Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 9071236, 11 pages
https://doi.org/10.1155/2019/9071236

Research Article

A Particle Swarm Optimization-Based Method for Numerically Solving Ordinary Differential Equations

Xian-Ci Zhong, Jia-Ye Chen, and Zhou-Yang Fan

School of Civil Engineering and Architecture, Guangxi University, Nanning, Guangxi 530004, China

Correspondence should be addressed to Xian-Ci Zhong; xczhong@gxu.edu.cn

Received 14 August 2019; Revised 27 September 2019; Accepted 1 October 2019; Published 12 December 2019

Academic Editor: Haopeng Zhang

The Euler method is a typical one for numerically solving initial value problems of ordinary differential equations. Particle swarm optimization (PSO) is an efficient algorithm for obtaining the optimal solution of a nonlinear optimization problem. In this study, a PSO-based Euler-type method is proposed to solve the initial value problem of ordinary differential equations. In the typical Euler method, the equidistant grid points are always used to obtain the approximate solution. The existing shortcoming is that when the iteration number is increasing, the approximate solution could be greatly away from the exact one. Here, it is considered that the distribution of the grid nodes could affect the approximate solution of differential equations on the discrete points. The adopted grid points are assumed to be free and nonequidistant. An optimization problem is constructed and solved by particle swarm optimization (PSO) to determine the distribution of grid points. Through numerical computations, some comparisons are offered to reveal that the proposed method has great advantages and can overcome the existing shortcoming of the typical Euler formulae.

1. Introduction

Due to the complexity of differential equations arising in science and engineering, it is a continuously hot topic to propose various methods for numerically solving the initial and boundary value problems [1–7]. One of the techniques is to transform nonlinear differential equations to linear ordinary differential equations such as the homotopy analysis method [8]. The other of the techniques is to directly discrete differential equations [9]. It is worth noting that many typical methods have been proposed such as the finite difference and element ones [7, 10]. A basic technique of the two typical methods is to mesh the domain of the function governed by differential equations. Then, the approximate solutions on the nodes of the mesh are computed by a feasible approach. One can see that the mesh must be always predefined, which forms the base of the two-type numerical methods. In the present study, a typical initial value problem of the first order ordinary differential equation is considered and represented as follows:

$$\begin{cases} \dfrac{\mathrm{d}u(x)}{\mathrm{d}x} = f(x,u), & x \geq x_0, \\ u(x_0) = u_0, \end{cases} \tag{1}$$

where the function $u(x)$ is to be solved. $f(x,u)$ is known, and u_0 is a constant. In order to numerically solve (1), it is worth noting that the most typical yet simple method is the Euler formula [11]. That is, under a series of nodes $x_0 < x_1 < x_2 < \cdots < x_k < \cdots$, one has the following form:

$$u_{k+1} = u_k + (x_{k+1} - x_k) f(x_k, u_k), \quad k = 0, 1, 2, \ldots, \tag{2}$$

where the term u_k denotes the approximate solution of $u(x_k)$. In general, the nodes are always chosen to be equidistant with $x_k = x_0 + kh$, where h is the step size. Then, formula (2) can be further simplified as

$$u_{k+1} = u_k + hf(x_k, u_k), \quad k = 0, 1, 2, \ldots. \tag{3}$$

It is noted that the Euler method has been extended and used widely [11–14]. One of the shortcomings is that the approximate solution obtained by the Euler method could be deviated greatly away from the exact one with a large iteration number k, unless the step size h is very small.

On the other hand, in many practical situations and the problems of sciences and engineering, various optimization problems have been created [15]. In order to obtain the optimal solutions, a great deal of typical methods have been proposed [15, 16] such as gradient methods, Newton's method, conjugate direction methods, and others. Recently, the heuristic approaches have attracted much attention, such as genetic algorithms [17], particle swarm optimization (PSO) [18, 19], and others. In particular, it is worth noting that the PSO algorithm is efficient in solving many nonlinear optimization problems, and it has been developed widely [20–24] with application to various engineering problems [25–28]. The objective of the present paper is attempted to combine the typical Euler method and the PSO algorithm. Then, the existing shortcoming in the typical Euler method can be overcome. In addition, it is seen that the predefined mesh in the typical numerical methods could be not necessary. Hence, the mesh-free methods have been developed and applied to solve various engineering problems [4]. Inspired by the idea of mesh-free methods, we consider that the grid nodes are free and nonequidistant. When the Euler method is used, it is found that different distributions of the nodes $x_0, x_1, \ldots, x_k$ lead to different values of u_{k+1}. There could exist an optimal distribution such that the approximate solution u_{k+1} has the best accuracy. The remaining issue is how to find the optimal distribution of $x_0, x_1, \ldots, x_k$. Motivated by the above considerations, here an optimization problem is constructed according to the error estimate. By using the particle swarm optimization [18], the constructed optimization problem is solved. Then, the nodes are obtained and used to determine the approximate solutions. Numerical results are carried out to show the advantages of the proposed approach through comparing with the typical Euler method. The observation reveals that a feasible distribution of the nodes $x_0, x_1, \ldots, x_k$ can be indeed used to increase the accuracy of the approximate solution u_{k+1}. The paper is constructed as follows. Section 2 is focused on a brief review on the Euler method and its modified versions. In Section 3, for constructing an optimization problem, two approaches involving differential and integration techniques are provided. Section 4 offers numerical results by solving the optimization problems based on the PSO algorithm. By comparing with some known methods, the novelty and advantages of the proposed method are shown. The main conclusions are covered in Section 5.

2. A Brief Review on the Euler-Type Methods

Since the Euler formula (2) for initial value problems of ordinary differential equations is proposed, many modified ones have been addressed. It is convenient to call the Euler formula and its modified versions as the Euler-type methods. In what follows, let us offer some reviews on the Euler-type methods.

Now, we consider the case of $x \in [x_k, x_{k+1}]$ with the equidistant nodes $x_k = x_0 + kh$ and give the following equality in terms of (1):

$$u(x_{k+1}) = u(x_k) + \int_{x_k}^{x_{k+1}} f(x,u)\mathrm{d}x. \tag{4}$$

When $f(x,u)$ in (4) is replaced by $f(x_k, u_k)$, the Euler formula (3) can be obtained. Similarly, replacing $f(x,u)$ by $f(x_{k+1}, u_{k+1})$, the implicit Euler formula can be determined as

$$u_{k+1} = u_k + hf(x_{k+1}, u_{k+1}), \quad k = 0, 1, 2, \ldots. \tag{5}$$

It is seen that the implicit Euler formula (5) exhibits the same linear convergence as (3) with $h \longrightarrow 0$ [11]. This means that formula (5) has no improvement with respect to (3) in accuracy. When applying the rectangular rule to the integration in (4), the trapezoidal rule is derived as

$$u_{k+1} = u_k + \frac{h}{2}\left[f(x_k, u_k) + f(x_{k+1}, u_{k+1})\right], \quad k = 0, 1, 2, \ldots. \tag{6}$$

Moreover, the application of (3) to (6) leads to the improved Euler formula as follows:

$$u_{k+1} = u_k + \frac{h}{2}\left[f(x_k, u_k) + f(x_{k+1}, u_k + hf(x_k, u_k))\right], \quad k = 0, 1, 2, \ldots. \tag{7}$$

It is found that formulae (6) and (7) have more accuracy than (3) and (5). In general, according to (4), the single-step method can be developed as follows:

$$u_{k+1} = u_k + h\varphi(x_k, u_k; h), \quad k = 0, 1, 2, \ldots, \tag{8}$$

where $\varphi(x_k, u_k; h)$ is a real-valued function related to $f(x,u)$. Hence, the single-step methods with higher order convergence can be constructed, such as the Runge–Kutta methods [11].

Furthermore, when we consider the case of $x \in [x_{k+r-j}, x_{k+r}]$ with $1 \le j \le r$, it follows

$$u(x_{k+r}) = u(x_{k+r-j}) + \int_{x_{k+r-j}}^{x_{k+r}} f(x,u)\mathrm{d}x. \tag{9}$$

Then, the multistep methods can be developed as follows:

$$u_{k+r} + a_{r-1}u_{k+r-1} + \cdots + a_0 u_k = h\varphi(x_k, u_k, \ldots, u_{k+r-1}; h), \tag{10}$$

where φ is a function given by f. Following the idea in (10), some explicit and implicit formulae have been constructed such as the Adams–Bashforth method, the Milne-Thomson method, and others [11]. The above Euler-type methods have been extended to solve fractional initial value problems [12], stochastic differential equations [13, 14], and others.

It is noted from the above-developed methods that they are based on the equidistant nodes to increase the accuracy of the typical Euler formula. Different from the above-mentioned methods, here we consider that the nodes are nonequidistant. By reasonably arranging the positions of the nodes, the accuracy of the typical Euler method can be increased. The proposed method can be further developed to improve the other numerical methods of differential and integral equations [7, 11].

3. The Methods of Constructing Optimization Problems

In this section, we construct several optimization problems to determine the grid points $x_0 < x_1 < x_2 < \cdots < x_n$. Two approaches are introduced in solving the initial value problem of ordinary differential equations (1).

3.1. Differential Method. First, the differential method is used to form an optimization problem. By considering the interval $[x_k, x_{k+1}]$ and the differential mean value theorem [29], there is at least one $\xi_k \in [x_k, x_{k+1}]$ such that

$$\frac{u(x_{k+1}) - u(x_k)}{x_{k+1} - x_k} = \left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_{x=\xi_k} = f(\xi_k, u(\xi_k)), \quad k = 0, 1, 2, \ldots. \tag{11}$$

It is seen that when the value of ξ_k in (11) is approximated by using x_k, the Euler method (2) is given. When the value of ξ_k in (11) is replaced by x_{k+1}, the implicit Euler method (5) is formed. On the other hand, if the value of ξ_k can be determined explicitly, the application of (11) can lead to a perfect formula as follows:

$$u_{k+1} = u_k + (x_{k+1} - x_k) f(\xi_k, u(\xi_k)). \tag{12}$$

Unfortunately, it is always difficult to give an explicit value of ξ_k and $u(\xi_k)$, which just forms the reason behind the Euler-type methods. On the other hand, we can change a way of thinking to construct an optimization problem as follows:

$$\min\left|u(x_{k+1}) - u_{k+1}\right|. \tag{13}$$

In other words, if the unknown ξ_k is replaced by a constant $c_k \in [x_k, x_{k+1}]$, the optimization problem (13) can be rewritten as

$$\min\left|u(x_k) - u_k + (x_{k+1} - x_k)\left[f(\xi_k, u(\xi_k)) - f(c_k, u(c_k))\right]\right|. \tag{14}$$

In what follows, let us analyze the optimization problem (14). It is convenient to assume $u(x_k) = u_k$, meaning that the local truncation error is considered. Defining a new function

$$Q(x_k, x_{k+1}, \xi_k, c_k) = \left|(x_{k+1} - x_k)\left[f(\xi_k, u(\xi_k)) - f(c_k, u(c_k))\right]\right|, \tag{15}$$

we have the following result.

Theorem 1. *Suppose that $D \subset \mathbf{R}^2$ is a domain with $f(x,u) : D \longrightarrow \mathbf{R}$. If $f(x,u)$ is continuously differentiated on D, the following estimate holds:*

$$Q(x_k, x_{k+1}, \xi_k, c_k) \le (x_{k+1} - x_k)^2 (M_k + N_k \cdot L_k), \tag{16}$$

where

$$\begin{aligned} M_k &= \max_{x\in[x_k,x_{k+1}]}\left|\frac{\partial f}{\partial x}\right|, \\ L_k &= \max_{x\in[x_k,x_{k+1}]}|f|, \\ N_k &= \max_{x\in[x_k,x_{k+1}],(x,u)\in D}\left|\frac{\partial f}{\partial u}\right|. \end{aligned} \tag{17}$$

Proof. According to (16), it follows

$$\begin{aligned} \left|f(\xi_k, u(\xi_k)) - f(c_k, u(c_k))\right| &= \left|f(\xi_k, u(\xi_k)) - f(c_k, u(\xi_k)) \right. \\ &\quad \left. + f(c_k, u(\xi_k)) - f(c_k, u(c_k))\right| \\ &= \left|f(\xi_k, u(\xi_k)) - f(c_k, u(\xi_k)) \right. \\ &\quad \left. + f(c_k, u(\xi_k)) - f(c_k, u(c_k))\right| \\ &= \left|\frac{\partial f(\overline{\xi}_k, u(\xi_k))}{\partial x}\right|\left|\xi_k - c_k\right| \\ &\quad + \left|\frac{\partial f(c_k, \eta)}{\partial u}\right|\left|u(\xi_k) - u(c_k)\right| \\ &= \left|\frac{\partial f(\overline{\xi}_k, u(\xi_k))}{\partial x}\right|\left|\xi_k - c_k\right| \\ &\quad + \left|\frac{\partial f(c_k, \eta)}{\partial u}\right| \cdot \left|f(\eta_k, u(\eta_k))\right| \\ &\quad \cdot \left|\xi_k - c_k\right|, \end{aligned} \tag{18}$$

where $\overline{\xi}_k$ and η_k are located between ξ_k and c_k; η lies between $u(\xi_k)$ and $u(c_k)$. Then, defining

$$\begin{aligned} M_k &= \max_{x\in[x_k,x_{k+1}]}\left|\frac{\partial f}{\partial x}\right|, \\ L_k &= \max_{x\in[x_k,x_{k+1}]}|f|, \\ N_k &= \max_{x\in[x_k,x_{k+1}],(x,u)\in D}\left|\frac{\partial f}{\partial u}\right|, \end{aligned} \tag{19}$$

we have the following relation:

$$\left|f(\xi_k, u(\xi_k)) - f(c_k, u(c_k))\right| \le \left|\xi_k - c_k\right|(M_k + N_k \cdot L_k). \tag{20}$$

Under the consideration of $\xi_k, c_k \in [x_k, x_{k+1}]$, the following result is determined:

$$Q(x_k, x_{k+1}, \xi_k, c_k) \le (x_{k+1} - x_k)^2 (M_k + N_k \cdot L_k). \quad (21)$$

The proof is completed. □

As shown in Theorem 1, it is found that the values of M_k, N_k, and L_k are dependent on the interval $[x_k, x_{k+1}]$. That is, a new function is constructed as follows:

$$\overline{Q}_k(x_k, x_{k+1}) = (x_{k+1} - x_k)^2 (M_k + N_k \cdot L_k). \quad (22)$$

In addition, considering the error estimate of the approximate solution u_{k+1}, we give the following result.

Theorem 2. *Assume that the function $f(x,u)$ is defined on $D \subset \mathbf{R}^2$ with $f(x,u) : D \longrightarrow \mathbf{R}$. If $f(x,u)$ is continuously differentiated on D, the error estimate is obtained as follows:*

$$\left|u(x_{k+1}) - u_{k+1}\right| \le \sum_{j=0}^{k} \overline{Q}_j\left(x_j, x_{j+1}\right). \quad (23)$$

Proof. According to (14) and Theorem 1, we have

$$\begin{aligned}
\left|u(x_{k+1}) - u_{k+1}\right| &= \left|u(x_k) - u_k + (x_{k+1} - x_k)\right.\\
&\quad \left.\cdot \left[f(\xi_k, u(\xi_k)) - f(c_k, u(c_k))\right]\right|\\
&\le \left|u(x_k) - u_k\right| + \overline{Q}_k(x_k, x_{k+1})\\
&\le \left|u(x_{k-1}) - u_{k-1}\right| + \overline{Q}_k(x_k, x_{k+1})\\
&\quad + \overline{Q}_{k-1}(x_{k-1}, x_k)\\
&\cdots\cdots\cdots\cdots\\
&\le \left|u(x_0) - u_0\right| + \overline{Q}_k(x_k, x_{k+1})\\
&\quad + \overline{Q}_{k-1}(x_{k-1}, x_k) + \cdots + \overline{Q}_0(x_0, x_1)\\
&= \sum_{j=0}^{k} \overline{Q}_j\left(x_j, x_{j+1}\right).
\end{aligned} \quad (24)$$

The proof is completed. □

Making use of Theorem 2, we can form a new optimization problem as follows:

$$\min_{x_0 < x_1 < \cdots < x_{k+1}} \sum_{j=0}^{k} \overline{Q}_j\left(x_j, x_{j+1}\right). \quad (25)$$

Once the optimization problem (25) is solved, the nodes $x_j\,(j = 1, 2, \ldots, k+1)$ can be determined, and they are further used to determine u_j through an Euler-type method such as one of (2)–(7). Hence, the approximate solution of $u(x)\,(x_0 \le x \le x_{k+1})$ is given as

$$u_a(x) = \sum_{j=0}^{k+1} u_j l_j(x), \quad (26)$$

where

$$\begin{aligned}
l_0(x) &= \begin{cases} \dfrac{x - x_1}{x_0 - x_1}, & x \in [x_0, x_1], \\ 0, & x \in [x_1, x_{k+1}], \end{cases}\\
l_j(x) &= \begin{cases} \dfrac{x - x_{j-1}}{x_j - x_{j-1}}, & x \in \left[x_{j-1}, x_j\right], \\ \dfrac{x - x_{j+1}}{x_j - x_{j+1}}, & x \in \left[x_j, x_{j+1}\right], \\ 0, & x \notin \left[x_{j-1}, x_{j+1}\right], \end{cases}\\
l_{k+1}(x) &= \begin{cases} \dfrac{x - x_k}{x_{k+1} - x_k}, & x \in [x_k, x_{k+1}], \\ 0, & x \in [x_0, x_k]. \end{cases}
\end{aligned} \quad (27)$$

Of much interest is that the approximate solution u_{k+1} could have more accuracy than the result obtained by the typical Euler method with the equidistant grid points. This means that when one wants to obtain an approximate solution u_{k+1} with more accuracy, the optimization problem (25) can be solved to determine the distribution of grid points. The formats of the Euler-type methods (2)–(7) or their combinations can be used to give the approximate solution u_{k+1}. As compared with the typical Euler-type methods [11, 30], the main novelty of the present study is to assume that the grid points are free, and they are determined by constructing and solving an optimization problem.

3.2. Integration Method. Now, the integration method is introduced to construct an optimization problem. According to the initial value problem (1), we integrate both sides of the differential equation with respect to x from x_0 to x_{k+1}. The following Volterra integral equation is obtained:

$$u(x_{k+1}) = u_0 + \int_{x_0}^{x_{k+1}} f(x,u)\mathrm{d}x. \quad (28)$$

By considering all the grid points $x_0 < x_1 < x_2 < \cdots < x_{k+1}$, integral equation (28) is rewritten as

$$u(x_{k+1}) = u_0 + \sum_{j=0}^{k} \int_{x_j}^{x_{j+1}} f(x,u)\mathrm{d}x. \quad (29)$$

Under the assumption of a continuous function $f(x,u)$ on a domain D, there is at least one $\xi_j \in [x_j, x_{j+1}]$ such that

$$u(x_{k+1}) = u_0 + \sum_{j=0}^{k} \left(x_{j+1} - x_j\right) f\left(\xi_j, u\left(\xi_j\right)\right), \quad (30)$$

where the integral mean value theorem has been applied [29]. When the values of ξ_j and $u(\xi_j)$ can be given explicitly, formula (30) is also explicit. However, it is always difficult to give the explicit values of ξ_j and $u(\xi_j)$. Then, it is assumed that ξ_j and $u(\xi_j)$ are replaced by c_j and $u(c_j)$, respectively. Hence, the optimization problem is constructed as follows:

$$\begin{aligned}\min\left|u(x_{k+1})-u_{k+1}\right| &= \min_{c_j\in[x_j,x_{j+1}]}\left|\sum_{j=0}^{k}(x_{j+1}-x_j)\left[f(\xi_j,u(\xi_j))-f(c_j,u(c_j))\right]\right|\\ &= \min_{c_j\in[x_j,x_{j+1}]}\left|\sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}f(x,u)\mathrm{d}x-\sum_{j=0}^{k}(x_{j+1}-x_j)f(c_j,u(c_j))\right|\\ &= \min_{c_j\in[x_j,x_{j+1}]}\left|\sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}f(x,u)\mathrm{d}x-\sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}f(c_j,u(c_j))\mathrm{d}x\right|\\ &= \min_{c_j\in[x_j,x_{j+1}]}\left|\sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left[f(x,u)-f(c_j,u(c_j))\right]\mathrm{d}x\right|.\end{aligned}\tag{31}$$

Moreover, we introduce the Lipschitz conditions as follows:

$$\begin{aligned}\left|f(x,u)-f(c_j,u)\right| &\le \overline{L}_j\left|x-c_j\right|, \quad x,c_j\in[x_j,x_{j+1}],\ \forall u,\\ \left|f(x,u(x))-f(x,u(c_j))\right| &\le \widetilde{L}_j\left|u(x)-u(c_j)\right|, \quad \forall x\in[x_j,x_{j+1}],\end{aligned}\tag{32}$$

where $\overline{L}_j$ and $\widetilde{L}_j$ are positive Lipschitz constants. It gives

$$\begin{aligned}\left|\sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left[f(x,u)-f(c_j,u(c_j))\right]\mathrm{d}x\right| &= \left|\sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left[f(x,u)-f(c_j,u(x))+f(c_j,u(x))-f(c_j,u(c_j))\right]\mathrm{d}x\right|\\ &\le \sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left(\left|f(x,u)-f(c_j,u(x))\right|+\left|f(c_j,u(x))-f(c_j,u(c_j))\right|\right)\mathrm{d}x\\ &= \sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left(\overline{L}_j\left|x-c_j\right|+\widetilde{L}_j\left|u(x)-u(c_j)\right|\right)\mathrm{d}x.\end{aligned}\tag{33}$$

In addition, applying the differential mean value theorem, there is at least one η_j such that

$$\left|u(x)-u(c_j)\right| = \left|f(\eta_j,u(\eta_j))\right|\cdot\left|x-c_j\right|,\tag{34}$$

where η_j is located between x and c_j. By considering

$$L_j = \max_{x\in[x_j,x_{j+1}]}|f|,\tag{35}$$

we have

$$\begin{aligned}&\sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left(\overline{L}_j\left|x-c_j\right|+\widetilde{L}_j\left|u(x)-u(c_j)\right|\right)\mathrm{d}x\\ &\quad= \sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left(\overline{L}_j\left|x-c_j\right|+\widetilde{L}_j\left|f(\eta_j,u(\eta_j))\right|\cdot\left|x-c_j\right|\right)\mathrm{d}x\\ &\quad\le \sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left(\overline{L}_j+L_j\cdot\widetilde{L}_j\right)\left|x-c_j\right|\mathrm{d}x.\end{aligned}\tag{36}$$

Applying the Cauchy–Schwarz inequality to (34) together with (33), the following relation is given:

$$\begin{aligned}&\left|\sum_{j=0}^{k}\int_{x_j}^{x_{j+1}}\left[f(x,u)-f(c_j,u(c_j))\right]\mathrm{d}x\right|\\ &\quad\le \sum_{j=0}^{k}\left(\overline{L}_j+L_j\cdot\widetilde{L}_j\right)\int_{x_j}^{x_{j+1}}\left|x-c_j\right|\mathrm{d}x\\ &\quad\le \sum_{j=0}^{k}\left(\overline{L}_j+L_j\cdot\widetilde{L}_j\right)\left[\int_{x_j}^{x_{j+1}}1^2\mathrm{d}x\cdot\int_{x_j}^{x_{j+1}}\left|x-c_j\right|^2\mathrm{d}x\right]^{1/2}\\ &\quad= \sum_{j=0}^{k}\left(\overline{L}_j+L_j\cdot\widetilde{L}_j\right)\left\{\frac{x_{j+1}-x_j}{3}\left[(x_{j+1}-c_j)^3+(x_j-c_j)^3\right]\right\}^{1/2}.\end{aligned}\tag{37}$$

Then, the optimization problem (31) is transformed to the following form:

$$\min_{c_j\in[x_j,x_{j+1}]} \sum_{j=0}^{k}\left(\overline{L}_j + L_j \cdot \widetilde{L}_j\right)\left\{\frac{x_{j+1}-x_j}{3}\left[\left(x_{j+1}-c_j\right)^3 + \left(x_j - c_j\right)^3\right]\right\}^{1/2}. \tag{38}$$

It is easy to give the optimal solution of (38) as

$$\Phi = \frac{\sqrt{3}}{6}\sum_{j=0}^{k}\left(\overline{L}_j + L_j \cdot \widetilde{L}_j\right)\left(x_{j+1}-x_j\right)^2, \tag{39}$$

with $c_j = (x_j + x_{j+1})/2$. This means that when the optimization problem (38) is solved, the approximate solution of $u(x_{k+1})$ can be determined as follows:

$$u_{k+1} = u_0 + \sum_{j=0}^{k}\left(x_{j+1}-x_j\right) f\left(\frac{x_j + x_{j+1}}{2}, u\left(\frac{x_j + x_{j+1}}{2}\right)\right). \tag{40}$$

In other words, midpoint formula (40) has been retrieved by solving an optimization problem.

Furthermore, of much important is that here the distribution of the grid points is considered to be free. Hence, a new optimization problem is constructed as follows:

$$F = \min_{x_0<x_1<\cdots<x_{k+1}} \Phi. \tag{41}$$

Once optimization problem (41) is solved, the distribution of the grid point is determined. Formulae (2)–(7) and (40) can be used to obtain the approximation solution u_{k+1}. It is hopeful to give an approximate solution with more accuracy than that by using the equidistant grid points. Moreover, it is seen from (40) that there is an unknown term $u(x_j + x_{j+1}/2)$. For the sake of simplicity, the Euler formula is applied to give

$$u\left(\frac{x_j + x_{j+1}}{2}\right) \approx u_k + \frac{x_{j+1}-x_j}{2} f\left(x_k, u_k\right). \tag{42}$$

Formula (40) is rewritten as

$$\begin{aligned} u_{k+1} = u_0 + \sum_{j=0}^{k}\left(x_{j+1}-x_j\right) \\ \cdot f\left(\frac{x_j + x_{j+1}}{2}, u_j + \frac{x_{j+1}-x_j}{2} f\left(x_j, u_j\right)\right). \end{aligned} \tag{43}$$

As compared with the differential method, the obtained functions (25) and (39) are similar, when the Lipschitz constants $\overline{L}_j$ and $\widetilde{L}_j$ are replaced by M_k and N_k, respectively.

According to the above analysis, the two optimization problems (25) and (41) are constructed to determine the nonequivalent grids $x_0 < x_1 < \cdots < x_k$ such that the approximate solution u_{k+1} to $u(x_{k+1})$ has more accuracy than that by using the equivalent grids. Furthermore, when we consider the grids $a = x_0 < x_1 < \cdots < x_n = b$ and the error estimate on all the grids x_k $(k = 1, 2, \ldots, n)$, the optimization problem (13) can be reread as

$$\min \sum_{k=0}^{n-1}\left|u\left(x_{k+1}\right) - u_{k+1}\right|. \tag{44}$$

In terms of (44) and (25), the derived optimization problem is obtained as

$$\min_{x_0<x_1<\cdots<x_n} \sum_{k=0}^{n-1}\sum_{j=0}^{k} \overline{Q}_j\left(x_j, x_{j+1}\right). \tag{45}$$

In the following computations, the optimization problems (25) and (45) are all used to determine the grid points. Some comparisons are offered to illustrate the advantages of the proposed methods in giving a good approximation solution.

4. Numerical Results and Discussion

In order to show the effectiveness of the proposed method, some numerical results are reported in this section.

4.1. The Initial Value Problems of the First Order Ordinary Differential Equations

Example 1. Consider the following initial value problem of the first order differential equation [30]:

$$\begin{cases} \dfrac{du(x)}{dx} = x^3 - \dfrac{u}{x}, & x > 1, \\ u(1) = \dfrac{2}{5}, \end{cases} \tag{46}$$

with the exact solution

$$u(x) = \frac{1}{5}x^4 + \frac{1}{5x}. \tag{47}$$

First, let us construct an optimization problem according to (25). In terms of (46), it gives

$$f(x,u) = x^3 - \frac{u}{x}, \tag{48}$$

meaning that

$$\begin{aligned} \left|\frac{\partial f}{\partial x}\right| &= \left|3x^2 + \frac{u}{x^2}\right|, \\ \left|\frac{\partial f}{\partial u}\right| &= \frac{1}{|x|}. \end{aligned} \tag{49}$$

We further have the following estimates:

$$\begin{aligned} M_k &= \max_{x\in[x_k,x_{k+1}]}\left|3x^2 + \frac{u}{x^2}\right| \le 3x_{k+1}^2 + \frac{1}{x_k^2}\cdot \max_{x\in[x_k,x_{k+1}]}|u|, \\ N_k &= \max_{x\in[x_k,x_{k+1}]} \frac{1}{|x|} = \frac{1}{x_k}, \\ L_k &= \max_{x\in[x_k,x_{k+1}]}\left|x^3 - \frac{u}{x}\right| \le x_{k+1}^3 + \frac{1}{x_k}\max_{x\in[x_k,x_{k+1}]}|u|. \end{aligned} \tag{50}$$

Then, it follows

$$M_k + N_k \cdot L_k \le 3x_{k+1}^2 + \frac{x_{k+1}^3}{x_k} + \frac{2}{x_k^2}\max_{x\in[x_k,x_{k+1}]}|u|. \tag{51}$$

One can find from (51) that the value of $\max_{x\in[x_k,x_{k+1}]}|u|$ is unknown. Hence, the main term of (51) related to the nodes is extracted to yield the following function:

$$F=\sum_{j=0}^{k}\left(x_{j+1}-x_j\right)^2\left(3x_{j+1}^2+\frac{x_{j+1}^3}{x_j}+\frac{2}{x_j^2}\right). \tag{52}$$

The optimization problem

$$\min_{x_0<x_1<\cdots<x_{k+1}} F \tag{53}$$

is used to determine the values of the nodes. Hence, the application of (43) leads to the computation formula as follows:

$$\begin{aligned}u_{k+1}=u_0+\sum_{j=0}^{k}\left(x_{j+1}-x_j\right)\Bigg[&\left(\frac{x_j+x_{j+1}}{2}\right)^3\\&-\frac{2}{x_{j+1}+x_j}\left(\frac{3x_j-x_{j+1}}{2x_j}u_j+\frac{x_{j+1}-x_j}{2}x_j^3\right)\Bigg].\end{aligned} \tag{54}$$

Second, let us focus on the optimal solution of (53). The particle swarm optimization developed by Kennedy and Eberhart [18, 19] is utilized to achieve the objective. For convenience, the formulae of updating the position and velocity of a particle are given as follows [31]:

$$\begin{cases}\vec{v}_{i+1}=\omega\vec{v}_i+\vec{U}\left(0,\phi_1\right)\otimes\left(\vec{p}_i-\vec{x}_i\right)\\\quad+\vec{U}\left(0,\phi_2\right)\otimes\left(\vec{p}_g-\vec{x}_i\right),\\\vec{x}_{i+1}=\vec{x}_i+\vec{v}_i.\end{cases} \tag{55}$$

The meanings of the terms in the above formulae are explained in Table 1.

Moreover, for numerical computations, it is assumed that ω is monotonically decreasing from 0.9 to 0.4 with respect to the generation number and $\phi_i=2$ $(i=1,2)$. When applying the PSO algorithm to the optimization problem (53), a particle is considered as the following $(k+2)$-dimensional vector:

$$\vec{x}_i=\left(x_0,x_1,\ldots,x_{k+1}\right), \tag{56}$$

subject to the constraint condition:

$$1=x_0<x_1<\cdots<x_k<x_{k+1}=c, \tag{57}$$

where c is a prescribed constant. The size of particle swarm is 1000, and the maximum of the generation is 200. The symbols $\vec{p}_i$ and $\vec{p}_g$ stand for the individual best position of a particle and the global best position of 1000 particles, respectively.

Figure 1 is drawn to show the variation of the values of F versus the generation number with $k=4$ and $x_{k+1}=2$. It is seen from Figure 1 that the values of F decrease rapidly and then tend to a constant with the increasing generation number. This means that a stable and optimal solution has been obtained when the generation number is 200. The computed nonequidistant grid points are given as follows:

$$(1.0000, 1.2469, 1.4630, 1.6573, 1.8352, 2.0000). \tag{58}$$

The approximate solution using the nonequidistant grid points and formula (54) is obtained as $u_5=3.3046$. When the equidistant grid points and (2) are used, we obtain $\tilde{u}_5=2.8407$. The exact solution is $u(2)=3.3$. Clearly, the solution $u_5=3.3046$ is more approximate to $u(2)=3.3$ than $\tilde{u}_5=2.8407$. The observation reveals that the proposed Euler-type method with optimization is effective to obtain an approximate solution with more accuracy. In addition, under the condition $k=4$, we also choose $x_{k+1}=1.5, 2.5, 3.0, 3.5$ for computations, and the observations are shown in Table 2. It is found from Table 2 that the approximate solutions of $u(x_5)$ determined by the Euler method are away from the exact one when the values of x_5 are increasing. This is also attributed to the fact that the number of the chosen grid points is only 6. Fortunately, the proposed method is still effective regardless of the increasing values of x_5. In other words, it is interesting to find that the obtained results by using the proposed method are in good approximation to the exact ones. This means that the shortcoming of the typical Euler method can be overcome by using the developed method in the present study.

In what follows, we construct the other optimization problem in terms of (45) as follows:

$$\min_{x_0<x_1<\cdots<x_n} F_g, \tag{59}$$

with

$$F_g=\sum_{k=0}^{n-1}\sum_{j=0}^{k}\left(x_{j+1}-x_j\right)^2\left(3x_{j+1}^2+\frac{x_{j+1}^3}{x_j}+\frac{2}{x_j^2}\right). \tag{60}$$

For the sake of comparisons, we choose $n=5$ and $x_5=2.0$ for numerical computations. After running the PSO algorithm, the optimal solution is obtained, and the nonequivalent grids are given as

$$(1.0000, 1.1365, 1.2849, 1.4555, 1.6703, 2.0000). \tag{61}$$

The obtained approximation solution for $u(2)$ is $u_5=3.3068$. Expectedly, it has more errors than the value $u_5=3.3046$ in Table 2 by solving the optimization problem (53). For convenience, Figure 2 is drawn to show the variations of the absolute errors $|u(x)-u_{a1}|$ and $|u(x)-u_{a2}|$ versus the discrete grids, where u_{a1} and u_{a2} stand for the approximate solutions by solving (53) and (54), respectively. It is seen from Figure 2 that the absolute errors are increasing with the increasing values of the discrete points. The optimization problem (53) focuses on the minimization of the error on the point x_n. The aim of the optimization problem (54) is the minimization of the global error on all discrete points.

Example 2. In engineering applications, the widely studied Riccati differential equation is expressed as the following form [32, 33]:

$$\frac{\mathrm{d}u(x)}{\mathrm{d}x}=P(x)u+Q(x)u^2+R(x). \tag{62}$$

Table 1: The meanings of the terms in the formulae of particle swarm optimization.

$\vec{x}_i$	The current position of particle
$\vec{v}_i$	The current velocity of particle
$\vec{p}_i$	The individual best position of particle
$\vec{p}_g$	The global best position of all particles
$\vec{U}(0,\phi_i)\,(i=1,2)$	The vectors of random numbers uniformly distributed in $[0,\phi_i]$
ω	The inertia weight

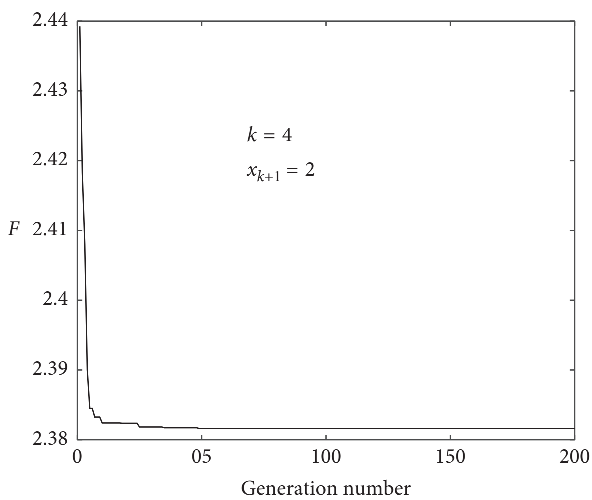

Figure 1: The variations of the values of F versus the generation number for $k = 4$ and $x_{k+1} = 2$.

Table 2: Comparisons with the typical Euler method.

	The approximate solution u_5		
x_5	Present study	Euler method	Exact solution
1.5	1.1465	1.0597	1.1458
2.0	3.3046	2.8407	3.3
2.5	7.9066	6.5183	7.8925
3.0	16.2985	13.0865	16.2667
3.5	30.1304	23.7542	30.0696

Here, we assume that it is subject to the initial value condition $u(0) = 0$ with $P(x) = 2$, $Q(x) = -1$, and $R(x) = 1$. Then, the explicit solution is given as [34, 35]

$$u(x) = 1 + \sqrt{2}\tanh\left[\sqrt{2}\,x + \ln\left(\sqrt{2}-1\right)\right]. \tag{63}$$

Under the consideration of (46), we have

$$f(x,u) = 2u - u^2 + 1,$$
$$\frac{\partial f}{\partial x} = 0, \tag{64}$$
$$\frac{\partial f}{\partial u} = 2 - 2u.$$

In virtue of (25), an optimization problem is constructed as follows:

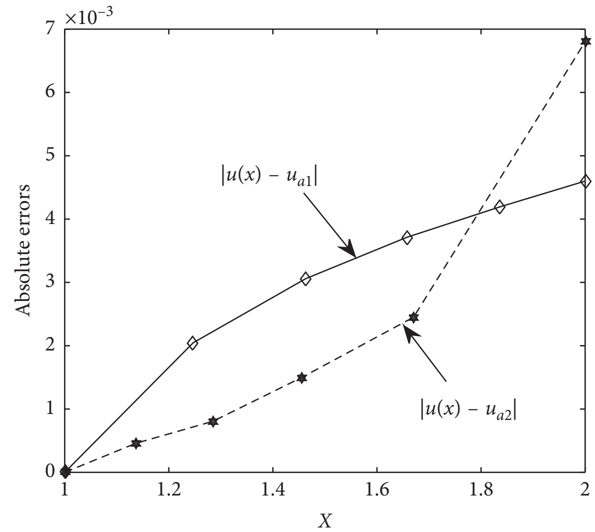

Figure 2: The variations of the absolute errors versus the discrete points.

$$H = \min_{x_0<x_1<\cdots<x_{k+1}} \sum_{j=0}^{k} \left(x_{j+1} - x_j\right)^2. \tag{65}$$

For numerical computations, we choose $x_{20} = 2$ with $k = 19$. By running the PSO algorithm, the optimization problem (65) can be solved, and the grid points are determined. Then, by using (43), the approximate solutions u_i $(i = 1, 2, \ldots, 20)$ are obtained as

$$u_{k+1} = u_0 + \sum_{j=0}^{k}\left(x_{j+1}-x_j\right)\left\{1 + 2\left[u_j + \frac{x_{j+1}-x_j}{2}\left(1+2u_j-u_j^2\right)\right] - \left[u_j + \frac{x_{j+1}-x_j}{2}\left(1+2u_j-u_j^2\right)\right]^2\right\}. \tag{66}$$

In order to show the advantages of the proposed approach, the approximate solutions in [34, 35] are recalled and given in the following:

$$u_e(x) = x + x^2 + \frac{1}{3}x^3 - \frac{2}{3}x^4 + \frac{2}{15}x^5,$$
$$u_t(x) = \frac{10x\left(x^3 - 9x^2 + 18x - 60\right)}{2x^5 - 31x^4 + 200x^3 - 630x^2 + 720x - 600}. \tag{67}$$

As shown in Table 3, the absolute errors $|u(x_k) - u_k|$, $|u(x_k) - u_e(x_k)|$, and $|u(x_k) - u_t(x_k)|$ are computed by virtue of different approaches. It is found from Table 3 that the better approximation solutions have been given by using the proposed method as compared to the known ones [34, 35]. In particular, the obtained results are stable, and the absolute errors are controlled in an acceptable domain.

4.2. The Initial Value Problem of Higher Order Ordinary Differential Equations. It is obvious that the developed

TABLE 3: The optimized points and the absolute errors.

x_k	$u(x_k)$	u_k	$\lvert u(x_k) - u_k\rvert$	$\lvert u(x_k) - u_e(x_k)\rvert$	$\lvert u(x_k) - u_t(x_k)\rvert$
	The present results			[34]	[35]
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.1149	0.1285	0.1277	0.0008	0.0000	0.0016
0.2316	0.2881	0.2865	0.0017	0.0005	0.0075
0.3408	0.4634	0.4612	0.0022	0.0016	0.0185
0.4552	0.6701	0.6677	0.0024	0.0023	0.0370
0.5660	0.8859	0.8837	0.0022	0.0003	0.0627
0.6790	1.1114	1.1097	0.0017	0.0106	0.0963
0.7910	1.3294	1.3282	0.0012	0.0326	0.1354
0.8989	1.5249	1.5239	0.0010	0.0670	0.1761
1.0072	1.7004	1.6993	0.0011	0.1140	0.2173
1.1091	1.8431	1.8417	0.0014	0.1659	0.2536
1.2136	1.9660	1.9643	0.0018	0.2210	0.2866
1.3113	2.0607	2.0587	0.0020	0.2675	0.3122
1.4064	2.1359	2.1338	0.0021	0.3012	0.3316
1.4967	2.1938	2.1916	0.0022	0.3167	0.3446
1.5820	2.2381	2.2360	0.0021	0.3119	0.3521
1.6680	2.2742	2.2723	0.0020	0.2840	0.3551
1.7498	2.3020	2.3002	0.0018	0.2329	0.3537
1.8369	2.3257	2.3241	0.0016	0.1497	0.3481
1.9208	2.3439	2.3425	0.0015	0.0400	0.3387
2.0000	2.3578	2.3565	0.0013	0.0911	0.3265

method in the previous section can be used to numerically solve the initial value problem of the higher order differential equations. For example, as shown in [36], we consider a damped oscillator circuit for reflecting the behaviour of a stiffness system.

Example 3. The initial value problem of the second order ordinary differential equation is expressed as follows:

$$\begin{cases} L\dfrac{\mathrm{d}^2 I}{\mathrm{d}t^2} + R\dfrac{\mathrm{d}I}{\mathrm{d}t} + \dfrac{I}{C} = 0, \quad t \geq 0, \\ I(0) = 0, \\ \dfrac{\mathrm{d}I(0)}{\mathrm{d}t} = 10, \end{cases} \tag{68}$$

where I is the electric current; L, R, and C are constants with $R/L = 20$ and $LC = 100$, respectively. The explicit solution can be computed as

$$I(t) = \frac{250\sqrt{1111}}{3333}\left(e^{(-(100+3\sqrt{1111})/50)t} - e^{(-(100-3\sqrt{1111})/50)t}\right). \tag{69}$$

In order to apply the proposed method, it is assumed that

$$\begin{aligned} u(t) &= I(t), \\ \frac{\mathrm{d}I(t)}{\mathrm{d}t} &= v(t). \end{aligned} \tag{70}$$

The initial problem (68) is rewritten as the system of the first order differential equations:

$$\begin{cases} \dfrac{\mathrm{d}u(t)}{\mathrm{d}t} = v(t), \\ \dfrac{\mathrm{d}v(t)}{\mathrm{d}t} = -\dfrac{u(t)}{LC} - \dfrac{R}{L}v(t), \end{cases} \tag{71}$$

subject to the initial value conditions:

$$\begin{aligned} u(0) &= 0, \\ v(0) &= 10. \end{aligned} \tag{72}$$

In addition, we can also reexpress (71) as the style of the vector. That is, it is supposed that

$$\begin{aligned} \mathbf{U}(t) &= \begin{bmatrix} u(t) \\ v(t) \end{bmatrix}, \\ \frac{\mathrm{d}\mathbf{U}(t)}{\mathrm{d}t} &= \begin{bmatrix} u'(t) \\ v'(t) \end{bmatrix}. \end{aligned} \tag{73}$$

Then, one has

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{U}(t)}{\mathrm{d}t} = A\mathbf{U}(t), \quad t \geq 0, \\ \mathbf{U}(0) = \mathbf{U}_0, \end{cases} \tag{74}$$

where

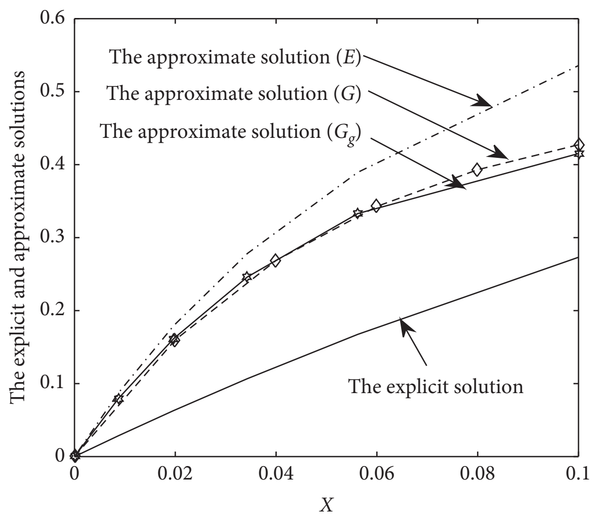

Figure 3: The variations of the explicit and the approximate solutions. The approximate solutions with the marks E, G, and G_g mean that they are obtained by using the typical Euler method, the proposed method under solving the optimization problem G and that with solving the optimization problem G_g, respectively.

$$A = \begin{bmatrix} 0 & 1 \\ -\left(\frac{1}{LC}\right) & -\left(\frac{R}{L}\right) \end{bmatrix},$$
$$\mathbf{U}_0 = \begin{bmatrix} 0 \\ 10 \end{bmatrix}. \tag{75}$$

For a series of grid points $0 = t_0 < t_1 < t_2 < \cdots < t_{k+1}$, the application of (43) leads to

$$\mathbf{U}_{k+1} = \mathbf{U}_0 + \sum_{j=0}^{k}\left(t_{j+1} - t_j\right)A\left(\mathbf{U}_j + \frac{t_{j+1} - t_j}{2}A\mathbf{U}_j\right), \tag{76}$$

where $\mathbf{U}_j$ denotes the approximate solution of $\mathbf{U}(t_j)$. When the optimal solution $\mathbf{U}_{k+1}$ is focused on, the grid points t_j $(j = 0, 1, \ldots, k+1)$ are determined by solving the following optimization problem:

$$G = \min_{t_0<t_1<\cdots<t_{k+1}} \sum_{j=0}^{k}\left(t_{j+1} - t_j\right)^2. \tag{77}$$

When considering the global optimal solution, the optimization problem for determining the grid points t_j $(j = 0, 1, \ldots, n)$ is constructed as follows:

$$G_g = \min_{t_0<t_1<\cdots<t_n} \sum_{k=0}^{n-1}\sum_{j=0}^{k}\left(t_{j+1} - t_j\right)^2. \tag{78}$$

In what follows, the grid points $0 = t_0 < t_1 < t_2 < \cdots < t_5 = 0.1$ are chosen for numerical computations. By running the PSO algorithm, the optimal solutions of (77) and (78) are obtained, and the following grid points are determined as

$$\begin{aligned}&(0.0000, 0.0200, 0.0400, 0.0600, 0.0800, 0.1000),\\ &(0.0000, 0.0088, 0.0197, 0.0343, 0.0562, 0.1000),\end{aligned} \tag{79}$$

respectively. One can see that the optimal solution of (77) yields the equivalent grid points. Figure 3 shows the variations of the approximate and explicit solutions versus the time t. The obtained results reveal that the approximate solutions are away from the explicit one with the increasing time. When the proposed methods are applied, the approximate solutions have more accuracy than those by using the typical Euler method.

5. Conclusions

In the typical Euler-type methods for numerically solving ordinary differential equations, the grid points are always assumed to be equidistant. When the iteration number is increasing, the approximate solution could be greatly away from the exact one. Here, we have improved the typical Euler method by reasonably arranging the nonequidistant nodes. Numerical results have been carried out to verify the novelty and advantages of the developed method by comparing with the known ones. The main findings are given as follows:

(i) A feasible distribution of the nonequidistant nodes can increase the accuracy of the typical Euler method based on the equidistant ones

(ii) The positions of the nonequidistant nodes can be determined by constructing and solving an optimization problem related to the error estimate

(iii) The shortcoming of the typical Euler method can be overcome to a certain degree

It is seen that the developed method is based on the novel idea different from those in the known Euler-type methods. In the future works, the idea in the proposed method could be used to develop the finite element method and the difference method in numerically solving ordinary and partial differential equations. The known software for numerically solving engineering and scientific problems could be further improved with more accuracy and efficiency.

Data Availability

The data used to support the findings of this study are included in this paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (nos. 11872155 and 11362002), the Guangxi Natural Science Foundation (no. 2016GXNSFAA380261), the 2017 Guangxi High School Innovation Team and Outstanding Scholars Plan, and the Innovation Project of Guangxi Graduate Education (no. YCSW2019045).

References

[1] Y. Huang and X.-F. Li, "Approximate solution of a class of linear integro-differential equations by Taylor expansion method," *International Journal of Computer Mathematics*, vol. 87, no. 6, pp. 1277–1288, 2010.

[2] M. Z. M. Kamali, N. Kumaresan, and K. Ratnavelu, "Solving differential equations with ant colony programming," *Applied Mathematical Modelling*, vol. 39, no. 10-11, pp. 3150–3163, 2015.

[3] X. Li, H. Li, and B. Wu, "A new numerical method for variable order fractional functional differential equations," *Applied Mathematics Letters*, vol. 68, pp. 80–86, 2017.

[4] G. R. Liu, *Mesh Free Methods: Moving beyond the Finite Element Method*, CRC Press, New York, NY, USA, 2002.

[5] K. Wu and D. Xiu, "Numerical aspects for approximating governing equations using data," *Journal of Computational Physics*, vol. 384, pp. 200–221, 2019.

[6] B. Yuttanan and M. Razzaghi, "Legendre wavelets approach for numerical solutions of distributed order fractional differential equations," *Applied Mathematical Modelling*, vol. 70, pp. 350–364, 2019.

[7] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, Butterworth-Heinemann, Oxford, UK, 5th edition, 2000.

[8] A. C. Cullen and S. R. Clarke, "A fast, spectrally accurate homotopy based numerical method for solving nonlinear differential equations," *Journal of Computational Physics*, vol. 385, pp. 106–118, 2019.

[9] C. Yang, F. Deluzet, and J. Narski, "On the numerical resolution of anisotropic equations with high order differential operators arising in plasma physics," *Journal of Computational Physics*, vol. 386, pp. 502–523, 2019.

[10] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial Value Problem*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1967.

[11] R. Kress, *Numerical Analysis*, Springer-Verlag, Berlin, Germany, 1998.

[12] M. Mazandarani and A. V. Kamyad, "Modified fractional Euler method for solving fuzzy fractional initial value problem," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 1, pp. 12–21, 2013.

[13] H. Wen, "Convergence rates of full-implicit truncated Euler–Maruyama method for stochastic differential equations," *Journal of Applied Mathematics and Computing*, vol. 60, no. 1-2, pp. 147–168, 2019.

[14] B. Yin and Z. Ma, "Convergence of the semi-implicit Euler method for neutral stochastic delay differential equations with phase semi-Markovian switching," *Applied Mathematical Modelling*, vol. 35, no. 5, pp. 2094–2109, 2011.

[15] R. A. Sark and C. S. Newton, *Optimization Modelling: A Practical Approach*, Taylor & Francis Group, New York, Ny, USA, 2008.

[16] E. K. P. Chong and S. H. Żak, *An Introduction to Optimization*, John Wiley & Sons, Hoboken, NJ, USA, 4th edition, 2013.

[17] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*, Springer, Berlin, Germany, 2008.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE Press, Perth, Australia, December 1995.

[19] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Publishers, Burlington, MA, USA, 2001.

[20] M. Clerc, *Particle Swarm Optimization*, ISTE Ltd., London, UK, 2006.

[21] A. R. Jordehi, "Particle swarm optimisation for dynamic optimisation problems: a review," *Neural Computing and Applications*, vol. 25, no. 7-8, pp. 1507–1516, 2014.

[22] A. Lin, W. Sun, H. Yu, G. Wu, and H. Tang, "Global genetic learning particle swarm optimization with diversity enhancement by ring topology," *Swarm and Evolutionary Computation*, vol. 44, pp. 571–583, 2019.

[23] H. Zhang and Q. Hui, "Parallel multiagent coordination optimization algorithm: implementation, evaluation, and applications," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 984–995, 2016.

[24] H. Zhang and Q. Hui, "Cooperative bat searching algorithm: a combined perspective from multiagent coordination and swarm intelligence," in *Proceedings of the 2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Xi'an, China, August 2017.

[25] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, and Y. Li, "A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease," *Neurocomputing*, vol. 320, pp. 195–202, 2018.

[26] N. Zeng, Z. Wang, Y. Li, M. Du, and X. Liu, "Identification of nonlinear lateral flow immunoassay state-space models via particle filter approach," *IEEE Transactions on Nanotechnology*, vol. 11, no. 2, pp. 321–327, 2012.

[27] N. Zeng, Z. Wang, B. Zineddin et al., "Image-based quantitative analysis of gold immunochromatographic strip via cellular neural network approach," *IEEE Transactions on Medical Imaging*, vol. 33, no. 5, pp. 1129–1136, 2014.

[28] N. Zeng, Z. Wang, H. Zhang, and F. E. Alsaadi, "A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay," *Cognitive Computation*, vol. 8, no. 2, pp. 143–152, 2016.

[29] C. Canuto and A. Tabacco, *Mathematical Analysis I*, Springer-Verlag Italia, Milano, Italy, 2008.

[30] P. W. Zhang and T. J. Li, *Numerical Analysis*, Peking University Press, Beijing, China, 2015, in Chinese.

[31] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[32] W. T. Reid, *Riccati Differential Equations*, Academic Press, New York, NY, USA, 1972.

[33] M. V. Soare, P. P. Teodorescu, and I. Toma, *Ordinary Differential Equations with Applications to Mechanics*, Springer, Dordrecht, The Netherlands, 2007.

[34] M. A. El-Tawil, A. A. Bahnasawi, and A. Abdel-Naby, "Solving Riccati differential equation using Adomian's decomposition method," *Applied Mathematics and Computation*, vol. 157, no. 2, pp. 503–514, 2004.

[35] B.-Q. Tang and X.-F. Li, "A new method for determining the solution of Riccati differential equations," *Applied Mathematics and Computation*, vol. 194, no. 2, pp. 431–440, 2007.

[36] A. C. Hindmarsh and L. R. Petzold, "Algorithms and software for ordinary differential equations and differential-algebraic equations, Part I: Euler methods and error estimation," *Computers in Physics*, vol. 9, no. 1, pp. 34–41, 1995.