International Journal of
Mathematical Modelling & Computations
Vol. 03, No. 03, 2013, 237- 244

Solving Integro-Differential Equation by Using B- Spline Interpolation

M. Amirfakhrian[a,*] and K. Shakibi[a]

[a] *Department of Mathematics, Islamic Azad University, Central Tehran Branch, Tehran, Iran.*

Abstract. In this paper, we introduce a hybrid approach based on modified neural networks and optimization teqnique to solve ordinary differential equation. Using modified neural network makes that training points should be selected over the open interval (a, b) without training the network in the range of first and end points. Therefore, the calculating volume involving computational error is reduced. In fact, the training points depending on the distance $[a, b]$ selected for training neural networks are converted to similar points in the open interval (a, b) by using a new approach, then the network is trained in these similar areas. In comparison with existing similar neural networks proposed model provides solutions with high accuracy. Numerical examples with simulation results illustrate the effectiveness of the proposed model.

Received:21 May 2013, Revised:16 July 2013, Accepted:25 August 2013.

Keywords: Ordinary Differential Equations, Modified Neural Networks.

Index to information contained in this paper

1. Introduction

Mathematical modeling of real-life problems usually results in functional equations, like ordinary or partial differential equations, integral and integro-differential equations and stochastic equations. Many mathematical formulation of physical phenomena contain integro-differential equations. These equations arises in many fields like fluid dynamics, biological models and chemical kinetics. Integro-differential equations are usually difficult to solve analytically so it is required to obtain an efficient approximate solution [4]. In [8], some methods are showed for solving Integro-differential equations such as El-gendi's, Wollfe's and Galerkin methods.

*Corresponding author. E-mail: amirfakhrian@iauctb.ac.ir

Recently, the first order linear Fredholm integro-differential equation is solved by using rationalized Haar functions method [9]. Chrysafinos [6] used wavelet-Galerkin method for solving integro-differential equations, Abbasa et al. [1] applied multi-wavelet direct method for solving integro-differential equations. In [5, 7, 10] others methods can be seen to solve integro-differential equation. In this paper we consider the Fredholm integro-differential equation of the form

$$u'(x) = \int_a^b k(x,t)u(t)dt + u(x) + f(x),$$

$$u(a) = u_0,$$

where the function $f(x)$ and the kernel $k(x,t)$ are known and $u(x)$ is the unknown function to be determined. To solve this equation. A numerical method based on B-spline interpolation is introduced and the unknown function is approximated by B-spline. The basis function of B-spline is constructed using piecewise polynomial function that satisfies C^2 continuity. General B-spline provide a highly versatile approach to describe curves in computer graphics. B-spline consist of sections of polynomial curves connected at points called knots. The polynomials of a given B-spline all have the same degree, which is the degree of the B-spline. The most used B-spline consist of cubic segments, and are called cubic B-spline.
The rest of this paper is as follows: In Section 2, the B-spline method is introduced. In Section 3, mathematical formulation of our method is explained. In Section 4, two examples are given to show the accuracy of the presented method, and finally Section 5 concludes the paper.

2. B-spline interpolation

In this section we introduce basic concept of B-spline.

Definition 1. Let a vector known as the knot vector be defined

$$T = \{t_0, t_1, ..., t_m\}, \tag{1}$$

where T is a nondecreasing sequence with $t_i \in [0,1]$, and $\mu_0, ..., \mu_n$ are considered as control points. The degree is introduced as follows

$$p \equiv m - n - 1.$$

The knots $t_{p+1}, ..., t_{m-p-1}$ are called internal knots. The jth B-spline of degree k is defined by

$$B_j^0(t) = \begin{cases} 1 & t_i \leqslant t < t_{i+1}, \\ 0 & otherwise \end{cases} \tag{2}$$

and

$$B_j^k(t) = \frac{t - t_j}{t_{j+k} - t_j} B_j^{k-1}(t) + \frac{t_{j+k+1} - t}{t_{j+k+1} - t_{j+1}} B_{j+1}^{k-1}(t). \tag{3}$$

A B-spline curve of degree k is a polynomial curve introduced as bellow

$$C(t) = \sum_{j=0}^{n} \mu_j B_j^k(t). \tag{4}$$

The B-spline basis functions also have some nice properties:

Local support: Each cubic curve piece is only a function of the four closest control points.

The basis functions are translates of each other:

$$B_j^k(x) = B_j^0(x - j).$$

The curve is C^{k-2} continuous.

The basis functions are a partition of unity.

Affine invariance.

3. Solving FIDE by using B-spline

Consider the following FIDE :

$$u'(x) = \int_a^b k(x,t)u(t)dt + u(x) + f(x), \tag{5}$$

$$u(a) = u_0,$$

where the function $f(x)$ and the kernel $k(x,t)$ are known and $u(x)$ is the unknown function. By considering Eq. (6) and integrating it from a to x, we get

$$u(x) - u(a) = \int_a^x (\int_a^b k(\varepsilon,t)u(t)dt + u(\varepsilon) + f(\varepsilon))d\varepsilon. \tag{6}$$

The unknown function $u(x)$ is approximated by B-spline interpolation which is introduced in pervious section

$$\sum_{j=1}^{n+d} \mu_j B_j(x) = u(a) + \int_a^x (\int_a^b k(\varepsilon,t) \sum_{j=1}^{n+d} \mu_j B_j(t)dt + \sum_{j=1}^{n+d} \mu_j B_j(\varepsilon) + f(\varepsilon))d\varepsilon. \tag{7}$$

So it can be written as follows

$$\sum_{j=1}^{n+d} \mu_j (B_j(x) - \int_a^x (\int_a^b k(\varepsilon,t)B_j(t)dt - B_j(\varepsilon))d\varepsilon = u(a) + \int_a^x f(\varepsilon)d\varepsilon. \tag{8}$$

If we put $x = x_i, \quad i = 0,1,2,...,n,$, we have $n+d$ unknown $\mu_1, \mu_2, ..., \mu_{n+d}$ and $n+1$ equation. So we set $d-1$ additional equation. For $d = 3$ to obtain two more conditions, require that the second derivatives at the endpoints be zero, so :

$$C^{(2)}(x_0) = C^{(2)}(x_n) = 0.$$

Equivalently, the following linear system of equations can be solved:

$$AX = b, \tag{9}$$

where $A = [a_{ij}]$, $i, j = 1, 2, 3, ..., n + d$ in which

$$a_{ij} = B_j(x_i) - (\int_a^{x_i} (\int_a^b k(\varepsilon, t)B_j(t)dt)d\varepsilon - (\int_a^{x_i} B_j(\varepsilon)d\varepsilon)) \tag{10}$$

and for $j = 1, 2, 3, ..., n + d, i = n + d, n + d - 1$

$$a_{ij} = B_j^{(2)}(x_i)$$

and

$$X = \begin{pmatrix} \mu_1 \\ \mu_2 \\ . \\ . \\ . \\ \mu_n \end{pmatrix}, \quad b = \begin{pmatrix} u(a) + \int_a^{x_1} f(\varepsilon)d\varepsilon \\ u(a) + \int_a^{x_2} f(\varepsilon)d\varepsilon \\ . \\ . \\ . \\ u(a) + \int_a^{x_{n+d}} f(\varepsilon)d\varepsilon \\ 0 \\ 0 \end{pmatrix}.$$

Now it is sufficient to substitute μ_j, $j = 1, ..., n + d$, in:

$$u^*(x) = \sum_{j=1}^{n+d} \mu_j B_j(x),$$

where $u^*(x)$ is an approximation of $u(x)$.

4. Examples

In this section, we apply our method to solve two integro-differential equations. The examples presented have been solved by commercial software Mathematica.
Example 1. Consider the following integro-differential equation

$$u'(x) = \int_0^1 xtu(t)dt + u(x) - 2\sin(x) - x(-1 + 2\sin(1)),$$

where

$$u(0) = 1,$$

and the exact solution is:

$$u(x) = \cos(x) + \sin(x).$$

We use cubic B-spline interpolation. The absolute errors are presented in Table 1 and we show the results in Figure 1.

Table 1. The results for example 1.

x	Exact solution	Numerical solution	Absolute error
0	1	1	0
0.1	1.09484	1.0948	0.0000327928
0.2	1.17874	1.17871	0.0000288661
0.3	1.25086	1.25082	0.0000359381
0.4	1.31048	1.31044	0.0000417093
0.5	1.35701	1.35696	0.0000494565
0.6	1.38998	1.38992	0.0000588735
0.7	1.40906	1.40899	0.0000688925
0.8	1.41406	1.41398	0.0000849139
0.9	1.40494	1.40485	0.0000879359
1	1.38177	1.38163	0.000148082

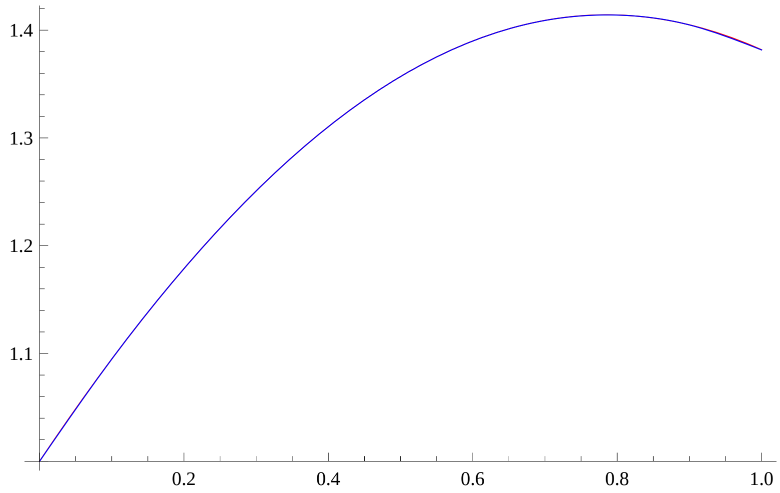

Figure 1. Error function of the Example 1.

Example 2. Consider the following integro-differential equation

$$u'(x) = \int_0^1 \exp(x+t)u(t)dt + u(x) - 0.5\exp(x)(\exp(2)-1),$$

where

$$u(0) = 1,$$

and the exact solution is:

$$u(x) = \exp(x).$$

We use cubic B-spline interpolation. The absolute errors are presented in Table 2 and we show the results in Figure 2.

Table 2. The results for example 2.

x	Exact solution	Numerical solution	Absolute error
0	1	1	$2.22045E^{-16}$
0.1	1.10517	1.10518	$4.6255E^{-16}$
0.2	1.2214	1.22137	0.0000339801
0.3	1.34986	1.34979	0.0000692134
0.4	1.49182	1.49171	0.000114455
0.5	1.64872	1.64855	0.000167833
0.6	1.82212	1.82189	0.000230738
0.7	2.01375	2.01345	0.000307301
0.8	2.22554	2.22515	0.000388929
0.9	2.4596	2.45909	0.000515254
1	2.71828	2.71773	0.00055131

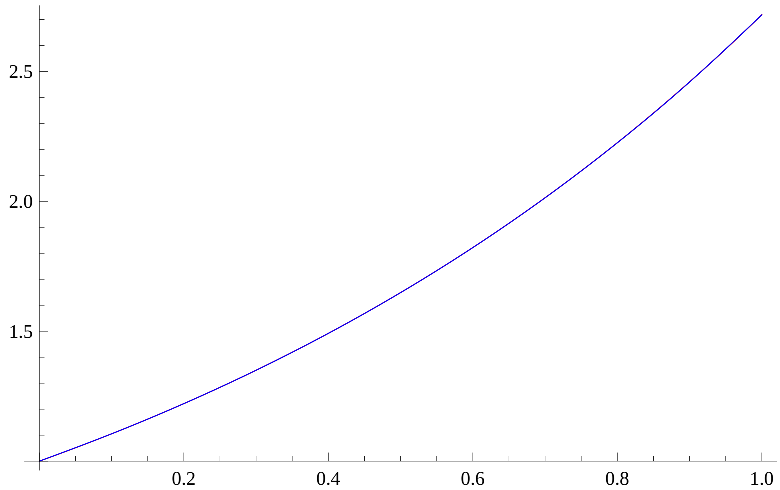

Figure 1. Error function of the Example 2.

5. Conclusion

In this paper, a numerical scheme based on B-spline is presented for solving for integro-differential equation. From the test examples, we can say that this scheme is feasible and the accuracy is acceptable.

References

[1] Abbasa Z., Vahdatia S., Atanb K.A., Nik Longa N. M. A., *Legendre multi-wavelets direct method for linear integro-differential equations*, Applied Mathematical Sciences, **3** (**14**) (2009) 693-700.

[2] Amirfakhrian M., *Approximation of 3D-Parametric Functions by Bicubic B-spline Functions*, International Journal of Mathematical Modelling & Computations, **2** (**3**) (2012) 211–220.

[3] Amirfakhrian M., Nouriani H., *Interpolation by Hyperbolic B-spline Functions*, International Journal of Mathematical Modelling & Computations, **1** (**3**) (2011) 175-181.

[4] Batiha B., Noorani M. S. M., Hashim I., *Numerical Solutions Of The Nonlinear Integro-Differential Equations*, Open Problems Compt. Math. **1** (**1**) (2008) 34–42.

[5] Crisci. M. R., *Stability results for one-step discretized collocation methods in the numerical treatment of Volterra integral equations*. Math. Comput., **197** (1992) 119-134.

[6] Chrysafinos K. *Approximations of parabolic integro-differential equations using wavelet-Galerkin compression techniques*, BIT Numerical Mathematics, **47** (2007) 487-505.

[7] Chuong N. M., Tuan N. V., *Spline collocation methods for Fredholm integro-differential equations of second order*, Acta Math. Vietnamica, **20** (1995) 85-98.

[8] Delves L. M., Mohamad J. L., *Computational methods for integral equation*. Cambridge university Press, Cambridge (1985).

[9] Maleknejad K., Mirzae F., Abbasbandy S., *Solving integro-differential equations system by using rationalized Haar functions method*, Applied mathematics and computation, **155** (2004) 317-328.

[10] Yalcinbas S., Sezer M., *The approximate solution of high-order linear Volterra Fredholm integro-differential equations in terms of Taylor polynomials*, Appl. Math. Comput., **112** (2000) 291-308.