Iranian Journal of Numerical Analysis and Optimization
Vol. 5, No. 2, (2015), pp 29-43

A practical review of the Adomian decomposition method: computer implementation aspects

A. Molabahrami

Abstract

In this paper, a practical review of the Adomian decomposition method, to extend the procedure to handle the strongly nonlinear problems under the mixed conditions, is given and the convergence of the algorithm is proved. For this respect, a new and simple way to generate the Adomian polynomials, for a general nonlinear function, is proposed. The proposed procedure, provides an explicit formula to calculate the Adomian polynomials of a nonlinear function. The efficiency of the approach will be shown by applying the procedure on several interesting integro-differential problems. The Mathematica programs generating the Adomian polynomials and Adomian solutions based on the procedures in this paper are designed.

Keywords: Adomian decomposition method; Adomian polynomials; Nonlinear integro-differential problems; Series solution; Strongly nonlinear problems; Explicit machine computation and programs.

1 Introduction

To construct series pattern solution for a problem with strong nonlinearity, it is necessary to construct nonlinear terms of the governing equation in the form of a series by using the components of the solution series. To the end, one of the best and suitable way is to use the Adomian polynomials. The so-called Adomian polynomials are used to deduce the recursive relation during the implementation of the Adomian decomposition method (ADM) while solving nonlinear problems. The main aim of the present paper is to provide a simple and new method to handle a strongly nonlinear problem by using ADM in the frame of a symbolic computer program so that by giving linear operator, it generates: initial guess, integral inverse of the linear operator, recursive relation and the terms of solution series automatically. To achieve this purpose, we first propose an explicit formula to calculate the

Received 7 October 2014; revised 20 December 2014; accepted 18 March 2015
A. Molabahrami
Department of Mathematics, Ilam University, PO Box 69315516, Ilam, Iran e-mail: bahrami@iust.ac.ir

Adomian polynomials and the Adomian series of a general nonlinear function and implement the proposed algorithms in Mathematica. In this respect, we first outline the modifications of some definitions as already given in [8]. Let u be a function of the parameter λ, whose Maclaurin series is given by

$$u(\lambda) = \sum_{n=0}^{+\infty} u_n \lambda^n. \tag{1}$$

This series is called the parametric series of u. Let ϕ be a function of the parameter λ, the mth-order parametric derivative of ϕ is

$$\mathbb{D}_m[\phi] = \frac{1}{m!} \frac{\partial^m \phi}{\partial \lambda^m}\bigg|_{\lambda=0}, \tag{2}$$

where $m \geq 0$ is an integer. The mth-order Adomian polynomial of ϕ is

$$A_m(\phi(u)) = \mathbb{D}_m \left[\phi(u(\lambda))\right], \tag{3}$$

where $m \geq 0$ is an integer and $A_m(\phi(u)) = A_m(\phi(u); u_0, u_1, ..., u_m)$.

Remark 1. For the case $0 \leq \lambda \leq 1$, the parametric series (1) and parametric derivative (2) reduce to homotopy series and homotopy derivative respectively [8].

Several algorithms [3,5,12,13] for symbolic programming have since been devised to efficiently generate the Adomian polynomials quickly to high orders, for example, a convenient formula for the Adomian polynomials is the rule of Rach, which reads (see Page 16 in [1] and Page 51 in [2])

$$A_n(f(u)) = \sum_{k=1}^{n} f^{(k)}(u_0) C(k,n), \ n \geq 1, \tag{4}$$

where the $C(k,n)$ are the sums of all possible products of k components $u_i, i = 1, 2, ..., n-k+1$, whose subscripts sum to n, divided by the factorial of the number of repeated subscripts. An equivalent expression of Equation (4) is

$$A_n(f(u)) = \sum_{p_1+2p_2+...+np_n=n} f^{(p_1+p_2+...+p_n)}(u_0) \prod_{s=1}^{n} \frac{u_s^{p_s}}{p_s!}, \ n \geq 1. \tag{5}$$

In the present paper, we propose an explicit formula to calculate the $C(k,n)$ in (4) and we show that for rapid computer-generation of the Adomian polynomials there is no need to use the (5).

2 Adomian polynomials for a general function

In this section, we first outline two theorems as already given in [7,10]. Then using them, we propose a new theorem which provides a new and simple way to calculate the Adomian polynomials for a general smooth function. Here, we mention them with a minor modification. For simplicity, we use the following notation

$$\widehat{u}_{m,n} = \sum_{i=n}^{m} u_i \lambda^i.$$

Theorem 1. *For function $f(u) = u^k$, the corresponding mth-order Adomian polynomial is given by*

$$A_m(u^k) = \sum_{r_1=0}^{m} u_{m-r_1} \sum_{r_2=0}^{r_1} u_{r_1-r_2} \sum_{r_3=0}^{r_2} u_{r_2-r_3} \cdots \sum_{r_{k-2}=0}^{r_{k-3}} u_{r_{k-3}-r_{k-2}} \sum_{r_{k-1}=0}^{r_{k-2}} u_{r_{k-2}-r_{k-1}} u_{r_{k-1}}, \qquad (6)$$

where $m \geq 0$ and $k \geq 0$ are positive integers.

Proof. Considering the definition (2), refer to [10]. □

Theorem 2. *For parametric series (1), it holds*

$$\mathbb{D}_m \left[f(u(\lambda)) \right] = \mathbb{D}_m \left[f(\widehat{u}_{m,0}) \right],$$

where f is a smooth function.

Proof. Refer to [7]. □

Corollary 1. *From Theorem 1, we find*

$$u^k(\lambda) = \left(\sum_{n=0}^{+\infty} u_n \lambda^n \right)^k = u_0^k + \sum_{m=1}^{+\infty} A_m(u^k) \lambda^m, \qquad (7)$$

where the Adomian polynomials $A_m(u^k)$ are given by (6) .

Remark 2. It is clear that $A_m(u^k)$ in Theorem 1 can easily be calculated by a simple code by using a symbolic software such as Mathematica. For this respect, the **Code. 1**, reported in Appendix, can be used in Mathematica. For instance, by ADPforPowerLaw[4,6], the $A_4(u^6)$ is calculated as follows

$$A_4(u^6) = 15u_0^2u_1^4 + 60u_0^3u_1^2u_2 + 15u_0^4u_2^2 + 30u_0^4u_1u_3 + 6u_0^5u_4.$$

Corollary 2. *From Theorem 2, for $m \geq n$, we find*

$$\mathbb{D}_m\left[(f(\widehat{u}_{\infty,n})\right] = \mathbb{D}_m\left[f(\widehat{u}_{m,n})\right],$$

where f is a smooth function.

Corollary 3. *From Corollary 2 and Theorem 1, for $m \geq k$, we find*

$$\mathbb{D}_m\left[\left(\hat{u}_{m,1}\right)^k\right] = \sum_{r_1=0}^{m-1} \sum_{\substack{r_2=0 \\ r_2\neq r_1}}^{r_1} \cdots \sum_{\substack{r_{k-2}=0 \\ r_{k-2}\neq r_{k-3}}}^{r_{k-3}} \sum_{\substack{r_{k-1}=0 \\ r_{k-1}\neq r_{k-2}}}^{r_{k-2}} u_{r_{k-1}} \prod_{j=0}^{k-2} u_{r_j - r_{j+1}}, \quad (8)$$

where $r_0 = m$.

Corollary 4. *It is easy to see that*

$$\mathbb{D}_m\left[\left(\hat{u}_{m,1}\right)^k\right] = \begin{cases} 0, & m < k, \\ u_1^k, & m = k. \end{cases} \quad (9)$$

Corollary 5. *Let $kn \geq m+1$ and $n \geq 1$, we find*

$$\mathbb{D}_m\left[\left(\widehat{u}_{\infty,n}\right)^k\right] = 0.$$

Remark 3. A sample Mathematica program for $\mathbb{D}_m\left[\left(\hat{u}_{m,1}\right)^k\right]$ in (8) is given by the **Code. 2** reported in Appendix. An alternative way is to use the Theorem 1 by taking $u_0 = 0$. To achieve this purpose, In Code 1, in the last command, the $Expand[D_{k,m}]$ is replaced by $Expand[D_{k,m}]/.u_0 \to 0$.

The following theorem provides a suitable and simple way to construct a recurrent relation for a general smooth function appeared within a structure with nonlinear terms in the equations.

Theorem 3. *Assume that $f(u)$ has the Taylor expansion with respect to u_0, then*

$$A_m\left(f\left(u\right)\right) = \sum_{k=1}^{m} \frac{f^{(k)}(u_0)}{k!} \mathbb{D}_m\left[\left(\widehat{u}_{m,1}\right)^k\right]. \quad (10)$$

Proof. Expanding $f(u)$ in Taylor series with respect to u_0, one has

$$f(u) = f(u_0) + \sum_{k=1}^{+\infty} \frac{f^{(k)}(u_0)}{k!} \left(u - u_0\right)^k. \quad (11)$$

From the (11), we have

$$A_m\left[f\left(u\right)\right] = \mathbb{D}_m\left[\sum_{k=1}^{\infty} \frac{f^{(k)}(u_0)}{k!} \left(u(\lambda) - u_0\right)^k\right],$$

recalling the Corollaries 5 and 2, we find

$$A_m\left(f\left(u\right)\right)=\sum_{k=1}^{m}\frac{f^{(k)}(u_0)}{k!}\mathbb{D}_m\left[\left(u(\lambda)-u_0\right)^k\right]=\sum_{k=1}^{m}\frac{f^{(k)}(u_0)}{k!}\mathbb{D}_m\left[\left(\widehat{u}_{m,1}\right)^k\right].$$

This ends the proof. □

Remark 4. The expansion (11) is rigorously guaranteed by the Cauchy-Kovaleski theorem. In fact, by using this theorem the Taylor expansion about the function $u_0(x)$ can be seen as an expansion in Banach space [4].

Remark 5. The expression $\mathbb{D}_m\left[\left(\widehat{u}_{m,1}\right)^k\right]$ in (10) can easily be calculated by (8) or Theorem 1 with taking $u_0=0$.

Corollary 5. *From Theorem 3, we find*

$$f(u(\lambda))=f(u_0)+\sum_{m=1}^{+\infty}w_m\lambda^m, \tag{12}$$

where $w_m=A_m(f(u))$ can be calculated by (10).

Remark 6. To calculate $A_m(f(u))$, the Code. 3, which is given in the Appendix, can be used in Mathematica. Using the $AdomianPolynomial[f[u],m]$, for $m=1,2,3,4$, the corresponding Adomian polynomials are given as follows

$A_1(f(u))=u_1f'(u_0)$,

$A_2(f(u))=u_2f'(u_0)+\frac{1}{2}u_1^2f''(u_0)$,

$A_3(f(u))=u_3f'(u_0)+u_1u_2f''(u_0)+\frac{1}{6}u_1^3f'''(u_0)$,

$A_4(f(u))=u_4f'(u_0)+\frac{1}{48}\left(24u_2^2+48u_1u_3\right)f''(u_0)+\frac{1}{2}u_1^2u_2f'''(u_0)+\frac{1}{24}u_1^4f^{(4)}(u_0)$.

Also, the Code. 4, reported in Appendix, shows an alternative way to calculate the Adomian polynomials by using the (3) and Corollary 2.

2.1 An improvement for the Rach's rule

According to (4) and Theorem 3, we find

$$C(k,n)=\frac{1}{n!}\mathbb{D}_n\left[\left(\widehat{u}_{n,1}\right)^k\right],$$

thus, the Rach's rule gets its explicit presentation.

3 A practical review of the Adomian decomposition method

In this section, we propose a practical review of the ADM and implementation it as an automatic program in the frame of a symbolic software such as Mathematica.

Adomian decomposition method [1], was first proposed by Adomian in 1988 and was further developed and improved by Adomian [2,3]. To illustrate the ADM for solving a general nonlinear problem, consider the following nonlinear problem

$$\begin{cases} \mathcal{N}(u) = f, \\ \mathbb{B}(u) = f_0, \end{cases} \tag{13}$$

where $\mathcal{N}$ is a general nonlinear operator, $\mathbb{B}$ is a linear initial/boundary operator, u is the unknown function that will be determined and f and g are given functions. An especial case of (13), $f = 0$, was given in [9]. The ADM consists in looking for the solution of Equation (13) in the series form

$$u = u_0 + \sum_{n=1}^{+\infty} u_n, \tag{14}$$

where u_0 is an initial guess and has the property

$$\mathbb{B}(u_0) = f_0, \tag{15}$$

and the other components of the solution series (14) have the property

$$\mathbb{B}(u_n) = 0, n \geq 1. \tag{16}$$

Thus, from (15) and (16) the solution series given by (14), satisfies the conditions of (13). For the nonlinear conditions, the ADM needs an improvement.

To construct series pattern solution, (14) by ADM, in the first step the nonlinear operator $\mathcal{N}$ is decomposed as

$$\mathcal{N}(u) = L(u) + N(u), \tag{17}$$

where L is a linear operator and $N = \mathcal{N} - L$. To continue in ADM, the nonlinear operator N is decomposed as

$$N(u) = \sum_{n=0}^{+\infty} A_n, \tag{18}$$

where A_n is the nth-order Adomian polynomial of N. Using (17), the Equation (13) becomes

$$\begin{cases} L(u) + N(u) = f, \\ \mathbb{B}(u) = f_0, \end{cases} \tag{19}$$

by ignoring the condition $\mathbb{B}(u) = f_0$, the solution of Equation (19) satisfies

$$u = u_g + L^{-1}[f] - L^{-1}[N(u)], \tag{20}$$

where u_g is the general solution of the linear equation $L(u) = 0$ and L^{-1} is the integral inverse of L. Now, by substituting (14) in (20) and considering (18), we get

$$\begin{cases} u_0 = u_g, \\ u_n = (1 - \chi_n)L^{-1}[f] - L^{-1}[A_{n-1}],\ n \geq 1. \end{cases} \tag{21}$$

where

$$\chi_n = \begin{cases} 0,\ n \leq 1, \\ 1,\ n \geq 2. \end{cases}$$

Recalling the condition $\mathbb{B}(u) = f_0$, we have

$$\begin{cases} u_0 = \mathbb{B}(u_g) = u_g^*, \\ \mathbb{B}(u_n) = 0,\ n \geq 1. \end{cases} \tag{22}$$

Thus, the recurrent relation to find a series pattern solution of Equation (13) by means of ADM is

$$\begin{cases} u_0 = u_g^*, \\ u_n = (1 - \chi_n)\, L^{-1}[f] - L^{-1}[A_{n-1}],\ \mathbb{B}(u_n) = 0, \end{cases} \tag{23}$$

let $uParticular = L^{-1}[-A_{n-1}] + (1 - \chi_n)\, L^{-1}[f]$, then, for $n \geq 1$, we find

$$\begin{cases} u_0 = u_g^*, \\ \\ u_n = uParticular + uParticularStar, \end{cases} \tag{24}$$

where $uParticularStar = \mathbb{B}\left(L^{-1}[A_{n-1}]\right) - (1 - \chi_n)\left(\mathbb{B}(L^{-1}[f])\right)$. It is important to notice that the definition of the integral inverse L^{-1} in (23), depends on the condition (16). L^{-1} in (24) can be defined such that the $L^{-1}[*]$ gives a particular solution of the equation $L(u) = *$.

Remark 7. It should be emphasized that the main step of the ADM is to choose a proper linear operator. According to (24), the linear operator should be defined so that

1. The following problem has a solution

$$\begin{cases} L(u_0) = 0, \\ \mathbb{B}(u_0) = f_0. \end{cases} \tag{25}$$

According to the Equations (20) and (21), the problem (25) gives the initial guess.

2. The following equation has a solution

$$\mathbb{B}\left(L^{-1}[A_{n-1}]\right) - (1-\chi(n))\,\mathbb{B}\left(L^{-1}[f]\right) = \mathbb{B}(u_g), \tag{26}$$

where $n \geq 1$. This equation helps to define the integral inverse of L.

3. The solution series, given by (14), is convergent. For this respect, some conditions have been given in [11].

3.1 Decomposition series and convergence analysis

An important hypothesis in ADM is the composition of the nonlinear operator N as shown in (18). There are some serious questions about (18) such as, from where does the series $\sum_{n=0}^{+\infty} A_n$ derive? Does it always converge? Is its sum really N? What are the other series that we could use instead of $\sum_{n=0}^{+\infty} A_n$? In order to answer these questions, and also, in order to explain some other issues, Gabet proposed a theory which explains and justifies the practical method [6]. In this subsection, we answer to the above mentioned questions by using the results of the section 2.

To derive (18), from (12), under the assumptions of the Theorem 3, we have

$$N(u(\lambda)) = N(u_0) + \sum_{n=1}^{+\infty} A_n \lambda^n, \tag{27}$$

where A_n is the nth-order Adomian polynomial of N. Let the parametric series (1) is convergent at $\lambda = 1$ and $s = \sum_{n=0}^{+\infty} u_n$, then

$$N(s) = A_0 + \sum_{n=1}^{+\infty} A_n. \tag{28}$$

According to the (23), it is clear that if the solution series (14) is convergent to s, then the decomposition series $\sum_{n=0}^{+\infty} A_n$ converges to $f - L(s)$. On the other hand, according to the (23), the decomposition series $\sum_{n=0}^{+\infty} A_n$ converges to $N(s)$. Therefore, if the solution series given by ADM, of the form (14) generated by (24), is convergent, then it must be an exact solution of the considered nonlinear problem, denoted by (13). Bearing in mind the above discussion, we can express the following theorem on the convergency of the ADM.

Theorem 4. *Assume that the ADM solution series given by (14) and (24) is convergent, then, under the assumptions of the Theorem 3, it must be an exact solution of the (13).*

4 Test examples

To show the efficiency of Theorem 4, described in the previous section, some examples are presented. We consider the mth-order nonlinear integro-differential equation with variable coefficients

$$\sum_{r=0}^{m} f_r(x)u^{(r)}(x) + \lambda \int_a^{\beta} K(x,t)F(u(t))dt = g(x), \;\; x \in [a,b], \tag{29}$$

where $F(u(x))$ is a function of $u(x)$, $K(x,t)$ is the kernel of the integro-differential equation, λ is a parameter, $g(x)$ is the data function, $f_r(x)$ is a function with respect to x, $u(x)$ is the unknown function that will be determined. For the Fredholm and Volterra kinds we take $\beta = b$ and $\beta = x$ respectively. We consider Equation (29) under the mixed conditions

$$\sum_{r=0}^{m-1} \Big(a_{rj}u^{(r)}(a) + b_{rj}u^{(r)}(b) + c_{rj}u^{(r)}(c) \Big) = \mu_j, \;\; j = 0,1,2,...,m-1, \tag{30}$$

where a_{rj}, b_{rj}, c_{rj} and μ_j are constants, and c, $a < c < b$, is a constant. For the linear case, it is assumed that $F(u(x)) = u(x)$. The computations will be performed using the program ***ADMforIDEs*** reported in the appendix. The program ***ADMforIDEs*** has designed in a general manner so that, for a given problem and its related linear operator, it calculates the integral inverse of the linear operator, initial guess, recursive relation and the terms of solution series automatically.

For Examples 1-4, we choose the linear operator as follows

$$L(u(x)) = \frac{d^2}{dx^2}u(x), \tag{31}$$

therefore, from the (22) and (31), we find

$$u_0(x) = x. \tag{32}$$

We emphasize that no attempt has been made here to obtain all solutions of a given problem.

Example 1. Consider the following nonlinear problem

$$\begin{cases} u''(x) + xu'(x) - u(x) + x\int_0^{\frac{\pi}{2}} t\sin\left(u(t)\right)dt = x \\ u(\frac{1}{2}) + u'(0) = \frac{3}{2}, \;\; 2u(\frac{1}{2}) + u'(1) = 2. \end{cases} \tag{33}$$

An exact solution is $u(x) = x$. Starting by (32), the recurrent relation (24) gives

$$u_n(x) = 0,\ n \geq 1,$$

therefore, using the ADM, the exact solution of Equation (33) is obtained as follows

$$u(x) = u_0(x) + \sum_{n=1}^{+\infty} u_n(x) = x. \tag{34}$$

Computations have been carried out by program ***ADMforIDEs***. The list of commands is as follows
$conditions[u\_] := \{(u[x]/.x \to \frac{1}{2}) + (\mathrm{D}[u[x], x]/.x \to 0) - \frac{3}{2},$
$2 * (u[x]/.x \to \frac{1}{2}) + (\mathrm{D}[u[x], x]/.x \to 1) - 2\};$
$Lcoefficients = \{0, 0, 1\};$
$Problemcoefficients = \{-1, x, 1\};$
ADMforIDEs$[\mathrm{Sin}[u], x, x * t, 0, \frac{\pi}{2}, 1, 2, 5]$

Example 2. Consider the following nonlinear problem

$$\begin{cases} u''(x) + xu'(x) - u(x) + 2x \int_0^x te^{-u^2(t)}dt = x - xe^{-x^2}, \\ u(\frac{1}{2}) + u'(0) = \frac{3}{2},\ 2u(\frac{1}{2}) + u'(1) = 2. \end{cases} \tag{35}$$

The exact solution is $u(x) = x$. Using the recurrent relation (24) with initial guess (32), we find

$$u_n(x) = 0,\ n \geq 1,$$

therefore, the ADM gives the exact solution of Equation (35) as follows

$$u(x) = u_0(x) + \sum_{n=1}^{+\infty} u_n(x) = x. \tag{36}$$

Computations have been carried out with the help of the program ***ADMforIDEs***. The list of commands is as follows
$conditions[u\_] := \{(u[x]/.x \to \frac{1}{2}) + (\mathrm{D}[u[x], x]/.x \to 0) - \frac{3}{2},$
$2 * (u[x]/.x \to \frac{1}{2}) + (\mathrm{D}[u[x], x]/.x \to 1) - 2\};$
$Lcoefficients = \{0, 0, 1\};$
$Problemcoefficients = \{-1, x, 1\};$
ADMforIDEs$[\mathrm{Exp}[-u^2], x - x * \mathrm{Exp}[-x^2], 2 * x * t, 0, x, 1, 2, 5]$

Example 3. Consider the following nonlinear problem

$$\begin{cases} u''(x) + xu'(x) - u(x) + (2m+2)x \int_0^1 tu^{2m}(t)dt = x, \\ u(\frac{1}{2}) + u'(0) = \frac{3}{2},\ 2u(\frac{1}{2}) + u'(1) = 2, \end{cases} \tag{37}$$

where m is a positive integer. An exact solution is $u(x) = x$. Recalling the recurrent relation (24) and initial guess (32), we obtain

$$u_n(x) = 0,\ n \geq 1,$$

therefore, the ADM obtains the exact solution of Equation (37) as follows

$$u(x) = u_0(x) + \sum_{n=1}^{+\infty} u_n(x) = x. \tag{38}$$

To achieve the computations by program ***ADMforIDEs***, the list of commands is as follows
$conditions[u\_] := \{(u[x]/.x \to \frac{1}{2}) + (\mathrm{D}[u[x], x]/.x \to 0) - \frac{3}{2},$
$2 * (u[x]/.x \to \frac{1}{2}) + (\mathrm{D}[u[x], x]/.x \to 1) - 2\};$
$Lcoefficients = \{0, 0, 1\};$
$Problemcoefficients = \{-1, x, 1\};$
$\mathrm{Assumptions} = m > 0 \&\& \mathrm{Element}[m, \mathrm{Integers}];$
$\mathrm{Refine}[\mathbf{ADMforIDEs}[u^{2m}, x, (2m+2) * x * t, 0, 1, 1, 2, 5]]$

Example 4. Consider the following nonlinear problem

$$\begin{cases} u''(x) + xu'(x) - u(x) + \frac{4m+1}{2m} x \int_0^1 t \sqrt[2m]{u(t)}dt = x, \\ u(\frac{1}{2}) + u'(0) = \frac{3}{2},\ 2u(\frac{1}{2}) + u'(1) = 2, \end{cases} \tag{39}$$

where m is a positive integer. An exact solution is $u(x) = x$. Putting to use the recurrent relation (24) and initial guess (32), we get

$$u_n(x) = 0,\ n \geq 1,$$

therefore, the ADM leads to the exact solution of Equation (39) as follows

$$u(x) = u_0(x) + \sum_{n=1}^{+\infty} u_n(x) = x. \tag{40}$$

For performing the computations with the help of the program ***ADMforIDEs***, the list of commands is as follows
$conditions[u\_] := \{(u[x]/.x \to \frac{1}{2}) + (\mathrm{D}[u[x], x]/.x \to 0) - \frac{3}{2},$
$2 * (u[x]/.x \to \frac{1}{2}) + (\mathrm{D}[u[x], x]/.x \to 1) - 2\};$
$Lcoefficients = \{0, 0, 1\};$
$Problemcoefficients = \{-1, x, 1\};$
$\mathrm{Assumptions} = m > 0 \&\& \mathrm{Element}[m, \mathrm{Integers}];$
$\mathrm{Refine}[\mathbf{ADMforIDEs}\ [\sqrt[2m]{u}, x, \frac{4m+1}{2m} * x * t, 0, 1, 1, 2, 5]]$

Remark 8. In Examples 1-4, the first-order iteration gives the exact solution.

Example 5. Consider the following problem

$$\begin{cases} u'(x) + u(x) - \int_0^1 u(t)dt = \frac{1}{2}\left(\frac{1}{e^2} - 1\right), \\ u(0) = 1, \end{cases} \tag{41}$$

we choose the linear operator as follows

$$L(u(x)) = \frac{d}{dx}u(x) + u(x), \tag{42}$$

therefore, from the (22) and (42), we find $u_0(x) = e^{-x}$. Putting to use the recurrent relation (24) and obtained initial guess, we have

$$u_n(x) = \frac{1}{2}(e-1)^2(e^x - 1)e^{-x-n-1},\ n \geq 1,$$

therefore, the ADM leads to the exact solution of Equation (39) as follows

$$u(x) = u_0(x) + \sum_{n=1}^{+\infty} u_n(x) = \frac{1}{2}e^{-x-1}\left(-e^x + e^{x+1} + 1 + e\right). \tag{43}$$

For performing the computations with the help of the program ***ADMforIDEs***, the list of commands is as follows

$conditions[u\_] := \{(u[x]/.x \to 0) - 1\};$
$Lcoefficients = \{1, 1\};$
$Problemcoefficients = \{1, 1\};$
ADMforIDEs $\left[u, \frac{1}{2}\left(\frac{1}{e^2} - 1\right), 1, 0, 1, -1, 1, 5\right]$

Remark 9. The program ***ADMforIDEs***, reported in the Appendix, monitors some terms of the solution series, the number of the terms is given by **SolutionOrder**. To generate the general term of the solution series (if possible) and the closed form of the solution series (if possible), the user must be added the following commands to the end of the list of commands . It is important to emphasis that the following commands must be used in Mathematica 7 or later.

m1 =Input["By evaluating the obtained terms solution series, please input the index of the term in which it is possible to generate the general term of the solution series?"];
SolutionTerms = Join[Table[$u[m]$, $\{m$, m1, SolutionOrder$\}$]];
SolutionCoefficents = FindSequenceFunction[SolutionTerms, n];
Print ["$u_n(x)$=", TraditionalForm[SolutionCoefficents]] ;
FinalSolution = Sum[SolutionCoefficents, $\{n$, m1, Infinity$\}$,GenerateConditions → True];
Sol1 = Simplify $\left[\sum_{n=0}^{\text{m1}-1} u[n] + \text{FinalSolution}\right]$;
Print $\left["u(x)=\sum_{n=0}^{+\infty} u_n(x)=", \text{TraditionalForm[Sol1]}\right]$

5 Concluding remarks

By calculating the parametric derivative for a function of a more general type, described in Theorem 3, the Adomian decomposition method becomes a powerful analytic approach for obtaining convergent series solutions of strongly nonlinear problems governing physical models in applied science and engineering. Using the parametric derivative, some lemmas and theorems provided in ADM papers in the literature have been unified and modified here via some new theorems and corollaries. Also, by means of the (24) the Adomian decomposition method was extended to handel the nonlinear problems with the mixed conditions.

Acknowledgements

The author would like to thank the anonymous referees especially the second referee for his/her constructive comments and suggestions. Many thanks are due to financial support from the Ilam University of Iran.

Appendix (Mathematica programs)

Code. 1(A sample Mathematica program for $A_m(u^k)$ given by (6))

$\mathbf{ADPforPowerLaw}[m\_, k\_] := \text{Module}[\{\}, D_{1,j\_} := u_j; D_{i\_,0} := u_0^i;$
$D_{2,j\_} := \sum_{r=0}^{j} u_{j-r}u_r;\ D_{i\_,order\_} := \sum_{r=0}^{order} u_{order-r}D_{i-1,r};$
$\text{Print}[''A''_m,''\,(\,'',u^k,''\,)\,='', \text{Expand}[\text{TraditionalForm}[D_{k,m}]]]];$

Code. 2(A sample Mathematica program for $\mathbb{D}_m\left[(\hat{u}_{m,1})^k\right]$ given by (8))

$\mathbf{DnonU0}[m\_, k\_] := \text{Module}[\{\}, D_{1,j\_} := \text{If}[j > 0, u_j, 0]; D_{i\_,0} := 0; D_{2,j\_} := \sum_{r=1}^{j-1} u_{j-r}u_r;$
$D_{i\_,order\_} := \sum_{r=2}^{order-1} u_{order-r}D_{i-1,r}; \text{Print}[''\mathbb{D}''_m,''\,('',\hat{u}^k_{m,''1''},''\,)\,='',$
$\text{Expand}[\text{TraditionalForm}[D_{k,m}]]];$

Code. 3(A sample Mathematica program for $A_m(f(u))$ given by (10))

$\mathbf{AdomianPolynomial}[function\_, m\_] := \text{Module}[\{\}, D_{1,j\_,x\_} := \text{If}[j > 0, x_j, 0];$
$D_{i\_,0,x\_} := 0; D_{2,j\_,x\_} := \sum_{r=1}^{j-1} x_{j-r}x_r;\ D_{i\_,order\_,x\_} := \sum_{r=2}^{order-1} x_{order-r}D_{i-1,r,x};$
$HD_{order\_} := \text{If}[m == 0, function/.u \to u_0,$

$\sum_{n=1}^{order}\left((\mathrm{D}[function,\{u,n\}]/.u\to u_0)/n!*D_{n,order,u}\right]);$
$\mathrm{Print}[''A''_m,''('',function,'')='',\mathrm{Expand}[\mathrm{TraditionalForm}[HD_m]]]];$

Code. 4(A sample Mathematica program for $A_m(f(u))$ by using the (3) & Corollary 2)

AdomianPolynomial1$[function_-,m_-]:=\mathrm{Module}[\{\},g[u_-]:=function;$
$A[m1_-]:=\frac{1}{\mathrm{Factorial}[m1]}(\mathrm{D}[g[u]/.u\to\mathrm{Sum}[u_n*p^n,\{n,0,m1\}],\{p,m1\}]/.p\to 0);$
$\mathrm{Print}[''A''_m,''('',function,'')='',\mathrm{Expand}[\mathrm{TraditionalForm}[A[m]]]];$

ADMforIDEs(A sample Mathematica program of ADM for solving IDEs given by (29))

ADMforIDEs$[functionF_-,functiong_-,kernel_-,a_-,\beta_-,\lambda_-,ProblemOrder_-,SolutionOrder_-]$
$:=\mathrm{Module}[\{\},F[u_-]:=functionF;g[x_-]:=functiong;k[x_-,t_-]:=kernel;$
$(*----$**Definition of the linear operator**$----------*)$
$L[f_-]:=\mathrm{Module}[\{\},\mathrm{Expand}[\sum_{i=0}^{ProblemOrder}(Lcoefficients[[i+1]]*\mathrm{D}[f,\{x,i\}])]];$
$(*----$**Definition of the integral inverse of linear operator**$-------*)$
$Linverse[f_-]:=\mathrm{Module}[\{Linv,u,solution\},Linv=\mathrm{DSolve}[L[u[x]]==f,u[x],x];$
$solution=Linv[[1,1,2]]/.C[\_]\to 0;\mathrm{Expand}[solution]];$
$Linverse[p\_Plus]:=\mathrm{Map}[Linverse,p];Linverse[c_-*f_-]:=c*Linverse[f]/;\mathrm{FreeQ}[c,x];$
$(*----$**Definition of the Adomian polynomials**$----------*)$
$HDN[horder_-]:=\mathrm{Module}[\{\},D_{1,j\_}:=\mathrm{If}[j>0,u[j],0];D_{i\_,0}:=0;$
$D_{2,j\_}:=\mathrm{If}[j>0,\sum_{r=1}^{j-1}u[j-r]*u[r]];D_{i\_,m\_}:=\sum_{r=2}^{m-1}u[m-r]*D_{i-1,r};$
$DH_{m\_,function\_}:=\mathrm{If}[horder==0,function/.u\to u[0],$
$\mathrm{Expand}[\sum_{n=1}^{m}((\mathrm{D}[function,\{u,n\}]/.u\to u[0])/(n!)*D_{n,m})]]];$
$(*----$**Definition of the initial guess**$----------*)$
$ugStar=\mathrm{DSolve}[L[u[x]]==0,conditions[u]==0,u[x],x];$
$u[0]=\ \mathrm{First}[u[x]/.ugStar];$
$(*----$**Definition of the $\mathbf{u_g}$**$----------*)$
$ug=\mathrm{DSolve}[L[u[x]]==0,u[x],x];$
$ug1=\mathrm{First}[u[x]/.ug];$
$(*----$**Some needed commands**$----------*)$
$X[m_-]:=\mathrm{If}[m\le 1,0,1];C1=\mathrm{Arrary}[C,ProblemOrder];w[x\_]:=ug1;$
$\mathrm{Which}[ProblemOrder==1,constants[u\_]:=\mathrm{First}[conditions[u]]/.\_[\_]->0,$
$ProblemOrder>1,constants[u\_]:=conditions[u]/.\_[\_]->0];$
$f11=constants[u1]-conditions[w];$
$(*----$**Main block**$----------*)$
$\mathrm{For}[m=1,m<SolutionOrder,m++,HDN[m-1];$
$\{A[m-1]=\sum_{i=0}^{ProblemOrder}(Problemcoefficients[[i+1]]-Lcoefficients[[i+1]])$
$*\mathrm{D}[u[m-1],\{x,i\}]+\lambda*\mathrm{Integrate}[k[x,t]*((DH_{m-1,F[u]})/.x\to t),\{t,a,\beta\}],$
$uParticular=Linverse[-A[m-1]+(1-X[m])*g[x]],w1[x\_]:=uParticular,$
$f12=conditions[w1]-constants[u1],ug2=\mathrm{Solve}[\{f11==f12\},C1],$
$uParticularStar=\mathrm{First}[ug1//.ug2],$

$u[m] = uParticular + uParticularStar,$
$\text{Print}[''u'',_m\ ,'' ='', \text{TraditionalForm}[\text{Collect}[u[m], x]]]\}]];$

References

1. Adomian, G. *Nonlinear Stochastic Systems Theory and Applications to Physics*, Kluwer Academic, Dordrecht, 1989.
2. Adomian, G. *Solving Frontier Problems of Physics: The Decomposition Method*, Kluwer Academic, Dordrecht, 1994.
3. Adomian, G. and Rach, R. *On composite nonlinearities and the decomposition method*, J. Math. Anal. Appl. 113 (1986) 504-509.
4. Cherruault, Y., Adomian, G., Abbaoui, K. and Rach R. *Further remarks on convergence of decomposition method*, International Journal of Bio-Medical Computing 38 (1995) 89-93.
5. Duan, J., Rach, R., Baleanu, D. and Wazwaz, A. *A review of the Adomian decomposition method and its applications to fractional differential equations*, Commun. Frac. Calc. 3 (2) (2012) 73-99.
6. Gabet, L. *The Theoretical Foundation of the Adomian Method*, Computers Math. Applic. 27 (1994) 41-52.
7. Ghorbani, A. *Beyond Adomian polynomails: He polynomails*, Chaos, Soliton and Fractals 39 (2009) 1486-1492.
8. Liao, S. J. *Notes on the homotopy analysis method: Some definitions and theorems*, Commun Nonlinear Sci Numer Simul 14 (2009) 983–997.
9. Molabahrami, A. *Integral mean value method for solving a general nonlinear Fredholm integro-differential equation under the mixed conditions*, Communications in Numerical Analysis 2013 (2013) 1-15. http://dx.doi.org/10.5899/2013/cna-00146.
10. Molabahrami, A. and Khani, F. *The homotopy analysis method to solve the Burgers-Huxley equation*, Nonlinear Anal-Real 10 (2009) 589-600.
11. Ngarhasta, N., Some, B., Abbaoui, K. and Cherruault, Y. *New numerical study of adomian method applied to a diffusion model*, Kybernetes 31 (2002) 61-75.
12. Rach, R. *A convenient computational form for the Adomian polynomials*, J. Math. Anal. Appl. 102 (1984) 415-419.
13. Rach, R. *A new definition of the Adomian polynomials*, Kybernetes 37 (2008) 910-955.

مروری عملی از روش تجزیه آدومیان: جنبه های پیاده سازی کامپیوتری

احمد ملابهرامی

دانشگاه ایلام، گروه ریاضی

چکیده : در این مقاله، مروی عملی از روش تجزیه آدومیان، جهت توسعه آن برای بررسی مسایل غیرخطی قوی تحت شرایط آمیخته، ارایه و همگرایی الگوریتم به صورت دقیق اثبات می شود. برای دست یابی به این هدف، یک روش جدید و ساده برای تولید چندجمله ای های آدومیان، برای فرم کلی یک تابع غیرخطی، ارایه می گردد. روش ارایه شده، یک فرمول صریح برای محاسبه چندجمله ای های آدومیان یک تابع غیرخطی فراهم می کند. کارایی رهیافت با به کارگیری آن روی چندین مسئله ی جالب انتگرال- دیفرانسیل نشان داده خواهد شد. برنامه های متمتیکای تولید چندجمله ای ها و جواب های آدومیان بر اساس رویه های این مقاله طرح یزی شده است.

کلمات کلیدی : روش تجزیه آدومیان؛ چندجمله ای های آدومیان؛ مسایل غیرخطی انتگرال- دیفرانسیل؛ جواب سری؛ مسایل غیرخطی قوی؛ محاسبات و برنامه های صریح ماشینی.