1}}$ is defined in terms of known $\boldsymbol{y_n}$

$$\boldsymbol{y'(t) = \frac{y(t+h) - y(t)}{h}} \quad \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \; \mathbf{(1.1)}$$

$$\boldsymbol{y(t+h) = y(t) + hy'(t)} \quad \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \; \mathbf{(1.2)}$$

$$\boldsymbol{y(t+h) = y(t) + hf(t, y(t))} \quad \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \; \mathbf{(1.3)}$$

$$\boldsymbol{y_{n+1} = y_n + hf(t_n, y_n)} \quad \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \; \mathbf{(1.4)}$$

Backward Euler method:-The backward Euler method, unlike the forward Euler method, is an implicit method in which the equation has to be solved to find out $\boldsymbol{y_{n+1}}$ **,** hence Newton-Raphson method is used. One disadvantage of an implicit method like Backward Euler method is the time for computation that is very high. However, the advantage is that implicit methods are more stable for solving stiff differential equations, and a large step size can be used.

$$\boldsymbol{y}'(\boldsymbol{t}) = \frac{\boldsymbol{y}(\boldsymbol{t}) - \boldsymbol{y}(\boldsymbol{t}-\boldsymbol{h})}{\boldsymbol{h}} \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\mathbf{1.5})$$

$$\boldsymbol{y}_{\boldsymbol{n}+\mathbf{1}} = \boldsymbol{y}_{\boldsymbol{n}} + \boldsymbol{h}\boldsymbol{f}(\boldsymbol{t}_{\boldsymbol{n}+\mathbf{1}}, \boldsymbol{y}_{\boldsymbol{n}+\mathbf{1}}) \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\mathbf{1.6})$$

The Euler method is often not very accurate as it considers only first order equations ignoring higher order equations. In such multistep methods, one gets to use the previously computed value $\boldsymbol{y}_{\boldsymbol{n}}$ to determine new value $\boldsymbol{y}_{\boldsymbol{n}+\mathbf{1}}$ **.** In case if more points are used in the interval, it leads to Runge-Kutta method.

1.2 ORTHOGONAL FUNCTIONS

In mathematics, two functions f and g are called orthogonal if their inner product (f, g) is zero for $f \neq g$. A typical definition of inner product is$\int\ f * (x)g(x)$, where $f * (x)$the complex conjugate of function is'f'. The inner product of f and g can be roughly approximated as the dot product between two vectorsf, g. Thus, two functions are orthogonal if their approximating vectors are perpendicular. Orthogonality of functions is a generalization concept of orthogonalization of vectors. Suppose we define V to be set of variables on which functions f and g operate then if $V = \{x\}$, x is the only parameter to f and g, thus there is one parameter; hence one integral sign is required to determine orthogonality.

Orthogonal functions can be broadly classified in to three families; namely, the piecewise constant, polynomial, and sine-cosine family. Harr functions, Block pulse functions, Delay-unit step functions, Slant functions, Triangular functions, Rademacher functions, Walsh function and block pulse function belong to the piecewise constant family while Chebyshev polynomial of first and second kind, Laguerre polynomial, Hermite polynomials, Jacobi polynomials together with their special cases the Gegenbauer polynomials, and Legendre polynomials belong to the polynomial family. Functions at any time can be synthesized using a set of orthogonal functions with a tolerable degree of accuracy. An orthogonal polynomial sequence is a family of polynomials such that any two different polynomials in the sequence are orthogonal to each other under some inner product.

The sine-cosine functions or orthogonal polynomials can represent a continuous function, however, becomes unsatisfactory for representing functions with discontinuities, jumps or dead-time. For representing such functions, piece-wise constant orthogonal functions such as Walsh functions or block pulse functions can be used. Each class of orthogonal functions

forms a basis for series expansion of a square integrable function, OFs' are commonly called as basis functions.

Orthogonal functions are used to construct operational matrices for solving, identification and optimization problems of dynamic systems. They help in dealing with various problems of dynamic systems as it reduces to those of solving algebraic equations. By using this approach, differential equations are converted into integral equations through integration, approximating various signals involved in the equation by truncated orthogonal functions and using operational matrices of integration to eliminate integral operation.

1.2.1 Block-pulse Functions

An orthogonal block-pulse function has been used to obtain the dynamics of concentration and temperature of the continuous stirring tank reactors. The operational matrix, P of the block-pulse functions eliminates integration operation and hence simplifies the system of state equations into a set of algebraic equations. The algorithm using BPF has been developed in the MATLAB platform. The idea is to present the states and outputs in terms of these block-pulse functions. The present method is advantageous over the existing methods like Runge-Kutta, Laplace transformations, State-space approach in terms of its simplicity in operation and accuracy.

Operational Matrix Derivation

An m-set of BPF is defined as follows:

$$\boldsymbol{\Phi_i(t)} = \begin{cases} \mathbf{1}, \boldsymbol{ih \le t \le (i+1)h} \\ \mathbf{0}, \boldsymbol{otherwise}, \end{cases} \quad \textbf{(1.7)}$$

Where $i = 1,2,\ldots\ldots,m-1$ with positive integer values for m, and h=T/m, and m are arbitrary positive integers. There are some properties for BPFs, e.g. disjointness, orthogonality, and completeness.

Disjointness: - The block-pulse functions are disjoint with each other; i.e.,

$$\boldsymbol{\Phi_i(t)\Phi_j(t)} = \begin{cases} \boldsymbol{\Phi_i(t), i = j}, \\ \mathbf{0}, \quad \boldsymbol{i \neq j} \end{cases} \quad \textbf{(1.8)}$$

Where $i, j = 0,\ldots\ldots,m-1$.

Orthogonality: - The block-pulse functions are orthogonal with each other i.e.,

$$\int \boldsymbol{\Phi_i(t)\Phi_j(t)dt} = \begin{cases} \boldsymbol{h, i = j,} \\ \boldsymbol{0, otherwise} \end{cases} \qquad \textbf{(1.9)}$$

In the region of $t \in [0, T)$, where $i,\ j = 1, 2, \ldots\ldots, m-1$. This property is obtained from the disjointness property.

Completeness:-

$$\int_0^\infty \boldsymbol{f^2(t)dt} = \sum_0^\infty \boldsymbol{f^2} \left\| \boldsymbol{\Phi_i(t)} \right\|^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(1.10)}$$

Where,

$$\boldsymbol{f_i = 1/h} * \left(\int_0^1 \boldsymbol{f(t)\Phi_i(t)}\right)\boldsymbol{dt} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(1.11)}$$

The set of BPFs may be written as an m-vector (t):

$$\boldsymbol{\Phi(t)\Phi^T(t) = 1}, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(1.12)}$$

Where t ∈ [0, 1). From the above representation and disjointness property, it follows:

$$\boldsymbol{\Phi(t) = [\Phi_0(t), \ldots\ldots, \Phi_{m-1}(t)]^T}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(1.13)}$$

$$\boldsymbol{\Phi\ (t)\Phi^T(t)} = \begin{bmatrix} \boldsymbol{\Phi_0(t)} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{\Phi_{m-1}(t)} \end{bmatrix} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(1.14)}$$

$$\boldsymbol{\Phi(t)\Phi^T(t)V = \widetilde{V}\Phi(t)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(1.15)}$$

Where V is an m-vector and $V = diag(V)$. moreover, it can be clearly concluded that for every m x m matrix A:

$$\boldsymbol{\Phi}(\boldsymbol{t})^{\boldsymbol{T}}\boldsymbol{A}\boldsymbol{\Phi}(\boldsymbol{t}) = \widehat{\boldsymbol{A}}^{\boldsymbol{T}}\boldsymbol{\Phi}(\boldsymbol{t}) \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.16})$$

Where A is an m-vector with elements equal to the diagonal entries of matrix A.

Functions approximation:-

A function $f(t), L€([0,1))$ may be expanded by the BPFs as:

$$\boldsymbol{F}(\boldsymbol{t}) = \sum \boldsymbol{F}_{\boldsymbol{i}}\boldsymbol{\Phi}_{\boldsymbol{i}}(\boldsymbol{t}) = \boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{\Phi}(\boldsymbol{t}) = \boldsymbol{\Phi}^{\boldsymbol{T}}(\boldsymbol{t})\boldsymbol{F} \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.17})$$

where F is a m-vector given by

$$\boldsymbol{F} = [\boldsymbol{f}_{\mathbf{0}}, \ldots\ldots\ldots \boldsymbol{f}_{\boldsymbol{m-1}}]^{\boldsymbol{T}} \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.12})$$

$$\boldsymbol{\Phi}(\boldsymbol{t}) = [\boldsymbol{\Phi}_{\mathbf{1}}(\boldsymbol{t}), \boldsymbol{\Phi}_{\mathbf{2}}(\boldsymbol{t}), \ldots\ldots\ldots, \boldsymbol{\Phi}_{\boldsymbol{m-1}}(\boldsymbol{t})]^{\boldsymbol{T}} \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.13})$$

The block-pulse $coefficients\ f_i$ are obtained as:

$$\boldsymbol{f}_{\boldsymbol{i}} = \frac{\mathbf{1}}{\boldsymbol{h}}\int \boldsymbol{f}(\boldsymbol{t})\boldsymbol{\Phi}_{\boldsymbol{i}}(\boldsymbol{t})\,\boldsymbol{dt} \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.14})$$

such that error between f(t), and its block-pulse expansion in the region of t ∈ [0,1)

$$\boldsymbol{\varepsilon} = \int_{\boldsymbol{i=0}}^{\boldsymbol{m-1}} (\boldsymbol{f} - \sum \boldsymbol{f}_{\boldsymbol{i}}\boldsymbol{\Phi}_{\boldsymbol{i}}(\boldsymbol{t}))^{2}\boldsymbol{dt} \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.15})$$

$$\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{t}) = \boldsymbol{\Phi}^{\boldsymbol{T}}(\boldsymbol{x})\boldsymbol{K}\boldsymbol{\Phi}(\boldsymbol{t}) \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.16})$$

$$\boldsymbol{k}_{\boldsymbol{ij}} = \boldsymbol{m}_{\mathbf{1}}\boldsymbol{m}_{\mathbf{2}} \iint \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{t})\boldsymbol{\Phi i}(\boldsymbol{x})\boldsymbol{\Phi j}(\boldsymbol{t})\boldsymbol{dxdt} \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.17})$$

Operational matrix Integration:-

We compute $\int_0^t \boldsymbol{\Phi}_i(\zeta)d\zeta$ as

$$\int_0^t \boldsymbol{\Phi}_i(\zeta)d\zeta = \begin{cases} \mathbf{0}, & t \le ih \\ t - ih, & ih \le t \le (i+1)h \\ h, & (i+1)h \le t \le 1 \end{cases} \qquad (1.18)$$

Then the above expression can be written as

$$\int \boldsymbol{\Phi}_i(\zeta)\, d\zeta = (t - ih)\boldsymbol{\Phi} i(t) + h \sum_{0}^{t} \boldsymbol{\Phi}_j(t) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.\ldots.... (\mathbf{1.19})$$

we have

$$X = \frac{1}{2h}\left(\sum_{i-1}^{m-1} \left[\left((i+1)h\right)^2 - (ih)^2\right] \boldsymbol{\Phi} i(t) \right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.\ldots. \quad (\mathbf{1.20})$$

By using orthogonal property, for0 $\le i \le m$, we have

$$\int_0^t \boldsymbol{\Phi}_i(\zeta)\, d\zeta = 1/2h\left(\sum_{j=0}^{m-1} [((j+1)h)^2 - (jh)^2]\right)\boldsymbol{\Phi}_j(t)\boldsymbol{\Phi}_i(t) - (ih)\boldsymbol{\Phi}_i(t) + h\sum \boldsymbol{\Phi}_i(t)$$

$$= 1/2h[((i+1)h)^2 - (ih)^2]\boldsymbol{\Phi}_i((t) - (ih)\boldsymbol{\Phi}_i((t) + h\sum \boldsymbol{\Phi}_i(t)$$

$$= h/2\boldsymbol{\Phi}_i + h\sum \boldsymbol{\Phi}_j(t)$$

$$\ldots.\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.\ldots..\ldots\ldots (\mathbf{1.21})$$

The integration of the vector $\Phi(t)$ may be obtained as:

$$\int \boldsymbol{\Phi}(\zeta)\, d\zeta \quad = \quad \boldsymbol{P}\, \boldsymbol{\Phi}(t) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{1.22})$$

Where P is called operational matrix of integration which can be represented by

$$\boldsymbol{P} = \frac{\boldsymbol{h}\begin{bmatrix}\boldsymbol{1} & \cdots & \boldsymbol{2}\\ \vdots & \ddots & \vdots\\ \boldsymbol{0} & \cdots & \boldsymbol{1}\end{bmatrix}}{\boldsymbol{2}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots.\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..\ldots.(\boldsymbol{1.23})$$

And their integrals in matrix form

$$\begin{pmatrix}\int \boldsymbol{\Phi_0} & \cdots & \boldsymbol{0}\\ \vdots & \ddots & \vdots\\ \int \boldsymbol{\Phi_{m-1}} & \cdots & \boldsymbol{0}\end{pmatrix} = \frac{\boldsymbol{h}}{\boldsymbol{2}}\begin{bmatrix}\boldsymbol{1} & \cdots & \boldsymbol{2}\\ \vdots & \ddots & \vdots\\ \boldsymbol{0} & \cdots & \boldsymbol{1}\end{bmatrix}\begin{pmatrix}\boldsymbol{\Phi_0} & \cdots & \boldsymbol{0}\\ \vdots & \ddots & \vdots\\ \boldsymbol{\Phi_{m-1}} & \cdots & \boldsymbol{0}\end{pmatrix} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(\boldsymbol{1.24})$$

By using matrix we can express the integral of a function f(t) into block pulse series

$$\int_{\boldsymbol{0}}^{\boldsymbol{t}} \boldsymbol{f_m}(\zeta)\, \boldsymbol{d}\zeta \quad = \int_{\boldsymbol{0}}^{\boldsymbol{t}} \boldsymbol{F^T \Phi_m dt} = \boldsymbol{F^T P\ \Phi_m}(\boldsymbol{t}) \ldots\ldots.\ldots\ldots\ldots\ldots.\ldots\ldots...(\boldsymbol{1.25})$$

Example

Consider a first order D.E such as $\frac{dy}{dt} = y; y(0) = 1; t \in [0,1]; h = 0.25$

From N-fold integration property, the L.H.S of the D.E can be integrated as $y(t) - y(0)$

Then the R.H.S can be represented as $Jy(t)$

Combining the expressions in L.H.S and R.H.S. we get

$y(t) = Jy(t) + y(0)$

This can be written in terms of BPF coefficients

$$\boldsymbol{C^T}\psi(\boldsymbol{t}) - \boldsymbol{C_0}^{\boldsymbol{T}}\psi(\boldsymbol{0}) = \boldsymbol{C^T P}\psi(\boldsymbol{t}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.\ldots\ldots..(\boldsymbol{1.26})$$

$$\boldsymbol{C} - \boldsymbol{C_0} = \boldsymbol{C} * \boldsymbol{P} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots....(\boldsymbol{1.27})$$

Now taking m=4; $C_0^T = [1\ 1\ 1\ 1]$; $P = \frac{h}{2}\begin{bmatrix}1 & 2 & 2 & 2\\ 0 & 1 & 2 & 2\\ 0 & 0 & 1 & 2\\ 0 & 0 & 0 & 1\end{bmatrix}$

Eqn (1.27) can be rewritten as

$$(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{C^T} = \boldsymbol{C_0}^{\boldsymbol{T}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(\boldsymbol{1.28})$$

Where I is the identity matrix of order 4x4, P is the operational matrix of order 4x4.On solving the eqn. (1.28) gives the unknown BPF coefficient $\boldsymbol{C^T}$ which is the output of the above stated first order D.E

$$\boldsymbol{C^T} = [\boldsymbol{1.1429} \quad \boldsymbol{1.4964} \quad \boldsymbol{1.8892} \quad \boldsymbol{2.4921}] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(\boldsymbol{1.29})$$

1.2.2 Triangular function

Apart from Walsh functions and block pulse functions, the other orthogonal function which is equally potential in simulations of process systems is the triangular function set. The triangular function set can also approximate function and form operational matrices P_1 and P_2 to resolve differential and integral equations.

1.2.3 Optimal control with BPF

To derive the control policies, there exists a mathematical optimization method called optimal control theory. Optimal control finds a control law for a specific system that achieves certain optimality criterion. There exists a cost function in control problems that is a function of state and control variables. Therefore in order to minimize the cost functional, optimal control describes the paths of control variables. The optimal control problems are nonlinear in nature and cannot be solved by analytical methods. Thus, numerical methods can be employed to solve them. Block pulse function is applied to the control problem to establish the optimality criterion.

1.3 OBJECTIVE

The objective of the present dissertation is to utilize the potency of orthogonal functions namely block-pulse function (BPF) and triangular function (TF) in simulation and control of different kinds of continuous stirred tank reactors.

1.4 SCOPE

Our work is fragmented into five main divisions in which model equations of various reactions in a CSTR are considered. These model equations are simulated by the following ways:

1) Simulation of a reversible reaction, irreversible reaction, and jacketed heater by operational matrix derived using block pulse functions. These systems come into the category of linear systems.
2) Simulation of Glucose and Insulin concentrations in a Biochemical Reactor by operational matrix derived using block pulse functions (BPF). These are non-linear systems.
3) An alternative approach to simulate linear and non-linear systems is the recursive relation developed in both BPF and triangular function (TF) domains.

The recurrence relation in both domains is applied to two systems of irreversible reactions.

4) Recurrence relation in BPF domain is applied to two non-linear systems: Biochemical Reactor and Non-isothermal CSTR.
5) Optimal Control of CSTR's using block-pulse functions.

1.5 THESIS ORGANIZATION

The thesis consists of six chapters in all .They are

Chapter 1: Introduction of numerical techniques

Chapter 2: Literature survey; in this chapter the details of other significant works done on the applicability of orthogonal functions are presented.

Chapter 3: Simulation of linear and non-linear systems using operational matrices of block pulse functions; the output variables are determined by converting model equations into a system of linear algebraic equations with the help of operational matrix and this technique includes no direct integration.

Chapter 4: Simulation of linear and non-linear systems via orthogonal functions by recurrence relation; the output variables are determined by using the recurrence algorithm developed from block pulse functions. The model equations are expressed in state-space model and the matrices are substituted in recurrence relation which gives the output variables i.e. the dynamics of the system can be established.

Chapter 5: Optimal control of CSTR's using block pulse functions; this chapter deals with the analysis of linear optimal control by incorporating observers. The error vector, state vector and input vector are determined using the optimal control law and recurrence algorithm using block pulse functions.

Chapter 6: Conclusions and Future Scope; in this chapter the extensions of the work and the conclusions of present work are presented.

CHAPTER-2

LITERATURE SURVEY

LITERATURE SURVEY

A Substantial amount of research work has been carried out globally on the application of orthogonal functions to various fields of engineering. These intensive studies laid a foundation stone for the application of orthogonal functions to chemical engineering. Here are a list of works briefly detailing their scope and magnitude of orthogonal functions.

The Legendre polynomial originated in determining the force of attraction exerted by solids of revolution and their orthogonal properties were established by A. M. Legendre during 1784-1790. Piecewise constant OF's basis functions having constant functional values within any subinterval of time period. Block pulse function is a complete OF, which provides elegant solution to the areas of parameter estimation, analysis and control. Substantial amount of research work has been carried out globally on the application of orthogonal functions to various fields of engineering. These intensive researches laid foundation stone for the application of orthogonal functions to chemical engineering. Here is a list of works already being carried out.

Solving Integral Equations:-

Maleknejad et al., (2013) conducted works on the linear Fredholm integral equations of the second kind that were solved using a combination of Block pulse functions and orthonormal Berstein functions .The integral equations are converted to a system of linear equations, and the results of the proposed method are compared with true solutions. The advantage of this method is that there is only addition and multiplication of matrices and needs no integration. It is an efficient and simple way in terms of applicability.

Maleknejad et al., (2012) presented the numerical solution of Volterra Integral equations using an iterative method, whose results are compared with that of the direct method, collocation method and iterated collocation method. The convergence results showed that the proposed method is at least as rapid as the direct method. The proposed method is not very much efficient than direct method but is of interest in many cases.

System Identification:-

Block pulse functions are used not only in solving various higher order integral equations but also in system identification, parameter identification that forms an important part of analysing the dynamic systems. One of such significant works on identification of continuous systems done by **G.P Rao and H.Unbehauen** stated that continuous time domain though is native to system identification; the advent of discrete time models has undermined continuous-time domain models .An orthogonal approach in the continuous time domain has been proposed.

In another research work conducted by **Anish et al., (2013)** emphasis was given to the identification of SISO control systems using non-optimal block pulse functions .The non-optimal method BPF coefficient computation employs trapezoidal integration instead of exact integration that uses only samples of functions expanded via BPF reducing computation burden drastically. Further results obtained by this approach contained fewer errors than the results obtained by the traditional BPF approach. The results of identification are also found to be superior by non-optimal BPF over optimal BPF.

P. Sannuti worked on development of a recurrence relation using block pulse functions which is used in integrating a system of differential equations thereby gives piecewise constant solutions with minimal mean square error and is computationally similar to trapezoidal rule of integration .This technique can be applied to both linear and nonlinear system of differential equations. This method simplifies the design of piecewise constant controls and feedback gains for dynamic systems.

Triangular function sets are used to determine the convolution of two TF components or trains, which in turn determine the output of linear SISO control system via an operational matrix technique.

Optimal Control of Systems:-

Ogata (2002) proposed a parameter optimization technique for computing feedback controller parameters using the downhill simple method that is a pattern search algorithm has been proposed. By this method optimal parameter that minimise the objective function and performance index are determined.

Simulation of control systems can be done more easily via orthogonal functions where the expansion coefficients are derived from the samples of the related functions. Microprocessor based simulation of discrete time as well as sample and hold systems have been carried out via sample-and-hold function set and Dirac delta function set (derived using triangular function and BPFs). In identification of control systems with known inputs and outputs, such simulations prove to be useful.

In state feedback controller design, the estimation of state variables is a paramount part. Since all the states are immeasurable, design of observers is a necessity; either full order or reduced order to estimate the states. Luenberger observer is used in this regard in general, which may produce erroneous estimate in the noisy environment. Fortunately the OFs; have some inherent filters embedded in them due to the involvement of an integration process, that causes a smoothing effect. OFs' can act as a filter in the noisy environment, while estimating the states. From the open literature it is clear that, till date two attempts have been made for state estimation problems using BPF and shifted chebyshev polynomial of first kind.

In comparison with other basis functions or polynomials, BPFs can lead more easily to recursive computations to solve concrete problems and among piecewise constant basis functions, the BPFs set has proved to be the most fundamental. These functions have been directly used for solving different problems especially integral equations. In control engineering; the optimal tracking problem is to determine control inputs so that the system states track the desired state trajectory. Typically optimal tracking problem of large scale systems is to determine control inputs. In this case, optimal trajectory problems will lead to a very high order system, thus it cause any difficulties to solve the optimal tracking problem. In addition, due to the high order system controller, computational burden can be increased. Thus, by applying the hierarchical system theory and orthogonal functions, such as Walsh functions, block pulse functions and Haar functions, we can solve these problems.

The minimization of a quadratic performance index to control the linear systems gives rise to a time varying gain for the linear state feedback, and the solution is obtained by the Riccati equation. Chen and Shiao (1975) applied the Walsh functions and obtained a numerical solution of the Riccati equation and the time varying gain .OF approach was successfully applied to problems in systems and control. The problem of optimal control incorporating observers has been successfully addressed using different classes of OFs' including BPFs, shifted Ligendre polynomials (SLPs), shifted Jacobi polynomials (SJPs), general orthogonal

polynomials (GOPs), among which some are recursive and some are non-recursive in nature. Synthesis of optimal control laws by integro- differential equations had been studied using dynamic programming method and subsequently by BPFs .The LQG control design problem had been solved using GOPs. By using the GOPs the non-linear Riccati equations were reduced to non-linear algebraic equations, which then solved to get the final solution. Singular systems deserve immense significance so far its control is concerned. Harr wavelet approach, sine cosine functions (SCFs), SCP1s, and Legendre wavelets have been applied for solving optimal control problems of singular systems. They considered time invariant system with one delay in state and one delay in control. Amount of work on optimal control of non-linear systems is not substantial so far. Lee and Chang (1975) appeared to be the pioneer researchers in optimal control of non-linear system using GOPs. Chebyshev polynomials of first kind (CP1s) were used for solving non-linear control problems. A general framework for solving non-linear optimal control problems using BPFs has been provided by Shienyu.

CHAPTER-3

SIMULATION OF LINEAR AND NON-LINEAR SYSTEMS USING OPERATIONAL MATRICES OF BLOCK PULSE FUNCTIONS

3.1 LINEAR SYSTEMS

The system of Differential equations arises quite quickly from naturally occurring situations. The mathematical model of the system using a linear operator is said to be a linear system. It exhibits simple features and properties, unlike the general case non-linear systems. Linear systems are applied in automatic control theory, signal processing and telecommunications. A linear system can be defined by the operator H, that maps an input, x(t) as a function of t to an output y(t). Linear systems satisfy the properties of superposition and homogeneity.

Given two valid inputs $x_1(t), x_2(t)$ and their respective outputs

$$\boldsymbol{y_1(t) = H\{x_1(t)\}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.\dots\dots.\dots\dots..\mathbf{(3.1)}$$

$$\boldsymbol{y_2(t) = H\{x_2(t)\}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.\dots\dots.\dots\dots..\mathbf{(3.2)}$$

Then the linear system must satisfy

$$\boldsymbol{\alpha y_1(t) + \beta y_2(t) = H\{x_1(t) + x_2(t)\}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.\mathbf{(3.3)}$$

for any scalar values α β

The system is then defined by the equation $\boldsymbol{H(x(t)) = y(t)}$ where $y(t)$some arbitrary function of time is and $x(t)$ is the system state. Given $y(t)$ and H, $x(t)$ can be solved for. In non-linear systems, there is no such relation that makes the solution to model equations simpler than many non-linear systems. Linearity is the basis of impulse response or frequency response methods for time-invariant systems.

Laplace Transforms are used to analyse differential equations of linear time invariant systems in the continuous case while Z-transforms are used for analysis in the discrete case. Linear models describe the non-linear system by linearization, which is a kind of mathematical convenience.

So far, a formal introduction has been given about various numerical approximation techniques that are to be used. These numerical methods are applied to a system of differential equations categorized into linear and non-linear differential equations. Appreciating the simplicity and flexibility of linear systems, new methods are primarily applied to such systems. When results produced using such methods for linear systems are consistent, a further step of introducing them to complex non-linear systems can be done. Otherwise, if

results turn out to be unsatisfactory, any new numerical method can be terminated at this stage itself which saves time and cost of computation. Therefore, orthogonal functions majorly block pulse functions with operational matrices are implemented on four set of differential equations comprising of chemical reactions in a continuous stirred tank reactor. These four systems include reversible reactions, irreversible reactions and jacketed heater**.**

3.2 SIMULATION OF LINEAR SYSTEMS

3.1) Simulation of two reactions occurring in CSTR: -

Problem Statement: - Consider the following set of differential equations in an isothermal CSTR. The block-pulse functions are used in solving the two differential model equations in an isothermal CSTR. The states are the concentration of A and B in the reactor.C_A is the concentration of the reactant A and C_B is the concentration of B. The parameter values are k_1=5/6min^{-1}, k_2=5/3 min^{-1},k_3=1/6mol/lt.min. The input values used in the following simulation are F/V=4/7min^{-1}, C_{Af}=10 mol/lt are listed in Table-3.1

Model equations:-

$$\frac{dC_A}{dt} = \frac{F(C_{Af} - C_A)}{V} - k_1 C_A - k_3 C_A \quad \ldots\ldots(3.4)$$

$$\frac{dC_B}{dt} = -\frac{FC_B}{V} + k_1 C_A - k_2 C_B \quad \ldots\ldots(3.5)$$

Now expressing the two differential equations in terms of block-pulse functions as follows:

Let $C_A = x_1, C_B = x_2$

After substituting the given parameters in the above differential equations (3.4) and (3.5), we get the following equations,

$$\frac{dx_1}{dt} = 5.7143 - 3.0714x_1 \quad \ldots\ldots(3.6)$$

$$\frac{dx_2}{dt} = 0.8333x_1 - 0.7381x_2 \quad \ldots\ldots(3.7)$$

On integrating the above equations (3.6) and (3.7), we get

$$x_1(t) - x_1(0) = 5.7143Jdt - 3.071Jx_1(t) \quad \ldots\ldots(3.8)$$

$$x_2(t) - x_2(0) = 0.8333Jx_1(t) - 0.7381Jx_2(t) \quad \ldots\ldots(3.9)$$

$$C_1{}^T\psi(t) - C_{10}{}^T\psi(0) = 5.7143t - 3.071C_1{}^T P\psi(t) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.10)$$

$$C_2{}^T\psi(t) - C_{20}{}^T\psi(0) = 0.8333C_1{}^T P\psi(t) - 0.7381C_2{}^T P\psi(t) \ldots\ldots\ldots\ldots\ldots (3.11)$$

Equations (3.10) and (3.11) are further solved to obtain the values of C_1 and C_2.Here J is the integration operator, C_1 and C_2 are the block pulse coefficients, C_1 (0), C_2 (0) are the initial steady state values, P is the operational matrix and ψ represents block pulse function.

Table-3.1:- Parameter values of series reactions in CSTR

Notations	Parameters	Steady state values
C_{Af}	10mol/lt	
k_1	5/6 min^{-1}	
k_2	5/3 min^{-1}	
k_3	1/6 mol/lt.min	
F/V	4/7min^{-1}	
C_{As}	-	2
C_{Bs}	-	1.117

The Block-pulse function code is compiled which gives a set of output values of C_A, C_B which are plotted against the time intervals t=5, 10 and 20 with matrix sizes 100, 500.

Graphs

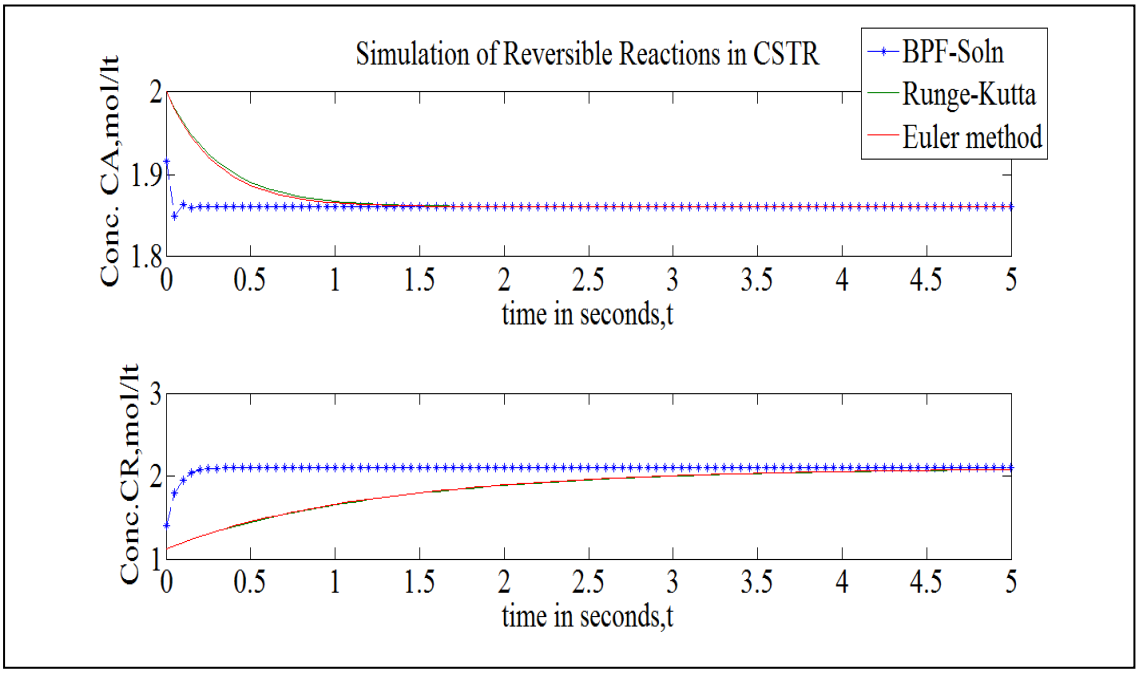

Figure-3.1: Concentration C_A, C_R vs. time and for n=100, T=5sec

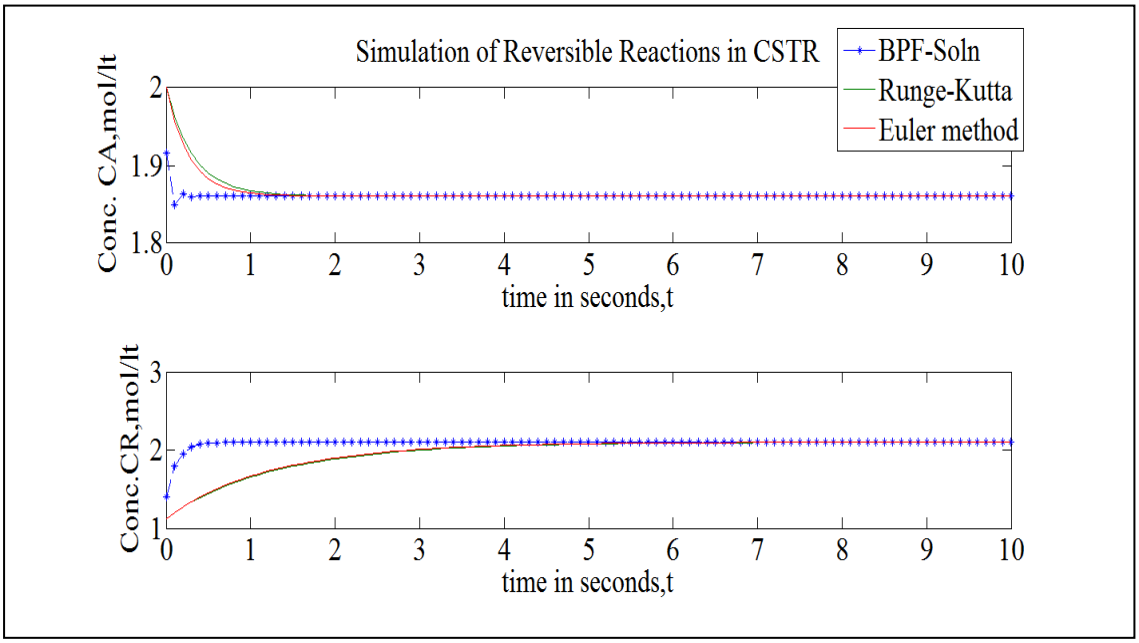

Figure-3.2: Concentration C_A, C_R vs. time and for n=100, T=10sec

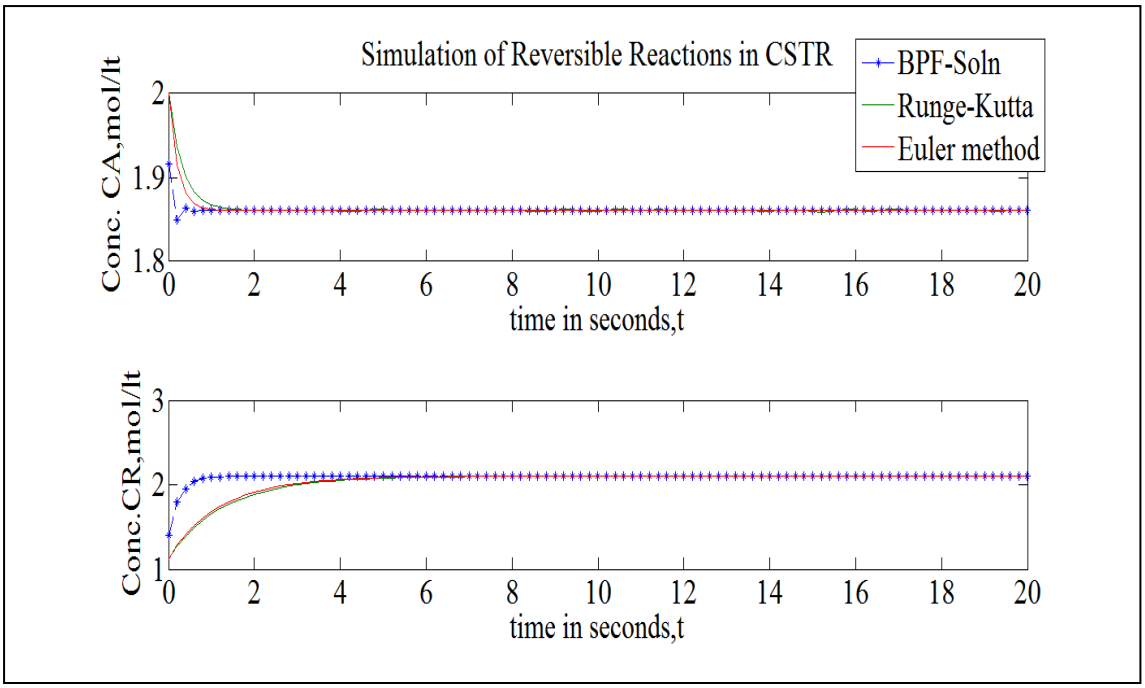

Figure-3.3: Concentration C_A, C_R vs. time and for n=100, T=20sec

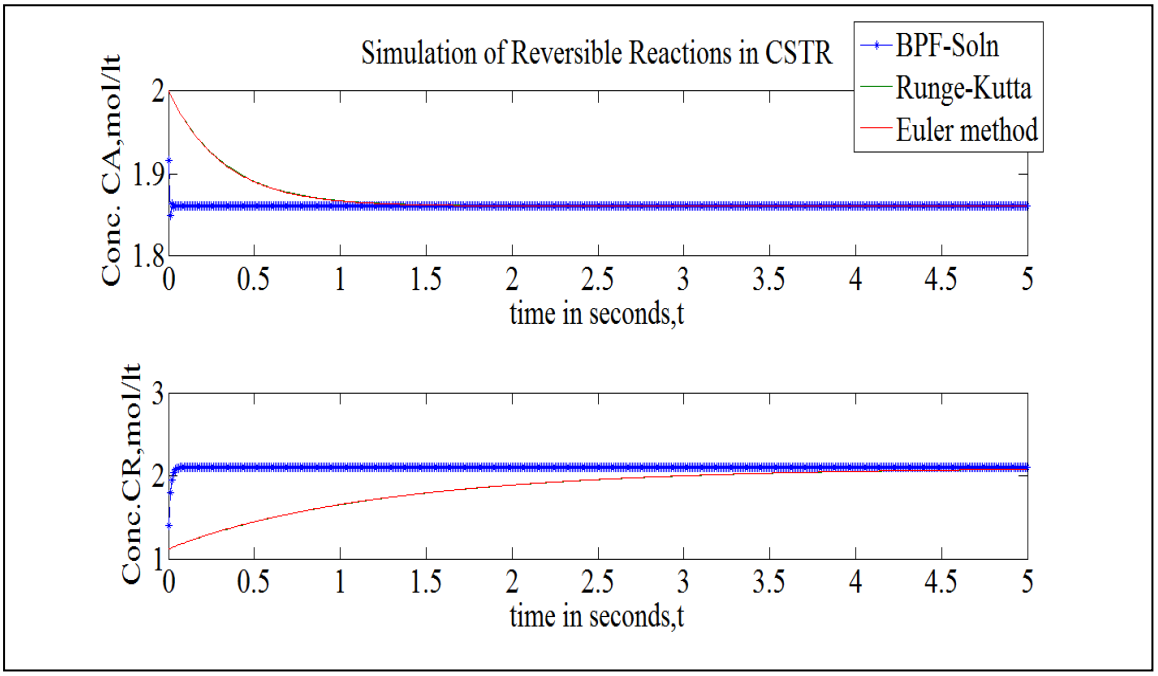

Figure-3.4: Concentration C_A, C_R vs. time and for n=500, T=5sec

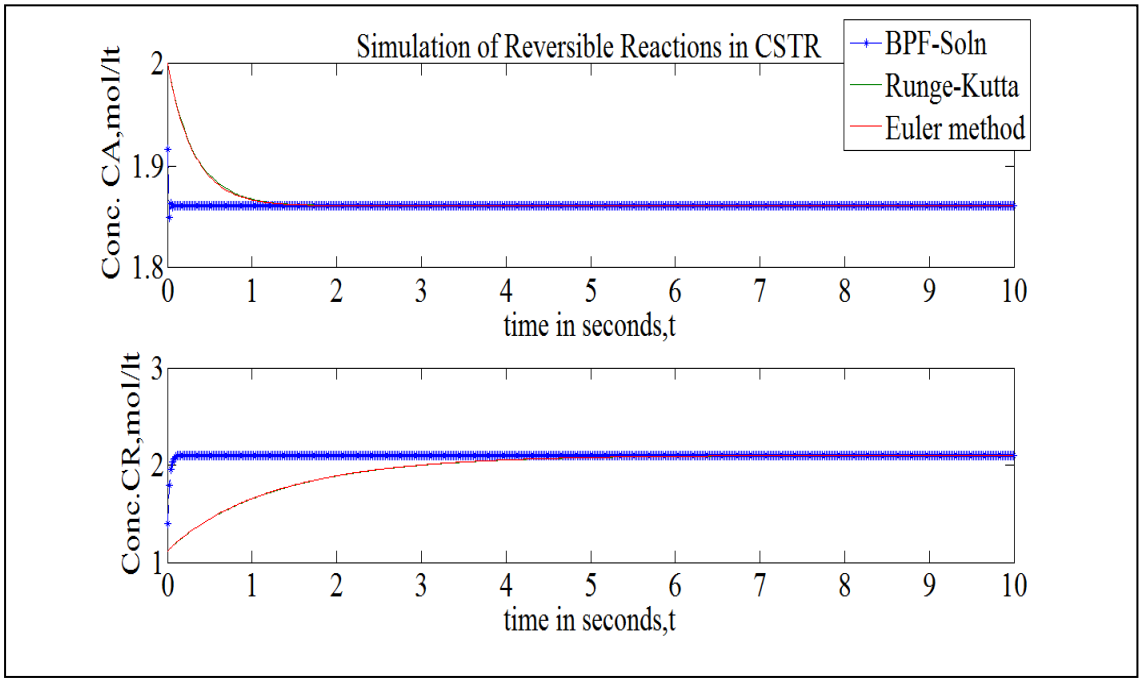

Figure-3.5: Concentration C_A, C_R vs. time and for n=500, T=10sec

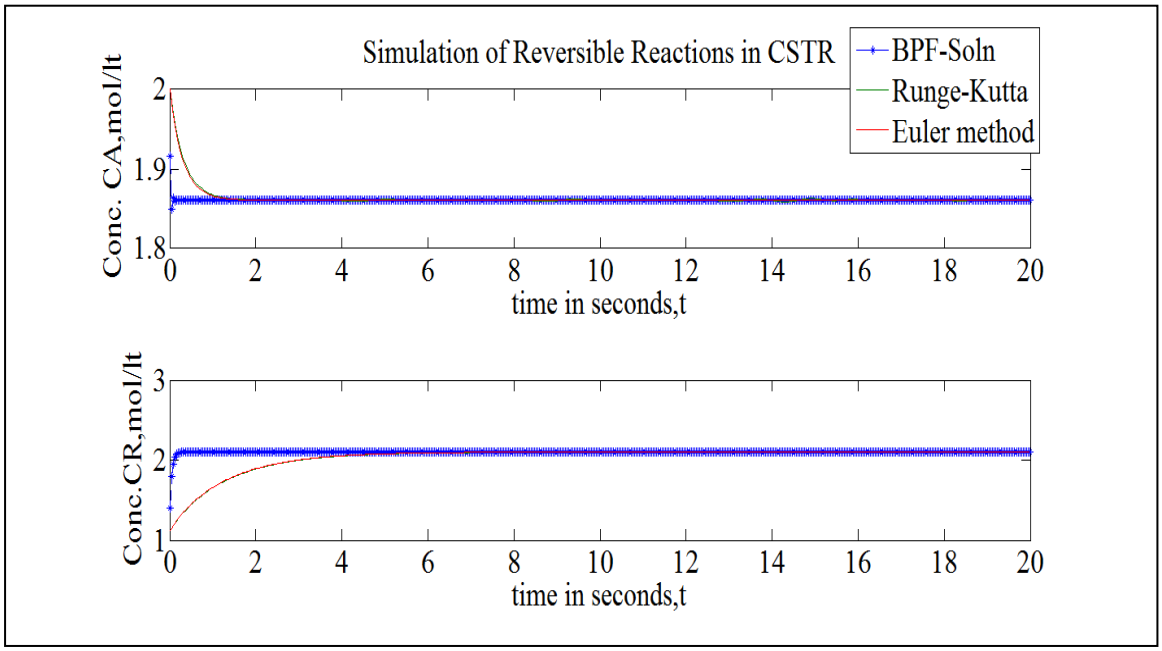

Figure-3.6: Concentration C_A, C_R vs. time and for n=500, T=20sec

3.2) Simulation of a Reversible reaction in a CSTR

Problem Statement:-Consider a set of differential equations for a first order reversible reaction in an isothermal CSTR. The block-pulse functions are used in solving the two differential model equations in an isothermal CSTR. The states are the concentration of A and R in the reactor. C_A is the concentration of the reactant A and C_R is the concentration of R. The parameter values are k_1=0.125 min^{-1}; k_2=0.325 min^{-1}. The input values used in the following simulation are F/V=1/8min^{-1}, C_{Af}=1000 mol/lt are listed in Table-3.2

Model Equations:-

$$\frac{dC_A}{dt} = \frac{F(C_{Af} - C_A)}{V} - k_1 C_A + k_2 C_R \quad (3.12)$$

$$\frac{dC_B}{dt} = -\frac{FC_R}{V} + k_1 C_A - k_2 C_R \quad (3.13)$$

Now expressing the two differential equations in terms of block-pulse functions as follows:

Let $C_A = x_1, C_R = x_2$

After substituting the given parameters in the above differential equations (3.12) and (3.13), we get the following equations,

$$\frac{dx_1}{dt} = 125 - 0.25x_1 + 0.325x_2 \quad (3.14)$$

$$\frac{dx_2}{dt} = -0.45x_1 + 0.125x_2 \quad (3.15)$$

On integrating the above equations (3.14) and (3.15), we get

$$x_1(t) - x_1(0) = 125Jdt - 0.25Jx_1(t) + 0.325Jx_2(t) \quad (3.16)$$

$$x_2(t) - x_2(0) = -0.45Jx_1(t) + 0.125Jx_2(t) \quad (3.17)$$

$$C_1{}^T\psi(t) - C_{10}{}^T\psi(0) = 125t - 0.25C_1{}^T P\psi(t) + 0.325C_2{}^T P\psi(t) \quad (3.18)$$

$$C_2{}^T\psi(t) - C_{20}{}^T\psi(0) = -0.45C_1{}^T P\psi(t) + 0.125C_2{}^T P\psi(t) \quad (3.19)$$

Equations (3.18) and (3.19) are further solved to obtain the values of C_1 and C_2.Here J is the integration operator, C_1 and C_2 are the block pulse coefficients, C_1 (0), C_2 (0) are the initial steady state values, P is the operational matrix and ψ represents block pulse function.

Table-3.2:- Parameter values of reversible reaction

Notations	Parameters	Steady state values
C_{Af}	1000mol/lt	
k_1	0.125 min^{-1}	
k_2	0.325 min^{-1}	
k_3	-	
F/V	1/8min^{-1}	
C_{As}	-	782.61
C_{Rs}	-	217.39

The Block-pulse function, Runge-Kutta (MATLAB solver) and Euler method codes are compiled which gives a set of output values of C_A, C_R which are plotted against the time intervals t=10, 20 and 40 with matrix sizes 100 and 500.

Graphs:-

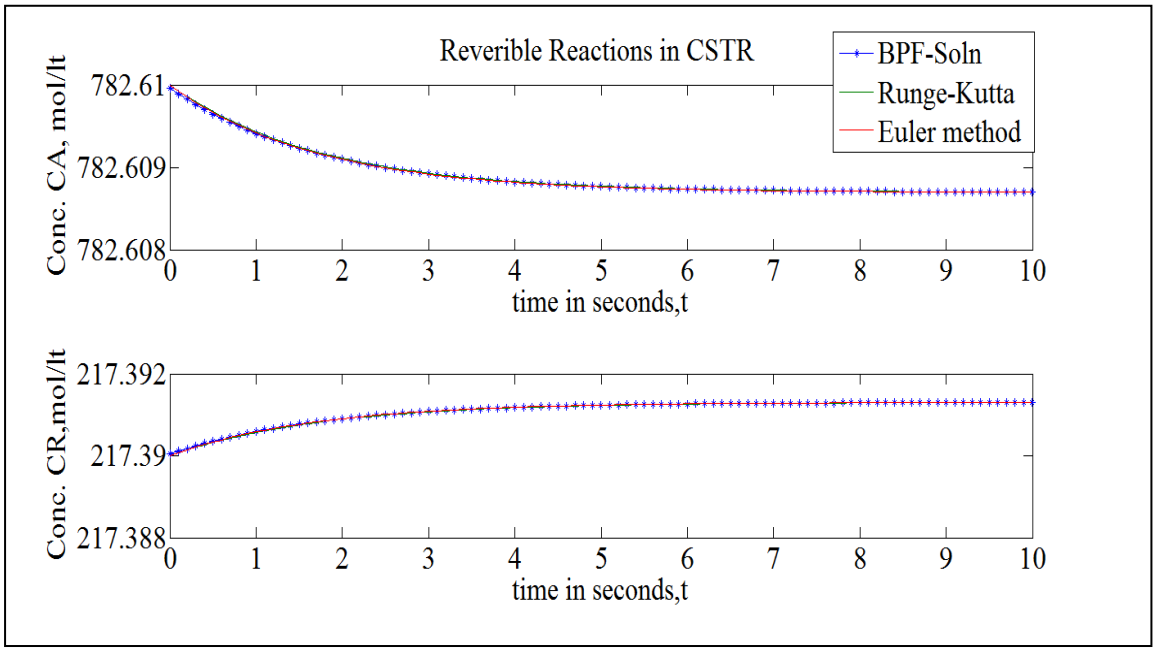

Figure-3.7: Concentration C_A, C_R vs. time and for n=100, T=10sec

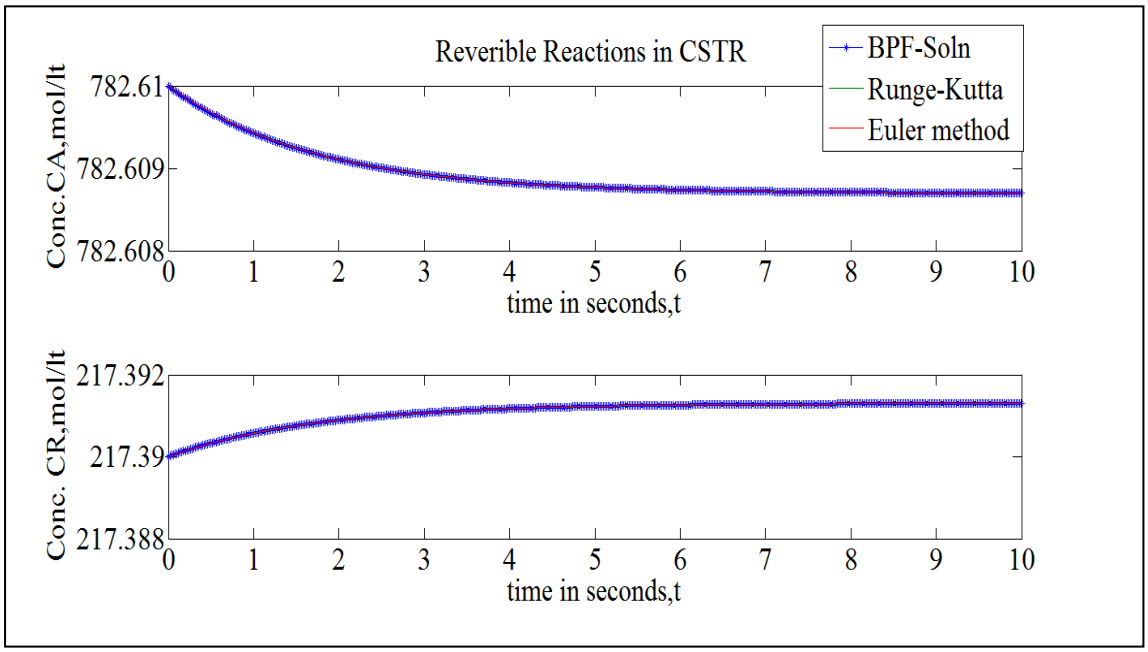

Figure-3.8: Concentration C_A, C_R vs. time and for n=500, T=10sec

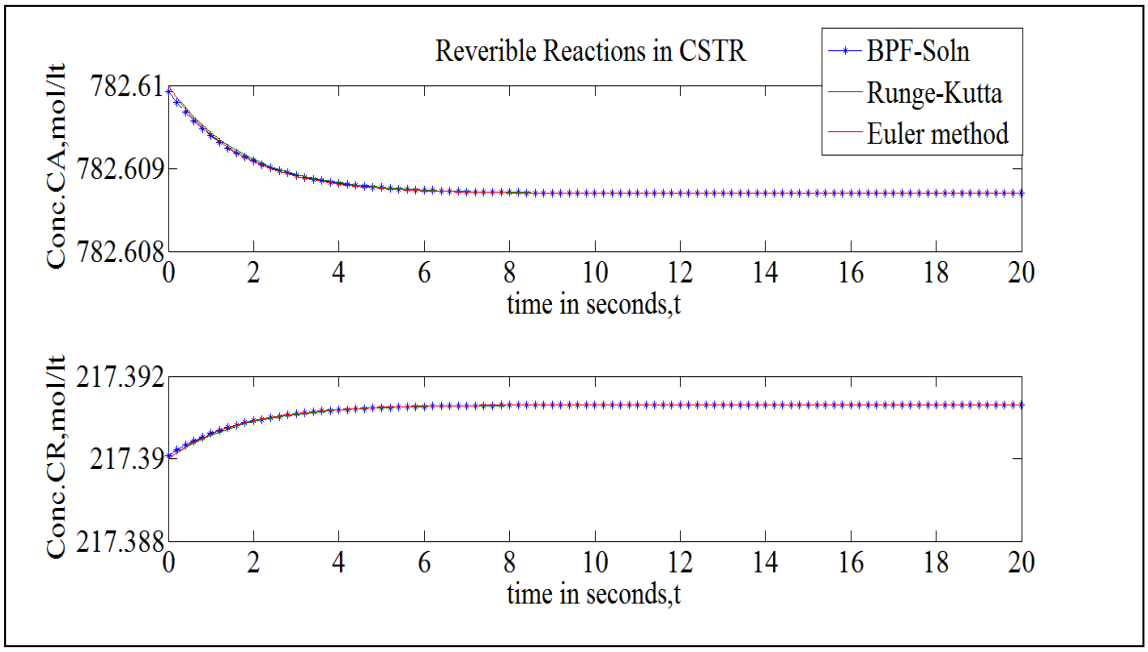

Figure-3.9: Concentration C_A, C_R vs. time and for n=100, T=20sec

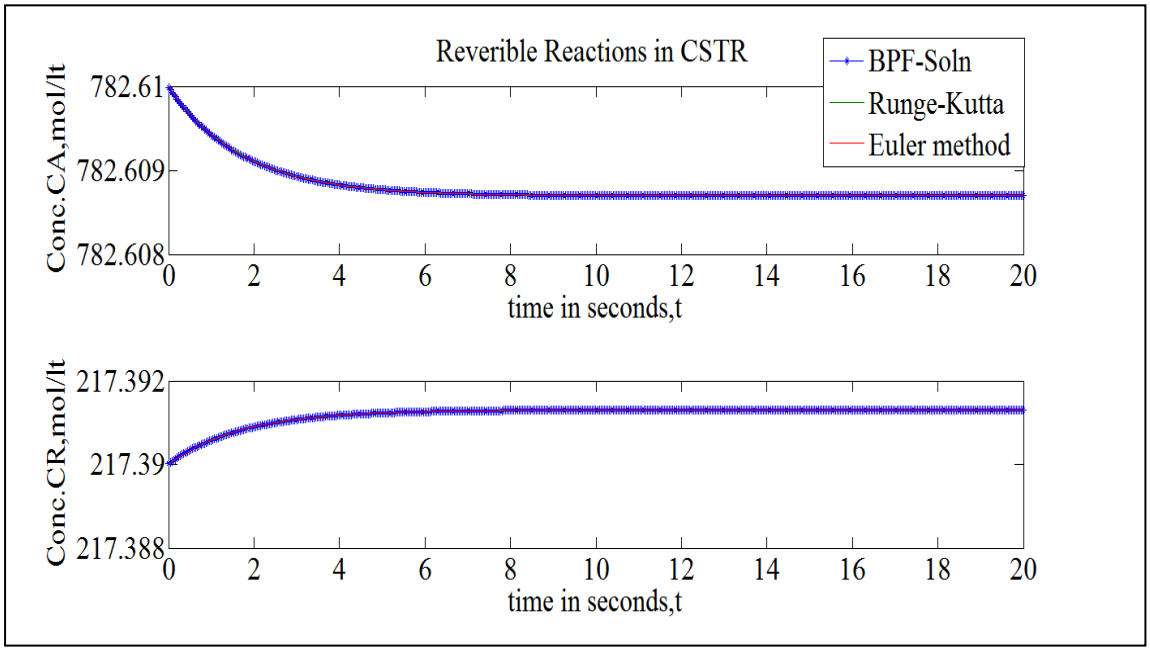

Figure-3.10: Concentration C_A, C_R vs. time and for n=500, T=20sec

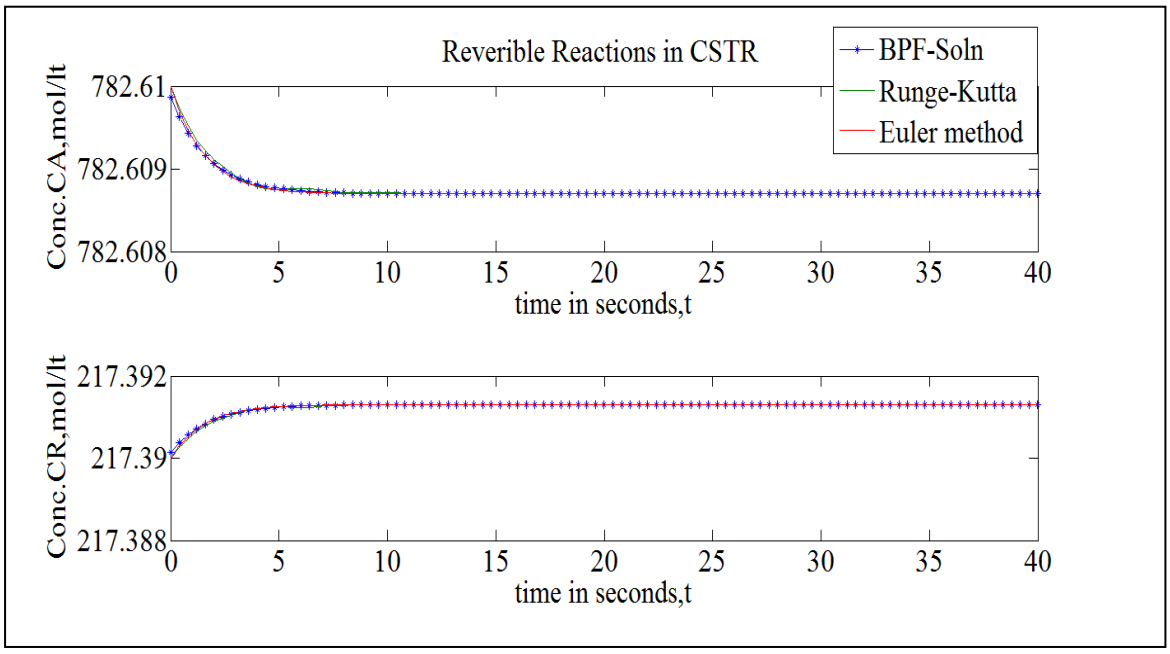

Figure-3.11: Concentration C_A, C_R vs. time and for n=100, T=40sec

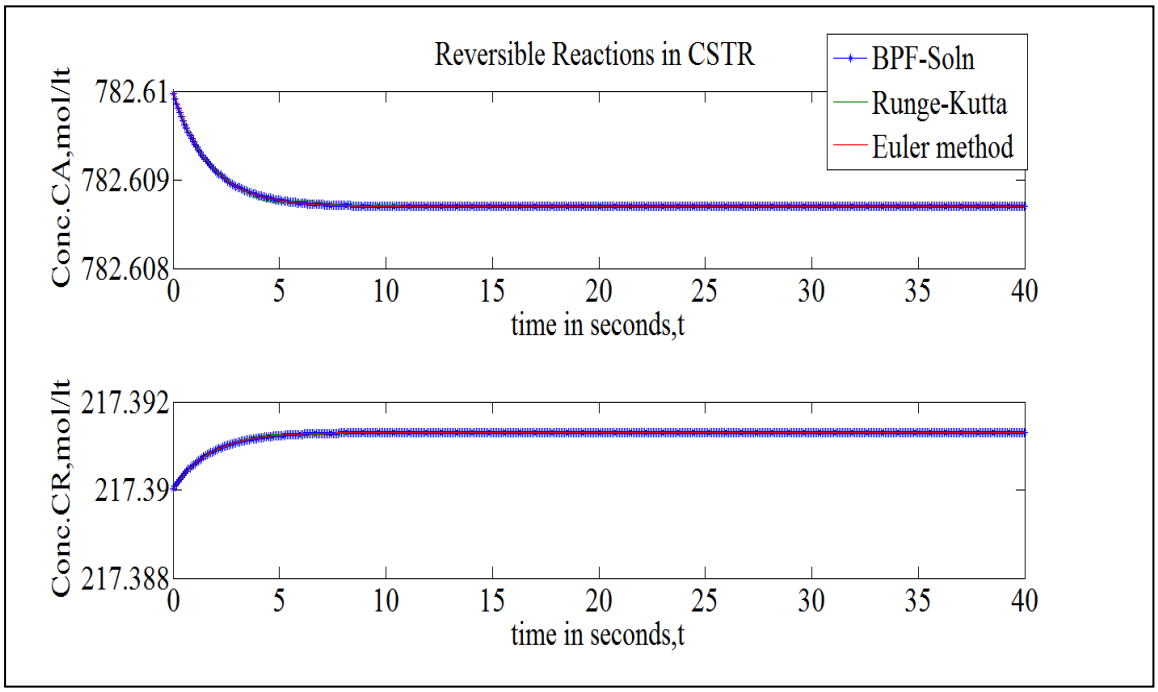

Figure-3.12: Concentration C_A, C_R vs. time and for n=500, T=40sec

The above plots generated from BPF algorithm using MATLAB software and from Euler and Runge-Kutta methods (inbuilt MATLAB solver ODE-45), represent the transient response to initial conditions from steady state values, C_{As}=782.61, C_{Rs}=217.39.The new steady state values reached by the C_A and C_R are 782.6088 and 217.3915 respectively. The plots show that a similar trend in concentrations is observed with three kinds of data (BPF& Euler and Runge –Kutta methods) obtained.

3.3) Simulation of an Irreversible Reaction in a CSTR

Problem Statement: - Consider a set of differential equations for a first order irreversible reaction in an isothermal CSTR. Ethylene oxide (A) is reacted with water (B) to produce ethylene glycol(R). Water is in large excess. A CSTR is used at a constant temperature. The block-pulse functions are used in solving the two differential model equations in an isothermal CSTR. The states are the concentration of A and R in the reactor. C_A is the concentration of the reactant A and C_R is the concentration of R. The parameter values are k_1=0.311 min^{-1}. The input values used in the following simulation are F/V=0.0777min^{-1}, C_{Af} =0.5mol/lt are listed in Table-3.3

Model Equations:-

$$\frac{dC_A}{dt} = \frac{F(C_{Af} - C_A)}{V} - k_1 C_A \quad \ldots\ldots (3.20)$$

$$\frac{dC_R}{dt} = -\frac{FC_R}{V} + k_1 CA \quad \ldots\ldots (3.21)$$

Now expressing the two differential equations in terms of block-pulse functions as follows:

Let $C_A = x_1, C_R = x_2$

After substituting the given parameters in the above differential equations (3.22) and (3.23) we get the following equations,

$$\frac{dx_1}{dt} = 0.0389 - 0.3887 x_1 \quad \ldots\ldots (3.22)$$

$$\frac{dx_2}{dt} = 0.311 x_1 + 0.0777 x_2 \quad \ldots\ldots (3.23)$$

On integrating the above equations (3.24) and (3.25), we get

$$x_1(t) - x_1(0) = 0.0389 Jdt - 0.3887 J x_1(t) \quad \ldots\ldots (3.24)$$

$$x_2(t) - x_2(0) = 0.311 J x_1(t) + 0.0777 J x_2(t) \quad \ldots\ldots (3.25)$$

$$C_1{}^T \psi(t) - C_{10}{}^T \psi(0) = 0.0389 t - 0.3887 C_1{}^T P \psi(t) \quad \ldots\ldots (3.26)$$

$$C_2{}^T \psi(t) - C_{20}{}^T P \psi(0) = 0.311 C_1{}^T P \psi(t) + 0.0777 C_2{}^T P \psi(t) \quad \ldots\ldots (3.27)$$

Equations (3.26) and (3.27) are further solved to obtain the values of C_1 and C_2.Here J is the integration operator, C_1 and C_2 are the block pulse coefficients, C_1 (0), C_2 (0) are the initial steady state values, P is the operational matrix and ψ represents block pulse function.

Table-3.3:- Parameter values of irreversible reactions in CSTR

Notations	Parameters	Steady state values
C_{Af}	0.5mol/lt	
k_1	0.311 min^{-1}	
k_2	-	
k_3	-	
F/V	0.0777min^{-1}	
C_{As}	-	0.1047
C_{Rs}	-	0.395

The Block-pulse function, Runge-Kutta (MATLAB solver) and Euler method codes are compiled which gives a set of output values of C_A, C_R which are plotted against the time intervals t=10, 20 and 40 with matrix sizes 100 and 500.

Graphs

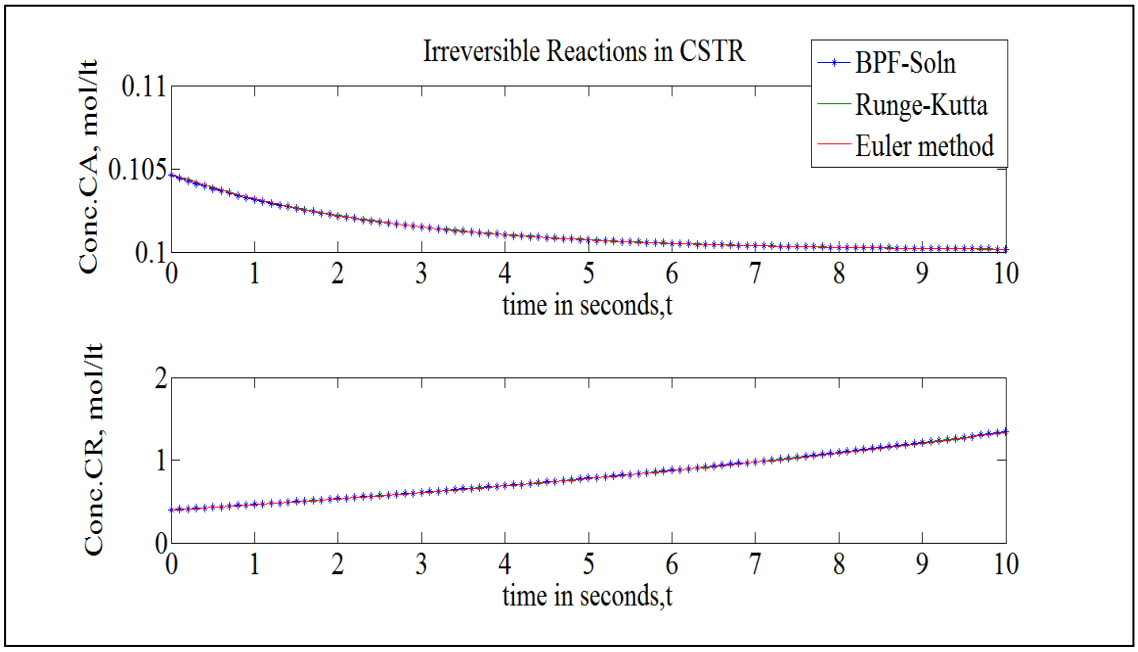

Figure-3.13: Concentration C_A, C_R vs. time and for n=100, T=10sec

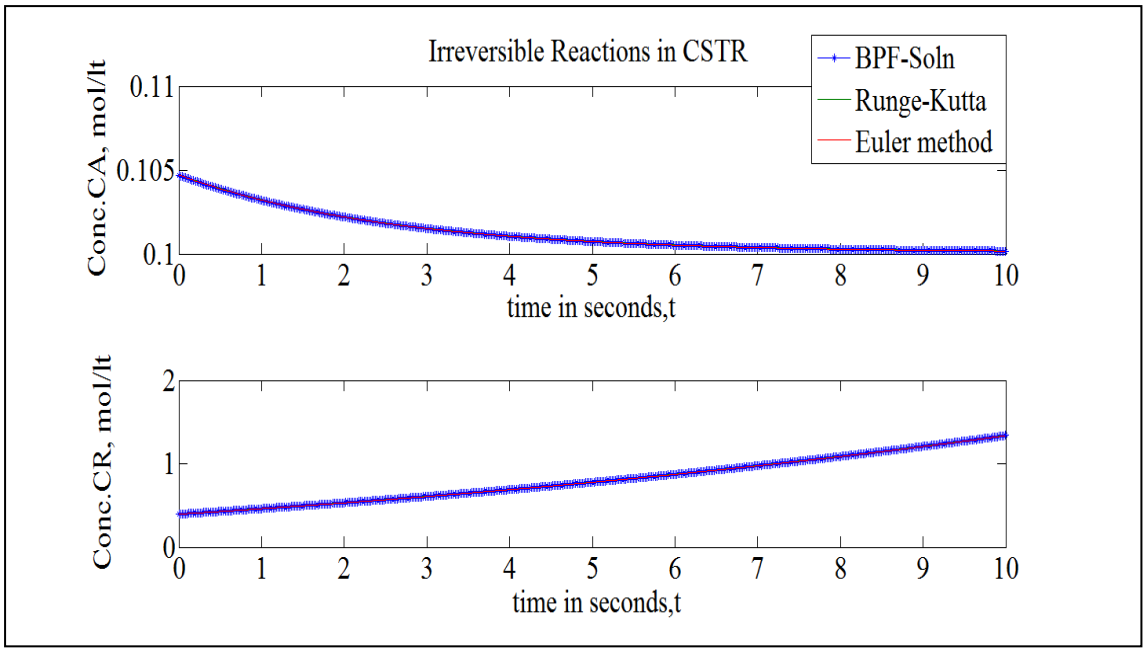

Figure-3.14: Concentration C_A, C_R vs. time and for n=500, T=10sec

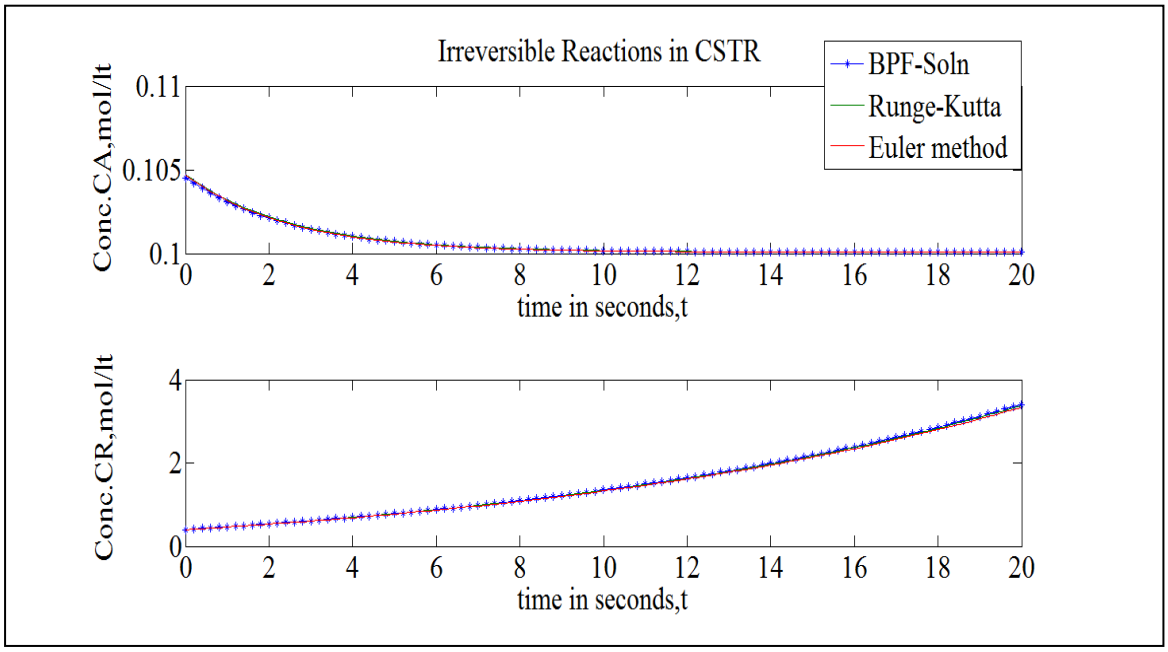

Figure-3.15: Concentration C_A, C_R vs. time and for n=100, T=20sec

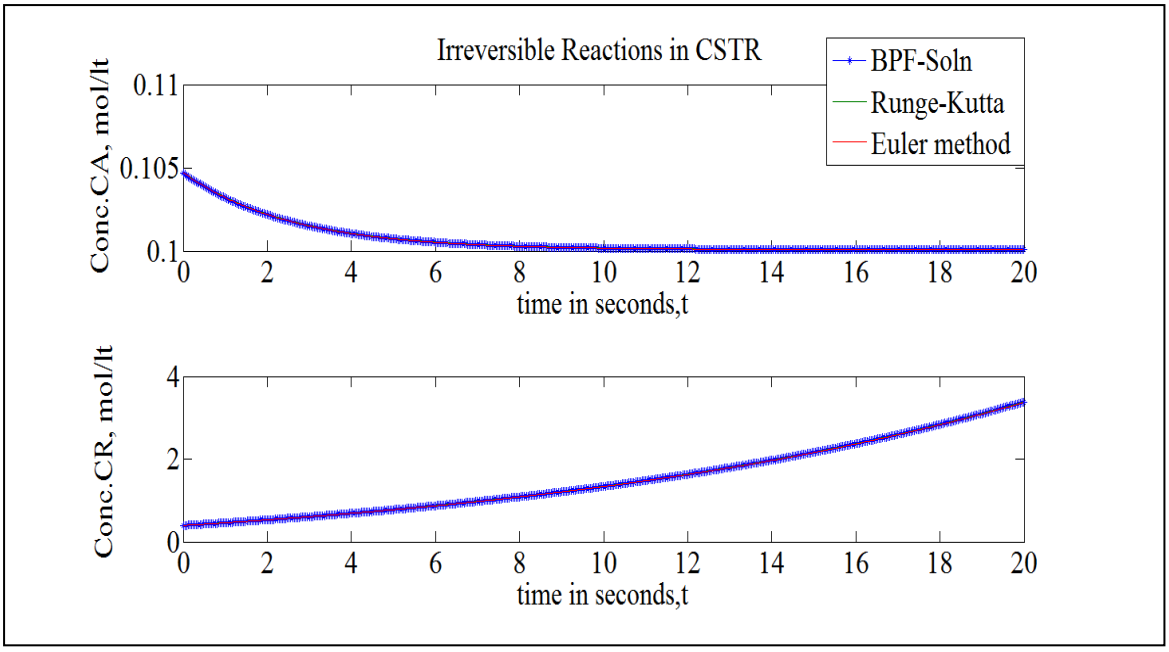

Figure-3.16: Concentration C_A, C_R vs. time and for n=500, T=20sec

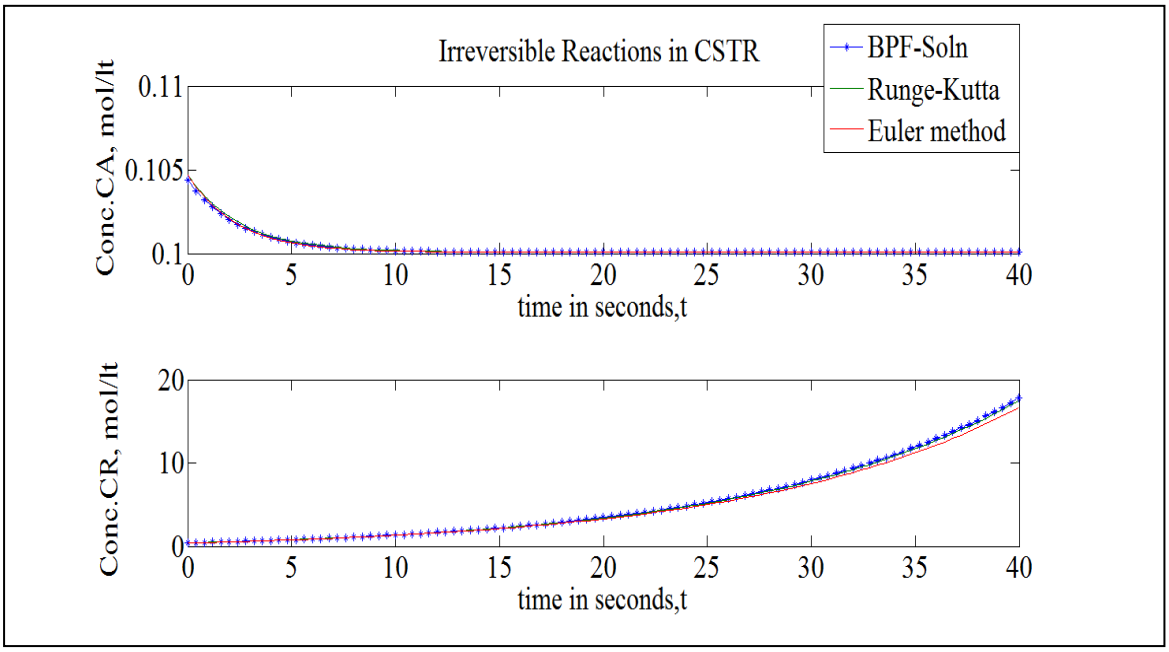

Figure-3.17: Concentration C_A, C_R vs. time and for n=100, T=40sec

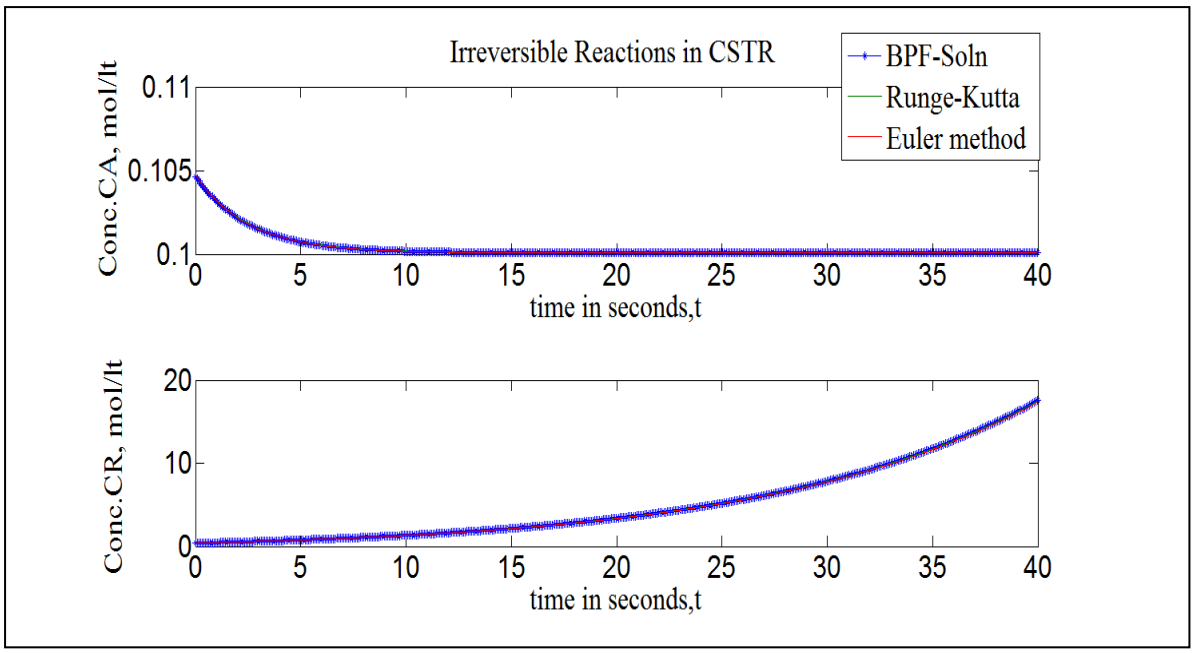

Figure-3.18: Concentration C_A, C_R vs. time and for n=500, T=40sec

The above plots generated from BPF algorithm using MATLAB software and from Euler and Runge-Kutta methods (inbuilt MATLAB solver ODE-45), represent the transient response to initial conditions from steady state values, C_{As}=0.1047, C_{Rs}=0.395.The plots show that a similar trend in concentrations is observed with three kinds of data (BPF& Euler and Runge – Kutta methods) obtained.

3.4) Simulation of a Jacketed Heater

Problem Statement: - Consider a jacketed stirred tank heater in which heat is circulated through the jacket and heat flows between the jacket and vessel increase energy content of the vessel fluid. The following parameters and steady-state values are used for the simulation. F_s=1ft^3/min,ρC_p=61.3Btu/(^{0}F.ft^3),$\rho_j C_{pj}$=61.3Btu/(^{0}F.ft^3),T_{is}=50^{0}F,T_s=125^{0}F,V=10ft^3,T_{jis}=200^{0}F ,T_{js}=150^{0}F,V_j=2.5ft^3,UA=183.9Btu/(^{0}F.min),F_{js}=1.5ft^3/min.The states are the fluid temperature, T flowing through the stirred tank heater and the jacket temperature flowing in the jacket surrounding the heater, T_j. The rate of heat transfer from jacket fluid to vessel fluid is given as:

Model Equations:-

$$Q = UA(T_j - T) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (3.28)$$

$$\frac{dT}{dt} = \frac{F(T_i - T)}{V} + \frac{UA(T_j - T)}{V\rho Cp} \ldots \ldots \ldots \ldots \ldots \ldots (3.29)$$

$$\frac{dT_j}{dt} = -\frac{F_j(T_{jin} - T_j)}{V_j} - \frac{UA(T_j - T)}{V_j \rho_j C_{pj}} \ldots \ldots \ldots \ldots \ldots (3.30)$$

Now expressing the two differential equations in terms of block-pulse functions as follows:

Let $T = x_1, T_j = x_2$

After substituting the given parameters in the above differential equations, we get the following equations,

$$\frac{dx_1}{dt} = 51.245 - 0.4083x_1 + 0.3x_2 \ldots \ldots \ldots \ldots \ldots (3.31)$$

$$\frac{dx_2}{dt} = 120 + 1.2x_1 - 1.8x_2 \ldots \ldots \ldots \ldots \ldots (3.32)$$

On integrating the above equations (3.31) and (3.32) we get

$$x_1(t) - x_1(0) = 51.245Jdt - 0.4083Jx_1(t) + 0.3Jx_2(t) \ldots \ldots \ldots (3.33)$$

$$x_2(t) - x_2(0) = 120t + 1.2Jx_1(t) - 1.8Jx_2(t) \ldots \ldots \ldots \ldots (3.34)$$

$$C_1{}^T\psi(t) - C_{10}{}^T\psi(0) = 51.245t - 0.4083C_1{}^TP\psi(t) + 0.3C_2{}^TP\psi(t) \ldots \ldots \ldots (3.35)$$

$$C_2{}^T\psi(t) - C_{20}{}^T\psi(0) = 120t + 1.2C_1{}^TP\psi(t) - 1.8C_2{}^TP\psi(t) \ldots \ldots \ldots (3.36)$$

Equations (3.35) and (3.36) are further solved to obtain the values of C_1 and C_2.Here J is the integration operator, C_1 and C_2 are the block pulse coefficients, C_1 (0), C_2 (0) are the initial steady state values, P is the operational matrix and ψ represents block pulse function.

Table-3.4:- Parameter values of Jacketed Heater

Notations	Parameters	Steady state values
F_s	$1ft^3/min$	
ρC_p	$61.3Btu/(^0F.ft^3)$,	
ρC_{pj}	$61.3Btu/(^0F.ft^3)$,	
T_{is}	50^0F	
T_s		125^0F
V	$10ft^3$	
T_{jis}	200^0F	
T_{js}		150^0F
V_j	$2.5ft^3$	
UA	$183.9Btu/(^0F.min)$	
F_{js}	$1.5ft^3/min$	

The Block-pulse function, Runge-Kutta (MATLAB solver) and Euler method codes are compiled which gives a set of output values of C_A, C_R which are plotted against the time intervals t=10, 20 and 40 with matrix sizes 100 and 500.

Graphs

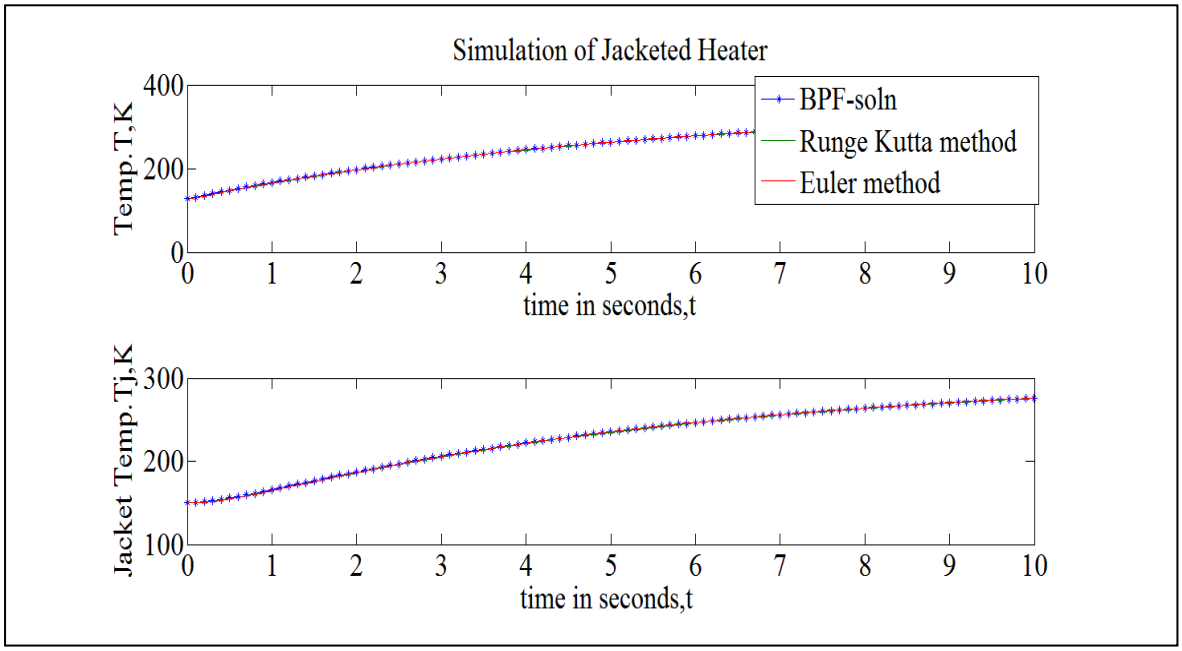

Figure-3.19: Temperature T, T_j vs. time and for n=100, T=10 sec

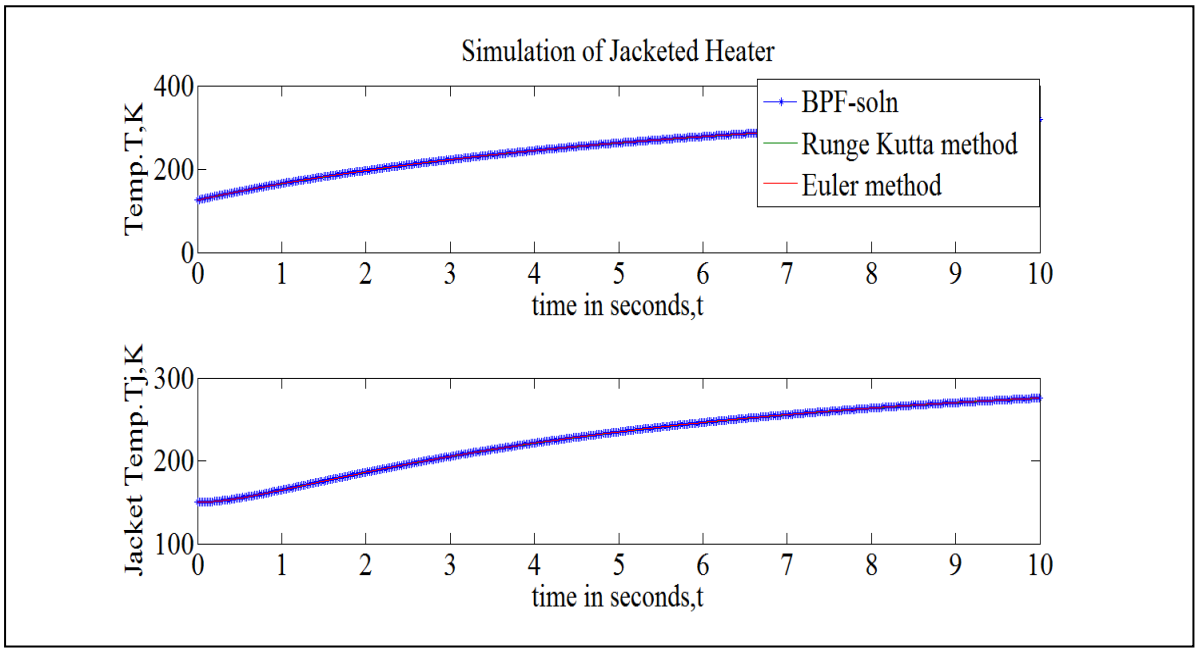

Figure-3.20: Temperature T, T_j vs. time and for n=500, T=10 sec

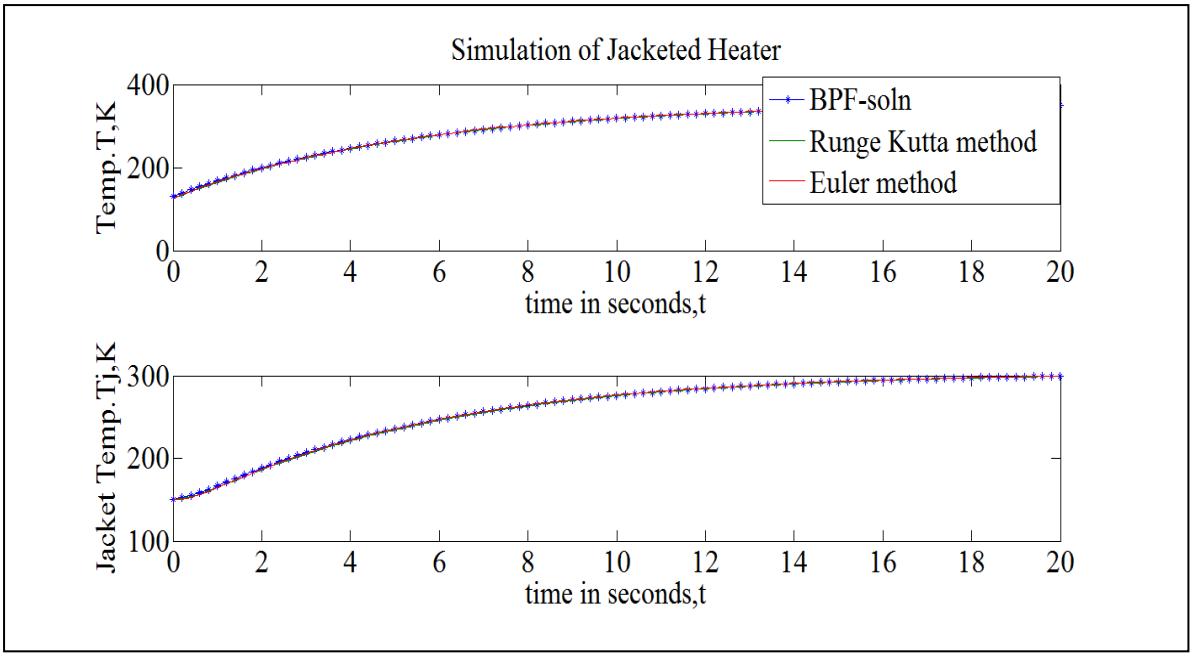

Figure-3.21: Temperature T, T_j vs. time and for n=100, T=20 sec

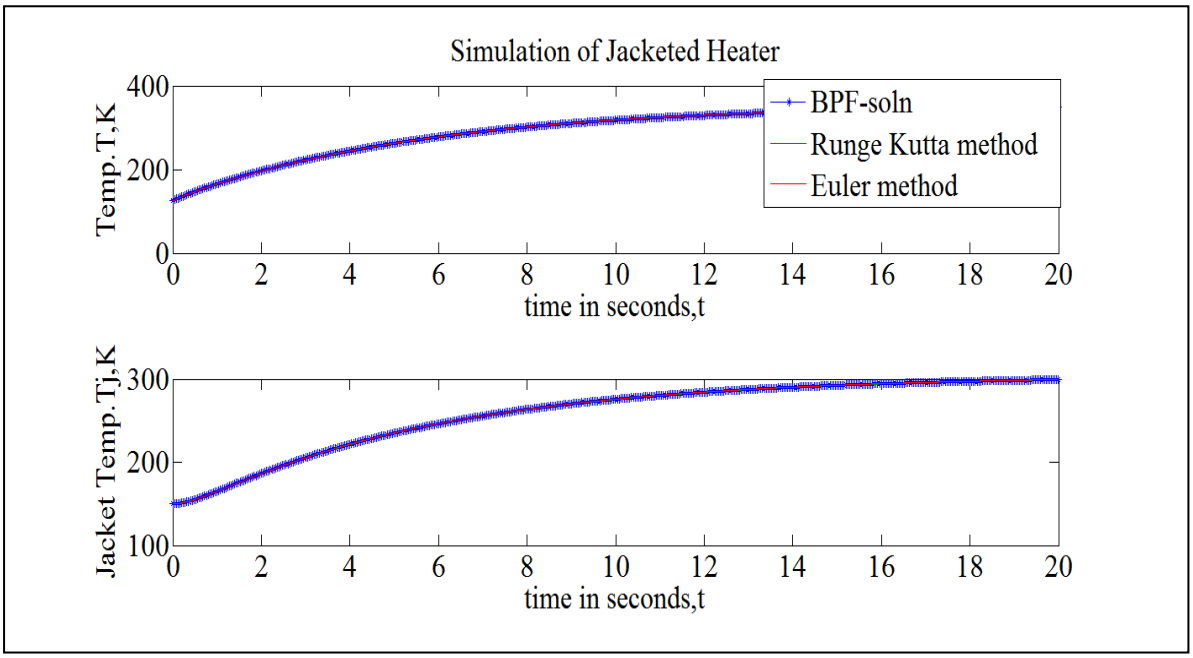

Figure-3.22: Temperature T, T_j vs. time and for n=500, T=20 sec

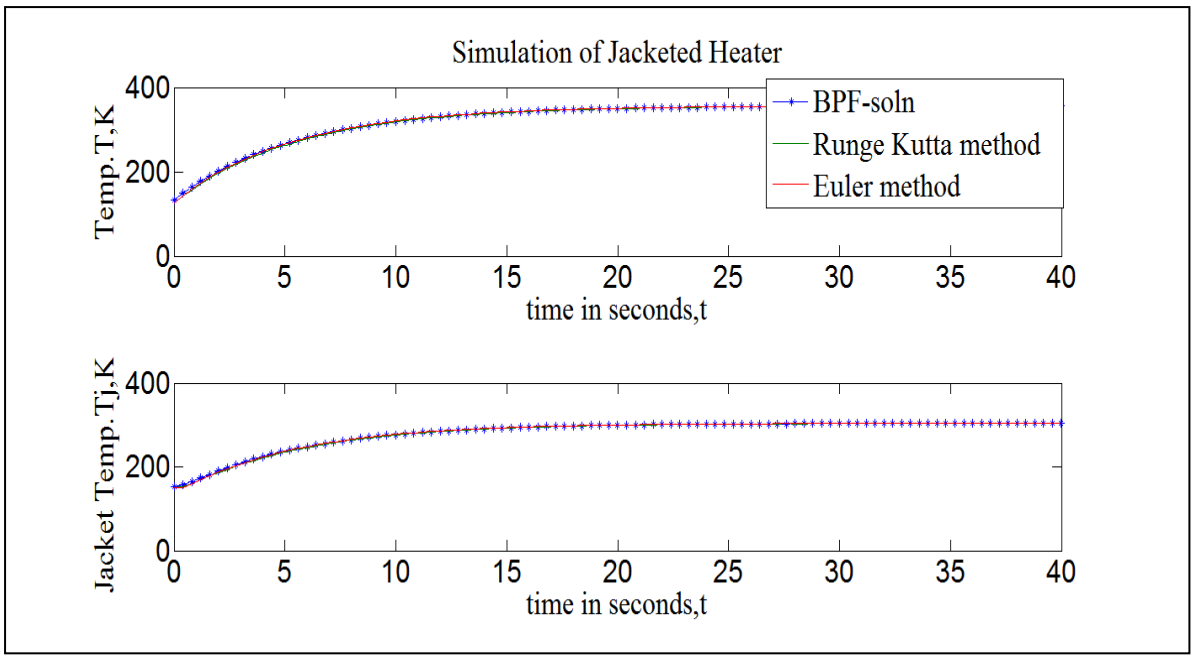

Figure-3.23: Temperature T, T_j vs. time and for n=100, T=40 sec

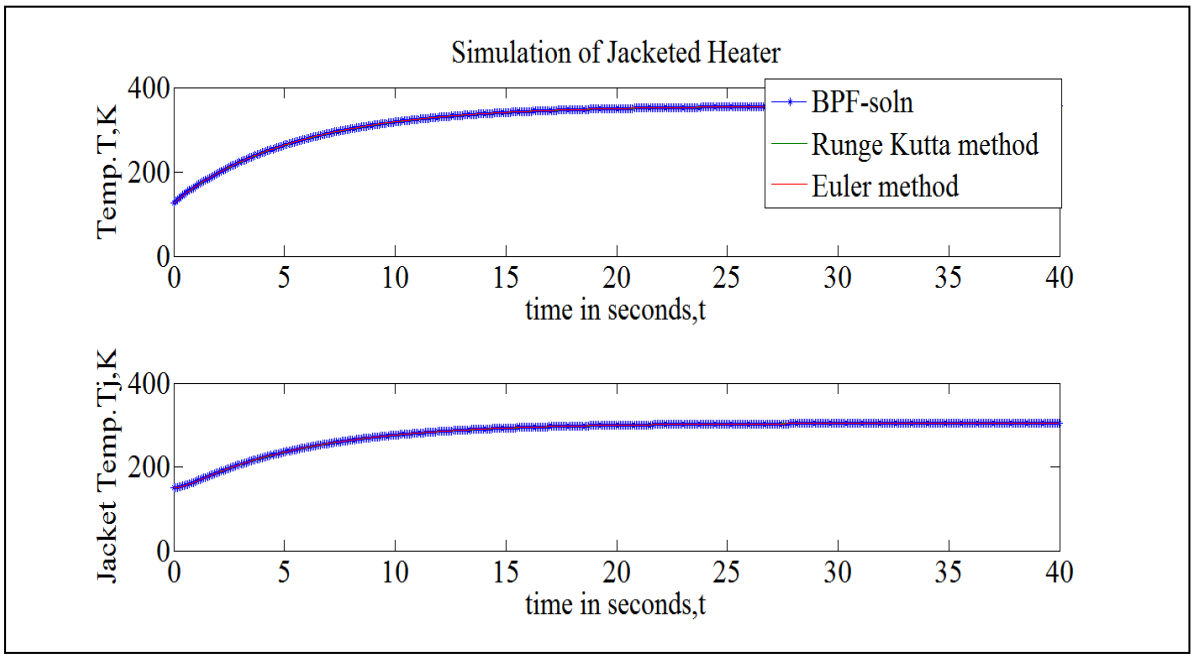

Figure-3.24: Temperature T, T_j vs. time and for n=500, T=40 sec

The above plots generated from BPF algorithm using MATLAB software and from Euler and Runge-Kutta methods (inbuilt MATLAB solver ODE-45), represent the transient response to initial conditions from steady state values, T_s=125,Tj_s=150. The plots show that a similar trend in temperatures is observed with three the kinds of data (BPF& Euler and Runge –Kutta methods) obtained.

3.2.1 Results and Discussions

- The state/output obtained from dynamic simulation using block-pulse functions (BPF) is plotted against time. This plot is compared with the plots generated using Euler method and Runge-Kutta method (MATLAB solver ODE-45). The plots are in a good agreement to each other, which proves the potency of BPF for simulation of such kinds of systems considered.
- Consistent results are obtained for matrix dimensions 100 and 500 at various time intervals such as 10, 20 and 40 seconds. Similarly, results can be extended further to other matrix dimensions.
- For small time intervals, the plots show a drop or an increase in concentration in a linear manner. However as time proceeds, i.e. for large time intervals, the complete curve gets developed. Thus, the results of all four simulations are in a good fit with each other.
- With small operational matrix size, there may be a deviation in the results of BPF compared to those obtained using other numerical methods such as Euler and Runge-Kutta. However, as the operational matrix size increases largely, the deviation from the two solutions reduces. Besides existing numerical techniques, block pulse functions also provide an efficient solution for the CSTR simulation.

3.3 NON LINEAR SYSTEMS

According to physics and other sciences, a nonlinear system does not obey the superposition principle – which means the output of the system is not directly related to the input, contrary to the linear systems. Many physical quantities have an upper bound which once reached the system loses linearity; these quantities are vehicle's velocity or electrical signals. The differential equations of some thermal, biological systems are inherently nonlinear in nature. Thus considering the non-linearity directly while analysing and designing controllers for such systems is always an advantage. In addition, many mechanical systems are subject to non-linearity by friction. Relays, which are part of many practical control systems, are inherently nonlinear. Other examples include ferromagnetic cores in electrical machines and transformers are often described with nonlinear magnetization curves and equations.

Mathematically, a nonlinear system of equations is defined as the set of simultaneous equations in which the unknowns (or the unknown functions in the case of differential equations) appear as variables of a polynomial of degree higher than one. Otherwise, it can be defined as that system of equations to be solved which cannot be written as a linear combination of unknown variables or functions that appear in it (them). Non-linearity in known functions within the equations is treated insignificant. A differential equation is linear if it is linear in terms of the unknown function and its derivatives, even if nonlinear in terms of the other variables appearing in it. A nonlinear system of equations describes the behaviour of a nonlinear system.

Engineers, physicists, mathematicians and many other scientists are primarily interested in nonlinear problems because of their inherent nonlinear in nature. Because of the difficulty involved in solving nonlinear equations, they are commonly approximated by linear equations (linearization). Linearization achieves some amount of accuracy and some range for the input values, but some interesting phenomena such as chaos and singularities are hidden. Behaviour of a nonlinear system can be characterized to be chaotic, unpredictable or counter intuitive. Although such chaotic behaviour may resemble random behaviour, it is absolutely not random. For example, some aspects of the weather are seen to be chaotic, where the system becomes very sensitive for small changes in inputs producing complex effects throughout. Thus accurate long-term forecasts are impossible with current technology because of non-linearity in systems.

3.3.1 Differences between Linear and Nonlinear systems:-

- Linear systems obey the properties of superposition and homogeneity.
- Non-Linear systems do not obey superposition and homogeneity.
- Linear systems possess one equilibrium point at the origin while many equilibrium points are possessed by non-linear systems.
- For nonlinear systems, stability needs to be defined precisely.
- For forced response for the nonlinear systems principle of superposition does not hold good.
- Linearity cannot be classified however Non-linearity can be broadly classified.

3.3.2 Definition of Linear and Nonlinear Systems:-

Linear systems must obey two important properties, superposition and homogeneity. According to the principle of superposition for two different inputs, x and y, in the domain of the function f,

$$\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{y})=\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{f}(\boldsymbol{y}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{3.37})$$

The property of homogeneity states that for a given input, x, in the domain of the function f, and for any real number k,

$$\boldsymbol{f}(\boldsymbol{kx})=\boldsymbol{kf}(\boldsymbol{x}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{3.38})$$

Any function that does not satisfy superposition and homogeneity is non-linear; there is no unifying characteristic of nonlinear systems, except for not satisfying the two above-mentioned properties.

Linear Time Invariant (LTI) systems are commonly described by the equation:

$$\frac{\boldsymbol{dx}}{\boldsymbol{dt}}=\boldsymbol{Ax}+\boldsymbol{Bu} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{3.39})$$

In this equation, x is the vector of n state variables, u is the control input, and A is a matrix of size (n-by-n), and B is a vector of appropriate dimensions. The equation determines the dynamics of the response. It is sometimes called a state-space realization of the system.

Non-Linear Systems are commonly described by the equation:

$$\frac{\boldsymbol{dx}}{\boldsymbol{dt}}=\boldsymbol{f}(\boldsymbol{x}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{3.40})$$

General Properties of Linear and Nonlinear Systems:-

	Linear Systems & $\frac{dx}{dt} = Ax$	Non-Linear Systems & $\frac{dx}{dt} = f(x)$
EQUILIBIUM POINTS A point where the system can stay forever without moving.	**UNIQUE** If A has rank n, then x_e=0, otherwise the solution lies in the null space of A.	**MULTIPLE** $f(x_e)$=0 n nonlinear equations in n unknowns 0 → +∞ solutions
ESCAPE TIME	x → +∞ as t → +∞	The state can go to infinity in finite time.
STABILITY	The equilibrium point is stable if all eigenvalues of A have negative real part, regardless of initial conditions.	About an equilibrium point: • Dependent on IC • Local vs. Global stability important • Possibility of limit cycles **LIMIT CYCLES** • A unique, self-excited oscillation • A closed trajectory in the state space • Independent of IC
FORCED RESPONSE	$\frac{dx}{dt} = Ax + Bu$ • The principle of superposition holds. • I/O stability → bounded input, bounded output	$\frac{dx}{dt} = f(x, u)$ • The principle of superposition does not hold in general. • The I/O ratio is not unique in general, may also not be single valued.

	• Sinusoidal input → sinusoidal output of same frequency	**CHAOS** Complicated steady-state behavior, may exhibit randomness despite the deterministic nature of the system.

3.3.3 Non Linear Differential Equations:-

A system of differential equations not obeying homogeneity and superposition are said to be nonlinear. Extremely diverse are problems involving nonlinear differential equations, and methods of solution or analysis are problem dependent and cannot be generalized. The Navier–Stokes equations in fluid dynamics, Lotka–Volterra equations in biology and Non-Isothermal CSTR in reaction engineering are some of the important examples of nonlinear differential equations.

A combination of known solutions into new solutions is not generally possible when dealing with nonlinear problems which can be regarded as one of the greatest difficulties. A family of linearly independent solutions can be used to construct general solutions through the superposition principle for linear systems. A good example of this is one-dimensional heat transport with Dirichlet boundary conditions, the solution of which can be written as a time-dependent linear combination of sinusoids of differing frequencies; making solutions very flexible. For nonlinear equations, several very specific solutions can be found very often; however new solutions cannot be constructed as it does not satisfy the superposition principle.

Ordinary differential equations:-

First order ordinary differential equations are often exactly solved by separation of variables, especially for autonomous equations. For example, the nonlinear equation

$$\frac{du}{dx} = -u^2 \quad \ldots\ldots (3.41)$$

$$u = \frac{1}{x+c} \quad \ldots\ldots (3.42)$$

has eq-5 as a general solution (and also $u = 0$ as a particular solution, corresponding to the limit of the general solution when C tends to the infinity). The equation is nonlinear because it may be written as

$$\frac{du}{dx} + u^2 \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad (3.43)$$

and the left-hand side of the equation is not a linear function of u and its derivatives. Note that if the u^2 term were replaced with u, the problem would be linear (the exponential decay problem).

Second and higher order ordinary differential equations (more generally, systems of nonlinear equations) rarely yield closed form solutions, though implicit solutions and solutions involving non-elementary integrals are encountered.

Common methods for the qualitative analysis of nonlinear ordinary differential equations include:

- Examination of any conserved quantities, especially in Hamiltonian systems.
- Examination of dissipative quantities (see Lyapunov function) analogous to conserved quantities.
- Linearization via Taylor expansion.
- Change of variables into something easier to study.
- Bifurcation theory.
- Perturbation methods

3.4 SIMULATION OF NON-LINEAR SYSTEMS

Blood Glucose Control in Diabetic Patients:-

There are many innate feedback control loops within a human body. For instance, pancreas regulates blood glucose by producing insulin. Food is consumed and later broken down by the digestive system; as a result blood glucose level rises which in turn stimulates insulin production. Glucose is broken down by the cells with the help of insulin. Insulin is not produced by diabetes mellitus patients of Type I .Hence patient must be administered insulin shots at several regular time intervals to regulate the blood glucose level. Thus, a typical patient here is serving as a control system. Administration of insulin shots coinciding with the meal concentration comprises the feed forward control nature. Other actions such as dosage changes based on glucose measurements obtained from finger pricks and analysis of glucose

strips. Hyperglycaemia leads to blindness, cardiovascular problems in the long run while fainting, diabetic comas are problems due to it in the short run.

Development of closed-loop insulin delivery systems has been highly motivated and external pumps provided with insulin reservoirs which can deliver insulin directly instead of shots are used by the current technology.

3.5) Problem Statement: - Consider a diabetic that is modelled using the following set of parameters

Model equations:-

$$\frac{dG}{dt} = -p_1G - x(G_b + G) + \frac{G_{meal}}{V_1} \quad \ldots (3.44)$$

$$\frac{dX}{dt} = -p_2X + p_3I \quad \ldots (3.45)$$

$$\frac{dI}{dt} = -n(I + I_b) + \frac{U}{V_1} \quad \ldots (3.46)$$

Where G and I represent the deviation in blood glucose and insulin concentrations respectively. Also X is proportional to insulin concentration in a remote compartment. The inputs are G_{meal}, a meal disturbance input of glucose, U manipulated insulin infusion rate. The parameters include p_1, p_2, p_3, n, V_1.Other parameters are G_b, I_b are the basal values of blood and insulin concentration. These values are used to determine the basal infusion rate of insulin necessary to maintain a steady state.

Let G=x_1, X=x_2, I=x_3

After substituting the given parameters in the above differential equations, we get the following equations:

$$\frac{dx_1}{dt} = -16.667x_1 - 4.5x_2 - x_1x_2 + 0.375 \quad \ldots (3.47)$$

$$\frac{dx_2}{dt} = -0.025x_2 + 0.000013x_3 \quad \ldots (3.48)$$

$$\frac{dx_3}{dt} = -0.0926x_3 + 0.9722 \quad \ldots (3.49)$$

On integrating the above equations we get

$$x_1(t) - x_1(0) = 0.375Jdt - 16.667Jx_1(t) - 0.45Jx_2(t) - Jx_1(t)x_2(t) \ldots (3.50)$$

$$x_2(t) - x_2(0) = -0.025Jx_2(t) + 0.000013Jx_3(t) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.51)$$

$$x_3(t) - x_3(0) = -0.0926Jx_3(t) + 0.9722Jdt \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.52)$$

$$C_1{}^T\psi(t) - C_{10}{}^T\psi(0) = 0.375t - 16.667C_1{}^TP\psi(t) - 0.45C_2{}^TP\psi(t) - C_1{}^TPC_2{}^TP\psi(t)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.53)$$

$$C_2{}^T\psi(t) - C_{20}{}^T\psi(0) = -0.025C_2{}^TP\psi(t) + 0.000013C_3{}^TP\psi(t) \ldots\ldots\ldots (3.54)$$

$$C_3{}^T\psi(t) - C_{30}{}^T\psi(0) = -0.0926C_3{}^TP\psi(t) + 0.9722t \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.55)$$

Equations (3.53), (3.54) and (3.55) are further solved to obtain the values of C_1,$C_2$and C_3.Here J is the integration operator, C_1,$C_2$and $C_3$are the block pulse coefficients, $C_1(0)$, C_2 (0) and $C_3(0)$are the initial steady state values, P is the operational matrix and ψ represents block pulse function.

The Block-pulse function and Runge-Kutta (MATLAB solver) method codes are compiled which gives a set of output values of C_G, C_X and C_I which are plotted against the time intervals t=100, 500 and 1000 with matrix sizes 100, 500 and 1000.

Graphs

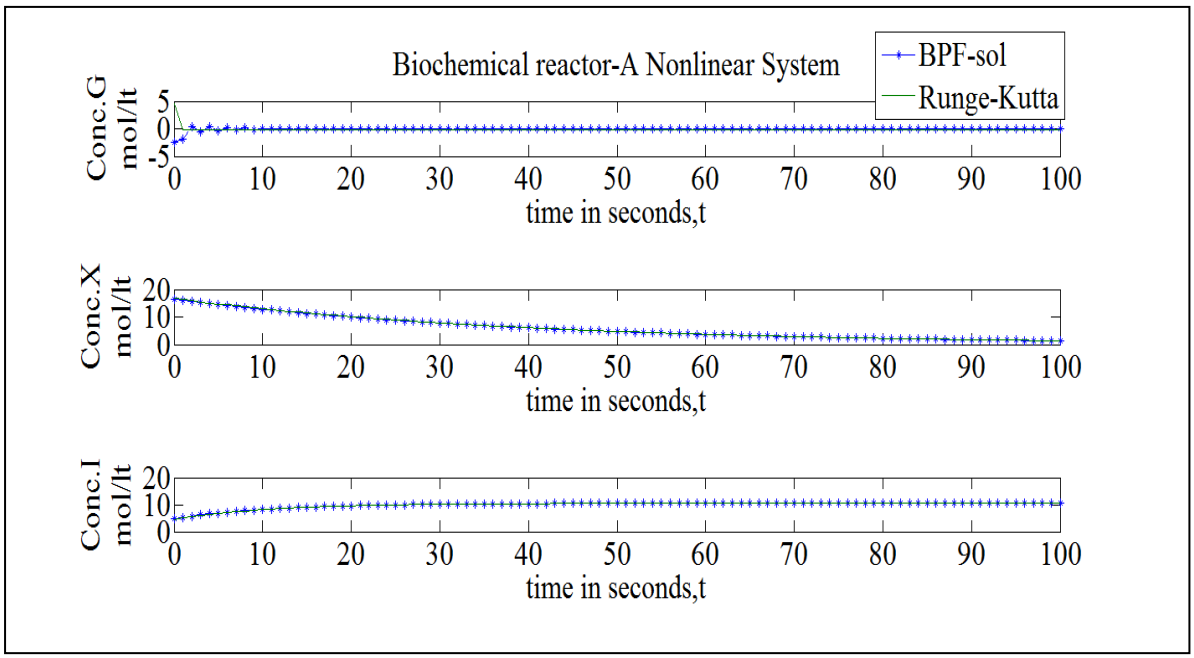

Figure-3.25: Concentration C_G, C_X and C_I vs. time and for n=100, T=100sec

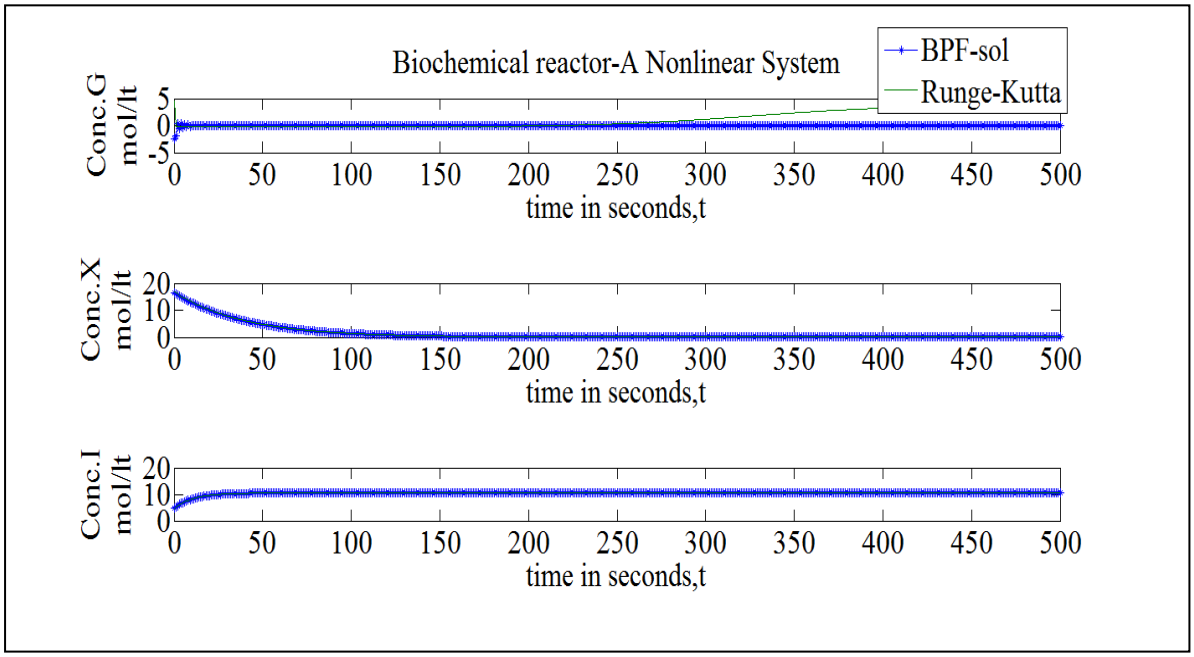

Figure-3.26: Concentration C_G, C_X and C_I vs. time and for n=500, T=500sec

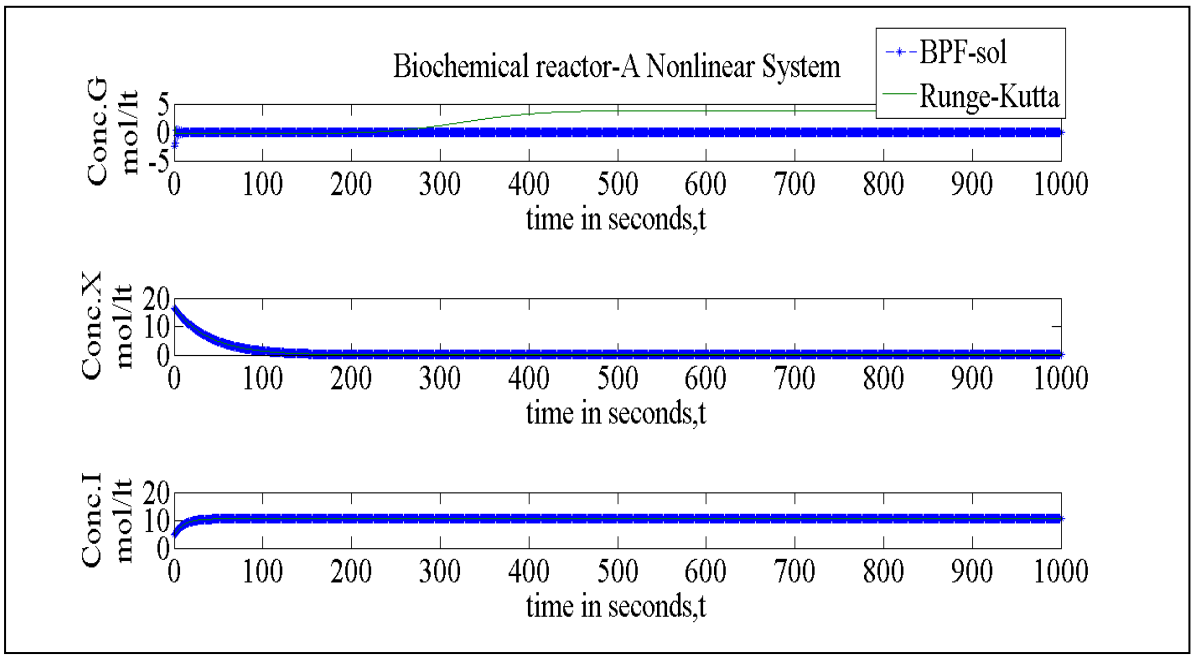

Figure-3.27: Concentration C_G, C_X and C_I vs. time and for n=1000, T=1000sec

3.4.1 Results and Discussions

- The given steady state values are 4.5, 16.667, 4.5 for G, X, I concentrations respectively and new steady state values established are 0.02, 1.356, 10.4984 respectively.
- For a matrix size of n=100, t=100 sec, a large aberration observed in the blood glucose concentration, G plotted using BPF data from that of the plots obtained by MATLAB solver ODE-45 data.
- Similarly, for the same matrix size but at different time intervals, deviation in insulin concentration (I), proportional concentration (X) are observed for plots obtained using BPF data from that of the plots obtained by MATLAB solver ODE-45 data.
- Deviations are observed to be minimizing with an increase in operational matrix sizes, such as that observed for n=500, t=500sec in plot-2, and there is even small deviation observed in concentrations of G, X, I for n=1000, t=1000sec.
- By increasing the number of subintervals within an interval the accuracy obviously increases because block pulse functions are binary in nature.
- Large time intervals will not increase the computation time to a great extent as the resulting operational matrix comprises of non-zero elements in the principal diagonal of the matrix, zeroes in the upper triangular matrix and ones in lower triangular matrix which makes computation simple even for large matrix sizes such as n=1000

CHAPTER-4

SIMULATION OF LINEAR AND NON-LINEAR SYSTEMS VIA ORTHOGONAL FUNCTIONS USING RECURRENCE RELATION

A Time- series is a sequence of data points measured typically at successive points in uniform intervals of time. Two methods are available for time-series analysis: frequency domain analysis, time domain analysis. Modern control theory essentially uses time domain approach while conventional control theory uses frequency domain analysis. The present work is premised on time domain approach in analysing the dynamic systems in chemical processes using two important orthogonal functions: block pulse functions, linear triangular functions via state space approach. By using vector-matrix notation, nth order differential equation may be expressed as a first order vector-matrix differential equation. The state space approach uses this vector-matrix notation to relate the input –output relationship. The knowledge of the state variables of a dynamic system at initial value $t=t_0$ combined with input at $t \geq t_0$ completely determines the behaviour of the system for any time $t \geq t_0$.

4.1 RECURRENCE RELATION USING TRIANGULAR FUNCTIONS FOR LINEAR SYSTEMS

Consider the time –invariant linear SISO dynamic system modelled by

$$\dot{x} = Ax + Bu \text{ and } x(0) = x_0 \quad \ldots \ldots (4.1)$$

$$\int_0^t \dot{x}(\tau)\, d\tau = x(t) - x(0) \quad \ldots \ldots (4.2)$$

Where, x is a state vector, u is an input vector, and A and B are matrices of appropriate dimensions. The rate vector $\frac{dx}{dt}$, the state vector x and overall control vector Bu are expanded into triangular function domain as given below:

$$\dot{x}(t) = \sum_{i=0}^{m-1} C_{1,i+1}^T T_{1,i+1}(t) + \sum_{i=0}^{m-1} C_{2,i+1}^T T_{2,i+1}(t) \triangleq C_1 T_1(t) + C_2 T_2(t) \quad \ldots \ldots (4.3)$$

$$x(t) = \sum_{i=0}^{m-1} D_{1,i+1}^T T_{1,i+1}(t) + \sum_{i=0}^{m-1} D_{2,i+1}^T T_{2,i+1}(t) \triangleq D_1 T_1(t) + D_2 T_2(t) \quad \ldots \ldots (4.4)$$

$$Bu(t) = \sum_{i=0}^{m-1} E_{1,i+1}^T T_{1,i+1}(t) + \sum_{i=0}^{m-1} E_{2,i+1}^T T_{2,i+1}(t) \triangleq E_1 T_1(t) + E_2 T_2(t) \quad \ldots \ldots (4.5)$$

Here $C_{1,i+1}, C_{2,i+1}, D_{1,i+1,i+1}, D_{2,i+1}, E_{1,i+1}$ and $E_{2,i+1}$ are n-vectors and form the (i+1)th column of the n-by-m matrices C_1, C_2, D_1, D_2, E_1 and E_2 respectively.For a given input u(t), E_1 and E_2 are known using the function approximation. To solve the state vector x in the TF domain equation 2 is used.

Using equation 4.3 in equation 4.22 and the TF property of integration, we get

$$[\boldsymbol{C_1} + \boldsymbol{C_2}] \int \boldsymbol{T1}\,(t) = \boldsymbol{D_1 T_1}(\boldsymbol{t}) + \boldsymbol{D_2 T_2}(\boldsymbol{t}) - \boldsymbol{2\widetilde{x_0}} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.6})$$

Using the TF property of integration and equation 1 and dropping the argument (t), we can write

$$[\boldsymbol{C_1} + \boldsymbol{C_2}][\boldsymbol{P_1 T_1} + \boldsymbol{P_2 T_2}] = \boldsymbol{D_1 T_1} + \boldsymbol{D_2 T_1} - \widetilde{\boldsymbol{x_0}}[\boldsymbol{T_1} + \boldsymbol{T_1}] \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.7})$$

Equating like coefficients of the basis functions T1 and T2, we obtain

$$[\boldsymbol{C_1} + \boldsymbol{C_2}]\boldsymbol{P_1} = \boldsymbol{D_1} - \widetilde{\boldsymbol{x_0}} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.8})$$

$$[\boldsymbol{C_1} + \boldsymbol{C_2}]\boldsymbol{P_2} = \boldsymbol{D_2} - \widetilde{\boldsymbol{x_0}} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.9})$$

Adding equations (4.8) and (4.9), we get

$$[\boldsymbol{C_1} + \boldsymbol{C_2}][\boldsymbol{P_1} + \boldsymbol{P_2}] = [\boldsymbol{D_1} + \boldsymbol{D_2}] - \boldsymbol{2\widetilde{x_0}} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.10})$$

Putting $C1 + C2 \triangleq C, D1 + D2 \triangleq D$ and using the relation $P1 + P2 \triangleq P$, we have

$$\boldsymbol{CP} = \boldsymbol{D} - \boldsymbol{2\widetilde{x_0}} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.11})$$

$$(\boldsymbol{CP})_{\boldsymbol{i+1}} = \boldsymbol{D_{i+1}} - \boldsymbol{2\widetilde{x_0}} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.12})$$

To proceed with the TF domain solution, we use the property of the matrix P and can write

$$(\boldsymbol{CP})_{\boldsymbol{i+1}} = \frac{\boldsymbol{T}}{\boldsymbol{m}} \sum_{\boldsymbol{j=1}}^{\boldsymbol{i}} \boldsymbol{C_j} + \frac{\boldsymbol{T}}{\boldsymbol{2m}} \boldsymbol{C_{i+1}}\,, \boldsymbol{i} \leq \boldsymbol{j} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.13})$$

Writing in a recurrence relation form

$$(\boldsymbol{CP})_{\boldsymbol{1}} = \boldsymbol{2}\frac{\boldsymbol{T}}{\boldsymbol{m}} \boldsymbol{C_1} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.14})$$

$$(\boldsymbol{CP})_{\boldsymbol{i+2}} = (\boldsymbol{CP})_{\boldsymbol{i+1}} + \frac{\boldsymbol{T}}{\boldsymbol{m}}(\boldsymbol{C_{i+1}} + \boldsymbol{C_{1i+2}}) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\boldsymbol{4.15})$$

for all $i \geq 0$

Combining equations (4.12), (4.14) and (4.15), we get solutions for state vectors as

$$\boldsymbol{D_1} = \frac{\boldsymbol{T}}{\boldsymbol{2m}}\boldsymbol{C_1} + \boldsymbol{2x_0} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (\mathbf{4.16})$$

$$\boldsymbol{D_{i+2}} = \boldsymbol{D_{i+1}} + \frac{\boldsymbol{T}}{\boldsymbol{2m}}(\boldsymbol{C_{i+1}} + \boldsymbol{C_{1i+2}}) \ldots \ldots \ldots \ldots (\mathbf{4.17})$$

for all $i \geq 0$

$$\boldsymbol{D_1} = \boldsymbol{2}\left[\boldsymbol{I} - \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]^{-1}\boldsymbol{x_0} + \frac{\boldsymbol{T}}{\boldsymbol{2m}}\left[\boldsymbol{I} - \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]^{-1}\boldsymbol{E_1} \ldots \ldots \ldots \ldots (\mathbf{4.18})$$

$$\boldsymbol{D_{i+2}} = \left[\boldsymbol{I} + \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]\left[\boldsymbol{I} - \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]^{-1}\boldsymbol{D_{i+1}} + \frac{\boldsymbol{T}}{\boldsymbol{2m}}\left[\boldsymbol{I} - \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]^{-1}(\boldsymbol{E_{i+1}} + \cdots \boldsymbol{E_{i+2}}) \ldots \ldots (\mathbf{4.19})$$

$$\boldsymbol{D_1} = \boldsymbol{K}\left[\boldsymbol{I} - \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]^{-1}\boldsymbol{x_0} + \frac{\boldsymbol{T}}{\boldsymbol{2m}}\left[\boldsymbol{I} - \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]^{-1}\boldsymbol{E_1} \ldots \ldots \ldots \ldots (\mathbf{4.20})$$

$$\boldsymbol{D_{i+2}} = \left[\boldsymbol{I} + \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]\left[\boldsymbol{I} - \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]^{-1}\boldsymbol{D_{i+1}} + \frac{\boldsymbol{T}}{\boldsymbol{2m}}\left[\boldsymbol{I} - \frac{\boldsymbol{AT}}{\boldsymbol{2m}}\right]^{-1}(\boldsymbol{E_{i+1}} + \cdots \boldsymbol{E_{i+2}}) \ldots \ldots (\mathbf{4.21})$$

The equations (4.20) and (4.21) represent recurrence relation in TF domain when K=2 and in BPF domain when K=1.

Now these equations are applied to a set of differential equations which are the model equations of reactions taking place in a system. Here in our present work two reacting systems are considered. The dynamics of these systems is analysed using recurrence relation that was developed previously.

4.2 SIMULATION OF LINEAR SYSTEMS VIA RECURRENCE RELATION

4.1) Simulation of an Irreversible Reaction in a CSTR

Problem Statement: - Consider a set of differential equations for a first order irreversible reaction in an isothermal CSTR. Ethylene oxide (A) is reacted with water (B) to produce ethylene glycol(R). Water is in large excess. A CSTR is used at a constant temperature. The block-pulse functions are used in solving the two differential model equations in an isothermal CSTR. The states are the concentration of A and R in the reactor. C_A is the concentration of the reactant A and C_R is the concentration of R. The parameter values are k_1=0.311 min^{-1}. The input values used in the following simulation are F/V=0.0777min^{-1}, C_{Af} =0.5mol/lt.

Model Equations:-

$$\frac{dC_A}{dt} = \frac{F(C_{Af} - C_A)}{V} - k_1 C_A \quad \ldots\ldots(4.22)$$

$$\frac{dC_R}{dt} = -\frac{FC_R}{V} + k_1 CA \quad \ldots\ldots(4.23)$$

Now expressing the two differential equations in terms of block-pulse functions as follows:

Let $C_A = x_1, C_R = x_2$

After substituting the given parameters in the above differential equations 42, 43, we get the following equations,

$$\frac{dx_1}{dt} = 0.0389 - 0.3887x_1 \quad \ldots\ldots(4.24)$$

$$\frac{dx_2}{dt} = 0.311x_1 + 0.0777x_2 \quad \ldots\ldots(4.25)$$

Expressing the above model equations in terms of state space matrices, we get

$$A = \begin{bmatrix} -0.0389 & 0 \\ 0.311 & -0.0777 \end{bmatrix}; x_0 = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}; E = \begin{bmatrix} 0.0389 \\ 0 \end{bmatrix}$$

For a given matrix size m and for a given time interval t the matrices A, x_0 and E are substituted in the recurrence relation to obtain the state vector coefficient in TF domain and non-optimal BPF domain. Now the obtained output variables are compared over the variables obtained using actual solution, function approximation using BPF and TF domains.

The actual solution of the system is obtained as:

$$x_1(t) = -0.0965e^{-0.077t} + 0.8482e^{-0.389t} + 0.1006 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.26)$$

$$x_2(t) = 1.253e^{-0.077t} - 2.1805e^{-0.389t} + 0.4033 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.27)$$

The function approximation using BPF and TF domain is given as following respectively:

$$c_{i,BPF} = \frac{1}{h}\int_0^1 x_1(t)\Phi_i(t)dt \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.28)$$

$$c_{i,TF} = x_1(ih); d_i = x_1\big((i+1)h\big) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.29)$$

The results are plotted using MATLAB against various time intervals, t=10,50 and 100 sec for various matrix sizes, m=10,50 and 100 as shown below:

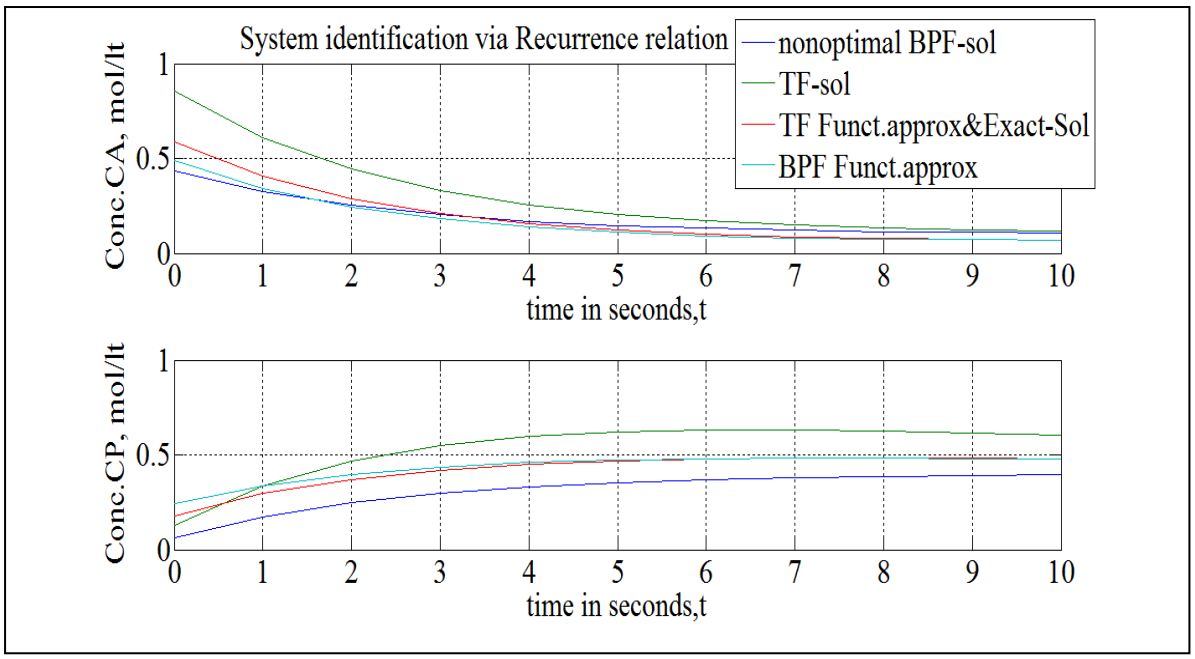

Figure-4.1: Concentration C_A, C_P vs. time and for n=10, T=10sec

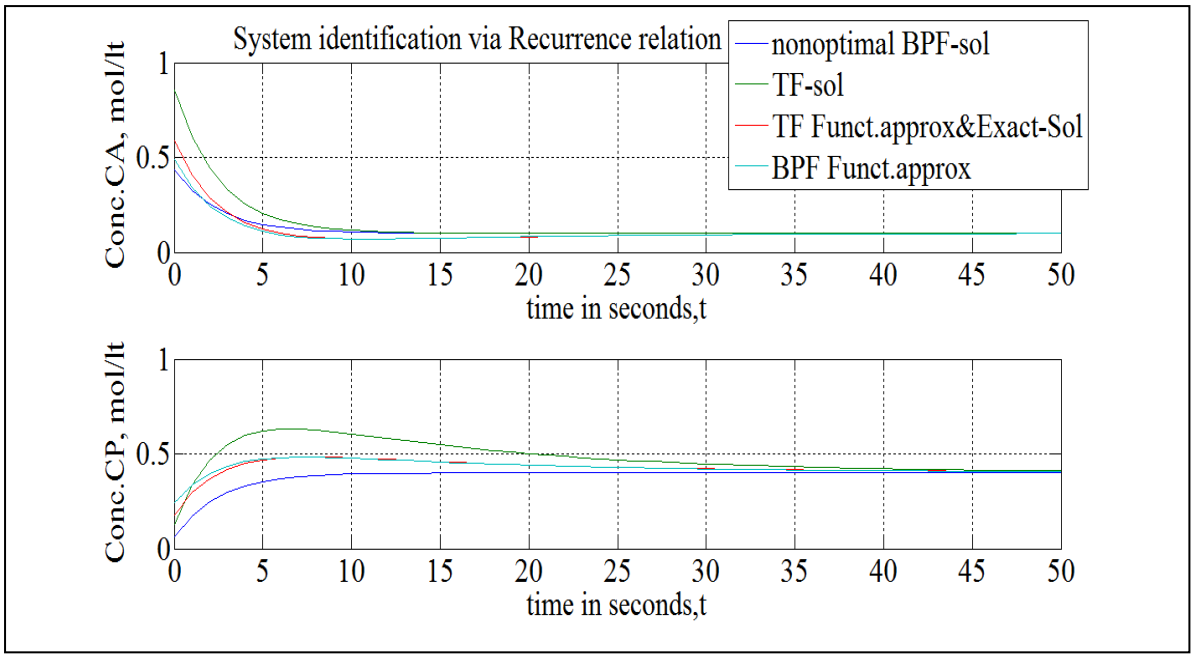

Figure-4.2: Concentration C_A, C_P vs. time and for n=50, T=50sec

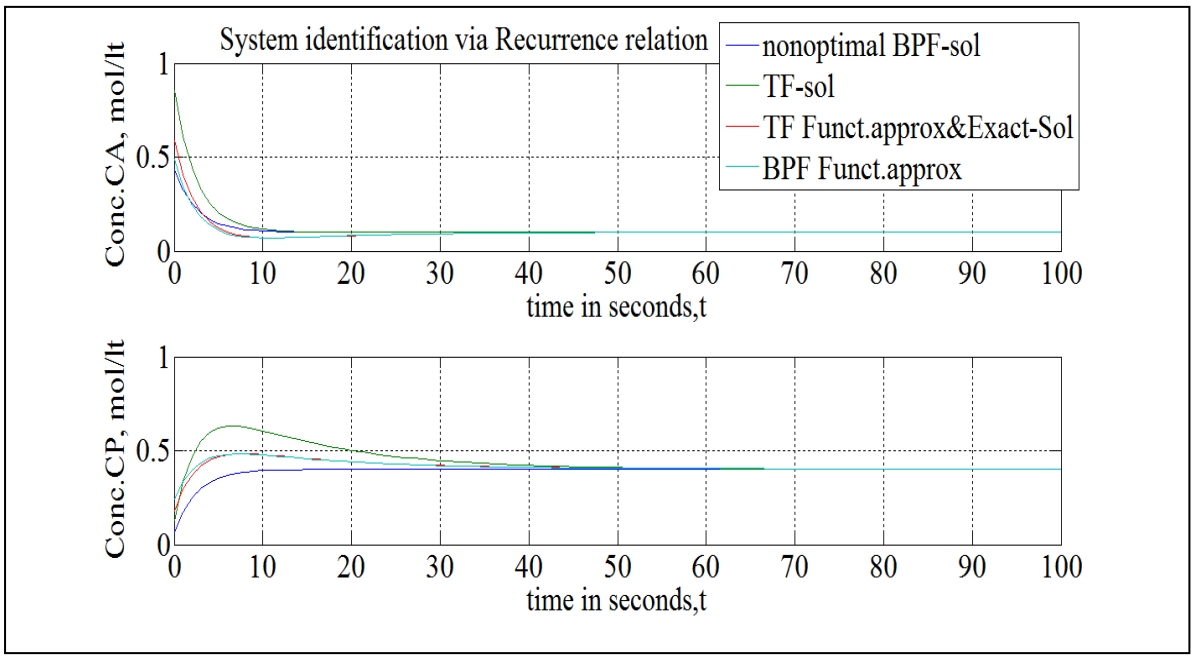

Figure-4.3: Concentration C_A, C_P vs. time and for n=100, T=100sec

Table 4.1:- Results Comparison Table

T(sec)	m(size)	x_{1BPF}	x_{2BPF}	x_{1TF}	x_{2TF}	y_{1BPF}	y_{2BPF}	y_{1TF}	y_{2TF}
10	10	0.1066	0.3936	0.1148	0.6065	0.0705	0.4676	0.071	0.4784
50	50	0.1001	0.4003	0.1001	0.4101	0.0988	0.407	0.098	0.4071
100	100	0.1001	0.4003	0.1001	0.4005	0.1006	0.4034	0.1006	0.4034

4.2) Simulation of a SISO system

Consider a linear time invariant SISO system whose model equations are given below

Model Equations:-

$$\frac{dx_1}{dt} = x_2 \quad (4.30)$$

$$\frac{dx_2}{dt} = -2x_1 - 3x_2 + 1 \quad (4.31)$$

Expressing the above model equations in terms of state space matrices, we get

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}; x_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; E = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

For a given matrix size m and for a given time interval t the matrices A, x_0 and E are substituted in the recurrence relation to obtain the state vector coefficient in TF domain and non-optimal BPF domain: Now the obtained output variables are compared over the variables obtained using actual solution, function approximation using BPF and TF domains.

The actual solution of the system is obtained as:

$$x_1(t) = -e^{-t} + 0.5e^{-2t} + 0.5 \quad (4.32)$$

$$x_2(t) = e^{-t} - e^{-2t} \quad (4.33)$$

The function approximations using BPF and TF domain are calculated using equations (4.28) and (4.29).The results are plotted using MATLAB against various time intervals, t=10, 20 and 30sec for various matrix sizes, m= 100 as shown below:

Graphs

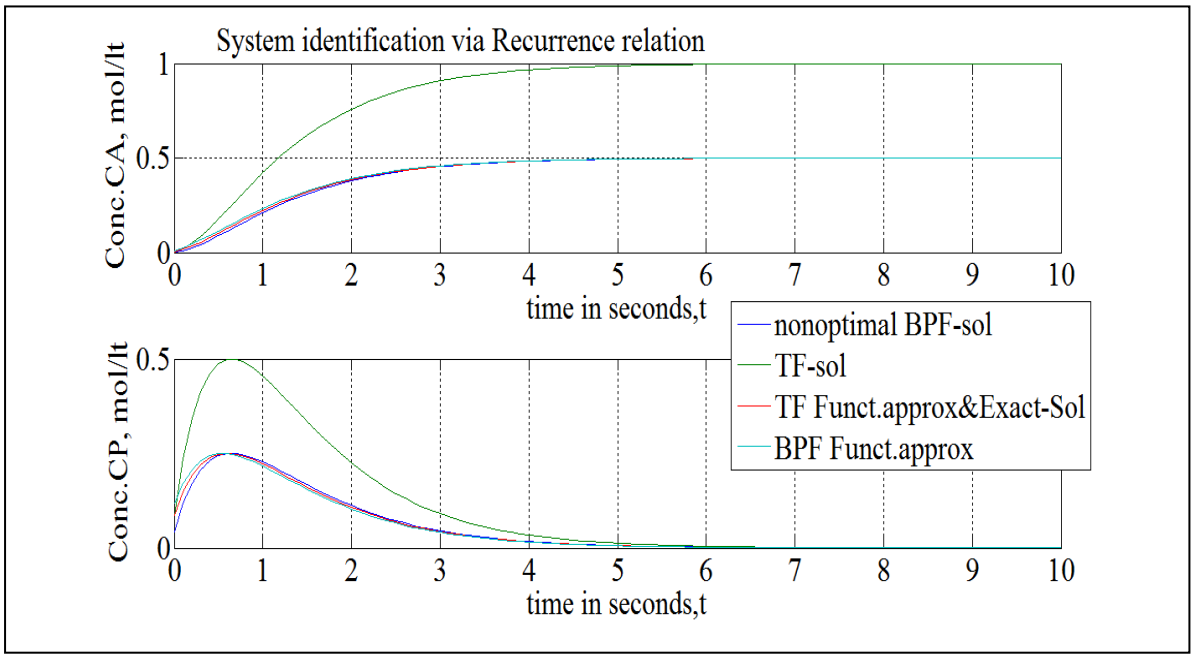

Figure-4.4: Concentration C_A, C_P vs. time and for n=100, T=10sec

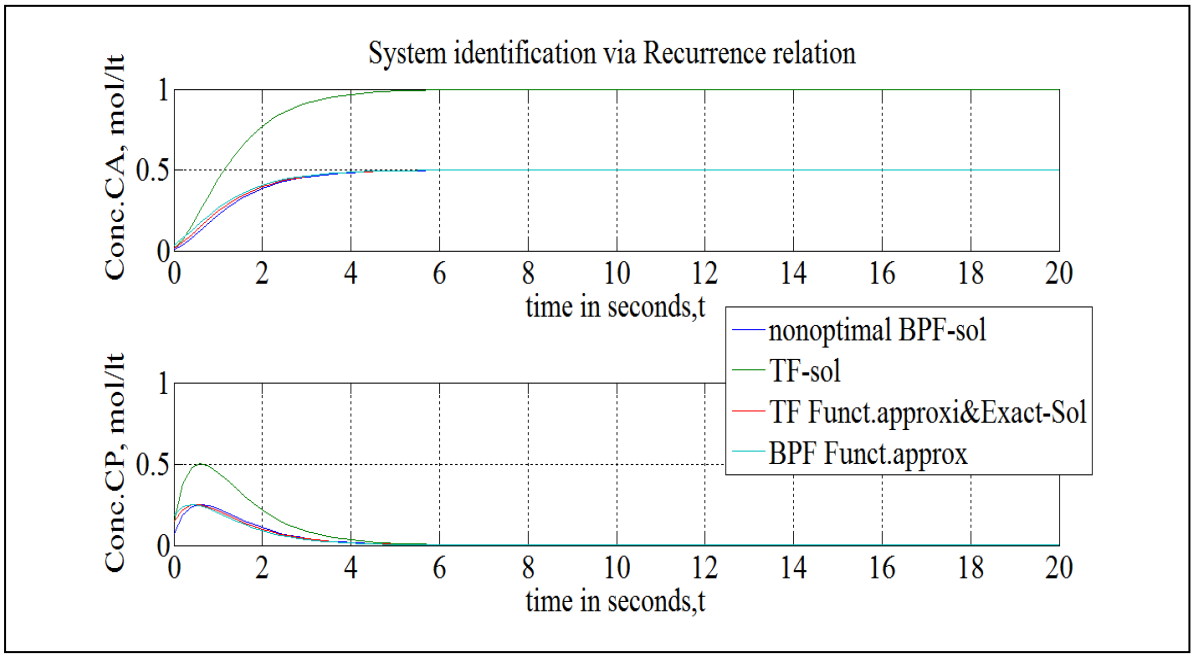

Figure-4.5: Concentration C_A, C_P vs. time and for n=100, T=20sec

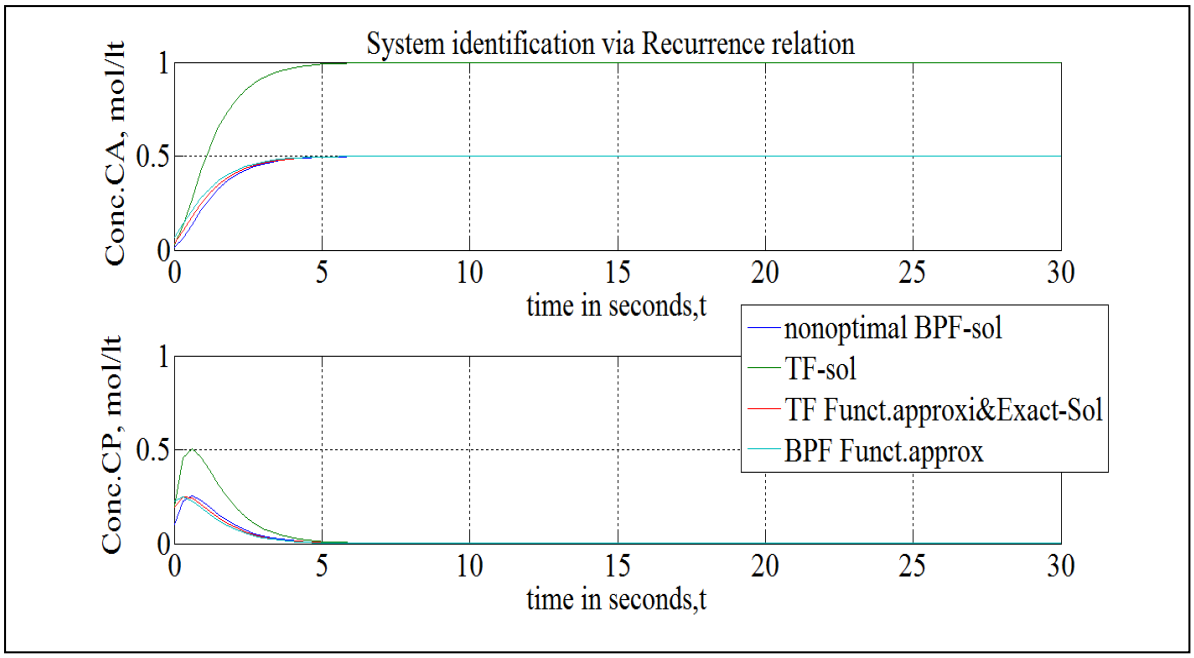

Figure-4.6: Concentration C_A, C_P vs. time and for n=100, T=30sec

Table 4.2:- Results Comparison Table

T(sec)	m(size)	x_{1BPF}	x_{2BPF}	x_{1TF}	x_{2TF}	y_{1BPF}	y_{2BPF}	y_{1TF}	y_{2TF}
10	100	0.5	0	0.9999	0.0001	0.5	0	0.5	0
20	100	0.5	05	1	0	0.5	0	0.5	0
30	100	0.5	0.5	1	0	0.5	0	0.5	0

4.2.1 Results and Discussions

- From the two reacting systems, it is clearly seen that TF has greater accuracy than BPF and hence TF domain solution is more efficient. The TF function approximation is same as the true solution.
- The non-optimal BPF solution can approach to TF with large matrix dimension size, m. Hence for large m, results obtained in TF and BPF domain are almost similar. Non-optimal BPF solution is the average of TF solution.
- From the two result tables 4.1 and 4.2, it is observed that the new steady state values have been reached by the systems within 30seconds, i.e. response time is very small. For the second system, even with large m, say 1000, the x and y values are same in recurrence relation (TF &BPF) and function approximation (TF & BPF) as of with m=100.Thus, it can be stated that new convergence of plots for second system can be achieved even with small matrix size.
- For the first system, for t=10sec and m=10, small deviations are observed in the plots however for large time intervals and matrix sizes, say t=50 and 100seconds and m=50 and 100 almost zero deviations are observed in all the plots.

4.3 SIMULATION OF NON-LINEAR SYSTEMS VIA RECURRENCE RELATION

Non-linearity is always a complex issue to deal with, hence any new methodology solving non-linear problems involves lot of assumptions and deviations from original solutions. Solving non-linear problems analytically becomes tough and almost impossible; as a result no method can rightly help in obtaining the exact solution to non-linear differential equations. As a result traditional numerical methods like Runge-Kutta been taken as a reference to compare the results of simulations produced by recurrence relation using BPF.

The recurrence relation developed for linear systems in BPF and TF domain cannot be used for non-linear systems. For a non-linear system, the functions under consideration should also be recurred followed by the recurrence of output variable. The recurrence relation for non-linear systems is thus an extension to that of the relation developed for linear systems. The linear results are used as initial values for the recurrence of non-linear systems. From equations (4.20) and (4.21), the linear output values can be determined which are further used in calculating the output variable of corresponding non-linear system.

Let us consider a general non-linear system of the form

$$\dot{X} = f(x, u, t) \dots \dots (4.34)$$

$$D_1 = \frac{1}{2m} f(D_{linear}, u_i, t_i) + X_0 \dots \dots (4.35)$$

$$D_{i+1} = \frac{1}{2m} f(D_{linear+1}, u_{i+1}, t_{i+1} + D_{linear}, u_i, t_i) + X_0 \dots \dots (4.36)$$

Where X_0 represents the initial values vector of the system, D_{linear} is the output vector calculated from linear recurrence relation. Now let us apply these equations to the two reacting systems: Non-isothermal CSTR and a Biochemical reactor.

4.3) Simulation of Irreversible reaction in Non-Isothermal CSTR using Recurrence relation:-

Problem Statement: - Consider a continuous stirred tank reactor for non-isothermal conditions. The reaction $A \rightarrow B$ is exothermic and the heat generated in the reactor is removed via cooling system. The effluent temperature is different from the inlet temperature due to heat generation by exothermic reaction. Assuming constant density, the microscopic total mass balance, mass component and energy balance equations are considered. The dependence of the rate constant on the temperature; $k = k_0 e^{\frac{-E_a}{RT}}$ shall be taken into account. The non-isothermal CSTR is modelled but three ordinary differential equations (ODE's).

Model equations:-

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Af} - C_A) - k_0 e^{\frac{-E_a}{RT}} C_A \dots \dots (4.37)$$

$$\frac{dT}{dt} = \frac{F}{V}(T_f - T) + \frac{(-\Delta H)}{\rho C_p} k_0 e^{\frac{-E_a}{RT}} C_A - \frac{UA}{\rho C_p}(T - T_j) \dots \dots (4.38)$$

$$\frac{dT_j}{dt} = \frac{F_{jf}}{V}(T_{jf} - T_j) + \frac{UA}{\rho_j V_j C_{pj}}(T - T_j) \dots \dots (4.39)$$

The parameters are substituted in the given differential equations and these differential equations are expressed in terms of state-space model.

$$A = \begin{bmatrix} -0.079907 & -0.00013674 & 0 \\ 29.229 & 0.0456 & 0.146 \\ 0 & 0.04782 & -0.0589 \end{bmatrix}; x_0 = \begin{bmatrix} 0.132 \\ 520.67 \\ 460.67 \end{bmatrix}; E = \begin{bmatrix} 0 \\ 0 \\ -0.0326 \end{bmatrix}$$

For a given matrix size m and for a given time interval t the matrices A, x_0 and E are substituted in the recurrence relation equations (4.34),(4.35) and (4.36) to obtain the state vector coefficient in BPF domain. Now the obtained output variables are compared over the outputs obtained using Runge-Kutta method.

The results are plotted using MATLAB against various time intervals, t=50, 100and 500 sec for various matrix sizes, m=1, 2 and5 for 30 iterations as shown below:

Graphs

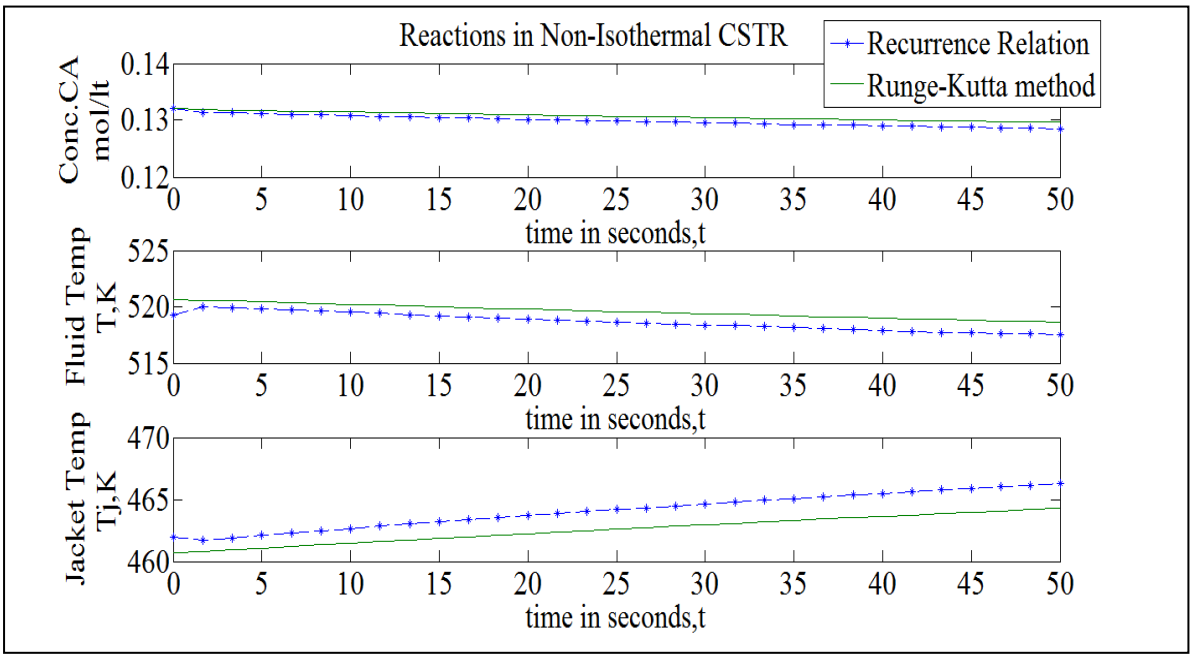

Figure-4.7: Concentration C_A, T and T_j vs. time and for m=2, T=50sec

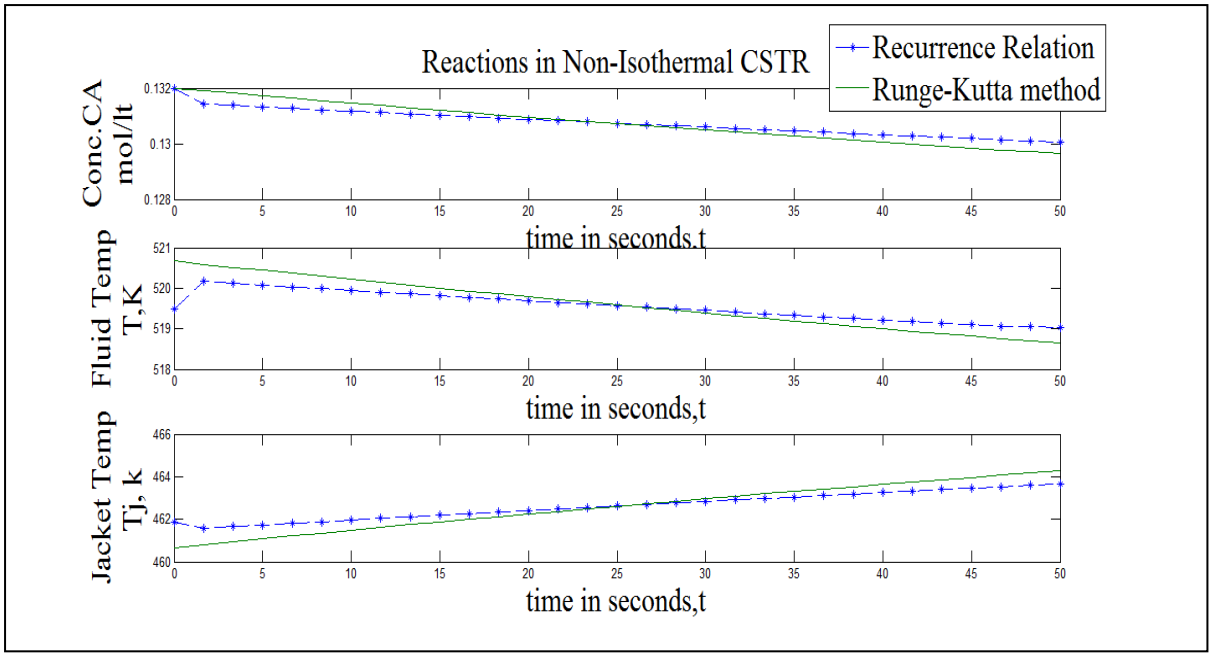

Figure-4.8: Concentration C_A, T and T_j vs. time and for m=5, T=50sec

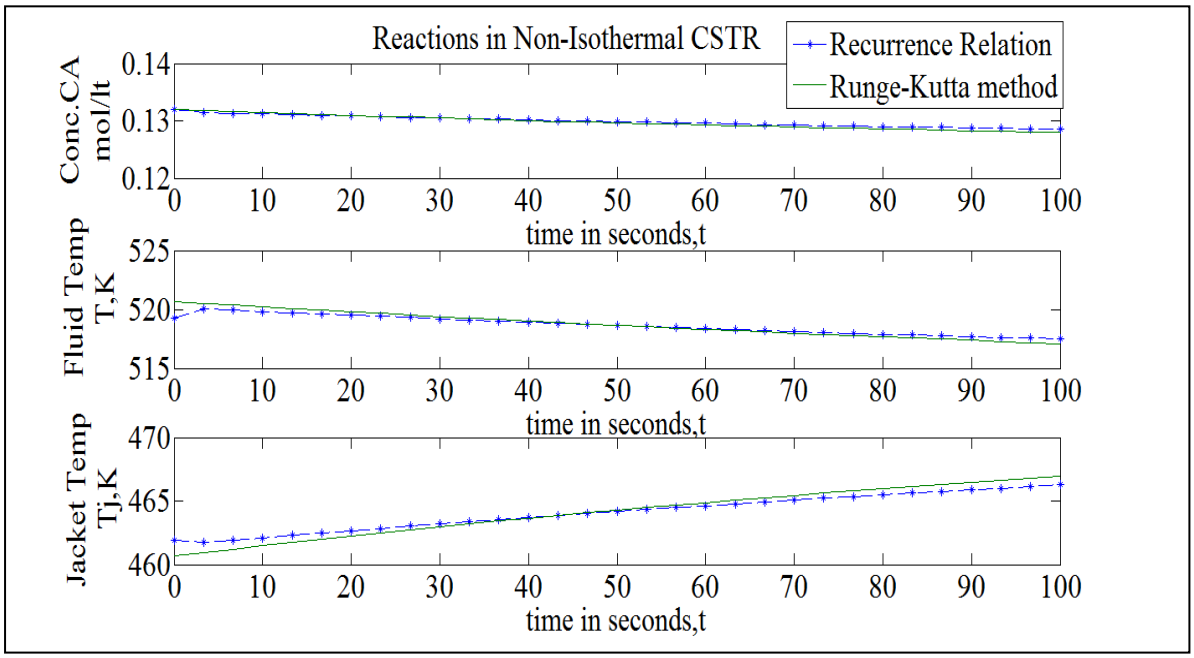

Figure-4.9: Concentration C_A, T and T_j vs. time and for m=2, T=100sec

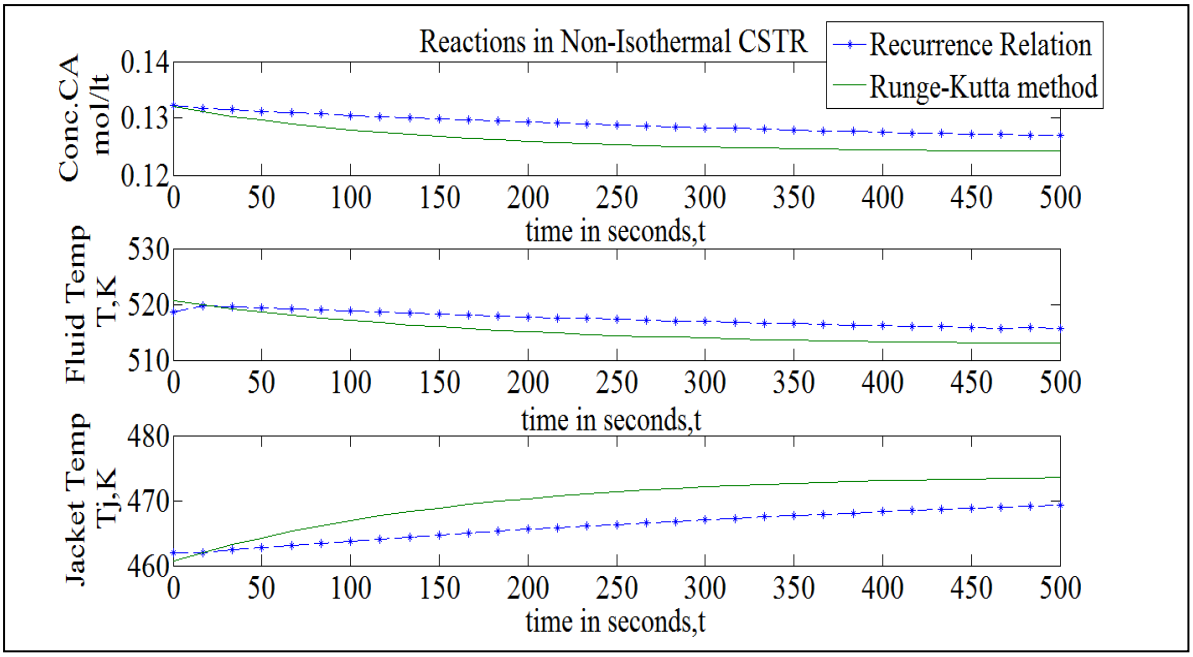

Figure-4.10: Concentration C_A, T and T_j vs. time and for m=1, T=500sec

Another non-linear system which is simulated using the recurrence form is the biochemical reactor in which bold glucose concentration is determined for the corresponding insulin injected with the body. The system described previously in problem (3.5) is considered. Those differential equations are multiplied by 0.01.The parameters are substituted in the differential equations of problem (3.5) and these differential equations are expressed in terms of state-space model.

$$A = 0.01 * \begin{bmatrix} -16.667 & -4.5 & 0 \\ 29.229 & -0.025 & 0.000013 \\ 0 & 0 & -0.0926 \end{bmatrix}; x_0 = \begin{bmatrix} 4.5 \\ 16.667 \\ 4.5 \end{bmatrix}; E = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{12} \end{bmatrix}$$

For a given matrix size m and for a given time interval t the matrices A, x_0 and E are substituted in the recurrence relation equations (4.34), (4.35) and (4.36) to obtain the state vector coefficient in BPF domain. Now the obtained output variables are compared over the outputs obtained using Runge-Kutta method.

The results are plotted using MATLAB against various time intervals, t=10, 20and 30 sec for various matrix sizes, m=2 for 30 iterations as shown below:

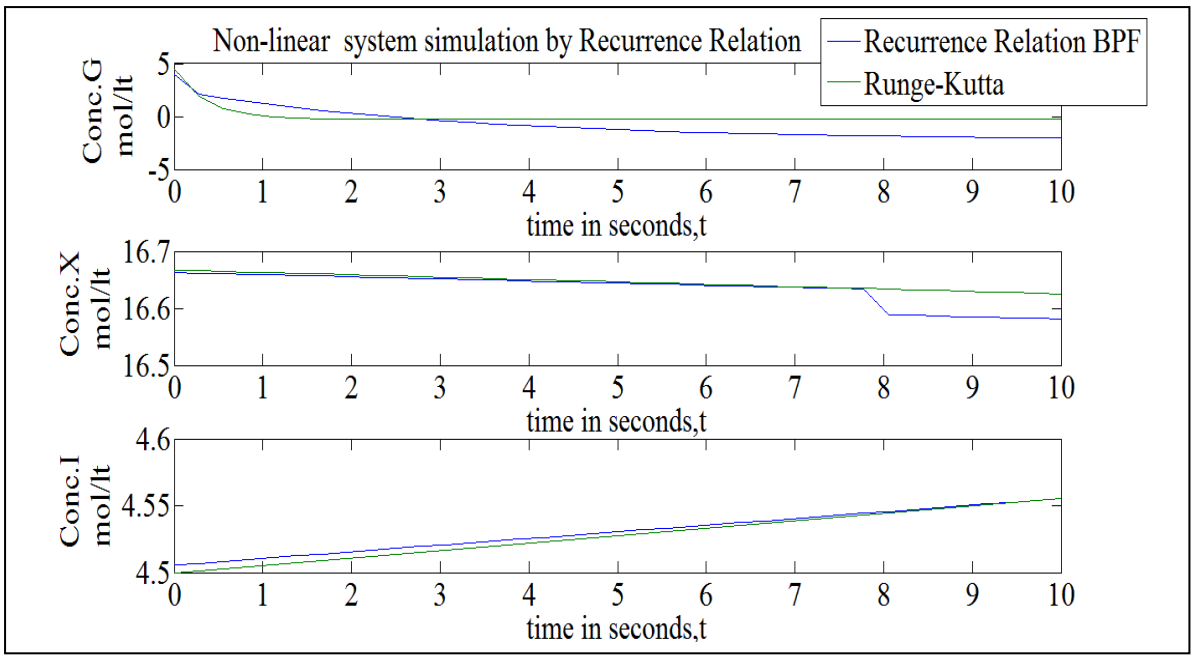

Figure-4.11: Concentration C_G, C_X and C_I vs. time and for m=2, T=10sec

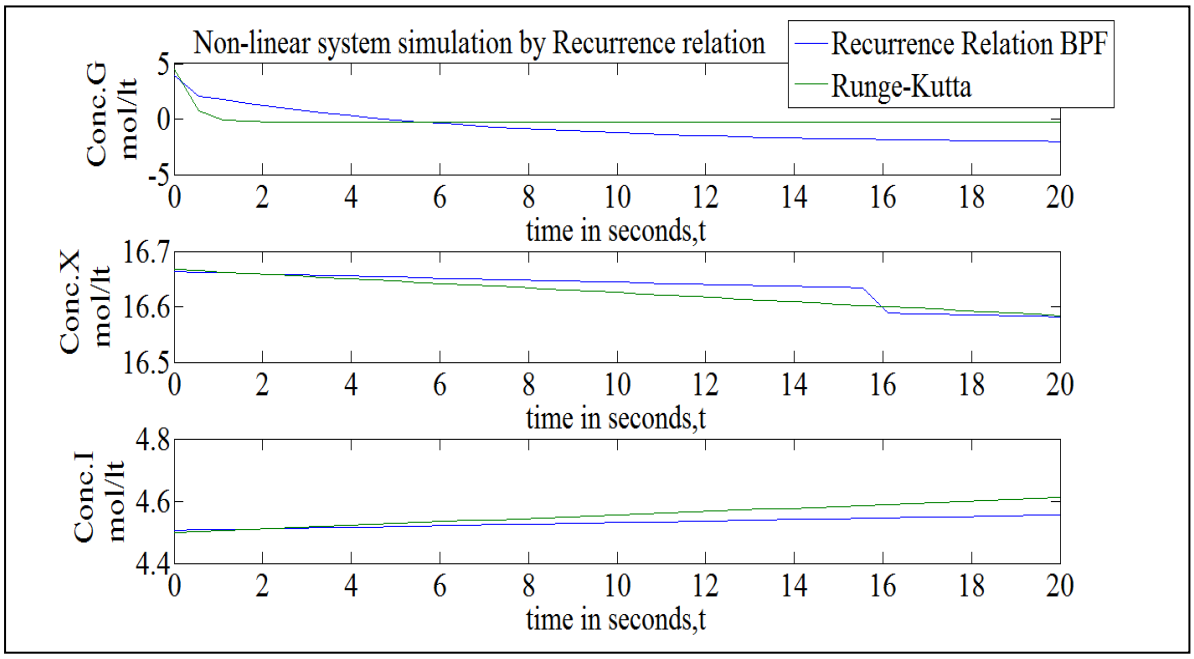

Figure-4.12: Concentration C_G, C_X and C_I vs. time and for m=2, T=20sec

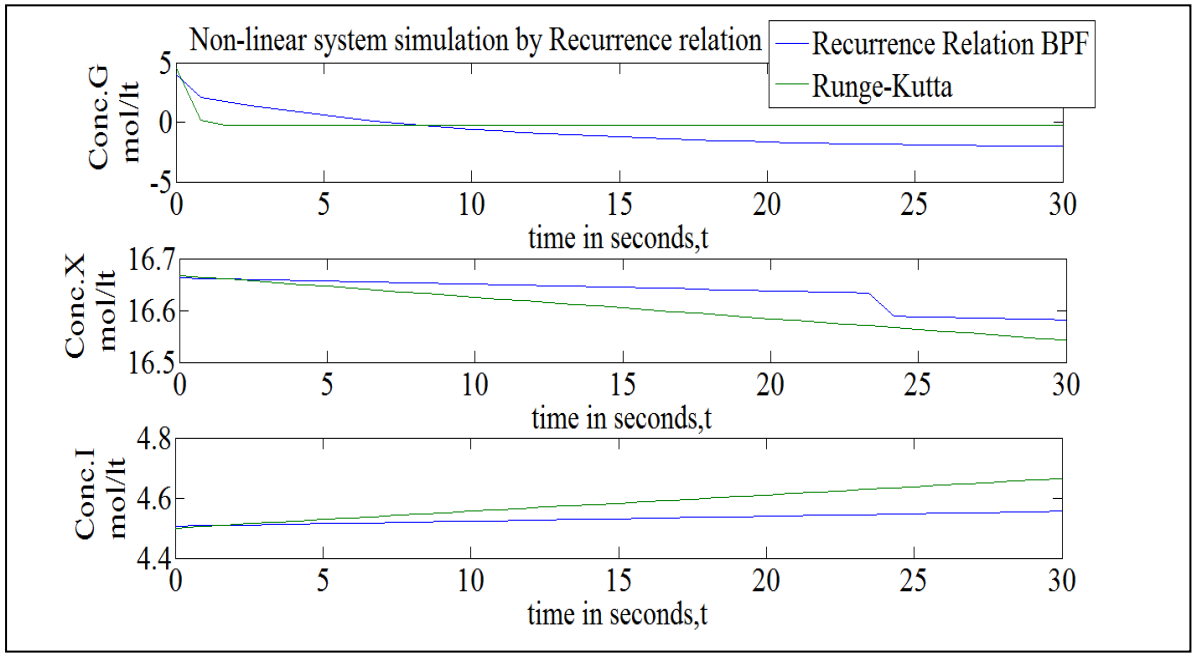

Figure-4.13: Concentration C_G, C_X and C_I vs. time and for m=2, T=30sec

4.3.1 Results and Discussions

- The two non-linear reacting systems are simulated using recurrence relation and the results are compared with those results obtained from Runge-Kutta method. The plots clearly show that our proposed method follows the trend of the Runge-Kutta method in establishing the dynamics of the two systems.
- Unlike the previous method of BPF with operational matrix which requires matrix size of 100 and 500 to converge, recurrence method requires very small matrix dimension, m=2 even for large time intervals of t=500 seconds. Deviations are not completely eliminated as observed from the plots however the average deviation in both the systems is about 2% and in CSTR the average error percentage is less than 0.5%.
- These results are produced with less sub iterations say about 30, thus to minimize the error or to eliminate it completely sub iterations can be compared when the systems are to be simulated for large time intervals say more than 1000 seconds for non-isothermal system and 100 seconds for biochemical reactor.

CHAPTER-5

OPTIMAL CONTROL OF CSTR'S USING ORTHOGONAL FUNCTIONS

5.1 OPTIMAL CONTROL OF SYSTEMS

The basis for conventional control theory is the input-output relationship or transfer function. On a contrary, the modern control theory is based on vector-matrix differential equation that simplifies the mathematical representation of equations to a large extent. Even though the number of states, inputs and outputs get increased, the complexity of the system does not get increased. There had been several numerical techniques adopted earlier to obtain the states and outputs by solving the matrices in state-space representation. Some of those methods are recursive while some are non-recursive. Now, the block pulse functions are employed to determine the optimality criterion and states of the system. The recursive computation derived from block pulse functions is used in this regard.

Pole Placement method can determine the states only when all states are measurable, i.e. when the system is completely controllable. In practise, not all states are completely measurable; hence states estimation is not possible by pole placement method. By incorporating observers within the system all states can be estimated.

Consider a linear time invariant completely observable and controllable system described by

$$\dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{A}\boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{B}\boldsymbol{u}(\boldsymbol{t}) \quad \ldots\ldots (\mathbf{5.1})$$

$$\boldsymbol{y}(\boldsymbol{t}) = \boldsymbol{C}\boldsymbol{x}(\boldsymbol{t}) \quad \ldots\ldots (\mathbf{5.2})$$

Where u (t), x (t) and y (t) are plant input, state and output vectors respectively and A, B, C are n×n, n×r and p×n real constant matrices respectively. Assume that the rank of C is p.

An observer described by

$$\dot{\boldsymbol{z}}(\boldsymbol{t}) = \boldsymbol{F}\boldsymbol{z}(\boldsymbol{t}) + \boldsymbol{G}\boldsymbol{u}(\boldsymbol{t}) + \boldsymbol{H}\boldsymbol{y}(\boldsymbol{t}) \quad \ldots\ldots (\mathbf{5.3})$$

$$\hat{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{L_1}\boldsymbol{y}(\boldsymbol{t}) + \boldsymbol{L_2}\boldsymbol{z}(\boldsymbol{t}) \quad \ldots\ldots (\mathbf{5.4})$$

Can provide the estimate $\hat{x}(t)$ for the stare x (t), where F, G, H, L_1, L_2 are real constant q×q, q×r, q×p, n×p and n×q matrices respectively and q=n-p when the following conditions are satisfied.

$$\boldsymbol{G} = \boldsymbol{\mathcal{T}}\boldsymbol{B} \quad \ldots\ldots (\mathbf{5.5})$$

Where $\mathcal{T}$ the q×n matrix which is the solution of the matrix equation

$$\boldsymbol{\mathcal{T}A - F\mathcal{T} = HC} \quad \boldsymbol{(5.6)}$$

$$\boldsymbol{z(t) = \mathcal{T}x(t) + e(t)} \quad \boldsymbol{(5.7)}$$

$$\boldsymbol{\dot{e}(t) = Fe(t)} \quad \boldsymbol{(5.8)}$$

$$\boldsymbol{L_1C + L_2\mathcal{T} = I_n} \quad \boldsymbol{(5.9)}$$

When an observer is incorporated to generate an estimate $\hat{x}(t)$ of the plant state vector, we need to choose the matrix K in the feedback law.

$$\boldsymbol{u(t) = -K\hat{x}(t)} \quad \boldsymbol{(5.10)}$$

So that cost function

$$J = \frac{1}{2}\int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \quad (5.11)$$

is a minimum. The n×n matrix Q and the r×r matrix R are real symmetric positive semi definite and real symmetric positive definite respectively.

Substituting eqs. (5.2), (5.4), (5.7) and (5.9) into eqn.(5.10), we obtain

$$\boldsymbol{u^*(t) = -K[x(t) + L_2e(t)]} \quad \boldsymbol{(5.12)}$$

Inserting eq. (5.12) into eq. (5.1) yields

$$\boldsymbol{\dot{x}(t) = \tilde{A}x(t) + \tilde{B}e(t)} \quad \boldsymbol{(5.13)}$$

Where

$$\boldsymbol{\tilde{A} = A - BK} \quad \boldsymbol{(5.14)}$$

$$\boldsymbol{\tilde{B} = -BKL_2} \quad \boldsymbol{(5.15)}$$

From eq. (5.12), it can be seen that the solutions of eq. (5.8) and (5.13) are necessary to compute the optimal control law $u^*(t)$.

5.1.1 Analysis of Linear Optimal Control Systems incorporating Observers

We express the state vector x (t) and error vector e (t) in terms of orthogonal functions as

$$\boldsymbol{x(t) \approx \sum_{i=0}^{m-1} x_i \phi_i\,(t) = X\phi(t)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.16)}$$

$$\boldsymbol{e(t) \approx \sum_{i=0}^{m-1} e_i \phi_i\,(t) = E\phi(t)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.17)}$$

Where

$$\boldsymbol{X = \left[x_{0,}x_1, \ldots\ldots\ldots\ldots\ldots\ldots\ldots, x_{m-1}\right]} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.18)}$$

$$\boldsymbol{E = \left[e_{0,}e_1, \ldots\ldots\ldots\ldots\ldots\ldots\ldots, e_{m-1}\right]} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.19)}$$

Which are unknowns and $\phi(t)$ is a BPF vector

On integration of equation (5.8), we get

$$\boldsymbol{E = E_0 + FEP} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.20)}$$

Where

$$\boldsymbol{E_0 = \left[e(t)_{0,}e(t_0), \ldots\ldots\ldots\ldots\ldots\ldots\ldots, e(t_0)\right]} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.21)}$$

Similarly from eq. (5.13), we get

$$\boldsymbol{X = X_0 + \left(\widetilde{A}X + \widetilde{B}E\right)P} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.22)}$$

Where

$$\boldsymbol{X_0 = \left[x(t)_{0,}x(t_0), \ldots\ldots\ldots\ldots\ldots\ldots\ldots, x(t_0)\right]} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.23)}$$

Eqs. (5.20) and (5.22) are to be solved for the unknowns E and X, which in turn determines the control law u*(t) from eq. (5.12)

$$\boldsymbol{u^*(t) = -K[X + L_2E]\phi(t)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(5.24)}$$

Now the eqs. (5.20) and (5.22) are solved by a recursive algorithm via BPF's

Substituting the operational matrix P in eqs, (5.20) and (5.22) and simplifying we obtain

$$\boldsymbol{e_0} = \left(\boldsymbol{I_q} - \mathbf{0.5}\boldsymbol{TF}\right)^{-\mathbf{1}}\boldsymbol{e}(\boldsymbol{t_0}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. (\mathbf{5.25})$$

$$\boldsymbol{e_1} = \left(\boldsymbol{I_q} - \mathbf{0.5}\boldsymbol{TF}\right)^{-\mathbf{1}}\left(\boldsymbol{I_q} + \mathbf{0.5}\boldsymbol{TF}\right)\boldsymbol{e_{i-1}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. (\mathbf{5.26})$$

For i=1, 2, 3…, m-1

$$\boldsymbol{x_0} = \left(\boldsymbol{I_n} - \mathbf{0.5}\boldsymbol{T\widetilde{A}}\right)^{-\mathbf{1}}\left[\boldsymbol{x}(\boldsymbol{t_0}) + \mathbf{0.5}\boldsymbol{T\widetilde{B}e_0}\right] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. \ (\mathbf{5.27})$$

$$\boldsymbol{x_i} = \left(\boldsymbol{I_n} - \mathbf{0.5}\boldsymbol{T\widetilde{A}}\right)^{-\mathbf{1}}\left[\left(\boldsymbol{I_n} + \mathbf{0.5}\boldsymbol{T\widetilde{A}}\right)\boldsymbol{x}(\boldsymbol{t_0}) + \mathbf{0.5}\boldsymbol{T\widetilde{B}}(\boldsymbol{e_{i-1}} + \boldsymbol{e_i})\right] \ldots\ldots\ldots\ldots\ldots\ldots .. (\mathbf{5.28})$$

For i=1, 2, 3…, m-1

Problem Statement: - Consider a linear time-invariant system which is represented by the state space matrices given in the following

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u(t)$$

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} -0.6 \\ 0.35 \end{bmatrix}$$

$$y(t) = [1 \quad 0][x_1(t) \quad x_2(t)]^T$$

Where the optimal control law can be taken to be

$$u^*(t) = -K\hat{x}(t) = [1.5 \quad 1][\hat{x}_1(t) \quad \hat{x}_2(t)]^T$$

There is no complete measurement of state, hence $\hat{x}(t)$ can be obtained by the Luenberger observer. Let us choose

$$L_1 = [1 \quad 1.5]^T \text{ and } L_2 = [0 \quad 1]^T$$

Substituting L_1 and L_2 in eq. (5.4) we get

$$\begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} y(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} z(t)$$

$\mathcal{T}$ is determined from eq. (5.9) , G is known from eq. (5.5) and F,H are known from eq. (5.6). The obtained F, G and H values are substituted in eq. (5.3)

$$\dot{z}(t) = -1.5z(t) - u(t) - 1.25Hy(t) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.29)$$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 \\ -1 \end{bmatrix}; C = [1 \quad 0]; K = [1.5 \quad 1]; F = -1.5; G = -1; H = -1.25;$$

$$L_1 = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix}; L_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \mathcal{T} = [-1.5 \quad 1]$$

$$\tilde{A} = A - BK = \begin{bmatrix} 0 & 1 \\ -0.5 & -1 \end{bmatrix} \text{ and } \tilde{B} = -BKL_2$$

Substituting F in eqs. (5.25) and (5.26) to compute error vector. Similarly $\tilde{A}$ and $\tilde{B}$ are substituted in recursive algorithm in eqs. (5.27) and (5.28) to compute state vector. Therefore, for m=4 and 9, the states and error vector is computed from the recurrence relation via BPF's. The results are plotted against different time intervals and compared with the plots of the true solutions.

Graphs

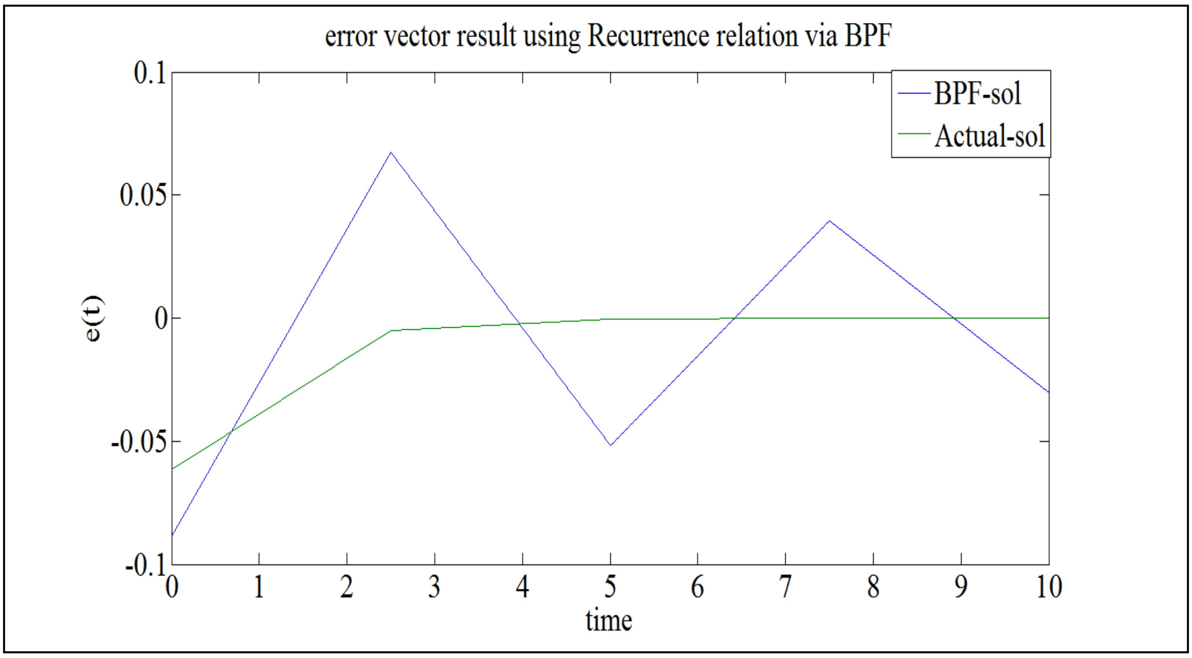

Figure 5.1: Error versus time, t=10sec, m=4

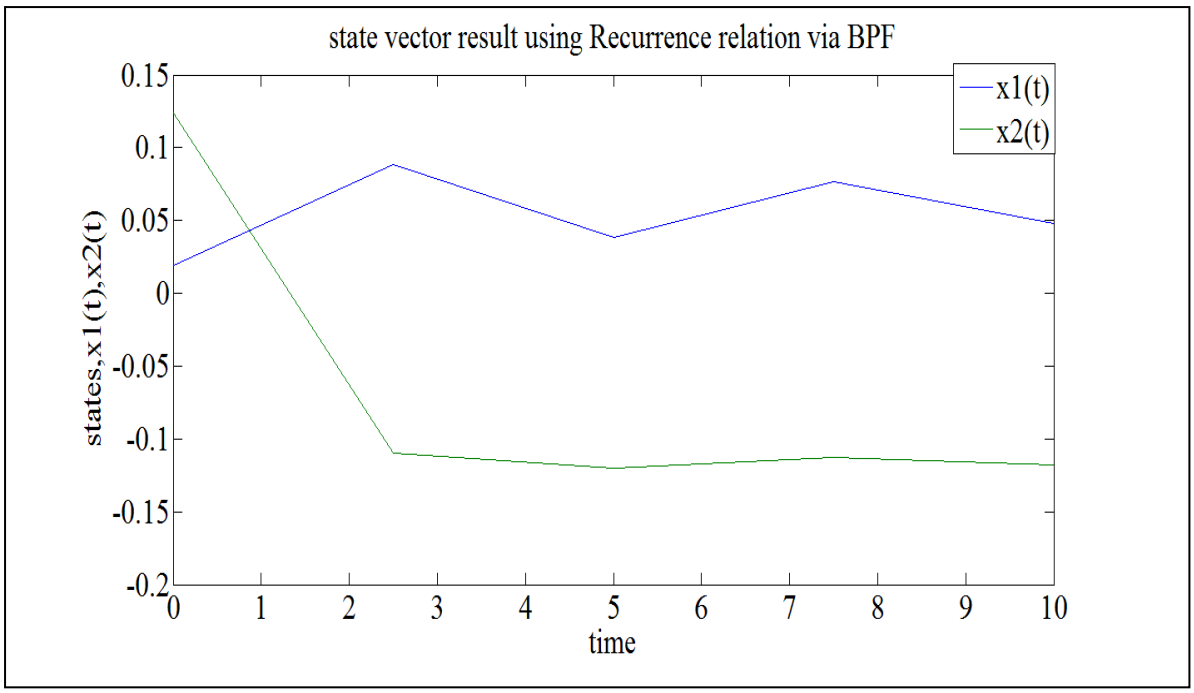

Figure 5.2: State variables versus time, t=10sec

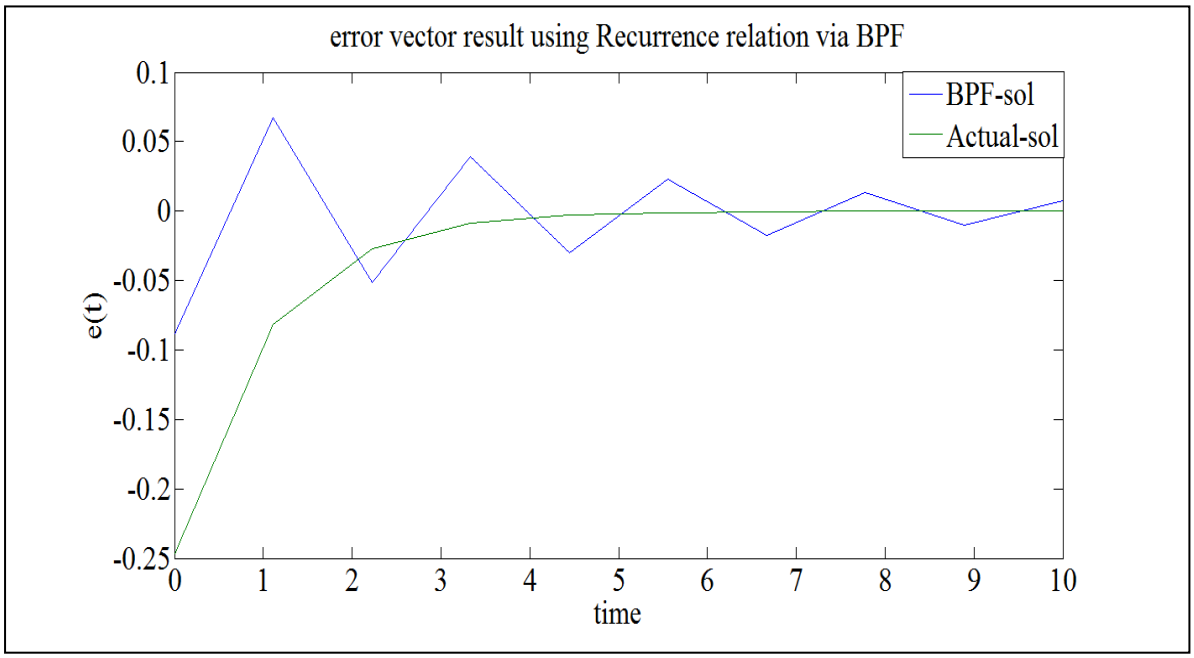

Figure 5.3: Error versus time, t=10sec, m=9

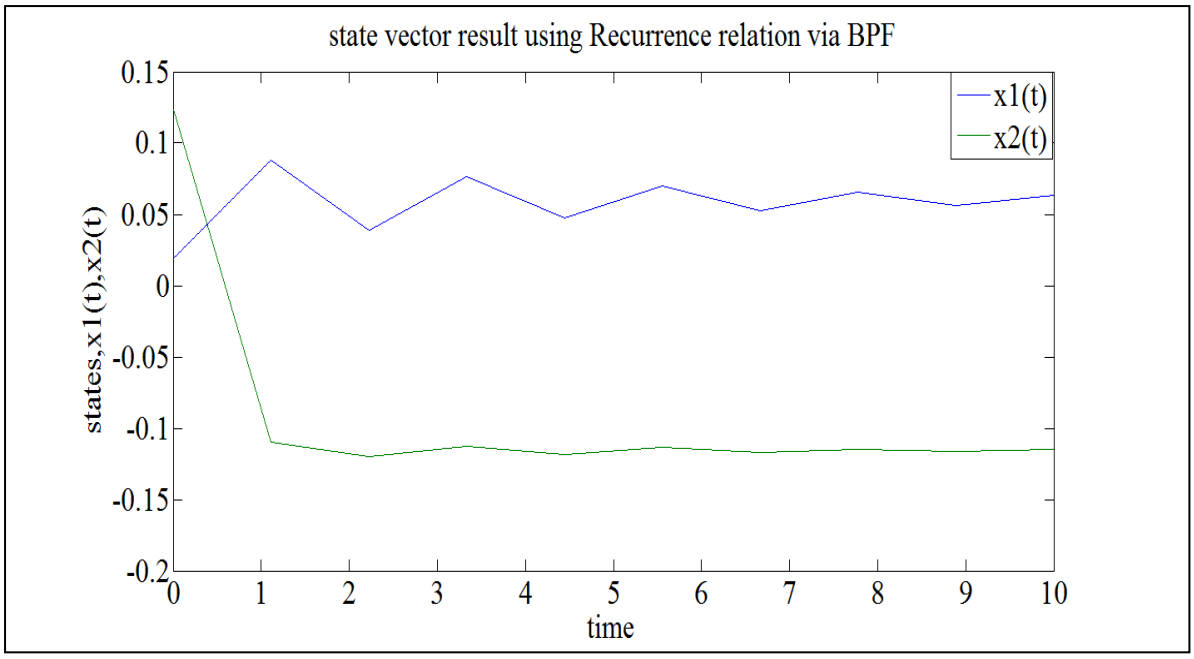

Figure 5.4: State variables versus time, t=10sec, m=9

5.1.2 Results and Discussions

- The states and error are determined successfully by the recursive algorithm obtained via BPF and the results are plotted against different time intervals and matrix dimensions.
- For small matrix dimension say m=4, there is a large difference in the error estimated by the proposed method and the error resulting from actual solution.
- For a little large m say 9, previously observed deviations got minimized.
- From literature it can be stated that recursive algorithms are computationally more attractive than non-recursive methods.
- As block pulse functions are piece-wise constant, they approach the original solution with large matrix sizes however they are computationally very fast.

CHAPTER-6

CONCLUSIONS AND FUTURE SCOPE

CONCLUSION

- BPF with an operational matrix, P is an efficient method in simulating the differential equations of linear reacting systems.
- However they are not that efficient in simulating all the non-linear systems. Non-linear differential equations with exponential terms in them are difficult to be solved by this method.
- Recurrence relation in BPF and TF domain is proved to be effective in solving all the linear system of differential equations.
- Recurrence relation in BPF domain can successfully solve non-linear CSTR system.
- Optimality criterion for control problems can be achieved by recursive algorithms via BPF and also the states, input and error can be determined.

FUTURE SCOPE

- Block pulse functions can be applied to Heat exchangers, distillation processes and coal plant simulations. In addition, the other important orthogonal functions like triangular function sets with operational matrices can be applied to these processes.
- The recurrence relation using triangular functions can be developed and implemented on the processes.
- System identification, parameter estimation and control are the important phenomena of system analysis. As system identification has been worked upon, parameter estimation of chemical processes can be worked in the future.

REFERENCES

1) A. Ganti Prasad Rao, “Piecewise Constant Orthogonal Functions and their Application to System and Control,” Springer-Verlag (1983).
2) A.Deb, G .Sarkar and S.K. Sen (1994),”Block pulse functions, the most fundamental of all piecewise constant basis functions,” Int. J. Syst. Sci., 25(2), 351-363.
3) Beauchamp K.G, “Applications of Walsh and Related Functions with an Introduction to Sequency theory,” Academic Press, London, 1984.
4) C.F.Chen, Y.T.Tsay and T.T.Wu, “Walsh operational matrices for fractional calculus and their applications to distributed systems”, Journal- The Franklin Institute, vol.303, no.3, pp:267-284, (1977)
5) C.Hwang, D.H.Shih and F.C.Kung, “Use of Block-pulse functions in optimal control of deterministic systems”, International Journal of Control, vol.44, no.2, pp: 613-624,(1986).
6) C.T Liou and Y.S Chou, “Piecewise linear polynomial functions and their applications to analysis and parameter identification,” International Journal of Systems and Science, vol.18, no. 10, pp.1919-1929, (1987).
7) D.Leunberger, “Observing the state of a linear system,” IEEE Trans. on Military electronics,vol.8,pp:74-80,1964
8) E. Babolian and Z. Masouri, “Direct method to solve Volterra integral equation of the first kind using operational matrix with block-pulse functions”, Applied Mathematics and Computation, 220 (2008) 51-57.
9) E. Coddington and N. Levinson, “Theory of Ordinary Differential Equations”, McGraw Hill, New York, (1955).
10) E. Hashemizadeh, K. Maleknejad and M.R.Mohsenyzadeh, “Hybrid Orthonormal Bernstein and Block-Pulse Functions for Solving Fredholm Integral Equations”, Proceedings of the World Congress on Engineering 2013 Vol I, WCE(2013), London, U.K.
11) G. Birkhoff and G. Rota, “Ordinary Differential Equations”, Blaisdell, Waltham, (1969).
12) G. Froment, and K.B Biscoff, “Chemical Reactor Analysis and Design”, John Wiley &sons, 2nd Edition, New York (1990).
13) G.P.Rao., L Sivakumar. (1978) “Analysis and synthesis of dynamic systems containing time-delays via block-pulse functions,” Proc. IEE, 125, 1064-1068.
14) H. Khalil, “Nonlinear Systems”, Macmillan, (1992).

15) H.R. Marzban., M. Razzaghi, (2004), "Optimal control of linear delay systems via hybrid of block-pulse and Legendre polynomials," J. Franklin Inst., 341, 279-293.
16) I.S.MetCalfe, "Chemical Reaction Engineering", Oxford University Press Oxford UK (1997).
17) J.h.Chou and I.R.Horng, "Shifted Legendre series analysis of linear optimal control systems incorporating observers", International Journal of Systems and Science, vol.16, no.7, pp: 863-867,(1985).
18) J.H.Park, Korea National University of Transportation, Republic of Korea, "Trajectory Optimization for Large Scale Systems via Block Pulse Functions and Transformations", International Journal of Control and Automation Vol. 5, No. 4, December, (2012).
19) K. Maleknejad and K. Mahdiani ,"Iterated Block-Pulse Method for Solving Volterra Integral Equations", Applied Mathematics (2012), 2(1): 17-20
20) K. Ogata, Modern control engineering (4^{th} ed.), Pearson Education Asia, New Delhi, 2002.
21) K.B.Datta and B.M.Mohan, "Orthogonal Functions in Systems and Control", Advanced Series in Electrical and Computer Engineering, vol.9, World Scientific, Singapore, (1995).
22) M. Vidyasagar, "Nonlinear Systems Analysis", Prentice Hall, 2nd Edition (1992).
23) M.Razzaghi, H.R. Marzban (2000) "Direct method for variational problems via hybrid of block-pulse and Chebyshev functions," Math. Probl. Eng., 6, 85-97.
24) N.K.Sinha and Z.Q.Jie, "State estimation using block pulse functions", International Journal of Systems and Science, vol.15,no.4,pp:341-350,(1984)
25) P. Sannuti "Analysis and Synthesis of dynamic systems containing time delays via block pulse functions," Proc.IEE, vol.125, no.9, pp.1064-1068,(1978).
26) P. Sannuti "Analysis and synthesis of dynamic systems via block-pulse functions," Proc. Inst. Electr. Eng., 124, 569-571, (1977).
27) P. Schiavane, C. Constanda, A. Mioduchowski' (2002); "Integral Methods in Science and Engineering'', Birk¨auser, Boston.
28) R. Cole, "Theory of Ordinary Differential Equations" Appleton-Century- Crofts, New York, (1968).
29) T.T.Lee, S.C.Tsay and I.R.Horng, "Shifted Jacobi series analysis of linear optimal control systems incorporating observers", Journal The Franklin Institute, vol.321, no.5, pp: 289-298,(1986).

30) Z.h.Ziang and W.Schaufelberger, “Block pulse functions and their Applications in Control Systems”, LNCIS 179, Springer, Berlin,(1992).