RESEARCH REVIEW JOURNALS
Volume-04
Issue-04
April-2019

ISSN: 2455-3085 (Online)
RESEARCH REVIEW International Journal of Multidisciplinary
www.rrjournals.com[UGC Listed Journal]

# A Study of Numerical Method for Solution of Multistep Parameter

[1]Ashok Kumar Mishra and [2]Dr. Satendra Kumar

[1]Research Scholar, Opjs University, Churu Rajasthan
[2]Professor, Opjs University, Churu Rajasthan

**ARTICLE DETAILS**
**Article History**
*Published Online:* 15April2019

**Keywords**
Delay differential equations, Variable order variable step size, Multistep methods, Predictor–corrector, Stability regions.

**ABSTRACT**
This paper considers the numerical solution of delay differential equations. The predictor–corrector scheme based on generalized multistep methods are implemented in variable order variable step size techniques. The formulae are represented in divided difference form where the integration coefficients are computed by a simple recurrence relation. This representation produces simpler calculation as compared with the modified divided difference form, but no sacrifice is made in efficiency and accuracy of the method. Numerical results prove that the method is reliable, efficient and accurate. The P- and Q-stability regions for a fixed step size of the predictor–corrector scheme are illustrated for various orders.

## 1. Introduction

In this paper, we consider systems of first order delay differential equations (DDEs) of the form:

$$\begin{cases} y'(x) = f(x, y(x), y(x-\tau_1), y(x-\tau_2), \ldots, y(x-\tau_n)), & x \in [a, b], \tau_i > 0, \\ y(x) = \phi(x), & x \in [\bar{a}, a], \end{cases} \quad (1)$$

where φ(x) is the initial function and τi , for i = 1,..., n, where n is an integer such that n ≥ 1 are the lag functions and $\bar{a}$ = min$_{x\in[a,b]}$(x − τi). The lag can be constant, time dependent where τi = τi(x) or state dependent, that is τi = τi(x, y(x)). The expression y(x − τi) is the solution of the delay term, or simply called the delay term. The function f , where f : [a, b] × C([ $\bar{a}$, b], Rm) × C([ $\bar{a}$, b], Rm) × ⋯ × C([ $\bar{a}$, b], Rm) → Rm is continuous and satisfies a Lipschitz condition which guarantees the existence of a unique solution of (1). Here, C([ $\bar{a}$, b], Rm) denotes the space of continuous functions mapping [ $\bar{a}$, b] into Rm for an integer m ≥ 1.

DDEs play an important role in modeling many phenomena in applied sciences including the fields as diverse as engineering, biology and economy. For example, Baker et al. [3] stated some of the application areas which include population dynamics, infectious disease, physiological and pharmaceutical kinetics and chemical kinetics, the navigational control of ships and aircraft and more general control problems. More examples of the modeling equations involving DDEs can be referred to Driver [5]. Finding analytical solutions for DDEs is hard and at times impossible to do. Therefore, scientists resort to numerical solutions which can be made as accurately as possible. Numerical methods for solving DDEs are adapted from that of numerical methods for ordinary differential equations (ODEs). Most of the popular methods are linear multistep and Runge–Kutta type of methods. For some of the earlier works, refer to Jackiewicz and Lo [8,9], Jackiewicz [7], Kemper [10], Suleiman and Ismail [15] and Shampine and Thompson [13].

In order to achieve greater efficiency, most modern codes for solving differential equations are implemented in variable order and variable stepsize techniques. While varying the stepsize, the coefficients of the methods need to be recomputed at every step. This leads to huge computational effort and reduces the efficiency of the methods as more time is allocated for the process. Shampine and Gordon [12] implemented the Adams–Bashforth–Moulton predictor–corrector methods for systems of ODEs in modified divided difference form. Later, Jackiewicz and Lo [9] developed predictor–corrector formulae for neutral functional differential equations based on Adams–Bash forth–Moulton methods in modified divided difference form for efficient implementation.

## 2. Formulae derivation for the predictor and the corrector

For simplicity, we consider only single-delay scalar DDE. However, the results are also valid for systems of equations with multiple delays. Consider the DDE of the form:

$$\begin{cases} y'(x) = f(x, y(x), y(x-\tau)), & x \in [a, b], \tau > 0, \\ y(x) = \phi(x), & x \in [\bar{a}, a], \end{cases} \quad (2)$$

where φ is a continuous function. The non-uniform grid is given by $\bar{a}$ ≤ a = x0 < ⋯ <xn< xn+1 < ⋯ <xN = b. We denote by yh(x) the approximation to- y(x), where y is the solution of (2), and also denote fn = f (xn, yh(xn), yh(xn − τ )). The predicted value of yh(x) is denoted as ph(x). Assume that yh is already computed for x ∈ [ $\bar{a}$, xn]. The immediate task is to evaluate yh(xn+1) and the delay term yh(xn+1−τ ). The Adams methods are implemented in PECE mode where P stands for an application of a predictor, E

stands for an evaluation of a function f , and C stands for an application of a corrector. The predictor uses the Adams–Bashforth method of order k while the corrector uses the Adams–Moulton method of order k + 1. In PECE mode, a predictor of order k and a corrector of order k+1 leads to a predictor–corrector method of order k + 1. Meanwhile, if the predictor and the corrector are of order k, then the order of the predictor–corrector method is also k. By using the corrector of higher order, we gain the advantage of increasing the accuracy.

Let Pk,n be the interpolating polynomial of degree k − 1 interpolating f at points (xn, fn), (xn−1, fn−1), . . . , (xn−k+1, fn−k+1). In divided difference form, the interpolating polynomial can be written as

$$P_{k,n}(x) = f[x_n] + (x - x_n) f[x_n, x_{n-1}] + (x - x_n)(x - x_{n-1}) f[x_n, x_{n-1}, x_{n-2}] + \cdots + (x - x_n)(x - x_{n-1}) \ldots (x - x_{n-k+2}) f[x_n, x_{n-1}, \ldots, x_{n-k+1}]$$

Where,

$$f[x_n, x_{n-1}, \ldots, x_{n-k+1}] = \frac{f[x_n, x_{n-1}, \ldots, x_{n-k+2}] - f[x_{n-1}, x_{n-2}, \ldots, x_{n-k+1}]}{x_n - x_{n-k+1}}. \tag{3}$$

In [12] and [9], the divided difference is replaced by the modified divided difference ˆf [xn, xn−1,..., xn−k+1]
Where

$$\hat{f}[x_n, x_{n-1}, \ldots, x_{n-k+1}] = (x_n - x_{n-1})(x_n - x_{n-2}) \cdots (x_n - x_{n-k+1}) \times f[x_n, x_{n-1}, \ldots, x_{n-k+1}].$$

By using the modified divided difference, the general term of Pk,n,

$$(x - x_n)(x - x_{n-1}) \cdots (x - x_{n-k+2}) f[x_n, x_{n-1}, \ldots, x_{n-k+1}]$$

is written as

$$\frac{(x - x_n)(x - x_{n-1}) \cdots (x - x_{n-k+2})}{(x_n - x_{n-1})(x_n - x_{n-2}) \cdots (x_n - x_{n-k+1})} \hat{f}[x_n, x_{n-1}, \ldots, x_{n-k+1}]. \tag{4}$$

In order to calculate (xn − xn−1)(xn − xn−2)···(xn − xn−k+1) in (4), the stepsizes, hi = xi − xi−1 and sums of the stepsizes, ψi(n) = hn + hn−1 +···+ hn+1−i = xn − xn−i are introduced as required quantities. In order to calculate these quantities, other additional quantities are also introduced. These extra quantities create additional arithmetic operations that need to be performed. Detailed description of the implementation using the modified divided difference can be found in [12] and [9]. Meanwhile, when Pk,n is represented in divided difference form, (3) is conveniently computed using basic arithmetic operations consisting of two subtraction operations and a division.

Integrating (2) from xn to x, for xn< x ≤ xn+1 and replacing f with Pk,n yields

$$p_h(x) = y_h(x_n) + \int_{x_n}^{x} P_{k,n}(\xi) \mathrm{d}\xi$$

$$= y_h(x_n) + \sum_{i=0}^{k-1} f[x_n, x_{n-1}, \ldots, x_{n-i}] \int_{x_n}^{x} p_{n,i}(\xi) \mathrm{d}\xi$$

Where

$$p_{n,i}(x) = \begin{cases} 1, & i = 0, \\ (x - x_n)(x - x_{n-1}) \ldots (x - x_{n-i+1}), & i \geq 1. \end{cases}$$

The predicted values of the derivative is given by

$$p_h'(x) = \sum_{i=0}^{k-1} p_{n,i}(x) f[x_n, x_{n-1}, \ldots, x_{n-i}].$$

 

For the corrector value, consider the interpolating polynomial Pk+1,n+1 of degree k that interpolates *f* at points

$$(x_n, f_n), (x_{n-1}, f_{n-1}), \ldots, (x_{n-k+1}, f_{n-k+1}), (x_{n+1}, f^p_{n+1})$$

Where,

$$f^p_{n+1} = f(x_{n+1}, p_h(x_{n+1}), p_h(x_{n+1} - \tau)).$$

We use the notation $f^p[x_{n+1}, x_n, \ldots, x_{n-k+1}]$ to denote the divided difference obtained from the (x, f)-pairs listed above. Clearly the interpolating polynomial $P_{k+1,n+1}$ can be written as

$$P_{k+1,n+1}(x) = P_{k,n}(x) + (x - x_n)\cdots(x - x_{n-k+1}) f^p[x_{n+1}, x_{n-1}, \ldots, x_{n-k+1}].$$

Replacing *f* in (2) with the polynomial $P_{k+1,n+1}$ and integrating, the corrector formula is obtained as follows,

$$y_h(x) = p_h(x) + f^p[x_{n+1}, x_n, x_{n-1}, \ldots, x_{n-k+1}] \int_{x_n}^{x} p_{n,k}(\xi)\mathrm{d}\xi$$

And,

$$y'_h(x) = p'_h(x) + p_{n,k}(x) f^p[x_{n+1}, x_n, x_{n-1}, \ldots, x_{n-k+1}].$$

Let $p^{(-t)}_{n,i}(x)$ denote the $t$-fold integral of $p_{n,i}(x)$. Now define

$$g_{i,t}(x) = \begin{cases} p_{n,i}(x), & t = 0, \\ p^{(-t)}_{n,i}(x), & t \geq 1. \end{cases}$$

The formula for gi,t(x) can be obtained recursively by the relation

$$g_{i,t}(x) = (x - x_{n-i+1}) g_{i-1,t}(x) - t g_{i-1,t+1}(x), \qquad t \geq 1$$

With the adoption of the following notation,

$$y^{[0]}_h(x) = p_h(x) \quad \text{and} \quad z^{[0]}_h(x) = p'_h(x),$$

the PECE mode based on the Adams methods at the grid point for the numerical solution of Eq. (2) is defined by

$$\mathrm{P}: \begin{cases} y^{[0]}_h(x_{n+1}) = y_h(x_n) + \sum_{i=0}^{k-1} g_{i,1}(x_{n+1}) f[x_n, x_{n-1}, \ldots, x_{n-i}], \\ z^{[0]}_h(x_{n+1}) = \sum_{i=0}^{k-1} g_{i,0}(x_{n+1}) f[x_n, x_{n-1}, \ldots, x_{n-i}], \end{cases}$$

$$\mathrm{E}: \quad z^{[1]}_h(x_{n+1}) = f^{[1]}(x_{n+1}, y^{[0]}_h(x_{n+1}), y^{[0]}_h(\alpha)),$$

$$\mathrm{C}: \begin{cases} y^{[1]}_h(x_{n+1}) = y^{[0]}_h(x_{n+1}) + g_{k,1}(x_{n+1}) f^p[x_{n+1}, x_n, \ldots, x_{n-k+1}], \\ z^{[1]}_h(x_{n+1}) = z^{[0]}_h(x_{n+1}) + g_{k,0}(x_{n+1}) f^p[x_{n+1}, x_n, \ldots, x_{n-k+1}], \end{cases}$$

$$\mathrm{E}: \quad z^{[2]}_h(x_{n+1}) = f^{[2]}(x_{n+1}, y^{[1]}_h(x_{n+1}), y^{[1]}_h(\alpha)),$$

 

where α = xn+1 − τ . The numerical solution is defined by

$$y_h(x_{n+1}) = y_h^{[1]}(x_{n+1}).$$

Let *e* be defined as the difference between subsequent iteration values of the derivative at xn+1. That is,

$$e = z_h^{[1]}(x_{n+1}) - z_h^{[0]}(x_{n+1}).$$

Therefore,

$$e = g_{k,0}(x_{n+1}) f^p[x_{n+1}, x_n, \ldots, x_{n-k+1}]. \qquad (5)$$

From Eq. (5), the corrector formulae can be written as

$$\mathrm{C}: \begin{cases} y_h^{[1]}(x_{n+1}) = y_h^{[0]}(x_{n+1}) + g_{k,1}(x_{n+1}) \dfrac{e}{g_{k,0}(x_{n+1})}, \\ z_h^{[1]}(x_{n+1}) = z_h^{[0]}(x_{n+1}) + e. \end{cases} \qquad (6)$$

The solution of the delay term is denoted by yh(α). We now describe how the prediction and the correction of the delay term y[s] h (α),s = 0, 1 are being carried out. Also note that yh(α) = y[1] h (α). The value of α is known since α = xn+1 −τ . We seek the exact interval that contains α. Thus, the value of j such that xj< α ≤ xj+1 is determined because the calculation of the delay term depends upon this interval.

**3. Local error estimation and the order and step size changing strategy**

A reliable way to control the local error at the grid point is to compare the formulae of different orders. Following the discussion in Suleiman [14], the local error at the grid point E¯k can be estimated as

$$\bar{E}_k \approx y_{n+1}(k+1) - y_{n+1}(k)$$

Where $y_{n+1}(k)$ and $y_{n+1}(k+1)$ are the results of $y_h(x_{n+1})$ by using the iterative mode $\mathrm{PEC}_k\mathrm{E}$ and $\mathrm{PEC}_{k+1}\mathrm{E}$, respectively. Let

$$E_k = y_{n+1}(k+1) - y_{n+1}(k) \qquad (7)$$

denotes the estimated error in $y_{n+1}(k)$ at $x_{n+1}$. It follows from (6) that

$$y_{n+1}(k) = p_h(x_{n+1}) + \frac{g_{k-1,1}(x_{n+1})}{g_{k-1,0}(x_{n+1})}\bar{e}$$

$$y_{n+1}(k+1) = p_h(x_{n+1}) + \frac{g_{k,1}(x_{n+1})}{g_{k,0}(x_{n+1})}\hat{e}$$

Where,

$$\bar{e} = y'_{n+1}(k) - p'_h(x_{n+1})$$

And

$$\hat{e} = y'_{n+1}(k+1) - p'_h(x_{n+1}).$$

Now,

$$\begin{aligned} \bar{e} &= g_{k-1,0}(x_{n+1})(x_{n+1} - x_{n-k+1}) f[x_{n+1}, x_n, \ldots, x_{n-k+1}] \\ &= g_{k,0}(x_{n+1}) f[x_{n+1}, x_n, \ldots, x_{n-k+1}] \\ &= \hat{e} \\ &= e. \end{aligned} \qquad (8)$$

Using Eq. (8), we can rewrite (7) as

 

$$E_k = \frac{-g_{k-1,2}(x_{n+1})}{g_{k,0}(x_{n+1})} e.$$

This variable order variable step size algorithm for DDE starts with method of order one, that is k = 1. As the integration proceeds, the order k takes the value in the range 1 ≤ k ≤ 12. It is necessary to estimate the local errors of different orders because these estimates will facilitate the order changing strategy and the optimal step size selection in the algorithm. The estimated errors of orders k − 1 and k − 2 can be calculatedwithout actually taking the steps of orders k −1 and k −2, respectively. The estimated local error of order k − 2 is denoted by Ek−2. The derivation for Ek−2 follows from that of Ek . The formula is given by

$$E_{k-2} = -g_{k-3,2}(x_{n+1}) f[x_{n+1}, x_n, \ldots, x_{n-k+3}]. \qquad (9)$$

Similarly, the estimates of the local error Ek−1 and Ek+1 can be calculated by replacing k −2 with k −1 and k +1 in (9). Ek−1 and Ek+1 are local error estimates of orders k − 1 and k + 1, respectively. Ek+1 can only be calculated in the case of a successful step. The pattern developed by these estimates is used in the order selection strategy. The order is either increased or reduced by one or kept the same. The application of order changing strategy follows from that of the same strategy used in Suleiman [14]. Once the new order is selected, it is renamed k and the corresponding Ek is used in selecting the new stepsize. In the case of a step failure, the new stepsize is reduced by half. After four repeated failures, the new stepsize is reduced further by a factor of 10. In this case, the order of the method is reduced to k = 1. The integration stops if there happens to be twelve repeated failures.

**4. Absolute stability**

In the development of numerical methods, it is of practical importance to study the behavior of the global error. An important aspect is for the numerical solution to behave as the exact solution does for a fixed h as x approaches infinity. Many authors, including Barwell [4], Wiederholt [16], Al-Mutib [2] and Jackiewicz [6] studied various concepts of stability for DDEs based on test equations with known stability regions. The simplest test equation is

$$\begin{aligned} y'(x) &= \mu y(x-\tau), \quad x \geq x_0, \\ y(x) &= \phi(x), \qquad -\tau \leq x \leq x_0, \end{aligned} \qquad (10)$$

and a more general test equation is

$$\begin{aligned} y'(x) &= \lambda y(x) + \mu y(x-\tau), \quad x \geq x_0, \\ y(x) &= \phi(x), \qquad -\tau \leq x \leq x_0. \end{aligned} \qquad (11)$$

Barwell introduced the notions of Q- and GQ-stability related to Eq. (10) and P- and GP-stability related to Eq. (11). Wiederholt determined the stability regions for second order Milne predictor–corrector method and third order Adams predictor– corrector method for Eq. (11). Al-Mutib determined the stability regions related to Eqs. (10) and (11) for the trapezium rule, Runge–KuttaMerson method and the fourth order implicit Runge–Kutta method. Jackiewicz investigated the stability regions for θ-methods with regard to Eq. (11).

In this section, we present the stability regions for various orders of Adams methods in PECE scheme as applied to (10) and (11). We consider a fixed stepsize h such that xn = x0 + nh and mh = τ, m ∈ I +. Let H1 = hλ and H2 = hμ. We shall adopt the following definitions of P- and Q-stability regions suggested by Al-Mutib [2].

**5. Conclusion**

We have presented a variable order variable stepsize algorithm for the numerical solution of DDEs. The method is implemented in PECE scheme where the predictor– corrector formulae are represented in divided difference form. From the numerical results we conclude that this algorithm is efficient, accurate and reliable for solving DDEs.

**References**


1. Al-Mutib, A.N.: Numerical methods for solving delay differential equations. Ph.D.thesis, University of Manchester, UK (1977)
2. Al-Mutib, A.N.: Stability properties of numerical methods for solving delay differential equations. J. Comput. Appl. Math. 10, 71–79 (1984)
3. Baker, C.T.H., Paul, C.A.H, Wille, D.R.: Issues in the numerical solution of evolutionary delay differential equations. Technical Report No. 248, University of Manchester, UK (1994)
4. Barwell, V.K.: Special stability problems for functional differential equations. BIT 15, 130–135 (1975)
5. Driver, R.D.: Ordinary and Delay Differential Equations. Springer, New York (1977)
6. Jackiewicz, Z.: Asymptotic stability analysis of θ-methods for functional differential equations. Numer. Math. 43, 389–396 (1984)
7. Jackiewicz, Z.: Variable-step variable-order algorithm for the numerical solution of neutral functional differential equations. Appl. Numer. Math. 3, 317–329 (1987)
8. Jackiewicz, Z., Lo, E.: The numerical solution of neutral functional differential equations by Adams predictor–corrector methods. Appl. Numer. Math. 8, 477–491 (1991)


 

9. Jackiewicz, Z., Lo, E.: Numerical solution of neutral functional differential equations by Adams methods in divided difference form. J. Comput. Appl. Math. 189, 592–605 (2006)
10. Kemper, G.A.: Linear multistep methods for a class of functional differential equations. Numer. Math. 19, 361–372 (1972)
11. Lambert, J.D.: Computational Methods in Ordinary Differential Equations. Wiley, London (1973)
12. Shampine, L.F., Gordon, M.K.: Computer Solution of Ordinary Differential Equations: The Initial Value Problem. W. H. Freeman, San Francisco (1975)
13. Shampine, L.F., Thompson, S.: Solving DDEs in MATLAB. Appl. Numer. Math. 37, 441–458 (2001)
14. Suleiman, M.B.: Generalized multistep Adams and backward differentiation methods for the solution of stiff and non-stiff ordinary differential equations. Ph.D.thesis, University of Manchester, UK (1979)
15. Suleiman, M.B., Ismail, F.: Solving delay differential equations using componentwise partitioning by Runge–Kutta method. Appl. Math. Comput. 122, 301–323 (2001)
16. Wiederholt, L.F.: Stability of multistep methods for delay differential equations. Math. Comp. 30, 283– 290 (1976)

 