Hindawi
Discrete Dynamics in Nature and Society
Volume 2023, Article ID 5844407, 10 pages
https://doi.org/10.1155/2023/5844407

Research Article

Forecasting Pollution Using Numerical Simulation Implementing Artificial Bee Colony Optimization

Geeta Arora,[1] **Harshdeep Kaur**,[1] **Homan Emadifar**,[2,3] **Samaneh Roudgarnejad**,[4,5] **and Hesam Emadifar**[6]

[1]*Department of Mathematics, Lovely Professional University, Punjab, India*
[2]*Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran*
[3]*MEU Research Unit, Middle East University, Amman, Jordan*
[4]*Department of Agronomy, Chalous Branch, Islamic Azad University, Chalous, Iran*
[5]*Young Researchers and Elite Clube, Chalous Branch, Islamic Azad University, Chalous, Iran*
[6]*Department of Agronomy, Kermashah Branch, Islamic Azad University, Kermanshah, Iran*

Correspondence should be addressed to Homan Emadifar; homan_emadi@yahoo.com

Received 11 May 2023; Revised 27 July 2023; Accepted 9 August 2023; Published 23 August 2023

Academic Editor: Anibal Coronel

In this article, an optimization strategy is presented for the numerical solution of Burgers' equations, which play an important role in estimating and forecasting pollution. The method involves the exponential B-spline basis function as the basis function in the differential quadrature method. Since exponential B-spline involves a parameter, the artificial bee colony optimization algorithm is implemented to find the unknown parameters that result in the minimum error. Among the metaheuristic optimization algorithms, the artificial bee colony (ABC) is one that has received the greatest attention from researchers and has been successfully implemented to solve various problems in engineering and sciences. The proposed work emphasizes the calculation of the parameter of exponential basis functions, a major factor that plays a role in the error calculation using the ABC optimization algorithm. The acquired findings are provided as tables, and the physical behaviour is showcased in the form of figures and tables. The results are in good conformity with the earlier studies.

1. Introduction

Optimization is the theory of methods that make a mathematical function or system maximally useful or minimize its disadvantages. Optimization methods are used in various disciplines of research to identify solutions that maximize or decrease some parameters of the subject. The purpose of optimization is to discover an optimal or near-optimal solution with little computing effort. There has been a steady increase since 1960s in the pursuit of developing robust algorithms for challenging optimization problems by modelling them after biological systems [1]. Nowadays, while discussing optimization methods, researchers talk about evolutionary algorithms. Goat's genetic programming [2] and Fogel et al.'s evolutionary algorithm [3] are among the most well-known algorithms in this category; other notable contributors are Holland's evolutionary methods [4], Rechenberg's, and Schwefel's work [5]. Hybrid versions combining several paradigms are also very prevalent. Among the many reports, the work by Back and Schwefel [6], Michalewicz et al. [7], and Fogel [8] has provided a comprehensive overview of the current state of evolutionary algorithms. Perhaps the most well-known evolutionary computing approaches today are genetic algorithms, which are effective stochastic search and optimization strategies with broad applicability. Recently, the genetic algorithm community has focused much of its effort on industrial engineering optimization issues, resulting in a growing amount of research and practical implementations.

Swarm intelligence algorithms [9] have recently caught the attention of numerous research experts in related domains. Swarm intelligence (SI) is the collective behaviour of self-organized systems. Typically, SI systems consist of a population of simple agents interacting locally with one another and their environment. The source of inspiration is frequently nature, particularly biological systems [10]. Fleeces of birds, schools of fish, and colonies of social insects like termites, ants, and bees are all well-known instances of such swarms. Despite the fact that honey bee colonies exhibit the self-organizing features necessary for SI, researchers have been interested in the behaviour of these swarm systems to characterize novel intelligent techniques.

The artificial bee colony (ABC) [11] is one of the metaheuristic optimization algorithms that has received the greatest attention and has been successfully implemented in various applications such as the forecasting of transportation energy and energy demand [12, 13].

Due to the applicability of the ABC algorithm in optimizing the objective function effectively in terms of time and computational complexity, the method is applied in the numerical technique to obtain the unknown parameter. The present work is intended to optimize (minimize) the maximum numerical errors (L_∞) for the solutions of Burgers' equation using the differential quadrature method involving an exponential B-spline basis function using artificial bee colony optimization (ABC-EDQ) for the unknown parameter involved in exponential B-spline. The application of the method reduces the considered partial differential equation to an ordinary differential equation (ODE) by approximating the space derivatives through the usage of a differential quadrature method. The subsequent step includes computing the solution to the converted equations through the use of MATLAB solvers.

Burgers' equation came into existence in 1915 by Harry Bateman [14] followed by Burgers in 1948 [15] and hence was named "Burgers' equation." Burgers' equation makes one think of the Navier–Stokes equation for one dimension. Burgers' equation is an elementary partial differential equation which is widely practiced for applications in various fields of applied mathematics such as fluid mechanics, nonlinear acoustics, gas dynamics, and traffic flow.

Recently in 2022, researchers established an energy equation combined with the viscous Burgers' equation as a quantitative model for estimating river water thermal pollution [16]. Thermal pollution is a pollution that causes a decrease in the oxygen level of water, making aquatic life harder. It occurs due to various human activities that includes electric generating plants, steel melting facilities, and industrial boilers. Thermal pollution in rivers, lakes, and waterways can be studied using the energy equation and the viscous Burgers' equation. The proposed model is a nonlinear system of partial differential equations (PDEs), which may be thought of as an initial and boundary value problem (IBVP). The authors have examined an explicit second-order Lax–Wendroff type technique to obtain the numerical solution of the equation and visually represented the numerical solutions as a temperature profile, which exhibits good qualitative agreement with the real heat transport occurrence.

In the context of pollution, this equation plays an important role. In a work reported by researchers in 2019, Burgers' equation has also been used as a model to forecast pollution, emphasizing hydro and water pollution [17].

In this paper, the authors propose a correction strategy based on Burgers' equations to improve the estimation and prediction of pollution.

Burgers' equation is given by

$$u_t + \alpha u u_x - \nu u_{xx} = 0. \tag{1}$$

Here, the coefficient of kinematic viscosity is given as ν, a parameter which is greater than zero with a small positive value while α represents a positive constant which behaves as a driving force.

Due to various applications of Burgers' equation, there are various well-known numerical methods reported in the literature to solve this equation.

Various well-known numerical methods have been applied to find the numerical solution of Burgers' equation in the last few years, such as the differential quadrature method (DQM) with modified cubic B-spline [18], cubic B-spline basis functions in standard form [19], modified trigonometric cubic B-spline [20], and exponential modified cubic B-spline [21]. The finite element method has also been applied to the equation with the Hopf–Cole transformation by transforming the equation to the linear heat equation [22] and with the Galerkin finite element approach [23]. The equation has also been experimented with the collocation approach with quadratic B-spline basis functions [24], quartic B-spline [25], cubic B-spline [26], and modified cubic B-splines [27]. Many researchers have also implemented the finite difference approach in different forms, such as the finite difference method, which has been used with the parameter-uniform implicit difference scheme [28], the fourth-order finite difference method [29], and the implicit fourth-order compact finite difference scheme [30].

This is how the research is presented. In Section 1, an introduction to the scheme and its significant contributions in several areas are presented. Following this is an explanation of the numerical scheme in Section 2, which includes a discussion of ABC, exponential B-spline, and their implementation in the differential quadrature method. In Section 3, numerical examples of the equation are provided with various aspects of parameters and for different time levels. The thoughts on the usefulness of the system are included in Section 4.

2. Scheme Description (ABC-EDQ)

The proposed method uses artificial bee colony optimization to establish the parameter for the execution of the exponential B-spline using the differential quadrature method.

2.1. Exponential B-Spline Differential Quadrature Method (EDQ). The differential quadrature technique is a well-known method for solving partial differential equations that have been used with a variety of basic functions [31]. Many problems, including Fisher's equation [32], the Telegraph equation [33], the Korteweg-De Vries equation [34], the nonlinear Schrodinger equation [35], and many more, have been successfully solved numerically using this method. This approach has also been used for fractional differential equations [36], indicating its applicability beyond partial differential equations.

To make this method work, first consider the domain $x\epsilon [a, b]$ as $a = x_1 < x_2 < \ldots < x_N = b$, where N characterizes its distribution. The solution $u(x,t)$ is a smooth function over the solution domain, approximated for its r^{th} derivatives with respect to x as a linear sum of all functional values in terms of weighting coefficients $a_{i,j}$ as follows:

$$\frac{d^{(r)}u}{dx^{(r)}}|x_i = \sum_{j=1}^{N} a_{i,j}^{(r)} u(x_j), \; i = 1,2,\ldots,N, r = 1,2,\ldots,N-1. \tag{2}$$

To determine the requisite weighting factors, the suggested method employs exponential B-splines, a version of ordinary B-splines. The purpose is to represent the solution using the features of the B-spline basis function but with the addition of a parameter that depicts the form of the piecewise polynomial. Many attempts have been made to solve different types of differential equations using exponential basis functions, but they have always been limited by the requirement to incorporate a parameter, the value of which has only been approximated intuitively. An exponential B-spline in the piecewise domain is defined as follows:

$$\psi_m(x) = \frac{1}{h^3}\begin{cases} \beta_2(x_{m-2}-x) - \dfrac{\beta_2}{\omega}(\sin h(\omega(x_{m-2}-x))), x\epsilon [x_{m-2}, x_{m-1}\Big), \\ \alpha_1 + \beta_1(x_m - x) + \gamma_1 e^{\omega(x_m - x)} + \delta_1 e^{-\omega(x_m-x)}, x\epsilon [x_{m-1}, x_m\Big), \\ \alpha_1 + \beta_1(x - x_m) + \gamma_1 e^{\omega(x - x_m)} + \delta_1 e^{-\omega(x-x_m)}, x\epsilon [x_m, x_{m+1}\Big), \\ \beta_2(x - x_{m+2}) - \dfrac{\beta_2}{\omega}(\sin h(\omega(x - x_{m+2}))), x\epsilon [x_{m+1}, x_{m+2}\Big), \end{cases} \tag{3}$$

where h is the uniform space partition and other parameters are reported as follows:

$$\begin{aligned} \alpha_1 &= \frac{\omega hc}{\omega hc - s}, \\ \beta_1 &= \frac{\omega}{2}\left(\frac{c(c-1)+s^2}{(\omega hc - s)(1-c)}\right), \\ \beta_2 &= \frac{\omega}{2\,(\omega hc - s)}, \\ \gamma_1 &= \frac{1}{4}\left(\frac{(1-c+s)e^{-\omega h} - s}{(\omega hc - s)(1-c)}\right), \\ \delta_1 &= \frac{1}{4}\left(\frac{(-1+c+s)e^{\omega h} - s}{(\omega hc - s)(1-c)}\right), \\ c &= \cos h(\omega h), \\ s &= \sin h(\omega h). \end{aligned} \tag{4}$$

The numerical values of the function and the derivatives at nodal point can be obtained as follows:

$$\begin{aligned} &\psi_m(x_{m-1}) = \psi_m(x_{m+1}) = \frac{s-\omega h}{2(\omega hc - s)}, \\ &\psi_m(x_m) = 1 \\ &\psi_m'(x_{m-1}) = \frac{\omega(c-1)}{2(\omega hc - s)}; \psi_m'(x_{m+1}) \\ &\quad = \frac{\omega(1-c)}{2(\omega hc - s)}; \psi_m'(x_m) = 0, \\ &\psi_m''(x_{m-1}) = \psi_m''(x_{m+1}) = \frac{\omega^2 s}{2(\omega hc - s)}; \psi_m''(x_m) \\ &\quad = \frac{-\omega^2 s}{(\omega hc - s)}. \end{aligned} \tag{5}$$

The basic functions at the border grid points are redefined to meet the criterion of a diagonally dominating matrix before the basis function is implemented, as illustrated in [18].

$$\begin{aligned}
\varphi_1(x) &= \psi_1(x) + 2\psi_0(x), \varphi_2(x) = \psi_2(x) - \psi_0(x),\\
\varphi_k(x) &= \psi_k(x)\, for\, k = 3, 4, \ldots, N-2,\\
\varphi_{N-1}(x) &= \psi_{N-1}(x) - \psi_{N+1}(x), \varphi_N(x) = \psi_N(x) + 2\varphi_{N+1}(x).
\end{aligned} \tag{6}$$

2.2. Artificial Bee Colony Optimization. To optimize numerical issues, Karaboga [37] proposed the artificial bee colony (ABC) algorithm in 2005. The ABC algorithm is a swarm-based metaheuristic algorithm. It was influenced by how honey bees used intelligence in their foraging. Various algorithms, tailored to the unique intelligence of bee swarms, have been created throughout the years. A swarm of bees is a large group of bees that have gathered to form a colony [10]. The primary feature of a swarm is the foraging behaviour of the bees. A bee will collect nectar from a flower and put it in its honeycomb when it discovers a good source of nourishment. Enzymes are produced, and nectar is poured into the hive's vacated cells to begin the honey-making process. Worker bees dance to communicate with each other and tell each other about the location of potential food sources. They dance in one of the defined ways, depending on how well the food supply is doing economically. Bees perform a dance called the waggle dance to indicate the direction of the sun and, by extension, the location of food sources. How quickly the dance moves indicate the distance between the hive and the food supply. A trembling bee is a sign that the bee is uncertain about the present food supply's potential for profit [38]. There are a number of parameters involved in defining the algorithm based on the performance of bees. Figure 1 explains the types of bees and their roles. Figure 2 depicts the role of unemployed bees in hunting for food.

2.2.1. Procedure. To find optimal solutions, swarm-based algorithms rely on a cooperative process of trial and error. Peer-to-peer learning's social colony behaviour is the engine that propels ABC optimization algorithms [39]. ABC generates a set of candidate solutions and then continuously selects the best one. Evolution in an ABC population is driven by variation and selection. Each iteration of the variational process investigates a new region of the search space. The selection process ensures that prior knowledge will be put to good use.

The ABC algorithm comprises four stages: the beginning, when no data are available, the working bees, and the observing phase. Each solution in ABC's initial population is a dimension vector, and the population is generated randomly. The number of dimensions is proportional to the number of variables in the population's food-source optimization issue. The seeded bees adjust the current solution based on their own experiences and the fitness benefits of the change. The bee will replace the old food source with the new one if the fitness value of the new food source is greater than the old one. To update the position, the dimension vectors specified in the introductory phase are being used together with the necessary step sizes to calculate the new coordinates. The increment might range from −1 to 1.

The employed bee phase and the position update procedure are shown in Figure 3. Here, X_i indicates where the bee is right now, and the highlighted box indicates the randomly chosen direction. The X_k bee was chosen at random. *o* achieve this, the random bee's direction is subtracted from that of the chosen bees. The step size is an arbitrary positive integer that is multiplied by this difference. Finally, this number is added to the vector of dimensions X_i to determine the size of the new food location "V." This vector is created in the neighbourhood of X_i and has the same metric values.

Bees that are hard at work in the hive share their position and information about the nectar quality of newly developed solutions (food sources). Uninvolved bees analyse the available information and choose the best options based on the fitness probability they offer. The worker bee's observer counterpart, meanwhile, updates the position in its memory, and evaluates the upcoming resource's suitability. If the new area is more suitable than the previous one, the bee will remember it and abandon the old one.

If the food source's location hasn't been updated in a certain amount of time, it has been deemed abandoned. If a food source is abandoned, the bee assigned to it becomes a scout bee and is sent to investigate a new food source elsewhere in the search area. The ABC exit limit, or the predetermined number of cycles, is a key regulating factor. Figure 4 explains the workings of the ABC algorithm in the form of a flow chart that is also presented below as a given pseudocode (Algorithm 1).

Pseudocode of the algorithm:

3. Numerical Applications and Discussion

Consider the following boundary values for numerically investigating Burgers' equation using the provided methodology:

$$u(a,t) = \psi_1(t), u(b,t) = \psi_2(t), \tag{7}$$

and initial conditions

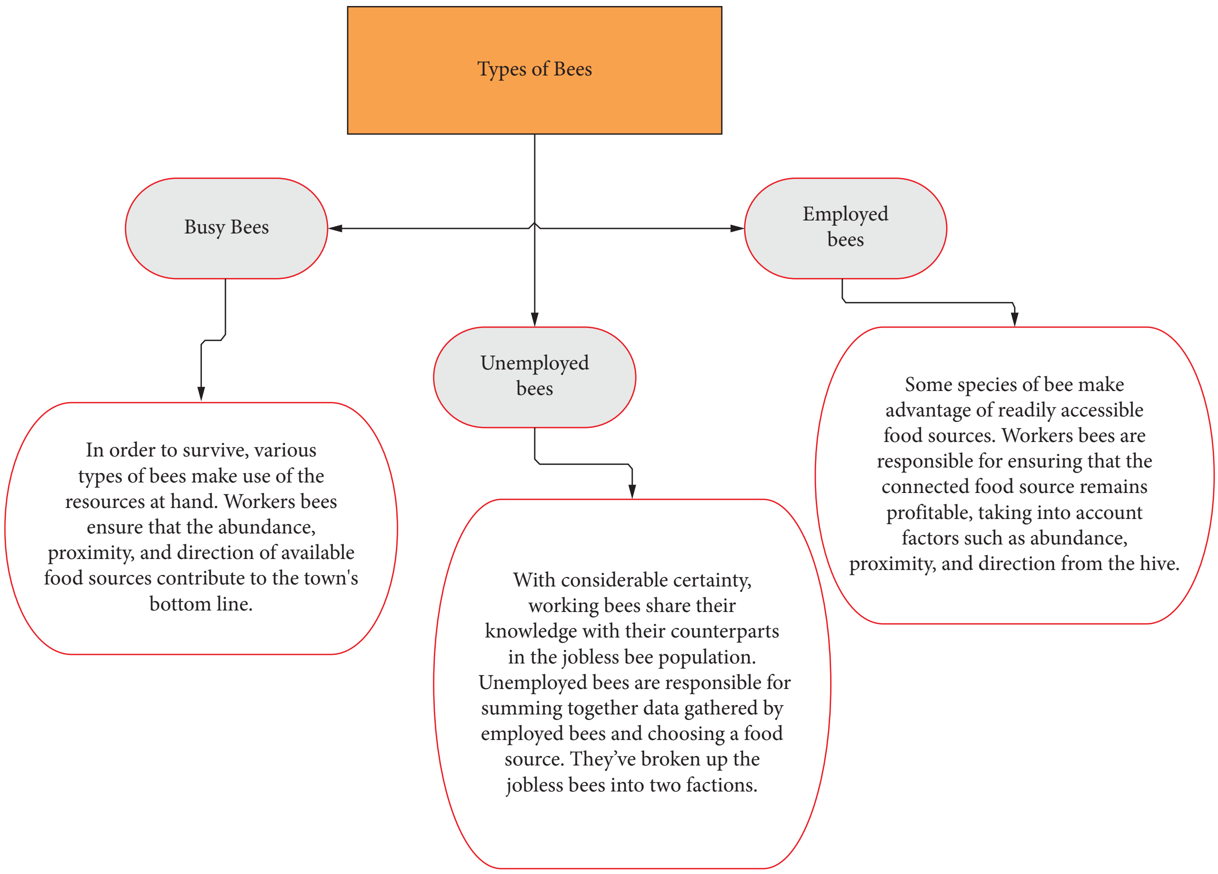

FIGURE 1: The different types of bees and their role.

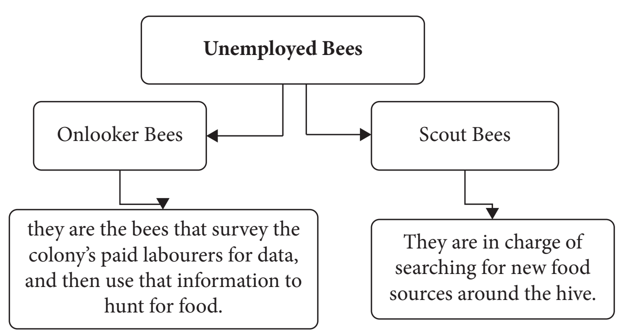

FIGURE 2: Role of unemployed bees in hunting for the food.

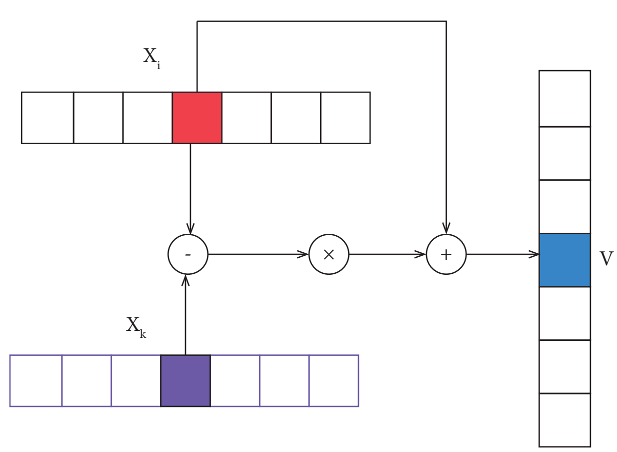

FIGURE 3: The employed bee phase and the position update procedure.

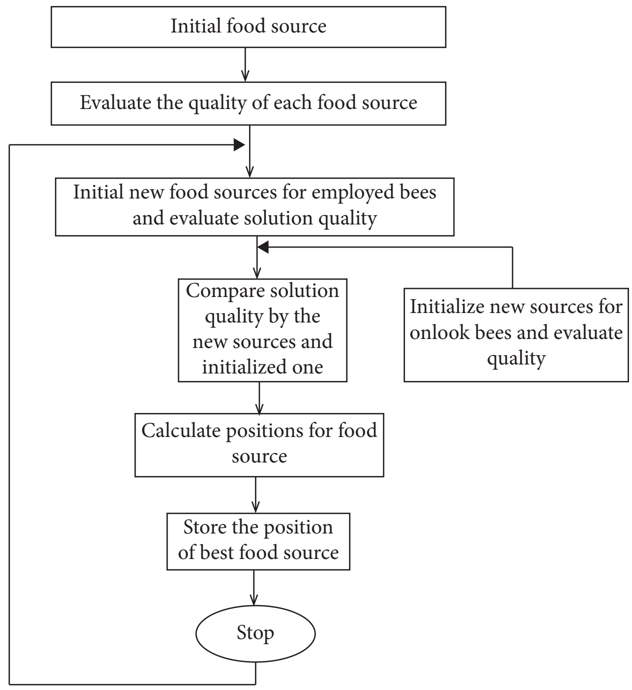

Figure 4: Working of the ABC algorithm.

```
(1) Input parameters = Population, fit, t, lb, ub, limit, Np.
(2) Calculate objective function value (f) & fitness value (fit).
(3) Set trial = 0.
(4) for i = 1 to t
    Evaluate employed bee stage
    Determined probability
    Apply Onlooker bee phase for generating food source
    Memorise the best food source
    if trial > limit
       enter into Scout bee stage
    end
    end
```

Algorithm 1: Pseudocode of the ABC algorithm.

Table 1: Solution of the equation using ABC for parameter at time $t = 0.1$ for Example 1.

$t = 0.1$	h	k	$\nu = 10^{-2}$		$\nu = 10^{-4}$		$\nu = 10^{-6}$	
			L_2	L_∞	L_2	L_∞	L_2	L_∞
Present $P = 150$	0.5	10^{-2}	$1.7755e-02$	$1.6304e-03$	$2.0226e-06$	$1.953e-06$	$2.0253e-10$	$1.9584e-10$
$P = 200$	0.025	10^{-3}	$3.7941e-03$	$3.6881e-03$	$4.2152e-07$	$4.076e-07$	$4.2195e-11$	$4.0800e-11$
$P = 200$	0.1	10^{-2}	$6.1673e-03$	$5.8412e-03$	$6.7453e-07$	$6.520e-07$	$6.7511e-11$	$6.5281e-11$
[40] $P = 150$	0.1	10^{-2}	$3.364e-03$	$3.810e-03$	$3.490e-07$	$3.979e-07$	$3.915e-11$	$4.002e-11$
[40] $P = 200$	0.1	10^{-2}	$3.381e-03$	$3.829e-03$	$3.509e-07$	$4.001e-07$	$3.510e-11$	$4.512e-11$

Table 2: Solution of the equation using ABC for parameter at time $t = 1$ for Example 1.

$t = 1$	h	k	$\nu = 10^{-2}$		$\nu = 10^{-4}$		$\nu = 10^{-6}$	
			L_2	L_∞	L_2	L_∞	L_2	L_∞
Present	0.5	10^{-2}	$2.392e-04$	$2.1617e-04$	$2.2504e-08$	$2.1760e-08$	$2.504e-12$	$2.1760e-12$
$P = 200$	0.025	10^{-3}	$4.670e-06$	$4.515e-06$	$4.6883e-10$	$4.5334e-10$	$4.6883e-14$	$4.5334e-14$
$P = 200$	0.1	10^{-2}	$7.471e-05$	$7.218e-05$	$7.5013e-09$	$7.2534e-09$	$7.5013e-13$	$7.2535e-13$
[40] $p = 200$	0.1	10^{-2}	$2.476e-02$	$2.748e-02$	$3.166e-06$	$3.6070e-06$	$3.1742e-10$	$3.6190e-10$
[40] $p = 150$	0.1	10^{-2}	$2.483e-02$	$2.757e-02$	$3.183e-06$	$3.6270e-06$	$3.1914e-10$	$3.6308e-10$

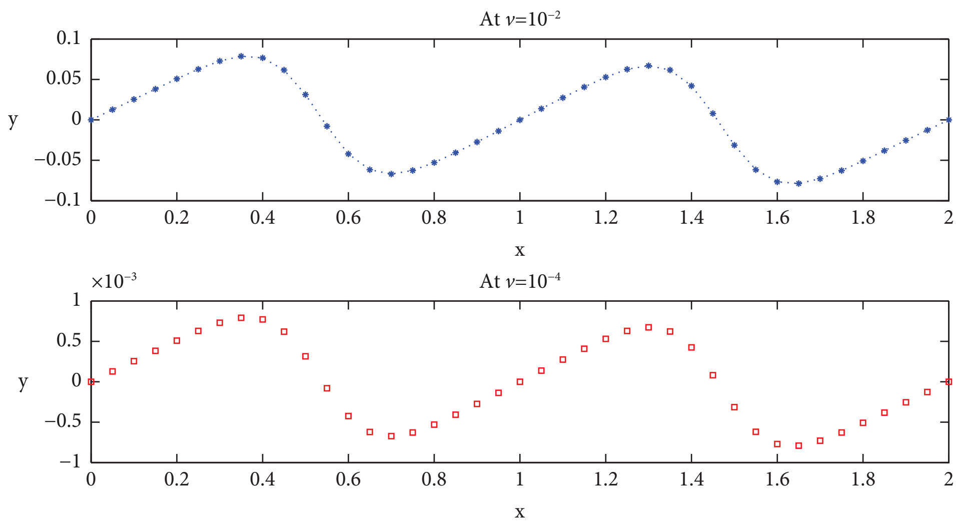

Figure 5: Behaviour of the solution that decreases in amplitude for small values of parameter ν.

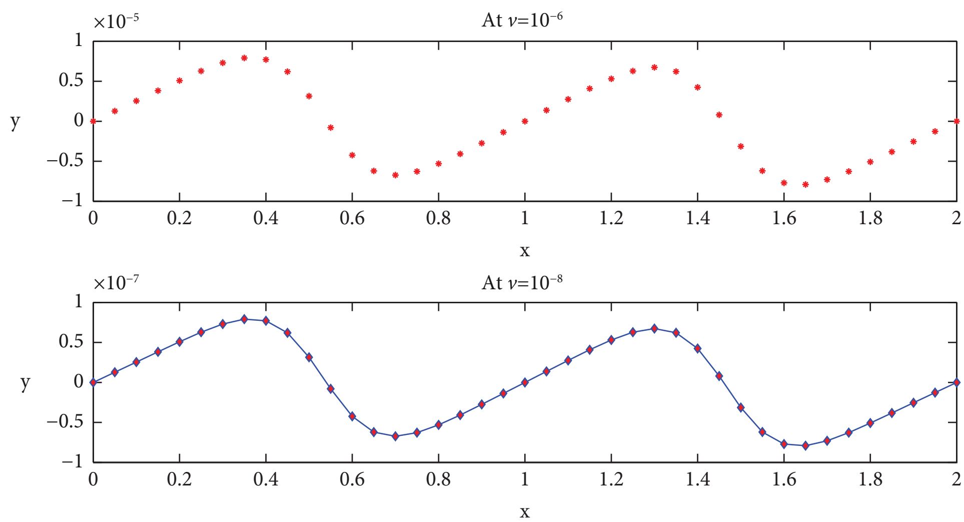

Figure 6: Behaviour of the solution of Example 1 that decreases in amplitude for very small values of ν.

$$u(x,0) = f(x), \tag{8}$$

from the exact solution on the domain $[a,b]$.

When the exponential B-spline differential quadrature technique is used to substitute the space derivatives in Burgers' equation, the equation is changed into a series of nonlinear ordinary differential equations with time dependence, as shown in the following equation:

$$\frac{\partial u_n}{\partial t} = -\alpha\, u \sum_{j=1}^{N} a_{i,j}^{(1)} u(x_j) + \nu \sum_{j=1}^{N} a_{i,j}^{(2)} u(x_j) = 0, \tag{9}$$

with $i = 1, 2, \ldots, n$.

In this paper, the MATLAB 2014 programming approach is used to determine the numerical solution of the equation while applying EDQ alongside ABC, and the results are displayed as errors. The two test problems presented are considered for determining the numerical answer with the defined approach:

Example 1. Consider Burgers' equation (1) with domain [0, 2] and zero boundary condition, as well as the initial condition from the precise solution for $\alpha = 1$. The precise solution obtained analytically is used to verify the equation's solution as follows:

$$u(x,t) = 2\pi\nu \frac{\sin(\gamma)e^{\beta} + 4\sin(2\gamma)e^{-4\beta}}{4 + \cos(\gamma)e^{\beta} + 2\cos(2\gamma)e^{-4\beta}}, \tag{10}$$

where $\gamma = \pi x, \beta = -\pi^2\nu^2 t$.

The numerical solution of the problem is achieved at various time levels for $k = 0.01$ at various ν values. The value of the parameter for the exponential B-spline is derived using ABC for application in the differential quadrature technique. The obtained numerical findings are compared

Table 3: Solution of the equation using ABC for parameter at different time levels for Example 2.

$N=121$	p	$t=1.7$		$t=3.1$	
		L_2	L_∞	L_2	L_∞
Present	1	$1.9076e-06$	$7.7721e-06$	$7.4189e-04$	$2.2055e-03$
[40]	1	$1.9076e-06$	$7.7210e-06$	$6.5722e-07$	$3.3167e-06$
Present	2	$1.9115e-06$	$7.7902e-06$	$7.4188e-04$	$2.2055e-03$
[40]	2	$1.9115e-06$	$7.7905e-06$	$6.5874e-07$	$3.3167e-06$
$N=151$					
Present	0.5	$7.4195e-04$	$2.2093e-03$	$7.4195e-04$	$2.2093e-03$
[40]	0.5	$8.7867e-07$	$3.6908e-06$	$5.1592e-07$	$3.5718e-06$
Present	1	$7.4195e-04$	$2.2093e-03$	$7.4195e-04$	$2.2093e-03$
[40]	1	$8.7908e-07$	$3.6927e-06$	$5.1601e-07$	$3.5718e-06$
[21]	—	$7.7700e-06$	$7.7500e-06$	$4.700e-07$	$1.5800e-06$
Present	2	$7.4195e-04$	$2.2093e-03$	$7.4195e-04$	$2.2093e-03$
[40]	2	$8.8075e-07$	$3.7002e-06$	$5.1637e-07$	$3.5718e-06$

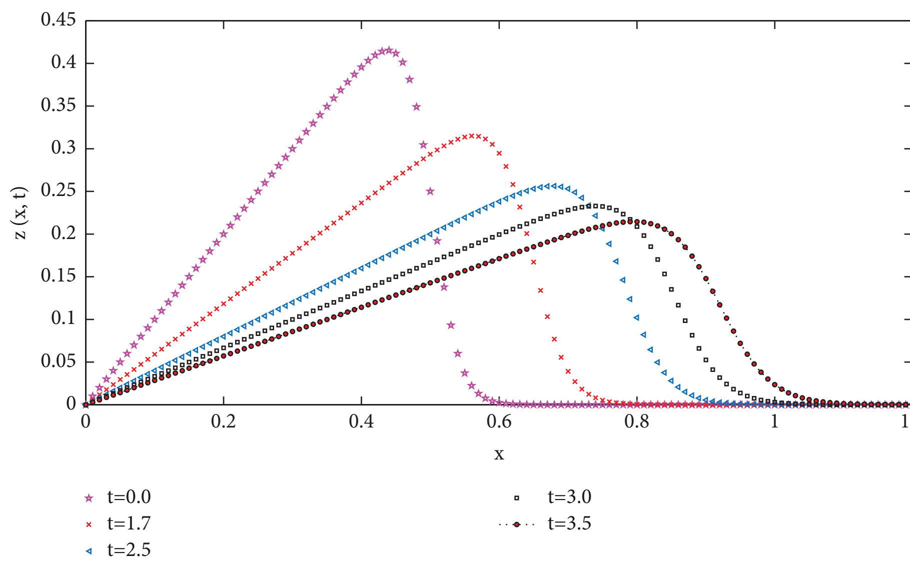

Figure 7: Behaviour of the solution for Example 2 for the different time levels.

with the exact solutions in Tables 1 and 2 and are reported in the form of errors for $t=0.1$ and $t=1$ as accessible in the literature.

The important parameters are swarm size: 5, maximum iterations: 50, inertia weight is linearly reduced, and social and cognitive coefficients are $c_1=c_2=2.05$. The numerical solutions are in good agreement with the precise solution, as evidenced by the obtained results in the form of L_2 and L_∞ errors.

Figures 5 and 6 depict the physical behaviour of the solution obtained at $t=1$ for various values of ν. From the figures, the physical behaviour of the equation over time can be seen to remain unchanged in behaviour but changing in amplitude.

It can be observed from the results presented in the Tables 1 and 2 that the algorithm has played an important role in minimizing the errors even at the same value of the parameter.

Example 2. Consider the equation (1) to be solved in the domain $[0, 1.2]$ with boundary and initial condition taken from the exact solution [19] for $\alpha = 1$, given as follows:

$$u(x,t) = \frac{x/t}{1+\sqrt{(t/g)}\exp\left(x^2/4\nu t\right)},$$
$$g = \frac{0.125}{\nu} \text{ for } t \geq 1. \tag{11}$$

The numerical solution has been obtained with $\nu = 0.005$ at different time levels for $t = 1.7$ and $t = 3.1$.

In Table 3, the errors are estimated at different time levels for time step, $k = 0.01$ and the number of domain partition as 121 and 151, the findings are compared with available data in literature. Using ABC, results are calculated for the parameters in Table 3 considering swarm size: 5; maximum iterations: 20; inertia weight is linearly decreased; and social and cognitive coefficients are taken as $c_1 = c_2 = 2.05$ and $k =$

1. The collected findings show that the numerical results are affected by the parameter as well as the number of domain divisions. The acquired findings are equivalent to the exact solutions that are accessible. Figure 7 depicts the solution behaviour along with the physical behaviour of the equation over time.

4. Concluding Remarks

The ABC technique is used to acquire the parameter for the exponential B-spline-based differential quadrature approach, which is used to calculate the numerical solution to Burgers' problem. To demonstrate the accuracy of the procedure the problems are solved with the different values of the coefficient of kinematic viscosity ranging from small to high. To illustrate the errors, different time steps and the number of domain divisions are employed. The obtained results are compared to the applied numerical method with particle swarm optimization algorithm in comparison to the ABC algorithm. It can be observed that the errors obtained by ABC algorithm are less as compared to the PSO algorithm when applied for minimizing the error while the value of parameter is same for different applied domain partitions. The proposed strategy presented in the work may be experimented using some other optimization algorithm such as spider monkey and grey wolf optimization algorithm.

Abbreviations

ABC: Artificial bee colony optimization
DQM: Differential quadrature method
E-DQM: Exponential differential quadrature method
SI: Swarm intelligence
IBVP: Initial boundary value problem
PDEs: Partial differential equations.

Data Availability

All data generated or analysed during this study are included in the article.

Disclosure

I hereby declare that the results provided are true and complete to the best of my knowledge and the paper has not been submitted, in whole or in part in any other journal.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, NY, USA, 1999.

[2] S. K. De, "The goat search algorithms," *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8265–8301, 2022.

[3] D. B. Fogel, "An overview of evolutionary programming," in *Evolutionary Algorithms*, pp. 89–109, Springer, New York, NY, USA, 1999.

[4] T. Bäck and H. P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23, 1993.

[5] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[6] T. Back and H. P. Schwefel, "Evolutionary computation: an overview," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 20–29, IEEE, New York, NY, USA, May 1996.

[7] Z. Michalewicz, D. Dasgupta, R. G. Le Riche, and M. Schoenauer, "Evolutionary algorithms for constrained engineering problems," *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 851–870, 1996.

[8] D. B. Fogel, "Evolutionary computation: a new transactions," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 1-2, 1997.

[9] Y. Qawqzeh, M. T. Alharbi, A. Jaradat, and K. N. Abdul Sattar, "A review of swarm intelligence algorithms deployment for scheduling and optimization in cloud computing environments," *PeerJ Computer Science*, vol. 7, p. e696, 2021.

[10] Y. Zhang, P. Agarwal, V. Bhatnagar, S. Balochian, and J. Yan, "Swarm intelligence and its applications," *The Scientific World Journal*, vol. 2013, Article ID 528069, 3 pages, 2013.

[11] J. C. Bansal, H. Sharma, and S. S. Jadon, "Artificial bee colony algorithm: a survey," *International Journal of Advanced Intelligence Paradigms*, vol. 5, no. 1/2, pp. 123–159, 2013.

[12] D. Özdemi̇r and S. Dörterler, "An adaptive search equation-based artificial bee colony algorithm for transportation energy demand forecasting," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 4, pp. 1251–1268, 2022.

[13] D. Özdemir, S. Dörterler, and D. Aydın, "A new modified artificial bee colony algorithm for energy demand forecasting problem," *Neural Computing & Applications*, vol. 34, no. 20, pp. 17455–17471, 2022.

[14] H. Bateman, "Some recent researches on the motion of fluids," *Monthly Weather Review*, vol. 43, no. 4, pp. 163–170, 1915.

[15] J. M. Burgers, "A mathematical model illustrating the theory of turbulence," *Advances in Applied Mechanics*, vol. 1, pp. 171–199, 1948.

[16] P. Biswas, L. S. Andallah, and K. M. Eusha-Bin-Hafiz, "Estimation of thermal pollution using numerical simulation of energy equation coupled with viscous Burgers' equation," *American Journal of Computational Mathematics*, vol. 12, no. 03, pp. 306–313, 2022.

[17] T. T. Ha, N. H. Phong, F. X. Le Dimet, and H. S. Hoang, "Data assimilation and pollution forecasting in Burgers' equation with model error function," *Comptes Rendus Mecanique*, vol. 347, no. 5, pp. 423–444, 2019.

[18] G. Arora and B. K. Singh, "Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method," *Applied Mathematics and Computation*, vol. 224, pp. 166–177, 2013.

[19] A. Korkmaz and İ. Dağ, "Cubic B-spline differential quadrature methods and stability for Burgers' equation," *Engineering Computations*, vol. 30, no. 3, pp. 320–344, 2013.

[20] G. Arora and V. Joshi, "A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 1087–1098, 2018.

[21] M. Tamsir, V. K. Srivastava, and R. Jiwari, "An algorithm based on exponential modified cubic B-spline differential

quadrature method for nonlinear Burgers' equation," *Applied Mathematics and Computation*, vol. 290, pp. 111–124, 2016.
[22] T. Öziş, E. N. Aksan, and A. Özdeş, "A finite element approach for solution of Burgers' equation," *Applied Mathematics and Computation*, vol. 139, no. 2-3, pp. 417–428, 2003.
[23] A. Dogan, "A Galerkin finite element approach to Burgers' equation," *Applied Mathematics and Computation*, vol. 157, no. 2, pp. 331–346, 2004.
[24] I. Dag, D. Irk, and A. Sahin, "B-spline collocation methods for numerical solutions of the Burgers' equation," *Mathematical Problems in Engineering*, vol. 2005, no. 5, pp. 521–538, 2005.
[25] B. Saka and İ. Dağ, "Quartic B-spline collocation method to the numerical solutions of the Burgers' equation," *Chaos, Solitons & Fractals*, vol. 32, no. 3, pp. 1125–1137, 2007.
[26] R. C. Mittal and G. Arora, "Numerical solution of the coupled viscous Burgers' equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 3, pp. 1304–1313, 2011.
[27] R. C. Mittal and R. K. Jain, "Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method," *Applied Mathematics and Computation*, vol. 218, no. 15, pp. 7839–7855, 2012.
[28] M. K. Kadalbajoo, K. K. Sharma, and A. Awasthi, "A parameter-uniform implicit difference scheme for solving time-dependent Burgers' equations," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 1365–1393, 2005.
[29] I. A. Hassanien, A. A. Salama, and H. A. Hosham, "Fourth-order finite difference method for solving Burgers' equation," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 781–800, 2005.
[30] W. Liao, "An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation," *Applied Mathematics and Computation*, vol. 206, no. 2, pp. 755–764, 2008.
[31] R. Bellman, B. G. Kashef, and J. Casti, "Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations," *Journal of Computational Physics*, vol. 10, no. 1, pp. 40–52, 1972.
[32] B. K. Singh and G. Arora, "A numerical scheme to solve Fisher-type reaction-diffusion equations," *Nonlinear Studies/Mesa-Mathematics in Engineering, Science and Aerospace*, vol. 5, no. 2, pp. 153–164, 2014.
[33] G. Arora, D. F. Mackay, D. I. Conway, and J. P. Pell, "Ethnic differences in oral health and use of dental services: cross-sectional study using the 2009 Adult Dental Health Survey," *BMC Oral Health*, vol. 17, no. 1, pp. 1–8, 2016.
[34] S. Mishra, G. Arora, H. Emadifar, S. K. Sahoo, and A. Ghanizadeh, "Differential quadrature method to examine the dynamical behavior of soliton solutions to the korteweg-de vries equation," *Advances in Mathematical Physics*, vol. 2022, Article ID 8479433, 10 pages, 2022.
[35] G. Arora, R. Rani, and H. Emadifar, "Numerical solutions of nonlinear Schrodinger equation with applications in optical fiber communication," *Optik*, vol. 266, Article ID 169661, 2022.
[36] G. Arora and Pratiksha, "Solution of inverse fractional Fisher's equation by differential quadrature method," *Journal of Physics: Conference Series*, vol. 1531, no. 1, Article ID 012088, 2020.
[37] D. Karaboga and B. Basturk, "June). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *International Fuzzy Systems Association World congress*, pp. 789–798, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
[38] T. Ye, W. Wang, H. Wang et al., "Artificial bee colony algorithm with efficient search strategy based on random neighborhood structure," *Knowledge-Based Systems*, vol. 241, p. 108306, 2022.
[39] J. C. Bansal, P. K. Singh, and N. R. Pal, *Evolutionary and Swarm Intelligence Algorithms*, Springer, Cham, Switzerland, 2019.
[40] G. Arora, P. Chauhan, M. I. Asjad, V. Joshi, H. Emadifar, and F. Jarad, "Particle swarm optimization for solving sine-gordan equation," *Computer Systems Science and Engineering*, vol. 46, no. 1, 2023.