British Journal of Mathematics & Computer Science
5(5): 562-570, 2015, Article no.BJMCS.2015.041
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Symbolic Computation of Adomian Polynomials Based on Rach's Rule

M. Kaliyappan[1*] and S. Hariharan[1]

[1]*Division of Mathematics, VIT University, Chennai, India.*

Article Information

DOI: 10.9734/BJMCS/2015/14067
Editor(s):
(1) Kai-Long Hsiao, Taiwan Shoufu University, Taiwan.
Reviewers:
(1) Anonymous, Celal Bayar University, Turkey.
(2) Anonymous, Shanghai Institute of Technology, China.
(3) Anonymous, National University of Computers and Emerging Sciences Peshawar Campus, Pakistan.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=729&id=6&aid=6999

Original Research Article

Received: 16 September 2014
Accepted: 31 October 2014
Published: 18 November 2014

Abstract

This paper presents a simple way of computing Adomian polynomials by applying the decomposition of positive integers as a subscript of the variable 'u' for nonlinear terms much similarly motivated by Abbaoui et al. through the use of MATLAB software. The proposed MATLAB program exploits symbolic programming for generating Adomian polynomials.

Keywords: Adomian decomposition method, MATLAB, nonlinear terms, Adomian polynomials.

1 Introduction

Adomian formally introduced formulas for generating Adomian polynomials for all forms of nonlinearity [1-3]. Several authors have been focused in this area to develop a practical method for the calculation of Adomian polynomials [4-8]. R.Rach have presented a convenient computational form to obtain Adomian polynomials [9]. Wenhai Chen et al. [10] have developed an algorithm which is easily programmable in MAPLE, to calculate Adomian polynomials for nonlinear terms in the differential equations. Duan have developed analytic recurrence algorithm for obtaining Adomian Polynomials and an efficient algorithm for getting multivariable Adomian polynomial and the MATHEMATICA subroutines for the algorithms [11-13]. J. Biazar and S. M. Shafiof [14] developed a simple algorithm for generating Adomian polynomials.The theoretical basis of the Adomian decomposition method was discussed in detail by L. Gabet [15].

*Corresponding author: kaliyappan.m@vit.ac.in;

Hooman Fatoorhchi et al. [16] and H.W. Choi et al. [17] have developed MATLAB and MATHEMATICA subroutines for computing Adomian polynomials based on the technique proposed by A.M. Wazwaz respectively.

Abbaoui et al. [18] have presented a computational procedure for obtaining Adomian polynomials through all possible non negative integer solutions of the equations

$$\alpha_1 + \alpha_2 + ... + \alpha_n = n \text{, with } \alpha_1 \geq \alpha_2 \geq ... \geq \alpha_n \qquad \text{…………..(1)}$$

In this paper we have obtained all possible non - negative integer solutions of the equation (1) by decomposition of positive integer 'n' as an array and used this array elements as subscripts of u's to obtain Adomian polynomial as given by G. Adomian [19].

Organization of this paper is as follows, Section 2 represents the basic procedure for generating Adomian polynomials. Decomposition of positive integers for generating Adomian polynomials was explained in detail in Section 3. Section 4 presents the MATLAB code for generating Adomian polynomials `through symbolic computation and provides Adomian polynomials for few nonlinear terms. The computation time for generating Adomian polynomials have also been presented in Section 4.

2 Adomian Polynomials

In the Adomian decomposition method, the solution of a differential equation is written as $u = \sum_{n=0}^{\infty} u_n$ and $f(u) = \sum_{n=0}^{\infty} A_n(u_0, u_{1,...,} u_n)$, where u_0 is a function involving initial or boundary conditions, the forcing function and an integral operator.

This leads to the assumption that the solutions and functions of the solutions are expanded in the A_n polynomials, because $\sum_{n=0}^{\infty} A_n$ reduces to $\sum_{n=0}^{\infty} u_n$ for $f(u) = u$ and $f(u) = \sum_{n=0}^{\infty} A_n$.

The expression $A_n = \frac{1}{n!}\left[\frac{d^n}{d\lambda^n} f(u(\lambda))\right]_{\lambda=0}$ given by Adomian for any positive integer 'n' can be generated using integer partition of 'n'.

A convenient way to generate A_n presented by Adomian (Rach's rule) in his book on the decomposition method is given as

$$A_n = \sum_{v=1}^{n} c(v,n) f^{(v)}(u_0) \qquad (2)$$

Where $c(v,n)$ are products (or sum of products) of v components of u whose subscripts sum up to n, with the result divided by the factorial of the number of repeated subscripts. Also, a reference list of the Adomian polynomials for f(u)presented by Adomian is shown below

$$A_0 = f(u_0)$$
$$A_1 = u_1 f'^{(1)}(u_0)$$
$$A_2 = u_2 f'^{(1)}(u_0) + \frac{1}{2!}u_1^2 f^{(2)}(u_0)$$
$$A_3 = u_3 f'^{(1)}(u_0) + u_1 u_2 f^{(2)}(u_0) + \frac{1}{3!}u_1^3 f^{(3)}(u_0)$$
$$A_4 = u_4 f'^{(1)}(u_0) + [\frac{1}{2!}u_2^2 + u_1 u_3] f^{(2)}(u_0) + \frac{1}{2!}u_1^2 u_2 f^{(3)}(u_0) + \frac{1}{4!}u_1^4 f^{(4)}(u_0)$$
$$\vdots$$

The convergence of the Adomian decomposition method are discussed in detail by Cherruault, G. Adomian and K. Abbaoui [20-22]. Cherruault et al. [23] presented some new results for convergence of Adomian decomposition method applied to integral equations.

3 Decomposition of Integers for Generating Adomian Polynomials

A computational procedure for obtaining Adomian polynomials through all possible non negative integer solutions of the equation (1) was presented by Abbaoui.

In this paper we have obtained all possible non - negative integer solutions of the equation (1) by decomposition of positive integer 'n' as an arrays\ and used this array elements as subscripts of u's to obtain $c(v,n)$ of (2).

For example, let us consider the case $c(v,3)$ where v = 1,2,3.

The decomposition of integers 3 provide an array [[3], [2, 1], [1, 1, 1]], make use of this array following manner to obtain $c(v,n)$, v = 1,2,3.

$c(1,3) = u_3$ (use [3] as a subscript of u)

$c(2,3) = u_2 u_1$ (use [2, 1], as subscripts of u and multiply u's if more than one elements in the group exist)

$c(3,3) = \frac{u_1^3}{3!}$ (use[1, 1, 1] as subscript of u and multiply u's if more than one element exist and divide it by the factorial of the number of repeated subscripts.)

The following is the procedure to obtain $c(v,n)$

3.1 Procedure

Step 1

Enter the positive integer n.

Step 2:

Decompose the positive integer n.

Step 3:

Make the integers in the decomposed array as subscripts of u and multiply u's if more than one integers present (in number) in a particular sub array.

Step 4:

Count the number of repeated integers in the sub array (say k) and multiply with 1/ factorial (k) [if more than one integers are repeated in a sub array then multiply each case separately].

Step 5:

Multiply the derivative and then add together.

Using the above procedure one can compute any number of Adomian polynomials elegantly.

4MATLAB Program for computing Adomian Polynomials

MATLAB helps us to perform many mathematical tasks such as solving ODE, PDE etc. Here we have used MATLAB for symbolic computation of Adomian polynomials

```
clear all
clc
syms u u_0
n = input('enter the number: ');
y = input('enter the non-linear term in terms of u_0')
g1 = intpartgen(n);
y1(1) = diff(y,1);
result(1)=u_1*y1(1);
for n2 = 2:n
  n3=n2;
  g = g1{n2+1};
y1(n2) = diff(y,n2);
  b=g;
  a = zeros(size(b));
  [m1 n1] = size(b);
for k = 1:m1
  [i1,j1,s1] = find(b(k,:));
    [m2 n2] = size(j1);
j2(k) = n2;
end
  j3 = sort(j2);
j3(m1+1) = 0;
for k1 = 1:m1
for k2 = 1:m1
if j3(1,k1)==j2(1,k2)
```

```
a(k1,:) = b(k2,:);
if k1<m1
          k1 = k1+1;
end
j2(1,k2) = 0;
        k2 = m1+1;
end
end
end
  i1 = 1;
  i2 = 1;
  temp2 = '';
for i=1:size(a)
if t~=0
for j=1:n
      t = sum(a(i,:)==j);

a(i,:);
j;
t;
        t1 = strcat('u_',num2str(j));
if t>1
ft = factorial(t);
          t2 = strcat('(',t1,'^',num2str(t),'/',num2str(ft),')');
else
          t2 = t1;
end
if (strcmp(temp2,''))
          temp2 = t2;
elseif (strcmp(temp2(length(temp2)),'+'))
          temp2 = strcat(temp2,t2);
else
          temp2 = strcat(temp2,'*',t2);
end
end
end
if j3(i1)==j3(i1+1)
      temp2 = strcat(temp2,'+');
else
term(i2) = sym(temp2);
      temp2 = '';
      i2 = i2+1;
end
    i1 = i1+1;
end
l1=length(term);
  t1=0;
for k3=1:l1
 t1 = t1+ term(k3)*y1(k3);
```

```
end
result(n3)= t1;
end
result
```

We have called three built-in functions in MATLAB intpartgen, intpartgen 2 and integerparttable to decompose the positive integer which are freely available to download.

4.1 Examples

The computation of c(v,n) for n = 2.4.5...8 and v=1 to n are given in Table 1.

Table 1. Computing c(v,n) for A_n

c(v,n)	Decomposition
c(v,2)	[u_2, u_1^2/2]
c(v,3)	[u_3, u_1*u_2, u_1^3/6]
c(v,4)	[u_4, u_2^2/2 + u_1*u_3, (u_1^2*u_2)/2, u_1^4/24]
c(v,5)	[u_5, u_1*u_4 + u_2*u_3, (u_3*u_1^2)/2 + (u_1*u_2^2)/2, (u_1^3*u_2)/6,u_1^5/120]
c(v,6)	[u_6, u_3^2/2 + u_1*u_5 + u_2*u_4, (u_4*u_1^2)/2 + u_3*u_1*u_2 + u_2^3/6, (u_3*u_1^3)/6 + (u_1^2*u_2^2)/4, (u_1^4*u_2)/24, u_1^6/720]
c(v,7)	[u_7, u_1*u_6 + u_2*u_5 + u_3*u_4, (u_5*u_1^2)/2 + u_4*u_1*u_2 + (u_1*u_3^2)/2 + (u_2^2*u_3)/2, (u_4*u_1^3)/6 + (u_3*u_1^2*u_2)/2 + (u_1*u_2^3)/6, (u_3*u_1^4)/24 + (u_1^3*u_2^2)/12, (u_1^5*u_2)/120, u_1^7/5040]
c(v,8)	[u_8, u_4^2/2 + u_1*u_7 + u_2*u_6 + u_3*u_5, (u_6*u_1^2)/2 + u_5*u_1*u_2 + u_4*u_1*u_3 + (u_4*u_2^2)/2 + (u_2*u_3^2)/2, (u_5*u_1^3)/6 + (u_4*u_1^2*u_2)/2 + (u_1^2*u_3^2)/4 + (u_1*u_2^2*u_3)/2 + u_2^4/24, (u_4*u_1^4)/24 + (u_3*u_1^3*u_2)/6 + (u_1^2*u_2^3)/12, (u_3*u_1^5)/120 + (u_1^4*u_2^2)/48, (u_1^6*u_2)/720, u_1^8/40320]

Once we get c(v,n), then multiply it with $f^{(v)}(u_0)$ v =1,2,3…n provides us A_n

Adomian polynomials A_5 for few nonlinear terms are presented in Table 2.

Table 2. Adomian Polynomial A_5 for few nonlinear terms

Nonlinear terms	Adomian polynomials A_5
u^2	2*u_0*u_5 + 2*u_1*u_4 + 2*u_2*u_3
u^3	3*u_1*u_2^2 + 3*u_1^2*u_3 + 3*u_0^2*u_5 + 6*u_0*(u_1*u_4 + u_2*u_3)
u^4	12*u_0^2*(u_1*u_4 + u_2*u_3) + 24*u_0*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + 4*u_1^3*u_2 + 4*u_0^3*u_5]
u^5	20*u_0^3*(u_1*u_4 + u_2*u_3) + 5*u_0^4*u_5 + u_1^5 + 60*u_0^2*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + 20*u_0*u_1^3*u_2.
e^u	(u_1^5*exp(u_0))/120 + exp(u_0)*(u_1*u_4 + u_2*u_3) + u_5*exp(u_0) + exp(u_0)*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + (u_1^3*u_2*exp(u_0))/6
$\log(u)$	u_1^5/(5*u_0^5) - (u_1*u_4 + u_2*u_3)/u_0^2 + u_5/u_0 + (2*((u_3*u_1^2)/2 + (u_1*u_2^2)/2))/u_0^3 - (u_1^3*u_2)/u_0^4
$\sin(u)$	(u_1^5*cos(u_0))/120 - sin(u_0)*(u_1*u_4 + u_2*u_3) + u_5*cos(u_0) - cos(u_0)*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + (u_1^3*u_2*sin(u_0))/6

The computation time for generating the Adomian polynomials is given in Table 3.

Table 3. Computation time (tic-toc) for Adomian Polynomials upto A_n for e^u

A_n 's	Computation Time (in seconds)
10	0.786148
15	1.478237
20	4.171372
25	15.231208
30	54.270038

5 Conclusion

The MATLAB code which we have presented here is a simple one and also easily understandable. The main advantage of our procedure is the generation of Adomian polynomials for nonlinear terms as presented by G. Adomian. Extension of the above procedure for the multivariable case is also possible.

Acknowledgement

The Authors would like to thank anonymous reviewers for their valuable comments to improve the content of the article. We also thank Mr, David Holdaway for keeping his MATLAB code as an open source to partition the integers.

References

[1] Adomian G, Rach R. Generalization of Adomian polynomials to functions of several variables. Computer Math. Applic. 1992;24(5/6):11-24.

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

[3] Adomian G. Nonlinear stochastic operator equations. Academic Press, New York; 1986.

[4] Abbaoui K, Cherruault Y. New ideas for proving convergence of decomposition methods, Comput. Math. Appl. 1995;29(7):103-108.

[5] Adomian G. Explicit solutions of nonlinear partial differential equations. Appl. Math. Comput. 1997;88:117-126.

[6] Guellal S, Cherruault Y. Practical formulae for calculation of Adomians polynomials and application to the convergence of the decomposition method. Int. J. Biomed. Comput. 1994;36:223-228.

[7] Seng V, Abbaoui K, Cherruault Y. Adomian polynomials for nonlinear operators. Math. Comput. Model. 1996;24:59-65.

[8] Wazwaz AM. A new algorithm for calculating Adomian polynomials for nonlinear operators. Appl. Math. Comput. 2000;111:53-69.

[9] Rach R. A convenient computational form for the Adomian polynomials. J. Math. Anal. Appl. 1984;102:415–419.

[10] Wenhai Chen, Zhengyi Lu. Symbolic implementation of the algorithm for calculating Adomian polynomials. Applied Mathematics and Computation. 2004;159:221–235.

[11] Duan JS. An efficient algorithm for the multivariable Adomian polynomials. Applied Mathematics and Computation. 2010;217(6a):2456-67.

[12] Duan JS. Recurrence triangle for Adomian polynomials. Applied Mathematics and Computation. 2010;216(4b):1235-41.

[13] Duan JS. Convenient analytic recurrence algorithms for the Adomian polynomials. Applied Mathematics and Computation. 2011;217(13):6337-48.

[14] Biazar A, Shafiof SM. A simple algorithm for calculating Adomian polylnomials. Int. J. Math. Sciences. 2007;2(20):975-982.

[15] Gabet L. The theoretical foundation of the Adomian method. Comput. Math. Appl. 1994;27(12):41-52.

[16] Hooman Fatootehchi, Hossein Abolghasemi. On calculation of Adomian polynomials by MATLAB. Journal of Applied Computer Science & Mathematics. 2011;11(5):85-88.

[17] Choi HW, Shin AG. Symbolic implementation of the algorithm for calculating Adomian polylnomials. Applied Mathematics and Computation. 2003;146:257-271.

[18] Abbaoui K, Cherruault Y. Convergence of Adomian's method applied to differential equations. Comput. Math. Appl. 1994;28(5):103-109.

[19] Adomian G. Solving frontier problems of physics: the decomposition method. Kluwer Acad. Publ.; 1994.

[20] Abbaoui K, Cherruault Y. Convergence of Adomian's method applied to nonlinear equations. Math. Comput. Model. 1994;20(9):60-73.

[21] Cherruault Y. Convergence of Adomian's method. Kybernetes. 1989;18:31-38.

[22] Cherruault Y, Adomian G. Decomposition methods: a new proof of convergence. Math. Comput. Model. 1993;18:103-106.

[23] Cherruault Y, Saccomandi G, Some B. New results for convergence of Adomian's method applied to integral equations. Math. Comput. Modell. 1992;16(2):85-93.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=729&id=6&aid=6999