Research Article

Solving Second-Order Delay Differential Equations by Direct Adams-Moulton Method

Hoo Yann Seong, 1 Zanariah Abdul Majid, 1,2 and Fudzial Ismail 1,2

1 Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, Serdang, 43400 Selangor DE, Malaysia
2 Institute for Mathematical Research, Universiti Putra Malaysia, Serdang, 43400 Selangor DE, Malaysia

Correspondence should be addressed to Zanariah Abdul Majid; zana_majid99@yahoo.com

Received 22 June 2013; Revised 6 November 2013; Accepted 11 November 2013

Academic Editor: Ebrahim Momoniat

Copyright © 2013 Hoo Yann Seong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper will consider the implementation of fifth-order direct method in the form of Adams-Moulton method for solving directly second-order delay differential equations (DDEs). The proposed direct method approximates the solutions using constant step size. The delay differential equations will be treated in their original forms without being reduced to systems of first-order ordinary differential equations (ODEs). Numerical results are presented to show that the proposed direct method is suitable for solving second-order delay differential equations.

1. Introduction

In the recent years, there are rigorous and numerous researches undertaken in the areas of science and engineering that are skewed towards the developments of the mathematical models involving the delay differential equations. In mathematics, the DDEs are differing from ODEs in which the derivative of the unknown function at a certain time is given in terms of the values of the function at previous times. Several numerical methods have been proposed to solve first-order DDEs such as in [1–4]. However, less attention was made to solve second-order DDEs. Spline collocation methods [5] and Adomian decomposition method [6] have been proposed to solve second-order DDEs directly. There were few numerical methods that have been proposed for solving ODEs and those methods have been extended to solve DDEs with some modifications in the algorithm. These are some works carried out for solving first-order DDEs using the extended version such as in San et al. [1], Ismail et al. [2], Radzi et al. [3], and Ishak et al. [4].

In this paper, we are concerned with solving second-order delay differential equations (DDEs) as follows:

\[
y'' = f(t, y(t), y(t-\tau)), \quad a \leq t \leq b, \quad \tau > 0,
\]

where \( f(t, y(t), y(t-\tau)) \) is the function given, \( a \) is the initial time, \( b \) is the final time, and \( \tau \) is the delay term.

\[
y'(a) = \Omega,
\]

\[
y(t) = \phi(t), \quad \alpha \leq t \leq a, \quad 0 \leq \tau \leq |a-\alpha|,
\]

where \( \phi(t) \) is the initial function and \( \tau \) is the delay term.

The direct Adams-Moulton methods were studied by several researchers and the methods have shown their ability to solve first-, second- and higher-order ODEs [7, 8] and boundary value problems [9] effectively and accurately. Hence, in this paper, we aim to propose the direct method of order five in the form of Adams-Moulton method for solving (1) using constant step size. The proposed direct method has the advantage to solve the second-order delay differential equations directly without reducing to system of first order. Therefore, the second-order DDEs problems will be handled in their original forms. This idea will reduce the computation cost at each step for the proposed direct method.

2. The Direct Method

2.1. Formulation of the Method. Most numerical methods for ODEs can be used to solve DDEs. In this paper, we adopted direct method proposed by Majid and Suleiman [8] to solve DDEs. The following is the derivation of one point direct Adams-Moulton method.
The point \( y_{m+1} \) at \( x_{n+1} \) can be obtained by integrating
\[
y''(t, y(t), y') = f(t, y(t), y')
\]
once and twice as follows.

Integrating once
\[
\int_{x_n}^{x_{n+1}} y''(x) \, dx = \int_{x_n}^{x_{n+1}} f(x, y, y') \, dx,
\]
\[
y'(x_{n+1}) = y'(x_n) + \int_{x_n}^{x_{n+1}} f(x, y, y') \, dx.
\]
Integrating twice
\[
\int_{x_n}^{x_{n+1}} \int_{x_n}^{x} y''(x) \, dx \, dx = \int_{x_n}^{x_{n+1}} \int_{x_n}^{x} f(x, y, y') \, dx \, dx,
\]
\[
y(x_{n+1}) - y(x_n) - hy'(x_n)
\]
\[
= \int_{x_n}^{x_{n+1}} (x_{n+1} - x) f(x, y, y') \, dx.
\]

The function \( f(x, y, y') \) in (6) and (8) will be approximated using Lagrange interpolation polynomial and the interpolation points involved are five points, that is, \( \{x_{n-3}, x_{n-2}, x_{n-1}, x_n, x_{n+1} \} \). Taking \( s = (x - x_{n+1})/h \) and replacing \( dx = h \, ds \), the value of \( y_{n+1} \) can be obtained by integrating (6) and (8) using MAPLE. The direct method is the combination of predictor of one order less than the corrector. The following is the predictor and corrector of the direct method.

**Predictor**
\[
y'(x_{n+1}) = y'(x_n) + \frac{h}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}),
\]
\[
y(x_{n+1}) = y(x_n) + hy'(x_n)
\]
\[
+ \frac{h^2}{360} (323f_n - 264f_{n-1} + 159f_{n-2} - 38f_{n-3}).
\]

**Corrector**
\[
y'(x_{n+1}) = y'(x_n) - \frac{h}{720} \times (-251f_{n+1} - 646f_n + 264f_{n-1} - 106f_{n-2} + 19f_{n-3}),
\]
\[
y(x_{n+1}) = y(x_n) + hy'(x_n) - \frac{h^2}{1440} \times (-135f_{n+1} - 752f_n + 246f_{n-1} - 96f_{n-2} + 17f_{n-3}).
\]

2.2. **Order and Error Constant of the Method.** The order of this developed method is calculated by referring to [10–14]. Linear \( k \)-step method can be written in the form of
\[
\alpha Y_m = h\beta Y_m' + h^2\gamma F_m,
\]
where \( \alpha, \beta, \) and \( \gamma \) are the coefficients with the \( m \)-vector. \( Y_m, Y_m' \), and \( F_m \) are
\[
Y_m = [y_{n-1}, y_n]^T, \quad Y_m' = [y_{n-1}', y_n']^T,
\]
\[
F_m = [f_{n-1}, f_n]^T.
\]
The formulae is defined as
\[
C_0 = \alpha_0 + \alpha_1 + \cdots + \alpha_k
\]
\[
C_1 = \alpha_1 + 2\alpha_2 + \cdots + k\alpha_k - (\beta_0 + \beta_1 + \cdots + \beta_k)
\]
\[
C_2 = \frac{1}{2!} (\alpha_1 + 2\alpha_2 + \cdots + k^2\alpha_k) - (\beta_1 + 2\beta_2 + \cdots + k\beta_k)
\]
\[
- (\gamma_0 + \gamma_1 + \cdots + \gamma_k)
\]
\[
\vdots
\]
\[
C_q = \sum_{j=0}^{k} \left( \frac{j^q}{q!} \alpha_j - \frac{j^{q-1}}{(q-1)!} \beta_j - \frac{j^{q-2}}{(q-2)!} \gamma_j \right),
\]
where \( q = 3, 4, 5, \ldots \).

The method has order \( p \) if \( C_0 = C_1 = \cdots = C_p = C_{p+1} = 0 \) and \( C_{p+2} \neq 0 \) is the error constant.

By referring to the corrector formulae in (11) and the general multistep method in (12), we will obtain
\[
\alpha_0 = 0, \quad \alpha_1 = 0,
\]
\[
\alpha_2 = -1, \quad \alpha_3 = 0, \quad \alpha_4 = 1;
\]
\[
\beta_0 = 0, \quad \beta_1 = 0,
\]
\[
\beta_2 = 0, \quad \beta_3 = 1, \quad \beta_4 = 0;
\]
\[
y_0 = -\frac{17}{1440}, \quad y_1 = \frac{96}{1440},
\]
\[
y_2 = -\frac{246}{1440}, \quad y_3 = \frac{752}{1440}, \quad y_3 = \frac{135}{1440}
\]
and substitute those values into (14):
\[
C_0 = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0,
\]
\[
C_1 = \alpha_1 + 2\alpha_2 + 3\alpha_3 + 4\alpha_4
\]
\[
- (\beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4) = 0,
\]
Mathematical Problems in Engineering

\[ C_2 = \frac{1}{2!} (\alpha_1 + 2^2 \alpha_2 + 3^2 \alpha_3 + 4^2 \alpha_4) \]
\[ - (\beta_1 + 2 \beta_2 + 3 \beta_3 + 4 \beta_4) \]
\[ - (\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4) = 0, \]
\[ C_3 = \frac{1}{3!} (\alpha_1 + 2^2 \alpha_2 + 3^2 \alpha_3 + 4^2 \alpha_4) \]
\[ - \frac{1}{2!} (\beta_1 + 2^2 \beta_2 + 3^2 \beta_3 + 4^2 \beta_4) \]
\[ - (\gamma_1 + 2 \gamma_2 + 3 \gamma_3 + 4 \gamma_4) = 0, \]
\[ C_4 = 0, \]
\[ C_5 = 0, \]
\[ C_6 = 0, \]
\[ C_7 = - \frac{41}{5040} \neq 0. \]  

(16)

Hence, \( C_0 = C_1 = \cdots = C_6 = 0 \) and \( C_7 = -(41/5040) \neq 0 \) is the error constant.

The corrector of the direct method is of order five and the error constant is \(-41/5040\). The method is said to be consistent if it has at least one order. Since the proposed method is order five, hence the method is said to be consistent.

2.3. Stability Analysis. The method is zero stable provided the roots \( R_j \) of the first characteristic polynomial \( \rho(R) \) specified as \( \rho(R) = \det \left[ \sum_{k=0}^{\infty} A^k R^{k-i} \right] = 0 \) and satisfy \( |R_j| \leq 1 \).

We rewrite (10) and (11) in the matrix form:

\[
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
y_{n+1} \\
y_n \\
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
y_n \\
y_n \\
\end{bmatrix}
+ h
\begin{bmatrix}
251 & 0 \\
0 & 720 \\
\end{bmatrix}
\begin{bmatrix}
f_n' \\
f_{n+1}' \\
\end{bmatrix}
+ h
\begin{bmatrix}
646 & 0 \\
0 & 720 \\
\end{bmatrix}
\begin{bmatrix}
f_n & f_n' \\
f_{n+1} & f_{n+1}' \\
\end{bmatrix}
+ h
\begin{bmatrix}
264 & 0 \\
0 & 720 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-1}' \\
f_{n-1} \\
\end{bmatrix}
+ h
\begin{bmatrix}
106 & 0 \\
0 & 720 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-2}' \\
f_{n-2} \\
\end{bmatrix}
+ h
\begin{bmatrix}
19 & 0 \\
0 & 720 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-3}' \\
f_{n-3} \\
\end{bmatrix}
+ h
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_{n+1}' \\
f_{n+1} \\
\end{bmatrix}
+ h^2
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_n' \\
f_n \\
\end{bmatrix}
+ h^2
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-1}' \\
f_{n-1} \\
\end{bmatrix}
+ h^2
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-2}' \\
f_{n-2} \\
\end{bmatrix}
+ h^2
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-3}' \\
f_{n-3} \\
\end{bmatrix}
+ h^2
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-4}' \\
f_{n-4} \\
\end{bmatrix}
+ h^2
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-5}' \\
f_{n-5} \\
\end{bmatrix}
+ h^2
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-6}' \\
f_{n-6} \\
\end{bmatrix}
+ h^2
\begin{bmatrix}
0 & 0 \\
0 & 1440 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-7}' \\
f_{n-7} \\
\end{bmatrix}
\]

(17)

The first characteristic polynomial of the method is given as

\[ \rho(R) = \det[R A^0 - A^1] = 0, \]

where \( A^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \) and \( A^1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \). Consider that

\[ \rho(R) = \det \begin{bmatrix} R - 1 & 0 \\ 0 & R - 1 \end{bmatrix} = 0, \]

\[ (R - 1)^2 = 0, \]

\[ R = 1, 1. \]

Since \( |R_j| \leq 1 \), the method is said to be zero stable.

There are many concepts of stability for numerical methods when applied to DDEs, depending on the test equation as well as the delay term involved. We would like to study the stability of the method by substituting the following test equation:

\[ f = \lambda y(x) + \mu y(x - \tau) \]  

into the proposed method (10) and (11). The method can be described in the following matrix form:

\[
A_0 Y_{N+1} = A_1 Y_N + h^4 \sum_{i=0}^{4} B_{i+1} F_{N-i} + h^2 \sum_{i=0}^{4} C_{i+1} F_{N-i},
\]

(20)

where

\[ Y_{N+1} = \begin{bmatrix} y'_{n+1} \\ y_{n+1} \end{bmatrix}, \]

\[ Y_N = \begin{bmatrix} y'_{n} \\ y_{n} \end{bmatrix}, \]

\[ F_{N-i} = \begin{bmatrix} y'_{n+i} \\ y_{n+i} \end{bmatrix}, \]

\[ A_0 = \begin{bmatrix} 1 & - \frac{251}{720} h \lambda \\ 0 & 1 - \frac{1}{32} h^3 \lambda \end{bmatrix}, \]

\[ A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \]

\[ B_1 = \begin{bmatrix} 1 & - \frac{323}{360} h \lambda + \frac{251}{720} h \mu \\ 0 & 1 \end{bmatrix}, \]

\[ B_2 = \begin{bmatrix} 1 & - \frac{11}{30} h \lambda + \frac{323}{360} h \mu \\ 0 & 1 \end{bmatrix}. \]
\[ B_3 = \begin{bmatrix} 0 & 53/360 h \lambda - 11/30 h \mu \\ 0 & 0 \end{bmatrix}, \]
\[ B_4 = \begin{bmatrix} 0 & 19/720 h \lambda + 53/360 h \mu \\ 0 & 0 \end{bmatrix}, \]
\[ B_5 = \begin{bmatrix} 0 & -19/720 h \mu \\ 0 & 0 \end{bmatrix}, \]
\[ C_1 = \begin{bmatrix} 0 & 0 \\ 0 & 47/90 h \lambda + 3/32 h \mu \end{bmatrix}, \]
\[ C_2 = \begin{bmatrix} 0 & 0 \\ 0 & -11/60 h \lambda + 47/90 h \mu \end{bmatrix}, \]
\[ C_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1/15 h \lambda - 11/60 h \mu \end{bmatrix}, \]
\[ C_4 = \begin{bmatrix} 0 & 0 \\ 0 & -17/1440 h \lambda + 1/15 h \mu \end{bmatrix}, \]
\[ C_5 = \begin{bmatrix} 0 & 0 \\ 0 & -17/1440 h \mu \end{bmatrix}. \]

We solve the determination of \( d = 0 \), where
\[
\begin{align*}
    d &= t^5 A_0 - t^4 \left( A_1 + h B_1 + h^2 C_1 \right) - t^3 \left( h B_2 + h^2 C_2 \right) \\
    &\quad - t^2 \left( h B_3 + h^2 C_3 \right) - t \left( h B_4 + h^2 C_4 \right) - \left( h B_5 + h^2 C_5 \right).
\end{align*}
\]

The following stability polynomial is obtained by letting \( H_1 = h^2 \mu \) and \( H_2 = h^2 \lambda \):
\[
\begin{align*}
    \left( 1 - \frac{3}{32} H_2 \right) t^{10} + \left( -2 + \frac{3}{32} H_1 - \frac{373}{480} H_2 \right) t^9 \\
    + \left( 1 - \frac{373}{480} H_1 - \frac{23}{120} H_2 \right) t^8 + \left( -\frac{23}{120} H_1 + \frac{7}{60} H_2 \right) t^7 \\
    + \left( \frac{7}{60} H_1 - \frac{11}{160} H_2 \right) t^6 + \left( -\frac{11}{160} H_1 + \frac{7}{480} H_2 \right) t^5 \\
    + \frac{7}{480} H_1 &= 0.
\end{align*}
\]

The boundary of the stability region in \( H_1 \)-\( H_2 \) plane is determined by substituting the values of \( t = 1, -1, \) and \( e^{i \theta} \) into the stability polynomial, where \( 0 \leq \theta \leq 2 \pi \). Figure 1 shows the stability region of the direct method and the stable region is the bounded shaded region.

### 2.4. Convergence Analysis

A basic property for an effective numerical method is that the method needs to converge. A linear multistep method is convergent if and only if it is stable and consistent [15].

By definition, a linear multistep method of the form
\[
\sum_{j=0}^{k} \alpha_j y_{n+j} = h^2 \sum_{j=0}^{k} \beta_j f_{n+j}
\]

is said to be consistent if the LMM is of order \( p \geq 1 \).

The proposed direct method is of order five, where \( p = 5 \) and has error constant \( \frac{-3}{160} \) and \( \frac{-41}{5040} \).

Since the method is order five which is \( \geq 1 \), therefore, the method is consistent according to the definition. In the previous section, it has been shown that the method is zero stable. Therefore, we can conclude that this method is convergent.

### 3. Implementation

In the code of PECE scheme \( P \) 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 developed code starts by using Adams-Bashforth method once at the beginning to calculate the three starting initial points. Once the points are calculated, then the proposed method can be applied until the end of the interval. The values of the delay term will be stored for future use. For \( x_n - \tau \leq 0 \), the delay term is calculated using the initial function given, \( \phi(x) \). Otherwise the delay term depends on the location of \((x - \tau)\). From the location we are able to recall the value which we had stored earlier. In this paper, no interpolation is required due to the implementation of constant step sizes. The algorithms of the proposed method were developed in C language.

In the code, the selection of step sizes is predetermined.

#### 3.1. Algorithm of Direct Method

**Step 1.** Set starting value \( a \), ending value \( b \), and step size \( h \), given initial value and given initial function \( g(x) \).

**Step 2.** For \( n = 1, 2, 3 \), set \( x_{n+1} = a + nh \), compute function \( f_{n'} \), and delay term \( d_{n'} \).

Evaluate the approximate value \( y_{n+1} \) with direct Adams-Bashforth method.
Step 3. While \( x_n < b \), do Step 4.

Step 4. Set \( x_{n+1} = x_n + h \), compute function \( f_n \), and delay term \( d_n \).

Evaluate the approximate value \( y_{n+1} \) with proposed method.

Computing \( y'_{n+1} \) and \( y_{n+1} \), we use the predictor formulas as follows:
\[
y'_{n+1} = y'(x_n) + \frac{h}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}),
\]
\[
y_{n+1} = y(x_n) + hy'(x_n)
\]
\[
+ \frac{h^2}{360} (323f_n - 264f_{n-1} + 159f_{n-2} - 38f_{n-3}).
\]
(25)

Computing \( y''_{n+1} \) and \( y_{n+1} \), we use the corrector formulas as follows:
\[
y''_{n+1} = y''(x_n) - \frac{h}{720} \times (-251f_n - 646f_{n-1} + 264f_{n-2} - 106f_{n-3} + 19f_{n-4}),
\]
\[
y_{n+1} = y(x_n) + hy'(x_n) - \frac{h^2}{1440} \times (-135f_n - 752f_{n-1} + 246f_{n-2} - 96f_{n-3} + 17f_{n-4}).
\]
(26)

Step 5. Complete.

4. Numerical Result and Discussion

In order to study the efficiency of the proposed direct method, we presented three second-order DDE problems with constant delay in the following test problems. The numerical results of the direct method when solving Problems 1, 2, and 3 will be compared with cubic spline in [5], variable multistep method (VMM) in [16], and dde23 in MATLAB solver, respectively.

**Problem 1.** Consider
\[
y''(t) = \frac{1}{2} y(t) + \frac{1}{2} y(t - \pi), \quad t \in [0, \pi],
\]
(27)
y
(28)

Exact solution
\[
y(t) = 1 - \sin(t).
\]

**Problem 2.** Consider
\[
y''(t) + y(t) = y(t - 1), \quad t \in [0, 1],
\]
\[
y(t) = t^2 + 3t + 2, \quad -1 \leq t \leq 0,
\]
(29)
y
(30)

The algorithm of the C language was executed on the Microsoft Visual C++ environment. The notations at the end of the paper are used in Tables 1–3.

The numerical results for solving Problems 1–3 are displayed in Tables 1–3.

In Problem 1, we solved the DDEs by the direct method and compare our results with the cubic spline method in [1]. Table 1 shows that the direct method managed to obtain highly accurate results compared to the cubic spline method at the same values of \( h \). At larger step size, that is, \( h = \frac{\pi}{10} \), we observed that the maximum error obtained by the direct method and the cubic spline is 6.128 \(- 06 \) and 1.848 \(- 02 \), respectively. We could also observe that the maximum error for the direct method and the cubic spline was comparable, that is, 6.128 \(- 06 \) and 4.528 \(- 06 \) at \( h = \frac{\pi}{10} \) and \( \frac{\pi}{(10 \cdot 2^6)} \), respectively. Hence, these results show that the direct method is able to obtain comparable maximum error compared to cubic spline at larger step size and therefore less expensive. The total function calls for the direct method during the computation are also shown in Table 1. Figure 2 display the comparison of the maximum error at different values of step sizes for solving Problem 1.

Table 2 displays the results for the direct method with the range of step sizes from 0.0128 to 0.1 compared to the variable multistep size method in [10] with varies step sizes (range from 0.0016 to 0.002) at different values of \( t \) when solving Problem 2. The direct method is clearly superior compared to variable multistep method since it is able to obtain comparable and better results at larger step sizes. Hence, the direct method has less computational cost. The total function calls are also shown in Table 2. Figure 3 shows the comparison of maximum error at the given value of \( t \).
Table 2: Comparison of the numerical results for solving Problem 2.

<table>
<thead>
<tr>
<th>Variable multistep method [10]</th>
<th>Direct method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>MAXE</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0.1280</td>
<td>$4.981077E-08$</td>
</tr>
<tr>
<td>0.2360</td>
<td>$1.726081E-07$</td>
</tr>
<tr>
<td>0.3418</td>
<td>$3.588444E-07$</td>
</tr>
<tr>
<td>0.4444</td>
<td>$5.915465E-07$</td>
</tr>
<tr>
<td>0.5440</td>
<td>$8.587104E-07$</td>
</tr>
<tr>
<td>0.6412</td>
<td>$1.149353E-06$</td>
</tr>
<tr>
<td>0.7346</td>
<td>$1.449843E-06$</td>
</tr>
<tr>
<td>0.8264</td>
<td>$1.756201E-06$</td>
</tr>
<tr>
<td>0.9136</td>
<td>$2.051398E-06$</td>
</tr>
<tr>
<td>1.0000</td>
<td>$2.341980E-06$</td>
</tr>
</tbody>
</table>

Table 3: Comparison of the numerical results for solving Problem 3.

<table>
<thead>
<tr>
<th>RT</th>
<th>AT</th>
<th>MATLAB solver dde23</th>
<th>Direct method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1E-3$</td>
<td>$1E-6$</td>
<td>$6.5248E-05$</td>
<td>$9.9016E-05$</td>
</tr>
<tr>
<td>$1E-4$</td>
<td>$1E-6$</td>
<td>$7.1874E-06$</td>
<td>$7.7395E-06$</td>
</tr>
<tr>
<td>$1E-5$</td>
<td>$1E-6$</td>
<td>$7.1018E-07$</td>
<td>$2.4408E-07$</td>
</tr>
<tr>
<td>$1E-6$</td>
<td>$1E-7$</td>
<td>$7.1144E-08$</td>
<td>$1.6530E-08$</td>
</tr>
<tr>
<td>$1E-7$</td>
<td>$1E-8$</td>
<td>$7.1382E-09$</td>
<td>$1.0953E-09$</td>
</tr>
<tr>
<td>$1E-8$</td>
<td>$1E-9$</td>
<td>$7.1507E-10$</td>
<td>$4.6437E-10$</td>
</tr>
<tr>
<td>$1E-9$</td>
<td>$1E-10$</td>
<td>$7.1574E-11$</td>
<td>$4.8654E-11$</td>
</tr>
</tbody>
</table>

Table 3 displays the numerical results for solving Problem 3 for the MATLAB solver dde23 and the present direct method. Since dde23 is a variable step size method, therefore, the user may control the efficiency and accuracy of the solutions by changing the real tolerance and the absolute tolerance. The table will only show the compatible maximum absolute error with the least function calls. We could observe that the direct method managed to obtain comparable
maximum errors with lesser total function calls and total step and to have no failure step compared to dde23. Hence, the direct method has less computational cost compared to dde23. Figure 4 shows the comparison results for maximum error and total function calls.

5. Conclusion

In this study, we have shown that the proposed direct Adams-Moulton method using constant step size is suitable for solving second-order DDEs directly. The proposed direct method has solved the second-order DDEs in their original forms without being reduced to first-order ODEs. This approach has given advantage in terms of computational cost to the direct method. The method has shown superiority in terms of accuracy and it has less computational cost.

Notations

- $h$: Step size
- MAXE: Magnitude of the maximum absolute error
- FCN: Total function calls
- TS: Total steps
- TFS: Total failure steps
- RT: Real tolerance
- AT: Absolute tolerance
- dde23: MATLAB solver dde23 based on the explicit Runge-Kutta (2,3) pair [17].

Acknowledgment

The authors gratefully acknowledged the financial support of University Putra Malaysia Grant.

References
