

Solving inverse Huxley problem using a combined numerical method and improved teaching–learning-based optimization algorithm

Ahmad Aliyari Boroujeni^a, Reza Pourgholi^{b,*}, Seyed Hashem Tabasi^c

^aDepartment of Mathematics and Computer Science, Faculty of Sciences, University of Zanjan, 45371-38791, Zanjan, Iran

^bSchool of Mathematics and Computer Science, Damghan University, 36715-364, Damghan, Iran

^cDepartment of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

(Communicated by Saeid Abbasbandy)

Abstract

In this paper, inverse Huxley partial differential equations problem is solved using a combination of numerical methods and improved teaching–learning-based optimization (ITLBO) meta-heuristic algorithm. The ITLBO algorithm, which is a population-oriented algorithm, is very efficient for solving inverse problems by increasing the discovery capacity and diversity of input data. In the method presented in this article, by considering each input data of the ITLBO algorithm as an answer for inverse Huxley problem, the problem is solved by numerical methods. The results obtained from the implementation of the combination of the numerical method and the ITLBO algorithm show that it is possible to get accurate answers for inverse Huxley problem without guessing the function type of the inverse problem. The results also indicate that combining Implicit and Explicit numerical methods, along with the features of the ITLBO algorithm, which searches across various intervals, can lead to solving inverse problems like Huxley without considering the type of the unknown function. The exact results obtained from the implementation of the presented method were obtained on a 2.5 GHz 4-core Intel CPU.

Keywords: Inverse problems, Huxley equation, ITLBO algorithm, Numerical method

2020 MSC: Primary 68Wxx; Secondary 35R30, 65M32

1 Introduction

Partial differential equations (PDE) play a role in many phenomena related to the fields of physics, engineering, and other sciences such as magnetism, mechanics, dynamics, fluids, optics, heat transfer, etc [1, 3, 2, 4, 9, 13, 24, 28, 29]. Many methods have been presented in analytical or numerical solutions [1, 3, 2, 4, 9, 13, 24, 28, 29] for the direct category of these problems, but solving their inverse problems has always been associated with challenges [14, 16]. PDEs are generally divided into two categories: linear problems and non-linear problems. Inverse problems of both categories are very ill-posed problems, but in nonlinear equations, due to the nonlinear form that appears in the

*Corresponding author

Email addresses: aliyari@du.ac.ir (Ahmad Aliyari Boroujeni), pourgholi@du.ac.ir (Reza Pourgholi), tabasi@du.ac.ir (Seyed Hashem Tabasi)

equations and their solution using numerical or analytical methods, their ill-posed condition worsens, and the smallest noise or the difference with exact answer can greatly affect the answers to the problem.

[17], to simulation the ionic mechanisms for the electrical description of neurons, proposed Huxley equation. This equation also describes a wide range of non-linear physical phenomena such as the interaction between reaction mechanisms and convection and diffusion effects [21]. This equation and its generalized equations are used in many fields such as mathematics, metallurgy, engineering, biology, combustion and chemistry. The Huxley equation and its generalized equations such as Burger-Huxley describe several phenomena such as bursting oscillations [12], neural system [20], population genetics [5], branching and chaos [34], and so far several analytical methods to solve the equation are proposed. For example, using Hirota's method [30], an exact individual solution for this equation has been obtained. From the generalized Burger-Huxley equation, [32] developed an exact solitary wave solution. In [33], some wave solutions for the generalized forms of Huxley's equation have been obtained using the tanh function technique. [7] studied Huxley problem using improved physics-informed neural network (PINN) method based on deep learning. Also, there are several researches to solve Huxley and its generalized equations with numerical methods such as different classes of Adomian decomposition method [18], finite difference scheme [23], variational iteration method [8], spline method [22] is introduced. However, the nature of solving the Burger-Huxley problem is ill-posed and most of the presented articles have dealt with the direct solution of the Huxley problem and its generalized equations.

Meta-heuristic algorithms are widely used in solving many problems, and according to the type and purpose of the problem, they can be used by combining them with other methods. One of the most important algorithms that has a very good ability to solve engineering design optimization problems and nonlinear programming, is the teaching-learning-based optimization algorithm (TLBO), and this point is one of the most useful and important features of this algorithm. For this reason, improvements have been provided for it according to its functionality. For example, [26] used it to solve nonlinear continuum problems on a large scale. To escape local optima, [11] improved the TLBO algorithm with a mutation operator and modify the learning step. [19] presented a improved TLBO algorithm by integrating three different algorithms. [1] has improved this algorithm to solve nonlinear and complex problems, by adding several phases of elitism, as well as grouping students and assigning a teacher to each group. Considering this feature, in this article, inverse Huxley partial differential equations are investigated using the ITBLO [1] optimized algorithm, and a method is presented by integrating it with numerical methods such as the finite difference method. The combination of implicit forward and backward methods for discretizing the partial differential equations of the Huxley problem is introduced. By integrating this approach with ITLBO Algorithm, it becomes possible to solve the inverse Huxley problem without guessing the type of the unknown function. In the following, numerical methods, improved TLBO algorithm and their combination for solving inverse Huxley problem are discussed.

2 Inverse Huxley partial differential equations problem

In this section, Inverse Huxley partial differential equations is investigated and its numerical solution is implemented using a combination of implicit and explicit methods [31]. For the numerical solution, first the governing equations of the problem are discretized. In this article, the differential equations governing Huxley problem are as follows:

$$\begin{aligned} u(x, t)_t &= u(x, t)_{xx} + u(x, t)(1 - u(x, t))(u(x, t) - 1), \\ a \leq x \leq b, \quad 0 \leq t \leq T, \end{aligned} \tag{2.1a}$$

$$u(x, 0) = f(x), \quad a \leq x \leq b \tag{2.1b}$$

$$u(a, t) = p(t), \quad 0 \leq t \leq T \tag{2.1c}$$

$$u(b, t) = q(t), \quad 0 \leq t \leq T \tag{2.1d}$$

and the over-specified condition:

$$u(x_1, t) = g(t), \quad a \leq x_1 \leq b, \quad 0 \leq t \leq T \tag{2.1e}$$

where $f(x)$ is function of the initial condition, $p(t)$ and $q(t)$ are function of the boundary condition of the problem and T represents the final existence time for the problem. In solving examples of Huxley's inverse problems that will be examined below, boundary conditions $p(t)$ and $q(t)$ are unknown, and an additional condition, $g(t)$, is required to solve this problem. By discretizing the equations using implicit and explicit methods, we have:

$$\begin{aligned}
& -ru_{i-1,j+1} + (1+2r)u_{i,j+1} - ru_{i+1,j+1} = \quad j \geq 0, \\
& (1-k)u_{i,j} + 2ku_{i,j}^2 - ku_{i,j}^3 + kF_{i,l}, \\
& i = 1, 2, 3, \dots, N-1,
\end{aligned} \tag{2.2a}$$

$$u_{i,0} = f(i \times h), \quad i = 1, 2, 3, \dots, N-1, \quad j = 0, \tag{2.2b}$$

$$u_{0,j} = p(j \times k), \quad i = 0 \quad j = 0, 1, 2, 3, \dots, M-1, \tag{2.2c}$$

$$u_{N,j} = q(j \times k), \quad N \times h = 1, \quad j = 0, 1, 2, \dots, M-1, \tag{2.2d}$$

where $x = i \times h$, $t = j \times k$ and $r = \frac{k}{h^2}$. By using Eq. (2.2), following algebraic system of equations is obtained:

$$\begin{aligned}
& \begin{pmatrix} 1+2r & -r & 0 & 0 & 0 & 0 & 0 \\ -r & 1+2r & -r & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & -r & 1+2r & -r \\ 0 & 0 & 0 & 0 & 0 & -r & 1+2r \end{pmatrix} \begin{pmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \cdot \\ \cdot \\ \cdot \\ u_{N-2,j+1} \\ u_{N-1,j+1} \end{pmatrix} \\
& -r \begin{pmatrix} p_{j+1} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ q_{j+1} \end{pmatrix} = (1-k) \begin{pmatrix} u_{1,j} \\ u_{2,j} \\ \cdot \\ \cdot \\ \cdot \\ u_{N-2,j} \\ u_{N-1,j} \end{pmatrix} + 2k \begin{pmatrix} u_{1,j} \\ u_{2,j} \\ \cdot \\ \cdot \\ \cdot \\ u_{N-2,j} \\ u_{N-1,j} \end{pmatrix}^2 \\
& -k \begin{pmatrix} u_{1,j} \\ u_{2,j} \\ \cdot \\ \cdot \\ \cdot \\ u_{N-2,j} \\ u_{N-1,j} \end{pmatrix}^3 - k \begin{pmatrix} F_{1,j} \\ F_{2,j} \\ \cdot \\ \cdot \\ \cdot \\ F_{N-2,j} \\ F_{N-1,j} \end{pmatrix}.
\end{aligned} \tag{2.3a}$$

As it is clear from (2.3), the answer of the problem in each time step depends on the answers obtained from the past times of powers 2 and 3, and this issue makes the investigation of these problems and especially its inverse problems be of special importance. In the following, these issues are investigated with the presented method based on the ITLBO algorithm combined with the numerical solution presented in this section.

3 Improved teaching-learning-based optimization

TLBO algorithm was first presented by [26] for solving optimization problems and then his own algorithm was first improved for solving unconstrained optimization problems [27]. One of the very good features of this algorithm is its lack of dependence on the input parameters of the problem, and this makes this algorithm always be considered and optimized by researchers. The primary optimization algorithm was based on two phases of teaching and learning. In [1], this algorithm was improved by adding two phases of grouping and elitism, and its test results on optimization problems were much better than the original algorithm and other optimized TLBO algorithms and even several other algorithms, and it was used to solve inverse partial differential equation problems. In the following, this algorithm and its implementation phases will be discussed so that it can be used to solve Huxley inverse problems.

3.1 Teaching Phase

In the teaching phase, the best student of the class or group is chosen as the class teacher, and other students increase their knowledge about the teacher. using the following equations:

$$\text{Difference Mean}_i = r_i(T_i - TF \times M_i) \quad (3.1a)$$

$$X_{\text{new},i} = \text{Difference Mean}_i + X_i \quad (3.1b)$$

where TF is teaching factor, T_i is the teachers, M_i is the calculated mean of the knowledge of all students in the class or groups, $X_{\text{new},i}$ is the student with the knowledge updated by the teacher's information and X_i is the student with the basic knowledge.

3.2 Learning phase

In this phase, each student can increase his knowledge by interacting with other randomly selected knowledge. The following equations describe learning phase:

$$x_j^{\text{new}} = x_j + r(x_k - x_j) \quad \text{if } f(x_j) \leq f(x_k) \quad (3.2a)$$

$$x_j^{\text{new}} = x_j + r(x_j - x_k) \quad \text{if } f(x_j) > f(x_k) \quad (3.2b)$$

where x_j and x_k are randomly learners and x_j^{new} is updated knowledge learner.

3.3 Grouping

This phase has been added to prevent optimization problems from getting stuck in local optima. In this phase, the class is divided into several groups and teachers are randomly assigned to each group. This will give weak students a chance to improve. The governing equations in this phase are given below

$$r_{i,j,p} = \text{rand}[1 - p] \quad (3.3a)$$

$$X_{i,j}^{\text{teacher}} = X_{r_{i,j,p}} \quad (3.3b)$$

The groups of students are formed according to the teachers selected in 3.3a, 3.3b as follows:

$$G_{i,1} = \{X_{i,1}, \dots, X_{r_{i,1,p}-1}, X_{i,1}^{\text{teacher}}\} \quad (3.3c)$$

$$G_{i,2} = \{X_{r_{i,1,p}+1}, \dots, X_{r_{i,2,p}-1}, X_{i,2}^{\text{teacher}}\} \quad (3.3d)$$

.

.

.

$$G_{i,T} = \{X_{r_{i,T-1,p}+1}, \dots, X_{r_{i,T,p}-1}, X_{i,T}^{\text{teacher}}\} \quad (3.3e)$$

$$G_{i,T+1} = \{X_{r_{i,T,p}+1}, \dots, X_{i,n-1}, X_{i,n}\} \quad (3.3f)$$

where $X_{i,j}^{\text{teacher}}$ as the random teacher selected from population size in each i iteration, $G_{i,j}$, are groups of students. The teacher of the last group is also selected as the best student of the group.

3.4 Elitism

In this phase, by determining the rate of elitism, elite people are selected in each group and the best member of the class is considered as their teacher. Elitism rate (ER) increases and decreases the speed of convergence to the exact answer, because in most optimization problems, convergence should be considered while paying attention to scattered answers. The relationships between elitism rate and elite group are given below.

$$ER_{i,j} = \text{rand}[0 - 1] \quad i = 1, 2, 3, \dots, G, \quad (3.4a)$$

$$j = 1, 2, 3, \dots, T, \quad (3.4a)$$

$$W_{i,j} = \text{round}(ER_{i,j} \times (\text{rand}(1, SG_{i,j}))) \quad (3.4b)$$

where EG_i is elite group, $SG_{i,j}$ is number of students in each group (i.e. population size of the each group $p = 1, 2, 3, \dots$) and $W_{i,j}$ is Weight to determine the number of selected student from each group.

In the two phases of grouping and elitism, the teaching and learning and self-learning [27] phases are implemented each time. The general flowchart of this algorithm is shown in 1.

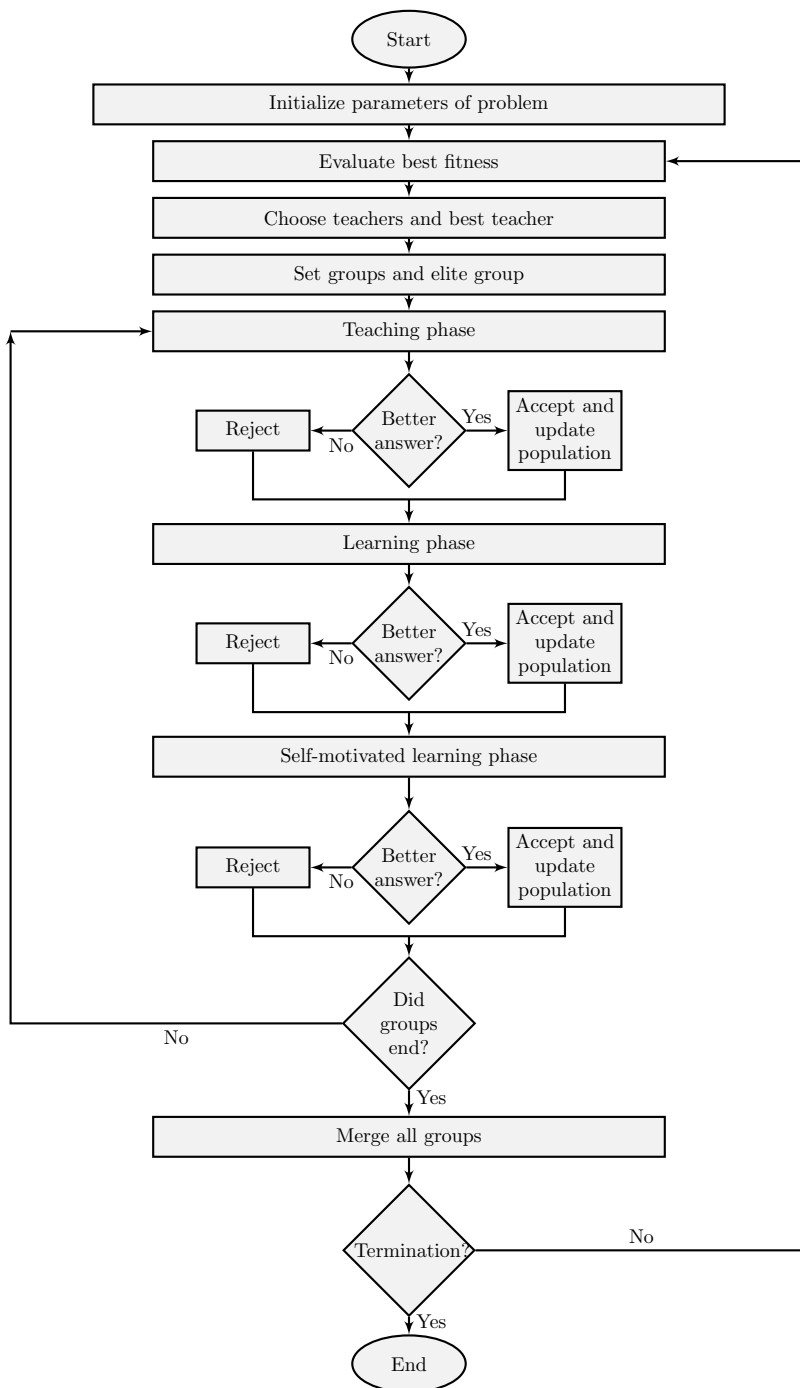


Figure 1: Overall flowchart of ITLBO algorithm

4 ITLBO for solving inverse Huxley problem

In this section, the method of solving inverse Huxley problems using the ITLBO algorithm is described. Using ITLBO algorithm, each student is considered as an answer for the unknown condition of Huxley partial differential equations problem, and the problem becomes a direct problem and is solved using the numerical method presented in Section 2. The procedure for doing this is given below:

1. Algorithm parameters are defined and the initial population is created randomly.
2. Cost function values are calculated for each student.
3. Students are grouped and a teacher is assigned to each group.
4. By the elitism rate determined in step 1, the elite group is formed.
5. For all groups and elite groups, teaching phase, learning phase and self-learning are implemented.
6. The values of the cast function of each student in the groups are calculated and if the answers are better, they replace the previous answers.
7. Steps 2 to 6 are repeated until the termination condition is met.

In this paper we approximate the unknown boundary condition of PDE problem using ITLBO algorithm. To do this, the fitness function is defined as follow:

$$f(S) = \sum_{j=1}^m (U_j - s_j)^2, \quad (4.1)$$

Eq. (2.2) must be minimum. $U_j, j = 1, 2, 3, \dots, m$ are values calculated at a point $(x, t) = (\alpha, s_j)$ by numerical methods approximated by the TLBO algorithm and $s_j = s(t_j), j = 1, 2, 3, \dots, m$ are method measured values.

Remark 4.1. Each student is estimated boundary conditions $p(t)$ or $q(t)$ at points $t_i, i = 1, 2, 3, \dots, m$, and the array composed of the characteristics of each student is interpolated for the unknown boundary condition of the PDE problem without using the type of the unknown function of the problem.

5 Numerical results and discussion

In this section, an example is examined to show the efficiency of the presentation method and the convergence of the obtained answers. Consider the following example discussed in [6]:

$$\begin{aligned} u(x, t)_t &= u(x, t)_{xx} + u(x, t)(1 - u(x, t))(u(x, t) - 1), \\ 0 \leq x \leq 1, \quad 0 \leq t \leq T, \end{aligned} \quad (5.1a)$$

$$u(x, 0) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{2\sqrt{2}}x\right), \quad 0 \leq x \leq 1 \quad (5.1b)$$

$$u(0, t) = p(t), \quad 0 \leq t \leq T \quad (5.1c)$$

$$u(1, t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{2\sqrt{2}}\left(1 - \frac{t}{\sqrt{2}}\right)\right), \quad 0 \leq t \leq T \quad (5.1d)$$

and the over-specified condition:

$$\begin{aligned} u(x_i, t) &= \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{2\sqrt{2}}\left(x_i - \frac{t}{\sqrt{2}}\right)\right), \\ i = 1, 2, 3, \dots, m \quad 0 \leq x_i \leq 1, \quad 0 \leq t \leq T \end{aligned} \quad (5.1e)$$

The exact solution of this example is

$$\begin{aligned} u(x, t) &= \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{2\sqrt{2}}\left(x - \frac{t}{\sqrt{2}}\right)\right), \\ 0 \leq x \leq 1, \quad 0 \leq t \leq T, \end{aligned} \tag{5.1f}$$

To better compare and demonstrate the efficiency of the method provided for solving the inverse Huxley problem, this problem was solved once by guessing the type of $P(t)$ function as an exponential function and another time without any guessing of the type of unknown function of problem, and its results and plots are given below. When the unknown condition function type conjecture method was used, a population of 60 students in 5 groups and with an elitism rate of 0.8 ($w = 0.8$) were used to obtain it. The number of iterations of the algorithm for optimizing values is placed on 80 and the search range $[-5 \ 5]$. To implement the presented method (without any conjecture of the unknown function type), the numerical solution in this example is the length of time and place steps on 0.2 ($h, k = 0.2$) to examine the problem in a 5×5 mesh. The ITLBO algorithm also searches for the best student for the unknown condition of the problem with a population of 60 students and in 5 groups in $[-10 \ 10]$ numerical interval. It should be noted that given the form of the boundary and initial condition functions ($f(x), q(t)$) that are present in the problem, we could have made the numerical range smaller and between 0 and 1, but we made the search range larger so that the efficiency of the method and algorithm presented would also be examined. Also the elitism rate is placed on 0.8 ($w = 0.8$) and implemented in 40 iterations to optimize the response.

Remark 5.1. *The overall effect of all errors, including deterministic and stochastic errors are considered as follows, [10]. Deterministic error is unavoidable bias deviation and stochastic error is the variance due to the amplification of measurement errors.*

$$S = \left[\frac{1}{N-1} \sum_{i=1}^N (\hat{P}_i - P_i)^2 \right]^{\frac{1}{2}}, \tag{5.2}$$

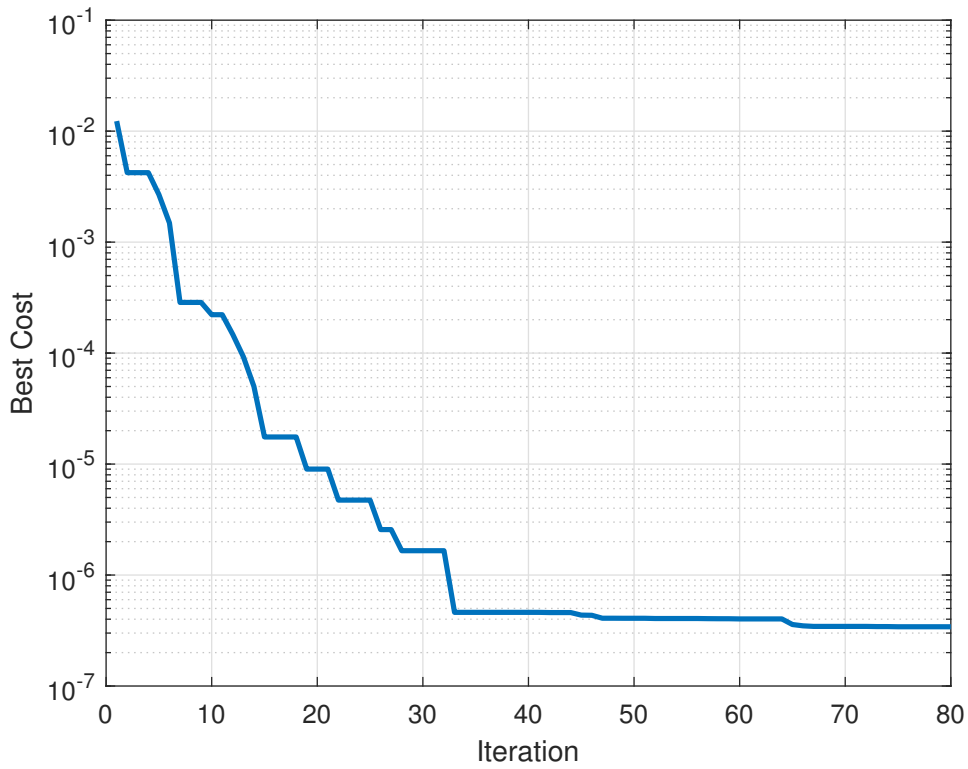
Where \hat{P}_i are calculated values from described method in N values, and P_i are exact values of $P(t)$.

Fig 2 illustrates the operation of the ITLBO algorithm in optimizing this problem. As shown in Fig. 2b, this algorithm always improves the answer by escaping from the local optimum, and this is one of the most important strengths of this algorithm. In Fig. 2b, ITLBO algorithm also tries to improve and escape from local optimum, but due to the faster convergence in this method, the speed of improvement of the fitness function is very good at first and then drops.

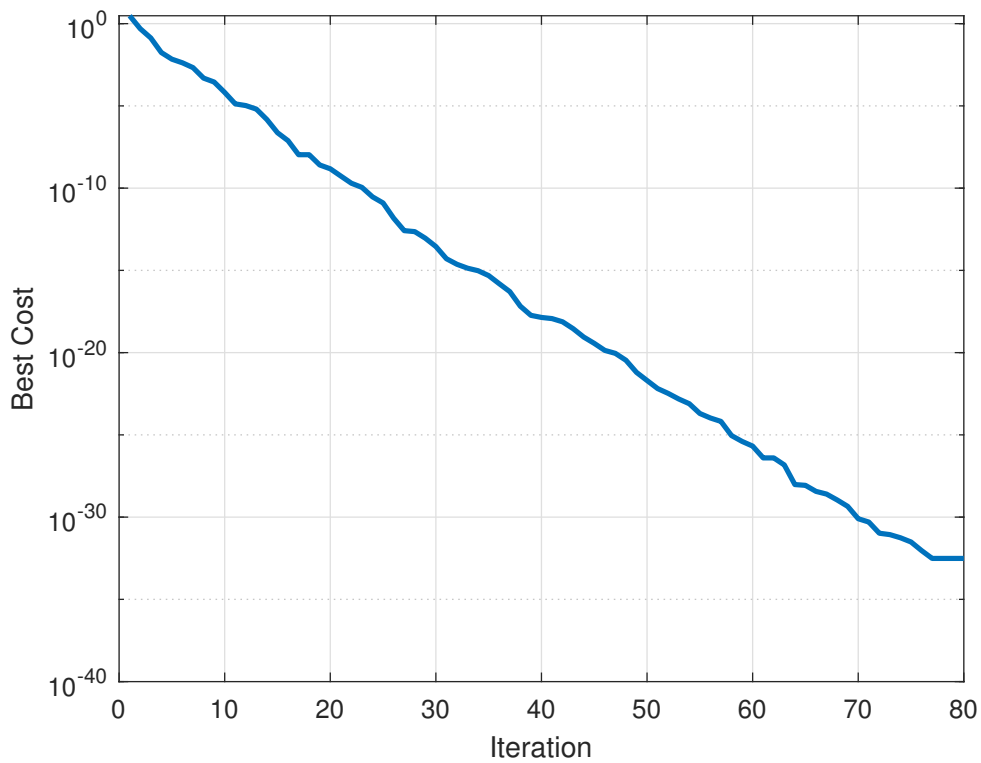
Fig. 3 shows the real value of $U(x, t)$ relative to the value of $U(x, t)$ obtained from the $P(t)$ estimate by the guessing the type of unknown function and the $P(t)$ estimate by the method provided. As these plots show, the all two answer is very close to the actual answer to the problem. In implementing this method without guessing the type of function, and also with guessing the type of unknown function, the boundary condition is obtained, which is very accurate. To better represent accuracy of answers, the contour plot is derived from the real value and the value obtained with the estimated $P(t)$ show in Fig. 4. As is evident in this form, the accuracy of the resulting response is very high. Table 1 shows the numerical results of the exact answer of the problem $U(x, t)$ in comparison with the answer obtained from the numerical solution of inverse Huxley problem with and without using the guess of the unknown function type. As it is clear from the numerical values of the table, the obtained answers are very accurate and close to the real answer of the problem.

Fig. 5 shows the actual $P(t)$ value relative to the $P(t)$ interpolated from the implementation of the method provided and also $P(t)$ obtained from the guessing the type of unknown function. As this figure also shows, the obtained $P(t)$ s are very close to the actual value, and where there is a slight difference, this difference is magnified. As is clear in the figure, both numerical answers obtained are accurate and close to the actual answer but according to that, in some problems we do not know the type of functions of the boundary conditions, the method base on without guessing the type of unknown function is very accurate and efficient to use. Table 2 shows the numerical values of $P(t)$ resulting from the implementation of the presented methods and the actual $P(t)$ values at the 5 estimated points $i = 1, 2, 3, \dots, n$ by the algorithm. As it clear, they are also very accurate and close to the actual $P(t)$.

Table 3 shows the performance comparison of ITLBO algorithm for implementing methods to obtain the unknown function of inverse Huxley problem. As it is clear from the results, the error is a little better by without using guess

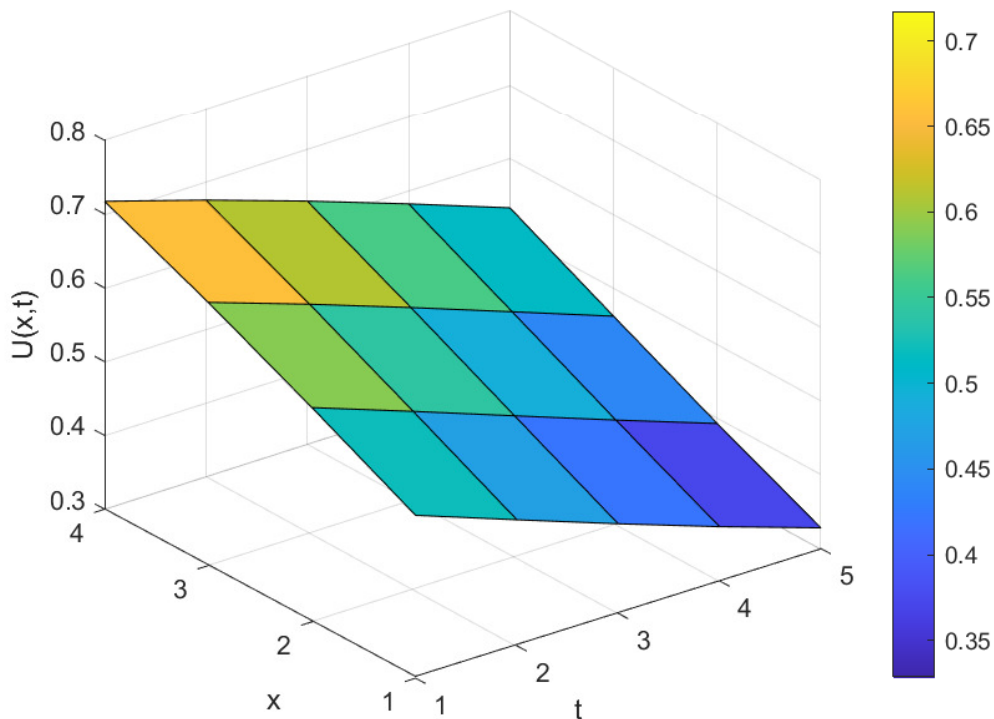


(a) The best costs of ITLBO with guess type of unknown function.

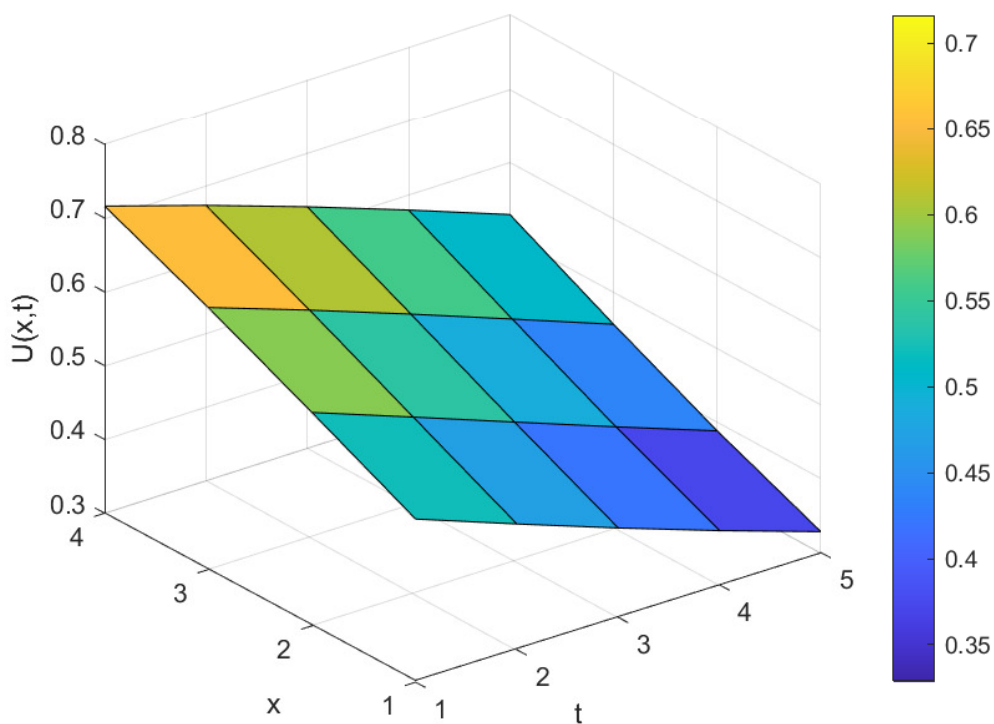


(b) The best costs of ITLBO without guess type of unknown function.

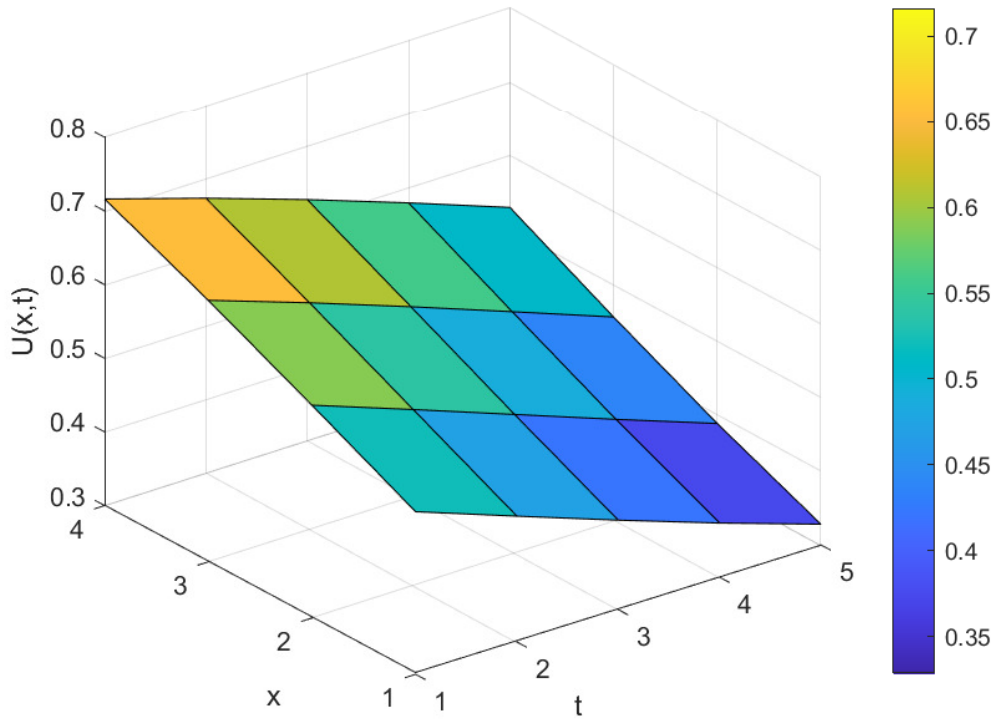
Figure 2: The best costs of ITLBO algorithm with and without guessing type of unknown function.



(a) The answer of Huxley problem $U(x, t)$ (exact).



(b) The answer of Huxley problem $U(x, t)$ using guess type of $P(t)$.

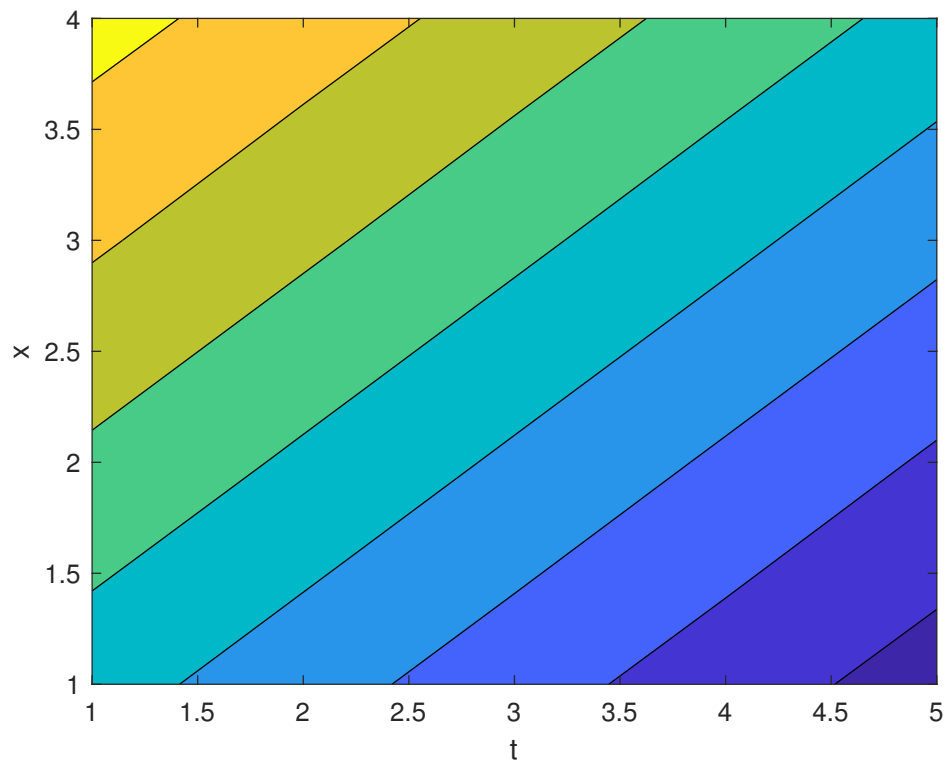


(c) The answer of Huxley problem $U(x, t)$ without guess type of $P(t)$.

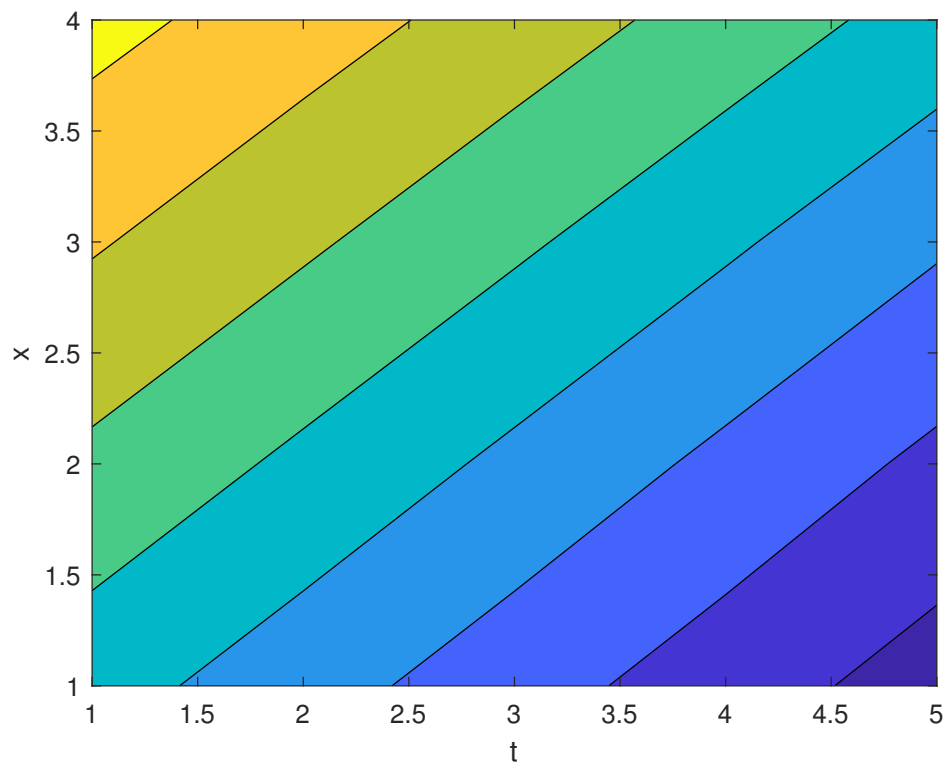
Figure 3: The comparison between the exact $U(x, t)$ and $U(x, t)$ obtained from methods.

Table 1: The numerical solution of the Huxley problem $U(x, t)$ using two methods compared with the exact solution.

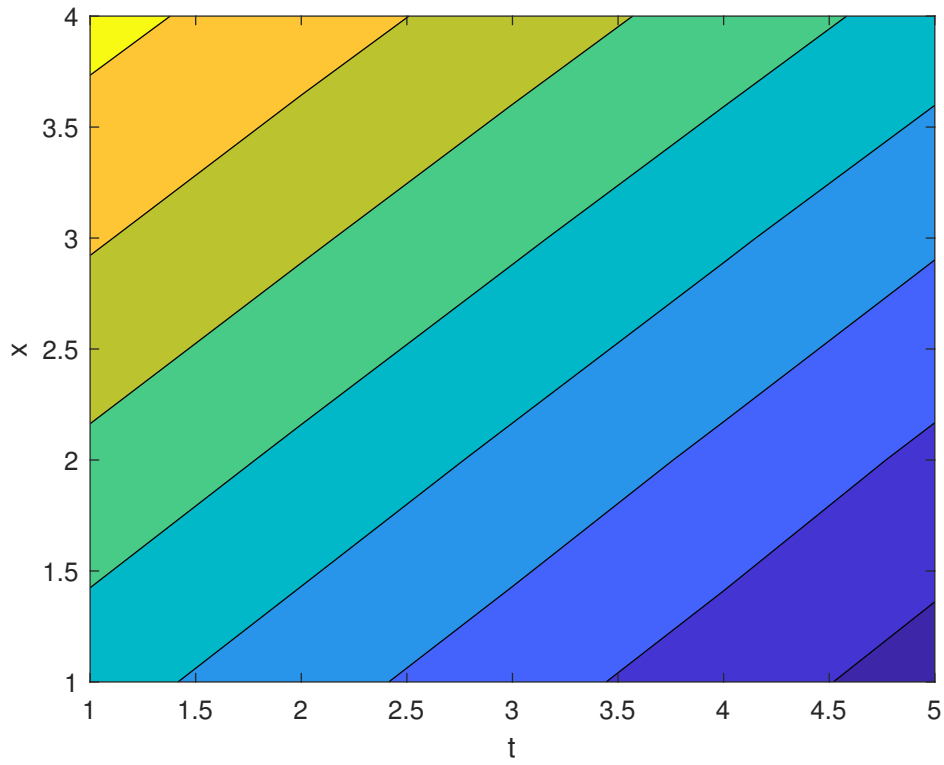
Method	$U(x_i, t_j)$					
	Location \ Time	0.2	0.4	0.6	0.8	1
With guess	0.2	0.5210	0.4704	0.4210	0.3735	0.3282
	0.4	0.5892	0.5390	0.4883	0.4380	0.3885
	0.6	0.6552	0.6077	0.5583	0.5077	0.4567
	0.8	0.7163	0.6734	0.6275	0.5792	0.5290
Without guess	0.2	0.5207	0.4707	0.4214	0.3735	0.3280
	0.4	0.5890	0.5392	0.4885	0.4380	0.3885
	0.6	0.6550	0.6078	0.5584	0.5078	0.4566
	0.8	0.7162	0.6734	0.6276	0.5792	0.5291
Exact	Location \ Time	0.2	0.4	0.6	0.8	1
	0.2	0.5207	0.4707	0.4214	0.3735	0.3280
	0.4	0.5904	0.5413	0.4914	0.4416	0.3930
	0.6	0.6567	0.6102	0.5618	0.5121	0.4622
	0.8	0.7173	0.6751	0.6298	0.5821	0.5327



(a) The contour answer of Huxley problem $U(x, t)$.



(b) The contour answer of Huxley problem $U(x, t)$ using guess type of $P(t)$.



(c) The contour answer of Huxley problem $U(x, t)$ without guess type of $P(t)$.

Figure 4: Contour plot of exact $U(x, t)$ and $U(x, t)$ obtained from methods.

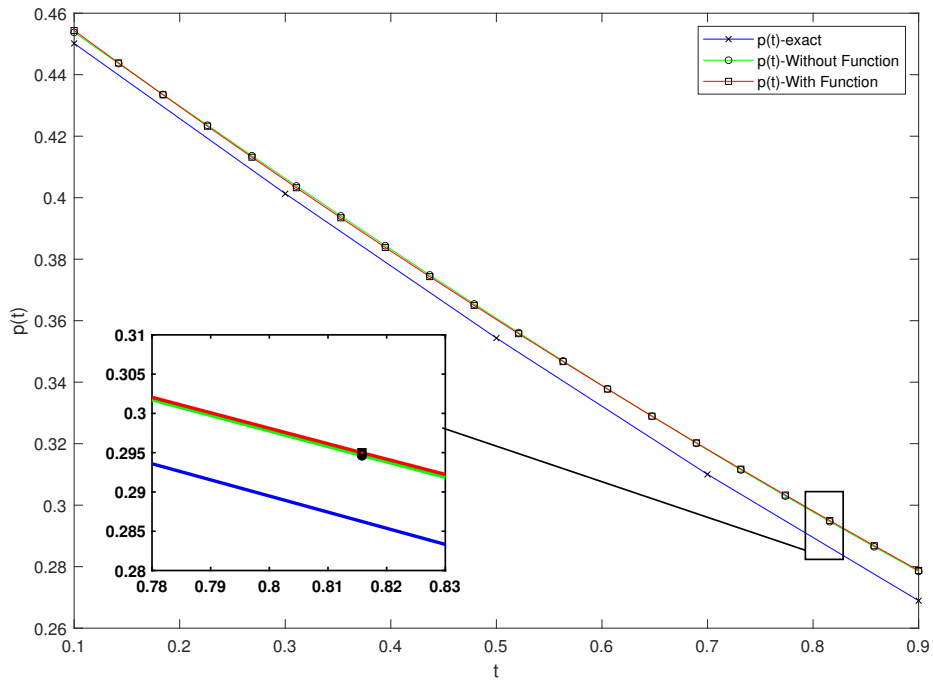


Figure 5: The comparison between exact $P(t)$ and $P(t)$ obtained from methods.

Table 2: The numerical value of obtained $P(t)$ using two methods compared with the exact $P(t)$.

Method	P_i				
With guess	P_1 0.4544	P_2 0.4057	P_3 0.3604	P_4 0.3181	P_5 0.2788
Without guess	P_1 0.4537	P_2 0.4062	P_3 0.3607	P_4 0.3179	P_5 0.2784
Exact	P_1 0.4502	P_2 0.4013	P_3 0.3543	P_4 0.3100	P_5 0.2689

Table 3: The results of implementing methods with ITLBO algorithm.

Method	Population	Iteration	Interval	Best Cost	S
With guess	60	80	[-5 5]	3.4185e-07	8.8523e-3
Without guess	60	80	[-10 10]	3.0815e-33	8.7900e-3

the type of the unknown function of the problem, but due to the fact that in some problems we do not know the type of boundary and initial functions of the problem, the results of the presented method without using guess the type of the unknown function of the problem are also very good and acceptable, and can be very effective for solving these problems and especially inverse Huxley problems which are nonlinear and very ill-posed. Also, the values of the cost function show that the ITLBO algorithm has a very good ability to optimize these values in the additional condition of the problem, and it can be used well to obtain the unknown boundary conditions.

6 Conclusion

Using ITLBO algorithm and the presented numerical method, the unknown boundary condition was obtained in the inverse Huxley PDE, and the following results have been obtained:

1. Using the method presented, unknown $p(t)$ can be obtained without guessing the type of unknown function in the Huxley inverse problem. Numerical results show that this value is very accurate and close to the actual value.
2. With the presented method, it is possible to obtain the answer of the problem both by guessing the type of the unknown function of the inverse problem and without guessing the type of the function, but in some problems there is no information about the type of functions of the boundary and initial conditions. This method can be very practical and useful in solving this category of problems, especially in the Huxley problem, which is a non-linear problem with forms of power 3.
3. Huxley's problem is a nonlinear problem with nonlinear forms of the power of 3. The answers obtained from the implementation of the mentioned method showed that this method can solve even the inverse type of these problems which are very ill-posed and get accurate answers. The answer to the problem $U(x, t)$ obtained using the unknown condition, is very accurate and close to the actual answer to the problem, which indicates the efficiency and convergence of this method.
4. The results of the implementation of ITLBO Algorithm show that this algorithm escapes from local optimizations and always tries to improve the fitness function defined for it.
5. The numerical method presented using the combination of implicit and explicit methods can provide very accurate answers to the problem when the inverse problem becomes a direct problem or in general for direct problems.
6. Given the convergence to the answer in this method and the efficiency of ITLBO, even by increasing the search range to find the unknown boundary condition, the exact answer to the boundary condition can be reached and it can be found.
7. The results were obtained within minutes and processed on a single-core 2.5 GHz CPU.

References

- [1] A. Aliyari Boroujeni, R. Pourgholi, and S.H. Tabasi, *A new improved teaching-learning-based optimization (ITLBO) algorithm for solving nonlinear inverse partial differential equation problems*, *Comp. Appl. Math.* **42** (2023), 99.
- [2] A. Aliyari Boroujeni, R. Pourgholi, and S.H. Tabasi, *Solving inverse partial differential equations problems by using teaching learning based optimization algorithm*, *TWMS J. Appl. Eng. Math.* (2024).
- [3] A. Aliyari Boroujeni, R. Pourgholi, and S.H. Tabasi, *Numerical solutions of KDV and mKDV equations: Using sequence and multi-core parallelization implementation*, *J. Comput. Appl. Math.* **454** (2025), 116184.
- [4] R.S. Anderssen, *Inverse problems: a pragmatist's approach to the recovery of information from indirect measurements*, *Aust. N. Z. Ind. Appl. Math. J.* **46** (2005), C588–C622.
- [5] D.G. Aronson and H.F. Weinberger, *Multidimensional nonlinear diffusion arising in population genetics*, *Adv. Math.* **30** (1978), no. 1, 33–76.
- [6] E. Babolian and J. Saeidian, *Analytic approximate solutions to Burgers, Fisher, Huxley equations and two combined forms of these equations*, *Commun. Nonlinear Sci. Numer. Simul.* **14** (2009), 1984–1992.
- [7] Y. Bai, T. Chaolu, and S. Bilige, *Solving Huxley equation using an improved PINN method*, *Nonlinear Dyn.* **105** (2021), 3439–3450.
- [8] B. Batiha, M.S.M. Noorani, and I. Hashim, *Numerical simulation of the generalized Huxley equation by He's variational iteration method*, *Appl. Math. Comput.* **186** (2007), no. 2, 1322–1325.
- [9] J.V. Beck and D.C. Murio, *Combined function specification-regularization procedure for solution of inverse heat conduction problem*, *AIAA J.* **24** (1986), 180–185.
- [10] J.M.G. Cabeza, J.A.M. Garcia, and A.C. Rodriguez, *A sequential algorithm of inverse heat conduction problems using singular value decomposition*, *Int. J. Therm. Sci.* **44** (2005), 235–244.
- [11] H. Dong, Y. Xu, D. Cao, W. Zhang, Z. Yang, and X. Li, *An improved teaching-learning-based optimization algorithm with a modified learner phase and a new mutation-restarting phase*, *Knowl.-Based Syst.* **258** (2022), 109989.
- [12] L. Duan and Q. Lu, *Bursting oscillations near codimension-two bifurcations in the Chay Neuron model*, *Int. J. Nonlinear Sci. Numer. Simul.* **7** (2006), no. 1, 59–64.
- [13] S. Foadian, R. Pourgholi, and S.H. Tabasi, *Cubic B-spline method for the solution of an inverse parabolic system*, *Appl. Anal.* **97** (2017), no. 3, 438–465.
- [14] S. Foadian, R. Pourgholi, S.H. Tabasi, and J. Damirchi, *The inverse solution of the coupled nonlinear reaction-diffusion equations by the Haar wavelets*, *Int. J. Comput. Math.* **96** (2019), no. 1, 105–125.
- [15] S. Foadian, R. Pourgholi, and A. Esfahani, *Numerical solution of the linear inverse wave equation*, *Int. J. Nonlinear Anal. Appl.* **13** (2022), no. 2, 1907–1926.
- [16] S. Foadian, R. Pourgholi, and A. Esfahani, *Numerical solution of the linear inverse wave equation*, *Int. J. Nonlinear Anal. Appl.* **13** (2022), no. 2, 1907–1926.
- [17] A.L. Hodgkin and A.F. Huxley, *A Quantitative description of ion currents and its applications to conduction and excitation in nerve membranes*, *J. Physiol.* **117** (1952), 500–544.
- [18] M. Javidi and A. Golbabai, *A new domain decomposition algorithm for generalized Burger's-Huxley equation based on Chebyshev polynomials and preconditioning*, *Chaos Solitons Fractals* **39** (2009), no. 2, 849–857.
- [19] Z. Jiang, F. Zou, D. Chen, S. Cao, H. Liu, and W. Guo, *An ensemble multi-swarm teaching-learning-based optimization algorithm for function optimization and image segmentation*, *Appl. Soft Comput.* **130** (2022), 109653.
- [20] S. Liu, T. Fan, and Q. Lu, *The spike order of the winnerless competition (WLC) model and its application to the inhibition neural system*, *Int. J. Nonlinear Sci. Numer. Simul.* **6** (2005), no. 2, 133–138.
- [21] L.B. Liu, Y. Liang, J. Zhang, and X. Bao, *A robust adaptive grid method for singularly perturbed Burger-Huxley equations*, *Electron. Res. Arch.* **28** (2020), no. 4, 1439–1457.

- [22] R. Mohammadi, *B-spline collocation algorithm for numerical solution of the generalized Burger's-Huxley equation*, Numer. Methods Partial Differ. Equ. **29** (2013), no. 4, 1173–1191.
- [23] R.K. Mohanty, W. Dai, and D. Liu, *Operator compact method of accuracy two in time and four in space for the solution of time dependent Burgers-Huxley equation*, Numer. Algorithms **70** (2015), no. 3, 591–605.
- [24] R. Pourgholi, S.H. Tabasi, and H. Zeidabadi, *Numerical techniques for solving system of nonlinear inverse problem*, Eng. Comput. **34** (2018), 487–502.
- [25] R.V. Rao, V.J. Savsani, and D.P. Vakharia, *Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems*, Comput.-Aided Des. **43** (2011), 303–315.
- [26] R.V. Rao, V.J. Savsani, and D.P. Vakharia, *Teaching learning-based optimization: an optimization method for continuous non-linear large scale problems*, Inf. Sci. **183** (2012), 1–15.
- [27] R.V. Rao and V. Patel, *An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems*, Sci. Iran. **20** (2013), no. 3, 710–720.
- [28] A. Saeedi, S. Foadian, and R. Pourgholi, *Applications of two numerical methods for solving inverse Benjamin–Bona–Mahony–Burgers equation*, Eng. Comput. **36** (2020), no. 4, 1453–1466.
- [29] C.W. Sahabandu, D. Karunarathna, P. Sewvandi, et al., *A method of directly defining the inverse mapping for a nonlinear partial differential equation and for systems of nonlinear partial differential equations*, Comput. Appl. Math. **40** (2021), 234.
- [30] J. Satsuma, *Exact Solutions of Burgers' Equation with Reaction Terms*, in Topics in Soliton Theory and Exact Solvable Nonlinear Equations, 1987.
- [31] G.D. Smith, *Numerical Solution of Partial Differential Equations*, Oxford University Press, New York, 1965.
- [32] X.Y. Wang, Z.S. Zhu, and Y.K. Lu, *Solitary wave solutions of the generalised Burgers-Huxley equation*, J. Phys. A: Math. Gen. **23** (1990), no. 3, 271–274.
- [33] A.M. Wazwaz, *Travelling wave solutions of generalized forms of Burgers, Burgers-KdV and Burgers-Huxley equations*, Appl. Math. Comput. **169** (2005), no. 1, 639–656.
- [34] G.J. Zhang, J.X. Xu, H. Yao, and R.X. Wei, *Mechanism of bifurcation-dependent coherence resonance of an excitable neuron model*, Int. J. Nonlinear Sci. Numer. Simul. **7** (2006), no. 4, 447–450.