Acta Universitatis Apulensis No. 22/2010
ISSN: 1582-5329 pp. 65-78

THE EXTENDED ADOMIAN DECOMPOSITION METHOD FOR FOURTH ORDER BOUNDARY VALUE PROBLEMS

G. Ebadi and S. Rashedi

Abstract. In this paper, we use an efficient numerical algorithm for solving two point fourth-order linear and nonlinear boundary value problems, which is based on the Adomian decomposition method (ADM), namely, the extended ADM (EADM). The proposed method is examined by comparing the results with other methods. Numerical results show that the proposed method is much more efficient and accurate than other methods with less computational work.

2000 *Mathematics Subject Classification*: 65l05; 65l99.

1. Introduction

In recent years, much attention have been given to solve the fourth-order boundary value problems, which have application in various branches of pure and applied sciences. Various numerical methods including finite difference, B-spline [7], homotopy perturbation method [11], variational iteration method [14] and Adomian decomposition method [8, 13] were developed for solving fourth-order boundary value problems . To be more precise, we consider the following fourth-order boundary value problem

$$u^{(4)}(x) = f(x, u, u', u'', u'''), \tag{1}$$

with the boundary conditions

$$\begin{aligned} u(a) &= \alpha_1, \qquad u'(a) = \alpha_2, \\ u(b) &= \beta_1, \qquad u'(b) = \beta_2, \end{aligned} \tag{2}$$

or

$$\begin{aligned} u(a) &= \alpha_1, \qquad u''(a) = \alpha_2, \\ u(b) &= \beta_1, \qquad u''(b) = \beta_2, \end{aligned} \tag{3}$$

where f is a continuous function on [a,b] and the parameters α_i and β_i, i = 1,2 are finite real arbitrary constants.

Adomian [1, 2] introduced an approximate analytical method that allows the solution of nonlinear functional equations without linearization or assumption of infinitesimally small parameters. Since its discovery, Adomians decomposition method (ADM) has been shown by several investigators [1, 2, 12, 16] to be extremely efficient and versatile. The ADM yields rapidly convergent series solution with much less computational work [1, 4].

2. Adomian decomposition method (ADM)

In the ADM [1], we first write (1) in the operator form,

$$Lu = Nu + \phi, \tag{4}$$

where $L = d^4/dx^4$ and N is the nonlinear operator that can be defined by $N = \hat{f}$, where $\hat{f}(x, u, u', u'', u''') = f(x, u, u', u'', u''') - \phi(x)$. Assume that the 4-fold operator L^{-1} exists and is easily obtained. Applying L^{-1} on both sides of (4) and using the boundary conditions yields

$$u = g + L^{-1}\phi + L^{-1}Nu, \tag{5}$$

where g represents the term arising from the given boundary conditions. The ADM [1] takes the solution u and the nonlinear function $N(u)$ as infinite series, respectively

$$u = \sum_{n=0}^{\infty} u_n, \qquad N(u) = \sum_{n=0}^{\infty} A_n(u_0, u_1, \cdots, u_n),$$

where Adomian polynomials A_n are given by [1]

$$A_n(u_0, u_1, \cdots, u_n) = \frac{1}{n!}\Big[\frac{d^n}{d\lambda^n} N(\sum_{i=0}^{n} u_i \lambda^i)\Big]_{\lambda=0}$$

Let us consider the inverse operator L^{-1} with the boundary conditions as follows

$$L^{-1}(.) = \int_a^x \int_a^x \int_a^x \int_a^x (.)\ dx\ dx\ dx\ dx. \tag{6}$$

Applying the standard ADM yields the following recursive scheme

$$u_0(x) = u(a) + u'(a)(x-a) + \tfrac{1}{2!}u''(a)(x-a)^2 + \tfrac{1}{3!}u'''(a)(x-a)^3 + L^{-1}(\phi),$$

$$u_{n+1}(x) = L^{-1}(A_n), \qquad n = 0,\ 1, \ldots.$$

Convergence aspects of the ADM have been investigated in [4]. For later numerical computation, let the expression

$$S_n(x) = \sum_{i=0}^{n-1} u_i(x), \tag{7}$$

denote the n-term approximation to $u(x)$. In order to determine all other components $u_n(x)$, $n \geq 1$, the zeroth component $u_0(x)$ has to be determined. However, $u''(a)$ and $u'''(a)$ are not defined by the boundary conditions (2) so that the zeroth component cannot be directly determined.

Many authors [3, 6, 9, 17] have proposed modified ADMs to overcome this difficulty. In [9, 17] $u''(a)$ and $u'''(a)$ are set to be constants, $u''(a) = c$ and $u'''(a) = d$, and they can be determined such that the nth partial sum $S_n(x, c)$ and $S'_n(x, c)$ satisfy the boundary conditions (2) at $x = b$ because $S_n(a, c) = u(a)$ and $S'_n(a, c) = u(a)$. In this case, it requires additional computational work to solve the nonlinear equations $S_n(b, c) = u(b)$ and $S'_n(b, c) = u'(b)$, also in [6, 10] $u''(a)$ and $u'''(a)$ are set by series

$$u''(a) = \sum_{n=0}^{\infty} c_n, \qquad u'''(a) = \sum_{n=0}^{\infty} d_n,$$

each component $u_n(x)$ can be obtained as follows

$$u_0(x) = u(a) + u'(a)(x-a) + \frac{1}{2!}c_0(x-a)^2 + \frac{1}{3!}d_0(x-a)^3 + L^{-1}(\phi), \tag{8}$$

$$u_{n+1}(x) = \frac{1}{2!}c_{n+1}(x-a)^2 + \frac{1}{3!}d_{n+1}(x-a)^3 + L^{-1}(A_n), \quad n = 0, 1, \cdots \tag{9}$$

In order to determine the unknown constants $c_n, d_n,\ n = 0, 1, \cdots$ it is also required that the nth partial sums S_n and S'_n satisfy the boundary conditions (2). It is obvious that $S_n(a) = u(a)$ and $S'_n(a) = u'(a)$, thus unknown constants c_0, d_0 can be determined by satisfying the following conditions

$$u_0(b) = u(b), \qquad u'_0(b) = u'(b). \tag{10}$$

Also each constants $c_{n+1}, d_{n+1},\ n = 1, 2, \cdots$ can be determined by solving

$$u_{n+1}(b) = 0, \qquad u'_{n+1}(b) = 0. \tag{11}$$

It also requires additional computational work.
In this work, a new modification of the ADM is proposed to overcome difficulties occurred in the standard ADM for solving fourth-order boundary value problems,

namely, the extended ADM (EADM). Main idea of the EADM is to create a canonical form containing all boundary conditions so that the zeroth component is explicitly determined without additional calculations and all other components are also easily determined.
Our aim in this work is to determine the accuracy and efficiency of the EADM in solving linear and nonlinear fourth-order boundary value problems. Numerical comparisons are made between the approximate numerical results and other methods.

3.Analysis of EADM

As described before, the standard ADM requires additional computational work in determining each component $u_n(x)$. It is easy to see that these difficulties originate from the fact that all canonical forms contain the unknown constants. Thus, our main goal is to create a new canonical form containing all boundary conditions so that each component $u_n(x)$ in the recursive scheme can be explicitly determined without additional computational work. Here we define the inverse operator L^{-1} as follows

$$L^{-1}(.) = \int_a^x \int_a^x \int_a^x \int_a^x (.)\ dx\ dx\ dx\ dx. \tag{12}$$

Applying L^{-1} on both sides of (4) and using the boundary conditions (2) yields

$$\begin{aligned} u(x) &= u(a) + u'(a)(x-a) + \frac{1}{2!}u''(a)(x-a)^2 \\ &\quad + \frac{1}{3!}u'''(a)(x-a)^3 + L^{-1}(\phi) + L^{-1}(Nu). \end{aligned} \tag{13}$$

The extended form of the Adomian decomposition method [10] will be implemented here. Our aim is now to determine the constants $u''(a)$ and $u'''(a)$. This can be achieved by imposing the remaining two boundary conditions (2) at $x = b$ to the S_n. Thus we have

$$S_n(b) = u(b), \qquad S_n'(b) = u'(b).$$

Solving for $u''(a)$ and $u'''(a)$ yields

$$\begin{aligned} u''(a) &= \tfrac{2}{(a-b)^2}\Big(u'(b)(a-b) + 2u'(a)(a-b) + 3(u(b)-u(a)) \\ &\quad + (b-a)\Big[(L^{-1}\phi)'\Big]_{x=b} + (b-a)\Big[(L^{-1}Nu)'\Big]_{x=b} \\ &\quad -3\Big[(L^{-1}\phi)\Big]_{x=b} - 3\Big[(L^{-1}Nu)\Big]_{x=b}\Big), \end{aligned}$$

$$\begin{aligned} u'''(a) &= \tfrac{6}{(a-b)^3}\Big(u'(a)(a-b) + u'(b)(a-b) + 2(u(b)-u(a)) \\ &\quad + (b-a)\Big[(L^{-1}\phi)'\Big]_{x=b} + (b-a)\Big[(L^{-1}Nu)'\Big]_{x=b} \\ &\quad -2\Big[(L^{-1}\phi)\Big]_{x=b} - 3\Big[(L^{-1}Nu)\Big]_{x=b}\Big). \end{aligned}$$

By substituting $u''(a)$ and $u'''(a)$ into (13), we have

$$\begin{aligned}
u(x) &= u(a)+u'(a)(x-a)+q(x)^2\Big(u'(b)(a-b)+2u'(a)(a-b)\\
&+3(u(b)-u(a))+(b-a)\Big[(L^{-1}\phi)'\Big]_{x=b}+(b-a)\Big[(L^{-1}Nu)'\Big]_{x=b}\\
&-3\Big[(L^{-1}\phi)\Big]_{x=b}-3\Big[(L^{-1}Nu)\Big]_{x=b}\Big)-q(x)^3\Big(u'(a)(a-b)\\
&+u'(b)(a-b)+2(u(b)-u(a))+(b-a)\Big[(L^{-1}\phi)'\Big]_{x=b}\\
&+(b-a)\Big[(L^{-1}Nu)'\Big]_{x=b}-2\Big[(L^{-1}\phi)\Big]_{x=b}-2\Big[(L^{-1}Nu)\Big]_{x=b}\Big)\\
&+L^{-1}(\phi)+L^{-1}(Nu),
\end{aligned} \tag{14}$$

where $q(x)=(x-a)/(b-a)$.
The components $u_n(x),\ \ n=0,\ 1,\cdots$ can be elegantly determined by using the recursive relation

$$\begin{aligned}
u_0(x) &= u(a)+u'(a)(x-a)+q(x)^2\Big(u'(b)(a-b)+2u'(a)(a-b)\\
&+3(u(b)-u(a))+(b-a)\Big[(L^{-1}\phi)'\Big]_{x=b}-3\Big[(L^{-1}\phi)\Big]_{x=b}\Big)\\
&-q(x)^3\Big(u'(a)(a-b)+u'(b)(a-b)+2(u(b)-u(a))\\
&+(b-a)\Big[(L^{-1}\phi)'\Big]_{x=b}-2\Big[(L^{-1}\phi)\Big]_{x=b}\Big)+L^{-1}(\phi),
\end{aligned} \tag{15}$$

$$\begin{aligned}
u_{n+1}(x) &= q(x)^2\Big((b-a)\Big[(L^{-1}A_n)'\Big]_{x=b}-3\Big[(L^{-1}A_n)\Big]_{x=b}\Big)\\
&-q(x)^3\Big((b-a)\Big[(L^{-1}A_n)'\Big]_{x=b}-2\Big[(L^{-1}A_n)\Big]_{x=b}\Big)+L^{-1}A_n,
\end{aligned} \tag{16}$$

where A_n is the Adomian polynomials associated with the nonlinear operator N. It is worth noting that the canonical form (14) consists of all boundary conditions. Moreover, the nth partial sum S_n and S'_n from the recursive schemes, (15) and (16), always satisfy the boundary conditions for any n. Thus, it is not necessary to determine the unknown constant $u''(a)$ and $u'''(a)$ by extra calculations.

Remark 1. *We note that each component $u_n(x),\ n=0,\ 1,\cdots$ in (15), (16) is identical to the components $u_n(x),\ n=0,\ 1,\cdots$ in (8) and (9). Because applying (10) and (11) to $u_n,\ \ n=0,\ 1,\cdots$ in (8) and (9) yields*

$$\begin{aligned}
c_0 &= \frac{2}{(a-b)^2}\Big(2u'(a)(a-b)+u'(b)(a-b)+3(u(b)-u(a))\\
&+(b-a)\Big[(L^{-1}\phi)'\Big]_{x=b}-3\Big[(L^{-1}\phi)\Big]_{x=b}\Big),
\end{aligned}$$

$$\begin{aligned}
d_0 &= \frac{6}{(a-b)^3}\Big(u'(a)(a-b)+u'(b)(a-b)+2(u(b)-u(a)) \\
&\quad +(b-a)\Big[(L^{-1}\phi)'\Big]_{x=b}-2\Big[(L^{-1}\phi)\Big]_{x=b}\Big), \\
c_{n+1} &= \frac{-2}{(a-b)^2}\Big((a-b)\Big[(L^{-1}A_n)'\Big]_{x=b}+3\Big[(L^{-1}A_n)\Big]_{x=b}\Big), \\
d_{n+1} &= \frac{-6}{(a-b)^3}\Big((a-b)\Big[(L^{-1}A_n)'\Big]_{x=b}+2\Big[(L^{-1}A_n)\Big]_{x=b}\Big).
\end{aligned}$$

As discussed in [10] there are several types of fourfold definite integrals, it is possible to produce different components for each twofold definite integral. We prove that EADM is independent on the inverse operator which is defined by any fourfold definite integral. For this we consider the inverse operator L_k^{-1} defined by

$$L_k^{-1}=\int_{\nu_k}^{x}\int_{\omega_k}^{x}\int_{\zeta_k}^{x}\int_{\eta_k}^{x} dx\ dx\ dx\ dx, \qquad k=1,2,\cdots,16, \tag{17}$$

where $\nu_k,\ \omega_k,\ \zeta_k,\ \eta_k$ are introduced in table 1. Let us define u_n^k by the component induced by L_k^{-1} in EADM. Applying the procedures in EADM with the inverse operator L_1^{-1} yields

$$u_0^1(x)=u_0(x), \qquad u_{n+1}^1(x)=u_{n+1}(x),$$

where $u_0(x)$ and $u_{n+1}(x)$ obtained in (15) and (16).

Table 1: Parameter values used for illustration purposes

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ν_k	a	a	a	a	a	a	a	a	b	b	b	b	b	b	b	b
ω_k	a	a	a	a	b	b	b	b	a	a	a	a	b	b	b	b
ζ_k	a	a	b	b	a	a	b	b	a	a	b	b	a	a	b	b
η_k	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b

Lemma 1. $L_1^{-1}\psi-q(x)^2\Big((a-b)\Big[(L_1^{-1}\psi)'\Big]_{x=b}+3\Big[(L_1^{-1}\psi)\Big]_{x=b}\Big)$ $+\,q(x)^3\Big((a-b)\Big[(L_1^{-1}\psi)'\Big]_{x=b}+2\Big[(L_1^{-1}\psi)\Big]_{x=b}\Big)=L_2^{-1}\psi-q(x)^2$ $\Big((a-b)\Big[(L_2^{-1}\psi)'\Big]_{x=b}+3\Big[(L_2^{-1}\psi)\Big]_{x=b}\Big)+q(x)^3\Big((a-b)\Big[(L_2^{-1}\psi)'\Big]_{x=b}$ $+\,2\Big[(L_2^{-1}\psi)\Big]_{x=b}\Big)$

Proof. Let us define ψ_1, ψ_2, ψ_3, and ψ_4 by $(\psi_1)' = \psi, (\psi_2)' = \psi_1, (\psi_3)' = \psi_2$ and $(\psi_4)' = \psi_3$. Then we have

$$L_1^{-1}\psi = \psi_4(x) - \psi_4(a) - (x-a)\psi_3(a) - \frac{(x-a)^2}{2!}\psi_2(a) - \frac{(x-a)^3}{3!}\psi_1(a),$$
$$L_2^{-1}\psi = \psi_4(x) - \psi_4(a) - (x-a)\psi_3(a) - \frac{(x-a)^2}{2!}\psi_2(a) - \frac{(x-a)^3}{3!}\psi_1(b).$$

Therefore, we have:

$$\begin{aligned} L_1^{-1}\psi \quad & -q(x)^2\Big((a-b)\Big[(L_1^{-1}\psi)'\Big]_{x=b} + 3\Big[(L_1^{-1}\psi)\Big]_{x=b}\Big) \\ & +q(x)^3\Big((a-b)\Big[(L_1^{-1}\psi)'\Big]_{x=b} + 2\Big[(L_1^{-1}\psi)\Big]_{x=b}\Big) \\ & = \frac{-1}{(a-b)^3}\Big(2\psi_4(b)x^3 - 2\psi_4(a)x^3 + a\psi_3(a)x^3 - b\psi_3(b)x^3 - \\ & b\psi_3(a)x^3 + a\psi_3(b)x^3 - 2a^2\psi_3(b)x^2 + b^2\psi_3(b)x^2 + a\psi_3(b)bx^2 \\ & +3\psi_4(a)ax^2 + 3\psi_4(a)bx^2 - a^2\psi_3(a)x^2 - a\psi_3(a)bx^2 - 3\psi_4(b)bx^2 \\ & +2b^2\psi_3(a)x^2 - 3\psi_4(b)ax^2 + 6a\psi_4(b)bx - 6a\psi_4(a)bx - ab^2\psi_3(a)x \\ & -2ab^2\psi_3(b)x + 2a^2b\psi_3(a)x + a^2b\psi_3(b)x + a^3\psi_3(b)x - \psi_3(a)b^3x\psi_4(x)a^3 \\ & +a^3\psi_4(b) - a^2b^2\psi_3(a) + a^2b^2\psi_3(b) - 3a^2\psi_4(b)b + \psi_4(x)b^3 - \psi_4(a)b^3 \\ & +\psi_3(a)ab^3 + 3\psi_4(x)a^2b - 3\phi_4(x)ab^2 + 3\psi_4(a)ab^2 - a^3b\psi_3(b)\Big). \end{aligned} \tag{18}$$

In the same manner, we have the same result for

$$L_2^{-1}\psi - q(x)^2\Big((a-b)\Big[(L_2^{-1}\psi)'\Big]_{x=b} + 3\Big[(L_2^{-1}\psi)\Big]_{x=b}\Big) + q(x)^3\Big((a-b)\Big[(L_2^{-1}\psi)'\Big]_{x=b} + 2\Big[(L_2^{-1}\psi)\Big]_{x=b}\Big).$$

Corollary 1. *From Lemma 1, it is easy to see that* $u_n^1(x) = u_n^2(x), \quad n = 0,\ 1, \cdots.$

Remark 2. *Now let us consider each component* $u_n^3(x), \quad n = 0,\ 1, \cdots$ *by taking* L_3^{-1} *as the inverse operator. Each component* $u_n^3(x), \quad n = 0,\ 1, \cdots$ *can be easily obtained by the same procedures in EADM as follows:*

$$\begin{aligned} u_0^3(x) \quad = \quad & u(a) + u'(a)(x-a) - q(x)^2\Big(2u'(b)(a-b) + u'(a)(a-b) \\ & +3(u(b)-u(a)) + 2(b-a)\Big[(L_3^{-1}\phi)'\Big]_{x=b} - 3\Big[(L_3^{-1}\phi)\Big]_{x=b}\Big) \\ & -r(x)^3\Big(u'(a)(a-b) + u'(b)(a-b) + 2(u(b)-u(a)) + \\ & (b-a)\Big[(L_3^{-1}\phi)'\Big]_{x=b} - 2\Big[(L_3^{-1}\phi)\Big]_{x=b}\Big) - \Big(u'(a)(a-b) + \end{aligned} \tag{19}$$

$$u'(b)(a-b)+2(u(b)-u(a))+(b-a)\Big[(L_4^{-1}\phi)'\Big]_{x=b}-2\Big[(L_3^{-1}\phi)\Big]_{x=b}\Big)$$
$$+3q(x)\Big(u'(a)(a-b)+u'(b)(a-b)+2(u(b)-u(a))+$$
$$(b-a)\Big[(L_3^{-1}\phi)'\Big]_{x=b}-2\Big[(L_3^{-1}\phi)\Big]_{x=b}\Big)+L_3^{-1}\phi,$$

$$\begin{aligned} u^3_{n+1}(x) \quad = \quad & -q(x)^2\Big(2(b-a)\Big[(L_3^{-1}A_n)'\Big]_{x=b}-3\Big[(L_3^{-1}A_n)\Big]_{x=b}\Big) \\ & -q(x)^3\Big((b-a)\Big[(L_3^{-1}A_n)'\Big]_{x=b}-2\Big[(L_3^{-1}A_n)\Big]_{x=b}\Big) \\ & -\Big((b-a)\Big[(L_3^{-1}A_n)'\Big]_{x=b}-2\Big[(L_3^{-1}A_n)\Big]_{x=b}\Big) \\ & +3q(x)\Big((b-a)\Big[(L_3^{-1}A_n)'\Big]_{x=b}-2\Big[(L_3^{-1}A_n)\Big]_{x=b}\Big) \\ & +L_3^{-1}A_n, \qquad n=0,\ 1,\cdots, \end{aligned} \tag{20}$$

where $r(x)=(x-b)/(b-a)$.

Lemma 2. $u^1_n(x)=u^3_n(x), \quad n=0,\ 1,\cdots.$

Proof.

$$1-r(x)=1-\frac{x-b}{b-a}=\frac{x-a}{b-a}=q(x).$$

This implies that

$$\begin{aligned} & u(a)+u'(a)(x-a)-q(x)^2\Big(2u'(b)(a-b)+u'(a)(a-b)+3(u(b)- \\ & u(a))\Big)-r(x)^3\Big(u'(a)(a-b)+u'(b)(a-b)+2(u(b)-u(a))\Big)- \\ & \Big(u'(a)(a-b)+u'(b)(a-b)+2(u(b)-u(a))\Big)+3q(x)\Big(u'(a)(a-b) \\ & +u'(b)(a-b)+2(u(b)-u(a))\Big)=u(a)+u'(a)(x-a)- \\ & q(x)^2\Big(2u'(a)(b-a)+u'(b)(b-a)+3(u(a)-u(b))\Big)+ \\ & q(x)^3\Big(u'(a)(b-a)+u'(b)(b-a)+2(u(a)-u(b))\Big). \end{aligned}$$

Thus, it is sufficient to show that

$$\begin{aligned} L_1^{-1}\psi \qquad & -q(x)^2\Big((a-b)\Big[(L_1^{-1}\psi)'\Big]_{x=b}+3\Big[(L_1^{-1}\psi)\Big]_{x=b}\Big) \\ & +q(x)^3\Big((a-b)\Big[(L_1^{-1}\psi)'\Big]_{x=b}+2\Big[(L_1^{-1}\psi)\Big]_{x=b}\Big) \\ & =L_3^{-1}\psi-q(x)^2\Big(2(b-a)\Big[(L_3^{-1}\psi)'\Big]_{x=b}-3\Big[(L_3^{-1}\psi)\Big]_{x=b}\Big) \\ & -r(x)^3\Big((b-a)\Big[(L_3^{-1}\psi)'\Big]_{x=b}-2\Big[(L_3^{-1}\psi)\Big]_{x=b}\Big)-\Big((b-a)\Big[(L_3^{-1}\psi)'\Big]_{x=b} \\ & -2\Big[(L_3^{-1}\psi)\Big]_{x=b}\Big)+3q(x)\Big((b-a)\Big[(L_3^{-1}\psi)'\Big]_{x=b}-2\Big[(L_3^{-1}\psi)\Big]_{x=b}\Big). \end{aligned} \tag{21}$$

With the assumption of lemma 1 we have

$$L_3^{-1}\psi = \psi_4(x) - \psi_4(a) - (x-a)\psi_3(a) - \frac{(x-a)^2}{2!}\psi_2(b) - \frac{(x-b)^3}{3!}\psi_1(a)$$
$$+\frac{(a-b)^3}{3!}\psi_1(a) + \frac{(a-b)^2}{2!}(x-a)\psi_1(a).$$

By substituting $L_3^{-1}\psi$ in the second side of (21) we get the same result in (18). It completes the proof.

Since the boundary conditions (2) are symmetric in terms of a and b, therefore by exchanging a and b, we get

$$u_n^{(k)}(x) = u_n^{(16-k+1)}(x), \quad k = 1, 2, \cdots, 8.$$

So for proving the equality of $u_n^{(k)}(x)$'s for $k = 1, 2, \cdots, 16$ it suffices to prove the equality of $u_n^{(k)}(x)$'s for $k = 1, 2, \cdots, 8$. For this, we have olready shown the equality of $u_n^{(1)}(x)$, $u_n^{(2)}(x)$ and $u_n^{(3)}(x)$ in lemmas 1 and 2, and the rest of the equalities can be proved in a similar way. Thus, the following conclusion can be obtained.

Theorem 1. *Every component* $u_n^k(x), \quad n = 0,\ 1, \cdots$ *induced by* $L_k^{-1}, \quad k = 1, \ldots, 16$ *in EADM is identical. In other words, EADM is independent on the inverse operator which is defined by any fourfold definite integral.*

We note that all of these lemmas and theorem are true with the boundary condition (3)

4. Examples

In this section, we demonstrate the effectiveness of the EADM with several illustrative examples. All numerical results obtained by EADM are compared with the results obtained by various numerical methods.

Example 1. Consider the following linear problem [13],

$$u^{(4)}(x) = (1+c)u''(x) - cu(x) + \frac{1}{2}cx^2 - 1, \qquad 0 < x < 1,$$

subject to

$$u(0) = 1, \qquad u'(0) = 1,$$
$$u(1) = 1.5 + sinh(1), \qquad u'(1) = 1.5 + cosh(1).$$

The exact solution for this problem is $u(x) = 1 + \frac{1}{2x^2} + sinh(x)$. For each test point, the absolute error between the exact solution and the results obtained by the HPM, ADM [13] and the EADM is compared in Table 2 and 3 for $c = 1$ and $c = 10$. Our approximate solutions obtained using EADM are in good agreement with the exact solution when the parameter c is less than 10. With only two iterations a better approximation S_2 has been obtained than the results by HPM and ADM.

Table 2: Absolute errors of the first-order approximate solution when $c = 1$.

k	*Analytical solution*	E_{HPM}	E_{ADM}	E_{EADM}
0.0	1.0000000000	0.0000	0.0000	0.0000
0.1	1.1051667500	$7.4E-5$	$7.4E-5$	$1.1488E-6$
0.2	1.2213360025	$2.5E-4$	$2.5E-4$	$3.2027E-7$
0.3	1.3495202934	$4.6E-4$	$4.6E-4$	$1.1328E-5$
0.4	1.4907523258	$6.5E-4$	$6.5E-4$	$3.4636E-5$
0.5	1.6460953055	$7.6E-4$	$7.6E-4$	$6.6411E-5$
0.6	1.8166535821	$7.5E-4$	$7.5E-4$	$9.6330E-5$
0.7	2.0035837018	$6.1E-4$	$6.1E-4$	$1.1038E-4$
0.8	2.2081059822	$3.8E-4$	$3.8E-4$	$9.6471E-5$
0.9	2.4315167257	$1.3E-4$	$1.3E-4$	$5.2931E-5$
1.0	2.6752011936	0.0000	0.0000	0.0000

Example 2. Consider the following linear fourth-order integro-differential equation [15],

$$y^{(4)}(x) = x(1+e^x) + 3e^x + y(x) - \int_0^x y(x)dx, \qquad 0 < x < 1,$$

subject to the boundary conditions:

$$y(0) = 1, \qquad y''(0) = 2, \quad y(1) = 1 + e, \quad y''(1) = 3e.$$

The exact solution is $y(x) = 1 + xe^x$.

The numerical results of EADM compared with exact solution and VIM [14] are presented in Table 4. These results are evaluated at $n = 2$ term of the recurrence relation (7). It can be seen from the numerical results in Table 4 that the EADM is more accurate than the VIM solution in [14]. Even though the ADM [8] shows a better performance than EADM, it is easy to obtain a similar accurate approximation with a few more iterations in EADM.

Table 3: Absolute errors of the first-order approximate solution obtained by HPM, ADM and EADM for Example 1 when $c = 10$.

k	*Analytical solution*	E_{HPM}	E_{ADM}	E_{EADM}
0.0	1.0000000000	0.0000	0.0000	0.0000
0.1	1.1051667500	$1.7E-4$	$1.7E-4$	$4.50410E-6$
0.2	1.2213360025	$5.7E-4$	$5.7E-4$	$3.02581E-5$
0.3	1.3495202934	$1.0E-3$	$1.0E-3$	$8.72832E-5$
0.4	1.4907523258	$1.4E-3$	$1.4E-3$	$1.67419E-4$
0.5	1.6460953055	$1.6E-3$	$1.6E-3$	$2.44493E-4$
0.6	1.8166535821	$1.6E-3$	$1.6E-3$	$2.83793E-4$
0.7	2.0035837018	$1.2E-3$	$1.2E-3$	$2.58064E-4$
0.8	2.2081059822	$7.6E-3$	$7.6E-3$	$1.66169E-4$
0.9	2.4315167257	$2.5E-3$	$2.5E-3$	$4.94701E-5$
1.0	2.6752011936	0.0000	0.0000	0.0000

Example 3. Consider the following nonlinear fourth-order integro-differential equation [15],

$$y^{(4)}(x) = 1 + \int_0^x e^{-x} y^2(x)dx, \qquad 0 < x < 1.$$

subject to the boundary conditions:

$$y(0) = 1, \qquad y''(0) = 1, \quad y(1) = e, \quad y''(1) = e.$$

The exact solution is $y(x) = e^x$. Table 5 shows the accuracy of the approximate solution $y_1(x)$, where $E(x) = |e^x - y_1(x)|$ be the absolute error. Also, it is evident that the overall errors can be made smaller by adding new terms from the iterative formulas.

Example 4. Consider the following nonlinear fourth-order integro-differential equation [14],

$$u^{(4)}(x) = u^2(x) + 1, \quad 0 < x < 2,$$

subject to the boundary conditions $u(0) = u'(0) = u(2) = u'(2) = 0$.

Table 6 displays the comparison of the EADM with the trapezoidal rule solution [5] for some values of x. The obtained solution is of remarkable accuracy, as shown in Table 6.

Table 4: Absolute errors of the first-order approximate solution obtained by VIM and EADM for Example 2.

k	*Exact solution*	E_{VIM}	E_{EADM}
0.0	1.0	0.00	0.00
0.2	1.24428	$1.65737E-4$	$5.816195E-5$
0.4	1.59673	$2.84804E-4$	$9.206495E-5$
0.6	2.09327	$3.12185E-4$	$8.912111E-5$
0.8	2.78043	$2.13844E-4$	$5.314847E-5$
1.0	3.71828	0.00	0.00

Table 5: Absolute errors of the first-order approximate solution obtained by VIM and EADM for Example 3.

k	*Exact solution*	E_{VIM}	E_{EADM}
0.0	1.0	0.00	0.00
0.2	1.2214	$1.11471E-03$	$4.01367E-05$
0.4	1.49182	$1.88476E-03$	$6.73710E-05$
0.6	1.82212	$2.00882E-03$	$7.08922E-05$
0.8	2.22554	$1.32855E-03$	$4.61244E-05$
1.0	2.71828	$5.27578E-13$	$2.00000E-15$

5. Conclusion

In this work, the EADM is extended to solve the two point fourth order boundary value problems. ADM has been successful for solving many application problems with simple calculations. However, it has difficulties in dealing with boundary conditions for solving two-point boundary problems. Many approaches have been presented to overcome these difficulties. However, they require additional computational work since all boundary conditions are not included in the canonical form. Our fundamental goal is to create the canonical form containing all boundary conditions. The EADM does not require us to calculate the unknown constant which is usually a derivative at the boundary. All numerical approximations by EADM are compared with the results in many other methods such as homotopy perturbation method, variational iteration method and the trapezoidal rule (TRAP). From the results in illustrative examples, it is concluded that EADM is a very effective algorithm which provides promising results with simple calculations.

Table 6: Comparison of the first-order approximate solution obtained by VIM with numerical solution obtained by the trapezoidal rule (TRAP) [5].

k	VIM	$EADM$	$TRAP[1]$
0.0	0.000000000	0.00000000	0.00000000
0.1	0.001514138	0.001505652	0.00150566
0.2	0.005437536	0.005405472	0.00540548
0.3	0.010916675	0.108487561	0.01084878
0.4	0.017198041	0.017084801	0.01708483
0.5	0.023628128	0.023462922	0.02346295
0.6	0.029653459	0.029432459	0.02943250
0.7	0.034820601	0.034542813	0.03454287
0.8	0.038776177	0.038443468	0.03844355
0.9	0.041266870	0.040884029	0.04088413
1.0	0.042139406	0.041714244	0.04171435

References

[1] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method*, Kluwer Academic, Dordrecht, 1994.

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

[3] G. Adomian and R. Rach,*Modified decomposition solution of linear and nonlinear boundary-value problems*, Nonlinear Anal., 23, 5, (1994), 615-619.

[4] K. Abbaoui and Y. Cherruault,*New ideas for proving convergence of decomposition methods*, Comput. Math. Appl., 29, (1995), 103-108.

[5] Waleed Al-Hayani and Luis Casasu s,*Approximate analytical solution of fourth order boundary value problems*, Numer Algorithms, 40, (2005), 67-78.

[6] M. Benabidallah and Y. Cherruault,*Application of the Adomain method for solving a class of boundary problems*, Kybernetes, 33, 1, (2004), 118-132.

[7] E. Doedel,*Finite difference methods for nonlinear two-point boundary-value problem*, SIAM Journal of Numerical Analysis, 16, (1979), 173-185.

[8] I. Hashim,*Adomian decomposition method for solving BVPs for fourth-order integro-differential equations*, J. Comput. Appl. Math. , 193, (2006), 658-664.

[9] M. Inc and D. J. Evans,*The decomposition method for solving of a class of singular two-point boundary value problems*, Int. J. Comput. Math., 80, 7, (2003), 869-882.

[10] B. Jang,*Two-point boundary value problems by the extended Adomian decomposition method*, J. Comput. Appl. Math., 219, (2008), 253-262.

[11] S.T. Mohyud-Din and M.A. Noor,*Homotopy perturbation method for solving fourth order boundary value problems*, Math. Problems Eng., (2007), 1-15, article ID 98602.

[12] S. Momani and K. Moadi,*A reliable algorithm for solving fourth-order boundary value problems*, J. Appl. Math. Comput., 22, 3, (2006), 185-197.

[13] S. Momani and M. A. Noor,*Numerical comparison of methods for solving a special fourth-order boundary value problem*, Appl. Math. Comput., 191, (2007), 218-224.

[14] Lan. Xu,*The variational iteration method for fourth order boundary value problems*, Chaos, Solitons and Fractals, 39, 3, (2009), 1386-1394.

[15] N. H. Sweilam,*Fourth order integro-differential equations using variational iteration method*, Int. J. Comput. Math., 54, (2007), 1086-1091.

[16] A. M. Wazwaz,*The numerical solution of special fourth-order boundary value problems by the modified decomposition method*, Int. J. Comput. Math. 79, 3, (2002), 345-356.

[17] A. M. Wazwaz,*A reliable algorithm for obtaining positive solutions for nonlinear boundary value problems*, Comput. Math. Appl., 41, (2001), 1237-1244.

G. Ebadi and S. Rashedi
Department of Applied mathematics
Faculty of mathematical Sciences
University of Tabriz
Tabriz, Iran.
email: *ghodrat_ebadi@yahoo.com* or *gebadi@tabrizu.ac.ir* (G. Ebadi)
email: *s.rashedi.t@gmail.com* (S. Rashedi)