Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Improved Predictor-corrector Method for the Initial-value Problem

Mohammad Al-Towaiq [1]

[1] *Jordan University of Science and Technology, Department of Mathematics and Statistics, Irbid, Jordan*

Abstract:- In this paper, we develop a predictor-corrector scheme based on a semi-open and closed-cotes quadrature process for solving an initial value problem of ordinary differential equation. The analysis showed that the method is stable, of order O(h5), and accurate. Numerical examples are given to demonstrate the validity and applicability of the proposed scheme. In addition, the numerical results show that the proposed scheme is very accurate and efficient.

Keywords: *Initial-Value Problems, Multistep Methods, Predictor-Corrector Methods, Numerical Solution.*

1. Introduction

Many real-life problems in applied science and engineering are modeled by a differential equation. Some of these problems need the solution of the Initial-value problems (IVP) of the form

$$x'(t) = f\big(t, x(t)\big), x(t_0) = x_0 \tag{1}$$

This type of equation often has difficulty to find an analytical solution, so the use of numerical scheme to find an approximate solution would be an excellent approach. Usually, there are two types of numerical schemes being used to solve equation (1). First, schemes called the one step methods since the solution at t_{i+1} involves information only from the one point t_i . Also, may these methods use functions evaluation at points in $[t_i , t_{i+1}]$, (e.g. see [1]-[9]). These methods are simple, but have relatively low accuracy. The other methods called multi-steps methods, where the integral of (1) can be computed over several intervals, so the solution will be available at some points before computing the solution at t_{i+1}. Integrate both sides of (1) over the interval $[t_{i-k} , t_{i+1}]$, we get

$$x_{i+1} = x_{i-k} + \int_{t_{i-k}}^{t_{i+1}} f\big(t,\ x(t)\big)dt. \tag{2}$$

Equation (2) called *k*-step method uses values of *x*(*t*) and *f*(*t*, *x*(*t*)) at *k* previous points t_{i-k}, *k* =0,1,2,…, *i*,

such as the well-known Adams-Moulton implicit method, the explicit Adams-Bashforth method, and other methods using some variation of Runge-Kutta method. For example, [10]-[13] presented what they call it "interval methods of Runge-Kutta type and multistep methods of Adams type". The authors concluded that the explicit interval methods are more accurate than Adams-Bashforth type. And the implicit of Milne-Simpson type has more accuracy than Adams-Moulton type.

The implicit methods have a weakness of converting it to explicit one, and this is not always possible. However, the implicit multi-step methods are used to improve the accuracy

of a multi-step method for the solution of equation (1) approximated by the explicit methods. This combination is called a predictor-corrector method, such as Milne-Simpson, Adams-Bashforth-Moulton, (e.g. see [14], [15]).

Recently, several methods were proposed for the solution of (1) (e.g. see [16]- [18]). In addition, many numerical techniques have been improved for solving fractional differential equations, such as, in [19], the authors used quadratic Lagrange interpolant to approximate the nonlinear part of Volterra integral equation using Adams type predictor corrector method. In [20] followed [19] by suggesting a higher order numerical scheme of the predictor corrector scheme for solving fractional differential equations using Lagrange interpolant to approximate the numerical part of Volterra integral. In [21], the author proposed a predictor corrector scheme for solving the IVP involving Caputo fractional derivative. For further reading see [22], [23].

In this work, we will develop a predictor-corrector method based on a semi-open and closed-cotes quadrature formula to improve the solution of (1).

The paper organized as follows; in section two we introduce the proposed method, in section three we give some numerical examples to demonstrate the efficiency and applicability of the method, we conclude the paper in section IV.

2. The proposed method

We begin by assuming that the solution of the initial value problem (1) exists and is unique.

To drive the proposed method, we follow the following procedures:

1. Integrate (1) over the interval $[t_{i-3}, t_{i+1}]$, that is

$$x_{i+1} = x_{i-3} + \int_{t_{i-3}}^{t_{i+1}} f\big(t, x(t)\big)dt \tag{3}$$

2. Assume that the $x(t)$ is known at the four equally spaced nodes $t_{i-3}, t_{i-2}, t_{i-1}$, and t_i, then we approximate the integral of (3), using a semi-open-cote quadrature formula

$$\int_{t_{i-3}}^{t_{i+1}} f\big(t, x(t)\big)dt \cong \sum_{k=i-3}^{i} w_k f(t_k, x_k) \tag{4}$$

Using Lagrange polynomials over the interval $[t_{i-3}, t_{i+1}]$ to estimate the weights, that is

$w_j = \int_{t_{i-3}}^{t_{i+1}} L_j(t)dt\,,\ j = i-3, \ldots, i,$ we obtain

$w_{i-3} = 2.2917h, w_{i-2} = 0.2083\,h, w_{i-1} = 0.2083\,h,$ and $w_i = 2.2917\,h.$

Substitute the weights in (4), then in (3), we obtain the predictor formula

$$\hat{x}_{i+1} = x_{i-3} + h[2.2917 f(t_i, x_i) + 0.2083 f(t_{i-1}, x_{i-1}) + 0.2083 f(t_{i-2}, x_{i-2}) + 2.2917 f(t_{i-3}, x_{i-3})]. \tag{5}$$

It is easy to see the formula is of order five ($O(h^5)$).

3. To improve the accuracy of the solution approximated by equation (4), we develop a corrector implicit multi-step method as follows:

   3.1. Integrate equation (1) over the interval $[t_{i-2}, t_{i+1}]$, we obtain

$$x_{i+1} = x_{i-2} + \int_{t_{i-2}}^{t_{i+1}} f\big(t, x(t)\big)dt \tag{6}$$

   3.2. Approximate the integral in (6), using a closed -cote quadrature formula

$$\int_{t_{i-2}}^{t_{i+1}} f\big(t, x(t)\big)dt \cong w_0 f(t_{i-2}, x_{i-2}) + w_1 f(t_{i-1}, x_{i-1}) + w_2 f(t_i, x_1) + w_3 f(t_{i+1}, x_{i+1}) \tag{7}$$

   3.3. We estimate the values of the weights using Lagrange polynomials over the interval $[t_{i-2}, t_{i+1}]$, we get $w_0 = 0.375h, w_1 = 1.125\,h, w_2 = 1.125h,$ and $w_3 = 0.375\,h.$

Now, substitute the weights in (7) then in (6), we obtain the corrector formula:

$$x_{i+1} = x_{i-2} + 0.375hf(t_{i+1}, \hat{x}_{i+1}) + 1.125\,hf(t_i, x_i) + 1.125hf(t_{i-1}, x_{i-1}) + 0.375\,hf(t_{i-2}, x_{i-2}) \tag{8}$$

You can see easily that the order term of this formula is five ($O(h^5)$).

4. Hence, the algorithm for finding an approximate solution for the IVP (1) at equally spaced n subintervals is as follows:

   - Start with initial condition x_0, then use one step method to calculate $x_1, x_2,$ and x_3
   - For i = 3, 4, 5, … do the following:

- Compute the predictor using (5).
- Compute the corrector using (8).

3. Numerical Demonstration

In this section, we consider some examples to illustrate the accuracy, efficiency, and to confirm the validity of our proposed technique (nppc).

Example 1: Consider the IVP, $x'(t) = \frac{1}{1+t^2} - 2x^2(t),\ x(0) = 0$ on the interval [0, 1]. The exact solution is

$x(t) = \frac{t}{1+t^2}$.

First, we used modified Euler's method to predicate $x_1, x_2, and\ x_3$, then we used Mathematica to generate x_i , $4 \leq i \leq 19$, using $h = 0.05$. The results shown in table 1 and figure 1.

Table 1. Comparison between the exact solution and the nppc solution of example 1

t_i	x_{i_exact}	x_{i_nppc}	Abs_Error
0	0	0	0
0.05	0.04987531172	0.04981265586	6.26558607*10^-5
0.1	0.09900990099	0.09888430787	1.255931201*10^-4
0.15	0.1466992665	0.1465112488	1.880177037*10^-4
0.2	0.1923076923	0.1921904729	1.172193654*10^-4
0.25	0.2352941176	0.2351149985	1.791191191*10^-4
0.3	0.2752293578	0.2751267151	1.02642712*10^-4
0.35	0.3118040089	0.3116387738	1.652351296*10^-4
0.4	0.3448275862	0.3447421199	8.546632525*10^-5
0.45	0.3742203742	0.3740702132	1.501610033*10^-4
0.5	0.4	0.3999325711	6.742889751*10^-5
0.55	0.4222648752	0.4221295246	1.353506175*10^-4
0.6	0.4411764706	0.4411266764	4.979417895*10^-5
0.65	0.4569420035	0.4568202402	1.217632899*10^-4
0.7	0.4697986577	0.4697652931	3.336460077*10^-5
0.75	0.48	0.4798900559	1.099440871*10^-4
0.8	0.487804878	0.4877863229	1.855513215*10^-5
0.85	0.4934687954	0.4933686826	1.001127399*10^-4
0.9	0.4972375691	0.497232077	5.492026496*10^-6
0.95	0.4993429698	0.4992507093	9.226043154*10^-5
1	0.5	0.5000058895	5.889488083*10^-6

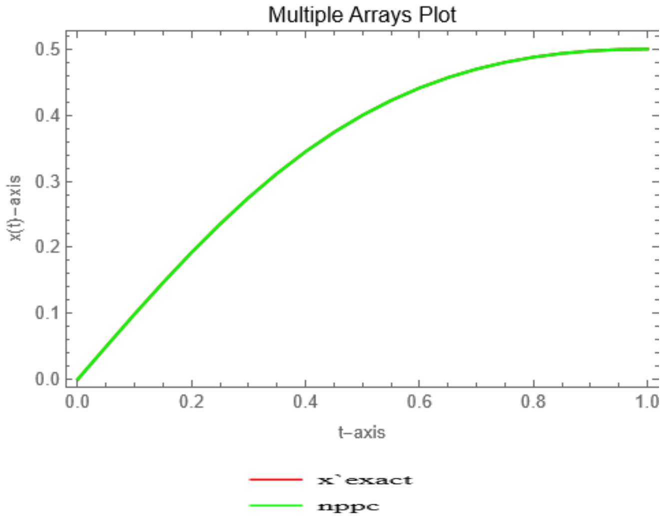

Fig. 1. Comparison between the exact solution and nppc of example 1

Example 2: Consider the following IVP

$x'(t) = (2-t)x(t), 2 \leq t \leq 3, x(2) = 1.$ The exact solution is $x(t) = e^{-0.5(t-2)^2}$.

We used the modified Euler's method to predicate $x_1, x_2, and\ x_3$, then we used Mathematica to generate x_i , $4 \leq i \leq 49$, using $h = 0.02$. The results for some selected points are shown in table 2 and figure 2.

Table 2. Comparison between the exact solution and the nppc solution of example 2

t_i	$x_{i-exact}$	x_{i-nppc}	Abs-Error
2	1.	1	0.
2.02	0.99980002	0.9998	1.999866672*10^{-8}
2.04	0.9992003199	0.99920028	3.991468367*10^{-8}
2.06	0.998201619	0.9982015539	6.512843731*10^{-8}
2.08	0.9968051145	0.9968050747	3.98321135*10^{-8}
2.1	0.9950124792	0.9950124141	6.50471168*10^{-8}
2.12	0.9928258579	0.9928258182	3.965951045*10^{-8}
2.18	0.9839305143	0.9839304496	6.465946256*10^{-8}
2.2	0.9801986733	0.9801986342	3.911012914*10^{-8}
2.22	0.9760904721	0.9760904078	6.438433386*10^{-8}
2.28	0.9615583782	0.9615583399	3.829224748*10^{-8}
2.3	0.9559974818	0.9559974182	6.367466476*10^{-8}
2.32	0.9500886338	0.950088596	3.778417179*10^{-8}
2.38	0.9303448082	0.9303447455	6.275750652*10^{-8}
2.4	0.9231163464	0.9231163098	3.657336101*10^{-8}
2.42	0.9155777429	0.9155776806	6.222334714*10^{-8}
2.48	0.8911878885	0.8911878534	3.510874003*10^{-8}
2.5	0.8824969026	0.8824968416	6.10092844*10^{-8}
2.52	0.873541186	0.8735411517	3.428378448*10^{-8}

2.58	0.8451847808	0.8451847212	$5.960972327*10^{-8}$
2.6	0.8352702114	0.835270179	$3.245476554*10^{-8}$
2.62	0.8251418236	0.8251417648	$5.884430454*10^{-8}$
2.64	0.8148102622	0.8148102307	$3.145363225*10^{-8}$
2.66	0.8042862828	0.8042862248	$5.803758951*10^{-8}$
2.68	0.793580734	0.7935807036	$3.039699059*10^{-8}$
2.7	0.7827045382	0.7827044811	$5.719168061*10^{-8}$
2.72	0.7716686739	0.7716686446	$2.928675435*10^{-8}$
2.74	0.7604841569	0.7604841006	$5.630889732*10^{-8}$
2.76	0.7491620228	0.7491619947	$2.81250343*10^{-8}$
2.78	0.737713309	0.7377132536	$5.539178372*10^{-8}$
2.8	0.7261490371	0.7261490102	$2.691414469*10^{-8}$
2.82	0.7144801955	0.714480141	$5.444311379*10^{-8}$
2.84	0.7027177229	0.7027176972	$2.565660484*10^{-8}$
2.86	0.6908724913	0.6908724379	$5.346589205*10^{-8}$
2.88	0.6789552903	0.6789552659	$2.43551378*10^{-8}$
2.9	0.6669768109	0.6669767584	$5.246335055*10^{-8}$
2.92	0.6549476305	0.6549476075	$2.301266455*10^{-8}$
2.94	0.6428781982	0.6428781468	$5.143894299*10^{-8}$
2.96	0.6307788205	0.6307787989	$2.163229551*10^{-8}$
2.98	0.6186596475	0.6186595971	$5.039633444*10^{-8}$
3	0.6065306597	0.6065306395	$2.021731604*10^{-8}$

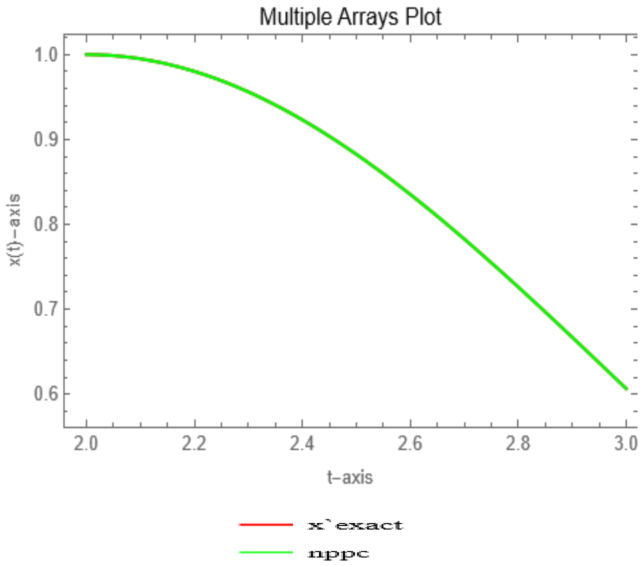

Fig. 2. Comparison of the exact solution and the nppc solution of example 2

The results of examples 1 and 2 shown the comparison of the absolute error of the exact solution and the proposed methods in tables 1 and 2, which indicates that our proposed scheme is efficient and accurate. Moreover, figures 1 and 2 show that the proposed scheme match with the exact solution.

4. Conclusion

We derive a predictor-corrector method for solving the IVP of ODEs based on a Gaussian quadrature process. We have shown that the derived method has a local truncation error of order five. We gave some numerical examples to demonstrate the efficiency, accuracy, and applicability of the proposed method.

References

[1] K. Gajda, A. Marciniak, B. Szyszka, Three- and Four-Stage Implicit Interval Methods of Runge-Kutta Type, Computational Methods in Science and Technology, 6, 41-59, (2000).

[2] A Marciniak, B. Szyszka, On Representations of Coefficients in Implicit Interval Methods of Runge-Kutta Type, Computational Methods in Science and Technology, 10(1), 57-71 (2004).

[3] A Marciniak, Implicit Interval Methods for Solving the Initial Value Problem, Numerical Algorithms, 37, 241-251(2004).

[4] M. Jankowska, A. Marciniak, Implicit Interval Multistep Methods for Solving the Initial Value Problem, Computational Methods in Science and Technology, 8(1), 17-30 (2002).

[5] Ehigie J.O., Jator S.N., Sofoluwe, A.B. and Okunuga, S.A. (2014) Boundary Value Technique for Initial Problems with Continuous Second Derivative Multistep Method of Enright. SIAM Journal on Numerical Analysis , 33, 81-93.

[6] R. I. Okuonghae and M. N. O. Ikhile, A Continuous Formulation of A(α)-Stable Second Derivative Linear Multistep Methods for Stiff IVPs in ODEs, Journal of Algorithms & Computational Technology, Vol. 6 No. 1, 2011, 79-100.

[7] Mehdiyeva, G.Y.; Ibrahimov, V.R. On the Research of Multi-Step Methods with Constant Coefficients; Lambert Academic Publishing: Saarbrücken, Germany, 2013; p. 314.

[8] M. Al-Towaiq and Osama Alayed, "AN EFFICIENT ALGORITHM BASED ON THE CUBIC SPLINE FOR THE SOLUTION OF BRATU-TYPE EQUATION", J. of Interdisciplinary Mathematics, V. 17 (2014) No. 5 & 6, pp 471-484.

[9] M. Al-Towaiq, Abdallah Obeidat, and Assma Al-Momani, "Clustered Diagnally Explicit Runge-Kutta Method for the Solution of Systems of Differential Equations", J. of Parallel and Cloud Computing, Vol. 2, Iss. 3, (2013), PP 58-64

[10] M. Jankowska, A. Marciniak, On Explicit Interval Methods of Adams-Bashforth Type, Computational Methods in Science and Technology, 8(2), 46-57 (2002).

[11] M. Jankowska, A. Marciniak, On Two Families of Implicit Interval Methods of Adams-Moulton Type, Computational Methods in Science and Technology, 12(2), 109-113 (2006).

[12] A Marciniak, B. Szyszka, One- and Two-Stage Implicit Interval Methods of Runge-Kutta Type, Computational Methods in Science and Technology, 5, 53-65 (1999).

[13] Gurjinder, S., Kanwar, V. and Saurabh, B. (2013) Exponentially Fitted Variants of the Two-Step Adams-Bashforth Method for the Numerical Integration of Initial Value Problem. Journal of Application and Applied Mathematics, 8, 741-755.

[14] Bolaji, B. Fully, Implicit Hybrid Block–Predictor Corrector Method for the Numerical Integration y ‴ = f (x, y, y′, y″), y (xo) = yo, y′(xo) = y1, y″(xo) =y3. J. Sci. Res. Rep. 2015, 6, 165–171.

[15] K. Diethelm, N. J. Ford, A. D. Freed, A predictor-corrector approach for the numerical solution of fractional differential equations. Nonlinear Dynamics, 29 (2002) 3-22.

[16] Ibrahimov, V.R.; Mehdiyeva, G.; Imanova, M. On the construction of the multistep Methods to solving for initial-value problem for ODE and the Volterra intego-differential equations. In Proceedings of the International Conference on Indefinite Applied Energy, Oxford, UK, 14–15 March 2019.

[17] Emmanuel A. Areo, Oluwatoyin A. Edwin, Multi-Derivative Multistep Method for Initial Value Problems Using Boundary Value Technique, Open Access Library Journal, 2020, Volume 7, e6063.

[18] Nur Auni Baharum, Zanariah Abdul Majid, Norazak Senu and Haliza Rosali Diagonally Multistep Block Method for Solving Volterra Integro-Differential Equation with Delay, Menemui Matematik (Discovering Mathematics), Vol. 45, No. 2: 208- 223 (2023).

[19] Nguyen, T. B., Jang, B. (2017). A high-order predictor-corrector method for solving nonlinear differential equations of fractional order, Fractional Calculus and applied analysis, 20(2), 447-476.

[20] S. Aljhani, Mohd Salmi Md Noorani, Redouane Douai, and Salem Abdelmalek, A high-order predictor-corrector method for initial value problems with fractional derivative involving Mittag-Le er kernel, Mathematical Methods in the Applied Sciences, January 22, 2023.

[21] Zaid Odibat, A universal predictor-corrector algorithm for numerical simulation of generalized fractional differential equations, Nonlinear Dyn (2021) 105:2363-2374.

[22] Douai, R., and Abdelmalek, S. (2019). A predictor-corrector method for fractional delay-differential system with multiple lags, Communications in Nonlinear Analysis, 6(1), 78-88.

[23] Douai, R., Bendoukha, S., Abdelmalek, S. (2021). A Newton interpolation-based predictor corrector numerical method for fractional differential equations with an activator inhibitor case study. Mathematics and Computers in Simulation, 187, 391-413.