*Turkish Journal of Computer and Mathematics Education Vol.12 No.12 (2021), 461-479*

*Research Article*

# IRFBEN method for solving higher order differential equations

**Emen . A . Hussain [1], Mayada . G. Mohammed [2], Haiffa .M. Buite [3], Ahmed. J. Hussain [4]**

Department of Mathematics, collage of Education for Pure Sciences, University of Thi-Qar.
eman_Ali.math@utq.edu.iq , Mayadagassab20@utq.edu.au, Haiffa_Muhsan @utq.edu.iqa,
hmedjabbbar1974_math@utq.edu.iq

**Article History**: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

**Abstract:**
This paper provides a computational method for approximating a radial basis function networks (RBFNS) are used to model a function and its derivatives (scattered data interpolation), The solution of differential equations of the fourth order Kuromoto-Sivashansky Eq and Nikolaevsky equation of the sixth order is presented in this article. It could begin with a fourth-order derivative method. a RBFN is used to approximate the fourth order derivative. Integration yields the derivative of the third degree, then the derivative of the second degree, and finally the derivative of the first degree. Finally, the original function is derived in the same way, i.e. the derivative can be performed on a sixth-order equation using IRBFN.

**key words: Radial basis function , Differential equations, Integrated radial basis functions method**

## 1. INTRODUCTION

In the field of numerical analysis, radial-basis-function networks (RBFNs) are commonly used. Partial differential equations are solved numerically. (PDEs) has received a lot of interest from both science and engineering research groups over the last 20 years. Since they can approximate any normal function [3,4,5], RBFNs (radial base function networks) are useful for a variety of tasks [1–2]. The RBFN combines self-organized and supervised learning, making it faster to prepare than a multilayer perceptron [2]. As a consequence, the approach's generalization is analogous to interpolating the test results using a multidimensional surface [6]. Introduced: Mark is a University of Introduced: Mark is a University of Introduced: Mark is a University of Introduced: Mark Orr, J.L. (1996) However, Radial base function networks (RBFNs) can be defined as a linear network in many ways for achieving strong generalizations. S. Haykin, S. Haykin, S. Haykin, S. Haykin, S (1999) In a high-dimensional space, the geometry of an RBFN is regarded as a curve-fitting (approximation) problem. Nam Mai-Duy and Thanh Tran-Cong are two Vietnamese actors who are better known for roles in the films Nam Mai-Duy and Thanh Tran-Cong (2003) The indirect radial basis function network (IRBFN) approximation is proposed as an alternative to the standard direct approach. The improvement is due to integration, which is a smoothing operation that is more numerically robust. Approximations for a given point were added in 2005 (Mai-Duy and Tran-Cong), resulting in considerable savings over an earlier 2D-IRBF interpolation system. Sarler and Vertnik (2006) suggested an explicit local radial base function collocation scheme for ambiguity problems. In terms of stream function and vorticity, it can be used to simulate 2-D incompressible viscous ows. Ngo-Cong, Mai-Duy, Karunasena, and Tran-Cong suggested a local moving least square-one dimensional interconnected radial base function network system (2012). (LMLS-1D-IRBFN). Using radial basis function networks, this paper provides a statistical approach for approximating a function and its derivatives (RBFNS) (interpolation of dispersed data), This paper presents the solution of Kuromoto-Sivashansky Eq of fourth order and Nikolaevsky Eq of sixth order differential equations.

## 2. PRELIMINARIES

In this section we present the indirect method for higher ranks and some of the required equations and results.

### 2.1 Indirect method to higher ranks :

In the Integral Radial Basis Function Networks(indirect method) is the original formulation of the problem with the decomposition of the function into (RBF s ). (IRBFN). To create a main function expression, the derivative expression is combined with the main function expression. The problem is then solved using the general linear least squares theorem if a sufficient set of discrete data points is given. If n is the highest order of the derivative in question, then a component of the mechanism is as follows:

$$\frac{d^n f(x)}{dx^n} = \sum_{i=1}^{m} w^{(i)} g^{(i)}(x) \tag{1}$$

$$\frac{d^{n-1} f(x)}{dx^{n-1}} = \int \sum_{i=1}^{m} w^{(i)} g^{(i)}(x)dx + t_1 = \Sigma \sum_{i=1}^{m} w^{(i)} \int g^{(i)}(x)dx + t_1 = \sum_{i=1}^{m} w^{(i)} H_{[n-1]}^{(i)} + t_1$$
$$= \sum_{i=1}^{m} w^{(i)}, H_{[n-1]}^{(i)}, \tag{2}$$

$$\frac{d^{n-2} f(x)}{dx^{n-2}} = \sum_{i=1}^{m} w^{(i)} \int H_{|n-2|}^{(i)} dx + t_1 x + t_2 = \sum_{i=1}^{m} w^{(i)} H_{[n-2)}^{(i)} dx + t_1 x + t_2 = \sum_{i=1}^{m} w^{(n)} H_{[n-2]}^{(i)} \tag{3}$$

(2.2.3)

$$\frac{df(x)}{dx} = \sum_{i=1}^{m} w^{(i)} \int H_{[2]}^{(0)} dx + t_1 \frac{x^{n-2}}{(n-2)!} + t_2 \frac{x^{n-3}}{(n-3)!} + \cdots + t_{n-2}x + t_{n-1} = \sum_{i=1}^{m} w^{(1)}, H_{[1]}^{(0)} \tag{4}$$

$$f(x) = \sum_{i=1}^{m} w^{(i)} \int H_{[1]}^{(i)} dx + t_1 \frac{x^{n-1}}{(n-1)!} + t_2 \frac{x^{n-2}}{(n-2)!} + \cdots + t_{n-1}x + L_n = \sum_{i=1}^{m} w^{(n)} H_{|0|}^{(j)} \tag{5}$$

Such that the functions $H_{[\cdot]}^{(i)}$ are new basis functions obtained by integrating the radial basis function g, and their corresponding known basis functions (polynomial) on the right hand sides in Gutierrez RH(1995) are also denoted by notations for convenience $w^i$ and $H_{[\cdot]}^{(i)}$ respectively, and $i > m$ Mai-Duy (2006) [ 7].

**2.1.1 Kuramoto – Sivashansky eq :**

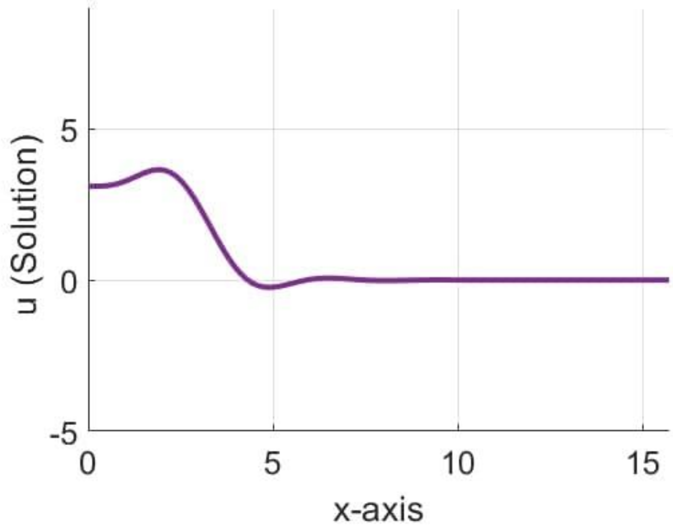
$$\frac{\partial u}{\partial t} = \frac{\partial^2 \mathrm{u}}{\partial x^2} + \left(\frac{\partial u}{\partial x}\right)^2 - \frac{\partial^4 \mathrm{u}}{\partial x^4}$$

A nonlinear fourth order partial differential equation is the Kuramoto-Sivashinsky equation. Yoshiki Kuramoto and Gregory Sivashinsky developed the equation in the late 1970s to model the propagation of instability in the laminar flame front. The disorderly action of the Cora Muto - Sivashinsky equation is well-known. [8]

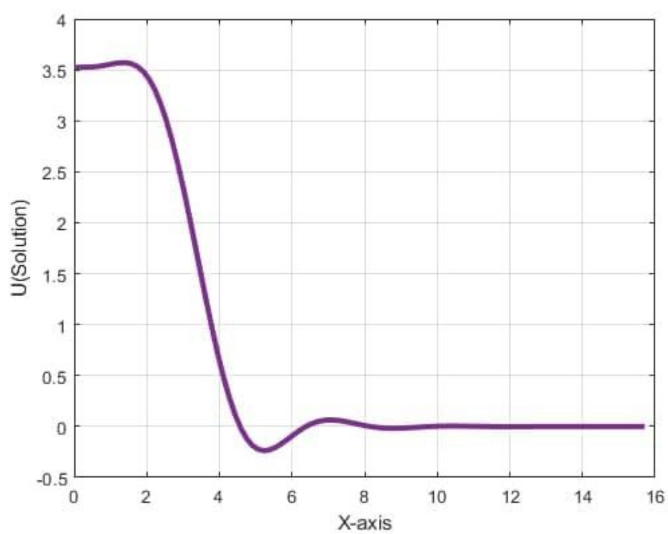
- Solve 4th order partial differential equation.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 \mathrm{u}}{\partial x^2} + \left(\frac{\partial u}{\partial x}\right)^2 - \frac{\partial^4 \mathrm{u}}{\partial x^4}$$

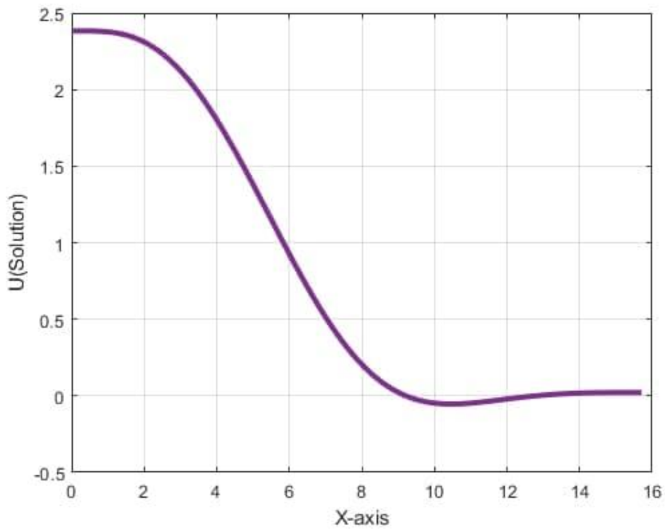
1. We will solve this equation using IRBFN and we devloped matlab code for it.
   We used intial condition which is $u_0 = 5\exp(M)$ with time priod [0,5pi] , $c = [xA: dx: xB]$
   $M = -(-x-2)(x-2)$ and $t = 10$ and we get the following resultis.



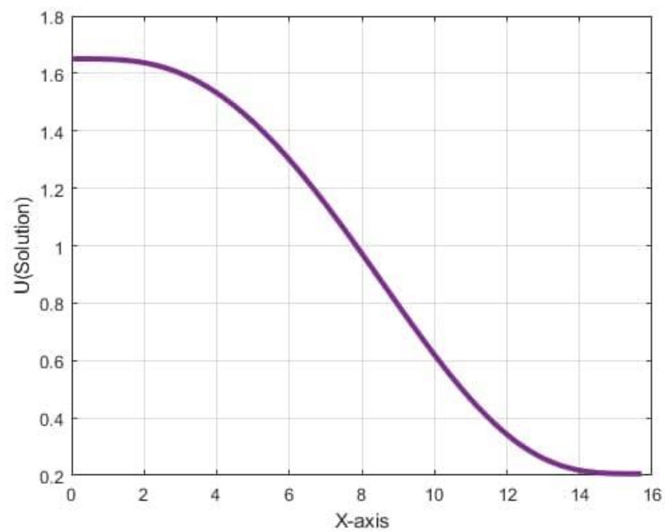

**Figer 1:** The solution at total time =20



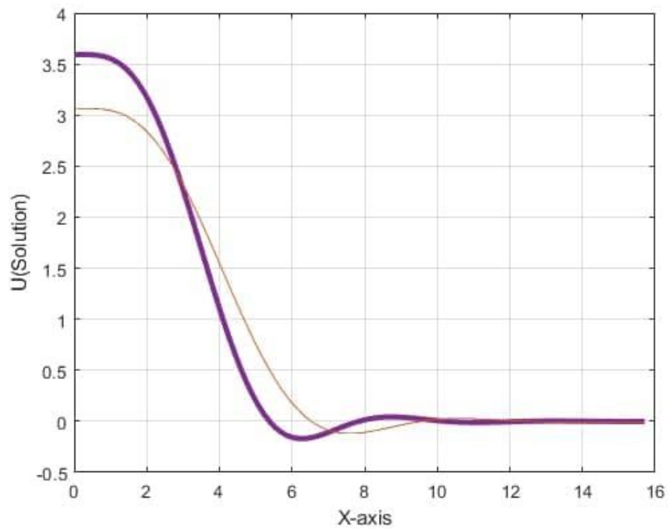

**Figer 2:** The solution at time step 60




**Figer 3:** The solution at time step 4000

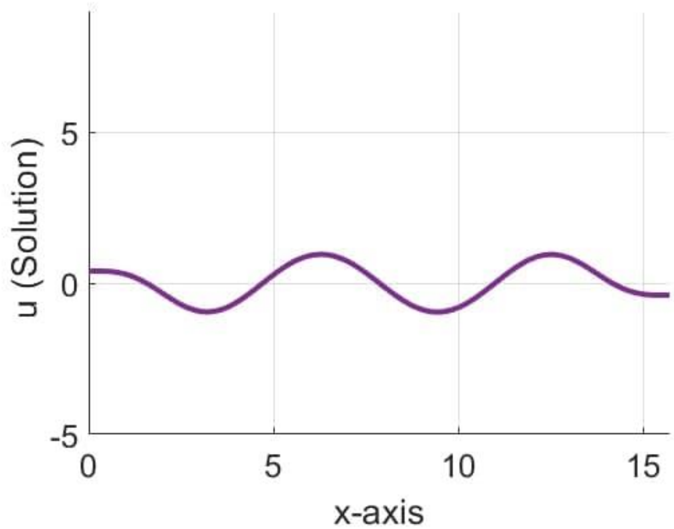



**Figer 4:** The solution at time step 20001

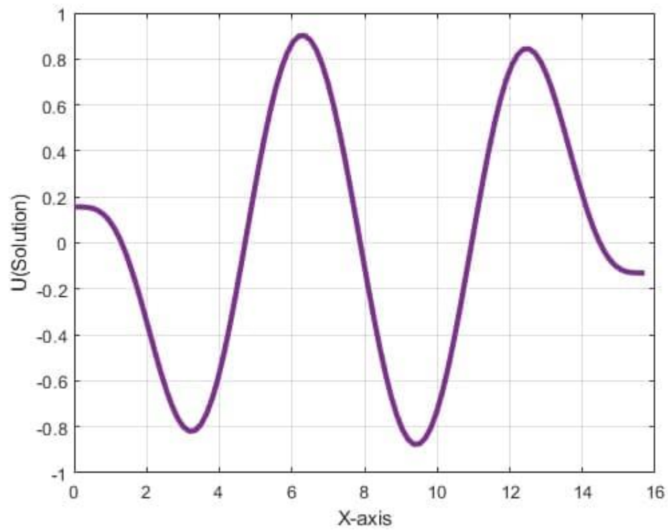



**Figer 4:** The solution at different times (300 , 1050 )

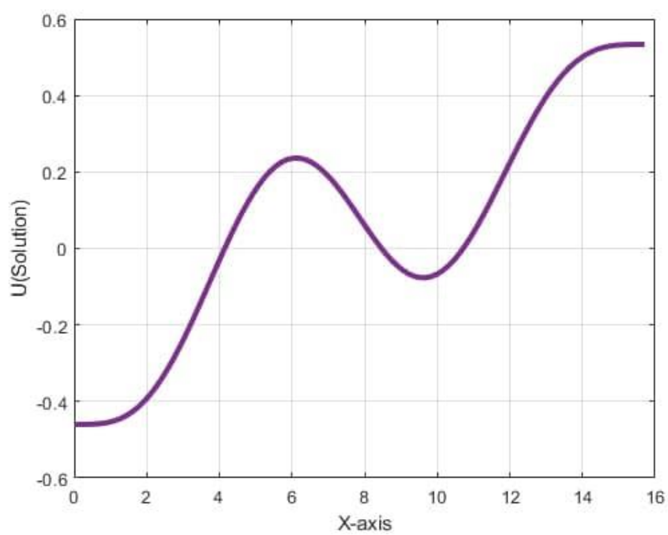
2. We will solve this equation using IRBFN and we devloped matlab code for it.
   We used intial condition which is $u_0 = \cos(x)$ with time priod [0,5pi]
   $c = [xA: dx: xB]$ , $M = -(-x-2)(x-2)$ and $t = 10$ and we get the following resultis.



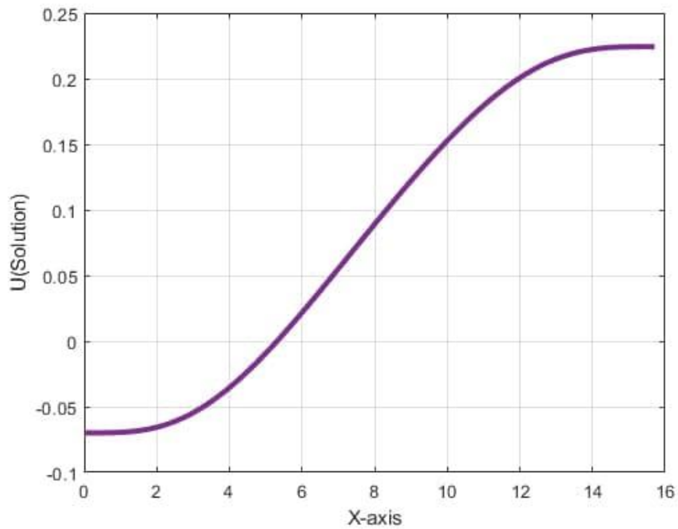

**Figer 5:** The solution at total time =20



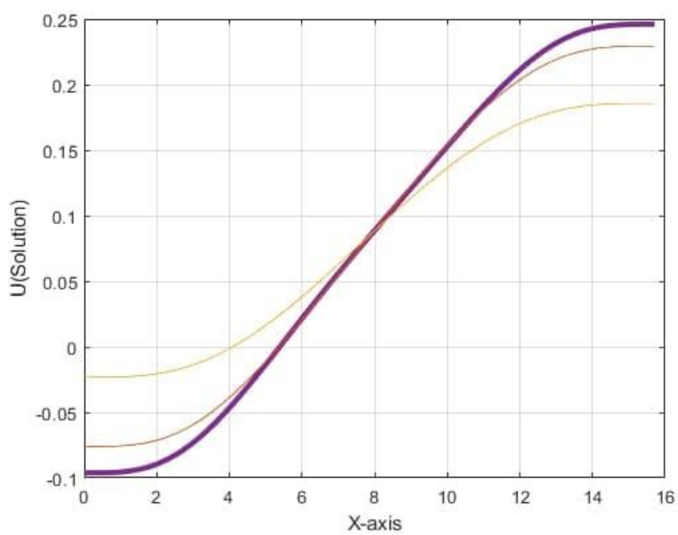

**Figer 6:** The solution at time step 80




**Figer 7:** The soluttion at time step 1500

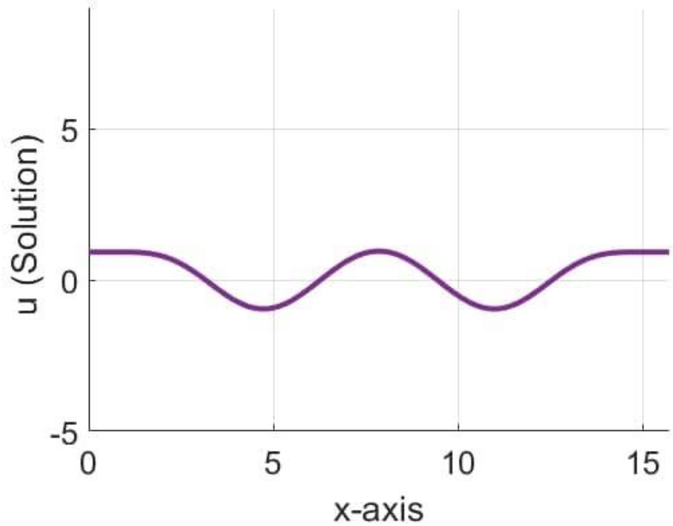



**Figer 8:** The solution at time step 10800

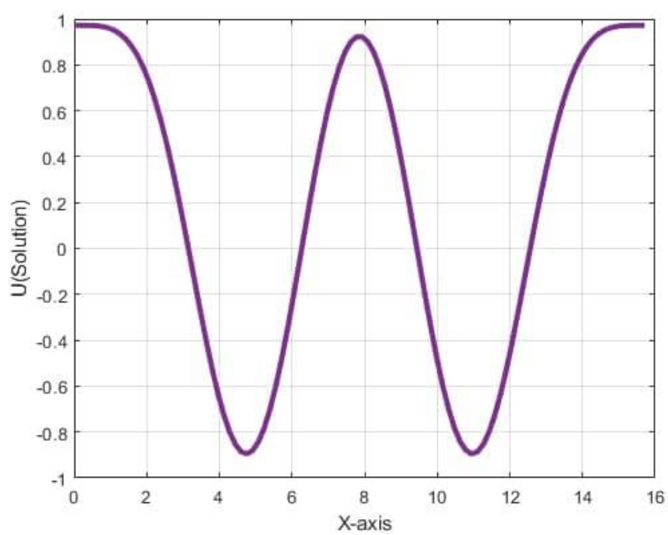



**Figer 9:** The solution at different times (8000 , 10000 , 20001 )

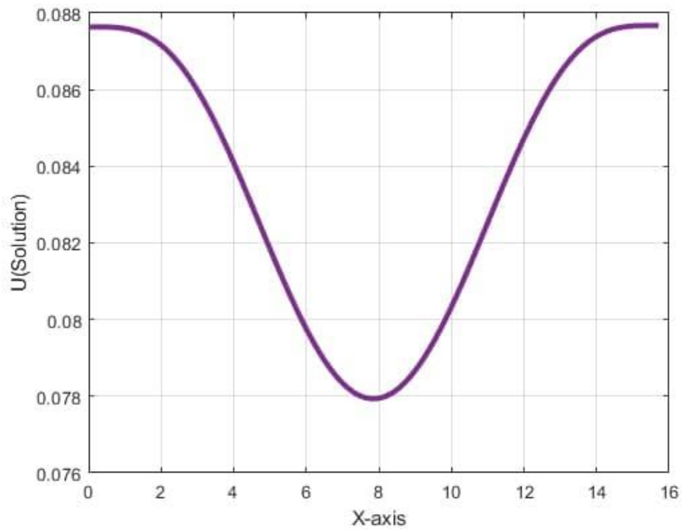
3. We will solve this equation using IRBFN and we devloped matlab code for it.
   We used intial condition which is $u_0 = \sin(x)$ with time priod [0,5pi] , $c = [xA: dx: xB]$ $M = -(-x-2)(x-2)$ and $t = 10$ and we get the following resultis.



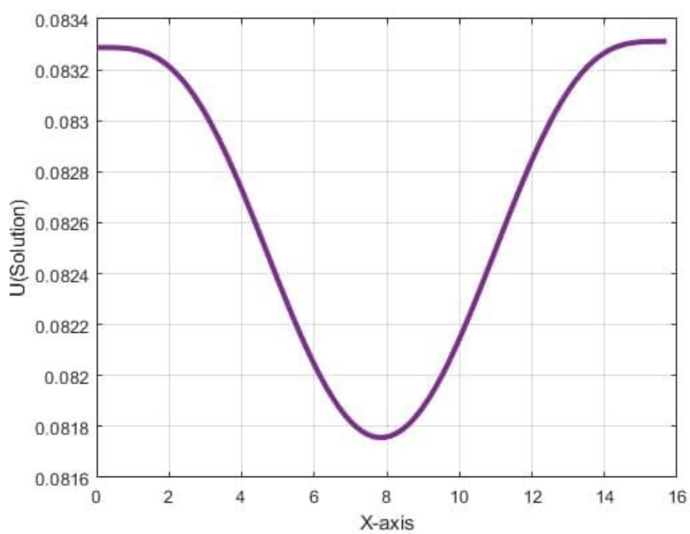

**Figer 10:** The solution at total time =20



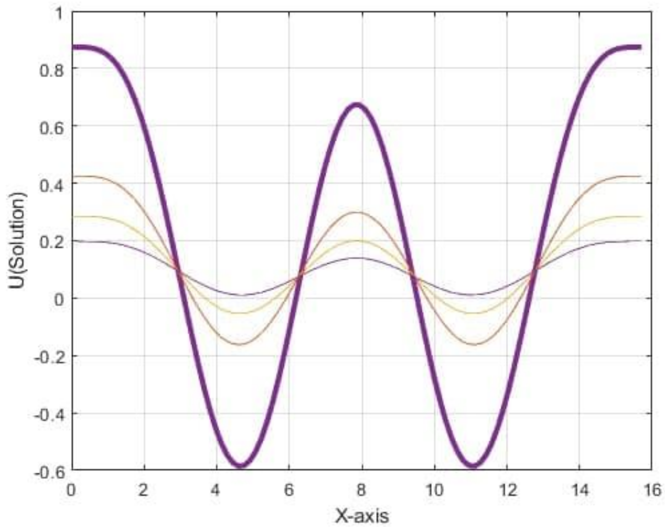

**Figer 11:** The solution at time step 60




**Figer 12:** The solution at time step 9000




**Figer 13:** The solution at time step 20001




**Figer 14:** The solution at different times (300 , 1050 , 1500 , 2000 )

### 2.1.2 Nikolaevsky equation:

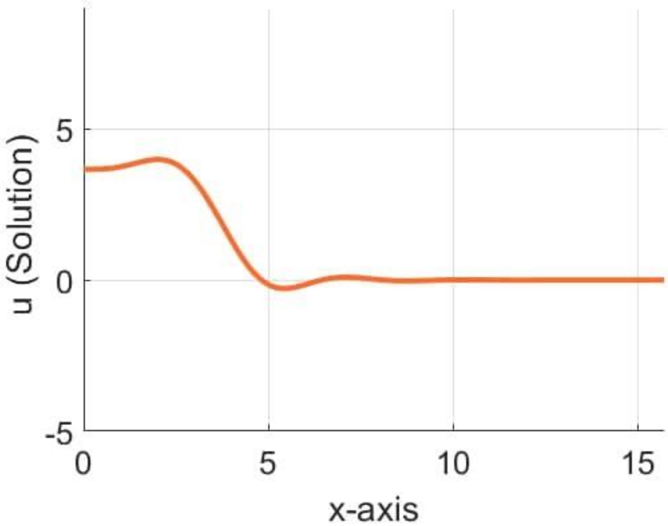
$$\frac{\partial u}{\partial t} = \frac{\partial^2 \mathrm{u}}{\partial x^2} + \left(\frac{\partial u}{\partial x}\right)^2 + \frac{\partial^4 \mathrm{u}}{\partial x^4} + \frac{\partial^6 \mathrm{u}}{\partial x^6}$$

Nikolaevskiy equation It is an equation of the sixth rank The Nikolaevskiy equation is also used to describe other physical problems such as the phase equation for reaction-diffusion systems (Fujisaka & Yamada, 2001; Tanaka, 2004; Tanaka, 2006). In addition, it also describes transverse instabilities of fronts at finite wave number (Cox & Matthews, 2007) and condensed matter evaporated by a laser beam (Anisimov, Tribel'skii, & Épel'baum, 1980). The Nikolaevskiy equation is an interesting field of study due to its physical applications and chaotic solutions.

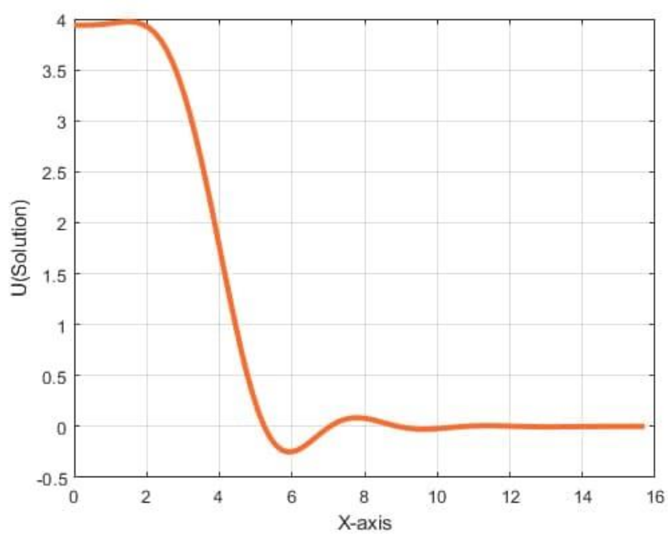
- Solve $6^{\text{th}}$ order partial differential equation.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 \mathrm{u}}{\partial x^2} + \left(\frac{\partial u}{\partial x}\right)^2 + \frac{\partial^4 \mathrm{u}}{\partial x^4} + \frac{\partial^6 \mathrm{u}}{\partial x^6}$$

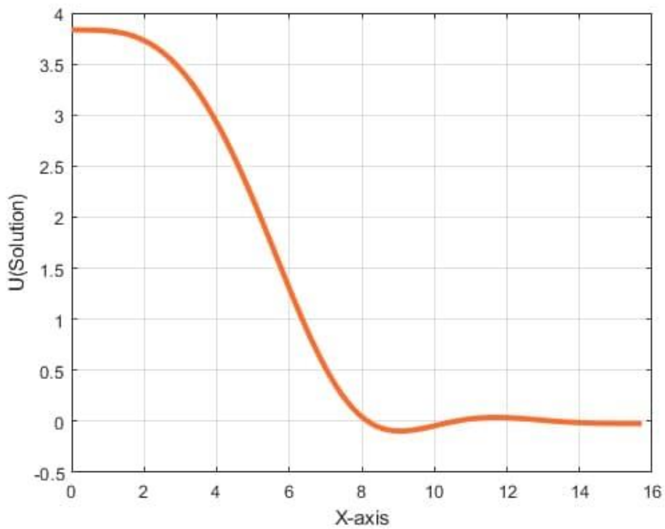
1. We will solve this equation using IRBFN and we devloped matlab code for it.
   We used intial condition which is $u_0 = 5\exp(M)$ with time priod $[0{,}5\mathrm{pi}]$ , $c = [xA{:}\,dx{:}\,xB]$
   $M = -(-x-2)(x-2)$ and $t = 10$ and we get the following resultis.



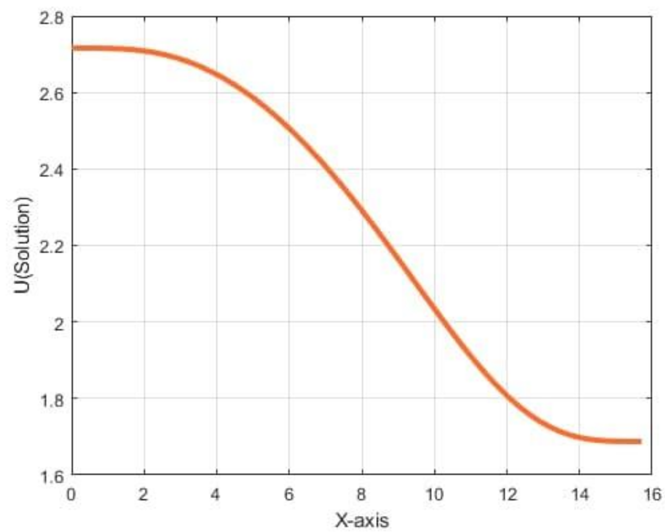

**Figer 1:** The solution at total time =20



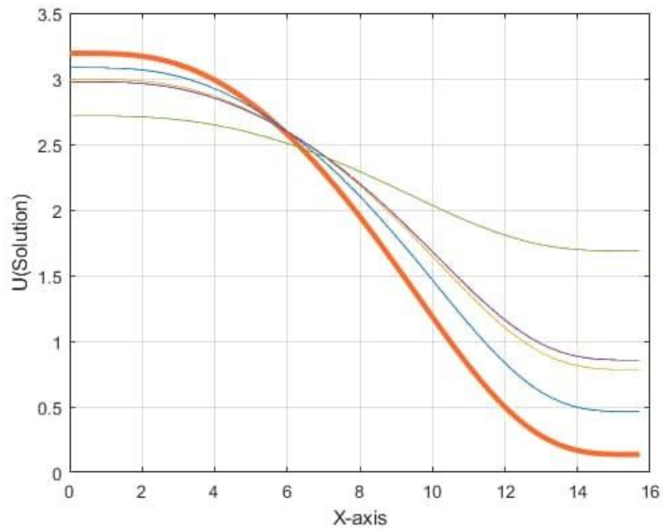

**Figer 2:** The solution at time step 60




**Figer 3:** The solution at time step 800

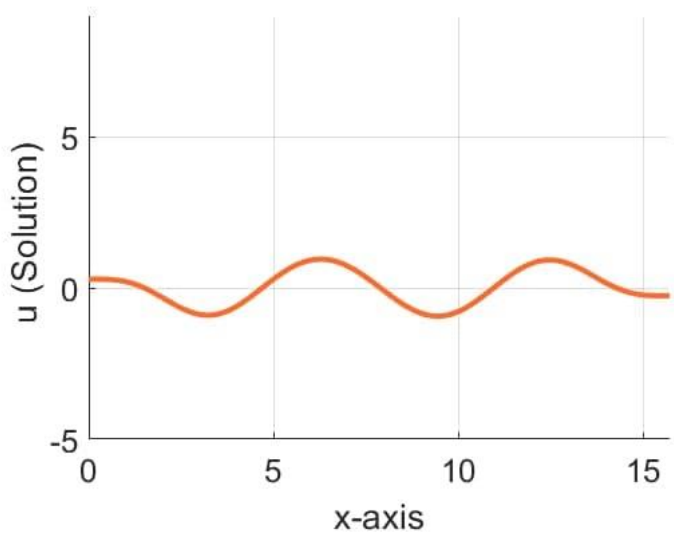



**Figer 4:** The solution at time steps =20001

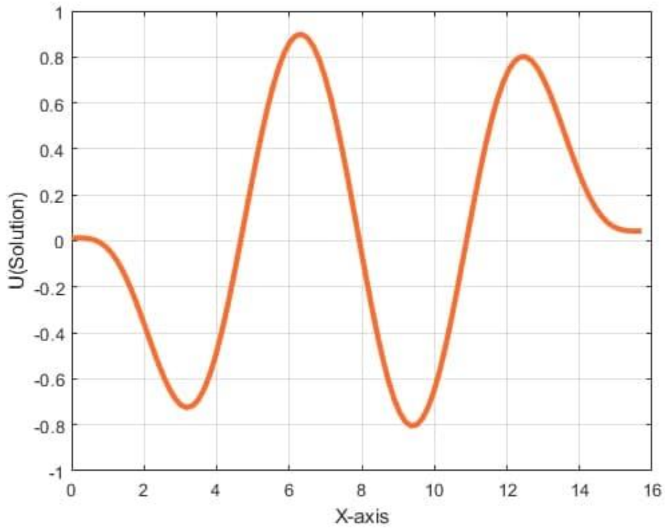



**Figer 4:** The solution at different times (6000 , 8000 , 10000 , 10500 , 20001)

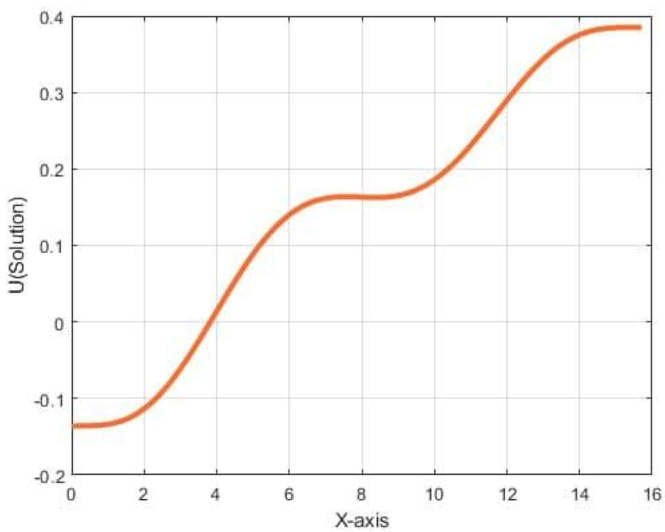
2. We will solve this equation using IRBFN and we devloped matlab code for it.
   We used intial condition which is $u_0 = \cos(x)$ with time priod [0,5pi]
   $c = [xA: dx: xB]$ , $M = -(-x-2)(x-2)$ and $t = 10$ and we get the following resultis.



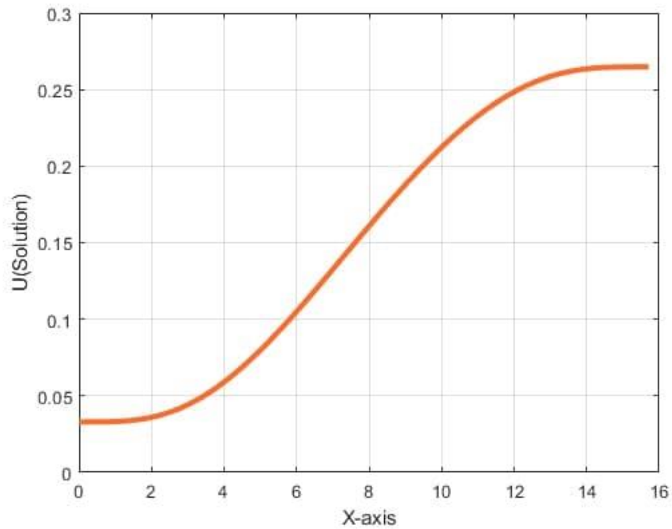

**Figer 5:** The solution at total time =20



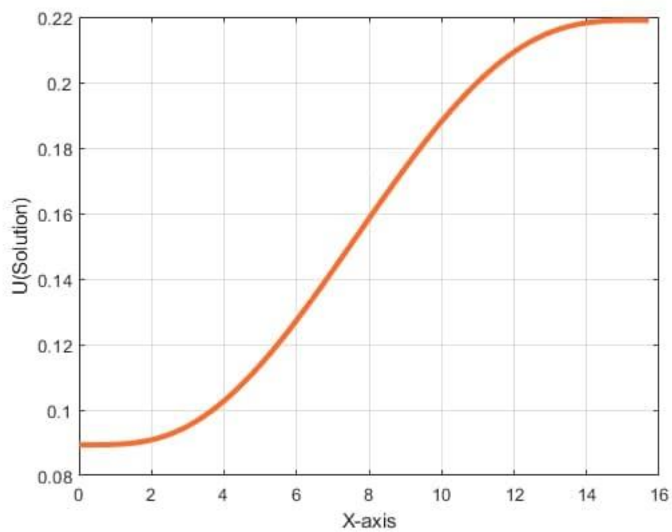

**Figer 6:** The soluttion at tim step 80



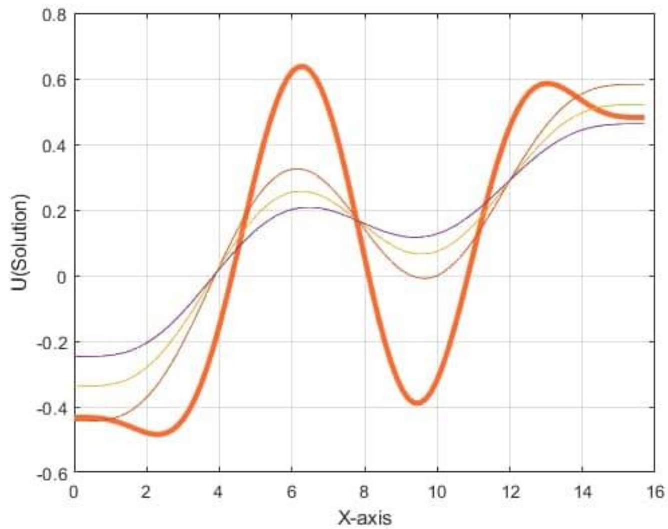

**Figer 7:** The solution at time step 3000




**Figer 8:** The solution at time step 10500

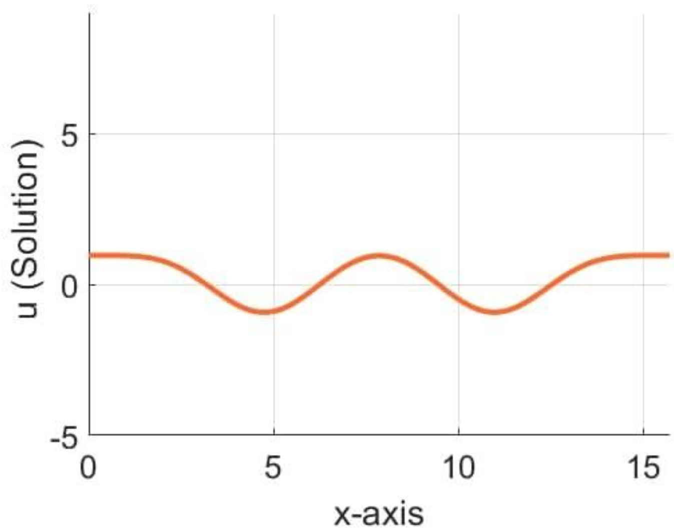



**Figer 9:** The solution at time step 20001

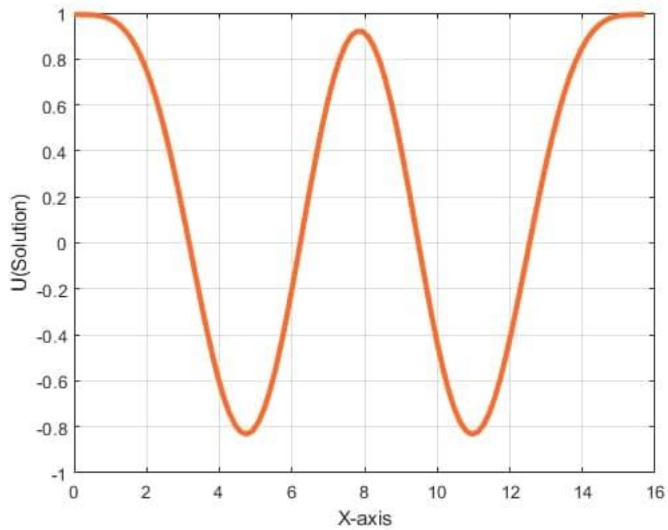



**Figer 9:** The solution at different times (300 , 1050 , 1500 , 2000 )

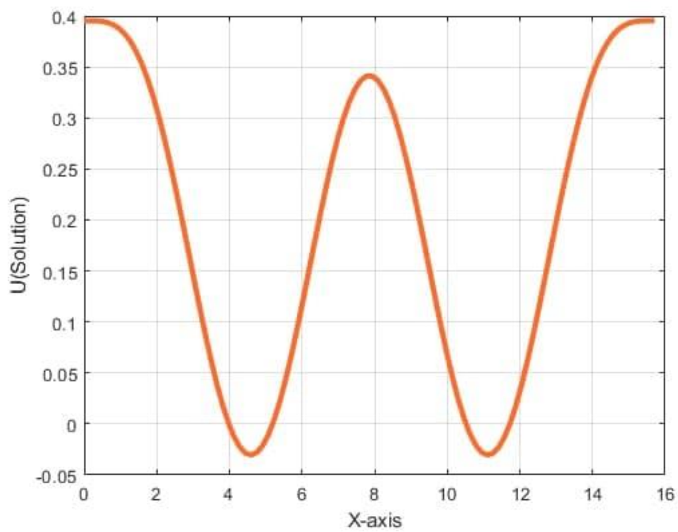
3. We will solve this equation using IRBFN and we devloped matlab code for it.
   We used intial condition which is $u_0 = \sin(x)$ with time priod [0,5pi] , $c = [xA{:}\,dx{:}\,xB]$ , $M = -(-x-2)(x-2)$ and $t = 10$ and we get the following resultis.



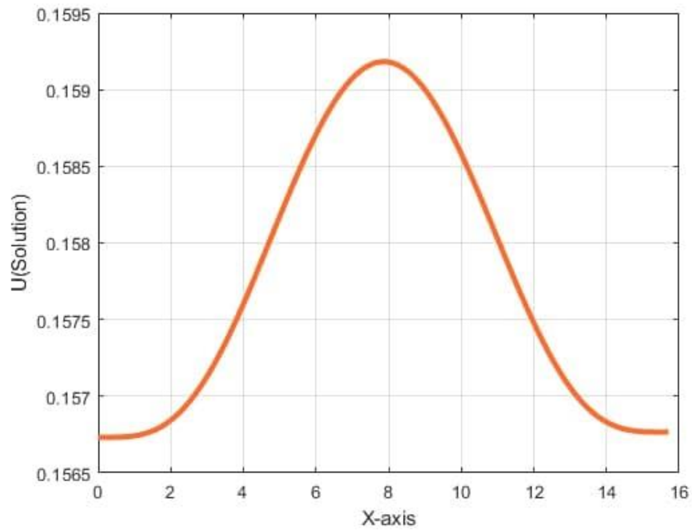

**Figer 10:** The solution at total time =20



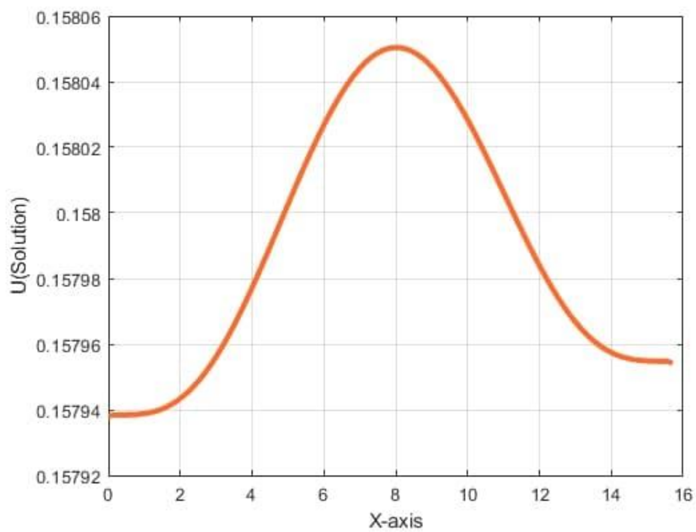

**Figer 11:** The solution at time step 60



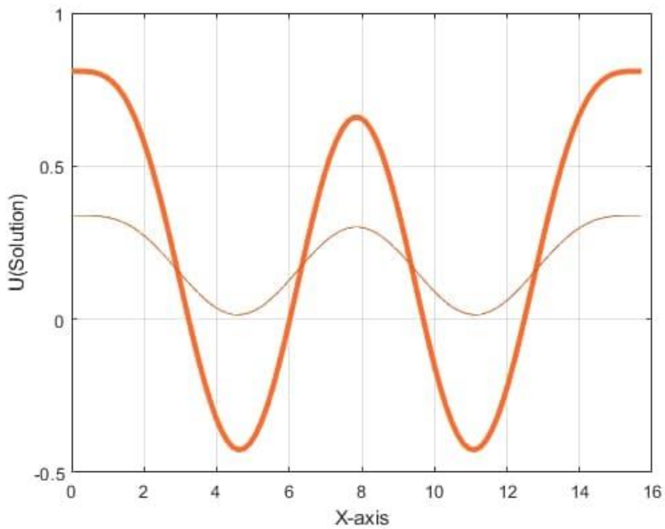

**Figer 12:** The solution at time step 900




**Figer 13:** The solution at time step 8000




**Figer 14:** The solution at time step 20001




**Figer 14:** The solution at different time (300 , 1050 )

3. CONCLUSION

We present a higher-level solution to the IRBFN phase order differential equations in this article. The approximation procedure starts with the derivative function obtained by integrations with the RBFNS original function. This is the place to be. The diagram below shows the general existence of "derivative function" and "initial function." Suppose a function $f(x)$ and its derivatives $\bar{f}\,(x)$ and $\bar{\bar{f}}\,(x)$ are to be approximated. The procedure consists of two stages. In the first stage, $f(x)$ corresponds to the original function and $\bar{f}\,(x)$ the derivative function. In the second stage the $\bar{f}\,(x)$ obtained in stage 1 corresponds to the original function and $\bar{\bar{f}}\,(x)$ the derivative function. Starting with the fourth derivative is an alternative/procedure. RBFN is used to approximate the fourth order derivative first, and then the third order derivative is obtained. The second-degree derivative and the first-degree derivative are then integrated. Finally, by integrating the initial function is derived using the first derivatiye function. The fourth indirect form, also known as IRBFN 4, is the fourth method. The derivative of the and the sixth rank are also included.

**References**

[1] M.J.D. Powell, Radial basis functions for multivariable interpolation: a review, in: J.C. Watson, M.G. Cox (Eds.), IMA Conference on Algorithms for the Approximation of Function and Data, Royal Military College of Science, Shrivenham, England, 1985, pp. 143–167.

[2] J. Moody, C.J. Darken, Fast learning in networks of locally-tuned processing units, Neural Computation (1989) 281–294.

[3] T. Poggio, F. Girosi, Networks for approximation and learning, in: Proceedings of the IEEE 78, 1990, pp. 1481–1497 .

[4] F. Girosi, T. Poggio, Networks and the best approximation property, Biological Cybernetics 63 (1990) 169–176.

[5] J. Park, I.W. Sandberg, Approximation and radial basis function networks, Neural Computation 5 (1993) 305–316.

[6] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall, New Jersey, 1999.

[7] Gutierrez RH, L.P.(1995) , Vibrations of non- uniform rings studied by means of the differential quadrature method , journal of sound and Vibration 1995 , 507-513.

[8] S.Kutluay and A.R. Bahadir , Numerical solution of one – dimensionall Burgers equation , journal of Computationall and Applied Mathematics (1999) , 251-261