Solution of a class of nonlinear differential equations using an accelerated version of Adomian decomposition method

A. Y. Sayed¹, K. M. AbdElgaber², A. R. Elmahdy³ and I. L. El-Kalla⁴

¹Department of Engineering Mathematics and Physics, Faculty of Engineering El-Mataria, PO 11718 Cairo, Helwan University, Egypt (AHMED_BADR@m-eng.helwan.edu.eg).
ORCID: 0000-0001-5691-4993

²Department of Engineering Mathematics and Physics, Faculty of Engineering El-Mataria, PO 11718 Cairo, Helwan University, Egypt (khaled.abdelgaber@m-eng.helwan.edu.eg)

³Department of Engineering Mathematics and Physics, Faculty of Engineering El-Mataria, PO 11718 Cairo, Helwan University, Egypt (alielmahdy@m-eng.helwan.edu.eg) ORCID: 0000-0002-4268-1399

⁴Department of Physics & Engineering Mathematics, Faculty of Engineering, Mansoura University, PO 35516, Mansoura, Egypt. (al_kalla@mans.edu.eg)

Abstract
In this work, we proposed a reliable polynomials called El-kalla polynomials which are faster than the traditional Adomian polynomials in solving some classes of nonlinear differential equations by Adomian decomposition method (ADM). The main advantages of El-kalla polynomials can be summerized in the following main three points:

- El-kalla polynomials are recursive and do not have derivative terms so, El-kalla formula is easy in programming and save much time on the same processor compared with the traditional Adomian polynomials formula.
- Solution using El-Kalla polynomials converges faster than the traditional Adomian polynomials.
- El-Kalla polynomials used directly in estimating the maximum absolute truncated error of the series solution.

Some convergence remarks are studied and some numerical examples are solved to verify the above advantages. In all applied cases, we obtained an excellent performance that may lead to a promising approach for many applications.

Keywords: Adomian polynomial, El-Kalla polynomial, nonlinear differential equations.

1. Introduction
Adomian decomposition method (ADM) is a semi-analytical method for solving linear and nonlinear differential equations, linear and nonlinear integral and integro-differential
equations, this powerful method was developed from 1970s to 1990s by George Adomian [7,8,9,10,22], chair of the center for applied mathematics at the university of Georgia in USA. In this work we will illustrate the using of the Adomian polynomials and El-Kalla polynomials proposed by El-kalla [13] in some examples and compare the two solutions with the exact solution.

2. Adomian decomposition method (ADM)

First, we will highlight briefly the main points of the Adomian decomposition method in case of nonlinear ordinary and partial differential equations which solution contains Adomian polynomial or El-Kalla polynomial.

Consider the general nonlinear ordinary differential equation

\[ L y + R y + N y = f(x), \]  

(1)

where \( y \) is the unknown function, \( L = \frac{d^n}{dx^n} \) is the linear differential operator of higher order which is easily invertible. Assume its inverse is \( L^{-1} \) and it will be an integral operator (integration \( n \) times the \( n^{th} \) derivative from 0 to \( x \)), \( N \) is the nonlinear operator, \( R \) is the remaining linear part and \( f(x) \) is a given function.

The solution algorithm will be as following :

Take \( L^{-1} \) to both sides of (2.1.1) to get :

\[ L^{-1}L y = L^{-1} f(x) - L^{-1}N y - L^{-1}R y, \]  

(2)

\[ y = y(0) + L^{-1}f(x) - L^{-1}N y - L^{-1}R y, \]  

(3)

Let, \( y(0) = \varphi(x) \),

\[ y = \varphi(x) + L^{-1}f(x) - L^{-1}N y - L^{-1}R y, \]  

(4)

The ADM assumes that solution \( y \) of the given equation can be decomposed into infinite series.

\[ y = \sum_{n=0}^{\infty} y_n = y_0 + y_1 + y_2 + y_3 + \cdots, \]  

(5)

\[ y_n = \varphi(x) + L^{-1}f(x) - L^{-1} \sum_{n=0}^{\infty} A_n - L^{-1}R \sum_{n=0}^{\infty} y_n, \]  

(6)

And the nonlinear term \( N(y) \) can be written as infinite series

\[ N(y) = \sum_{n=0}^{\infty} A_n = A_0 + A_1 + A_2 + A_3 + \cdots, \]  

(7)

where \( A_n \) are the traditional Adomian polynomial or we can use the new El-Kalla polynomial \( \bar{A}_n \) that can be calculated from formula (8) or (9), where

the Adomian polynomial formula \( A_n = \frac{1}{n!} \left( \frac{d^n}{dx^n} \left[ f \left( \sum_{i=0}^{n} \lambda^i y_i \right) \right] \right)_{\lambda=0}. \)  

(8)

and the El-Kalla polynomial formula \( \bar{A}_n = f(S_n) - \sum_{i=0}^{n-1} \bar{A}_i. \)  

(9)
we can obtain the solution of the given equation as follows:

\[ y_0 = \varphi(x) + L^{-1}f(x), \]  
\[ y_{n+1} = -L^{-1}(A_n + R y_n), \]

where \( n = 0,1,2,3, \ldots \).

Consider the general nonlinear partial differential equation.

\[ L_t u + L_x u + R u(x,t) + f(u(x,t)) = f(x,t), \]  
(12)

where \( L_t \) is the highest derivative respect to the variable \( t \), \( L_x \) is the highest derivative respect to the variable \( x \), \( R u(x,t) \) is the remainder from the operator, \( f(u(x,t)) \) is the nonlinear term and \( f(x,t) \) is the free term. We will separate the highest derivative in the variable \( (x) \) or in the variable \( (t) \) in the left side of the equation.

\[ L_x u = f(x,t) - L_t u - R u(x,t) - f(u(x,t)), \]  
(13)

and then we operate \( L_x^{-1} \) to both sides, where \( L_x^{-1} \) is the inverse operator of \( L_x \) to get:

\[ u(x,t) = u(0,t) + L_x^{-1} f(x,t) + L_x^{-1}(L_t u) + L_x^{-1}(R u(x,t)) + L_x^{-1}(A_n), \]  
(14)

\[ u_0 = u(0,t) + L_x^{-1} f(x,t), \]  
(15)

\[ u_{n+1} = L_x^{-1}(L_t u_n) + L_x^{-1}(R u_n(x,t)) + L_x^{-1}(A_n), \]  
(16)

where \( n = 0,1,2,3, \ldots \),

the Adomian polynomials & El-Kalla polynomials of the nonlinear term \( y^2 \) are shown in Table 1 we can see that the terms using El-Kalla polynomials appear faster than Adomian polynomials.

**Table 1 : Adomian polynomials and El-Kalla polynomials of the nonlinear term \( y^2 \)**

<table>
<thead>
<tr>
<th>Adomian polynomials of ( y^2 )</th>
<th>El-Kalla polynomials of ( y^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_0 = y_0^2 )</td>
<td>( \bar{A}_0 = y_0^2 )</td>
</tr>
<tr>
<td>( A_1 = 2y_0y_1 )</td>
<td>( \bar{A}_1 = 2y_0y_1 + y_1^2 )</td>
</tr>
<tr>
<td>( A_2 = y_1^2 + 2y_0y_2 )</td>
<td>( \bar{A}_2 = 2y_0y_2 + 2y_1y_2 + y_2^2 )</td>
</tr>
<tr>
<td>( A_3 = 2y_1y_2 + 2y_0y_3 )</td>
<td>( \bar{A}_3 = 2y_0y_3 + 2y_1y_3 + 2y_2y_3 + y_3^2 )</td>
</tr>
<tr>
<td>( A_4 = y_2^2 + 2y_1y_3 + 2y_0y_4 )</td>
<td>( \bar{A}_4 = 2y_0y_4 + 2y_1y_4 + 2y_2y_4 + 2y_3y_4 + y_4^2 )</td>
</tr>
</tbody>
</table>

Also the nonlinear term \( y^3 \) the Adomian polynomials & El-kalla polynomials are shown in Table 2 we can see that the terms of El-kalla polynomials appear faster than Adomian polynomials.

**Table 2 : Adomian polynomials and El-kalla polynomials of the nonlinear term \( y^3 \)**

<table>
<thead>
<tr>
<th>Adomian polynomials of ( y^3 )</th>
<th>El-kalla polynomials of ( y^3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_0 = y_0^3 )</td>
<td>( \bar{A}_0 = y_0^3 )</td>
</tr>
<tr>
<td>( A_1 = 3y_0^2y_1 )</td>
<td>( \bar{A}_1 = 3y_0^2y_1 + y_1^3 )</td>
</tr>
<tr>
<td>( A_2 = 2y_1^3 + 3y_0^2y_2 )</td>
<td>( \bar{A}_2 = 2y_0y_2 + 3y_1y_2 + y_2^3 )</td>
</tr>
<tr>
<td>( A_3 = 3y_0^2y_3 + 2y_0y_2y_4 )</td>
<td>( \bar{A}_3 = 3y_0y_3 + 2y_1y_3 + y_3^3 )</td>
</tr>
<tr>
<td>( A_4 = y_2^3 + 2y_1y_3 + 4y_0y_4 )</td>
<td>( \bar{A}_4 = 2y_0y_4 + 2y_1y_4 + y_4^3 )</td>
</tr>
</tbody>
</table>
Convergence of the Adomian method when applied to some classes of ordinary and partial differential equations is discussed by many authors. For example, K. Abbaoui and Y. Cherruault [5,20,21] proved the convergence of the Adomian method for differential and operator equations. E. Babolian And J. Biazar, contemplate the order of the convergence of the Adomian method in [4]. Also Wazwaz and Khuri discussed applications of the Adomian decomposition method to a class of Fredholm integral equations that occurs in acoustics [3]. Zhang [23] presented a modified Adomian decomposition method to solve a class of nonlinear singular boundary-value problems, which arise as normal model equations in nonlinear conservative systems. Zhu et al. [24] presented a new algorithm for calculating Adomian polynomials for nonlinear operators. Also, many modifications were made to this method by numerous researchers in an attempt to improve the accuracy or extend the applications of this method [1, 2, 6, 11, 19]. Also, El-Kalla polynomial was discussed by El-Kalla In [13, 14, 15, 16, 17, 18], and conclude that El-kalla polynomial was directly used to estimate the maximum absolute truncated error of the Adomian series solution which cannot be estimated using the traditional polynomials.

### 4. Numerical Examples

#### 4.1. Example 1

Consider the nonlinear ordinary differential equation

\[ y''' + e^{-2x}y^3 = 2e^x, \quad y(0) = 1, \quad y'(0) = 1. \]  

(17)

We will solve this problem using using Adomian polynomials and El-Kalla polynomials.

#### 4.1.1 Solution by using Adomian polynomials

Let, the solution

\[ y = \sum_{i=0}^{\infty} y_i = y_0 + y_1 + y_2 + \ldots, \]  

(18)
\[ y'' = 2e^x - e^{-2x}y^3 \]  

(19)

Make integration of both sides from 0 to \( x \), we get

\[ \int_0^x f_1(x) \, dx + \int_0^x f_2(x) \, dx = \int_0^x 2e^x \, dx = \int_0^x e^{-2x}y^3 \, dx . \]  

(20)

\[ y = 2e^x - x - 1 \]  

(21)

\[ (y_0 + y_1 + y_2 + \ldots) = 2e^x - x - 1 - \int_0^x e^{-2x}(A_0, A_1, A_2, \ldots) \, dx . \]  

(22)

\[ y_0 = 2e^x - x - 1, \]  

(23)

\[ y_1 = -\int_0^x \int_0^x e^{-2x}A_0 \, dx \, dx \]  

\[ y_2 = -\int_0^x \int_0^x e^{-2x}A_1 \, dx \, dx \]  

\[ y_3 = -\int_0^x \int_0^x e^{-2x}A_2 \, dx \, dx . \]  

(24)

where the nonlinear term is \( y^3 \), we calculate \( A_0, A_1, A_2, \ldots \) from the Eq. (8),

\[ A_0 = (y_0)^3 = (2e^x - x - 1)^3 \]  

\[ A_1 = \frac{1}{11}[N(y_0 + \lambda y_1)]_{\lambda=0} = 3 * (x - 2e^x(x) + 1)^2 * ((23 * e^{(-2x)} - 66 * e^{-x} - 157/8 - 8 * e^{-x} + \frac{27x^2 * e^{(-2x)}}{8} - 6 * x^2 * e^{(-x)} + \frac{3x^2 * e^{(-2x)}}{2} + \frac{x^3 * e^{(-2x)}}{4} + 6 * x^2 + 2 * x^3 + \frac{569}{8} \]  

(25)

\[ y_1 = -\int_0^x \int_0^x e^{-2x}A_0 \, dx \, dx = -\int_0^x \int_0^x e^{-2x} (2e^x - x - 1)^3 \, dx \, dx, \]  

(26)

\[ y_2 = -\int_0^x \int_0^x e^{-2x}A_1 \, dx \, dx . \]  

So, the solution will be

\[ y = \sum_{i=0}^{\infty} y_i = y_0 + y_1 + y_2 + \ldots . \]  

(27)

where the exact solution.

\[ y(x) = e^x . \]  

(28)
4.1.2 Solution using El-Kalla polynomials

The solution is the same as before in equations (18 - 23) except when we calculate El-kalla polynomials using formula (9) as follow.

\[ \tilde{A}_0 = (y_0)^3 = (2e^x - x - 1)^3, \]

\[ \tilde{A}_1 = (y_0 + y_1)^3 - \tilde{A}_0, \]

\[ y_1 = - \int_0^x \int_0^x e^{-2x} \tilde{A}_0 \, dx \, dx = - \int_0^x \int_0^x e^{-2x} (2e^x - x - 1)^3 \, dx \, dx, \]

Note that: \( \exp(x) \) means \( e^x \)

The solution is

\[ y = \sum_{i=0}^{\infty} y_i = y_0 + y_1 + y_2 + \ldots = \frac{31189803239931391290771125507+x}{293778833960736000000000} + \frac{23287419+\exp(\text{-}x)+32}{4096} \]

\[ + \frac{727362423+\exp(-2\times x)+371990333+\exp(-3\times x)}{2592} - \frac{2474122895+\exp(-4\times x)}{262144} + \frac{184453414601+\exp(-5\times x)}{62500000} \]

\[ + \frac{758933335+\exp(-6\times x)}{2985984} + \frac{111909141909+\exp(-7\times x)}{1291315424} - \frac{2253818711+\exp(-8\times x)}{2147483648} + 218 \cdot \exp(x) + \ldots \]

As shown we take three terms of the series solution \( y = y_0 + y_1 + y_2 \) to graph.

![Figure 1. Solution using Adomian polynomials, Solution using El-Kalla polynomials and Exact solution of](image-url)
Figure 2. The difference between Exact solution and Solution using Adomian polynomials of
\[ y'' + e^{-2x}y^3 = 2e^x \]

Figure 3. The difference between Exact solution and Solution using El-Kalla polynomials of
\[ y'' + e^{-2x}y^3 = 2e^x \]

In Table 3, we introduce the absolute relative error (ARE) between the exact solution and the solution deduced by using El-Kalla polynomials. Also, (ARE) between the exact solution and the solution deduced by using Adomian polynomials for some values of \( x \), where we take three terms of the series solution \( y = y_0 + y_1 + y_2 \) in Example 1.

Table 3: the absolute relative error (ARE) between the exact solution and solution deduced by using El-Kalla polynomials, also between the exact solution and the solution deduced by using Adomian polynomials for some values of \( x \) in Example 1.
The time elapsed of the program that calculate the solution of Example 1 in Matlab R2014a by using Adomian polynomials = 3.8822 seconds, and by using El-Kalla polynomials = 2.2105 seconds.

### 4.2. Example 2

Consider the nonlinear partial differential equation

\[ u_t - uu_x = 0 \quad , \quad u(x,0) = x \]  

We will solve this problem by using Adomian polynomials and El-Kalla polynomials.

#### 4.2.1 Solution using Adomian polynomial

\[ \int_0^t u_t \ dt = \int_0^t uu_x \ dt , \]  

\[ u(x,t) - u(x,0) = \int_0^t uu_x \ dt , \]  

\[ u(x,t) = x + \int_0^t uu_x \ dt . \]  

Let, the solution

\[ u = \sum_{i=0}^{\infty} u_i = u_0 + u_1 + u_2 + ... \]  

\[ (u_0 + u_1 + u_2 + ...) = x + \int_0^t (A_0 + A_1 + A_2 + ...) \ dt \]  

\[ u_0 = x , \]  

\[ u_1 = \int_0^t A_0 \ dt , \]  

\[ u_2 = \int_0^t A_1 \ dt , \]  

\[ u_3 = \int_0^t A_2 \ dt . \]  

where \( A_0, A_1, A_2, \ldots \) are Adomian polynomials calculated from Equation (8), where the nonlinear term \( N(u) = uu_x \), such that
\[ A_0 = x \]
\[ A_1 = 2xt \]
\[ A_2 = 3xt^2 \]
\[ \vdots \]
\[ u_1 = \int_0^t A_0 \, dt = \int_0^t x \, dt = xt \]
\[ u_2 = \int_0^t A_1 \, dt = \int_0^t 2xt \, dt = x^2t \]
\[ u_3 = \int_0^t A_2 \, dt = \int_0^t 3xt^2 \, dt = x^3t \]
\[ \vdots \]
\[ u = \sum_{i=0}^{\infty} u_i = u_0 + u_1 + u_2 + \ldots \]
\[ u = x + xt + xt^2 + xt^3 + \ldots = x(1 + t + t^2 + t^3 + \ldots) = \frac{x}{1-t} \]

which equal to the exact solution.

**4.2.2 Solution using El-Kalla polynomials**

For the term \( u \), El-Kalla polynomials will be in the form
\[ \bar{A}_0 = u_0(u_0)_x = x \]
\[ \bar{A}_1 = 2xt + xt^2 \]
\[ \bar{A}_2 = 3xt^2 + \frac{9}{3}xt^3 + \frac{5}{3}xt^4 + \frac{2}{3}xt^5 + \frac{1}{9}xt^6 \]
\[ \vdots \]
\[ u_1 = \int_0^t \bar{A}_0 \, dt = \int_0^t x \, dt = xt \]
\[ u_2 = \int_0^t \bar{A}_1 \, dt = \frac{1}{3}xt^3 \]
\[ u_3 = \frac{2}{3}xt^3 + \frac{2}{3}xt^4 + \frac{1}{3}xt^5 + \frac{1}{9}xt^6 + \frac{1}{63}xt^7 \]
\[ \vdots \]
\[ u = \sum_{i=0}^{\infty} u_i = u_0 + u_1 + u_2 + u_3 + \ldots \]
\[ u = x + xt + xt^2 + xt^3 + \frac{2}{3}xt^4 + \frac{1}{3}xt^5 + \frac{1}{9}xt^6 + \frac{1}{63}xt^7 \]

where we take six terms of the series solution \( u = u_0 + u_1 + u_2 + u_3 + u_4 + u_5 + u_6 \).

In Table 4, we introduce the absolute relative error (ARE) between the exact solution and solution using El-Kalla polynomials. Also, (ARE) between the Exact solution and Solution
using Adomian polynomials for some values of \( t \) at \( x = 0.2 \) where we take seven terms of the series solution \( u = u_0 + u_1 + u_2 + u_3 + u_4 + u_5 + u_6 \) in Example 2.

Table 4: the absolute relative error (ARE) between the exact solution and solution using El-Kalla polynomials, also between the Exact solution and Solution using Adomian polynomials for some values of \( t \) in Example 2 at \( x = 0.2 \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>(ARE) of solution using Adomian polynomials at ( x = 0.2 )</th>
<th>(ARE) of solution using El-Kalla polynomials at ( x = 0.2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2.2222222*10^{-8}</td>
<td>3.657479*10^{-10}</td>
</tr>
<tr>
<td>0.2</td>
<td>3.2*10^{-6}</td>
<td>6.970819*10^{-8}</td>
</tr>
<tr>
<td>0.3</td>
<td>6.24857*10^{-5}</td>
<td>1.847009*10^{-6}</td>
</tr>
<tr>
<td>0.4</td>
<td>5.46133*10^{-4}</td>
<td>2.25723*10^{-4}</td>
</tr>
<tr>
<td>0.5</td>
<td>3.125*10^{-3}</td>
<td>1.8735022*10^{-4}</td>
</tr>
<tr>
<td>0.6</td>
<td>1.39968*10^{-2}</td>
<td>1.2737864*10^{-3}</td>
</tr>
<tr>
<td>0.7</td>
<td>5.4902866*10^{-2}</td>
<td>8.0283917*10^{-3}</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2097152</td>
<td>5.29241209*10^{-2}</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9565938</td>
<td>0.455370633</td>
</tr>
</tbody>
</table>

The time elapsed of the program that calculate the solution of Example 2 in Matlab R2014a by using Adomian polynomials = 1.9600 seconds and by using El-Kalla polynomials = 1.8810 seconds.

4.3. Example 3

Consider the nonlinear partial differential equation

\[
\frac{1}{36}x(u_{xx})^2 = x^3, \quad u(x, 0) = 0 .
\]  

We will solve this problem by using Adomian polynomials and El-Kalla polynomials.

4.3.1 Solution using Adomian polynomials

Let, the solution

\[
\sum_{i=0}^{\infty} u_i = u_0 + u_1 + u_2 + \ldots ,
\]

\[
u = \int_{0}^{t} x^3 dt - \frac{1}{36}X \int_{0}^{t} (u_{xx})^2 dt ,
\]

\[
(u_0 + u_1 + u_2 + \ldots) = tx^3 - \frac{1}{36}X \int_{0}^{t} (A_0 + A_1 + A_2 + \ldots) dt ,
\]

where \( A_0, A_1, A_2, \ldots \) are Adomian polynomials

\[
A_0 = (u_0)_{xx}^2 = (6xt)^2 = 36x^2t^2
\]

\[
A_1 = -24x^2t^4 ,
\]
\[ u_0 = tx^3, \]
\[ u_1 = -\frac{1}{3} x^3 t^3, \]
\[ u_2 = \frac{2}{15} x^3 t^5, \]
\[ \vdots \]
\[ u = u_0 + u_1 + u_2 + \cdots \]
\[ u(x,t) = tx^3 - \frac{1}{3} x^3 t^3 + \frac{2}{15} x^3 t^5 + \cdots \]

That converges to the exact solution \( x^3 \tanh(t) \).

**4.4.2 Solution using El-Kalla polynomials**

The solution is the same as before in equations (48 - 50) except when we calculate El-Kalla polynomials as follow, where \( \bar{A}_0, \bar{A}_1, \bar{A}_2, \ldots \) are El-Kalla polynomials calculated using the formula in the Eq. (14).

\[ (u_0 + u_1 + u_2 + \cdots) = tx^3 - \frac{1}{36} x \int_0^t (\bar{A}_0 + \bar{A}_1 + \bar{A}_2 + \cdots) \, dt \]

\[ \bar{A}_0 = (u_0)_{xx}^2 = (6xt)^2 = 36x^2 t^2, \quad \bar{A}_1 = 4x^2 t^6 - 24x^2 t^4 \]
\[ \vdots \]

\[ u_0 = tx^3, \]
\[ (u_0)_{xx} = 6xt, \]
\[ u_1 = -\frac{1}{3} x \int_0^t \bar{A}_0 \, dt = -\frac{1}{3} x^3 t^3, \quad u_2 = -\frac{1}{36} x \int_0^t \bar{A}_1 \, dt = -\frac{1}{63} x^3 t^7 + \frac{2}{15} x^3 t^5, \]
\[ \vdots \]
\[ u = u_0 + u_1 + u_2 + \cdots \]
\[ u(x,t) = tx^3 - \frac{1}{3} x^3 t^3 - \frac{1}{63} x^3 t^7 + \frac{2}{15} x^3 t^5 + \cdots. \]

where we take five terms of the series solution \( u = u_0 + u_1 + u_2 + u_3 + u_4 \).

In Table 5, we introduce the absolute relative error (ARE) between the Exact solution and Solution using El-Kalla polynomials. Also, (ARE) between the Exact solution and Solution using Adomian polynomials for some values of \( t \) at \( x=0.1 \) in Example 3.
Table 5: the absolute relative error (ARE) between the Exact solution and Solution using El-Kalla polynomials, also between the exact solution and solution using Adomian polynomials for some values of \( t \) in Example 3 at \( x=0.1 \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>(ARE) of solution using Adomian polynomials at ( x=0.1 )</th>
<th>(ARE) of solution using El-Kalla polynomials at ( x=0.1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>8.827459*10^{-17}</td>
<td>1.525276*10^{-17}</td>
</tr>
<tr>
<td>0.2</td>
<td>1.7862333612*10^{-13}</td>
<td>3.0402870449*10^{-14}</td>
</tr>
<tr>
<td>0.3</td>
<td>1.5148409094*10^{-11}</td>
<td>2.5151368377*10^{-12}</td>
</tr>
<tr>
<td>0.4</td>
<td>3.491126763*10^{-10}</td>
<td>5.6013697159*10^{-11}</td>
</tr>
<tr>
<td>0.5</td>
<td>3.9296006604*10^{-9}</td>
<td>6.0387302385*10^{-10}</td>
</tr>
<tr>
<td>0.6</td>
<td>2.806134482*10^{-8}</td>
<td>4.0960869202*10^{-9}</td>
</tr>
<tr>
<td>0.7</td>
<td>1.462178507*10^{-7}</td>
<td>2.0119183251*10^{-8}</td>
</tr>
<tr>
<td>0.8</td>
<td>6.045396213*10^{-7}</td>
<td>7.7870697982*10^{-8}</td>
</tr>
<tr>
<td>0.9</td>
<td>2.093969172*10^{-6}</td>
<td>2.50926822026*10^{-7}</td>
</tr>
<tr>
<td>1</td>
<td>6.30707861*10^{-6}</td>
<td>6.9918170107*10^{-7}</td>
</tr>
</tbody>
</table>

The time elapsed of the program that calculate the solution of Example 3 in Matlab R2014a by using Adomian polynomials= 2.1672 seconds and by using El-Kalla polynomials= 1.6457 seconds.

4.4. Example 4
Consider the following nonlinear ordinary differential equation that we do not know it's exact solution.

\[
y' = x^3 + \frac{1}{10}y^2 \quad , \quad y(0) = 0 . \quad (60)
\]

We will solve this problem by using Adomian polynomials and El-Kalla polynomials.

4.4.1 Solution by using Adomian polynomials
Let, the solution

\[
y = \sum_{i=0}^{\infty} y_i = y_0 + y_1 + y_2 + ... \quad (61)
\]

Make integration of both sides to eq. (60) from 0 to \( x \), we get

\[
\int_0^x y' \, dx = \int_0^x x^3 + \frac{1}{10}y^2 \, dx \quad (62)
\]

\[
y = \int_0^x x^3 \, dx + \frac{1}{10}\int_0^x y^2 \, dx \quad (63)
\]

\[
(y_0 + y_1 + y_2 + ...) = \frac{x^4}{4} + \frac{1}{10}\int_0^x (A_0, A_1, A_2, ... ) \, dx \quad (64)
\]
\[ y_0 = \frac{x^4}{4}, \]
\[ y_1 = \frac{1}{10} \int_0^x A_0 \, dx, \]
\[ y_2 = \frac{1}{10} \int_0^x A_1 \, dx, \]
\[ y_3 = \frac{1}{10} \int_0^x A_2 \, dx. \]

Where the nonlinear term is \( y^2 \), we calculate \( A_0, A_1, A_2, \ldots \) from the Eq. (8),

\[ A_0 = y_0^2 = \frac{x^8}{16} \]
\[ A_1 = \frac{1}{1!} \frac{d}{d\lambda} [N(y_0 + \lambda y_1)]_{\lambda=0} = \frac{x^{13}}{2880} \]
\[ A_2 = \frac{x^{18}}{580608} \]
\[ A_3 = \frac{11x^{23}}{1378944000} \]

\[ y_1 = \int_0^x A_0 \, dx = \frac{x^9}{1440} \]
\[ y_2 = \int_0^x A_1 \, dx = \frac{x^{14}}{403200} \]
\[ y_3 = \int_0^x A_2 \, dx = \frac{x^{19}}{110315520} \]
\[ y_4 = \int_0^x A_3 \, dx = \frac{11x^{24}}{330946560000} \]

So, the Solution is

\[ y = \sum_{i=0}^{\infty} y_i = y_0 + y_1 + y_2 + \ldots \]

\[ y = \frac{x^4}{4} + \frac{x^9}{1440} + \frac{x^{14}}{403200} + \frac{x^{19}}{110315520} + \frac{11x^{24}}{330946560000} + \ldots \]

### 4.4.2 Solution using El-Kalla polynomials

The solution is the same as before in equations (61 - 65) except that when we calculate El-kalla polynomials as follow.

\[ \bar{A}_0 = y_0^2 = \frac{x^8}{16} \]
\[ \tilde{A}_1 = (y_0 + y_1)^2 - \tilde{A}_0 = \left( \frac{x^9}{1440} + \frac{x^4}{4} \right)^2 - \frac{x^8}{16} \]

\[ \tilde{A}_2 = (y_0 + y_1 + y_2)^2 - (\tilde{A}_0 + \tilde{A}_1) \]

\[ = \left( \frac{x^{14} \cdot (7 \cdot x^5 + 6840)}{2757888000} + \frac{x^4}{4} + \frac{x^9}{1440} \right)^2 - \left( \frac{x^9}{1440} + \frac{x^4}{4} \right) \]

\[ \tilde{A}_3 = (y_0 + y_1 + y_2 + y_3)^2 - (\tilde{A}_0 + \tilde{A}_1 + \tilde{A}_2) \]

\[ = \left( \frac{x^{19} \cdot (24157 \cdot x^{20} + 54152280 \cdot x^{15} + 48795739200 \cdot x^{10})}{1462395279823994880000000} \right) \]

\[ + \frac{x^{19} \cdot (2872236931200 \cdot x^5 + 95446642636800000)}{1462395279823994880000000} \]

\[ + \frac{x^{14} \cdot (7 \cdot x^5 + 6840)}{2757888000} + \frac{x^4}{4} + \frac{x^9}{1440} \]

\[ - \left( \frac{x^{14} \cdot (7 \cdot x^5 + 6840)}{2757888000} + \frac{x^4}{4} + \frac{x^9}{1440} \right)^2 \]

\[ y_1 = \int_0^x \tilde{A}_0 \, dx = \frac{x^9}{1440} \]

\[ y_2 = \int_0^x \tilde{A}_1 \, dx = \frac{x^{14} \cdot (7 \cdot x^5 + 6840)}{2757888000} \]

\[ y_3 = \int_0^x \tilde{A}_2 \, dx \]

\[ = \frac{x^{19} \cdot (24157 \cdot x^{20} + 54152280 \cdot x^{15})}{1462395279823994880000000} \]

\[ + \frac{x^{19} \cdot (48795739200 \cdot x^{10} + 2872236931200 \cdot x^5 + 95446642636800000)}{1462395279823994880000000} \]
The solution is

\[ y = \sum_{i=0}^{\infty} y_i = y_0 + y_1 + y_2 + y_3 + y_4 + \ldots \]  

(72)

where we take five terms of the series solution \( y = y_0 + y_1 + y_2 + y_3 + y_4 \).

The time elapsed of the program that calculate the solution of Example 5 in Matlab R2014a By using Adomian polynomials = 4.2985 seconds and by using El-Kalla polynomials = 2.2884 seconds.

we do not know the exact solution of this problem but, from [13] we can estimate directly the maximum absolute truncation error of the series solution using the formula that proved by El-kalla:

\[ \max \left| y(x) - \sum_{i=0}^{m} y_i(x) \right| \leq \max \frac{\alpha^m}{1 - \alpha} |y_1(x)| \]  

(73)

\[ \alpha = \frac{LMT^k}{k!}, \quad L = 2, \quad M = \frac{1}{10}, \quad T = 1, \quad k = 1 \quad \text{and} \quad \alpha = \frac{2^{*}1}{10^{*}1} = 0.2 \]

where the number of iteration \( m = 4 \).

In Table 6, we introduce the maximum absolute truncation error is estimated to some values of \( x \) in Example 4.
Table 6: the Maximum Absolute Truncation Error ($\text{MATE}=\max |y(x) - \sum_{i=0}^{m} y_i(x)|$) is estimated to some values of $x$ in Example 4.

| $x$   | $\max |y(x) - \sum_{i=0}^{m} y_i(x)| \leq \max \frac{a^m}{1-a} |y_1(x)|$ |
|-------|---------------------------------------------------------------|
| 0.1   | MATE $\leq 2.6064438 \times 10^{-43}$                       |
| 0.2   | MATE $\leq 1.25876608 \times 10^{-34}$                      |
| 0.3   | MATE $\leq 1.608985837 \times 10^{-29}$                     |
| 0.4   | MATE $\leq 6.75774325 \times 10^{-26}$                      |
| 0.5   | MATE $\leq 4.36781602 \times 10^{-23}$                      |
| 0.6   | MATE $\leq 8.643319455 \times 10^{-21}$                     |
| 0.7   | MATE $\leq 7.55872213 \times 10^{-19}$                      |
| 0.8   | MATE $\leq 3.637177685 \times 10^{-17}$                     |
| 0.9   | MATE $\leq 1.109412954 \times 10^{-15}$                     |
| 1     | MATE $\leq 2.363048877 \times 10^{-14}$                     |

5. Discussions and Conclusions
From the previous examples it is clear that the time elapsed in the program that calculate the solution by using El-Kalla polynomial is less than the time elapsed in the program that calculate the solution by using Adomian polynomial. Also, the maximum error between the exact solution and the solution using El-Kalla polynomials is less than the maximum error between the exact solution and the solution using Adomian polynomial in calculations, so El-Kalla polynomial is faster and more accurate than Adomian polynomial as shown in Tables 3, 4, 5. Also, we use the equation (73) to calculate the Maximum Absolute Truncation Error (MATE) as shown in Table 6.

References


