Research Article

Mathematical Methods in the Applied Sciences

Received XXXX

(www.interscience.wiley.com) DOI: 10.1002/sim.0000
MOS subject classification: 34B15; 37M15 ; 65D05; 65L60; 65K05

A numerical scheme based on the collocation and optimization methods for accurate solution of sensitive boundary value problems

M. A. Mehrpouya*, **R. Salehi**

Despite the significant advances in the numerical solution of nonlinear boundary value problems, most of the existing methods still encounter with a high sensitivity to the initial guess. The aim of this paper is to propose a less sensitive robust numerical scheme for accurate solution of sensitive boundary value problems. For this purpose, an orthogonal collocation approach for discretization of the problem is utilized. Thereby, the problem is converted to the solution of nonlinear algebraic equations. However, due to the difficulties of solving the obtained system of nonlinear equations, particularly in providing the proper initial guess, the obtained system of equations is transferred to an optimization problem in which the values of the solution at collocation points are considered as decision parameters. The method finds good results even using not good initial guess for decision parameters and even using a small number of discretization points. Numerical results of two benchmark examples are presented and the efficiency of the method is reported.

Keywords: Boundary value problem; Numerical approximation; Interpolation; Orthogonal collocation; Optimization

1. Introduction

The nonlinear boundary value problem, in this paper, is considered of the form

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t)), \quad t_0 \leq t \leq t_f, \tag{1}$$

or could be transformed into this form, together with the boundary conditions

$$\psi(\mathbf{y}(t_0), \mathbf{y}(t_f)) = \mathbf{0}, \tag{2}$$

where, $\mathbf{y} = [y_1, \ldots, y_p]^T : [t_0, t_f] \to \mathbb{R}^p$, $\mathbf{f} = [f_1, \ldots, f_p]^T : [t_0, t_f] \times \mathbb{R}^p \to \mathbb{R}^p$, and $\psi : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^p$. Obviously, Eq. (1), is a system of first-order ordinary differential equations. Boundary value problems are the large class of problems that are arisen in a variety of application areas, such as, engineering, biology, applied mathematics and mathematical modeling of a lot of real world problems. Because of the complexity in most applications, boundary value problems are most often solved numerically.

So, different kinds of methods and softwares for numerical solution of boundary value problems have appeared. For instances, we can refer to the simple and multiple shooting methods, finite difference and various weighted residual methods [1, 2, 3]. According to our knowledge, almost all of these methods reduce the solution of the boundary value problem to the solution of a system of algebraic equations which should be solved by the root finding methods such as Newton's method. So, convergence to the optimal solution can suffer from the extreme sensitivity of Newton's or Newton-like methods, because of the lack of the good initial guesses for the solution [4]. Especially, if the problem (1) and (2) is unstable, small changes in the initial conditions lead to the large changes in the solution. Consequently, solving the problem numerically, is very difficult, require many iterations and often impossible [5].

Department of Applied Mathematics, Faculty of Mathematical Sciences, Tarbiat Modares University, P.O. Box 14115-134, Tehran, Iran. E-mail: r.salehi@modares.ac.ir
**Correspondence to: Mohammad Ali Mehrpouya, Department of Mathematics, Tafresh University, 39518-79611, Tafresh, Iran. E-mail: mehrpouya@tafreshu.ac.ir*

The aim of this paper is to present a robust numerical method with less sensitivity to the initial guess, for efficient solution of such sensitive boundary value problems. For this purpose, using an orthogonal collocation approach for discretization, the problem is transferred to an optimization problem. Orthogonal collocation methods which are usually known as pseudospectral methods, are a family of spectral methods which classically were applied in fluid dynamics [6, 7]. The main advantage of them is that, they have faster convergence rates than other methods and are easy to implement [8]. Orthogonal collocation methods mainly comprehend two stages which is led to a stable numerical approximation corresponding to the solution of differential equations. The first stage is choosing a proper finite representation of the solution which is usually done by polynomial interpolation of the solution based on some suitable points [9]. The second stage is discretizing the equations at the collocation points and finally obtaining the values of unknown solution at these points in a root finding problem.

In this paper, an orthogonal collocation approach using the Legendre-Gauss-Radau (LGR) points for discretization is utilized in the first stage. In the second stage, against the other traditional collocation methods [10, 11, 12, 13, 14, 15], by considering the values of the solution at the LGR points as decision parameters, the problem is transferred to an optimization problem. Indeed, instead of confronting a root finding problem, an optimization problem is met in the second stage. Because the numerical solution of the obtained optimization problem does not need a good initial guess to start, so, its solution is much easier than solving the sensitive root finding problem which is obtained from the second stage of the traditional collocation methods [16]. This leads, the proposed method is robust and more suitable and can be more easily applicable from the other traditional collocation methods.

2. The proposed method

In order to approximate the solution of the system of equations (1) and (2), an orthogonal collocation approach is applied. Let

$$\hat{\xi}_j = ((t_0+t_f)\xi_j + (t_0+t_f))/2, \quad j=1,\dots,n,$$

for n LGR points $\{\xi_j\}_{j=1}^{n}$, where $\xi_1=-1$ and $\xi_n<+1$, be the associated LGR points to the interval (t_0,t_f). It is noted that, the LGR points $\{\xi_j\}_{j=1}^{n}$ is the set of n roots of the sum of the Legendre polynomials of degree n and $n-1$, $P_n(t)+P_{n-1}(t)$, that are given by the following recurrence relation [6]

$$\begin{aligned} &P_0(t)=1, \quad P_1(t)=t,\\ &P_{n+1}(t)=\frac{2n+1}{n+1}tP_n(t)-\frac{n}{n+1}P_{n-1}(t), \quad n=1,2,\dots \end{aligned}$$

By adding the point $\hat{\xi}_{n+1}=t_f$, a set of $n+1$ points $\{\hat{\xi}_1=t_0,\hat{\xi}_1,\dots,\hat{\xi}_{n+1}=t_f\}$ is constituted which are used for approximating $\mathbf{y}(t)$ as

$$\mathbf{y}(t)\simeq\hat{\mathbf{y}}_n(t)=\sum_{k=1}^{n+1}\hat{\mathbf{c}}_k\hat{\varphi}_k(t), \tag{3}$$

in which, $\hat{\mathbf{c}}_k=[\hat{c}_{1,k},\dots,\hat{c}_{p,k}]^T\simeq\mathbf{y}(\hat{\xi}_k)=[y_1(\hat{\xi}_k),\dots,y_p(\hat{\xi}_k)]^T$, and $\{\hat{\varphi}_k(t)\}_{k=1}^{n+1}$ are Lagrange polynomials associated with $\hat{\xi}_k$ or shifted Lagrange polynomials on (t_0,t_f), with

$$\hat{\varphi}_k(t)=\varphi_k((2/(t_f-t_0))t-(t_f+t_0)/(t_f-t_0)),$$

and $\{\varphi_k(t)\}_{k=1}^{n+1}$ are the nth-order Lagrange interpolating polynomials, defined as

$$\varphi_k(t)=\prod_{l=1,l\neq k}^{n+1}\frac{t-\xi_l}{\xi_k-\xi_l}, \quad k=1,\dots,n+1,$$

with the Kronecker property

$$\varphi_k(\xi_l)=\delta_{kl}=\begin{cases}0, & \text{if } k\neq l,\\ 1, & \text{if } k=l.\end{cases} \tag{4}$$

Note that, $\hat{\xi}_{n+1}=t_f$ is not a shifted LGR point but it is used in defining of Lagrange polynomials $\hat{\varphi}_k(t)$. Obviously, from Eq. (3), and by using the Kronecker property (4), we get

$$\begin{aligned} &\mathbf{y}(\hat{\xi}_j)\simeq\hat{\mathbf{y}}_n(\hat{\xi}_j)=\sum_{k=1}^{n+1}\hat{\mathbf{c}}_k\hat{\varphi}_k(\hat{\xi}_j)=\hat{\mathbf{c}}_j, \quad j=1,\dots,n,\\ &\mathbf{y}(\hat{\xi}_{n+1}=t_f)\simeq\hat{\mathbf{y}}_n(\hat{\xi}_{n+1}=t_f)=\sum_{k=1}^{n+1}\hat{\mathbf{c}}_k\hat{\varphi}_k(\hat{\xi}_{n+1})=\hat{\mathbf{c}}_{n+1}. \end{aligned}$$

Using Eq. (3), $\dot{\mathbf{y}}$ can be approximated as

$$\dot{\mathbf{y}}(t) \simeq \dot{\hat{\mathbf{y}}}_n(t) = \sum_{k=1}^{n+1} \hat{\mathbf{c}}_k \dot{\hat{\varphi}}_k(t). \tag{5}$$

Consequently, substituting the Eqs. (3) and (5) into the Eq. (1), next collocating in $\hat{\xi}_j$, $j = 1, \dots, n$, we get

$$\sum_{k=1}^{n+1} \hat{\mathbf{c}}_k \dot{\hat{\varphi}}_k(\hat{\xi}_j) \ \simeq \ \mathbf{f}(\hat{\xi}_j, \sum_{k=1}^{n+1} \hat{\mathbf{c}}_k \hat{\varphi}_k(\hat{\xi}_j)), \quad j = 1, \dots, n. \tag{6}$$

So, the Eq. (6) is reduced to

$$\frac{2}{t_f - t_0} \sum_{k=1}^{n+1} \hat{\mathbf{c}}_k d_{kj} \simeq \mathbf{f}(\hat{\xi}_j, \hat{\mathbf{c}}_j), \quad j = 1, \dots, n, \tag{7}$$

where

$$d_{kj} = \dot{\varphi}_k(\xi_j),$$

is the (k, j)-th element of the matrix $\mathbf{D}$, which is named as the differentiation matrix [7]. It is noted that, the Eqs. (7) provide np algebraic equations which is demanded to find the unknown vectors $\hat{\mathbf{c}}_k$, $k = 1, \dots, n+1$, and of course p another algebraic equations should be obtained from the boundary conditions. So, the boundary conditions (2) can be discretized as

$$\psi(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_{n+1}) = \mathbf{0}. \tag{8}$$

Obviously, the Eqs. (7) and (8) are a nonlinear system of algebraic equations for finding the desired $\hat{\mathbf{c}}_k$, $k = 1, \dots, n+1$, as follows

$$\begin{cases} \frac{2}{t_f - t_0} \sum_{k=1}^{n+1} \hat{\mathbf{c}}_k d_{kj} \simeq \mathbf{f}(\hat{\xi}_j, \hat{\mathbf{c}}_j), \quad j = 1, \dots, n, \\ \psi(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_{n+1}) = \mathbf{0}. \end{cases} \tag{9}$$

Then, one can provide the approximate solution of the system of boundary value problems (1) and (2) with

$$\hat{\mathbf{y}}_n(t) = \big[\sum_{k=1}^{n+1} \hat{c}_{1,k} \hat{\varphi}_k(t), \dots, \sum_{k=1}^{n+1} \hat{c}_{p,k} \hat{\varphi}_k(t)\big]^T.$$

It is noted that, solving the system of equations (9) with root finding methods such as Newton's ones may be difficult, as mentioned in the previous section. So, in this paper, instead of confronting the system of equations (9), a different strategy is utilized. This strategy is based on transferring (9) to the optimization problem. It should be noted that, this strategy is formed in two different ways in this paper.

2.1. Transferring the system of equations to a constrained optimization problem

To this aim, with considering the discretized boundary conditions in Eq. (8) in the form of

$$J = \big(\psi(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_{n+1})\big)^2,$$

as an objective function and considering the Eqs. (7) as the constraints, the system of equations (9) is finally transferred to the following constrained optimization problem

$$\begin{cases} \textbf{Min} \ \ J = \big(\psi(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_{n+1})\big)^2, \\ \textbf{S.t.} \\ \qquad \frac{2}{t_f - t_0} \sum_{k=1}^{n+1} \hat{\mathbf{c}}_k d_{kj} \simeq \mathbf{f}(\hat{\xi}_j, \hat{\mathbf{c}}_j), \quad j = 1, \dots, n. \end{cases} \tag{10}$$

This strategy is named as the strategy (I), in this paper. Noted that, the constrained optimization problem (10) can be solved by any well-developed optimization algorithm [17]. In this paper, the Matlab function `fmincon` is used and we set this solver to use the `interior-point` numerical algorithm. In this solver, we can specify the values of termination tolerance on the objective function value, `TolFun`, tolerance on the constraint violation, `TolCon`, and termination tolerance on decision variables, `TolX`, are less than 10^{-10}.

2.2. Transferring the system of equations to an unconstrained optimization problem

For this purpose, considering the system of equations (9) in the form of

$$\textbf{Min} \ \ J = \big(\frac{2}{t_f - t_0} \sum_{k=1}^{n+1} \hat{\mathbf{c}}_k d_{kj} - \mathbf{f}(\hat{\xi}_j, \hat{\mathbf{c}}_j)\big)^2 + \big(\psi(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_{n+1})\big)^2, \tag{11}$$

the problem (9) is finally transferred to an unconstrained optimization problem (11). This strategy is also named as the strategy (II), in this paper. Here, the unconstrained optimization problem (11) is solved by the Matlab function `fminunc` in which the `trust-region` numerical algorithm is used in this solver. In `fminunc`, we can specify the values of termination tolerance on the objective function value, `FunctionTolerance`, termination tolerance on the first-order optimality, `OptimalityTolerance`, and termination tolerance on decision variables, `StepTolerance`, are less than 10^{-10}. In the next section, we will examine the performance of both strategies in implementing the proposed method.

3. Illustrative examples

In this section, two nontrivial examples are given to demonstrate the applicability of our method. In the first example, the robustness of the method in solving a benchmark example from boundary layer theory compared to the two famous and common methods in solving boundary value problems, is highlighted. One of these famous and common methods is the multiple shooting method and the other is a finite difference method that utilizes the four-stage Lobatto formula implemented by means of the Matlab function `bvp5c` [5], where the relative and absolute error tolerances are set to be 10^{-10} for both. It is noted that, in the multiple shooting method, the origin interval of the problem is divided into M subintervals and in i-th subinterval, an initial value problem is met [18]. So, the solution of the boundary value problem is finally reduced to the solution of nonlinear equations which are named as shooting equations. It is worthwhile to note that, the shooting equations obtained from the multiple shooting method in this paper, are solved by the Matlab function `fsolve` and two parameters `TolFun` and `TolX` of this solver, in which, the former specifies the termination tolerance on function value and the latter specifies the termination tolerance on the parameter value are set to be 10^{-10}. In the second example, the proposed method will be faced with a benchmark example from a family of Sturm-Liouville boundary value problems. It is necessary to mention that, the first example has no analytical solution while the second example has an analytical solution. Therefore, using the Euclidian norm of error at collocation points in the second example as

$$\mathbf{E}_n = \left(\sum_{k=1}^{n+1} [\hat{\mathbf{y}}(\hat{\xi}_k) - \mathbf{y}^*(\hat{\xi}_k)]^2 \right)^{1/2},$$

where, $\hat{\mathbf{y}}$ is the approximated solution and $\mathbf{y}^*$ is the exact solution, an overview of the rate of convergence by plotting $\mathbf{E}_n$ appears. Noted that, all computations are performed on a 2.53 GHz Core i5 PC Laptop with 4 GB of RAM running in Matlab R2016a.

3.1. Example 1.

Consider the nonlinear boundary value problem

$$\begin{cases} \dot{y}_1(t) = y_2(t), \\ \dot{y}_2(t) = y_3(t), \\ \dot{y}_3(t) = -\frac{1}{2}(3-m)y_1(t)y_3(t) - my_2^2(t) - y_4^2(t) + sy_2(t) + 1, \\ \dot{y}_4(t) = y_5(t), \\ \dot{y}_5(t) = -\frac{1}{2}(3-m)y_1(t)y_5(t) - (m-1)y_2(t)y_4(t) + s(y_4(t) - 1), \\ y_1(0) = y_2(0) = y_4(0) = 0, \\ y_2(t_f) = 0, \quad y_4(t_f) = 1, \end{cases}$$

which is related to the theory of the boundary layer of an incompressible fluid [19]. It is known that, for sufficiently large values of t_f, this problem is hyper-sensitive. Therefore, finding it's solution is very difficult. In this paper, we consider $t_f = 20$. Now, the problem is solved using the proposed method. The approximated solution for $m = -0.1$, $s = 0.2$ and $n = 30$ discretization points, is shown in Fig. 1. Also, to show the numerical convergence and accuracy of the method, different values of $\mathbf{y}$ for some discretization points are shown in Table 1. Because this problem has no analytical solution, so it's reference solution can be obtained either with the multiple shooting method or by means of the Matlab function `bvp5c` and the results are shown in Table 2. Comparing Tables 1 and 2, it can be concluded that the accuracy of the proposed method is high and the convergence is achieved by using a small number of discretization points. In addition, it can be seen that, the performance of the proposed method with the strategy (I) is much more effective than using the strategy (II) due to its low cpu time.

Now, in order to show the robustness of the proposed method, a comparison is made between the sensitivity of the presented method and two common multiple shooting and `bvp5c` methods, which in both of them, the solution of the problem is reduced to the solution of nonlinear system of algebraic equations [1, 5]. So, the proposed method, the multiple shooting and `bvp5c` methods are separately utilized for 100 initial values randomized distributed in the interval $[0, 1]$, and the average of the CPU time of each method is also reported. The results are shown in Table 3. According to Table 3, both `bvp5c` and the multiple shooting methods are more sensitive to the initial guess than the proposed method. However, the robustness of the proposed method besides to its low cpu time can be clearly seen in this table.

3.2. Example 2.

The second example is the Bratu boundary value problem, a class of nonlinear eigenvalue problems which arises in the model of thermal reaction process and the Chandrasekhar model of the expansion of the universe [20]. The Bratu boundary value problem

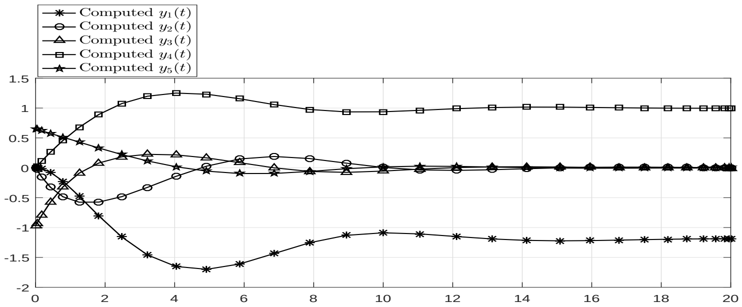

Figure 1. Approximated Solution of Example 1 for $n = 30$.

Table 1. The initial and final values of $y_3(t)$ and $y_5(t)$ for different number of discretization points obtained by the presented method in Example 1.

		Present method with strategy (I)			Present method with strategy (II)		
t	n	$y_3(t)$	$y_5(t)$	CPU Time(s)	$y_3(t)$	$y_5(t)$	CPU Time(s)
0	20	-0.96631801	0.65290641	1.78	-0.96631801	0.65290641	36.28
	25	-0.96631161	0.65290974	2.64	-0.96631161	0.65290974	60.51
	30	-0.96631181	0.65290957	3.22	-0.96631180	0.65290958	97.23
	35	-0.96631181	0.65290957	4.44	-0.96631180	0.65290958	122.83
20	20	-0.00867774	0.00879331	1.78	-0.00867774	0.00879332	36.28
	25	-0.00867617	0.00879305	2.64	-0.00867617	0.00879307	60.51
	30	-0.00867621	0.00879306	3.22	-0.00867621	0.00879307	97.23
	35	-0.00867622	0.00879307	4.44	-0.00867621	0.00879307	122.83

Table 2. The initial and final values of $y_3(t)$ and $y_5(t)$ obtained by bvp5c and multiple shooting method in Example 1.

	$y_3(0)$	$y_5(0)$	$y_3(20)$	$y_5(20)$
Solution by bvp5c	-0.96631180	0.65290958	-0.00867621	0.00879307
Solution by the multiple shooting method with $M = 5$	-0.96631180	0.65290958	-0.00867621	0.00879307
Solution by the multiple shooting method with $M = 10$	-0.96631180	0.65290958	-0.00867621	0.00879307

Table 3. The convergence percentage of bvp5c, the multiple shooting method (with $M = 5, 10$) and the proposed method (with $n = 15, 20$) for 100 random initial guess in the the interval $[0, 1]$, in Example 1.

	bvp5c	The multiple shooting method		The proposed method with strategy (I)	
		$M = 5$	$M = 10$	$n = 15$	$n = 20$
$[0, 1]$	14	58	65	100	100
The average of CPU time(s)	2.35	Over 1000		1.63	

has the form

$$\ddot{y}(x) + \lambda e^{y(x)} = 0, \quad y(0) = y(1) = 0, \qquad 0 \le x \le 1,$$

where, $\lambda > 0$ is a constant. It is simple to show that, by using a substitution

$$\begin{aligned} y_1 &= y, \\ y_2 &= \dot{y}, \end{aligned}$$

the order of the problem is reduced to one and the following system of first-order boundary value problems is derived

$$\begin{cases} \dot{y}_1(x) = y_2(x), \\ \dot{y}_2(x) = -\lambda e^{y_1(x)}, \\ y_1(0) = y_1(1) = 0. \end{cases}$$

This problem has the exact solution, $y_1^*(x) = -2\ln\left[\frac{\cosh((x-\frac{1}{2})\frac{\theta}{2})}{\cosh(\frac{\theta}{4})}\right]$ and $y_2^*(x) = -\frac{\theta\sinh((x-\frac{1}{2})\frac{\theta}{2})}{\cosh((x-\frac{1}{2})\frac{\theta}{2})}$, where θ is the solution of the equation

$$\theta = \sqrt{2\lambda}\cosh(\frac{\theta}{4}). \tag{12}$$

Now, the problem is solved using the proposed method, when λ is considered as 2. It is noted that in this situation, the Eq. (12) has two roots, i.e., $\theta = 2.35755105385$ and $\theta = 8.50719957071$. The exact, approximated and error functions, for the various

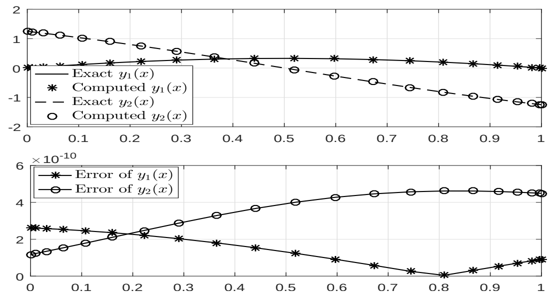

Figure 2. Solution of Example 2 with strategy (I) for $n = 20$, $\lambda = 2$ and $\theta = 2.35755105385$.

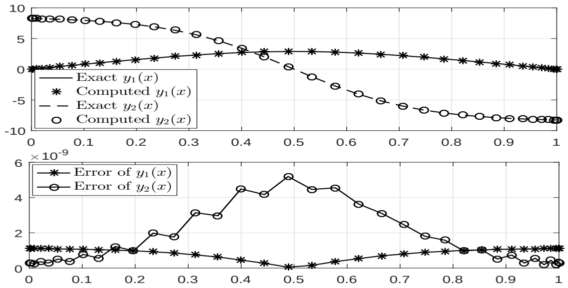

Figure 3. Solution of Example 2 with strategy (I) for $n = 35$, $\lambda = 2$ and $\theta = 8.50719957071$.

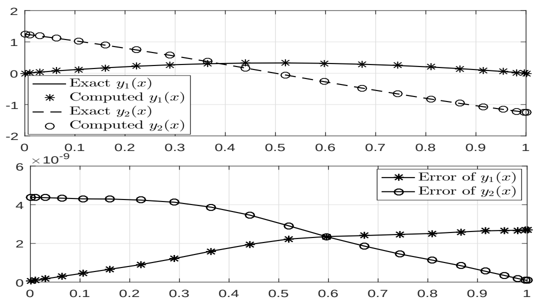

Figure 4. Solution of Example 2 with strategy (II) for $n = 20$, $\lambda = 2$ and $\theta = 2.35755105385$.

discretization points, are shown in Figs. 2-5. Also, for exploring the dependence of error of the solutions on the parameter n, the proposed method is applied on this problem for various values of n and the results are shown in Figs. 6 and 7. Furthermore, in Table 4, the absolute error of the proposed method when using $n = 20$ discretization points and with strategy (I) in comparison with five other numerical methods is given. The results show that, the proposed method approximates the exact solution with the high precision. Finally, to show the robustness of the proposed method, once, it is used in a situation that it does not utilize the strategies (I) or (II) and only solves the system of algebraic equations (9), and again when the proposed method uses the strategy (I). Noted that, for solving the system of algebraic equations (9) in this paper, the Matlab function `fsolve` is utilized and we have considered the Bratu boundary value problem with $\lambda = 2$ and $\theta = 8.50719957071$. The results are shown in Fig. 8. It is clear that, by utilizing the optimization strategy, the CPU time, the number of iterations and the number of function evaluation is significantly reduced.

4. Conclusion

In the present work, an accurate and robust numerical method was applied for efficient solution of the first-order system of nonlinear sensitive boundary value problems. The method reduced the solution of the problem to the solution of an optimization problem which is easier to solve than the sensitive root finding problem appears in the traditional collocation methods. The proposed method is especially useful where the classical methods fail for sensitivity or stiffness of the problem. Illustrative examples were given to demonstrate the validity and applicability of the proposed method. It was seen that, good results are obtained even using a small number of discretization points, the rate of convergence is high and the proposed method did not

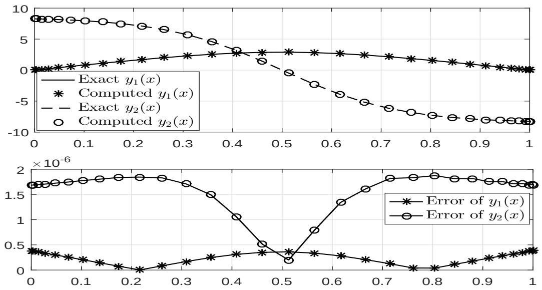

Figure 5. Solution of Example 2 with strategy (II) for $n = 30$, $\lambda = 2$ and $\theta = 8.50719957071$.

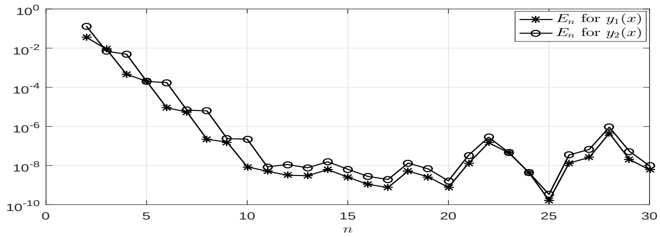

Figure 6. Euclidian norm of error versus n with strategy (I) for $\lambda = 2$ and $\theta = 2.35755105385$, in Example 2.

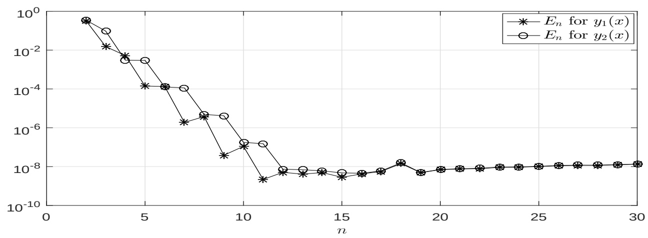

Figure 7. Euclidian norm of error versus n with strategy (II) for $\lambda = 2$ and $\theta = 2.35755105385$, in Example 2.

Table 4. The absolute error for $\lambda = 2$ and $\theta = 2.35755105385$, in Example 2.

t	Method of [21]	Method of [22]	Method of [23]	Method of [24]	Method of [25]	Proposed method
0.1	$1.52e-02$	$2.13e-03$	$1.52e-02$	$1.72e-05$	$4.04e-08$	$2.47e-10$
0.2	$1.47e-02$	$4.21e-03$	$1.47e-02$	$3.26e-05$	$2.76e-08$	$2.27e-10$
0.3	$5.89e-03$	$6.19e-03$	$5.89e-03$	$4.49e-05$	$4.59e-08$	$2.00e-10$
0.4	$3.25e-03$	$8.00e-03$	$3.25e-03$	$5.28e-05$	$3.03e-08$	$1.68e-10$
0.5	$6.98e-03$	$9.60e-03$	$6.98e-03$	$5.56e-05$	$2.83e-08$	$1.31e-10$
0.6	$3.25e-03$	$1.09e-03$	$3.25e-03$	$5.28e-05$	$4.11e-08$	$8.98e-11$
0.7	$5.89e-03$	$1.19e-02$	$5.89e-03$	$4.49e-05$	$3.37e-08$	$4.58e-11$
0.8	$1.47e-02$	$1.24e-02$	$1.47e-02$	$3.26e-05$	$1.79e-08$	$1.75e-14$
0.9	$1.52e-02$	$1.09e-02$	$1.52e-02$	$1.72e-05$	$2.95e-08$	$4.61e-11$

need to start with a good initial guess. We believe that the idea used in this work can also be utilized for robust solution of partial differential equations.

References

1. Keller H. *Numerical Solution of Two Point Boundary Value Problems*. CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM: Philadelphia, PA, 1976.
2. Atkinson K, Han W, Stewart D. *Numerical Solution of Ordinary Differential Equations*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts, Wiley, 2009.

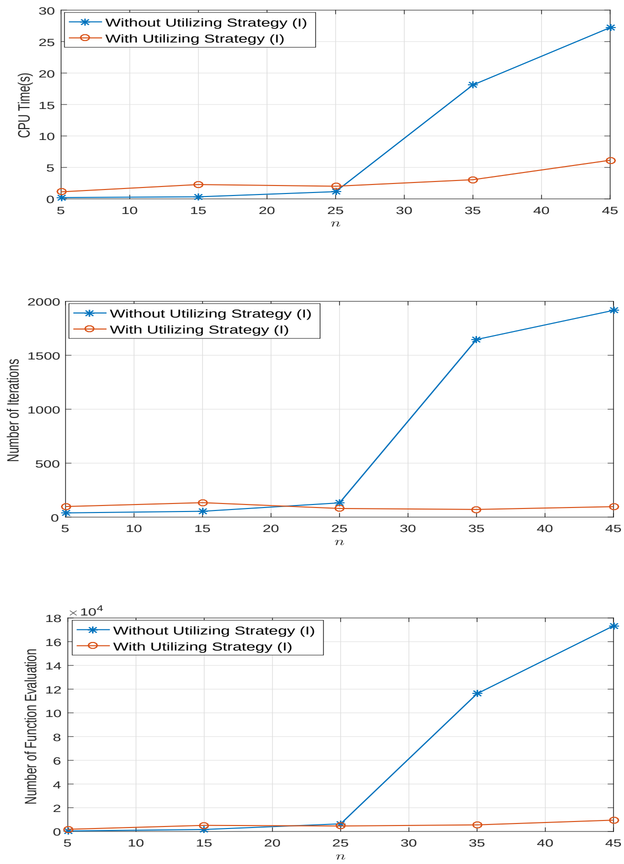

Figure 8. Comparing the ability of the proposed method with and without utilizing the optimization strategy for $\lambda = 2$ and $\theta = 8.50719957071$, in Example 2.

3. Boyd J. *Chebyshev and Fourier Spectral Methods: Second Revised Edition*. Dover Books on Mathematics, Dover Publications, 2001.
4. Kelley C. *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, 2003.
5. Shampine L, Gladwell I, Thompson S. *Solving ODEs with MATLAB*. Cambridge University Press, 2003.
6. Canuto C, Hussaini M, Quarteroni A, Zang T. *Spectral Methods in Fluid Dynamics*. Springer Series in Computational Physics, Springer: Berlin, 1991.
7. Fornberg B. *A Practical Guide to Pseudospectral Methods*. Cambridge University Press: Cambridge, 1998.
8. Trefethen LN. *Spectral methods in MATLAB*. SIAM: Philadelphia, 2000.
9. Gautschi W. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, Oxford, 2004.
10. Fatemi SZ, Shamsi M, Razmjooy N. Collocation method for differential variational inequality problems. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 2019; 32(1):e2466.
11. Oshagh MKe, Shamsi M. An adaptive wavelet collocation method for solving optimal control of elliptic variational inequalities of the obstacle type. *Computers & Mathematics with Applications* 2018; 75(2):470–485.
12. Fakhar-Izadi F, Dehghan M. Fully spectral collocation method for nonlinear parabolic partial integro-differential equations. *Applied Numerical Mathematics* 2018; 123:99–120.
13. Nabati M, Emamjome M, Jalalvand M. Reproducing kernel pseudospectral method for the numerical investigation of nonlinear multi-point boundary value problems. *Computational and Applied Mathematics* 2018; 37(5):6530–6543.
14. Patrício MS, Ramos H, Patrício M. Solving initial and boundary value problems of fractional ordinary differential equations by using collocation and fractional powers. *Journal of Computational and Applied Mathematics* 2019; 354:348–359.
15. Delkhosh M, Parand K. Generalized pseudospectral method: Theory and applications. *Journal of Computational Science* 2019; 34:11–32.
16. Dennis J, Schnabel R. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, 1996.
17. Nocedal J, Wright SJ. *Numerical Optimization*. 2nd edn., Springer: New York, 2006.
18. Osborne MR. On shooting methods for boundary value problems. *Journal of mathematical analysis and applications* 1969; 27(2):417–433.
19. Kubicek M, Hlavacek V. *Numerical solution of nonlinear boundary value problems with applications*. Courier Corporation, 2008.
20. Tomar S, Pandey R. An efficient iterative method for solving Bratu-type equations. *Journal of Computational and Applied Mathematics* 2019; 357:71–84.

21. Deeba E, Khuri S, Xie S. An algorithm for solving boundary value problems. *Journal of Computational Physics* 2000; 159(2):125–138.
22. Khuri SA. A new approach to Bratus problem. *Applied mathematics and computation* 2004; 147(1):131–136.
23. Wazwaz AM. Adomian decomposition method for a reliable treatment of the Bratu-type equations. *Applied Mathematics and computation* 2005; 166(3):652–663.
24. Caglar H, Caglar N, Özer M, Valarıstos A, Anagnostopoulos AN. B-spline method for solving Bratu's problem. *International Journal of Computer Mathematics* 2010; 87(8):1885–1891.
25. Raja MAZ. Solution of the one-dimensional Bratu equation arising in the fuel ignition model using ANN optimised with PSO and SQP. *Connection Science* 2014; 26(3):195–214.