ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/273960318

An Efficient Algorithm based on the Cubic Spline for the Solution of Bratu-Type Equation

Article *in* Journal of Interdisciplinary Mathematics · November 2014

DOI: 10.1080/09720502.2013.842050

CITATIONS
4

READS
109

2 authors:

Mohammad Al-Towaiq
Jordan University of Science and Technology
42 PUBLICATIONS 64 CITATIONS
SEE PROFILE

Osama Ala'yed
Jadara University
8 PUBLICATIONS 21 CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project An efficient algorithm for the solution of second order PDE's using Taylor expansion polynomials View project

Project Renewable Energy Optimization View project

All content following this page was uploaded by Mohammad Al-Towaiq on 16 August 2015.

The user has requested enhancement of the downloaded file.

This article was downloaded by: [Jordan Univ. of Science & Tech]
On: 05 April 2015, At: 02:46
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

CrossMark
Click for updates

Journal of Interdisciplinary Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tjim20

An Efficient Algorithm based on the Cubic Spline for the Solution of Bratu-Type Equation

M. Al-Towaiq[a] & Osama Ala'yed[a]

[a] Jordan University of Science and Technology Department of Mathematics and Statistics P.O. Box 3030, Irbid 22110 Jordan. E-mails: ;

Published online: 30 Jan 2015.

To cite this article: M. Al-Towaiq & Osama Ala'yed (2014) An Efficient Algorithm based on the Cubic Spline for the Solution of Bratu-Type Equation, Journal of Interdisciplinary Mathematics, 17:5-6, 471-484, DOI: 10.1080/09720502.2013.842050

To link to this article: http://dx.doi.org/10.1080/09720502.2013.842050

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the “Content”) contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms &

Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

An Efficient Algorithm based on the Cubic Spline for the Solution of Bratu-Type Equation

M. Al-Towaiq*
Osama Ala'yed†
Jordan University of Science and Technology
Department of Mathematics and Statistics
P.O. Box 3030, Irbid 22110
Jordan

Abstract

In this paper, we propose an algorithm using the cubic spline interpolation on the finite difference method to solve the Bratu-type equation. The algorithm has been successfully implemented. Numerical results are also given to demonstrate the validity and the applicability of the proposed algorithm. The results we obtained show that the proposed algorithm perform better than some existing methods in the literature.

Keywords: *Bratu-Type Equation, Cubic Spline, Finite Difference, Differential Equations.*

1. Introduction

The Bratu's equation occurs in a large variety of applications such as the solid fuel ignition model in the thermal combustion theory, the model of thermal reaction process, the Chandrasekhar model of the expansion of the Universe, questions in geometry and relativity about the Chandrasekhar model, chemical reaction theory, and the radiative heat transfer and nanotechnology [1, 2, 3, 4].

Without loss of generality, we consider the following Bratu's problem in one-dimensional planar coordinates of the form:

$$\begin{gathered} u'' + \lambda e^{u} = 0, 0 < x < 1, \lambda > 0 \\ u(0) = 0, u(1) = 0. \end{gathered} \tag{1}$$

*E-mail: towaiq@just.edu.jo
†E-mail: os_al2009@yahoo.com

Journal of Interdisciplinary Mathematics
Vol. 17 (2014), No. 5 & 6, pp. 471–484
© Taru Publications

The Bratu's problem has zero, one or two solutions when $\lambda > \lambda_c$, $\lambda = \lambda_c$ and $\lambda < \lambda_c$ respectively, where the critical value λ_c satisfies the equation

$$1 = \frac{1}{4}\sqrt{2\lambda_c}\,\sinh(\frac{\theta}{4}).$$

It has been evaluated in [1,5,6], $\lambda_c = 3.513830719$.

Many researchers studied a numerical and analytical solutions of the Bratu's type equation. Syam and Hamdan presented a numerical technique for solving Bratu's equation, is based on the Laplace Adomain decomposition method which produces an implicit equation in two variables. Which the authors used the predictor corrector technique to trace the solution curve generated from this equation, and concluded that the technique works efficiently and accurately with Bratu's equation.

Buckmire presented a numerical approximations to the exact solution of the one-dimensional planar Bratu problem using various numerical methods. Of particular interest is the application of nonstandard finite-difference schemes known as Mickens finite-difference to solve the problem and compared the results.

Deeba *et al*. presented a numerical algorithm, based on the decomposition technique, for solving a class of nonlinear boundary value problems. The method was implemented on Bratu's problem. The scheme is shown to be highly accurate, and only a few terms are required to obtain accurate computable solutions.

Khuri presented a Laplace transform numerical technique for solving the nonlinear Bratu's problem. The numerical algorithm illustrates how the Laplace transform integral operator and the decomposition technique can be both efficiently manipulated to approximate the solution of this nonlinear boundary value problem. He concluded that the method converges rapidly and approximates the exact solution very accurately using only few iterates of the recursive scheme.

Li and Liao based on a new kind of analytic method, namely the homotopy analysis method , an analytic approach of searching for multiple solutions of strongly nonlinear problems was described by using Gelfand equation as an example. They concluded that the approach was convenient and efficient and the validity was verified by comparing the approximation series with the known exact solution.

I. H. Abdel-Halim Hassan and V. Ertürk applied differential transformation method to the one-dimensional planar Bratu problem and compare the known exact solutions to numerical solutions produced by differential transformation method.

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Lin Jin solved the Bratu-type problem by modified variational iteration method together with the Taylor series. The efficiency of this modified approach was verified by three examples.

This paper is organized as follows: In section 2 we present the proposed algorithm, we discuss the performance evaluation of the proposed algorithm in section 3, and we conclude the paper in section 4.

2. The Proposed Algorithm

In this section, we introduce the proposed algorithm using cubic spline on the finite difference method to solve the following Bratu-type equation.

$$\begin{aligned} u''(x) + \lambda e^{u(x)} = 0, 0 < x < 1, \lambda > 0 \\ u(0) = 0, u(1) = 0. \end{aligned} \tag{2.1}$$

***Step* 1.** For eqn. (2.1) select a set of equally spaced points $x_0, x_1, \ldots, x_N$ on the interval [0,1] by letting $x_i = ih$ with $h = \frac{1}{N}$, $i = 0, 1, \ldots, N$.

***Step* 2.** Apply the finite difference method on eqn. (2.1), we obtain the following non-linear system of equations:

$$\begin{aligned} -\frac{2}{h^2} w_1 + \frac{1}{h^2} w_2 + \lambda e^{w_1} &= 0 \\ \frac{1}{h^2} w_1 - \frac{2}{h^2} w_2 + \frac{1}{h^2} w_3 + \lambda e^{w_2} &= 0 \\ \vdots \qquad &\vdots\vdots \\ \frac{1}{h^2} w_{N-2} - \frac{2}{h^2} w_{N-1} + \lambda e^{w_{N-1}} &= 0 \end{aligned} \tag{2.2}$$

Solve this system to compute the approximate solution at the nodal points $x_0, x_1, \ldots, x_N$.

***Step* 3.** It is well known that the natural cubic spline function is given by:

$$\begin{aligned} s_i(x) = M_i \frac{(x_{i+1} - x)^3}{6h} + M_{i+1} \frac{(x - x_i)^3}{6h} + (w_i - \frac{M_i h^2}{6}) \frac{x_{i+1} - x}{h} 2.3 \\ + \left(w_{i+1} - \frac{M_{i+1} h^2}{6} \right) \frac{x - x_i}{h}, \end{aligned} \tag{2.3}$$

where $i = 0,1,\ldots,N-1$ and $M_i = s_i''(x_i)$. Using the continuity condition of $s''(x)$, we obtain the following linear system of equations:

$$M_{i-1}^* + 4M_i^* + M_{i+1}^* = \frac{6}{h^2}[w_{i-1} - 2w_i + w_{i+1}], \text{ for } i = 1,2,\ldots,N-2$$

In matrix notation:

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} M_1^* \\ M_2^* \\ \vdots \\ M_{N-3}^* \\ M_{N-2}^* \\ M_{N-1}^* \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} w_0 - 2w_1 + w_2 \\ w_1 - 2w_2 + w_3 \\ \vdots \\ w_{N-4} - 2w_{N-3} + w_{N-2} \\ w_{N-3} - 2w_{N-2} + w_{N-1} \\ w_{N-2} - 2w_{N-1} + w_N \end{bmatrix} \tag{2.4}$$

Solve this system for $M_1^*, M_2^*, \cdots, M_{N-1}^*$ to obtain an approximate solution to the coefficients of the cubic spline functions.

***Step* 4.** Using $w_0, w_1, \cdots, w_N$ and $M_1^*, M_2^*, \cdots, M_{N-1}^*$, obtained on steps 2 and 3 respectively to construct the cubic spline $s_i^*(x)$ solution for eqn. (2.1)

$$s_i^*(x) = M_i^* \frac{(x_{i+1}-x)^3}{6h} + M_{i+1}^* \frac{(x-x_i)^3}{6h} + \left(w_i - \frac{M_i^* h^2}{6}\right)\frac{(x_{i+1}-x)}{h}$$
$$+ \left(w_{i+1} - \frac{M_{i+1}^* h^2}{6}\right)\frac{(x-x_i)}{h} \tag{2.5}$$

for $i = 0,1,2,\ldots,N-1$, on the interval $[x_i, x_{i+1}]$. So the cubic spline solution $s_i^*(x)$ which interpolates $u(x)$ on the interval [0,1] is given by

$$S^*(x) = \begin{cases} s_0^*(x) & 0 \le x \le x_1 \\ s_1^*(x) & x_1 \le x \le x_2 \\ \vdots & \vdots \\ s_{N-1}^*(x) & x_{N-1} \le x \le 1 \end{cases}.$$

3. Performance Evaluation

In this section, we analyze the complixity of the proposed algorithm, then we give some numerical exampls to test the validity of the algorithm.

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

3.1. *Analysis*

The execution time of the proposed algorithm consists of two main tasks; applying the finite difference method and computing the coefficients of the cubic spline functions, steps 2 and 3, respectively in the algorithm. For step 2, the computational time for solving the non-linear system of equations requires approximately N^2 partial derivatives and solving $N \times N$ linear system at each step; which cost $\frac{N^3}{3} + N^2 - \frac{N}{3}$ for multiplications and divisions, and $\frac{N^3}{3} + \frac{N^2}{2} - \frac{5N}{6}$ for additions and subtractions, which is of order (N^3) for large N, [12]. For step 3, the computational time is equal the amount of time to solve the linear system of equations which is of order (N^3). It is clear that the total cost is of $O(N^3)$ for large N.

Moreover, the jacobians in the systems (2.2) and (2.4) are diagonally dominents matrices and that guaranteed the convergence of the systems to the exact solutions.

3.2. *Numerical Experiments*

In this section, we give some numerical examples appeared in the litrature to test the validity and the applicability of the proposed algorithm and for comparisons purposes.

Example: Here we consider the Bratu-type equation presented in [7]:

$$u'' + e^u = 0 \tag{3.2.1}$$

with the boundary conditions

$$u(0) = 0 \text{ and } u(1) = 0$$

The exact solution of eqn. (3.2.1) is

$$u(x) = -2\ln\left[\frac{\cosh((x-0.5)\,0.758582)}{\cosh(0.379291)}\right].$$

Now, by taking $N = 10$, $h = \frac{1}{10}$, we select a set of equally spaced points $x_0, x_1, \ldots, x_{10}$ on the interval [0,1] by letting $x_i = ih$ with $i = 0, 1, \ldots, 10$.

When the finite difference method is applied on eqn. (3.2.1), we get the non-linear system.

$$
\begin{aligned}
-200w_1+100w_2+e^{w_1} &= 0\\
100w_1-200w_2+100w_3+e^{w_2} &= 0\\
\vdots \quad & \vdots\ \vdots\\
100w_7-200w_8+100w_9+e^{w_8} &= 0\\
100w_8-200w_9+e^{w_9} &= 0
\end{aligned}
$$

By using the mathematica software (see Appendix A) to compute the approximate solution $w_1,\cdots,w_9$, at the nodal points $x_1,\ldots,x_9$, we obtain :

$$
\begin{pmatrix}
0.0498946\\
0.0892777\\
0.117727\\
0.134927\\
0.140682\\
0.134927\\
0.117727\\
0.0892777\\
0.0498946
\end{pmatrix}
$$

From the boundary conditions we have

$$w_0=w_{10}=0.$$

Now, to compute $M_1^*, M_2^*, \cdots, M_9^*$, we use the continuity conditions:

$$M_{i-1}^*+4M_i^*+M_{i+1}^*=\frac{6}{h^2}\left(w_{i-1}-2w_i+w_{i+1}\right),$$

where $i=1,2,\ldots,9$. So, we obtain the linear system

$$
\begin{bmatrix}
4 & 1 & 0 & \cdots & 0 & 0 & 0\\
1 & 4 & 1 & \cdots & 0 & 0 & 0\\
0 & 1 & 4 & \cdots & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & \cdots & 4 & 1 & 0\\
0 & 0 & 0 & \cdots & 1 & 4 & 1\\
0 & 0 & 0 & \cdots & 0 & 1 & 4
\end{bmatrix}
\begin{bmatrix}
M_1^*\\ M_2^*\\ M_3^*\\ \vdots\\ M_7^*\\ M_8^*\\ M_9^*
\end{bmatrix}
=\frac{6}{h^2}
\begin{bmatrix}
w_0-2w_1+w_2\\
w_1-2w_2+w_3\\
w_2-2w_3+w_4\\
\vdots\\
w_6-2w_7+w_8\\
w_7-2w_8+w_9\\
w_8-2w_9+w_{10}
\end{bmatrix}.
$$

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Substitute $w_0, w_1, \cdots, w_{10}$ and $M_1^*, M_2^*, \cdots, M_9^*$, on the cubic spline eqn. (2.5), we get

$$\begin{array}{ll}
s_0^*(x) = -3.38323x^3 + 0.831201x & 0 \le x \le 0.1 \\
s_1^*(x) = -3.38323(0.2-x)^3 - 2.71211(x-0.1)^3 + 0.831201(0.2-x) + 1.45941(x-0.1) & 0.1 \le x \le 0.2 \\
s_2^*(x) = -2.71211(0.3-x)^3 - 3.07823(x-0.2)^3 + 1.45941(0.3-x) + 1.92489(x-0.2) & 0.2 \le x \le 0.3 \\
s_3^*(x) = -3.07823(0.4-x)^3 - 3.10302(x-0.3)^3 + 1.92489(0.4-x) + 2.20567(x-0.3) & 0.3 \le x \le 0.4 \\
s_4^*(x) = -3.10302(0.5-x)^3 - 3.15351(x-0.4)^3 + 2.20567(0.5-x) + 2.30028(x-0.4) & 0.4 \le x \le 0.5 \\
s_5^*(x) = -3.15351(0.6-x)^3 - 3.10302(x-0.5)^3 + 2.30028(0.6-x) + 2.20567(x-0.5) & 0.5 \le x \le 0.6 \\
s_6^*(x) = -3.10302(0.7-x)^3 - 3.07823(x-0.6)^3 + 2.20567(0.7-x) + 1.92489(x-0.6) & 0.6 \le x \le 0.7 \\
s_7^*(x) = -3.07823(0.8-x)^3 - 2.71211(x-0.7)^3 + 1.92489(0.8-x) + 1.45941(x-0.7) & 0.7 \le x \le 0.8 \\
s_8^*(x) = -2.71211(0.9-x)^3 - 2.19995(x-0.8)^3 + 1.45941(0.9-x) + 0.819368(x-0.8) & 0.8 \le x \le 0.9 \\
s_9^*(x) = -2.19995(1.0-x)^3 + 0.819368(1.0-x) & 0.9 \le x \le 1.0.
\end{array}$$

Solve the system using the mathematica software (see Appendix A), we obtain:

$$\begin{pmatrix} -1.31997 \\ -1.02232 \\ -1.14516 \\ -1.13985 \\ -1.15476 \\ -1.13985 \\ -1.14516 \\ -1.02232 \\ -1.31997 \end{pmatrix}$$

Table 1 shows the absolute errors of the numerical results of the proposed algorithm and the Decomposition method presented in [7] for example 3.1. The table indicates that the proposed algorithm is more accurate than the Decomposition method.

Example 2. We Consider here the Bratu-type equation presented in

Table 1

Approximate solution and absolute errors of example (3.1)

x	Exact solution given in [7]	The Proposed Algorithm Solution	Absolute errors for the Decomposition method, [7]	Absolute errors for the proposed Algorithm
0.1	0.0498467900	0.0498946347	2.68510×10^{-3}	4.78447×10^{-5}
0.2	0.0891899350	0.0892776660	2.02194×10^{-3}	8.7731×10^{-5}
0.3	0.1176090956	0.1177268553	1.52342×10^{-3}	1.1776×10^{-5}
0.4	0.1347902526	0.1349266765	2.20175×10^{-3}	1.36424×10^{-5}
0.5	0.1405392142	0.1406819691	3.01547×10^{-3}	1.42755×10^{-5}
0.6	0.1347902526	0.1349266765	2.20175×10^{-3}	1.36424×10^{-5}
0.7	0.1176090956	0.1177268553	1.52342×10^{-3}	1.1776×10^{-5}
0.8	0.0891899350	0.0892776660	2.02194×10^{-3}	8.7731×10^{-5}
0.9	0.0498467900	0.0498946347	2.68510×10^{-3}	4.78447×10^{-5}

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

$$u'' + 2e^{u} = 0 \tag{3.2.2}$$

with the boundary conditions

$$u(0) = 0 \text{ and } u(1) = 0.$$

The exact solution of eqn. (3.2.2) is

$$u(x) = -2 \ln\left[\frac{\cosh((x-0.5)\,1.17878)}{\cosh(0.589388)}\right].$$

Now, by taking $N = 10$, $h = \frac{1}{10}$, we select a set of equally spaced points $x_0, x_1, \ldots, x_{10}$ on the interval [0,1] by letting $x_i = ih$ with $i = 0, 1, \ldots, 10$.

When the finite difference method is applied on eqn. (3.2.2), we get the non-linear system

$$\begin{aligned} -200w_1 + 100w_2 + 2e^{w_1} &= 0 \\ 100w_1 - 200w_2 + 100w_3 + 2e^{w_2} &= 0 \\ \vdots \quad & \vdots\ \vdots \\ 100w_7 - 200w_8 + 100w_9 + 2e^{w_8} &= 0 \\ 100w_8 - 200w_9 + 2e^{w_9} &= 0. \end{aligned}$$

By using the mathematica software to compute the approximate solution $w_1, \cdots, w_9$, at the nodal points we obtain:

$$\begin{pmatrix} 0.114688 \\ 0.206945 \\ 0.274603 \\ 0.315942 \\ 0.32985 \\ 0.315942 \\ 0.274603 \\ 0.206945 \\ 0.114688 \end{pmatrix}.$$

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

From the boundary conditions we have

$$w_0 = w_{10} = 0.$$

Now, to compute $M_1^*, M_2^*, \cdots, M_9^*$, we use the continuity conditions, we get

$$M_{i-1}^* + 4M_i^* + M_{i+1}^* = \frac{6}{h^2}\,(w_{i-1} - 2w_i + w_{i+1}),$$

where $i = 1, 2, \ldots, 9$. So, we obtain the linear system

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} M_1^* \\ M_2^* \\ M_3^* \\ \vdots \\ M_7^* \\ M_8^* \\ M_9^* \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} w_0 - 2w_1 + w_2 \\ w_1 - 2w_2 + w_3 \\ w_2 - 2w_3 + w_4 \\ \vdots \\ w_6 - 2w_7 + w_8 \\ w_7 - 2w_8 + w_9 \\ w_8 - 2w_9 + w_{10} \end{bmatrix},$$

Now, by solving the system using the mathematica software we obtain:

$$\begin{pmatrix} -2.78368 \\ -2.32356 \\ -2.68104 \\ -2.74437 \\ -2.80009 \\ -2.74437 \\ -2.68104 \\ -2.32356 \\ -2.78368 \end{pmatrix}$$

Substitue $w_0, w_1, \cdots, w_{10}$ and $M_1^*, M_2^*, \cdots, M_9^*$, on the cubic spline eqn. (2.4), we get

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

$$
\begin{aligned}
s_0^*(x) &= -4.63946x^3 + 1.19327x && 0 \le x \le 0.1\\
s_1^*(x) &= -4.63946(0.2-x)^3 - 3.8726(x-0.1)^3 + 1.19327(0.2-x) + 2.10817(x-0.1) && 0.1 \le x \le 0.2\\
s_2^*(x) &= -3.8726(0.3-x)^3 - 4.4684(x-0.2)^3 + 2.10817(0.3-x) + 2.79072(x-0.2) && 0.2 \le x \le 0.3\\
s_3^*(x) &= -4.4684(0.4-x)^3 - 4.57395(x-0.3)^3 + 2.79072(0.4-x) + 3.20516(x-0.3) && 0.3 \le x \le 0.4\\
s_4^*(x) &= -4.57395(0.5-x)^3 - 4.66682(x-0.4)^3 + 3.20516(0.5-x) + 3.34516(x-0.4) && 0.4 \le x \le 0.5\\
s_5^*(x) &= -4.66682(0.6-x)^3 - 4.57395(x-0.5)^3 + 3.34516(0.6-x) + 3.20516(x-0.5) && 0.5 \le x \le 0.6\\
s_6^*(x) &= -4.57395(0.7-x)^3 - 4.4684(x-0.6)^3 + 3.20516(0.7-x) + 2.79072(x-0.6) && 0.6 \le x \le 0.7\\
s_7^*(x) &= -4.4684(0.8-x)^3 - 3.8726(x-0.7)^3 + 2.79072(0.8-x) + 2.10817(x-0.7) && 0.7 \le x \le 0.8\\
s_8^*(x) &= -3.8726(0.9-x)^3 - 4.63946(x-0.8)^3 + 2.10817(0.9-x) + 1.19327(x-0.8) && 0.8 \le x \le 0.9\\
s_9^*(x) &= -4.63946(1.0-x)^3 + 1.19327(1.0-x) && 0.9 \le x \le 1.0.
\end{aligned}
$$

Table 2

Approximate solution and absolute errors of example (3.2)

x	Exact solution [8]	The Proposed Algorithm Solution	Absolute errors for the Laplace Decomposition method, [8]	Absolute errors for the Proposed Algorithm
0.1	0.1144107440	0.1146875199	2.1290299×10^{-3}	2.76776×10^{-4}
0.2	0.2064191156	0.2069445813	4.2096994×10^{-3}	5.25466×10^{-4}
0.3	0.2738793116	0.2746033545	6.1868058×10^{-3}	7.24043×10^{-4}
0.4	0.3150893646	0.3159419560	8.0019140×10^{-3}	8.52591×10^{-4}
0.5	0.3289524214	0.3298495446	9.5991920×10^{-3}	8.97123×10^{-4}
0.6	0.3150893646	0.3159419560	$1.09295243 \times 10^{-2}$	8.52591×10^{-4}
0.7	0.2738793116	0.2746033545	$1.19334207 \times 10^{-2}$	7.24043×10^{-4}
0.8	0.2064191156	0.2069445813	$1.23778084 \times 10^{-2}$	5.25466×10^{-4}
0.9	0.1144107440	0.1146875199	$1.08733655 \times 10^{-2}$	2.76776×10^{-4}

Table 2 shows that the absolute errors of the numerical results of the proposed algorithm and the Laplce decomposition method for example (3.2). From the table we can see that the proposed algorithm is more accurate than the Laplace decomposition method.

4. Conclusion

In this paper, we have studied a cubic spline on the finite difference method to find an approximate solution to the Bratu type equation. We presented some numerical examples appeared in the literature for introducing the main idea behind our approach and for comparisons purposes. The numerical results show that our technique has been successfully applied to the Bratu type equation. The results in Tables 1 and 2 show that the proposed algorithm perform better than the Decomposition and Laplace Decomposition methods presented in [7] and [8] respectively.

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

Appendix A

To construct the cubic spline interpolation $S^*(x)$ for equation (2.1): "compute the approximate solution at the nodal points "

Input N

Set $i = 0, 1, \ldots, N$

$$h = \frac{1}{N}$$

$$x_i = ih.$$

$$-\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} + f(x_i, w_i) = 0, \; i = 1, 2, \ldots, N-1$$

$$\mathit{Find\ Root}\left[-\frac{2}{h^2} w_1 + \frac{1}{h^2} w_2 + \lambda e^{w_1} == 0, \frac{1}{h^2}\right.$$

$$w_1 - \frac{2}{h^2} w_2 + \frac{1}{h^2} w_3 + \lambda e^{w_2} == 0,$$

$$\cdots, \frac{1}{h^2} w_{N-2} - \frac{2}{h^2} w_{N-1} + \lambda e^{w_{N-1}} == 0, \{\{w_1, \alpha_1\}, \{w_2, \alpha_2\}, \{w_3, \alpha_3\},$$

$$\left.\cdots, \{w_{N-2}, \alpha_{N-2}\}, \{w_{N-1}, \alpha_{N-1}\}\}\right]$$

"Compute the coefficients of the natural cubic spline $M_1^*, M_2^*, \cdots, M_9^*$"

$$B = \mathit{Sparse\ Array}\,[\{\{i\_, i\_\} \to 4, (\{i\_, j\_\} /; \mathit{Abs}[i - j] == 1 \to 1)\},$$
$$\{N-1, N-1\}]$$

$$\mathit{Linear\ Solve}\left[B, \left\{\frac{6}{h^2}(w_0 - 2w_1 + w_2), \frac{6}{h^2}(w_1 - 2w_2 + w_3), \cdots,\right.\right.$$

$$\left.\left.\frac{6}{h^2}(w_{N-2} - 2w_{N-1} + w_N)\right\}\right]$$

Now, the cubic spline interpolation for eqn. (2.1) us:

$$s_i^*(x) = M_i^* \frac{(x_{i+1} - x)^3}{6h} + M_{i+1}^* \frac{(x - x_i)^3}{6h} + \left(w_i \frac{M_i^* h^2}{6}\right) \frac{(x_{i+1} - x)}{h}$$
$$+ \left(w_{i+1} - \frac{M_{i+1}^* h^2}{6}\right) \frac{(x - x_i)}{h}$$

References

[1] R. Buckmire. Application of a Mickens finite-difference scheme to the cylindrical Bratu–Gelfand problem, *Numer. Methods Partial Differential Equations*, Vol. 20(3), 2004, pp. 327–337.

[2] M. I. Syam, and A. Hamdan. An effcient method for solving Bratu equations, *Appl.Math. Comput.*, Vol. 176, 2006, pp. 704--713.

[3] J. S. McGough. Numerical continuation and the Gelfand problem, *Appl. Math. Comput.*, Vol. 89, 1998, pp. 225–239.

[4] A. S. Mounim, and B. M. de Dormale. From the fitting techniques to accurate schemes for the Liouville–Bratu–Gelfand problem, *Numer. Methods Partial Differential Equations*, Vol. 22(4), 2006, pp. 761–775.

[5] J. Jacobsen, and K. Schmitt, The Liouville–Bratu–Gelfand problem for radial operators, *J. Differential Equations*, Vol. 184, 2002, pp. 283–298.

[6] R. Buckmire. On exact and numerical solutions of the one-dimensional planar Bratu problem, Preprint. Available at http://faculty.oxy.edu/ron/research/bratu/bratu.pdf.

[7] E. Deeba, S. A. Khuri, and S. Xie. An algorithm for solving boundary value problems, *J. Comput. Phys.*, Vol. 159, 2000, pp. 125–138.

[8] S. A. Khuri. A new approach to Bratu's problem, *Appl. Math. Comput.*, Vol. 147, 2004, pp. 131–136.

[9] S. Li, and S. J. Liao. An analytic approach to solve multiple solutions of a strongly nonlinear problem. *Appl. Math. Comput*. Vol. 169, 2005, pp. 854–865.

[10] I. H. Abdel-Halim Hassan, and V. Ertürk. Applying differential transformation method to the one-dimensional planar Bratu problem, *Int. J. Contemp. Math. Sciences*. Vol. 2, 2007, (29-32):1493-1504.

[11] Lin Jin. Application of Modified Variational Iteration Method to the Bratu-Type Problems, *Int. J. Contemp. Math. Sciences*. Vol. 5, 2010, pp. 153–158.

[12] R. L. Burden, and J. D. Faires. Numerical Analysis, seventh edition, Wadsworth Group. Brooks/Cole, Thomson Learning, Inc. 2001.

[13] Y. Q. Wan, Q. Guo and N. Pan. Thermo-electro-hydrodynamic model for electrospinning process, *International Journal of Nonlinear Sciences and Numerical Simulation,* Vol. 5, 2004, pp. 5–8.

[14] O. Ala'yed. MSc. Thesis, An Efficient Algorithm For The Solution Of Bratu-Type Equation Using Cubic Spline Method, Department of Mathematics and Statistics, *Jordan University of Science and Technology*, May 2011.

Received November, 2012

Downloaded by [Jordan Univ. of Science & Tech] at 02:46 05 April 2015

View publication stats