Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/120108

Modeling and Analyzing Neural Networks Using Reproducing Kernel Hilbert Space Algorithm

Zainah Momani[1], *Mohammad Al Shridah*[1], *Omar Abu Arqub*[2,*], *Mohammad Al-Momani*[3] *and Shaher Momani*[4,5]

[1] Department of Computer Science, King Abdullah II School of Information Technology, The University of Jordan, Amman 11942, Jordan
[2] Department of Mathematics, Faculty of Science, Al-Balqa Applied University, Salt 19117, Jordan
[3] Department of Management Information Systems, Faculty of Economics and Administrative Sciences, Zarqa University, Zarqa 13132, Jordan
[4] Department of Mathmatics,The University of Jordan, Amman 11942, Jordan
[5] Nonlinear Analysis and Applied Mathematics (NAAM) Research Group, Faculty of Science, King Abdulaziz University, Jeddah, 21589, Kingdom of Saudi Arabia

Received: 26 Sep. 2017, Revised: 2 Nov. 2017, Accepted: 7 Nov. 2017
Published online: 1 Jan. 2018

Abstract: In this paper, we present a new method for solving some certain differential systems in the artificial neural networks field. The analytic and approximate solutions are given with series form in the spaces W[a,b] and H[a,b]. The method used in this thesis has several advantages; first, it is of global nature in terms of the solutions obtained as well as its ability to solve other mathematical, physical, and engineering problems; second, it is accurate, need less effort to achieve the results, and is developed especially for the nonlinear cases; third, in the proposed method, it is possible to pick any point in the interval of integration and as well the approximate solutions will be applicable; fourth, the method does not require discretization of the variables, and it is not effected by computation round off errors and one is not faced with necessity of large computer memory and time. Results presented in this thesisshow potentiality, generality, and superiority of our method as compared with the Range Kutta method.

Keywords: Reproducing kernel algorithm; Artificial neural networks; differential systems

1 Introduction

Artificial neural networks (ANN) are consisting of group from the virtual neurons thats designed by computerized programs which use a various of mathematical fractional equations. These equations put in the hidden layers to process the data that comes from the neurons. The ANN we can call it also simulated neural networks SNN because the ANN simulate the mechanism of the biological neural networks, its consist of nodes may be processing elements or neurons. Every connection between the nodes has a weight and by these weights of connections we can define the produced values of each node by calculate the weights values that of each connection that comes to the specific node. By brief words the ANN is a programmed attempt to simulate the way the human brain work and the appendages of nerve bound the nerve cells to build the neural networks in the human brain. The human brain has a millions of the connected nerve cells, the ANN able to allow human mental activities like:

1. The ability of store the information.
2. The ability to sense, hearing, taste and smell.
3. Learning many things.
4. Distinguish between the things and decision making.
5. The possibility of remembering the shapes, colors, the pictures, and people.

For example, to learn specific type of smart phones we will save in our mind the shape of phone its features, its color, its type, and its size. This is an allowed activity of neural networks and we can remember all of these features by also ANN.

* Corresponding author e-mail: o.abuarqub@bau.edu.jo

1.1 Components of the artificial neurons

Every neuron consists of three elements:

1. Weights: each neuron has a weight expressed as a real number and according to these weights of each neuron we can determine the strength of neurons contribution.
2. Weighted Sum: its a summation of all the income weights to the specific neuron thats come from other neurons. In ANN this factor will help in trying to gather a best set of the income weights values for each neuron to know which the neuron has real and useful values that comes from specific action issued by the neurons.
3. Activation Factor: when the neuron gives some action for another one. The issued neuron for this action will send it by signal to the activation factor which would express this signal as an exponential relation to process this signal to get the final result.

The human brain able to process the inputsor givens, so that every part of the brain has a specific function and implement some task. Its called plasticity to simulate the computer thats consists of multiprocessor and each of the will execute task which called parallel computing.

1.2 The weights and the weighted matrix in ANN

As we said the technical neural network consist of many connected neurons or processing units that associate with each other to exchange the information by connections. Each of these connections has a weight, and its called connection weight. For example, we have two connected neurons the connection weight of them will be expressed as W_{ij}. We can represent the weight in weight matrix or in weight vector;the number of rows in the matrix weight determines the cell that carried out the connection to the target cells or to the desired cells to receive information that is sent. In other words, from the row number we can identify from any neuron the connection begins between two neurons, and by the column number determines receptor neurons to the information or signals. In case of the column number was equal zero this means no connection exist, this matrix also called Hin-ton diagrams.

1.3 Mechanism of Artificial Neural Networks

In this section,summed up the work of the mechanism of ANN in data processing across three stages and each stage outputs are input to the next stage, respectively, and we will show this on example form. Stages of information processing are as follows:

1. Propagation function stage
2. Activation function stage
3. Output function stage

1.4 Network Topologies

1. After we became familiar with the nerve cell combination, here we will give an overview about the famous topologies of the neural networks.to construct neural network that consists of the previous components we will firstly decide what will be the based topology of the network that we will describe it with its map on diagram to enable the reader understanding the features and its approach on the neural networks.
2. In the Hinton diagram the dotted weights expressed as a gray field, the solid weights are expressed as a dark gray field. the input and output arrows added to identify the sent neuron by put its name as a line and the destination neuron will be as a column number or column name.

1.5 learning procedures by input training patterns

The purpose of the neural networks to predict to make the input patterns as a common form by training procedure which called generalization. Here in this section we will propose some of the learning procedures and give some of the basic principles to learning procedures approaches [1]. Firstly, we will preview the operations that the neural networks can do them such as:

1. The neural networks can develop a new connections or create them between the neurons and as we said its called weighted connection. Later on we will explain how we can set the weight value to reduce the difference value between the desired outputs and the current outputs to give a good result.
2. The neural network can delete an exist connection between the neurons by put the connected weight value in the weighted matrix between the connected neurons as we learned zero value to cancel the connection.
3. The neural network can change the connected weight value by specific rules they will be explained later on.
4. The neural network able to change the threshold value of the neurons.
5. The neurons in the neural network able to implement a main task (propagation function, activation function and output function).
6. The neural network can add or delete neurons.

The neural network form characterized by their ability to predict the solutions that fit the problem in accurate form. For example, in the medicine domain by input the symptoms of the disease and then by processing the information by the neural networks it will predict the disease or diagnosis of the disease. Another example when give the neural network input patterns we will predict of it to give the desired pattern or desired output.

This could be executed by the training of the network by using an input pattern training to train the network give the desired output by learning algorithms:

1.6 Hopfield Neural Networks

This type of the wide domain of artificial neural networks is same idea or the Hopfield neural network come from idea that the particles which situated in magnetic domain and each of these particles is connect via the magnetic force with all the particles in the domain in completely linked or in other words in fully connected form. So we will apply the previous topology of the neural network which means completely linked topology. We can say about the particle is neuron and these particles when they connect as a fully connected with each other to try to become their activation states are suitable and we can know the minimum activation of the neurons or particles by themselves. When the Hopfield network neurons will be in rotating state they will encourage each other to spin this idea the rotation of the particles is to process the information which means they will be in the activation case. For example, if we have a two particles are in rotating state to process the information this called in the Hopfield neural networks is binary activation [2].

1.7 Problem Statement

Let us consider the following theorem that summarize our work:

Theorem 1.1.[3,4,5,6] *If there exist a number $L > 0$ such that $\parallel h(s_1) - h(s_2) \parallel \geq L \parallel s_1 - s_2 \parallel$ for all $s_1, s_2 \in R^m$ and the number $\mid \varepsilon \mid$ is sufficiently small, then there exists a neighborhood N of the orbitally stable limit cycle of the equation $u^{'} = -Du + Wg(u)$ such that the solutions of the equation $u^{'} = -Du + Wg(u) + \varepsilon h(x(t))$ which start inside N behave chaotically around the limit cycle, that is the solutions are sensitive and there are infinitely many unstable periodic solutions.*

To illustrate the result of the above theorem, let us consider the following general HNN:

$$x_1^{'} = f_1(t, x_1, x_2, ..., x_n), T \in t,$$

$$x_2^{'} = f_2(t, x_1, x_2, ..., x_n), T \in t,$$

$$.$$

$$.$$

$$.$$

$$x_n^{'} = f_n(t, x_1, x_2, ..., x_n), T \in t,$$

subject to the following initial conditions:

$$x_1(0) = \alpha_1,$$

$$x_2(0) = \alpha_2,$$

$$.$$

$$.$$

$$.$$

$$x_n(0) = \alpha_n$$

over the long interval T=[a,b]. In fact, the above system consists of several cases on the Hopfield neural network.

2 Reproducing Kernel Hilbert Spaces Method

The theory of reproducing kernel was used for the first time at the beginning of the 20th century as a novel solver for the boundary value problems of harmonic and biharmonic functions types. This theory, which is representative in the reproducing kernel Hilbert spaces (RKHS) method, has been successfully applied to various important application in numerical analysis, computational mathematics, image processing, machine learning, probability and statistics, and finance [7,8,9, 10]. The RKHS method is a useful framework for constructing numerical solutions of great interest to applied sciences. In the recent years, based on this theory, extensive work has been proposed and discussed for the numerical solutions of several integral and differential operators side by side with their theories. The reader is kindly requested to go through [11,12,13,14,15,16,17, 18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34, 35] in order to know more details about the RKHS method, including its modification and scientific applications, its characteristics and symmetric kernel functions, and others.

2.1 Multistep Approach

The major aim of this work is to find the approximate solutions over along interval. In this section, we utilize the multistep RKHS procedure. To do so, we consider the nonlinear differential equations of Eqs. (2.1) and (2.2). Indeed, let $T = [a, b]$ be the interval over which we want to find the solution. Assume that the interval T is divided into M subintervals $[\beta^{m-1}, \beta^m], m = 1, 2, , M$ of equal step size $h = \dfrac{b-a}{M}$, by using the nodes $\beta^m = mh$.Firstly, we apply the RKHS method over the interval $[a, \beta^1]$, to obtain the approximate solution. For $m \geqslant 2$ and at each subinterval $[\beta^{m-1}, \beta^m]$, we will use the initial conditions obtained over $[a, \beta^1]$, then apply the RKHS technique directly. The process is repeated and generates a sequence of approximate solutions over $[a, \beta^1], [\beta^1, \beta^2]$, and $[\beta^{M-1}, \beta^M]$.

2.2 Computational RKHS Algorithm

Software packages have great capabilities for solving mathematical, physical, and engineering problems. The aim of the next algorithm is to implement a procedure to solve system od differmtial equations in numeric form in terms of their grid nodes based on the use of RKHS method.

Algorithm 1 To approximate the solution $x^{\eta}(t)$ of $x_r(t)$, we do the following steps:

Input: The endpoints of $[a,b]$, the unit truth interval [0,1], the integers n and m, the kernel functions $G_t(s) and H_t(s)$, the differential operator L_r, the initial condition α_r, and the function f_r.

Output: Approximate solution $x^{\eta}(t)$ of $x_r(t)$.

Step i. Fixed t in $[a,b]$ and set $s \in [a,b]$;

If $s \leqslant t$, set

$G_t(s) = \frac{1}{6}(s-a)(2a^2 - s^2 + 6t + 3st - a(6+s+3t))$;

else set

$G_t(s) = \frac{1}{6}(t-a)(2a^2 - t^2 + 3s(2+t) - a(6+3s+t))$;

For $i = 1,2,...,n, h = 1,2,...,m$, and $j = 1,2,,n$, do the following:

Set $t_i = \frac{i-1}{n-1}$; Set $\psi_{ij}(t) = L_s[G_t(s)]_{s=t_i}$;

Output: the orthogonal function system $\psi_{ij}(t)$.

Step ii. For $l = 2,3...,n$ and $k = 1,2...,l$, do the following:

If $k \neq l$ then set

$$\beta_{lk}^{ij} = \frac{-\sum_{p=k}^{l-1} \langle \psi_{lk}(t), \overline{\psi}_{ij}(t) \rangle_W \beta_{pk}^{ij}}{\sqrt{\| \psi_{lk} \|_W^2 - \sum_{p=1}^{l-1} \langle \psi_{lk}(t), \overline{\psi}_{ij}(t) \rangle_W^2}};$$

else set

$$\beta_{lk}^{ij} = \frac{1}{\sqrt{\| \psi_{lk} \|_W^2 - \sum_{p=1}^{l-1} \langle \psi_{lk}(t), \overline{\psi}_{ij}(t) \rangle_W^2}};$$

else set

$$\beta_{11}^{ij} = \frac{1}{\|\psi_{11}\|_W};$$

Output: the orthogonalization coefficients β_{lk}^{ij}.

Step iii. For $l = 2,3...,n-1$and $k = 1,2...,l-1$, do the following:

Set $\overline{\psi}_{ij}(t) = \sum_{l=1}^{i} \sum_{k=1}^{j} \beta_{lk}^{ij} \psi_{lk}(t)$;

Output: the orthonormal function system $\overline{\psi}_{ij}(t)$.

Step iv. Set $x_{r_h}^0(t_1) = x_(r_h)(t_1) = 0$;

Set $B_{ij} = \sum_{l=1}^{i} \sum_{k=1}^{j} \beta_{lk}^{ij} f_k(t_l, x^{l-1}(t_l))$;

Set $x^i(t)(t) = \sum_{i=1}^{i} \sum_{j=1}^{2} B_{ij} \overline{\psi}_{ij}(t)$;

Output: the approximate solution $x^{\eta}(t)$ of $x(t)$.

Step v. Stop.

3 Numerical Simulation

In order to solve the given ANN problems numerically on a computer, the equation is approximated by a discrete one. Continuous functions are approximated by finite arrays of values. Algorithms are then sought which approximately solve the mathematical problem efficiently and accurately. To show behavior, properties, efficiency, and applicability of the present RKHS technique, five problems will be solved numerically in this chapter.

In this section, we have five nonlinear numerical examples thats solved by using the RKHS method and the RK method of order 4, in order to show the accuracy and the ability of these methods for solving such systems. Here, we give the accuracy results especially for the RKHS method. After we obtained the current results or in other words the periodical solutions we compare them between the two presented algorithms. In the process of computation, all the symbolic and numerical computations are performed by using MATHEMATICA 9 software package.

Example 3.1

Consider the following nonlinear Hopfield neural network:

$$x_1^{'} = -x_1 + 34tanh(x_1) - 1.6tanh(x_2) + 0.7tanh(x_3),$$

$$x_2^{'} = -x_2 + 2.5tanh(x_1) + 0.95tanh(x_3),$$

$$x_3^{'} = -x_3 - 3.5tanh(x_1) + 0.5tanh(x_2),$$

subject to initial conditions:

$$x_1(0) = -0.109, x_2(0) = -0.832, x_3(0) = 1.721,$$

where the period equal $t \in [0,250]$

This is Hopfield neural network equation system with initial conditions that we need to use it in our algorithm to enable us get the results and the periodical solutions from zero to 250. Here, we selected this long period to enable the RKHS draw the results in utmost precision. Thus, and according to this accuracy result we can judge or decide how to act or behavior of the system. So, we will solve the equations for x_1, x_2 and x_3, then we will solve pairs of these equation by RKHS method. After applying the previous steps, we will repeat the same approach on the same system and initial condition but by using the RK method of order 4. Finally, we will compare the produced numerical solutions.

According to the numerical solutions in Figure 1 that produced by using the RKHS kernel method,we can conclude that these solutions are not chaotic, which means that, the form of the function behaviour was not chaotic. Anyhow, to make it chaotic system we added some values to be chaotic and this is existing in Example 3.3. So, in general the RKHS method gave solutions to

this system and can solve it. In Figure 2, the numerical solutions are not chaotic also, and they take the same behavior of the solutions using the RKHS method. So, from these results we can conclude the solution between the RKHS and RK methods are agreement.
In Figure3, we note in image (a) the result roughly equal and the same case in image (c). There is small difference in image (b) but also its agree with the other images so we can say that the RKHS able to solve the equations system of the Hopfield neural network in very efficiency and accuracy.

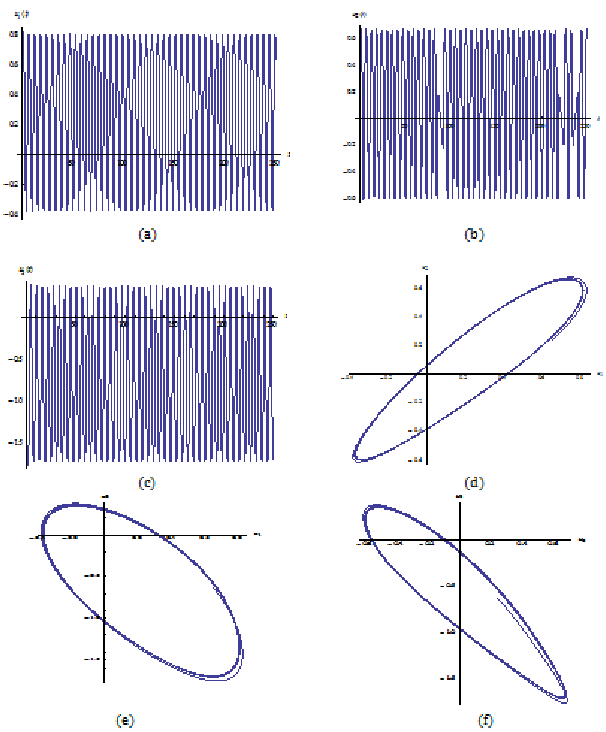

Fig. 1: Some numerical results of Example3.1 using the RKHS kernel method.

Example 3.2

Consider the following non-linear Hopfield Neural Network:

$$x_1^{'} = -x_1 + 2tanh(x_1) - 1.2tan(x_2),$$

$$x_2^{'} = -x_2 + 2.5tanh(x_1) + 1.71tanh(x_2) + 1.15tanh(x_3),$$

$$x_3^{'} = -x_3 - 4.75tanh(x_1) + 1.1tanh(x_3),$$

subject to initial conditions:

$$x_1(0) = -0.109, x_2(0) = -0.832, x_3(0) = 1.721,$$

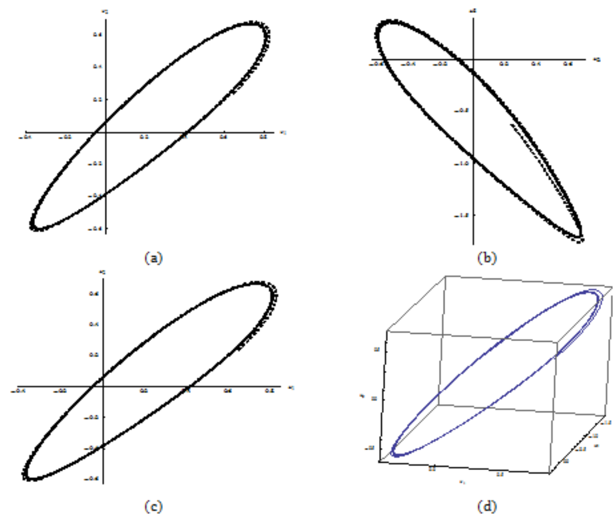

Fig. 2: Some numerical results of Example3.1 using the RK kernel method.

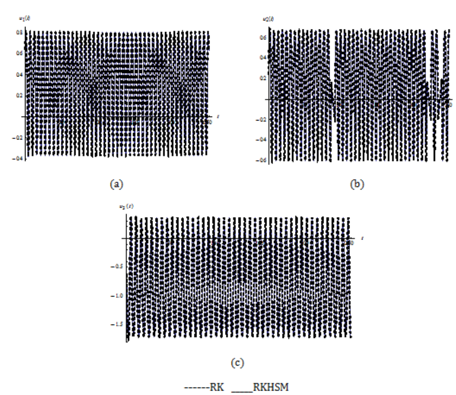

Fig. 3: The solution of the equations system of the Hopfield neural network in Example3.1 using RKHS method

where the period equal $t \in [0, 250]$.
We solved this system by using the RKHS method and the RK method of order 4, to show the difference of the results between these two systems, and to show the performance of them to enable us decide which one the efficient method. After applying the two presented method, as in figure 4 , we solved $x_1(t)$,$x_2(t)$, and $x_3(t)$ equation each one alone by using the RKHS method, thats shown in images (a) , (b) , and (c) to compare these results with the results of the RK method of order 4 that will solve the same equations. We obtained the results thats appear in Figure 4.

In this second part of the example as shown in figure 5, we resolved this system by using the RK method of order 4. As a one time we resolved by it the equationsx_1 with x_2 and the result was in image (a). In the second step, we also resolved the equationsx_1 with x_3 to obtain a chaotic periodical solution for these equations as in image (b), in image (c) we applied the RK method to solve equations x_2 with x_3 and the periodical solutions is also chaotic. Image (d) illustrates the group of solutions to the x_1, x_2 and x_3 which means we could have resolved each one separately by the RK method, but in this image we collected the three periodical results in one image to give chaotic form which means we are solving an chaotic Hopfield neural network equations system. In image (e) we showed you the result of solving equation x_1 by using the RKHS method and the RK method to display how each method solved this equation, and how was the performance of it, so we repeated this step also in equations x_2 and x_3. To compare between them by their results and their performance, and then decide if they can solve the Hopfield neural network equations system and in efficiently form. Virtually the results in image (e), (f), and (g) are approximately symmetric.

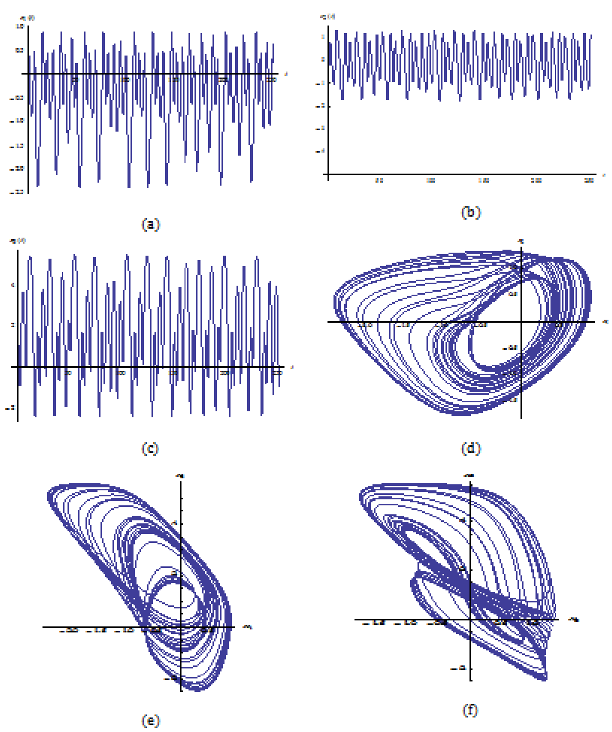

Fig. 4: The RKHS method solved the pairs of equation in Example3.2 like: x_1 with x_2 in image (d), x_1 with x_3 like image (e) to compare the same pair of this equations in the second method, and the same case in image (f) when we solved by the RKHS method the equations pair x_2 with x_3.

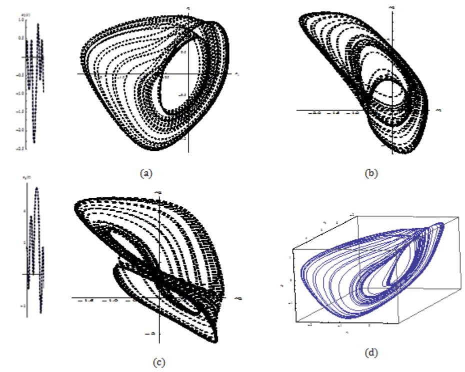

Fig. 5: The results of using RK4 method for the system in Example3.2 and after obtained these results we compared them with the RKHS algorithm results for the same system.

Example 3.3

Making use of the solution of Example 3.2 as external inputs for Example 3.1, we set up the following Hopfield neural network:

$x_1' = -y_1 + 3.4tanh(x_1) - 1.6tanh(y_2) + 0.7tanh(y_3) +$
$0.0136tanh(x_1) - 0.0015tanh(x_2) + 0.0025tanh(x_3),$
$x_2' = -x_2 + 2.5tanh(x_1) + 0.95tanh(y_3) +$
$0.0004tanh(x_1) + 0.0212tanh(x_2) - 0.0005tanh(x_3),$
$x_3' = -x_2 - 3.5tanh(x_1) + 0.5tanh(x_2) +$
$0.0012tanh(x_1) + 0.0023tanh(x_2) + 0.015tanh(x_3),$
subject to initial conditions:

$$x_1 = -0.645, x_2 = 0.243, x_3 = -0.628,$$

where the period equal $t \in [0, 250]$. The following step we will solve the previous numerical Hopfield neural network system and calculate x_1, x_2, x_3 sequentially by using the RKHS method. The goal of these steps and computations to prove the ability of the RKHS method in solving the Hopfield neural network systems, and in efficiency form in addition to compare the numerical solutions of using RKHS method with the most famous mathematical method which means The Rang Kutta method of order 4, to show for the reader the results after the computations for the two methods are agreement. So after applying the RKHS as we see in figure 6 the previous results calculated by the RKHS method with a specific initial conditions and on the specific period thats equal 250. For example in image (a) the equation x_1 have a chaotic periodical solution from [0,250] period, also the same case in x_2 and x_3 equations thats shown in (b) and (c) images. Then we solved by the RKHS method the pairs of equation like x_1 and x_2, x_1 and x_3, x_2 and x_3, and they are shown in (e), (f), (g) images sequentially to watch their results thats produced by the RKHS method then compare them with RK method in the same cases and the same pairs of the equations .

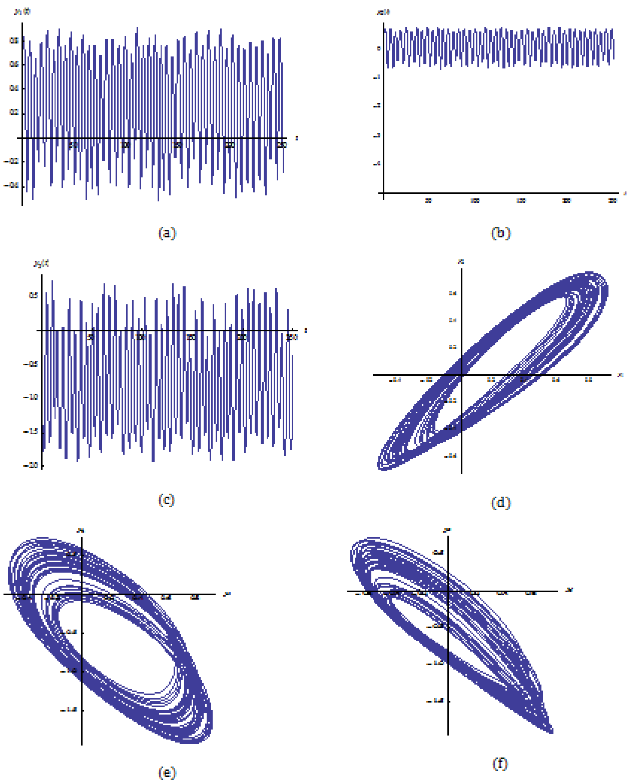

Fig. 6: The result of using RKHS algorithm for the HNN equation system in Example3.3 by two steps; the first one is solving it as one equation and the second is taking the system equation as pairs.

Example 3.4

Consider the following non-linear Hopfield Neural Network:

$$x_1^{'} = -x_1 + 2tanh(x_1) - tan(x_2),$$

$$x_2^{'} = -x_2 + 1.7tanh(x_1) + 1.71tanh(u_2) + 1.1tanh(x_3),$$

$$x_3^{'} = -x_3 - 4.75tanh(x_1) + 1.1tanh(x_2),$$

subject to initial conditions:

$$x_1 = -0.109; x_2 = -0.832; x_3 = 1.721,$$

where the period equal $t \in [0,250]$. As the previous approaches we will apply RKHS method on x_1, x_2 and x_3 to get the desired result and apply them also RK method to compare the current result and judge are they agreement or not. So we will begin with applying RKHS method on this numerical system firstly. See Figures 9 and 10.

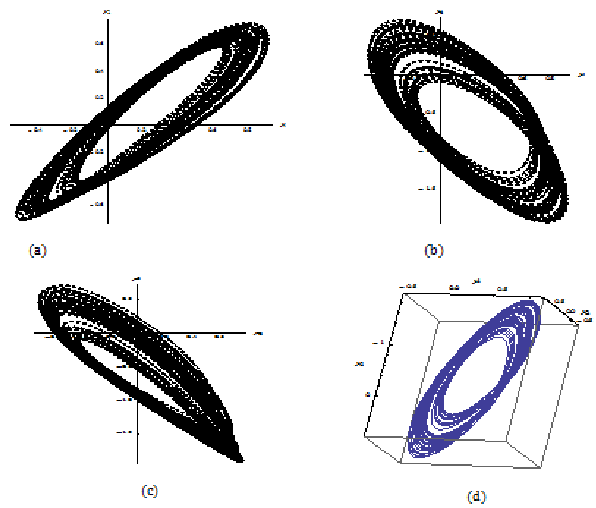

Fig. 7: The equations pairs x_1 and x_2, x_1 and x_3, x_2 and x_3 resolved by the RK method as its clear in (a), (b), (c) and (d) images but in the last image we resolved the HNN equation system in Example3.3 thats consist of x_1, x_2 and x_3 firstly by separate form then we collect the results in this image. by RK method we obtained these periodical solutions.

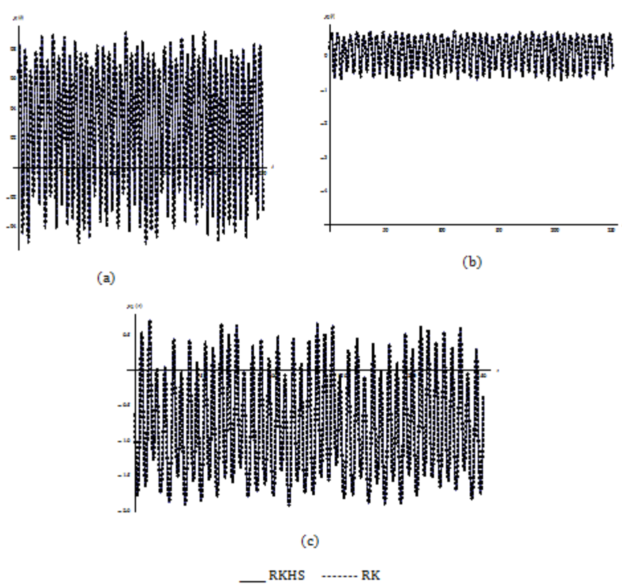

Fig. 8: The images (a), (b) and (c) illustrates each equation which means x_1, x_2 andx_3 resolved by rang kutta 4 method and by RKHS method to show you their results and to prove our used algorithm which means RKHS is able to solve these HNN equations system in Example3.3. We can understand from these periodical solutions almost the results were symmetric.

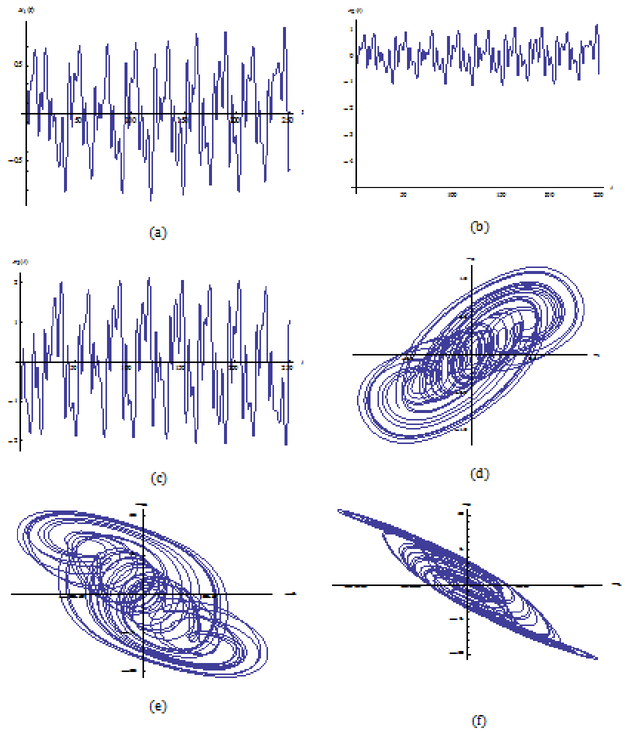

Fig. 9: The system equation in Example3.4 has solved by RKHS method to show and compare these solutions with the other mathematical method we have which means RK method.

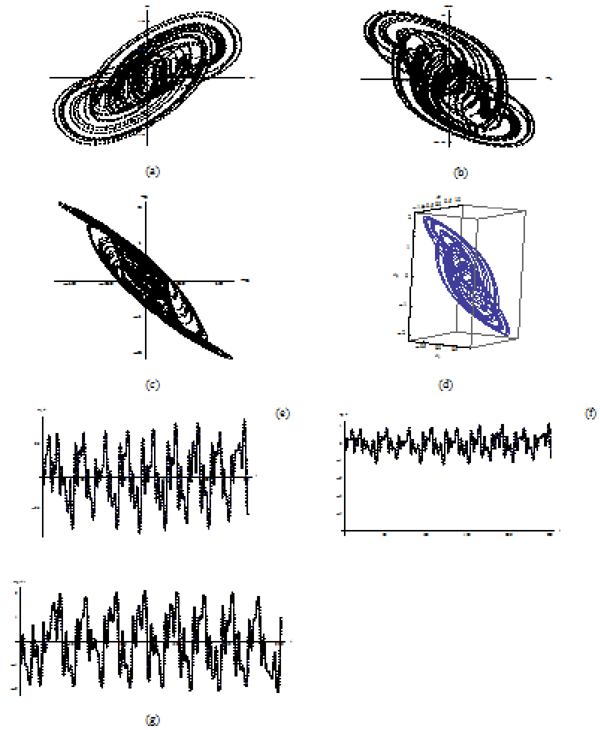

Fig. 10: These previous periodical solutions are chaotic in this Hopfield neural network equations system and they resolved by using the RK method in images a, b, c, d ,e ,f and g. But in the last three images we also collect the results of solving the equations x_1, x_2 and x_3 thats solved previously by the RKHS method to judge is the our mathematical algorithm able to deal with HNN equations system and solve them.

Example 3.5

Consider the following non-linear Hopfield Neural Network [7]:
$x_1^{'} = -x_1 + 3.4tanh(x_1) - 1.6tanh(x_2) + 0.7tanh(x_3) + 0.02tanh(x_1) + 0.035tanh(x_3)$,
$x_2^{'} = -x_2 + 2.5tanh(x_1) + 0.95tanh(x_3) + 0.025tanh(x_2)$,
$x_3^{'} = -x_3 - 3.5tanh(x_1) + 0.5tanh(x_2) + 0.004tanh(x_1) - 0.01tanh(x_2) + 0.05tanh(x_3)$,
subject to initial conditions:

$$x_1(0) = -0.236, x_2(0) = 0.543, x_3(0) = -0.745,$$

where the period equal $t \in [0,250]$.
The following step we will solve the previous numerical system and calculate x_1, x_2 and x_3 by using reproducing kernel Hilbert space method then so we will find the numerical solutions to make sure of the reproducing kernel Hilbert space method able to solve this hop field neural network chaotic system. Let us begin solve this numerical system firstly by RKHS method then we will resolve it by RK method as we did in the previous examples but this time with different initial conditions. See Figures 11,12 and 13.

4 Conclusions

We were able to prove thatthe RKHS method able to solve any numerical equations systems of the Hopfield neural network equations systems in accuracy and efficiency form according to the obtained results that we compared them with the numerical results which solved by using the other famous mathematical method RK method of order 4, and from this point we can decide or judge the numerical results that solved by these two methods are agreement so these agreement results encourage our invention. Finally, we can employ the reproducing kernel Hilbert space method in solving the Hopfield neural networks equations systems as we saw.

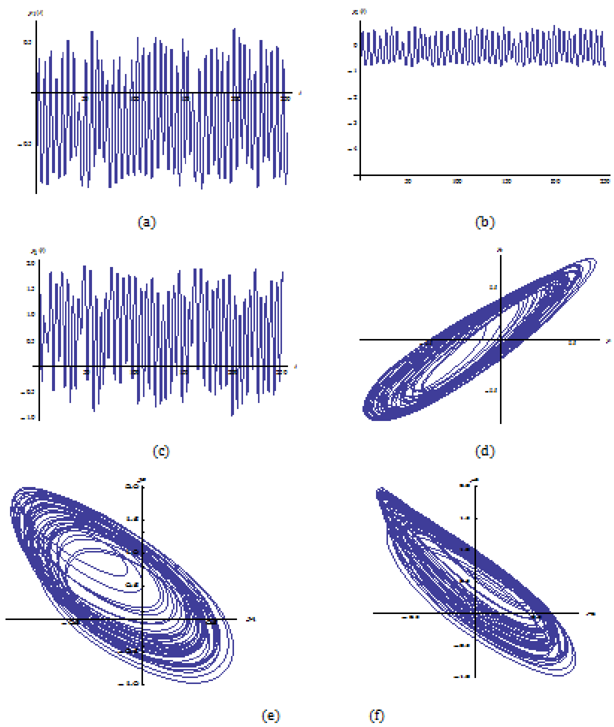

Fig. 11: As we saw, these numerical solutions are produced by applying the RKHS method in HNN equation system that the RKHS could solve and give accuracy solutions for one equation like y_1 or to the pairs of equations system like x_1 with x_2,x_2 with x_3 and x_1 with x_3. We can conclude as a final image the reproducing kernel Hilbert space has a good efficiency in solve HNN equation system.

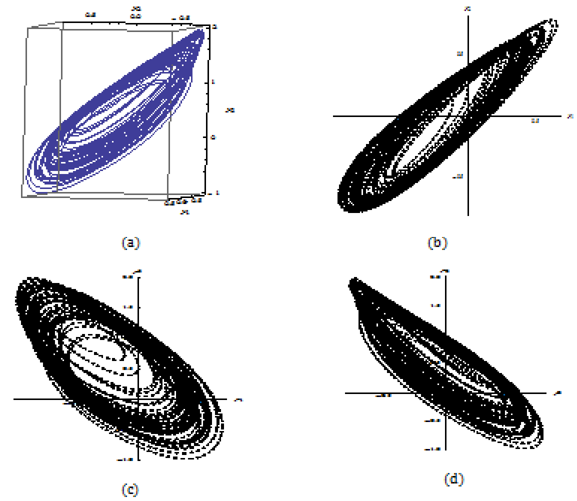

Fig. 12: We solved by RK 4 method the equations y_1,y_2,and y_3 then put the produced solutions in one image which means image a. Then we began solve the pairs of equations system like the previous pairs of the equations system that solved by RKHS method to facilitate the comparison operation between them.

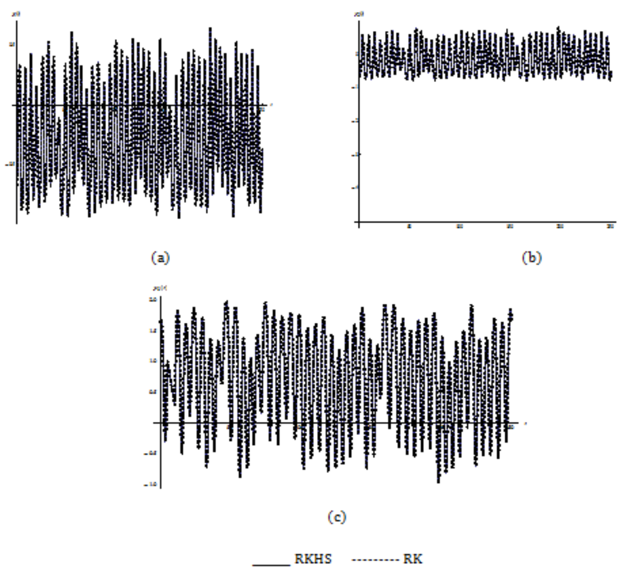

Fig. 13: This figure illustrates a collection operation for the results of the previous numerical system in equation by equation to make sure the two mathematical method will be agreement or not. And according the periodical solutions the results in a or b or c are almost symmetric which means RKHS gives an accuracy result in our Hopfield neural.

References

[1] David Kriesel, A Brief Introduction to Neural Networks, chapter four Fundamentals on learning and training samples, published online, Germany, 2005.
[2] David Kriesel, A Brief Introduction to Neural Networks, chapter eight Hopfield Neural Networks, published online, Germany,2005.
[3] Marat Akhmet,Mehmet Onur Fen, Replication of Chaos in Neural Networks,Economics and Physics, acid-free paper, USA, Turkey, 2016.
[4] M.U. Akhmet, M.O. Fen, Replication of chaos. Commun. Nonlinear Sci. Numer. Simul. 18 (2013) 26262666.
[5] X.-S. Yang, Q. Yuan, Chaos and transient chaos in simple Hopfield neural networks. Neurocomputing 69 (2005) 232241.
[6] J. Li, F. Liu, Z.-H. Guan, T. Li, A new chaotic Hopfield neural network and its synthesis via parameter switchings. Neurocomputing 117 (2013) 3339.
[7] M. Cui, Y. Lin, Nonlinear Numerical Analysis in the Reproducing Kernel Space, Nova Science, New York, NY, USA, 2009.
[8] A. Berlinet, C.T. Agnan, Reproducing Kernel Hilbert Space in Probability and Statistics, Kluwer Academic Publishers, Boston, Mass, USA, 2004.
[9] A. Daniel, Reproducing Kernel Spaces and Applications, Springer, Basel, Switzerland, 2003.

[10] H.L. Weinert, Reproducing Kernel Hilbert Spaces: Applications in Statistical Signal Processing, Hutchinson Ross, 1982.
[11] Y. Lin, M. Cui, L. Yang, Representation of the exact solution for a kind of nonlinear partial differential equations, Applied Mathematics Letters 19 (2006) 808-813.
[12] Y. Zhoua, M. Cui, Y. Lin, Numerical algorithm for parabolic problems with non-classical conditions, Journal of Computational and Applied Mathematics 230 (2009) 770-780.
[13] O. Abu Arqub, Numerical solutions for the Robin time-fractional partial differential equations of heat and fluid flows based on the reproducing kernel algorithm, International Journal of Numerical Methods for Heat and Fluid Flow (2017). doi:10.1108/HFF-07-2016-0278.
[14] O. Abu Arqub, Fitted reproducing kernel Hilbert space method for the solutions of some certain classes of time-fractional partial differential equations subject to initial and Neumann boundary conditions, Computers & Mathematics with Applications 73 (2017) 1243-1261.
[15] O. Abu Arqub, N. Shawagfeh, Application of reproducing kernel algorithm for solving Dirichlet time-fractional diffusion-Gordon types equations in porous media, Journal of Porous Media (2017). In Press.
[16] O. Abu Arqub, H. Rashaideh, The RKHS method for numerical treatment for integrodifferential algebraic systems of temporal two-point BVPs, Neural Computing and Applications (2017) 1-12. doi:10.1007/s00521-017-2845-7.
[17] O. Abu Arqub, The reproducing kernel algorithm for handling differential algebraic systems of ordinary differential equations, Mathematical Methods in the Applied Sciences 39 (2016) 4549-4562.
[18] O. Abu Arqub, M. Al-Smadi, N. Shawagfeh, Solving Fredholm integro-differential equations using reproducing kernel Hilbert space method, Applied Mathematics and Computation 219 (2013) 8938-8948.
[19] O. Abu Arqub, M. Al-Smadi, Numerical algorithm for solving two-point, second-order periodic boundary value problems for mixed integro-differential equations, Applied Mathematics and Computation 243 (2014) 911-922.
[20] S. Momani, O. Abu Arqub, T. Hayat, H. Al-Sulami, A computational method for solving periodic boundary value problems for integro-differential equations of Fredholm-Voltera type, Applied Mathematics and Computation 240 (2014) 229-239.
[21] O. Abu Arqub, M. Al-Smadi, S. Momani, T. Hayat, Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method, Soft Computing 20 (2016) 3283-3302.
[22] O. Abu Arqub, M. Al-Smadi, S. Momani, T. Hayat, Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems, Soft Computing (2016) 1-16. doi:10.1007/s00500-016-2262-3.
[23] O. Abu Arqub, Adaptation of reproducing kernel algorithm for solving fuzzy Fredholm-Volterra integrodifferential equations, Neural Computing & Applications (2015) 1-20. doi:10.1007/s00521-015-2110-x.
[24] O. Abu Arqub, Approximate solutions of DASs with nonclassical boundary conditions using novel reproducing kernel algorithm, Fundamenta Informaticae 146 (2016) 231-254.
[25] O. Abu Arqub, B. Maayah, Solutions of Bagley-Torvik and Painlev equations of fractional order using iterative reproducing kernel algorithm, Neural Computing & Applications (2016) 1-15. doi:10.1007/s00521-016-2484-4.
[26] M. Al-Smadi, O. Abu Arqub, N. Shawagfeh, S. Momani. Numerical investigations for systems of second-order periodic boundary value problems using reproducing kernel method, Applied Mathematics and Computation 291 (2016) 137-148.
[27] M. Ahmad, S. Momani, O. Abu Arqub, M. Al-Smadi, A. Alsaedi, An Efficient Computational Method for Handling Singular Second-Order, Three Points VolterraIntegrodifferential Equations, Journal of Computational and Theoretical Nanoscience 13 (2016) 7807-7815.
[28] S. Bushnaq, B. Maayah, S. Momani, O. Abu Arqub, M. Al-Smadi, A. Alsaedi, Analytical Simulation of Singular Second-Order, Three Points Boundary Value Problems for Fredholm Operator Using Computational Kernel Algorithm, Journal of Computational and Theoretical Nanoscience 13 (2016) 7816-7824.
[29] S. Momani, O. Abu Arqub, A. Freihat, M. Al-Smadi, Analytical approximations for Fokker-Planck equations of fractional order in multistep schemes, Applied and Computational Mathematics 15 (2016) 319-330.
[30] M. Al-Smadi, A. Freihat, O. Abu Arqub, N. Shawagfeh, A Novel Multistep Generalized Differential Transform Method for Solving Fractional-order L Chaotic and Hyperchaotic Systems, Journal of Computational Analysis & Applications 19 (2015) 713-724.
[31] F.Z. Geng, S.P. Qian, Reproducing kernel method for singularly perturbed turning point problems having twin boundary layers, Applied Mathematics Letters 26 (2013) 998-1004.
[32] W. Jiang, Z. Chen, A collocation method based on reproducing kernel for a modified anomalous subdiffusion equation, Numerical Methods for Partial Differential Equations 30 (2014) 289-300.
[33] F.Z. Geng, S.P. Qian, S. Li, A numerical method for singularly perturbed turning point problems with an interior layer, Journal of Computational and Applied Mathematics 255 (2014) 97-105.
[34] F.Z. Geng, M. Cui, A reproducing kernel method for solving nonlocal fractional boundary value problems, Applied Mathematics Letters 25 (2012) 818-823.
[35] W. Jiang, Z. Chen, Solving a system of linear Volterra integral equations using the new reproducing kernel method, Applied Mathematics and Computation 219 (2013) 10225-10230.

Zainah Momani graduated from University of Jordan in 2014 with a B.Sc. in Computer Science (CS). In 2016, she received her Master degree in CS from University of Jordan. Her research interests focus on Networks, Databases and Information systems and Sciences.

Mohammad Al Shridah is a Professor of Software Engineering in the Department of Computer Science at the University of Jordan, Jordan. He received his B.Sc. degree in Computer Science in 1988 from Mutah University, Jordan. He also earned from the University of Jordan, Master degree in Computer Science in 2000. Furthermore, he obtained his Ph.D. degree in Computer Science from University of Hull, UK, in 2007. During his graduate studies, he obtained a fellowship from the University of Jordan. Now he currently is working as Dean of Graduate school at the University of Jordan. He was granted several academic projects. He has published more than thirty papers in his research areas. He participated in many workshops, seminars and conferences in the field of software engineering, and computing. His research interests include Software Testing, cost models, software engineering environments, and knowledge-based software engineering, computation Intelligence.

Omar Abu Arqub received his Ph.D. from the university of Jordan (Jordan) in 2008. He then began work at Al Balqa applied university in 2008 as an assistant professor of applied mathematics and promoted to associate professor in 2013. His research interests focus on numerical analysis, optimization techniques, optimal control, fractional calculus theory, and fuzzy calculus theory.

Mohammad Al-Momani graduated from Al-Balqa Applied University in 2006 with a B.S. in Computer Information Systems (CIS) and in M.S. degree in CIS in 2009 from Near East University (Cyprus). In 2014, he received his Ph.D in Management Information Systems (MIS) from Girne American University (Cyprus). Dr. Al-Momani is now an Assistant Professor at Zarqa University (Jordan) and is currently serving as a head of MIS department. His research interests focus on Networks, Databases, Online learning and Information systems and Sciences.

Shaher Momani received his Ph.D from the university of Wales (UK) in 1991. He then began work at Mutah university in 1991 as assistant professor of applied mathematics and promoted to full Professor in 2006. He left Mutah university to the university of Jordan in 2009 until now. His research interests focus on the numerical solution of fractional differential equations in fluid mechanics, non-Newtonian fluid mechanics and numerical analysis. Prof. Momani has written over 250 research papers and warded several national and international prizes. Also, he was classified as one of the top ten scientists in the world in fractional differential equations according to ISI web of knowledge.