Journal of King Saud University – Science (2018) 30, 91–99

King Saud University
Journal of King Saud University – Science
www.ksu.edu.sa
www.sciencedirect.com

Optimal perturbation iteration method for Bratu-type problems

Sinan Deniz, Necdet Bildik *

Celal Bayar University, Faculty of Arts and Science, Department of Mathematics, 45047 Manisa, Turkey

Received 4 May 2016; accepted 5 September 2016
Available online 14 September 2016

KEYWORDS

Optimal perturbation iteration method;
Perturbation methods;
Bratu-type equations

Abstract In this paper, we introduce the new optimal perturbation iteration method based on the perturbation iteration algorithms for the approximate solutions of nonlinear differential equations of many types. The proposed method is illustrated by studying Bratu-type equations. Our results show that only a few terms are required to obtain an approximate solution which is more accurate and efficient than many other methods in the literature.

1. Introduction

Many nonlinear differential equations are used in many scientific studies and most of them cannot be solved analytically using traditional methods. Therefore these problems are often handled by a broad class of analytical and numerical methods such as Adomian decomposition method (Adomian, 1988; Deniz and Bildik, 2014), Taylor collocation method (Bildik and Deniz, 2015), differential transform method (Bildik and Konuralp, 2006), homotopy perturbation method (Öziş and Ağırseven, 2008), variational iteration method (He, 2003). These methods can give accurate solutions to nonlinear problems but they have also some problems about the convergence region of their series solution. These regions are generally small according to the desired solution. In order to cope with this task, researchers have recently proposed some new methods (Marinca and Herişanu, 2008; Liao, 2012; Idrees et al., 2010). Perturbation iteration method is one of them and has been recently developed by Pakdemirli et.al. It has proven that this method is very effective for solving many nonlinear equations arising in the scientific world (Aksoy and Pakdemirli, 2010; Aksoy et al., 2012; Şenol et al., 2013; Dolapçı et al., 2013; Khalid et al., 2015). In the presented study, we construct a new optimal perturbation iteration method which is applicable to a wide range of equations and does not require special transformations. In order to show the efficiency of the proposed method, we try to solve Bratu initial and boundary value problems which are used in a large variety of applications, such as the fuel ignition model of the theory of thermal combustion, the thermal reaction process model, radioactive heat transfer, nanotechnology and theory of chemical reaction (Doha et al., 2013; He et al., 2014; Raja, 2014).

* Corresponding author.
E-mail addresses: sinan.deniz@cbu.edu.tr (S. Deniz), n.bildik@cbu.edu.tr (N. Bildik).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksus.2016.09.001
1018-3647

2. Perturbation iteration method

Pakdemirli and his co-workers have modified the well-known perturbation method to construct perturbation iteration method (PIM). PIM has been efficiently applied to some strongly nonlinear systems and yields very approximate results (Aksoy et al., 2012; Şenol et al., 2013). In this section; we give basic information about perturbation iteration algorithms. They are classified with respect to the number of correction terms (n) and with respect to the degrees of derivatives in the Taylor expansions (m). Briefly, this process is represented as PIA (n, m).

2.1. PIA (1,1)

In order to illustrate the algorithm, consider a second-order differential equation in closed form:

$$F(y'', y', y, \varepsilon) = 0 \tag{2.1}$$

where $y = y(x)$ and ε is the perturbation parameter. For PIA (1,1), we take one correction term from the perturbation expansion:

$$y_{n+1} = y_n + \varepsilon (y_c)_n \tag{2.2}$$

Substituting (2.2) into (2.1) and then expanding in a Taylor series gives

$$F(y_n'', y_n', y_n, 0) + F_y (y_c)_n \varepsilon + F_{y'} (y_c')_n \varepsilon + F_{y''} (y_c'')_n \varepsilon + F_\varepsilon \varepsilon = 0 \tag{2.3}$$

Rearranging Eq. (2.3) yields a linear second order differential equation:

$$(y_c'')_n + \frac{F_{y'}}{F_{y''}} (y_c')_n + \frac{F_y}{F_{y''}} (y_c)_n = -\frac{\frac{F}{\varepsilon} + F_\varepsilon}{F_{y''}} \tag{2.4}$$

We can easily obtain $(y_c)_0$ from Eq. (2.4) by using an initial guess y_0. Then first approximation y_1 is determined by using this information.

2.2. PIA (1,2)

As distinct from PIA (1,1), we need to take $n = 1,\ m = 2$ to obtain PIA (1,2). In other words, second order derivatives must be taken into consideration:

$$\begin{aligned} &F(y_n'', y_n', y_n, 0) + F_y (y_c)_n \varepsilon + F_{y'} (y_c')_n \varepsilon + F_y'' (y_c'')_n \varepsilon + F_\varepsilon \varepsilon \\ &\quad + \frac{1}{2} \varepsilon^2 F_{y''y''} (y_c'')_n^2 + \frac{1}{2} \varepsilon^2 F_{y'y'} (y_c')_n^2 + \frac{1}{2} \varepsilon^2 F_{yy} (y_c)_n^2 \\ &\quad + \varepsilon^2 F_{y''y'} (y_c'')_n (y_c')_n + \varepsilon^2 F_{y'y} (y_c')_n (y_c)_n + \varepsilon^2 F_{y''y} (y_c'')_n (y_c)_n \\ &\quad + F_{\varepsilon y''} (y_c'')_n \varepsilon^2 + F_{\varepsilon y'} (y_c')_n \varepsilon^2 + F_{\varepsilon y} (y_c)_n \varepsilon^2 + \frac{1}{2} \varepsilon^2 F_{\varepsilon\varepsilon} = 0 \end{aligned} \tag{2.5}$$

or by rearranging

$$\begin{aligned} &(y_c'')_n \left(\varepsilon F_{y''} + \varepsilon^2 F_{\varepsilon y''} \right) + (y_c')_n \left(\varepsilon F_{y'} + \varepsilon^2 F_{\varepsilon y'} \right) + (y_c)_n \left(\varepsilon F_y + \varepsilon^2 F_{\varepsilon y} \right) \\ &\quad + (y_c'')_n^2 \left(\frac{\varepsilon^2}{2} F_{y''y''} \right) + (y_c')_n^2 \left(\frac{\varepsilon^2}{2} F_{y'y'} \right) + (y_c)_n^2 \left(\frac{\varepsilon^2}{2} F_{yy} \right) \\ &\quad + (y_c)_n (y_c')_n \left(\varepsilon^2 F_{y'y} \right) + (y_c'')_n (y_c')_n \left(\varepsilon^2 F_{y'y''} \right) + (y_c'')_n (y_c)_n \left(\varepsilon^2 F_{yy''} \right) \\ &\quad = -F - F_\varepsilon \varepsilon - \frac{\varepsilon^2 F_{\varepsilon\varepsilon}}{2}. \end{aligned} \tag{2.6}$$

Note that all derivatives and functions are calculated at $\varepsilon = 0$. By means of (2.2) and (2.6), iterative scheme is developed for the equation under consideration.

3. Optimal perturbation iteration method

To illustrate the basic concept of the optimal perturbation iteration method (OPIM), we first reconsider Eq. (2.1) as:

$$F(y'', y', y, \varepsilon) = Ly + N(y'', y', y, \varepsilon), \quad B(y, y') = 0 \tag{3.1}$$

where L is a linear operator, N denotes the nonlinear terms and B is a boundary operator respectively. We then expand only nonlinear terms in a Taylor series to decrease the volume of calculations. Because, it is useless and unnecessary to expand the whole equation for each problem. This is the first step of OPIM to decrease the time needed for computations.

After Eqs. (2.4) and (2.6) in the solution processes for PIAs (1,m), we offer to use the formula

$$y_{n+1} = y_n + S_n(\varepsilon)(y_c)_n \tag{3.2}$$

to increase the accuracy of the results and effectiveness of the method. Here $S_n(\varepsilon)$ is an auxiliary function which provides us to adjust and control the convergence. This is the crucial point of OPIM. The choices of functions $S_n(\varepsilon)$ could be exponential, polynomial, etc. In this study, we select auxiliary function in the form

$$S_n(\varepsilon) = C_0 + \varepsilon C_1 + \varepsilon^2 C_2 + \varepsilon^3 C_3 + \cdots = \sum_{i=0}^{n} \varepsilon^i C_i \tag{3.3}$$

where $C_0, C_1, \ldots$ are constants which are to be determined later.

The following algorithm can be used for OPIM:

a) Take the governing differential equation as:

$$Ly + N(y'', y', y, \varepsilon) = 0, \quad y = y(x), \quad a \leqslant x \leqslant b \tag{3.4}$$

b) Substitute (2.2) into the nonlinear part of (3.4) and expand it in a Taylor series:

$$N(y_n'', y_n', y_n, 0) + N_y (y_c)_n \varepsilon + N_{y'} (y_c')_n \varepsilon + N_{y''} (y_c'')_n \varepsilon + N_\varepsilon \varepsilon = 0 \tag{3.5}$$

and

$$\begin{aligned} &N + N_y (y_c)_n \varepsilon + N_{y'} (y_c')_n \varepsilon + N_\varepsilon \varepsilon + N_{\varepsilon y} (y_c)_n \varepsilon^2 + N_{\varepsilon y'} (y_c')_n \varepsilon^2 \\ &+ \tfrac{N_{\varepsilon\varepsilon} \varepsilon^2}{2} + \tfrac{N_{yy} \varepsilon^2 (y_c)_n^2}{2} + \tfrac{N_{y'y'} \varepsilon^2 (y_c')_n^2}{2} = 0 \end{aligned} \tag{3.6}$$

c) After finding $(y_c)_0$ for each algorithm as in PIAs (1,m), substitute it into Eq. (3.2) to find the first approximate result:

$$y_1 = y_0 + S_0(\varepsilon)(y_c)_0 = y_0 + C_0 (y_c)_0 \tag{3.7}$$

By using initial condition and setting $\varepsilon = 1$ yields

$$y_1 = y(x, C_0) \tag{3.8}$$

Using Eq. (3.8) and repeating the similar steps, we have:

$$\begin{aligned} &y_2(x, C_0, C_1) = y_1 + S_1(\varepsilon)(y_c)_0 = y_1 + (C_0 + C_1)(y_c)_1 \\ &y_3(x, C_0, C_1, C_2) = y_2 + (C_0 + C_1 + C_2)(y_c)_2 \\ &\quad \vdots \\ &y_m(x, C_0, \ldots, C_{m-1}) = y_{m-1} + (C_0 + \cdots + C_{m-1})(y_c)_{m-1} \end{aligned} \tag{3.9}$$

d) Substitute the approximate solution y_m into Eq. (3.4) and the general problem results in the following residual:

$$R(x,C_0,\ldots,C_{m-1}) = L(y_m(x,C_0,\ldots,C_{m-1})) + N(y_m(x,C_0,\ldots,C_{m-1})) \tag{3.10}$$

Obviously, when $R(x,C_0,\ldots,C_{m-1})=0$ then the approximation $y_m(x,C_0,\ldots,C_{m-1})=y^{(m)}(x,C_i)$ will be the exact solution. Generally it doesn't happen, especially in nonlinear equations. To determine the optimum values of $C_0,C_1,\ldots$; we here use the equations

$$R(x_1,C_i)=R(x_2,C_i)=\cdots=R(x_m,C_i)=0,\quad i=0,1,\ldots,m-1 \tag{3.11}$$

where $x_i\in(a,b)$. Generally it is quite impossible to solve the system of Eq. (3.11) other than numerically. Therefore, one needs to use a computer program such that Mathematica, Maple etc. Note that the solution of the system (3.11) is not unique, but all obtained constants would yield the same approximate solutions.

The constants $C_0,C_1,\ldots$ can also be defined from the method of least squares:

$$J(C_0,\ldots,C_{m-1}) = \int_a^b R^2(x,C_0,\ldots,C_{m-1})dx \tag{3.12}$$

where a and b are selected from the domain of the problem. Putting these constants into the last one of Eq. (3.9), the approximate solution of order m is well-determined. It should be also emphasized that, Eq. (3.12) is not always useful to find the constants $C_0,C_1,\ldots$ especially for strongly nonlinear equations. So,we use Eq. (3.11) to get those constants in this work. For much more information and different usage about this process, please see Herisanu et al. (2015) and Marinca and Herişanu (2012)

4. Applications

Example 1. Consider the following nonlinear differential equation (Wazwaz, 2005):

$$y''-2e^{y}=0,\quad y(0)=y'(0)=0,\quad 0\leqslant x\leqslant 1. \tag{4.1}$$

which has the exact solution $y=-2\ln(\cos x)$.

4.1. OPIA (1,1)

Consider Eq. (4.1) as:

$$F(y'',y,\varepsilon)=y''-2e^{\varepsilon y}=y''+N(y,\varepsilon). \tag{4.2}$$

OPIA (1,1) requires to compute:

$$N(y_n,0)+N_y(y_n,0)(y_c)_n\varepsilon+N_\varepsilon\varepsilon=0 \tag{4.3}$$

which is approximately half of the volume of calculations that in PIA (1,1). Using Eqs. (2.2), (4.3) and setting $\varepsilon=1$ yields

$$(y_c'')_n=-y_n''+2y_n+2 \tag{4.4}$$

One may start the iteration by taking a trivial solution which satisfies the given initial conditions:

$$y_0=0. \tag{4.5}$$

Substituting (4.5) into Eq. (4.4), we have

$$(y_c)_0=x^2+c \tag{4.6}$$

Now, Eq. (4.6) is inserted into Eq. (3.2) and applying the initial conditions we get

$$y_1=y_0+S_n(\varepsilon)(y_c)_0=C_0x^2 \tag{4.7}$$

It is worth mentioning that y_1 does not represent the first correction term; rather it is the approximate solution after the first iteration. Following the same procedure, we obtain new and more approximate results:

$$y_2=C_0x^2+(C_0+C_1)\left(x^2-C_0x^2+\frac{C_0x^4}{6}\right) \tag{4.8}$$

$$\begin{aligned} y_3 = {} & [2C_0+C_1-C_0(C_0+C_1)+(-1+C_0)(-1+C_0\\ & +C_1)(C_0+C_1+C_2)]x^2\\ & +\left[\frac{1}{6}C_0(C_0+C_1)+\frac{1}{15}\left(-5C_0^2-5C_0(-1+C_1)\right.\right.\\ & \left.\left.+\frac{5C_1}{2}\right)(C_0+C_1+C_2)\right]x^4\\ & +\left[\frac{1}{90}C_0(C_0+C_1)(C_0+C_1+C_2)\right]x^6 \end{aligned} \tag{4.9}$$

To determine the constants, we proceed as in Section 3. First, the residual

$$\begin{aligned} R(x,C_0,C_1,C_2) = {} & L(y_3(x,C_0,C_1,C_2))+N(y_3(x,C_0,C_1,C_2))=2C_0+(C_0+C_1)(2-2C_0+2C_0x^2)\\ & +\frac{(C_0+C_1+C_2)}{15} times\left[30(-1+C_0)(-1+C_0+C_1)+12\left(-5C_0^2-5C_0(-1+C_1)+\frac{5C_1}{2}\right)x^2+5C_0(C_0+C_1)x^4\right]\\ & -2Exp\left[\begin{array}{l} C_0x^2+(C_0+C_1)\left(x^2-C_0x^2+\frac{C_0x^4}{6}\right)+\frac{1}{6}C_0(C_0+C_1)x^6\\ \frac{(C_0+C_1+C_2)}{15}(15(-1+C_0)(-1+C_0+C_1)x^2+\left(-5C_0^2-5C_0(-1+C_1)+\frac{5C_1}{2}\right)x^4) \end{array}\right] \end{aligned} \tag{4.10}$$

is constructed for the third order approximation. Using Eq. (3.11) with $x=0.3,0.6,0.9$, we get

$$\begin{aligned} & C_0=1.00096007239,\ C_1=0.034138423506,\\ & C_2=-0.049127633506 \end{aligned} \tag{4.11}$$

Inserting the constants into Eq. (4.9), we obtain the approximate solution of the third order:

$$y_3(x) = 1.00112456947x^2 + 0.152984774463x^4 + 0.076778117636x^6 \quad (4.12)$$

Note that some complex numbers arise from solving Eq. (4.10). They can also be used instead of C_0, C_1, C_2 to get the same result. We here give only real solutions for simplicity.

4.2. OPIA (1,2)

One can construct the OPIA (1,2) by taking one correction term in the perturbation expansion and two derivatives in the Taylor series. Note that one needs to enter the data in Eq. (2.4) into the computer for PIA (1,2). But, it is sufficient to use

$$N + N_y(y_c)_n\varepsilon + N_\varepsilon\varepsilon + N_{\varepsilon y}(y_c)_n\varepsilon^2 + \frac{N_{\varepsilon\varepsilon}\varepsilon^2}{2} + \frac{N_{yy}\varepsilon^2(y_c)_n^2}{2} = 0 \quad (4.13)$$

for OPIA (1,2). After making the relevant calculations, the algorithm takes the simplified form:

$$(y_c'')_n - 2(y_c)_n = -y_n'' + 2y_n + y_n^2 + 2 \quad (4.14)$$

Using the trivial solution $y_0 = 0$, we have second order problem

$$(y_c'')_0 - 2(y_c)_0 = 2 \quad (4.15)$$

Using Eqs. (3.2), (4.15) and the initial conditions, we obtain

$$y_1 = C_0\left(\cosh\left(\sqrt{2}x\right) - 1\right) \quad (4.16)$$

Following the same procedure using (4.16), the second iteration is obtained as

$$y_2 = \frac{1}{3}\begin{pmatrix} -3C_0 + 3C_0\cosh[\sqrt{2}x] - \frac{3C_0{}^2(C_0+C_1)x\sinh[\sqrt{2}x]}{\sqrt{2}} \\ +\left((C_0+C_1)\left(6 + C_0(-6+5C_1) + C_0^2\cosh[\sqrt{2}x]\right)\right)\sinh\left[\frac{x}{\sqrt{2}}\right]^2 \end{pmatrix} \quad (4.17)$$

One can easily realize that, we have functional expansion for OPIA (1,2) instead of a polynomial expansion.

Following the same procedure, from the residual

$$R(x, C_0, C_1) = L(y_2) + N(y_2) = \frac{1}{3}\begin{bmatrix} -2(-3C_1 + C_0(3(-2+C_1) + C_0(3 + C_0 + C_1)))\cosh[\sqrt{2}x] \\ +C_0^2(C_0 + C_1)\left(2\cosh[2\sqrt{2}x] - 3\sqrt{2}x\sinh[\sqrt{2}x]\right) \end{bmatrix} - 2Exp\left[\frac{1}{3}\begin{pmatrix} (C_0+C_1)\left(6 + C_0(-6+5C_0) + C_0^2\cosh[\sqrt{2}x]\right)sinh[\frac{x}{\sqrt{2}}]^2 \\ -\frac{3C_0{}^2(C_0+C_1)x\sinh[\sqrt{2}x]}{\sqrt{2}} - 3C_0 + 3C_0\cosh[\sqrt{2}x] \end{pmatrix}\right] \quad (4.18)$$

the constants C_0 and C_1 can be determined as

$$C_0 = 1.000861120478, \quad C_1 = 0.0266135748038 \quad (4.19)$$

Thus, we have the second-order approximate solution:

$$y_2(x) = -1.1784311655118591x\sinh(\sqrt{2}x) + 2.1406095945289634\cosh(\sqrt{2}x) + 0.13215241067298802\cosh(2\sqrt{2}x) - 2.2728821098149927 \quad (4.20)$$

One can also compute more approximate results by following the same procedure with a computer program. We do not give higher iterations due to huge amount of calculations. Fig. 1 and Table 1 shows a comparison of OPIAs and exact solution. It is clear that the results obtained by OPIM are more accurate than those of PIM in Aksoy and Pakdemirli (2010).

Example 2. Bratu's first boundary value problem is given as (Wazwaz, 2005):

$$y'' + \lambda e^y = 0, \quad 0 \leqslant x \leqslant 1, \quad y(0) = y(1) = 0 \quad (4.21)$$

with the exact solution $y(x) = -2\ln\left[\frac{\cosh\left(\left(x-\frac{1}{2}\right)\frac{\theta}{2}\right)}{\cosh\left(\frac{\theta}{4}\right)}\right]$ where θ satisfies

$$\theta = \sqrt{2\lambda}\cosh\left(\frac{\theta}{4}\right).$$

4.3. OPIA (1,1)

An artificial perturbation parameter is inserted for Eq. (4.21) as follows

$$F(y'', y, \varepsilon) = y'' + \lambda e^{\varepsilon y} = Ly + N(y, \varepsilon) = 0. \quad (4.22)$$

By making necessary computations using Eqs. (2.2), (4.3) and setting $\varepsilon = 1$, we easily get

$$(y_c'')_n = -\lambda y_n - \left(y_n'' + \lambda\right). \quad (4.23)$$

One may start with the trivial solution

$$y_0 = 0 \quad (4.24)$$

and using Eq. (3.2) the iterations are reached as follows:

$$y_1 = -\frac{\lambda C_0}{2}\left(x^2 - x\right) \quad (4.25)$$

$$y_2 = -\frac{\lambda C_0}{2}(x^2 - x) + \frac{(C_0 + C_1)}{24}(-1 + x)x\lambda\left[-12 + \left(12 + (-1 - x + x^2)\lambda\right)C_0\right] \quad (4.26)$$

$$y_3 = \frac{\lambda x}{720} \times \left[\begin{array}{l} 30(C_0 + C_1)x(-12 + C_0(12 + \lambda(-2 + x)x)) - 360C_0(-1 + x) - \\ (C_0 + C_1 + C_2)x \left\{ \begin{array}{l} 360(-1 + C_0)(-1 + C_0 + C_1) - 60C_0(-1 + C_0 + C_1)\lambda x \\ +30(-C_1 + 2C_0(-1 + C_0 + C_1))\lambda x^2 - 3C_0(C_0 + C_1)\lambda^2 x^3 \\ +C_0(C_0 + C_1)\lambda^2 x^4 \end{array} \right\} \end{array} \right] \tag{4.27}$$

For the constants C_0, C_1 and C_2, the method given in Section 3.(d) is used, and we obtain the following values for $\lambda = 1$:

$$C_0 = 0.00896621251, \;\; C_1 = 0.086955412771,$$
$$C_2 = -0.000213669444 \tag{4.28}$$

for the $x_i = 0.3, 0.6, 0.9$. Thus, the approximate solution of the third order is:

$$\begin{aligned} y_3(x) = &0.549359811237294x - 0.5001682773565349x^2 \\ &- 0.09044598683883211x^3 \\ &+ 0.02537329293225415 8x^4 + 0.02382881872172111 5x^5 \\ &- 0.00794805048551452x^6 \end{aligned} \tag{4.29}$$

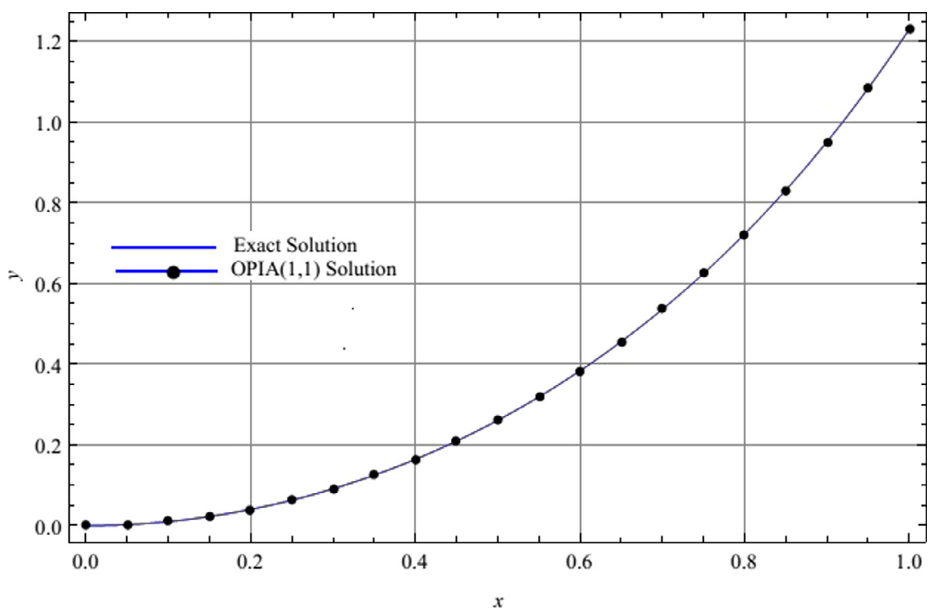

Fig. 1 Comparison between the three-term OPIA (1,1) approximate solution and the exact solution for Example 1.

Table 1 Comparison of absolute errors of Example 1 at different orders of approximations.

x	Absolute errors for OPIA (1,1) solutions			Absolute errors for OPIA (1,2) solutions		Exact solution
	$\lvert y - y_1\rvert$	$\lvert y - y_2\rvert$	$\lvert y - y_3\rvert$	$\lvert y - y_1\rvert$	$\lvert y - y_2\rvert$	$y = -2\ln(\cos x)$
0.1	0.000449452	0.000169553	9.9097×10^{-6}	2.402×10^{-6}	9.4728×10^{-6}	0.010016711
0.2	0.001595127	0.000583911	2.5126×10^{-5}	9.453×10^{-6}	3.3152×10^{-5}	0.040269546
0.3	0.002812140	0.000976872	1.3047×10^{-5}	1.9420×10^{-5}	2.7254×10^{-5}	0.091383311
0.4	0.003000543	0.000963454	4.7213×10^{-5}	2.4899×10^{-6}	4.4563×10^{-6}	0.164458038
0.5	0.000485555	0.000139394	0.000126132	4.916×10^{-6}	5.55112×10^{-8}	0.261168480
0.6	0.007148548	0.001752633	0.000116507	8.8755×10^{-5}	7.2047×10^{-5}	0.383930338
0.7	0.023329621	0.004551758	0.000144037	0.000354849	7.0044×10^{-5}	0.536171515
0.8	0.052947212	0.007229526	0.000727717	0.000982654	0.000128213	0.722781493
0.9	0.103126097	0.007116353	0.001202366	0.002323371	0.000452361	0.950884887
1	0.184637089	0.001509956	0.000365479	0.005024005	4.44089×10^{-8}	1.231252940

4.4. OPIA (1,2):

One just needs to construct

$$N + N_y(y_c)_n\varepsilon + N_\varepsilon\varepsilon + N_{\varepsilon y}(y_c)_n\varepsilon^2 + \frac{N_{\varepsilon\varepsilon}\varepsilon^2}{2} + \frac{N_{yy}\varepsilon^2(y_c)_n^2}{2} = 0 \tag{4.30}$$

where

$$N(y,\varepsilon) = \lambda e^{\varepsilon y}. \tag{4.31}$$

After making the relevant calculations, the algorithm takes the simplified form:

$$(y_c'')_n + \lambda(y_c)_n = -\lambda y_n - y_n'' - \lambda - \frac{\lambda}{2}y_n^2. \tag{4.32}$$

Using Eqs. (3.2), (4.24), (4.32) and the initial conditions, we obtain

$$y_1 = C_0\left(-1 + \cos[x\sqrt{\lambda}] + \sin[x\sqrt{\lambda}]\tan\left[\frac{\sqrt{\lambda}}{2}\right]\right) \tag{4.33}$$

$$\begin{aligned}
y_2 = {} & C_0\left(-1 + \cos[x\sqrt{\lambda}] + \sin[x\sqrt{\lambda}]\tan[\tfrac{\sqrt{\lambda}}{2}]\right) + \tfrac{(C_0+C_1)}{48}\left(\sec[\tfrac{\sqrt{\lambda}}{2}]^2(-24(1 - C_0 + C_0^2)\cos[x\sqrt{\lambda}]^2\right) \\
& +(C_0+C_1)\left(\begin{array}{l}(12 - 12C_0 + C_0^2)\cos[\sqrt{\lambda}] - C_0^2\cos[(1-3x)\sqrt{\lambda}] + 3(4 - 4C_0 + 3C_0^2 \\ +C_0^2\cos[(1-2x)\sqrt{\lambda}] + 2(-1 + C_0)\cos[(-1+x)\sqrt{\lambda}] + C_0^2\cos[2x\sqrt{\lambda}] \\ -2\cos[(1+x)\sqrt{\lambda}] + 2C_0\cos[(1+x)\sqrt{\lambda}] - C_0^2\cos[(1+x)\sqrt{\lambda}] \\ -2C_0^2x\sqrt{\lambda}\sin[\sqrt{\lambda}])\end{array}\right) \\
& +2(C_0+C_1)\cos[x\sqrt{\lambda}] + (C_0+C_1)\sec[\tfrac{\sqrt{\lambda}}{2}]\sin[x\sqrt{\lambda}]\times \\
& \times\left[\begin{array}{l}6C_0^2(-2+3x)\sqrt{\lambda}\cos[\tfrac{\sqrt{\lambda}}{2}] + 6C_0^2x\sqrt{\lambda}\cos[\tfrac{3\sqrt{\lambda}}{2}] + 12\sin[\tfrac{\sqrt{\lambda}}{2}] - 12C_0\sin[\tfrac{\sqrt{\lambda}}{2}] + 17C_0^2\sin[\tfrac{\sqrt{\lambda}}{2}] \\ +12\sin[\tfrac{3\sqrt{\lambda}}{2}] - 12C_0\sin[\tfrac{3\sqrt{\lambda}}{2}] + C_0^2\sin[\tfrac{3\sqrt{\lambda}}{2}] + C_0^2 Sin[\tfrac{1}{2}(1-6x)\sqrt{\lambda}] - 6C_0^2\sin[\tfrac{1}{2}(1-4x)\sqrt{\lambda}] \\ -3C_0^2\sin[\tfrac{1}{2}(3-4x)\sqrt{\lambda}] + 18\sin[\tfrac{1}{2}(1-2x)\sqrt{\lambda}] - 18C_0\sin[\tfrac{1}{2}(1-2x)\sqrt{\lambda}] + 18C_0^2\sin[\tfrac{1}{2}(1-2x)\sqrt{\lambda}] \\ +C_0^2\sin[\tfrac{3}{2}(1-2x)\sqrt{\lambda}] + 6\sin[\tfrac{1}{2}(3-2x)\sqrt{\lambda}] - 6C_0\sin[\tfrac{1}{2}(3-2x)\sqrt{\lambda}] + 6C_0^2\sin[\tfrac{1}{2}(3-2x)\sqrt{\lambda}] - \\ 18\sin[\tfrac{1}{2}(1+2x)\sqrt{\lambda}] + 18C_0\sin[\tfrac{1}{2}(1+2x)\sqrt{\lambda}] - 15C_0^2\sin[\tfrac{1}{2}(1+2x)\sqrt{\lambda}] \\ -6\sin[\tfrac{1}{2}(3+2x)\sqrt{\lambda}] + 6C_0\sin[\tfrac{1}{2}(3+2x)\sqrt{\lambda}] - 3A^2\sin[\tfrac{1}{2}(3+2x)\sqrt{\lambda}] + 3C_0^2\sin[\tfrac{1}{2}(1+4x)\sqrt{\lambda}]\end{array}\right]
\end{aligned} \tag{4.34}$$

For the constants C_0 and C_1 in Eq. (4.34), we proceed as earlier and get

$$C_0 = -1.0002036577189,\ \ C_1 = 0.099502786321 \tag{4.35}$$

for $\lambda = 1$. Thus, we have the second-order approximate solution:

$$\begin{aligned}
y_2(x) = {} & -1.078485122090 - 0.004293531433x \\
& + 1.105765327206\cos[x] - 0.0279349653844\cos[2x] \\
& + 0.00065493410502\cos[3x] + 0.6114581430450\sin[x] \\
& - 0.091877388999\cos[x]\sin[x] \\
& + 0.011352766729676\sin[3x]
\end{aligned} \tag{4.36}$$

for OPIA (1,2). It can be readily seen from Fig. 2 and Table 2 that approximate solutions obtained by the OPIAs are identical with that given by the analytical methods (Wazwaz, 2005). Note that more components in the solution series can be computed to enhance the approximation.

Example 3. Consider Bratu's second boundary value problem (Wazwaz, 2005)

$$y'' + \pi^2 e^{-y} = 0,\quad 0 \leqslant x \leqslant 1,\quad y(0) = y(1) = 0. \tag{4.37}$$

Exact solution of this problem is mistakenly given as

$$y(x) = \ln[1 + \sin(1 + \pi x)] \tag{4.38}$$

in Wazwaz (2005) and Batiha (2010), whereas the correct exact solution is

$$y(x) = \ln[1 + \sin(\pi x)]. \tag{4.39}$$

4.5. OPIA (1,1)

By rearranging Eq. (4.37) as

$$F(y'', y, \varepsilon) = y'' + \pi^2 e^{-\varepsilon y} = Ly + N(y,\varepsilon) \tag{4.40}$$

and using Eqs. (2.2) and (4.3) with $\varepsilon = 1$, we have

$$(y_c'')_n = \pi^2 y_n - (y_n'' + \pi^2). \tag{4.41}$$

Without going into details here, we just give the successive iterations:

$$y_0 = 0 \tag{4.42}$$

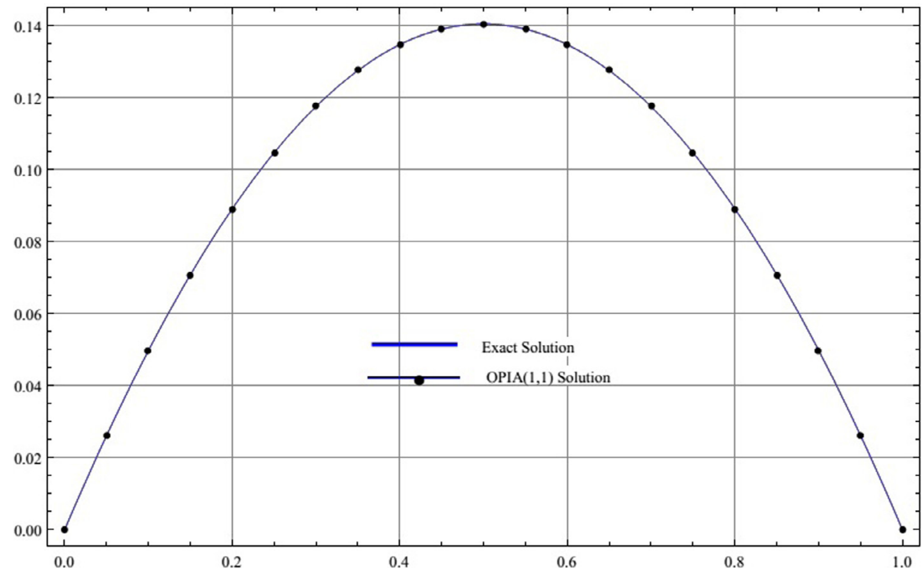

Fig. 2 Comparison between the three-term OPIA (1,1) approximate solution and the exact solution for Example 2.

Table 2 Comparison of absolute errors of Example 2 at different orders of approximations.

x	Absolute errors for OPIA (1,1) solutions			Absolute errors for OPIA (1,2) solutions		Exact solution
	$\lvert y-y_1\rvert$	$\lvert y-y_2\rvert$	$\lvert y-y_3\rvert$	$\lvert y-y_1\rvert$	$\lvert y-y_2\rvert$	for $\lambda=1$
0.1	1.05236×10^{-6}	8.05698×10^{-7}	1.19748×10^{-7}	5.22201×10^{-10}	1.23154×10^{-16}	0.0498465
0.2	1.08547×10^{-5}	7.5067×10^{-7}	3.35942×10^{-8}	7.90215×10^{-9}	2.36014×10^{-15}	0.0891894
0.3	4.96318×10^{-5}	1.00521×10^{-6}	1.12813×10^{-8}	5.20476×10^{-9}	5.10365×10^{-13}	0.1176084
0.4	9.55681×10^{-5}	5.96014×10^{-8}	9.08115×10^{-9}	2.63391×10^{-11}	5.30158×10^{-15}	0.1347894
0.5	7.56419×10^{-6}	7.22085×10^{-8}	7.33394×10^{-10}	9.89661×10^{-10}	5.60972×10^{-15}	0.1405383
0.6	0.000121368	5.00123×10^{-7}	1.13418×10^{-9}	2.00569×10^{-11}	9.12054×10^{-13}	0.1347894
0.7	0.000802364	4.20161×10^{-6}	6.13948×10^{-9}	4.11057×10^{-11}	2.03606×10^{-13}	0.1176084
0.8	0.000110879	1.00907×10^{-5}	1.00907×10^{-8}	8.05698×10^{-10}	7.45236×10^{-12}	0.0891894
0.9	0.000569203	2.10102×10^{-5}	7.75262×10^{-8}	2.05471×10^{-9}	1.00612×10^{-12}	0.0498465

$$y_1=\frac{\pi^2C_0}{2}\left(x-x^2\right) \tag{4.43}$$

$$y_2=\frac{\pi^2C_0}{2}\left(x-x^2\right)-\frac{(C_0+C_1)}{24}(-x+x^2)\pi^2\left[-12+\left(12+(-1-x+x^2)\pi^2\right)C_0\right] \tag{4.44}$$

Proceeding as earlier we find constants $C_0,\ C_1$ and C_2:

$$C_0=0.00839960142,\ C_1=0.08178563321,$$
$$C_2=-0.000193602314 \tag{4.46}$$

Inserting the constants into Eq. (4.45), we obtain the approximate solution of the third order:

$$y_3=-\frac{\pi^2C_0}{2}(-x+x^2)-\frac{x\pi^2}{24}(C_0+C_1)(-1+x)\left(12+C_0\left(-12+(-1+(-1+x)x)\pi^2\right)\right)+\frac{(C_0+C_1+C_2)\pi^2}{720}\begin{bmatrix}-360(-1+C_0)(-1+C_0+C_1)(-1+x)x\\+30(-C_1+2C_0(-1+C_0+C_1))(x-2x^3+x^4)\pi^2\\-C_0(C_0+C_1)x(-3+5x^2-3x^4+x^5)\pi^4\end{bmatrix} \tag{4.45}$$

Table 3 Comparison of absolute errors of Example 3 at different orders of approximations.

x	Absolute errors for OPIA (1,1) solutions			Absolute errors for OPIA (1,2) solutions		Exact solution
	$\lvert y-y_1\rvert$	$\lvert y-y_2\rvert$	$\lvert y-y_3\rvert$	$\lvert y-y_1\rvert$	$\lvert y-y_2\rvert$	
0.1	0.0000752784	0.0000608251	0.0000719575	9.05621×10^{-7}	8.8864×10^{-7}	0.269276469
0.2	0.0004108547	0.0001009657	0.0000183205	4.03657×10^{-7}	1.88504×10^{-7}	0.462340122
0.3	0.0000296314	0.0000723684	4.31405×10^{-6}	3.99521×10^{-7}	1.23351×10^{-7}	0.592783600
0.4	0.0000955682	0.0000135841	5.90773×10^{-6}	2.60399×10^{-6}	1.18017×10^{-7}	0.668371029
0.5	0.0002856413	2.90365×10^{-5}	1.28998×10^{-6}	3.05668×10^{-6}	2.73484×10^{-7}	0.693147180
0.6	0.0000213685	8.10269×10^{-6}	1.07103×10^{-6}	8.70569×10^{-7}	2.68334×10^{-7}	0.668371029
0.7	0.0000723646	9.30855×10^{-6}	1.1606×10^{-6}	5.19005×10^{-7}	9.46642×10^{-8}	0.592783600
0.8	0.0000108799	9.99237×10^{-6}	2.05027×10^{-6}	8.05111×10^{-7}	4.36073×10^{-7}	0.462340122
0.9	0.0005692033	0.000111947	0.0000523313	1.22014×10^{-6}	5.3818×10^{-7}	0.269276469

$$\begin{aligned} y_3(x) =& 3.134717936805843x - 4.811906503098512x^2 \\ &+ 4.266200757140372x^3 \\ &- 4.407364172185969x^4 + 2.7222682887263185x^5 \\ &- 0.9036537597026911x^6 \end{aligned} \tag{4.47}$$

4.6. OPIA (1,2)

After making the relevant calculations, the algorithm

$$N + N_y(y_c)_n\varepsilon + N_\varepsilon\varepsilon + N_{\varepsilon y}(y_c)_n\varepsilon^2 + \frac{N_{\varepsilon\varepsilon}\varepsilon^2}{2} + \frac{N_{yy}\varepsilon^2(y_c)_n^2}{2} = 0 \tag{4.48}$$

reduces to

$$(y_c'')_n - \pi^2(y_c)_n = \pi^2 y_n - \frac{\pi^2}{2}y_n^2 - y_n'' - \pi^2. \tag{4.49}$$

Using Eqs. (3.2), (4.49) and the initial conditions, we get

$$y_0 = 0 \tag{4.50}$$

$$y_1 = C_0\left(-1 + \cosh[\pi x] + \sinh[\pi x]\tanh\left[\frac{\pi}{2}\right]\right) \tag{4.51}$$

$$\begin{aligned} y_2(x) = & C_0\left(-1 + \cosh[\pi x] + \sinh[\pi x]\tanh\left[\frac{\pi}{2}\right]\right) + \frac{e^{-2\pi x}(-1+e^\pi)}{12(1+e^\pi)^2}(C_0+C_1)(-1+\coth[\pi])\times \\ & \times\left[\begin{array}{l} -6e^{\pi x}(1+e^\pi)^2(-1+e^{\pi x})(-e^\pi+e^{\pi x}) + 6Ae^{\pi x}(1+e^\pi)^2(-1+e^{\pi x})(-e^\pi+e^{\pi x}) \\ -C_0^2\left\{\begin{array}{l} e^{2\pi}+e^{3\pi}-3e^{2\pi x}+e^{4\pi x}-15e^{2\pi(1+x)}-3e^{\pi(3+2x)}-15e^{\pi+2\pi x}+e^{\pi+4\pi x} \\ +e^{\pi+\pi x}(2+3\pi(-1+x))+e^{3\pi x}(2-3\pi x)+3e^{\pi+3\pi x}(4+\pi-2\pi x)+ \\ e^{\pi(3+x)}(2+3\pi x)+3e^{\pi(2+x)}(4+\pi(-1+2x))+e^{\pi(2+3x)}(2-3\pi(-1+x)) \end{array}\right\} \end{array}\right] \end{aligned} \tag{4.52}$$

for OPIA (1,2). Using Eq. (3.11), the following values of C_0 and C_1 are obtained:

$$C_0 = -1.0286083214317654,\ C_1 = 2.029583070812236 \tag{4.53}$$

By using the above values, the approximate solution of the second order is:

$$\begin{aligned} y_2(x) =& 1.56204116701300 \\ &+ (-1.4080552742209 - 1.4777172978873x)\cosh[\pi x] \\ &+ 0.15341644132458\sinh[2\pi x] \\ &- 0.15399004069623\cosh[2\pi x] \\ &+ (1.163463843313 + 1.6111742346200355x)\sinh[\pi x]. \end{aligned} \tag{4.54}$$

One can easily observe from Table 3 and Fig. 3 that the results agree very well with the exact solution.

5. Conclusions

In this paper, a new technique OPIM is employed for the first time to obtain a new analytic approximate solution of Bratu-type differential equations. This new method provides us with an easy way to optimally control and adjust the convergence

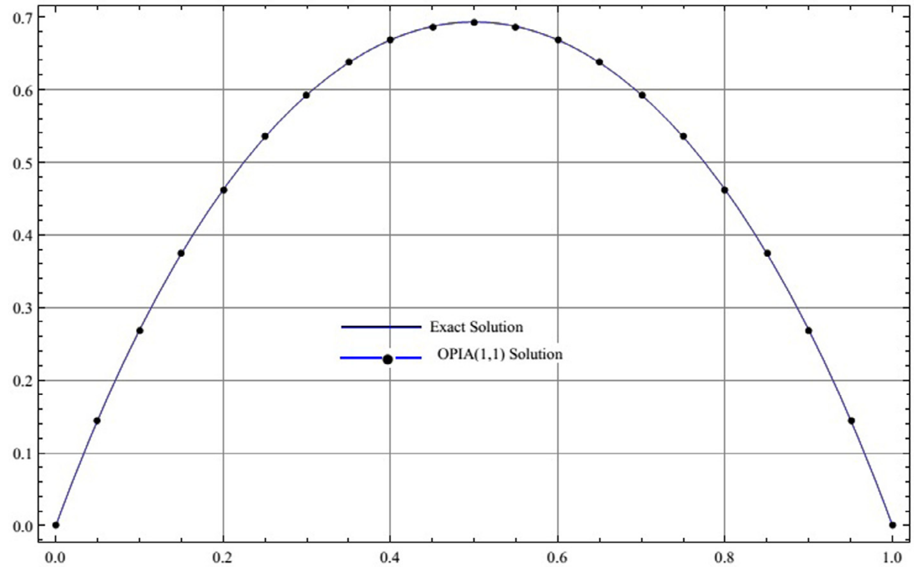

Fig. 3 Comparison between the three-term OPIA (1,1) approximate solution and the exact solution for Example 3.

solution series. OPIM gives a very good approximation even in a few terms which converges to the exact solution. This fact is obvious from the use of the auxiliary function $S_n(\varepsilon)$ which depends on n coefficients $C_0,\ C_1,\dots,C_n$. The results obtained in this paper confirm that the OPIM is a powerful and efficient technique for finding nearly exact solutions for differential equations which have great significance in many different fields of science and engineering.

References

Adomian, G., 1988. A review of the decomposition method in applied mathematics. J. Math. Anal. Appl. 135 (2), 501–544.

Aksoy, Y., Pakdemirli, M., 2010. New perturbation–iteration solutions for Bratu-type equations. Comput. Math. Appl. 59 (8), 2802–2808.

Aksoy, Y., Pakdemirli, M., Abbasbandy, S., Boyaci, H., 2012. New perturbation–iteration solutions for nonlinear heat transfer equations. Int. J. Numer. Methods Heat Fluid Flow 22 (7), 814–828.

Batiha, B., 2010. Numerical solution of Bratu-type equations by the variational iteration method. Hacettepe J. Math. Stat. 39 (1).

Bildik, N., Deniz, S., 2015. Comparison of solutions of systems of delay differential equations using taylor collocation method, Lambert W function and variational iteration method. Sci. Iran. Trans. D, Comput. Sci. Eng. Electr. 22 (3), 1052.

Bildik, N., Konuralp, A., 2006. The use of variational iteration method, differential transform method and Adomian decomposition method for solving different types of nonlinear partial differential equations. Int. J. Nonlinear Sci. Numer. Simul. 7 (1), 65–70.

Deniz, S., Bildik, N., 2014. Comparison of Adomian decomposition method and Taylor matrix method in solving different kinds of partial differential equations. Int. J. Model. Optim. 4 (4), 292.

Doha, E., Bhrawy, A., Baleanu, D., Hafez, R., 2013. Efficient Jacobi-gauss collocation method for solving initial value problems of Bratu type. Comput. Math. Math. Phys. 53 (9), 1292–1302.

He, J.-H., 2003. Variational approach to the Lane–Emden equation. Appl. Math. Comput. 143 (2), 539–541.

He, J.-H., Kong, H.-Y., Chen, R.-X., Hu, M.-S., Chen, Q.-L., 2014. Variational iteration method for Bratu-like equation arising in electrospinning. Carbohydrate Polym. 105, 229–230.

Herisanu, N., Marinca, V., Madescu, G., 2015. An analytical approach to non-linear dynamical model of a permanent magnet synchronous generator. Wind Energy 18 (9), 1657–1670.

Idrees, M., Islam, S., Haq, S., Islam, S., 2010. Application of the optimal homotopy asymptotic method to squeezing flow. Comput. Math. Appl. 59 (12), 3858–3866.

Khalid, M., Sultana, M., Zaidi, F., Khan, F.S., 2015. Solving polluted lakes system by using perturbation-iteration method. Int. J. Comput. Appl. 114 (4).

Liao, S., 2012. Optimal homotopy analysis method. In: Homotopy Anal. Method Nonlinear Differ. Equ. Springer, pp. 95–129.

Marinca, V., Herişanu, N., 2008. Application of optimal homotopy asymptotic method for solving nonlinear equations arising in heat transfer. Int. Commun. Heat Mass Transfer 35 (6), 710–715.

Marinca, V., Herişanu, N., 2012. Optimal parametric iteration method for solving multispecies Lotka-Volterra equations. Discr. Dyn. Nat. Soc.

Öziş, T., Ağrseven, D., 2008. He's homotopy perturbation method for solving heat-like and wave-like equations with variable coefficients. Phys. Lett. A 372 (38), 5944–5950.

Raja, M.A.Z., 2014. Solution of the one-dimensional Bratu equation arising in the fuel ignition model using ann optimised with PSO and SQP. Connection Sci. 26 (3), 195–214.

Senol, M., Timuçin Dolapç, I., Aksoy, Y., Pakdemirli, M., 2013. Perturbation–iteration method for first-order differential equations and systems. In: Abstract and Applied Analysis, vol. 2013. Hindawi Publishing Corporation.

Timuçin Dolapç, I., Senol, M., Pakdemirli, M., 2013. New perturbation iteration solutions for Fredholm and Volterra integral equations. J. Appl. Math.

Wazwaz, A.-M., 2005. Adomian decomposition method for a reliable treatment of the Bratu-type equations. Appl. Math. Comput. 166 (3), 652–663.