On Some Computational Methods for Solving an Ordinary Differential Equation with Initial Condition.

1Oshinubi I. K., 2Adeyeye, F.J., 3Amao, F.A & 4Longe, O.B.
1Laboratoire Jean Kuntzmann, Mathématiques applique informatique
Universite Grenoble, Alpes, France
2Department of Math/Computer Science, Federal University of Petroleum Resources,
Effurun, Nigeria
3Department of Mathematics, Faculty of Science, Adeleke University, Ede, Nigeria.
4School of IT & Computing, American University of Nigeria, Yola, Nigeria
E-mails: 1kayode.oshinubi@univ-grenoble-alpes.fr, 2adeyeye.fola@fupre.edu.ng;
3fola_amao@yahoo.com; 4longeolumide@fulbrightmail.org

Abstract: This research presents computational methods for solving ordinary differential equation with initial condition. Ogunrinde et al.(2012) presented Euler's method and Runge Kutta method for solving initial value problem in ordinary differential equation. We shall use Adomian decomposition method to solve the problem solved by Ogunrinde et al.(2012) and compare the results and error. The error incurred is undertaken to determine the accuracy and consistency of the three methods

Keywords: Adomian Decomposition Method (ADM), Differential Equation, Euler's Method, Error, Stability Runge Kutta Method,.

1. INTRODUCTION

Ordinary Differential equations can describe nearly all system undergone change. Many mathematicians have studied the nature of these equations and many complicated systems can be described quite precisely with compact mathematical expressions. However, many systems involving differential equations are so complex. It is in these complex systems where computer simulations and numerical approximations are useful. The techniques for solving differential equations based on numerical approximations were developed before programmable computers existed. The problem of solving ordinary differential equations is classified into initial value and boundary value problems, depending on the conditions specified at the end points of the domain.
There are numerous methods that produce numerical approximations to solution of initial value problems in ordinary differential equations such as Euler’s method which was the oldest and simplest method originated by Leonhard Euler in 1768. Improved Euler’s method and Runge Kutta methods described by Carl Runge and Martin Kutta in 1895 and 1905 respectively.

There are many excellent and exhaustive texts on this subject that may be consulted, such as [8], [4],[6],[5],[17],[18] and [1] just to mention few. In this work we present the practical use of Adomian decomposition method for solving ordinary differential equations with initial conditions.

2. NUMERICAL METHOD

Numerical method forms an important part of solving initial value problems in ordinary differential equations, most especially in cases where there is no closed form solution. Next we present three numerical methods namely Euler’s Method, Runge Kutta method, and Adomian Decomposition method.

2.1 Runge Kutta Method
Runge Kutta method is a technique for approximating the solution of ordinary differential equation. This technique was developed around 1900 by the mathematicians Carl Runge and Wilhelm Kutta. Runge Kutta method is popular because it is efficient and used in most computer programs for differential equation.

The following are the orders of Runge Kutta Method as listed below:
-Runge Kutta method of order one is called Euler’s method.
-Runge Kutta method of order two is the same as modified Euler’s or Heun’s Method.
-Runge Kutta method of order four is called classical Runge Kutta method

2.2 Euler’s Method
Euler’s method is also called tangent line method and is the simplest numerical method for solving initial value problem in ordinary differential equation, particularly suitable for quick programming which was originated by Leonhard Euler in 1768. This method subdivided into three namely,
-2.2.1 Forward Euler’s method.
-2.2.2 Improved Euler’s method.
-2.2.3 Backward Euler’s method.

2.3 Adomian Decomposition Method
The Adomian decomposition method, proposed by Adomian initially with the aims to solve frontier physical problem, has been applied to a wide class of deterministic and stochastic problems, linear and nonlinear, in physics, biology and chemical reactions etc. For nonlinear models, the method has shown reliable results in supplying analytical approximation that converges very rapidly. It is well known that the key of the method is to decompose the nonlinear term in the equations into a peculiar series of polynomials are the so-called Adomian polynomials. Adomian formally introduced formulas that can generate Adomian polynomials for all forms of nonlinearity. Recently, a great deal of interests has been focused to develop a practical method for the calculation of Adomian polynomials An.
However, the methods developed by [9]-[15] also require a huge size of calculations. [16] Established a promising algorithm that can be easily programmed in Maple, and be used to calculate Adomian polynomials for nonlinear terms in the differential equations. Let us first recall the basic principles of the Adomian decomposition methods for solving differential equations.

Consider the general equation $F \frac{d^q u}{dt^q} = g$, where $F$ represents a general nonlinear differential operator involving both linear and nonlinear terms, the linear term is decomposed into $L \circ R$, where $L$ is easily invertible and $R$ is the remainder of the linear operator. For convenience, $L$ may be taken as the highest order derivate. Thus the equation may be written as:

$$Lu + Ru + Nu = g,$$  \hspace{1cm} (1)

Where $Nu$ represents the nonlinear terms. Solving $Lu$ from (1), we have $Lu = g - Ru - Nu$ \hspace{1cm} (2)

Because $L$ is invertible, the equivalent expression is $Lu = L^{-1}g - L^{-1}Ru - L^{-1}Nu$ \hspace{1cm} (3)

If $L$ is a second order operator, for example, $L^{-1}$ is a twofold integration operator and $L^{-1}Lu = u - u(0) - t(0)$, then (3) for $u$ yields

$$u = a + bt + L^{-1}g - L^{-1}Ru - L^{-1}Nu$$ \hspace{1cm} (4)

Therefore, $u$ can be presented as a series

$$u = \sum_{n=0}^{\infty} u_n$$ \hspace{1cm} (5)

with $u_0$ identified as $a + bt + L^{-1}g$ and $u_n (n > 0)$ is to be determined. The nonlinear term $Nu$ will be decomposed by the infinite series of Adomian polynomials.
NUMERICAL EXPERIMENTS

We shall now proceed by using ADM Method In order to confirm the applicability and suitability of the methods for solution of initial value problems in ordinary differential equations, it was computerized in Maple software package. The performance of the methods was checked by comparing their accuracy and efficiency with methods used by Ogunrinde et al.(2012). The efficiency was determined from the number iterations counts and number of functions evaluations per step while the accuracy is determined by the size of the discretization error estimated from the difference between the exact solution and the numerical approximations.

\[ N u = \sum_{n=0}^{\infty} A_n, \]  
\[ \text{where } A_n \text{'s are obtained by writing} \]
\[ v(\lambda) = \sum_{n=0}^{\infty} \lambda^n u_n, \]
\[ N(v(\lambda)) = \sum_{n=0}^{\infty} \lambda^n A_n. \]

Here \( \lambda \) is a parameter introduced for convenience. From (7) and (8), we deduce
\[ A_n = \frac{1}{n!} \left[ \frac{d^n}{d\lambda^n} N(v(\lambda)) \right]_{\lambda=0}, \quad n = 0, 1, \ldots \]  

Now, substituting (5) and (6) into (4), we obtain
\[ \sum_{n=0}^{\infty} u_n = u_0 - L^{-1} R \sum_{n=0}^{\infty} u_n - L^{-1} \sum_{n=0}^{\infty} A_n. \]

Consequently, we can write
\[ u_0 = a + bt + L^{-1} g, \]
\[ u_1 = -L^{-1} Ru_0 - L^{-1} A_0, \]
\[ \vdots \]
\[ u_{n+1} = -L^{-1} Ru_0 - L^{-1} A_n. \]

All of \( u_n \) are calculable, and \( u = \sum_{n=0}^{\infty} u_n \). Since the series converges and does so very rapidly, the \( n \)-term partial sum \( \varphi_n = \sum_{i=0}^{n-1} u_i \) can serve as a practical solution.

3. NUMERICAL EXPERIMENTS

We shall now proceed by using ADM Method In order to confirm the applicability and suitability of the methods for solution of initial value problems in ordinary differential equations, it was computerized in Maple software package. The performance of the methods was checked by comparing their accuracy and efficiency with methods used by Ogunrinde et al.(2012). The efficiency was determined from the number iterations counts and number of functions evaluations per step while the accuracy is determined by the size of the discretization error estimated from the difference between the exact solution and the numerical approximations.
Example:
We use Adomian Decomposition method to approximate the solution of the initial value problem \( y' \), \( y(0) = 2 \), with step size \( h = 0.1 \) on the interval \( 0 \leq x \leq 1 \) whose exact solution is given by \( y(x) = e^x + x + 1 \). The results obtained shown in Table 1 and Table 2, the comparison of the method to Euler, Runge Kutta and exact solution and the error incurred respectively.

### 3.1 Table of Results

**Table 1: The Comparative Result Analysis of Adomian Decomposition Method, Runge Kutta Method and Euler’s Method**

<table>
<thead>
<tr>
<th>( n )</th>
<th>( x_n )</th>
<th>( y(x_n) )</th>
<th>Runge Kutta</th>
<th>Euler</th>
<th>Adomian</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>1</td>
<td>0.1</td>
<td>2.2051</td>
<td>2.2051</td>
<td>2.2000</td>
<td>1.8048</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>2.4213</td>
<td>2.4213</td>
<td>2.4100</td>
<td>1.6186</td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>2.6497</td>
<td>2.6497</td>
<td>2.6310</td>
<td>1.4402</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>2.8918</td>
<td>2.8918</td>
<td>2.8641</td>
<td>1.2684</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>3.1487</td>
<td>3.1486</td>
<td>3.1105</td>
<td>1.1018</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>3.4221</td>
<td>3.4221</td>
<td>3.3716</td>
<td>0.9392</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>3.7137</td>
<td>3.7137</td>
<td>3.6487</td>
<td>0.7792</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>4.0255</td>
<td>4.0255</td>
<td>3.9436</td>
<td>0.6203</td>
</tr>
<tr>
<td>9</td>
<td>0.9</td>
<td>4.3596</td>
<td>4.3596</td>
<td>4.2579</td>
<td>0.4611</td>
</tr>
<tr>
<td>10</td>
<td>1.0</td>
<td>4.7182</td>
<td>4.7182</td>
<td>4.5937</td>
<td>0.3000</td>
</tr>
</tbody>
</table>
Table 2: Error incurred in Adomian Decomposition Method, Runge Kutta Method and Euler’s Method

<table>
<thead>
<tr>
<th>n</th>
<th>$x_n$</th>
<th>Runge kutta Error</th>
<th>Euler Error</th>
<th>Adomian Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.0001</td>
<td>0.0052</td>
<td>0.4004</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.0001</td>
<td>0.0114</td>
<td>0.8028</td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>0.0001</td>
<td>0.0188</td>
<td>1.2096</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>0.0000</td>
<td>0.0277</td>
<td>1.6234</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.0001</td>
<td>0.0382</td>
<td>2.0469</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>0.0000</td>
<td>0.0505</td>
<td>2.4829</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>0.0000</td>
<td>0.0650</td>
<td>2.9345</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.0000</td>
<td>0.0819</td>
<td>3.4052</td>
</tr>
<tr>
<td>9</td>
<td>0.9</td>
<td>0.0000</td>
<td>0.1017</td>
<td>3.8985</td>
</tr>
<tr>
<td>10</td>
<td>1.0</td>
<td>0.0000</td>
<td>0.1245</td>
<td>4.4182</td>
</tr>
</tbody>
</table>

3.2 Graph of Result
Given below is the graph of results

![Graphical Presentation of Runge Kutta Error and Adomian Decomposition Error](image)

Fig. 1: Graphical Presentation of Runge Kutta Error and Adomian Decomposition Error

DISCUSSION OF RESULTS
We noticed that in Table 2, the error incurred in Adomian Decomposition Method is greater than that of Runge Kutta method and Euler Method and at the same time get larger as $n$ increases. Hence Runge Kutta method is more accurate than its counterpart Euler’s method and Adomian Decomposition Method as we can see from Table 2.

5. CONCLUSION

We have in our disposal three numerical methods for solving initial value problems in ordinary differential equations. In general, numerical method has its own advantages and disadvantages of use. From the problem solved using MAPLE, it is observed that a lot of useful insights into numerical solution of initial value problem have been gained. We conclude that Runge Kutta method is consistent, convergent, quite stable and more accurate than Euler's method and Adomian Decomposition Method and it is widely used in solving initial value problems in ordinary differential equations.

REFERENCES

