

Original Research Article

Symbolic Computation of Adomian Polynomials Based On Rach's Rule

Author details:

M. Kaliyappan^{1*}, S. Hariharan²

¹Associate Professor, VIT University Chennai.

²Assistant Professor (sr.), VIT University Chennai* Tel.: +91 44 39931232;

fax: +91 44 39932555.

E-mail address: kaliyappan.m@vit.ac.in

kaliyappan.m@vit.ac.in

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[4–7]. Several authors have been focused in this area to develop a practical method for the calculation of Adomian polynomials[1,3,8,10,18,20-22]. R.Rach have presented a convenient computational form to obtain Adomian polynomials [20]. Wenhai Chen et.al have developed an algorithm which is easily programmable in MAPLE, to calculate Adomian polynomials for nonlinear terms in the differential equations [23]. 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[14-16]. J. Biazar and S. M.Shafiof [9] developed a simple algorithm for generating Adomian polynomials. The theoretical basis of the Adomian decomposition method was discussed in detail by L. Gabet[17]

Hooman Fatoorhchi et.al[13]. and H.W.Choi et.al[19] have developed MATLAB and MATHEMATICA subroutines for computing Adomian polynomials based on the technique proposed by A.M. Wazwaz respectively.

Abbaoui et al.[1] have presented a computational procedure for obtaining Adomian polynomials through all possible non negative integer solutions of the equations

$$\alpha_1 + \alpha_2 + \dots + \alpha_n = n, \text{ with } \alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n \dots \quad (1)$$

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 [5].

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

$f(u) = \sum_{n=0}^{\infty} A_n(u_0, u_1, \dots, u_n)$, 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 as [5]

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

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.

A reference list of the Adomian polynomials for f(u) presented by G.Adomian [5] is shown below

$$A_0 = f(u_0)$$

$$A_1 = u_1 f^{(1)}(u_0)$$

$$A_2 = u_2 f^{(1)}(u_0) + \frac{1}{2!} u_1^2 f^{(2)}(u_0)$$

$$A_3 = u_3 f^{(1)}(u_0) + u_1 u_2 f^{(2)}(u_0) + \frac{1}{3!} u_1^3 f^{(3)}(u_0)$$

$$A_4 = u_4 f^{(1)}(u_0) + \left[\frac{1}{2!} u_2^2 + u_1 u_3 \right] f^{(2)}(u_0) + \frac{1}{2!} u_1^2 u_2 f^{(3)}(u_0) + \frac{1}{4!} u_1^4 f^{(4)}(u_0)$$

.

The convergence of the Adomian decomposition series and the series of the Adomian polynomials have proved by Cherruault, G.Adomian and K.Abbaoi[1,2,3,11,12]. Cherruault et al, presented some new results for convergence of Adomian decomposition method[10].

3. DECOMPOSITION OF INTEGERS FOR GENERATING ADOMIAN'S POLYNOMIALS

Abbaoui et al.[1] have presented a computational procedure for obtaining Adomian's polynomials through all possible non negative integer solutions of the equation (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 arrays\ and used this array elements as subscripts of u's to obtain $c(v, n)$ of (2).

85 For example, let us consider the case $c(v,3)$ where $v = 1,2,3$.
 86
 87 The decomposition of integers 3 provide an array $[[3], [2, 1], [1, 1, 1]]$, make use of this array following manner to obtain
 88 $c(v,n)$, $v = 1,2,3$.
 89 $c(1,3) = u_3$ (use [3] as a subscript of u)
 90 $c(2,3) = u_2 u_1$ (use [2, 1], as subscripts of u and multiply u's if more than one elements in the group exist)
 91 $c(3,3) = \frac{u_1^3}{3!}$ (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
 92 the number of repeated subscripts.)
 93
 94 The following is the procedure to obtain $c(v,n)$

96 3.1 Procedure:

97
 98 Step 1
 99 Enter the positive integer n.
 100 Step 2:
 101 Decompose the positive integer n.
 102 Step 3:
 103 Make the integers in the decomposed array as subscripts of u and multiply u's if more than one integers present (in
 104 number) in a particular sub array.
 105 Step 4:
 106 Count the number of repeated integers in the sub array (say k) and multiply with $1/\text{factorial}(k)$
 107 [if more than one integers are repeated in a sub array then multiply each case separately].
 108 Step 5:
 109 Multiply the derivative and then add together.

110
 111 Using the above procedure one can compute any number of Adomian polynomials elegantly.

113 4. MATLAB Program for computing Adomian polynomials

114
 115 MATLAB helps us to perform many mathematical tasks such as solving ODE, PDE etc. Here we have used MATLAB for
 116 symbolic computation of Adomian polynomials

```
117
118 clear all
119 clc
120 syms u u_0
121 n = input('enter the number: ');
122 y = input('enter the non-linear term in terms of u_0')
123 tic
124 g1 = intpartgen(n);
125 y1(1) = diff(y,1);
126 for n2 = 2:n
127     g = g1{n2+1};
128     % [M N] = size(g);
129     y1(n2) = diff(y,n2);
130     % b = g{N};
131     b=g;
132     a = zeros(size(b));
133     [m1 n1] = size(b);
134     for k = 1:m1
135         [i1,j1,s1] = find(b(k,:));
136         [m2 n2] = size(j1);
137         j2(k) = n2;
138     end
139     j3 = sort(j2);
140     j3(m1+1) = 0;
```

```

141 for k1 = 1:m1
142     for k2 = 1:m1
143         if j3(1,k1)==j2(1,k2)
144             a(k1,:) = b(k2,:);
145             if k1<m1
146                 k1 = k1+1;
147             end
148             j2(1,k2) = 0;
149             k2 = m1+1;
150         end
151     end
152 end
153 i1 = 1;
154 i2 = 1;
155 temp2 = "";
156 for i=1:size(a)
157     for j=1:n
158         t = sum(a(i,:)==j);
159         if t~=0
160             a(i,:);
161             j;
162             t;
163             t1 = strcat('u_',num2str(j));
164             if t>1
165                 ft = factorial(t);
166                 t2 = strcat('(',t1,'^',num2str(t),',',num2str(ft),')');
167             else
168                 t2 = t1;
169             end
170             if (strcmp(temp2,""))
171                 temp2 = t2;
172             elseif (strcmp(temp2(length(temp2)),'+'))
173                 temp2 = strcat(temp2,t2);
174             else
175                 temp2 = strcat(temp2,'*',t2);
176             end
177         end
178     end
179     if j3(i1)==j3(i1+1)
180         temp2 = strcat(temp2,'+');
181     else
182         term(i2) = sym(temp2);
183         temp2 = "";
184         i2 = i2+1;
185     end
186     i1 = i1+1;
187 end
188 result(n2)=sum(term.*y1);
189 end
190 toc
191 result(1) = y;
192 result
193

```

We have called three built-in functions in MATLAB `intpartgen`, `intpartgen2` and `integerparttable` to decompose the positive integer which are freely available to download.

4.1 Examples

The computation of $c(v,n)$ for $n = 2,4,5\dots 8$ and $v=1$ to n are given in Table 1.

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

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

202

203

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

204

Table 2: Adomian polynomial A_5 for the nonlinear terms $u^2, u^3, u^4, u^5, e^u, \log(u)$ and $\sin(u)$

205

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	$12*u_0^2*(u_2^2/2 + u_1*u_3) + 4*u_0^3*u_4 + u_1^4 + 12*u_0*u_1^2*u_2$
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	$(u_1^5*\exp(u_0))/120 + \exp(u_0)*(u_1*u_4 + u_2*u_3) + u_5*\exp(u_0) + \exp(u_0)*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + (u_1^3*u_2*\exp(u_0))/6$
$\log(u)$	$u_1^5/(5*u_0^5) - (u_1*u_4 + u_2*u_3)/u_0^2 + u_5/u_0 + (2*((u_3*u_1^2)/2 + (u_1*u_2^2)/2))/u_0^3 - (u_1^3*u_2)/u_0^4$
$\sin(u)$	$(u_1^5*\cos(u_0))/120 - \sin(u_0)*(u_1*u_4 + u_2*u_3) + u_5*\cos(u_0) - \cos(u_0)*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + (u_1^3*u_2*\sin(u_0))/6$

206

207

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

208

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.

REFERENCES

1. Abbaoui K, Cherruault Y. Convergence of Adomian's method applied to differential equations, Comput. Math. Appl. 1994;28(5):103–109.
2. Abbaoui K, Cherruault Y. Convergence of Adomian's method applied to nonlinear equations, Math. Comput. Model. 1994;20(9):60–73
3. Abbaoui K, Cherruault Y. New ideas for proving convergence of decomposition methods, Comput. Math. Appl. 1995;29(7):103–108
4. Adomian G, Rach R. Generalization of Adomian polynomials to functions of several variables, Computer Math.Applic 1992;24(5/6):11-24.
5. Adomian G. Solving frontier problems of physics: the decomposition method, Kluwer Acad.Publ.; 1994.
6. Adomian G. A review of the decomposition method in applied mathematics, J. Math. Anal. Appl. 1988;135:501–544.
7. Adomian G. Nonlinear stochastic operator equations, Academic Press, New York; 1986.
8. Adomian G. Explicit solutions of nonlinear partial differential equations, Appl. Math. Comput. 1997;88:117–126.
9. Biazar A, Shafiof S. M., A Simple Algorithm for Calculating Adomian Polynomials, Int. J. Math. Sciences, 2007, 2 (20): 975-982.
10. 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.
11. Cherruault Y. Convergence of Adomian's method, Kybernetes, 1989;18:31–38.
12. Cherruault Y, Adomian G. Decomposition methods: a new proof of convergence, Math. Comput. Model. 1993;18:103–106.
13. Choi H.W, Shin A.G. Symbolic implementation of the algorithm for calculating Adomian Polynomials, Applied Mathematics and Computation, 2003;146:257-271
14. Duan J.S. An efficient algorithm for the multivariable Adomian polynomials, Applied Mathematics and Computation, 2010;217(6a):2456-67.
15. Duan J.S. Recurrence triangle for Adomian polynomials, Applied Mathematics and Computation, 2010;216(4b):1235-41.
16. Duan J.S. Convenient analytic recurrence algorithms for the Adomian polynomials ,Applied Mathematics and Computation, 2011;217(13):6337-48
17. Gabet L. The theoretical foundation of the Adomian method, Comput. Math. Appl. 1994; 27(12):41–52.
18. 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.
19. Hooman Fatootehchi, Hossein Abolghasemi, On Calculation of Adomian Polynomials by MATLAB, Journal of Applied Computer Science & Mathematics, 2011;11(5):85-88.
20. Rach R. A convenient computational form for the Adomian polynomials, J. Math. Anal. Appl. 1984;102:415–419.
21. Seng V, Abbaoui K, Cherruault Y. Adomian polynomials for nonlinear operators, Math. Comput. Model. 1996;24:59–65.
22. Wazwaz A.M. A new algorithm for calculating Adomian polynomials for nonlinear operators, Appl. Math. Comput. 2000;111:53–69.
23. Wenhai Chen, Zhengyi Lu. Symbolic implementation of the algorithm for calculating Adomian polynomials, Applied Mathematics and Computation 2004;159:221–235.