International Journal of Advanced Research in Engineering and Technology (IJARET)
Volume 11, Issue 6, June 2020, pp. 298-306, Article ID: IJARET_11_06_026
Available online athttp://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=11&IType=6
ISSN Print: 0976-6480 and ISSN Online: 0976-6499
DOI: 10.34218/IJARET.11.6.2020.026

SOLVING NONLINEAR INTEGRAL EQUATIONS AND NONLINEAR INTEGRO - DIFFERENTIAL EQUATIONS USING LAPLACE ADOMIAN DECOMPOSITION METHOD THROUGH SAGEMATH

M. Kaliyappan and G. Hannah Grace
Division of Mathematics,
VIT - Vellore Institute of Technology, Chennai Campus, Tamilnadu, India
Email: kaliyappan.m@vit.ac.in; hannahgrace.g@vit.ac.in

ABSTRACT

Adomain decomposition method is a powerful technique to solve nonlinear differential equations and nonlinear Integro - differential equations. This paper presents a novel way of computing Adomian polynomials for nonlinear terms occur in nonlinear differential equations through partial Bell polynomial function(subroutine) in SageMath software. The obtained Adomian polynomials is used to solve nonlinear integral equations and nonlinear Integro - differential equations by Laplace Adomian decomposition method through symbolic computation in SageMath software.

Key words: Adomian polynomials, Bell polynomials. Nonlinear integral equations and Nonlinear Integro - differential equations, Adomain Decomposition Method(ADM)

Cite this Article: M. Kaliyappan and G. Hannah Grace, Solving Nonlinear Integral Equations and Nonlinear Integro - Differential Equations using Laplace Adomian Decomposition Method through Sagemath, *International Journal of Advanced Research in Engineering and Technology*, 11(6), 2020, pp. 298-306.
http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=11&IType=6

1. INTRODUCTION

Most scientific phenomena modeled as nonlinear ordinary or partial differential equations. Adomian decomposition method is one of the best approximation technique to solve ordinary differential equations, partial differential equations, integro-differential equations, stochastic differential equations, fuzzy differential equations and fuzzy fractional differential equations [5,7,8,9,31]. In Adomian decomposition method, generation of Adomian polynomials for nonlinear terms plays an essential role. George Adomian presented methods for generating Adomian polynomials for all forms of nonlinear terms. Several researchers have focused to find the methods or algorithms for generating Adomian polynomials

[3,6,12,17,18,19,22,29,30,32]. Many researchers developed subroutines to generate Adomian polynomials in mathematical software such as MATLAB, MAPLE, MATHEMATICA and SAGEMATH [16,18,23,24,25]. The convergence of Adomian decomposition methods for differential equations and news proffs for convergence and new results of the convergence are discussed by the several authors[1,2,4,13,14,15]. The theoretical basis of the Adomian decomposition method was discussed in detail by L. Gabet [21] .The procedure for solving nonlinear integral equations and nonlinear Integro - differential equations using Laplace Adomian decomposition method was provided by Wazwaz [33,34]. Several researchers have applied different methods to solve nonlinear integral equations and solving nonlinear integro-differential equations [10,20,27]. Algorithm for generating Adomian polynomials through partial Bell polynomials functions in MATLAB and MATHEMATICA was presented by the author [26].

In this article, Section 2 describes the basic definitions of Adomian polynomials and partial Bell polynomials and relationship between them. The procedure for getting Adomian polynomials using Partial Bell polynomial function and computation time for generating Adomian polynomials A_n are presented in Section 3. Solution of nonlinear Volterra integral equations and nonlinear Integro - differential equations using Laplace ADM is presented in Section 4.

2. ADOMIAN POLYNOMIALS AND PARTIAL BELL POLYNOMIALS

2.1 Adomian Polynomials

While solving nonlinear integral equations and integro-differential equations using Adomain decomposition method generation of Adomian polynomials plays an important role. The procedure for solving nonlinear integral equations and integro-differential equations using Adomian decomposition methods is presented in[33].

A few formulae or rules for generation of Adomain polynomial for nonlinear terms which occurs in the nonlinear differential equations are provided below[5].

Definition 2.1

For the n^{th} order differential function f, Adomian polynomial A_n are given by

$$A_n(u_0, u_1, \dots, u_n) = \frac{1}{n!}\frac{d^n}{d\lambda^n} f\left(\sum_{i=0}^{n} \lambda^i u_i\right)\Big|_{\lambda=0} \quad (1)$$

Definition 2.2

Rach [24] developed an elegant way to generate A_n is $A_n = \sum_{v=1}^{n} c(v,n)\, f^{(v)}(u_0)$ (2)

Several researchers have focused in this area and developed many algorithms for generating Adomian polynomials.

2.2. Partial Bell Polynomials

The Bell polynomial is a special type of polynomials having many applications in Combinatorial analysis[11]. Bell polynomials are useful to obtain Stirling numbers, Lah numbers and idempotent numbers.

Definition 2.3

The $(n,k)^{th}$ incomplete or partial exponential Bell polynomial $B_{n,k}$ in $, u_2, \dots, u_{n-k+1}$variables may be defined by the explicit formula

$$B_{n,k}(u_1, u_2, \dots, u_{n-k+1}) = \sum_{\substack{l_1+l_2+\cdots+l_{n-k+1}=k \\ l_1+2l_2+\cdots+(n-k+1)l_{n-k+1}=n}} n!\ \prod_{j=1}^{n-k+1} \frac{1}{l_j!}\left(\frac{u_j}{j!}\right)^{l_j} \quad (3)$$

Definition 2.4

The n^{th} complete exponential Bell polynomial is given by

$$B_n(u_1, u_2, \dots, u_n) = \sum_{k=1}^{n} B_{n,k}(u_1, u_2, \dots, u_{n-k+1}) \quad (4)$$

Abbaoui et.al [3] discussed the relationship between Bell polynomials and Adomian polynomials. The relationship between partial exponential Bell polynomials and Adomian polynomials is discussed in [28] as follows:

Let $A_n, n \geq 1$, be the nth Adomian polynomial for nonlinear term $N(u)$. Then

$$A_n(u_0, u_1, u_2, \dots, u_n) = \frac{1}{n!}\sum_{k=1}^{n} B_{n,k}(1!\,u_1, 2!\,u_2, \dots, (n-k+1)!\,u_{n-k+1})\, N^k(u_0). \quad (5)$$

Also the authors proved that

$$c(k,n) = \frac{1}{n!} B_{n,k}(1!\,u_1, 2!\,u_2, \dots, (n-k+1)!\,u_{n-k+1}) \quad (6)$$

From the equation (6) , it is quite easy to generate $c(k,n)\ k = 1,2,\dots,n$ using partial Bell polynomial subroutine in SageMath. Multiplying derivatives of the nonlinear term with $c(k,n)$ and summing over provides Adomian polynomials $A_n, n \geq 1$ $(Rach\ rule)$ equa.(2).

3. COMPUTATION ADOMIAN POLYNOMIALS USING PARTIAL BELL POLYNOMIAL FUNCTION THROUGH SAGEMATH

SageMath is a free open-source mathematics software. This software is useful for teaching and research. This software useful for learning Algebra, Linear algebra, Graph theory, Combinatorics, Calculus and differential equations etc. The function *partial_bell_polynomial(n, k)* was developed by Peter Luschny [36] in SageMath. This function generates Bell polynomials based on the variables $x_0, x_1, \dots$. Also this function provides Bell polynomials depend only on the variables x_1, x_2, ...,by substituting $x_0 = 0$.

The following code provides generation of partial Bell polynomial $B_{n,k}$ for $n = 5$ and $k = 1,2,\dots,5$ using the function partial_bell_polynomial(n, k).

```
n=5
s1 = [0]*(n)
for k in (1..n):
  s1[k-1]= partial_bell_polynomial(n, k).subs(x_0=0)
show(s1)
```

The output of above code is

$$\left[x_5, 10\,x_2x_3 + 5\,x_1x_4, 15\,x_1x_2^2 + 10\,x_1^2x_3, 10\,x_1^3x_2, x_1^5\right]$$

The following algorithm provides generation of Adomian polynomial using partial Bell polynomial function.

Algorithm 3.1.

Input: n, and $f(u_0)$ (nonlinear term)

Step 1: In the function '*partial_bell_polynomial(n, k)*' substitute $x_1 = 1!\,u_1, x_2 = 2!\,u_2, \dots, x_N = n!\,u_N$

Step 2: Obtain $f^n(u_0),\ for\ n = 1,2,3,\dots,n$ (derivatives of nonlinear function) and store them in an array

Step 3: $(\frac{1}{n!})$*(Step 1

Step 4: Call Step 3 function for $k = 1,2,3, \dots, n$ and store the output in an array and substitute $x\_0 = 0$

Step 5: Do scalar product of two arrays in Step 2 and Step 4

Output: Adomian polynomial A_n , n ≥2

The following SageMath function *Adomian_polynomial(n,f)* is developed using the algorithm 3.1.

```
def Adomian_polynomial(n,f)
  dfun = [0]*(n)
  s1 = [0]*(n)
  s2 = [0]*(n)
  dfun[0]=f.derivative(u_0)
  for k in range(1,n):
     dfun[k]=dfun[k-1].derivative(u_0)
  for k in(1..n):
     s1[k-1]= (1/factorial(n))*partial_bell_polynomial_modified(n,k).subs(x_0=0)
  for k in range(0,n):
  s2[k]=dfun[k]*s1[k]
  s3=sum(s2)
  return(s3)
```

The function partial_bell_polynomial_modified(n,k) in the above code is obtained by replacing $x_1 = 1!\,u_1, x_2 = 2!\,u_2, \dots, x_N = n!\,u_n$ in partial_bell_polynomial(n, k).

Example 1

For $n = 5$, and the nonlinear terms u^5, after Step 3 in the algorithm 3.1 we obtain

After Step 2, we obtain

$$\left[5\,u_0^4, 20\,u_0^3, 60\,u_0^2, 120\,u_0, 120\right]$$

After Step 4, we obtain

$$\left[u_5, u_2u_3 + u_1u_4, \frac{1}{2}\,u_1u_2^2 + \frac{1}{2}\,u_1^2u_3, \frac{1}{6}\,u_1^3u_2, \frac{1}{120}\,u_1^5\right]$$

By doing scalar product of above two vectors(arrays) we obtain (Step 5)

$$u_1^5 + 20\,u_0u_1^3u_2 + 5\,u_0^4u_5 + 20\left(u_2u_3 + u_1u_4\right)u_0^3 + 30\left(u_1u_2^2 + u_1^2u_3\right)u_0^2$$

Table 1 The following table shows the Adomian polynomial A_5 for various nonlinear terms using the algorithm 3.1.

Nonlinear terms	Adomian polynomial A_5
u^2	$2\,u_2u_3 + 2\,u_1u_4 + 2\,u_0u_5$
u^3	$3\,u_1u_2^2 + 3\,u_1^2u_3 + 3\,u_0^2u_5 + 6\,(u_2u_3 + u_1u_4)u_0$
u^5	$u_1^5 + 20\,u_0u_1^3u_2 + 5\,u_0^4u_5 + 20\,(u_2u_3 + u_1u_4)u_0^3 + 30\,(u_1u_2^2 + u_1^2u_3)u_0^2$
e^u	$\frac{1}{120}\,u_1^5e^{u_0} + \frac{1}{6}\,u_1^3u_2e^{u_0} + \frac{1}{2}\,(u_1u_2^2 + u_1^2u_3)e^{u_0} + (u_2u_3 + u_1u_4)e^{u_0} + u_5e^{u_0}$
$\log(u)$	$\frac{u_1^5}{5\,u_0^5} - \frac{u_1^3u_2}{u_0^4} + \frac{u_5}{u_0} - \frac{u_2u_3 + u_1u_4}{u_0^2} + \frac{u_1u_2^2 + u_1^2u_3}{u_0^3}$
$cos(u)$	$-\frac{1}{120}\,u_1^5\sin(u_0) + \frac{1}{6}\,u_1^3u_2\cos(u_0) - (u_2u_3 + u_1u_4)\cos(u_0) + \frac{1}{2}\,(u_1u_2^2 + u_1^2u_3)\sin(u_0) - u_5\sin(u_0)$

Table 2 The following table shows the CPU times and Wall time for computing Adomian polynomial A_n using the function *Adomian_polynomial(n,f)*

n(no.of terms)	CPU time and Wall time
10	CPU times: user 126 ms, sys: 3.59 ms, total: 130 ms Wall time: 150 ms
15	CPU times: user 800 ms, sys: 2.93 ms, total: 803 ms Wall time: 859 ms
20	CPU times: user 5.13 s, sys: 11.9 ms, total: 5.14 s Wall time: 5.63 s
25	CPU times: user 2min 15s, sys: 219 ms, total: 2min 16s Wall time: 2min 23s

4. SOLVING NONLINEAR INTEGRAL EQUATIONS AND NONLINEAR INTEGRO - DIFFERENTIAL EQUATIONS USING LAPLACE ADOMIAN DECOMPOSITION METHOD

We have applied the above function *Adomian_polynomial(n.f)* to solve nonlinear integral equations and integra-differential equations using Laplace transform - Adomian decomposition method. We have modified the SageMath code(subroutine) for solving nonlinear differential equations in[25] to solve nonlinear integral equations and nonlinear integro-differential equations.

Example 2

Consider the following nonlinear Volterra integral equation[33]

$$u(x) = x + \int_0^x u^2(t)dt \qquad (7)$$

The exact solution of (7) is $u(x) = \tan(x)$ (8)

The solution by Laplace ADM for (7) upto 11 terms obtained through SageMath software is shown below:

$$u(x)=\frac{18888466084}{194896477400625}x^{21}+\frac{443861162}{1856156927625}x^{19}+\frac{6404582}{10854718875}x^{17}+\frac{929569}{638512875}x^{15}+\frac{21844}{6081075}x^{13}+\frac{1382}{155925}x^{11}+\frac{62}{2835}x^{9}+\frac{17}{315}x^{7}+\frac{2}{15}x^{5}+\frac{1}{3}x^{3}+x$$

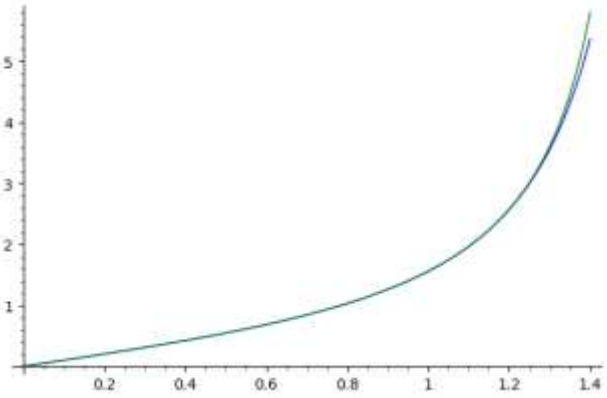

Figure 1 Comparison plot of Laplace ADM solution upto 11 terms (Blue) and the exact solution (Green) in the interval [0, 1.4].

The computational time required to obtain Laplace ADM solution of (7) upto 11 terms is given below:

CPU times: user 1min 59s, sys: 1.28 s, total: 2min Wall time: 2min 2s

Example 3

Consider the following nonlinear Volterra integral equation[20,27]

$$u(x) = 2x - \frac{x^4}{12} + 0.25\int_0^x (x-t)u^2(t)dt\,, 0 \le x \le 1 \quad (9)$$

The exact solution of (9) is $u(x) = 2x$ (10)

The solution by Lalplace ADM for (9) upto 4 terms obtained through SageMath software is shown below:

$$u(x)=\frac{1}{31784246968320}x^{22}-\frac{193}{10620547891200}x^{19}+\frac{7237}{1521657446400}x^{16}-\frac{127}{226437120}x^{13}+\frac{23}{1451520}x^{10}+\frac{1}{1008}x^{7}+\frac{1}{24}x^{4}+2x$$

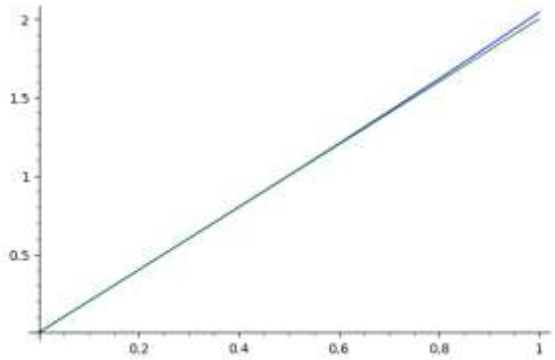

Figure 2 Comparison plot of Laplace ADM solution(Blue) and the exact solution (Green) in the interval [0,1].

Example 4

Consider the following nonlinear integro-differential equation[10]

$$u'(x) = -1 + \int_0^x u^2(t)dt\,, u(0) = 0, 0 \le x \le 1 \qquad (11)$$

The solution by Laplace ADM for (11) upto 8 terms obtained through SageMath software is shown below:

$$u(x)=\frac{73}{252975550464}x^{22} - \frac{25}{3011613696}x^{19} + \frac{37}{158505984}x^{16} - \frac{1}{157248}x^{13} + \frac{1}{6048}x^{10} - \frac{1}{252}x^{7} + \frac{1}{12}x^{4} - x$$

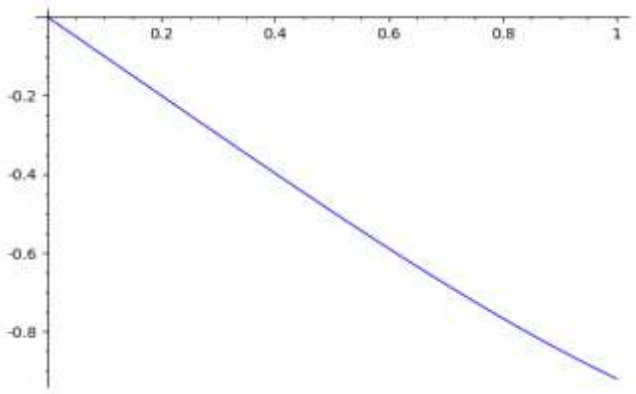

Figure 3 Plot of Laplace ADM solution upto 5 terms in the interval [0, 1]

Example 5

Consider the following nonlinear integro-differential equation [34]

$$u'(x) = \frac{3}{2}e^x - \frac{1}{2}e^{3x} + \int_0^x e^{x-t}u^3(t)dt\,,\ u(0) = 1. \qquad (12)$$

The exact solution of (12) is $u(x) = e^x$ (13)

The initial term is

$$u\_0(x)= -\frac{1}{6}e^{(3x)} + \frac{3}{2}e^{x} - \frac{1}{3}$$

The solution by Laplace ADM for (12) upto 2 terms obtained through SageMath software is shown below:

$$u(x)=\frac{1}{2}xe^{x} + \frac{1}{27}x - \frac{1}{15552}e^{(9x)} + \frac{1}{336}e^{(7x)} - \frac{1}{1080}e^{(6x)} - \frac{9}{160}e^{(5x)} + \frac{1}{24}e^{(4x)} + \frac{167}{432}e^{(3x)} - \frac{9}{8}e^{(2x)} + \frac{529}{320}e^{x} + \frac{6661}{68040}$$

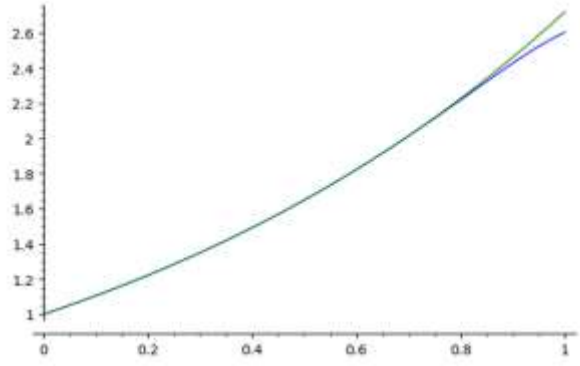

Figure 4 Comparison plot of Laplace ADM solution (Blue) upto 5 terms and the exact solution (Green) in the interval [0,1]

5. CONCLUSION

This article presents generation of Adomian polynomials for generating nonlinear terms using partial Bell polynomial function in SageMath. This algorithm is simple and easy to use. The entire code for generating Adomian polynomials is presented here. We have used the Adomian polynomials generated by the algorithm 3.1 and modified the subroutine in[25] to solve nonlinear integral equations and nonlinear Integro-differential equations. SageMath is an open-source software, the algorithm and the functions written in SageMath are useful for the researchers working in nonlinear integral equations and integro- differential equations in under developed countries.

REFERENCES

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

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

[3] Abbaoui. K., Cherruault. Y. and Seng. V., (1995), Practical Formulae for the Calculus of Multivariable Adomian Polynomials, Mathematical and Computer Modelling, 22 (1) 89-93

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

[5] Adomian G,(1994) Solving Frontier Problems of Physics: The Decomposition Method, Springer Science + Business Media, Dordhecht,

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

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

[8] Adomian G. (1986) Nonlinear stochastic operator equations. Academic Press, New York.

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

[10] Bahuguna. D., Ujlayan. A, Pandey, D.N. (2009) A comparative study of numerical methods for solving an integro-differential equation, Computers and Mathematics with Applications 57-1485–149

[11] Bell, E. T, (1934) Exponential polynomials, ann. of math., 35, 258–277.

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

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

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

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

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

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

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

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

[20] Dimple Rani, Vinod Mishra, (2018) Modification of Laplace Adomian decomposition method for solving nonlinear Volterra integral and integro-differential equations based on Newton Raphson formula, European Journal of Pure and Applied Mathematics, 11(1), 202-214

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

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

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

[24] Kaliyappan.M, and Hariharan. S. (2015) Symbolic computation of Adomian polynomial based on Rach's rule", British Journal of Mathematics & Computer Science, *5(5): 562-570.*

[25] Kaliyappan,M., Hariharan.S, (2019) Solving Nonlinear Differential Equations Using Adomian Decomposition Method Through Sagemath, International Journal of Innovative Technology and Exploring Engineering ,8(6);510-515

[26] M.Kaliyappan, (2020) Symbolic Computation of Adomian Polynomials Through partial Bell Polynomial Functions in MATLAB and Mathematica, Test engineering and Management, 83; 22427–22433

[27] Kamyad, A V, Mehrabinezhad, M and Nadja, J S, (2010) A numerical approach for solving linear and nonlinear Volterra integral equations with controlled error. International Journal of Applied Mathematics, 40.

[28] Kuldeep Kumar Kataria, Palaniappan Vellaisamy, (2017) Correlation between Adomian and partial exponential Bell polynomials, 2017.C. R.Acad.Sci.Paris,Ser.I355, 929–936

[29] Rach R. A (1984) Convenient computational form for the Adomian polynomials. J. Math. Anal. Appl. 102:415–419.

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

[31] Wazwaz, A.M (2009) Partial Differential Equations and Solitary Waves Theory, Springer

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

[33] Wazwaz. A M, (2011) Linear and Nonlinear Integral Equations: Methods and Applications. Springer.

[34] Wazwaz. A M, (2010) The combined Laplace transform–Adomian decomposition method for handling nonlinear Volterra integro–differential equations, Applied Mathematics and Computation 216, 1304–1309.

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

[36] https://oeis.org/wiki/User:Peter_Luschny/BellTransform