AIMS Mathematics, 10(3): 5399–5420.
DOI: 10.3934/math.2025249
Received: 26 September 2024
Revised: 17 February 2025
Accepted: 24 February 2025
Published: 10 March 2025

https://www.aimspress.com/journal/Math

Research article

Hybrid B-spline collocation method with particle swarm optimization for solving linear differential problems

Seherish Naz Khalid Ali Khan and Md Yushalify Misro*

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Gelugor, Pulau Pinang, Malaysia

* **Correspondence:** Email: yushalify@usm.my.

Abstract: B-spline collocation methods were developed to provide simpler numerical solutions for differential problems. Over the years, various types of B-splines have been established, including the cubic B-spline collocation method (CBSM), cubic trigonometric B-spline collocation method (CTBSM), extended cubic B-spline collocation method (ECBSM), and cubic hybrid B-spline collocation method (CHBSM). Among these methods, CHBSM has been shown to produce the most accurate approximations due to the presence of a free parameter, γ, which allows for greater flexibility in the basis functions. However, the accuracy of the CHBSM is highly dependent on the value of γ, which must be optimized for improved results. While traditional brute-force optimization methods can achieve minimal errors, they often require significant computational time and effort. Therefore, this study has proposed using particle swarm optimization (PSO) to efficiently determine the optimal γ value for the CHBSM. The optimized CHBSM (OCHBSM) was tested on four examples of linear two-point boundary value problems (BVPs), including a linear BVP system. For comparison, the well-established CBSM and CTBSM were also applied to the same problems. The numerical results were analyzed and compared with analytical solutions revealing that the OCHBSM provided the most accurate approximations among the methods tested. Moreover, an average improvement percentage of 99.83% was achieved across all examples, indicating that our method outperforms the compared methods significantly.

Keywords: boundary value problem; numerical approximation; collocation method; hybrid B-spline; particle swarm optimization
Mathematics Subject Classification: 34K10, 34K28, 65D05, 65D07

1. Introduction

Ordinary differential equations (ODEs) are often used to model many practical applications in the fields of applied mathematics, physics, chemistry, and engineering. Generally, ODEs can be classified

into two types: linear and non-linear ODEs. An ODE is said to be linear if the dependent variables and all their derivatives are of degree one, respectively. In contrast with the non-linear ODE, the dependent variables and their derivatives can be of degree greater than one and/or with any multiplication between the variables and derivatives. The general form of the nth-order ODE can be written as

$$F(y(x), y'(x), \ldots, y'(x)^{n-1}, y^n(x)) = f(x), \qquad x \in [a,b], \tag{1.1}$$

for some $n \in \mathbb{N}, n \geq 2$, where y is a function of x, $y'(x)$ is the first derivative with respect to x, and $y^n(x)$ is the nth derivative with respect to x. In this paper, a second-order linear two-point BVP will be considered in the following form:

$$\begin{aligned} y''(x) + q(x)y'(x) + r(x)y(x) &= f(x), \qquad x \in [a,b], \\ y(a) = \alpha_1,\ y(b) &= \alpha_2, \end{aligned} \tag{1.2}$$

where $\alpha, b, \alpha_1, \alpha_2$ are real numbers. Note that the boundary conditions in Eq (1.2) are called the Dirichlet conditions where are the most common type of boundary conditions and will be considered in this study. Moreover, Dirichlet boundary conditions ensure that the continuity of the solution will be obtained [1]. To solve physical problems is sometimes very challenging and requires extensive effort. Therefore, it is recommended to use numerical solutions for solving real-life application problems. Several numerical techniques, including the variational approach [2], finite difference method (FDM) [3–5], finite element method (FEM) [3,5], finite volume method (FVM) [3,5], and the shooting method (LSM) [4], have been applied to the two-point BVP solutions.

The cubic B-spline interpolation method (CBI) was proposed by [5] to solve for two-point BVP solutions. This study can be considered as a breakthrough as it gives the fundamental idea of the B-spline collocation method. Based on the computed maximum errors, it can be concluded that this method was more precise than the methods by [3]. Various numerical techniques based on CBI have been extensively used since then to solve both linear and nonlinear BVPs [6–8]. The cubic trigonometric B-spline collocation method (CTBSM) and extended cubic B-spline collocation method (ECBSM) were investigated by [9] as solutions to linear two-point BVPs. In comparison with the CBSM, the CTBSM produced better results when dealing with trigonometric problems. A hybrid version of the CBSM and CTBSM schemes has been presented in [10] to address nonlinear two-point BVPs. This research was proven to give promising results when the free parameter γ was optimized. This is a crucial step in order to produce results with high accuracy.

In 2019, an improved method to solve singular BVPs of second order was constructed by leveraging an extended cubic B-spline basis [11]. A variety of third-order Emden-Flower-type problems were solved using a new cubic B-Spline approximation (NCBSA) by [12]. For the solutions of non-linear higher-order Korteweg-de Vries equations, a new CBS approximation was provided using the Taylor series method [13]. A novel quintic B-spline approximation was applied to solve Boussinesq equations [14]. Septic- and nonic-order splines have also been developed and applied to surfaces as tensor product schemes [15–17]. The convexity of the closed shapes was also discussed in [16]. Many authors and researchers have used higher-order numerical schemes to solve ODEs and PDEs, but these

schemes have shown computationally higher costs and inefficiency in building the solutions for a large subsequent matrix system with unknown constants due to the rigidity they possess, as compared to the spline approximations.

The CHBSM has been shown to be the superior method for approximating second-order linear BVPs compared to other spline methods [10]. The main reason for this is because the CHBSM has a free parameter, γ, which is optimized using the brute force method initially proposed in [9]. To our knowledge, the best optimization method specifically for the CHBSM has not been found. Instead, the optimization approach by [18] for parameter λ in the ECBSM was adopted and applied to the CHBSM. According to this approach, Newton's method was chosen and implemented for the minimization process. This method was proven to give promising results [6, 9, 10, 19, 20] and thus became the reference of this study.

An optimization algorithm is a process or technique designed to identify the optimal solution or set of solutions to a problem by maximizing or minimizing a specific objective function [33]. Particle swarm optimization (PSO) is a widely adopted, straightforward, and efficient technique. It belongs to the population-based algorithms which were inspired by population group behavior. PSO was originally discovered by [21] when the authors observed the social behavior of a flock of birds. Since then, PSO has been extensively used to optimize various types of problems mostly in the field of engineering [22]. For instance, the one-dimensional nonlinear Schrödinger equation was effectively solved using PSO with an exponential B-spline [22]. The authors then introduced the use of the exponential modified cubic B-spline differential quadrature method (Expo-MCB-DQM) with PSO to tackle Sine–Gordon equations [23]. The method significantly enhanced the stability and accuracy of the solution compared to earlier methods. To find the optimal parameter value of the radial basis functions for solving PDEs, the "PSO with Kansa's method" based on collocation techniques was integrated, and its performance superiority was demonstrated through numerical results [24].

As an application in geometric modeling and computer-aided design, scattered data points were effectively fitted with ball B-spline curves using PSO [25]. The PSO algorithm has also been applied into developable surfaces for shape optimization. A highly accurate developability GHT-Bézier surface was visualized in [26] where the authors applied PSO to find the optimum shape control parameters of the surface. However, there is a major lack of study on the implementation of PSO in numerical approximations, specifically in the B-spline collocation method. This is because most research opt to modify their basis functions [27, 28] or use the aforementioned brute force with Newton's method, instead of using an optimization approach. This issue has motivated us to explore PSO as a potential optimization method for spline approximations. Therefore, the primary objective of this paper is to incorporate the use of the PSO technique to obtain an optimized value of the free parameter, γ, in the OCHBSM for solving linear second-order BVPs.

This paper is organized as follows: Section 2 introduces the basis functions of the cubic hybrid B-spline approach and explains the formation of the collocation method. It also details the integration of particle swarm optimization (PSO) with the CHBSM. Section 3 provides a few numerical examples of linear two-point boundary value problems (BVPs) where both the proposed method and existing approaches such as the CBSM and CTBSM were applied. The results are then analyzed and discussed in Section 4. Finally, the conclusion of the study with suggestions for future work are presented in Section 5.

2. Methodology

The basis function of the cubic hybrid B-spline method is presented in this section. Subsequently, a collocation method is formed, namely the CHBSM. A particle swarm optimization approach is employed to optimize the free parameter in the CHBSM, which is called the OCHBSM.

2.1. Cubic hybrid B-spline

The hybrid B-spline is defined as the linear combination of the B-spline function, $B_j^k(x)$, and the trigonometric B-spline function, $T_j^k(x)$. The j-th hybrid B-spline basis function of k-th order is defined in [20] as:

$$H_j^k(x) = \gamma B_j^k(x) + (1-\gamma)T_j^k(x), \qquad \gamma \in R. \tag{2.1}$$

It is important to point out that the value of the parameter γ in Eq (2.1) has significant importance in the hybrid B-spline basis functions. When $\gamma = 1$, the basis function is reduced to the B-spline basis function and when $\gamma = 0$, the basis function becomes the trigonometric B-spline basis function. The relationship between the order, k, and degree, d, of a basis function is defined by $k = d+1$ [29]. For instance, if a B-spline function of degree $d = 4$, is used, the value of k will be 5, as the order is simply the degree plus one. In our study, B-spline basis functions of degree $d = 3$, were used; hence, the order $k = d+1 = 4$ was substituted. However, it is possible to use other values of k, as a B-spline curve can also be defined by the control points, n. Based on [29], a B-spline curve can be constructed up to order $n+1$ by adjusting the locality of the B-spline. Subsequently, the cubic hybrid B-spline basis can be obtained by substituting $k = 4$ into Eq (2.1) resulting in

$$H_j^4(x) = \gamma B_j^4(x) + (1-\gamma)T_j^4(x), \qquad \gamma \in R. \tag{2.2}$$

The definitions of $B_j^4(x)$ and $T_j^4(x)$ in [20] are as follows:

$$B_j^4(x) = \frac{1}{6h^3}\begin{cases} (x-x_j)^3, & x=[x_j,x_{j+1}], \\ h^3+3h^2(x-x_{j+1})+3h(x-x_{j+1})^2-3(x-x_{j+1})^3, & x=[x_{j+1},x_{j+2}], \\ h^3+3h^2(x_{j+3}-x)+3h(x_{j+3}-x)^2-3(x_{j+3}-x)^3, & x=[x_{j+2},x_{j+3}], \\ (x_{j+4}-x)^3, & x=[x_{j+3},x_{j+4}], \\ 0, & otherwise, \end{cases} \tag{2.3}$$

$$T_j^4(x)=\frac{1}{\varphi}\begin{cases}\delta^3(x_j,x), & x=[x_j,x_{j+1}],\\ \delta(x_j,x)[\delta(x_j,x)]\theta(x_{j+2},x)+\theta(x_{j+3},x)\delta(x_{j+1},x)]+\theta(x_{j+4},x)\delta^2(x_{j+1},x), & x=[x_{j+1},x_{j+2}],\\ \delta(x_j,x)\theta^2(x_{j+3},x)+\theta(x_{j+4},x)[\delta(x_{j+1},x)]\theta(x_{j+3},x)\theta(x_{j+4},x)\delta(x_{j+2},x)], & x=[x_{j+2},x_{j+3}],\\ \theta^3(x_{j+4},x), & x=[x_{j+3},x_{j+4}],\\ 0, & otherwise,\end{cases} \tag{2.4}$$

where $h=\frac{(b-a)}{n}$, $\delta(x_j,x)=\sin\frac{x-x_j}{2}$, $\theta(x_j,x)=\sin\frac{x_j-x}{2}$, and $\varphi=\sin(\frac{h}{2})\sin(h)\sin(\frac{3h}{2})$.

2.2. Collocation method

This section explains the collocation method of the cubic hybrid B-spline. An approximate solution of Eq (1.2) using the CHBSM can be written as:

$$S(x)=\sum_{j=-3}^{n-1}C_jH_j^4(x)\approx f(x), \qquad x\in[x_0,x_n], \tag{2.5}$$

where $C_j(x)$ are the unknown real coefficients and $H_j^4(x)$ are the cubic hybrid B-spline basis functions presented in Eq (1.2). The term $f(x)$ is the continuous function in the domain $[x_0,x_n]$ located at the right-hand side of the BVP. Analogous to the CBSM and CTBSM, there are three nonzero terms at each knot, x_j, namely $H_j^4(x)$, $H_{j-2}^4(x)$, and $H_{j-1}^4(x)$ due to the local support property of the B-spline basis. The values are tabulated in Table 1.

Table 1. Values of $H_j^4(x)$, $\frac{d}{dx}H_j^4(x)$, and $\frac{d^2}{dx^2}H_j^4(x)$.

	x_j	x_{j+1}	x_{j+2}	x_{j+3}	x_{j+4}
$H_j^4(x)$	0	h_1	h_2	h_1	0
$\frac{d}{dx}H_j^4(x)$	0	h_3	0	h_4	0
$\frac{d^2}{dx^2}H_j^4(x)$	0	h_5	h_6	h_5	0

The values of h_j for $j=1,2,3,...,6$ are as follows:

$$h_1=\frac{\gamma}{6}+\frac{(1-\gamma)\sin^2(\frac{h}{2})}{\sin(h)\sin(\frac{3h}{2})}, \qquad h_2=\frac{4\gamma}{6}+\frac{2(1-\gamma)\sin^2(\frac{h}{2})}{\sin(\frac{3h}{2})}, \qquad h_3=\frac{\gamma}{2h}+\frac{3(1-\gamma)}{4\sin(\frac{3h}{2})}, \qquad h_4=\frac{-\gamma}{2h}-\frac{3(1-\gamma)}{4\sin(\frac{3h}{2})},$$

$$h_5=\frac{\gamma}{h^2}+\frac{3(1-\gamma)[\sin(\frac{h}{2})-2\sin^3(\frac{h}{2})+sin(\frac{3h}{2})]}{8\sin(\frac{h}{2})\sin(h)\sin(\frac{3h}{2})}, \qquad h_6=\frac{-2\gamma}{h^2}-\frac{3(1-\gamma)[\sin(2h)+2\sin^2(\frac{h}{2})+sin(h)]}{4\sin(\frac{h}{2})\sin(h)\sin(\frac{3h}{2})}.$$

Suppose $S(x)$ is the approximated solution for the analytical solution $y(x)$ in Eq (1.2). We can replace $y(x)$ and its derivatives by substituting $S(x)$ and its corresponding derivatives. By substituting x_j and applying the the boundary conditions into Eq (1.2), it can be rewritten as:

$$S''(x_j)+q(x_j)S'(x_j)+r(x_j)S(x_j)\approx f(x_j), \qquad x\in[x_0,x_n], \tag{2.6}$$
$$S(x_0)=\alpha_1,\ S(x_n)=\alpha_2,$$

for $j=0,1,...,n-1$. Thus, the approximated solution in Eq (2.5) and its derivatives with respect to x, and the three nonzero terms at each knot x_j, given in Table 1, are substituted into Eq (2.5) yielding

$$S(x_j)=C_{j-3}H^4_{j-3}(x_j)+C_{j-2}H^4_{j-2}(x_j)+C_{j-1}H^4_{j-1}(x_j)\approx f(x_j), \tag{2.7}$$
$$\text{or } S(x_j)=C_{j-3}(A_1)+C_{j-2}(A_2)+C_{j-1}(A_1)\approx f(x_j),$$

$$S'(x_j)=C_{j-3}\frac{d}{dx}H^4_{j-3}(x_j)+C_{j-2}\frac{d}{dx}H^4_{j-2}(x_j)+C_{j-1}\frac{d}{dx}H^4_{j-1}(x_j)\approx f'(x_j), \tag{2.8}$$
$$\text{or } S'(x_j)=C_{j-3}(A_3)+C_{j-2}(0)+C_{j-1}(A_3)\approx f'(x_j),$$

$$S''(x_j)=C_{j-3}\frac{d^2}{dx^2}H^4_{j-3}(x_j)+C_{j-2}\frac{d^2}{dx^2}H^4_{j-2}(x_j)+C_{j-1}\frac{d^2}{dx^2}H^4_{j-1}(x_j)\approx f''(x_j), \tag{2.9}$$
$$\text{or } S''(x_j)=C_{j-3}(A_4)+C_{j-2}(A_5)+C_{j-1}(A_4)\approx f''(x_j),$$

where

$$A_i=\gamma\sigma_i+(1-\gamma)\eta_i,\ \text{for } i=1,2,...,5 \tag{2.10}$$

and

$$\sigma_1=\frac{1}{6},\qquad \sigma_2=\frac{4}{6},\qquad \sigma_3=\frac{-1}{2h},\qquad \sigma_4=\frac{1}{h^2},\qquad \sigma_5=\frac{-2}{h^2},$$
$$\kappa_1=\sin(\frac{h}{2}),\qquad \kappa_2=\sin(h),\qquad \kappa_3=\sin(\frac{3h}{2}),\qquad \kappa_4=\sin(2h),$$
$$\eta_1=\frac{\kappa_1^2}{\kappa_2\kappa_3},\qquad \eta_2=\frac{2\kappa_2}{\kappa_3},\qquad \eta_3=\frac{-3}{4\kappa_3},\qquad \eta_4=\frac{3(\kappa_1^2\kappa_1^3+\kappa_3)}{8\kappa_1\kappa_2\kappa_3},\qquad \eta_5=\frac{-3(\kappa_4+2\kappa_1^2\kappa_2)}{4\kappa_1\kappa_2\kappa_3}.$$

Equations (2.7)–(2.9) are then substituted into Eq (1.2) giving

$$\begin{aligned}&[C_{j-3}[\gamma\delta_4+(1-\gamma)\eta_4]+C_{j-2}[\gamma\delta_5+(1-\gamma)\eta_5]+C_{j-1}[\gamma\delta_4+(1-\gamma)\eta_4]]\\&+q(x_j)[[C_{j-3}(-\gamma\delta 3-(1-\gamma)\eta_3)]+C_{j-1}r(x_j)[(\gamma\delta_3+(1-\gamma)\eta_4]]\\&+r(x_j)[[C_{j-3}[\gamma\delta_1+(1-\gamma)\eta_1]+C_{j-2}[(\gamma\delta_2+(1-\gamma)\eta_2]+C_{j-1}[\gamma\delta_1+(1-\gamma)\eta_1]].\end{aligned} \tag{2.11}$$

Collecting the terms that contain C_{j-3}, C_{j-2}, and C_{j-1} from Eq (2.11) leads to

$$
\begin{aligned}
C_{j-3}[\gamma\delta_4+(1-\gamma)\eta_4-q(x_j)(\gamma\delta 3+(1-\gamma)\eta_3)+r(x_j)(\gamma\delta_1+(1-\gamma)\eta_1] \\
+C_{j-2}[\gamma\delta_5+(1-\gamma)\eta_5+r(x_j)(\gamma\delta_2+(1-\gamma)\eta_2] \\
+C_{j-1}[\gamma\delta_4+(1-\gamma)\eta_4+q(x_j)(\gamma\delta_3+(1-\gamma)\eta_3] \\
+r(x_j)(\gamma\delta_1+(1-\gamma)\eta_1]=f(x_j),
\end{aligned} \tag{2.12}
$$

with boundary conditions

$$S_H(x_0)=C_{-3}[\gamma\delta_1+(1-\gamma)\eta_1]+C_{-2}[\gamma\delta_2+(1-\gamma)\eta_2]+C_{-1}[\gamma\delta_1+(1-\gamma)\eta_1]=\alpha_1, \tag{2.13}$$

$$S_H(x_n)=C_{n-3}[\gamma\delta_1+(1-\gamma)\eta_1]+C_{n-2}[\gamma\delta_2+(1-\gamma)\eta_2]+C_{n-1}[\gamma\delta_1+(1-\gamma)\eta_1]=\alpha_2. \tag{2.14}$$

Equations (2.12)–(2.14) result in a tri-diagonal matrix system of size $(n+3)(n+3)$. This system can be denoted as $BC_j=F$. The matrix C is a column matrix of the coefficients $C=(C_0,C_1,\ldots,C_{n-1},C_n)$, and the right-hand-side matrix is the function $F=(\alpha_1,f(x_0),f(x_1),\ldots,f(x_n),\alpha_2)^T$ where the coefficients of matrix B are presented as follows:

$$
B=\begin{bmatrix}
Q_1 & Q_2 & Q_1 & 0 & 0 & \ldots & \ldots & 0 \\
\lambda_0(x_0) & \mu_0(x_0) & \rho_0(x_0) & 0 & 0 & \ldots & \ldots & 0 \\
0 & \lambda_1(x_1) & \mu_1(x_1) & \rho_1(x_1) & 0 & \ldots & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\
\vdots & \ddots & \ldots & 0 & 0 & \lambda_n(x_n) & \mu_n(x_n) & \rho_n(x_n) \\
0 & \ldots & \ldots & 0 & 0 & Q_1 & Q_2 & Q_1
\end{bmatrix}
$$

where

$$Q_1=\gamma\delta_1+(1-\gamma)\eta_1, \quad Q_2=\gamma\delta_2+(1-\gamma)\eta_2, \tag{2.15}$$

$$\lambda_j(x_j)=[\gamma\delta_4+(1-\gamma)\eta_4]-q(x_j)[(\gamma\delta 3+(1-\gamma)\eta_3)]+r(x_j)(Q_1), \tag{2.16}$$

$$\mu_j(x_j)=[\gamma\delta_5+(1-\gamma)\eta_5]+r(x_j)(Q_2), \tag{2.17}$$

$$\rho_j(x_j)=[\gamma\delta_4+(1-\gamma)\eta_4]-q(x_j)[(\gamma\delta 3+(1-\gamma)\eta_3)]+r(x_j)(Q_1) \tag{2.18}$$

for $j=0,1,\ldots,n-1$. The values of the unknown C_j for $j=0,1,\ldots,n-1$ can be obtained by solving the matrix system using

$$C_j=B^{-1}F, \tag{2.19}$$

and are substituted in Eq (1.2) to get the approximated solution of the second-order linear two-point BVP in Eq (2.5). Note that matrix B is a square matrix where the determinant is nonzero. Hence, it is a non-singular matrix such that the inverse of it exists.

2.3. Optimization of γ

The accuracy level of the approximated solutions of second-order linear two-point BVPs using the CHBSM is highly dependent on the value of the free parameter γ. This section explains the method of determining γ with the use of swarm optimization, namely particle swarm optimization, after establishing the collocation method in the previous section. This method is a simple and direct method which can be applied to the splines. Figure 1 and Algorithm 1 show the flowchart and process of the proposed OHCBSM, respectively.

Algorithm 1 Numerical approximation of the CHBSM with a PSO scheme

1: Define the basis function of the cubic hybrid B-spline, $H(x)$, as in Eq (2.2).
2: Develop a collocation method for each problem as discussed in Section 2.
3: Solve the tridiagonal matrix and calculate the value of unknown parameter, c.
4: Get the spline function by substituting the value of the unknown parameter, c into Eq (2.5).
5: Optimize the value of the free parameter γ using the PSO algorithm.
Start with PSO
6: The objective function is the L_2 norm in Eq (2.21).
7: Initialize the PSO parameters.
8: Calculate the fitness of each particle and find the best one.
9: **for** $i \leq$ swarm size **do**
10: Calculate the maximum and minimum velocities and positions of the particle.
11: Obtain the *pbest* and *gbest* values.
12: **if** $newbestvalue \leq bestvalue$ **then**
13: Update the new *pbest* and *gbest* values.
14: **else**
15: No update on the *pbest* and *gbest* values.
16: **end if**
17: The error of the new best particle after substituting in Eq (2.2) is less than 1×10^{-4}.
18: **end for**
19: Calculate the fitness of each particle.
20: Return the new best particle and the position.
21: Apply the OCHBSM on the numerical examples.
22: Calculate the errors as in Eq (2.21) and Eq (2.22).
23: Construct the graphs of exact and approximated solutions.

PSO is an evolutionary computation method similar to the genetic algorithm (GA). It was proposed by [21] in 1995. In PSO, swarms called particles can be defined as $P_i \in [a,b]$ where $i = 1,2,3,\ldots,D$ and $a,b \in R$ where D is the dimension. Each particle has its own velocity and position which are randomly initialized at the start. Each particle has to maintain its positions *pbest*, known as the local best position, and *gbest*, known as the global best position, among all the particles. The following equations are used to update the position and velocity of the particle [30].

$$\begin{aligned} V_i(t+1) &= V_i(t) + c_1 \times r_1(pbest - n_j(b)) + c_2 \times r_2(gbest - x_i(t)) \\ X_i(t+1) &= X_i(t) + V_i(t+1) \end{aligned} \tag{2.20}$$

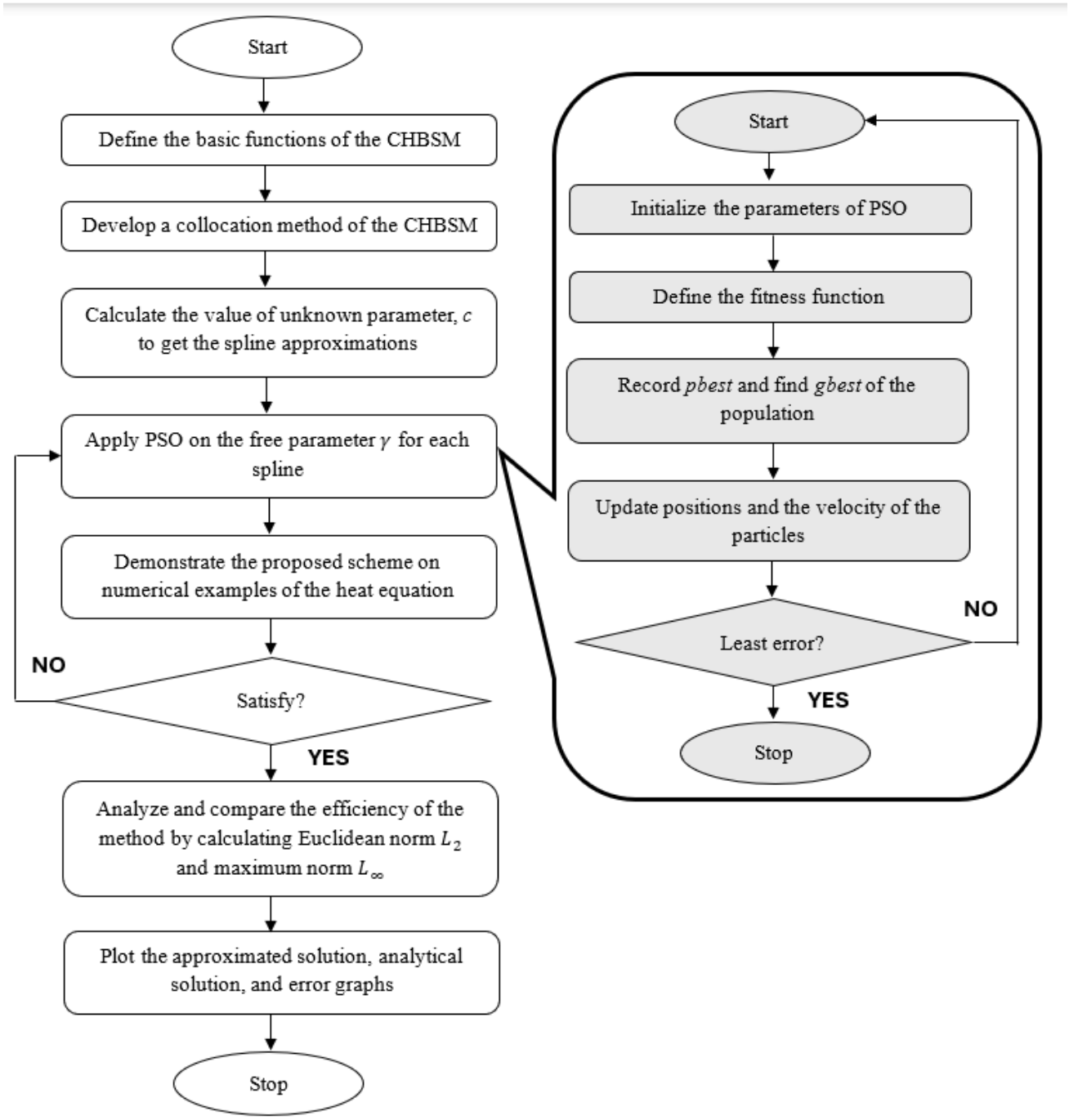


Figure 1. Flowchart of the proposed OCHBSM scheme.

We are given that V_i is the velocity, X_i is the particle position, *pbest* is the personal best particle position, and *gbest* is the global best particle position. r_1 and r_2 are two random numbers ranging from $[0, 1]$ where c_1 and c_2 are the cognitive coefficients. When selecting parameters for PSO, the goal is to balance exploration (searching broadly) and exploitation (focusing on promising areas) by

adjusting factors such as population size, inertia weight, w, and cognitive coefficients, based on the problem's complexity, in order to achieve fast convergence to a near-optimal solution while avoiding local optima.

In this study, the parameters are initialized as follows: number of particles (population size, N) = 20, weight inertial coefficient, $w = 0.9$, cognitive coefficient, c_1 = 2, social coefficient, c_2 = 2, number of iterations = 50, lower bound = -2, and upper bound = 8. The population size chosen is small (i.e., 20), as our problem is less complex. Generally, complex problems require a larger population size to explore a wider search space while simpler problems may require a smaller population. The inertia weight, w, controls the exploration of particles whether to move further across the search space or to focus on areas near the best solution, and the given range is between 0 and 1 [31]. According to most research, the standard inertia weight implemented in PSO is $w = 0.9$, which is closer to 1 to encourage further exploration of particles in the search space [32]. Moreover, a recent work [33] showed that the results obtained using the stated inertia weight are reliable.

For cognitive and social coefficients, they are typically set equal (e.g., $c_1 = c_2 = 2$) in many standard implementations such as in [33,34], but tuning these values can help in fine-tuning the balance between exploration and exploitation. Other values are also possible, but large values can lead to instability whereas very small values might slow down convergence. Adjusting these coefficients can significantly affect the algorithm's performance in finding optimal solutions. Based on our test run, the errors converge more quickly as the number of iterations increases. Hence, we simply choose a large number of iterations where the errors will be significantly small.

The values of upper and lower boundaries are adopted from [9] which states that the boundary values of γ must be between -2 and 8 to satisfy the B-spline properties. Eq (2.12) with boundary conditions in Eq (2.13) and Eq (2.14) is the fitness function for x_j for $j = 0, 1, 2, \ldots, n$. The values of *pbest* and *gbest* are constantly updated for each iteration. The work by [30] discussed PSO in greater detail. To determine the accuracy of the scheme, the maximum absolute error L_∞ and the Euclidean norm L_2 are calculated as follows:

$$L_2 = \sqrt{\sum_{j=1}^{n-1} [S(x_j) - y(x_j)]^2}, \tag{2.21}$$

$$L_\infty = \max \sum_{j=1}^{n-1} [S(x_j) - y(x_j)]. \tag{2.22}$$

The maximum percentage error and improvement percentage can also be calculated to further verify the accuracy and efficiency of our proposed method using the following formulas, respectively.

$$\text{Maximum Percentage Error} = \left|\frac{\text{Exact value - Approximated value}}{\text{Exact value}}\right| \times 100\%, \tag{2.23}$$

$$\text{Improvement Percentage} = \left|\frac{\text{Initial error - New error}}{\text{Initial error}}\right| \times 100\%. \tag{2.24}$$

The numerator of the maximum percentage error in Eq (2.23) corresponds to L_∞ in Eq (2.22), which represents the maximum difference between the approximated solution $S(x_j)$ and the analytical

solution $y(x_j)$. Meanwhile, Eq (2.24) is used to compute the error improvement between the two methods, where the initial error is defined as the maximum percentage error of the CBSM or CTBSM, and the new error corresponds to the maximum percentage error of our proposed method.

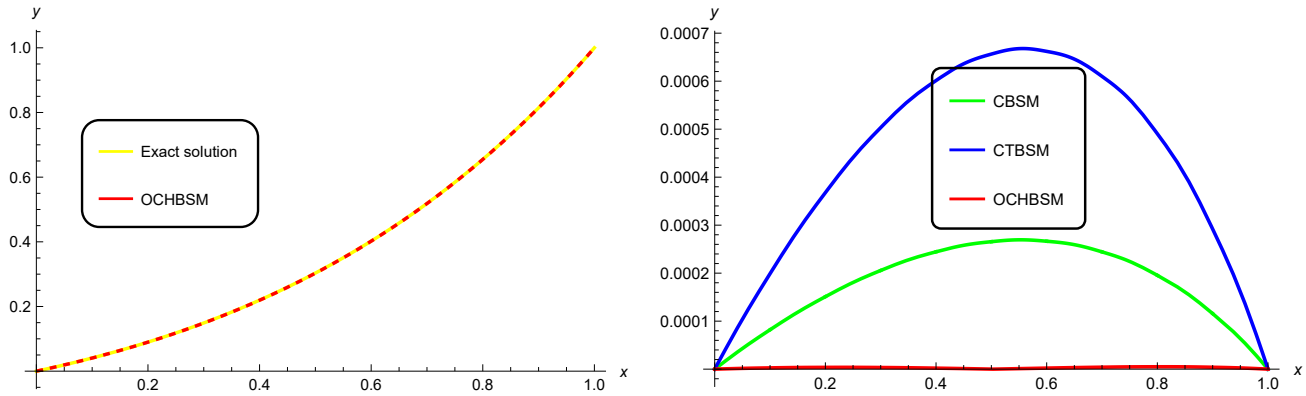
3. Results and discussion

In this section, four numerical examples were studied to demonstrate the accuracy of the proposed method. The results obtained are compared with the analytical solution of each example. Our approach shows significant improvements over Caglar's method [5] in both accuracy and usability. For all examples, PSO was iterated several times to determine the optimal parameter value, γ. A value is considered optimal when the associated errors are smaller than those from previous results. Unlike prior methods, our approach considered the value of γ at each knot, allowing for enhanced local control. We recorded the optimized γ values at each knot and calculated their average, which was then used in Eq (2.2).

Example 1. [4]
Consider the following linear equation:

$$y''(x) - y(x) = 2e^{x-1}. \tag{3.1}$$

The analytical solution is given by xe^{x-1}. Figure 2(a) compares the approximated and analytical solutions using the OCHBSM with $n = 10$, showing that the approximated solution closely matches the analytical one as both solutions overlapped, indicating minimal errors. Figure 2(b) shows the errors for the approximated solutions using the CBSM, CTBSM and OCHBSM. It is evident that the approximation error is smallest for the OCHBSM, followed by the CBSM and CBTSM. The smaller error for the OCHBSM suggests that its approximated solution is closest to the analytical solution compared to the other methods.



(a) Exact solution. **(b)** Error plots.

Figure 2. Exact solution and error plots of the approximated solution for Example 1 by the OCHBSM.

Table 2 presents the absolute errors and norms upon applying the CBSM, CTBSM, CHBSM with brute force optimization [35], and OCHBSM with PSO optimization on Example 1 where Table 3

displays the chosen value of parameter γ using PSO in the OCHBSM at each knot. According to Table 2, the maximum absolute error L_∞ and the Euclidean norm L_2 for the OCHBSM are 4.95596×10^{-6} and 9.93104×10^{-6}, respectively, which are the lowest compared to other methods. Additionally, the results from the CBSM and CTBSM are consistent with those reported in [35]. Evidently, our proposed method produced better approximations compared with the earlier methods.

Table 2. The absolute errors and norms for Example 1.

x	**CBSM** [4]	**CTBSM**	**CHBSM** [35] ($\gamma = -0.70786$)	**OCHBSM** ($\gamma = -0.67923$)
0.1	8.20393×10^{-5}	1.98907×10^{-4}	6.86513×10^{-7}	2.66012×10^{-6}
0.2	1.50958×10^{-4}	3.67729×10^{-4}	2.48477×10^{-6}	3.72270×10^{-6}
0.3	2.05606×10^{-4}	5.03258×10^{-4}	5.09038×10^{-6}	3.43324×10^{-6}
0.4	2.44454×10^{-4}	6.01275×10^{-4}	8.12575×10^{-6}	2.09225×10^{-6}
0.5	2.65555×10^{-4}	6.56420×10^{-4}	1.11229×10^{-5}	6.99347×10^{-8}
0.6	2.66489×10^{-4}	6.62037×10^{-4}	1.35035×10^{-5}	2.17657×10^{-6}
0.7	2.44305×10^{-4}	6.09998×10^{-4}	1.45553×10^{-5}	4.08326×10^{-6}
0.8	1.95449×10^{-4}	4.90498×10^{-4}	1.34050×10^{-5}	4.95596×10^{-6}
0.9	1.15684×10^{-4}	2.91808×10^{-4}	8.98678×10^{-6}	3.94327×10^{-6}
L_∞	2.66490×10^{-4}	6.62037×10^{-4}	1.45553×10^{-5}	4.95596×10^{-6}
L_2	6.19963×10^{-4}	1.53512×10^{-3}	2.96136×10^{-5}	9.93104×10^{-6}

Table 3. The value of parameter γ chosen using PSO at each knot for Example 1.

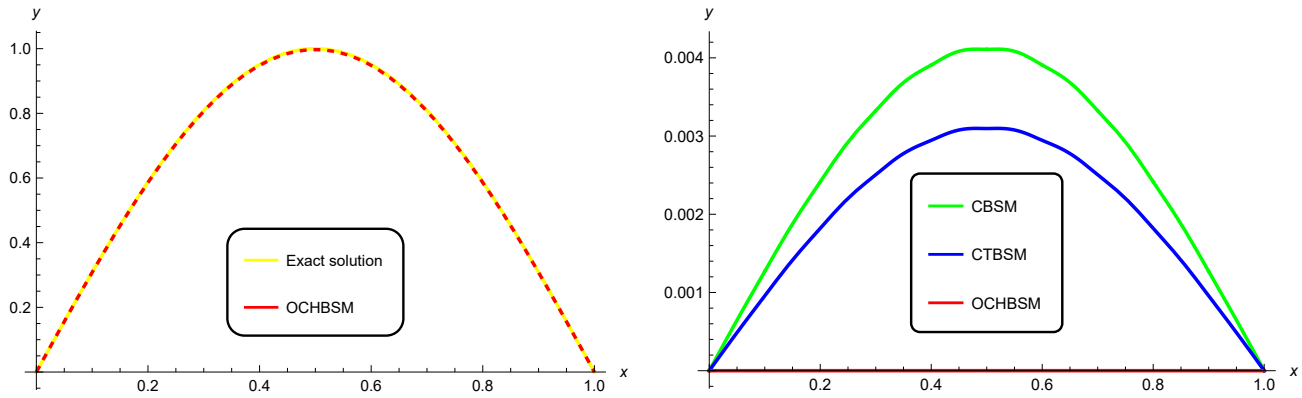
x	γ
0.1	-7.02400×10^{-1}
0.2	-6.97440×10^{-1}
0.3	-6.92450×10^{-1}
0.4	-6.84440×10^{-1}
0.5	-6.80410×10^{-1}
0.6	-6.74430×10^{-1}
0.7	-6.63360×10^{-1}
0.8	-6.59420×10^{-1}
0.9	-6.58700×10^{-1}
Avg	-6.79230×10^{-1}

Example 2. [36]
Consider the following linear equation with a trigonometric function:

$$y''(x) - \pi^2 y(x) = -2\pi^2 \sin \pi x. \tag{3.2}$$

The analytical solution is given by $\sin \pi x$. Figure 3(a) illustrates the analytical solution and approximated solution for the OCHBSM with $n = 10$. Again, both solutions overlap each other, indicating a good agreement between the approximated and the analytical solutions. The comparison of the errors between the proposed method, CBSM, and CTBSM is shown in Figure 3(b). As observed

in the figure, the OCHBSM has the smallest approximation error, followed by the CTBSM and CBSM. While the CTBSM performs slightly better than the CBSM for equation problems involving trigonometric functions, the performance of the OCHBSM exceeded both earlier methods in giving the best approximation to the analytical solution.



(a) Exact solution. **(b)** Error plots.

Figure 3. Exact solution and error plots of the approximated solution for Example 2 by the OCHBSM.

Table 4 presents the absolute errors and norms upon applying the CBSM, CTBSM, and proposed OCHBSM on Example 2 and Table 5 displays the value of parameter γ chosen using PSO in the OCHBSM at each knot. According to Table 4, the maximum absolute error L_∞ and the Euclidean norm L_2 for the OCHBSM are 3.95268×10^{-7} and 8.83843×10^{-7}, respectively. Both errors are the smallest compared to the other methods. Besides that, the results of the CBSM and CTBSM agree with those found in [19, 37].

Table 4. The absolute errors and norms for Example 2.

x	**CBSM** [19]	**CTBSM** [37]	**OCHBSM** ($\gamma = 4.05816$)
0.1	1.26968×10^{-3}	9.56780×10^{-4}	1.22144×10^{-7}
0.2	2.41508×10^{-3}	1.82199×10^{-3}	2.32330×10^{-7}
0.3	3.32407×10^{-3}	2.50488×10^{-3}	3.19776×10^{-7}
0.4	3.90768×10^{-3}	2.94467×10^{-3}	3.75920×10^{-7}
0.5	4.10877×10^{-3}	3.09620×10^{-3}	3.95268×10^{-7}
0.6	3.90768×10^{-3}	2.94467×10^{-3}	3.75922×10^{-7}
0.7	3.32407×10^{-3}	2.50488×10^{-3}	3.19778×10^{-7}
0.8	2.41508×10^{-3}	1.81990×10^{-3}	2.32332×10^{-7}
0.9	1.26968×10^{-3}	9.56780×10^{-4}	1.22144×10^{-7}
L_∞	4.10877×10^{-3}	3.09620×10^{-3}	3.95268×10^{-7}
L_2	9.18750×10^{-3}	6.92332×10^{-3}	8.83843×10^{-7}

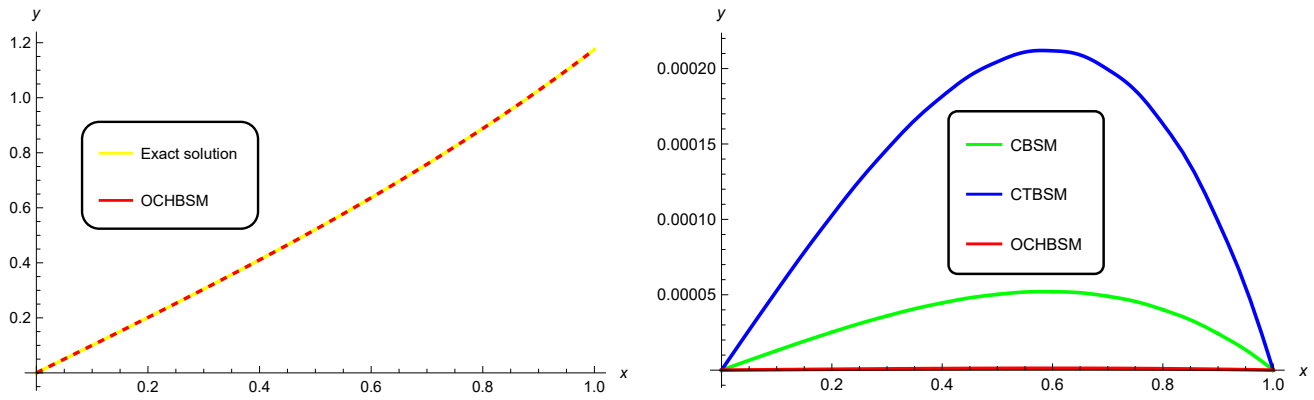
Table 5. The value of parameter γ chosen using PSO at each knot for Example 2.

x	γ
0.1	4.05831
0.2	4.05635
0.3	4.05680
0.4	4.05848
0.5	4.05771
0.6	4.06115
0.7	4.06204
0.8	4.05477
0.9	4.05780
Avg	4.05816

Example 3. [18]
Consider another linear equation:

$$y''(x) - y(x) = 0. \tag{3.3}$$

The analytical solution is given by $\sinh x$. Again, from Figure 4(a) and Figure 4(b), it was observed that our presented method is more precise than the earlier methods, the CBSM and CTBSM.



(a) Exact solution. **(b)** Error plots.

Figure 4. Exact solution and error plots of the approximated solution for Example 3 by the OCHBSM.

Table 6 lists the absolute errors and norms upon applying the CBSM, CTBSM, and OCHBSM on Example 3 and Table 7 reports the values of the parameter γ involved. From Table 6, the maximum absolute error L_∞ and the Euclidean norm L_2 are found to be 1.23029×10^{-6} and 2.78941×10^{-6}, respectively, which are the lowest among the methods evaluated. The results tabulated for the CBSM and CTBSM match with those reported in [5, 37].

Table 6. The absolute errors and norms for Example 3.

x	**CBSM** [5]	**CTBSM** [37]	**OCHBSM** ($\gamma = -3.33250 \times 10^{-1}$)
0.1	1.29420×10^{-5}	5.26942×10^{-5}	3.05327×10^{-7}
0.2	2.51763×10^{-5}	1.02508×10^{-4}	5.94097×10^{-7}
0.3	3.59800×10^{-5}	1.46497×10^{-4}	8.49359×10^{-7}
0.4	4.45988×10^{-5}	1.81591×10^{-4}	1.05338×10^{-6}
0.5	5.02311×10^{-5}	2.04526×10^{-4}	1.18721×10^{-6}
0.6	5.20111×10^{-5}	2.11777×10^{-4}	1.23029×10^{-6}
0.7	4.89906×10^{-5}	1.99482×10^{-4}	1.15996×10^{-6}
0.8	4.01204×10^{-5}	1.63366×10^{-4}	9.50989×10^{-7}
0.9	2.42288×10^{-5}	9.86598×10^{-5}	5.75017×10^{-7}
L_∞	5.20111×10^{-5}	2.11777×10^{-4}	1.23029×10^{-6}
L_2	1.17941×10^{-4}	4.80226×10^{-4}	2.78941×10^{-6}

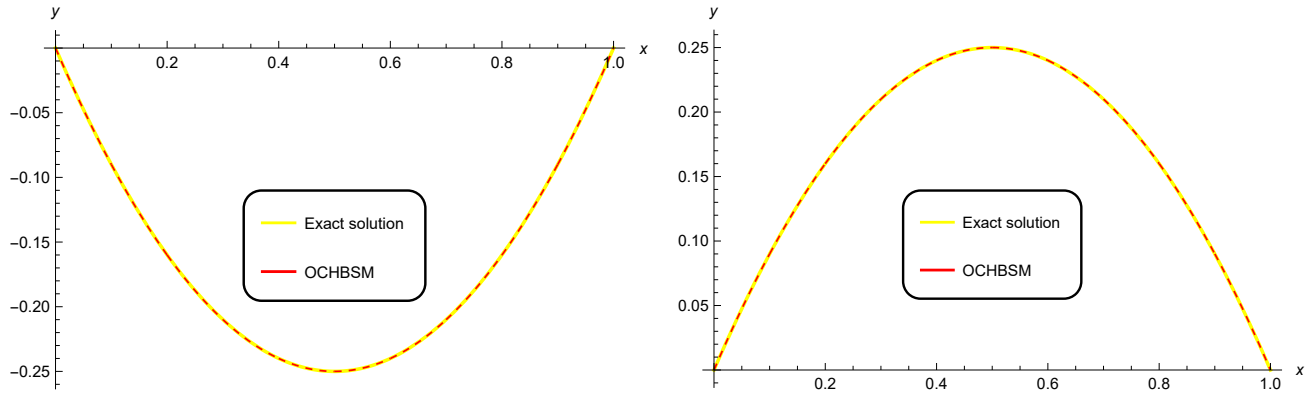
Table 7. The value of parameter γ chosen using PSO at each knot for Example 3.

x	γ
0.1	-3.29060×10^{-1}
0.2	-3.26410×10^{-1}
0.3	-3.21360×10^{-1}
0.4	-3.25480×10^{-1}
0.5	-3.25650×10^{-1}
0.6	-3.29480×10^{-1}
0.7	-3.07630×10^{-1}
0.8	-3.31130×10^{-1}
0.9	-4.03020×10^{-1}
Avg	-3.33250×10^{-1}

Example 4. [38]
To further test the efficacy of the proposed method, we will consider a linear system as follows:

$$\begin{aligned} u''(x) + xu(x) + xv(x) &= 2 \\ v''(x) + 2xv(x) + 2xu(x) &= -2 \end{aligned} \tag{3.4}$$

The analytical solution is given by $x^2 - x$ for $u(x)$ and $x - x^2$ for $v(x)$. Figure 5(a) and Figure 5(b) illustrate the approximated analytical solutions for $u(x)$ and $v(x)$, respectively. The close overlap of these results indicates good approximations, as corroborated by Table 8.



(a) Exact solution for $u(x)$. **(b)** Exact solution for $v(x)$.

Figure 5. Plot of exact solutions for $u(x)$ and $v(x)$ in Example 4 using the OCHBSM.

Table 8. The absolute errors and norms for $u(x)$ in Example 4.

x	**CBSM** [38]	**CTBSM** [39]	**OCHBSM** ($\gamma = 9.38154 \times 10^{-5}$)
0.1	-1.48492×10^{-15}	3.35626×10^{-4}	2.35536×10^{-7}
0.2	-1.33227×10^{-15}	5.98839×10^{-4}	8.46607×10^{-8}
0.3	-1.22125×10^{-15}	7.86848×10^{-4}	8.72577×10^{-8}
0.4	-9.99201×10^{-16}	8.99654×10^{-4}	7.59788×10^{-8}
0.5	-6.38378×10^{-16}	9.37255×10^{-4}	6.05528×10^{-8}
0.6	-6.10623×10^{-16}	8.99654×10^{-4}	1.87472×10^{-8}
0.7	-3.33067×10^{-16}	7.86848×10^{-4}	3.34422×10^{-8}
0.8	-5.55112×10^{-17}	5.98839×10^{-4}	8.10685×10^{-9}
0.9	1.11022×10^{-16}	3.35626×10^{-5}	1.83466×10^{-8}
L_∞	-1.48492×10^{-15}	9.37255×10^{-4}	2.35536×10^{-7}
L_2	1.17941×10^{-15}	2.16286×10^{-3}	2.02910×10^{-7}

The numerical results are tabulated in Table 8. From the table, the results for the CBSM and CTBSM are consistent with those presented in [38, 39]. On the other hand, the parameter values γ at each knot for $u(x)$ are listed in Table 9. The average of these optimized values was calculated to be 9.38154×10^{-5}. It is important to highlight that the solutions for $u(x)$ and $v(x)$ are negatives of each other. Therefore, to derive the approximation for $v(x)$, it is adequate to apply a negative sign to $u(x)$. For simplicity, the same value of γ is used to approximate the solutions of $v(x)$, resulting in results similar to $u(x)$ except for the number signs. For this reason, the results of $v(x)$ were not tabulated.

The maximum absolute error, L_∞, for the functions $u(x)$ and $v(x)$ is recorded in Table 10 for $n = 10$ and $n = 21$. It should be noted that the maximum absolute errors for both $u(x)$ and $v(x)$ are the same, as $u(x)$ is the negative of $v(x)$, leading to identical error values. At $n = 21$, the results obtained using the CBSM method align with those reported in [38]. The CBSM provides the best results due to the absence of trigonometric functions in the problem; in cases involving trigonometric functions, the OCHBSM method would produce the smallest maximum absolute error. For $n = 10$, the parameter γ is 9.38154×10^{-5} as indicated in Table 9, while for $n = 21$, γ is 1.65796×10^{-4} upon applying PSO.

Table 9. The value of parameter γ chosen using PSO at each knot for $u(x)$ of Example 4.

x	γ
0.1	8.08347×10^{-4}
0.2	1.41375×10^{-4}
0.3	1.10895×10^{-4}
0.4	-8.44533×10^{-5}
0.5	6.46065×10^{-5}
0.6	2.08382×10^{-5}
0.7	-4.25014×10^{-5}
0.8	-1.35375×10^{-5}
0.9	-5.46638×10^{-5}
Avg	9.38154×10^{-5}

Table 10. The maximum absolute error, L_∞, for $u(x)$ and $v(x)$ at different values of n in Example 4.

n	**CBSM**	**CTBSM**	**OCHBSM**
10	1.61676×10^{-15}	9.37255×10^{-4}	8.79290×10^{-8}
21	1.52073×10^{-13}	1.15141×10^{-4}	9.63323×10^{-6}

For comparative analysis, the computational time and maximum percentage error alongside the improvement percentage for each method are computed and summarized in Tables 11 and 12, respectively. It is important to note that two different software platforms, MATLAB R2024a and Mathematica 14.0, were employed in this study. For the OCHBSM, the B-spline collocation method was implemented in Mathematica 14.0 while the PSO algorithm was executed in MATLAB R2024a. Therefore, the computational times reported may not be entirely accurate, as they were calculated separately in each software and subsequently combined. Additionally, each iteration of PSO for each spline may vary due to the initialization of random numbers, r_1 and r_2, within the range [0,1]. To mitigate this variability, the PSO algorithm was run three times for each spline, and the average computational time was recorded.

For methods other than the OCHBSM, all computations were performed exclusively in Mathematica software. As shown in Table 11, it is expected that the computational time for the OCHBSM is higher than that of CBSM and CTBSM, due to the inclusion of two basis functions and an additional optimization method. Thus, the computational work for the OCHBSM is approximately double, as both the CBSM and CTBSM must be computed prior to the OCHBSM. Although the execution time for the OCHBSM is slightly higher than the other two methods, the difference in computational time is minimal, typically on the order of a few seconds. That said, one can anticipate that as problem complexity increases, the computational time for the OCHBSM will also rise, but this is justified by the improved accuracy it offers.

From Table 12, the OCHBSM exhibits the lowest maximum percentage error and a significantly higher improvement percentage compared to the CBSM and CTBSM across all examples, except for Example 4. This discrepancy is likely due to the absence of trigonometric functions in the equation for Example 4. It needs to be highlighted that the percentage improved in the table was calculated

based on the difference between the maximum percentage error of the CBSM or CTBSM, and the new error, which corresponds to the maximum percentage error of our proposed method, as shown in Eq (2.24). The exact value is determined by using the L_∞ norm as given in Eq (2.22) to obtain the knot value where the error is the biggest. Hence, it is possible that the knot chosen in computing the exact value differs for each method, which can be seen in Example 4. In this example, the L_∞ norm was attained at knot $x = 0.1$ for the CBSM and OCHBSM while the knot value $x = 0.5$ was recorded for the CTBSM. These values are then substituted into the analytical and approximated solutions to get the corresponding values. On average, an improvement percentage of $99.828352\% \approx 99.83\%$ was achieved, indicating that our method outperforms the compared methods significantly though the differences of improvement between the CBSM and CTBSM are quite small. Therefore, these results support the claim of the accuracy and efficiency of the proposed method.

Table 11. Computational time using the CBSM, CTBSM, and OCHBSM for Example 1–Example 4.

	Computational Time (seconds, s)			
Method/Example	1	2	3	4
CBSM	1.264	1.984	2.078	0.579
CTBSM	0.765	0.531	1.062	1.141
OCHBSM	7.165	6.979	7.803	4.666

Table 12. Maximum percentage error and improvement percentage of the CBSM, CTBSM, and OCHBSM for Example 1–Example 4.

Example	**Method**	**Approx. value**	**Exact value**	**Max % error**	**% Improved**
1	**CBSM**	0.65479153	0.65498460	0.02984024	99.48887480
	CTBSM	0.65449410	0.65498460	0.07488695	99.79611930
	OCHBSM	0.65498400	0.65498460	0.00015252	-
2	**CBSM**	0.02714658	0.02741213	0.96874986	99.97890500
	CTBSM	0.02675571	0.02741213	0.06564200	99.70322350
	OCHBSM	0.02741208	0.02741213	0.00019481	-
3	**CBSM**	0.52104330	0.52109531	0.00998119	99.97067110
	CTBSM	0.52088353	0.52109531	0.04064084	99.99279690
	OCHBSM	0.52109529	0.52109531	0.00000293	-
4	**CBSM**	-0.09000000	-0.09000000	0.00000000	0.00000000
	CTBSM	-0.25093700	-0.25000000	0.37480000	99.93033350
	OCHBSM	-0.09000024	-0.09000000	0.00026111	-
Avg					99.82835153

4. Conclusions

In conclusion, this study demonstrated the effectiveness of particle swarm optimization (PSO) to optimize the free parameter, γ, in the cubic hybrid B-spline collocation method (CHBSM). The numerical results showed that the OCHBSM outperformed traditional B-spline methods, including

the CBSM and CTBSM, in approximating second-order linear BVPs. This superior performance is attributed to the flexibility provided by the free parameter in the OCHBSM. However, the CBSM was able to solve linear systems of BVPs in Example 4 more accurately than the CTBSM and our proposed method. This is probably due to the lack of trigonometric functions in the particular example. According to [18], the CTBSM can offer better approximations than the CBSM for problems involving trigonometric functions due to the existence of trigonometric basis functions. Since the OCHBSM is a linear combination of trigonometric B-spline basis functions, it is expected that the OCHBSM would also provide improved solutions for problems involving trigonometric functions. A limitation of this study is the use of a fixed set of initialized parameters for the PSO. Consequently, the results may vary for different sets of initial parameters, potentially leading to different values of γ and, possibly, improved results. Additionally, the proposed method was applied only to linear second-order BVPs. Therefore, future work may focus on extending the PSO approach to more complex problems such as non-linear PDEs or to other spline methods with multiple free parameters or a significantly large number of nodes. On top of that, factors such as the order of the non-linear problems, the type of boundary conditions, and the selection of PSO parameters are important considerations for future improvements. Further exploration of alternative optimization techniques, such as whale optimization, could also enhance the performance of the OCHBSM.

Author contributions

Seherish Naz Khalid Ali Khan: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Visualisation, Writing–original draft; Md Yushalify Misro: Project administration, Supervision, Writing–review and editing. All authors have read and approved the final version of the manuscript for publication.

Acknowledgments

This research was supported by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme (FRGS/1/2023/STG06/USM/03/4) and the School of Mathematical Sciences, Universiti Sains Malaysia. The authors are very grateful to the anonymous referees for their valuable suggestions.

Conflict of interest

The authors declare that they have no conflicts of interest.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

References

1. N. Asaithambi, *Numerical Analysis: Theory and Practice*, Saunders College Pub., 1995.

2. S. Q. Wang, A variational approach to nonlinear two-point boundary value problems, *Comput. Appl. Math.*, **58** (2009), 2452–2455. https://doi.org/10.1016/j.camwa.2009.03.050
3. Q. Fang, T. Tsuchiya, T. Yamamoto, Finite difference, finite element and finite volume methods applied to two-point boundary value problems, *J. Comput. Appl. Math.*, **139** (2002), 9–19. https://doi.org/10.1016/s0377-0427(01)00392-2
4. S. Shafie, A. Majid, Approximation of cubic b-spline interpolation method, shooting and finite difference methods for linear problems on solving linear two-point boundary value problems, *World Appl. Sci. J.*, **17** (2012), 1–9.
5. H. Caglar, N. Caglar, K. Elfaituri, B-spline interpolation compared with finite difference, finite element and finite volume methods which applied to two-point boundary value problems, *Appl. Math. Comput.*, **175** (2006), 72–79. https://doi.org/10.1016/j.amc.2005.07.019
6. A. S. Heilat, R. S. Hailat, Extended cubic B-spline method for solving a system of non-linear second-order boundary value problems, *J. Math. Comput. Sci.*, **21** (2020), 231–242. http://dx.doi.org/10.22436/jmcs.021.03.06
7. A. Khalid, A. Ghaffar, M. N. Naeem, K. S. Nisar, D. Baleanu, Solutions of bvps arising in hydrodynamic and magnetohydro-dynamic stability theory using polynomial and non-polynomial splines, *Alex. Eng. J.*, **60** (2021), 941–953. https://doi.org/10.1016/j.aej.2020.10.022
8. A. Tassaddiq, A. Khalid, M. N. Naeem, A. Ghaffar, F. Khan, S. A. A. Karim, et al., A new scheme using cubic B-spline to solve non-linear differential equations arising in visco-elastic flows and hydrodynamic stability problems, *Mathematics*, **7** (2019), 1078. https://doi.org/10.3390/math7111078
9. N. N. Abd Hamid, A. A. Majid, A. I. M. Ismail, Extended cubic B-spline method for linear two-point boundary value problems, *Sains Malaysiana*, **40** (2011), 1285–1290.
10. A. S. Heilat, A. I. M. Ismail, Hybrid cubic B-spline method for solving non-linear two-point boundary value problems, *Int. J. Pure Appl. Math.*, **110** (2016), 369–381. https://doi.org/10.12732/ijpam.v110i2.11
11. I. Wasim, M. Abbas, M. K. Iqbal, A new extended B-spline approximation technique for second order singular boundary value problems arising in physiology, *J. Math. Comput. Sci.*, **19** (2019), 258–267. http://dx.doi.org/10.22436/jmcs.019.04.06
12. M. K. Iqbal, M. Abbas, I. Wasim, New cubic B-spline approximation for solving third order Emden–Flower type equations, *Appl. Math. Comput.*, **331** (2018), 319–333. https://doi.org/10.1016/j.amc.2018.03.025
13. M. Abbas, M. K. Iqbal, B. Zafar, S. B. M. Zin, New cubic B-spline approximations for solving non-linear third-order Korteweg-de Vries equation, *Indian J. Sci. Technol.*, **12** (2019), 1–9. https://dx.doi.org/10.17485/ijst/2019/v12i6/141953
14. T. Nazir, M. Abbas. M. Iqbal, A new quintic B-spline approximation for numerical treatment of Boussinesq equation, *J. Math. Comput. Sci.*, **20** (2020), 30–42. http://dx.doi.org/10.22436/jmcs.020.01.04

15. A. Ghaffar, M. Iqbal, M. Bari, S. Muhammad Hussain, R. Manzoor, K. Sooppy Nisar, et al., Construction and application of nine-tic B-spline tensor product ss, *Mathematics*, **7** (2019), 675. https://doi.org/10.3390/math7080675

16. M. Iqbal, S. A. Abdul Karim, A. Shafie, M. Sarfraz, Convexity preservation of the ternary 6-point interpolating subdivision scheme, in *Towards Intelligent Systems Modeling and Simulation: With Applications to Energy, Epidemiology and Risk Assessment*, **383** (2021), 1–23. https://doi.org/10.1007/978-3-030-79606-8_1

17. M. Iqbal, N. Zainuddin, H. Daud, R. Kanan, R. Jusoh, A. Ullah, et al., Numerical solution of heat equation using modified cubic B-spline collocation method, *J. Adv. Res. Numer. Heat Trans.*, **20** (2024), 23–35. https://doi.org/10.37934/arnht.20.1.2335

18. N. N. Abd Hamid, A. A. Majid, A. I. M. Ismail, Cubic trigonometric B-spline applied to linear two-point boundary value problems of order two, *Int. J. Math. Comput. Sci.*, **4** (2010), 1377–1382. https://doi.org/10.5281/zenodo.1081989

19. A. S. Heilat, H. Zureigat, B. Batiha, New spline method for solving linear two-point boundary value problems, *Eur. J. Pure Appl. Math.*, **14** (2021), 1283–1294. https://doi.org/10.29020/nybg.ejpam.v14i4.4124

20. S. Mat Zin, A. Abd Majid, A. I. M. Ismail, M. Abbas, Application of hybrid cubic B-spline collocation approach for solving a generalized nonlinear Klein-Gordon equation, *Math. Probl. Eng.*, **2014** (2014), 108560. https://doi.org/10.1155/2014/108560

21. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-international conference on neural networks*, **4** (1995), 1942–1948. http://dx.doi.org/10.1109/ICNN.1995.488968

22. R. Rani, G. Arora, H. Emadifar, M. Khademi, Numerical simulation of one-dimensional nonlinear Schrodinger equation using pso with exponential B-spline, *Alex. Eng. J.*, **79** (2023), 644–651. https://doi.org/10.1016/j.aej.2023.08.050

23. R. Rani, G. Arora, Particle swarm optimization numerical simulation with exponential modified cubic B-spline DQM, *Int. J. Appl. Comput. Math.*, **10** (2024), 135. https://doi.org/10.1007/s40819-024-01697-6

24. J. Koupaei, M. Firouznia. S. Hosseini, Finding a good shape parameter of rbf to solve pdes based on the particle swarm optimization algorithm, *Alex. Eng. J.*, **57** (2018), 3641–3652. https://doi.org/10.1016/j.aej.2017.11.024

25. Z. Wu, X. Wang, Y. Fu, J. Shen, Q. Jiang, Y. Zhu, Fitting scattered data points with ball B-spline curves using particle swarm optimization, *Comput. Graph.*, **72** (2018), 1–11. https://doi.org/10.1016/j.cag.2018.01.006

26. S. Bibi, M. Y. Misro, M. Abbas, Shape optimization of GHT-Bézier developable surfaces using particle swarm optimization algorithm, *Optim. Eng.*, **24** (2023), 1321–1341. https://doi.org/10.1007/s11081-022-09734-3

27. B. Latif, M. Y. Misro, S. A. Abdul Karim, I. Hashim, An improved symmetric numerical approach for systems of second-order two-point bvps, *Symmetry*, **15** (2023), 1166. https://doi.org/10.3390/sym15061166

28. M. Iqbal, N. Zainuddin, H. Daud, R. Kanan, H. Soomro, R. Jusoh, et al., A modified basis of cubic B-spline with free parameter for linear second order boundary value problems: Application to engineering problems, *J. King Saud Univ. Sci.*, **36** (2024), 103397. https://doi.org/10.1016/j.jksus.2024.103397

29. D. Salomon, *Curves and surfaces for computer graphics*, Springer Science & Business Media, 2007.

30. M. Imran, R. Hashim, N. E. Abd Khalid, An overview of particle swarm optimization variants, *Procedia Eng.*, **53** (2013), 491–496. https://doi.org/10.1016/j.proeng.2013.02.063

31. A. Pradhan, S. K. Bisoy, Chapter 6 - inertia weight strategies for task allocation using metaheuristic algorithm, in *BDCC*, 2022, 131–146. https://doi.org/10.1016/B978-0-323-85117-6.00004-2

32. W. Zhang, D. Ma, J. Wei, H. Liang, A parameter selection strategy for particle swarm optimization based on particle positions, *Expert Syst. Appl.*, **41** (2014), 3576–3584. https://doi.org/10.1016/j.eswa.2013.10.061

33. M. E. Nordin, M. Y. Misro, Optimized bi-quadratic trigonometric Bézier curve using particle swarm optimization, *ARASET*, **33** (2023), 258–278. https://doi.org/10.37934/araset.33.3.258278

34. J. Liu, X. Ren, H. Ma, A new pso algorithm with random c/d switchings, *Appl. Math. Comput.*, **218** (2012), 9579–9593. https://doi.org/10.1016/j.amc.2012.02.059

35. S. N. K. A. Khan, *B-spline collocation methods for linear two-point boundary Value problems*, Master's thesis, School of Mathematical Sciences, Universiti Sains Malaysia, 2022.

36. R. L. Burden, Jd faires numerical analysis, *Boston: Brooks-Cole. Pub*, 672–674.

37. N. Hamid, A. Majid, A. Ismail, Cubic trigonometric B-spline applied to linear two-point boundary value problems of order two, *Int. J. Math. Comput. Sci.*, **4** (2010), 1377–1382.

38. N. Caglar, H. Caglar, B-spline method for solving linear system of second-order boundary value problems, *Comput. Math. Appl.*, **57** (2009), 757–762. https://doi.org/10.1016/j.camwa.2008.09.033

39. A. S. Heilat, N. N. A. Hamid, A. I. M. Ismail, Extended cubic b-spline method for solving a linear system of second-order boundary value problems, *Springerplus*, **5** (2016), 1314. https://doi.org/10.1186/s40064-016-2936-4

AIMS Press