Memetic Heuristic Computation for Nonlinear Problems in Engineering

By

Suheel Abdullah Malik
Reg. No. 16-FET/PHDEE/S08

A dissertation submitted to I.I.U. in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Electronic Engineering
Faculty of Engineering and Technology
INTERNATIONAL ISLAMIC UNIVERSITY
ISLAMABAD
2015

Electronic engineering

Electronics - technology

DEDICATED TO

My Grandmothers (Late)

My Parents

My Father in Law (Late)

My Wife

My Daughters Saroosh and Sundus

CERTIFICATE OF APPROVAL

Title of Thesis: Memetic Heuristic Computation for Nonlinear Problems in Engineering

Name of Student: Suheel Abdullah Malik

Registration No: 16-FET/PHDEE/S08

Accepted by the Department of Electronic Engineering, Faculty of Engineering and Technology, International Islamic University, Islamabad, in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electronic Engineering.

Viva voce committee:

Prof. Dr. Aqdas Naveed Malik

Dean FET, IIU Islamabad

Prof. Dr. Muhammad Amir (Chairman)

Chairman DEE, FET, IIUI

Prof. Dr. Abdul Jalil (External Examiner-1)
Pakistan Institute of Engineering and Applied Science
Nilore, Islamabad

Prof. Dr. Tanveer Ahmed Cheema (External Examiner-2)
HOD, DEE, ISRA University, Islamabad Campus

Dr. Adnan Umar Khan (Internal Examiner)
Assistant Professor, DEE, FET, IIUI

Prof. Dr. Ijaz Mansoor Qureshi (Supervisor)

IMQureshi

Dr Ex-Pakistan leave

Department of Electrical Engineering
Air University, Islamabad

Dr. Muhammad Zubair (Co-supervisor)

Assistant Professor, DEE, FET, IIUI

Abstract

This dissertation presents new heuristic computational schemes for solving the nonlinear problems in engineering that are governed by nonlinear ordinary differential equations (NODEs) and nonlinear partial differential equations (NPDEs).

The heuristic schemes comprising of Evolutionary Algorithms (EAs) and a linear combination of some basis functions are presented for solving NODEs. The approximate solution of NODEs is deduced as a linear combination of some basis functions with unknown parameters. Three different basis functions including log sigmoid, Bernstein polynomials, and polynomial basis have been used for the approximate modeling. A fitness function is used to convert the NODE into an equivalent global error minimization problem. Two popular EAs including Genetic Algorithm (GA) and Differential Evolution (DE), and local search techniques, such as, Interior Point Algorithm (IPA) and Active Set Algorithm (ASA) are used to solve the minimization problem and to obtain the unknown parameters. The memetic algorithm schemes combining GA with IPA (GA-IPA) and GA with ASA (GA-ASA) are also explored. The schemes have been tested on various nonlinear problems including Bratu problem, Duffing van der pol oscillator, Michaelis-Menten biochemical reaction system, and power-law fin-type problem.

An elegant hybridization of Exp-function method with nature inspired computing (NIC) has been presented for the numerical solution of NPDEs. Exp-function method is used to express the travelling wave solution of the given NPDE. The NPDE is converted into an optimization problem. Two popular NIC techniques including GA and particle swarm optimization (PSO) are used to solve the optimization problem. The scheme has been successfully tested on some important NPDEs including generalized Burger-Fisher, Burger-Huxley, and Fisher equations.

The proposed numerical solutions are found in a good agreement with the exact solutions and quite competent with those reported by some well-known classical methods like adomian decomposition method (ADM), variational iteration method (VIM), and homotopy perturbation method (HPM). It is also observed that the memetic algorithm schemes are good choice for the optimization of such problem.

The presented schemes are simple as well as efficient, and they provide the numerical solution not only at the grid points but also at any value in the solution domain.

LIST OF PUBLICATIONS

[1] **S. A. Malik**, I. M. Qureshi, M. Zubai, and M. Amir, "Hybrid heuristic computational approach to the Bratu problem," *Res. J. Recent Sci.*, vol. 2, no 10, pp. 1-8, 2013.

[2] **S. A. Malik**, I. M. Qureshi, M. Zubair, and M. Amir, "Numerical solution to Troesch's problem using hybrid heuristic computing," *J. Basic Appl. Sci. Res.*, vol. 3, no. 7, pp. 10-16, 2013.

[3] **S. A. Malik**, I. M. Qureshi, M. Amir, and I. Haq, "Nature inspired computational technique for the numerical solution of nonlinear singular boundary value problems arising in physiology," *The Scientific World Journal*, vol. 2014, Article ID 837021, doi:10.1155/2014/837021, 2014.

[4] **S. A. Malik**, I. M. Qureshi, M. Zubair, and I Haq, "Memetic heuristic computation for solving nonlinear singular boundary value problems arising in physiology," *Res. J. Recent Sci.*, vol. 2, no. 9, pp. 1-8, 2013.

[5] **S. A. Malik**, I. M. Qureshi, M. Amir, and I. Haq, "Numerical solution of Lienard equation using hybrid heuristic computation," *World Applied Sciences Journal*, vol. 28, no. 5, pp. 636-643, 2013.

[6] **S. A. Malik**, I. M. Qureshi, M. Zubair, and I. Haq, "Solution to force-free and forced Duffing-van der pol oscillator using memetic computing," *J. Basic Appl. Sci. Res.*, vol. 2, no. 11, pp. 11136-11148, 2012.

[7] **S. A Malik**, I. M. Qureshi, M. Amir, and I. Haq, "Numerical solution to nonlinear biochemical reaction model using hybrid polynomial basis differential evolution technique, " *Advanced Studies in Biology*, vol. 6, no. 3, pp. 99 – 113, 2014.

[8] **S. A. Malik**, I. M. Qureshi, M. Amir, and A. N. Malik, "Nature inspired computational approach to solve the model for HIV infection of CD4$^+$T cells," *Res. J. Recent Sci.*, vol. 3, no. 6, pp. 1-7, 2014.

[9] **S. A. Malik**, I. M. Qureshi, M. Amir, A. N. Malik, and I. Haq, "Numerical solution to Riccati equations using evolutionary algorithm technique ybridized with Bernstein polynomials," *MAGNT Research Report*, vol. 2, no. 6, pp. 49-60, 2014.

[10] **S.A. Malik**, A Ullah, I. M. Qureshi, and M. Amir, "Numerical solution to Duffing equation using hybrid genetic algorithm technique" *MAGNT Research Report*, vol. 3, no.2, pp. 21-30, 2015.

[11] A. A. Khaliq, J. A. Shah, and **S. A. Malik**, " De-noising of functional magnetic resonance imaging (fMRI) data using nonlinear anisotropic 1D and 2D filters," *International Journal of Advanced Research in Computer Science*, vol. 4, no. 4, pp. 13-17, 2013.

[12] A. A. Khaliq, I. M. Qureshi, J. A. Shah, and **S. A. Malik**, "Differential covariance based algorithm for blind source separation," *International Journal of Advanced Studies in Computer Science and Engineering*, vol. 2, no. 2, pp. 8-12, 2013.

[13] A. A. Khaliq, I. M. Qureshi, J. A. Shah, and **S. A. Malik**, "High order differential covariance based source separation of Monkey's fMRI data," *International Journal of Engineering Sciences and Research Technology*, vol. 2, no. 5, pp. 1287-1292, 2013.

[14] A. A. Khaliq, I. M. Qureshi, **S. A. Malik**, and J. A. Shah," Covariance based BSS algorithm for functional magnetic resonance imaging (fMRI) data source separation," *Research Journal of Recent Sciences*, vol. 2 , no. 9, pp. 86-91, 2013.

[15] A. A. Khaliq, I. M. Qureshi, **S. A. Malik**, and J. A. Shah, "Eigen decomposition based blind source separation," *International Journal of Engineering Sciences and Research Technology*, vol. 2, no. 11, pp. 3277-3280, 2013.

[16] A. A. Khaliq, I. M. Qureshi, J. A. Shah, **S. A. Malik**, and **I.** Haq, "Blind source separation of fMRI signals using Joint diagonalization algorithm," *Research Journal of Applied Sciences, Engineering and Technology* , vol. 7, no. 2, pp. 233-239, 2014.

[17] A. A. Khaliq, I. M. Qureshi, J. A. Shah, **S. A. Malik**, and I. Haq, "De-noising functional magnetic resonance imaging (fMRI) data using exponential gradient

filter," *Middle East Journal of Scientific Research*, vol. 18 (9), pp. 1349-1356, 2013.

[18] S. Hafeez, A. S. Fazal, **S. A. Malik**, and K.S.Alimgeer, " Comparison of lumped element UWB printed filter with discrete components", *Research Journal of Recent Sciences*, vol. 3, no. 1, pp. 91-96, 2014.

[19] S. Khursheed, A. A. Khaliq, J. A. Shah, **S. A. Malik**, and S. Khan, "A hybrid logarithmic algorithm for poisson noise removal in medical images," *Advanced Studies in Biology*, vol. 6, no. 4, pp. 181-192, 2014.

[20] S. Khursheed, A. A. Khaliq, J. A. Shah, **S. A. Malik**, and S. Khan, " Third order NLM filter for poisson noise removal from medical images," *MAGNT Research Report*, vol.2, no. 7: PP. 482-489, 2014.

[21] S. Khursheed, A. A. Khaliq, **S. A. Malik**, J. A. Shah, and S. Khan, "Poison noise removal using wavelet transformer, *Science International* (in press).

[22] K. J. Krishnan, **S. A. Malik**, A. Kalam, and A. Zayegh, "Need for hydrogen energy and fuel cell technology in Australia", in *Proceedings of the International Conference on Thermal Energy and Environment*, paper number EMR 177, Madurai, India, 2011.

LIST OF SUBMITTED PAPERS

[1] **S. A. Malik**, I. M. Qureshi, A. N. Malik, M. Amir, and I. Haq, "Exp-function method combined with memetic heuristic computation for solving Fisher's equations," *The Scientific World Journal*.

[2] **S. A. Malik**, I. M. Qureshi, M. Amir, A. N. Malik, and I. Haq, "Numerical solution to generalized Burger-Fisher equation using Exp-function method hybridized with heuristic computation," *PLOS ONE*.

[3] **S. A. Malik**, I. M. Qureshi, M. Amir, A. N. Malik, and I. Haq, "A new heuristic scheme for solving the generalized Burgers-Huxley equation," *Applied Mathematics and Information Sciences.*

[4] **S. A. Malik**, I. M. Qureshi, M. Amir, A. N. Malik, and I. Haq, "Hybrid evolutionary algorithm technique combined with Bernstein polynomials for solving nonlinear fin problem," *Science International.*

ACKNOWLEDGEMENTS

First and foremost I am thankful to almighty Allah for his eventual blessing and guidance at each step of this work. Millions of Praise for the wonderful human and the supreme of wisdom Prophet Muhammad (Peace be upon him).

I would like to truthfully acknowledge the brilliant supervision and superb guidance of Dr. Ijaz Mansoor Qureshi during this research work. He is my ideal teacher and true source of inspiration. His valuable suggestions and articulate approach helped me to conceive the overall direction of the research and contributed immensely to the success of the work.

I am grateful to my colleagues Dr. Muhammad Amir, Dr. Ihsanul Haq, and Mr. Jawad Ali for their fruitful suggestions. I thank all my friends who helped me when I needed them. I am highly indebted to Dean FET Dr. Aqdas Naveed, Dr. Muhammad Afzal, Dr. Saeed Badshah, and chairman DEE Dr. Amir for their kind encouragement and for providing me a comfortable working environment.

I appreciate gratefully to my parents who remain desirous for my best state of Emaan, health and success. My special thanks to my brother Mr. Ubaid-ul-Islam for his services rendered during the publications. I am also thankful to my brother Tajamul for providing me technical assistance and printing facilities whenever I needed them. I am thankful to my brothers Dr. Junaid, Dr. Muslim, and Mr. Waleed Rasool for their moral support during the research work. I am also thankful to Aunty and my sisters for their preyers. My special thanks to my little nieces Taqdees and Nosiaba, nephew Udaan, and brother Masoom for their innocent good wishes.

My special gratitude to my daughters Saroosh Jan and Sundus Bazigha who at many occasions forfeited their wishes for my studies, and whose modest gestures were an elusive source of inspiration to me. Last but not the least I am thankful to my wife Sabia for her endless encouragement and support, who proved to be embodiment of endurance and sacrifice throughout untiring research years. I am also thankful to I.I.U. for giving me full fee waiver throughout the period of PhD studies.

(Suheel Abdullah Malik)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

ADM	Adomian Decomposition Method
ASA	Active Set Algorithm
BPCM	B-spline Based Collocation Method
CNODE	Coupled Nonlinear Ordinary Differential Equation
DE	Differential Evolution
DTM	Differential Transform Method
EA	Evolutionary Algorithm
GA	Genetic Algorithm
GA-ASA	GA Hybridized with ASA
GA-IPA	GA Hybridized with IPA
GA-PS	GA Hybridized with PS
HPM	Homotopy Perturbation Method
HWM	Haar Wavelet Method
IPA	Interior Point Algorithm
MA	Memetic Algorithm
MHPM	Modified Homotopy Perturbation Method
MVIM	Modified Variational Iteration Method
NIA	Nature Inspired Algorithm
NIC	Nature Inspired Computation
NODE	Nonlinear Ordinary Differential Equation
NPDE	Nonlinear Partial Differential Equation
ODE	Ordinary Differential Equation
OHAM	Optimal Homotopy Asymtotic Method
PS	Pattern Search
PSO	Particle Swarm Optimization
RK4	Fourth-Order Runge-Kutta Method
VHPM	Variational Homotopy Perturbation Method
VIM	Variational Iteration Method

CHAPTER 1

INTRODUCTION

1.1 DESCRIPTION OF THE PROBLEM

Most of the nonlinear problems arising in diverse fields of engineering and science are by and large modeled by the nonlinear differential equations. The systems of nonlinear ordinary differential equations (NODEs) have sustained their importance due to their wide ranging applications from classical areas of engineering to more recent use in biology, chemistry, economics, and ecology. Nonlinear partial differential equations (NPDEs) govern the fundamental laws of nature and are crucial to many areas of engineering and science such as fluid dynamics, population models, plasma physics, and nonlinear optics etc. Due to their central role in engineering and science, NODEs and NPDEs are studied extensively by the research community including mathematicians, engineers, and scientists.

A good number of nonlinear problems governed by NODEs and NPDEs have been solved effectively using the classical and modern techniques such as Runge Kutta method, Variational iteration method (VIM), Adomian decomposition method (ADM), homotopy perturbation method (HPM), and Exp-function method.

The advent of high speed computing and the consequent emergence of scientific computation have witnessed much interest in new analytical and numerical approximation techniques coupled with computational algorithms. Recently, the evolutionary computation based techniques have been employed by some of the researchers as an alternate to the standard methods for numerically solving the nonlinear problems. However, comparatively lesser amount of work has been carried for the numerical solution of nonlinear problems modeled by NODEs, coupled nonlinear ordinary differential equations (CNODEs), and NPDEs. A lot more needs to be explored and a good number of these problems need to be solved numerically. Therefore, investigating the numerical solutions to these nonlinear problems especially CNODEs and NPDEs is still a crucial task and a great area of research.

Motivated by the potential of nature inspired computation based techniques and the importance of the nonlinear problems in engineering, this research work considers numerical solution of NODEs, CNODEs, and NPDEs through nature inspired computation. The key issues which have to be addressed in this regard are as follows.

- To devise efficient heuristic technique for solving systems of nonlinear differential equations using nature inspired computation (NIC).
- The log sigmoid based NIC technique suggested and used in [1] should be exploited for solving other NODEs.
- One should investigate the combination of polynomial basis functions and heuristic computation based methodology for numerically solving nonlinear NCODEs in chemical reaction system and biomedical engineering problems.
- A novel scheme based on the elegant hybrid approach of the well-known Exp-function method and NIC should be investigated for the numerical solution of NPDEs.

- The applicability, efficacy, and reliability of the proposed heuristic schemes should be investigated on various nonlinear problems.

1.2 CONTRIBUTIONS OF THE DISSERTATION

This dissertation presents some new stochastic heuristic schemes for obtaining the numerical solution of nonlinear problems in engineering governed by NODEs, CNODEs, and NPDEs. Although we have adopted four different heuristic schemes for solving NODEs, CNODEs, and NPDEs, but all the schemes share the common concept of formulating a problem exclusive fitness function, which represents a trial solution of the given nonlinear problem.

The basic idea of the proposed heuristic schemes is to transform the given nonlinear problem into an equivalent optimization problem using a problem exclusive fitness function with unknown parameters. The optimization problem is solved using the application of nature inspired optimization techniques. The main contributions of this thesis are given below.

1. *Application of Evolutionary Computing Technique With Log Sigmoid Basis For Solving Nonlinear ODEs*

   This technique which was suggested and used in [1] has been employed to solve the Bratu problem, Troech's problem, Duffing van der pol oscillator equation, as well as nonlinear singular boundary value problems in physiology. The approximate solutions are found in a good agreement with the exact solutions and quite comparable or better than some of the classical methods.

2. *Polynomial Basis Along With Hybrid Evolutionary Algorithm Technique for Solving CNODEs*

   A heuristic approach based on the combination of polynomial basis and hybrid evolutionary algorithms is applied for solving CNODEs. A linear combination of polynomial basis with unknown parameters is used to construct the approximate solution. The CNODE is converted into a global error minimization problem. Hybrid evolutionary algorithms are used to solve the minimization problem and to achieve the unknown parameters. The technique has been applied for the first time to Michaelis-Menten nonlinear biochemical reaction system and HIV infection model of $CD4^{+}T$ cells. The proposed scheme has shown supremacy on some well-known traditional method in obtaining the solution of biochemical reaction model with greater accuracy.

3. *Hybridization of Exp-function Method with Nature Inspired Computing for Solving Nonlinear Partial Differential Equations (NPDEs).*

   An elegant hybrid approach of Exp-function method and nature inspired computing is for the first time attempted to obtain the numerical solution of NPDEs. Exp-function method is used to express the travelling wave solution of the given NPDE which is converted into an optimization problem. Nature inspired algorithms are used to solve the optimization problem. The scheme has been successfully tested on some important NPDEs including generalized Burger-Fisher, Burger-Huxley and Fisher equations. The numerical results from the proposed scheme are found in sharp agreement with the exact solutions.

4. *Bernstein Polynomials based Stochastic Technique for Solving NODEs.*

A heuristic technique based on the couple of Bernstein polynomials and nature inspired algorithms is applied for obtaining the approximate solution of NODEs. Bernstein polynomial basis with unknown coefficients are used to construct the approximate solution of the NODE. Hybrid genetic algorithm is used to solve the optimization problem and to obtain the unknown coefficients. The technique has been applied to the strongly nonlinear power law-fin-type problem and nonlinear Riccati equation. Comparisons of numerical results validate the effectiveness and reliability of the suggested technique.

1.3 ORGANIZATION OF THE DISSERTATION

Chapter 2, starts with an introduction of nature inspired optimization algorithms, followed by an overview of global and local search techniques. It also provides the literature review regarding the applications of nature inspired algorithms in solving diverse optimization problems in engineering and science.

Chapter 3, gives the description of the hybrid log sigmoid basis evolutionary algorithm technique for solving NODEs. Moreover procedural steps of hybrid genetic algorithm (HGA) are provided. The application of the devised technique is illustrated on several nonlinear problems including the Bratu problem, Troesch's problem, Duffing van der pol oscillator equation, and nonlinear singular boundary value problems in physiology.

Chapter 4, introduces the hybrid polynomials basis evolutionary technique for solving nonlinear coupled ordinary differential equations. It gives essential details of evolutionary algorithms such as Genetic Algorithm (GA) and Differential Evolution

(DE). It illustrates the application of the proposed approach to nonlinear biochemical reaction model and HIV infection model of CD4$^+$T cells. To our knowledge these problems are solved for the first time using such a scheme.

Chapter 5, presents a new scheme based on the couple of Exp-function method and nature inspired algorithms for solving NPDEs. It gives an overview of the Exp-function method. Some essential procedural steps of the nature inspired algorithms are provided. An elegant couple of Exp-function method and nature inspired algorithms is for the first time attempted as per our literature survey. The scheme is implemented to solve well-known NPDEs including generalize Burger-Fisher, Burger-Huxley, and Fisher equations. The viability of the proposed scheme is also illustrated by various simulations.

Chapter 6, starts with an introduction of the Bernstein polynomials. Heuristic approach combining Bernstein polynomials and nature inspired algorithms is described. The proposed approach is applied to solve nonlinear Riccati differential equations and power-law fin-type problem with high order nonlinearity to illustrate its effectiveness. The reliability is also tested by many simulations. A study of comparative analysis between log sigmoid based technique and Bernstein polynomials technique is also presented.

Chapter 7, gives the summary of the work described in this dissertation. Some future work directions also made at the end.

CHAPTER 2

NATURE INSPIRED OPTIMIZATION ALGORITHMS

2.1 IINTRODUCTION

In past few decades, many optimization methods have been suggested for solving optimization problems. Although deterministic optimization methods perform well on many problems, they are not efficient in solving highly nonlinear and large scale combinatorial problems [2]. Also, these methods need huge computational efforts, which tend to fail as the problem size increases [3]. Nature inspired algorithms (NIAs) are metaheuristics that mimics the nature for solving optimization problems. NIAs are computationally efficient and derivative-free global optimization methods that work well on noisy target functions which have many local optima. The main thrust behind NIAs is the nature itself, which is the real source of inspiration for solving complex and stiff problems efficiently. Moreover, NIAs can handle highly nonlinear and high dimensional problems efficiently due to their characteristics such as adaptability, parallelization, robustness, and cooperation, which also makes these techniques suitable for scientific computing [4], [5]. In recent years many metaheuristic techniques have been developed, however we confine our study to the evolutionary algorithms (EAs) such as genetic algorithm (GA) and differential evolution (DE), particle swarm optimization (PSO), and pattern search (PS) algorithms, which are briefly introduced in the next sections.

2.2 EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EAs) are subclass of NIAs which are based on the idea of biological evolution in nature. EAs are population based stochastic computational algorithms that employ survival of the fittest philosophy for solving optimization problems. EAs maintain a population of individuals, which represent the potential solutions to the given problem. Each solution is subject to a fitness based selection criteria. New solutions are formed by applying genetic operators, and better solutions are selected for the next generation in an iterative manner. EAs are robust and able to cope with problems with discontinuities. The family of EAs comprises of genetic algorithm (GA), differential evolution (DE), genetic programming (GP) etc. however our study is primarily concerned with GA and DE which are briefly discussed next. The interested readers can find good source of material on EAs in books [6], [7].

2.2.1 GENETIC ALGORITHMS

The genetic algorithm (GA) invented by Holland [8] is one of the most renowned stochastic global search optimization technique in EAs. GA uses the principle of natural selection and genetics by mimicking the nature to solve optimization problems [9], [10]. GA is attractive in solving various optimization problems because they are easy, efficient, and robust [11], [12]. GAs have been used in the optimization of diverse problems in engineering and science such as nonlinear heat conduction problem [13], predictive controller for nonlinear system [14], antenna array thinning problem [15], parameter estimation of chromatography process [16], fuzzy nonlinear problem [17]. Specifically GA has been effectively used as the optimization algorithm for solving various systems of linear and nonlinear differential equations [18], [19], [20], [21], [22], [23], [24], [25],

[26], [27]. Moreover many authors have used GA optimized artificial neural networks (ANNs) for solving various NODEs [1], [28], [29], [30], [31], [32]. Although there is a long list of applications of GA as optimization tool in various problems, only some have been narrated as a reference to confirm the great potential and broad applicability of GAs. The GA commences by creating a population of individuals called chromosome. A chromosome consists of genes typically encoded as a string of values. Each chromosome is regarded as a possible solution to the given problem. Each chromosome is assigned a fitness value which indicates how good solution a particular individual is to the given problem. The individuals within a population are evaluated using a fitness function that is specific to the problem at hand. The algorithm evolves population of individuals iteratively by means of three genetic operations, selection, crossover, and mutation [33]. The parents are selected on the basis of their fitness values which produce offspring. Offspring act as parents for the new generation. It is expected that new offspring would give better solutions over the course of generations. The pseudo-code of the GA is given in algorithm 2.1 and its flow chart is given in Fig. 2.1.

Algorithm 2.1: Genetic Algorithm (GA)

```
Begin
    Initialize population of candidate solutions
    Evaluate each candidate solution
Repeat until (stoppage criterion is satisfied)
    Select parents
    Crossover and generate offspring
    Mutate offspring
```

```
        Evaluate offspring

        Select new population

    End
```

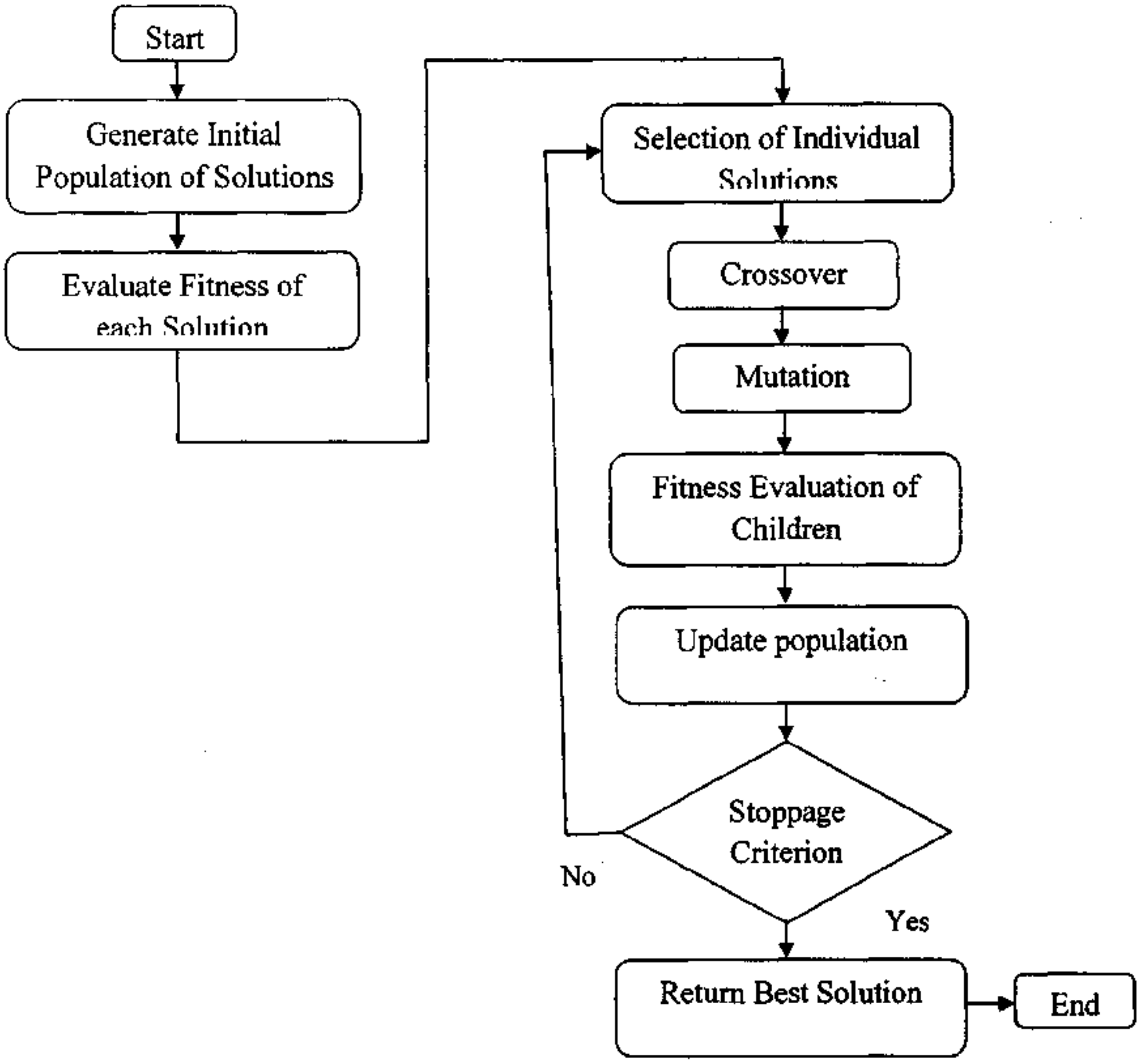
Start
Generate Initial Population of Solutions
Evaluate Fitness of each Solution
Selection of Individual Solutions
Crossover
Mutation
Fitness Evaluation of Children
Update population
Stoppage Criterion
No
Yes
Return Best Solution
End

Fig. 2.1 Flow chart for GA Optimization

The steps of working of GA are now briefly explained. A detailed discussion on the elements and its working of GA can be found in the books [33], [34].

Encoding: The first and important step in GA is the appropriate encoding that is representation of the chromosome. The chromosome is normally represented as a fixed length binary string of 0's and 1's. The choice of encoding greatly affects the performance of GA [33].

Initialization of population: As mentioned before GA is a population based search method. A population is initially created randomly in the search space. The performance of GA largely depends of the size of the initial population [33].

Selection: After the encoding and random population initialization, selection of chromosomes that will act as parents is performed. Selection picks the chromosome in a manner similar to the Darwin's theory of natural selection. At the end of each generation a new population of candidate solutions is selected. The chromosomes with better performance or with the relatively high value of fitness survive through generations. Selection of chromosomes is performed using a problem specific fitness function. Over the years many types of selection operators have been developed which include stochastic uniform, rank, roulette wheel, and tournament.

Crossover: crossover operator is like natural mating in which two different chromosomes swap their genes to produce offsprings. Crossover is explorative, it makes a big jump to an area somewhere "in between" two parent areas. In GAs crossover is carried in many different forms like scattered, heuristic, single point, etc.

Mutation: Finally mutation operator is used to provide genetic diversity from one generation of a population to next to search a broader space. It makes small random changes in the individuals of the population. It is exploitative, it creates random small diversions, thereby staying near around the parent. Some of the types of mutation used in GA are adaptive feasible, Gaussian, and uniform.

This process of selection, crossover, and mutation is continued until the termination criterion such as number of generations, or fitness value, is satisfied.

2.2.2 DIFFERENTIAL EVOLUTION

Differential evolution (DE) developed by Price and Storn [35] is another popular and powerful parallel search global optimization algorithm in EAs. It is a population based algorithm like other EAs, but it differs from others, such as GA, in the mutation and recombination stage. DE produces offspring by mutating the solution vectors with a weighted difference of two randomly picked population vectors. Moreover, DE adopts a one-to-one logic for reproduction which allows replacement of an individual only if the offspring gives better fitness value than its corresponding parent [36]. DE has illustrated its strength and robustness in diverse applications, such as nonlinear system identification [37], edge detection in images [38], control and synchronization of chaotic systems [39], process engineering problems [40], and learning of neural networks (NNs) [41], [42], [43], [44], [45]. Further a survey of variants of DE proposed in recent years and their applications can be found in [36].

DE like other EAs begins by creating a random initial population of individuals in the search space. At each generation, for each individual three unique vectors are selected in the population. The weighted difference of two vectors is added to the third one. A trial vector is produced and compared to the target vector in the population. The one with the lower fitness value survives and becomes parent for the next generation. The basic implementation steps of DE are described below [7], [35].

Population initialization: A population of N chromosomes is randomly generated within the user defined bounds. Each chromosome consists of D number of genes.

Mutation: Mutation operation creates a trial vector for each individual of the current population by perturbing the target vector with a weighted difference of two vectors. For each parent, $x^{i,g}$, three individuals n$_1$, n$_2$, and n$_3$ which are mutually distinct and also different from *i*, are randomly chosen from the population. The mutation is applied to the target vector to produce a perturbed vector according to the following formula.

$$y^{i,g+1} = x^{n_1,g} + F(x^{n_2,g} - x^{n_3,g}) \qquad (2.1)$$

where $F \in [0,2]$ is a real constant which controls the amplification of the weighted difference vector. The mutated vectors perturbed vectors are

Crossover: Subsequent to the mutation crossover operation is applied to the population which introduces diversity in the mutated vectors. The crossover recombines the trial vector and the parent vector to produce offspring as follows.

$$z_j^{i,g+1} = \begin{cases} y_j^{i,g+1} & if\ rand \leq CR\ or\ j = j_{rand} \\ x_j^{i,g} & otherwise \end{cases} \qquad (2.2)$$

where $CR \in [0,1]$ is called crossover constant.

Selection: Following recombination selection is applied to decide which individual should become the member in the next generation. At this stage trial vector is compared with the target vector, and only the fitter one is selected for the next generation. If the trial vector gives better fitness than the target vector it replaces the target vector otherwise the target vector is retained. The selection method is as follows (if a minimization problem is considered).

$$x^{i,g+1} = \begin{cases} z^{i,g+1} & if\ fit(z^{i,g}) < fit(z^{i,g+1}) \\ x^{i,g} & otherwise \end{cases} \qquad (2.3)$$

The above procedure of mutation, crossover, and selection is continued until some stopping criterion like desired fitness or maximum number of generations is reached.

2.3 PARTICLE SWARM OPTIMIZATION

The Particle swarm optimization (PSO) belongs to a broad class of swarm intelligence (SI). The fundamental idea behind the SI techniques is derived from the natural behavior and social interactions of flock of birds, a school of fish, and ant colonies etc. where individuals in a group interact and exchange local information and ultimately solve the complex global objective efficiently [33].

PSO is a stochastic global search optimization algorithm which was introduced by Eberhart and Kennedy in 1995 [46], [47]. PSO applies the concept of simulating the social behavior of birds within a flock for solving optimization problems. Due to the simple concept, ease in implementation, and computational efficiency, PSO has attracted many researchers and practitioners. PSO has been successfully applied to solve many optimization problems involving ODEs such as Riccati equation [48], Wessiger's equation [49], [50], Bagley-Torvik equation [51], nonlinear damped pendulum [52], fluid flow and heat transfer problem [53], fractional ODEs [54], [55]. Although PSO has been used for solving wide variety of problems, only few of its applications to solve optimization problems that are modeled by the differential equations are reported here as a reference. For a comprehensive detail of applications of standard PSO and its variants proposed in recent years, refer to the book [33] and references [56], [57], [58].

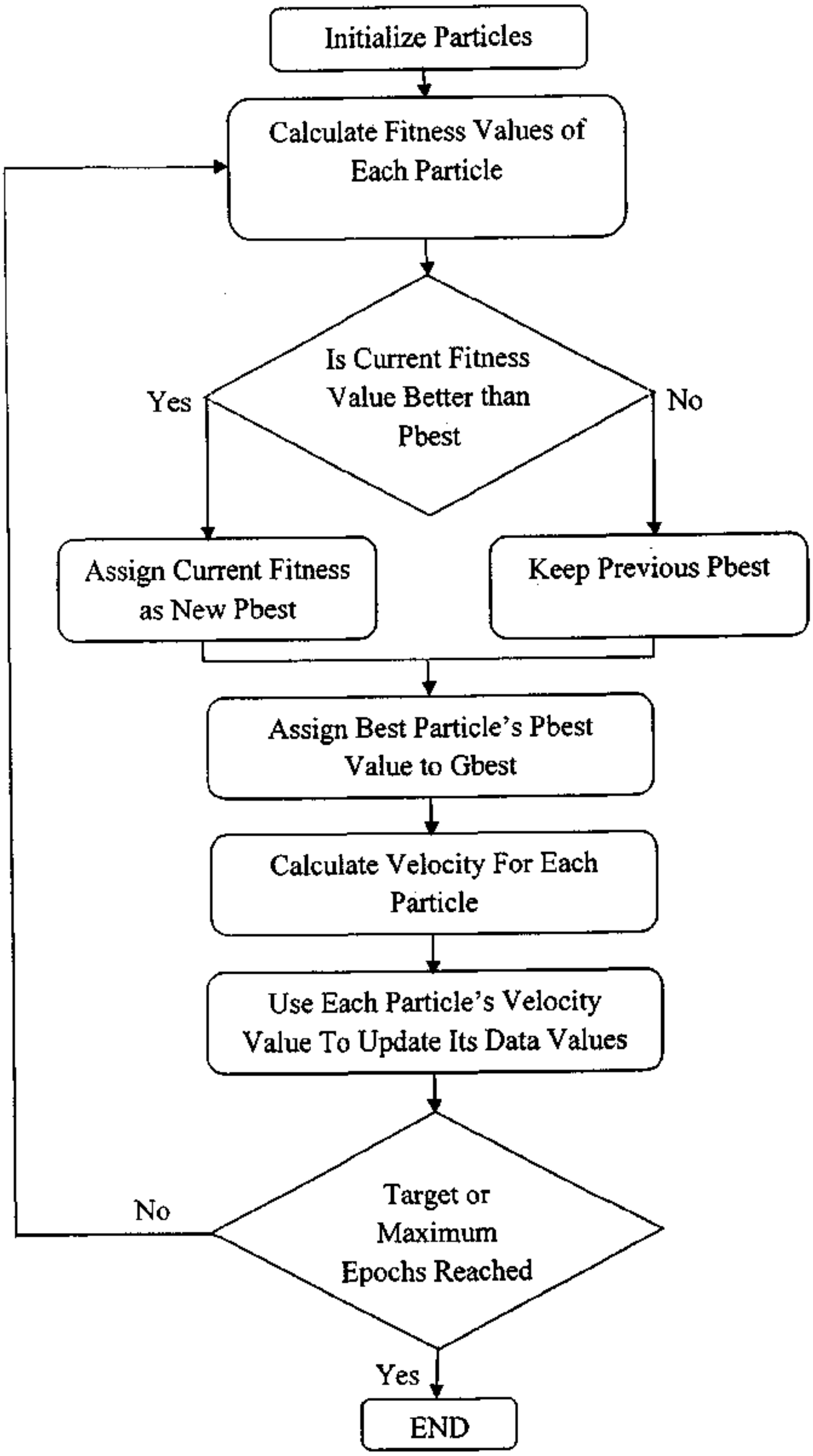
Initialize Particles
Calculate Fitness Values of Each Particle
Is Current Fitness Value Better than Pbest
Yes
No
Assign Current Fitness as New Pbest
Keep Previous Pbest
Assign Best Particle's Pbest Value to Gbest
Calculate Velocity For Each Particle
Use Each Particle's Velocity Value To Update Its Data Values
Target or Maximum Epochs Reached
No
Yes
END

Fig. 2.2 Flow Chart of PSO

PSO is a population based search method in which individuals, called as particles are grouped into a swarm. Each particle in the swarm is a candidate solution. Starting with a randomly generated population the particles move around in the search space for the best solution. All particles exhibit velocities that direct flying of the particles and fitness values which are computed by the problem exlclusive fitness function. Each particle in search space keeps track of its best solution obtained called Pbest, and the best value achieved by any paricle called Gbest, and adjusts its travelling speed dynamically according to its personal flying experience as well as flying of colleague particles. It is expected that the particles will move towards a optimum global solution area. At each iteration velocity and the position of each particle are updated accordingly. We have employed following relations for updating the velocity and position of the particle [59].

$$v_i^{j+1} = \varphi^j v_i^j + c_1[\gamma_{1,i}(Pbest_i - x_i^j)] + c_2[\gamma_{2,i}(Gbest - x_i^j)] \quad (2.4)$$

$$x_i^{j+1} = x_i^j + x_i^{j+1} \quad (2.5)$$

where c_1 and c_2 are positve acceleration constants, i= 0, 1, 2,....M, where M is the total number of paticles in the swarm, $\gamma_{1,2}$ are random numbers chosen uniformly in the range [0,1], φ is the linearly decreasing inertia weight, x_i^j, and v_i^j are current position and velocity vectors respectively. The generic PSO is given in Fig. 2.2 in the form of the flow chart.

2.4. MEMETIC ALGORITHMS

Memetic algorithms (MAs) are optimization techniques based on the hybrid approach of global search EAs and local search [60]. MA term was first introduced by Moscato in [61], and was regarded as a population based hybrid genetic algorithm (HGA). MAs are inspired by both Darwinian evolution and the cultural evolution [62]. MAs also referred in more diverse context as hybrid evolutionary algorithms have been proved to be more accurate and computationally efficient than EAs [60]. In recent years a number of hybrid evolutionary techniques have been reported for solving various problems such as generation maintenance scheduling [63], brain computer interface [64], and nonlinear flight control [65]. Further a detail of several hybrid evolutionary schemes and their applications can be found in [60]. The application of hybrid evolutionary algorithm ANNs are also reported in [27], [28] for solving many systems of ODEs effectively.

In this study I have used the hybrid approach of stochastic global search algorithm such as GA with local search algorithms including interior point algorithm (IPA), active set algorithm (ASA), and pattern search (PS). In our approach, GA has been used as global search optimizer which finds global best solutions, and IPA, ASA, and PS are used as local optimizers for the fine tuning and improvement of the solutions. The memetic algorithm approach adopted in this dissertation is shown in Fig 2.3.

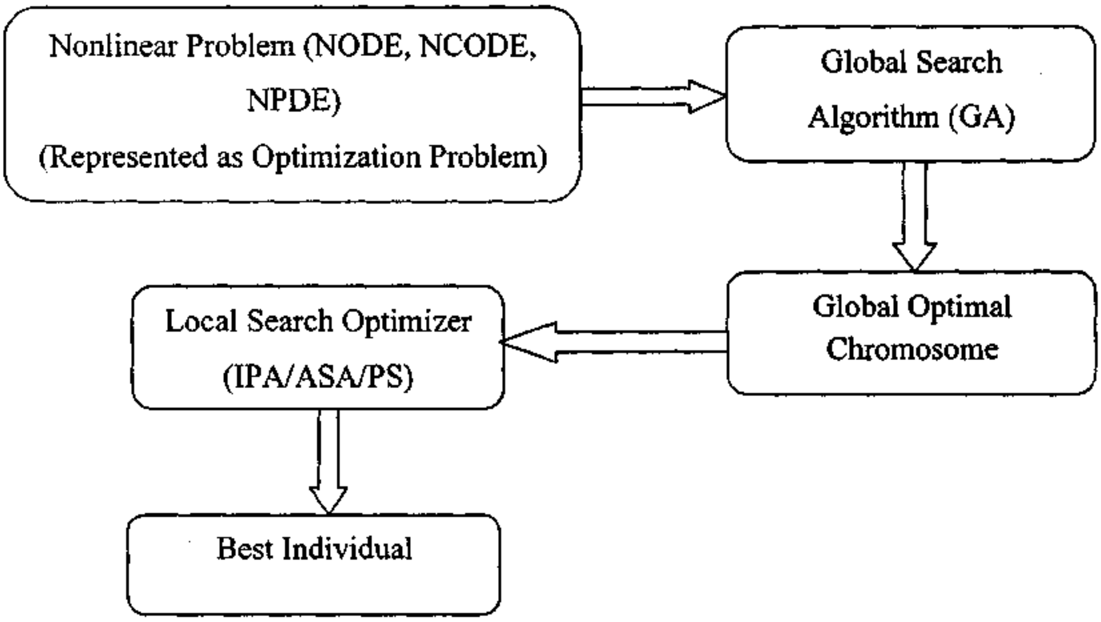
Nonlinear Problem (NODE, NCODE, NPDE)
(Represented as Optimization Problem)
Global Search Algorithm (GA)
Global Optimal Chromosome
Local Search Optimizer (IPA/ASA/PS)
Best Individual

Fig. 2.3 Flow Diagram of Memetic Algorithm

The global best chromosome found by GA for the given problem is fed into the local search optimizers which perform the local search refinement, and consequently improved solutions are obtained. The local search algorithms such as IPA, ASA, and PS used in this dissertation are briefly explained below.

2.5 INTERIOR POINT ALGORITHMS

Interior-point algorithms (IPAs) although were introduced in late 1940's [66], they only became popular since the revolutionary work of Kamarkarin in 1984 [67]. IPA also referred to as barrier method navigates through interior feasible region following a middle path to reach an optimal solution [68]. At each iteration IPA attempts to solve a sequence of approximate minimization problems using either direct step also called Newton step or conjugate gradient (CG) step [69].

The algorithm tries the direct step by default. When the approximate problem is not locally convex near the current iterate then the algorithm attempts CG step. In the direct step the algorithm uses linear approximation to solve the Karush-Kuhn-Tucker (KKT) equations and a CG step is attempted using trust region. At each iteration, the algorithm decreases a merit function that is specific to problem. In case the attempted step does not give any decrease in the merit function, it is rejected and a new step is attempted by the algorithm. The algorithm iteratively tracks the middle path of the feasible region while decreasing the barrier parameter to ultimately reach an optimal point [70].

2.6 ACTIVE SET ALGORITHM

Active set algorithm (ASA) belongs to the larger class of quadratic programming (QP) [71]. ASA is an iterative method that creates a sequence of approximate solutions to the given problem. The objective of the algorithm is to maintain and update a prediction of the optimal sets of active and inactive constraints. The prediction usually turns wrong therefore these methods contain procedures for testing and altering the current prediction [72]. At each iteration the algorithm attempts to solve KKT equations defined by the active constraints.

The classical active set methods usually work in two phases such as feasibility and optimality. In the feasibility phase the algorithm attempts to compute the feasible point for the constraints, while ignoring the objective. In the optimality phase the algorithm maintains feasibility and attempts to compute an optimal point by minimizing the objective [71], [72].

2.7 PATTERN SEARCH

The Pattern Search (PS) first introduced by Hooke and Jeevs [73] belongs to the direct search methods (DSM). DSMs does not involve the gradient of the optimization problems, hence PS can be applied to optimization problems that are non-continuous, non-differentiable, and multimodal [74]. The PS algorithm proceeds by computing a set of points that approach to the optimal point. The algorithm begins by searching a set of points called mesh, around the given point that is computed in the previous step of the algorithm. The mesh is formed by adding the current point to a scalar multiple of vectors called a pattern. If PS finds that a point in the mesh improves the objective function at the current point, the new point becomes the current point in the next iteration. PS is simple, easy, and computationally efficient, and also possesses adaptability for local search refinement [75].

2.8 SUMMARY

This chapter presents the review of evolutionary computing techniques and their applications to many problems in engineering. It also presents an overview of hybrid genetic algorithms known as memetic algorithms.

CHAPTER 3

APPLICATION OF EVOLUTIONARY COMPUTING TECHNIQUE FOR SOLVING NONLINEAR ORDINARY DIFFERENTIAL EQUATIONs

This chapter provides the detail of the suggested heuristic computation method, combining log sigmoid basis functions and evolutionary algorithm for solving nonlinear ODEs. The approximate solution of NODE is deduced as a linear combination of log sigmoid basis functions with some unknown parameters. A fitness function is used to convert the NODE into an equivalent global error minimization problem. One of the popular EAs such as GA and memetic algorithms combining GA with IPA, ASA, and PS are used to solve the minimization problem and to obtain the unknown parameters. The numerical applications of the suggested method are extensively studied for solving several nonlinear problems. The material provided in the following sections is mostly from the published work [76], [77], [78], [79], [80].

3.1 INTRODUCTION

Nonlinear problems appearing in many physical phenomena, engineering and scientific applications are modeled with nonlinear ordinary differential equations (NODEs). Mostly these NODEs are formulated as initial and/or boundary value problems. Some of the useful applications of NODEs include the modeling of gas dynamics, beam deflection, thermodynamics, optimization theory, atomic physics, nuclear, chemical reactions, and fluid dynamics [81], [82]. Solving such NODEs is vital to get the insight of the systems behavior. Since many NODEs either do not have an exact solution or obtaining the same is difficult analytically, therefore these problems are tackled using various approximate analytical and numerical techniques. There are many traditional methods like adomian decomposition method (ADM), variational iteration method (VIM), and homotopy perturbation method (HPM) applied for the solutions of NODEs [83]. These methods have their own strengths as well as some limitations that are addressed in [82].

Recently, there has been a growing interest in using stochastic solvers based on evolutionary computation as an alternative to the traditional methods for solving NODEs. The efficiency of these stochastic solvers has been demonstrated by many authors, for example Blasius equation [84], MHD Jeffery-Hamel problem [85], Emden-Fowler equation [86], and van der pol oscillator equation [87] are among several nonlinear problems that have been successfully solved using these techniques. The main advantage of these techniques is that they can provide the numerical solution of NODE on continuous points as compared to the traditional methods which give the solution only at pre-defined grid points. Moreover the complexity of these techniques does not grow with the increase in sampling points, which is one of the main drawbacks of most traditional methods [49].

Although a good number of NODEs has been solved using the traditional as well as stochastic methods, still new methods are sought to handle these and many other such problems. The strengths of stochastic solvers need to be explored on NODEs of physical interest in engineering and science.

3.2. METHODOLOGY FOR SOLVING NODEs

In this section, the methodology for solving NODEs is presented. The method is basically heuristic in nature which combines log sigmoid basis functions and evolutionary algorithms (EAs). An approximate mathematical model employing the linear combinations of log sigmoid basis functions with unknown parameters is deduced. The given NODE is converted into an equivalent optimization problem. EA is used to solve the optimization problem. The method is explained below.

Consider a general nth order ODE given in the following form.

$$\frac{d^n y}{dx^n} = g(x, y, y', y'', y''', \dots\dots\dots, y^{n-1}), \qquad 0 \le x \le T \tag{3.1}$$

subject to the following initial and boundary conditions respectively

$$\frac{d^k y}{dx^k} y(0) = \alpha_k, \qquad k = 0,1,2,3, \dots\dots, n-1 \tag{3.2}$$

$$\frac{d^k y}{dx^k} y(T) = \beta_k, \qquad k = 0,1,2,3, \dots\dots, n-1 \tag{3.3}$$

where g represents the nonlinear function, prime denotes the derivation with respect to *x*, T is the upper bound of the solution span, α_k and β_k are real constants denoting the initial and boundary conditions respectively.

To solve (3.1), we assume that the approximate solution $y(x)$ and its n derivatives $y'(x), y''(x), y'''(x), \dots\dots\dots, and\ y^n(x)$ are a linear combination of basis functions, which can be expressed as follows.

$$y(x) = \sum_{i=1}^{m} a_i \, \varphi(b_i x + c_i) \tag{3.4}$$

$$y'(x) = \sum_{i=1}^{m} a_i \gamma_i \, \varphi'(b_i x + c_i) \qquad (3.5)$$

$$y''(x) = \sum_{i=1}^{m} a_i {\gamma_i}^2 \, \varphi''(b_i x + c_i) \qquad (3.6)$$

$$\vdots$$

$$\vdots$$

$$y^n(x) = \sum_{i=1}^{m} a_i {\gamma_i}^n \, \varphi^n(b_i x + c_i) \qquad (3.7)$$

where a_i , b_i , and c_i are real valued unknown parameters to be determined, m is the number of basis functions, and $\varphi(x)$ is assumed to be the log sigmoid function which is given by

$$\varphi(x) = \frac{1}{1+e^{-x}} \qquad (3.8)$$

The derivates of $y(x)$ given by (3.5) – (3.7) can be simply obtained using the basic calculus or any available mathematical tool. Besides, in this chapter I have solved NODEs of second order only however this methodology can be applied to higher order NODEs as well. Hence in view of the second order NODE as a special case of (3.1), first two derivates of (3.4) are given by (3.9) and (3.10) respectively as follows.

$$y'(x) = \sum_{i=1}^{m} a_i b_i \frac{e^{-(b_i x + c_i)}}{(1+e^{-(b_i x + c_i)})^2} \qquad (3.9)$$

$$y''(x) = \sum_{i=1}^{m} a_i {b_i}^2 \left(\frac{2e^{-2(b_i x + c_i)}}{(1+e^{-(b_i x + c_i)})^3} - \frac{e^{-(b_i x + c_i)}}{(1+e^{-(b_i x + c_i)})^2} \right) \qquad (3.10)$$

The objective is to find the values of unknown parameters $(a_i, b_i, \ and\ c_i)$ in (3.4), which consequently yields the approximate numerical solution $y(x)$ of the given problem. To determine the values of these unknown parameters $(a_i, b_i, \ and\ c_i)$, the given NODE along with its initial and/or boundary conditions is converted into an equivalent optimization problem using a fitness function defined below.

3.2. 1 FITNESS FUNCTION

The fitness function (FF) denoted as (ε_j) basically represents the global error associated with the given NODE along with its initial and/or boundary conditions to be solved using the approximate model defined above. FF consists of the sum of two parts, first part represents the mean of sum of square errors associated with the given NODE denoted by (ε_1) , the second part represents the mean of sum of square errors linked with the given initial and/or boundary conditions denoted by (ε_2). Assuming a second order NODE (ε_j) is developed as follows.

$$\varepsilon_1 = \frac{1}{N}\sum_{i=1}^{N}\left(y''(x_i) - g(x_i, y(x_i), y'(x_i))\right)^2 \qquad (3.11)$$

$$\varepsilon_2 = \frac{1}{n}\sum_{k=0}^{n-1}\left(\frac{d^k y}{dx^k}y(0) - a_k\right)^2 + \frac{1}{n}\sum_{k=0}^{n-1}\left(\frac{d^k y}{dx^k}y(T) - b_k\right)^2 \qquad (3.12)$$

where $y(x)$, $y'(x)$, and , $y''(x)$ are given by (3.4), (3.9), and (3.10) respectively, N is the total number of steps on the interval [0, T]. The FF is accordingly written as

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \quad j = 1,2,3 \ldots \qquad (3.13)$$

where j is the generation number/iteration count of the algorithm.

The fitness function (ε_j) represents the global error minimization problem. As it is evident that FF contains unknown parameters $(a_i, b_i, \ and\ c_i)$, therefore it solely depends on the values of these unknown parameters. It is also quite obvious that smaller the ε_j the better the approximate solution. The error minimization problem given by (3.13) is solved using the evolutionary algorithm (EA) to find the values of the unknown parameters $(a_i\,, b_i\,,\ and\, c_i)$ that correspond to the best possible minimum ε_j. Consequently the approximate numerical solution $y(x)$ of the given NODE is straightforward obtained by using the values of unknown parameters in (3.4).

3.2.2 HYBRID GENETIC ALGORITHMS

In this section, evolutionary algorithm used throughout the chapter for solving the minimization problem given by the fitness function (3.13) is introduced. The GA, IPA, ASA, and PS, and three hybrid schemes combining GA with PS, IPA, and ASA have been employed for solving the fitness function and to obtain the unknown parameters $(a_i\,, b_i\,,\ and\ c_i)$. The three hybrid schemes used are referred to as GA-IPA, GA-ASA, and GA-PS in the rest of the chapter. The GA has been used as global optimizer while IPA, ASA, and PS have been utilized for local search refinement. The procedural steps of the hybrid schemes are given in algorithm 3.1.

Algorithm 3.1: Hybrid Genetic Algorithm (HGA)

Step 1: (Population Initialization)

A population of N chromosomes or individuals is generated using random number generator. Each population consists of M number of genes. The number of genes is equal to the number of unknown adjustable parameters.

Step 2: (Fitness Evaluation)

Fitness of each chromosome is computed in the current population using the problem exclusive fitness function (FF). Rank the individuals according their fitness values.

Step 3: (Stoppage Criteria)

The algorithm stops if the maximum number of generations/iterations has exceeded or a predefined fitness value is achieved. If the stoppage criterion is satisfied then go to step 6 for local search refinement, else continue and repeat steps 2 to 5.

Step 4: (Selection and Reproduction)

A new generation is populated using the crossover operation. Parents are selected on the basis of their fitness which produces offspring (children) to act as parents for the next generation.

Step 5: Mutation

This operation is optional and it is carried if there is no improvement in the fitness in a generation. Mutation introduces intermittent changes in the genes to preserve the genetic diversity.

Step 6: (Local Search Fine Tuning)

The optimal chromosome achieved by the GA is fed to the IPA, ASA, and PS as a starting point for fine tuning and improvement.

3.3. NUMERICAL APPLICATIONS

In this section, the methodology described above is applied to many nonlinear problems including the Bratu problem, Troesch's problem, Duffiing van der pol oscillator problem, and nonlinear singular boundary value problems in physiology, to assess and test its performance. The nonlinear problems solved in this chapter using the proposed methodology have been selected due to their physical importance in diverse applications of engineering.

In order to demonstrate the accuracy, efficiency, and viability of the presented method, comparisons of the numerical solutions are made with the exact solutions and the solutions obtained by some traditional methods.

For implementation Matlab 7.6 and its built in optimization tool has been utilized throughout.

3.3.1 BRATU PROBLEM

Example1. We consider the classical Bratu problem in one-dimensional planner coordinates represented by the boundary value problem of the following form [88], [89], [90], [91], [92], [93], [94], [95].

$$y''(x) + \lambda e^{y(x)} = 0, \;\; x \in [0,1] \qquad (3.14)$$

$$y(0) = 0, and\, y(1) = 0 \qquad (3.15)$$

The classical Bratu problem has wide spread applications in engineering and science including the model of fuel ignition, chemical reaction theory, radiative heat transfer, Chandrasekhar model of the expansion of the universe, and nanotechnology [88], [89], [90], [91], [92], [93], [94],[95].

The Bratu problem has received much attention due to its diverse applications and many methods have been utilized for the solution of the standard Bratu and Bratu-type problems [88], [89], [90], [91], [92], [93], [94], [95]. The methods include DM by Deeba and Khuri [88], LTDM by Khuri [89], ADM by Wazwaz [5], RADM by Vahidi, and Hasanzade [92], non-poly spline by Rashidinia and Jalilian [93], cubic B-spline collocation by Abukhalid [91].

The exact solution of (3.14) for $\lambda > 0$ is given in [90], [91], [92], [93], [94], [95] and is given by the equation below

$$y_{exact}(x) = -2ln\left[\frac{cosh\left(\frac{\theta}{2}\left(x-\frac{1}{2}\right)\right)}{cosh\left(\frac{\theta}{4}\right)}\right] \qquad (3.16)$$

where θ satisfies

$$\theta = \sqrt{2\lambda}\cosh\left(\frac{\theta}{4}\right) \qquad (3.17)$$

The approximate numerical solution $y(x)$ of the Bratu problem (3.14) – (3.15) is obtained using the suggested method in the interval [0, 1] with a step of 0.1, for two different values of $\lambda = 1, 2$, for a direct comparison with some other methods.

To apply the proposed method, the given problem (3.14) – (3.15) is converted into an equivalent global error minimization problem by formulating its fitness function for each case $\lambda = 1\ and\ \lambda = 2$. The number of basis functions is taken equal to 10. The fitness function ε_j for $\lambda = 1$ is developed as follows.

$$\varepsilon_1 = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) + 1\left(e^{y(x_i)}\right)\right)^2 \qquad (3.18)$$

$$\varepsilon_2 = \frac{1}{2}\left(\left(y(0)\right)^2 + \left(y(1)\right)^2\right) \tag{3.19}$$

Consequently the fitness function ε_j is given as follows

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) + 1\left(e^{y(x_i)}\right)\right)^2 + \frac{1}{2}\left(\left(y(0)\right)^2 + \left(y(1)\right)^2\right) \tag{3.20}$$

Similarly the fitness function for $\lambda = 2$ is formulated which is given by

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) + 2\left(e^{y(x_i)}\right)\right)^2 + \frac{1}{2}\left(\left(y(0)\right)^2 + \left(y(1)\right)^2\right) \tag{3.21}$$

where $y(x)$ and $y''(x)$ are given by (3.4) and (3.10) respectively.

The fitness functions given by (3.20) and (3.21) are minimized by applying GA, IPA, ASA, and PS, and three hybrid schemes GA-IPA, GA-ASA, and GA-PS for achieving the values of unknown adjustable parameters $(a_i\,, b_i\,,\ and\ c_i)$.

The parameter settings and values used for the implementation of the algorithms are given in Table 3.1 for GA and IPA and in Table 3.2 for PS and ASA respectively.

Since we have taken the number of basis functions equal to 10, therefore the size of chromosome i.e. the number of unknown adjustable parameters $(a_1\,, a_2, \ldots, a_{10};\ b_1, b_2, \ldots, b_{10};\ c_1, c_2, \ldots, c_{10}\,)$ are chosen equal to 30. The values of these unknown adjustable parameters are restricted between -20 and + 20. This was observed by several simulations that by bounding these unknown adjustable parameters to the specified interval we get good results.

Table 3.1 Parameter values and settings of GA and IPA for Bratu problem

GA			IPA		
Parameters Name	Settings/Value		Parameters Name	Settings/Value	
	Example 1	Example 2		Example 1	Example 2
Population size	240	240	Start point	Random/best chromosome from GA	Random/best chromosome from GA
Chromosome size	30	30	Maximum function evaluations	48000	200000
Selection function	Stochastic uniform	Stochastic uniform	Derivative type	Forward differences	Central differences
Mutation function	Adaptive feasible	Adaptive feasible	Hessian	BFGS	BFGS
Crossover function	Heuristic	Heuristic	Subproblem algorithm	ldl factorization	ldl factorization
Crossover fraction	0.8	0.8	X tolerance	1e-10	1e-10
No. of generations	1500	2000	Maximum iterations	1000	1000
Function tolerance	1e-22	1e-18	Function tolerance	1e-22	1e-18
Bounds	-20, +20	-20, +20	Bounds	-20, +20	-20, +20

Table 3.2 Parameter values and settings of PS and ASA for Bratu problem

PS			ASA		
Parameters Name	Settings/Value		Parameters Name	Settings/Value	
	Example 1	Example 2		Example 1	Example 2
Start point	Optimal chromosome from GA	Optimal chromosome from GA	Start point	Random/best chromosome from GA	Random/best chromosome from GA
Poll method	GPS positive basis 2N	MADS positive basis 2N	Maximum iterations	400	400
Polling order	Random	consecutive	Maximum function evaluations	48000	200000
Maximum iterations	3000	4000	Function tolerance	1e-22	1e-18
Maximum function evaluation	200000	230000	Nonlinear constraint tolerance	1e-10	1e-18
Function tolerance	1e-22	1e-22	SQP constraint tolerance	1e-6	1e-6

The algorithms are executed according to the prescribed settings to achieve the minimum value of fitness function (ε_j). The algorithms run iteratively until the termination criteria of either maximum number of generations exceeds or the desired fitness value is achieved. The best chromosome found by the algorithms is accordingly chosen as the values of unknown parameters. The values of unknown parameters acquired by the algorithms are used in (3.4) to yield the approximate solution $y(x)$.

In Table 3.3 we provide the optimal values of unknown parameters acquired by GA for $\lambda = 1,2$, and in Table 3.4 and Table 3.5 values of unknown parameters acquired by three hybrid schemes GA-IPA, GA-ASA, and GA-PS are given for $\lambda = 1$ and $\lambda = 2$ respectively. Further in Table 3.6 – Table 3.8 the values of unknown parameters acquired by IPA, ASA, and PS are given respectively.

Table 3.3 Optimal values of unknown parameters acquired by GA for Bratu problem (example 1) with $\lambda = 1,2$

	$\lambda = 1$			$\lambda = 2$		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.1873	-0.9927	4.3567	2.0106	1.8214	0.1774
2	0.0532	2.8125	1.3375	-0.2814	-0.1561	0.0798
3	3.3774	0.0092	-2.8024	1.0925	3.6038	2.0671
4	-2.3967	-1.4546	-0.3605	-0.5204	-0.0327	0.8305
5	-1.9747	-0.2414	1.5959	-2.3394	0.7914	-0.3817
6	0.4933	-0.7706	2.8568	0.4159	-1.7345	4.6111
7	-2.7251	1.2393	-1.7020	-0.6076	2.2922	-1.8711
8	-2.0717	1.5492	3.2516	-3.8191	1.6470	-2.8017
9	2.6098	-1.4604	3.9947	-1.6690	-2.2575	0.0809
10	3.0735	2.0118	3.1644	0.3176	-0.6789	-0.1111

Table 3.4 Optimal values of unknown parameters acquired by hybrid schemes for Bratu problem (example 1) with $\lambda = 1$

	GA-PS			GA-ASA			GA-IPA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.2026	-0.9780	4.7183	-1.1873	-0.9927	4.3567	-0.9870	-0.8866	4.0878
2	0.0636	3.1262	1.7586	0.0532	2.8125	1.3375	0.1371	2.5885	1.3474
3	3.3970	0.0286	-2.8143	3.3774	0.0092	-2.8024	3.1203	-0.1468	-2.6795
4	-2.4075	-1.4589	-0.3726	-2.3967	-1.4546	-0.3605	-2.2438	-1.5079	-0.2057
5	-1.9894	-0.2529	1.6472	-1.9747	-0.2414	1.5959	-1.7019	-0.2075	1.4432
6	0.4889	-0.8683	3.0274	0.4933	-0.7706	2.8568	0.4568	-0.7396	2.6768
7	-2.7638	1.2354	-1.6975	-2.7251	1.2393	-1.7020	-2.5884	1.1878	-1.5518
8	-2.0535	1.6769	3.3943	-2.0717	1.5492	3.2516	-1.7665	1.4070	3.1085
9	2.6244	-1.4582	4.0706	2.6098	-1.4604	3.9947	2.3478	-1.4449	3.6874
10	3.0827	2.0025	3.2143	3.0735	2.0118	3.1644	2.6066	2.0180	2.9024

Table 3.5 Optimal values of unknown parameters acquired by hybrid schemes for Bratu problem (example 1) with $\lambda = 2$

	GA-PS			GA-ASA			GA-IPA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	2.0497	2.0166	0.2739	1.9681	1.9209	0.2519	1.9109	1.9301	0.2668
2	-0.2482	-0.8913	0.1337	-0.3238	-0.1363	0.0881	-0.2924	-0.1368	0.0864
3	1.0784	3.5803	2.3166	1.1145	3.4939	2.3152	1.0969	3.4901	2.3596
4	-0.5860	-0.0930	1.0323	-0.5827	0.0042	0.8449	-0.5221	0.0024	0.8409
5	-2.2732	0.7748	-0.1741	-2.3987	0.7396	-0.4401	-2.3208	0.6664	-0.4480
6	0.4298	-2.3960	5.2107	0.4101	-1.8643	4.5666	0.3630	-1.9026	4.5434
7	-0.5210	2.5587	-2.2779	-0.8380	2.3214	-2.1652	-0.8158	2.3580	-2.1672
8	-4.0139	1.6085	-2.7957	-4.0283	1.5534	-3.0031	-4.0585	1.5711	-2.9750
9	-1.7985	-2.0488	-0.0763	-1.7691	-2.0924	-0.0573	-1.7780	-2.0643	-0.0848
10	0.4287	-1.5663	-0.3880	0.2635	-0.6650	-0.1063	0.2335	-0.6549	-0.1033

The approximate solution $y(x)$ of the Bratu problem (3.14) is obtained straightforward by using the values of unknown parameters in (3.4). The solutions obtained by the proposed method with GA and three hybrid schemes GA-IPA, GA-ASA, and GA-PS are presented in Table 3.9 and Table 3.10 for $\lambda = 1\ and\ \lambda = 2$ respectively, also exact solutions are given for the comparison purpose.

Table 3.6 Optimal Values of unknown parameters acquired by IPA for Bratu problem (example 1) with $\lambda = 1, 2$

	$\lambda = 1$			$\lambda = 2$		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	0.3992	-0.0455	-0.9373	-5.1128	1.6163	-2.8336
2	0.8232	-0.1400	-0.8866	1.0195	0.8413	0.7468
3	-1.6445	-1.3328	-0.9705	-3.0910	-1.6747	0.6457
4	-3.4513	1.2628	-2.9330	-1.5674	-1.5443	-1.1531
5	1.6397	0.0244	-0.7520	-0.8372	-0.8444	-0.6081
6	-1.3173	-1.8875	-2.1328	-1.3665	-1.1407	-0.8240
7	0.6709	1.7674	-0.0797	-1.6926	-1.5776	-2.2070
8	0.0044	-0.3470	-0.2056	0.6572	-0.7454	1.6052
9	-0.8562	1.4735	-1.6205	2.5609	-2.1051	1.6029
10	-1.6134	0.0690	-1.4985	0.3224	-0.3243	0.4474

Table 3.7 Optimal Values of unknown parameters acquired by ASA for Bratu problem (example 1) with $\lambda = 1, 2$

	$\lambda = 1$			$\lambda = 2$		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	-0.5927	1.6649	-1.5896	0.2217	-0.3299	-1.6089
2	0.6306	0.7886	-2.1247	-4.9753	-1.6655	-0.1554
3	-1.4929	1.3215	-1.5765	-0.8901	0.2419	1.8635
4	0.3649	0.5369	-1.1720	1.7046	-0.8554	-1.2245
5	3.2966	0.2018	-0.6828	-0.2667	-1.3758	-1.0384
6	0.3009	-0.0477	-1.4860	-3.6200	1.8887	-4.0655
7	3.2056	0.0620	-0.5341	0.4903	-0.5940	-0.4413
8	-4.9089	1.2200	-3.0732	0.6553	-0.1726	-1.5118
9	-0.6137	0.2596	0.1714	-2.2965	-2.5983	-2.9132
10	-5.7031	-0.8584	-0.9471	2.9963	-1.9038	1.8238

To show the accuracy and the efficacy of the proposed method, absolute errors ($|y_{exact}(x) - y(x)|$) have been computed and presented in Table 3.11 and Table 3.12 for $\lambda = 1 \; and \; \lambda = 2$ respectively. For comparisons the absolute errors obtained by the standard methods DM [88], LTDM [89], and B-spline [91] are also provided.

Table 3.8 Optimal Values of unknown parameters acquired by PS for Bratu problem (example 1) with $\lambda = 1, 2$

	$\lambda = 1$			$\lambda = 2$		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	-0.0491	1.3713	1.7478	-0.4768	-4.1539	-2.6383
2	-15.3581	0.3592	-0.7001	1.6841	0.4060	-0.3223
3	-9.3463	-18.2663	-12.0579	0.4716	1.5749	1.1711
4	3.3143	1.8380	1.7670	0.8389	4.8710	4.4206
5	0.9526	-2.4088	-3.0868	-0.1825	-5.5634	0.1554
6	0.8614	-0.3402	1.2269	-4.6172	1.8696	-2.6017
7	3.3925	1.4900	0.0493	-1.2924	-0.0795	0.1325
8	-0.1187	2.8442	-0.0739	-2.2351	-2.0640	-0.2275
9	-0.0284	-0.1801	1.7186	-0.0548	7.4645	-0.4996
10	-0.1181	0.3970	-0.0463	0.5227	0.1003	-0.0645

Table 3.9 Comparison of numerical results for Bratu problem (example 1) with $\lambda = 1$

		Proposed Method $y(x)$			
x	$y_{exact}(x)$	**GA**	**GA-IPA**	**GA-ASA**	**GA-PS**
0.1	0.049847	0.049845	0.049847	0.049847	0.049847
0.2	0.089190	0.089187	0.089190	0.089191	0.089190
0.3	0.117609	0.117606	0.117609	0.117610	0.117609
0.4	0.134790	0.134785	0.134790	0.134791	0.134790
0.5	0.140539	0.140533	0.140539	0.140539	0.140539
0.6	0.134790	0.134782	0.134790	0.134789	0.134790
0.7	0.117609	0.117601	0.117609	0.117608	0.117609
0.8	0.089190	0.089181	0.089190	0.089189	0.089190
0.9	0.049847	0.049838	0.049847	0.049847	0.049847

Comparison of the absolute errors reveals that the proposed method based on the heuristic computational approach yields the results of the Bratu problem (3.14) for two special cases $\lambda = 1$ *and* $\lambda = 2$ with the significantly greater accuracy, with an average absolute error of 5.20E-08 for $\lambda = 1$ and 3.09E-07 for $\lambda = 2$ respectively.

Table 3.10 Comparison of numerical results for Bratu pronlem (example 1) with $\lambda = 2$

		Proposed Method $y(x)$			
x	$y_{exact}(x)$	**GA**	**GA-IPA**	**GA-ASA**	**GA-PS**
0.1	0.114411	0.114442	0.114411	0.114411	0.114430
0.2	0.206419	0.206454	0.206419	0.206419	0.206435
0.3	0.273879	0.273924	0.273880	0.273880	0.273893
0.4	0.315089	0.315146	0.315090	0.315090	0.315102
0.5	0.328952	0.329014	0.328952	0.328953	0.328963
0.6	0.315089	0.315149	0.315089	0.315089	0.315098
0.7	0.273879	0.273937	0.273879	0.273879	0.273886
0.8	0.206419	0.206480	0.206419	0.206419	0.206423
0.9	0.114411	0.114474	0.114411	0.114411	0.114411

Furthermore comparison shows that the absolute errors obtained by standard methods DM, LTDM, and B-spline are quite high compared to the absolute errors obtained by the proposed method, which illustrates the supremacy of the proposed method over the standard methods DM, LTDM, and B-spline.

The improved performance achieved by the hybrid schemes GA-IPA, GA-ASA, and GA-PS are also quite evident from the comparison.

Table 3.11 Comparison of absolute errors for Bratu problem (example 1) with $\lambda = 1$

	Proposed Method				Standard Methods		
x	GA	GA-IPA	GA-ASA	GA-PS	DM	B-spline	LTDM
0.1	1.62E-06	1.11E-07	3.59E-07	1.57E-08	2.68E-03	2.98E-06	6.25E-07
0.2	2.62E-06	2.35E-08	6.27E-07	6.25E-08	2.02E-03	5.46E-06	4.36E-07
0.3	3.51E-06	1.55E-07	8.71E-07	2.65E-08	1.52E-04	7.33E-06	2.26E-07
0.4	4.86E-06	1.71E-07	5.20E-07	7.23E-08	2.20E-03	8.50E-06	4.76E-07
0.5	6.53E-06	6.82E-08	2.68E-07	6.72E-08	3.01E-03	8.89E-06	8.06E-08
0.6	7.91E-06	3.34E-07	8.81E-07	3.75E-08	2.20E-03	8.50E-06	8.76E-07
0.7	8.58E-06	3.01E-07	8.84E-07	9.33E-08	1.52E-04	7.33E-06	1.01E-06
0.8	8.72E-06	6.46E-08	4.54E-07	1.30E-08	2.02E-03	5.46E-06	3.14E-07
0.9	9.04E-06	3.03E-07	2.83E-07	8.02E-08	2.28E-03	2.98E-06	2.18E-07

Table 3.12 Comparison of absolute errors for Bratu problem (example 1) with $\lambda = 2$

	Proposed Method				Standard Methods	
x	GA	GA-IPA	GA-ASA	GA-PS	DM	LTDM
0.1	3.15E-05	2.98E-07	5.19E-07	1.89E-05	1.52E-02	2.13E-03
0.2	3.51E-05	2.53E-07	3.45E-07	1.59E-05	1.47E-02	4.21E-03
0.3	4.48E-05	2.18E-07	2.37E-07	1.36E-05	5.89E-03	6.19E-03
0.4	5.68E-05	4.03E-07	7.12E-07	1.23E-05	3.25E-03	8.00E-03
0.5	6.18E-05	1.34E-07	8.07E-08	1.09E-05	6.98E-03	9.60E-03
0.6	5.95E-05	3.43E-07	4.91E-07	9.12E-06	3.25E-03	1.09E-02
0.7	5.79E-05	3.11E-08	2.32E-07	6.70E-06	5.89E-03	1.19E-02
0.8	6.14E-05	1.51E-07	1.19E-07	3.68E-06	1.47E-02	1.24E-02
0.9	6.34E-05	1.47E-07	4.95E-08	6.82E-07	1.52E-02	1.09E-02

Example 2: Consider an initial value problem of the Bratu-type model given by the following equation [90], [91], [92].

$$y''(x) - 2e^{y(x)} = 0, \;\; 0 \leq x \leq 1 \qquad (3.22)$$
$$y(0) = 0, and\, y'(0) = 0$$

The exact solution of (3.23) is given by the following equation [90], [91].

$$y_{exact}(x) = -2\ln\big(\cos(\mathrm{x})\big) \qquad (3.23)$$

The approximate solution $y(x)$ of (3.22) is obtained in the domain [0, 1] using the proposed method by formulating the fitness function as follows.

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) - 2\big(e^{y(x_i)}\big)\right)^2 + \frac{1}{2}\left(\big(y(0)\big)^2 + \big(y'(0)\big)^2\right) \qquad (3.24)$$

The global error minimization problem given by (3.24) is solved using the HGAs to obtain the optimal values of the unknown parameters.

The number of basis functions is taken same as for the example 1. The values and settings of parameter used for the implementation of the algorithms for this example are given in Table 3.1 for GA and IPA and in Table 3.2 for PS and ASA respectively.

The optimal values of unknown parameters acquired by three hybrid schemes are provided in Table 3.13, while the values of parameters acquired by GA, IPA and ASA are given in Table 3.14 and Table 3.15 respectively.

Table 3.13 Optimal values of unknown parameters acquired by hybrid schemes for Bratu problem (example 2)

	GA-PS			GA-ASA			GA-IPA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.574	-4.3007	-5.3328	-0.7058	-2.9094	-6.5500	-0.3453	-1.5097	-2.5190
2	5.0112	6.1085	-9.9512	5.6296	6.1657	-10.000	5.9163	5.1799	-8.0063
3	3.2318	-2.0084	-1.787	3.5729	-1.6589	-1.3569	1.5015	-2.7157	-2.4613
4	9.9683	1.5681	-3.2017	9.7022	0.9337	-2.0803	1.6462	0.4369	-0.6561
5	2.2362	-3.2781	-2.8949	2.9962	-4.6725	-5.6242	1.5190	-2.4307	-1.8001
6	5.0632	2.0861	-6.0192	5.0085	1.3898	-6.2992	2.1032	0.3709	-2.5095
7	-5.178	-2.058	-2.8193	-4.8552	-0.5026	-3.1264	-2.0347	-0.6000	-1.5870
8	-7.2745	0.6289	-3.9587	-7.3139	0.3284	-3.5974	-1.9510	-0.8493	-2.6926
9	-0.5584	-3.154	3.9479	-1.4498	-3.3044	4.2272	-0.6270	-4.6463	4.6490
10	0.1407	3.2648	-9.9211	0.1360	3.2655	-9.9216	0.6655	3.3812	-2.3110

Table 3.14 Optimal values of unknown parameters acquired by GA for Bratu problem (example 2)

i	a_i	b_i	c_i
1	-1.5776	-4.3045	-5.3297
2	5.0105	6.1082	-9.9508
3	3.2303	-2.0088	-1.7872
4	9.9685	1.5680	-3.2014
5	2.2362	-3.2779	-2.8930
6	5.0646	2.0867	-6.0184
7	-5.1779	-2.0583	-2.8192
8	-7.2720	0.6292	-3.9584
9	-0.5584	-3.1541	3.9491
10	0.1439	3.2662	-9.9204

Using the values of unknown parameters achieved by the algorithms, we can find the approximate solution at any point in the solution domain of x.

Table 3.15 Optimal values of unknown parameters acquired by IPA, ASA, and PS for Bratu problem (example 2)

	IPA			ASA			PS		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	1.4581	0.5270	0.4738	-0.3680	-1.1426	0.0403	-2.6757	-0.8110	1.1234
2	0.9208	3.1691	-2.2430	-0.7143	-0.7902	-1.1973	-0.5042	-4.1937	3.2996
3	0.6426	1.1818	0.7531	-2.1814	-3.0543	3.8211	2.2801	4.7116	-6.8452
4	0.5124	0.3398	-0.6950	2.1474	0.5028	2.8979	3.6878	2.6565	-5.1608
5	0.4135	0.1116	0.1689	3.2240	0.6879	0.4128	-0.2803	5.6840	0.6241
6	5.9228	5.2682	-8.1379	-1.4715	3.2768	3.7599	10.0000	-10.0000	-7.3151
7	-1.5984	3.0668	2.0979	1.9487	-2.1496	-0.4357	0.8328	6.5450	8.1848
8	0.2180	-1.2097	-1.0061	10.0000	5.5137	-9.5235	3.2812	0.1775	7.3812
9	1.2828	-0.8675	-1.5663	3.9595	-2.5669	-3.3636	3.0230	4.3748	-7.5306
10	-0.6828	-4.5946	4.6801	-1.5912	-1.3059	0.3872	-1.7997	0.4498	1.4555

The approximate solutions obtained using GA and hybrid schemes for different values of x are given in Table 3.16. Further Table 3.17 shows the comparison of absolute errors ($|y_{exact}(x) - y(x)|$) between the proposed method and the standard methods ADM [92] and RADM [92].

It is observed from the comparisons that the proposed method provides satisfactory results of the Bratu-type problem (3.22) which are in a good agreement with the exact solution. The comparison further shows that the absolute errors obtained from the proposed method based on the hybrid approaches are comparable with those ADM and RADM. However, it can be seen from Table 3.17 that ADM and RADM give much smaller absolute error in the vicinity of x, but as x increases towards 1 absolute error also rises drastically. If we compute the average absolute errors from Table 3.17 by considering the range of x in two partitions such as $x \in [0.1,0.5]$ and $x \in [0.5,1.0]$, Our method gives average absolute errors of 3.96E-06 and 1.74E-05, while ADM gives 1.04E-06, and 1.91E-03, and RADM gives 2.35E-07 and 1.93E-04 average absolute errors respectively for the mentioned intervals of x. Also the average absolute errors in the interval $x \in [0.1,1.0]$ obtained by the proposed method, ADM, and RADM are 1.07E-05, 9.54E-04, and 9.67E-05 respectively. This proves the effectiveness and reliability of the proposed method and its accuracy for large values of x. Furthermore the effectiveness of the hybrid schemes is also evident in this example.

Table 3.16 Comparison of numerical results for Bratu problem (example 2)

		Proposed Method $y(x)$			
x	$y_{exact}(x)$	**GA**	**GA-IPA**	**GA-ASA**	**GA-PS**
0.0	0.000000	-0.000127	0.000007	0.000004	-0.000073
0.1	0.010017	0.009903	0.010022	0.010012	0.009947
0.2	0.040270	0.040198	0.040275	0.040255	0.040232
0.3	0.091383	0.091355	0.091382	0.091363	0.091377
0.4	0.164458	0.164447	0.164453	0.164429	0.164459
0.5	0.261168	0.261160	0.261165	0.261124	0.261162
0.6	0.383930	0.383936	0.383922	0.383870	0.383929
0.7	0.536172	0.536212	0.536154	0.536102	0.536199
0.8	0.722781	0.722853	0.722765	0.722703	0.722834
0.9	0.950885	0.950962	0.950859	0.950782	0.950937
1.0	1.231253	1.231367	1.231234	1.231133	1.231336

Table 3.17 Comparison of absolute errors for Bratu problem (example 2)

	Proposed Method				Other Methods	
x	GA	GA-IPA	GA-ASA	GA-PS	ADM	LADM
0.1	1.14E-04	5.11E-06	5.05E-06	7.02E-05	4.39E-13	9.30E-14
0.2	7.15E-05	4.99E-06	1.42E-05	3.79E-05	4.54E-10	9.72E-11
0.3	2.87E-05	1.06E-06	2.07E-05	5.98E-06	2.66E-08	5.78E-09
0.4	1.10E-05	4.84E-06	2.92E-05	9.75E-07	4.85E-07	1.07E-07
0.5	8.82E-06	3.49E-06	4.41E-05	6.69E-06	4.67E-06	1.06E-06
0.6	5.21E-06	8.32E-06	6.01E-05	1.08E-06	3.01E-05	7.07E-06
0.7	4.07E-05	1.73E-05	6.91E-05	2.75E-05	1.48E-04	3.62E-05
0.8	7.15E-05	1.66E-05	7.80E-05	5.27E-05	6.00E-04	1.54E-04
0.9	7.68E-05	2.56E-05	1.03E-04	5.24E-05	2.11E-03	5.74E-04
1.0	1.14E-04	1.92E-05	1.20E-04	8.33E-05	6.65E-03	1.95E-04

3.3.2 TROESCH'S PROBLEM

We now investigate the numerical solution of the Troesch's problem using the proposed method. Troesch's problem is a boundary value problem which appears in the investigation of the confinement of a plasma column by radiation pressure, theory of gas porous electrodes, and applied physics [96], [97], [98], [99], [100], [101], [102], [103], [104]. This problem was formulated and solved by Weibel [105]. Troesch's obtained the numerical solution of this problem by the shooting method [106].

The governing equation of the Troesch's problem is given by [96], [97], [98], [99], [100], [101], [102], [103].

$$y''(x) = \sigma \sinh(\sigma\, y(x)) \quad 0 \le x \le 1 \qquad (3.25)$$

with the boundary conditions

$$y(0) = 0, and\; y(1) = 1 \qquad (3.26)$$

where σ is a positive constant.

The closed form solution of (3.25) is given as follows [96], [97], [98], [99].

$$y_{exact}(x) = \frac{2}{\sigma} sinh^{-1}\left\{\frac{y'(0)}{2}\ sc\left(\sigma x|1 - \frac{1}{4}(y'(0))^2\right)\right\} \qquad (3.27)$$

The approximate numerical solution $y(x)$ of the Troesch's problem is obtained in the domain $x \in [0,1]$ using the proposed method for three special values of the constant σ for a direct comparison with some standard methods.

Case 1: $\sigma = 0.5$

To apply the proposed method, fitness function (FF) is developed as given below.

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) - 0.5sinh\big(0.5y(x_i)\big)\right)^2 + \frac{1}{2}\left(\big(y(0)\big)^2 + (y(1) - 1)^2\right) \quad (3.28)$$

The number of basis functions m has been taken 10. The FF given by (3.28) is minimized by applying heuristic optimization algorithms GA, PS, IPA, GA-PS, and GA-IPA for obtaining the unknown parameters.

The parameter values and settings for the execution of the optimization algorithms are given in Table 3.18. Since 10 number of basis functions have been taken therefore the unknown parameters that need to be tailored are 30. The values of these unknown parameters are bounded between real numbers -20 and + 20.

Table 3.18 Parameter values and settings of algorithms for Troesch's problem

GA		PS		IPA	
Parameters	**Value/Settings**	**Parameters**	**Value/Settings**	**Parameters**	**Value/Settings**
Population size	240	Start point	Random/ best values from GA	Start point	Random/ best values from GA
Chromosome size	30	Poll method	GPS positive basis 2N	Derivative type	Central differences
Selection function	Stochastic uniform	Polling order	consecutive	Hessian	BFGS
Mutation function	Adaptive feasible	Search method	Nelder mead	subproblem algorithm	ldl factorization
Crossover function	Heuristic	Maximum function evaluation	150000	Maximum function evaluations	150000
Function tolerance	1e-18	Function tolerance	1e-18	Function tolerance	1e-18
No. of generations	1000	Maximum iterations	3000	Maximum iterations	1000
Bounds	-20, +20	Bounds	-20, +20	Bounds	-20, +20

The best values of unknown parameters$(a_1, a_2, \dots, a_{10};\ b_1, b_2, \dots, b_{10};\ c_1, c_2, \dots, c_{10})$ found by the algorithms GA, GA-IPA, and GA-PS are given in Table 3.19, while Table 3.20 show the values of unknown parameters acquired by IPA and PS respectively.

Table 3.19 Optimal values of unknown parameters acquired by GA and hybrid schemes (for $\sigma = 0.5$)

	GA			GA-IPA			GA-PS		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	1.7119	1.5684	2.0633	2.7198	1.3023	2.7030	1.7159	1.5582	2.0632
2	-4.6814	-0.6693	1.4850	-6.8853	-0.6605	1.9212	-4.6783	-0.6688	1.4864
3	4.6063	-0.5614	7.2784	6.8810	-0.7873	11.0064	4.6092	0.2025	6.6903
4	-0.5755	-1.3349	0.2917	-0.7906	-1.0790	-0.3645	-0.5712	-1.3301	0.2898
5	0.3034	0.0733	-1.5361	0.4717	0.1718	-2.2140	0.2864	0.0082	-1.4850
6	1.2785	-0.1976	-1.5037	1.8254	-0.0013	-2.1874	1.2685	-0.1756	-1.5029
7	-1.8026	-1.5340	4.1413	-2.7284	-1.8211	6.7238	-1.8016	-1.5304	4.1614
8	1.2235	-0.5719	-5.0429	1.7635	-0.8101	-7.4410	0.6954	0.1338	-4.5161
9	-0.7416	-0.9677	3.3371	-1.0727	-1.5059	4.8666	-0.7450	-0.9923	3.3092
10	0.2802	0.1051	1.3109	0.5266	0.2231	1.8975	0.2801	-0.0254	1.2470

The values of the unknown parameters can be used to obtain the approximate solution of the Troesch's problem at any value in $x \in [0,1]$. The numerical solutions obtained by the proposed method at different values of x are presented in Table 3.21. Further in Table 3.22 absolute errors (y_{exact}-$y(x)$) obtained by the proposed method are provided, also absolute errors obtained by the standard methods HPM [96], VIM [98], and ADM [99] are given for the purpose of comparison to show the accuracy of the proposed method.

Table 3.20 Optimal values of unknown parameters acquired by IPA and PS (for $\sigma = 0.5$)

	IPA			PS		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	1.0537	1.3713	1.3713	-1.0290	-2.9378	5.8257
2	-1.2478	0.3592	0.3592	0.2431	1.7351	-0.5637
3	2.1226	-18.2663	-18.2663	-1.2566	-1.5175	2.1363
4	-2.2443	1.8380	1.8380	-0.8462	-1.9903	-0.2576
5	4.5561	-2.4088	-2.4088	15.1573	0.0714	-1.4095
6	2.2651	-0.3402	-0.3402	-19.9617	-3.6835	-8.4486
7	-1.7096	1.4900	1.4900	-1.0211	-0.0309	0.3251
8	-2.9082	2.8442	2.8442	-19.9915	-3.6868	-8.4472
9	-1.9689	-0.1801	-0.1801	0.1737	4.4795	3.1516
10	3.7907	0.3970	0.3970	-0.1161	-3.1513	2.9255

From the comparison it is observed that the absolute errors by the proposed method are quite similar to ADM and HPM, and much smaller than VIM, which confirms that the proposed method is more accurate than VIM.

Table 3.21 Comparison of numerical solution (for $\sigma = 0.5$)

		Proposed Method $y(x)$				
x	$y_{exact}(x)$	**GA**	**PS**	**IPA**	**GA-PS**	**GA-IPA**
0.1	0.095177	0.095953	0.095815	0.095944	0.095944	0.095944
0.2	0.190634	0.192135	0.192018	0.192129	0.192128	0.192129
0.3	0.286653	0.288798	0.288716	0.288794	0.288794	0.288794
0.4	0.383523	0.386188	0.386118	0.386185	0.386184	0.386185
0.5	0.481537	0.484550	0.484463	0.484547	0.484548	0.484547
0.6	0.581002	0.584136	0.584027	0.584133	0.584135	0.584133
0.7	0.682235	0.685202	0.685092	0.685201	0.685203	0.685201
0.8	0.785572	0.788016	0.787931	0.788017	0.788017	0.788017
0.9	0.891367	0.892852	0.892792	0.892854	0.892855	0.892854

Table 3.22 Comparison of absolute errors for (for $\sigma = 0.5$)

	Proposed Method			**Standard Methods**		
x	**GA**	**GA-PS**	**GA-IPA**	**HPM**	**VIM**	**ADM**
0.1	7.76E-04	7.68E-04	7.68E-04	7.71E-04	4.87E-03	7.62E-04
0.2	1.50E-03	1.50E-03	1.50E-03	1.50E-03	9.70E-03	1.48E-03
0.3	2.15E-03	2.14E-03	2.14E-03	2.15E-03	1.45E-02	2.13E-03
0.4	2.67E-03	2.66E-03	2.66E-03	2.67E-03	1.92E-02	2.65E-03
0.5	3.01E-03	3.01E-03	3.01E-03	3.02E-03	2.37E-02	2.99E-03
0.6	3.13E-03	3.13E-03	3.13E-03	3.14E-03	2.81E-02	3.12E-03
0.7	2.97E-03	2.97E-03	2.97E-03	2.98E-03	3.22E-02	2.95E-03
0.8	2.44E-03	2.45E-03	2.45E-03	2.45E-03	3.61E-02	2.43E-03
0.9	1.49E-03	1.49E-03	1.49E-03	1.49E-03	3.96E-02	1.48E-03

Case 2: $\sigma = 1$

The fitness function is formulated as follows.

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) - sinh\big(y(x_i)\big)\right)^2 + \frac{1}{2}\left(\big(y(0)\big)^2 + (y(1) - 1)^2\right) \qquad (3.29)$$

The algorithms GA, PS, IPA, GA-PS, and GA-IPA are used with the same parameter values and settings as far case 1, for solving the FF given by (3.29) and to obtain the unknown parameters.

The best values of unknown parameters$(a_1, a_2, \ldots, a_{10};\ b_1, b_2, \ldots, b_{10};\ c_1, c_2, \ldots, c_{10})$ found by the algorithms GA, GA-IPA, and GA-PS are given in Table 3.23, while Table 3.24 show the values of unknown parameters acquired by IPA and PS respectively.

In Table 3.25 numerical results obtained by the proposed method are given. To prove the accuracy again a comparison of absolute errors is made in Table 3.26 between the proposed method and other methods VIM, HPM, and ADM. The comparison evidently shows the reliability of the proposed methods, as the results are quite comparable to ADM and HPM and relatively good than VIM.

Table 3.23 Optimal values of unknown parameters acquired by GA and hybrid schemes (for $\sigma = 1$)

	GA			GA-IPA			GA-PS		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.9309	-0.7643	8.8285	-0.1373	-0.6229	6.4797	-1.9309	-0.7995	8.8559
2	5.4753	-8.9500	-19.9842	2.6492	-2.7798	-10.1327	-15.3919	-4.2390	-12.5779
3	5.4965	-5.5327	-12.9705	4.2095	-3.6319	-8.0170	19.9965	-8.8608	-12.9705
4	3.0886	-19.1565	-19.8576	2.3400	1.3419	-3.9916	-17.9739	-19.9924	-15.7951
5	12.9644	3.5159	16.0530	4.6751	2.8041	6.6194	12.9644	19.9924	15.6780
6	1.3739	3.5675	-7.2071	7.6424	2.8621	-7.5088	1.3739	3.5675	-7.2071
7	1.4540	-1.4889	-3.9448	1.6334	0.0463	-2.6546	1.4540	-1.4889	-3.9448
8	0.8380	3.7772	-17.6828	0.2192	1.6717	-4.6563	-19.1933	-19.9962	-16.0578
9	-11.0564	-1.0704	3.2561	-4.6509	-1.0621	2.4865	-11.0564	-1.0704	3.2561
10	-13.7740	-1.1367	-3.4685	-8.3942	-1.0788	-3.0064	-13.7740	-1.1367	-3.4685

Table 3.24 Optimal values of unknown parameters acquired by IPA and PS (for $\sigma = 1$)

	IPA			PS		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.2911	-0.2756	-4.6943	-0.5322	0.8219	1.9901
2	-2.4965	-0.9767	-4.2146	-0.0804	-1.0385	-4.0600
3	-3.6982	-1.4786	-4.4117	0.8681	0.0628	-0.0766
4	10.7375	2.8061	-7.6192	1.0898	-2.3449	-1.8634
5	-2.9104	-2.1790	-5.8197	0.2645	-3.8994	-3.3029
6	4.5015	1.2481	-2.7211	4.7255	-1.2716	1.6378
7	-2.0397	-1.5369	-4.3454	-0.1395	-0.1811	0.4929
8	0.9157	0.2675	-0.9205	-0.6078	-3.9700	-2.3049
9	-2.0222	-0.7856	-1.8465	-8.0373	-0.6668	0.1342
10	-1.1094	-0.8660	-1.9246	10.6662	1.3687	-3.3266

Table 3.25 Comparison of numerical solution (for $\sigma = 1$)

		Proposed Method $y(x)$				
x	$y_{exact}(x)$	**GA**	**PS**	**IPA**	**GA-PS**	**GA-IPA**
0.1	0.081797	0.084687	0.084676	0.084662	0.084657	0.084661
0.2	0.164531	0.170193	0.170166	0.170172	0.170168	0.170171
0.3	0.249167	0.257411	0.257402	0.257393	0.257391	0.257392
0.4	0.336732	0.347235	0.347255	0.347222	0.347221	0.347220
0.5	0.428347	0.440611	0.440623	0.440599	0.440600	0.440597
0.6	0.525274	0.538545	0.538491	0.538535	0.538538	0.538533
0.7	0.628971	0.642139	0.641981	0.642129	0.642135	0.642129
0.8	0.741168	0.752615	0.752374	0.752608	0.752613	0.752607
0.9	0.863970	0.871366	0.871120	0.871362	0.871366	0.871361

Case 3: $\sigma = 10$

For $\sigma = 10$ the Troesch's problem is difficult to be solved, as reported in [103]. We simply formulate its fitness function as follows.

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) - 10sinh\left(10y(x_i)\right)\right)^2 + \frac{1}{2}\left(\left(y(0)\right)^2 + (y(1)-1)^2\right) \quad (3.30)$$

Without any change in the algorithm settings, the FF given by (3.30) is solved for achieving the unknown parameters and consequently the approximate solution, $y(x)$.

Table 3.26 Comparison of absolute errors (for $\sigma = 1$)

	Proposed Method			**Standard Methods**		
x	**GA**	**GA-PS**	**GA-IPA**	**HPM**	**VIM**	**ADM**
0.1	2.89E-03	2.86E-03	2.86E-03	1.42E-02	1.84E-02	2.45E-03
0.2	5.66E-03	5.64E-03	5.64E-03	2.76E-02	3.68E-02	4.90E-03
0.3	8.24E-03	8.22E-03	8.22E-03	3.96E-02	5.54E-02	7.25E-03
0.4	1.05E-02	1.05E-02	1.05E-02	4.95E-02	7.41E-02	9.35E-03
0.5	1.23E-02	1.23E-02	1.23E-02	5.62E-02	9.30E-02	1.11E-02
0.6	1.33E-02	1.33E-02	1.33E-02	5.89E-02	1.12E-01	1.21E-02
0.7	1.32E-02	1.32E-02	1.32E-02	5.62E-02	1.31E-01	1.21E-02
0.8	1.15E-02	1.15E-02	1.14E-02	4.69E-02	1.50E-01	1.06E-02
0.9	7.40E-03	7.40E-03	7.39E-03	2.89E-02	1.69E-01	6.94E-03

The best values of unknown parameters $(a_1, a_2, \dots, a_{10};\ b_1, b_2, \dots, b_{10};\ c_1, c_2, \dots, c_{10})$ acquired by the algorithms GA, GA-IPA, and GA-PS are given in Table 3.27 and for IPA and PS in Table 3.28. In Table 3.29 we provide the numerical results obtained by the proposed method, while in Table 3.30 numerical results obtained by classical methods including combined reproducing kernel method and ADM (RKM-ADM), ADM, VIM, and modified HPM (MHPM) given in [103] are provided.

Further in Table 3.31 a comparison of absolute errors between the proposed method and the above mentioned classical methods is carried to demonstrate the efficacy, accuracy, and reliability of the proposed method.

From the comparison the quite good accuracy of the proposed scheme is remarkable. The comparison evidently shows the ascendancy of the proposed method over some of the classical methods like ADM, VIM, and MHPM. As it can be seen from the comparison of Table 3.23 ADM, VIM, and MHPM methods completely fail to solve the Troesch's problem with the constant parameter $\sigma = 10$, while the proposed method has yielded the solution conveniently and accurately which are also fairly competent with RKM-ADM.

Table 3.27 Optimal values of unknown parameters acquired by GA and hybrid schemes (for $\sigma = 10$)

	GA			GA-IPA			GA-PS		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	9.1928	-8.0971	-16.6108	1.3415	0.2033	-1.5843	5.8178	12.7154	-17.1889
2	-2.5445	-7.7444	-17.2812	-1.2057	0.3585	-1.3811	19.1118	5.0369	-13.2187
3	2.4610	-8.0616	12.8335	1.2506	-16.9949	17.0689	2.4610	-8.0616	12.8335
4	2.7639	1.7930	17.4762	1.3900	3.6624	3.3795	2.7639	-0.9570	7.7731
5	9.8742	17.0661	10.8081	1.3582	2.5388	3.1580	9.8742	-11.3402	17.3081
6	-15.2671	-9.2333	-17.2564	0.6712	3.7427	-2.3670	-19.9859	10.7510	-17.2564
7	-12.5161	-10.9707	15.8814	-5.8458	-12.5778	14.1888	-12.5161	-10.9707	15.8814
8	3.7853	5.5099	-10.8241	4.5510	0.2360	-3.7174	5.0665	5.5099	-10.8241
9	5.6594	-9.2024	-16.6433	2.6373	-0.7653	-2.8890	8.6907	-2.7337	-8.5183
10	-2.5833	15.9531	11.6347	1.8428	-1.8201	2.1505	-2.5833	-4.6250	10.6347

Table 3.28 Optimal values of unknown parameters acquired by IPA and PS (for $\sigma = 10$)

	IPA			PS		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	1.6138	4.7208	-2.7690	1.3429	-0.0055	2.5223
2	1.6102	-18.7004	17.5195	0.2089	0.1615	1.2086
3	7.8896	5.6415	-6.3116	0.0831	0.9156	0.4843
4	0.8073	5.4891	-3.5872	0.3747	0.2848	-1.3420
5	-1.8685	-4.2408	3.4157	0.1696	-0.3804	0.8463
6	1.2527	-6.4293	2.5012	0.3241	0.6353	0.3136
7	-0.5993	3.8931	0.2186	-0.1001	0.6786	0.0983
8	-1.4033	8.8531	-5.1692	-1.8718	0.3702	2.4907
9	-5.9635	-0.5063	-1.9422	-0.1235	0.2935	1.3210
10	-2.0724	11.1143	-8.6891	0.0693	-0.4508	0.3006

Table 3.29 Approximate numerical solution by the proposed schemes and comparison with the exact solution (for $\sigma = 10$)

		Proposed method $y(x)$			
x	$y_{exact}(x)$	**GA**	**IPA**	**GA-PS**	**GA-IPA**
0.1	0.0000763	-0.0001750	0.0006507	0.0000959	-0.0022146
0.2	0.0001299	-0.0000617	0.0003901	-0.0001366	-0.0008775
0.3	0.0003589	0.0000972	0.0001604	-0.0001633	-0.0005250
0.4	0.0009779	0.0003808	0.0006313	0.0001016	0.0002556
0.5	0.0026590	0.0009327	0.0013073	0.0008483	0.0018240
0.6	0.0072289	0.0021076	0.0039797	0.0025192	0.0043090
0.7	0.0196640	0.0048757	0.0096452	0.0062620	0.0104308
0.8	0.0537303	0.0120360	0.0306525	0.0154046	0.0300570
0.9	0.1521140	0.0318843	0.0640741	0.0403545	0.0801700

Table 3.30 Numerical solution by classical methods and comparison with the exact solution (for $\sigma = 10$)

x	$y_{exact}(x)$	ADM-RKM	VIM	ADM	MHPM
0.1	0.0000763	0.0000576	0.1186109866	667081.1874	17.61750
0.2	0.0001299	0.0001902	0.4461962517	1333955.1189	33.69333
0.3	0.0003589	0.0005676	3.8003366781	1999860.1189	46.78583
0.4	0.0009779	0.0016654	79.89147273	2661970.7366	55.65333
0.5	0.0026590	0.0048331	1880.3539472	3310585.4201	59.35417
0.6	0.0072289	0.0137488	41642.365193	3914127.8659	57.34667
0.7	0.0196640	0.0374013	878764.64189	4374578.5342	49.58917
0.8	0.0537303	0.0936540	18064027.967	4406724.4178	36.64000
0.9	0.1521140	0.2189270	366613074.02	3290268.6374	19.75750

Table 3.31 Comparison of absolute errors (for $\sigma = 10$)

	Proposed Method			Classical Methods			
x	GA	IPA	GA-PS	ADM-RKM	ADM	MHPM	VIM
0.1	2.51E-04	5.74E-04	1.96E-05	1.87E-05	6.67E+05	1.76E+01	1.19E-01
0.2	1.92E-04	2.60E-04	2.67E-04	6.03E-05	1.33E+06	3.37E+01	4.46E-01
0.3	2.62E-04	1.99E-04	5.22E-04	2.09E-04	2.00E+06	4.68E+01	3.80E+00
0.4	5.97E-04	3.47E-04	8.76E-04	6.88E-04	2.66E+06	5.57E+01	7.99E+01
0.5	1.73E-03	1.35E-03	1.81E-03	2.17E-03	3.31E+06	5.94E+01	1.88E+03
0.6	5.12E-03	3.25E-03	4.71E-03	6.52E-03	3.91E+06	5.73E+01	4.16E+04
0.7	1.48E-02	1.00E-02	1.34E-02	1.77E-02	4.38E+06	4.96E+01	8.79E+05
0.8	4.17E-02	2.31E-02	3.83E-02	3.99E-02	4.41E+06	3.66E+01	1.81E+07
0.9	1.20E-01	8.80E-02	1.12E-01	6.68E-02	3.29E+06	1.96E+01	3.67E+08

3.3.3 DUFFING VAN DER POL OSCILLATOR

Duffing van der pol (DVP) oscillator is one of the most extensively studied dynamical system, which can be used as a model in engineering, electronics, physics, biology, neurology, and many other disciplines [107], [108], [109],. Moreover the chaotic behavior and coupling of the Duffing van der pol oscillator (DVP) makes it useful in applications, such as chaos communication systems, synchronization in communication engineering, image processing, electrical and automation engineering [107], [110].

The DVP oscillator investigated in this dissertation is given by the following second order NODE [111].

$$y''(x) - \mu\left(1 - y^2(x)\right)y' + \alpha x(t) + \beta y^3(x) = g(f, \omega, x) \qquad (3.31)$$

where $g(f, \omega, t) = f\, cos(\omega t)$, which represents the periodic excitation function for a forced DVP oscillator, for a force-free Duffing van der pol oscillator $g(f, \omega, t) = 0$. ω is the angular frequency of the driving force, f is the amplitude of the excitation, $\mu > 0$ is the damping parameter of the system, while α and β are constant parameters.

The DVP oscillator equation has three main physically fascinating situations, (a) single-well ($\alpha > 0$, $\beta > 0$), (b) double-well ($\alpha < 0$, $\beta > 0$), and (c) double- hump ($\alpha > 0$, $\beta < 0$) [111].

Many authors have investigated solution of DVP oscillator from different perspectives, and in this regard many methods have been utilized. Among many authors, Cordshooli and Vahidi [112] used ADM, Chen and Liu [113] applied homotopy analysis method (HAM) to study the limit cycle of DVP oscillator, Sajadi et al. [114] used HPM and VIM to investigate the problem of single-well, double-well and double-hump of DVP oscillator. Khan et al. [115] studied the force-free DVP equation using modified version of homotopy pertubation method (NHPM).

Keeping in view the importance of DVP oscillator, we aim to investigate the solution of the DVP oscillator using the proposed heuristic computation approach. Further to exploit the application of HGA the force-free and forced DVP oscillator problems and its three special situations such as single-well, double-well, and double- hump are studied. The effectiveness and reliability of the proposed method are illustrated in comparison to some well-known classical approximate analytical and numerical methods.

Example 1: Consider the forced DVP oscillator given by (3.31) with the initial conditions $y(0) = 1, y'(0) = 0$, and the values of parameters as follows [114].

a) α =0.5, β =0.5, μ=0.1, ω=0.79, f=0.5 α > 0, β > 0 (single-well situation)

b) α = -0.5, β =0.5, μ=0.1, ω=0.79, f=0.5 α < 0, β > 0 (double- well situation)

c) α =0.5, β = -0.5, μ=0.1, ω=0.79, f=0.5 α > 0, β < 0 (double-hump situation)

The approximate solution is obtained in the domain $x \in [0, 1]$ with a step of 0.1 and m = 10. To apply the proposed method fitness function for each of the three cases is formulated. For instance the fitness function for single well situation is given as follows.

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) - 0.1(1-y^2)y + 0.5y'(x_i) + 0.5y^3(x_i) - 0.5cos(0.79x_i)\right)^2 + \frac{1}{2}\left((y(0)-1)^2 + \left(y'(0)\right)^2\right) \tag{3.32}$$

Similarly fitness functions for double well and double hump are formulated. GA, IPA, and hybrid scheme GA-IPA are used to solve the minimization problem corresponding to each case to find the best values of unknown parameters$(a_1, \ldots, a_{10}; b_1, \ldots, b_{10}; c_1, \ldots c_{10})$. The parameters settings used for the implementation of the algorithms are given in Table 3.32 and Table 3.33 for GA and IPA respectively. Further the approximate solution is obtained for $x \in [0,1]$.

Table 3.32 Parameter values and settings of GA for DVP oscillator (example 1)

	Value/Setting		
Parameter Name	**Single well**	**Double well**	**Double hump**
Population Size	240	240	240
Chromosome Size	30	30	30
Creation function	Uniform	Uniform	Uniform
Fitness scaling function	Proportional	Proportional	Proportional
Selection function	Stochastic Uniform	Stochastic uniform	Remainder
Mutation function	Adaptive feasible	Adaptive feasible	Adaptive feasible
Crossover function	Heuristic	Heuristic	Heuristic
Crossover fraction	0.6	0.8	0.6
Reproduction elite count	2	3	3
Migration direction	Both	Forward	Both
No. of generations	1000	1500	1500
Function tolerance	1e-20	1e-22	1e-24
Bounds	[-10,10]	[-10,10]	[-20,20]

The values of unknown parameters achieved by the algorithms are provided in Table 3.34, Table 3.35, and Table 3.36 for single-well, double-well, and double-hump situations respectively.

Table 3.33 Parameter values and settings of IPA for DVP oscillator (example 1)

Parameter Name	Value/Setting		
	Single well	**Double well**	**Double hump**
Start point	Random/best chromosome from GA	Random/best chromosome from GA	Random/best chromosome from GA
Maximum iterations	1000	1000	1000
Maximum function evaluations	200000	60000	150000
Function tolerance	1e-20	1e-18	1e-22
Derivative type	Forward differences	Forward differences	Central differences
Hessian	BFGS	BFGS	BFGS
Subproblem algorithm	ldl factorization	ldl factorization	ldl factorization

Table 3.34 Optimal values of unknown parameters (single-well)

	GA			IPA			GA-IPA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.2659	1.0928	-2.5220	-3.3665	-0.6959	-1.4886	-1.5957	0.8992	-2.1621
2	-0.0810	-2.6850	0.9051	-0.4120	-0.1052	0.2921	0.4326	-1.7698	1.0654
3	1.6722	-1.0819	1.7361	-0.2152	-0.6109	-0.7739	1.1261	-0.8825	1.7507
4	-0.0940	-3.1047	-0.6228	-0.5178	0.5514	-0.6724	-0.0252	-2.0079	-0.8485
5	-0.4393	1.8360	-0.9288	0.8251	-1.3982	0.6871	0.0000	0.9947	-0.5034
6	1.7550	1.8984	3.1466	-0.2017	1.4037	3.1358	0.9758	1.7730	1.7857
7	0.0057	-0.4851	0.2978	0.1842	-0.0729	-2.2518	0.0722	-0.3246	0.2143
8	-1.4339	-0.3037	4.6513	1.2980	1.0659	0.4704	-0.7689	-0.0419	2.9356
9	0.9572	0.4222	-1.4989	-2.6753	0.6193	-1.0091	0.5494	-0.1841	-1.1014
10	-0.7418	-1.3007	1.0751	1.7129	-0.8858	3.1321	-0.6217	-1.4058	0.5313

Table 3.35 Optimal values of unknown parameters (double-well)

	GA			IPA			GA-IPA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.1064	0.5200	0.2427	-2.8392	0.9916	-2.5073	-0.2841	-0.2368	0.2452
2	2.0328	1.2182	2.4292	2.2350	-1.0415	-0.5402	0.6256	1.1292	0.7149
3	2.2182	0.8602	-2.8779	0.6521	0.2876	1.6866	1.2648	-0.6402	-1.0964
4	-0.2162	1.2268	-1.4041	-1.5160	1.1629	-1.1019	-0.0425	0.5539	-0.6396
5	-0.3778	-1.4876	2.5962	-1.7064	0.1104	-0.8143	-0.5624	-1.6808	0.7901
6	-2.3695	0.3318	-1.4263	0.8760	1.4883	-0.0927	-1.1647	0.5725	-0.6754
7	-0.7805	1.9331	1.5199	-0.6281	0.2778	0.3370	0.2726	-0.1679	1.5727
8	-1.4112	1.9310	3.8129	-0.1592	1.0519	0.7380	-1.0453	1.3442	1.5174
9	2.4571	1.3904	-1.9188	3.6263	1.3949	-1.8049	1.5484	1.6832	-2.2322
10	2.2683	-1.5445	3.3774	0.7386	1.2720	-0.4530	1.7858	-1.0023	2.9674

Table 3.36 Optimal values of unknown parameters (double-hump)

	GA			IPA			GA-IPA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	1.1929	-0.7118	1.0438	3.8178	0.1871	-3.3957	1.2406	-0.7222	1.0955
2	2.0498	0.1176	-0.1452	5.3321	1.4184	-5.2507	2.1829	0.1944	-0.1591
3	0.0017	4.3052	1.5300	-2.1584	-3.0855	7.7026	0.0100	4.5949	1.6283
4	1.5534	1.6096	-1.8847	-1.3983	-0.5699	1.0850	1.6996	1.5871	-2.1010
5	-1.5118	-0.6555	0.0933	2.1968	-0.7480	-0.4469	-1.6212	-0.6521	0.1255
6	1.4405	-1.0716	1.7971	2.1958	0.1615	3.9974	1.4824	-1.0859	1.9104
7	-1.3925	1.6797	2.0564	-1.7812	-0.9191	1.4696	-1.4521	1.7219	1.9197
8	0.2457	-1.3267	2.0866	2.5775	0.7387	0.9743	0.2035	-1.4070	2.2227
9	2.5294	2.4623	-5.3330	8.9235	-0.7729	-2.5057	2.7916	2.5695	-5.8750
10	-0.4822	2.0900	5.2569	-0.2738	-0.5004	-1.3017	-0.4848	2.1950	5.6379

The values of parameters from Table 3.34, Table 3.35 and Table 3.36 are used in (3.4) to obtain the numerical solutions for each case at any desired value in the interval [0,1]. The numerical results obtained by the proposed method with GA and GA-IPA for the three situations, single-well, double-well, and double-hump are presented in Table 3.37, Table 3.38, and Table 3.39 respectively.

Since there isn't exact solution available to this problem, therefore fourth-order Runge-Kutta (RK4) method has been used for comparing the results. From the comparison of numerical results it can be clearly seen that the proposed method yields the approximate solutions for single-well, double-well, and double hump situations of DVP oscillator with significantly greater accuracy with an average absolute error of 9.29E-07, 6.03E-07, and 7.31E-07 for single-well, double-well, and double-hump respectively.

Furthermore in Table 3.40 we give a comparison of numerical solutions between the proposed method and the classical methods HPM and VIM reported in [114] at various values of x. The comparison of numerical solutions clearly shows that the proposed method is more accurate than HPM and VIM.

Table 3.37 Comparison of numerical solution (for single-well)

		Proposed method $y(x)$		**Absolute error $\lvert y_{RK4} - y(x) \rvert$**	
x	y_{RK4}	**GA**	**GA-IPA**	**GA**	**GA-IPA**
0.0	1.000000	1.000003	1.000000	3.49E-06	8.33E-08
0.1	0.997503	0.997506	0.997503	3.53E-06	5.09E-08
0.2	0.990045	0.990049	0.990045	3.64E-06	4.06E-07
0.3	0.977726	0.977729	0.977725	3.22E-06	6.21E-07
0.4	0.960702	0.960705	0.960702	2.85E-06	7.15E-07
0.5	0.939183	0.939186	0.939182	2.80E-06	9.53E-07
0.6	0.913415	0.913418	0.913414	2.60E-06	1.35E-06
0.7	0.883673	0.883676	0.883672	2.01E-06	1.60E-06
0.8	0.850249	0.850251	0.850248	1.43E-06	1.48E-06
0.9	0.813436	0.813437	0.813435	1.15E-06	1.37E-06
1.0	0.773522	0.773523	0.773521	6.63E-07	1.75E-06

Table 3.38 Comparison of numerical solution (for double-well)

		Proposed method $y(x)$		Absolute error $\lvert y_{RK4} - y(x)\rvert$	
x	y_{RK4}	GA	GA-IPA	GA	GA-IPA
0.0	1.000000	0.999998	1.000000	1.90E-06	3.08E-09
0.1	1.002497	1.002495	1.002497	1.96E-06	9.84E-08
0.2	1.009945	1.009945	1.009945	-1.45E-08	3.69E-07
0.3	1.022222	1.022224	1.022221	-2.46E-06	4.80E-07
0.4	1.039115	1.039118	1.039114	-3.43E-06	4.09E-07
0.5	1.060322	1.060325	1.060322	-2.87E-06	4.47E-07
0.6	1.085449	1.085451	1.085448	-2.41E-06	7.39E-07
0.7	1.114001	1.114005	1.114000	-3.65E-06	1.05E-06
0.8	1.145385	1.145391	1.145384	-6.43E-06	1.06E-06
0.9	1.178907	1.178915	1.178906	-8.54E-06	9.00E-07
1.0	1.213778	1.213787	1.213777	-8.41E-06	1.08E-06

Table 3.39 Comparison for numerical solution (for double-hump)

		Proposed method $y(x)$		Absolute error $\lvert y_{RK4} - y(x)\rvert$	
x	y_{RK4}	GA	GA-IPA	GA	GA-IPA
0.0	1.000000	0.999997	1.000000	2.57E-06	5.97E-09
0.1	1.002501	1.002498	1.002501	2.61E-06	1.09E-07
0.2	1.010012	1.010010	1.010012	2.49E-06	3.27E-07
0.3	1.022563	1.022560	1.022563	2.71E-06	3.87E-07
0.4	1.040203	1.040199	1.040202	3.18E-06	4.51E-07
0.5	1.063007	1.063004	1.063007	3.28E-06	6.90E-07
0.6	1.091089	1.091086	1.091088	2.92E-06	9.44E-07
0.7	1.124605	1.124602	1.124604	3.00E-06	1.03E-06
0.8	1.163775	1.163770	1.163774	4.45E-06	1.11E-06
0.9	1.208901	1.208895	1.208899	6.36E-06	1.42E-06
1.0	1.260394	1.260388	1.260392	6.33E-06	1.57E-06

Table 3.40 Comparison numerical solutions and absolute errors for DVP oscillator (example 1)

						Absolute errors		
	x	RK-4	GA-IPA	HPM	VIM	GA-IPA	HPM	VIM
single-well	0.2	0.99004	0.99004	0.99004	0.99004	0.0	0.0	0.0
	0.4	0.96070	0.96070	0.96075	0.96070	0.0	5.0E-05	0.0
	0.6	0.91341	0.91341	0.91383	0.91341	0.0	4.2E-04	0.0
	0.8	0.85024	0.85024	0.85216	0.85025	0.0	1.9E-03	1.0E-05
	1.0	0.77352	0.77352	0.77973	0.77353	0.0	6.2E-03	1.0E-05
double-well	0.2	1.00994	1.00994	1.00994	1.00994	0.0	0.0	0.0
	0.4	1.03911	1.03911	1.03918	1.03911	0.0	7.0E-05	0.0
	0.6	1.08544	1.08544	1.08621	1.08544	0.0	7.7E-04	0.0
	0.8	1.14538	1.14538	1.14937	1.14539	0.0	4.0E-03	1.0E-05
	1.0	1.21377	1.21377	1.22785	1.21382	0.0	1.4E-02	5.0E-05
double-hump	0.1	1.00250	1.00250	1.00250	1.00250	0.0	0.0	0.0
	0.2	1.01001	1.01001	1.01001	1.01001	0.0	0.0	0.0
	0.5	1.06300	1.06300	1.06296	1.06300	0.0	4.0E-05	0.0
	0.75	1.14346	1.14346	1.14209	1.14346	0.0	1.3E-03	0.0
	1.0	1.26039	1.26039	1.25055	1.26035	0.0	10.0E-03	4.0E-05

Example 2: Consider the following particular case of force-free DVP oscillator equation taken from [115]

$$y''(t) + \left(\frac{4}{3} + 3y^2\right)y' + \frac{1}{3}y + y^3 = 0 \tag{3.33}$$

with initial conditions,

$$y(0) = -0.28868, \qquad y'(0) = 0.12$$

The approximate solution of (3.33) is obtained using the proposed method in the interval $x \in [0,1]$. Converting (3.33) into an equivalent global error minimization problem by formulating its fitness function given below.

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) - \left(\frac{4}{3} + 3y^2(x_i)\right)y' + \frac{1}{3}y(x_i) + y^3(x_i)\right)^2 \tag{3.34}$$

$$+\frac{1}{2}((y(0) + 0.28868)^2 + (y'(0) - 0.12)^2)$$

The minimization problem given by (3.34) is solved using GA, IPA, and GA-IPA to achieve the best values of unknown parameters. The algorithms are executed according to the prescribed parameters values and settings given in Table 3.41.

Table 3.41 Parameter values and settings of the algorithms for DVP oscillator (example 2)

GA		IPA	
Parameters	**Value/Settings**	**Parameters**	**Value/Settings**
Population size	240	Start point	Random/ best values from GA
Chromosome size	30	Derivative type	Forward difference
Selection function	Stochastic uniform	Hessian	BFGS
Mutation function	Adaptive feasible	Subproblem algorithm	Ldl factorization
Crossover function	Heuristic	Maximum function evaluations	200000
Reproduction crossover fraction	0.6	Nonlinear constraint tolerance	1e-10
Function tolerance	1e-24	Function tolerance	1e-24
No. of generations	1500	Maximum iterations	1000
Bounds	-10, +10	Bounds	-10, +10

The unknown parameters achieved by the algorithms are given in Table 3.42, using these parameter values in (3.4) one can find the solution of (3.34) at any value of x in the solution domain. For the purpose of comparison with other methods the solution of (3.34) obtained using GA-IPA at different values of x in the interval [0,0.1] are presented in Table 3.43, also the solutions by RK4, NHPM, and DTM reported in [115] are provided.

To show the efficiency and accuracy of the proposed method absolute errors of the proposed method, NHPM, and DTM, computed relative to the RK4 method are provided in Table 3.44.

Table 3.42 Optimal values of unknown parameters for DVP oscillator (example 2)

	GA			**IPA**			**GA-IPA**		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.9131	0.0218	1.2990	-1.4770	-0.7548	-1.1181	-0.7596	-0.1201	0.7729
2	1.7054	0.9885	0.1404	0.0858	0.6864	0.3618	0.2300	1.3452	-0.0264
3	1.4322	1.3488	2.6614	-0.1268	1.4177	-0.3862	1.0234	1.2464	2.2217
4	0.7786	-0.2738	1.1724	0.1645	-1.9591	-0.9329	0.0264	0.2895	0.5620
5	0.3101	-0.6118	-1.1765	-0.2491	-1.6587	-1.2621	0.1240	-0.3583	-0.5593
6	-0.8639	1.0202	1.8268	-0.1192	-0.0368	-0.5265	-0.4020	1.1062	0.3713
7	0.0784	-0.6968	-0.2652	0.0011	-0.7248	0.3644	0.0995	-0.3095	-0.1391
8	-1.5082	0.9693	0.0739	0.4054	-0.9014	-1.1719	-1.2477	0.0225	0.2815
9	1.0304	0.1999	-1.3700	0.0226	1.7136	-0.2219	0.4482	0.7606	-0.7004
10	-0.9040	-0.5012	-0.1366	0.0448	-0.7260	-0.0224	-0.2430	0.5741	-0.1272

The comparison of absolute errors clearly illustrates the competency of the proposed method. The results obtained from the proposed method are found in an excellent agreement with the numerical method based on RK4 and more accurate than DTM and NHPM.

The higher accuracy of the proposed method can be best ascertained by comparing the absolute errors in Table 3.44, as it is found that DTM provides significantly high absolute errors with an average absolute error of 1.48E-4, NHPM gives an average absolute error of 8.37E-4, while the average absolute error yielded by the proposed method is 2.24E-8 in the interval $x \in [0,.1]$. Furthermore the improved performance of hybrid scheme GA-IPA is quite significant from the results.

Table 3.43 Comparison of numerical results for DVP oscillator (example 2)

		Proposed method		**Classical methods**	
x	**RK4**	**GA**	**GA-IPA**	**NHPM**	**DTM**
0.0	-0.28868000	-0.28868030	-0.28867998	-0.28868000	-0.28868000
0.01	-0.28748349	-0.28748376	-0.28748346	-0.28748347	-0.28748523
0.02	-0.28629387	-0.28629413	-0.28629384	-0.28629386	-0.28630661
0.03	-0.28511110	-0.28511135	-0.28511107	-0.28511108	-0.28514429
0.04	-0.28393510	-0.28393534	-0.28393507	-0.28393509	-0.28399838
0.05	-0.28276583	-0.28276606	-0.28276580	-0.28044716	-0.28286905
0.06	-0.28160320	-0.28160343	-0.28160318	-0.27929767	-0.28175642
0.07	-0.28044718	-0.28044740	-0.28044716	-0.27814566	-0.28066066
0.08	-0.27929769	-0.27929791	-0.27929768	-0.27701806	-0.27951892
0.09	-0.27815468	-0.27815490	-0.27815468	-0.27815466	-0.27852038
0.1	-0.27701808	-0.27701831	-0.27701811	-0.27701806	-0.27747621

Table 3.44 Comparison of absolute errors for DVP oscillator (example 2)

	Proposed method			**Classical methods**	
x	**GA**	**IPA**	**GA-IPA**	**NHPM**	**DTM**
0.0	2.97E-07	1.92E-08	1.56E-08	0.00E+00	0.00E+00
0.01	2.70E-07	1.12E-09	3.01E-08	1.75E-08	1.73E-06
0.02	2.59E-07	1.00E-10	2.95E-08	1.75E-08	1.27E-05
0.03	2.48E-07	4.00E-10	2.97E-08	1.74E-08	3.32E-05
0.04	2.39E-07	1.32E-09	2.96E-08	1.72E-08	6.33E-05
0.05	2.31E-07	6.02E-09	2.79E-08	2.32E-03	1.03E-04
0.06	2.25E-07	1.43E-08	2.39E-08	2.31E-03	1.53E-04
0.07	2.22E-07	2.64E-08	1.68E-08	2.30E-03	2.13E-04
0.08	2.22E-07	4.30E-08	6.03E-09	2.28E-03	2.21E-04
0.09	2.25E-07	6.41E-08	9.00E-09	1.71E-08	3.66E-04
0.1	2.32E-07	8.97E-08	2.86E-08	1.68E-08	4.58E-04

3.3.4 NONLINEAR SINGULAR BOUNDRY VALUE PROBLEMS ARISING IN PHYSIOLOGY

The numerical treatment of singular boundary value problems (BVPs) has been considered by many authors due to their substantial significance in engineering and science such as gas dynamics, atomic structures, atomic calculations, chemical reactions etc. [82].

Motivated by the potential applications of the nonlinear singular boundary value problems and the real challenge to solve such problems, we aim to investigate the numerical solution of nonlinear singular BVPs of the following form [116] using the proposed method.

$$y''(x) + \left(a + \frac{l}{x}\right)y'(x) = f(x,y), \qquad 0 \le x \le 1 \tag{3.35}$$

$$\eta_1 y(0) + \xi_1 y'(0) = \gamma_1 \tag{3.36}$$

$$\eta_2 y(1) + \xi_2 y'(1) = \gamma_2 \tag{3.37}$$

The assumptions normally applied on $f(x,y)$ are that it is continuous, $\frac{\partial f}{\partial y}$ exists and is continuous and $\frac{\partial f}{\partial y} \ge 0$, $\forall\, 0 \le x \le 1$. The singular BVP (3.35) – (3.37) occurs in numerous applications, especially with $l = 0,1,2$ and $a = 0$, in the study of many tumor growth problems [117], [118], with linear $f(x,y)$ and with nonlinear $f(x,y)$ of the following form.

$$f(x,y) \equiv f(y) = \frac{ny}{y+k} \ , n > 0, k > 0 \tag{3.38}$$

and with $l = 2$ and $a = 0$ in the study of steady state oxygen diffusion in a spherical cell with Michaelis-Menten uptake kinetics [119], [120], [121], [122].

A similar problem for $l = 2$ and $a = 0$ also arises in the study of the distribution of heat sources in the human head [123], [124], [125] in which

$$f(x,y) \equiv f(y) = -ne^{-nky}, n > 0, k > 0 \qquad (3.39)$$

A glance at the literature reveals that the BVPs of the form (3.35) have been given much attention and many analytical and numerical methods including non-polynomial cubic spline method (NPCSM) [116], finite difference method (FDM) [126], modified decomposition combined with B-spline collocation technique (MDM-BSC) [127], B-spline method (BSF) [128], cubic spline method (CSM) [129], nonclassical pseudo-spectral method (NCPSM) [130] have been utilized for their numerical solution.

We shall consider some special forms of (3.34) and obtain their numerical solutions by the proposed method to show its reliability and potency.

Example 1: Consider the following special case of (3.35) which arises in thermal explosions [127], [128]

$$y'' + \frac{1}{x}y' = -e^{y} \qquad (3.40)$$

subject to the boundary conditions

$$y'(0) = 0\,, \quad y(1) = 0 \qquad (3.41)$$

The exact solution of (3.40) is given by

$$y_{exact}(x) = 2\ln\left(\frac{c+1}{cx^2+1}\right) \qquad (3.42)$$

where $c = 3 - 2\sqrt{2}$

The approximate numerical solution $y(x)$ of (3.40) is obtained in the domain $x \in [0,1]$ using the proposed method by transforming the given problem into an equivalent global error minimization problem by formulating its fitness function ε_j. Assuming the number of basis functions m = 10 the fitness function is developed as follows.

$$\varepsilon_1 = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) + \frac{1}{x_i}y' + e^{y}\right)^2 \tag{3.43}$$

$$\varepsilon_2 = \frac{1}{2}\left(\left(y'(0)\right)^2 + \left(y(1)\right)^2\right) \tag{3.44}$$

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) + \frac{1}{x_i}y' + e^{y}\right)^2 + \frac{1}{2}\left(\left(y'(0)\right)^2 + \left(y(1)\right)^2\right) \tag{3.45}$$

The fitness function given by (3.45) is minimized by employing the algorithms GA, IPA, ASA and two hybrid schemes GA-IPA and GA-ASA for the determination of the optimal values of unknown adjustable parameters (α_i, b_i , c_i).

The parameter values and settings used for the implementation of the algorithms GA, IPA, GA-IPA and GA-ASA are given in Table 3.45.

Table 3.45 Parameter values and settings of algorithms for physiology problem (example 1)

GA		ASA		IPA	
Parameter Name	**Value/Settings**	**Parameter Name**	**Value/Settings**	**Parameter Name**	**Value/Settings**
Population size	240	Start point	Optimal values from GA	Start point	Optimal values from GA
Chromosome size	30	Maximum function evaluations	150000	Maximum function evaluations	150000
Selection function	Stochastic uniform	Minimum perturbation-	1e-8	Derivative type	Central differences
Mutation function	Adaptive feasible	Maximum perturbation	0.1	Hessian	BFGS
Crossover function	Heuristic	Nonlinear constraint tolerance	1e-18	Nonlinear constraint tolerance	1e-18
Crossover fraction	0.8	X tolerance	1e-6	X tolerance	1e-10
No. of generations	2000	Maximum iterations	400	Maximum iterations	1000

The size of the chromosome i.e the number of unknown adjustable parameters (α_i, b_i, c_i) are chosen equal to 30. The values of these unknown adjustable parameters are bounded between -15 and + 15.

The algorithms are executed according to the prescribed settings in Table 3.45 to achieve the minimum fitness. The optimal values of the unknown adjustable parameters corresponding to the minimum fitness are acquired.

The optimal values achieved by the hybrid schemes GA-IPA and GA-ASA are given in Table 3.46, while the values of unknown parameters achieved by IPA and ASA are provided in Table 3.47 respectively. The optimal values of the parameters are used in (3.4) and consequently the approximate numerical solution $y(x)$ of example 1 is obtained.

Table 3.46 Optimal values of unknown parameters acquired by hybrid schemes for physiology problem (example 1)

	GA-IPA			**GA-ASA**		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	-0.8811	0.1759	2.6939	-1.2227	0.1892	8.2529
2	-1.9142	1.0248	-0.2728	-2.5235	0.9329	-1.2987
3	0.9019	1.5552	0.2088	0.0937	2.9289	1.2779
4	-1.9473	0.2465	2.8167	-3.1845	0.0323	7.1516
5	0.3192	0.3115	2.5890	0.8419	0.6241	6.0167
6	0.3388	1.4130	2.1349	0.5436	2.6099	4.6536
7	1.8157	-1.1190	2.5583	2.1806	-1.2464	5.6357
8	0.5979	1.4398	0.8316	1.0886	1.6358	0.6876
9	-0.9108	-2.0555	-2.2809	-1.1748	-3.9638	-5.3264
10	0.9210	-0.5012	1.1042	1.0171	-0.8336	2.2371

The numerical results obtained by the proposed method are given in Table 3.48. Further for the accuracy of numerical results and the potency of the proposed method, absolute errors obtained by the proposed method are also presented in Table 3.49.

The comparisons are made with the exact solutions and the absolute errors obtained by the classical method MDM-BSC used in [127], further in [127] the authors used two different approaches of their method MDM-BSC, therefore the absolute errors corresponding to both the approaches are given for the purpose of comparison. It is quite evident from comparison of Table 3.49 that the absolute errors relative to the exact solutions by the proposed method with hybrid schemes GA-IPA and GA-ASA are much smaller than the errors by the approach I, whereas they are relatively smaller than approach II of MDM-BSC used in [127]. This means that the proposed method yields the approximate solutions more accurately than MDM-BSC.

Table 3.47 Optimal values of unknown parameters acquired by GA, IPA, and ASA for physiology problem (example 1)

	GA			**IPA**			**ASA**		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.3785	0.1894	8.2529	1.0795	0.1077	-0.7024	-0.1764	0.8695	0.1437
2	-2.5646	0.9838	-1.6204	-2.0467	-0.6301	-1.3108	-0.7016	-1.4298	-1.3980
3	-0.0133	2.9216	1.3046	-0.0374	0.4182	0.7255	-1.2910	-0.9288	-1.9555
4	-3.6260	0.0321	7.1516	1.5718	-0.1663	-0.2336	-0.2463	-0.9022	-1.1689
5	0.9861	0.6213	6.0177	-0.1613	-0.4959	-1.0897	-0.7336	-1.0942	-1.6778
6	0.6469	2.5866	4.6814	-3.9504	0.8494	-0.1398	-0.2519	-0.4502	-1.6248
7	2.4711	-1.0249	5.6964	0.2630	-0.5200	-0.6565	-0.1269	-0.0376	-0.5292
8	1.0356	1.8424	0.9401	1.1727	-0.0055	-0.6617	-1.9183	1.3167	-3.0009
9	-1.1208	-3.9339	-5.3886	-4.0559	-1.0826	-0.8891	1.2639	-1.3530	1.1566
10	1.0204	-0.6813	2.2690	2.5942	-0.8618	-0.0224	-0.2430	0.5741	-0.1272

Table 3.48 Comparison of numerical results for physiology problem (example 1)

		Proposed method $y(x)$				
x	$y_{exact}(x)$	GA	IPA	ASA	GA-IPA	GA-ASA
0	0.316694	0.316666	0.316724	0.316690	0.316692	0.316696
0.1	0.313266	0.313235	0.313289	0.313263	0.313264	0.313267
0.2	0.303015	0.302984	0.303037	0.303012	0.303014	0.303016
0.3	0.286047	0.286017	0.286066	0.286045	0.286046	0.286048
0.4	0.262531	0.262502	0.262546	0.262529	0.262530	0.262532
0.5	0.232697	0.232667	0.232709	0.232695	0.232696	0.232697
0.6	0.196827	0.196796	0.196837	0.196825	0.196826	0.196826
0.7	0.155248	0.155219	0.155259	0.155246	0.155248	0.155247
0.8	0.108323	0.108297	0.108335	0.108321	0.108322	0.108322
0.9	0.056439	0.056416	0.056451	0.056437	0.056438	0.056438
1.0	0.000000	-0.000020	0.000011	-0.000001	0.000000	0.000000

Moreover in Table 3.50 a comparison of maximum absolute errors obtained by the proposed method are made with maximum absolute errors obtained by MDM-BC [127] and BSM [128], for different numbers of mesh points (N and h). It may be worth to mention here that we have solved the given example (3.41) in the interval [0, 1] with a step of 0.1, which means total number of mesh points N = 11 in our case. The maximum absolute error obtained by the proposed method with hybrid schemes GA-IPA and GA-ASA are 3.99E-06 and 2.33E-06 respectively, while the maximum absolute errors by MDM-BSC (approach I) and MDM-BSC (approach II) are 1.05E-05 with N=20 and 2.00E-06 with N=20 respectively, also the maximum absolute errors of BSM are 3.50E-06 with h=1/60 and 1.55E-06 with h=1/90 respectively. One can clearly see from the comparison that the proposed method with fewer number of mesh points yield better accuracy than MDM-BSC and BSM, it can be henee inferred that the proposed method is much efficient and accurate.

Table 3.49 Comparison of absolute errors for physiology problem (example 1)

	Proposed method			MDM-BSC	
x	GA	GA-IPA	GA-ASA	Approach I (with N = 20)	Approach II (with N = 20)
0.0	2.87E-05	3.99E-06	-2.13E-06	1.05E-05	2.00E-06
0.1	3.04E-05	3.26E-06	-1.36E-06	1.05E-05	1.99E-06
0.2	3.13E-05	3.05E-06	-6.17E-07	1.03E-05	1.97E-06
0.3	2.99E-05	2.65E-06	-9.22E-07	1.02E-05	1.94E-06
0.4	2.91E-05	2.17E-06	-9.39E-07	9.93E-06	1.83E-06
0.5	2.98E-05	1.94E-06	-2.89E-07	9.62E-06	1.78E-06
0.6	3.04E-05	2.05E-06	5.00E-07	6.93E-06	1.67E-06
0.7	2.93E-05	2.17E-06	8.25E-07	4.75E-06	1.34E-06
0.8	2.62E-05	1.93E-06	5.64E-07	2.93E-06	9.20E-07
0.9	2.26E-05	1.36E-06	1.29E-07	1.37E-06	4.57E-07
1.0	2.04E-05	9.65E-07	6.22E-08	0	0

Table 3.50 Comparison of maximum absolute errors for physiology problem (example 1)

Proposed Method		MDM-BSC (Approach I)	MDM-BSC (Approach II)	BSM
3.13E-5	(GA)	1.05E-5 (N = 5)	3.22E-5 (N = 5)	3.16E-5 (h=20)
2.92E-5	(IPA)	1.05E-5 (N = 10)	8.06E-6 (N = 10)	7.87E-6 (h=1/40)
2.37E-6	(ASA)	1.05E-5 (N = 20)	2.00E-6 (N = 20)	3.50E-6 (h=1/60)
3.99E-6	(GA-IPA)	-	-	1.55E-6 (h=1/90)
2.13E-6	(GA-ASA)	-	-	4.97E-7 (h=1/161)

Example 2: We consider the following nonlinear singular boundary value problem [116]

$$y''(x) + \left(1 + \frac{l}{x}\right) y'(x) = \frac{5x^3(5x^5 e^y - x - l - 4)}{4 + x^5} \tag{3.46}$$

with the boundary conditions

$$y'(0) = 0\,,\quad y(1) + 5y'(1) = \ln\left(\frac{1}{5}\right) - 5 \tag{3.47}$$

To obtain the approximate numerical solution of (3.46) subject the boundary conditions (3.47) using the proposed method its fitness function with $m = 10$ is developed as follows.

$$\varepsilon_1 = \frac{1}{11} \sum_{i=1}^{11} \left(y''(x_i) + \left(1 + \frac{l}{x_i}\right) y'(x_i) - \frac{5x_i^3(5x_i^5 e^y - x_i - l - 4)}{4 + x_i^5} \right)^2 \tag{3.48}$$

$$\varepsilon_2 = \frac{1}{2}\left(\left(y'(0)\right)^2 + \left(\left(y(1) + 5y'(1)\right) - ln\left(\frac{1}{5}\right) + 5 \right)^2 \right) \tag{3.49}$$

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \tag{3.50}$$

The algorithms GA, IPA, GA-IPA and GA-ASA are executed with the same parameter values and settings used for example 1 and prescribed in Table 3.45 for the minimization of (3.50). To prove the effectiveness and viability of the proposed method the approximate numerical solutions are obtained for various values of the parameter ($l = 0.25, 1, 2, and\ 8$). The fitness function corresponding to each value of l is constructed and the minimization is performed using the above mentioned algorithms.

The optimal values of the unknown parameters are achieved corresponding to each value of l, which consequently provides the solution using (3.4).

In Table 3.51, Table 3.52, Table 3.53, and Table 3.54, the values of unknown parameters for l = 0.25, l = 1, l = 2, and l = 8 acquired by GA, GA-IPA, and GA-ASA are given respectively. Further in Table 3.55, Table 3.56, Table 3.57, and Table 3.58 the values of parameters for l = 0.25, l = 1, l = 2, and l = 8 acquired by IPA and ASA are given respectively.

The approximate numerical solutions obtained by the proposed method with hybrid schemes for l = 0.25, 1 are provided in Table 3.59 and for l = 2, 8 in Table 3.60 respectively and compared with the exact solutions. Comparisons show that the results are in a good agreement with the exact solutions.

Furthermore the comparison of the maximum absolute errors between the proposed method and the standard numerical methods including finite difference method (FDM) [126] and non-polynomial cubic spline method (NPCSM) [116] are presented in Table 3.61 for l = 0.25 and l =1 and in Table 3.62 for l = 2 and l = 8 respectively. From the comparison it is seen that the proposed method provides more accurate solutions with fewer mesh points as compared to FDM and NPCSM. For example the maximum absolute errors obtained by FDM and NPCSM for l = 0.25 and N = 64 are 7.64E-5 and 9.20E-5 respectively, whereas our method gives maximum absolute error of 7.17E-5 for l = 0.25 with total steps N = 11. Similarly the comparison of other values in Table 3.61 and Table 3.62 show that our method is more accurate than FDM and NPCSM even with less number of mesh points. More importantly our method is quite capable of providing the solutions of the given problem at any point in the solution domain, and not only at the mesh points, whereas classical methods lack this feature. The comparison of numerical results and absolute errors hence testify the effectiveness and reliability of the proposed method.

Table 3.51 Optimal values of unknown parameters (for $l = 0.25$)

	GA			GA-IPA			GA-ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	0.4645	-11.0029	-12.7637	2.0000	-2.5333	-2.0444	0.4645	-11.0029	-12.7637
2	-3.3333	2.1132	-5.3719	-1.2860	0.6286	-3.0862	-3.3332	2.1132	-5.3719
3	-0.1945	0.4747	-0.0423	0.7652	-0.8261	-0.4480	-0.1943	0.4747	-0.0424
4	-3.4541	2.7010	-3.7574	-3.3632	1.9412	-2.2492	-3.4541	2.7010	-3.7572
5	-5.4880	-14.9999	-13.8060	-2.8175	1.9073	-2.6633	-5.4880	-14.9999	-13.8060
6	-1.2864	-2.9748	2.6265	-0.7491	-3.1092	1.5960	-1.2865	-2.9750	2.6267
7	-6.5630	-5.1595	-8.0904	-2.5675	-1.6468	-2.3696	-6.5630	-5.1596	-8.0901
8	1.9884	2.4267	-5.9786	1.6710	3.3098	-2.9503	1.9884	2.4267	-5.9786
9	-9.0556	-1.1570	-11.8681	-2.9193	-1.4379	-1.4352	-9.0556	-1.1570	-11.8681
10	-1.0514	-1.2376	-10.0917	-0.5162	-0.6235	-2.7035	-1.0514	-1.2376	-10.0917

Table 3.52 Optimal values of unknown parameters (for $l = 1$)

	GA			GA-IPA			GA-ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	3.8195	-1.9143	0.2293	3.7882	-2.0878	1.2707	1.8965	-2.1341	2.9564
2	7.2250	-3.2295	-6.4082	7.2667	-3.4787	-6.1098	0.3267	1.9719	-5.2003
3	-4.9797	-1.9766	7.1616	-4.6588	-1.1020	7.4933	-0.8850	1.2080	1.7777
4	-3.1273	-1.3689	-0.4379	-2.8993	-1.4443	0.0504	-2.2760	2.8463	-2.6211
5	4.5041	1.7432	-1.0191	4.3725	2.5080	-2.0967	1.9020	3.3901	-3.1131
6	-1.9877	1.5358	-0.5245	-1.6835	1.1045	0.0682	-0.7209	-1.1057	2.5836
7	-0.0330	-5.6962	0.5521	-0.0596	-4.5078	1.3921	-2.0803	-2.7215	1.8019
8	2.3172	1.9496	4.0610	2.3887	1.6406	4.3001	0.9292	1.8675	0.7355
9	0.3648	-1.4280	-5.1585	0.3849	-1.4387	-5.1515	0.0021	-0.3118	-1.2402
10	-3.4563	3.1205	-4.4858	-4.5847	2.5148	-3.4766	-4.8986	1.3781	-2.1028

Table 3.53 Optimal values of unknown parameters (for $l = 2$)

	GA			GA-IPA			GA-ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-8.6140	1.6960	-0.1481	-1.5719	-1.6444	-0.7302	-9.7449	1.6242	-1.5700
2	1.0170	5.3433	-1.6950	-1.8035	1.8918	-1.5266	1.6474	2.9120	-1.3303
3	-3.1255	4.5555	-8.3842	-3.5063	2.2638	-2.8817	-2.7850	0.7339	-4.2018
4	3.4662	-1.3151	7.6439	0.7128	1.8895	-0.7040	1.2212	-1.5083	7.5182
5	-1.7994	7.1615	-0.9825	-1.4399	-2.6231	1.4387	-0.8603	4.4232	0.8013
6	4.7603	5.9590	-0.1783	-0.3132	0.4443	0.7348	5.0458	2.5000	1.4183
7	-0.5780	-0.2694	-2.1789	1.2425	-0.6745	0.0592	-0.6824	0.2961	-2.1216
8	-2.6991	8.7355	1.6223	2.4579	-1.3260	1.3594	-3.1398	7.1871	4.7598
9	1.6005	4.4085	-3.1078	0.8418	-0.9377	-1.4212	2.8864	3.0900	-2.6822
10	-0.3619	-1.7673	4.9763	-2.1032	-3.1070	2.8363	-1.7047	-0.4566	5.7473

Table 3.54 Optimal values of unknown parameters (for $l = 8$)

	GA			GA-IPA			GA-ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-0.2257	-1.1836	-2.8669	-1.3239	-1.1273	0.0434	-0.7399	-1.5733	-2.7226
2	0.3734	3.4546	-2.5224	1.4318	2.6117	-1.0620	0.8242	3.0552	-2.5513
3	-0.7075	-0.1457	1.5161	-1.0112	0.9984	0.6244	-1.0746	0.0356	1.3827
4	-0.0419	-2.2342	-1.2315	0.4112	-3.1178	0.6917	0.1894	-2.2031	-1.5957
5	-3.8394	0.9589	-4.1366	-4.0473	2.1213	-2.5848	-3.8033	1.2021	-4.0579
6	2.1366	-3.5035	4.9886	2.6130	-1.6064	2.2561	2.5461	-3.0677	4.3265
7	1.2592	1.9825	5.3292	0.7952	1.0222	-0.0842	1.4097	2.5280	4.9714
8	0.6915	-0.9305	-2.9232	-1.0968	0.4663	-0.8065	0.5381	-0.4168	-2.8960
9	-2.3620	-4.4942	14.8668	-1.7548	-3.2295	2.9002	-2.4955	-4.4851	14.8720
10	-1.8242	0.3485	4.3434	-1.6845	0.0721	0.7767	-1.9805	0.2997	4.3166

Table 3.55 Optimal values of unknown parameters acquired by IPA and ASA (for $l = 0.25$)

	IPA			ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	-5.0450	-2.1297	0.2688	-2.5341	0.5575	0.4658
2	-0.7089	2.1228	0.4124	-6.4918	0.2307	-3.6270
3	0.4363	0.0338	-5.1717	-1.6216	-2.8541	2.5707
4	-2.0510	-3.0286	2.7527	-1.2127	-0.3124	0.9431
5	-3.7253	-0.8941	-1.7940	1.4676	-0.2832	1.6961
6	1.6650	2.1654	3.9908	4.0139	-2.6245	3.6650
7	-3.9611	2.3016	-0.1608	-0.5124	-0.0296	-1.1720
8	6.1221	-2.1664	2.8072	2.1264	0.4966	-2.8326
9	-0.4042	0.6195	1.6191	-2.3181	-0.1038	2.7637
10	-0.9159	3.9117	3.6583	-1.6515	-1.0750	-1.7132

Table 3.56 Optimal values of unknown parameters acquired by IPA and ASA (for $l = 1$)

	IPA			ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	1.4269	1.7359	2.7372	-0.6465	-0.3778	-2.2004
2	-0.5171	0.6346	-0.5211	-1.6257	-0.7012	-4.3560
3	-5.5356	2.3600	-3.0655	-3.2384	2.8295	-3.9248
4	-0.8349	-1.7210	1.0385	-2.6482	-2.6118	2.3807
5	-0.4766	0.0978	-0.5076	-1.5857	0.7136	-1.5254
6	0.4918	-1.7178	-0.9725	-3.4426	-1.0424	-1.1269
7	0.0063	-0.8879	0.6653	-5.3317	0.8744	-8.9517
8	1.3780	0.0851	0.5231	0.2110	2.6660	-8.8604
9	-1.2440	0.6034	-2.0543	1.3939	-2.1289	2.0037
10	-2.5210	-2.8455	2.6489	7.2136	-0.6346	-1.7331

Table 3.57 Optimal values of unknown parameters acquired by IPA and ASA (for $l = 2$)

	IPA			ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	1.2185	0.8865	-0.3997	-3.3992	2.9605	-4.0291
2	-1.4334	-2.8795	1.1542	-4.0197	2.1469	-1.4973
3	1.4202	0.1907	0.7721	-2.0958	-2.4289	2.6983
4	-0.0849	0.2228	-1.5794	6.6361	0.0988	1.3600
5	0.4499	-0.3395	-1.0997	-0.4092	-1.8121	-2.1186
6	-1.5452	2.7385	-1.0849	-1.6187	1.4943	-2.6948
7	-0.9775	0.5348	-0.9554	0.9990	-5.4605	-5.7493
8	-1.3872	-3.1154	2.7321	1.5214	0.4419	-8.0573
9	0.5762	-0.3566	-0.2574	-4.3467	-2.2675	1.7185
10	-4.3952	2.4202	-3.3998	-0.9211	1.7231	-2.3386

Table 3.58 Optimal values of unknown parameters acquired by IPA and ASA (for $l = 8$)

	IPA			ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i
1	1.6413	2.1312	1.5469	-2.2039	2.8183	1.2797
2	0.5038	2.9201	-1.5619	-1.0631	0.3609	1.9999
3	1.3585	0.2000	-2.2557	5.7234	0.0890	-5.1640
4	0.6267	-1.7196	2.4872	3.5446	-2.7146	4.1850
5	1.4304	-1.6661	-1.2063	-0.5995	3.8796	-4.5984
6	-1.1030	-3.3340	2.9756	-0.4438	-1.2752	1.8647
7	-1.4255	0.5231	4.2991	2.6149	-2.5180	-2.3388
8	-0.8620	-1.6547	-1.6612	-3.4821	-2.9503	-1.5066
9	-4.0077	2.3362	-3.2787	-1.5088	-2.6187	2.7029
10	-1.2676	2.0930	2.0654	-1.0031	0.8921	-3.0921

Table 3.59 Comparison of numerical results (for l = 0.25, 1)

x	y_{exact}	Proposed method $y(x)$		$\lvert y_{exact} - y(x)\rvert$	
		$l = 0.25$	$l = 1$	$l = 0.25$	$l = 1$
0.0	-1.386294	-1.386223	-1.386291	7.17E-05	3.21E-06
0.1	-1.386297	-1.386232	-1.386296	6.47E-05	6.58E-07
0.2	-1.386374	-1.386316	-1.386377	5.80E-05	2.49E-06
0.3	-1.386902	-1.386849	-1.386907	5.26E-05	5.21E-06
0.4	-1.388851	-1.388803	-1.388858	4.83E-05	7.19E-06
0.5	-1.394077	-1.394032	-1.394085	4.47E-05	8.57E-06
0.6	-1.405548	-1.405507	-1.405558	4.13E-05	1.04E-05
0.7	-1.427453	-1.427415	-1.427465	3.85E-05	1.16E-05
0.8	-1.465032	-1.464995	-1.465044	3.65E-05	1.21E-05
0.9	-1.523987	-1.523953	-1.524000	3.41E-05	1.35E-05
1.0	-1.609438	-1.609405	-1.609452	3.28E-05	1.38E-05

Table 3.60 Comparison of numerical results (for l = 2, 8)

		Proposed method y(x)		$\lvert y_{exact} - y(x)\rvert$	
x	y_{exact}	$l = 2$	$l = 8$	$l = 2$	$l = 8$
0.0	-1.386294	-1.38632	-1.386328	2.91E-05	3.35E-05
0.1	-1.386297	-1.38626	-1.386332	3.27E-05	3.47E-05
0.2	-1.386374	-1.38633	-1.386409	4.45E-05	3.46E-05
0.3	-1.386902	-1.38686	-1.386936	4.38E-05	3.47E-05
0.4	-1.388851	-1.3888	-1.388886	5.28E-05	3.46E-05
0.5	-1.394077	-1.39402	-1.394111	5.17E-05	3.45E-05
0.6	-1.405548	-1.4055	-1.405583	4.45E-05	3.48E-05
0.7	-1.427453	-1.42741	-1.427488	4.69E-05	3.46E-05
0.8	-1.465032	-1.46498	-1.465066	5.47E-05	3.44E-05
0.9	-1.523987	-1.52393	-1.524022	5.27E-05	3.50E-05
1.0	-1.609438	-1.60939	-1.609472	4.67E-05	3.46E-05

Table 3.61 Comparison of maximum absolute errors (for $l = 0.25, 1$)

Proposed method (N = 10)			Other methods				
				FDM		NPCSM	
	$l = 0.25$	$l = 1$	N	$l = 0.25$	$l = 1$	$l = 0.25$	$l = 1$
GA	1.36E-4	6.46E-4	16	1.17E-4	2.07E-4	1.46E-3	1.71E-3
IPA	1.10E-4	1.43E-4	32	3.04E-4	1.87E-4	3.68E-4	1.87E-4
ASA	1.42E-4	3.23E-4	64	7.67E-5	3.88E-5	9.20E-5	1.96E-5
GA-IPA	7.17E-5	1.14E-5	128	1.92E-5	8.10E-5	2.30E-5	1.72E-5

Table 3.62 Comparison of maximum absolute (for $l = 2, 8$)

Proposed method (N = 11)			Other methods			
				FDM		NPCSM
	$l = 2$	$l = 8$	N	$l = 2$	$l = 8$	$l = 2$
GA	9.04E-2	1.11E-4	16	1.82E-3	7.71E-3	4.11E-3
IPA	1.26E-4	1.1E-4	32	4.52E-4	7.78E-5	9.76E-4
ASA	9.52E-5	1.42E-4	64	9.20E-5	7.05E-5	2.38E-4
GA-IPA	5.47E-5	6.47E-5	128	2.80E-5	6.45E-6	5.89E-5

Example 3: Consider the special case of (3.35) - (3.37), the non-linear heat conduction model of the human head as follows [127], [128], [129], [130].

$$y''(x) + \frac{2}{x}y'(x) = -e^{-y} \qquad (3.51)$$

with the following boundary conditions

$$y'(0) = 0 \qquad (3.52)$$

$$y(1) + y'(1) = 0 \qquad (3.53)$$

To apply the proposed method for obtaining the approximate numerical solution of this problem its fitness function ε_j is formulated. Assuming the number of basis functions, m = 10 then ε_j can be expressed as follows.

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\left(y''(x_i) + \frac{2}{x_i}y'(x_i) + e^{-y}\right)^2 \qquad (3.54)$$
$$+\tfrac{1}{2}\left(\left(y(0)\right)^2 + ((y(1) + y'(1)))^2\right)$$

where $y(x), y'(x)$ *and* $y''(x)$ are given by (3.4) , (3.9), and (3.10) respectively.

The FF given by (3.54) is minimized by applying GA, IPA and hybrid scheme GA-IPA for the obtaining the unknown parameters$(a_1, \dots, a_{10}; b_1, \dots, b_{10}; \textit{and } c_1 \dots, c_{10})$.

The algorithms are implemented with the parameter values and settings prescribed in Table 3.63. The size of the chromosome is chosen equal to 30 which basically equals the number of the unknown parameters with the assumed number of basis functions.

The optimal values of the unknown parameters achieved by the algorithms GA, IPA, and GA-IPA are given in Table 3.64. Consequently the approximate solution $y(x)$ is obtained by using the values of the unknown parameters in (3.4).

The approximate numerical results by our method are given in Table 3.65 and compared with results obtained by some standard methods including CSM and NCPSM. Since there isn't any exact solution of this problem, comparisons are therefore made with the approximate results reported in the literature, in order to prove the validly of our results. The comparison shows that the proposed method provides numerical results that are fairly comparable with other methods.

Table 3.63 Parameter values and settings of the algorithms for physiology problem (example 3)

GA		IPA	
Parameter Name	**Value/Setting**	**Parameter name**	**Value/Setting**
Population size	240	Start point	Random/Optimal chromosome from GA
Chromosome size	30	Maximum iterations	1000
Selection function	Stochastic uniform	Maximum function evaluations	150000
Mutation function	Adaptive feasible	Function tolerance	1e-18
Crossover function	Heuristic	Derivative type	Central differences
Hybridization	IPA	Hessian	BFGS
No. of generations	1000	Sub problem algorithm	ldl factorization
Function tolerance	1e-18	Initial barrier parameter	0.1
Bounds	-10, +10	Bounds	-10, +10

Table 3.64 Optimal values of unknown parameters for physiology problem (example 3)

	GA			IPA			GA-ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-1.1233	1.7934	-2.1238	-0.6193	0.0376	0.2301	-1.2971	0.9206	-1.6694
2	-3.1302	0.4825	2.5170	-3.0118	1.5522	-5.1618	-2.1063	0.3835	1.3038
3	1.7488	2.7557	1.8540	-0.1009	-0.2647	0.6070	0.9493	1.9724	1.3476
4	-0.0342	-4.6162	-1.2431	0.7647	1.4617	2.3210	0.2877	-2.2574	-0.8831
5	0.6347	1.3466	-3.8221	-1.1154	0.0916	-0.5656	-0.2777	0.0754	-2.2691
6	1.0341	-0.3715	-4.8659	0.8104	0.1709	0.4989	0.6410	-0.2561	-2.3409
7	1.3022	2.0246	-1.0301	0.6769	-1.1856	1.9077	0.3875	1.9560	-0.7369
8	3.2473	-1.0030	0.8715	-0.1090	0.1419	-1.3340	2.4039	-0.6392	1.3026
9	-0.8915	-2.3917	2.8963	0.1920	0.0710	-1.2344	-0.5890	-1.8764	2.3467
10	0.5030	-4.1038	-1.8094	-0.9833	-0.0720	0.4972	-0.3548	-1.8158	-0.7197

Table 3.65 Comparison of numerical results for physiology problem (example 3)

	Proposed method			Other methods	
x	GA	IPA	GA-IPA	CSM	NCPSM
0.0	0.367508	0.367518	0.367516	0.367518	0.367517
0.1	0.366348	0.366363	0.366361	0.366363	0.366362
0.2	0.362879	0.362894	0.362893	0.362895	0.362894
0.3	0.357082	0.357098	0.357095	0.357099	0.357098
0.4	0.348932	0.348948	0.348946	0.348950	0.348948
0.5	0.338395	0.338412	0.338411	0.338413	0.338412
0.6	0.325427	0.325444	0.325442	0.325445	0.325444
0.7	0.309968	0.309986	0.309984	0.309987	0.309986
0.8	0.291951	0.291971	0.291969	0.291972	0.291971
0.9	0.271296	0.271317	0.271315	0.271318	0.271317
1.0	0.247908	0.247928	0.247926	0.247929	0.247928

Example 4: We Consider another special case of (3.35) – (3.37), the oxygen diffusion problem, given by (3.55) – (3.57) [127], [128], [129], [130], as the final example to illustrate the effectiveness of the proposed method.

$$y''(x) + \frac{2}{x}y'(x) = \frac{0.76129y}{y + 0.03119} \qquad (3.55)$$

with the boundary conditions

$$y'(0) = 0 \qquad (3.56)$$

$$5y(1) + y'(1) = 5 \qquad (3.57)$$

As explained above we formulate the fitness function (FF) to convert the given problem into an equivalent global error minimization problem. We have taken m = 10 therefore the FF is expressed as given below.

$$\varepsilon_j = \frac{1}{11}\sum_{i}^{11}\left(y''(x_i) + \frac{2}{x_i}y'(x_i) - \frac{0.76129y}{y + 0.03119}\right)^2 + \tfrac{1}{2}\left(\left(y(0)\right)^2 + ((5y(1) + y'(1)) - 5)^2\right) \qquad (3.58)$$

The FF given by (3.58) is minimized by applying GA, IPA and hybrid scheme GA-IPA for the obtaining the unknown parameters$(a_1, \dots, a_{10}; b_1, \dots, b_{10}; and\ c_1 \dots, c_{10})$.

The algorithms are implemented with the same parameter values and settings prescribed in Table 3.63 for example 3.

The optimal values of the unknown parameters achieved by the algorithms GA, IPA, and GA-IPA are given in Table 3.66. Consequently the approximate solution $y(x)$ of the oxygen diffusion problem is obtained by using the values of the unknown parameters in (3.4).

The approximate numerical results by our method are presented in Table 3.67, also results obtained by other methods including BSM, CSM, and MDM-BSC are shown for the sake of comparison and validity of our results. From the comparison the accuracy of the proposed method is found comparable to other methods.

Table 3.66 Optimial values of unknown parameters for physiology problem (example 4)

	GA			IPA			GA-ASA		
i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	-2.5066	0.9151	-0.3361	-0.0721	0.9923	0.7528	-1.5397	0.4938	0.1192
2	0.1773	-1.4705	-0.9257	1.6288	0.5059	-0.8761	0.1912	-1.3684	-1.0948
3	2.0467	0.5311	-0.3352	0.4121	0.7771	0.1800	1.8041	-0.0003	0.1195
4	0.8442	0.6962	-0.8461	2.0729	0.8476	-2.0114	0.8651	0.4690	-0.9862
5	0.6371	-1.8234	-1.5871	-0.6035	0.6472	1.0943	0.9935	-1.1294	-2.3936
6	-0.4674	-0.1594	-2.5157	-0.6359	-0.0892	-1.8421	-0.4392	-0.1436	-2.7392
7	0.8110	1.0952	0.5406	1.4881	-0.9799	-1.2437	0.1778	1.0984	0.4652
8	1.7931	0.8252	-2.0905	-0.2568	-0.6686	0.2576	2.6786	0.7694	-2.7146
9	-0.2965	-0.2861	-0.8390	-0.2557	0.7775	-0.0619	-0.1124	-0.2406	-0.8413
10	0.1647	0.6359	-1.8501	0.4901	-1.2738	1.5504	0.8953	0.7298	-1.9954

Table 3.67 Comparison of numerical solutions for physiology problem (example 4)

	Proposed method			Traditional methods		
x	GA	IPA	GA-IPA	CSM	BSM	MDM-BSC
0.1	0.82845	0.82847	0.82848	0.82848	0.82848	0.82848
0.2	0.82969	0.82970	0.82970	0.82971	0.82971	0.82971
0.3	0.83336	0.83337	0.83337	0.83337	0.83337	0.83337
0.4	0.83948	0.83949	0.83949	0.83949	0.83949	0.83949
0.5	0.84804	0.84805	0.84805	0.84805	0.84805	0.84805
0.6	0.85906	0.85907	0.85906	0.85906	0.85906	0.85906
0.7	0.87252	0.87253	0.87253	0.87253	0.87253	0.87253
0.8	0.88843	0.88845	0.88845	0.88845	0.88845	0.88845
0.9	0.90680	0.90682	0.90682	0.90682	0.90682	0.90682
1.0	0.92763	0.92765	0.92765	0.92765	0.92765	0.92765

3.4 CONCLUSION

A simple yet an efficient heuristic computation method based on the hybrid approach of polynomial basis and evolutionary algorithm has been successfully applied to the solution of nonlinear problems governed by NODEs. The accuracy and reliability of the proposed method have been demonstrated by numerically solving several important nonlinear problems arising in diverse fields of engineering.

The proposed method has been applied to the Bratu boundary value problem and an initial value problem of the Bratu-type. The numerical solutions obtained by the proposed method are found in a good agreement with the exact solutions and more accurate than some of the standard methods.

The validity of the proposed method has also been demonstrated by solving the Troesch's problem with three special cases of the parameter σ governing the equarion, and especially for $\sigma = 10$ for which methods like ADM, VIM, and MHPM fail to solve this problem, the proposed method has obtained approximate solutions with good accuracy.

Furthermore the efficiency of the proposed method has been illustrated by solving the three interesting situations, single-well, double-well, and double-hump of forced DVP oscillator. It has been established from the comparisons of numerical solutions that the proposed method gives more accurate solutions than some of the well-known classical methods.

Moreover, the strength of proposed method has been demonstrated by solving several nonlinear problems appearing in physiology. The results from the proposed method are found in an excellent agreement with the exact solutions and quite comparable (or more accurate) than some of the classical methods.

On the basis of the simulation results and comparisons made with some standard methods and exact solutions, it can be concluded that the proposed heuristic computing method is effective, handy, and possesses a great potential and viability for broad range of NODEs. Moreover the beauty of the proposed method is that it can provide the approximate solution at any point in the solution domain once the optimal values of unknown parameters are obtained.

3.5 SUMMARY

This chapter provides the detail of the heuristic computation method combining log sigmoid basis functions and evolutionary algorithm for solving nonlinear ODEs. It also presents an extensive study of numerical applications of the designed method. The chapter gives the basic idea of fitness function used for transforming the NODE into an equivalent optimization problem in the feasible form for evolutionary algorithm. The procedural steps of memetic algorithms combining GA with local search algorithms IPA, ASA, and PS for solving the optimization problem are also presented. The designed method has been successfully applied to solve some important nonlinear problems including the classical Bratu problem, Troesch's problems, Duffing van der pol oscillator equation, nonlinear oxygen diffusion problem, nonlinear heat conduction model of human head and various other nonlinear singular BVPs arising in physiology. From the simulation results presented in this chapter it has been established that the proposed method is quite competent and viable for solving broad range of nonlinear ODEs. Moreover the overall performance of the memetic algorithm schemes GA-IPA, GA-ASA, and GA-PS have been found quite better than using GA alone as a global search optimizer.

CHAPTER 4

HYBRID EVOLUTIONARY ALGORITHM TECHNIQUE FOR SOLVING COUPLED NONLINR ORDINARY DIFFERNTIAL EQUATIONs

This chapter provides the detail of the proposed heuristic computation method, combining polynomial basis functions and EAs for solving coupled nonlinear ODEs (CNODEs). The approximate solution of CNODE is deduced as a linear combination of polynomial basis functions with some unknown parameters. A fitness function is used to convert the system of CNODEs into an equivalent global error minimization problem. Two popular EAs including GA and DE and memetic algorithms schemes GA-IPA, and GA-ASA are employed to solve the minimization problem and to obtain the unknown parameters. The numerical applications of the proposed method are investigated for solving two important problems modeled by the systems of CNODEs. The material provided in the rest of the chapter is mostly from the published work [131], [132].

4.1 INTRODUCTION

Coupled nonlinear ODEs (CNODEs) occur in many situations in applied science and engineering including the models of biological systems, enzyme reactions, and fluid mechanics. Most of such nonlinear problems do not admit the analytical solution, therefore these problems are solved using some approximate techniques. Many standard methods including VIM, ADM, HPM, and LADM have been proposed for solving such CNODEs.

In this dissertation, an alternate heuristic computation method to the existing standard methods is presented for solving the systems of CNODEs. The method is based on the combination of polynomial basis functions and EAs, which is described in the next section. The method has been successfully applied to two important problems including Michaelis-Menten biochemical reaction model and the HIV infection model of CD4$^+$T cells. Although stochastic solvers based on EAs have been used in recent years by many authors for solving various nonlinear problems of NODEs, but nobody as yet has attempted the method that is proposed in this dissertation for solving these CNODEs.

4.2 METHODOLOGY FOR SOLVING CNODEs

In this section, the methodology for solving CNODEs is described. The method is heuristic which employs the hybrid approach of polynomial basis functions and EAs. The approximate solution of CNODE is expanded as a linear combination of polynomial basis with unknown parameters. The CNODE is transformed into an equivalent global error minimization problem. A trial solution is formulated using a fitness function with unknown parameters. EAs are used to solve the minimization problem and to obtain the unknown parameters. The description of the method is given below.

Consider a system of CNODEs of the following form.

$$\frac{dy_1(t)}{dt} = f_1\big(x, y_1(t), y_2(t), \dots \dots \dots, y_n(t)\big) \tag{4.1}$$

$$\frac{dy_2(t)}{dt} = f_2\big(t, y_1(t), y_2(t), \dots \dots \dots, y_n(t)\big) \tag{4.2}$$

$$\vdots$$

$$\vdots$$

$$\frac{dy_n(t)}{dt} = f_n\big(t, y_1(t), y_2(t), \dots \dots \dots, y_n(t)\big) \tag{4.3}$$

with the following initial conditions

$$y_1(0) = k_1, \qquad y_2(0) = k_2, \quad y_n(0) = k_3 \tag{4.4}$$

where k_1, k_2, and k_3 are real constants, and $f_1, f_2, \dots, f_n$ represent some nonlinear functions. To solve the system of CNODEs given by (4.1) – (4.3), we may assume that the approximate solutions $\hat{y}_1(t), \hat{y}_2(t), \dots\dots, \hat{y}_n(t)$ and their first derivatives $\frac{d\hat{y}_1(t)}{dt}, \frac{d\hat{y}_2(t)}{dt}, \dots, \frac{d\hat{y}_n(t)}{dt}$ are a linear combinations of some polynomial basis functions,$t^i, t^j, \dots, t^k$ which can be expressed as follows.

$$\hat{y}_1(t) = \sum_{i=0}^{m} a_i\, t^i \tag{4.5}$$

$$\hat{y}_2(t) = \sum_{j=0}^{m} b_j\, t^j \tag{4.6}$$

$$\vdots$$

$$\vdots$$

$$\hat{y}_n(t) = \sum_{k=0}^{m} c_k\, t^k \tag{4.7}$$

$$\frac{d\hat{y}_1(t)}{dt} = \sum_{i=0}^{m} a_i \, i \, t^{i-1} \tag{4.8}$$

$$\frac{d\hat{y}_2(t)}{dt} = \sum_{j=0}^{m} b_j \, j \, t^{j-1} \tag{4.9}$$

$$\vdots$$

$$\vdots$$

$$\frac{d\hat{y}_n(t)}{dt} = \sum_{k=0}^{m} c_k k \, j \, t^{k-1} \tag{4.10}$$

where a_i, b_j, and c_k are unknown real parameters, m is the number of basis functions. The aim is to find the unknown parameters (a_i , b_j , and c_k) in (4.5) – (4.7), which consequently gives the approximate solutions $\hat{y}_1(t), \hat{y}_2(t), \ldots\ldots, \hat{y}_n(t)$ of the given system of CNODEs. To find the required unknown coefficients the given system of CNODE is converted into an equivalent global error minimization problem using a problem exclusive fitness function.

4. 2.1 FITNESS FUNCTION

The fitness function (FF) denoted as (ε_j) represents the global error associated with the given system of CNODEs, and is represented as follows

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_n \quad j = 1,2,3 \ldots \tag{4.11}$$

where j is the generation number/iteration count of the algorithm, and ε_1, $\varepsilon_2, \ldots, \varepsilon_n$ are defined as the mean of sum of square errors linked with each ODE, which are given by

$$\varepsilon_1 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{d}y_1(t_i)}{dt} - f_1\big(t_i, \hat{y}_1(t_i), \hat{y}_2(t_i), \dots\dots\dots, \hat{y}_n(t_i)\big)\right)^2 \tag{4.12}$$

$$\varepsilon_2 = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\hat{d}y_2(t_j)}{dt} - f_1\left(t_j, \hat{y}_1(t_j), \hat{y}_2(t_j), \dots\dots\dots, \hat{y}_n(t_j)\right)\right)^2 \tag{4.13}$$

$$\vdots$$

$$\varepsilon_n = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{\hat{d}y_n(t_k)}{dt} - f_1\big(t_k, \hat{y}_1(t_k), \hat{y}_2(t_k), \dots\dots\dots, \hat{y}_n(t_k)\big)\right)^2 \tag{4.14}$$

where $\hat{y}_1(t), \hat{y}_2(t), \dots\dots, \hat{y}_n(t)$ and their first derivatives $\frac{d\hat{y}_1(t)}{dt}, \frac{d\hat{y}_2(t)}{dt}, \dots, \frac{d\hat{y}_n(t)}{dt}$ are given by (4.5) – (4.10) respectively. *N* is the total number of steps taken in the solution domain of time *t*.

The error minimization problem given by (4.11) is solved using EAs and the optimal values of the unknown coefficients are achieved. The optimal values of these unknown coefficients are used in (4.5) – (4.7), which provide the approximate solutions $\hat{y}_1(t), \hat{y}_2(t), \dots\dots, and\ \hat{y}_n(t)$ of the given system of CNODEs.

4.2. 2 EVOLUTIONARY ALGORITHMS

In this section, evolutionary algorithms used for solving the minimization problem given by the fitness function (4.11) are introduced.

GA, GA-IPA, GA-ASA, and DE, have been employed for solving FF and to obtain the unknown parameters $(a_i\,, b_i\,,\ and\ c_i)$. The procedural steps of DE are given in algorithm 4.1, while the procedural steps of HGA's are given in algorithm 3.1.

Algorithm 4.1: Differential Evolution (DE)

Step 1: (Population Initialization)

A population of N chromosomes $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ is generated randomly. Each chromosome has D number of genes representing the number of unknown parameters.

Step 2: (Updating): $\forall\, i = 1\ to\ N$

Update the target vector, $\mathbf{x}_i$ using the following operations.

a) Mutation: choose 3 members randomly between 1 and N, all three different among themselves and also different from i to generate a mutant vector as follows

$$y_i = \mathbf{x}_{n1} + F(\mathbf{x}_{n2} - \mathbf{x}_{n3})$$

b) Crossover: The trial vector is formed as follows

$$z_{j,i} = \begin{cases} y_{j,i} & if\ rand_j \leq CR\ or\ j = jrand \\ \mathbf{x}_{j,i} & o.w \end{cases}$$

where $CR \in [0,1]$ and $1 \leq jrand \leq D$

c) Selection: Fitness of parents and mutants are evaluated for next generation. Parents and mutants are sorted according to their fitness values.

$$\mathbf{x}_i = \begin{cases} y_{g,i} & if\ f_{xi} \leq f_{yi} \\ \mathbf{x}_{g,i} & o.w \end{cases}$$

where f_{xi} and f_{yi} are the fitness values of parent $\mathbf{x}_{g,i}$ and mutant $\mathbf{y}_{g,i}$.

Step 3: (Stopping Criterion)

The algorithm stops if the number of generations or a desired fitness reaches, else the algorithm goes to step 2.

4.3 NUMERICAL APPLICATIONS

In this section, the methodology described above is applied to two main problems including the biochemical reaction model and the HIV infection model of CD4$^+$T cells, to illustrate its effectiveness and viability. These problems are of practical importance therefore the investigation of their solutions has been considered by many authors. Since both these problems do not have exact solution, hence many approximate analytical and numerical methods have been utilized. But the literature survey reveals that nobody as yet has attempted to solve these problems using the stochastic solver based on polynomial basis and EAs presented in this dissertation.

In order to show the efficacy and reliability of the suggested method, comparisons of the numerical solutions are made with the RK4 and some well known classical methods.

4.3.1 BIOCHEMICAL REACTION MODEL

We consider the well-known Michaelis-Menten biochemical reaction model given by [133], [134], [135], [136], [137].

$$E + A \rightleftharpoons Y \rightarrow E + X \tag{4.15}$$

where E is the enzyme, A the substrate, Y the intermediate complex and X the product. The time evolution of scheme (4.15) can be determined from the solution of the system of CNODEs [133], [134], [135], [136], [137].

$$\frac{dA}{dt} = -k_1EA + k_{-1}Y \tag{4.16}$$

$$\frac{dE}{dt} = -k_1EA + (k_{-1} + k_2)Y \tag{4.17}$$

$$\frac{dY}{dt} = k_1EA - (k_{-1} + k_2)Y \tag{4.18}$$

$$\frac{dX}{dt} = k_2Y \tag{4.19}$$

subject to the initial conditions

$$A(0) = A_0, \qquad E(0) = E_0, \qquad Y(0) = 0 \;\; and \;\; X(0) = 0 \tag{4.20}$$

where the parameters k_1, k_{-1} and k_2 are positive rate constants for each reaction. The system of CNODEs (4.15) – (4.19) can be reduced to only two equations for A and Y and in dimensionless form of concentrations of substrate, x, and intermediate complex between enzyme and substrate, y, [133], [134], [135], [136], [137].

$$\frac{dx}{dt} = -x + (\beta - \alpha)y + xy \tag{4.21}$$

$$\frac{dy}{dt} = \frac{1}{\varepsilon}(x - \beta y - xy) \qquad (4.22$$

subject to initial conditions

$$x(0) = 1, \qquad y(0) = 0 \tag{4.23}$$

where α , β , and ε are dimensionless parameters.

The reader may refer [133] and references therein for a detailed mathematical formulation of equations (4.21) and (4.23).

Our aim is solve the system of CNODEs given by (4.21) and (4.22) with the initial conditions (4.23) and obtain the approximate numerical solutions $\hat{x}(t)\, and\, \hat{y}(t)$. The values of dimensionless parameters are taken as $\alpha = 0.3752$, $\beta = 1.0$, $\varepsilon = 0.1$, for a direct comparison with the reported numerical results by some classical methods including HPM [134], VIM [135], and modified Picard-Padé method (PPM) [137].

In view of the suggested method with m = 7 the approximate solutions $\hat{x}(t)$ *and* $\hat{y}(t)$ and their first derivatives $\frac{d\hat{x}(t)}{dt}$ *and* $\frac{d\hat{y}(t)}{dt}$ can be expressed as follows.

$$\hat{x}(t) = \sum_{i=0}^{7} a_i\, t^i \qquad (4.24)$$

$$\hat{y}(t) = \sum_{j=0}^{7} b_j\, t^j \qquad (4.25)$$

$$\frac{d\hat{x}(t)}{dt} = \sum_{i=0}^{7} a_i\, i\, t^{i-1} \qquad (4.26)$$

$$\frac{d\hat{y}(t)}{dt} = \sum_{j=0}^{7} b_j\, j\, t^{j-1} \qquad (4.27)$$

where $(a_0, a_1, \dots, a_7 \; and \; b_0, b_1, \dots, b_7)$ are real valued unknown parameters to be determined.

Using the given initial conditions (4.23) in (4.24) and (4.25) we get $a_0 = x(0) = 1$ and $b_0 = y(0) = 0$. The remaining unknown parameters $(a_1, a_2, \dots, a_7 \; and \; b_1, b_2, \dots b_7)$ that need that need to be tailored is 14. The values of these unknown parameters are obtained by formulating the fitness function ε_j and then applying EAs such as GA and DE for the minimization of ε_j. The numerical solutions are found in the interval $0 \leq t \leq 1$ with a time step of 0.1, therefore ε_j is developed as given below.

$$\varepsilon_1 = \frac{1}{10}\sum_{i=1}^{11}\left(\frac{d\hat{x}(t_i)}{dt} + \hat{x}(t_i) - (1.0 - 0.3752)\hat{y}(t_i) - \hat{x}(t_i)\hat{y}(t_i)\right)^2 \qquad (4.28)$$

$$\varepsilon_2 = \frac{1}{11}\sum_{i=1}^{11}\left(\frac{d\hat{y}(t_i)}{dt} - \frac{1}{0.375}\big(\hat{x}(t_i) - 1.0\hat{y}(t_i) - \hat{x}(t_i)\hat{y}(t_i)\big)\right)^2 \qquad (4.29)$$

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \qquad (4.30)$$

where $\hat{x}(t), \hat{y}(t)$, $\frac{d\hat{x}(t)}{dt}$ *and* $\frac{d\hat{y}(t)}{dt}$ are given by (4.24) – (4.27) respectively.

The GA and DE are executed with the parameter values and settings as prescribed in Table 1 and Table 2 for the minimization of (4.30) in order to find the optimal values of the unknown parameters. The size of chromosome is chosen equal to the total number of unknown parameters which is 14.

The optimal values of the unknown parameters corresponding to one of the minimum fitness achieved by GA and DE are provided in Table 4.3 and Table 4.4 respectively.

Table 4.1 Parameter settings for GA for biochemical reaction model

Parameter Name	Setting
Population creation function	Uniform
Fitness scaling function	Proportional
Selection function	Stochastic uniform
Mutation function	Adaptive feasible
Crossover function	Heuristic

The approximate numerical solutions $\hat{x}(t)$ and $\hat{y}(t)$ of the biochemical reaction model (3.21) and (3.22) are obtained by using the optimal values of unknown parameters from Tables 4.3 and Table 4.4 in (4.5) and (4.6).

Table 4.2 Parameter values of GA and DE

Parameter	Value	
	GA	DE
Population size	400	400
Chromosome size	14	14
No. of generations	1000	1000
Crossover fraction	0.8	-
Crossover constant (CR)	-	0.9
Amplification factor (F)	-	0.3

Table 4.3 Optimal values of unknown parameters achieved by GA for various values of m

index (i)	$m = 3$		$m = 5$		$m = 7$	
	a_i	b_i	a_i	b_i	a_i	b_i
1	-0.8353	2.9369	-0.6916	5.4836	-0.7514	7.2345
2	0.7355	-5.3780	1.9908	-22.4917	3.4798	-44.6143
3	-0.2613	2.8396	-3.5280	41.7696	-11.1921	145.3710
4	---	---	2.9292	-35.9838	20.5707	-271.3640
5	---	---	-0.9192	11.6568	-21.5824	290.5740
6	---	---	---	---	12.0233	-165.7713
7	---	---	---	---	-2.7582	39.0119

Table 4.4 Optimal values of unknown parameters achieved by DE for different values of *m*

index (i)	$m=3$		$m=5$		$m=7$	
	a_i	b_i	a_i	b_i	a_i	b_i
1	-0.8354	2.9369	-0.6928	5.4836	-0.6534	5.8836
2	0.7359	-5.3780	1.9990	-22.4926	2.0945	-26.8811
3	-0.2616	2.8396	-3.5457	41.7737	-3.9277	57.4274
4	---	---	2.9446	-35.9897	4.4977	-56.1427
5	---	---	-0.9238	11.6595	-4.4623	12.9218
6	---	---	---	---	3.2402	15.1704
7	---	---	---	---	-0.9993	-7.9352

The approximate numerical solutions $\hat{x}(t)$ and $\hat{y}(t)$ obtained by the proposed method are presented in Table 4.5 and Table 4.6 respectively, also numerical solutions obtained by RK4, HPM, VIM, and PP are given for comparison.

Table 4.5 Comparison of numerical solution for *x(t)*

		Proposed Method		**Classical Methods**		
t	**RK4**	**GA**	**DE**	**HPM**	**VIM**	**PPM**
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
0.1	0.947400	0.950319	0.952082	0.950847	0.927545	0.947400
0.2	0.925070	0.926115	0.927634	1.254025	0.684203	0.925068
0.3	0.906893	0.907912	0.914159	5.389823	-0.121830	0.906858
0.4	0.889444	0.890244	0.903458	28.827507	-2.047393	0.889234
0.5	0.872255	0.872766	0.890432	114.931424	-5.883123	0.871539
0.6	0.855261	0.855848	0.872700	358.179651	-12.669723	0.853491
0.7	0.838455	0.839271	0.85054	934.695192	-23.669732	0.834879
0.8	0.821837	0.822619	0.826642	2140.089715	-40.302786	0.815531
0.9	0.805407	0.805998	0.805171	4434.619833	-64.052391	0.795307
1.0	0.789166	0.789671	0.789640	8495.655924	-96.352183	0.774092

Table 4.6 Comparison of numerical solution for y*(t)*

		Proposed Method		Classical Methods		
t	RK4	GA	DE	HPM	VIM	PPM
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.1	0.421519	0.398289	0.371506	0.382724	0.648910	0.421520
0.2	0.473657	0.473996	0.476073	-3.241408	3.126695	0.473650
0.3	0.476763	0.475826	0.482292	-50.263656	11.166230	0.476676
0.4	0.473092	0.472756	0.472026	-316.226688	28.767390	0.472754
0.5	0.468437	0.469874	0.46989	-1293.903524	59.853051	0.467684
0.6	0.463611	0.463852	0.471657	-4057.909720	107.190086	0.462402
0.7	0.458731	0.457704	0.46759	-10612.848782	170.815371	0.457180
0.8	0.453813	0.454513	0.456699	-24324.990805	246.206781	0.452164
0.9	0.448860	0.449759	0.447926	-50437.484352	322.439191	0.447436
1.0	0.443875	0.441936	0.444256	-96669.101562	380.565477	0.443037

Further to illustrate the validity and accuracy of the proposed solutions, comparisons of absolute errors computed relative to the RK4 are given in Table 4.7. The accuracy and effectiveness of the proposed method is quite evident from the comparisons, one can clearly see that the proposed method gives approximate solutions that are in good agreement with RK4, while the classical methods including HPM and VIM utterly diverge after $t = 0.1$. The comparison also shows that our results are quite comparable to the results obtained by PPM.

The influence of different number of basis functions (i.e. change in m) on the accuracy of the approximate solution and convergence of the evolutionary algorithms GA and DE are analyzed next, to demonstrate the effectiveness and reliability of the proposed method.

We used $m = 3, 5$ for evaluating the performance, therefore the number of unknown parameters to be tailored are 6 and 10 respectively. Without any other change in the parameter values except the chromosome size 6 and 10 for $m = 3$ and 7 respectively, GA and DE are executed to acquire the unknown parameters. The optimal values of the unknown parameters acquired by GA and DE are given in Table 4.3 and Table 4.4 respectively.

Table 4.7 Comparison of absolute errors for biochemical reaction model

		Proposed Method		Classical Methods		
	t	**GA**	**DE**	**HPM**	**VIM**	**PPM**
x(t)	0.0	0	0	0	0	0
	0.1	2.92E-03	4.68E-03	3.45E-03	1.99E-02	9.06E-08
	0.2	1.05E-03	2.57E-03	3.29E-01	2.41E-01	1.83E-06
	0.3	1.02E-03	7.27E-03	4.48E+00	1.03E+00	3.54E-05
	0.4	8.00E-04	1.40E-02	2.79E+01	2.94E+00	2.10E-04
	0.5	5.11E-04	1.82E-02	1.14E+02	6.76E+00	7.16E-04
	0.6	5.87E-04	1.74E-02	3.57E+02	1.35E+01	1.77E-03
	0.7	8.16E-04	1.21E-02	9.34E+02	2.45E+01	3.58E-03
	0.8	7.83E-04	4.81E-03	2.14E+03	4.11E+01	6.31E-03
	0.9	5.91E-04	2.35E-04	4.43E+03	6.49E+01	1.01E-02
	1.0	5.05E-04	4.74E-04	8.50E+03	9.71E+01	1.51E-02
y(t)	0.0	0	0	0	0	0
	0.1	2.32E-02	5.00E-02	3.88E-02	2.27E-01	6.28E-08
	0.2	3.39E-04	2.42E-03	3.72E+00	2.65E+00	7.31E-06
	0.3	9.38E-04	5.53E-03	5.07E+01	1.07E+01	8.70E-05
	0.4	3.36E-04	1.07E-03	3.17E+02	2.83E+01	3.38E-04
	0.5	1.44E-03	1.45E-03	1.29E+03	5.94E+01	7.53E-04
	0.6	2.40E-04	8.05E-03	4.06E+03	1.07E+02	1.21E-03
	0.7	1.03E-03	8.86E-03	1.06E+04	1.70E+02	1.55E-03
	0.8	7.00E-04	2.89E-03	2.43E+04	2.46E+02	1.65E-03
	0.9	8.99E-04	9.34E-04	5.04E+04	3.22E+02	1.42E-03
	1.0	8.99E-04	9.34E-04	5.04E+04	3.22E+02	1.42E-03

In Fig. 4.1 and Fig 4.2 we provide the approximate solutions *x(t)* and *y(t)* obtained using the proposed method with m = 3,5, also solutions using RK4 and with m = 7 by the proposed method are shown for the comparison.

It is evident from Fig. 4.1 and Fig 4.2 that the accuracy of approximate solutions improves with an increase in m (number of basis functions). In Table 4.8 we also give a comparison of average absolute errors yielded by the proposed method in the interval [0,1] for m = 3, 5, 7 and the number of generations taken for achieving the desired minimum fitness by GA and DE.

From Table 4.8 it is seen that the proposed method yields improved accuracy with increase in m but at the cost of large number of generations consequently high computational cost. Nonetheless the proposed method provides the solution of the biochemical reaction model with better accuracy even with m = 3 as compared to popular classical methods HPM and VIM which proves the effectiveness and reliability of our method.

Table 4.8 Effect of change in number of basis functions *(m)*

m	**Number of Generations**	**Average Absolute Errors**	
		$\lvert x_{RK4} - \hat{x} \rvert$	$\lvert y_{RK4} - \hat{y} \rvert$
3	294	1.095E-01	5.561E-02
5	517	6.542E-03	1.135E-02
7	1000	8.707E-04	2.826E-03

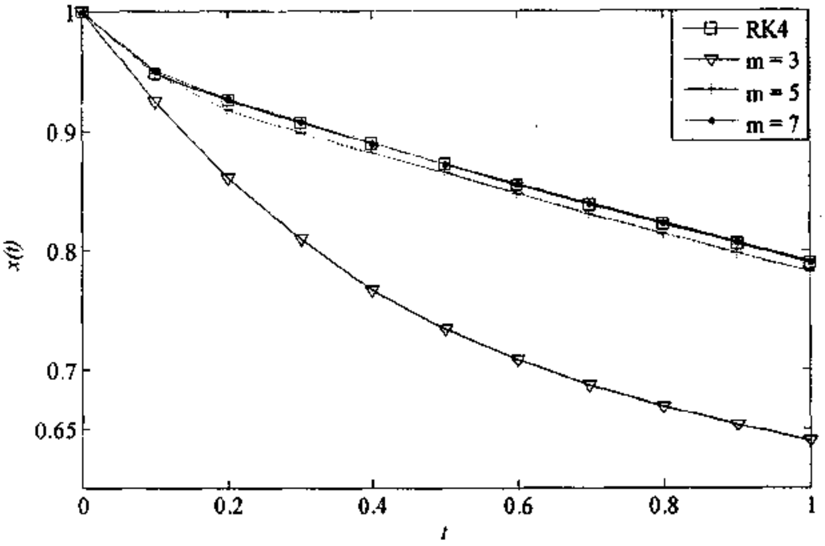

Fig. 4.1 Effect of change in m on approximate solution *x(t)* and comparison with RK4

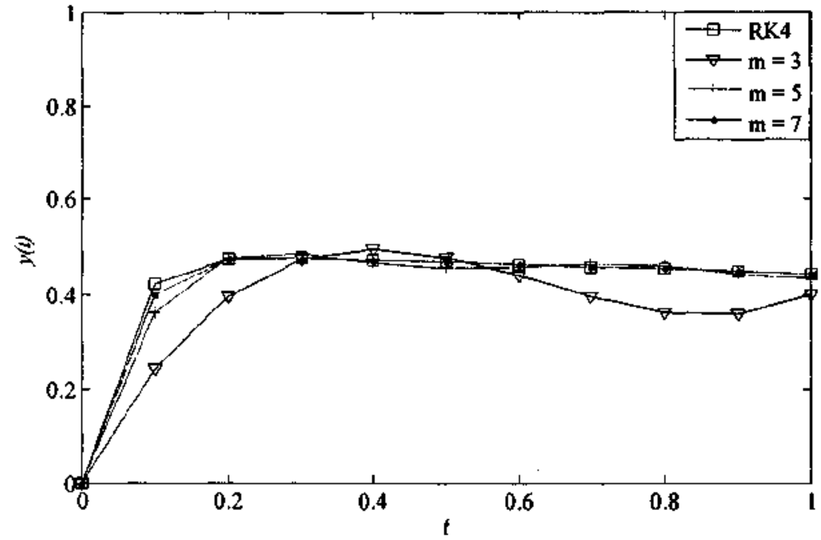

Fig. 4.2 Effect of change in m on approximate solution *y(t)* and comparison with RK4

4.3.2 HIV INFECTION MODEL OF CD4$^+$T CELLS

We shall now apply the suggested method to the numerical solution of the HIV infection model of CD4$^+$T cells [138], which is governed by the following system of CNODEs [139], [140], [141], [142], [143], [144].

$$\begin{aligned}
\frac{dT}{dt} &= p - \alpha T + rT\left(1 - \frac{T+I}{T_{max}}\right) - kVT \\
\frac{dI}{dt} &= kVT - \beta I \\
\frac{dV}{dt} &= N\beta I - \gamma V
\end{aligned} \qquad (4.31)$$

subject to the initial conditions

$$T(0) = T_0, \qquad I(0) = I_0, \qquad and\ V(0) = V_0 \qquad (4.32)$$

In the model of HIV (4.31), the dependent variables T(t), I(t), and V(t) denote the concentration of susceptible CD4$^+$T cells, the concentration of CD4$^+$T cells infected by HIV viruses, and free HIV virus particles in the blood respectively. The parameters α, β, and γ denote the natural turnover rates of uninfected T cells, infected T cells, and virus particles respectively. The term $\left(1 - \frac{T+I}{T_{max}}\right)$ describes the logistic growth of the healthy CD4$^+$T cells, while the proliferation of the infected CD4$^+$T cells is neglected. The term kVT describes the incidence of HIV infection of healthy CD4$^+$T cells, where k > 0 is the infection rate. Each infected CD4$^+$T cell is assumed to produce *N* virus particles during its lifetime, including any of its daughter cells. The body is believed to produce CD4$^+$T cells from precursors in the bone marrow and thymus at a constant rate p. T cells multiply through mitosis with a rate r when they are stimulated by antigen or mitogen. T_{max} denotes the maximum leve' of CD4^{+}T cell concentration in the body [141], [142], [143], [144].

The numerical solution of the HIV infection model of CD4^{+}T cells has been obtained by several authors using several different standard methods such as HPM by Merdan [139], LADM by Ongun [145], multi-stage VIM by Merdan et al. [146], Bessel collocation method (BCM) by Yüzbaş [147], VIM by Merdan et al. [140], and HAM by Goreishi et al. [148].

To apply the proposed method to the HIV infection model of CD4+T cells (4.31) we take m = 5, therefore approximate solutions $\hat{T}(t), \hat{I}(t), and\ \hat{V}(t)$ and their first derivates $\frac{d\hat{T}}{dt}, \frac{d\hat{I}}{dt}, and\ \frac{d\hat{V}}{dt}$ can be expressed as follows.

$$\hat{T}(t) = \sum_{i=0}^{5} a_i\, t^i \qquad (4.33)$$

$$\hat{I}(t) = \sum_{j=0}^{5} b_j\, t^j \qquad (4.34)$$

$$\hat{V}(t) = \sum_{k=0}^{5} c_k\, t^k \qquad (4.35)$$

$$\frac{d\hat{T}(t)}{dt} = \sum_{i=0}^{5} a_i\, i\, t^{i-1} \qquad (4.36)$$

$$\frac{d\hat{I}(t)}{dt} = \sum_{j=0}^{5} b_j\, j\, t^{j-1} \qquad (4.37)$$

$$\frac{d\hat{V}(t)}{dt} = \sum_{k=}^{5} c_k\, k\, t^{k-1} \qquad (4.38)$$

The approximate numerical solution of the HIV infection model of CD4^{+} T cells (4.31) is obtained in the interval $0 \le t \le 1$, with initial conditions $T(0) = 0.1$, $I(0) = 0.0, \mathit{and}\ V(0) = 0.1$, also setting $p = 0.1$, $\alpha = 0.02$, $\beta = 0.3$, $r = 3$, $\gamma = 2.4$, $k = 0.0027$, $T_{max} = 1500$, and $N = 10$ in (4.31) for a direct comparison of our results with the results reported by other methods used in [139], [140], [147].

From the given initial conditions we get $a_0 = T(0) = 0.1$, $b_0 = I(0) = 0.0$, $\mathit{and}\ c_0 = V(0) = 0.1$.

Now formulating the fitness function (ε_j) as follows

$$\varepsilon_1 = \frac{1}{11}\sum_{i=1}^{11}\left(\frac{dT(t_i)}{dt} - 0.1 + 0.02T(t_i) - 3T(t_i)\left(1 - \frac{T(t_i) + I(t_i)}{1500}\right) + 0.0027V(t_i)T(t_i)\right)^2 \qquad (4.39)$$

$$\varepsilon_2 = \frac{1}{11}\sum_{i=1}^{11}\left(\frac{dI(t_i)}{dt} - 0.0027V(t_i)T(t_i) + 0.3I(t_i)\right)^2 \qquad (4.40)$$

$$\varepsilon_3 = \frac{1}{11}\sum_{i=1}^{11}\left(\frac{dV(t_i)}{dt} - 10 * 0.3I(t_i) + 2.4V(t_i)\right)^2 \qquad (4.41)$$

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \qquad (4.42)$$

The FF given by (4.42) is subject to minimization by using GA, PS, IPA, and memetic algorithm schemes GA-IPA, and GA-ASA for acquiring the unknown parameters $(a_1, \ldots a_5; b_1, \ldots b_5; c_1, \ldots, c_5)$. The algorithms are executed according to the prescribed settings and values given in Table 4.9 for GA and in Table 4.10 for IPA and ASA respectively, to achieve the optimal values of the unknown parameters.

Table 4.9 Parameter values and settings of GA for HIV model

Parameter	**Value/Setting**
Population size	[240 240]
Chromosome size	15
Fitness scaling function	Proportional
Selection function	Stochastic uniform
Crossover function	Heuristic
Mutation function	Adaptive feasible
Reproduction crossover fraction	0.8
Generations	2000
Function tolerance	1E-18
Nonlinear constraint tolerance	1E-18

The optimal values of the unknown parameters achieved by GA, GA-IPA, and GA-ASA are provided in Table 4.11, while the values of these unknown parameters achieved by IPA and ASA are given in Table 4.12.

Once we have achieved the optimal values of unknown parameters the approximate solutions $\hat{T}(t), \hat{I}(t), and\ \hat{V}(t)$ can be obtained easily by using these values in (4.33) – (4.35) respectively.

The approximate numerical solutions obtained by the proposed method with memetic algorithm scheme GA-IPA are shown in Fig. 4.3, Fig. 4.4, and Fig. 4.5 for $T(t)$, I(t), and $V(t)$ respectively, also numerical solutions using RK4 are shown for the purpose of comparison. From the comparison the numerical solutions obtained by the proposed method are found in a good agreement with RK4.

Table 4.10 Parameter values and settings of IPA and ASA for HIV model

	Value/Setting	
Parameter	**IPA**	**ASA**
Start point	Random/optimal chromosome from GA	Random/optimal chromosome from GA
Maximum iterations	200	200
Maximum function evaluations	60000	60000
Maximum perturbation	0.1	0.1
Function tolerance	1E-18	1E-18
Nonlinear constraint tolerance	1E-18	1E-18
Derivative type	Central differences	-------
Hessian	BFGS	-------
Subproblem algorithm	Ldl factorization	-------

Table 4.11 Optimal values of unknown parameters for HIV model

index	GA			GA-IPA			GA-ASA		
(i)	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
1	0.4014	0.0001	-0.2399	0.4016	0.0000	-0.2399	0.4016	0.0000	-0.2399
2	0.4355	-0.0008	0.2846	0.4353	0.0003	0.2853	0.4354	0.0003	0.2851
3	1.3520	0.0020	-0.2123	1.3571	-0.0008	-0.2143	1.3569	-0.0009	-0.2133
4	-0.9844	-0.0020	0.0977	-0.9905	0.0009	0.0995	-0.9902	0.0010	0.0984
5	1.2734	0.0007	-0.0209	1.2791	-0.0004	-0.0216	1.2790	-0.0004	-0.0211

Table 4.12 Optimal values of unknown parameters acquired by IPA and ASA for HIV model

index	IPA			ASA		
(i)	a_i	b_i	c_i	a_i	b_i	c_i
1	0.40159	0.00001	-0.23993	0.40158	0.00001	-0.23990
2	0.43529	0.00026	0.28535	0.43539	0.00028	0.28500
3	1.35713	-0.00077	-0.21428	1.35681	-0.00077	-0.21318
4	-0.99052	0.00089	0.09954	-0.99014	0.00085	0.09827
5	1.27910	-0.00035	-0.02158	1.27896	-0.00032	-0.02108

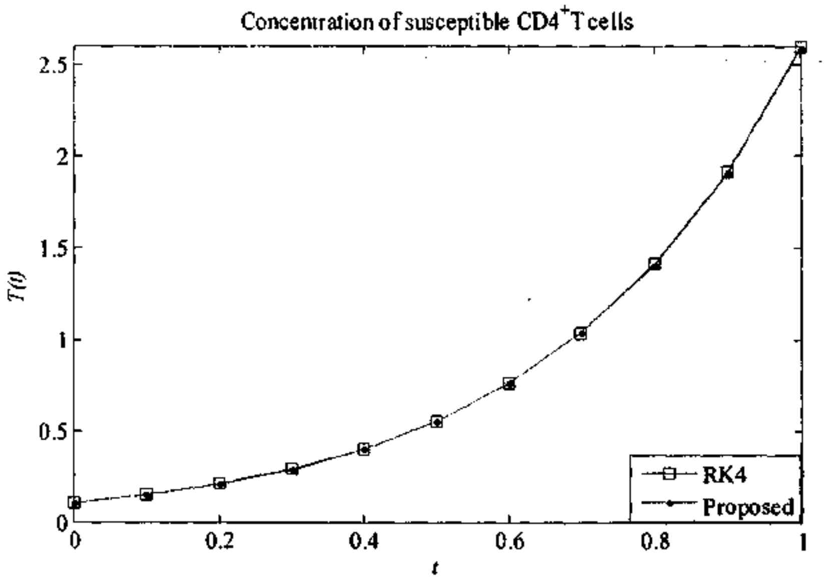

Fig. 4.3 Comparison of numerical solutions for T(t)

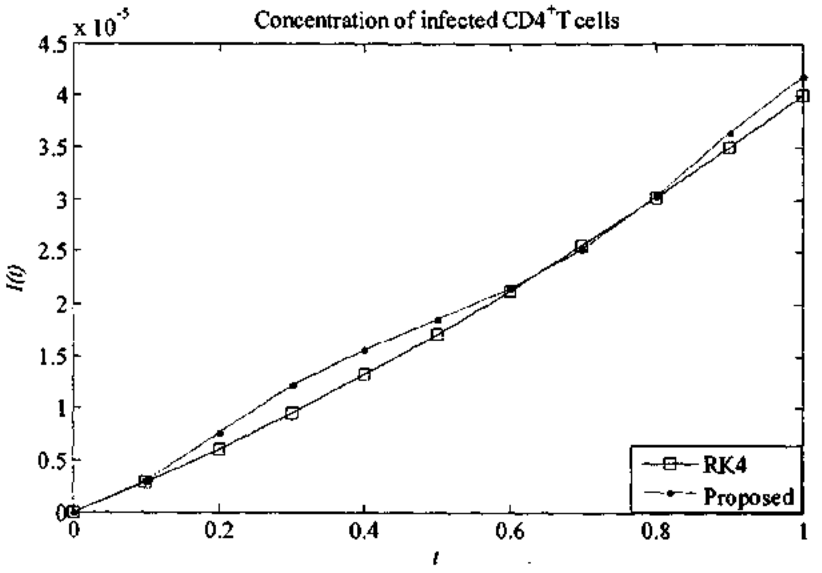

Fig. 4.4 Comparison of numerical solutions for I(t)

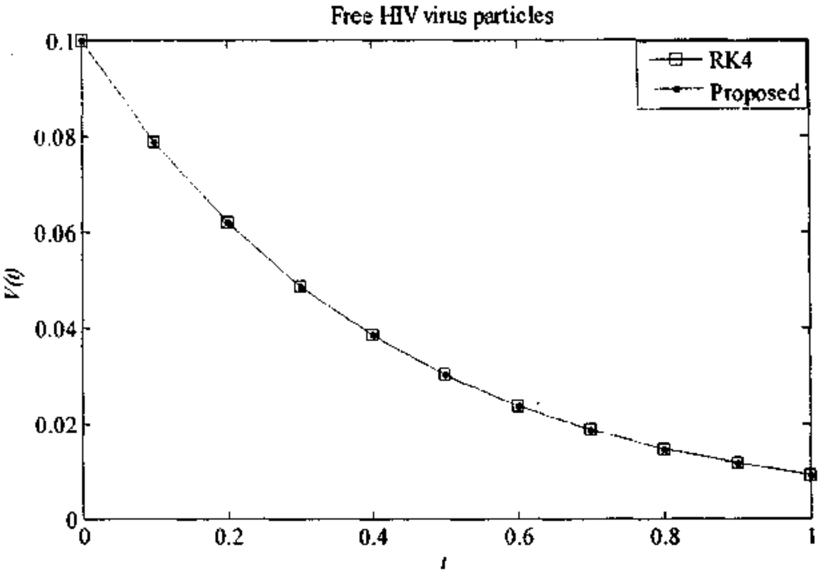

Fig. 4.5 Comparison of numerical solutions for V(t)

Moreover in Table 4.13, Table 4.14, and Table 4.15 a comparison of numerical solutions obtained by the proposed method are made with VIM, BCM, and HPM. To show the accuracy of our results absolute errors obtained by the proposed method and VIM, BCM, and HPM are computed relative to RK4 and provided in Table 4.16.

Table 4.13 Comparison of numerical solutions for *T(t)*

t	**RK4**	**Proposed**	**VIM**	**HPM**	**BCM**
0.0	0.1	0.1	0.1	0.1	0.1
0.2	0.40594071	0.40487784	0.40613466	0.40613583	0.38033093
0.4	0.76357978	0.76188751	0.76245304	0.76247621	0.69546238
0.6	1.41195609	1.40812421	1.39788059	1.39808281	1.27596244
0.8	2.58677696	2.58258960	2.50674667	2.50787401	2.38322774
1.0	0.20872956	0.20741048	0.20880732	0.20880733	0.20386166

Table 4.14 Comparison of numerical solutions for I*(t)*

t	RK4	Proposed	VIM	HPM	BCM
0.0	0	0	0	0	0
0.2	6.031510E-06	7.693902E-06	6.032634E-06	6.032706E-06	6.247872E-06
0.4	1.315302E-05	1.561454E-05	1.314878E-05	1.315890E-05	1.293552E-05
0.6	2.121060E-05	2.138633E-05	2.101417E-05	2.123298E-05	2.035267E-05
0.8	3.015178E-05	3.033203E-05	2.795130E-05	3.024270E-05	2.837302E-05
1.0	3.999421E-05	4.187400E-05	2.431562E-05	4.033321E-05	3.690842E-05

Table 4.15 Comparison of numerical solutions for V*(t)*

t	RK4	Proposed	VIM	HPM	BCM
0.0	0.1	0.1	0.1	0.1	0.1
0.2	0.06187985	0.06186666	0.06187995	0.06187995	0.06187992
0.4	0.03829490	0.03829830	0.03830820	0.03830818	0.03829493
0.6	0.02370455	0.02370762	0.02392029	0.02391982	0.02370432
0.8	0.01468036	0.01467245	0.01621705	0.01621234	0.01467957
1.0	0.00910082	0.00910746	0.01608419	0.01605502	0.02370432

The comparison of the absolute errors from Table 4.16 reveals that the proposed technique yields the results of the HIV infection model of CD4^{+}T cells (4.21) with fairly good accuracy. Furthermore from the comparison it is observed that the average absolute errors for T(t) by the proposed method are fairly smaller than BCM and comparable to VIM and HPM while the absolute errors of V(t) yielded by proposed schemes are much smaller than VIM and HPM. However for I(t) our method gives relatively greater absolute errors as compared to VIM and HPM but fairly comparable with BCM. Nonetheless the overall performance of the proposed method is fairly comparable with the standard methods VIM, HPM, and BCM in comparison with RK4.

Table 4.16 Comparison of absolute errors for HIV model

	t	Proposed	VIM	HPM	BCM
T(t)	0.0	0	0	0	0
	0.2	1.32E-03	7.78E-05	7.78E-05	4.87E-03
	0.4	1.06E-03	1.94E-04	1.95E-04	2.56E-02
	0.6	1.69E-03	1.13E-03	1.10E-03	6.81E-02
	0.8	3.83E-03	1.41E-02	1.39E-02	1.36E-01
	1.0	4.19E-03	8.00E-02	7.89E-02	2.04E-01
I(t)	0.0	0	0	0	0
	0.2	1.662E-06	1.124E-09	1.196E-09	2.164E-07
	0.4	2.462E-06	4.240E-09	5.880E-09	2.175E-07
	0.6	1.757E-07	1.964E-07	2.238E-08	8.579E-07
	0.8	1.802E-07	2.200E-06	9.092E-08	1.779E-06
	1.0	1.880E-06	1.568E-05	3.390E-07	3.086E-06
V(t)	0.0	0	0	0	0
	0.2	1.320E-05	1.004E-07	1.004E-07	6.590E-08
	0.4	3.404E-06	1.330E-05	1.328E-05	3.790E-08
	0.6	3.070E-06	2.157E-04	2.153E-04	2.306E-07
	0.8	7.905E-06	1.537E-03	1.532E-03	7.872E-07
	1.0	6.640E-06	6.983E-03	6.954E-03	1.460E-02

4.4 CONCLUSION

A simple and effective stochastic heuristic scheme based on **hybrid** approach of polynomial basis functions and EAs has been suggested for numerically solving system of CNODEs.

The effectiveness of the presented method has been illustrated by numerically solving the nonlinear biochemical reaction model. The comparisons have revealed that the proposed method outperforms some well-known classical methods including HPM and VIM.

The proposed method has also been successfully applied to solve the HIV infection model of CD4$^+$T cells. The approximate numerical solutions by the proposed method are found in a fairly good agreement with RK4. Moreover, it is also established from the comparisons that the proposed method provides approximate solutions that are fairly comparable with some of the classical methods including VIM, HPM, and BCM.

On the basis of numerical results and comparisons, it can be concluded that the proposed method is effective and viable for solving such CNODEs.

4.5 SUMMARY

This chapter provides the detail of the heuristic computation method combining polynomial basis functions and EAs for solving systems of CNODEs. It also presents the investigation of numerical applications of the designed method. The chapter gives the basic idea of fitness function used for transforming the CNODEs into an equivalent optimization problem. The procedural steps of DE for solving the optimization problem are also presented. The designed method has been successfully applied to solve two problems of practical importance including the Michaelis-Menten biochemical reaction model biochemical reaction model and the HIV infection model of CD4$^+$T cells. From the simulation result presented in this chapter it has been illustrated that the proposed method is quite competent and viable for solving systems of CNODEs like HIV and biochemical reaction models.

CHAPTER 5

HYBRIDIZATION OF EXP-FUNCTION METHOD AND NATURE INSPIRED COMPUTING FOR SOLVING NONLINEAR PARTIAL DIFFERENTIAL EQUATIONs

This chapter provides the detail of the heuristic scheme based on the hybridization of the Exp-function method and nature inspired computing for solving NPDEs. The given NPDE is converted into a NODE using a wave transformation variable. The approximate solution of NODE is expressed using the Exp-function method with unknown constants. The unknown constants are achieved by transforming the NODE into an optimization problem. Two popular nature inspired algorithms including GA and PSO are used to solve the optimization problem and to obtain the unknown constants. The numerical applications of the proposed scheme are tested on some well-know NPDEs including the Fisher, Burger-Fisher, and Burger-Huxley equations.

5.1 INTRODUCTION

Nonlinear wave propagation problems appear in numerous engineering and scientific fields including plasma physics, nonlinear optics, fluid dynamics, chemical kinetics, quantum field theory, population models etc. These problems are by and large governed by the systems of nonlinear partial differential equations (NPDEs). The investigation of solitary solutions to such NPDEs has been of great interest to the scientific community. A glance at the literature reveals that a number of effective methods have been reported in past few years, such as tanh-function method [149,[150], [151], Hirota's bilinear method [152], Haar wavelets method (HWM) [153], VIM [154], [155], ADM [156], [157], [158], HPM [159], [160] for handling NPDEs. These methods have their built in deficiencies and limitations and therefore the research community is paying much attention to seek new and efficient methods for solving the crucial systems of NPDEs.

Recently He and Wu [161] introduced a simple and straightforward method known as the Exp-function method for obtaining the generalized solitonary and periodic solutions of nonlinear wave equations. The method has attracted much attention and it has been successfully applied to a wide variety of problems including the Burger-Fisher equation [162], Fisher equation [163], Burger-Huxley equation [164], Burger equation [165], Kawahara equation [166], and many other problem of PDEs and NPDEs see for example [167], [168], [169], [170], [171], [172], [173] and references therein.

In recent years, stochastic solvers based on evolutionary computation (EC) and artificial neural networks (ANNs) have been increasingly used as an alternate tool for solving variety of differential equations. However, there has been comparatively little work reported where EC and ANN has been used for NPDEs. Nonetheless, the strength of these methods for solving PDEs has been illustrated on several problems, for example Samir et al. [174] used ANN based approach for solving Burger-Fisher and Burger-Huxly equations, Puffer et al. [175] used cellular NN for solving some NPDEs including Burger equation, S. He at al. [176] applied NN based technique for solving linear PDEs, and Ramuhalli et al. [177] employed the merger of ANN and finite element method (FEM) for the approximate solution of various PDEs.

In this dissertation, a novel scheme based on the elegant couple of Exp-function method and nature inspired computing (NIC) is proposed for the numerical solution of NPDEs. The literature survey reveals that the couple of Exp-function method and NIC is attempted for the first time to tackle NPDEs numerically. The proposed method has been successfully tested on some popular NPDEs including Fisher equations, Burger-Fisher, and Burger-Huxley equations. The details of the proposed scheme follow in the next section.

5.2 THE EXP-FUNCTION METHOD

To illustrate the basic idea of the Exp-Function method, an NPDE solved by He and Wu in [161] using the same method is presented.

Consider an NPDE given in the following form

$$N(u, u_x, u_t, u_{xx}, u_{tt}, u_{xt}) = 0 \tag{5.1}$$

A transformation variable η is introduced below

$$\eta = kx + \omega t \tag{5.2}$$

The transformation variable converts (5.1) into an NODE of the following form

$$P(u, ku', \omega u', k^2 u'', \ldots.) = 0 \tag{5.3}$$

where k and ω are unknown constants, and prime denotes derivation with respect to η.

According to the Exp-function method, the solution of (5.3) can be expressed in the following form:

$$u(\eta) = \frac{\sum_{n=-c}^{d} a_n \exp(n\eta)}{\sum_{m=-p}^{q} b_m \exp(m\eta)} = \frac{a_c \exp(c\eta) + \ldots\ldots.. + a_{-d} \exp(-d\eta)}{b_p \exp(p\eta) + \ldots\ldots.. + b_{-q} \exp(-q\eta)} \tag{5.4}$$

where c, d, p, and q are unknown positive integers, a_n and b_m are unknown constants.

The values of c and p are determined by balancing the linear term of highest order in (5.3) with the highest order nonlinear term. Similarly the values of d and q are determined by balancing the lowest order of linear and nonlinear terms in (5.3). Next the unknown constants a_n and b_m are determined by substituting (5.4) into (5.3) and equating the coefficients of $\exp(n\eta)$ to zero, which results into a set of algebraic equations with unknown constants. The systems of algebraic equations are solved using some software package like Matlab, Maple or Mathematica for determining the unknown constants a_n and b_m. Consequently the solution of NPDE (5.1) is obtained.

Now consider the following example for the illustration of the method described above.

$$u_t + u^2 u_x + u_{xxx} = 0 \tag{5.5}$$

Using the transformation given by (5.2) leads to the following ODE

$$\omega u' + k u^2 u' + k^3 u''' = 0 \tag{5.6}$$

Assume the solution of (5.6) is given by (5.4). The values of *c and p* are determined by balancing the linear and nonlinear terms of highest and lowest orders in (5.6) as follows from [161] after some calculations.

$$u''' = \frac{c_1 \exp[(7p+c)\eta] + ...}{c_2 \exp[8p\eta] + ...} \tag{5.7}$$

and

$$u^2 u' = \frac{c_3 \exp[(p+3c)\eta] + ...}{c_4 \exp[4p\eta] + ...} = \frac{c_3 \exp[(5p+3c)\eta] + ...}{c_4 \exp[8p\eta] + ...} \tag{5.8}$$

In order to determine the values of c and p balance the highest order of Exp-function in (5.7) and (5.8) yields

$$7p + c = 5p + 3c \tag{5.9}$$

Simplifying (5.9) gives

$$p = c \tag{5.10}$$

Similarly d and q are found by balancing the linear term of lowest order in (5.6)

$$u''' = \frac{....+d_1 \exp[-(7q+d)\eta]}{....+d_2 \exp[-8q\eta]} \tag{5.11}$$

and

$$u^2 u' = \frac{....+d_3 \exp[-(q+3d)\eta]}{d_4 \exp[(-4q\eta)]} = \frac{....+d_3 \exp[-(5q+3d)\eta]}{d_4 \exp[(-8q\eta)]} \tag{5.12}$$

Balancing the lowest order of Exp-function in (5.11) and (5.12) yields

$$-(7q+d) = -(5q+3d) \tag{5.13}$$

which gives

$$q = d \tag{5.14}$$

Once the c, p, d, and q are determined, their values are freely chosen. For instance in [161] authors have set $p = c = 1$ and $d = q = 1$ as a simple case, therefore (5.4) becomes

$$u(\eta) = \frac{a_1 \exp(\eta) + a_0 + a_{-1} \exp(-\eta)}{\exp(\eta) + b_0 + b_{-1} \exp(-\eta)} \tag{5.15}$$

Equation (5.15) is substituted into (5.6) and some software package like Matlab or Maple is utilized to solve the system of algebraic equations as follows [161].

$$\frac{1}{A}\begin{bmatrix} C_3 \exp(3\eta) + C_2 \exp(2\eta) + C_1 \exp(\eta) + C_0 + C_{-1} \exp(-\eta) \\ + C_{-2} \exp(-2\eta) + C_{-3} \exp(4\eta) \end{bmatrix} = 0 \tag{5.16}$$

where

$$A = (\exp(\eta) + b_0 + b_{-1} \exp(-\eta))^4,$$

$$C_3 = \omega a_1 b_0 + k a_1^3 b_0 - k^3 a_0 - \omega a_0 - k a_a^2 a_0 + k^3 a_1 b_0,$$

$$\begin{aligned}C_2 &= 8k^3a_1b_{-1} + 2ka_1^3b_{-1} - 4k^3a_1b_0^2 - 2\omega a_{-1} - 2ka_1a_0^2 + 2\omega a_1b_{-1} + 4k^3a_0b_0 \\ &\quad - 2ka_1^2a_{-1} + 2ka_1^2a_0b_0 + 2\omega a_1b_0^2 - 2\omega a_0b_0 - 8k^3a_{-1},\end{aligned}$$

$$\begin{aligned}C_1 &= \omega a_1b_0^3 + 6\omega a_1b_0b_{-1} - \omega a_0b_0^2 - k^3a_0b_0^2 - 18k^3a_1b_0b_- - 6ka_1a_0a_{-1} + ka_1a_0^2b_0 \\ &\quad - ka_1^3 + 23k^3a_0b_{-1} - \omega a_0b_{-1} - 5\omega a_{-1}b_0 + k^3a_1b_0^3 - 5k^3a_{-1}b_0 + ka_1^2a_{-1}b_0 + 5ka_1^2a_0b_{-1},\end{aligned}$$

$$\begin{aligned}C_0 &= 4\omega a_1b_{-1}^2 - 4ka_1a_{-1}^2 + 32k^3a_{-1}b_{-1} + 4ka_1a_0^2b_{-1} - 32k^3a_1b_{-1}^2 + 4k^3a_1b_0^2b_{-1} \\ &\quad - 4\omega a_{-1}b_{-1} - 4k^3a_{-1}b_0^2 - 4ka_0^2a_{-1} - 4\omega a_{-1}b_0^2 + 4ka_1^2a_{-1}b_{-1} + 4\omega a_1b_0^2b_{-1},\end{aligned}$$

$$\begin{aligned}C_{-1} &= 18k^3a_{-1}b_0b_{-1} - 6\omega a_{-1}b_0b_{-1} - k^3a_{-1}b_0^3 + k^3a_0b_{-1}b_0^2 + \omega a_0b_{-1}^2 - 5ka_0a_{-1}^2 \\ &\quad + 5\omega a_1b_0b_{-1}^2 + \omega a_0b_{-1}b_0^2 - \omega a_{-1}b_0^3 - ka_{-1}a_{-1}^2b_0 - 23k^3a_0b_{-1}^2 - ka_0^2a_{-1}b_0 \\ &\quad + 5k^3a_1b_0b_{-1}^2 + ka_0^3b_{-1} + 6ka_1a_0a_{-1}b_{-1}\end{aligned}$$

$$\begin{aligned}C_{-2} &= 2\omega a_0b_{-1}^2b_0 - 2\omega a_{-1}b_{-1}^2 - 2ka_{-1}^3 + 2ka_1a_{-1}^2b_{-1} + 2\omega a_1b_{-1}^3 - 4k^3a_0b_{-1}^2b_0 \\ &\quad - 2\omega a_{-1}b_0^2b_{-1} + 4k^3a_{-1}b_0^2b_{-1} - 8k^3a_{-1}b_{-1}^2 + 2ka_0^2a_{-1}b_{-1} - 2ka_0a_{-1}^2b_0 + 8k^3a_1b_{-1}^3,\end{aligned}$$

$$C_{-3} = ka_0a_{-1}^2b_{-1} + \omega a_0b_{-1}^3 - ka_{-1}^3b_0 + k^3a_0b_{-1}^3 - \omega a_{-1}b_0b_{-1}^2 - k^3a_{-1}b_0b_{-1}^2.$$

Setting the coefficients of $\exp(n\eta)$ to zero

$$C_3 = 0,\quad C_2 = 0,\quad C_1 = 0,\quad C_0 = 0,\quad C_{-1} = 0,\quad C_{-2} = 0,\quad C_{-3} = 0. \qquad (5.17)$$

The system of (5.17) is solved which gives the unknown constants and consequently the solution of (5.5).

5.3 PROPOSED SCHEME FOR SOLVING NPDEs

In this section, the description of the proposed scheme for solving NPDEs is presented. The scheme is based on the elegant hybridization of Exp-function method with NIC. The given NPDE is converted into a corresponding NODE given by (5.3) using the transformation (5.2). In view of the Exp-function method the solution of (5.3) can be expressed by (5.4). As mentioned above the values of *c, p, d, q* can be freely chosen. The rest of the unknown parameters a_n, b_m, *k*,

ω are acquired by transforming the NODE given by (5.3) into an equivalent global error minimization problem by formulating a problem exclusive fitness function.

Nature inspired algorithms (NIAs) such as GA and PSO are employed to solve the minimization problem and to achieve the unknown parameters. The suggested novel approach for determining the unknown constants and consequently the approximate solution is simple and straightforward. The detail of the scheme is given below.

We consider the transformed NODE given by (5.3) subject to the initial condition given by

$$u(x,0)=f(x) \tag{5.18}$$

We assume that the approximate solution of (5.3) is expressed in the following form in view of the Exp-function method.

$$\hat{u}(\eta)=\frac{a_c\exp(c\eta)+........+a_{-d}\exp(-d\eta)}{b_p\exp(p\eta)+........+b_{-q}\exp(-q\eta)} \tag{5.19}$$

As mentioned above the values of *c, p, d, q* can be freely chosen, also it has been illustrated that $p = c$, and $d = q$, therefore we set their values freely. The rest of the unknown constants existing in (5.19) including ($a_c,....,a_{-d};b_p,....,b_{-q}$; k and ω) are achieved using the application of nature inspired algorithms (NIAs) such as GA and PSO. To apply the NIAs such as GA and swarm intelligence PSO the unknown parameters in (5.19) are chosen as a chromosome/particle. The transformed NODE (5.3) along with the given initial condition (5.18) is converted into a global error minimization problem by developing fitness function (FF) given by

$$\varepsilon_j=\varepsilon_1+\varepsilon_2 \tag{5.20}$$

where j is the generation/iteration index.

The first part in the fitness function represents the mean of the sum of square errors associated with the transformed NODE (5.3) and the second part represents the mean of the sum of square errors associated with the given initial condition (5.18), given, respectively, as follows.

$$\varepsilon_1 = \frac{1}{N \times S} \sum_{i=1}^{N} \sum_{j=1}^{S} P\left(u\left(kx_j + \omega t_i\right), ku'\left(kx_j + \omega t_i\right), k^2 u''\left(kx_j + \omega t_i\right), \ldots\right)^2 \quad (5.21)$$

$$\varepsilon_2 = \frac{1}{S} \sum_{j=1}^{S} \left(u\left(x_j, 0\right) - f\left(x_j\right)\right)^2 \quad (5.22)$$

where N and S are the total number of steps taken in the solution domain of *x* and *t*. The FF given by (5.20) contains unknown constants in the form of a chromosome/particle for the stochastic algorithms such as GA and PSO. The objective is to solve the minimization problem (5.20) and to achieve the optimal chromosome/particle which represents the values of unknown constants ($a_c, \ldots, a_{-d}; b_p, \ldots, b_{-q}$; k and ω). Consequently the approximate solution $\hat{u}(\eta)$ of the given problem is obtained straightforward by using these optimal values in (5.19).

5.4 NATURE INSPIRED OPTIMIZATION ALGORITHMS

In this section, NIAs used throughout the chapter for solving the minimization problem given by the FF (5.20) are introduced. The PSO, GA, and memetic algorithm scheme GA-IPA have been employed for solving the FF and to obtain the unknown constants ($a_c, \ldots, a_{-d}; b_p, \ldots, b_{-q}$; k and ω). The basic procedural steps of these algorithms used for the implementation are given in algorithm 3.1 for GA hybridized with IPA (HGA) and in algorithm 5.1 for PSO respectively.

Algorithm 5.1: Particle Swarm Optimization (PSO)

Step 1: (Initialization)

A population (swarm) of N particles is randomly created. Each particle has n number of elements which represent the number of unknown parameters. Assign values to parameters c_1 and c_2 and set j=j$_{max}$.

Step 2: (Fitness Evaluation)

Fitness of each particle is computed using a problem exclusive fitness function.

Step 3: (Updating velocity and position)

The velocity and position of each particle are updated using the following equations respectively.

$$\mathbf{v}_i^{j+1} = w^j \mathbf{v}_i^j + c_1 \gamma_{1,i}\left(Pbest - \mathbf{x}_i^j\right) + c_2 \gamma_{2,i}\left(Gbest - \mathbf{x}_i^j\right)$$

$$\mathbf{x}_i^{j+1} = \mathbf{x}_i^j + \mathbf{v}_i^{j+1}$$

Step 4a: (Updating Local Bests)

If a particle's current position gives better fitness than its previous best position, then replace Pbest with it and preserve its fitness.

Step 4b: (Updating Global Best)

If any particle gives better fitness than Gbest, replace Gbest with that Pbest and preserve the new fitness of new Gbest.

Step 5: (Stoppage Criteria)

If fitness of Gbest meets criteria or number of iterations is greater than j$_{max}$ then algorithm terminates, else go to step 2.

5.5 NUMERICAL APPLICATIONS

In this section, the proposed scheme described above is tested on some popular systems of NPDEs arising in diverse practical applications of engineering, including Fisher's equations, Burger's-Fisher equation, Huxley equation, and Burger's-Huxley equation. In order to demonstrate the effectiveness and accuracy of the proposed scheme comparisons of the numerical solutions are made with the exact solutions and the solutions obtained by some traditional methods. Moreover, an extensive investigation with the help of simulations is carried to show the viability of the proposed scheme.

5.5.1 FISHER EQUATION

The Fisher equation introduced by Fisher in 1937 [178] is encountered in many applications such as chemical kinetics, tissue engineering, gene propagation, flame propagation, neurophysiology, branching Brownian motion, autocatalytic chemical reactions, and nuclear reactor theory [179], [180], [181].

The Fisher's equations have been paid much attention due to their diverse applications. A glance at the literature reveals that an ample of powerful analytical and numerical methods have been utilized to solve the Fisher equations, such as ADM [182], (VIM) [183], modified VIM (MVIM) [184], variational HPM (VHPM) [179], [185], Differential quadrature method (DQM) [180], DTM [186], HWM [187], optimal homotopy asymptotic method (OHAM) [187], and many others see for example [188], [189], [190], [191], [192], [193], [194] and references therein. But nobody as yet has tackled Fisher equations using the scheme proposed here. We have solved some three different forms of Fisher equations including the generalized Fisher equation as follows.

Example 1: We consider the Fisher's equation of the following form [179], [184]

$$u_t = u_{xx} + 6u(1-u) \tag{5.23}$$

subject to the initial condition

$$u(x,0) = \frac{1}{(1+\exp(x))^2} \tag{5.24}$$

the exact solution is given by

$$u(x,t) = \frac{1}{(1+\exp(x-5t))^2} \tag{5.25}$$

The approximate solution of (5.23) is obtained using the proposed scheme in domain $x,t \in [0,1]$ with a step of 0.05, which means the total number of steps S = 21.

Converting (5.23) into a NODE using the transformation variable $\eta = kx + \omega t$ yields

$$\omega u' = k^2 u'' + 6u(1-u) \tag{5.26}$$

The approximate solution of (5.26) is expressed using Exp-method given by (5.19). As mentioned above the values of *c, d, p*, and *q* can be freely chosen, we set $p = c = 2$ and $d = q = 2$ in (5.19) and we get following trial function

$$\hat{u}(\eta) = \frac{a_2\exp(2\eta) + a_1\exp(\eta) + a_0 + a_{-1}\exp(-\eta) + a_{-2}\exp(-2\eta)}{b_2\exp(2\eta) + b_2\exp(2\eta) + b_0 + a_{-1}\exp(-\eta) + b_{-2}\exp(-2\eta)} \tag{5.27}$$

The unknown constants ($a_2, \ldots, a_{-2}, b_2, \ldots, b_{-2},$ *and* k, ω) in (5.27) are achieved using EAs by transforming (5.26) along with the initial condition (5.24) into a global error minimization problem using a fitness function (ε_j) as follows

$$\varepsilon_1 = \frac{1}{231}\sum_{i=1}^{21}\sum_{j=1}^{11}\begin{pmatrix} \omega u'(kx_j + \omega t_i) - k^2 u''(kx_j + \omega t_i) - \\ 6u(kx_j + \omega t_i)(1 - u(kx_j + \omega t_i)) \end{pmatrix}^2 \tag{5.28}$$

$$\varepsilon_2 = \frac{1}{21}\sum_{j=1}^{11}\left(u(x_j, 0) - \frac{1}{(1+\exp(x_j))^2} \right)^2 \tag{5.29}$$

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \tag{5.30}$$

The FF given by (5.30) is minimized by applying GA, IPA, and GA-IPA for obtaining the optimal values of unknown constants.

The parameter settings used for the implementation of the algorithms, such GA and IPA are given in Table 5.1. The number of unknown constants ($a_2, \ldots, a_{-2}$, $b_2, \ldots, b_{-2}$, *and* k, ω) which need to be tailored is 12, therefore the size of chromosome is chosen as 12. The values of these unknown constants are restricted between -20 and +20. The search algorithms GA, IPA, and GA-IPA are executed to achieve the minimum fitness, with the prescribed parameter settings and values given in Table 5.1.

Table 5.1 Parameter settings and values for GA and IPA for Fisher equation

GA		IPA	
Parameter Name	**Setting/Value**	**Parameter Name**	**Setting/value**
Population size	400	Start point	Random/Optimal values from GA
No. of generations	1000	Maximum iterations	1000
Selection function	Stochastic uniform	Maximum function evaluations	90000
Mutation function	Adaptive feasible	Function tolerance	1e-18
Crossover function	Heuristic	Nonlinear constraint tolerance	1e-18
Crossover fraction	0.8	Hessian	BFGS

The optimal chromosome corresponding to the minimum fitness achieved by the algorithms GA, IPA, and GA-IPA are provided in Table 5.2. The approximate solution $\hat{u}(\eta)$ of the Fisher's equation (5.23) is consequently obtained by using the values of unknown constants from Table 5.2 in (5.27).

In Table 5.3 we present the numerical solutions from the proposed scheme GA-IPA for different values of time t and x, also exact solutions are provided for the comparison purpose.

Table 5.2 Optimal values of unknown constants for Fisher equation

Constant	GA	IPA	GA-IPA
a_2	16.446380	-0.179080	9.007126
a_1	-14.714076	0.787201	-8.006905
a_0	17.292580	-1.154062	1.907914
a_{-1}	-0.002424	0.514752	-0.000004
a_{-2}	0.000309	0.081439	0.000000
b_2	16.446390	-0.387261	9.007126
b_1	18.177991	1.418956	10.007317
b_0	4.310516	-1.502830	-5.098783
b_{-1}	19.870224	0.598411	-4.191039
b_{-2}	17.284500	0.073787	1.907886
k	-1.000016	0.224904	-1.000001
ω	5.000003	-1.125617	4.999999

From the comparison, numerical solutions are found in an excellent agreement with the exact solutions with an average absolute error of 1.21E-07 in the solution domain [0, 1]. Further in Table 5.4 and Table 5.5 we show a comparison of absolute errors at various values of t and x, obtained by the proposed scheme and some well-known classical methods, including VIM [179], VHPM [179], ADM [184], and MVIM [184].

The comparison from Table 5.4 and Table 5.5 reveals that the proposed method yields the numerical solutions of the Fisher's equation (5.23) with remarkably greater accuracy and much accurate than methods ADM, MVIM, VIM, and VHPM.

Table 5.3 Comparison of numerical solutions at different values of time t.

	$t = 0.0$			$t = 0.8$		
x	**Exact** $u(x,t)$	**Proposed scheme (GA-IPA)** $\hat{u}(\eta)$	**Absolute error** $\lvert u(x,t)-\hat{u}(\eta)\rvert$	**Exact** $u(x,t)$	**Proposed scheme (GA-IPA)** $\hat{u}(\eta)$	**Absolute error** $\lvert u(x,t)-\hat{u}(\eta)\rvert$
0.0	0.25000	0.25000	3.93E-07	0.96435	0.96435	2.36E-08
0.1	0.22564	0.22565	3.36E-07	0.96071	0.96071	2.29E-08
0.2	0.20265	0.20265	2.80E-07	0.95672	0.95672	2.17E-08
0.3	0.18110	0.18110	2.27E-07	0.95233	0.95233	1.98E-08
0.4	0.16105	0.16105	1.77E-07	0.94751	0.94751	1.73E-08
0.5	0.14254	0.14254	1.30E-07	0.94223	0.94223	1.40E-08
0.6	0.12556	0.12556	8.86E-08	0.93645	0.93645	9.69E-09
0.7	0.11010	0.11010	5.19E-08	0.93012	0.93012	4.34E-09
0.8	0.09612	0.09612	2.02E-08	0.92320	0.92320	2.22E-09
0.9	0.08355	0.08355	7.38E-09	0.91564	0.91564	1.01E-08
1.0	0.07233	0.07233	3.06E-08	0.90740	0.90740	1.96E-08

Table 5.4 Comparison of absolute errors at t = *0.2* and t = *0.4*

	$t = 0.2$			$t = 0.4$		
x	**Proposed (GA-IPA)**	**ADM**	**MVIM**	**Proposed (GA-IPA)**	**ADM**	**MVIM**
0.0	4.48E-07	7.22E-03	2.69E-03	2.499E-07	5.75E-02	5.01E-02
0.2	3.90E-07	9.89E-03	2.15E-03	2.373E-07	1.61E-01	5.27E-02
0.4	3.19E-07	1.09E-02	1.13E-03	2.137E-07	1.39E-01	4.12E-02
0.6	2.37E-07	1.04E-02	7.38E-04	1.78E-07	1.51E-01	2.25E-02
0.8	1.51E-07	8.50E-03	5.73E-04	1.301E-07	1.43E-01	5.28E-03
1.0	6.57E-08	5.87E-03	9.07E-04	7.128E-08	1.93E-01	4.23E-03

Table 5.5 Comparison of absolute errors at t = 0.1

x	**Proposed scheme (GA-IPA)**	**VIM**	**VHPM**
0.0	2.499E-07	2.07E-05	3.90E-07
0.1	2.373E-07	2.33E-05	4.03E-07
0.2	2.137E-07	2.42E-05	1.02E-06
0.3	1.78E-07	2.35E-05	1.35E-06
0.4	1.301E-07	2.12E-05	1.37E-06
0.5	7.128E-08	1.77E-05	1.09E-06

In order to investigate the reliability of the proposed scheme and also the effect of change in the values of *c, p, d,* and *q* in (5.19) on the accuracy of the approximate solution, we now set $p = c = 1$ and $d = q = 2$ in (5.19), therefore the trial solution can be expressed as follows

$$u(\eta) = \frac{a_1 \exp(\eta) + a_0 + a_{-1} \exp(-\eta) + a_{-2} \exp(-2\eta)}{b_1 \exp(\eta) + b_0 + b_{-1} \exp(-\eta) + b_{-2} \exp(-2\eta)} \qquad (5.31)$$

The unknown constants ($a_1, \ldots, a_{-2}; b_1, \ldots, b_{-2}$; *k and* ω) are achieved using the procedure described above. The heuristic algorithms GA, IPA, and GA-IPA are executed with the same parameter values and settings prescribed in Table 5.1, except a change in the chromosome size now chosen as 10 which determine the number of unknown constants. The unknown constants are achieved and consequently the approximate solution of (5.23).

In Table 5.6 and Table 5.7 we provide the values of unknown constants and the approximate numerical solutions obtained using the proposed scheme respectively.

The comparison with exact solution from Table 5.7 shows that numerical results are in a good agreement with the exact solution and yet accurate than some standard methods such as ADM and MVIM, which confirms the reliability of the proposed scheme. However a considerable raise in absolute error is observed as compared to the case with $p = c = 2$ and $q = d = 2$.

Table 5.6 Optimal values of unknown constants with $p = c = 1$ and $q = d = 2$

Constant	GA	IPA	GA-IPA
a_1	10.736310	4.768013	9.521559
a_0	-13.482789	-6.238337	-12.492168
a_{-1}	6.223437	2.970817	5.958235
a_{-2}	-1.031807	-0.506884	-1.018934
b_1	10.803787	4.799643	9.579296
b_0	-14.809442	-6.849699	-13.658821
b_{-1}	18.279926	8.371573	16.930231
b_{-2}	-4.455646	-2.351191	-4.982365
k	-0.391884	-0.396776	-0.402247
ω	1.964916	1.945464	1.962834

Table 5.7 Comparison of numerical solutions for Fisher equation (example 1)

	$t = 0.2$			$t = 1.0$		
x	**Exact** $u(x,t)$	**Proposed scheme (GA-IPA)** $\hat{u}(\eta)$	**Absolute error** $\lvert u(x,t)-\hat{u}(\eta)\rvert$	**Exact** $u(x,t)$	**Proposed scheme (GA-IPA)** $\hat{u}(\eta)$	**Absolute error** $\lvert u(x,t)-\hat{u}(\eta)\rvert$
0.0	0.53445	0.53481	3.68E-04	0.98666	0.98755	8.92E-04
0.1	0.50545	0.50558	1.31E-04	0.98527	0.98621	9.36E-04
0.2	0.47606	0.47598	8.09E-05	0.98374	0.98471	9.68E-04
0.3	0.44647	0.44622	2.58E-04	0.98205	0.98304	9.85E-04
0.4	0.41687	0.41648	3.96E-04	0.98020	0.98118	9.87E-04
0.5	0.38746	0.38697	4.89E-04	0.97815	0.97912	9.73E-04
0.6	0.35843	0.35789	5.35E-04	0.97589	0.97683	9.39E-04
0.7	0.32998	0.32945	5.33E-04	0.97341	0.97429	8.85E-04
0.8	0.30232	0.30183	4.86E-04	0.97067	0.97148	8.08E-04
0.9	0.27560	0.27521	3.97E-04	0.96766	0.96837	7.06E-04
1.0	0.25000	0.24973	2.71E-04	0.96435	0.96493	5.76E-04

Example 2: We consider the following generalized Fisher equation [180], [181], [184], [185], [186], [187].

$$u_t = u_{xx} + u\left(1 - u^{\alpha}\right) \tag{5.32}$$

subject to the following initial condition

$$u(x,0) = \left\{\frac{1}{2}\tanh\left(-\frac{\alpha}{2\sqrt{2\alpha+4}}x\right)+\frac{1}{2}\right\}^{\frac{2}{\alpha}} \tag{5.33}$$

The exact solution is given by

$$u(x,t) = \left\{\frac{1}{2}\tanh\left\{-\frac{\alpha}{2\sqrt{2\alpha+4}}\left(x-\frac{\alpha+4}{\sqrt{2\alpha+4}}t\right)\right\}+\frac{1}{2}\right\}^{\frac{2}{\alpha}} \tag{5.34}$$

The numerical solution of (5.32) is obtained using the proposed scheme in domain [0, 1], for $\alpha = 3$ and $\alpha = 6$. We assume that approximate solution is given by (5.27) in view of Exp-function method. The corresponding FF of this problem is given as follows.

$$\varepsilon_j = \frac{1}{N \times S}\sum_{i=1}^{N}\sum_{j=1}^{S}\begin{pmatrix} \omega u'(kx_j + \omega t_i) - k^2 u''(kx_j + \omega t_i) \\ - u(kx_j + \omega t_i)(1 - u^{\alpha}(kx_j + \omega t_i)) \end{pmatrix}^2 + \\ \frac{1}{S}\sum_{j=1}^{S}\left(u(x_j, 0) - \left\{ \frac{1}{2}\tanh\left(-\frac{\alpha}{2\sqrt{2\alpha + 4}} x_j \right) + \frac{1}{2} \right\}^{\frac{2}{\alpha}} \right)^2 \tag{5.35}$$

Here we have taken *N, S = 11* for $\alpha = 3$, and for $\alpha = 6$ *N, S = 21* respectively. The minimization problem given by (5.35) is solved using GA, IPA, and GA-IPA to achieve the constants ($a_2,...,a_{-2}$, $b_2,...,b_{-2}$, *and* k, ω) and consequently the approximate solution $\hat{u}(\eta)$ of (5.32).

The algorithms GA, IPA, and GA-IPA are executed with the same parameter settings as prescribed in Table 5.1 and values of the unknown constants are achieved, which are provided in Table 5.8. One can use the values of unknown constants from Table 5.8 in (5.27) and obtain the numerical solution of (5.32) at any value of x and t in the solution domain [0, 1]. In Table 5.9 we provide numerical solutions by the proposed scheme for $\alpha = 3$ and $\alpha = 6$ at different values of *t* and *x*, also exact solutions are presented for the comparison. To further illustrate the accuracy of the proposed scheme comparison of absolute errors obtained by the proposed scheme at various values of time *t* are made with absolute errors obtained using classical methods VIM [185], VHPM [185] and DTM [186] in Table 5.10, ADM [184] and MVIM [184] in Table 5.11 for $\alpha = 3$ and $\alpha = 6$

respectively. Furthermore Table 5.12 shows a comparison of absolute errors between the proposed scheme, HWM [187], and OHAM [187] for $\alpha = 6$ at different values of time t. From the comparisons of numerical solutions and absolute errors it is observed that the proposed scheme yields the approximate solutions of Fisher equation (5.32) with a remarkably great accuracy and in a sharp agreement with the exact solutions as compared to the methods used in [184], [185], [186], [187].

Table 5.8 Optimal values of unknown constants for $\alpha = 3$ and $\alpha = 6$

Constant	$\alpha = 3$			$\alpha = 6$		
	GA	**IPA**	**GA-IPA**	**GA**	**IPA**	**GA-IPA**
a_2	1.956183	2.396043	1.955010	11.842042	-0.776000	10.788171
a_1	3.472735	0.989144	3.475511	16.008676	16.577801	13.547594
a_0	7.935255	0.440256	7.933308	13.109621	10.249838	11.263619
a_{-1}	3.253756	0.758158	3.251794	6.548537	11.892683	5.824778
a_{-2}	0.083236	-0.007743	0.086775	0.265929	0.319823	0.122577
b_2	1.958695	2.387092	1.956615	11.839029	-0.773863	10.787785
b_1	4.824608	1.213233	4.810081	16.088964	16.546023	13.522874
b_0	10.020968	2.578967	10.024919	17.658665	10.202153	14.001583
b_{-1}	8.299114	0.148455	8.279836	11.410827	16.208127	11.013277
b_{-2}	1.410277	0.935976	1.442175	3.194830	6.027368	3.019663
k	-0.967265	-0.569803	-0.960679	-0.786989	-0.734617	-0.705098
ω	2.140420	1.261315	2.126651	1.967286	1.836550	1.762658

Table 5.9 Comparison of numerical solutions (for $\alpha = 3,6$)

	$\alpha = 3$			$\alpha = 6$		
	$t = 0.3$			$t = 0.6$		
x	**Exact** $u(x,t)$	**Proposed scheme (GA-IPA)** $\hat{u}(\eta)$	**Absolute error** $\lvert u(x,t)-\hat{u}(\eta)\rvert$	**Exact** $u(x,t)$	**Proposed scheme (GA-IPA)** $\hat{u}(\eta)$	**Absolute error** $\lvert u(x,t)-\hat{u}(\eta)\rvert$
0.0	0.75228	0.75228	3.29E-08	0.96715	0.96715	1.48E-06
0.1	0.73543	0.73543	2.77E-08	0.96223	0.96222	8.67E-07
0.2	0.71794	0.71794	2.92E-08	0.95663	0.95663	2.43E-07
0.3	0.69987	0.69987	1.15E-08	0.95029	0.95029	3.12E-07
0.4	0.68126	0.68126	7.24E-08	0.94312	0.94312	7.15E-07
0.5	0.66217	0.66217	1.30E-07	0.93507	0.93507	9.03E-07
0.6	0.64266	0.64266	1.67E-07	0.92604	0.92604	8.57E-07
0.7	0.62278	0.62278	1.72E-07	0.91598	0.91598	6.11E-07
0.8	0.60262	0.60262	1.45E-07	0.90482	0.90482	2.46E-07
0.9	0.58224	0.58224	9.81E-08	0.89251	0.89251	1.20E-07
1.0	0.56172	0.56172	5.10E-08	0.87901	0.87901	3.68E-07

Table 5.10 Comparison of absolute errors at $t = 0.1$ (for $\alpha = 3$)

	Proposed Scheme			**Classical Methods**		
x	GA	IPA	GAIPA	VHPM	VIM	DTM
0.0	5.01E-05	4.96E-08	3.85E-06	1.22E-04	3.76E-05	5.23E-05
0.1	5.79E-05	2.06E-09	3.76E-06	1.28E-04	2.58E-05	1.06E-04
0.2	6.57E-05	1.88E-08	3.40E-06	1.32E-04	1.45E-05	2.43E-04
0.3	7.32E-05	7.03E-09	2.66E-06	1.35E-04	4.23E-06	3.81E-04
0.4	7.99E-05	3.21E-08	1.49E-06	1.35E-04	4.90E-06	4.35E-04
0.5	8.58E-05	8.30E-08	1.51E-08	1.33E-04	1.26E-05	3.18E-04

Table 5.11 Comparison of absolute errors (for $\alpha = 6$) at $t = 0.2, 0.4$

	$t = 0.2$			$t = 0.4$		
x	**Proposed (GA-IPA)**	**ADM**	**MVIM**	**Proposed (GA-IPA)**	**ADM**	**MVIM**
0.0	1.01E-06	5.24E-02	4.54E-02	7.09E-07	1.21E-01	1.97E-01
0.2	9.42E-07	7.79E-02	4.17E-02	3.69E-07	2.17E-01	8.39E-02
0.4	8.53E-08	1.10E-01	3.23E-02	4.14E-07	3.41E-01	9.22E-04
0.6	6.7E-07	1.51E-01	1.91E-02	7.52E-07	4.94E-01	4.10E-02
0.8	5.2E-07	1.99E-01	5.03E-03	1.67E-07	6.74E-01	4.10E-02
1.0	1.27E-07	2.55E-01	7.85E-03	8.24E-07	8.78E-01	1.46E-02

Table 5.12 Comparison of absolute errors (for $\alpha = 6$) at $t = 0.4, 0.8$

	$t = 0.4$			$t = 0.8$		
x	**Proposed (GA-IPA)**	**HWM**	**OHAM**	**Proposed (GA-IPA)**	**HWM**	**OHAM**
0.0	7.09E-07	5.48E-03	4.07E-03	5.92E-07	2.76E-03	7.41E-02
0.1	6.40E-07	1.06E-02	2.34E-03	1.04E-06	5.82E-03	6.78E-02
0.2	3.69E-07	1.51E-02	4.31E-04	1.20E-06	8.93E-03	5.81E-02
0.3	2.10E-08	1.86E-02	1.56E-03	1.14E-06	1.17E-02	4.53E-02
0.4	4.14E-07	2.06E-02	3.52E-03	9.94E-07	1.37E-02	3.00E-02
0.5	6.89E-07	2.10E-02	5.34E-03	8.67E-07	1.45E-02	1.31E-02
0.6	7.52E-07	1.94E-02	6.94E-03	8.77E-07	1.40E-02	4.63E-03
0.7	5.66E-07	1.59E-02	8.25E-03	1.12E-06	1.21E-02	2.21E-02
0.8	1.67E-07	1.11E-02	9.23E-03	1.67E-06	9.09E-03	3.84E-02
0.9	3.43E-07	5.58E-03	9.86E-03	2.55E-06	5.58E-03	5.27E-02
1.0	8.24E-07	5.48E-03	4.07E-03	3.71E-06	2.76E-03	7.41E-02

Example 3: We consider the following Fisher equation [180]

$$u_t = u_{xx} + u^2(1-u) \tag{5.36}$$

subject to the following initial condition

$$u(x,0) = \left(\frac{1}{1+\exp\left(\frac{x}{\sqrt{2}}\right)} \right) \tag{5.37}$$

The exact solution is given by

$$u(x,t) = \left(\frac{1}{1+\exp\left(\frac{1}{\sqrt{2}}\left(x - \frac{1}{\sqrt{2}}t\right)\right)} \right) \tag{5.38}$$

The numerical solution of (5.23) is obtained using the proposed scheme in domain [0, 1] with total steps N, S = 11 taken in the solution domain. We assume that approximate solution of (5.36) is given by (5.27). The corresponding FF is accordingly formulated as follows.

$$\begin{aligned} \varepsilon_j = {} & \frac{1}{121}\sum_{i=1}^{11}\sum_{j=1}^{11}\left(\omega u'(kx_j+\omega t_i) - k^2 u''(kx_j+\omega t_i) - u^2(kx_j+\omega t_i)(1-u(kx_j+\omega t_i))\right)^2 \\ & + \frac{1}{11}\sum_{1}^{11}\left(u(x_j,0) - \left(\frac{1}{1+\exp\left(\frac{x_j}{\sqrt{2}}\right)} \right)\right)^2 \end{aligned} \tag{5.39}$$

The minimization problem given by (5.39) is solved using GA, IPA, and GA-IPA to achieve the constants (*a*$a_2,\ldots,a_{-2}$, $b_2,\ldots,b_{-2}$, *and* k, ω). Algorithms are executed with the parameter settings prescribed in Table 5.1.

Table 5.13 Optimal values of unknown constants for Fisher equation (example 3)

Constant	GA	IPA	GA-IPA
a_2	-0.016912	0.000009	-0.001872
a_1	0.082512	3.231307	0.047214
a_0	-5.099919	-0.100825	-3.968665
a_{-1}	-18.600571	1.807064	-0.000854
a_{-2}	-1.135735	0.806983	-1.622407
b_2	-3.807989	3.231250	-4.053147
b_1	-18.660492	3.130570	0.179000
b_0	-7.552964	1.706175	-5.632102
b_{-1}	-18.339570	2.614088	0.039252
b_{-2}	-1.180987	0.806984	-1.626181
k	0.369348	0.707122	0.347313
ω	-0.261127	-0.500008	-0.245582

In Table 5.13 we provide the values of unknown constants acquired by the algorithms GA, IPA, and GA-IPA. Once we have acquired the unknown constants the approximate solution of (5.36) can easily be obtained by using these values in (5.27).

The numerical solutions obtained at various values of time t using the proposed scheme are compared with the exact solution in Table 5.14. Further in Table 5.15 we present comparison of numerical solutions and absolute errors obtained by the proposed scheme with those reported by the DQM used in [180]. From the comparisons of numerical solutions in Table 5.14 and Table 5.15, it can be seen that the proposed scheme gives the approximate solution in an excellent agreement with the exact solutions and fairly accurate than DQM.

Table 5.14 Comparison of numerical solutions for Fisher equation (example 3)

	$t=0.5$			$t=1.0$		
x	**Exact** $u(x,t)$	**Proposed scheme (GA-IPA)** $\widehat{u}(\eta)$	**Absolute error** $\lvert u(x,t)-\widehat{u}(\eta)\rvert$	**Exact** $u(x,t)$	**Proposed scheme (GA-IPA)** $\widehat{u}(\eta)$	**Absolute error** $\lvert u(x,t)-\widehat{u}(\eta)\rvert$
0.0	0.56218	0.56218	1.74E-07	0.62246	0.62246	2.29E-08
0.1	0.54470	0.54470	1.17E-07	0.60570	0.60570	2.88E-08
0.2	0.52712	0.52712	5.69E-08	0.58870	0.58870	8.32E-08
0.3	0.50947	0.50947	5.54E-09	0.57147	0.57147	1.40E-07
0.4	0.49179	0.49179	7.04E-08	0.55408	0.55408	1.99E-07
0.5	0.47413	0.47413	1.38E-07	0.53655	0.53655	2.60E-07
0.6	0.45654	0.45654	2.07E-07	0.51892	0.51892	3.22E-07
0.7	0.43906	0.43906	2.78E-07	0.50126	0.50126	3.86E-07
0.8	0.42173	0.42173	3.51E-07	0.48358	0.48358	4.52E-07
0.9	0.40459	0.40458	4.24E-07	0.46595	0.46595	5.19E-07
1.0	0.38767	0.38767	4.99E-07	0.44841	0.44841	5.87E-07

Table 5.15 Comparison of numerical solutions between the proposed method and DQM

t	x	**Exact** $u(x,t)$	**Proposed** **(GA-IPA)** $\widehat{u}(\eta)$	**DQM**	**Absolute errors**	
					Proposed **(GA-IPA)**	**DQM**
	0.25	0.51830	0.51830	0.51831	2.60E-08	1.23E-05
0.5	0.5	0.47413	0.47413	0.47415	1.38E-07	1.52E-05
	0.75	0.43037	0.43037	0.43039	3.14E-07	1.71E-05
	0.25	0.58011	0.58011	0.58012	1.11E-07	1.04E-05
1.0	0.5	0.53655	0.53655	0.53656	2.60E-07	1.36E-05
	0.75	0.49242	0.49242	0.49243	4.19E-07	1.19E-05

5.5.2 GENERALIZED BURGER-FISHER EQUATION

The generalized Burger-Fisher (B-F) equation is one of the important NPDE which appear in various applications such as fluid dynamics, shock wave formation, turbulence, heat conduction, traffic flow, gas dynamics, sound waves in viscous medium, and some other fields of applied science [195], [196], [197].

The generalized B-F equation is of the form [198], [199]

$$u_t + \alpha u^{\delta} u_x - u_{xx} = \beta u\left(1 - u^{\delta}\right) \qquad \forall\ 0 \le x \le 1,\ t \ge 0 \qquad (5.40)$$

subject to the following initial condition

$$u(x,0) = \left(\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}x\right)\right)^{\frac{1}{\delta}} \qquad (5.41)$$

The exact solution is given by

$$u_{exact}(x,t) = \left(\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}\left(x - \left(\frac{\alpha}{\delta+1} + \frac{\beta(\delta+1)}{\alpha}\right)t\right)\right)\right)^{\frac{1}{\delta}} \qquad (5.42)$$

Many analytical and numerical methods such as OHAM [197], ADM [198], HPM [199], collocation based radial basis functions (RBF) [200], and several others [195], [196], [201], [202], [203], [204], have been utilized for solving B-F equation (5.40).

We obtain the numerical solution of the generalized B-F equation (5.40) using the proposed scheme in the following two examples.

Example 1: Consider the generalized B-F equation (5.40) and apply the transformation variable $\eta = kx + \omega t$ yields the following NODE.

$$\omega u' + \alpha u^{\delta} k u' - k^2 u'' = \beta u\left(1 - u^{\delta}\right) \qquad (5.43)$$

The numerical solution of (5.43) with the initial condition given by (5.41) is obtained in the domain $x \in (0,1)$ and $t \in (0,1)$ for various values of $\alpha, \beta,$ and δ as follows.

Case 1: $\alpha = \beta = 0.001, \delta = 1$

Case 2: $\alpha = \beta = 0.1, \delta = 1$

Case 3: $\alpha = \beta = 0.5, \delta = 1$

Case 4: $\alpha = \beta = 1, \delta = 2$

We assume the approximate solution $\hat{u}(\eta)$ is given by (5.27). The unknown constants ($a_2,\ldots,a_{-2}$, $b_2,\ldots,b_{-2}$, *and* k, ω) in equation (5.27) are achieved using GA by formulating the FF corresponding to each case. For instance the FF corresponding to case 1 is given by

$$\varepsilon_1 = \frac{1}{121}\sum_{i=1}^{11}\sum_{j=1}^{11}\begin{pmatrix} \omega u'(kx_j + \omega t_i) + (0.1)u(kx_j + \omega t_i)ku'(kx_j + \omega t_i) - k^2 u''(kx_j + \omega t_i) \\ -(0.1)u(kx_j + \omega t_i)(1 - u(kx_j + \omega t_i)) \end{pmatrix}^2 \quad (5.44)$$

$$\varepsilon_2 = \frac{1}{11}\sum_{j=1}^{11}\left(u(x_j,0) - \left(\frac{1}{2} + \frac{1}{2}\tanh\left(-\frac{0.1}{4}x_j\right)\right)\right)^2 \quad (5.45)$$

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \quad (5.46)$$

Similarly we formulate FF corresponding to each case defined above. The parameter settings and values used for the implementation of GA are given in Table 5.16. The number of unknown constants *($a_2,\ldots,a_{-2}$, $b_2,\ldots,b_{-2}$, and k, ω)* which need to be tailored is 12, therefore the size of chromosome is chosen as 12. The values of these unknown constants are restricted between -10 and +10. The GA is executed to achieve the minimum fitness, with the prescribed parameter settings and values given in Table 5.16.

Using the values of unknown constants from Table 5.17 in equation (5.27) provides the numerical solution $\hat{u}(\eta)$ of the generalized B-F equation at any value of x and t in the solution domain.

The numerical solutions obtained by the proposed scheme at time $t = 0.1$ are presented in Table 5.18 for case 1 and case 2, and in Table 5.19 for case 3 and case 4 respectively, also exact solutions are given for comparison. It is seen from the comparisons of Table 5.18 and Table 5.19 that numerical solutions obtained by the proposed scheme are in an excellent agreement with the exact solutions with average absolute errors of 1.20E-8, 1.49E-08, 4.40E-07, and 1.85E-06 for case 1, case 2, case 3, and case 4 respectively.

Table 5.18 Comparison of numerical solutions for B-F equation at $t = 0.1$ (for case 1, case2)

	Case 1		**Case 2**		**Absolute errors** $\lvert u_{exact} - \hat{u}(\eta)\rvert$	
x	u_{exact}	**Proposed** $\hat{u}(\eta)$	u_{exact}	**Proposed** $\hat{u}(\eta)$	**Case 1**	**Case 2**
0.0	0.500025	0.500025	0.502562	0.502562	2.236E-08	8.009E-08
0.1	0.500013	0.500012	0.501312	0.501312	1.988E-08	7.001E-08
0.2	0.500000	0.500000	0.500062	0.500062	1.706E-08	5.985E-08
0.3	0.499988	0.499987	0.498813	0.498812	1.390E-08	4.967E-08
0.4	0.499975	0.499975	0.497563	0.497562	1.040E-08	3.954E-08
0.5	0.499963	0.499962	0.496313	0.496313	6.547E-09	2.954E-08
0.6	0.499950	0.499950	0.495063	0.495063	2.354E-09	1.972E-08
0.7	0.499938	0.499938	0.493813	0.493813	2.182E-09	1.018E-08
0.8	0.499925	0.499925	0.492563	0.492563	7.062E-09	9.795E-10
0.9	0.499913	0.499913	0.491313	0.491313	1.228E-08	7.780E-09
1.0	0.499900	0.499900	0.490064	0.490064	1.785E-08	1.601E-08

To show the accuracy of our results in comparison to the numerical solutions of this problem reported in the literature by some classical methods, we provide the comparison of numerical solutions with the exact solutions at various values of t and x in Table 5.20, Table 5.21, Table 5.22, and Table 5.23 for case 1, case 2, case 3, and case 4 respectively. Also the absolute errors obtained by the proposed scheme are compared with the absolute errors obtained by OHAM [197] and ADM [198], in Table 5.20 and Table 5.23, and HPM [199] in Table 5.21 and Table 5.22.

From the comparison of numerical solutions and absolute errors the accuracy of the proposed scheme is quite notable. It is observed from the comparisons that the absolute errors yielded by the proposed scheme are relatively smaller than absolute errors obtained using OHAM [197], ADM [198], and HPM [199], which confirms the effectiveness and efficiency of the proposed scheme.

Table 5.19 Comparison of numerical solutions for B-F equation at $t = 0.1$ (for case 3, case 4)

	Case 3		**Case 4**		**Absolute errors** $\lvert u_{exact} - u(\eta) \rvert$	
x	u_{exact}	**Proposed** $\hat{u}(\eta)$	u_{exact}	**Proposed** $\hat{u}(\eta)$	**Case 3**	**Case 4**
0.0	0.514059	0.514057	0.745203	0.745205	1.669E-06	1.396E-06
0.1	0.507812	0.507811	0.734037	0.734038	1.165E-06	8.651E-07
0.2	0.501562	0.501562	0.722639	0.722640	7.771E-07	3.266E-07
0.3	0.495313	0.495312	0.711024	0.711024	4.836E-07	2.146E-07
0.4	0.489064	0.489064	0.699207	0.699206	2.670E-07	7.568E-07
0.5	0.482819	0.482819	0.687205	0.687204	1.123E-07	1.303E-06
0.6	0.476580	0.476580	0.675035	0.675033	6.852E-09	1.859E-06
0.7	0.470347	0.470347	0.662715	0.662713	5.971E-08	2.436E-06
0.8	0.464124	0.464124	0.650264	0.650261	9.571E-08	3.047E-06
0.9	0.457912	0.457912	0.637701	0.637698	1.074E-07	3.704E-06
1.0	0.451713	0.451714	0.625046	0.625042	9.900E-08	4.418E-06

Table 5.20 Comparison of numerical solutions and absolute errors for $\alpha = \beta = 0.001$ and $\delta = 1$

x	t	u_{exact}	Proposed	Absolute errors		
			$\hat{u}(\eta)$	**Proposed**	**ADM**	**OHAM**
0.1	0.001	0.499988	0.499988	1.97E-08	1.94E-06	2.25E-08
	0.005	0.499989	0.499989	1.97E-08	9.69E-06	1.12E-07
	0.01	0.499990	0.499990	1.97E-08	1.94E-06	2.25E-07
0.5	0.001	0.499938	0.499938	3.58E-09	1.94E-06	4.58E-08
	0.005	0.499939	0.499939	3.71E-09	9.69E-06	2.29E-07
	0.01	0.499940	0.499940	3.88E-09	1.94E-06	4.58E-07
0.9	0.001	0.499888	0.499888	1.80E-08	1.94E-06	4.58E-08
	0.005	0.499889	0.499889	1.77E-08	9.69E-06	2.29E-07
	0.01	0.499890	0.499890	1.74E-08	1.94E-06	4.58E-07

Table 5.21 Comparison of numerical solutions and absolute errors for $\alpha = \beta = 0.1$ and $\delta = 1$

t	x	u_{exact}	**Proposed** $\hat{u}(\eta)$	Absolute error	
				Proposed	**HPM**
0.1	0.2	0.500062	0.500062	5.98E-08	4.32E-08
	0.4	0.497563	0.497562	3.95E-08	1.08E-07
	0.6	0.495063	0.495063	1.97E-08	1.74E-07
	0.8	0.492563	0.492563	9.80E-10	2.40E-07
0.4	0.2	0.507749	0.507749	6.75E-08	3.85E-07
	0.4	0.505250	0.505250	4.89E-08	6.65E-07
	0.6	0.502750	0.502750	2.93E-08	1.71E-06
	0.8	0.500250	0.500250	9.08E-09	2.76E-06
0.8	0.2	0.517992	0.517992	5.09E-08	7.28E-06
	0.4	0.515495	0.515495	4.27E-08	3.08E-06
	0.6	0.512997	0.512997	3.09E-08	1.12E-06
	0.8	0.510498	0.510498	1.63E-08	5.32E-06

Table 5.22 Comparison of numerical solutions and absolute errors for $\alpha = \beta = 0.5$ and $\delta = 1$

t	x	u_{exact}	**Proposed** $\tilde{u}(\eta)$	**Absolute error**	
				Proposed	**HPM**
0.1	0.2	0.501562	0.501562	7.77E-07	6.17E-08
	0.4	0.489064	0.489064	2.67E-07	1.60E-05
	0.6	0.476580	0.476580	6.85E-09	2.58E-05
	0.8	0.464124	0.464124	9.57E-08	3.54E-05
0.4	0.2	0.543639	0.543631	7.40E-06	7.87E-05
	0.4	0.531209	0.531205	4.69E-06	7.89E-05
	0.6	0.518741	0.518738	2.95E-06	2.36E-04
	0.8	0.506250	0.506248	1.87E-06	3.92E-04
0.8	0.2	0.598688	0.598635	5.22E-05	1.24E-03
	0.4	0.586618	0.586583	3.48E-05	6.22E-04
	0.6	0.574443	0.574419	2.32E-05	2.80E-06
	0.8	0.562177	0.562161	1.54E-05	6.28E-04

Table 5.23 Comparison of numerical solutions and absolute errors for $\alpha = \beta = 1$ and $\delta = 2$

x	t	u_{exact}	**Proposed** $\tilde{u}(\eta)$	**Absolute errors**		
				Proposed	**ADM**	**OHAM**
0.1	0.0001	0.695266	0.695267	1.08E-06	2.80E-04	1.17E-05
	0.0005	0.695426	0.695427	1.08E-06	1.40E-03	5.87E-05
	0.001	0.695625	0.695626	1.08E-06	2.80E-03	1.17E-04
0.5	0.0001	0.646130	0.646129	1.14E-06	2.69E-04	5.33E-05
	0.0005	0.646297	0.646296	1.14E-06	1.34E-03	1.06E-05
	0.001	0.646506	0.646505	1.14E-06	2.69E-03	1.06E-05
0.9	0.0001	0.595310	0.595306	4.12E-06	2.55E-04	9.29E-06
	0.0005	0.595481	0.595477	4.12E-06	1.27E-03	4.64E-05
	0.001	0.595695	0.595691	4.12E-06	2.55E-03	9.29E-04

Example 2: With $\beta = 0$ and $\alpha = 1$ equation (5.40) is reduced to the generalized Burger's equation [198].

The approximate solution is obtained by the proposed scheme for three different values of $\delta = 1,2,3$ in the domain $x \in (0,1)$ and $t \in (0,2)$ for $\delta = 1,2$, and $t \in (0,5)$ for $\delta = 3$.

We assume the solution is expressed by Exp-function method given by equation (5.27). The fitness function is developed for each value of δ with $\beta = 0$ and $\alpha = 1$. For example the fitness function for $\delta = 3$ is given as follows

$$\varepsilon_j = \frac{1}{121}\sum_{i=1}^{11}\sum_{j=1}^{11}\begin{pmatrix} \omega u'(kx_j + \omega t_i) + u^3(kx_j + \omega t_i)ku'(kx_j + \omega t_i) \\ -k^2 u''(kx_j + \omega t_i) \end{pmatrix}^2 + \frac{1}{11}\sum_{j=1}^{11}\left(u(x_j,0) - \left(\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-3}{8}x_j\right)\right)^{\frac{1}{3}} \right)^2 \quad (5.47)$$

GA is used to solve the minimization problem such as given by equation (5.47) and to obtain the optimal values of unknown constants in (5.27). GA is executed with the parameter settings and values as prescribed in Table 5.16.

The optimal values of unknown constants achieved by GA are given in Table 5.24 for each value of $\delta = 1,2$ and 3. The numerical solutions of generalized Burger's equation are obtained consequently by using the values of unknown constants in equation (5.27).

In Tables 5.25, Table, 5.26, Table 5.27, and Table 5.28 we provide the comparison of numerical solutions for $\delta = 1,2$ and 3 obtained by the proposed scheme with the exact solutions, and the solutions obtained by ADM [198] and RBF [200]. The comparisons of numerical solutions and absolute errors reveals that the proposed scheme is quite competent with other methods including ADM and RBF used in [198, 200] for solving the generalized Burger equation. The comparison further reveals that the proposed scheme is capable to achieve the approximate solutions in the larger domain of time t with greater accuracy. Moreover, it is observed from Table 5.27 that our scheme gives more accurate solutions than ADM for the interval $t \in (0.0001, 0.001)$ and $\delta = 3$.

Table 5.24 Optimal values of unknown constants for different values of δ

Constant	$\delta = 1$	$\delta = 2$	$\delta = 3$
a_2	-0.02125	-1.17068	9.75002
a_1	0.09513	6.45543	-0.68817
a_0	4.33794	1.20187	5.57958
a_{-1}	3.83626	5.95373	0.67409
a_{-2}	2.79395	9.85137	-0.33117
b_2	4.85151	9.54284	9.39347
b_1	3.33969	7.28427	1.56672
b_0	7.41892	-3.83427	-0.00436
b_{-1}	3.65409	9.41931	8.97744
b_{-2}	2.81992	9.11302	-1.05417
k	0.24546	0.18000	-0.23131
ω	-0.12273	-0.05999	0.05784

Table 5.25 Comparison of numerical solutions for $\beta = 0, \alpha = 1$, and $\delta = 1$

t	x	u_{exact}	**Proposed** $\hat{u}(\eta)$	**ADM**	**RBF**	**Absolute errors** **Proposed**	**ADM**	**RBF**
0.5	0.1	0.518741	0.518740	0.518741	0.518739	1.14E-07	6.34E-08	2.00E-06
	0.5	0.468791	0.468791	0.468791	0.468790	1.13E-07	5.66E-08	1.00E-06
	0.9	0.419458	0.419459	0.419458	0.419449	1.56E-06	4.12E-08	9.00E-06
1.0	0.1	0.549834	0.549833	0.549832	0.549831	1.17E-06	2.02E-06	3.00E-06
	0.5	0.500000	0.499999	0.499998	0.499998	3.79E-08	1.84E-06	2.00E-06
	0.9	0.450166	0.450167	0.450165	0.450157	1.28E-06	1.37E-06	9.00E-06
2.0	0.1	0.610639	0.610638	0.610575	0.610635	8.44E-07	6.42E-05	4.00E-06
	0.5	0.562177	0.562176	0.562116	0.562175	1.16E-07	6.06E-05	2.00E-06
	0.9	0.512497	0.512498	0.512450	0.512488	9.72E-07	4.75E-05	9.00E-06

Table 5.26 Comparison of numerical solutions for $\beta = 0, \alpha = 1$, and $\delta = 2$

t	x	u_{exact}	**Proposed**	**ADM**	**Absolute errors**	
			$\widehat{u}(\eta)$		**Proposed**	**ADM**
0.5	0.1	0.714919	0.714918	0.714919	7.43E-07	1.25E-08
	0.5	0.666837	0.666836	0.666837	1.16E-06	1.49E-08
	0.9	0.616567	0.616565	0.616567	2.38E-06	1.39E-08
1.0	0.1	0.734037	0.734034	0.734037	2.94E-06	1.25E-08
	0.5	0.687205	0.687202	0.687205	3.22E-06	4.75E-07
	0.9	0.637701	0.637697	0.637701	4.20E-06	4.39E-07
2.0	0.1	0.770284	0.770277	0.770272	7.21E-06	1.18E-05
	0.5	0.726464	0.726456	0.726449	7.35E-06	1.49E-05
	0.9	0.679109	0.679101	0.679095	8.03E-06	1.43E-05

Table 5.27 Comparison of numerical solutions for $\beta = 0, \alpha = 1$, and $\delta = 3$

t	x	u_{exact}	**Proposed**	**ADM**	**Absolute errors**	
			$\widehat{u}(\eta)$		**Proposed**	**ADM**
0.0001	0.1	0.783660	0.783659	0.784106	4.55E-07	4.46E-04
	0.5	0.741285	0.741285	0.743145	5.66E-07	1.86E-03
	0.9	0.696157	0.696158	0.697089	7.00E-07	9.32E-04
0.0005	0.1	0.783670	0.783670	0.784115	4.57E-07	4.45E-04
	0.5	0.741296	0.741296	0.743150	5.63E-07	1.85E-03
	0.9	0.696169	0.696170	0.697089	6.98E-07	9.20E-04
0.001	0.1	0.783683	0.783682	0.784127	4.60E-07	4.44E-04
	0.5	0.741309	0.741309	0.743157	5.61E-07	1.85E-03
	0.9	0.696183	0.696184	0.697088	6.95E-07	9.05E-04

Table 5.28 Comparison of numerical with RBF for $\beta = 0, \alpha = 1$, and $\delta = 3$

		u_{exact}	Proposed		Absolute errors	
t	x		$\widehat{u}(\eta)$	RBF	Proposed	RBF
0.5	0.1	0.796173	0.796174	0.796176	1.00E-06	3.00E-06
	0.5	0.75487	0.754871	0.754877	1.00E-06	7.00E-06
	0.9	0.710485	0.710486	0.710486	1.00E-06	1.00E-06
1.0	0.1	0.808297	0.808299	0.808299	2.00E-06	2.00E-06
	0.5	0.768157	0.768159	0.768165	2.00E-06	8.00E-06
	0.9	0.724622	0.724625	0.724623	3.00E-06	1.00E-06
2.0	0.1	0.831283	0.831288	0.831286	5.00E-06	3.00E-06
	0.5	0.793701	0.793706	0.793709	5.00E-06	8.00E-06
	0.9	0.752176	0.752182	0.752177	6.00E-06	1.00E-06
5.0	0.1	0.889248	0.88926	0.889252	1.20E-05	4.00E-06
	0.5	0.860439	0.860452	0.860452	1.30E-05	1.30E-05
	0.9	0.826825	0.826839	0.826828	1.40E-05	3.00E-06

Finally we study the effect of change in the values of p, q, c, and d in equation (5.27) on the accuracy of approximate solution, and show reliability of the proposed scheme. We used following test cases

Case (i): $p = c = 1 \quad q = d = 1$

Case (ii): $p = c = 2 \quad q = d = 2$

Case (iii): $p = c = 3 \quad q = d = 3$

Case (iv): $p = c = 1 \quad q = d = 2$

We consider the generalized B-F equation (5.40) with $\alpha = \beta = 0.001$ and $\delta = 1$. The approximate solution is obtained in the domain $x, t \in (0,1)$.

The approximate solution is expressed in view of the Exp-function method by choosing the values of p, q, c and d in (5.27), we get following trial functions for case (i) – case (iv) respectively.

$$\tilde{u}(\eta)=\frac{a_1\exp(\eta)+a_0+a_{-1}\exp(-\eta)}{b_1\exp(\eta)+b_0+b_{-1}\exp(-\eta)} \tag{5.48}$$

$$\tilde{u}(\eta)=\frac{a_2\exp(2\eta)+a_1\exp(\eta)+a_0+a_{-1}\exp(-\eta)+a_{-2}\exp(-2\eta)}{b_2\exp(2\eta)+b_2\exp(2\eta)+b_0+a_{-1}\exp(-\eta)+b_{-2}\exp(-2\eta)} \tag{5.49}$$

$$\tilde{u}(\eta)=\frac{\begin{pmatrix}a_3\exp(3\eta)+a_2\exp(2\eta)+a_1\exp(\eta)+a_0\\+a_{-1}\exp(-\eta)+a_{-2}\exp(-2\eta)+a_{-3}\exp(-3\eta)\end{pmatrix}}{\begin{pmatrix}b_3\exp(3\eta)+b_2\exp(2\eta)+b_1\exp(\eta)+b_0\\+b_{-1}\exp(-\eta)+b_{-2}\exp(-2\eta)+b_{-3}\exp(-3\eta)\end{pmatrix}} \tag{5.50}$$

$$\tilde{u}(\eta)=\frac{a_1\exp(\eta)+a_0+a_{-1}\exp(-\eta)+a_{-2}\exp(-2\eta)}{b_1\exp(\eta)+b_0+b_{-1}\exp(-\eta)+b_{-2}\exp(-2\eta)} \tag{5.51}$$

The GA has been used to solve the FF given by (5.46) with same settings for all the four cases (i) - (iv) as prescribed in Table 5.16 for example 1, except with a change in chromosome size for each case which is 8, 12, 16, and 10 for case (i), case (ii), case (iii), and case (iv) respectively. The optimal values of unknown constants acquired by GA are provided in Table 5.29.

The numerical solutions have been obtained for each case and absolute errors have been computed. In Table 5.30 we present a comparison of numerical solutions obtained by the proposed scheme with each case at time t = 0, also exact solutions are shown for the comparison. To further evaluate the influence of the parameters *p, q, c, and d* average absolute errors obtained by the proposed scheme for each case (i) - (iv) in the solution domain [0, 1] have been computed and presented in Table 5.31. Moreover, computational time and number of generations utilized are also given for the sake of comparison in Table 5.31. From the comparison of Table 5.31, it is observed that the average absolute error corresponding to case (i) with $p = c = 1$ and $d = q = 1$ is relatively high compared to other cases (ii) – (iv). It is also observed that the accuracy is fairly equal for the remaining cases (ii) - (iv), however the computational time is quite different. It can be seen from Table 5.31 that for case (iv) we get the average absolute error quite comparable to cases (ii) and (iii), but with lesser number generations and smaller computational time. Therefore it can be concluded on the basis of the simulation results that the choice of *p, q, c,* and *d* have influence on the accuracy of numerical solutions and computational time. Nonetheless the comparison clearly demonstrates the accuracy and reliability of the proposed scheme.

Table 5.29 Optimal values of unknown constants with different values of *p, q, c,* and *d*

Constant	Case (i)	Case (ii)	Case (iii)	Case (iv)
a_3	-	-	0.289132	-
a_2	-	0.355869	-0.27761	-
a_1	0.318111	0.233491	0.40672	0.759563
a_0	0.19323	0.889993	0.193522	0.071117
a_{-1}	0.266052	0.526815	1.221299	0.574267
a_{-2}	-	0.148751	-0.01553	0.557091
a_{-3}	-	-	0.683148	-
b_3	-	-	-0.19988	-
b_2	-	0.831958	0.483044	-
b_1	-0.43654	0.85859	0.736614	0.802849
b_0	0.715427	0.746341	0.683484	0.602095
b_{-1}	1.275903	0.762522	0.576841	1.076806
b_{-2}	-	1.110428	1.404818	1.442326
b_{-3}	-	-	1.31645	-
k	0.000214	-0.00153	-0.00105	-0.00075
ω	0.006147	0.003067	0.002102	0.00151

Table 5.30 Comparison of numerical solutions for B-F equation

		Proposed Scheme, $\bar{u}(\eta)$			
x	u_{exact}	Case (i)	Case (ii)	Case (iii)	Case (iv)
0	0.500025	0.499641	0.500025	0.500025	0.500025
0.1	0.500013	0.499629	0.500012	0.500013	0.500013
0.2	0.500000	0.499616	0.500000	0.500000	0.500000
0.3	0.499988	0.499604	0.499988	0.499988	0.499988
0.4	0.499975	0.499591	0.499975	0.499975	0.499975
0.5	0.499963	0.499579	0.499963	0.499962	0.499963
0.6	0.499950	0.499566	0.499950	0.499950	0.499950
0.7	0.499938	0.499554	0.499938	0.499937	0.499938
0.8	0.499925	0.499541	0.499925	0.499925	0.499925
0.9	0.499913	0.499529	0.499913	0.499912	0.499913
1	0.499900	0.499516	0.499900	0.499900	0.499900

Table 5.31 Effect of change in *p, q, c,* and *d* on the performance of the scheme

Values of p, q, c, d	Average absolute error	No. of generations	Computational time in sec
Case (i)	1.91E-03	196	80
Case(ii)	1.97E-07	457	177
Case (iii)	1.42E-07	279	97
Case (iv)	1.76E-7	51	40

5.5.3 GENERALIZED BURGER-HUXLEY EQUATION

In this section, we consider the generalized Burger-Huxley (B-H) equation as a final NPDE problem and obtain its numerical solution using the proposed scheme. To prove the efficacy and viability of the proposed scheme generalized B-H equation is numerically solved for various values of the parameters governing the equation, besides the numerical solutions of the generalized Huxley equation are obtained for various choices of the parameters.

We consider the generalized B-H equation as follows [205], [206], [207]

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u\left(1 - u^\delta\right)\left(u^\delta - \gamma\right), \quad 0 \le x \le 1, t \ge 0 \tag{5.52}$$

subject to the initial condition

$$u(x,0) = \left[\frac{\gamma}{2} + \frac{\gamma}{2}\tanh(\sigma\gamma x)\right]^{\frac{1}{\delta}} \tag{5.53}$$

with the exact solution given by

$$u_{exact}(x,t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2}\tanh\left\{\sigma\gamma\left(x - \left\{\frac{\gamma\alpha}{1+\delta} - \frac{(1+\delta-\gamma)(\rho-\alpha)}{2(1+\delta)}\right\}t\right)\right\}\right]^{\frac{1}{\delta}} \tag{5.54}$$

where $\sigma = \delta(\rho - \alpha)/4(1+\delta)$ and $\rho = \sqrt{\alpha^2 + 4\beta(1+\delta)}$. The generalized B-H equation (5.39) describes the prototype models of interaction between reaction mechanisms, convection effects, and diffusion transports [205], [206], [207]. When $\alpha = 0$, equation (5.39) reduces to the generalized Huxley equation, which describes wall motion in liquid crystals [207], [208], [209], [210]. B-H equation has been paid much attention and many methods have been proposed so far, but no one yet has tried the method proposed here.

Example 1: We consider the generalized B-H equation (5.52) and obtain its approximate solution in the domain $x,t \in [0,1]$, with following values of the parameters for a direct comparison with other methods reported in [205], [206].

Case 1: $\alpha = \beta = \delta = 1, \gamma = 0.001$

Case 2: $\alpha = \beta = 1, \gamma = 0.01$, $\delta = 2,4,6$

To apply the proposed scheme we first convert the given NPDE into corresponding NODE by applying the wave the transformation variable η to the generalized Burgers-Huxley equation to get the following equation.

$$\omega u' + \alpha u^{\delta} k u' - k^2 u'' = \beta u\left(1 - u^{\delta}\right)\left(u^{\delta} - \gamma\right) \tag{5.55}$$

The unknown constants ($a_2,\ldots,a_{-2}; b_2,\ldots,b_{-2}$; k and ω) in (5.27) are achieved using GA and PSO by formulating the FF for each case. For instance the FF for case 1 that is $\alpha = \beta = \delta = 1, \gamma = 0.001$ is given by

$$\begin{aligned}\varepsilon_j = \frac{1}{(144)}\sum_{i=1}^{11}\sum_{j=1}^{11}&\begin{pmatrix}\omega u'\left(kx_j + \omega t_i\right) + u^{\delta}\left(kx_j + \omega t_i\right)ku'\left(kx_j + \omega t_i\right) - k^2 u''\left(kx_j + \omega t_i\right)\\ - u\left(kx_j + \omega t_i\right)\left(1 - u\left(kx_j + \omega t_i\right)\right)\left(u\left(kx_j + \omega t_i\right) - 0.001\right)\end{pmatrix}^2\\ &+ \frac{1}{11}\sum_{j=1}^{11}\left(u\left(x_j,0\right) - \left[\frac{0.001}{2} + \frac{0.001}{2}\tanh\left(\frac{0.001}{4}x_j\right)\right]\right)^2\end{aligned} \tag{5.56}$$

The FF given by (5.56) is minimized by applying GA and PSO for obtaining the optimal values of unknown parameters.

The parameter settings and values used for the implementation of the algorithms GA and PSO are given in Table 5.32.

The number of unknown constants ($a_2,....,a_{-2};b_2,....,b_{-2}$, k and ω) which need to be tailored is 12, therefore the size of chromosome and the size of particle each is chosen as 12 respectively for GA and PSO.

The values of these unknown constants are restricted between -20 and +20. The GA and PSO are executed to achieve the minimum fitness, with the prescribed parameter settings and values given in Table 5.32.

Table 5.32 Parameter values and settings of GA and PSO for B-H equation

GA		PSO	
Parameter Name	**Setting/Value**	**Parameter Name**	**Value**
Population size	[310 310]	Population size	620
Chromosome size	12	Particle size	12
Scaling function	Rank	Cognitive constant	0.5
Selection function	Stochastic uniform	Social constant	1.0
Mutation function	Adaptive feasible	No. of generations	1000
Crossover function	Heuristic	Function tolerance	1e-18
Crossover fraction	0.8	Bounds	-20, +20
No. of generations	1000	-	-
Function tolerance	1e-18	-	-
Bounds	-20, +20	-	-

The optimal chromosome corresponding to the minimum fitness achieved by GA and PSO are given in Table 5.33 and Table 5.34 for case 1 and case 2 respectively.

The approximate solution of the generalized B-H equation (5.52) is consequently obtained by using the values of unknown parameters from Table 5.33 and Table 5.34 in (5.27).

Table 5.33 Optimal values of unknown constants for $\alpha = \beta = \delta = 1, \gamma = 0.001$

Constant	Value	
	GA	PSO
a_2	0.072043	1.046513
a_1	-0.143229	-1.357303
a_0	-0.107314	4.013094
a_{-1}	0.528938	-2.355655
a_{-2}	-0.345736	-1.334002
b_2	2.886257	7.220108
b_1	0.278548	2.643407
b_0	0.601896	6.631879
b_{-1}	3.129988	8.380038
b_{-2}	2.508507	0.419568
k	0.000007	0.000001
ω	-0.000014	-0.000001

Table 5.34 Optimal values of unknown constants for $\alpha = \beta = 1, \gamma = 0.01$, and $\delta = 2,4$, and 6

Constant	Value					
	$\delta = 2$		$\delta = 4$		$\delta = 6$	
	GA	PSO	GA	PSO	GA	PSO
a_2	2.6333	-0.5878	4.6845	-1.1357	5.6753	-3.5955
a_1	-1.1713	-1.1993	-6.2895	7.1417	6.7001	1.2889
a_0	-7.4537	1.9339	11.3438	0.7822	1.8124	-1.0281
a_{-1}	13.7534	-0.6767	-0.2318	-2.7754	8.1234	-1.1954
a_{-2}	-3.2001	-0.6929	6.5612	4.3251	5.3674	-3.3303
b_2	12.0289	-7.0723	16.3038	4.0919	18.3226	-13.3246
b_1	9.5465	-0.0046	7.4003	3.6792	4.7712	3.8662
b_0	13.3712	-4.2442	1.1396	16.2994	18.6110	-2.1033
b_{-1}	9.8656	-2.0549	17.7174	-7.3819	11.5079	3.8682
b_{-2}	19.6997	-3.9111	17.8647	14.6671	13.7217	-11.3150
k	-0.0046	-0.0402	-0.0046	0.0091	-0.0237	-0.0034
ω	5.0000	1.9649	1.9455	-0.0248	0.0757	0.0109

Table 5.35 Numerical solutions by proposed scheme for $\alpha = \beta = \delta = 1$ and $\gamma = 0.001$

			Proposed scheme $\hat{u}(\eta)$		**Absolute errors** $\lvert u_{exact} - \hat{u}(\eta) \rvert$	
t	x	u_{exact}	**GA**	**PSO**	**GA**	**PSO**
0.2	0.2	0.0005001	0.0005000	0.0005000	7.61E-08	7.50E-08
	0.4	0.0005001	0.0005000	0.0005000	7.48E-08	7.50E-08
	0.6	0.0005001	0.0005000	0.0005000	7.36E-08	7.50E-08
	0.8	0.0005001	0.0005001	0.0005001	7.24E-08	7.50E-08
	1.0	0.0005002	0.0005001	0.0005001	7.12E-08	7.50E-08
0.4	0.2	0.0005001	0.0004999	0.0004999	1.51E-07	1.50E-07
	0.4	0.0005001	0.0005000	0.0005000	1.50E-07	1.50E-07
	0.6	0.0005001	0.0005000	0.0005000	1.49E-07	1.50E-07
	0.8	0.0005001	0.0005000	0.0005000	1.47E-07	1.50E-07
	1.0	0.0005002	0.0005000	0.0005000	1.46E-07	1.50E-07
0.6	0.2	0.0005001	0.0004999	0.0004999	2.26E-07	2.25E-07
	0.4	0.0005001	0.0004999	0.0004999	2.25E-07	2.25E-07
	0.6	0.0005001	0.0004999	0.0004999	2.24E-07	2.25E-07
	0.8	0.0005002	0.0005000	0.0005000	2.22E-07	2.25E-07
	1.0	0.0005002	0.0005000	0.0005000	2.21E-07	2.25E-07
0.8	0.2	0.0005001	0.0004998	0.0004998	3.01E-07	3.00E-07
	0.4	0.0005001	0.0004999	0.0004999	3.00E-07	3.00E-07
	0.6	0.0005002	0.0004999	0.0004999	2.98E-07	3.00E-07
	0.8	0.0005002	0.0004999	0.0004999	2.97E-07	3.00E-07
	1.0	0.0005002	0.0004999	0.0004999	2.96E-07	3.00E-07

In Table 5.35 and Table 5.36 we present numerical solutions obtained by the proposed scheme for case 1 at different values of x and t, also the exact solution and absolute errors obtained by classical methods ADM [205] and VIM [206] are provided for the comparison purpose. From the comparison of absolute errors approximate solutions are found in good agreement with the exact solution with an average absolute error of 1.874E-07 in the solution domain $x, t \in (0,1)$.

Table 5.36 Comparison of numerical solutions for $\alpha = \beta = \delta = 1$ and $\gamma = 0.001$

			Proposed $\widehat{u}(\eta)$		Absolute Errors			
x	t	u_{exact}	GA	PSO	GA	PSO	ADM	VIM
0.1	0.05	0.00050002	0.00050000	0.00050000	2.05E-08	1.87E-08	1.87E-08	1.87E-08
	0.1	0.00050003	0.00049999	0.00049999	3.92E-08	3.75E-08	3.74E-08	3.74E-08
	1.0	0.00050014	0.00049976	0.00049976	3.76E-07	3.75E-07	3.74E-07	3.74E-07
0.5	0.05	0.00050007	0.00050005	0.00050005	1.80E-08	1.87E-08	1.87E-08	1.87E-08
	0.1	0.00050008	0.00050004	0.00050004	3.68E-08	3.75E-08	3.74E-08	3.74E-08
	1.0	0.00050019	0.00049981	0.00049981	3.74E-07	3.75E-07	3.74E-07	3.74E-07
0.9	0.05	0.00050012	0.00050010	0.00050010	1.56E-08	1.87E-08	1.87E-08	1.87E-08
	0.1	0.00050013	0.00050009	0.00050009	3.43E-08	3.75E-08	3.74E-08	3.74E-08
	1.0	0.00050024	0.00049987	0.00049986	3.72E-07	3.75E-07	3.74E-07	3.74E-07

Table 5.37 Comparison of numerical solutions for $\alpha = \beta = 1, \gamma = 0.01$ and $\delta = 2,4$

		$\delta = 2$		$\delta = 4$		Absolute errors	
t	x	u_{exact}	Proposed $\widehat{u}(\eta)$	u_{exact}	Proposed $\widehat{u}(\eta)$	$\delta = 2$	$\delta = 4$
0.2	0.2	0.070781	0.070740	0.266180	0.265744	4.12E-05	4.35E-04
	0.4	0.070812	0.070757	0.266275	0.265840	5.49E-05	4.35E-04
	0.6	0.070842	0.070788	0.266370	0.265935	5.45E-05	4.35E-04
	0.8	0.070873	0.070834	0.266464	0.266030	3.96E-05	4.34E-04
	1.0	0.070904	0.070894	0.266559	0.266125	1.02E-05	4.34E-04
0.6	0.2	0.070861	0.070741	0.266518	0.265211	1.20E-04	1.31E-03
	0.4	0.070891	0.070759	0.266613	0.265306	1.32E-04	1.31E-03
	0.6	0.070922	0.070791	0.266707	0.265402	1.31E-04	1.31E-03
	0.8	0.070952	0.070838	0.266802	0.265498	1.15E-04	1.30E-03
	1.0	0.070983	0.070899	0.266896	0.265594	8.43E-05	1.30E-03
1.0	0.2	0.070940	0.070742	0.266856	0.264674	1.98E-04	2.18E-03
	0.4	0.070970	0.070761	0.266950	0.264770	2.10E-04	2.18E-03
	0.6	0.071001	0.070794	0.267044	0.264867	2.07E-04	2.18E-03
	0.8	0.071032	0.070842	0.267138	0.264963	1.90E-04	2.17E-03
	1.0	0.071062	0.070904	0.267232	0.265059	1.58E-04	2.17E-03

Moreover the absolute errors yielded by the proposed scheme are found quite comparable to ADM and VIM, which shows the efficiency of the proposed scheme. Further a comparison of numerical solutions and absolute errors obtained by the proposed scheme is made with the exact solutions and absolute errors obtained using ADM and VIM for $\alpha = \beta = 1, \gamma = 0.01$, and $\delta = 2,4$ in Table 5.37 and Table 5.38 and for $\alpha = \beta = 1, \gamma = 0.01$, and $\delta = 6$ in Table 5.39 respectively.

The comparison of absolute errors from Table 5.38 and Table 5.39 reveals that the proposed scheme gives approximate solution relatively with greater accuracy than ADM and VIM for $\delta = 2$ while for $\delta = 4$ and $\delta = 6$ the accuracy of the proposed scheme is quite comparable to ADM and VIM used in [205], [206].

Table 5.38 Comparison of absolute errors for $\alpha = \beta = 1, \gamma = 0.01$and $\delta = 2,4$

			$\delta = 2$			$\delta = 4$	
x	t	**Proposed**	**ADM**	**VIM**	**Proposed**	**ADM**	**VIM**
0.1	0.1	9.120E-06	5.516E-05	5.516E-05	2.178E-04	2.178E-04	2.177E-04
	0.2	2.890E-05	1.103E-04	1.103E-04	4.357E-04	4.357E-04	4.353E-04
	0.3	4.880E-05	1.655E-04	1.655E-04	6.537E-04	6.537E-04	6.528E-04
	0.4	6.860E-05	2.207E-04	2.206E-04	8.719E-04	8.718E-04	8.703E-04
	0.5	8.830E-05	2.760E-04	2.757E-04	1.090E-03	1.090E-03	1.088E-03
0.3	0.1	3.030E-05	5.514E-05	5.513E-05	2.176E-04	2.176E-04	2.175E-04
	0.2	4.980E-05	1.103E-04	1.103E-04	4.352E-04	4.352E-04	4.348E-04
	0.3	6.940E-05	1.655E-04	1.654E-04	6.530E-04	6.530E-04	6.521E-04
	0.4	8.890E-05	2.206E-04	2.205E-04	8.709E-04	8.709E-04	8.693E-04
	0.5	1.080E-04	2.758E-04	2.756E-04	1.089E-03	1.089E-03	1.086E-03
0.5	0.1	3.720E-05	5.511E-05	5.511E-05	2.173E-04	2.173E-04	2.172E-04
	0.2	5.650E-05	1.102E-04	1.102E-04	4.348E-04	4.348E-04	4.344E-04
	0.3	7.580E-05	1.654E-04	1.653E-04	6.523E-04	6.523E-04	6.514E-04
	0.4	9.500E-05	2.205E-04	2.204E-04	8.700E-04	8.700E-04	8.684E-04
	0.5	1.140E-04	2.757E-04	2.755E-04	1.088E-03	1.088E-03	1.085E-03

Table 5.39 Comparison of numerical solutions for $\alpha = \beta = 1, \gamma = 0.01$ and $\delta = 6$

			Proposed	Absolute errors	
t	x	u_{exact}	$\hat{u}(\eta)$	Proposed	ADM
0.1	0.1	0.413725	0.413377	3.481E-04	3.481E-04
	0.2	0.413789	0.413441	3.479E-04	3.478E-04
	0.3	0.413854	0.413506	3.476E-04	3.476E-04
	0.4	0.413918	0.413571	3.474E-04	3.473E-04
	0.5	0.413983	0.413636	3.471E-04	3.470E-04
0.2	0.1	0.413866	0.413169	6.965E-04	6.965E-04
	0.2	0.413930	0.413234	6.960E-04	6.960E-04
	0.3	0.413995	0.413299	6.955E-04	6.954E-04
	0.4	0.414059	0.413364	6.949E-04	6.949E-04
	0.5	0.414124	0.413429	6.944E-04	6.943E-04
0.3	0.1	0.414007	0.412962	1.045E-03	1.045E-03
	0.2	0.414071	0.413027	1.044E-03	1.044E-03
	0.3	0.414136	0.413092	1.044E-03	1.044E-03
	0.4	0.414200	0.413157	1.043E-03	1.043E-03
	0.5	0.414264	0.413222	1.042E-03	1.042E-03
0.4	0.1	0.414148	0.412754	1.394E-03	1.394E-03
	0.2	0.414212	0.412819	1.393E-03	1.393E-03
	0.3	0.414276	0.412884	1.392E-03	1.392E-03
	0.4	0.414340	0.412949	1.391E-03	1.391E-03
	0.5	0.414404	0.413015	1.390E-03	1.390E-03
0.5	0.1	0.414288	0.412545	1.743E-03	1.743E-03
	0.2	0.414352	0.412610	1.742E-03	1.742E-03
	0.3	0.414416	0.412676	1.741E-03	1.741E-03
	0.4	0.414480	0.412741	1.739E-03	1.739E-03
	0.5	0.414544	0.412806	1.738E-03	1.738E-03

Example 2: Consider the generalized B-H equation (5.52) with $\alpha = 0$, the equation is reduced to the generalized Huxley equation as follows.

$$u_t - u_{xx} = \beta\left(1 - u^{\delta}\right)\left(u^{\delta} - \gamma\right) \qquad (5.57)$$

The numerical solution of (5.57) subject to the initial condition (5.53) is obtained by the proposed scheme in domain $x, t \in (0,1)$, for $\beta = 1, \gamma = 0.001$, and for different values of $\delta = 1,2$ and 3. We assume that approximate solution is given by (5.27) in view of Exp-function method. Equation (5.57) is first converted into a NODE using the transformation variable η and fitness function (ε_j) with the parameter values $\beta = 1, \gamma = 0.001$, and for each value of $\delta = 1,2$ and 3 is developed. For example ε_j corresponding to $\beta = 1$, $\gamma = 0.001$, and $\delta = 3$ is given by

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}\sum_{j=1}^{11}\begin{pmatrix}\omega u'\left(kx_j + \omega t_i\right) - k^2 u''\left(kx_j + \omega t_i\right) - \\ u\left(kx_j + \omega t_i\right)\left(1 - u^3\left(kx_j + \omega t_i\right)\right)\left(u^3\left(kx_j + \omega t_i\right) - 0.001\right)\end{pmatrix}^2 + \\ \frac{1}{11}\sum_{j=1}^{11}\left(u\left(x_j, 0\right) - \left[\frac{0.001}{2} + \frac{0.001}{2}\tanh\left(\frac{(0.001)\sqrt{8}}{8}x_j\right)\right]^{\frac{1}{3}}\right)^2 \qquad (5.58)$$

The minimization problem given by (5.58) is solved using GA and PSO to achieve the constants *(a_2,...,a_{-2}, b_2,...,b_{-2}, and k, ω)* in (5.27), and consequently the approximate solution of (5.57) is obtained.

The algorithms are executed with the parameter values and settings as prescribed in Table 5.32. The optimal chromosome/particle learned by the algorithms GA and PSO which represents the values of unknown constants in are provided in Table 5.40. The values of unknown constants are used in (5.27) to obtain the approximate numerical solution at any value of x and t in the solution domain [0, 1].

Table 5.40 Optimal values of unknown constants for B-H equation (example 2)

	Value					
	$\delta=1$		$\delta=2$		$\delta=3$	
Constants	**GA**	**PSO**	**GA**	**PSO**	**GA**	**PSO**
a_2	-0.395617	3.305147	0.051950	-6.998792	-0.197379	2.248346
a_1	0.904366	-9.510479	0.035122	17.029130	0.594860	-7.656524
a_0	-0.429999	7.191818	-0.332515	-5.869583	-0.311415	8.519216
a_{-1}	-0.268263	5.349152	0.377499	-12.827567	0.309808	-10.646525
a_{-2}	0.193587	-6.324107	-0.071184	9.225269	0.105479	4.451424
b_2	2.893155	3.250626	0.362974	14.916439	-0.293531	-17.832675
b_1	1.172939	2.926159	-0.473849	-6.381976	1.604009	18.849973
b_0	-0.397558	4.423938	-0.260857	1.864866	2.640544	-16.071581
b_{-1}	2.435372	8.635413	1.648454	2.600298	1.373907	-10.351738
b_{-2}	2.045568	3.826029	1.445436	11.975325	0.991738	-13.450688
k	-0.000253	0.000001	-0.014268	-0.000064	-0.000937	0.000251
ω	0.000335	-0.000001	0.002208	0.000111	0.001860	-0.000507

Table 5.41 Comparison of numerical solutions for $\beta=1, \gamma=0.001$ and $\delta=1$

x	t		**Proposed scheme $\hat{u}(\eta)$**		**Absolute errors**	
		u_{exact}	GA	PSO	GA	PSO
0.1	0.05	0.00050003	0.00050001	0.00050001	1.70E-08	2.50E-08
	0.1	0.00050004	0.00050000	0.00049999	4.18E-08	5.00E-08
	1.0	0.00050027	0.00049978	0.00049977	4.90E-07	5.00E-07
0.5	0.05	0.00050010	0.00050009	0.00050008	1.33E-08	2.50E-08
	0.1	0.00050011	0.00050008	0.00050006	3.81E-08	5.00E-08
	1.0	0.00050034	0.00049985	0.00049984	4.85E-07	5.00E-07
0.9	0.05	0.00050017	0.00050016	0.00050015	9.82E-09	2.50E-08
	0.1	0.00050018	0.00049993	0.00050013	2.56E-07	5.00E-08
	1.0	0.00050041	0.00050015	0.00049991	2.59E-07	5.00E-07

Table 5.41, Table 5.42, and Table 5.43 show a comparison of numerical solutions obtained by the proposed schemes GA and PSO with the exact solutions for $\delta = 1$, $\delta = 2$, and $\delta = 3$ respectively. It can be seen from the comparison that numerical solutions are in a good agreement with the exact solutions.

Table 5.42 Comparison of numerical solutions for $\beta = 1, \gamma = 0.001$ and $\delta = 2$

x	t	u_{exact}	**Proposed scheme $\tilde{u}(\eta)$**		**Absolute errors**	
			GA	**PSO**	**GA**	**PSO**
0.1	0.05	0.022362	0.022361	0.022361	1.12E-06	1.12E-06
	0.1	0.022362	0.022360	0.022360	2.24E-06	2.23E-06
	1.0	0.022373	0.022350	0.022350	2.24E-05	2.23E-05
0.5	0.05	0.022364	0.022363	0.022363	1.12E-06	1.12E-06
	0.1	0.022365	0.022363	0.022363	2.23E-06	2.24E-06
	1.0	0.022375	0.022353	0.022353	2.24E-05	2.23E-05
0.9	0.05	0.022367	0.022366	0.022366	1.12E-06	1.12E-06
	0.1	0.022368	0.022365	0.022365	2.23E-06	2.24E-06
	1.0	0.022378	0.022355	0.022355	2.23E-05	2.23E-05

Table 5.43 Comparison of numerical solutions for $\beta = 1, \gamma = 0.001$ and $\delta = 3$

x	t	u_{exact}	**Proposed scheme $\tilde{u}(\eta)$**		**Absolute errors**	
			GA	**PSO**	**GA**	**PSO**
0.1	0.05	0.079374	0.079370	0.079370	4.07E-06	3.88E-06
	0.1	0.079376	0.079368	0.079368	8.04E-06	7.85E-06
	1.0	0.079412	0.079332	0.079332	7.95E-05	7.93E-05
0.5	0.05	0.079382	0.079378	0.079378	4.00E-06	3.96E-06
	0.1	0.079384	0.079376	0.079376	7.97E-06	7.93E-06
	1.0	0.079420	0.079340	0.079340	7.94E-05	7.93E-05
0.9	0.05	0.079390	0.079386	0.079386	3.93E-06	4.04E-06
	0.1	0.079392	0.079384	0.079384	7.90E-06	8.01E-06
	1.0	0.079428	0.079348	0.079348	7.93E-05	7.94E-05

Table 5.44 Comparison of absolute errors for $\beta = 1, \gamma = 0.001$ and $\delta = 1$

			Absolute errors			
x	t	GA	PSO	VIM	HPM	ADM
0.1	0.05	1.70E-08	2.50E-08	2.50E-08	2.50E-08	2.50E-08
	0.1	4.18E-08	5.00E-08	5.00E-08	5.00E-08	5.00E-08
	1.0	4.90E-07	5.00E-07	5.00E-07	5.00E-07	5.00E-07
0.5	0.05	1.33E-08	2.50E-08	2.50E-08	2.50E-08	2.50E-08
	0.1	3.81E-08	5.00E-08	5.00E-08	5.00E-08	5.00E-08
	1.0	4.85E-07	5.00E-07	5.00E-07	5.00E-07	5.00E-07
0.9	0.05	9.82E-09	2.50E-08	2.50E-08	2.50E-08	2.50E-08
	0.1	2.56E-07	5.00E-08	5.00E-08	5.00E-08	5.00E-08
	1.0	2.59E-07	5.00E-07	5.00E-07	5.00E-07	5.00E-07

Table 5.45 Comparison of absolute errors for $\beta = 1, \gamma = 0.001$ and $\delta = 2$

			Absolute errors			
x	t	GA	PSO	VIM	HPM	ADM
0.1	0.05	1.12E-06	1.12E-06	1.12E-06	1.12E-06	1.12E-06
	0.1	2.24E-06	2.23E-06	2.24E-06	2.24E-06	2.24E-06
	1.0	2.24E-05	2.23E-05	2.23E-05	2.24E-05	2.24E-05
0.5	0.05	1.12E-06	1.12E-06	1.12E-06	1.12E-06	1.12E-06
	0.1	2.23E-06	2.24E-06	2.23E-06	2.23E-06	2.23E-06
	1.0	2.24E-05	2.23E-05	2.23E-05	2.24E-05	2.24E-05
0.9	0.05	1.12E-06	1.12E-06	1.12E-06	1.12E-06	1.12E-06
	0.1	2.23E-06	2.24E-06	2.23E-06	2.23E-06	2.23E-06
	1.0	2.23E-05	2.23E-05	2.23E-05	2.23E-05	2.23E-05

Further, to show the accuracy of the proposed scheme in comparison to the classical methods, we provide comparisons of absolute errors obtained by the proposed scheme with the absolute errors obtained by HPM [208], ADM [209], and VIM [210] in Table 5.44, Table 5.45, and Table 5.46 for $\delta = 1$, $\delta = 2$, and $\delta = 3$ respectively. One can clearly see that the absolute errors obtained by the proposed scheme for the generalized Huxley equation (5.57) are quite similar to the absolute errors obtained by HPM [208], ADM [209], and VIM [210] which confirms the accuracy of the proposed scheme.

Table 5.46 Comparison of absolute errors for $\beta = 1, \gamma = 0.001$ and $\delta = 3$

		Absolute errors				
x	t	**GA**	**PSO**	**VIM**	**HPM**	**ADM**
0.1	0.05	4.07E-06	3.88E-06	3.97E-06	3.97E-06	3.96E-06
	0.1	8.04E-06	7.85E-06	7.93E-06	7.93E-06	7.93E-06
	1.0	7.95E-05	7.93E-05	7.93E-05	7.93E-05	7.93E-06
0.5	0.05	4.00E-06	3.96E-06	3.97E-06	3.97E-06	3.96E-06
	0.1	7.97E-06	7.93E-06	7.93E-06	7.93E-06	7.93E-06
	1.0	7.94E-05	7.93E-05	7.93E-05	7.93E-05	7.93E-06
0.9	0.05	3.93E-06	4.04E-06	3.97E-06	3.97E-06	3.96E-06
	0.1	7.90E-06	8.01E-06	7.93E-06	7.93E-06	7.93E-06
	1.0	7.93E-05	7.94E-05	7.93E-05	7.93E-05	7.93E-06

Finally, we investigate the influence of change in the values of p, q, c and d on the accuracy of the approximate solution and show the reliability of the proposed scheme. We consider the generalized Huxley equation (5.54) with $\beta = 1, \gamma = 0.001$ and $\delta = 2$. The numerical solution using the proposed scheme is obtained in domain $x, t \in (0,1)$ for various values of p, q, c and d as follows.

Case (i): $p = c = 1$; $q = d = 1$

Case (ii): $p = c = 1$; $q = d = 2$

Case (iii): $p = c = 3$; $q = d = 3$

The approximate solution is expressed in view of the Exp-function method by choosing the values of p, q, c and d in (5.27), we get following trial functions for case (i) – case (iii) respectively.

$$\hat{u}(\eta)=\frac{a_1\exp(\eta)+a_0+a_{-1}\exp(-\eta)}{b_1\exp(\eta)+b_0+b_{-1}\exp(-\eta)} \tag{5.59}$$

$$\hat{u}(\eta)=\frac{a_1\exp(\eta)+a_0+a_{-1}\exp(-\eta)+a_{-2}\exp(-2\eta)}{b_1\exp(\eta)+b_0+b_{-1}\exp(-\eta)+b_{-2}\exp(-2\eta)} \tag{5.60}$$

$$\hat{u}(\eta)=\frac{\begin{pmatrix}a_3\exp(3\eta)+a_2\exp(2\eta)+a_1\exp(\eta)+a_0\\+a_{-1}\exp(-\eta)+a_{-2}\exp(-2\eta)+a_{-3}\exp(-3\eta)\end{pmatrix}}{\begin{pmatrix}b_3\exp(3\eta)+b_2\exp(2\eta)+b_1\exp(\eta)+b_0\\+b_{-1}\exp(-\eta)+b_{-2}\exp(-2\eta)+b_{-3}\exp(-3\eta)\end{pmatrix}} \tag{5.61}$$

GA and PSO are used to solve the minimization problem given by the fitness function corresponding to each case to achieve the unknown constants in (5.59) - (5.61).

The GA and PSO are implemented with the same parameter values and settings for all the cases as prescribed in Table 5.32, except a change in the chromosome/particle size which equals the number of unknown constants as 8, 10, and 16 for case (i), case (ii), and case (iii) respectively. The unknown constants acquired by the algorithms GA and PSO are provided in Table 5.47.

Table 5.47 Optimal values of unknown constants for B-H equation for different values of p, q, c, and d

	Value					
	Case (i)		Case (ii)		Case (iii)	
Constants	GA	PSO	GA	PSO	GA	PSO
a_3	-	-	-	-	-0.015473	15.458505
a_2	-	-	-	-	-0.210453	17.221482
a_1	-0.020537	-14.892131	1.006563	0.681380	0.320629	-16.779725
a_0	-0.025723	19.533463	-0.992386	-1.409252	0.167226	-18.366993
a_{-1}	0.099027	-5.681394	-0.241572	-2.995651	0.135173	4.088015
a_{-2}	-	-	-	-	-0.006768	-2.434400
a_{-3}	-	-	-	-	-0.017180	2.237701
b_3	-	-	-	-	0.475341	-5.641686
b_2	-	-	0.310559	2.917210	0.907889	4.272639
b_1	0.167858	-9.508666	0.283217	-2.366959	1.900287	10.518373
b_0	0.657426	-19.197997	2.239849	-4.895796	0.517934	14.094300
b_{-1}	1.534536	-17.806362	-0.486807	-19.811375	0.750430	13.389810
b_{-2}	-	-	1.682959	-8.985287	0.682116	19.990291
b_{-3}	-	-	-	-	0.043197	7.085685
k	0.000944	-0.000032	0.000035	0.000079	-0.002557	0.000007
ω	-0.005106	0.000058	-0.000061	-0.000137	0.004433	-0.000012

In Table 5.48 and Table 5.49 we present a comparison of numerical solutions and absolute errors obtained by the proposed scheme for each case at different values of t and x with exact solutions, also approximate solutions with $p = c = 2$ and $q = d = 2$ obtained above are given for sake of comparison. To further evaluate the effect of change in the values of p, q, c and d, we have computed average absolute errors in the solution domain $x, y \in (0,1)$, corresponding to each i.e. cases (i) , case (ii), and case (iii), also average absolute errors with $p=c=2$ and $q=d=2$ are computed and shown in Table 5.50.

Table 5.48 Comparison of numerical solutions for $\beta=1, \gamma=0.001$ and $\delta=2$ for different values of *p, q, c, d*

			Proposed Scheme $\hat{u}(\eta)$			
			$p=c=1$	$p=c=1$	$p=c=3$	$p=c=2$
t	x	u_{exact}	$q=d=1$	$q=d=2$	$q=d=3$	$q=d=2$
0.1	0.2	0.022363	0.022361	0.022361	0.022361	0.022361
	0.5	0.022365	0.022363	0.022363	0.022363	0.022363
	0.8	0.022367	0.022365	0.022365	0.022365	0.022365
	1.0	0.022368	0.022366	0.022366	0.022366	0.022366
0.5	0.2	0.022368	0.022356	0.022356	0.022356	0.022356
	0.5	0.022369	0.022358	0.022358	0.022358	0.022358
	0.8	0.022371	0.022360	0.022360	0.022360	0.022360
	1.0	0.022373	0.022361	0.022362	0.022362	0.022362
1.0	0.2	0.022373	0.022350	0.022351	0.022351	0.022351
	0.5	0.022375	0.022352	0.022353	0.022353	0.022353
	0.8	0.022377	0.022354	0.022355	0.022355	0.022355
	1.0	0.022378	0.022355	0.022356	0.022356	0.022356

From the comparison it is observed that the accuracy of the proposed scheme is quite comparable for all cases, which illustrates the effectiveness and reliability of the proposed scheme.

Table 5.49 Comparison of absolute errors for different values of *p*, *q*, *c*, and *d*

		$p=c=1$	$p=c=1$	$p=c=3$	$p=c=2$
t	x	$q=d=1$	$q=d=2$	$q=d=3$	$q=d=2$
0.1	0.2	2.28E-06	2.24E-06	2.24E-06	2.24E-06
	0.5	2.28E-06	2.24E-06	2.24E-06	2.24E-06
	0.8	2.28E-06	2.23E-06	2.24E-06	2.24E-06
	1.0	2.28E-06	2.23E-06	2.23E-06	2.23E-06
0.5	0.2	1.14E-05	1.12E-05	1.12E-05	1.12E-05
	0.5	1.14E-05	1.12E-05	1.12E-05	1.12E-05
	0.8	1.14E-05	1.12E-05	1.12E-05	1.12E-05
	1.0	1.14E-05	1.12E-05	1.12E-05	1.12E-05
1.0	0.2	2.28E-05	2.24E-05	2.23E-05	2.23E-05
	0.5	2.28E-05	2.23E-05	2.23E-05	2.23E-05
	0.8	2.28E-05	2.23E-05	2.23E-05	2.23E-05
	1.0	2.28E-05	2.23E-05	2.23E-05	2.23E-05

Table 5.50 Effect of *p*, *q*, *c*, and *d* on the accuracy of approximate solutions

	Average Absolute Error	
Value of p, q, c, d	**GA**	**PSO**
$p=c=1;\quad q=d=1$	2.307E-05	1.142E-05
$p=c=1;\quad q=d=2$	1.117E-05	1.117E-05
$p=c=3;\quad q=d=3$	1.120E-05	1.117E-05
$p=c=2;\quad q=d=2$	1.118E-05	1.117E-05

5.6 CONCLUSION

A simple and straightforward scheme based on the elegant couple Exp-function method and evolutionary algorithm has been 'presented for numerically solving NPDEs. The proposed scheme has been applied for the numerical solution of some well-known NPDEs to illustrate its usefulness.

The effectiveness of the proposed method has been tested on the Fishers equations. The simulation results have revealed that the proposed scheme shows supremacy on some of the classical methods including ADM, VIM, MVIM, VHPM, HWM, and OHAM.

The proposed heuristic scheme has also been successfully implemented for obtaining the numerical solutions of the generalized Burger's-Fisher, and Burger's equations. The numerical solutions obtained by the proposed scheme are found in a good agreement with the exact solutions, and also quite comparable or more accurate than some of the classical methods including ADM, HPM, OHAM, and RBF. Also the reliability of the proposed scheme has been illustrated by solving the generalized Burger's-Fisher and Burger's equations with different choice of parameters governing the equations.

The suggested method based on the couple of Exp-function method and nature inspired computing has been effectively applied for the numerical solutions of the generalize Burger's-Huxley (B-H) and Huxley equations. The B-H and Huxley equations have been considered with several different choices of involved parameters to show the accuracy and efficacy of the proposed scheme. Further the performance of the proposed scheme with different choices of parameters for approximating the solution has also been demonstrated with comprehensive simulations. It has been established from the comparison that the proposed scheme gives numerical solutions which are in a complete agreement with the solutions reported by some classical methods like ADM, VIM, and HPM.

On the basis of the simulation results and comprehensive comparisons, it can be concluded that the proposed scheme is a viable tool for solving such NPDEs. Furthermore the extensive investigation of change in the parameters of the trial solution has demonstrated the efficacy and reliability of the proposed scheme. Moreover the proposed scheme can provide the approximate solution of the given NPDE at any point in the solution domain once the unknown constants are achieved.

5.7 SUMMARY

This chapter provides the details for hybridization of the Exp-function method with nature inspired computing scheme for solving NPDEs. It also gives the basic idea of the Exp-function method. The detailed description of the fitness function used for converting the given NPDE into an optimization problem is presented. The procedural steps of stochastic global search algorithm PSO are also provided. The designed scheme has been successfully applied for the numerical solution of some well-known NPDEs including the Fisher equations, Burger-Fisher equation, Burger-Huxley equation, and Huxley equation. On the basis of simulation results and comparisons made with some popular classical methods it can be concluded that the proposed heuristic scheme is a promising and viable tool for solving such systems of NPDEs tackled in this dissertation.

CHAPTER 6

BERNSTEIN POLYNOMIALS BASED STOCHASTIC TECHNIQUE FOR SOLVING NONLINEAR ORDINARY DIFFERENTIAL EQUATIONs

In this chapter, a heuristic method based on the hybrid approach of Bernstein polynomials and evolutionary algorithms (EAs) is introduced for solving power-law fin type problem and Riccati equations. Earlier in chapter 3 we used log sigmoid basis functions for the approximation of numerical solution to NODEs, however in this chapter different basis functions such as Bernstein polynomial basis are investigated.

6.1 INTRODUCTION

Bernstein polynomials (B-polynomials) were originally introduced by S. N. Bernstein [211] more than a century ago, however these polynomials began to enjoy widespread use only after the advent of digital computers. In recent years, B-polynomials have received tremendous attention from researchers of diverse fields, and the methods based on these polynomials have been extensively used for solving wide variety of problems including systems of ODEs [212], [213], [214], [215], [216], integral equations [217], [218], [219], and integro-differential equations [220], [221].

Although B-polynomials have been utilized by many authors, but to the best of my knowledge nobody so far has used these polynomials in combination with EAs for solving NODEs. With the aim to seek some different basis functions for a better approximation and computational efficiency, besides their feasibility for hybridization with the EAs for solving NODEs, B-polynomials are investigated and explored in this dissertation.

This dissertation suggests a simple and straightforward, yet an efficient heuristic method for numerically solving NODEs. The scheme is based on the combination of B-polynomials and EAs such as GA and DE. B-polynomials basis with unknown coefficients are used to construct an approximate solution of the NODE. The NODE is transformed into an equivalent global error minimization problem. A trial solution is formulated using an exclusive fitness function with unknown coefficients. GA, DE, and GA-IPA are used to solve the minimization problem and to obtain the unknown coefficients. In the following sections we first introduce B-polynomials, then give the description of the proposed scheme and finally numerical applications are studied.

6.2 AN OVERVIEW OF BERNSTEIN POYLNOMIALS

The B-polynomials of nth degree are defined on the interval [0, T] as follows [212], [213], [214].

$$B_{i,n}(x) = \binom{n}{i}\frac{x^i(T-x)^{n-i}}{T^n} \qquad (6.1)$$

where

$$\binom{n}{i} = \frac{n!}{i!\,(n-i)!} \qquad (6.2)$$

There are $n+1$ nth degree B-polynomials. Usually we set $B_{i,n}(x) = 0,$ if $i < 0$ or $i > n$, for mathematical convenience. Each B-polynomial is positive, i.e., $B_{i,n}(x) > 0$ and also the sum of all the B-polynomials is unity for all real $x \in [0, T]$, i.e.

$$\sum\nolimits_{i=0}^{n} B_{i,n}(x) = 1 \tag{6.3}$$

B-polynomials defined over the interval form a complete basis [212], [213], [214] over the interval [0, T]. For detail of B-polynomials and their properties see [212], [213], [214] and references therein. These polynomials are quite easy to write and can be generated recursively. The ith nth degree B-polynomial and derivates of B-polynomials over the interval [0, T] are given by [212], [213], [214] [222].

$$B_{i.n}(x) = \frac{(T-x)}{T} B_{i.n-1}(x) + \frac{x}{T} B_{i-1.n-1}(x) \tag{6.4}$$

$$B'_{i,n}(x) = \frac{n}{T}(B_{i-1.n-1} - B_{i.n-1}) \tag{6.5}$$

$$B''_{i,n}(x) = \frac{n(n-1)}{T^2}(B_{i-2.n-2} - 2B_{i-1.n-2} + B_{i.n-2}) \tag{6.6}$$

6.3 B-POLYNOMIALS BASED HEURISTIC SCHEME

In this section, the description of the methodology hybridizing B-polynomials and EAs is presented. Since the systems of NODEs of first and second order have been solved in this chapter, therefore a second order ODE is considered to illustrate the methodology as follows

$$\frac{d^2y}{dx^2} = g(x, y, y'), \qquad 0 \le x \le T \tag{6.7}$$

subject to the following initial and/or boundary conditions respectively

$$\frac{d^k y}{dx^k} y(0) = b_k, \quad k = 0, 1 \tag{6.8}$$

$$\frac{d^k y}{dx^k} y(T) = c_k, \quad k = 0, 1 \tag{6.9}$$

where g represents the nonlinear function, prime denotes the derivation with respect to x, T is the upper bound of the solution span, b_k and c_k are real constants denoting the initial and boundary conditions respectively.

To solve (6.7), we assume that the approximate solution $\hat{y}(x)$ and its first and second derivates $\hat{y}'(x)$ and $\hat{y}''(x)$ can be expressed as linear combinations of Bernstein polynomials (B-polynomials) basis functions of degree n as follows.

$$\hat{y}(x) = \sum_{i=0}^{n} a_i B_{i,n}(x), \qquad n \geq 1 \tag{6.10}$$

$$\hat{y}'(x) = \sum_{i=0}^{n} a_i B'_{i,n}(x) \tag{6.11}$$

$$\hat{y}''(x) = \sum_{i=0}^{n} a_i B''_{i,n}(x) \tag{6.12}$$

where $(a_0, a_1, \dots, a_n)$ are real valued unknown coefficients, to be further determined, $B_{i,n}(x)$, $B'_{i,n}(x)$, and $B''_{i,n}(x)$ are given by (6.4) – (6.6) respectively.

To obtain the unknown coefficients $(a_0, a_1, \dots, a_n)$ and consequently the approximate solution $\hat{y}(x)$ given by (6.10), the given NODE is transformed into an equivalent optimization problem using a fitness function (ε_j) as follows.

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \quad j = 1,2,3 \dots \tag{6.13}$$

where j is the generation/iteration index of the algorithm.

The fitness function (FF) denoted as (ε_j) basically represents the global error associated with the given NODE along with its initial and/or boundary conditions. The first part in FF represents the mean of sum of square errors associated with the given NODE denoted by (ε_1) , the second part represents the mean of sum of square errors linked with the given initial and/or boundary conditions denoted by (ε_2), which are given respectively as follows.

$$\varepsilon_1 = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}''(x_i) - g(x_i, \hat{y}(x_i), \hat{y}'(x_i))\right)^2 \tag{6.14}$$

$$\varepsilon_2 = \frac{1}{2}\sum_{k=0}^{1}\left(\frac{d^k y}{dx^k}\hat{y}(0) - b\right)^2 + \frac{1}{2}\sum_{k=0}^{1}\left(\frac{d^k y}{dx^k}\hat{y}(T) - c\right)^2 \tag{6.15}$$

where $\hat{y}(x)$ $\hat{y}'(x)$, and , $\hat{y}''(x)$ are given by (6.10), (6.11), and (6.12) respectively, N is the total number of steps on the interval [0, T].

The minimization problem given by (6.13) is solved using GA and hybrid scheme GA-IP to find the optimal values of the unknown coefficients $(a_0, a_1, \dots, a_n)$. Consequently the approximate numerical solution $\hat{y}(x)$ of the given NODE is straightforward obtained by using the optimal values of coefficients in (6.10).

6.4 NUMERICAL APPLICATIONS

In this section, the methodology described above is applied to solve Riccati equations and power-law fin-type problem to assess and test its performance. In order to illustrate the validity and efficacy of the suggested scheme comparisons of numerical solutions are made with some state of the art classical methods as well as with the exact solutions. Moreover extensive analysis, through simulations is carried to show the reliability and accuracy of the proposed scheme.

6.4.1 RICCATI EQUATION

In this section, we used the proposed scheme to solve two different forms of well-known Riccati NODE of the following form [223], [224], [225], 226], 227], [228].

$$y'(x) = P(x) + Q(x)y + R(x)y^2 \qquad (6.16)$$

subject to the initial condition

$$y(0) = c \qquad (6.17)$$

where P(x), Q(x), and R(x) are real functions and c is an arbitrary constant.

The differential equation (6.16) introduced by the mathematician Count Jacopo Francesco Riccati (1676–1754) is imperative due to its existence in diverse fields of engineering and science such as optimal control, random processes, quantum mechanics, diffusion problems, etc. [223], [225], [227]. To date an incredibly great number of methods have been utilized for solving various forms of the Riccati equation. These methods also include OHAM [223], HPM [224], MHPM [225], ADM [226], VIM [227], and B-polynomials based collocation method (BPCM) [212].

Example 1: Consider the following Riccati nonlinear differential equation [223], [224], [225], [226], [227], [228].

$$y'(x) = -y^2(x) + 1, \qquad y(0) = 0 \qquad (6.18)$$

The exact solution of (6.18) is given by

$$y_{exact}(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \qquad (6.19)$$

To apply the proposed scheme we assume the approximate solution of (6.18) is given by (6.10) with $n = 7$ (B-polynomial of degree 7) as follows.

$$\hat{y}(x) = \sum_{i=0}^{7} a_i \, B_{i,n}(x), \qquad n \geq 1 \qquad (6.20)$$

The given NODE (6.18) is transformed into an optimization problem using the FF to find the unknown coefficients $(a_0, a_1, \dots, a_7)$. The numerical solution is obtained in the interval $x \in (0,1)$ with a step of 0.1, therefore FF is given by

$$\varepsilon_j = \frac{1}{11} \sum_{i=1}^{11} (\hat{y}'(x_i) + \hat{y}^2(x_i) - 1)^2 + \left(y(0)\right)^2 \qquad (6.21)$$

GA has been employed to solve the minimization problem given by (6.21) and to achieve the unknown coefficients $(a_0, a_1, \dots, a_7)$. The GA is executed according to the prescribed parameter values and settings given in Table 6.1. The number of unknown coefficients that need to be adapted is 8 therefore the size of chromosome for GA is also chosen to be 8. The optimal values of the unknown coefficients achieved by GA are given in Table 6.2 for various values of n.

Table 6.1 Parameter values and settings of GA for Riccati equation

Parameter Name	Parameter Settings/Value
Population size	[100 100]
Chromosome size	6, 7, 8 (for n = 5, 6, 7)
No. of generations	1000
Selection function	Stochastic uniform
Mutation function	Adaptive feasible
Crossover function	Heuristic
Crossover fraction	0.8

Table 6.2 Values of unknown coefficients obtained by GA for Riccati equation (example 1)

	Value			
	$x \in [0,1]$			$x \in [0,5]$
Coefficients	$n = 5$	$n = 6$	$n = 7$	$n = 7$
a_0	0.000001	0.000000	0.000000	-0.000007
a_1	0.199932	0.166663	0.142857	0.151233
a_2	0.401322	0.333444	0.285690	0.295511
a_3	0.562517	0.483054	0.419155	0.124550
a_4	0.677527	0.600108	0.532988	0.258568
a_5	0.761581	0.691610	0.626355	0.193401
a_6	---	0.761600	0.701599	0.211702
a_7	---	---	0.761595	0.209841

Once the unknown coefficients have been found, the numerical solution of (6.18) can be obtained at any value of x in the interval [0, 1] by using the values of these coefficients in (6.10). In Table 6.3 comparison of numerical results obtained by the proposed scheme is made with the exact solution, ADM, VIM, and OHAM solutions, also a comparison of absolute errors are provided in Table 6.4. It is seen from the comparison of Table 6.4 that the average absolute error in the interval $x \in [0,1]$ obtained by the proposed scheme is

9.14E-07, while the average absolute errors of ADM, VIM, and OHAM are 8.35E-04, 6.74E-04, and 1.57E-06 respectively, which shows that the proposed scheme gives more accurate results. It is seen that the classical methods like ADM and VIM give better approximation initially but as the value of x increases their accuracy also decreases, whereas the accuracy of the proposed scheme is seen fairly steady in the solution domain of x.

Table 6.3 Comparison of numerical results for Riccati equation (example 1)

		Proposed method $\hat{y}(x)$ with (n=7)		**Other methods**	
x	$y_{exact}(x)$	**GA**	**ADM**	**VIM**	**OHAM**
0	0.000000	0.000000	0.000000	0.000000	0.000000
0.1	0.099668	0.099667	0.099668	0.099668	0.099668
0.2	0.197375	0.197374	0.197375	0.197376	0.197377
0.3	0.291313	0.291311	0.291313	0.291321	0.291315
0.4	0.379949	0.379947	0.379949	0.380009	0.379953
0.5	0.462117	0.462115	0.462121	0.462401	0.462120
0.6	0.53705	0.537049	0.537078	0.53805	0.537051
0.7	0.604368	0.604368	0.604514	0.607262	0.604369
0.8	0.664037	0.664036	0.664641	0.67129	0.664038
0.9	0.716298	0.716297	0.718392	0.732603	0.716300
1.0	0.761594	0.761595	0.767901	0.795262	0.761595

Table 6.4 Comparison of absolute errors for Riccati equation (example 1)

x	GA	ADM	VIM	OHAM
0	1.66E-09	0	0	0
0.1	1.17E-06	8.80E-14	1.50E-11	1.844E-07
0.2	1.00E-06	1.79E-10	4.22E-09	1.180E-06
0.3	1.22E-06	1.51E-08	1.56E-07	2.695E-06
0.4	2.07E-06	3.49E-07	1.98E-06	3.540E-06
0.5	1.95E-06	3.93E-06	1.38E-05	2.997E-06
0.6	7.76E-07	2.81E-05	6.62E-05	1.747E-06
0.7	1.38E-07	1.46E-04	2.43E-04	1.067E-06
0.8	6.32E-07	6.05E-04	7.36E-04	1.234E-06
0.9	5.63E-07	2.09E-03	1.92E-03	1.649E-06
1.0	5.33E-07	6.31E-03	4.43E-03	9.540E-07

The influence of the change in degree of B-polynomials i.e. change in n on the performance of the proposed scheme is analyzed next. We used $n = 5, 6$ in (6.10) for evaluating the performance, therefore the number of unknown coefficients to be tailored are 6 and 7 respectively for $n = 5$ and $n = 6$. GA has been used for solving the FF given by (6.21) and to achieve the unknown coefficients corresponding to each value of n mentioned. GA is implemented with the same parameter settings prescribed in Table 6.1 except a change in the chromosome size which is now chosen as 6 for $n = 5$ and 7 for $n = 6$ respectively.

The optimal values of the unknown coefficients acquired by GA are given in Tables 6.2 and using these values the approximate solution $\hat{y}(x)$ is obtained from (6.10). The approximate solutions obtained using the proposed method for different values of n = 5, 6, are presented in Table 6.5, also exact solution and solution with n = 7 obtained by the proposed method are given for comparison. From the comparison of Table 6.5 the improvement in the approximate solution is observed with the increase in *n* (i.e. increase in degree of B-polynomials).

To further investigate the effect of change in the degree of B-polynomials (i.e. change in *n*), a comparison of average absolute errors, computational time, and number of generations utilized by GA is given in Table 6.6. From the comparison of Table 6.6 it is observed that the accuracy of the solution drastically improves with the increase in *n* from 5 to 7, but at the cost of high computational time and more number of generations.

Table 6.5 Comparison of numerical results for various values of *n*

		Proposed method $\hat{y}(x)$			**Absolute errors** $\lvert(y_{exact} - \hat{y}(x))\rvert$		
x	$y_{exact}(x)$	$n = 5$	$n = 6$	$n = 7$	$n = 5$	$n = 6$	$n = 7$
0.0	0.000000	0.000001	0.000000	0.000000	7.90E-07	4.78E-07	1.66E-09
0.1	0.099668	0.099713	0.099673	0.099667	4.54E-05	5.43E-06	1.17E-06
0.2	0.197375	0.197464	0.197382	0.197374	8.85E-05	6.77E-06	1.00E-06
0.3	0.291313	0.291373	0.291313	0.291311	6.06E-05	6.05E-07	1.22E-06
0.4	0.379949	0.379956	0.379947	0.379947	6.79E-06	2.43E-06	2.07E-06
0.5	0.462117	0.462102	0.462119	0.462115	1.53E-05	1.64E-06	1.95E-06
0.6	0.537050	0.537061	0.537056	0.537049	1.10E-05	6.23E-06	7.76E-07
0.7	0.604368	0.604422	0.604371	0.604368	5.42E-05	3.52E-06	1.38E-07
0.8	0.664037	0.664100	0.664033	0.664036	6.37E-05	3.77E-06	6.32E-07
0.9	0.716298	0.716317	0.716295	0.716297	1.89E-05	2.37E-06	5.63E-07
1.0	0.761594	0.761581	0.761600	0.761595	1.31E-05	6.08E-06	5.33E-07

Moreover, in Table 6.7 we show a comparison of our results with the results reported in [212] using BPCM with same degree of B-polynomials (n = 5). It can be clearly seen from the comparison of Table 6.7 that the suggested scheme gives numerical solutions fairly comparable indeed slightly better than BPCM, which further confirms the accuracy and viability of our scheme.

Table 6.6 Effect of change in the degree n

Degree of B-polynomials (n)	No. of Generations	Computational Time (sec)	Average Absolute Error
5	240	11	3.439E-05
6	281	15	3.575E-06
7	339	17	9.140E-07

Table 6.7 Comparison of numerical results with BPCM (with $n = 5$)

		Proposed method		Absolute Error	
x	$y_{exact}(x)$	$\hat{y}(x)$ (with $n = 5$)	BPCM (with $n = 5$)	Proposed method	BPCM
0.0	0.000000	0.000001	0.000000	7.902E-07	0.000000
0.2	0.197375	0.197464	0.197427	8.855E-05	5.173E-05
0.4	0.379949	0.379956	0.379975	6.795E-06	2.597E-05
0.6	0.537050	0.537061	0.537090	1.100E-05	4.066E-05
0.8	0.664037	0.664100	0.664024	6.369E-05	1.239E-05
1.0	0.761594	0.761581	0.762346	1.312E-05	7.514E-04

We now investigate the reliability and accuracy of the proposed scheme in the larger interval $x \in [0, 5]$. The FF is formulated and GA is used for its minimization to achieve the values of the unknown coefficients. GA is executed with the same parameter settings as prescribed in Table 6.2. The values of unknown parameters achieved by GA are provided in Table 6.2. The comparison of our approximate solution is graphically made with two well-known classical methods ADM and VIM and the exact solution in Fig. 6.1.

The comparison of results in Fig. 6.1 evidently shows that the proposed scheme is quite capable of yielding the numerical solution of the Riccati equation (6.18) with a significant accuracy in the larger domain of x, while ADM and VIM diverge after $x = 1.4$ and $x = 1.1$ respectively.

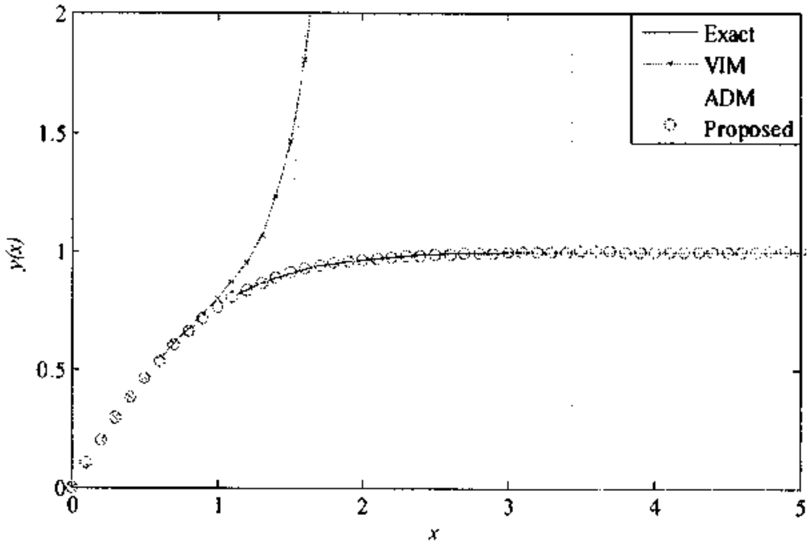

Fig 6.1 Comparison of numerical solutions for $x \in [0, 5]$

Example 2: We now consider another Riccati NODE as follows [223], [224], [225], [226], [227], [228].

$$y'(x) = -y^2(x) + 2y(x) + 1, \qquad y(0) = 0 \qquad (6.22)$$

the exact solution is given by

$$y_{exact}(x) = 1 + \sqrt{2}tanh\left(\sqrt{2}t + \frac{1}{2}log\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)\right) \qquad (6.23)$$

The approximate solution of (6.22) is obtained using the proposed scheme with $n = 5, 7$ (B-polynomial of degree 5, 7), in the interval $x \in (0,1)$ and $x \in (0,2)$ with a step of 0.1 for each interval of x. The FF is formulated for each interval of x separately, for instance FF for the interval $x \in (0, 2)$ given by

$$\varepsilon_j = \frac{1}{21}\sum_{i=1}^{21}\left((\hat{y}'(x_i) + \hat{y}^2(x_i) - 2\hat{y}(x_i) - 1)\right)^2 + \left(y(o)\right)^2 \quad (6.24)$$

The FF given by (6.23) is solved using GA for finding the optimal values of unknown coefficients corresponding to the minimum ε_j . GA is implemented according to the prescribed parameter values and settings given in Table 6.1. The values of the unknown coefficients achieved by GA are provided in Table 6. 8.

Table 6.8 Optimal values of coefficients obtained by GA for Riccati equation (example 2)

	Value		
	$x \in [0,1]$		$x \in [0,2]$
Coefficient	$n = 5$	$n = 7$	$n = 7$
a_0	0.000076	0.000002	0.000330
a_1	0.200673	0.142854	0.289078
a_2	0.487779	0.334228	0.697442
a_3	0.969543	0.577790	1.725555
a_4	1.384721	0.893010	2.115500
a_5	1.689281	1.204383	2.201938
a_6	---	1.471732	2.315525
a_7	---	1.689531	2.358231

Using the values of unknown coefficients in (6.10) yields the approximate numerical solution of (6.22) straightforward.

The numerical solutions obtained by the proposed method are presented in Table 6.9 for$x \in (0,1)$, Table 6.10 for $x \in (0,2)$, and in Fig. 6.2 graphically for $x \in (0,2)$, also the exact solutions and the approximate solutions reported by methods including OHAM [223], MHPM [224], VIM [225], MVIM [226], and BPCM [212] are given for the sake of comparison.

Table 6.9 Comparison of numerical results for $x \in (0, 1)$

		Proposed method $\hat{y}(x)$		**Other methods**			
				BPCM	**MHPM**	**VIM**	**OHAM**
x	$y_{exact}(x)$	$n = 5$	$n = 7$	$n = 5$			
0	0	0.000076	0.000002	0	0	0.000000	0
0.1	0.110295	0.109928	0.110350	---	0.110294	0.110295	0.110328
0.2	0.241977	0.241161	0.242024	0.241283	0.241965	0.241977	0.242273
0.3	0.395105	0.394495	0.395138	---	0.395106	0.395113	0.396178
0.4	0.567812	0.567624	0.567875	0.567187	0.568115	0.567845	0.570281
0.5	0.756014	0.755923	0.756097	---	0.757564	0.756086	0.759942
0.6	0.953566	0.953149	0.953631	0.952653	0.958259	0.953666	0.957123
0.7	1.152949	1.152145	1.152997	---	1.163459	1.153037	1.150521
0.8	1.346364	1.345546	1.346429	1.345970	1.365240	1.346379	1.326366
0.9	1.526911	1.526482	1.526980	---	1.554960	1.526411	1.468058
1.0	1.689498	1.689281	1.689531	1.682362	1.723810	1.686027	1.546858

Further Table 6.11 and Table 6.12 show a comparison of absolute errors yielded by our method and other methods including MHPM, VIM, MVIM, OHAM, and BPCM in contrast to exact solutions. The comparison clearly reveals that the proposed method provides the solution with significantly smaller absolute errors as compared to OHAM, MHPM, and VIM. Moreover, from Fig. 6.2 it is noticeable that VIM gives good accuracy only in the short interval $x \in (0,1.4)$ after that it diverges drastically, while our method provides the approximate solution with a high degree of accuracy in the interval$x \in (0,2)$. The comparison further reveals that numerical results by our method with $n = 5$ are fairly comparable to those reported by BPCM with $n = 5$.

Table 6.10 Comparison of numerical results for $x \in (0,2)$

		Proposed method **(with n=7)**	**Other methods**	
x	$y_{exact}(x)$	$\hat{y}(x)$	**MHPM**	**VIM**
0.2	0.241977	0.239594	0.239615	0.241978
0.4	0.567812	0.567019	0.562623	0.567846
0.6	0.953566	0.954069	0.946841	0.953666
0.8	1.346364	1.344518	1.340564	1.346379
1.0	1.689498	1.686834	1.686382	1.686027
1.2	1.951360	1.950877	1.950949	1.915051
1.4	2.131327	2.132003	2.132583	2.179132
1.6	2.246286	2.245645	2.248141	-50.982298
1.8	2.316325	2.315479	2.318124	-5338.782860
2.0	2.357772	2.358231	2.359242	-286352.732500

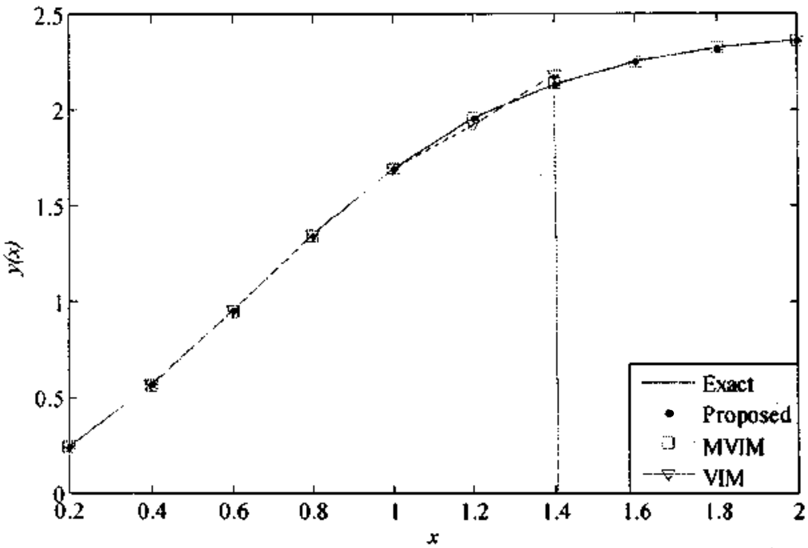

Fig 6.2 Comparison of numerical solutions for $x \in (0, 2)$

Table 6.11 Comparison of absolute errors for $x \in (0, 1)$

	Proposed method		**Other methods**			
			BPCM			
x	$n = 5$	$n = 7$	$n = 5$	**MHPM**	**VIM**	**OHAM**
0	7.58E-05	2.04E-06	0	0	0	0
0.1	3.67E-04	5.48E-05	---	1.20E-06	1.97E-07	3.24E-05
0.2	8.16E-04	4.68E-05	6.94E-04	1.18E-05	2.00E-07	2.97E-04
0.3	6.10E-04	3.36E-05	---	1.15E-06	8.15E-06	1.07E-03
0.4	1.88E-04	6.24E-05	6.25E-04	3.03E-04	3.28E-05	2.47E-03
0.5	9.09E-05	8.30E-05	---	1.55E-03	7.16E-05	3.93E-03
0.6	4.17E-04	6.47E-05	9.13E-04	4.69E-03	9.98E-05	3.56E-03
0.7	8.04E-04	4.77E-05	---	1.05E-02	8.80E-05	2.43E-03
0.8	8.17E-04	6.51E-05	3.94E-04	1.89E-02	1.53E-05	2.00E-02
0.9	4.29E-04	6.90E-05	---	2.80E-02	5.00E-04	5.89E-02
1.0	2.18E-04	1.16E-05	7.13E-03	3.43E-02	3.47E-03	1.43E-01

Table 6.12 Comparison of absolute errors for $x \in (0, 2)$

x	**Proposed** (with n = 7)	**MVIM**	**VIM**
0.2	2.38E-03	2.362E-03	1.033E-06
0.4	7.93E-04	5.189E-03	3.335E-05
0.6	5.03E-04	6.725E-03	9.982E-05
0.8	1.85E-03	5.800E-03	1.545E-05
1.0	2.66E-03	3.116E-03	3.471E-03
1.2	4.83E-04	4.109E-04	3.631E-02
1.4	6.76E-04	1.256E-03	4.780E-02
1.6	6.41E-04	1.855E-03	5.323E+01
1.8	8.45E-04	1.799E-03	5.341E+03
2.0	4.59E-04	1.470E-03	2.864E+05

6.4.2 NONLINEAR FIN PROBLEM

Fins are most commonly encountered in numerous engineering applications to aid the transfer of heat. In many situations the heat transfer coefficient is non-uniform and varies in a nonlinear manner with the temperature, which gives rise to the power-law type form [229], [230], [231], [232]. Accordingly, the equation of temperature happens to be highly nonlinear. The analytical solutions of such problems are not easy to obtain, therefore approximate analytical and numerical methods are being utilized for handling these problems. In view of the practical importance of fin problems in engineering applications like heat exchangers, transformers, and electronic devices, such problems have been considered by a number of authors and numerous methods have been reported to tackle these problems, see for example [233], [234], [235], [236] , [237], [238], [239], [240] and references therein.

The fin problem investigated here is a one-dimensional steady-state heat conduction equation given as follows in the dimensionless form [230], [231], [232], [233]

$$y'' - My^m = 0 \tag{6.25}$$

subject to the following boundary conditions

$$y'(0) = 0 \tag{6.26}$$

$$y(1) = 1 \tag{6.27}$$

where prime denotes the differentiation with respect to the dimensionless coordinate x, y is the dimensionless temperature, M is the convective-conductive parameter of the fin, and the exponent m depends on the heat transfer mode [230], [231], [232], [233]. The physical values of m that are of practical importance are 5/4 and 4/3 for laminar and turbulent convection, 3 for nucleate boiling, and 4 for radiation [230], [231], [232], [233]. The fin problem given by (6.25) – (6.27), has been considered by many authors and several approximate methods including ADM [232], Homotopy analysis method (HAM) [231], VIM [233], and Runge-Kutta shooting method (RKSM) [230] have been utilized for its solution.

Our aim is to investigate the numerical solution of (6.26) using the proposed method. The numerical solution will be obtained for various values of the parameters M, and m that govern the problem under consideration and also denote the degree of nonlinearity.

Case (1): $M = 1$, $m = 1$

For this case we have analytical solution of (6.25) given by [230]

$$y_{anl}(x) = \frac{e}{e^2 + 1}(e^x + e^{-x}) \tag{6.28}$$

accordingly we get the first derivate as follows

$$y'(x) = \frac{e}{e^2+1}(e^x - e^{-x}) \tag{6.29}$$

We assume that the approximate solution is expressed by (6.10) with $n = 6$ (B-polynomials of degree 6). The unknown coefficients $(a_0, a_1, \ldots, a_6)$ in (6.10) are obtained by transforming the given problem (6.25) – (6.27) into an equivalent global error minimization problem using FF as follows.

$$\varepsilon_1 = \frac{1}{11}\sum_{i=1}^{11}\left(\hat{y}''(x_i) - \hat{y}(x_i)\right)^2 \tag{6.30}$$

$$\varepsilon_2 = \frac{1}{2}\left(\left(\hat{y}'(0)\right)^2 + (\hat{y}(1) - 1)^2\right) \tag{6.31}$$

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \tag{6.32}$$

where ε_1 represents the mean of sum of square errors of (6.25) in the interval [0,1] with a step of 0.1, ε_2 represents the mean of sum of square errors of (6.26) and (6.27), and $\hat{y}(x)$, $\hat{y}'(x)$ and $\hat{y}''(x)$ are given by (6.10) - (6.12) respectively.

The global error minimization problem given by (6.32) is solved using GA, IPA, and GA-IPA to find the optimal values of the unknown coefficients, which consequently gives the approximate numerical solution of (6.25). GA and IPA are implemented with the parameter values and settings prescribed in Table 6.13. The number of unknown coefficients $(a_0, a_1, \ldots, a_6)$ that need to be tailored is 7, therefore the size of chromosome is chosen equal to 7.

Table 6.13 Parameter settings and values of GA and IPA for fin problem

GA		IPA	
Parameter Name	**Setting/Value**	**Parameter Name**	**Setting/value**
Population size	[100 100]	Start point	Random/Optimal values from GA
No. of generations	1000	Maximum iterations	500
Selection function	Stochastic uniform	Maximum function evaluations	3000
Mutation function	Adaptive feasible	Function tolerance	1e-15
Crossover function	Heuristic	Nonlinear constraint tolerance	1e-15
Function tolerance	1e-15	Hessian	BFGS
Nonlinear constraint tolerance	1e-15	Derivative type	Central differences
Bounds	-10, +10	Bounds	-10, +10

The unknown coefficients found by GA, IPA and GA-IPA corresponding to one of the best chromosome are provided in Table 6.14. Using the values of these coefficients in (6.10) we can obtain the approximate numerical solution $\hat{y}(x)$ of (6.25) at any value of x in the interval [0, 1].

Table 6.14 Optimal values of unknown coefficients with $M = 1$ and $m = 1$

	Value		
Coefficient	**GA**	**IPA**	**GA-IPA**
a_0	0.648033	0.648055	0.648055
a_1	0.648037	0.648055	0.648055
a_2	0.669644	0.669656	0.669656
a_3	0.712850	0.712858	0.712858
a_4	0.779462	0.779468	0.779468
a_5	0.873067	0.873068	0.873068
a_6	1.000005	1.000000	1.000000

The numerical solution obtained by our method with GA and GA-IPA are presented in Table 6.15, also the exact solution and solutions obtained by HAM [231] and RKSM [230] are presented for comparison.

Table 6.15 Comparison of numerical solutions with $M = 1$ and $m = 1$

		Proposed method $\hat{y}(x)$		Other methods	
x	$y_{exact}(x)$	**GA**	**GA-IPA**	**HAM**	**RKSM**
0	0.648054	0.648033	0.648055	0.648054	---
0.1	0.651297	0.651279	0.651298	0.651297	0.651297
0.2	0.661059	0.661043	0.661059	0.661059	0.661059
0.3	0.677436	0.677423	0.677436	0.677436	0.677436
0.4	0.700594	0.700582	0.700594	0.700594	0.700596
0.5	0.730763	0.730754	0.730763	0.730763	0.730763
0.6	0.768246	0.768239	0.768246	0.768246	0.768246
0.7	0.813418	0.813414	0.813418	0.813418	0.813418
0.8	0.866730	0.866729	0.866730	0.866731	0.866730
0.9	0.928718	0.928720	0.928718	0.928718	0.928717
1.0	1.000000	1.000005	1.000000	1.000000	1.000000

Further in Table 6.16 comparison of absolute errors is shown to prove the accuracy of results From the comparison approximate solutions by our method are found fairly close to the exact solution with an average absolute error of 1.27E-07 in the interval [0, 1]. The comparison further shows that solutions by the proposed method are quite comparable with HAM solutions which also give an average absolute error of 1.13E-07, however our method gives comparatively more error in comparison to RKSM which gives an average absolute error of 7.18E-08. But as illustrated earlier in Riccati problem above we can use

B-polynomials of higher degree to improve the accuracy of the solutions but at the cost of computational time. Moreover, the improved performance of GA-IPA in comparison to GA is also evident from the comparison.

Table 6.16 Comparison of absolute errors with $M = 1$ and $m = 1$

x	GA	GA-IPA	HAM	RKSM
0	2.148E-05	2.728E-07	1.160E-07	---
0.1	1.874E-05	2.647E-07	1.173E-07	5.400E-08
0.2	1.610E-05	2.074E-07	1.190E-07	8.000E-08
0.3	1.357E-05	1.366E-07	1.210E-07	8.000E-09
0.4	1.113E-05	1.053E-07	1.260E-07	2.029E-06
0.5	8.732E-06	1.178E-07	1.310E-07	2.600E-08
0.6	6.328E-06	1.365E-07	1.380E-07	1.000E-07
0.7	3.846E-06	1.140E-07	1.420E-07	1.380E-07
0.8	1.179E-06	4.979E-08	1.370E-07	3.320E-07
0.9	1.756E-06	5.942E-09	1.040E-07	3.570E-07
1.0	4.886E-06	5.433E-11	0.000E+00	5.000E-07

We now investigate the approximate solution of (6.25) with different value of M and m and show the validity and efficacy of the proposed method. As it can be seen from (6.25) that with larger values of M and m the nonlinearity increases, hence it is vital to show that the proposed method provides the numerical solution for higher values of M for which the problem is strongly nonlinear. We consider following test case studies.

Case (2a): $M = 1$, and $m = 5/4$, $4/3$, 3, and 4

Case (2b): $M = 2$, and $m = 5/4$, $4/3$, 3, and 4

Case (2c): $M = 5$, and $m = 5/4$, $4/3$, 3, and 4

To apply the proposed method we need to develop the FF for each value of *M* and *m* exclusively, For instance, FF with *M* = *1* and *m* = *5/4* is given by

$$\varepsilon_1 = \frac{1}{11}\sum_{i=1}^{11}\left(\hat{y}''(x_i) - \hat{y}^{\left(\frac{5}{4}\right)}(x_i)\right)^2 \qquad (6.33)$$

$$\varepsilon_2 = \frac{1}{2}\left(\left(\hat{y}'(0)\right)^2 + (\hat{y}(1) - 1)^2\right) \qquad (6.34)$$

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \qquad (6.35)$$

Similarly FF for other values of *M* and *m* is formulated. GA and GA-IPA are used to solve the FF such as given by (6.35) and to find the optimal values of the unknown coefficients. The GA and GA-IPA are implemented with the same parameter values as prescribed in Table 6.13 throughout for this problem for all the cases. The unknown coefficients acquired by GA, and GA-IPA are provided in Table 6.17, Table 6.18 and Table 6.19, for case (2a), case (2b), and case (2c) respectively.

Table 6.17 Optimal values of for *M* = *1*, and *m* = 5/4, 4/3, 3, and 4

	GA				GA-IPA			
	m=5/4	*m=4/3*	*m=3*	*m=4*	*m=5/4*	*m=4/3*	*m=3*	*m=4*
a_0	0.6677	0.6738	0.7517	0.7792	0.6679	0.6738	0.7516	0.7792
a_1	0.6677	0.6738	0.7517	0.7792	0.6679	0.6738	0.7516	0.7792
a_2	0.6879	0.6935	0.7658	0.7915	0.6880	0.6935	0.7658	0.7915
a_3	0.7281	0.7328	0.7940	0.8157	0.7283	0.7329	0.7939	0.8157
a_4	0.7904	0.7938	0.8389	0.8553	0.7906	0.7939	0.8388	0.8553
a_5	0.8784	0.8802	0.9028	0.9110	0.8786	0.8803	0.9027	0.9110
a_6	0.9998	0.9999	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000

Table 6.18 Optimal values of for $M = 2$, and $m = 5/4$, 4/3, 3, and 4

	GA				GA-IPA			
	$m=5/4$	$m=4/3$	$m=3$	$m=4$	$m=5/4$	$m=4/3$	$m=3$	$m=4$
a_0	0.4987	0.5102	0.6494	0.6944	0.4987	0.5102	0.6494	0.6944
a_1	0.4987	0.5102	0.6494	0.6944	0.4987	0.5102	0.6494	0.6944
a_2	0.5267	0.5374	0.6677	0.7100	0.5267	0.5374	0.6677	0.7100
a_3	0.5825	0.5916	0.7031	0.7392	0.5825	0.5916	0.7031	0.7392
a_4	0.6714	0.6782	0.7642	0.7930	0.6714	0.6782	0.7641	0.7930
a_5	0.8024	0.8058	0.8488	0.8631	0.8024	0.8058	0.8487	0.8631
a_6	1.0000	1.0000	0.9999	0.9995	1.0000	1.0000	0.9999	0.9995

Table 6.19 Optimal values of for $M = 5$, and $m = 5/4$, 4/3, 3, and 4

	GA				GA-IPA			
	$m=5/4$	$m=4/3$	$m=3$	$m=4$	$m=5/4$	$m=4/3$	$m=3$	$m=4$
a_0	0.2722	0.2900	0.5060	0.5730	0.2722	0.2900	0.5059	0.5730
a_1	0.2722	0.2900	0.5059	0.5729	0.2722	0.2900	0.5059	0.5729
a_2	0.3050	0.3221	0.5281	0.5916	0.3050	0.3221	0.5280	0.5916
a_3	0.3693	0.3845	0.5628	0.6165	0.3693	0.3845	0.5628	0.6165
a_4	0.4828	0.4958	0.6518	0.6979	0.4828	0.4958	0.6518	0.6979
a_5	0.6581	0.6646	0.7409	0.7625	0.6581	0.6646	0.7409	0.7625
a_6	1.0000	0.9999	0.9916	0.9788	1.0000	0.9999	0.9916	0.9788

The approximate solution is straightforward obtained by using the values of the coefficients in (6.10). The approximate numerical solutions for dimensionless temperature y and temperature gradient y' are depicted in Fig. 6.3 and Fig. 6.4 for case (2a), Fig. 6.5 and Fig. 6.6 for case (2b), and Fig. 6.7 and Fig. 6.8 for case (2c) respectively.

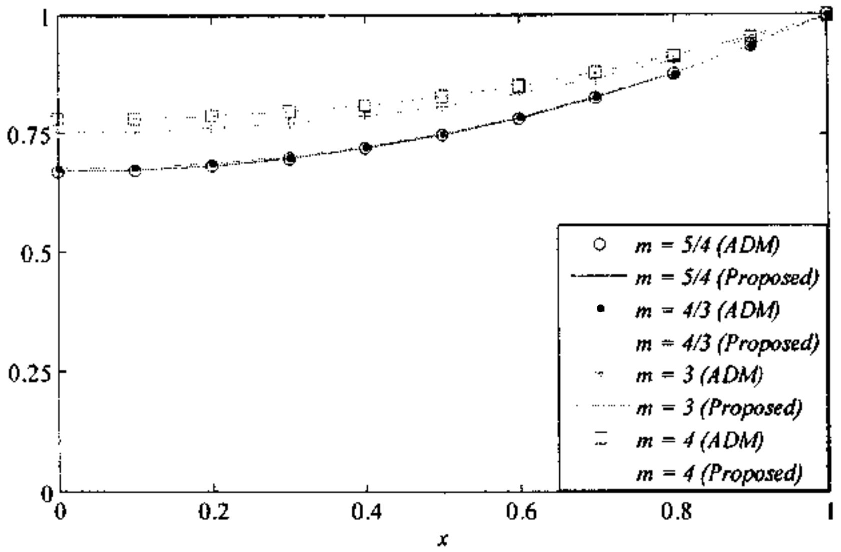

Fig. 6.3 Comparison of numerical solution for various values of *m* with *M = 1*

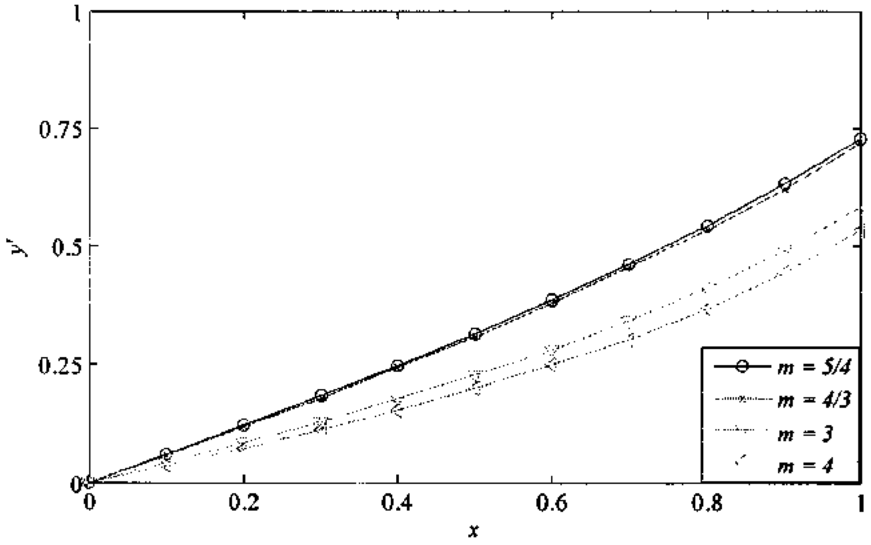

Fig. 6.4 Numerical solution *y'* for various values of *m* with *M = 1*

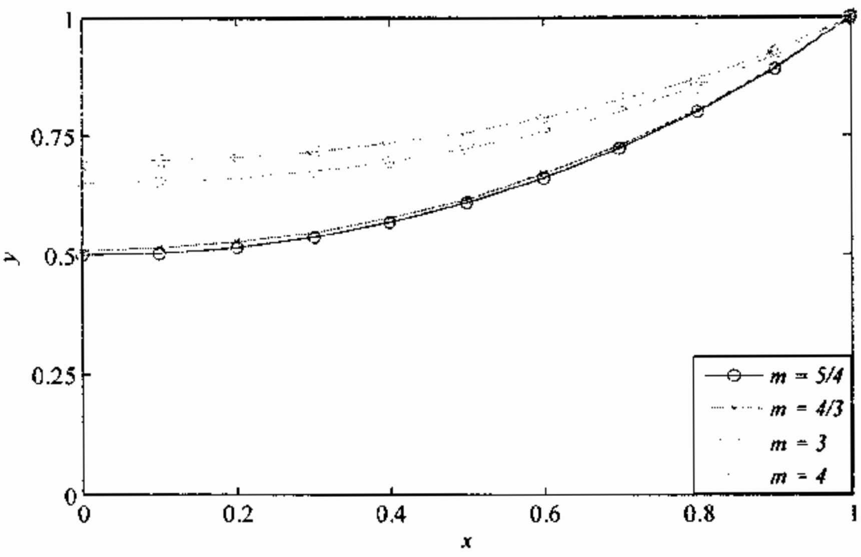

Fig 6.5 Numerical solution for various values of *m* with *M* = *2*

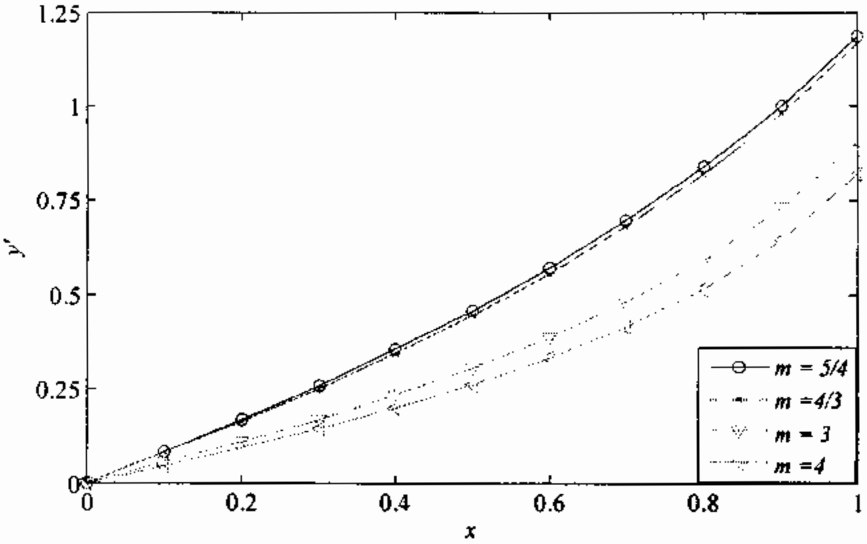

Fig 6.6 Numerical solution *y'* for various values of *m* with *M* = *2*

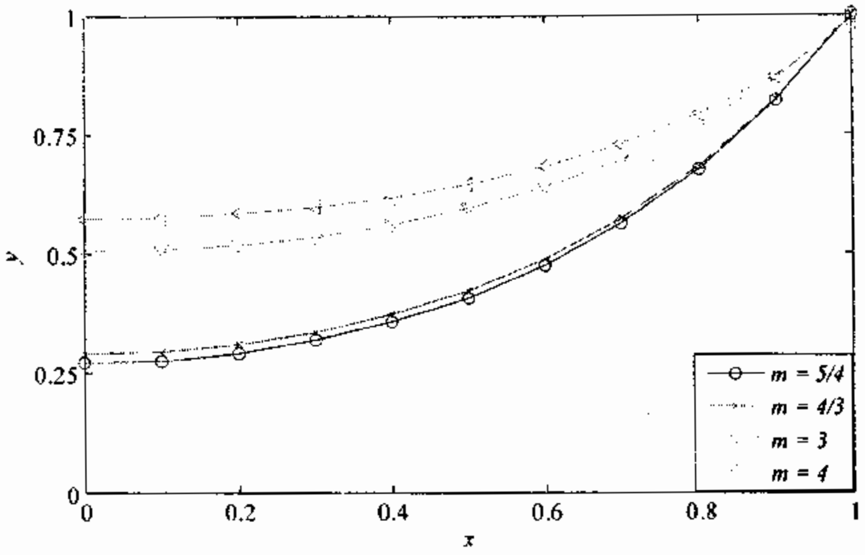

Fig 6.7 Numerical solution for various values of *m* with *M* = 5

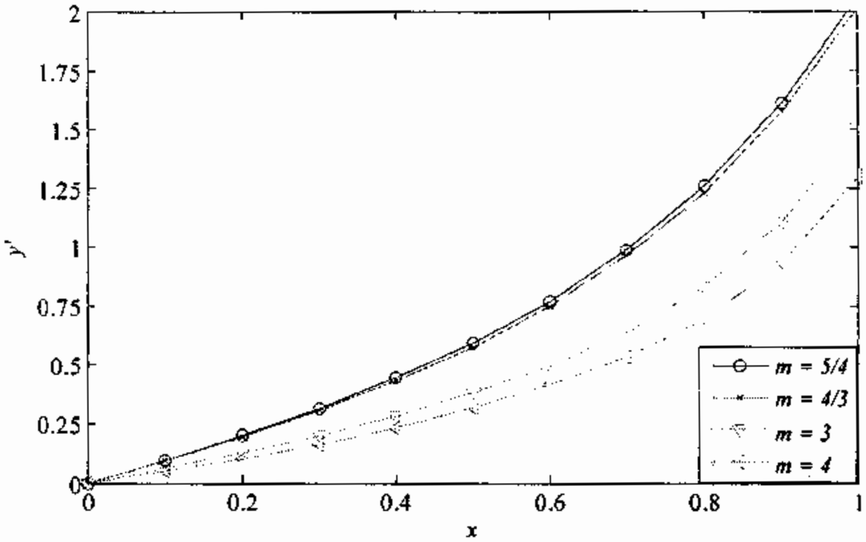

Fig 6.8 Numerical solution *y'* for various values of *m* with *M* = 5

In order to show the validity of our results, we also compare our results with some of the reported results by ADM [232] and RKSM [230]. In Fig. 6.3 comparison of numerical solutions is made with ADM solution for case (2a) and in Table 6.20 comparison is presented with RKSM for $M = 2$ and $m = 4/3$ respectively. It can be seen that proposed solutions are quite comparable with ADM and RKSM solutions. Further in Table 6.21 and Table 6.22 a comparison of $y(0)$ and $y'(1)$ values with RKSM is presented. It is seen that the proposed results are quite similar to RKSM results, which confirms the validity as well as efficacy of the proposed method.

Table 6.20 Comparison of numerical results for fin problem with $M = 2$ and $m = 4/3$

	$y(x)$		$y'(x)$	
x	**RKSM**	**Proposed**	**RKSM**	**Proposed**
0.0	0.510179	0.510192	0.000000	0.000000
0.1	0.514263	0.514275	0.081822	0.081804
0.2	0.526601	0.526611	0.165397	0.165354
0.3	0.547461	0.547466	0.252545	0.252512
0.4	0.577294	0.577298	0.345220	0.345218
0.5	0.616761	0.616765	0.445584	0.445602
0.6	0.666749	0.666755	0.556096	0.556102
0.7	0.728413	0.728418	0.679604	0.679580
0.8	0.803215	0.803216	0.819474	0.819433
0.9	0.892986	0.892985	0.979726	0.979714
1	1.000000	1.000000	1.165226	1.165244

Table 6.21 Comparison of $y(0)$ and $y'(1)$ values for different m and $M = 1$

	$y(0)$		$y'(1)$	
m	**RKSM**	**Proposed**	**RKSM**	**Proposed**
5/4	0.667898	0.667899	0.728303	0.728304
4/3	0.673814	0.673816	0.718308	0.718309
3	0.751622	0.751645	0.583459	0.583489
4	0.779145	0.779183	0.533989	0.534037

Table 6.22 Comparison of $y(0)$ and $y'(1)$ values for different m and $M = 2, 5$

	$M = 2$				$M = 5$			
	$y(0)$		$y'(1)$		$y(0)$		$y'(1)$	
m	**RKSM**	**Proposed**	**RKSM**	**Proposed**	**RKSM**	**Proposed**	**RKSM**	**Proposed**
5/4	0.4987	0.4987	1.1858	1.1858	0.2720	0.2721	2.0510	2.0512
4/3	0.5101	0.5101	1.1652	1.1652	0.2898	0.2899	2.0118	2.0120
3	0.6492	0.6493	0.9068	0.9068	0.5065	0.5059	1.5281	1.5042
4	0.6943	0.6944	0.8190	0.8184	0.5755	0.5729	1.3688	1.2981

6.5 CONCLUSION

The Bernstein polynomials based heuristic technique has been proposed for numerically solving NODEs. The effectiveness of the proposed technique has been demonstrated by numerically solving Riccai NODEs. The presented method has shown supremacy on some of the well known classical methods like VIM, MHPM, ADM and OHAM in terms of accuracy.

Further the efficacy and reliability of the proposed method have been illustrated by numerical solving the power-law fin type problem for several values of the convective-conductive parameter M and the exponent m involved in the governing equation. Moreover, the effect of nonlinearity due to M and m has been investigated. The numerical results by the proposed method have been found quite comparable to other methods like ADM and RKSM and also in a good agreement with the exact solutions. It can be concluded that the proposed method based on the hybrid approach of Bernstein polynomials and evolutionary algorithms is a promising tool and viable for solving such highly nonlinear problems. Furthermore the proposed method can give the numerical solution of the given NODE on the continuous values in the solution domain.

6.6 COMPARISON BETWEEN TWO HEURISTIC SCHEMES

In this section, the comparative analysis between two heuristic schemes used in this dissertation for solving NODEs is presented. In chapter 3, we used log sigmoid based EA scheme, and in the current chapter B-polynomials based EA scheme has been used for solving NODEs. The aim is to investigate the performance such as computational time and accuracy of two schemes.

The two schemes including the log sigmoid based EA approach and B-polynomials based EA approach are applied for solving the same problems. We consider Riccati equations (6.18) and (6.22), and fin problem (6.25) (with $M = 1$ and $m = 1$) solved above in section (6.4), for the purpose of comparison between two schemes. The approximate solutions of these problems are obtained in the interval [0, 1], with various values of n (i.e different degree of B-polynomials) and m (i.e. number of basis functions).

The FF corresponding to each case is formulated and GA is used for solving the FF and to achieve the unknown coefficients. The GA is implemented with the same parameter values and settings for both the schemes as prescribed in Table 6.23 for the minimization of the FF that represents the global error (ε_j) of the given problem.

The values of unknown coefficients as well as the approximate solutions have been omitted here, because our focus is to give a comparison of convergence speed of GA and accuracy of the approximate solutions for two schemes. The number of generations and the computational time of GA utilized to achieve the minimum fitness have been recorded for each problem. Further, the average absolute errors have been computed in the interval [0, 1] for each problem.

In Table 6.24, Table 6.25, and Table 6.26, comparison of two schemes such as log sigmoid based EA scheme and B-polynomials based EA scheme are provided. For the purpose of comparison number of generations, computational time, and the average absolute errors are shown corresponding to various values of *n* (degree of B-polynomials) and *m* (number of basis functions).

From the comparison of simulation results it is observed that overall B-polynomials based EA scheme is computationally efficient than log sigmoid based EA scheme, also the average absolute errors obtained by this scheme are relatively smaller than log sigmoid based EA scheme.

It is also seen from the comparison that increase in *m* from 5 to 7 does not give any significant improvement in the solutions but relative increase in computational time, while an increase in *n* shows relative improvement in solutions particularly for example 1 and example 2, but at the cost of increased computational time. However, it may be worth to mention here that the change in the input parameters of GA such as selection function, crossover fraction, population size etc. have great influence on its performance on different problems.

The comparative analysis of two schemes shown here corresponds to their performance on the considered problem (6.18), (6.22), and (6.25), with the prescribed input parameters in Table 6.23. It can be concluded on the basis of these comparisons that B-polynomials based EA scheme is more efficient than log sigmoid based EA scheme for solving such problems.

Table 6.23 Parameter values and settings of GA

Parameter Name	Parameter Setting/Value
Population size	[100 100]
Max. No. of generations	1000
Selection function	Stochastic uniform
Mutation function	Adaptive feasible
Crossover function	Heuristic
Crossover fraction	0.8
Function tolerance	1e-15
Bounds	-10, +10

Table 6.24 Comparison of convergence speed and accuracy for example 1

	B-polynomials based EA scheme			**Log sigmoid based EA scheme**		
Parameter	$n = 5$	$n = 6$	$n = 7$	$m = 5$	$m = 6$	$m = 7$
No. of generations	240	281	339	1000	401	1000
Computational time (sec)	11	15	17	148	88	152
Average absolute error	3.44E-05	3.58E-06	9.14E-07	3.89E-06	1.88E-06	2.11E-06

Table 6.25 Comparison of convergence speed and accuracy for example 2

	B-polynomials based EA scheme			**Log sigmoid based EA scheme**		
Parameter	$n = 5$	$n = 6$	$n = 7$	$m = 5$	$m = 6$	$m = 7$
No. of generations	228	309	539	1000	1000	1000
Computational time (sec)	11	16	18	179	180	180
Average absolute error	4.39E-04	1.11E-04	5.11E-05	1.99E-05	2.11E-05	1.06E-05

Table 6.26 Comparison of convergence speed and accuracy for fin problem

	B-polynomials based EA scheme			**Log sigmoid based EA scheme**		
Parameter	$n = 5$	$n = 6$	$n = 7$	$m = 5$	$m = 6$	$m = 7$
No. of generations	267	419	601	1000	1000	1000
Computational time (sec)	27	54	78	95	112	200
Average absolute error	6.13E-06	4.44E-06	3.84E-06	4.51E-05	7.48E-05	2.01E-05

6.7 SUMMARY

This chapter provides the detail of the heuristic computation method hybridizing the Bernstein polynomials basis and evolutionary algorithm for solving NODEs. The chapter gives the introduction of the Bernstein polynomials. The numerical applications of the method are presented for solving Riccati equations and power-law fin type problem. From the simulation results presented in this chapter it has been established that the proposed method is quite competent and viable for solving strongly nonlinear problems that are of practical interest to engineers. Moreover, from the comparative analysis between log sigmoid based EA scheme and B-polynomials based EA it can be concluded that B-polynomials based EA scheme is quite efficient and promising for solving NODEs.

CHAPTER 7

CONCLUSION

This chapter furnishes thesis conclusion and outlines some future research problems that one may attempt.

7.1 SUMMARY OF THE THESIS

This thesis work has mainly investigated the numerical solution to nonlinear problems governed by differential equations, using the heuristic computation based techniques. Four different heuristic computation schemes have been used for solving various nonlinear problems arising in diverse applications of engineering.

- A heuristic scheme comprising of EAs and the linear combinations of log sigmoid basis functions has been successfully applied for the numerical solution of several nonlinear problems of NODEs, including the Bratu problem, Troesch's problem, and Duffing van der pol oscillator. Further, the scheme has been successfully tested on some nonlinear singular boundary value problems, such as, heat conduction model of the human head, oxygen diffusion problem, and many other problems in physiology. It has been found that the scheme gives quite accurate results that are in good agreement with the exact solutions and quite competent with those reported by some traditional methods.
- The polynomial basis along with hybrid EA technique has been applied for numerically solving CNODEs. To validate the scheme it has been tested on

Michaelis-Menten nonlinear biochemical reaction system and HIV infection model of $CD4^{+}T$ cells. The proposed results have been found in a good agreement with RK4. Moreover it has been found that the proposed scheme shows significant supremacy on some well-known traditional methods including VIM, HPM, and DTM in obtaining the solution of biochemical reaction model.

- A simple and straightforward scheme based on the hybridization of Exp-function method with nature inspired computation has been presented for numerically solving NPDEs. The efficacy and viability of the proposed scheme has been demonstrated by numerically solving some important NPDEs including Fisher's equations, generalized Burger's-Fisher (B-F) equation, and generalized Burger's-Huxley (B-H) equation. The comparison of numerical results have revealed that the proposed scheme gives more accurate solution for the Fisher's and B-F equations, as compared some state of the art classical methods including ADM, VIM, MVIM, VHPM, HWM, and OHAM. Furthermore, it has been established that the proposed scheme provides the numerical solution to B-H equation that are quite comparable to ADM, VIM, and HPM.
- The Bernstein polynomials based hybrid EA scheme has been presented for the numerical solution of NODEs. The efficacy of the proposed approach has been demonstrated by numerically solving strongly nonlinear power-law fin-type problem. The numerical results have been found very close to the available exact solutions and quite comparable with some approximate numerical techniques like ADM and RKSM. Moreover, the reliability of the proposed approach has been illustrated by solving power-law fin-type problem for several values of the

parameters governing the equation and nonlinearity as well. Furthermore, the applicability of the proposed scheme has been tested on nonlinear Riccati equations. It has been found that the proposed approach yields highly accurate solution to Riccati equations in comparison to VIM, ADM, MHPM, and OHAM.

- In general the numerical simulations show that the proposed schemes are promising and viable for solving such nonlinear problems tackled in this dissertation. Furthermore, the simulation results show that memetic algorithm schemes GA-IPA, GA-ASA, and GA-PS on the average provide superior results.

7.2 DIRECTIONS OF FUTURE WORK

- One may attempt the proposed schemes for solving other NODEs, CNODEs, and NPDEs such as MHD squeezing flow, Chen system, and Schrödinger equations.
- The exploitation of different basis functions like exponential and hat functions can be investigated.
- Use of other nature inspired techniques like bee colony and ant colony optimization may be looked into as candidate solutions.
- The comparative analysis and computational aspect of different heuristic techniques and optimization algorithms can be an area of research.
- The applicability of the proposed schemes on fractional order nonlinear problems may be investigated.

References

[1]. J. A. Khan, "Applications of evolutionary computing techniques to non-linear systems," PhD Dissertation, International Islamic University, Islamabad, Pakistan, 2011.

[2]. Z. Beheshti, S. Mariyam Hj, and Shamsuddin, "A Review of Population-based Meta-Heuristic Algorithms," Int. J. Advance. Soft Comput. Appl., vol. 5, no. 1, 2013.

[3]. S. Binitha, and S.S. Sathya, "A Survey of Bio inspired Optimization Algorithms," International Journal of Soft Computing and Engineering, vol. 2, issue 2, 2012.

[4]. S. Gupta, S. Bhardwaj, and P. K. Bhatia, "A reminiscent study of nature inspired computation," International Journal of Advances in Engineering and Technology, vol. 1, issue 2, pp.117-125, 2011.

[5]. K. Manda, S. C. Satapathy, and B. Poornasatyanarayana, "Population-based Meta-Heuristic techniques for solving optimization problems: A selective survey," International Journal of Emerging Technology and Advanced Engineering, vol. 2, issue 11, 2012.

[6]. X. Yu, and M. Gen, "Introduction to Evolutionary Algorithms," Springer London Dordrecht Heidelberg New York, 2010.

[7]. A. P. Engelbrecht, "Computational Intelligence: An Introduction," Second Edition, John Wiley & Sons Ltd, 2007.

[8]. J. H. Holand, "Adaptation in neural and artificial systems", Ann Arbor, MI: University of Michigan press, 1975.

[9]. M. Mitchell, "Genetic Algorithms: An Overview," pp.31-39, 1995.

[10]. D. E. Goldberg, "Genetic algorithms in search optimization and machine learning," Addison Wesley, pp. 41, 1993.

[11]. M. Gen, and R. Cheng, "Genetic algorithms and engineering optimization," John Wiley & Sons, 2000.

[12]. P. J. Fleming, and R.C. Purshouse, "Genetic algorithms in control systems engineering," Control Systems, Robotics and Automation, vol. XVII.

[13]. M. B. Kadri, and W.A. Khan, "Application of genetic algorithms in nonlinear heat conduction problems," The Scientific World Journal, Article ID 451274, vol 2014, DOI:10.1155/2014/451274, 2014.

[14]. V. Rankovic, J. Radulovic, N. Grujovic, and D. Divac, "Neural network model predictive control of nonlinear systems using genetic algorithms," Int. J. Comput. Commun., vol.7, no. 3, pp. 540 - 549, 2012.

[15]. R. Jain and G. S. Mani, "Solving antenna array thinning problem using genetic algorithm," Applied Computational Intelligence and Soft Computing, vol. 2012.

[16]. R. H. Torres, M. I. Mesa, L. D. T. Camara, A.J. da Silva Neto, and O.L. Santiago, "Application of genetic algorithms for parameter estimation in liquid chromatography," RIELAC, vol. XXXII 3/2011, pp. 13-20, 2011.

[17]. N. R. Shankar, G. A. Rao, J. M. Latha and V. Sireesha, "Solving a fuzzy nonlinear optimization problem by genetic algorithm," Int. J. Contemp. Math. Sciences, vol. 5, no. 16, pp.. 791 – 803, 2010.

[18]. Z. A. Hammour, O. A. Arqub, S.Momani, and N. Shawagfeh, "Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm," Discrete Dynamics in Nature and Society, vol. 2014, 2014.

[19]. Z. S. Abo-Hammour, M. Yusuf, N. Mirza, S. Mirza, M. Arif, and J. Khurshid, "Numerical solution of second-order, two-point boundary value problems using continuous genetic algorithms," International Journal for Numerical Methods in Engineering, vol. 61, pp. 1219–1242, 2004.

[20]. O. Abu Arqub, Z. Abo-Hammour, S. Momani, and N. Shawagfeh, "Solving singular two-point boundary value problems using continuous genetic algorithm," Abstract and Applied Analysis, 2012.

[21]. O. Abu Arqub, Z. S. Abo-Hammour, and S. Momani, "Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems," Applied Mathematics and Information Sciences, vol. 8, pp. 235-248, 2014.

[22]. M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "Evolutionary computational intelligence in solving the Fractional differential Equations," *Lecture notes in Computer Science*, vol. 5990, part 1, Springer-Verlog, ACIIDS Hue City, Vietnam pp 231-240, 2010

[23]. P. Chaudhury and S. P. Bhattacharyya, "Numerical solutions of the Schrodinger equation directly or perturbatively by a genetic algorithm: test cases," Chemical Physics Letters, vol. 296, pp. 51– 60, 1998.

[24]. O. Guimaraes, J. R. C. Piqueira, and M.L. Netto, "Combining legendre's polynomials and genetic algorithm in the solution of nonlinear initial-value problems," Mathematical Problems in Engineering, vol.2011, 2011.

[25]. P. E. MacNeil, "Genetic algorithms and solutions of an interesting differential equation," Real World Applications Track, 2008.

[26]. K. Sersic, and I. Urbiha, "Parameter identification problem solving using genetic algorithm," Proceedings of the Conference on Applied Mathematics and Computation, pp. 253-261, 1999.

[27]. Z. S. Abo-Hammour, A. D. Samhouri, and Y.Mubarak, "Continuous genetic algorithm as a novel solver for stokes and nonlinear navier stokes problems," Mathematical Problems in Engineering, vol. 2014, 2014.

[28]. R. M. A. Zahoor, "Application of fractional calculus to engineering: A new computational approach," PhD Dissertation, International Islamic University, Islamabad, Pakistan, 2011.

[29]. C. Caetano, J. L. Reis Jr., J. Amorim, M. R. Lemes, and A. Dalpino Jr., "Using neural networks to solve nonlinear differential equations in atomic and molecular physics," International Journal of Quantum Chemistry, vol. 111, no. 12, pp. 2732-2740, 2011.

[30]. J. A. Khan, M. A. Z. Raja, and I. M. Qureshi, "Evolutionary computing approach for the solution of initial value problems in ordinary differential equations," World Academy of Science, Engineering and Technology, 31, 2009.

[31]. J. A. Khan, M. A. Z. Raja, and I. M. Qureshi, "An Application of evolutionary Computational technique to non-linear singular system arising in polytrophic

and isothermal sphere," Global Journal of Researches in Engineering, vol. 12, issue 1, 2012.

[32]. M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "Evolutionary computation technique for solving Riccati differential equation of arbitrary order," World Academy of Science, Engineering and Technology, 58, 2009.

[33]. M. Melanie, "An Introduction to Genetic Algorithms", MIT Press, 1999.

[34]. D. A Coley, "An introduction to genetic algorithms for scientists and engineers," World Scientific Publication, 1999.

[35]. R. Storn, and K.Price, "Differential evolution – A simple and efficient heuristic for global optimization over Continuous Spaces," Journal of Global Optimization, vol. 11, pp. 341-359, 1997.

[36]. F.Neri, and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," Artif. Intell. Rev., vol. 33, pp. 61-106, 2010.

[37]. B. Subudhi, and D. Jena, "Nonlinear system identification using memetic differential evolution trained neural networks," Neurocomputing, vol. 74, pp. 1696-1709, 2011.

[38]. A. Basturk, and E. Gunay, "Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm," Expert Systems with Applications, vol. 36, pp. 2645-2650, 2009.

[39]. B. Liu, L.Wang, Y.H. Jin, D.X. Huang, and F.Tang, "Control and synchronization of chaotic systems by differential evolution algorithm," Chaos, Solitons and Fractals, vol. 34, pp. 412-419, 2007.

[40]. A. Ponsich, and C.A. C. Coello, "Differential Evolution performances for the solution of mixed-integer constrained process engineering problems," Applied Soft Computing, vol. 11, pp. 399- 409, 2011.

[41]. B. Subudhia, and D. Jena, "A differential evolution based neural network approach to nonlinear system Identification," Applied Soft Computing, vol. 11, pp. 861-871, 2011.

[42]. A. Slowik, and M. Bialko, "Training of artificial neural networks using differential evolution algorithm," in Proceedings of IEEE conference, 2008.

[43]. G. D. Magoulas, V. P. Plagianakos, and M. N. Vrahatis, "Neural network-based colonoscopic diagnosis using on-line learning and differential evolution," Applied Soft Computing, vol. 4, pp. 369-379, 2004.

[44]. L. Gang, T. Yiqing, and T. Fu, "A fast evolutionary algorithm for neural network training using differential evolution," Fifth International Conference for Young Computer Scientists, vol. 1, pp. 507-511, 1999.

[45]. J. Ilonen, J. K. Kamarainen, and J. Lampinen, "Differential evolution training algorithm for feed-forward neural networks," Neural Processing Letters, vol. 17, pp. 93-105, 2003.

[46]. J. Kennedy, and R. Eberhart, "Particle swarm optimization, "Proceedings of IEEE international conference on neural networks (ICNN'95), Perth Australia: IEEE Press; vol. 4, pp. 1942-2948, 1995.

[47]. R. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory," Sixth International Symposium on Micro Machine and Human Science ,IEEE, 1995.

[48]. M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "A new stochastic approach for solution of Riccati differential equation of fractional order," Ann Math Artif Intell., vol. 60, no. 3, pp. 229-250, 2010.

[49]. M. Biglari, A. Ghoddosian, I. Poultangari, E. Aassareh, and M. Nedaei, "Using evolutionary algorithms for solving a differential equation," Global Journal of Science, Engineering and Technology, Issue 14, pp. 25-32, 2013.

[50]. J. A. Khan, M. A. Z. Raja, and I. M. Qureshi, "Swarm intelligence for the problems of nonlinear differential equations and its application to well-known Wessinger's equation," EJSR vol. 34, no 4, pp. 514-525, 2009.

[51]. M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "Solution of fractional order system of Bagley-Torvik equation using evolutionary computational intelligence," Mathematical problems in engineering, pp.1-18, 2011.

[52]. M. Ghalambaz, and E. Edalatifar, "Analytic Approximate solution for nonlinear damped pendulum by using hybrid artificial neural network and particle swarm optimization algorithm," International Journal of Advanced Engineering Applications, vol.7, pp. 82-89, 2014.

[53]. M. A. Behrang, M. Ghalambaz, E. Assareh, and A.R. Noghrehabadi, "A New Solution for Natural Convection of Darcian Fluid about a Vertical Full Cone Embedded in Porous Media Prescribed Wall Temperature by using a Hybrid Neural Network-Particle Swarm Optimization Method," World Academy of Science, Engineering and Technology, 49, 2011.

[54]. M. A. Z. Raja, I. M. Qureshi, and J. A. Khan, "Swarm intelligence optimized neural networks for solving fractional differential equations," International Journal of Innovative Computing, Information and Control, vol. 7, no. 11, pp. 6301-6318, 2011.

[55]. J. A. Khan, M. A. Z. Raja, and I. M. Qureshi, "Swarm intelligence for the solution of problems in differential equations," Second International Conference on Environmental and Computer Science, pp. 141-147, 2009.

[56]. Y. Liu, X. Ling, Z. Shi, M. Lv, J. Fang, and L. Zhang, "A survey on particle swarm optimization algorithms for multimodal function optimization," Journal of Software, vol. 6, no. 12, pp. 2449-2455, 2011.

[57]. K. Kameyama, "Particle swarm optimization-A survey," The Institute of Electronics, Information and Communication Engineers (IEICE) Trans. Inf. and Syst., vol. E-92-D, no. 7, pp. 1354-1361 2009.

[58]. Y. D. Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, and R.G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," IEEE Transactions on Evolutionary Computation, vol. 12, no. 2, pp. 171-195, 2008.

[59]. S. Ebbesen, P. Kiwitz, and L. Guzzella, "A generic particle swarm optimization Matlab function," American Control Conference, pp. 1519-1524, 2012.

[60]. C. Grosan, and A. Abraham, "Hybrid evolutionary algorithms: methodologies, architectures, and reviews," Studies in Computational Intelligence (SCI), vol. 75, pp.1-17, 2007.

[61]. P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms," Technical Report Caltech Concurrent

Computation Program, rep. 826. Pasadena (CA, USA): California Institute of Technology; 1989.

[62]. D.K. Dimitroulas, and P.S. Georgilakis, "A new memetic algorithm approach for the price based unit commitment problem," Applied Energy, vol. 88, pp. 4687-4699, 2011.

[63]. E. K. Burke, and A. J. Smith, "Hybrid evolutionary techniques for the maintenance scheduling problem," IEEE Transactions on Power Systems, vol. 1(1), pp. 122–128, 2000.

[64]. M. Fatourechi, A. Bashashati, R. K. Ward, and G. Birch, "A hybrid genetic algorithm approach for improving the performance of the LF-ASD brain computer interface, In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP05), Philadelphia, pp. 345-348, 2005.

[65]. P. P. Menon, D. G Bates, and I. Postlethwaite, "Hybrid evolutionary optimization methods for the clearance of nonlinear flight control laws," In Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, Seville, Spain, pp. 4053-4058, 2005.

[66]. T. Terlaky, "Interior point methods of mathematical programming," Springer science and business edition, 1996.

[67]. N. Karmarkar, "A new polynomial-time algorithm for linear programming," Proceedings of the sixteenth annual ACM symposium on theory of computing, pp. 302,DOI:10.1145/800057.808695,1984.

[68]. R. H. Byrd, M. E. Hribar, and J. Nocedal, "Interior point algorithms for large scale nonlinear programming," SIAM Journal on Optimization, vol. 9,no. 4, pp. 877-900, 1999.

[69]. R. A. Walt, J. L. Morales, J. Nocedal, and D. Orban, "An interior point algorithm for nonlinear optimization that combines line search and trust region steps," mathematical programming, vol. 107, no.3, pp. 391-408, 2006.

[70]. G. Lesaja,"Introducing interior-point methods for introductory operations research courses and/or linear programming courses," The Open Operations Research Journal, vol. 3, pp. 1-12, 2009.

[71]. C.M. Maes, "A regularized active-set method for sparse convex quadratic programming," PhD Dissertation, Standford University, 2010.

[72]. E. Wong, "Active-Set Methods for Quadratic Programming," PhD Dissertation, University of California, San Diego, 2011.

[73]. R.Hooke, and R.Jeevs, "Direct search solution of numerical and statistical problems," Journal of Association for Computing Machinery, vol. 8, pp.212-229, 1961.

[74]. R. M. Lewis, V. Torczon, and M. W. Trosset, "Direct search methods: then and now," Journal of Computational and Applied Mathematics, vol. 124, pp. 191-207, 2000.

[75]. A. K. Al-Othman, and K. M. EL-Nagger, "Application of pattern search method to power system security constrained economic dispatch," International Journal of Engineering and Applied Sciences, vol. 4(1), pp. 25-30, 2008.

[76]. S. A. Malik, I. M. Qureshi, M. Zubair, and M.Amir, "Hybrid heuristic computational app.roach to the Bratu problem," Research Journal of Recent Sciences (RJRS), vol. 2(10), pp. 1-8, 2013.

[77]. S. A. Malik, I. M. Qureshi, M. Zubair, and M.Amir, "Numerical solution to Troesch's problem using hybrid heuristic computing," J. Basic Appl. Sci. Res. vol. 3(7), pp. 10-16, 2013.

[78]. S. A. Malik, I. M. Qureshi, M. Zubair, and I. Haq,"Solution to force-free and forced Duffing van der pol oscillator using memetic computing," J. Basic App.l. Sci. Res., vol. 2(11), pp. 11136-11148, 2012.

[79]. S. A. Malik, I. M. Qureshi, M. Amir, and I. Haq, "Nature inspired computational technique for the numerical solution of nonlinear singular boundary value problems arising in physiology," The Scientific World Journal, vol.2014, Article ID 837021, 2014.

[80]. S. A. Malik, I. M. Qureshi, M. Zubair, and I. Haq, "Memetic heuristic computation for solving nonlinear singular boundary value problems arising in physiology," Res. J. Recent Sci., vol. 2(9), pp.1-8, 2013.

[81]. Y.G. Wang, H.F.Song, and D.Li, "Solving two-point boundary value problems using combined homotopy perturbation method and Green's function method," Applied Mathematics and Computation, vol. 212, pp.366-376, 2009.

[82]. M. Kumar, and N. Singh, "A collection of computational techniques for solving singular boundary-value problems," Advances in Engineering Software, vol. 40, pp. 288-297, 2009.

[83]. M. A. Noor, and S. T. Mohyud-Din, "Solution of singular and nonsingular initial and boundary value problems by modified variational iteration method," Mathematical Problems in Engineering, vol. 2008, 2008.

[84]. I. Ahmad, and M. Bilal, "Numerical solution of Blasius equation through neural networks algorithm," American Journal of Computational Mathematics, vol.4, pp. 223-232, 2014.

[85]. M. A. Z. Raja, and R. Samar, "Numerical treatment for nonlinear MHD Jeffery-Hamel problem using neural networks optimized with interior point algorithm," Neurocomputing, vol. 124, pp.178-193, 2014.

[86]. J. A. Khan, M. A. Z. Raja, and I. M. Qureshi, "Numerical treatment of nonlinear Emden-Fowler equation using stochastic technique," Annals of Mathematics and Artificial Intelligence, vol. 63, issue 2, pp. 185-207, 2011.

[87]. J. A. Khan, M. A. Z. Raja, and I. M. Qureshi, " Novel approach for a van der pol oscillator in the continuous time domain," Chinese Physics Letters, vol. 28, no. 11, 2011.

[88]. E. Deeba, S. A. Khuri, and S. Xie, "An algorithm for solving boundary value problems," J. Comput. Phys., vol. 159, pp. 125-138, 2000.

[89]. S.A. Khuri, "A new approach to Bratu's problem," Appl. Math. Comput., vol. 147, pp.131-136, 2004.

[90]. A. M. Wazwaz, "Adomian decomposition method for a reliable treatment of the Bratu-type equations," Appl. Math. Comput., vol. 166, pp. 652-663, 2005.

[91]. M. Abukhaled, S. A. Khuri, and A. Sayfy, "Spline-based numerical treatments of Bratu-type equations," Palestine J. Math., vol. 1, pp. 63-70, 2012.

[92]. A. R.Vahidi, and M. Hasanzade, "Restarted adomian's decomposition method for

the Bratu-type problem," Appl. Math. Sci., vol. 6, pp. 479-486, 2012.

[93]. J. Rashidinia, and R. Jalilian, "Spline solution of two point boundary value problems," Appl. Comput. Math., vol. 9, pp. 258-266, 2010.

[94]. M. A. Noor, and S. T. Mohyud-Din, "Variational iteration method for solving initial and boundary value problems of Bratu-type," Appl. Appl. Math.,vol. 3, pp.89- 99, 2008.

[95]. I. H. Hassan, "Applying differential transformation method to the one-dimensional planar Bratu problem," Int. J. Contemp. Math. Sci., vol. 2, pp.1493-1504, 2007.

[96]. S. H. Mirmoradi, I. Hosseinpour, S. Ghanbarpour and A. Barari, "Application of an approximate analytical method to nonlinear Troesch's problem," Appl. Math. Sci., 3(32): pp..1579 – 1585, 2009.

[97]. H. N. Hassan, and M. A. El-Tawil, "An efficient analytic approach for solving two-point nonlinear boundary value problems by homotopy analysis method," Math. Meth. Appl. Sci., vol. 34, pp. 977–989, 2011.

[98]. S. Momani, S. Abuasad, and Z.Odibat, "Variational iteration method for solving nonlinear boundary value problems," Appl. Math. Comput., vol. 183, pp.1351-1358, 2006.

[99]. E. Deeba, S. A. Khuri, and S. Xie, "An Algorithm for solving boundary value problems," J. Comput. Phy., vol. 159, pp. 125-138, 2000.

[100]. S. H. Chang, "Numerical solution of Troesch's problem by simple shooting method," Appl. Math. Comput., vol. 216, pp. 3303-3306, 2010.

[101]. S. T.Mohyu-din, "Solution of Troesch's problem using He's polynomials," Revista De La, Union Mathematica Argentina, 52: pp.143–148, 2011.

[102]. X. Feng, L. Mei, and G. He, "An efficient algorithm for solving Troesch's problem," Appl. Math. Comput., vol. 189, pp.500–507, 2007.

[103]. F. Geng, and M. Cui, "A novel method for nonlinear two-point boundary value problems: combination of ADM and RKM," Appl. Math. Comput., vol. 217, pp. 4676–4681, 2011.

[104]. S. A. Khuri, and A. Sayfy, "Troesch's problem: A B-spline collocation approach," Math. Comput. Modeling, vol. 54, pp. 1907-1918, 2011.

[105]. E. S. Weibel, and R. K. M. Landshoff, "The Plasma in magnetic field," Stanford University Press, Stanford, pp. 130, 1958.

[106]. B. A.Troesch,"A simple approach to a sensitive two-point boundary value problem," J. Comput. Phy., vol. 21, pp. 279-290, 1976.

[107]. K. Kyamakya, C. Ngoy, M. Tamasala, and J. Chedjou, "A novel image processing approach combining a coupled nonlinear oscillators-based paradigm with cellular neural networks for dynamic robust contrast enhancement," ISAST Transactions on Computers and Intelligent Systems, vol. 2, no. 1, 2010.

[108]. X. Li, J.C. Ji, C. H. Hansen, and C. Tan, "The response of a Duffing–van der pol oscillator under delayed feedback control," Journal of Sound and Vibration, vol. 291, pp. 644-655, 2006.

[109]. A. N. Njah, "Synchronization and anti-synchronization of double hump Duffing-van der pol oscillators via active control," Journal of Information and Computing Science, vol. 4, no. 4, pp. 243-250, 2009.

[110]. N. H. Sweilam, and M. M. Khader, "A Note on He's parameter-expansion method

of coupled Van der pol–Duffing oscillators," Applications and Applied Mathematics, no.1, pp. 94-100, 2010.

[111]. A. Kimiaeifar, A.R. Saidi, G.H. Bagheri, M. Rahimpour, and D.G. Domairry, "Analytical solution for van der pol–duffing oscillators," Chaos, Solitons and Fractals, vol. 42, pp. 2660-2666, 2009.

[112]. G. A. Cordshooli, and A. R. Vahidi, "Solution of Duffing-van der pol equation using decomposition method," Adv. Studies Theor. Phys., vol. 5, no. 3, pp. 121-129, 2011.

[113]. Y.M. Chen and J.K. Liu, "Uniformly valid solution of limit cycle of the Duffing–van der Pol equation," Mechanics Research Communications, vol. 36, pp. 845–850, 2009.

[114]. H. Sajadi, D.D. Ganji, and Y. V. Shenas, "Application of numerical and semi-analytical approach on van der pol-duffing oscillators," Journal of Advanced Research in Mechanical Engineering, vol.1, Issue 3, pp. 136-141, 2010.

[115]. N. A. Khan, M. Jamil, S. A. Ali, and N. A. Khan, "Solutions of the force-free Duffing-van der pol oscillator equation," International Journal of Differential equations, 2011.

[116]. J. Rashidinia, R. Mohammadi and R. Jalilian," The numerical solution of non-linear singular boundary value problems arising in physiology," Applied Mathematics and Computation, vol. 185, pp. 360–367, 2007.

[117]. J. A. Adam, "A simplified mathematical model of tumor growth," Mathematical Biosciences, vol. 81, no. 2, pp. 229-244, 1986.

[118]. H. P. Greenspan, "Models for the growth of a solid tumor by diffusion," Studies in Applied Mathematics, vol. 4, pp. 317–340, 1972.

[119]. S. H. Lin, "Oxygen diffusion in a spherical cell with nonlinear oxygen uptake kinetics," Journal of Theoretical Biology, vol. 60, pp. 449-457, 1976.

[120]. D. L. S. McElwain, "A re-examination of oxygen diffusion in a spherical cell with Michaelis-Menten oxygen uptake kinetics," Journal of Theoretical Biology, vol. 71, pp. 255-263, 1978.

[121]. N. Anderson and A. M. Arthurs, "Complementary variational principles for diffusion problems with michaelis-menten kinetics," Bulletin of Mathematical Biology, vol. 42, pp. 131-135, 1980.

[122]. P. Hiltmann, and P. Lory, "On oxygen diffusion in a spherical cell with Michaelis–Menten oxygen uptake kinetics," Bulletin of Mathematical Biology, vol. 45, pp. 661-664, 1983.

[123]. R. C. Duggan, and A. M. Goodman, "Point wise bounds for a nonlinear heat conduction model of the human head," Bulletin of Mathematical Biology, vol. 48, pp. 229-236, 1986.

[124]. U. Flesch, "The distribution of heat sources in the human head: A theoretical consideration," Journal of Theoretical Biology, vol. 54, pp. 285-287, 1975.

[125]. J. B. Garner, and R. Shivaji, "Diffusion problems with mixed nonlinear boundary condition," Journal of Mathematical Analysis and Applications, vol. 148, pp. 422-430, 1990.

[126]. R. K. Pandey, and A. K. Singh, "On the convergence of a finite difference method for a class of singular boundary value problems arising in physiology," Journal of Computational and Applied Mathematics, vol. 166, pp. 553–564, 2004.

[127]. S. A. Khuri, and A. Sayfy, "A novel approach for the solution of a class of singular boundary value problems arising in physiology," Mathematical and Computer Modelling, vol. 52, pp. 626-636, 2010.

[128]. H. Caglar, N. Caglar, and M. Ozer, "B-spline solution of non-linear singular boundary value problems arising in physiology," Chaos, Solitons and Fractals, vol. 39, pp. 1232-1237, 2009.

[129]. A. S. V. Ravi Kanth, V. Bhattacharya, "Cubic spline for a class of non-linear singular boundary value problems arising in physiology," Appl. Math. Comput., vol. 174 , pp. 768-774, 2006.

[130]. A. Alipanah, "Nonclassical pseudospectral method for a class of singular boundary value problems arising in physiology," Appl. Math., vol. 2(2), pp. 1-4, 2012.

[131]. S. A. Malik, I. M. Qureshi, M. Amir and I. Haq, "Numerical solution to nonlinear biochemical reaction model using hybrid polynomial basis differential evolution technique," Advanced Studies in Biology, vol. 6, no. 3, pp. 99 – 113, 2014.

[132]. S. A. Malik, I. M. Qureshi, M. Amir, and A. N. Malik, "Nature inspired computational approach to solve the model for HIV Infection of $CD4^{+}T$ Cells," Research Journal of Recent Sciences, vol. 3(6), pp. 1-7, 2014.

[133]. A. K. Sen, "An application of the adomian decomposition method to the transient behavior of a model biochemical reaction," Journal of Mathematical Analysis and Applications, vol. 131, pp. 232-245, 1988.

[134]. I. Hashim, M. S. H. Chowdhury, and S. Mawa, "On multistage homotopy-perturbation method applied to nonlinear biochemical reaction model," Chaos, Solitons and Fractals, vol. 36, pp.823-827, 2008.

[135]. S. Alao, F.S. Akinboro, and F.O. Akinpelu, "Numerical solution of nonlinear biochemical reaction model using variational iteration method," The International Journal of Engineering and Science, vol. 2(10), pp. 94-96, 2013.

[136]. A. Batiha and B. Batiha, "Differential transformation method for a reliable treatment of the nonlinear biochemical reaction model," Advanced Studies in Biology, vol. 3(8), pp. 355-360, 2011.

[137]. M. Khader, "On the numerical solutions to nonlinear biochemical reaction model using Picard-Pade technique," World Journal of Modelling and Simulation, vol. 9(1), pp. 38-46, 2013.

[138]. N. Doğan, "Numerical treatment of the model for HIV Infection of $CD4^{+}T$ cells by using multistep laplace adomian decomposition method," Discrete Dynamics in Nature and Society, vol. 2012, 2012.

[139]. M. Merdan, "Homotopy perturbation method for solving model for HIV Infection of $CD4^{+}T$ cells," İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi Yıl: 6 Sayı:12 Güz 2007/2 s. pp.39-52, 2007.

[140]. M. Merdan, A. Gökdoğan, and A. Yildirim, "On the numerical solution of the model for HIV infection of $CD4^{+}$ T cells," Comput. Math. Appl., vol. 62, pp. 118-123, 2011.

[141]. A.S. Perelson, and P.W. Nelson, "Mathematical analysis of HIV-1 dynamics in vivo," SIAM Review, Vol. 41 (1), pp. 3–44, 1999.

[142]. L.Wang, and M.Y.Li,"Mathematical analysis of the global dynamics of a model for HIV infection of $CD4^{+}T$ cells," Math. Biosci. Vol. 200, pp. 44–57, 2006.

[143]. B. Asquith, and C.R.M. Bangham, "The dynamics of T-cell fratricide: application of a robust app.roach to mathematical modelling in immunology," J. Theoret. Biol., vol. 222, pp. 53-69, 2003.

[144]. M. Nowak, and R.May, "Mathematical biology of HIV infections: antigenic variation and diversity threshold," Math. Biosci., vol. 106, pp .1–21, 1991.

[145]. M.Y.Ongun, "The laplace adomian decomposition method for solving a model for HIV infection of CD4^{+}T cells," Math. Comput. Model., vol. 53, pp. 597–603, 2011.

[146]. M. Merdan, A. Gokdogan, and V.S. Erturk, "An approximate solution of a model for HIV infection of CD4^{+} T cells," Iranian J. Sci. Tech. pp. 9-12, 2011.

[147]. S. Yüzbaş, "A numerical approach to solve the model for HIV infection of CD4^{+}T cells," App.l. Math. Model., vol. 36, pp. 5876-5890, 2012.

[148]. M. Ghoreishi, Ismail, A.I.B.Md. and A.K. Alomari, "Application of the homotopy analysis method for solving a model for HIV infection of CD4^{+} T-cells," Math. Comput. Model., vol. 54, pp. 3007-3015, 2011.

[149]. C. L. Bai and H. Zhao, "Generalized extended tanh-function method and its application," Chaos, Solitons and Fractals, vol. 27, no. 4, pp. 1026-1035, 2006.

[150]. A. M. Wazwaz, "The tanh method: solitons and periodic solutions for the Dodd–Bullough–Mikhailov and the Tzitzeica–Dodd–Bullough equations," Chaos, Solitons and Fractals, vol.24, no. 1, pp. 55-63, 2005

[151]. S. A. Khuri. "A complex tanh-function method app.lied to nonlinear equations of Schroedinger type," Chaos, Solitons and Fractals, vol. 20, no 5, pp. 1037-40, 2004

[151] R. Hirota. "Exact solution of the KdV equation for multiple collisions of solitons," Physical Review Letter, vol. 27, pp. 1192-1194, 1971.

[153]. I. Çelik, "Haar wavelet method for solving generalized Burgers-Huxley equation," Arab J. Math. Sci., vol. 18, pp. 25–37, 2011. DOI: 0.1016/j.ajmsc.2011.08.003.

[154]. M. A. Abdou, and A. A. Soliman, "Variational iteration method for solving Burger's and coupled Burger's equations," J. Comput. App.l. Math, vol. 181, no. 2, pp. 245-251, 2005.

[155]. S. Momani, and S Abuasad, "Application of He's variational iteration method to Helmholtz equation," Chaos, Solitons and Fractals, vol. 27, no. 5, pp. 1119-1123, 2006.

[156]. S. M. El-Sayed, D. Kaya, and S. Zarea, "The decomposition method app.lied to solve high-order linear Volterra–Fredholm integrodifferential equations," Int J Nonlinear Sci Numer Simul, vol.5, no.2, pp.105-112, 2004.

[157]. T. S. El-Danaf, MA. Ramadan MA, and Alaal FEIA,"The use of adomian decomposition method for solving the regularized long-wave equation," Chaos, Solitons and Fractals, vol.26, no.3,pp.747-757, 2005.

[158]. Y. G. Zhu, QS Chang, and SC Wu, "Construction of exact solitary solutions for Boussinesq-like B(m,n) equations with fully nonlinear dispersion by the decomposition method", Chaos, Solitons and Fractals, vol.26, no.3, pp. 897-903, 2005.

[159]. J. H. He, "Application of homotopy perturbation method to nonlinear wave equations," Chaos, Solitons and Fractals, vol.26, no.3, pp. 695-700, 2005.

[160]. J. H. He, "Homotopy perturbation method for bifurcation of nonlinear problems," Int J Nonlinear Sci Numer Simul, vol.6, no.2, pp. 207-208, 2005.

[161]. J. He, and Xu-Hong Wu "Exp-function method for nonlinear wave equations," Chaos, Solitons and Fractals, vol.30, pp. 700-708, 2006.

[162]. Zhen-hui Xu, and Da-quan Xian, "Application of Exp-function method to generalized Burgers-Fisher equation," Acta Math. Appl. Sinica, vol.26, no.4, pp. 669=676, 2010. DOI: 10.1007/s10255-010-0031-0.

[163]. T. Öziş, and C. Köroğlu, "A novel approach for solving the Fisher equation using Exp-function method", Phys. Lett, vol. A-372, pp. 3836–3840, 2008.

[164]. C. Chun, "Application of Exp-Function method to the generalized Burgers-Huxley equation," J. Phys. vol. 96, 2008, DOI 012217 DOI:10.1088/1742-6596/96/1/012217

[165]. A. El-Halim Ebaid, "Generalization of He's Exp-function method and new exact solutions for Burgers equation." Z. Naturforsch, vol. 64a, pp. 604 – 608, 2009.

[166]. L. M.B. Assas, "New exact solutions for the Kawahara equation using Exp-function method," J. Comput. Appl. Math., vol.233, pp. 97-102, 2009.

[167]. M. Hu, Z. Jia, Q. Chen and S. Jia, "Exact solutions for nonlinear wave equations by the Exp-function method," Abstract and Applied Analysis, 2014, DOI: 10.1155/2014/252168

[168]. Li Yao, Lin Wang, and Xin-Wei Zhou, "Application of Exp-Function method to a Huxley Equation with variable coefficient," International Mathematical Forum, vol.4, no.1, pp. 27- 32, 2009.

[169]. Xin-Wei Zhou, "Exp-Function method for solving Huxley equation," Mathematical Problems in Engineering, Article ID 538489, 2008, DOI:10.1155/2008/538489

[170]. K. R. Raslan, "The App.lication of He's Exp-function method for MKdV and Burgers' equations with variable coefficients", International Journal of Nonlinear Science, vol.7, no.2, pp.174-181, 2009.

[171]. M.A. Abdou, "New application of Exp-function method for improved Boussinesq equations," Phys. Lett. A 369, pp. 469-475, 2009.

[172]. A. Bekir, and A. Boz, "Exact solutions for a class of nonlinear partial differential equations Exp-function method," Int. J. Nonlinear Sci. Numer. Simul., vol.8, no.5, pp. 505-512, 2007.

[173]. E. Misirli, and Y. Gurefe, "Exp-function method for solving nonlinear evolution equations," Mathematical and Computational Applications, vol. 16, no. 1, pp. 258-266, 2011.

[174]. B. Samir, Hadid, Shayma, Adil Murad, Amera Ismail Melhum, Rabha W, Ibrahim, and A. Hamid Jalab, Neural network technique to estimate the solutions of Burgers-Huxley and Burgers-Fisher equations," Antarctica J. Math., vol. 9, no.6, pp. 503-511, 2012.

[175]. F. Puffer, R. Tetzlaff, and D. Wolf, "Learning algorithm for cellular neural networks (CNN) solving nonlinear partial differential equations," Conference Proceedings of the International Symposium on Signals, Systems and Electronics, pp.501–504, 1995.

[176]. S. He, K Reif, and R. S. Unbehauen, "Multilayer neural networks for solving a differential equations," Neural Networks, vol.13, pp. 385–396, 2000.

[177]. P. Ramuhalli, L. Udpa, and S.S. Udpa, "Finite-element neural networks for solving differential equations," IEEE Transactions on Neural Networks, vol.16, pp.1381–1392, 2005.

[178]. R. A. Fisher, "The wave of advance of advantageous genes," Ann. Eugenics, vol. 7, pp. 355-369, 1937.

[179]. M. Matinfar, M. Mahdavi and Z. Raeisy, "The implementation of variational homotopy perturbation method for Fisher's equation," International Journal of Nonlienar Science, vol.9, no.2, pp.188-194, 2010.

[180]. R.C. Mittal and R Jiwari, "Numerical study of Fisher's equation by using differential quadrature method." International Journal of systems and information sciences, vol. 5, no.1, pp. 143-160, 2009.

[181]. M. Bastani and D. K. Salkuyeh, "A highly accurate method to solve Fisher's equation," Pranama Journal of Physics, vol. 78, no. 3, pp. 335–346, 2012.

[182]. A. M. Wazwaz and A. Gorguis, "An analytic study of Fisher_s equation by using adomian decomposition method," Applied Mathematics and Computation, vol. 154, pp.609–620, 2004.

[183]. M. Matinfar and M. Ghanbari, "Solving the Fisher's equation by means of variational iteration method," International Journal of Contemporary Mathematical Sciences, vol. 4, no. 7, pp. 343-348. 2009.

[184]. S. T. Mohyud-Din, and M. A. Noor, "Modified variational iteration method for solving Fisher's equations," Journal of Applied Mathematics and Computing, vol. 31, pp. 295–308, 2009.

[185]. M. Matinfar, and M.Mahdavi, 'The application of the variational homotopy perturbation method on the generalized Fisher's equation," International Journal of Mathematical Modelling and Computations, vol. 1, no. 2, pp. 101-107, 2011.

[186]. M. Matinfar, S. R. Bahar, and M. Ghasemi, "Solving the generalized Fisher's equation by differential transform method," Journal of Applied Mathematics and Informatics, vol. 30, no. 3-4, pp. 555-560, 2012.

[187]. A. K. Gupta, and S. Saha Ray, "On the solutions of fractional Burgers-Fisher and generalized Fisher's equations using two reliable methods," International Journal of Mathematics and Mathematical Sciences, Article ID 682910, 2014, DOI: 10.1155/2014/682910.

[188]. D. Olmos, and B. D. Shizgal, "A pseudospectral method of solution of Fisher's equation, Journal of Computational and Applied Mathematics," vol. 193, pp. 219-242, 2006.

[189]. R. C. Mittal, and R. K. Jain, "Numerical solutions of nonlinear Fisher's reaction–diffusion equation with modified cubic B-spline collocation method," Mathematical Sciences, vol.7, no.12, 2013.

[190]. R. C. Mittal, and S. Kumar, "Numerical study of Fisher's equation by wavelet Galerkin method, International Journal of Computer Mathematics," vol. 83, pp. 287-298, 2006. DOI: 10.1080/00207160600717758.

[191]. Y. Qiu, and D. M. Sloan, "Numerical solution of Fisher's equation using a moving mesh method," Journal of Computational Physics. vol. 146, pp. 726-746, 1998.

[192]. R. E. Micken, "A best finite-difference scheme for Fisher's equation," Numerical Methods in Partial Differential Equations, vol. 10, pp. 581-585, 1994.

[209] I. Hashim, M.S.M. Noorani, and B. Batiha, "A note on the Adomian decomposition method for the generalized Huxley equation," Applied Mathematics and Computation, vol.181, pp. 1439-1445, 2007.

[210]. B. Batiha, M.S.M. Noorani, and I. Hashim, "Numerical simulation of the generalized Huxley equation by He's variational iteration method," Applied Mathematics and Computation, vol. 186, pp. 1322-1325, 2007.

[211]. N. Bernstein, "D´emonstration du th´eor`eme de Weierstrass fond´ee sur le calcul des probabilit´es," Communications de la Soci´et´eMath´ematique de Kharkov 2. Series XIII No. 1, pp. 1–2., 1912.

[212]. Ş. Yüzbaşi, "Numerical solutions of fractional Riccati type differential equations by means of the Bernstein polynomials," Applied Mathematics and Computation, vol. 219, no.213, pp. 6328-6343.

[213]. M. Idrees Bhatti, and P. Bracken, "Solutions of differential equations in a Bernstein polynomial basis," Journal of Computational and Applied Mathematics, vol.205, pp. 272-280, 2007.

[214]. Y. Ordokhani, and S. Davaei far, "Approximate Solutions of Differential Equations by Using the Bernstein Polynomials," ISRN Applied Mathematics,vol. 2011, Article ID 787694, pp. 1-15, DOI:10.5402/2011/787694

[215]. D. Baleanu, M. Alipour, and H. Jafari, "The Bernstein Operational Matrices for Solving the Fractional Quadratic Riccati Differential Equations with the Riemann-Liouville Derivative," Abstract and Applied Analysis, vol. 2013, Article ID 461970, DOI:10.1155/2013/461970

[216]. A. A. Dascioglu, and N. Isler, "Bernstein collocation method for solving nonlinear differential equations," Mathematical and Computational Applications, vol. 18, no. 3, pp. 293-300, 2013.

[217]. B. N. Mandal, and S. Bhattacharya, "Numerical solution of some classes of integral equations using Bernstein polynomials," Applied Mathematics and Computation, vol. 190, no. 2, pp. 1707-1716, 2007.

[218]. A. Chakrabarti, and S. C. Martha, "Approximate solutions of Fredholm integral equations of the second kind," Applied Mathematics and Computation, vol. 211, no. 2, pp. 459-466, 2009.

[219]. S. Bhattacharya, and B. N. Mandal, "Use of Bernstein polynomials in numerical solutions of Volterra integral equations," Applied Mathematical Sciences, vol. 2, no. 36, pp. 1773-1787, 2008.

[220]. S. Bhattacharya, and B. N. Mandal, "Numerical solution of a singular integro-differential equation," Applied Mathematics and Computation, vol. 195, no. 1, pp. 346-350, 2008.

[221]. B. M. Pandya1, and D. C. Joshi, "Solution of a volterra's population model in a Bernstein polynomial basis," Applied Mathematical Sciences, vol. 5, no. 69, 3403-3410, 2011.

[222]. D. D. Bhatta, and M. I. Bhatti, "Numerical solution of KdV equation using modified Bernstein polynomials," Applied Mathematics and Computation, vol. 174, pp. 1255-1268, 2006.

[223]. F. Mabood, A. Izani bin Md Ismail, and I. Hashim, "Application of optimal homotopy asymptotic method for the approximate solution of Riccati Equation," Sains Malaysiana, vol.42, no.6, pp. 863-867, 2013.

[224]. S. Abbasbandy, "Homotopy perturbation method for quadratic Riccati differential equation and comparison with Adomian's decomposition method," Applied Mathematics and Computation, vol. 172, pp. 485–490, 2006.

[225]. Z. Odiba, and S. Momani, "Modified homotopy perturbation method: Application to quadratic Riccati differential equation of fractional order," Chaos, Solitons and Fractals, vol.36, pp. 167-174, 2008.

[226]. S. Momani, and N. Shawagfeh, "Decomposition method for solving fractional Riccati differential equations," Applied Mathematics and Computation, vol.182, pp.1083-1092, 2006.

[227]. B. Batiha, M. S. M. Noorani, and I. Hashim, "Application of variational iteration method to a general Riccati equation," International Mathematical Forum, vol.2, no. 56, pp. 2759 – 2770, 2007.

[228]. B Batiha, "A numeric-analytic method for approximating quadratic Riccati differential equation," International Journal of Applied Mathematical Research, vol.1, no.1, pp. 8-16, 2012.

[229] A. Moradi, "Analytical solution for fin with temperature ependent heat transfer coefficient," International Journal of Engineering & Applied Sciences (IJEAS), vol.3, no. 2, pp.1-12, 2011.

[230]. Rafael Cortell, "A numerical analysis to the non-linear fin problem," Journal of Zhejiang University Science A, J Zhejiang Univ Sci A, vol.9, no.5, pp.648-653, 2008.

[231]. M.S.H. Chowdhury, I. Hashim, and O. Abdulaziz, "Comparison of homotopy analysis method and homotopy-perturbation method for purely nonlinear fin-type problems," Communications in Nonlinear Science and Numerical Simulation, vol. 14, pp. 371–378, 2008.

[232]. D. Lesnik, and P.J. Heggs, "A decomposition method for power-law fin-type problems," Int. comm. Heat Mass Transfer, vol. 31, no.5, pp. 673-682, 2004.

[233]. M. T. Atay, and S. B. Cos kun, "Comparative analysis of power-law fin-type problems using variational iteration method and finite element method", Mathematical Problems in Engineering vol. 2008, Article ID 635231, pp. 1-9, DOI:10.1155/2008/635231

[234]. S. Mosayebidorcheh, D. D. Ganji, and M. Farzinpoorc, "Approximate solution of the nonlinear heat transfer equation of a fin with the power-law temperature-dependent thermal conductivity and heat transfer coefficient," Propulsion and Power Research, vol.3, no.1, pp. 41-47, 2014.

[235]. K. Saravanakumar, V. Ananthaswamy, M. Subha, and L. Rajendran, "Analytical solution of nonlinear boundary value problem for fin efficiency of convective straight fins with temperature-dependent thermal conductivity," ISRN Thermodynamics, vol. 2013, Article ID 282481, DOI:10.1155/2013/282481

[236]. M. H. Chang, "A decomposition solution for fins with temperature dependent surface heat flux," International Journal of Heat and Mass Transfer, vol. 48, pp. 1819–1824, 2005.

[237]. M. Torabi, and H.Yaghoobi, "Accurate solution for convective–radiative fsin with variable thermal conductivity and nonlinear boundary condition by DTM," Arab J. Sci. Eng, vol.38, pp. 3575-3585, 2013, DOI 10.1007/s13369-013-0636-8.

[238]. S. H. Chowdhury, "A comparison between modified homotopy perturbation method and adomian decomposition method for solving nonlinear heat transfer equations." Journal of Applied Sciences, vol.11, no.8, pp.1416-1420, 2011.

[239]. G. Domairry, and M. Fazeli, "Homotopy analysis method to determine the fin efficiency of convective straight fins with temperature-dependent thermal conductivity," Communications in Nonlinear Science and Numerical Simulation, vol.14, pp. 489–499, 2009.

[240]. M.S.H Chowdhury, and I. Hashim, "Analytical solutions to heat transfer equations by homotopy-perturbation method revisited," Phys. Lett. A, vol. 9, no. 15, 2007, DOI:10.1016/j.physleta.