An Efficient Technique for Solving Bratu Type Equation via Wavelet Orthonormal Boubaker polynomials

1. Introduction

Bratu’s equation represents one of many important problems in science and engineering applications (e.g., heat transfer, chemical reactor theory, nanotechnology ...etc.)[1, 2].

The general form of Bratu-type Equation is

\[ \Delta u + \lambda e^u = 0 \]

\[ u(0) = 0 \text{ on } \delta \Omega \]  

where \( \delta \Omega \) is the boundary of a region \( \Omega \) and \( \lambda > 0 \).

This equation can be defined by a 1-dimensional planar Bratu boundary value problem as follows [3].

\[ u_{xx} + \lambda e^u = 0 \quad 0 < x < 1, \quad \lambda > 0 \]

\[ u(0) = u(1) = 0 \]
which has the following exact solution

\[ u(x) = -2\ln\left(\frac{\cosh\left(x-0.5\frac{\theta}{2}\right)}{\cosh\left(\frac{\theta}{4}\right)}\right), \text{ where } \cosh \theta = \sqrt{2\lambda} \cosh \left(\frac{\theta}{4}\right). \]

The value of \( \lambda \) determines the number of solutions for this kind of equations such that

If \( \lambda < \lambda_c \), there would be two solutions. Where \( \lambda_c \) represents the critical value.

If \( \lambda = \lambda_c \), one solution and no solution when \( \lambda > \lambda_c \).

For this type of Bratu equation \( \lambda_c \) has been found to be equal to 3.513830719.

For the 2-dimensional case no exact solution has been found to the Eq. 1 (see Mohsen[4]),

This type of equations is defined as follows

\[ u_{xx} + u_{yy} + ce^u = 0 \]

On the region \( \Omega \), where \( x \in [0,1], y \in [0,1] \), and the boundary condition on \( \partial \Omega \) is

\[ u = 0 \]

The same as in 1-dimensional planar Bratu boundary value case, the equation has one solution for \( \lambda = \lambda_c \), two solutions for \( \lambda < \lambda_c \) and no solution for \( \lambda > \lambda_c \). For small \( \lambda_c \) the equation represents a steady state heat transfer while for \( \lambda > \lambda_c \) this thermal reaction will end with explosion which represents no solution for Bratu’s equation [5].

\( \lambda_c \) was found to be approximately 6.808124423.

This kind of equations has been solved by many mathematicians using different proceedings. Buckmire R., studied nonstandard finite difference method which is known as Mickens finite difference method and comparing with other methods (standard finite difference Boyd collocation, Adomian polynomial decomposition and shooting) for solving 1D planar Bratu boundary value problem[6]. Syam M. I. and Hamdan A. introduced a method based on the Laplace Adomian Decomposition method with a predictor corrector technique to demonstrate the founded solution curve[7]. Batiha B., applied the variational iteration method without discretization using a correction functional with Lagrange multiplier, finding its optimality by the aid of variational theory, this method can be used in nonlinear problems without small perturbation or linearization, it is proved to have faster convergence than Adomian’s method [8, 9]. Venkatesh S. G., Ayyaswamy S. K. and Hariharan G., used Haar wavelet method for solving 1D Bratu-type equation [10]. Rashidinia J. and Taher N., utilized sinc-collocation method for solving Bratu’s problem which was already presented by professor Stenger in 1993[5]. Changqang Y. and Jiahhua H., used Chebyshev wavelets with collocation method for solving 1D Bratu problem [11]. Hassan H. N. And Semary M., presented analytic approximate solutions using homotopy analysis method [12]. Saravi M., Hermann M. and Kaiser D., used He’s variational iteration method with three terms in expansion of nonlinear part for solving Bratu’s boundary value problem [13]. Zareinia and Hoshyar used a non-polynomial cubic spline method for solving the Bratu equation [14]. Mohsen A. presented a good survey of the properties
and different treatments of 1D and 2D Bratu problems, using finite – difference and nonstandard finite difference for solving this equation with a simple starting function for this object [4]. Inc M. et al, used the reproducing kernel Hilbert space method RKHSM (a functional analysis proceeding), which has given accurate results with respect to other known methods (Adomian decomposition, Laplace decomposition, B-spline, Non-polynomial spline and Lie-group shooting method), this method was already used by researchers for finding a solution to boundary value problems[15]. Bougoffa L. extracted a new exact solution for the generalized Bratu equation under suitable conditions of \( \lambda(x) \) and \( f(x) \) [16]. Ghomanjani F. and Shateyi S. used Bernstein polynomial approximation for solving 1D Bratu-type equation [17].

Many researchers have widely utilized Boubaker’s polynomials which have first appeared for solving heat equation inside physical model [18], then many attempts have been done to construct its wavelets by the following researchers. Ouda E. H., founded the orthonormal family for Boubaker polynomials using Gram-Schmidt method then the deduction of the operational matrices of derivative and integration and using them for solving optimal control problem by indirect method [19]. Ouda E. H., Shihab S. and Rasheed M., used the properties of Boubaker orthonormal polynomials for constructing the new Boubaker wavelet orthonormal functions, founding its operational matrix of derivative then utilizing it with collocation method for transforming a higher order integro-differential equation into linear algebraic equations which can easily be solved [20].

Bratu-type equation was already studied by many researchers as we have seen from the brief summary above. This kind of numerical method proceeding gives only the lower branch solution of Bratu-type equation for \( \lambda < \lambda_c \). In this paper, we introduced a method for solving 1D initial and boundary value problem of Bratu equations to show the capability of OBWP’s for this aim.

In this paper, we have been introduced a method for solving 1D initial and boundary value problem of Bratu equations. The paper is arranged as follows, in section 2 we present the wavelet orthonormal Boubaker polynomials. In section 3, the standard newton iterative method for nonlinear system is introduced. Numerical examples with comparison with the exact results, and illustrative graphs have been added in section 4. Finally, the conclusions, in section 5.

2. Orthonormal Boubaker Polynomials:

The Boubaker orthonormal polynomials were deduced using Gram-shmidt method, it is denoted by \( \beta_m(t) \) of order \( m \) and defined as follows [19]

\[
\begin{align*}
B_0(t) & = 1 \\
B_1(t) & = 2\sqrt{3}(t - \frac{1}{2}) \\
B_2(t) & = 6\sqrt{5}(t^2 - t + \frac{1}{6}) \\
B_3(t) & = 20\sqrt{7}(t^3 - \frac{3}{2}t^2 + \frac{3}{5}t - \frac{1}{20}) \\
B_4(t) & = 70\sqrt{9}(t^4 - 2t^3 + \frac{9}{7}t^2 - \frac{2}{7}t + \frac{1}{70})
\end{align*}
\]
In general, the \( m \)-th order orthonormal Boubaker polynomial can be obtained using the following binomial expansion.

\[
Bo_m(t) = (2m + 1)\frac{1}{2} \sum_{k=0}^{m} \frac{(-1)^{m+k}(m+k)!}{k! [(m-k)!]} t^k
\]

3. Orthonormal Boubaker Wavelet [20]

The orthonormal Boubaker wavelet polynomials have been found by the aid of orthonormal Boubaker polynomials, it can be defined as

\[
Bw_{n,m}(t) = Bw(k, n^*, m, t)
\]

have four arguments \( k = 1, 2, ..., n^* = 2n - 1 \), \( n = 1, 2, ..., 2^{k-1} \)

\( m \) is the order of Boubaker orthonormal polynomials.

\[
Bw_{n,m}(t) = \begin{cases}
2^{k-1} & \text{if } t \in \left[ \frac{n^*-1}{2^k}, \frac{n^*+1}{2^k} \right] \\
0 & \text{otherwise}
\end{cases}
\]

Where \( Bo_m \) are Boubaker orthonormal polynomials.

4. Orthonormal Boubaker wavelet operation matrix of derivatives

Let \( Bw(t) \) be the Boubaker orthonormal wavelet vector presented in Eq. (2), and then \( Bw(t) \) can be written as

\[
\frac{dBw(t)}{dt} = DBw(t)
\]

where \( D \) is the \( 2^k (M+1) \) operational Matrix of the derivatives expressed as

\[
D = \begin{bmatrix}
D_1 & 0 & \ldots & 0 \\
0 & D_1 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & D_1
\end{bmatrix}
\]

where \( D_1 \) is an \((M+1)\times(M+1)\) matrix and expressed as follows:

\[
D_1 = \sqrt[2K+1]{\frac{1}{2(2m+1)}} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & \sqrt{3} & 0 & 0 & 0 \\
1 & 0 & \sqrt{5} & 0 & 0 \\
0 & \sqrt{3} & \sqrt{7} & 0 & 0 \\
1 & 0 & \sqrt{5} & \sqrt{7} & 0 \\
\end{bmatrix}
\]

\( m \) represents Boubaker orthonormal order.
then
\[ \frac{d^n Bw(t)}{dt^n} = D^n Bw(t), \ n = 1,2,3, \ldots \] \hspace{1cm} \ldots (3)

5. The proposed method:

In this method, the modification is using Newton’s iteration with different new terms of orthonormal wavelet Boubaker polynomial for each step.

The method in steps would be as follows

1- Assuming \( u(t) = \sum_{m=0}^{M-1} c_{n,m} Bw_{n,m} \), with \( M = 4, k = 1 \).

2- For the 1st iteration, using Eq.3 with 2nd order differentiation of orthonormal Boubaker wavelet polynomials, the Eq.1 becomes

3- The suggested collocation points can be found according to

\[ t_i = \frac{1}{2} \left[ 1 + \cos \left( \frac{(i-1)\pi}{2^{k-1}M-1} \right) \right], \]

\[ i = 2,3, \ldots, 2^{k-1}M - 1 \] \hspace{1cm} \ldots (5)

These points are substituted in Eq.4 to find the collocated equations.

4- Using the boundary conditions equations with the collocated equations gives us a system of \( M \)-nonlinear equations.

5- Assuming initial values for the \( C \)'s then using Newton’s method the values of \( C \)'s for \( u \) can be found for the 1st iteration.

6- Repeating the same procedure for the following iterations with \( M = 5, 6 \) and \( k = 1 \) to find the new approximate \( u \).

In non-homogeneous case the same method can be used.

6. Numerical Examples

Example1

Consider the initial value Bratu’s problem

\[ u'' - 2e^u = 0 \]

\[ u(0) = u'(0) = 0. \]

with \( u_{exact} = -2 \ln(\cos t) \)

Assuming \( u(t) = \sum_{m=0}^{3} c_{n,m} Bw_{n,m} \), with \( M = 4, k = 1 \) , then
According to Eq. (6) the initial condition equations would be

\[ u(0) = 0 = c_{10} - 3\sqrt{3} c_{11} + 13\sqrt{5} c_{12} - 63\sqrt{7} c_{13} \quad \text{... (7)} \]

\[ \dot{u}(0) = 0 = 4\sqrt{3} c_{11} - 36\sqrt{5} c_{12} + 264\sqrt{7} c_{13}. \quad \text{... (8)} \]

and

\[ u''(t) = 48\sqrt{5} c_{12} + \sqrt{7} c_{13}(960t - 720). \]

Using Eq. (5), the collocation points would be 0.25, 0.75 for \( i = 2, 3 \).

Let \( F = u'' - 2e^u \)

We have now the following system of four equations with Eq. (7) and (8) with \( C \)'s as unknowns

\[ F(0.25) = 48\sqrt{5} c_{12} - 2e^{\left( c_{10} - 2\sqrt{3} c_{11} + \frac{(11\sqrt{5} c_{12})}{2} - 17\sqrt{7} c_{13} \right)} - 480\sqrt{7} c_{13} = 0.06316210 \]

\[ F(0.75) = 48\sqrt{5} c_{12} - 2e^{\left( c_{10} - \frac{(5\sqrt{12})}{2} \right)} = 0.62479979 \]

To find \( C \)'s, using Newton's method from assumed initial values of \( C \)'s, we obtain: for the 1\textsuperscript{st} iteration \((M = 4, k = 1)\) the following results

\[ c_{10} = 0.651173917794325, c_{11} = 0.277886089393695, c_{12} = 0.034390517276518, c_{13} = 0.001241459639318. \]

and

\[ u_1(t) = \frac{319}{607} t^3 + \frac{313}{472} t^2 + \frac{260}{10833} t - 5.807377148614457e - 14 \]

In the same way for the 2\textsuperscript{nd} iteration \((M = 5, k = 1)\), we get the following results

\[ c_{10} = 0.669917696510561, c_{11} = 0.276857334675699, c_{12} = 0.035454448724780, c_{13} = 0.002351369914077, c_{14} = 0.000134980050874. \]

\[ u_2(t) = \frac{1181}{2604} t^4 - \frac{3593}{9838} t^3 + \frac{2029}{1735} t^2 - \frac{250}{11153} t + 1.188229473581687e - 13 \]

Also in the same way for the 3\textsuperscript{rd} iteration \((M = 6, k = 1)\), we get the following results

\[ c_{10} = 0.663632989760171, c_{11} = 0.273895474804654, c_{12} = 0.034694806839367, c_{13} = 0.002228815444385, c_{14} = 0.000133676822432, c_{15} = 0.000007363658178. \]

\[ u_3(t) = \frac{335}{1701} t^5 - \frac{193}{4437} t^4 + \frac{303}{3400} t^3 + \frac{801}{814} t^2 + \frac{28}{19095} t + 1.004644626954481e - 13 \]
The approximate results for \( u(t) \) are shown in table (1) with comparison to the exact results.

<table>
<thead>
<tr>
<th>( t )</th>
<th>( u(t)(M=4) )</th>
<th>( u(t)(M=5) )</th>
<th>( u(t)(M=6) )</th>
<th>( u(t)_{\text{exact}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-5.8064664e-14</td>
<td>0.00000000000</td>
<td>0.00000000000</td>
<td>0.00000000000</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0095569606</td>
<td>0.0091225748</td>
<td>0.0100736698</td>
<td>0.0100167112</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0355298345</td>
<td>0.0399101953</td>
<td>0.0403608259</td>
<td>0.0402695461</td>
</tr>
<tr>
<td>0.3</td>
<td>0.0810718314</td>
<td>0.0916592487</td>
<td>0.0915349966</td>
<td>0.0913833118</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1493361610</td>
<td>0.1648543777</td>
<td>0.1646379784</td>
<td>0.1644580381</td>
</tr>
<tr>
<td>0.5</td>
<td>0.2434760331</td>
<td>0.2611684808</td>
<td>0.2613161678</td>
<td>0.2611684808</td>
</tr>
<tr>
<td>0.6</td>
<td>0.3666446574</td>
<td>0.3834627120</td>
<td>0.3840568926</td>
<td>0.3839303388</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5219952437</td>
<td>0.5357864807</td>
<td>0.5364247436</td>
<td>0.5361715151</td>
</tr>
<tr>
<td>0.8</td>
<td>0.7126810015</td>
<td>0.7233774520</td>
<td>0.7232979062</td>
<td>0.7227814936</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9418551408</td>
<td>0.9526615466</td>
<td>0.9511044919</td>
<td>0.9508848871</td>
</tr>
<tr>
<td>1</td>
<td>1.2126708712</td>
<td>1.2312529407</td>
<td>1.2280588698</td>
<td>1.2312529407</td>
</tr>
</tbody>
</table>

Table (1) Numerical solution of Example 1

Fig.1 illustrates the approximate results with the exact solution, which shows a good accuracy results.

![Figure1 Graphical illustration of Example1](image)

**Example 2**

Consider the nonlinear Bratu's problem

\[ u'' + 2e^u = 0 , \]

with \( u(0) = u(1) = 0 \).

The exact is \( u_{e}(t) = 2\ln\left(\frac{1.17877552}{\cosh(0.58938776+(1-2t))}\right) \).
Assuming \( u(t) = \sum_{m=0}^{3} c_{n,m} B_{n,m} \), with \( M = 4, k = 1 \), then
\[
u(t) = c_{10} + \sqrt{3} c_{11} (4t - 3) + \sqrt{5} c_{12} (24t^2 - 36t + 13) + \sqrt{7} c_{13} (160t^3 - 360t^2 + 264t - 63) \] ... (9)

According to Eq. (9) the initial condition equations would be
\[
u(0) = 0 = c_{10} - 3\sqrt{3} c_{11} + 13\sqrt{5} c_{12} - 63\sqrt{7} c_{13} \ldots (10)
\]
\[
u(1) = 0 = c_{10} + \sqrt{3} c_{11} + \sqrt{5} c_{12} + \sqrt{7} c_{13} \ldots (11)
\]
and
\[
u''(t) = 48\sqrt{5} c_{12} + \sqrt{7} c_{13} (960t - 720).
\]

Let \( F = 2e^u \).

We have now the following system of four equations with Eq.(10) and (11) with C's as unknowns
\[
F(0.125) = 2e^{\left(c_{10} - \frac{5\sqrt{3} c_{11}}{2} + \frac{71\sqrt{5} c_{12}}{8} - \frac{565\sqrt{7} c_{13}}{16}\right)} + 48\sqrt{5} c_{12} - 600\sqrt{7} c_{13} = 0.13960278
\]
\[
F(0.5) = 2e^{\left(c_{10} - \sqrt{3} c_{11} + \sqrt{5} c_{12} - \sqrt{7} c_{13}\right)} + 48\sqrt{5} c_{12} - 240\sqrt{7} c_{13} = 0.32895242
\]

To find C's, using Newton’s method from assumed initial values of C's, we obtain the following results for the 1st iteration \((M=4, k=1)\)
\[
c_{10} = 0.246962736146410, c_{11} = -0.102089915668847, c_{12} = -0.030575992704274, c_{13} = -0.000668172059051.
\]

In the same way for the 2nd iteration \((M=5, k=1)\), we get the following results,
\[
c_{10} = 0.216749920799177, c_{11} = -0.096081625621780, c_{12} = -0.023458207575246, c_{13} = 0.000708938658435, c_{14} = 0.000087281006539.
\]

In the same way for the 3rd iteration \((M=6, k=1)\), we get the following results
\[
c_{10} = 0.216843539460207, c_{11} = -0.096090113273145, c_{12} = -0.023465093655457, c_{13} = 0.000708938658435, c_{14} = 0.000087281006539.
\]

Also in the 3rd iteration \((M=6, k=1)\), we get the following results
\[
c_{10} = 0.216843539460207, c_{11} = -0.096090113273145, c_{12} = -0.023465093655457, c_{13} = 0.000707174078480, c_{14} = 0.0000869797774672, c_{15} = -0.00000087426089.
\]

The results are shown in table (2) with the comparison to the exact results.
Table (2) Numerical solution of Example 2

<table>
<thead>
<tr>
<th>t</th>
<th>u(t) (M=4)</th>
<th>u(t) (M=5)</th>
<th>u(t) (M=6)</th>
<th>u(t)_{exact}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.000000000000</td>
<td>-0.0000151553</td>
<td>0.000000000000</td>
<td>0.000000000000</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1146364981</td>
<td>0.1144873323</td>
<td>0.1145237562</td>
<td>0.1144107432</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2054810516</td>
<td>0.2067205504</td>
<td>0.2067805265</td>
<td>0.2064191164</td>
</tr>
<tr>
<td>0.3</td>
<td>0.2719025981</td>
<td>0.2742621431</td>
<td>0.2743404724</td>
<td>0.2738793118</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3132700754</td>
<td>0.3153935885</td>
<td>0.3154835864</td>
<td>0.3150893642</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3289524213</td>
<td>0.3291001989</td>
<td>0.3291968861</td>
<td>0.3289524213</td>
</tr>
<tr>
<td>0.6</td>
<td>0.3183185735</td>
<td>0.3150711204</td>
<td>0.3151716084</td>
<td>0.3150893642</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2807374697</td>
<td>0.2736993335</td>
<td>0.2738004032</td>
<td>0.2738793118</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2155780477</td>
<td>0.2060816523</td>
<td>0.2061745282</td>
<td>0.2064191164</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1222092452</td>
<td>0.1140187254</td>
<td>0.1140810422</td>
<td>0.1144107432</td>
</tr>
<tr>
<td>1</td>
<td>0.000000000000</td>
<td>0.0000150351</td>
<td>0.000000000000</td>
<td>0.000000000000</td>
</tr>
</tbody>
</table>

Fig.2 illustrates the approximate results with the exact solution, which shows a good accuracy results.

7. Conclusion:

The obtained numerical results have shown the good accuracy of this method using orthonormal Boubaker wavelet polynomials (OBWP’s). These results represent usually the lower branch of the solution as expected from this kind of methods, also it can be noticed that the odd iterations have more accurate results than others. From the illustrating graphs it can be seen that the numerical solutions have a good approximation to the exact with few iterations and fast convergence. This
method shows the ability of OBWP's for solving nonlinear Bratu-type equation, which can be verified in the future for finding solutions of other physical nonlinear problems.

References


