

Treatment for third-order nonlinear differential equations based on the Adomian decomposition method

Xueqin Lv and Jianfang Gao

ABSTRACT

The Adomian decomposition method (ADM) is an efficient method for solving linear and nonlinear ordinary differential equations, differential algebraic equations, partial differential equations, stochastic differential equations, and integral equations. Based on the ADM, a new analytical and numerical treatment is introduced in this research for third-order boundary-value problems. The effectiveness of the proposed approach is verified by numerical examples.

1. Introduction

In this paper, we apply the Adomian decomposition method (ADM) to the third-order nonlinear differential equation

$$\begin{cases} u^{(3)}(x) + h(x)f(u(x)) = g(x), & x \in [0, 1], \\ u(0) = \alpha, \quad u(1) = \beta, \quad u'(0) = \gamma, \end{cases} \quad (1.1)$$

where $h(x), f(u(x))$ are continuous functions.

The third-order nonlinear differential equation arises in different areas of applied mathematics and physics. Underground water flow and population dynamics can be reduced to nonlocal problems with boundary conditions [14, 15, 27]. Moreover, boundary-value problems (BVPs) with boundary conditions constitute a very interesting and important class of problems. They include two-, three- and multipoint BVPs as special cases. Therefore, the problems have attracted much attention and have been studied by many authors. There are many algorithms to solve the third-order nonlinear differential equation, for example: the direct block method [25], the Leray–Schauder continuation principle [19], the Green’s function method [17], the fixed-point theorem in cones [22] and the reproducing kernel method [16, 23].

The Adomian decomposition method and its modification (MADM) [1–6, 8–12, 18, 21, 26, 28–32] have been used to solve effectively and easily a large class of linear and nonlinear ordinary and partial differential equations. In this paper, the ADM is improved to deal with third-order nonlinear differential equations. This improvement is based on the ADM and a modification of Lesnic’s work [20].

2. Improvement of Ebaid’s work

2.1. Ebaid’s work

The differential operator L_x is given by

$$L_x(\cdot) = \frac{d^N(\cdot)}{dx^N}. \quad (2.1)$$

Received 11 June 2015; revised 8 June 2016.

2010 Mathematics Subject Classification 34K10 (primary).

The corresponding author for this article is Jianfang Gao.

From the formula (2.1), we can get:

- when $N = 1$, the inverse operator L_x^{-1} is

$$L_x^{-1}(\cdot) = \int_{x_0}^x (\cdot) dk; \tag{2.2}$$

- when $N = 2$, the inverse operator L_{xx}^{-1} is

$$L_{xx}^{-1}(\cdot) = \int_{x_0}^x dk \int_{x_0}^k (\cdot) ds; \tag{2.3}$$

- when $N = 3$, the inverse operator L_{xxx}^{-1} is

$$L_{xxx}^{-1}(\cdot) = \int_{x_0}^x dk \int_{x_0}^k ds \int_{x_0}^s (\cdot) dy. \tag{2.4}$$

In [21], Lesnic proposed the operator

$$L_{xx}^{-1}(\cdot) = \int_{x_0}^x dk \int_{x_0}^k (\cdot) ds - \frac{x - x_0}{1 - x_0} \int_{x_0}^1 dk \int_{x_0}^k (\cdot) ds \tag{2.5}$$

to solve the Dirichlet problem for the heat equation $u_t = u_{xx}, x_0 < x < 1, t > 0$, under the boundary conditions

$$u(x_0, t) = f_0(t), \quad u(1, t) = f_1(t) \tag{2.6}$$

and the initial condition

$$u(x, 0) = p(x). \tag{2.7}$$

Using the definition in (2.5), we note that

$$L_{xx}^{-1}(u''(x)) = u(x, t) - u(x_0, t) - \frac{x - x_0}{1 - x_0} [u(1, t) - u(x_0, t)], \tag{2.8}$$

that is, the boundary conditions can be used directly. However, from (2.5) we note that the lower bound of all integrations is restricted to the initial point x_0 . In fact, we can avoid this restriction by using a new definition of L_{xx}^{-1} which gives the same result as in equation (2.8) and is given by

$$L_{xx}^{-1}(\cdot) = \int_{x_0}^x dk \int_c^k (\cdot) ds - \frac{x - x_0}{1 - x_0} \int_{x_0}^1 dk \int_c^k (\cdot) ds, \tag{2.9}$$

where c is a free lower point. This free point plays an important role if the equation solved has a singular point.

2.2. Extension of Ebaid's work

In the extension of Ebaid's work, we have

$$L_{xxx}^{-1}(\cdot) = \int_a^x dk \int_c^k ds \int_g^s (\cdot) dy - x^2 \int_d^b dk \int_e^k ds \int_h^s (\cdot) dy. \tag{2.10}$$

In the next section, we will give a theoretical derivation of the operator given by equation (2.10).

3. Derivation of the proposed operator

First, we define L_{xxx}^{-1} as

$$L_{xxx}^{-1}(\cdot) = \int_a^x dk \int_c^k ds \int_g^s (\cdot) dy - z(x) \int_d^b dk \int_e^k ds \int_h^s (\cdot) dy, \tag{3.1}$$

where $z(x)$ is to be determined such that $L_{xxx}^{-1}(u(x))$ can be expressed only in terms of the boundary conditions given in (1.1). With this definition, we easily get

$$\begin{aligned} L_{xxx}^{-1}(u^{(3)}(x)) &= u(x) - u(a) - xu'(c) + au'(c) - xcu''(g) + acu''(g) + \frac{a^2}{2}u''(g) - \frac{x^2}{2}u''(g) \\ &\quad - z(x) \left[u(b) - u(d) - bu'(e) + du'(e) + beu''(h) \right. \\ &\quad \left. - de u''(h) + \frac{d^2}{2}u''(h) - \frac{b^2}{2}u''(h) \right] \\ &= u(x) - u(a) - xu'(c) + au'(c) - z(x)[u(b) - u(d) - bu'(e) + du'(e)] \\ &\quad + u''(g) \left[c(x - a) - \frac{x^2 - a^2}{2} \right] - z(x)u''(h) \left[e(b - d) - \frac{b^2 - d^2}{2} \right]. \end{aligned} \tag{3.2}$$

Setting $d = a, e = c$ and $h = g$, we obtain

$$\begin{aligned} L_{xxx}^{-1}(u^{(3)}(x)) &= u(x) - u(a) - xu'(c) + au'(c) + z(x)[u(a) - u(b) - au'(c) + bu'(c)] \\ &\quad + u''(g) \left\{ c(x - a) - \frac{x^2 - a^2}{2} + z(x) \left[c(a - b) + \frac{b^2 - a^2}{2} \right] \right\}. \end{aligned} \tag{3.3}$$

In order to express $L_{xxx}^{-1}(u^{(3)}(x))$ in terms of the three boundary conditions only, we have to eliminate the coefficients multiplying $u''(g)$ by setting

$$u''(g) \left\{ c(x - a) - \frac{x^2 - a^2}{2} + z(x) \left[c(a - b) + \frac{b^2 - a^2}{2} \right] \right\} = 0. \tag{3.4}$$

Let $a = 0, b = 1, c = 0$ and assume that $u''(g) \neq 0$; we get

$$\begin{aligned} z(x) &= \frac{2cx - 2ca - x^2 + a^2}{2cb - 2ca - b^2 - a^2} \\ &= x^2. \end{aligned} \tag{3.5}$$

Substituting (3.4) and (3.5) into (3.3), respectively, we obtain

$$\begin{aligned} L_{xxx}^{-1}(u^{(3)}(x)) &= u(x) - u(0) - xu'(0) + x^2[u(0) - u(1) + u'(0)] \\ &= u(x) - \alpha - x\gamma + x^2(\alpha - \beta + \gamma). \end{aligned} \tag{3.6}$$

In the next section, we will use the combination of this equation and the decomposition method to solve the third-order nonlinear differential equation.

4. Analysis of the improved ADM (IADM) for solving third-order nonlinear differential equations

In this section, the ADM with the modification of Lesnic’s work developed in the previous section is used to construct algorithms for solving equation (1.1).

Now, applying the operator $L_{xxx}^{-1}(\cdot)$ presented in the previous section and given by equation (3.6) on both sides of equation (1.1), we obtain

$$u(x) = \alpha + x\gamma - x^2(\alpha - \beta + \gamma) - L_{xxx}^{-1}h(x)f(u(x)) + L_{xxx}^{-1}g(x). \quad (4.1)$$

The ADM is based on decomposing u and the nonlinear term $f(u(x))$ as

$$u = \sum_{n=0}^{\infty} u_n \quad \text{and} \quad f(u(x)) = \sum_{n=0}^{\infty} A_n, \quad (4.2)$$

where $\{A_n\}_{n=0}^{\infty}$ are Adomian's polynomials for the nonlinear term $f(u(x))$ and can be found from the formula

$$A_n = \frac{1}{n!} \left[\frac{d^n}{d\lambda^n} f \left(\sum_{i=0}^{\infty} \lambda^i u_i \right) \right]_{\lambda=0}, \quad n \geq 0. \quad (4.3)$$

Substituting (4.2) into (4.1) and according to the ADM, the solution $u(x)$ can be elegantly computed by using the recurrence relation

$$\begin{cases} u_0(x) = \alpha + x\gamma - x^2(\alpha - \beta + \gamma) + L_{xxx}^{-1}g(x), \\ u_1(x) = L_{xxx}^{-1}(h(x)A_0(x)), \\ u_2(x) = L_{xxx}^{-1}(h(x)A_1(x)), \\ u_3(x) = L_{xxx}^{-1}(h(x)A_2(x)), \\ \vdots \\ u_n(x) = L_{xxx}^{-1}(h(x)A_{n-1}(x)), \quad n \geq 0 \end{cases} \quad (4.4)$$

and

$$\begin{cases} A_0 = f(u_0), \\ A_1 = u_1 f'(u_0), \\ A_2 = u_2 f'(u_0) + \frac{1}{2} u_1^2 f''(u_0), \\ A_3 = u_3 f'(u_0) + u_1 u_2 f''(u_0) + \frac{1}{3!} u_1^3 f'''(u_0), \\ \vdots \\ A_n = \frac{1}{n!} \left[\frac{d^n}{d\lambda^n} f \left(\sum_{i=0}^{\infty} \lambda^i u_i \right) \right]_{\lambda=0}, \quad n \geq 0. \end{cases} \quad (4.5)$$

We obtain an approximate solution of the equation

$$\tilde{u}_m(x) = \sum_{n=0}^m u_n(x). \quad (4.6)$$

In this work, $u(x)$ is assumed to be bounded for any $x \in J = [0, 1]$. The nonlinear term $f(u)$ is Lipschitz continuous with $|f(u) - f(v)| \leq L|u - v|$. In the paper [13], El-Kalla deduced another programmable formula for the Adomian polynomials:

$$A_n = f(S_n) - \sum_{i=0}^{n-1} A_i, \tag{4.7}$$

where the partial sum is $S_n = \sum_{i=0}^n u_i(x)$. The relations above have been obtained with the assumption of the convergence of the series (4.2). The conditions for such convergence are discussed in the following theorem.

THEOREM 4.1. *Suppose $0 < \alpha = MNL < 1$ and $|u_1(x)| < \infty$; then the series solution (4.2) of problem (4.1) using ADM converges, where $N = \max_{\forall x \in J} |h(x)|$ and, for any $u(x) \in C[J]$, $\exists M > 0$ such that $\|L_{xxx}^{-1}u\| \leq M\|u\|$.*

Proof. Denote $(C[J], \|\cdot\|)$ as the Banach space of all continuous functions on J with the norm $\|u(x)\| = \max_{\forall x \in J} |u(x)|$. Define S_n as the sequence of partial sums. Let S_n and S_m be arbitrary partial sums with $n \geq m$. We are going to prove that S_n is a Cauchy sequence in this Banach space.

$$\begin{aligned} \|S_n - S_m\| &= \max_{\forall x \in J} |S_n - S_m| \\ &= \max_{\forall x \in J} \left| \sum_{i=m+1}^n u_i(x) \right| \\ &= \max_{\forall x \in J} \left| \sum_{i=m+1}^n L_{xxx}^{-1}(h(x)A_{i-1}(x)) \right| \\ &= \max_{\forall x \in J} \left| \sum_{i=m}^{n-1} L_{xxx}^{-1}(h(x)A_i(x)) \right| \\ &= \max_{\forall x \in J} \left| L_{xxx}^{-1}(h(x) \sum_{i=m}^{n-1} A_i(x)) \right|. \end{aligned}$$

From (4.7), we have $\sum_{i=m}^{n-1} A_i = f(S_{n-1}) - f(S_{m-1})$, so

$$\|S_n - S_m\| = \max_{\forall x \in J} |L_{xxx}^{-1}(h(x)f(S_{n-1}) - f(S_{m-1}))|,$$

according to the boundedness of L_{xxx}^{-1} , $h(x)$ and $f(u)$ is Lipschitz continuous; then

$$\begin{aligned} \|S_n - S_m\| &= \max_{\forall x \in J} |L_{xxx}^{-1}(h(x)f(S_{n-1}) - f(S_{m-1}))| \\ &\leq MNL \max_{\forall x \in J} |S_{n-1} - S_{m-1}| \\ &\leq \alpha \|S_{n-1} - S_{m-1}\|. \end{aligned}$$

Let $n = m + 1$; then

$$\|S_{m+1} - S_m\| \leq \alpha \|S_m - S_{m-1}\| \leq \alpha^2 \|S_{m-1} - S_{m-2}\| \leq \dots \leq \alpha^m \|S_1 - S_0\|.$$

From the triangle inequality, we have

$$\begin{aligned} \|S_n - S_m\| &\leq \|S_{m+1} - S_m\| + \|S_{m+2} - S_{m+1}\| + \dots + \|S_n - S_{n-1}\| \\ &\leq (\alpha^m + \alpha^{m+1} + \dots + \alpha^{n-1}) \|S_1 - S_0\| \\ &\leq \alpha^m (1 + \alpha + \alpha^2 + \dots + \alpha^{n-m-1}) \|S_1 - S_0\| \\ &\leq \alpha^m \left(\frac{1 - \alpha^{n-m}}{1 - \alpha} \right) \|u_1\|. \end{aligned}$$

Since $0 < \alpha < 1$, we have $1 - \alpha^{n-m} < 1$; then $\|S_n - S_m\| \leq (\alpha^m / (1 - \alpha)) \max_{x \in J} \|u_1(x)\|$. As $|u_1(x)| < \infty$, as $m \rightarrow \infty$, $\|S_n - S_m\| \rightarrow 0$. We conclude that $\{S_n\}_{n=0}^\infty$ is a Cauchy sequence in $C[J]$; this implies that there exists a $u(x) \in C[J]$ such that $\lim_{n \rightarrow \infty} S_n = u$, that is, $u(x) = \sum_{n=0}^\infty u_n(x)$. The series converges and the proof is complete. \square

In the following, we give three examples to demonstrate the effectiveness of the algorithm.

5. Examples

EXAMPLE 1. Consider the following equation:

$$\begin{cases} u^{(3)}(x) - e^{u(x)}(0.5 - e^{u(x)}) = g(x), & x \in [0, 1], \\ u(0) = \ln(10000), \quad u(1) = \ln(10001), \quad u'(0) = \frac{1}{10000}, \end{cases} \quad (5.1)$$

where $h(x) = 1$, $g(x) = (-2 - \frac{1}{2}\sqrt{x}(9998 + x)(10000 + x))/(1000 + x)^3$; the exact solution is $u(x) = \ln(1/(x + 10000))$; by (4.4), (4.5), we select 100 points in $[0, 1]$ and get the approximate solution $\tilde{u}_5(x) = \sum_{i=0}^5 u_i(x)$; the results are shown in Table 1.

TABLE 1. Relative error of approximate solutions at $m = 5$.

Node	True solution	Approximate solution	Relative error
1/100	9.210347	9.210345	2.68994E-08
11/100	9.210355	9.210353	2.95887E-07
21/100	9.210368	9.210363	5.64863E-07
31/100	9.210370	9.210360	8.33828E-07
41/100	9.210380	9.210370	1.10278E-06
51/100	9.210390	9.210380	1.37172E-06
61/100	9.210400	9.210390	1.64066E-06
71/100	9.210410	9.210390	1.90958E-06
81/100	9.210420	9.210400	2.17849E-06
91/100	9.210430	9.210410	2.44738E-06

EXAMPLE 2. Consider another differential equation, where the nonlinear term is different from that in Example 1.

$$\begin{cases} u^{(3)}(x) - \ln(u(x)) = g(x), & x \in [0, 1], \\ u(0) = 1, \quad u(1) = \frac{10001}{10000}, \quad u'(0) = \frac{1}{10000}, \end{cases} \quad (5.2)$$

where $h(x) = 1, g(x) = -\ln(1 + x/10000)$; the exact solution is $u(x) = 1 + x/10000$. We choose 100 points in $[0, 1]$ by making use of (2.10) and get the approximate solution $\tilde{u}_5(x) = \sum_{i=0}^5 u_i(x)$; the relative errors are shown in Table 2.

TABLE 2. Relative error of approximate solutions at $m = 5$.

Node	True solution	Approximate solution	Relative error
1/100	1.000000	1.000000	4.95025E-07
11/100	1.000010	1.000020	5.44530E-06
21/100	1.000020	1.000030	1.03956E-05
31/100	1.000030	1.000050	1.53460E-05
41/100	1.000040	1.000060	2.02964E-05
51/100	1.000050	1.000080	2.52469E-05
61/100	1.000060	1.000090	3.01974E-05
71/100	1.000070	1.000110	3.51480E-05
81/100	1.000080	1.000120	4.00986E-05
91/100	1.000090	1.000140	4.50493E-05

EXAMPLE 3. We consider the following equation for comparing our algorithm with the existing algorithm [24].

$$\begin{cases} u^{(3)}(x) - xu''(x) - u'(x) - xu(x) - u^2(x) = g(x), \\ u(1) = 0, \quad u'(1) = 0, \quad u'(0) = 0, \end{cases} \tag{5.3}$$

where $h(x) = 1, g(x) = 18(x - 1)^2 + 36(x - 1)x - 2(x - 1)^3 + 6x^2 - (x - 1)^2x^2 - (x - 1)^3x^3 - (x - 1)^6x^4 - x(2(x - 1)^3 + 12x(x - 1)^2 + 6(-1 + x)x^2)$; the exact solution is $u(x) = x^2(x - 1)^2$. We select ten points in $[0, 1]$ by making use of (2.10) and get the approximate solution $\tilde{u}_5(x) = \sum_{i=0}^5 u_i(x)$; the comparison results are shown in Table 3.

TABLE 3. Comparison of the relative error at $m = 5$ in the paper [24].

Node	True solution	Approximate solution	Our algorithm	Algorithm in [24]
1/10	0.00108830	0.00108929	9.097650E-04	4.71412E-03
1/5	0.00116726	0.00116800	6.381110E-04	1.23304E-03
3/10	0.00121262	0.00121313	4.247230E-04	1.74306E-03
2/5	0.00121921	0.00121952	2.542720E-04	4.25259E-03
1/2	0.00118380	0.00118394	1.154000E-04	6.10987E-03
3/5	0.00110530	0.00110530	1.034070E-06	7.10274E-03
7/10	0.00098487	0.00098477	1.036380E-04	6.88828E-03
4/5	0.00082595	0.00082579	2.031170E-04	4.62244E-03
9/10	0.00063401	0.00063381	3.191620E-04	3.53199E-03

EXAMPLE 4. For comparing our algorithm with the existing algorithm in [7], we consider the following equation:

$$\begin{cases} u^{(3)}(x) + 2\alpha \operatorname{Re} u(x)u'(x) + (4 - Ha)u'(x) = 0, \\ u(0) = 1, \quad u(1) = 0, \quad u'(0) = 0, \end{cases} \quad (5.4)$$

where $\operatorname{Re} = 50$ and $Ha = 1000$. The same as in the paper [7], we introduce the error as follows:

$$\text{Error} = \left| \frac{\text{Numerical 1} - \text{Numerical 2}}{\text{Numerical 2}} \right|,$$

where Numerical 1 denotes the numerical solution of [7] and Numerical 2 denotes the numerical solution in our algorithm. We select twenty points in $[0, 1]$ by using (2.10) and get the approximate solution $\tilde{u}_{10}(x) = \sum_{i=0}^{10} u_i(x)$; the comparison results are shown in Table 4 for $\alpha = 5$.

TABLE 4. Comparison of the relative error at $\alpha = 5$ in the paper [7].

Node	Error 1 in [7]	Error 2 in [7]	Error in our algorithm
0.00	0.00E+00	0.00E+00	0.00E+00
0.05	1.00E-10	1.00E-10	6.66E-11
0.10	1.01E-10	2.02E-10	3.63E-11
0.15	8.18E-10	1.12E-09	1.02E-10
0.20	1.87E-09	6.24E-10	2.16E-11
0.25	2.76E-09	1.28E-09	3.85E-10
0.30	8.09E-09	5.47E-09	8.98E-11
0.35	1.66E-08	1.26E-08	1.24E-10
0.40	1.49E-08	1.28E-08	1.61E-10
0.45	8.29E-09	1.86E-09	2.01E-11
0.50	4.58E-09	3.27E-09	2.40E-11
0.55	2.38E-09	1.16E-08	2.77E-10
0.60	8.32E-09	1.77E-08	3.07E-11
0.65	1.15E-08	2.69E-08	3.28E-11
0.70	1.45E-08	5.40E-09	3.36E-11
0.75	1.59E-08	5.53E-08	3.27E-10
0.80	2.18E-08	4.49E-08	3.01E-09
0.85	2.06E-08	2.16E-08	2.54E-10
0.90	1.97E-08	1.53E-07	2.01E-08
0.95	2.49E-08	5.25E-07	1.87E-09
1.00	0.00E+00	0.00E+00	0.00E+00

6. Conclusions

In this paper, through the three examples, it can be seen that our algorithm is more accurate than the traditional algorithm. In comparison with other methods, the decomposition method we put forward is not only more accurate but also needs fewer steps; the process also saves a lot of time. Through the above numerical examples, we illustrated the practicality of the decomposition method we put forward for solving third-order nonlinear equations.

Acknowledgements. The authors of this research are grateful for the support of the National Natural Science Foundation of China (11401145), the Educational Department Scientific Technology Program of Heilongjiang Province (12531180 and 12541244) and the Academic Foundation for Youth of Harbin Normal University (11KXQ-04).

References

1. G. ADOMIAN, *Solving frontier problems of physics: the decomposition method* (Kluwer Academic, Boston, MA, 1994).
2. G. ADOMIAN, 'Explicit solutions of nonlinear partial differential equations', *Appl. Math. Comput.* 88 (1997) 117–126.
3. G. ADOMIAN, 'Solutions of nonlinear P.D.E.', *Appl. Math. Lett.* 11 (1998) no. 3, 121–123.
4. W. AL-HAYANI and L. CASASUS, 'The Adomian decomposition method in turning point problems', *Appl. Math. Comput.* 177 (2005) 187–203.
5. L. BOUGOFFA and R. C. RACH, 'Solving nonlocal boundary value problems for first- and second-order differential equations by the Adomian decomposition method', *Kybernetes* 42 (2013) no. 4, 641–664.
6. M. DEHGHAN, 'Application of the Adomian decomposition method for two-dimensional parabolic equation subject to nonstandard boundary specifications', *Appl. Math. Comput.* 157 (2004) 549–560.
7. A. DIB, A. HAIHAHEM and B. BOU-SAID, 'An analytical solution of the MHD Jeffery–Hamel flow by the modified Adomian decomposition method', *Comput. Fluids* 102 (2014) 111–115.
8. J. S. DUAN and R. C. RACH, 'A new modification of the Adomian decomposition method for solving boundary value problems for higher order nonlinear differential equations', *Appl. Math. Comput.* 218 (2011) no. 8, 4090–4118.
9. A. EBAID, 'A new numerical solution for the MHD peristaltic flow of a bio-fluid with variable viscosity in a circular cylindrical tube via Adomian decomposition method', *Phys. Lett. A* 372 (2008) 5321–5328.
10. A. EBAID, 'Exact solutions for the generalized Klein–Gordon equation via a transformation and expansion method and comparison with Adomian's method', *J. Comput. Appl. Math.* 223 (2009) 278–290.
11. A. EBAID, 'A new analytical and numerical treatment for singular two-point boundary value problems via the Adomian decomposition method', *J. Comput. Appl. Math.* 235 (2011) no. 8, 1914–1924.
12. N. T. ELDABE, E. M. ELGHAZY and A. EBAID, 'Closed form solution to a second order boundary value problem and its application in fluid mechanics', *Phys. Lett. A* 363 (2007) 257–259.
13. I. L. EL-KALLA, 'Convergence of Adomian's method applied to a class of Volterra type integro-differential equations', *Int. J. Differ. Equ. Appl.* 10 (2005) no. 2, 225–234.
14. R. E. EWING and T. LIN, 'A class of parameter estimation techniques for fluid flow in porous media', *Adv. Water Resour.* 14 (1991) 89–97.
15. L. FORMAGGIA, F. NOBILE, A. QUARTERONI and A. VENEZIANI, 'Multiscale modelling of the circulatory system: a preliminary analysis', *Comput. Vis. Sci.* 2 (1999) 75–83.
16. F. GENG, 'A novel method for nonlinear two-point boundary value problems: combination of ADM and RKM', *Appl. Math. Comput.* 217 (2011) 4676–4681.
17. L.-J. GUO, 'Existence of positive solutions for nonlinear third-order three-point boundary value problems', *Appl. Math. Comput.* 68 (2008) 3151–3158.
18. I. HASHIM, 'Adomian decomposition method for solving BVPs for fourth-order integro-differential equations', *Appl. Math. Comput.* 193 (2006) 658–664.
19. B. HOPKINS, 'Third-order boundary value problems with sign-changing solutions', *Appl. Math. Comput.* 67 (2007) 126–137.
20. D. LESNIC, 'A computational algebraic investigation of the decomposition method for time-dependent problems', *Appl. Math. Comput.* 119 (2001) 197–206.
21. D. LESNIC and L. ELLIOTT, 'The decomposition approach to inverse heat conduction', *J. Math. Anal. Appl.* 232 (1999) no. 82, 82–98.
22. Y. LI, 'Positive periodic solutions for fully third-order ordinary differential equations', *Appl. Math. Comput.* 59 (2010) 3464–3471.
23. X. LV and M. CUI, 'Solving a singular system of two nonlinear ODEs', *Appl. Math. Comput.* 198 (2008) 534–543.
24. X. LV and M. CUI, 'Existence and numerical method for nonlinear third-order boundary-value problem in the reproducing kernel space', *Bound. Value Probl.* 2010 (2010) 1–13.
25. S. MEHRKANOON, 'A direct variable step block multistep method for solving general third-order ODEs', *Numer. Algorithms* 57 (2011) 53–66.
26. N. T. SHAWAGFEH, 'Analytic approximate solution for a nonlinear oscillator equation', *Comput. Math. Appl.* 31 (1996) 135–141.
27. P. SHI, 'Weak solution to evolution problem with a nonlocal constraint', *SIAM J. Anal.* 24 (1993) 46–58.
28. A. M. WAZWAZ, 'A reliable modification of Adomian's decomposition method', *Appl. Math. Comput.* 102 (1999) 77–96.

29. A. M. WAZWAZ, 'A reliable algorithm for obtaining positive solutions for nonlinear boundary value problems', *Math. Comput. Simulation* 41 (2001) 1237–1244.
30. A. M. WAZWAZ, 'The modified decomposition method applied to unsteady flow of gas through a porous medium', *Appl. Math. Comput.* 118 (2001) 123–132.
31. A. M. WAZWAZ, 'A reliable algorithm for solving boundary value problems for higher-order integro-differential equations', *Appl. Math. Comput.* 118 (2001) 327–342.
32. A. M. WAZWAZ, 'The numerical solution of fifth-order boundary value problems by the decomposition method', *Appl. Math. Comput.* 136 (2001) 259–270.

Xueqin Lv
School of Mathematics and Sciences
Harbin Normal University
Harbin, Heilongjiang, 150025
China
hashidalvxueqin@126.com

Jianfang Gao
School of Mathematics and Sciences
Harbin Normal University
Harbin, Heilongjiang, 150025
China
09151108@163.com