

# A Simple Algorithm for Calculating Adomian Polynomials

J. Biazar <sup>1</sup> and S. M. Shafiof

Department of Mathematics, Faculty of Science  
University of Guilan, P.O. Box 1914, Rasht, Iran  
biazar@guilan.ac.ir, shafiof@yahoo.com

## Abstract

In this paper, we introduce a new Algorithm for calculating Adomian polynomials and present some example to show the simplicity of the new Algorithm. The new Algorithm requires less computations in comparison with the previous methods, and can be extended to calculate Adomian polynomials for nonlinear functional with several variables.

**Keywords:** Adomian decomposition method, Adomian polynomials, Non-linearity of several variables

## 1 Introduction

The Adomian decomposition method[1-2] is a technique for solving functional equations in the form:

$$u = f + G(u) \quad (1)$$

In some functional space, say F. The solution  $u$  is considered as the summation of a series, say;

$$u = \sum_{n=0}^{\infty} u_n \quad (2)$$

and  $G(u)$  as the summation of a series, say;

$$G(u) = \sum_{n=0}^{\infty} A_n(u_0, u_1, \dots, u_n) \quad (3)$$

Where  $A_n$  's, called Adomian polynomials, has been introduced by the Adomian himself by the formula

$$A_n(u_0, u_1, \dots, u_n) = \frac{1}{n!} \frac{d^n}{d\lambda^n} [G(\sum_{i=0}^{\infty} u_i \lambda_i)]_{\lambda=0} \quad (4)$$

---

<sup>1</sup>Corresponding author

Other authors have suggested different Algorithm for computing Adomian polynomials [3-6]. Using the Algorithm presented in Ref.[3], needs classification of the terms  $u_i$ ,  $i = 0, 1, 2, \dots, n$ , which is very complicate for large  $n$  's. Applying the Algorithm presented in [4], which uses Taylor expansion of the functional  $G(u)$ , will be very complicated especially when the unknown function  $u$ , appears in the denominator. Calculation of  $n$ 'th Adomian polynomials using the technique in Ref.[5], requires to computing the  $n$ 'th order derivative, that will be complicated for large values of  $n$ . In spite of difficulties in applying the used method in [6], it could not be applied for functionals with several variables. Here we suggest a new Algorithm, which seems to be more easier than other methods introduced so far. This Algorithm can also be extended for calculating Adomian polynomials in several variables. Some examples, will illustrate how easy is the new Algorithm for implementation .

## 2 The new Algorithm

The following is the Algorithm for calculating  $A_0, A_1, \dots, A_n$ :

Step 1: Input nonlinear term  $G(u)$  and  $n$ , the number of Adomian polynomials needed.

Step 2: Set  $A_0 = G(u_0)$

Step 3: For  $k = 0$  to  $n - 1$  do

$$A_k(u_0, \dots, u_k) := A_k(u_0 + u_1\lambda, \dots, u_k + (k + 1)u_{k+1}\lambda)$$

$$\{in \quad A_k : u_i \rightarrow u_i + (i + 1)u_{i+1}\lambda \quad for \quad i = 0, 1, \dots, k\}$$

Step 4: Taking the first order derivative of  $A_k$ , with respect to  $\lambda$ , and then let  $\lambda = 0$  :

$$\frac{d}{d\lambda} A_k|_{\lambda=0} = (k + 1)A_{k+1}.$$

End do

Step 5: Output  $A_0, A_1, \dots, A_n$  .

According to the above Algorithm, Adomian polynomials will be computed as follows:

$$A_0 = G(u_0),$$

$$A_1 = \frac{d}{d\lambda}G(u_0 + u_1\lambda)|_{\lambda=0} = u_1G'(u_0),$$

$$A_2 = \frac{1}{2}\frac{d}{d\lambda}((u_1 + 2u_2\lambda)G'(u_0 + u_1\lambda))|_{\lambda=0} = u_2G'(u_0) + \frac{u_1^2}{2!}G''(u_0),$$

$$A_3 = \frac{1}{3}\frac{d}{d\lambda}((u_2 + 3u_3\lambda)G'(u_0 + u_1\lambda) + \frac{1}{2!}(u_1 + 2u_2\lambda)^2G''(u_0 + u_1\lambda))|_{\lambda=0} = u_3G'(u_0) + u_1u_2G''(u_0) + \frac{u_1^3}{3!}G'''(u_0),$$

$$A_4 = \frac{1}{4}\frac{d}{d\lambda}((u_3 + 4u_4\lambda)G'(u_0 + u_1\lambda) + (u_1 + 2u_2\lambda)(u_2 + 3u_3\lambda)G''(u_0 + u_1\lambda) + \frac{1}{3!}(u_1 + 2u_2\lambda)^3G'''(u_0 + u_1\lambda))|_{\lambda=0} = u_4G'(u_0) + (u_1u_3 + \frac{u_2^2}{2})G''(u_0) + \frac{u_1^2u_2}{2}G'''(u_0) + \frac{u_1^4}{4!}G^4(u_0),$$

Continuing this course, we can get the other Adomian polynomials .

### 3 Application of the new Algorithm

Here we apply the above Algorithm to some nonlinear operators functionals.

**Example 1:**(The case of nonlinear polynomials)

$$G(u) = u^3 + u^4$$

Now by applying the new Algorithm we get

$$A_0 = u_0^3 + u_0^4,$$

$$A_1 = \frac{d}{d\lambda}((u_0 + u_1\lambda)^3 + (u_0 + u_1\lambda)^4)|_{\lambda=0} = 3u_1u_0^2 + 4u_1u_0^3,$$

$$A_2 = \frac{1}{2}\frac{d}{d\lambda}(3(u_1 + 2u_2\lambda)(u_0 + u_1\lambda)^2 + 4(u_1 + 2u_2\lambda)(u_0 + u_1\lambda)^3)|_{\lambda=0} = 3u_2u_0^2 + 3u_0u_1^2 + 4u_2u_0^3 + 6u_0^2u_1^2,$$

$$A_3 = \frac{1}{3}\frac{d}{d\lambda}(3(u_2 + 3u_3\lambda)(u_0 + u_1\lambda)^2 + 3(u_0 + u_1\lambda)(u_1 + 2u_2\lambda)^2 + 4(u_2 + 3u_3\lambda)(u_0 + u_1\lambda)^3 + 6(u_0 + u_1\lambda)^2(u_1 + 2u_2\lambda)^2)|_{\lambda=0} = 3u_3u_0^2 + 6u_0u_1u_2 + 4u_3u_0^3 + 12u_0^2u_1u_2 + 4u_0u_1^3 + u_1^3,$$

While by corresponding method presented in [3], one must calculate  $(u_0 + u_1 + u_2 + u_3 + \dots)^3$ ,  $(u_0 + u_1 + u_2 + u_3 + \dots)^4$  and reorder the terms in index, to obtain Adomian polynomials.

**Example 2:**( Case of nonlinear derivatives)

Consider  $G(u) = u^2u_x$ , According to the new Algorithm :

$$A_0 = G(u_0) = u_0^2u_{0x},$$

$$A_1 = \frac{d}{d\lambda}((u_0 + u_1\lambda)^2 + (u_{0x} + u_{1x}\lambda))|_{\lambda=0} = 2u_0u_1u_{0x} + u_0^2u_{1x},$$

$$A_2 = \frac{1}{2}\frac{d}{d\lambda}(2(u_0 + u_1\lambda)(u_1 + 2u_2\lambda)(u_{0x} + u_{1x}\lambda) + (u_0 + u_1\lambda)^2(u_{1x} + 2u_{2x}\lambda))|_{\lambda=0} = u_1^2u_{0x} + 2u_0u_2u_{0x} + 2u_0u_1u_{1x} + u_0^2u_{2x},$$

$$A_3 = \frac{1}{3}\frac{d}{d\lambda}((u_1 + 2u_2\lambda)^2(u_{0x} + u_{1x}\lambda) + 2(u_0 + u_1\lambda)(u_{0x} + u_{1x}\lambda)(u_2 + 3u_3\lambda) + 2(u_0 + u_1\lambda)(u_1 + 2u_2\lambda)(u_{1x} + 2u_{2x}\lambda) + (u_0 + u_1\lambda)^2(u_{2x} + 3u_{3x}\lambda))|_{\lambda=0} = 2u_1u_2u_{0x} + 2u_0u_3u_{0x} + 2u_0u_2u_{1x} + 2u_0u_1u_{2x} + u_1^2u_{1x} + u_0^2u_{3x},$$

$$A_4 = u_1^2u_{2x} + 2u_1u_2u_{1x} + 2u_1u_3u_{0x} + 2u_0u_4u_{0x} + 2u_0u_3u_{1x} + 2u_0u_1u_{3x} + 2u_0u_2u_{2x} + u_2^2u_{0x} + u_0^2u_{4x},$$

**Example 3:**( Case of hyperbolic and trigonometric nonlinearity)

$$G(u) = \sinh u + \sin^2 u \cos^2 u$$

By using the new Algorithm,the Adomian polynomials for G(u) are given as:

$$A_0 = G(u_0) = \sinh u_0 + \sin^2 u_0 \cos^2 u_0,$$

$$A_1 = \frac{d}{d\lambda}(\sinh(u_0 + u_1\lambda) + \sin^2(u_0 + u_1\lambda)\cos^2(u_0 + u_1\lambda))|_{\lambda=0} = u_1 \cosh u_0 + 2u_1 \sin u_0 \cos^3 u_0 - 2u_1 \sin^3 u_0 \cos u_0,$$

$$A_2 = \frac{1}{2}\frac{d}{d\lambda}((u_1 + 2u_2\lambda) \cosh(u_0 + u_1\lambda) + 2(u_1 + 2u_2\lambda) \sin(u_0 + u_1\lambda) \cos^3(u_0 + u_1\lambda) - 2(u_1 + 2u_2\lambda) \sin^3(u_0 + u_1\lambda) \cos(u_0 + u_1\lambda))|_{\lambda=0} = u_2 \cosh u_0 + \frac{u_2^2}{2!} \sinh u_0 + u_1^2 \cos^4 u_0 - 6u_1^2 \sin^2 u_0 \cos^2 u_0 + 2u_2 \sin u_0 \cos^3 u_0 + u_1^2 \sin^4 u_0 - 2u_2 \sin^3 u_0 \cos u_0,$$

$$A_3 = \frac{1}{3}\frac{d}{d\lambda}((u_2 + 3u_3\lambda) \cosh(u_0 + u_1\lambda) + \frac{(u_1 + 2u_2\lambda)^2}{2!} \sinh(u_0 + u_1\lambda) + (u_1 + 2u_2\lambda)^2 \cos^4(u_0 + u_1\lambda) - 6(u_1 + 2u_2\lambda)^2 \sin^2(u_0 + u_1\lambda) \cos^2(u_0 + u_1\lambda) + 2(u_2 + 3u_3\lambda) \sin(u_0 + u_1\lambda) \cos^3(u_0 + u_1\lambda) + (u_1 + 2u_2\lambda)^2 \sin^4(u_0 + u_1\lambda) - 2(u_2 + 3u_3\lambda) \sin^3(u_0 + u_1\lambda) \cos(u_0 + u_1\lambda))|_{\lambda=0} =$$

$$u_3 \cosh u_0 + u_1u_2 \sinh u_0 + \frac{u_1^3}{3!} \cosh u_0 - \frac{16}{3}u_1^3 \cos^3 u_0 \sin u_0 + 2u_1u_2 \cos^4 u_0 + \frac{16}{3}u_1^3 \sin^3 u_0 \cos u_0 - 12u_1u_2 \sin^2 u_0 \cos^2 u_0 + 2u_3 \sin u_0 \cos^3 u_0 + 2u_1u_2 \sin^4 u_0 - 2u_3 \sin^3 u_0 \cos u_0,$$

**Example 4:**(Case of exponential and logarithmic nonlinearity)

$$G(u) = e^u + \ln u$$

By using the new Algorithm, for calculating the Adomian polynomials for  $G(u) = e^u + \ln u$ , we have:

$$\begin{aligned}
 A_0 &= G(u_0) = e^{u_0} + \ln u_0, \\
 A_1 &= \frac{d}{d\lambda}(e^{(u_0+u_1\lambda)} + \ln(u_0 + u_1\lambda))|_{\lambda=0} = u_1 e^{u_0} + \frac{u_1}{u_0}, \\
 A_2 &= \frac{1}{2} \frac{d}{d\lambda}((u_1 + 2u_2\lambda)e^{(u_0+u_1\lambda)} + \frac{(u_1+2u_2\lambda)}{(u_0+u_1\lambda)})|_{\lambda=0} = u_2 e^{u_0} + \frac{1}{2} u_1^2 e^{u_0} + \frac{u_2}{u_0} - \frac{1}{2} \frac{u_1^2}{u_0^2}, \\
 A_3 &= \frac{1}{3} \frac{d}{d\lambda}((u_2+3u_3\lambda)e^{(u_0+u_1\lambda)} + \frac{1}{2}(u_1+2u_2\lambda)^2 e^{(u_0+u_1\lambda)} + \frac{(u_2+3u_3\lambda)}{(u_0+u_1\lambda)} - \frac{1}{2} \frac{(u_1+2u_2\lambda)^2}{(u_0+u_1\lambda)^2})|_{\lambda=0} = \\
 &u_3 e^{u_0} + u_1 u_2 e^{u_0} + \frac{1}{3!} u_1^3 e^{u_0} + \frac{u_3}{u_0} - \frac{u_1 u_2}{u_0^2} + \frac{1}{3} \frac{u_1^3}{u_0^3},
 \end{aligned}$$

**4. Extension of the new Algorithm for calculating Adomian polynomials in several variables:**

Now, this new approach for computing Adomian polynomials can be applied to functionals with several variables. Let we are concerned with a system of nonlinear equations  $u_i = f_i + N_i \quad i = 1, 2, \dots, p$  Where  $N_i(u_1, u_2, \dots, u_p)$  are nonlinear operators and  $u_i, N_i$  are represented as the summation of the following series:

$$\begin{aligned}
 u_i &= \sum_{j=0}^{\infty} u_{ij} \lambda^j \\
 N_i(u_1, u_2, \dots, u_p) &= \sum_{j=0}^{\infty} A_{ij}(u_{10}, \dots, u_{1j}, u_{20}, \dots, u_{2j}, \dots, u_{p0}, \dots, u_{pj}) \lambda^j,
 \end{aligned}$$

The following Algorithm calculates Adomian polynomials  $A_{i0}, A_{i1}, \dots, A_{in}$ :

For  $i = 1, 2, \dots, p$  do

Step 1: Input nonlinear term  $N_i(u_1, u_2, \dots, u_p)$  and  $n$ , the number of Adomian polynomials needed.

Step 2: Set  $A_{i0} = N_i(u_{10}, u_{20}, \dots, u_{p0})$

Step 3: For  $k = 0$  to  $n - 1$  do

$$\begin{aligned}
 &A_{ik}(u_{10}, \dots, u_{1k}, \dots, u_{p0}, \dots, u_{pk}) := \\
 &A_{ik}(u_{10} + u_{11}\lambda, \dots, u_{1k} + (k + 1)u_{1k+1}\lambda, \dots, u_{p0} + u_{p1}\lambda, \dots, u_{pk} + (k + 1)u_{pk+1}\lambda) \\
 &\quad \{ \text{in } A_{ik}: u_{ij} \rightarrow u_{ij} + (j + 1)u_{ij+1}\lambda \quad \text{for } j = 0, 1, \dots, k \}
 \end{aligned}$$

Step 4: By taking, first order derivative of  $A_{ik}$ , with respect to  $\lambda$ , and let  $\lambda = 0$ , we have :

$$\frac{d}{d\lambda} A_{ik}|_{\lambda=0} = (k + 1)A_{ik+1}.$$

End do

End do

Step 5: Output  $A_{i0}, A_{i1}, \dots, A_{in}$  .

#### 4.1 Two variables (p=2):

By using the new Algorithm, the Adomian polynomials for the following system of two functional equations can be computed:

$$\begin{cases} u_1 = f_1 + N_1 \\ u_2 = f_2 + N_2 \end{cases}$$

$$A_{i0} = N_i(u_{10}, u_{20}),$$

$$A_{i1} = \frac{d}{d\lambda}(N_i(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda))|_{\lambda=0} = u_{11} \frac{\partial N_i}{\partial u_1}(u_{10}, u_{20}) + u_{21} \frac{\partial N_i}{\partial u_2}(u_{10}, u_{20}),$$

$$\begin{aligned} A_{i2} = & \frac{1}{2} \frac{d}{d\lambda}((u_{11} + 2u_{12}\lambda) \frac{\partial N_i}{\partial u_1}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda) + (u_{21} + 2u_{22}\lambda) \\ & \frac{\partial N_i}{\partial u_2}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda))|_{\lambda=0} = u_{12} \frac{\partial N_i}{\partial u_1}(u_{10}, u_{20}) + u_{22} \frac{\partial N_i}{\partial u_2}(u_{10}, u_{20}) + \\ & \frac{1}{2!} u_{11}^2 \frac{\partial^2 N_i}{\partial u_1^2}(u_{10}, u_{20}) + \frac{1}{2!} u_{21}^2 \frac{\partial^2 N_i}{\partial u_2^2}(u_{10}, u_{20}) + u_{11} u_{21} \frac{\partial^2 N_i}{\partial u_1 \partial u_2}(u_{10}, u_{20}), \end{aligned}$$

$$\begin{aligned} A_{i3} = & \frac{1}{3} \frac{d}{d\lambda}((u_{12} + 3u_{13}\lambda) \frac{\partial N_i}{\partial u_1}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda) + (u_{22} + 3u_{23}\lambda) \\ & \frac{\partial N_i}{\partial u_2}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda) + \frac{1}{2!} (u_{11} + 2u_{12}\lambda)^2 \frac{\partial^2 N_i}{\partial u_1^2}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda) + \\ & \frac{1}{2!} (u_{21} + 2u_{22}\lambda)^2 \frac{\partial^2 N_i}{\partial u_2^2}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda) + (u_{11} + 2u_{12}\lambda)(u_{21} + 2u_{22}\lambda) \\ & \frac{\partial^2 N_i}{\partial u_1 \partial u_2}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda))|_{\lambda=0} = \\ & u_{13} \frac{\partial N_i}{\partial u_1}(u_{10}, u_{20}) + u_{23} \frac{\partial N_i}{\partial u_2}(u_{10}, u_{20}) + u_{11} u_{12} \frac{\partial^2 N_i}{\partial u_1^2}(u_{10}, u_{20}) + \\ & u_{21} u_{22} \frac{\partial^2 N_i}{\partial u_2^2}(u_{10}, u_{20}) + u_{12} u_{21} \frac{\partial^2 N_i}{\partial u_1 \partial u_2}(u_{10}, u_{20}) + u_{11} u_{22} \frac{\partial^2 N_i}{\partial u_2 \partial u_1}(u_{10}, u_{20}) + \\ & \frac{1}{2!} u_{11}^2 u_{21} \frac{\partial^2 N_i}{\partial u_1^2 \partial u_2}(u_{10}, u_{20}) + \frac{1}{2!} u_{21}^2 u_{11} \frac{\partial^2 N_i}{\partial u_1 \partial u_2^2}(u_{10}, u_{20}) + \frac{1}{3!} u_{11}^3 \frac{\partial^3 N_i}{\partial u_1^3}(u_{10}, u_{20}) + \\ & \frac{1}{3!} u_{21}^3 \frac{\partial^3 N_i}{\partial u_2^3}(u_{10}, u_{20}), \end{aligned}$$

To illustrate the application of the method for two variables, here an example is presented .

**Example 6:** Let  $N(u, v) = uvv_x$  , by applying the new Algorithm we get

$$A_0 = N(u_0, v_0) = u_0 v_0 v_{0x},$$

$$A_1 = \frac{d}{d\lambda}((u_0 + u_1\lambda)(v_0 + v_1\lambda)(v_{0x} + v_{1x}\lambda))|_{\lambda=0} = u_1 v_0 v_{0x} + u_0 v_1 v_{0x} + u_0 v_0 v_{1x},$$

$$\begin{aligned} A_2 = & \frac{1}{2} \frac{d}{d\lambda}((u_1 + 2u_2\lambda)(v_0 + v_1\lambda)(v_{0x} + v_{1x}\lambda) + (u_0 + u_1\lambda)(v_1 + 2v_2\lambda)(v_{0x} + v_{1x}\lambda) + \\ & (u_0 + u_1\lambda)(v_0 + v_1\lambda)(v_{1x} + 2v_{2x}\lambda))|_{\lambda=0} = u_2 v_0 v_{0x} + u_1 v_1 v_{0x} + u_0 v_2 v_{0x} + \\ & u_1 v_0 v_{1x} + u_0 v_1 v_{1x} + u_0 v_0 v_{2x}, \end{aligned}$$

$$\begin{aligned} A_3 = & \frac{1}{3} \frac{d}{d\lambda}((u_2 + 3u_3\lambda)(v_0 + v_1\lambda)(v_{0x} + v_{1x}\lambda) + (u_1 + 2u_2\lambda)(v_1 + 2v_2\lambda)(v_{0x} + v_{1x}\lambda) + \\ & (u_0 + u_1\lambda)(v_2 + 3v_3\lambda)(v_{0x} + v_{1x}\lambda) + (u_1 + 2u_2\lambda)(v_0 + v_1\lambda)(v_{1x} + 2v_{2x}\lambda) + \end{aligned}$$

$$(u_0 + u_1\lambda)(v_1 + 2v_2\lambda)(v_{1x} + 2v_{2x}\lambda) + (u_0 + u_1\lambda)(v_0 + v_1\lambda)(v_{2x} + 3v_{3x}\lambda)|_{\lambda=0} = u_3v_0v_{0x} + u_2v_1v_{0x} + u_0v_3v_{0x} + u_2v_0v_{1x} + u_1v_1v_{1x} + u_0v_2v_{1x} + u_1v_0v_{2x} + u_0v_1v_{2x} + u_0v_0v_{3x} + u_1v_2v_{0x},$$

#### 4.2 Three variables (p=3):

Using the new Algorithm, the Adomian polynomials for  $N_i(u_1, u_2, u_3)$  Where  $u_1, u_2, u_3, N_i$  are represented by series:

$$u_i = \sum_{j=0}^{\infty} u_{ij}\lambda^j \quad \text{for } i = 1, 2, 3,$$

$$N_i = \sum_{j=0}^{\infty} A_{ij}(u_{10}, \dots, u_{1j}, u_{20}, \dots, u_{2j}, u_{30}, \dots, u_{3j})\lambda^j$$

are thus given by

$$A_{i0} = N_i(u_{10}, u_{20}, u_{30}),$$

$$A_{i1} = \frac{d}{d\lambda}(N_i(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda, u_{30} + u_{31}\lambda))|_{\lambda=0} = u_{11}\frac{\partial N_i}{\partial u_1}(u_{10}, u_{20}, u_{30}) + u_{21}\frac{\partial N_i}{\partial u_2}(u_{10}, u_{20}, u_{30}) + u_{31}\frac{\partial N_i}{\partial u_3}(u_{10}, u_{20}, u_{30}),$$

$$A_{i2} = \frac{1}{2}\frac{d}{d\lambda}((u_{11} + 2u_{12}\lambda)\frac{\partial N_i}{\partial u_1}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda, u_{30} + u_{31}\lambda) + (u_{21} + 2u_{22}\lambda)\frac{\partial N_i}{\partial u_2}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda, u_{30} + u_{31}\lambda) + (u_{31} + 2u_{32}\lambda)\frac{\partial N_i}{\partial u_3}(u_{10} + u_{11}\lambda, u_{20} + u_{21}\lambda, u_{30} + u_{31}\lambda))|_{\lambda=0} = u_{12}\frac{\partial N_i}{\partial u_1}(u_{10}, u_{20}, u_{30}) + u_{22}\frac{\partial N_i}{\partial u_2}(u_{10}, u_{20}, u_{30}) + u_{32}\frac{\partial N_i}{\partial u_3}(u_{10}, u_{20}, u_{30}) + \frac{1}{2!}u_{11}^2\frac{\partial^2 N_i}{\partial u_1^2}(u_{10}, u_{20}, u_{30}) + \frac{1}{2!}u_{21}^2\frac{\partial^2 N_i}{\partial u_2^2}(u_{10}, u_{20}, u_{30}) + \frac{1}{2!}u_{31}^2\frac{\partial^2 N_i}{\partial u_3^2}(u_{10}, u_{20}, u_{30}) + \frac{1}{2!}u_{11}u_{21}\frac{\partial^2 N_i}{\partial u_1\partial u_2}(u_{10}, u_{20}, u_{30}) + \frac{1}{2!}u_{11}u_{31}\frac{\partial^2 N_i}{\partial u_1\partial u_3}(u_{10}, u_{20}, u_{30}) + \frac{1}{2!}u_{21}u_{31}\frac{\partial^2 N_i}{\partial u_2\partial u_3}(u_{10}, u_{20}, u_{30}),$$

Continue this course, we can get the Adomian polynomials for any p-variables functional operator.

## 4 Discussion

The Adomian decomposition method is a powerful device for solving a large class of nonlinear problems in sciences. The main part of this method is calculating Adomian polynomials for nonlinear polynomials. In this paper, we introduced a new Algorithm for calculating Adomian polynomials. The main specification of this Algorithm is no requirement to complicated calculations.

This Algorithm is very easy to implement, because it uses first order derivative, once. In each step to obtain the Adomian polynomials and will be extended to calculate Adomian polynomials for nonlinear functionals of several variables.

## 5 References

- [1] G. Adomian, G.E. Adomian, A global method for solution of complex systems, *Math. Model.* 5 (1984) 521-568.
- [2] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method*, Kluwer Academic Publishers, Dordrecht, 1994.
- [3] A. M. Wazwaz, A new algorithm for calculating Adomian polynomials for non-linear Operators, *Appl. Math. Comput.* 111 (2000) 53-69.
- [4] J. Biazar, E. Babolian, A. Nouri, R. Islam, An alternate algorithm for computing Adomian polynomials in special cases, *App. Math. Comp.* 138 (2003) 523-529.
- [5] Yonggui Zhu, Qianshun Chang, Shengchang Wu, A new algorithm for calculating Adomian polynomials *Appl. Math. Comput.* 169 (2005) 402-416.
- [6] E. Babolian, Sh. Javadi, New method for calculating Adomian polynomials, *Appl. Math. Comput.* 153 (2004) 253-259.

**Received: November 18, 2006**