https://dx.doi.org/10.4314/wojast.v15i2.34

ISSN: 2141 – 3290
www.wojast.com

ON NEWTON AND LAGRANGE INTERPOLATION METHOD FOR FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS

UDOH, P. J.[1], **NSIEN, E. F.**[2], **AND ABASIEKWERE U. A.**[1]
[1]*Department of Mathematics, University of Uyo, Nigeria*
[2]*Department of Statistics, University of Uyo, Nigeria*
**Correspondence: ubonabasiekwere@uniuyo.edu.ng*

ABSTRACT
In the recent past, there has been a tremendous growth of the solution of differential equations using algorithmic methods. This work deploys Newton's interpolation polynomial method (NIPM) and Lagrange interpolation polynomial method (LIPM) to create cubic polynomials for solving initial value problems of the first order. The results obtained show greater accuracy for the investigated method when compared with some known methods in literature. Illustrative examples are presented to demonstrate the validity and applicability of the technique.

KEYWORDS: Newton's interpolation, Lagrange interpolation, initial value problem, first order.

INTRODUCTION
Many mathematical models in science and engineering fields can be formulated in the form of linear and nonlinear ordinary differential equations which need an analytical method to solve the exact equations (Raza *et al*, 2020; Arqub, 2016; Singh and Singh (2019); Khan, 2022; Alkasassbeh *et al.*, 2019). However, in some problems, we cannot obtain the exact solutions by the analytical method. Therefore, numerical methods such as Euler's method, Runge-Kutta method and Runge-Kutta-Fehlberg method are important tools used to solve these kinds of problems. Many methods have been widely developed by a lot of researchers to solve these problems. Some problems in form of partial differential equations can even be converted to ordinary differential equations and solved computationally.
In this study, we consider only the first order ordinary differential equations with an initial condition (initial value problems) in the form
$$\frac{dy}{dx} = f(x,y); \qquad y(x_0) = y_0,$$
where $f(x,y)$ is a known function and the value of initial conditions x_0, y_0 are also known values.
The goal of this study is to estimate approximated solutions and absolute errors by comparing the results of our new method with other methods such as Euler's method.

First-order ODEs can be solved analytically, graphically, and also using numerical methods.
Numerical methods are methods used to find numerical approximations to the solutions of ordinary differential equations (ODEs). Their use is also known as "numerical integration", although this term can also refer to the computation of integrals. Many differential equations cannot be solved in closed forms using analytical methods. For practical purposes, however, such as in engineering; a numeric approximation to the solution is often sufficient. The algorithms studied here can be used to compute such an approximation. An alternative method is to use techniques from calculus to obtain a series expansion of the solution.
In our study, we use both Newton's interpolation and Lagrange polynomial to create cubic polynomials for solving initial value problems. By this method, it is simple to solve linear and nonlinear first order ordinary differential equations. Some numerical examples are provided to test the performance and illustrate the efficiency of this method. Research has also proven that these methods have been used to solve differential equations but we've not really thought of combining these two methods to solve problems.
We use numerical method as a tool to solve numerical problems. For instance, a differential equation

$$u'(x) = \cos x, \; 0 < x < 3,$$
written, as an equation involving some derivative of an unknown function u.
There is also a set of values allowed into the function of the differential equation (for the example; 0 <x< 3). In reality, a differential equation is then an infinite number of equations, one for each x in the domain. The analytic or exact solution is the functional expression of u or for the example case

$$u(x) = sinx + c,$$
where c is an arbitrary constant, because of this non uniqueness which is inherent in differential equations we typically include some additional equations.

Statement of the Problem
Research has proved that there are quite a number of numerical methods used in solving initial value problems for ordinary differential equations such as the Euler's method, the Runge-Kutta family of methods, the Taylor's series method etc.
We observe that most of the researches on numerical approach to the solution of ordinary differential equation tend to adopt the aforementioned methods in their traditional sense. A great deal of work has also been done using Newton's method and Lagrange methods separately to solve initial value problems. However, there is dearth of research in this area of combining the Newton's interpolation and Lagrange's interpolation methods to solve a first order initial value problem

The aim of this work is to formulate an iterative scheme that uses a combination of the Newton and the Lagrange interpolation methods to solve initial value problems for ordinary differential equation of the first order.

Main Formulations

Newton's Interpolation Polynomial Method

Isaac Newton (1641–1727) was one of the most brilliant scientists of all time. The late 17th century was a vibrant period for science and mathematics and Newton's work touched nearly every aspect of mathematics. His method for solving was introduced to find a root of the equation

$$y^3 - 2y - 5 = 0$$

Although he demonstrated the method only for polynomials, it is clear that he realized its broader applications.

Newton interpolation is a quadratic interpolation methodology used in numerical methods and outcomes. The interpolation formula in most classic procedures is particular to the data. This work discusses single and multivariable generalized Newton type polynomial interpolation approaches.

The forward difference formula and the backward difference formula are used in Newton polynomial interpolation.

$$y_0(x) = a_0, \quad (1)$$

$$y_1(x) = a_0 + a_1(x - x_0), \quad (2)$$

$$y_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1), \quad (3)$$

$$y_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0) + \cdots \ldots + a_n(x - x_0)(x - x_1) \ldots \ldots \ldots \ldots (x - x_{n-1}), \quad (4)$$

where

$$a_o = y_0 \quad (5)$$

$$a_1 = \frac{f(x_1) - f(x_0)}{(x_1 - x_0)} \quad (6)$$

$$a_2 = \frac{\frac{f(x_2)-f(x_1)}{(x_2-x_1)} - \frac{f(x_1)-f(x_0)}{(x_1-x_0)}}{(x_2 - x_0)} \quad (7)$$

$$a_3 = \frac{\frac{\frac{f(x_3)-f(x_2)}{(x_3-x_2)} - \frac{f(x_2)-f(x_1)}{(x_2-x_1)}}{(x_3-x_1)} - \frac{\frac{f(x_2)-f(x_1)}{(x_2-x_1)} - \frac{f(x_1)-f(x_0)}{(x_1-x_0)}}{(x_2-x_0)}}{(x_3-x_0)} \quad (8)$$

$$a_n = f[x_k, x_{k-1} \ldots x_1, x_0] = \frac{f[x_k, x_{k-1} \ldots x_1, x_0] - f[x_{k-1}, x_{k-2}, \ldots x_1, x_0]}{x_k - x_0} \quad (9)$$

Lagrange Interpolation Polynomial Method

Lagrange Interpolation is a way of finding the value of any function at any given point when the function is not given. We use other points on the function to get the value of the function at any required point.

Suppose we have a function $y = f(x)$ in which substituting the values of x gives different values of y. And we are given two points (x_1, y_1) and (x_2, y_2) on the curve then the value of y at x = a(constant) is calculated using Lagrange Interpolation Formula.

Lagrange Interpolation Formula

Given few real values $x_1, x_2, x_3, \ldots, x_n$ and $y_1, y_2, y_3, \ldots, y_n$ and there will be a polynomial P with real coefficients satisfying the conditions $P(x_i) = y_i, \forall\, i = (1, 2, 3, \ldots, n)$ and degree of polynomial P must be less than the count of real values i.e., degree$(P) < n$.

Lagrange Interpolation Formula for n^{th} Order

The Lagrange Interpolation formula for n^{th} degree polynomial is given below:

Lagrange Interpolation Formula for the n^{th} order is,

$$P(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)} y_0 + \frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)} y_1 + \frac{(x-x_0)(x-x_1)\ldots(x-x_n)}{(x_2-x_0)(x_2-x_1)\ldots(x_2-x_n)} y_2 + \cdots + \frac{(x-x_1)(x-x_2)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_{n-1})} y_n \quad (10)$$

Lagrange First Order Interpolation Formula

If the Degree of the polynomial is 1 then it is called the First Order Polynomial. Lagrange Interpolation Formula for 1st order polynomials is,

$$f(x) = \frac{(x-x_1)}{(x_0-x_1)} \times y_0 + \frac{(x-x_1)}{(x_1-x_0)} \times y_1$$

Lagrange Second Order Interpolation Formula

If the Degree of the polynomial is 2 then it is called Second Order Polynomial. Lagrange Interpolation Formula for 2nd order polynomials is,

$$f(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} \times y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \times y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \times y_2$$

Proof of Lagrange Theorem

Let's consider a nth-degree polynomial of the given form,

$$f(x) = A_0(x - x_1)(x - x_2)(x - x_3) \ldots (x - x_n) + A_1$$

$$(x - x_1)(x - x_2)(x - x_3) \ldots (x - x_n) + \ldots + A_{(n-1)}(x - x_1)(x - x_2)(x - x_3) \ldots (x - x_n)$$

Substituting observations x_i to get A_i

with $x = x_0$ then we get A_0

$$f(x_0) = y_0 = A_0(x_0 - x_1)(x_0 - x_2)(x_0 - x_3) \ldots (x_0 - x_n)$$

$$A_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)\ldots(x_0 - x_n)}$$

By substituting $x = x_1$ we get A_1

$$f(x_1) = y_1 = A_1(x_1 - x_0)(x_1 - x_2)(x_1 - x_3) \ldots (x_1 - x_n)$$

$$A_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_1)(x_1 - x_2)\ldots(x_1 - x_n)}$$

Similarly, by substituting $x = x_n$ we get A_n

$$f(x_n) = y_n = A_n(x_n - x_0)(x_n - x_1)(x_n - x_2) \ldots (x_n - x_{n-1})$$

$$A_n = \frac{y_n}{(x_n - x_0)(x_n - x_1)(x_n - x_2)\ldots(x_n - x_{n-1})}$$

If we substitute all values of A_i in function $f(x)$ where $i = 1, 2, 3, \dots n$ then we get Lagrange Interpolation Formula as,

$$P(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)}y_1 + \frac{(x-x_0)(x-x_1)\dots(x-x_n)}{(x_2-x_0)(x_2-x_1)\dots(x_2-x_n)}y_2 + \dots + \frac{(x-x_1)(x-x_2)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_{n-1})}y_n$$

Problems and Numerical Results

We now apply Newton and Lagrange interpolation polynomial method to solve sample initial value problems of the first order ordinary differential equations.

Problem 1:

Solve

$$\frac{dy}{dx} = 1 - y$$
$$y(0) = 0$$

Taking step $h = 0.1$

Using Newton's interpolation,

$a_0 = 0 = y_0$

$$a_1 = \frac{f(x_1)-f(x_0)}{(x_1-x_0)} = \left[\frac{dy}{dx}\right]_{0,0} = 1$$

Using eqn. (2)

$y_1 = 0 + 1(0.1 - 0) = 0.1$

Using eqn. (7)

$$a_2 = \frac{\frac{f(x_2)-f(x_1)}{(x_2-x_1)} - \frac{f(x_1-x_0)}{x_1-x_0}}{(x_2-x_0)} = \frac{\left[\frac{dy}{dx}\right]_{0.1,0.1} - \left[\frac{dy}{dx}\right]_{0,0}}{0.2-0} = 0.549999$$

Using eqn. (3)

$$y_2 = 0 + 1(0.2-0) + 0.49999(0.2-0)(0.2-0.1) = 0.1998$$

By Eqn. (8), we have

$$a_3 = \frac{\frac{\frac{f(x_3)-f(x_2)}{(x_3-x_2)} - \frac{f(x_2)-f(x_1)}{(x_2-x_1)}}{(x_3-x_1)} - \frac{\frac{f(x_2)-f(x_1)}{(x_2-x_1)} - \frac{f(x_1)-f(x_0)}{(x_1-x_0)}}{(x_2-x_0)}}{(x_3-x_0)}$$

$$= \frac{\frac{\left[\frac{dy}{dx}\right]_{0.2,0.199} - \left[\frac{dy}{dx}\right]_{0.1,0.1}}{0.3-0.1} - \frac{\left[\frac{dy}{dx}\right]_{0.1,0.1} - \left[\frac{dy}{dx}\right]_{0,0}}{0.2-0}}{0.3-0} = 0.16667$$

$$y_3 = 0 + 1(0.3-0) + 0.49999(0.3-0)(0.3-0.1) + 0.16667(0.3-0)(0.3-0.1)(0.3-0.2) = 0.0271$$

Applying (0,1), (0.1, 0.100000), (0.2, 0.199000), and (0.3, 0.271000) to find the cubic polynomial by Eqn. (10).

Forming quadratic equation using Lagrange polynomial

$$P(x) = \frac{(x-0.1)(x-0.2)(x-0.3)}{(0-0.1)(0-0.2)(0-0.3)} \times 0 + \frac{(x-0)(x-0.2)(x-0.3)}{(0.1-0)(0.1-0.2)(0.1-0.3)} \times 0.1 + \frac{(x-0)(x-0.1)(x-0.3)}{(0.2-0)(0.2-0.1)(0.2-0.3)} \times 0.1998 + \frac{(x-0)(x-0.1)(x-0.2)}{(0.3-0)(0.3-0.1)(0.3-0.2)} \times 0.0271$$

$$P_3 = 0.16667x^3 - 0.549997x^2 + 1.053333x$$

The equation is used to get the values for y at any given value of x

Table 1: The table showing results of the equation
$$\frac{dy}{dx} = 1 - y$$

X	NIPM and LIPM Method	Exact values	Percentage Error
0	0	0	0%
0.1	0.100000	0.095163	5.082858%
0.2	0.199000	0.181282	9.773723%
0.3	0.271000	0.259215	4.546419%
0.4	0.344000	0.329717	4.331897%
0.5	0.410000	0.393490	4.195786%
0.6	0.470000	0.451186	4.169899%
0.7	0.525002	0.503399	4.291427%
0.8	0.576003	0.550681	4.598307%
0.9	0.624004	0.593488	5.141806%
1.0	0.670006	0.632211	5.978226%

Problem 2

Consider the differential equation

$$\frac{dy}{dx} = x^2 - y$$

with initial conditions $y(0) = 1$

we will take step size $h = 0.1$

Using Newton Interpolation:

$a_0 = 1$

$y_0 = 1$

$$a_1 = \left[\frac{dy}{dx}\right]_{0,1} = -1$$

$y_1 = 1 - 1(0.1 - 0) = 0.9$

$$a_2 = \frac{\frac{f(x_2)-f(x_1)}{(x_2-x_1)} - \frac{f(x_1-x_0)}{x_1-x_0}}{(x_2-x_0)} = \frac{\left[\frac{dy}{dx}\right]_{0.1,0.9} - \left[\frac{dy}{dx}\right]_{0,1}}{0.2-0} = 0.549999$$

$y_2 = 1 - 1(0.2 - 0) - 0.549999(0.2 - 0)(0.2 - 0.1) = 0.811$

$$a_3 = \frac{\frac{\frac{f(x_3)-f(x_2)}{(x_3-x_2)} - \frac{f(x_2)-f(x_1)}{(x_2-x_1)}}{(x_3-x_1)} - \frac{\frac{f(x_2)-f(x_1)}{(x_2-x_1)} - \frac{f(x_1)-f(x_0)}{(x_1-x_0)}}{(x_2-x_0)}}{(x_3-x_0)}$$

$$= \frac{\frac{\left[\frac{dy}{dx}\right]_{0.2,0.811} - \left[\frac{dy}{dx}\right]_{0.1,0.9}}{0.3-0.1} - \frac{\left[\frac{dy}{dx}\right]_{0.1,0.9} - \left[\frac{dy}{dx}\right]_{0,1}}{0.2-0}}{0.3-0} = 0.149999$$

$y_3 = 1 - 1(0.3 - 0) - 0.549999(0.3 - 0)(0.3 - 0.1) + 0.149999(0.3 - 0)(0.3 - 0.1)(0.3 - 0.2) = 0.7339$

Applying (0, 1), (0.1, 0.900000), (0.2, 0.811000), and (0.3, 0.733900) to find the cubic polynomial by Eqn. (10).

Forming quadratic using Lagrange

$P(x) = \frac{(x-0.1)(x-0.2)(x-0.3)}{(0-0.1)(0-0.2)(0-0.3)} \times 1 + \frac{(x-0)(x-0.2)(x-0.3)}{(0.1-0)(0.1-0.2)(0.1-0.3)} \times 0.9 + \frac{(x-0)(x-0.1)(x-0.3)}{(0.2-0)(0.2-0.1)(0.2-0.3)} \times 0.811 + \frac{(x-0)(x-0.1)(x-0.2)}{(0.2-0)(0.2-0.1)(0.2-0.3)} \times 0.7339$

$y_n = 0.149999x^3 + 0.505x^2 + 1.052x + 1$

Table 2: The table showing results of the equation: $\frac{dy}{dx} = x^2 - y$

x	NIPM and LIPM Method	Exact values	Percentage Error
0	1.000000	1.000000	0%
0.1	0.900000	0.905163	-0.570395%
0.2	0.811000	0.821212	-1.243528%
0.3	0.733900	0.749005	-2.016675%
0.4	0.669599	0.689391	-2.870940%
0.5	0.618999	0.643129	-3.751969%
0.6	0.582999	0.610887	-4.565165%
0.7	0.562499	0.593241	-5.182042%
0.8	0.558399	0.590676	-5.464417%
0.9	0.571599	0.603586	-5.299493%
1.0	0.602999	0.632280	-4.631017%

Problem 3:
Consider the differential equation

$$\frac{dy}{dx} = \frac{x-y}{e^{x+y}}$$

with initial $y(0) = 1$.
we will take step size $h = 0.1$
Using Newton interpolation:

$$a_0 = 1$$

$y_0 = 1$

$a_1 = \left[\frac{dy}{dx}\right]_{0,1} = -0.367$

$y_1 = 1 - 0.367(0.1 - 0) = 0.963212$
Eqn. 3.21;

$a_2 = \frac{\frac{f(x_2)-f(x_1)}{(x_2-x_1)} - \frac{f(x_1-x_0)}{x_1-x_0}}{(x_2-x_0)} = \frac{\left[\frac{dy}{dx}\right]_{0.1,0.963212} - \left[\frac{dy}{dx}\right]_{0,1}}{0.2-0} =$ 0.344504

$y_2 = 1 - 0.367(0.2 - 0) + 0.344504(0.2 - 0)(0.2 - 0.1) = 0.933401$

a_3
$= \frac{\frac{\frac{f(x_3)-f(x_2)}{(x_3-x_2)} - \frac{f(x_2)-f(x_1)}{(x_2-x_1)}}{(x_3-x_1)} - \frac{\frac{f(x_2)-f(x_1)}{(x_2-x_1)} - \frac{f(x_1)-f(x_0)}{(x_1-x_0)}}{(x_2-x_0)}}{(x_3-x_0)}$
$= \frac{\frac{\left[\frac{dy}{dx}\right]_{0.2,0.9334901} - \left[\frac{dy}{dx}\right]_{0.1,0.963212}}{0.3-0.1} - \frac{\left[\frac{dy}{dx}\right]_{0.1,0.963212} - \left[\frac{dy}{dx}\right]_{0,1}}{0.2-0}}{0.3-0} =$
0.16665

$y_3 = 1 - 0.367(0.3 - 0) + 0.344504(0.3 - 0)(0.3 - 0.1) + 0.0115180(0.3 - 0)(0.3 - 0.1)(0.3 - 0.2) = 0.909791$

Applying (0,1), (0.1, 963212), (0.2, 0.933401), and (0.3, 0.909791) to find the cubic polynomial by Eqn. (10)

Forming quadratic using Lagrange

$P(x) = \frac{(x-0.1)(x-0.2)(x-0.3)}{(0-0.1)(0-0.2)(0-0.3)} \times 1 + \frac{(x-0)(x-0.2)(x-0.3)}{(0.1-0)(0.1-0.2)(0.1-0.3)} \times 0.963212 + \frac{(x-0)(x-0.1)(x-0.3)}{(0.2-0)(0.2-0.1)(0.2-0.3)} \times 0.933401 + \frac{(x-0)(x-0.1)(x-0.2)}{(0.3-0)(0.3-0.1)(0.3-0.2)} \times 0.909791$

$y_n = -0.129608x^3 + 0.387751x^2 - 0.405358x + 1$

Table 3: The table showing results of the equation: $\frac{dy}{dx} = \frac{x-y}{e^{x+y}}$

x	NIPM and LIPM Method	Exact values	Percentage Error
0	1.000000	1.000000	0%
0.1	0.963212	0.966759	-0.366895%
0.2	0.933401	0.940092	-0.711738%
0.3	0.909791	0.919197	-1.023284%
0.4	0.891602	0.903288	-1.293718%
0.5	0.878057	0.891643	-1.523703%
0.6	0.868380	0.883609	-1.723499%
0.7	0.861791	0.878595	-1.912599%
0.8	0.857514	0.876075	-2.118654%
0.9	0.854771	0.875591	-2.377822%
1.0	0.852784	0.876747	-2.733172%

In this study, Newton and Lagrange interpolation polynomial method are used to solve initial value problems for first order ordinary differential equation and we constructed cubic polynomials from the method as the solutions of linear and non-linear ordinary differential equations. We compared our numerical results with the results of exact solutions for the problems considered.
The initial value problems and tables above of problem 1 and 2, helps to compare the numerical results with analytical method. Our method gives numerical

approximate solutions which is also closer to the exact solutions as expressed in Tables 1 and 2.
While finding solution, we observe that this method aids faster convergence even as we repeat the calculations for larger step sizes. Also, this proposed method can capture the local curvature of the function allowing it to fit better between data points.

This method gives results very close to the exact value. This is noted by the percentage error that is very minor. The method is very accurate and easy to use after getting the cubic polynomial equation. Hence, one can get the value of y at any value of x without necessarily getting preceding values of y. Numerical results show that our proposed method has greater accuracy and remarkable performance as seen in the tables above. When observed closely, that the percentage error is small with respect to the exact values.

CONCLUSION

In conclusion, our approach represents a step forward in the field of numerical methods for solving differential equations. It addresses the time-consuming and cumbersome nature of traditional techniques by introducing adaptability, efficiency, and enhanced accuracy. With its broad applicability, we provide a valuable tool that can be employed to solve various systems of differential equations, making it particularly useful in scientific research and engineering projects.

REFERENCES

Alkasassbeh, M., Omar, Z., Mebarek-Oudina F, Raza, J., and Chamkha, A. (2019). Heat Transfer Study of Convective Fin with Temperature-dependent Internal Heat Generation by Hybrid Block method. *Heat Transfer*, 48(4):1225–1244

Arqub, O. A. (2016). The Reproducing Kernel Algorithm for Handling Differential Algebraic Systems of Ordinary Differential Equations. *Math. Methods in the App. Sci*., 39(15):4549– 4562.

Khan, U., Mebarek-Oudina, F., Zaib, A., Ishak, A., Bakar, S. A., Sherif, E. M., and Baleanu, D. (2022). An Exact Solution of a Casson Fluid Flow Induced by Dust Particles with Hybrid Nanofluid over a Stretching Sheet Subject to Lorentz Forces. *Waves in Random and Complex Media*, 32.

Raza, J., Mebarek-Oudina, F, and Ali, L. L. (2020). *The flow of Magnetized Convective Casson Liquid via a Porous Channel with Shrinking and Stationary walls*. Pramana, 96.

Singh, I. and Singh, G. (2019). *Numerical Solution of First Order Ordinary Differential Equations*. Int. J. Sci. Res. Rev. 8:593–602.