Hindawi
Journal of Mathematics
Volume 2021, Article ID 3497017, 8 pages
https://doi.org/10.1155/2021/3497017

Research Article

Subdivision Collocation Method for One-Dimensional Bratu's Problem

Ghulam Mustafa [1], **Syeda Tehmina Ejaz** [2], **Sabila Kouser** [2], **Shafqat Ali** [1], and **Muhammad Aslam** [3]

[1] *Department of Mathematics, The Islamia University of Bahawalpur, Bahawalpur, Pakistan*
[2] *Department of Mathematics, The Government Sadiq College Women University Bahawalpur, Bahawalpur, Pakistan*
[3] *Department of Mathematics, Lock Haven University of Pennsylvania, Lock Haven, PA, USA*

Correspondence should be addressed to Ghulam Mustafa; ghulam.mustafa@iub.edu.pk

Received 17 April 2021; Revised 16 July 2021; Accepted 22 July 2021; Published 3 August 2021

Academic Editor: Sumit Chandok

The purpose of this article is to employ the subdivision collocation method to resolve Bratu's boundary value problem by using approximating subdivision scheme. The main purpose of this researcher is to explore the application of subdivision schemes in the field of physical sciences. Our approach converts the problem into a set of algebraic equations. Numerical approximations of the solution of the problem and absolute errors are compared with existing methods. The comparison shows that the proposed method gives a more accurate solution than the existing methods.

1. Introduction

The general expression of Liouville–Bratu–Gelfand equation [1, 2]:

$$\begin{cases} \Delta\chi(r) + \alpha \exp(\chi(r)) = 0, & r \in \Omega_1, \\ \chi(r) = 0, & r \in \partial\Omega_1, \end{cases} \tag{1}$$

where parameter $\alpha > 0$ and Ω_1 is a bounded domain. We consider the Bratu's boundary value problem in one-dimensional planar coordinates [2–4] of the form

$$\chi^{\cdot\cdot}(r) + \alpha \exp(\chi(r)) = 0, \quad 0 < r < 1, \alpha > 0, \tag{2}$$

with conditions at the ends of the domain

$$\begin{aligned} \chi(0) &= 0, \\ \chi(1) &= 0. \end{aligned} \tag{3}$$

The detailed information of problem (2) is given in [4, 5]. The exact solution of (2) is

$$\chi(r) = -2 \ln\left[\frac{\cosh(0.5\theta(r-0.5))}{\cosh(0.25\theta)}\right], \quad \text{where } \theta = \sqrt{2\alpha}\cosh(0.25\theta). \tag{4}$$

The exponential term guarantees nonlinearity and the bifurcation phenomenon that follows up. In particular, one can verify the following for different values of α, i.e., problem (2) has no solution for $\alpha > \alpha_c$, unique solution for $\alpha = \alpha_c$, and two bifurcated solutions have been obtained for $0 < \alpha < \alpha_c$, where α_c is the critical value given as $\alpha_c = 3.51380719$. It is the solution of $1 = 0.25\sqrt{2\alpha_c}\sinh(0.25\theta)$.

In science and engineering, Bratu's problem is often used to characterize complex physical and chemical models. For example, Bratu's problem is used in a wide range of applications, including the thermal combustion theory's fuel ignition model, the model of the thermal reaction mechanism, the Chandrasekhar model of the universe's expansion, chemical reaction theory, radiative heat transfer, and nanotechnology.

Many researchers have developed analytical and numerical methods to solve the Bratu's problem, including the B-spline method [6], Adomian decomposition method

[7, 8], Chebyshev polynomial approximation method [2], homotopy analysis method [9], homotopy perturbation method [10, 11], differential transform method [12], Laplace transformed decomposition method [13], method of weighted residuals [14], and variational iteration method [15, 16]. Moreover, the solutions of the problem have been reported by Jalilian [17] using the nonpolynomial spline method, by Boyd [18] with the one-point pseudospectral collocation method, and by Abbasbandy et al. [19] with the Lie-group shooting method.

Our goal is to make use of subdivision schemes for solving Bratu's problem. Subdivision schemes-based algorithms are not frequently used to find numerical solutions of boundary value problems. The approximate solutions of boundary value problems have been found by subdivision-based algorithms. Initially, these algorithms were constructed by Qu and Agarwal [20, 21]. Their constructed algorithms were based on an interpolatory subdivision algorithm and formulated only for the second-order two-point boundary value problems. After that, Ejaz et al. [22, 23] constructed subdivision schemes-based algorithm for solutions of boundary value problems of third and fourth order. We present a subdivision collocation algorithm for solving Bratu's problem in this paper.

We organize our paper in the following way. In Section 2, we present some important properties of 6-point binary approximating subdivision scheme. In Section 3, subdivision collocation algorithm is formulated for the solution of (2). The convergence and error estimation of the proposed algorithm are also discussed in this section. Numerical results based on the proposed algorithm, comparison with other existing methods, and conclusion based on the obtained results are given in Section 4.

2. Subdivision Scheme and Derivatives of Its Two-Scale Relation

In this section, we define 6-point binary approximating subdivision scheme (6PBASS) [24] as

$$\begin{cases} Z_{2i}^{k+1} = a_0 Z_{i-2}^k + a_1 Z_{i-1}^k + a_2 Z_i^k + a_3 Z_{i+1}^k + a_4 Z_{i+2}^k, \\ Z_{2i+1}^{k+1} = b_0 Z_{i-2}^k + b_1 Z_{i-1}^k + b_2 Z_i^k + b_3 Z_{i+1}^k + b_4 Z_{i+2}^k + b_5 Z_{i+3}^k, \end{cases} \tag{5}$$

with $a_0 = a_4 = -(1/32)\alpha$, $a_1 = a_3 = (1/8)\alpha$, $a_2 = (1 - (3/16)\alpha)$, $b_0 = b_5 = (1-\alpha)\beta$, $b_1 = b_4 = -(1/16) - 3\beta(1-\alpha)$, $b_2 = b_3 = (9/16) + 2\beta(1-\alpha)$, where α and β are tension parameters. The 6PBASS scheme possesses some of the following properties:

(i) The scheme (5) is C^2-continuous for $\alpha = (1/5)$, $\beta = (13/1000)$.

(ii) It has support width $(5, 5)$.

(iii) Its approximation order is fourth.

(iv) Its fundamental solution is

$$\nu(i) = \begin{cases} 1, & \text{for } i = 0, \\ 0, & \text{for } i \neq 0, \end{cases} \tag{6}$$

and it satisfies the two-scale relation

$$\nu(x) = \sum_k a_k \nu(2x - k), \tag{7}$$

where a_k is the mask of the scheme (5). Since 6PBASS is C^2-continuous by [24], so its 2-scale relations $\nu(x)$ are also C^2-continuous.

(v) For the computation of the first- and second-order derivatives of (7), we adopt similar approach of [22, 23].

The first two derivatives of (7) are given in the following equations:

$$\begin{cases} \nu^{\cdot\cdot}(0) = 0, \\ \nu^{\cdot\cdot}(\pm 1) = \pm \dfrac{308375375}{103575526}, \\ \nu^{\cdot}(\pm 2) = \mp \dfrac{89875573}{155363289}, \\ \nu^{\cdot}(\pm 3) = \pm \dfrac{6037875}{1035755526}, \\ \nu^{\cdot}(\pm 4) = \pm \dfrac{186056}{155363289}, \end{cases} \tag{8}$$

$$\begin{cases} \nu^{\cdot}(0) = -\dfrac{2030241008}{65127633}, \\ \nu^{\cdot\cdot}(\pm 1) = \dfrac{1259384000}{65127633}, \\ \nu^{\cdot\cdot}(\pm 2) = -\dfrac{96510000}{21709211}, \\ \nu^{\cdot\cdot}(\pm 3) = \dfrac{43501000}{65127633}, \\ \nu^{\cdot\cdot}(\pm 4) = \dfrac{1765504}{65127633}. \end{cases} \tag{9}$$

3. Subdivision Collection Algorithm for Bratu's Problem

In this section, we have constructed a subdivision collocation algorithm for the solution of (2), which is based upon the fundamental solution of the subdivision scheme and its derivatives. Convergence and error estimations results are also presented in this section.

3.1. Formulation of Subdivision Collection Algorithm for Bratu's Problem. The detail of the proposed algorithm is given as follows:

Let

$$\mathbb{W}(y) = \sum_{i=-4}^{N+4} w_i v\left(\frac{y - y_i}{h}\right), \quad 0 \le y \le 1, \tag{10}$$

be an approximate solution of (2) and N must be greater than or equal to four, h is the step size and is defined as $h = (1/N)$, $y_i = ih$, where $i = -4$ to $N + 4$, and w_i are the unknowns to be determined. From (10), we get

$$D^2\mathbb{W}(y_i) = \frac{1}{h^2} \sum_{i=-4}^{N+4} w_i v^{\cdot\cdot}\left(\frac{y - y_i}{h}\right). \tag{11}$$

By using (10) and (11) in (2), we get

$$\frac{1}{h^2}\left[\sum_{i=-4}^{N+4} w_i v^{\cdot\cdot}\left(\frac{y_j - y_i}{h}\right)\right] + \alpha \exp\left[\sum_{i=-4}^{N+4} w_i v\left(\frac{y_j - y_i}{h}\right)\right] = 0, \tag{12}$$

where $j = 0, 1, \ldots, N$, and the conditions given at the ends of the domain (3) become

$$\begin{aligned} \mathbb{W}(0) &= w_0 = 0, \\ \mathbb{W}(1) &= w_N = 0. \end{aligned} \tag{13}$$

The matrix representation of equation (12) is

$$\mathscr{A}\mathscr{W} + \alpha\mathbb{D}_1 = 0, \tag{14}$$

where

$$\mathbb{A} = \begin{pmatrix} v^{\cdot\cdot}_4 & v^{\cdot\cdot}_3 & v^{\cdot\cdot}_2 & v^{\cdot\cdot}_1 & v^{\cdot\cdot}_0 & v^{\cdot\cdot}_{-1} & v^{\cdot\cdot}_{-2} & \cdots & 0 & 0 & 0 \\ 0 & v^{\cdot\cdot}_4 & v^{\cdot\cdot}_3 & v^{\cdot\cdot}_2 & v^{\cdot\cdot}_1 & v^{\cdot\cdot}_0 & v^{\cdot\cdot}_{-1} & \cdots & 0 & 0 & 0 \\ 0 & 0 & v^{\cdot\cdot}_4 & v^{\cdot\cdot}_3 & v^{\cdot\cdot}_2 & v^{\cdot\cdot}_1 & v^{\cdot\cdot}_0 & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & v^{\cdot\cdot}_{N-3} & v^{\cdot\cdot}_{N-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & v^{\cdot\cdot}_{N-2} & v^{\cdot\cdot}_{N-3} & v^{\cdot\cdot}_{N-4} \end{pmatrix}_{N+1 \times N+9}, \tag{15}$$

$$\mathbb{W} = (w_{-4}, w_{-3}, \ldots, w_{N+3}, w_{N+4})^T, \tag{16}$$

$$\mathbb{D}_1 = h^2 \times (\exp(\mathbb{W}(y_0)),\ \exp(\mathbb{W}(y_1)), \ldots,\ \exp(\mathbb{W}(y_{N-1})),\ \exp(\mathbb{W}(y_N)))^T. \tag{17}$$

Since system (14) is underdetermined because it has fewer equations than unknowns, so it requires eight more equations to get a unique solution. Two conditions are given in (13) at the ends of the domain of (2) and the detail of the remaining six conditions is given in the next section.

3.2. Forced Conditions. As we require six more conditions to get a unique solution of (14), so we will construct three conditions at the left and three conditions at the right end of the domain. Since 6PBASS reproduces third degree polynomial with order of approximation four, so the order of new conditions is four and these conditions are known as forced conditions. Let w_{-3}, w_{-2}, w_{-1} and $w_{N+1}, w_{N+2}, w_{N+3}$ represent the left end points and right end points. These left and right end points can be computed by using polynomial of degree three which interpolates the data (y_i, w_i), for $0 \le i \le 3$, i.e., left end conditions are obtained from

$$w_{-i} = S(-y_i), \quad i = 1, 2, 3, \tag{18}$$

where

$$S(y_i) = \sum_{j=1}^{4} \binom{4}{j} (-1)^{j+1} \mathbb{W}(y_{i-j}). \tag{19}$$

Since by (10), $\mathbb{W}(y_i) = w_i$ for $i = 1, 2, 3$ and substituting y_i by $-y_i$ in (19), we have

$$S(-y_i) = \sum_{j=1}^{4} \binom{4}{j} (-1)^{j+1} w_{j-i}. \tag{20}$$

So, the following conditions can be used at the left end and

$$\sum_{j=0}^{4} \binom{4}{j} (-1)^j w_{j-i} = 0, \quad i = 3, 2, 1. \tag{21}$$

Similarly, at the right end, we have the following conditions:

$$\sum_{j=0}^{4} \binom{4}{j} (-1)^j w_{i-j} = 0, \quad i = N+3, N+2, N+1. \tag{22}$$

Finally, we get system of $(N + 9) \times (N + 9)$ nonlinear equations

$$\mathbb{J}_c\mathbb{W} + \alpha\mathbb{D} = 0, \tag{23}$$

or

$$\mathbb{J}_c\mathbb{W} = -\alpha\mathbb{D}(w), \tag{24}$$

where

$$\mathbb{J}_c = \left(\mathbb{J}_{c_0}^T, \mathbb{A}^T, \mathbb{J}_{c_N}^T\right), \tag{25}$$

where $\mathbb{A}$ is defined in (15) and $\mathbb{J}_{c_0}$ and $\mathbb{J}_{c_N}$ are constrained as follows: the matrix $\mathbb{J}_{c_0}$ is obtained from all the conditions defined at the left end of the domain, i.e., first three rows obtained from (21) and fourth row of $\mathbb{J}_{c_0}$ obtained from (13) at $\mathbb{W}(0) = w_0 = 0$. Similarly, the matrix $\mathbb{J}_{c_N}$ is obtained from all the conditions defined at the right end of the domain, i.e., first row comes from (13) at $\mathbb{W}(1) = w_N = 0$ and remaining rows come from (22). Hence,

$$\begin{aligned}
\mathbb{J}_{c_0} &= \begin{pmatrix} 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \\
\mathbb{J}_{c_N} &= \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 \end{pmatrix}.
\end{aligned} \tag{26}$$

The column vector $\mathbb{W}$ is defined in (16) and $\mathbb{D}$ is defined as

$$\mathbb{D} = \left(0, 0, 0, \mathbb{W}(0), \mathbb{D}_1^T, \mathbb{W}(N), 0, 0, 0\right)^T, \tag{27}$$

where $\mathbb{D}_1$ is given in (17).

3.3. Iterative Algorithm. To find the numerical solutions of nonlinear system of equation (24), we define an iterative algorithm. The iterative algorithm includes the following steps:

(i) Formulation of initial solution: the initial approximate solution $\mathbb{W}^0$ is selected to find the following system:

$$\mathbb{J}_c\mathbb{W}^0 = -\alpha\mathbb{R}^0, \tag{28}$$

where

$$\begin{cases} \mathbb{R}^0 = \left(0, 0, 0, \mathbb{W}(0), \delta_0, \delta_1, \ldots, \delta_N, \mathbb{W}(N), 0, 0, 0\right)^T, \\ \delta_i = h^3\mathbb{D}_i\left(y_i, \mathbb{L}_i, \mathbb{D}_0^*\right), \\ \mathbb{L}_i = \mathbb{W}(0) + 3ih\left(\mathbb{W}(N) - \mathbb{W}(0)\right), \\ \mathbb{D}_0^* = \mathbb{W}(N) - \mathbb{W}(0), \end{cases} \tag{29}$$

where $i \in \{0, 1, \ldots, N\}$. The column vector $\mathbb{R}^0$ is the linear approximation of the column vector (27).

(ii) Iterative scheme: the following iterative scheme is used to find the approximate solution $\mathbb{W}^*$,

$$\mathbb{J}_c\mathbb{W}^{k+1} = \mathbb{D}\left(w^k\right), \quad k = 0, 1, 2, \ldots. \tag{30}$$

(iii) Terminating criteria: the following condition is used to stop the iteration at $k-$th level, for any ε; let $\varepsilon = 10^{-4}$,

$$\left\|\mathbb{W}^{k+1} - \mathbb{W}^k\right\|_\infty \le \varepsilon. \tag{31}$$

3.4. Convergence and Error Estimation. In this section, we present results of convergence and error estimation of the proposed iterative algorithm. The convergence of the iterative algorithm is guaranteed by the following proposition.

Proposition 1. *The approximate solution $\left\{\mathbb{W}^k\right\}$ founded by (28) and (30) linearly converges to the approximate solution $\mathbb{W}^*$ of (24) with the supposition that step size and the Lipschitz constants r_0, r_1 are small, i.e.,*

$$\left\|\mathbb{J}_c^{-1}\right\|_\infty\left(r_0h^2 + r_1h\frac{29773}{10000}\right) \le 1. \tag{32}$$

The proof is similar to [23].

The main result of error estimation is given by the following proposition.

Theorem 1. *Let exact solution $g(y) \in C^4[0,1]$ and w_i be obtained by solving (24) with the fourth order boundary treatment at the end points. Then, we have*

$$\left\|w\left(y_j\right) - g\left(y_j\right)\right\|_\infty = O\left(h^{4-j}\right), \quad j = 0, 1, 2. \tag{33}$$

4. Numerical Examples and Comparison

The numerical technique discussed previously is illustrated in this section by applying subdivision collection algorithm to the planar one-dimensional Bratu's problem (2) for three distinct values of α, which guarantee the existence of two locally unique solutions. We have created comparison tables using $\alpha = 1, 2$ and 3.51 to show the consistency of our approach in comparison to the exact solution as well as the solutions of other methods. All calculations have been performed using MATLAB .

(i) The fact regarding the solution of Bratu's problem for $\alpha = 1$ is obtained after third iteration, as shown in Table 1. Comparison between the numerical results and absolute errors obtained by our subdivision collection algorithm and decomposition method [25] are presented in Tables 2 and 3, respectively. From the tabulated results, we observed that the numerical results obtained by our subdivision collection algorithm are better than the decomposition method [25].

(ii) The fact regarding the solution of Bratu's problem for $\alpha = 2$ is obtained after fifth iteration, as shown in

Table 1: Numerical results of (2) by subdivision collection algorithm for case $\alpha = 1$.

x	Exact solution	By subdivision collection algorithm	Absolute error
0.0	0.0000000000	0.000000000000000	0.00000000000000
0.1	0.0498467900	0.049835942666664	$1.08473333363529e-5$
0.2	0.0891899350	0.089157036980790	$3.28980192095263e-5$
0.3	0.1176090956	0.117560379925558	$4.87156744417711e-5$
0.4	0.1347902526	0.134731783666151	$5.84689338488098e-5$
0.5	0.1405392142	0.140477341442747	$6.18727572535005e-5$
0.6	0.1347902526	0.134731783666151	$5.84689338488931e-5$
0.7	0.1176090956	0.117560379925558	$4.87156744419098e-5$
0.8	0.0891899350	0.089157036980790	$3.28980192097483e-5$
0.9	0.0498467900	0.049835942666663	$1.08473333366513e-5$
1.0	0.0000000000	0.000000000000000	$0.00000000000085e-5$

Table 2: Comparison between the numerical results of (2) for $\alpha = 1$.

x	Exact solution	By [25]	By subdivision collection algorithm
0.0	0.0000000000	0.0000000000	0.000000000000000
0.1	0.0498467900	0.0471616875	0.049835942666664
0.2	0.0891899350	0.0871680000	0.089157036980790
0.3	0.1176090956	0.1177614375	0.117560379925558
0.4	0.1347902526	0.1369920000	0.134731783666151
0.5	0.1405392142	0.1435546875	0.140477341442747
0.6	0.1347902526	0.1369920000	0.134731783666151
0.7	0.1176090956	0.1177614375	0.117560379925558
0.8	0.0891899350	0.0871680000	0.089157036980790
0.9	0.0498467900	0.0471616875	0.049835942666663
1.0	0.0000000000	0.0000000000	0.000000000000000

Table 3: Comparison between the absolute errors of (2) for case $\alpha = 1$.

x	By [25]	By subdivision collection algorithm
0.0	0.000000000000000000	0.00000000000000
0.1	$2.685102500000001e-3$	$1.08473333363529e-5$
0.2	$2.021935000000003e-3$	$3.28980192095263e-5$
0.3	$1.523418999999915e-4$	$4.87156744417711e-5$
0.4	$2.201747400000009e-3$	$5.84689338488098e-5$
0.5	$3.015473299999988e-3$	$6.18727572535005e-5$
0.6	$2.201747400000009e-3$	$5.84689338488931e-5$
0.7	$1.523418999999915e-4$	$4.87156744419098e-5$
0.8	$2.021935000000003e-3$	$3.28980192097483e-5$
0.9	$2.685102500000001e-3$	$1.08473333366513e-5$
1.0	0.000000000000000000	$0.00000000000085e-5$

Table 4: Numerical results of (2) for case $\alpha = 2$.

x	Exact solution	By subdivision collection algorithm	Absolute error
0.0	0.0000000000	0.000000000000000	0.00000000000000
0.1	0.1144107440	0.114588772999854	$1.78028999853849e-4$
0.2	0.2064191156	0.206717298179522	$2.98182579522244e-4$
0.3	0.2738793116	0.274280205962876	$4.00894362875881e-4$
0.4	0.3150893646	0.315559243360757	$4.69878760757103e-4$
0.5	0.3289524214	0.329446300901418	$4.93879501417605e-4$
0.6	0.3150893646	0.315559243360757	$4.69878760757159e-4$
0.7	0.2738793116	0.274280205962876	$4.00894362875992e-4$
0.8	0.2064191156	0.206717298179522	$2.98182579522410e-4$
0.9	0.1144107440	0.114588772999854	$1.78028999854057e-4$
1.0	0.0000000000	0.000000000000000	$0.00000000000017e-4$

TABLE 5: Comparison between numerical results of (2) for case $\alpha = 2$.

x	Exact solution	By subdivision collection algorithm	By [25]	By [5]
0.0	0.0000000000	0.000000000000000	0.0000000000	0.0000000000
0.1	0.1144107440	0.114248449272834	0.0991935000	0.1122817141
0.2	0.2064191156	0.206065066411653	0.1917440000	0.2022094162
0.3	0.2738793116	0.273375407642615	0.2679915000	0.2676925058
0.4	0.3150893646	0.314489410524746	0.3183360000	0.3070874506
0.5	0.3289524214	0.328319028557975	0.3359375000	0.3193532294
0.6	0.3150893646	0.314489410524746	0.3183360000	0.3041598403
0.7	0.2738793116	0.273375407642615	0.2679915000	0.2619458909
0.8	0.2064191156	0.206065066411653	0.1917440000	0.1940413072
0.9	0.1144107440	0.114248449272835	0.0991935000	0.1035373785
1.0	0.0000000000	0.000000000000000	0.0000000000	0.0000000000

TABLE 6: Comparison between absolute errors of (2) for case $\alpha = 2$.

x	By subdivision collection algorithm	By [25]	By [5]
0.0	$0.00000000000200e-4$	0.000000000000000000	0.000000000000000000
0.1	$1.78028999853849e-4$	$1.521724399999999e-2$	$2.129029899999996e-3$
0.2	$2.98182579522244e-4$	$1.467511560000001e-2$	$4.209699400000017e-3$
0.3	$4.00894362875881e-4$	$5.887811600000015e-3$	$6.186805800000028e-3$
0.4	$4.69878760757103e-4$	$3.246635400000031e-3$	$8.001913999999999e-3$
0.5	$4.93879501417605e-4$	$6.985078600000028e-3$	$9.599191999999979e-3$
0.6	$4.69878760757159e-4$	$3.246635400000031e-3$	$1.092952429999999e-2$
0.7	$4.00894362875992e-4$	$5.887811600000015e-3$	$1.193342070000003e-2$
0.8	$2.98182579522410e-4$	$1.467511560000001e-2$	$1.237780840000000e-2$
0.9	$1.78028999854057e-4$	$1.521724399999999e-2$	$1.087336650000000e-2$
1.0	$0.00000000000017e-4$	0.000000000000000000	0.000000000000000000

TABLE 7: Numerical results of (2) for case $\alpha = 3.51$.

x	Exact solution	By subdivision collection algorithm	Absolute error
0.0	0.0000000000	0.000000000000002	0.000000000000002
0.1	0.3958056990	0.395838508425970	0.000032809425970
0.2	0.7390974100	0.739653867710330	0.000556457710330
0.3	1.0087582600	1.009642835147183	0.000884575147183
0.4	1.1825366600	1.183595191371533	0.001058531371533
0.5	1.2427426900	1.243855461630791	0.001112771630791
0.6	1.1825366600	1.183595191371532	0.001058531371532
0.7	1.0087582600	1.009642835147182	0.000884575147182
0.8	0.7390974100	0.739653867710329	0.000556457710329
0.9	0.3958056990	0.395838508425968	0.000032809425968
1.0	0.0000000000	0.000000000000000	0.000000000000000

TABLE 8: Comparison between absolute errors of (2) for case $\alpha = 3.51$.

x	By [6]	By subdivision collection algorithm
0.0	0.00000000000000	0.00000000000002
0.1	$3.84172369550e-2$	$3.280942597000e-5$
0.2	$7.48135367780e-2$	$5.564577103300e-4$
0.3	$1.05827422823e-1$	$8.845751471830e-4$
0.4	$1.27116880861e-1$	$1.058531371533e-3$
0.5	$1.34752877607e-1$	$1.112771630791e-3$
0.6	$1.27116880861e-1$	$1.058531371532e-3$
0.7	$1.05827422823e-1$	$8.845751471820e-4$
0.8	$7.48135367780e-2$	$5.564577103290e-4$
0.9	$3.84172369550e-2$	$3.280942596800e-5$
1.0	0.00000000000000	0.000000000000

Table 4. Comparison between the numerical results and absolute errors obtained by our subdivision collection algorithm, decomposition [25], and Laplace method [5] are presented in Tables 5 and 6, respectively. From the tabulated results, we observed that the numerical results obtained by our subdivision collection algorithm are better than [5, 25].

(iii) The fact regarding the solution of Bratu's problem for $\alpha = 3.15$ is obtained after forty-two iterations, as shown in Table 7. Comparisons between the absolute errors obtained by our subdivision collection algorithm and B-spline [6] method are presented in Table 8. From the tabulated results, we observed that the numerical results obtained by our subdivision collection algorithm give better approximation than [6].

5. Concluding Remarks

In this paper, we have established a subdivision collocation algorithm for the solution of one-dimensional nonlinear Bratu's problem. The numerical results obtained by subdivision collection algorithm showed that the algorithm is suitable for the approximate solution of (2). We have concluded that the numerical results converge to the exact solution for the small step size. We have also presented a comparison of absolute errors of the solution obtained from subdivision collection algorithm with decomposition method [25], Laplace method, [5] and B-spline method [6] for different values of α. We conclude that our algorithm gives smaller absolute errors as compared with the other existing methods [5, 6, 25].

Data Availability

The data used to support the findings of the study are available within this paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] U. M. Ascher, R. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, USA, Classics in Applied Mathematics, 1995.

[2] J. P. Boyd, "Chebyshev polynomial expansions for simultaneous approximation of two branches of a function with application to the one-dimensional bratu equation," *Applied Mathematics and Computation*, vol. 143, no. 2-3, pp. 189–200, 2003.

[3] R. Buckmire, "Investigations of nonstandard, mickens-type, finite-difference schemes for singular boundary value problems in cylindrical or spherical coordinates," *Numerical Methods for Partial Differential Equations*, vol. 19, no. 3, pp. 380–398, 2003.

[4] J. Jacobsen and K. Schmitt, "The liouville-bratu-gelfand problem for radial operators," *Journal of Differential Equations*, vol. 184, no. 1, pp. 283–298, 2002.

[5] J. S. McGough, "Numerical continuation and the gelfand problem," *Applied Mathematics and Computation*, vol. 89, no. 1-3, pp. 225–239, 1998.

[6] H. Caglar, N. Caglar, M. Özer, A. Valarıstos, and A. N. Anagnostopoulos, "*B*-spline method for solving bratu's problem," *International Journal of Computer Mathematics*, vol. 87, no. 8, pp. 1885–1891, 2010.

[7] E. Deeba, S. A. Khuri, and S. Xie, "An algorithm for solving boundary value problems," *Journal of Computational Physics*, vol. 159, no. 2, pp. 125–138, 2000.

[8] A.-M. Wazwaz, "Adomian decomposition method for a reliable treatment of the bratu-type equations," *Applied Mathematics and Computation*, vol. 166, no. 3, pp. 652–663, 2005.

[9] S. Abbasbandy and E. Shivanian, "Prediction of multiplicity of solutions of nonlinear boundary value problems: novel application of homotopy analysis method," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 3830–3846, 2010.

[10] X. Feng, Y. He, and J. Meng, "Application of homotopy perturbation method to the bratu-type equations," *Topological Methods in Nonlinear Analysis*, vol. 31, no. 2, pp. 243–252, 2008.

[11] M. C. Wang, X. S. Zhao, and X. Liu, "Analytic solutions of a class of nonlinearly dynamic systems," *Journal of Physics: Conference Series*, vol. 96, no. 1, Article ID 012174, 2008.

[12] I. H. A.-H. Hassan and V. S. Erturk, "Applying differential transformation method to the one-dimensional planar bratu problem," *International Journal of Contemporary Mathematical Sciences*, vol. 2, no. 30, pp. 1493–1504, 2007.

[13] S. A. Khuri, "A new approach to bratu's problem," *Applied Mathematics and Computation*, vol. 147, no. 1, pp. 131–136, 2004.

[14] Y. A. S. Aregbesola, "Numerical solution of bratu problem using the method of weighted residual," *Electronic Journal of Southern African Mathematical Sciences Association*, vol. 3, no. 1, pp. 1–7, 2003.

[15] J.-H. He, "Some asymptotic methods for strongly nonlinear equations," *International Journal of Modern Physics B*, vol. 20, no. 10, pp. 1141–1199, 2006.

[16] J. H. He, "Variational approach to the bratu's problem," *Journal of Physics: Conference Series*, vol. 96, no. 1, Article ID 012087, 2008.

[17] R. Jalilian, "Non-polynomial spline method for solving bratu's problem," *Computer Physics Communications*, vol. 181, no. 11, pp. 1868–1872, 2010.

[18] J. P. Boyd, "One-point pseudospectral collocation for the one-dimensional bratu equation," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5553–5565, 2011.

[19] S. Abbasbandy, M. S. Hashemi, and C.-S. Liu, "The lie-group shooting method for solving the bratu equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 11, pp. 4238–4249, 2011.

[20] R. Qu and R. P. Agarwal, "Solving two point boundary value problems by interpolatory subdivision algorithms," *International Journal of Computer Mathematics*, vol. 60, no. 3-4, pp. 279–294, 1996.

[21] R. Qu and R. P. Agarwal, "A subdivision approach to the construction of approximate solutions of boundary-value problems with deviating arguments," *Computers & Mathematics with Applications*, vol. 35, no. 11, pp. 121–135, 1998.

[22] S. T. Ejaz and G. Mustafa, "A subdivision based iterative collocation algorithm for nonlinear third order boundary value problems," *Advances in Mathematical Physics*, vol. 2016, p. 15, Article ID 5026504, 2016.

[23] G. Mustafa, M. Abbas, S. T. Ejaz, A. I. M. Ismail, and F. Khan, "A numerical approach based on subdivision schemes for solving non-linear fourth order boundary value problems," *Journal of Computational Analysis and Applications*, vol. 23, no. 4, pp. 607–623, 2017.

[24] K. Rehan and S. S. Siddiqi, "A combined binary 6-point subdivision scheme," *Applied Mathematics and Computation*, vol. 270, pp. 130–135, 2015.

[25] S. Liao and Y. Tan, "A general approach to obtain series solutions of nonlinear differential equations," *Studies in Applied Mathematics*, vol. 119, no. 4, pp. 297–354, 2007.