NIPES
Journal of Science and Technology Research
Journal homepage: www.nipesjournals.org.ng

Modified Milne's Method for Solving IVP Using Matlab

***Subhi Abdalazim Aljily Osman*[1]*, Abdel Radi Abdel Rahman Abdel Gadir Abdel Rahman*[2]*, Arafat Abdelhameed Abdelrahman*[3] *and Ragaa Mohammed Haj Ibrahim*[4]**

[1]Department of Mathematics, Faculty of Computer Science and Information Technology, University of Abutana, Sudan.
[2] Department of Mathematics, Faculty of Education, Omdurman Islamic University, Omdurman, Sudan.
[3]Department of Mathematics, Faculty of Sciences and Technology, Omdurman Islamic University, Omdurman, Sudan
[4]Department of Mathematics, Faculty of Education, ElzaiemElAzhary University, Omdurman, Sudan

Article Info

Keywords:
Application,Numerical Method,Ordinary, Differential Equations ,Milne's Method

Received 2 September 2021
Revised 27 October 2021
Accepted 23 November 2021
Available online 12 December 2021

doi Crossref NIPES JOURNALS

https://doi.org/10.37933/nipes/3.4.2021.7

https://nipesjournals.org.ng
© 2021 NIPES Pub. All rights reserved.

Abstract

The solution of an initial value problem in ordinary differential equations expanded as Milne's method has been given as both a classical and a numerical method. In this paper, we first describe methods for solving initial value problems and then explain the Milne's method. When considering the numerical solution of ordinary differential equations (ODEs), a predictor–corrector method typically uses an explicit method for the predictor step and an implicit method for the corrector step. Our aims are proposing a modified form of the Milne's Predictor-Corrector formula for solving ordinary differential equation of first order and first degree. We followed the applied mathematical method numerically by MATLAB. We found that Milne's (modified) predictor-corrector formula gives better accuracy and it also can minimize the calculating time as it takes less number of iterations. In addition, toits a multi-step method to compute y for preceding values of y and y' is essentially required.

1. Introduction

Differential equations are among the most important mathematical tools used in producing models in the physical sciences, biological sciences and engineering **[1].**Differential equations are commonly used for mathematical modeling in since and engineering often there is no known analytic solution and numerical approximation are required **[2].** The core of the MATLAB system implements a set of functions to cope with some classical numerical problems. Although there is no need for a really deep knowledge of numerical analysis in order to use MATLAB, a grasp of the basics is useful in order to choose among competing methods and to understand what may go wrong with them. In fact, numerical computation is affected by machine precision and error propagation, in ways that may result in quite unreasonable outcomes. Hence, we begin by considering the effect of finite precision arithmetic and the issues of numerical instability and problem conditioning **[3].** Numerical differentiation is not a particularly accurate process. It suffers from a conflict between round off errors (due to limited machine precision) and errors inherent in interpolation. For this reason, a derivative of a function can never be computed with the same precision as the function itself **[4].** This paper used Milne's method that can be used to obtain approximate solutions of differential equations. Such approximations are necessary when no exact solution can be found **[5].**

1.1 Existence and Uniqueness of Solutions

This is the time to acknowledge that we have been avoiding a very important question: When we're trying to solve a differential equation, how do we know whether there is a solution? We could be looking for something that doesn't exist a waste of time, effort, and computer resources. Calculators and computers can mislead. They may present us with a solution where there is none. If there are several possible solutions, our user-friendly device may make its own selection, whether or not it is the one that we want for our problem. A skeptical attitude and a knowledge of mathematical theory will protect us against inappropriate answers. First, let's look at what can happen when we try to solve first-order initial-value problems. Then we'll discuss an important result guaranteeing when such IVPs have one and only one solution **[6].**

1.2 General Initial Value Ordinary Differential Equation (ODE) Problem

The general problem for a single initial-value ODE is simply stated as

$$\frac{dy}{dt} = f(y,t) \quad y(t_0) = y_0 \qquad (1)$$

y=dependent variable
t =independent variable
$f(y,t)$=derivative function
t_0 =initial value of the independent variable
y_0= initial value of the dependent variable
Equations (1) be termed a 1×1problem (one equation in one unknown). The solution of this 1×1problem is the dependent variable as a function of the independent variable, $y(t)$(this function when substituted into Equations(1) satisfies these equations). This solution may be a mathematicalfunction, termed an analytical solution **[7].**

1.3 Dependence on the Initial Value

Consider the initial value problem

$$\begin{cases} x' = f(t,x) \\ x(t_0) = s \end{cases}$$

Where the initial value is denoted by s instead of x_0, t_0emphasize that we allow now s to vary. Hence, the solution is can be considered as a function of two variables: $x = x(t,s)$our aim is to investigate the dependence on s.

As before, assume that f is continuous in an open set $\Omega \subset R^2$and is locally $Lipschitz$ in this set in x. Fix a point $(t_0, x_0)\epsilon\Omega$ and let ε, δ, L be the parameters from the local $Lipchitz$ condition at this point, that is, the rectangle

$$R = [t_0 - \delta, t_0 + \delta] \times [x_0 - \varepsilon, x_0 + \varepsilon]$$

Is contained in Ω and for all $(t,x), (t,y) \epsilon R$

$$|f(t,x) - f(t,y)| \leq L|x - y|$$ **[8]**

2. Numerical Methods

2.1 Milne's Method

Consider the first order differential equation

$$\frac{dy}{dx} = f(x,y) \quad y(x_0) = y_0$$

Newton's forward difference formula can be written as

$$f(x,y) = f_0 + n\Delta f_0 + \frac{n(n-1)}{2}\Delta^2 f_0 + \frac{n(n-1)(n-2)}{3!}\Delta^3 f_0 + \cdots \qquad (2)$$

Substituting this in the relation

$$y_4 = y_0 + \int_{x_0}^{x_0+4h} f(x,y)\,dx$$

we get

$$y_4 = y_0 + \int_{x_0}^{x_0+4h} [f_0 + n\Delta f_0 + \frac{n(n-1)}{2}\Delta^2 f_0 + \cdots]\,dx$$

Let $x = x_0 + nh$Therefore

$$y_4 = y_0 + h\int_0^4 [f_0 + n\Delta f_0 + \frac{n(n-1)}{2}\Delta^2 f_0 + \cdots]\,dn$$

$$= y_0 + h\left[4f_0 + 8\Delta f_0 + \frac{20}{3}\Delta^2 f_0 + \frac{8}{3}\Delta^3 f_0 + \cdots\right]$$

$$= y_0 + h\left[4y_0' + 8(E-1)y_0' + \frac{20}{3}(E^2 - 2E + 1)y_0' + \frac{8}{3}(E^3 - 3E^2 + 3E + 1)y_0'\right]$$

(neglecting fourth and higher order differences)

$$= y_0 + h\left[4y_0' + 8(y_1' - y_0') + \frac{20}{3}(y_2' - 2y_1' + y_0') + \frac{8}{3}(y_3' + -3y_2' + 3y_1' + y_0')\right]$$

$$= y_0 + h\left[\frac{8}{3}y_1' + \frac{4}{3}y_2' + \frac{8}{3}y_3'\right]$$

$$y_4 = y_0 + \frac{4h}{3}[2y_1' - y_2' + 2y_3']$$

If $x_0, x_1, \ldots, x_4$are any 5 consecutive values of x, then the above equation can be written as

$$y_{n+1,p} = y_{n-3} + \frac{4h}{3}[2y_{n-2}' - y_{n-1}' + 2y_n']$$

This is called Milne's Predictor formula (the subscript p indicates that it is a predicted value). This formula can be used to predict the value of $y_4$when those of y_0, y_1, y_2, y_3 are known. To get a corrector formula we substitute Newton's formula (2) in the relation

$$y_2 = y_0 + \int_{x_0}^{x_0+2h} f(x,y)\,dx$$

and we get

$$y_2 = y_0 + \int_{x_0}^{x_0+2h} [f_0 + n\Delta f_0 + \frac{n(n-1)}{2}\Delta^2 f_0 + \cdots]\,dx$$

$$y_2 = y_0 + h\int_0^4 [f_0 + n\Delta f_0 + \frac{n(n-1)}{2}\Delta^2 f_0 + \cdots]\,dn \text{ putting x } = x_0 + nh$$

$$= y_0 + h\left[2f_0 + 2\Delta f_0 + \frac{1}{3}\Delta^2 f_0 + \cdots\right]$$

$$= y_0 + h\left[2y_0' + 2(E-1)y_0' + \frac{1}{3}(E^2 - 2E + 1)y_0'\right]$$

(neglecting higher order differences)

$$= y_0 + h\left[2y_0' + 2(y_1' - y_0') + \frac{1}{3}(y_2' - 2y_1' + y_0')\right]$$

Thus

$$= y_0 + h\left[\frac{8}{3}y_1' + \frac{4}{3}y_2' + \frac{8}{3}y_3'\right]$$

$$y_2 = y_0 + \frac{h}{3}[y_0' + 4y_1' + y_2']$$
$$y_4 = y_0 + \frac{4h}{3}[2y_1' - y_2' + 2y_3']$$

If x_0, x_1, x_2 are any three corrective values of x, the above relation can be written as

$$y_{n+1,c} = y_{n-1} + \frac{h}{3}[y_{n-1}' + 4y_n' + y_{n+1}']$$

This is known Milne's corrector formula where the suffix c stands for corrector
an improved value of y_{n+1}' is computed and again the corrector formula is applied until we get y_{n+1} to the derived accuracy **[9].**

Milne's method also requires prior knowledge of several values of . It uses the predictor−corrector pair

$$y_{n+1} = y_{n-3} + \frac{4}{3}h[2f_n - f_{n-1} + 2f_{n-2}] \qquad (3)$$

and

$$y_{n+1} = y_{n-1} + \frac{1}{3}h[2f_{n+1} + 4f_n + f_{n-1}] \qquad (4)$$

The corrector formula of (2) serves as a check for the value

$$y_{n+1} = f(x_{n+1}, y_{n+1}) \qquad (5)$$

If y_{n+1} *and* y_{n+1} in (3) and (4) respectively, do not differ considerably, we accept y_{n+1} as the best approximation. If they differ significantly, we must reduce the interval h.

Example (1):

Use Milne's method to find the value of y corresponding to $x = 0.6$ for the differential equation

$$y' = 2y + x \qquad (6)$$

with the initial condition $y(0) = 1$.

Solution:

This is the same differential equation as in Example (1) where we found the following values:

n	x_n	y_n	$f_n = x_n + 2y_n$
2	0.2	1.5146	3.2292
3	0.3	1.8773	4.0546
4	0.4	2.3313	5.0626
5	0.5	2.8969	6.2938

and using the predictor formula we find

$$y_6 = y_2 + \frac{4}{3}(0.1)[2f_5 - f_4 + 2f_3]$$
$$= 1.5146 + \frac{4}{3} \times 0.1(2 \times 6.2938 - 5.0626 + 2 \times 4.0546)$$
$$= 3.599 \qquad (7)$$

Before we use the corrector formula of (4), we must find the value of f_6 this is found from

$$f_6 = x_6 + 2y_6$$

where $x_6 = 0.6$, and from example $y_6 = 3.599$. Then,

$$f_6 = x_6 + 2y_6$$
$$= 0.6 + 2 \times 3.599$$
$$= 7.7984$$

and

$$y_6 = y_4 + \frac{4}{3} \times 0.1(f_6 - 4f_5 + f_4)$$

$$y_6 = 2.3313 + \frac{4}{3} \times 0.1(7.7984 + 4 \times 6.2938 + 5.0626)$$
$$= 3.599 \qquad (8)$$

We see from (6) and (8) that the predictor−corrector pair is in very close agreement.
Milne's method can also be extended to second order differential equations of the form $y'' = (x, y, y')$ with initial conditions $y(x_0) = y_0$and $'(x_0) = y'_0$[5] .

2.2 Milne Method Code

```
m=inputdlg('Enter the 1st order ODE or slope function:');
s = m{:};
d =str2func(['@(x,y)' s]);
f =str2func(['@(x,y)'vectorize(s)]);
x =input('Select the initial x value:');
y =input('Select the initial y value:');
h =input('Select the increment value:');
order= 0;
order2=0;
n=0;
term=0;
while order= =0
order=input('\nSelect what order of Runge-Krutta to use (1 is min. and 5 is max.):');
if order<1||order>5
fprintf('Wrong input! The value must be between 1 and 5 only.\n');
order=0;
end
end
if order==2
while order2==0
order2=input('Choose "1" for Heun method, "2" for Midpoint method, and "3" for Ralston
method:');
if order2<1||order2>3
fprintf('Wrong input! The value must be between 1 and 3 only.\n\n');
   order2=0;
end
end
end
while (term<1)||(term>2)
term=input('\nPlease choose what termination criteria you desire.\nPress "1" for a specified x
value and "2" for nth iteration:');
if term==1
val=input('The specified x value is:');
itr=(val-x)/h;
break;
end
if term==2
itr=input('The desired nth iteration (iterations will start from n=0) is:');
break;
end
fprintf('Wrong input! Please try again.\n\n');
```

```
end
fprintf('\n');
%This is for labels
q='itrn';
w='x value';
b='current y';
t='solved y';
o='slope value';
s='f_n';
k='      ';

a=[q,k,w,k,b,k,t,k,o,s];
disp(a);

fori=1:itr+1
Y(i)=y;
X(i)=x;
    l=x;
    p=y;
f_n(i)=x+2*y;
    x=x+(0.50*h);
    k1=f(l,p);
    y=p+(0.50*k1*h);

    k2=f(x,y);
    y=p-(k1*h)+(2*k2*h);
    x=l+h;
    k3=f(x,y);
fr=(k1/6)+(4*k2/6)+(k3/6);
    y=p+(h*fr);
C(i)=l;
V(i)=p;
fprintf('%2.0f %13.4f %13.4f %13.4f %17.4f   %17.4f \n ',n,l,p,y,fr,f_n(i));

    n=n+1;
end

for k=itr:-1:1
f_new(k)=f_n(k+1)-f_n(k);

end
for k=itr-1:-1:1
    f_new1(k)=f_new(k+1)-f_new(k);
end
for k=itr-2:-1:1
    f_new2(k)=f_new1(k+1)-f_new1(k);
end

fori=3:itr+1
F(i)=X(i)+2*V(i);
```

```
end
result=V(3)+(4/3)*0.1*(2*F(6)-F(5)+2*F(4));
result2=0.6+2*result;
Final=V(5)+(1/3)*0.1*(result2+4*F(6)+F(5));

plot(C,F, '+b-')
title('Milne method '),...
xlabel('x'), ylabel('f_n'), grid
```

Output:

```
>> milne
Select the initial x value:0
Select the initial y value:1
Select the increment value:0.1

Select what order of Runge-Krutta to use (1 is min. and 5 is max.):3

Please choose what termination criteria you desire.
Press "1" for a specified x value and "2" for nth iteration:1
The specified x value is:0.5

itrn      x value      current y      solved y      slope valuef_n
 0        0.0000       1.0000         1.2267          2.2667              2.0000
  1        0.1000        1.2267         1.5146          2.8790              2.5533
  2        0.2000        1.5146         1.8773          3.6269              3.2291
  3        0.3000        1.8773         2.3313          4.5403              4.0545
  4        0.4000        2.3313         2.8969          5.6559              5.0626
  5        0.5000        2.8969         3.5987          7.0184              6.2938
>>
```

Milne Graph:

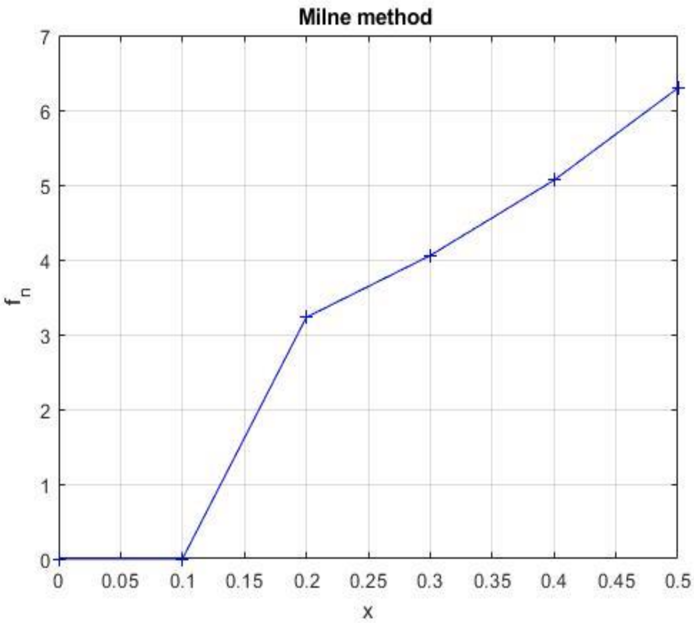

Fig1: Relation between values of x and f_n

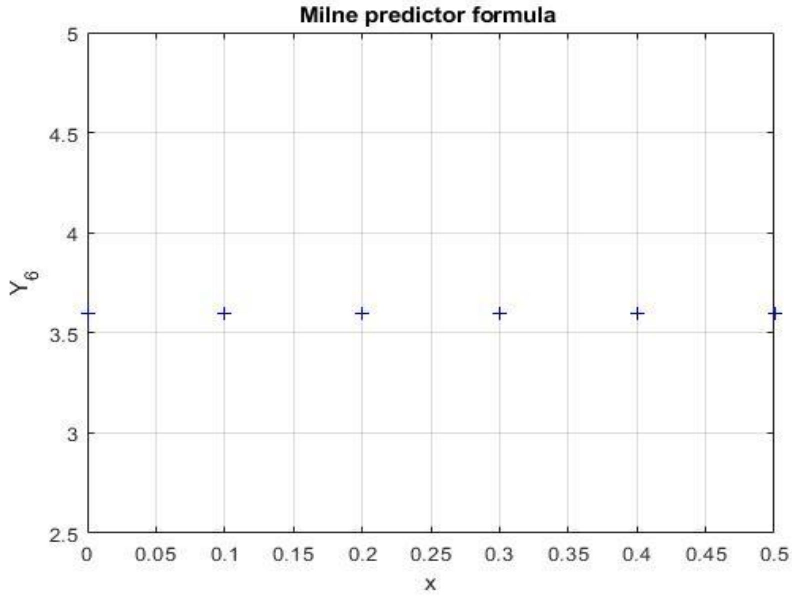

Fig 2: Milne predictor formula

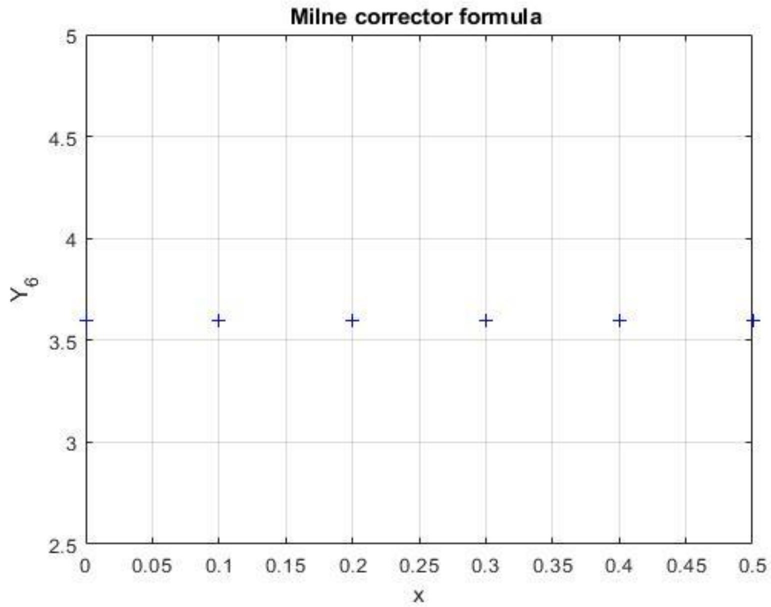

Fig 3: Milne Corrector Formula

3. Conclusion

The values of y_{n+1} computed by (2) may be called its predicted value and that computed by (3) is called the corrected value and respectively denoted by y_{n+1}^{p} and y_{n+1}^{c} . We have shown that Milne's (modified) predictor-corrector formulae gives better accuracy and it also can minimize the calculating time as it takes less number of iterations. It is a multi-step method i.e. to compute a knowledge of preceding values of y and y' is sent all y required.

References:

[1] Kendall atkinson, weimin han, david stewart, numerical solution of ordinary differential equations, a john wiley & sons, inc., publication published simultaneously in canada, 2009.
[2] John h. Mathews, kurtis d. Fink, numerical methods using matlab, third edition, prentice hall 1999.
[3] John wiley & sons, inc. All rights reserved numerical methods in finance and economics, second edition, printed in the united states of america, 2006.
[4] Jaankiusalaas, numerical methods in engineering with matlab, first published in print format 2005.
[5] Steven t. Karris, numerical analysis using matlab®and excel®, third edition, isbn-13: 978-1-934404-04-1, 2007
[6] Henry ricardo, amdern introduction to differentialequations,isbn 13: 978-0-12-374746-4,2009 .
[7] H.j. Lee and w.e. Schiesser, ordinary and partial differential equation routines in c, c++, fortran, java , maple , and matlab, copyright © 2004 by chapman & hall/crc.
[8] Alexander grigorian,ordinary differential equtions ,july 2008.
[9] P. Sam johnson, numerical solution of ordinary differential equations (part - 2), may 3, 2020.