International Journal of Mathematical Modelling and Computation
Vol 1, Issue 1 2025

The Epsilon Modified Block-Pulse Function is used to approximate the solution of the second order initial value problem.

M. Mohanuadi and S. Khezerlnu

Department of Mathematics, Islamic Azad university–south Tehran Branch, Tehran, Iran

Article Info

Received: 31-05-2025 Revised:08-06-2025 Accepted:19-06-2025 Published:30-06-2025

Abstract. The current study tackles the issue of converting second order initial value problems (IVPs) into a Volterra integral equation of the second type (VIE2) in order to approximate their solution. Since the IVPs may be converted to a lower triangular system of algebraic equations, we first solve them using Runge–Kutta of the forth–order technique (RK), after which we convert them into VIE2 and solve them using the εmodified block–pulse functions (εMBPFs) and their operational matrix. The suggested approach has an appropriate level of accuracy, as shown by numerical examples.

Keywords: Initial value problems; Runge–Kutta method; Volterra integral equation; εmodified block–pulse function.

Index to information contained in this paper

1. Introduction

Studies have shown the widespread usage of differential equations in mathematics [1, 16], physics [3], biology, and engineering [12, 13] to solve a variety of issues. Most of the issues listed need an IVP solution, which calls for the differential equation's solution to meet certain beginning conditions. Ordinary differential equation (ODE) solutions have been estimated using a variety of techniques in recent years.For instance, Mastorakis et al. [12] solved the second-order problem by combining the genetic algorithm with the Neider-Mead technique. Moreover, IVP has the form $y'' = f(x,y)$ Mateescu [13] approximated the solution of second order IVPs using the standard evolutionary method. Neural network adaptation was also shown in another study to handle second-order IVPs [8]. Furthermore, the adaptation of the differential evolution method to solve the second order IVPs of the type $y'' + a1(x)y' + a0(x)y = b(x)$ was suggested by Fatimah et al. [5]. By converting the second order IVPs into an optimization problem, Bilesanmi et al. [3] were able to get an approximate solution. Edeki et al. [4] used the DTM approach in a different research to solve both linear and nonlinear IVPs of second order ODEs. Additionally, it was suggested by Fatimah et al. [5] that the differential evolution (DE) technique may be used to produce very accurate approximations for second order IVPs. Lastly, Hasan examined the scientific computation of IVP in [6]. Considering the second-order differential equation:

$$y'' = f(x,y,y') \tag{1}$$

an IVPs for a second order differential eqation is the problem of finding a solution $y(x)$ to Eq. (1) that satisfies an initial conditions $y(x_0)$ and $y'(x_0)$ are the fixed states. We cosider the IVP as follows:

$$\begin{cases} y'' = f(x,y,y') \\ y(x_0) = y_0{}^{y'}(x_0) = y_1 \end{cases} \tag{2}$$

The present paper intended to solve Eq. (2) via Runge–Kutta of the forth–order method [1, 2]. For this purpose, IVP of the second order is converted into a VIE2 and this equation is solved by applying the basis function εmodified block-pulse [15]. One of the reasons for adopting this approach is that some problems of ODEs via analytical methods of ODE like the Euler method [14], Taylor method [14], Runge–Kutta method and so, did not give a good approximate; therefore, the problems were converted into VIE2 to improve the approximate solution.

It should be mentioned that section 2 reviews differential equation and RK method and section 3 reviews the εmodified block-pulse function. In addition, section 4 deals with the proposed method and section 5 presents Error analysis. Then, section 6 gives the numerical results, and we show the results of Theorem 5.2 for all examples in Table 5 and section 7 concludes the study.

2. Differential equation of the second–order

This section reviews differential eqation [3] and Rung–Kutta method [1] that we used for obtaining the approximate solution of differential equation of the second kind.

Definition 2.1 ([2]) It is widely accepted that a second order linear differential equation for function y would be

$$y'' + a_1(x)y' + a_0(x)y = b(t) \tag{3}$$

so that a_1, a_0, b refer to the functions on the interval $I \subset R$. In addition, Eq. (3) (a) would be homogeneos iff the source $b(x) = 0$ for all $x \in R$.

(b) would have constant coefficients if a_1 and a_0 are constants, and (c) would have variable coefficients if either a_1 or a_0 is not constant.

2.1 Runge–Kutta of forth–order method [1, 2]

The RK scheme is a method with the greatest utilization to solve the differential equation with numerical procedures. In order to compute the solution of a first order IVP. The following relations were utilized for a RK method of the forth– order [1]

$$\begin{cases} k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 = hf(x_n + h, y_n + k_3) \\ y_{n+1} = y_n + \frac{1}{6}(k_1 + 2K_2 + 2K_3 + k_4) + O(h^5) \end{cases}$$

Moreover, the RK method could be utilized for the second order differential equation of the form

$$y'' = f(x,y,y')$$

for the second order differential equations, thus, the forth–order formulas would be [2]

$$\begin{cases} k'_1 = hf(x_n, y_n, y'_n) \\ k'_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, y'_n + \frac{k'_1}{2}) \\ k'_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, y'_n + \frac{k'_2}{2}) \\ k'_4 = hf(x_n + h, y_n + k_3, y'_n + k'_3) \\ y'_{n+1} = y'_n + \frac{1}{6}(k'_1 + 2k'_2 + 2k'_3 + k'_4) + O(h^5) \end{cases}$$

3. εModified block–pulse function (εMBPFs)

It is notable that many authors investigated and reviewed the block–pulse functions (BPFs) method and used it to solve many problems, definition, vector forms, BPFs expansion and operational matrix [7, 11]. This section presents a review of εMBPFs.

Definition 3.1 ([10]) A $(n+1)$–set of εMBPFs $\phi_i(t)$, $i = 0,...,n-1$ on the interval $[0,T)$ is defined as:

$$\phi_0(t) = \begin{cases} 1 & , t \in \left[0, \frac{T}{n} - \varepsilon\right) = I_0 \\ 0 & , \text{o.w} \end{cases}$$

$$\phi_n(t) = \begin{cases} 1 & , t \in [T - \varepsilon, T) = I_n \\ 0 & , \text{o.w} \end{cases} \tag{4}$$

$$\phi_i(t) = \begin{cases} 1 & , t \in \left[\frac{iT}{n} - \varepsilon, \frac{(i+1)T}{n} - \varepsilon\right) = I_i, \quad 0 < i < n \\ 0 & , \text{o.w} \end{cases}$$

Therefore, there are some properties for εMBPFs as follows:
εMBPFs are disjoint and orthogonal

$$\phi_i(t)\phi_j(t) = \begin{cases} \phi_i(t) & , i = j \\ 0 & , i \neq j \end{cases}, \quad i, j = 0, \ldots, n$$

$$\int_0^1 \phi_i(t)\phi_j(t)dt = h\delta_{ij},$$

and εMBPFs like BPFs are complete:

$$\int_0^1 f^2(t)dt = \sum_{i=0}^{\infty} f_i^2 \|\phi_i(t)\|^2 .$$

Using notation $\Phi_n(t) = [\phi_0(t),...,\phi_n(t)]^T$, the following properties are achieved:

$\phi_0(t) 0 \quad 0 \ldots \quad 0$

$\Phi_{n+1}(t)\Phi_{n+1}^T(t) = 0 \ldots \quad \phi_1(t)\ 0 \ldots \quad 0_0$

$0 \quad \ldots 0 \ldots \phi_n(t)$

$\Phi^T_{n+1}(t)\Phi_{n+1}(t) = 1$

$$\Phi_{n+1}(t)\Phi^T_{n+1}(t)V = \tilde{V}\Phi_{n+1}(t) \tag{5}$$

$$\Phi^T_{n+1}(t)B\Phi_{n+1}(t) = \hat{B}^T\Phi_{n+1}(t), \tag{6}$$

if $h = \frac{T}{n}$ then the operational matrix of εMBPFs would be defined in this way:

$$P_{(n+1)\times(n+1)} = \begin{bmatrix} \frac{h-\varepsilon}{2} & h-\varepsilon & h-\varepsilon & \ldots & h-\varepsilon & h-\varepsilon \\ 0 & \frac{h}{2} & h & \ldots & h & h \\ 0 & 0 & \frac{h}{2} & \ldots & h & h \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \frac{h}{2} & h \\ 0 & 0 & 0 & \ldots & 0 & \frac{\varepsilon}{2} \end{bmatrix}, \tag{7}$$

and it has features and usage similar to the operational matrix BPFs in [3, 11].

Definition 3.2 ([10]) εMBPFs expansion of continuous function $f(t) \in L^2([0,1))$ based on ϕ_i, $i = 0,\ldots,n$ would be defined as:

$$f(t) \simeq \hat{f}_{n+1} = \sum_{i=0}^{n} f_i\phi_i(t),$$

where

$$f_i = \frac{1}{\Delta(I_i)}\int_0^1 f(t)\phi_i(t)dt,$$

and $\Delta(I_i)$ is the length of interval I_i defined (4).

4. Main idea

Let

$$\begin{cases} y'' = f(x,y,y') \\ y(x_0) = \alpha, y'(x_0) = \beta \end{cases}$$

or

$$\begin{cases} y'' + a_1(x)y' + a_0(x)y = b(x) \\ y(x_0) = \alpha, y'(x_0) = \beta \end{cases} \tag{8}$$

where α and β are the given constants.
For this work, we transform Eq. (8) into VIE2. Therefore, we consider

$$y''(x) = q(x). \tag{9}$$

Then, both sides of Eq. (9) from 0 to x is integrated and yielded

$$\int_0^x y''(x)dx = \int_0^x q(t)dt$$
$$y'(x) - y'(x_0) = \int_0^x q(t)dt$$
$$\Longrightarrow y'(x) = \beta + \int_0^x q(t)dt. \tag{10}$$

Again, both side of the Eq. (10) would be integrated based on x from 0 to x so that:

$$\int_0^x y'(x)dx = \int_0^x \beta dx + \int_0^x \int_0^{x_1} q(t)dtdx_1$$
$$\Longrightarrow y(x) = \alpha + \beta x + \int_0^x (x-t)q(t)dt. \tag{11}$$

Then, substituting Eq. (9), Eq. (10), and Eq. (11) into Eq. (8) gives

$$q(x) + a_1(x)(\beta + \int_0^x q(t)dt) + a_0(x)(\alpha + \beta x + \int_0^x (x-t)q(t)dt) = b(x)$$

Then,

$$q(t) = b(x) - \beta a_1(x) - \alpha a_0(x) - \beta x a_0(x) - \int_0^x (a_1(x) - a_0(x)(x-t))q(t)dt.$$

Finally, we obtain the VIE2 as follows:

$$q(x) = f(x) + \int_0^x k(x,t)q(t)dt, \tag{12}$$

so that

$$f(x) = b(x) - \beta a_1(x) - \alpha a_0(x) - \beta x a_0(x), \qquad k(x,t) = -(a_1(x) + a_0(x)(x-t)).$$

Then, applying εMBPFs method for solving Eq. (12) and approximating functions f, q and k with respect to εMBPFs would give:

$$k(x,t) \simeq \Phi^T(x)K\Phi(t)$$
$$f(x) \simeq F^t\Phi(x) = \Phi^T(x)F \tag{13}$$
$$q(x) \simeq Q^T\Phi(x) = \Phi^T(x)Q$$

Here, m–vectors F, Q, and $m \times m$ matrix K respectively stand for εMBPFs coefficients of f, q and k. Note that Q in Eq. (13) is the unknown vector and should be obtained. Therefore, substituting (13) into (12) gives

$$\begin{aligned} Q^T\Phi(x) &\simeq F^T\Phi(x) + \int_0^t \Phi^T(x)K\Phi(t)\Phi^T(t)Qds \\ &= F^T\Phi(x) + \Phi^T(x)K\int_0^t \Phi(t)\Phi^T(t)Qds. \end{aligned} \tag{14}$$

Using (5) and operational matrix P in (7), we have

$$Q^T\Phi(x) \simeq F^T\Phi(x) + \Phi^T(x)KQP^{\sim} \quad \Phi(x), \tag{15}$$

where, $K^{QP\sim}$ represets $(n+1)\times(n+1)$ matrix. Thus, if ε equals to 0, just n BPFs would exist and the vectors dimension and matrices decline to n. By Eq. (6), we can write:

$$Q^T\Phi(x) \simeq F^T\Phi(x) + Q^{\wedge T}\Phi(x),$$

where $Q^{\wedge}$ refers to $(n+1)$–vector with the components equivalent to diagonal entries of the matrix $KQP^{\sim}$. Finally,

$$Q - Q^{\wedge} = F.$$

Therefore, the vector $Q^{\wedge}$ could be written as follows:

$$\hat{Q} = \begin{bmatrix} \frac{h-\varepsilon}{2}k_{0,0}q_0 \\ (h-\varepsilon)k_{1,0}q_0 + \frac{h}{2}k_{1,1}q_1 \\ (h-\varepsilon)k_{2,0}q_0 + hk_{2,1}q_1 + \frac{h}{2}k_{2,2}q_2 \\ \vdots \\ (h-\varepsilon)k_{n,0}q_0 + hk_{n,1}q_1 + \cdots + hk_{n,(n-1)}q_{n-1} + \frac{\varepsilon}{2}k_{n,n}q_n \end{bmatrix}.$$

Now, substituting (15) into Eq. (14) would give:

$$G = \begin{bmatrix} 1-\left(\frac{h-\varepsilon}{2}\right)k_{0,0} & 0 & 0 & \dots & 0 \\ -(h-\varepsilon)k_{1,0} & 1-\left(\frac{h}{2}\right)k_{1,1} & 0 & \dots & 0 \\ -(h-\varepsilon)k_{2,0} & -hk_{2,1} & 1-\left(\frac{h}{2}\right)k_{2,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -(h-\varepsilon)k_{n,0} & -hk_{n,1} & -hk_{n,2} & \dots & 1-\left(\frac{\varepsilon}{2}\right)k_{n,n} \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}.$$

Now, replacing $\simeq$ with =, Eq. (11) reduces to a linear lower triangular system as:

$$G = F$$

Cosequently, unknown cofficients Q_j, $j = 0,1,...,n$ are calculated by solving this linear equation system. Now, if $\varepsilon_j = \frac{jh}{k}$, $j = 0,1,...,n-1$, there would be k numerical answers of $\hat{f}_{\varepsilon_j}$, $j = 0,...,n-1$ so that based on theorem 5.2, we can estimate the error of:

$$\bar{f}(t) = \left(\frac{1}{k}\right)\sum_{j=0}^{n-1}\hat{f}_{\varepsilon_j}(t).$$

5. Error analysis

This section addresses error analysis. For simplicity we assume $T = 1$ and $h = \frac{1}{n}$ in the following theorem.

Theorem 5.1 *If* $\hat{f}_n = \sum_{i=0}^{n} f_i\phi_i(t)$ *and* $f_i = \frac{1}{\Delta(I_i)}\int_0^1 f(t)\phi_i(t)dt,\ i = 0,\dots,n$*, then:*

$(i)\ \delta = \int_0^1 (f(t) - \sum_{i=0}^{n} f_i\phi_i(t))^2 dt$*, achieves its minimum value.*

(ii) $\{\hat{f}_n(t)\}$ *approaches* $f(t)$ *point wise.*

$(iii)\ \int_0^1 f^2(t)dt = \sum_{i=0}^{\infty} f_i^2\|\phi_i\|^2.$

Proof See [9]. ∎

Theorem 5.2 *Suppose that*

(1) $f(t)$ *is continuous and could be differentiated in* $[-h,1+h]$ *with bounded derivative so that* $|f'(t)| < M$.

$(2)\ \hat{f}_{\frac{ih}{k}}(t),\ i = 0,1,\dots,n-1$ *are correspondingly BPFs.*

$h\ MBPs,\ \dots,\ \frac{(k-1)h}{k}\_k$ *MBPs expansions of* $f(t)$ *base on* $(m+1)$ *εMBPFs over internal* $[0,1)$.

$(3)\ \bar{f}(t) = \left(\frac{1}{k}\right)\sum_{i=0}^{n-1}\hat{f}_{\frac{ih}{k}}(t),$

then, $\|f(t) - \hat{f}_{\frac{ih}{k}}(t)\| = O(h)$*, and* $\|f(t) - \bar{f}(t)\| = O(\frac{h}{k})$ *in* $[h,1-h]$.

Proof See [10]. ∎

6. Numerical examples

The εMBPFs, is applied for examples. As seen in the examples below, n refers to the number of the block–pulse function and represents the times of modification if k is equal to 0. Moreover, expansion was on the basis BPF; in other ways, expansion was on the basis of εMBPFs. In these examples, the approximate solution presented method was compared with the exact solution and RK method. Tables 1–4 presents the numerical results of Examples 1–4, respectively. Moreover, Table 5 reports the results of Theorem 5.2. For each example, we have two figures, one of the figures compared the approximate solution by RK method with the exact solution, and the other figure made a comparison between the approximate solution of the present method with the exact solution. The computations related to the examples were performed using Matlab R2017a.

Example 6.1 Consider the following IVP:

$$\begin{cases} y''(x) + y'(x) = -e^{-x} \\ y(0) = 0, y'(0) = 1 \end{cases} \tag{16}$$

with the exact solution $y(x) = xe^{-x}$. We converted Eq. (16) into VIE2 and considered $y''(x) = q(x)$ by integrating both sides. Finally, we have:

$$q(x) = 1 - e^{-x} + \int_0^x -q(t)dt.$$

Table 1 reports the numerical results. Moreover, Figure 1 shows the results of the exact and approximate solution by RK4 method and Figure 2 depicts the result of the present method.

Table 1. Numerical results for Example 6.1.

Value x	Exact solution	Approximate solution by RK4	$n = 128, k = 0$ Present method	$n = 128, k = 3$ Present method
0	0.000000	0.000000	0.003881	0.002915
0.1	0.090484	0.090325	0.088562	0.093328
0.2	0.163746	0.162999	0.163228	0.161943
0.3	0.222245	0.220562	0.222645	0.221632
0.4	0.268128	0.265234	0.269064	0.268281
0.5	0.303265	0.298945	0.304440	0.303853
0.6	0.329287	0.323378	0.328768	0.330052
0.7	0.347610	0.339994	0.347492	0.347199
0.8	0.359463	0.350058	0.359532	0.359357
0.9	0.365913	0.354667	0.366006	0.365928

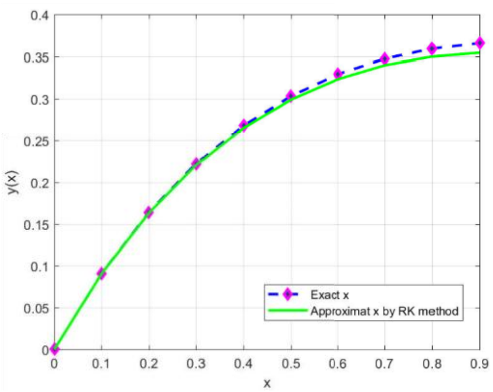


Figure 1. The result solution by RK4 (Exammethod (Example 6.1).

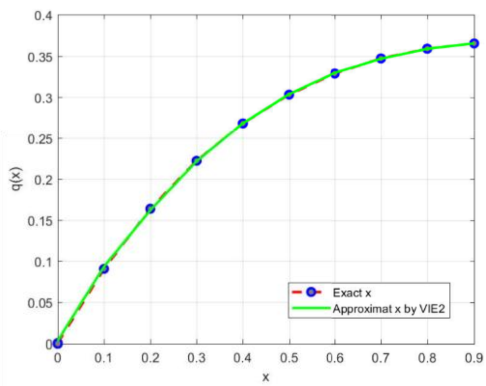


Figure 2. The result solution of the proposed scheme for $n = 128, k = 3$

ple 6.1).

Example 6.2 Consider the following IVP:

$$\begin{cases} y''(x) + y(x) = -2 + x - x^2 \\ y(0) = 0, y'(0) = 1 \end{cases} \tag{17}$$

with the exact solution $y(x) = x - x^2$. Therefore, we converted of Eq. (17) into VIE2 and considered $y''(x) = q(x)$ by integrating both sides in order to have:

$$q(x) = x - x^2 + \frac{x^3}{6} - \frac{x^4}{12} + \int_0^x (t - x)q(t)dt.$$

Table 2 peresents the numerical results. Moreover, Figure 3 shows the results of the exact and approximate solutions by RK4 method and Figure 4 depicts the results of the present method.

Table 2. Numerical result for Example 6.2.

Value x	Exact solution	Approximate solution by RK4	$n = 128, k = 0$ Present method	$n = 128, k = 3$ Present method
0	0.000000	0.000000	0.003886	0.002918
0.1	0.090000	0.089842	0.088115	0.092796
0.2	0.160000	0.159252	0.159526	0.158348
0.3	0.210000	0.208337	0.210308	0.209526
0.4	0.240000	0.237205	0.240459	0.240074
0.5	0.250000	0.245968	0.249981	0.249992
0.6	0.240000	0.234738	0.240459	0.239281
0.7	0.210000	0.203627	0.210308	0.211082
0.8	0.160000	0.152747	0.159526	0.160697
0.9	0.090000	0.082205	0.088115	0.089682

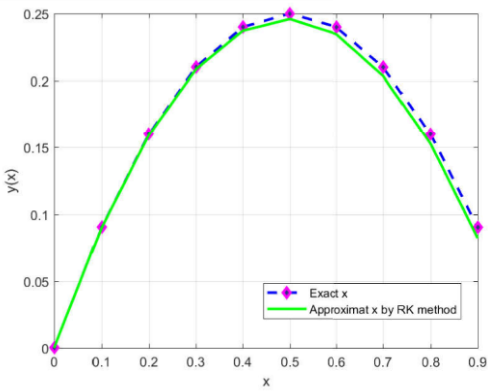


Figure 3. The result solution by RK4

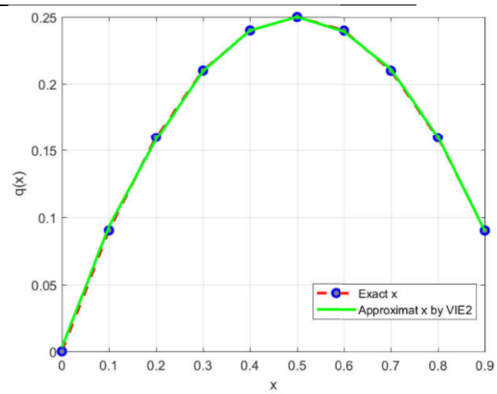


Figure 4. The result solution of the pro-

posed scheme for $n = 128$, $k = 3$ (Example 6.2).

method (Example 6.2).

Example 6.3 This example consists of the following IVP:

$$\begin{cases} y''(x) + y'(x) = e^x \\ y(0) = 0, y'(0) = 1 \end{cases} \tag{18}$$

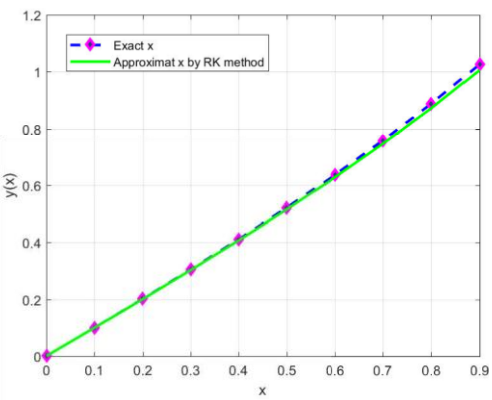
with the exact solution $y(x) = sinh(x)$. Therefore, we converted of Eq. (18) into VIE2 and considered $y''(x) = q(x)$, by integrating the both sides in order to have:

$$q(x) = e^x - 1 + \int_0^x -q(t)dt.$$

Table 3 presents the numerical results. Moreover, Figure 5 shows the results of the exact and approximate solution by RK4 method and Figure 6 depicts the results of the present method.

Table 3. Numerical results for Example 6.3.

Value x	Exact solution	Approximate solution by RK4	$n = 128$, $k = 0$ Present method	$n = 128$, $k = 3$ Present method
0	0.000000	0.000000	0.003901	0.002927
0.1	0.100167	0.100000	0.097807	0.103696
0.2	0.201336	0.200509	0.200535	0.198544
0.3	0.304520	0.302532	0.305334	0.303292
0.4	0.410752	0.407091	0.413285	0.411172
0.5	0.521095	0.515233	0.525502	0.523296
0.6	0.636654	0.628039	0.633875	0.640823
0.7	0.758584	0.746638	0.757602	0.755153
0.8	0.888106	0.872219	0.889150	0.886538
0.9	1.026517	1.006036	1.029877	1.027075



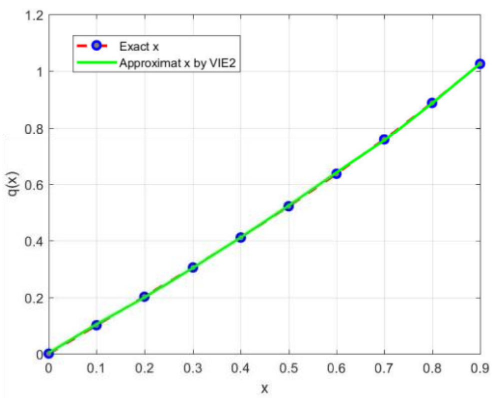


Figure 5. The result solution by RK4 (Exammethod (Example 6.3).

Figure 6. The result solution of the proposed scheme for n = 128, k = 3 ple 6.3).

Example 6.4 Consider the following IVP [14]

$$\begin{cases} y''(x) - 0.1y'(x) = -x \\ y(0) = 0, y'(0) = 1 \end{cases} \tag{19}$$

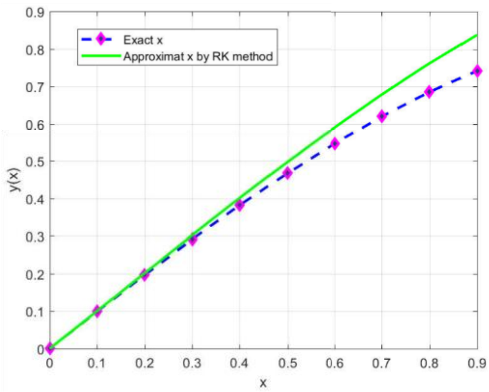
with the exact solution $y(x) = 100x - 5x^2 + 990(e^{-0.1x} - 1)$. Therefore, we converted of Eq. (19) into VIE2 and considered $y''(x) = q(x)$, by integrating the both sides in order to have:

$$q(x) = -\frac{1}{6}x^3 + x + \int_0^x -0.1q(t)dt.$$

Table 4 presented the numerical results. Moreover, Figure 7 depicts the results of the exact and approximate solution by RK4 method and Figure 8 shows the result of the present method.

Table 4. Numerical results for Example 6.4.

Value x	Exact solution	Approximate solution by RK4	n = 128, k = 0 Present method	n = 128, k = 3 Present method
0	0.000000	0.000000	0.003905	0.002929
0.1	0.099335	0.100502	0.097025	0.102796
0.2	0.196687	0.201512	0.195935	0.194058
0.3	0.291078	0.302030	0.291800	0.289991
0.4	0.381545	0.401047	0.383609	0.381888
0.5	0.467130	0.497543	0.470359	0.468745
0.6	0.546888	0.590486	0.545091	0.549573
0.7	0.619882	0.678837	0.619338	0.617982
0.8	0.685183	0.761544	0.685658	0.684463
0.9	0.741873	0.837546	0.743089	0.742074



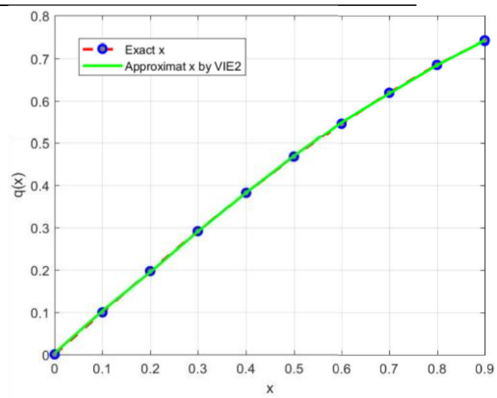


Figure 7. The result solution by RK4 Figure 8. The result solution of the promethod (Example 6.4). posed scheme for n = 128, k = 3 (Example 6.4).

Table 5. Bound of error.

Example	Bound of error $\|f - \bar{f}\|$	
	n = 128 , k = 0	n = 128 , k = 3
1	3.9×10^{-3}	2.9×10^{-3}
2	3.9×10^{-3}	2.9×10^{-3}
3	4.4×10^{-3}	4.2×10^{-3}
4	3.9×10^{-3}	3.5×10^{-3}

7. Conclusion

The present research solved an initial value problem by transformation into Volterra integral equation of the second type. It was found that the numerical solution of these equations using the expansion based on εMBPFs would be better than the numerical solution of the Runge–Kutta of the fourth–order method. Consequently, the results obtained in Tables 1–4 and bound of errors in Table 5 confirmed that method is efficient.

References

[1] A. O. Anidu, S. A. Arekete, A. O. Adedayo and A. O. Adekoya, Dynamic computation of rung– kutta forth order algorithm for first and second order ordinary differential equation using java, International Journal of Computer Science, **12 (3)** (2015) 211–218.

[2] K. Atkinson, W. Han and D. Stewart, Numerical solution of ordinary differential equations, John Wiley and Sons, (2009).

[3] A. A. Bilesanmi, A.Senapon-wusu, A. L. Olutimo, Solution of second–order ordinary differential equations via simulated annealing, Open Journal of Optimization, **8 (1)** (2019) 32–38.

[4] S. O. Edeki, H. I. Okagbue, A. A. Opanuga and S. A. Adeosun, A semi–analytical method for solutions of a certain class of second order ordinary differential equations, Journal of Applied Mathematics, **5** (2014) 2034–2041.

[5] B. O. Fatimah, W. A. Senapon and A. M. Adebowale, Solving ordinary differential equations with evolutionary algorithms, Journal of Optimization, **4 (3)** (2015) 69–73.

[6] A. Hasan, Numerical computation of initial value problem by various techniques, Journal of Science and Art, **1 (42)** (2018) 19–32.

[7] J. Jiange and W. Schaufelberger, Block Pulse Functions and Their Applications in Control Systems, Lecture Notes in Control and Information Sciences, Springer-Verlag Berlin Heidelberg, Berlin, Germany, (1992).

[8] A. Junaid, A. Z. Raja and I. M. Qureshi, Evolutionary computing approach for the solution of initial value problems in ordinary differential equations, World Academic of Science, Engineering and Technology, **31** (2009) 574–577.

[9] K. Maleknejad, M. Khodabin and F. Hosseini Shekarabi, Modified block-pulse functions for numerical solution of stachastic Volterra integral equations, Journal of Applied Mathematics, **2014** (2014), doi: 10.1155/2014/469308.

[10] K. Maleknejad and B. Rahimi, Modification of block–pulse functions and their application to solve numbering Volterra integral of the first kind, Communications in Nonlinear Science and Numerical Simulation, **16 (6)** (2011) 2469–2477.

[11] Z. Masouri, Numerical expansion–iterative method for solving second kind Volterra and fredholm integral equationss using block–pulse functions, Advanced Computational Techniques in Electromagnetics, **2012** (2012), doi: 10.5899/2012/acte-00108.

[12] N. E. Mastorakis, Numerical solution of non–linear ordinary differential equation via collocation method (finite elements) and genetic algorithms, Proceedings of the 6th WSEAS International Conference on Evolutionary Computing, (2005) 36–42.

[13] G. D. Mateescu, On the application of genetic algorithms to differential equations, Romanian Journal of Economic Forecasting, **7 (2)** (2006) 5–9.

[14] M. R. Nadir and A. Rahmoune, Initial value problems between Taylor and Volterra integral equations, MATLAB Journal, **1 (1)** (2018) 1-12.

[15] A. M. Wazwaz, Linear and nonlinear integral equations, Methods and Applications, Springer Berlin Heidelberg, (2011).

[16] N. Yizengaw, Numerical solution of initial value ordinary differential equations using finite difference method, Open Access Library Journal, **2** (2015) 1-7, doi: 10.4236/oalib.1101614.