

Original Research Article

Symbolic Computation of Adomian Polynomials Based On Rach's Rule

ABSTRACT

This paper presents a simple way of computing Adomian's 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 (ARIAL, BOLD, 11 FONT, LEFT ALIGNED, CAPS)

Adomian formally introduced formulas for generating Adomian's polynomials for all forms of nonlinearity[1–5]. Several authors have been focused in this area to develop a practical method for the calculation of Adomian polynomials[6–13]. R.Rach have presented a convenient computational form to obtain Adomian polynomials[6]. 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[15]. Duan et.al have developed analytic recurrence algorithms for obtaining Adomian Polynomials and an efficient algorithm for getting multivariable Adomian polynomial[17-19]. J. Biazar and S. M.Shafiof developed an algorithm for generating Adomian polynomials.

Hooman Fatoorhchi et.al. have developed a MATLAB program for computing Adomian polynomials based on the technique proposed by Wazwaz[16].

Abbaoui et al.[23] have presented a computational procedure for obtaining Adomian polynomials through all possible non negative integer solutions of the equations $\alpha_1 + \alpha_2 + \dots + \alpha_n = n$, with

$$\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n \quad \dots \quad (1)$$

In this paper we have obtained the all possible non - negative integer solutions of the equation (2) by decomposition of positive integer 'n' as an arrays and used this array elements as subscripts of u's to obtain Adomian polynomial as given by G.Adomian[2].

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 presented in Section 5.

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 solution that are expanded in the A_n polynomials based on the assumptions of the solution, because

$$\sum_{n=0}^{\infty} A_n \quad \text{reduces to} \quad \sum_{n=0}^{\infty} u_n \quad \text{for} \quad f(u) = u \quad \text{and} \quad f(u) = \sum_{n=0}^{\infty} A_n.$$

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 [2]

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

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 [2] 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.Abbaoui[20-22].

3. DECOMPOSITION OF INTEGERS FOR GENERATING ADOMIAN'S POLYNOMIALS.

Abbaoui et al.[23] have presented a computational procedure for obtaining Adomian's polynomials through all possible non negative integer solutions of the equations $\alpha_1 + \alpha_2 + \dots + \alpha_n = n$, with

$$\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n \quad \dots \quad (2)$$

In this paper we have obtained the all possible non - negative integer solutions of the equation (2) 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 (1).

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 u)

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

$c(3,3) = u_1^3$ (use [1, 1, 1] as subscript of u's and multiply u's if more than one elements are exists and divide by factorial of number of repeated subscript.

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 are (in number) in a particular sub array

Step 4:

Count the numbers repeated integers in the sub array and multiply with $1/\text{factorial}(k)$ where k = the number of times the integer repeated in a sub array). [if more than one integers are repeated in a sub array then multiply each case separately]

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

4. MATLAB 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's polynomials

```
clear all
clc
syms u
n = input('enter the number: ');
y = u^10;
tic
g1 = intpartgen(n);
y1(1) = diff(y,1);
for n2 = 2:n
    g = g1{n2+1};
    % [M N] = size(g);
    y1(n2) = diff(y,n2);
    % b = g{N};
    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
```

```

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

```

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 following Table 1 shows $c(v,n)$ for $n = 2,4,5\dots 8$ and $v=1$ to n

$c(v,n)$	$c(v,n)$
----------	----------

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,3)	[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]

Table 1: computing c(v,n) for A_n

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

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	$20*u_0^3*(u_1*u_4 + u_2*u_3) + 5*u_0^4*u_5 + u_1^5 + 60*u_0^2*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + 20*u_0*u_1^3*u_2$
$\exp(u)$	$(u_1^5*\exp(u_0))/120 + \exp(u_0)*(u_1*u_4 + u_2*u_3) + u_5*\exp(u_0) + \exp(u_0)*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + (u_1^3*u_2*\exp(u_0))/6$
$\log(u)$	$u_1^5/(5*u_0^5) - (u_1*u_4 + u_2*u_3)/u_0^2 + u_5/u_0 + (2*((u_3*u_1^2)/2 + (u_1*u_2^2)/2))/u_0^3 - (u_1^3*u_2)/u_0^4$
$\sin(u)$	$(u_1^5*\cos(u_0))/120 - \sin(u_0)*(u_1*u_4 + u_2*u_3) + u_5*\cos(u_0) - \cos(u_0)*((u_3*u_1^2)/2 + (u_1*u_2^2)/2) + (u_1^3*u_2*\sin(u_0))/6$

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

5. Conclusion

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

REFERENCES

1. Adomian G. Generalization of Adomian polynomials to functions of several variables, Computer Math.Applic 1992;24(5/6):11-24.
2. Adomian G. Solving frontier problems of physics: the decomposition method, Kluwer Acad.Publ.; 1994.
3. Adomian G. A review of the decomposition method in applied mathematics, J. Math. Anal. Appl. 1988;135:501–544.
4. Adomian G. Nonlinear stochastic operator equations, Academic Press, New York; 1986.
5. Adomian G. A new approach to nonlinear partial differential equations, J. Math. Anal. Appl. 1984;102:420–434.
6. Adomian G. Explicit solutions of nonlinear partial differential equations, Appl. Math. Comput. 1997;88:117–126.
7. Rach R. A convenient computational form for the Adomian polynomials, J. Math. Anal.Appl. 1984;102:415–419.
8. Seng V, Abbaoui K, Cherruault Y. Adomian polynomials for nonlinear operators, Math.Comput. Modell. 1996;24:59–65.

- 214 9. Abbaoui K, Cherruault Y. Convergence of Adomian_s method applied to differential
215 equations, Comput. Math. Appl. 1994;28(5):103–109.
- 216 10. Guellal S, Cherruault Y. Practical formulae for calculation of Adomians polynomials and application to the
217 convergence of the decomposition method, Int. J. Biomed. Comput. 1994;36:223–228.
- 218 11. Wazwaz A.M. A new algorithm for calculating Adomian polynomials for nonlinear
219 operators, Appl. Math. Comput. 2000;111:53–69.
- 220 12. Cherruault Y, Saccomandi G. Some New results for convergence of Adomian_s method applied to integral equations,
221 Math. Comput. Modell. 1992;16(2):85–93.
- 222 13. Abbaoui K, Cherruault Y. New ideas for proving convergence of decomposition methods, Comput. Math. Appl.
223 1995;29(7):103–108.
- 224 14. Gabet L. The theoretical foundation of the Adomian method, Comput. Math. Appl. 1994; 27(12):41–52.
- 225 15. Wenhai Chen, Zhengyi Lu. Symbolic implementation of the algorithm for calculating Adomian polynomials, Applied
226 Mathematics and Computation 2004;159:221–235.
- 227 16. Hooman Fatootehchi, Hossein Abolghasemi, On Calculation of Adomian Polynomials by MATLAB, Journal of Applied
228 Computer Science & Mathematics, 2011;11(5):85-88.
- 229 17. Duan J. S. An efficient algorithm for the multivariable Adomian polynomials, Applied Mathematics and Computation,
230 2010;217(6a):2456-67.
- 231 18. Duan J. S. Recurrence triangle for Adomian polynomials, Applied Mathematics and Computation, 2010;216(4b):1235-
232 41.
- 233 19. Duan J.S. Convenient analytic recurrence algorithms for the Adomian polynomials ,Applied Mathematics and
234 Computation, 2011;217(13):6337-48
- 235 20. Cherruault Y. Convergence of Adomian's method, Kybernetes, 1989;18:31–38.
- 236 21. Cherruault Y, Adomian G. Decomposition methods: a new proof of convergence, Math. Comput. Model. 1993;18:103–
237 106.
- 238 22. Abbaoui K, Cherruault Y. Convergence of Adomian's method applied to nonlinear equations, Math. Comput. Model.
239 1994;20(9):60–73.
- 240 23. Abbaoui K, Cherruault Y. Convergence of Adomian's Method Applied to Differential Equations Computers Math.
241 *Appl.* 1994;28(5):103-109.