Research Article

Direct Two-Point Block One-Step Method for Solving General Second-Order Ordinary Differential Equations

Zanariah Abdul Majid,[1,2] **Nur Zahidah Mokhtar,**[1] **and Mohamed Suleiman**[1,2]

[1] *Institute for Mathematical Research, Universiti Putra Malaysia, Selangor Darul Ehsan, 43400 Serdang, Malaysia*
[2] *Mathematics Department, Faculty of Science, Universiti Putra Malaysia, Selangor Darul Ehsan, 43400 Serdang, Malaysia*

Correspondence should be addressed to Zanariah Abdul Majid, zanariah@math.upm.edu.my

A direct two-point block one-step method for solving general second-order ordinary differential equations (ODEs) directly is presented in this paper. The one-step block method will solve the second-order ODEs without reducing to first-order equations. The direct solutions of the general second-order ODEs will be calculated at two points simultaneously using variable step size. The method is formulated using the linear multistep method, but the new method possesses the desirable feature of the one-step method. The implementation is based on the predictor and corrector formulas in the $PE(CE)^m$ mode. The stability and precision of this method will also be analyzed and deliberated. Numerical results are given to show the efficiency of the proposed method and will be compared with the existing method.

1. Introduction

In this paper, we are considering solving directly the general second-order initial value problems (IVPs) for systems of ODEs in the form

$$y'' = f(x, y, y'), \quad y(a) = y_0, \quad y'(a) = y'_0, \quad x \in [a, b]. \tag{1.1}$$

Equation in (1.1) arises from many physical phenomena in a wide spectrum of applications especially in the science and engineering areas such as in the electric circuit, damped and
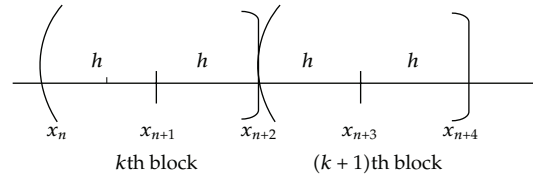
Figure 1: Block one-step method.

undamped spring mass and some other areas of application. We will also consider solving general second order as in (1.1) using the direct block one-step method. Block methods for numerical solutions of ODEs have been proposed by several researchers such as in [1–5]. The common block methods used to solve the problems can be categorized as one-step block method and multistep block method.

One-step block method such as the implicit Runge-Kutta method is also being referred to as one previous point to obtain the solution. The multistep block method in the form of Adams type formula is presented in [5, 6]. In [7], the block backward differentiation formula (BBDF) for solving stiff ODEs has been introduced and the solutions referred to as more than one previous point. The works in [6] showed the proposed two-point four-step block method presented as in a simple form of Adams Moulton method for solving second-order ODEs directly.

The block method of Runge-Kutta type has been explored in [1], and it is suggested that a block of new approximation values is used simultaneously for solving first-order ODEs. The works in [3, 8, 9] have been considered in solving (1.1) using the block one-step method, while [3] has proposed a two-point implicit block one-step method for solving second-order ODEs directly and suggested that the method is suitable to be parallel.

In [9], Majid et al. have derived the two-point block method for solving first-order ODEs by using the closest point in the interval, that is, $[x_n, x_{n+1}]$ and $[x_{n+1}, x_{n+2}]$. The Gauss Seidel iteration was implemented in the proposed block method. The approach in this research is to extend the idea in [9] for solving (1.1) directly without reducing system of first-order ODEs using two-point block one-step method.

2. Formulation of the Method

In order to compute the two approximation values of y_{n+1} and y_{n+2} simultaneously, the interval of $[a, b]$ is divided into a series of blocks with each block containing two points as shown in Figure 1.

In Figure 1, we observed that kth block contains x_n, x_{n+1} and x_{n+2}, where x_n becomes the starting point and x_{n+2} is the last point in the kth block with step size h. The approximations values of y_{n+1} and y_{n+2} are computed simultaneously. The evaluation solution at the last point in kth block will be restored as the initial values for $(k + 1)$th block. The same procedure is used to compute the solutions for the next block until the end of the interval. The evaluation information from the previous step in a block could be used for other steps of the same block only. During the calculations of iteration, the final values of y_{n+2} at the point x_{n+2} are taken as the initial values for the next iteration.

We obtained the approximation values of y_{n+1} and y_{n+2} at the points x_{n+1} and x_{n+2} by integrating once and twice over (1.1) with respect to x. We begin to evaluate the y'_{n+1} and y_{n+1} by integrating once and twice (1.1) over the interval $[x_n, x_{n+1}]$:

$$\int_{x_n}^{x_{n+1}} y''(x)dx = \int_{x_n}^{x_{n+1}} f(x,y,y')dx,$$

$$\int_{x_n}^{x_{n+1}} \int_{x_n}^{x} y''(x)dx\,dx = \int_{x_n}^{x_{n+1}} \int_{x_n}^{x} f(x,y,y')dx\,dx. \tag{2.1}$$

Let $x_{n+1} = x_n + h$ which gives

$$y'(x_{n+1}) - y'(x_n) = \int_{x_n}^{x_{n+1}} f(x,y,y')\ dx,$$

$$y(x_{n+1}) - y(x_n) - hy'(x_n) = \int_{x_n}^{x_{n+1}} (x_{n+1} - x)f(x,y,y')\ dx. \tag{2.2}$$

Then, $f(x,y,y')$ in (2.2) will be replaced with Lagrange interpolation polynomial that involves interpolation points in the block at $(x_n, f_n), (x_{n+1}, f_{n+1})$ and (x_{n+2}, f_{n+2}). Taking $x = x_{n+2} + sh, s = (x - x_{n+2})/h$, and $dx = h\,ds$ and replacing into (2.2). Then, the limit of integration in (2.2) will be from -2 to -1. The corrector formulae will be obtained using MATHEMATICA. The formulae of y'_{n+1} and y_{n+1} are obtained as follows:

$$y'_{n+1} = y'_n + \frac{h}{12}(5f_n + 8f_{n+1} - f_{n+2}),$$

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{24}(7f_n + 6f_{n+1} - f_{n+2}). \tag{2.3}$$

To approximate the value of y'_{n+2} and y_{n+2}, we take $x_{n+2} = x_n + 2h$ by integrating once and twice (1.1) over the interval $[x_{n+1}, x_{n+2}]$ and apply the same process. The limit of integration will be from -1 to 0, and the following corrector formulae will be obtained:

$$y'_{n+2} = y'_{n+1} + \frac{h}{12}(-f_n + 8f_{n+1} + 5f_{n+2}),$$

$$y_{n+2} = y_{n+1} + hy'_{n+1} + \frac{h^2}{24}(-f_n + 10f_{n+1} + 3f_{n+2}). \tag{2.4}$$

The formulae (2.3) and (2.4) may be rewritten in the form of matrix difference equation as follows:

$$
\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}
\begin{bmatrix} y_{n-1} \\ y_n \\ y_{n+1} \\ y_{n+2} \end{bmatrix}
= h \left(
\begin{bmatrix} 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\begin{bmatrix} y'_{n-1} \\ y'_n \\ y'_{n+1} \\ y'_{n+2} \end{bmatrix}
\right)
+ h^2 \left(
\begin{bmatrix}
0 & \dfrac{5}{12} & \dfrac{8}{12} & \dfrac{-1}{12} \\[2mm]
0 & \dfrac{7}{24} & \dfrac{6}{24} & \dfrac{-1}{24} \\[2mm]
0 & \dfrac{-1}{12} & \dfrac{8}{12} & \dfrac{5}{12} \\[2mm]
0 & \dfrac{-1}{24} & \dfrac{10}{24} & \dfrac{3}{24}
\end{bmatrix}
\begin{bmatrix} f_{n-1} \\ f_n \\ f_{n+1} \\ f_{n+2} \end{bmatrix}
\right).
$$

$$(2.5)$$

The order of this developed method is identified by referring to [10–12]. The two-point one-step block method for ODEs can be written in a matrix difference equation as follows:

$$
\alpha Y_m = h \beta Y'_m + h^2 \gamma F_m,
\tag{2.6}
$$

where α, β, and γ are the coefficients with the m-vector Y_m, Y'_m, and F_m be defined as

$$
Y_m = \left[y_{n-1}, y_n, y_{n+1}, y_{n+2} \right]^T, \qquad
Y'_m = \left[y'_{n-1}, y'_n, y'_{n+1}, y'_{n+2} \right]^T, \qquad
F_m = \left[f_{n-1}, f_n, f_{n+1}, f_{n+2} \right]^T.
\tag{2.7}
$$

By applying the formulae for the constants C_q, in [10] the formulae is defined as

$$
C_0 = \sum_{j=0}^{k} \alpha_j,
$$

$$
C_1 = \sum_{j=0}^{k} \left(j\alpha_j - \beta_j \right),
$$

$$
C_2 = \sum_{j=0}^{k} \left(\frac{j^2}{2!}\alpha_j - j\beta_j - \gamma_j \right),
\tag{2.8}
$$

$$
\vdots
$$

$$
C_q = \sum_{j=0}^{k} \left(\frac{j^q}{q!}\alpha_j - \frac{j^{q-1}}{(q-1)!}\beta_j - \frac{j^{q-2}}{(q-2)!}\gamma_j \right), \quad \text{where } q = 3,4,5,\ldots.
$$

Therefore, the order and error constant of the two-point one-step block method can be obtained by using equation in (2.8).

For $q = 0$,

$$C_0 = \sum_{j=0}^{3} \frac{1}{0!} j^0 \alpha_j = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.9}$$

For $q = 1$,

$$C_1 = \sum_{j=0}^{3} \frac{1}{1!} j^1 \alpha_j - \sum_{j=0}^{3} \frac{1}{0!} j^0 \beta_j$$

$$= 1^1 \cdot \alpha_1 + 2^1 \cdot \alpha_2 + 3^1 \cdot \alpha_3 - (\beta_0 + \beta_1 + \beta_2 + \beta_3)$$

$$= 1 \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.10}$$

For $q = 2$,

$$C_2 = \sum_{j=0}^{3} \frac{1}{2!} j^2 \alpha_j - \sum_{j=0}^{3} \frac{1}{1!} j^1 \beta_j - \sum_{j=0}^{3} \frac{1}{0!} j^0 \gamma_j$$

$$= \frac{1}{2} \left(1^2 \cdot \alpha_1 + 2^2 \cdot \alpha_2 + 3^2 \cdot \alpha_3 \right) - \left(1^1 \cdot \beta_1 + 2^1 \cdot \beta_2 + 3^1 \cdot \beta_3 \right)$$

$$- (\gamma_0 + \gamma_1 + \gamma_2 + \gamma_3)$$

$$= \frac{1}{2} \left(1 \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} + 4 \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} + 9 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right) - \left(1 \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} -1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \right)$$

$$- \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{5}{12} \\ \frac{7}{24} \\ \frac{-1}{12} \\ \frac{-1}{24} \end{bmatrix} + \begin{bmatrix} \frac{8}{12} \\ \frac{6}{24} \\ \frac{8}{12} \\ \frac{10}{24} \end{bmatrix} + \begin{bmatrix} \frac{-1}{12} \\ \frac{-1}{24} \\ \frac{5}{12} \\ \frac{3}{24} \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.11}$$

For $q = 3$,

$$C_3 = \sum_{j=0}^{3} \frac{1}{3!} j^3 \alpha_j - \sum_{j=0}^{3} \frac{1}{2!} j^2 \beta_j - \sum_{j=0}^{3} \frac{1}{1!} j^1 \gamma_j = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.12}$$

For $q = 4$,

$$C_4 = \sum_{j=0}^{3} \frac{1}{4!} j^4 \alpha_j - \sum_{j=0}^{3} \frac{1}{3!} j^3 \beta_j - \sum_{j=0}^{3} \frac{1}{2!} j^2 \gamma_j = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.13}$$

For $q = 5$,

$$C_5 = \sum_{j=0}^{3} \frac{1}{5!} j^5 \alpha_j - \sum_{j=0}^{3} \frac{1}{4!} j^4 \beta_j - \sum_{j=0}^{3} \frac{1}{3!} j^3 \gamma_j = \begin{bmatrix} \dfrac{1}{24} \\[2mm] \dfrac{1}{45} \\[2mm] -\dfrac{1}{24} \\[2mm] -\dfrac{7}{360} \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.14}$$

The method is order p if $C_0 = C_1 = \cdots = C_{p+1} = 0$ and $C_{p+2} \neq 0$ is the error constant. Thus, we conclude that the method in (2.3) and (2.4) is of the order 3 and the error constant is

$$C_{p+2} = C_5 = \left[\frac{1}{24}, \frac{1}{45}, -\frac{1}{24}, -\frac{7}{360} \right]^{\mathrm{T}}. \tag{2.15}$$

3. Implementation of the Method

The initial starting point at each block are obtained by using Euler method as predictor and the initial h used is as follows:

$$h_{\text{initial}} = \left(\frac{\text{TOL}}{2} \right)^{1/(p+1)}, \tag{3.1}$$

where TOL is the tolerance and p is the order of the method. Then, the calculations are corrected using the corrector formulae in (2.3) and (2.4). For the next block, the

same techniques are repeated to compute the approximation values of y_{n+1} and y_{n+2} simultaneously until the end of the interval. We use r function evaluations per step at the corrector formulae, that is, $PE(CE)^r$, where P, E, and C indicate the predictor, evaluate the function f, and the corrector, respectively.

Algorithm 3.1. Computing approximating $y'^{(p)}_{n+1,0}$, $y^{(p)}_{n+1,0}$, $y'^{(p)}_{n+2,0}$, and $y^{(p)}_{n+2,0}$ using the predictor formulas is as follows:

for $i = 1$ to 2 **do**

$$P : y'^{(p)}_{n+i,0} = y'^{(p)}_{n+(i-1)} + h f^{(p)}_{n+(i-1)},$$

$$y^{(p)}_{n+i,0} = y^{(p)}_{n+(i-1)} + h y'^{(p)}_{n+(i-1)} + \frac{h^2}{2} f^{(p)}_{n+(i-1)}, \tag{3.2}$$

$$E : f\left(x_{n+i}, y^{(p)}_{n+i,0}, y'^{(p)}_{n+i,0}\right)$$

end for

Computing approximations $y'^{(c)}_{n+1,0}$, $y^{(c)}_{n+1,0}$, $y'^{(c)}_{n+2,0}$, and $y^{(c)}_{n+2,0}$ using corrector formulas is as follows:

for $r = 0$ **do**

$$C : y'^{(c)}_{n+1,r+1} = y'^{(c)}_n + \frac{h}{12}\left(5 f^{(c)}_n + 8 f^{(p)}_{n+1,0} - f^{(p)}_{n+2,0}\right),$$

$$y^{(c)}_{n+1,r+1} = y^{(c)}_n + h y'^{(c)}_n + \frac{h^2}{24}\left(7 f^{(c)}_n + 6 f^{(p)}_{n+1,0} - f^{(p)}_{n+2,0}\right),$$

$$y'^{(c)}_{n+2,r+1} = y'^{(c)}_{n+1,r+1} + \frac{h}{12}\left(-f^{(c)}_n + 8 f^{(p)}_{n+1,0} + 5 f^{(p)}_{n+2,0}\right), \tag{3.3}$$

$$y^{(c)}_{n+2,r+1} = y^{(c)}_{n+1,r+1} + h y'^{(c)}_{n+1,r+1} + \frac{h^2}{24}\left(-f^{(c)}_n + 10 f^{(p)}_{n+1,0} + 3 f^{(p)}_{n+2,0}\right),$$

for $i = 1$ to 2 **do**

$$E : f\left(x_{n+i}, y^{(c)}_{n+i,r+1}, y'^{(c)}_{n+i,r+1}\right) \tag{3.4}$$

end for

end for

for $r = 1, 2, \ldots$ until convergence **do**

$$C : y'^{(c)}_{n+1,r+1} = y'^{(c)}_n + \frac{h}{12}\left(5f^{(c)}_n + 8f^{(c)}_{n+1,0} - f^{(c)}_{n+2,0}\right),$$

$$y^{(c)}_{n+1,r+1} = y^{(c)}_n + hy'^{(c)}_n + \frac{h^2}{24}\left(7f^{(c)}_n + 6f^{(c)}_{n+1,0} - f^{(c)}_{n+2,0}\right),$$

$$y'^{(c)}_{n+2,r+1} = y'^{(c)}_{n+1,r+1} + \frac{h}{12}\left(-f^{(c)}_n + 8f^{(c)}_{n+1,0} + 5f^{(c)}_{n+2,0}\right), \tag{3.5}$$

$$y^{(c)}_{n+2,r+1} = y^{(c)}_{n+1,r+1} + hy'^{(c)}_{n+1,r+1} + \frac{h^2}{24}\left(-f^{(c)}_n + 10f^{(c)}_{n+1,0} + 3f^{(c)}_{n+2,0}\right),$$

for $i = 1$ to 2 **do**

$$E : f\left(x_{n+i}, y^{(c)}_{n+i,r+1}, y'^{(c)}_{n+i,r+1}\right) \tag{3.6}$$

end for

end for

The convergence test:

$$\left|\frac{(y_{n+2,r+1})_t - (y_{n+2,r})_t}{A + B(y_{n+2,r+1})_t}\right| < 0.1 \times \text{TOL}, \tag{3.7}$$

where r is the number of iterations and $(y)_t$ is the tth component of the approximate. $A = 1$, $B = 0$ correspond to the absolute error test. $A = 1$, $B = 1$ correspond to the mixed test and finally $A = 0$, $B = 1$ correspond to the relative error test. The mixed error test is used in all tested problems. If the convergence test is satisfied, then we control the error in the current block by performing the local truncation error as follows:

$$\text{LTE} = \left|y^{(k)}_{n+2} - y^{(k-1)}_{n+2}\right|, \tag{3.8}$$

where k is the order of corrector formula. The errors calculated in the code is defined as

$$(E_i)_t = \left|\frac{(y_i)_t - (y(x_i))_t}{A + B(y(x_i))_t}\right|. \tag{3.9}$$

For the evaluation of maximum error, it is defined as follow:

$$\text{MAXE} = \max_{1 \leq i \leq \text{TS}}\left(\max_{1 \leq i \leq N}(E_i)_t\right), \tag{3.10}$$

where N is the number of equation in the system and TS is the number of successful steps.

In order to make the selection of the next step size, we follow again the techniques used in [9]. If LTE $\leq$ TOL, then the next step size remains constant or double, otherwise the step size will be half. The step size when the integration steps are successful is given by

$$h_{\text{new}} = C \times h_{\text{old}} \times \left(\frac{\text{TOL}}{\text{LTE}} \right)^{1/(k+1)},$$

$$\text{if } (h_{\text{new}} \geq 2 \times h_{\text{old}}), \text{ then } h_{\text{next}} = 2 \times h_{\text{old}},$$

$$\text{else } h_{\text{next}} = h_{\text{old}}, \tag{3.11}$$

where $C = 0.5$ is a safety factor. The algorithm when the step failure occurs is

$$h_{\text{next}} = \frac{1}{2} \times h_{\text{old}}. \tag{3.12}$$

4. Stability of the Method

In this section, we will discuss the stability of the proposed method derived in the previous section on a linear general second-order problem:

$$y'' = f = \theta y' + \lambda y. \tag{4.1}$$

Firstly, the test equation in (4.1) is substituted into the predictor formula (3.2). The evaluation of $f^p_{n+1,0}$ and $f^p_{n+2,0}$ will be substituted into the right-hand side of (3.3) when $r = 0$ as shown below:

$$f^p_{n+1,0} = \theta y'^p_{n+1,0} + \lambda y^p_{n+1,0},$$

$$f^p_{n+2,0} = \theta y'^p_{n+2,0} + \lambda y^p_{n+2,0}, \tag{4.2}$$

where $y'^p_{n+1,0}$, $y^p_{n+1,0}$, $y'^p_{n+2,0}$, and $y^p_{n+2,0}$ in (4.2) are obtained in (3.2) after substituted the test equation (4.1). Then, the formulae are written in the matrix form and setting the determinant of the matrix to zero. Hence, the stability polynomial is obtained:

$$r = 0:$$

$$t^2 \left(1 + \frac{25h\theta}{12} + \frac{9h^2\theta^2}{4} + \frac{5h^3\theta^3}{6} + \frac{h^4\lambda^4}{3} + \frac{h^2\lambda}{8} - \frac{h^3\theta\lambda}{2} + \frac{h^4\theta^2\lambda}{12} \right.$$

$$\left. - \frac{h^5\theta^3\lambda}{12} - \frac{37h^4\lambda^2}{144} - \frac{h^5\theta\lambda^2}{9} - \frac{h^6\theta^2\lambda^2}{9} - \frac{h^6\lambda^3}{36} + \frac{h^7\theta\lambda^3}{16} + \frac{h^8\lambda^4}{36} \right)$$
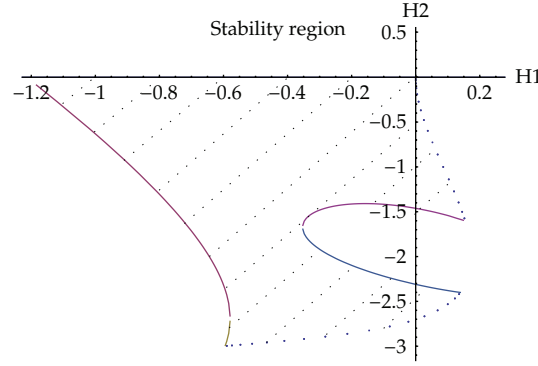
Figure 2: Stability region for direct block one-step method when $r = 0$.

$$+ t^3 \left(-2 - \frac{13h\theta}{6} - \frac{7h^2\theta^2}{3} - \frac{5h^3\theta^3}{6} - \frac{h^4\lambda^4}{3} - \frac{17h^2\lambda}{4} - \frac{47h^3\theta\lambda}{12} - \frac{7h^4\theta^2\lambda}{4} \right.$$

$$\left. - \frac{7h^5\theta^3\lambda}{12} - \frac{115h^4\lambda^2}{72} - \frac{19h^5\theta\lambda^2}{18} - \frac{7h^6\theta^2\lambda^2}{18} - \frac{7h^6\lambda^3}{36} - \frac{17h^7\theta\lambda^3}{144} - \frac{h^8\lambda^4}{72} \right)$$

$$+ t^4 \left(1 + \frac{h\theta}{12} + \frac{h^2\theta^2}{12} + \frac{h^2\lambda}{8} + \frac{h^3\theta\lambda}{12} + \frac{h^4\lambda^2}{48} \right) = 0.$$

(4.3)

Figure 2 showed the stability region of the direct block one-step method when $r = 0$.

In case $r = 1$, the stability polynomial is obtained as follows:

$r = 1$:

$$t^2 \left(1 + 2h\theta + \frac{25h^2\theta^2}{12} + \frac{19h^3\theta^3}{12} + \frac{h^4\theta^4}{3} + \frac{5h^5\theta^5}{18} + \frac{h^6\theta^6}{9} + \frac{7}{48}h^3\theta\lambda - \frac{13}{48}h^4\theta^2\lambda + \frac{1}{4}h^5\theta^3\lambda \right.$$

$$+ \frac{17}{72}h^6\theta^4\lambda + \frac{1}{18}h^7\theta^5\lambda + \frac{11h^4\lambda^2}{288} - \frac{83}{144}h^5\theta\lambda^2 + \frac{101}{432}h^6\theta^2\lambda^2 + \frac{13}{432}h^7\theta^3\lambda^2$$

$$- \frac{17}{432}h^8\theta^4\lambda^2 - \frac{199h^6\lambda^3}{1728} - \frac{53}{288}h^7\theta\lambda^3 - \frac{7}{216}h^8\theta^2\lambda^3 - \frac{5}{432}h^9\theta^3\lambda^3$$

$$\left. - \frac{29h^8\lambda^4}{324} - \frac{17h^9\theta\lambda^4}{1296} + \frac{97h^{10}\theta^2\lambda^4}{5184} - \frac{h^{10}\lambda^5}{648} + \frac{1}{96}h^{11}\theta\lambda^5 + \frac{h^{12}\lambda^6}{648} \right)$$

$$+ t^3 \left(-2 - 2h\theta - \frac{13h^2\theta^2}{6} - \frac{5h^3\theta^3}{3} - \frac{h^4\theta^4}{3} - \frac{5h^5\theta^5}{18} - \frac{h^6\theta^6}{9} - 4h^2\lambda - \frac{103}{24}h^3\theta\lambda \right.$$

$$- \frac{17h^4\theta^2\lambda}{6} - \frac{11h^5\theta^3\lambda}{12} - \frac{19h^6\theta^4\lambda}{24} - \frac{5h^7\theta^5\lambda}{18} - \frac{203h^4\lambda^2}{144} - \frac{199}{144}h^5\theta\lambda^2$$

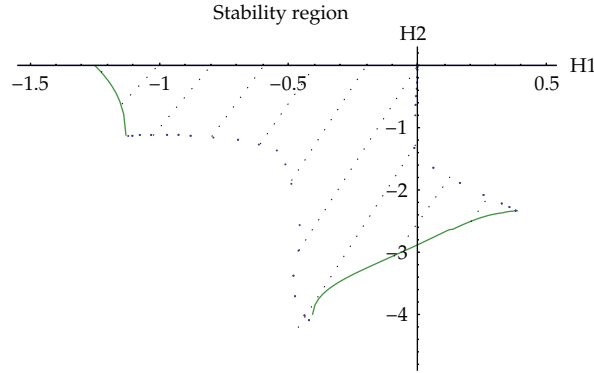**Figure 3:** Stability region for direct block one-step method when $r = 1$.

$$- \frac{365h^6\theta^2\lambda^2}{432} - \frac{361}{432}h^7\theta^3\lambda^2 - \frac{127h^8\theta^4\lambda^2}{432} - \frac{133h^6\lambda^3}{864} - \frac{269h^7\theta\lambda^3}{864} - \frac{23h^8\theta^2\lambda^3}{54}$$

$$- \frac{73h^9\theta^3\lambda^3}{432} - \frac{55h^8\lambda^4}{1296} - \frac{139h^9\theta\lambda^4}{1296} - \frac{289h^{10}\theta^2\lambda^4}{5184} - \frac{7h^{10}\lambda^5}{648} - \frac{13h^{11}\theta\lambda^5}{1296} - \frac{h^{12}\lambda^6}{1296} \Bigg)$$

$$+ t^4 \left(1 + \frac{h^2\theta^2}{12} + \frac{h^3\theta^3}{12} + \frac{7}{48}h^3\theta\lambda + \frac{5}{48}h^4\theta^2\lambda + \frac{11h^4\lambda^2}{288} + \frac{1}{24}h^5\theta\lambda^2 + \frac{h^6\lambda^3}{192} \right) = 0.$$

$$(4.4)$$

Figure 3 showed the stability region of the direct block one-step method when $r = 1$.

The stability region is plotted using MATHEMATICA, and the shaded region inside the boundary in Figures 2 and 3 demonstrate the stability region for the proposed method.

5. Numerical Results and Discussion

We have tested the performance of the proposed method on four problems. For the first three problems, a comparison is made between the solutions obtained in [9].

Problem 1. We have

$$y_1'' = -y_2 + \sin \pi x, \qquad y_1(0) = 0, \qquad y_1'(0) = -1,$$

$$y_2'' = -y_1 + 1 - \pi^2\sin \pi x, \qquad y_2(0) = 1, \qquad y_2'(0) = 1 + \pi, \qquad [0, 10].$$

$$(5.1)$$

Solution: $y_1(x) = 1 - e^x$, $y_2(x) = e^x + \sin \pi x$.
First-order systems:

$$y_1' = y_3, \qquad y_2' = y_4, \qquad y_3' = -y_2 + \sin \pi x, \qquad y_4' = -y_1 + 1 - \pi^2 \sin \pi x,$$

$$y_1(0) = 0, \qquad y_2(0) = 1, \qquad y_3(0) = -1, \qquad y_4(0) = 1 + \pi.$$

$$(5.2)$$

Solution: $y_1(x) = 1 - e^x, y_2(x) = e^x + \sin \pi x, y_3(x) = -e^x, y_4(x) = e^x + \pi \cos \pi x$.

Problem 2. We have

$$y_1'' = -e^{-x}y_2, \qquad y_1(0) = 1, \qquad y_1'(0) = 0,$$
$$y_2'' = 2e^{x}y_1', \qquad y_2(0) = 1, \qquad y_2'(0) = 1, \quad [0, \pi].$$

$$(5.3)$$

Solution: $y_1(x) = \cos x$, $y_2(x) = e^x \cos x$.
First-order systems:

$$y_1' = y_3, \qquad y_2' = y_4, \qquad y_3' = -e^{-x}y_2, \qquad y_4' = 2e^{x}y_3,$$
$$y_1(0) = 1, \qquad y_2(0) = 1, \qquad y_3(0) = 0, \qquad y_4(0) = 1.$$

$$(5.4)$$

Solution: $y_1(x) = \cos x$, $y_2(x) = e^x \cos x$, $y_3(x) = -\sin x$, $y_4(x) = e^x \cos x - e^x \sin x$.

Problem 3. We have

$$y_1'' = -y_2', \qquad y_1(0) = 0, \qquad y_1'(0) = \frac{1}{1 - e^{-1}},$$
$$y_2'' = -y_1', \qquad y_2(0) = 1, \qquad y_2'(0) = \frac{1}{1 - e^{-1}}, \quad [0, 10].$$

$$(5.5)$$

Solution: $y_1(x) = (1 - e^{-x})/(1 - e^{-1})$, $y_2(x) = (2 - e^{-1} - e^{-x})/(1 - e^{-1})$.
First-order systems:

$$y_1' = y_3, \qquad y_2' = y_4, \qquad y_3' = -y_4, \qquad y_4' = -y_3,$$
$$y_1(0) = 0, \qquad y_2(0) = 1, \qquad y_3(0) = \frac{1}{1 - e^{-1}}, \qquad y_4(0) = \frac{1}{1 - e^{-1}}.$$

$$(5.6)$$

Solution: $y_1(x) = (1 - e^{-x})/(1 - e^{-1})$, $y_2(x) = (2 - e^{-1} - e^{-x})/(1 - e^{-1})$, $y_3(x) = (e^{-x})/(1 - e^{-1})$, $y_4(x) = e^{-x}/(1 - e^{-1})$.

Problem 4 (Van Der Pol oscillator). We have

$$y'' - 2\xi\left(1 - y^2\right)y' + y = 0, \qquad y(0) = 0, \qquad y'(0) = 0.5, \quad [0, 10], \qquad (5.7)$$

where we take $\xi = 0.025$.

The codes are written in C language and executed on DYNIX/ptx operating system. The numerical results for Problems 1–3 in Tables 2–4 are solved using the proposed method and the method in [9]. The results in terms of total steps and execution times for solving Problems 1–3 are presented in histograms and graph lines in Figures 4, 5 and 6. The solutions of 2P(B) for solving Problem 4 are plotted in Figure 7 at tolerance 10^{-8} and compared with the solutions obtained by the MATLAB built-in solver **ode45**.

In Figures 4, 5 and 6, it is obvious that method 2P(B) requires less number of total steps as compared to method 2P(A) when solving the same given problems. It is also observed that

Table 1: The ratios execution times and steps for solving Problems 1 to 3.

TOL	PROB 1		PROB 2		PROB 3	
	RSTEP	RTIME	RSTEP	RTIME	RSTEP	RTIME
10^{-2}	2.53	1.90	2.83	1.55	2.56	1.20
10^{-4}	2.38	2.31	2.88	1.57	2.45	1.32
10^{-6}	2.89	3.05	3.11	2.84	3.19	2.30
10^{-8}	4.74	5.26	4.34	1.49	3.47	3.17

Table 2: Comparison between 2P(A) and 2P(B) methods for solving Problem 1.

TOL	Method	TS	FS	MAXE	AVERR	FCN	TIME
10^{-2}	2P(A)	48	2	$3.20591e-004$	$1.28467e-004$	385	1002
	2P(B)	19	0	$3.88324e-003$	$2.89565e-003$	121	527
10^{-4}	2P(A)	143	2	$3.69289e-006$	$1.48547e-006$	1145	2423
	2P(B)	60	0	$3.46171e-005$	$2.81402e-005$	367	1047
10^{-6}	2P(A)	546	3	$2.91502e-008$	$7.74544e-009$	4369	9287
	2P(B)	189	0	$3.43754e-007$	$2.76828e-007$	1133	3045
10^{-8}	2P(A)	2819	3	$2.43829e-011$	$9.79663e-012$	22553	47821
	2P(B)	595	0	$3.44148e-009$	$2.75811e-009$	3571	9093

the execution times of 2P(B) are faster than 2P(A) at all tested tolerances. This is expected since 2P(B) has less number of function calls; therefore, it has affected the computation time of 2P(B).

In Table 1, the RSTEP and RTIME are greater than 1.00 which shows that 2P(B) is more efficient compared to 2P(A). In fact, in some cases, the ratios are greater than 3.00, which indicates a clear advantage of method 2P(B) over 2P(A).

In Table 2, it can be observed that the maximum error of 2P(B) is one or two order larger than 2P(A) but still acceptable as it is within the given tolerance. This is expected since the code 2P(B) solved the given problem directly without reducing to system of first-order differential equations. We could observe that the RSTEP for Problem 1 is greater than 2 in Table 1.

In Tables 3 and 4, it is observed that the maximum error of 2P(B) is two or three order larger as compared to 2P(A) in solving Problems 2 and 3 but it is still within the given tolerances. In Figure 7, it is obvious that the 2P(B) solutions agree very well with the solutions obtained by the MATLAB built-in solver **ode45** when solving Problem 4.

6. Conclusion

In this paper, we have constructed the direct two-point block one-step method which is efficient and suitable for solving general second-order ODEs directly. The block method has shown acceptable solutions and managed to solve the second-order ODE faster compared to the existing method.
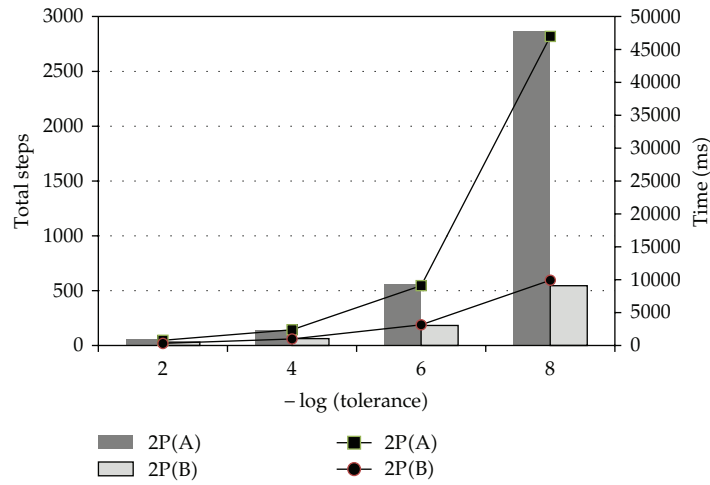
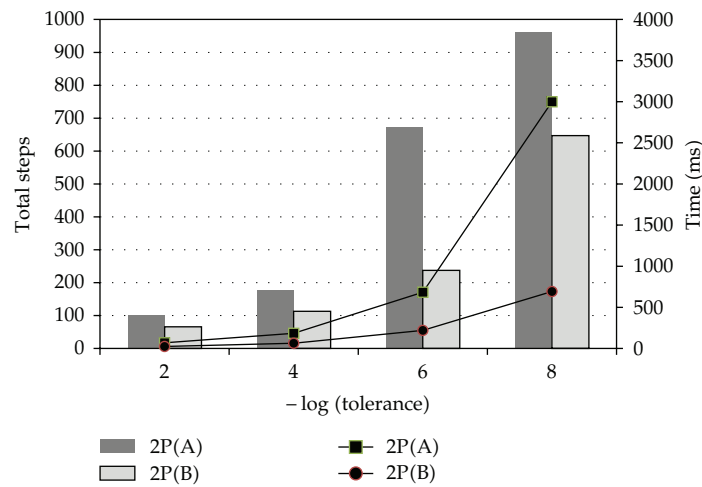**Figure 4:** Results of total steps and time for Problem 1.

Figure 5: Results of total steps and time for Problem 2.

Table 3: Comparison between 2P(A) and 2P(B) methods for solving Problem 2.

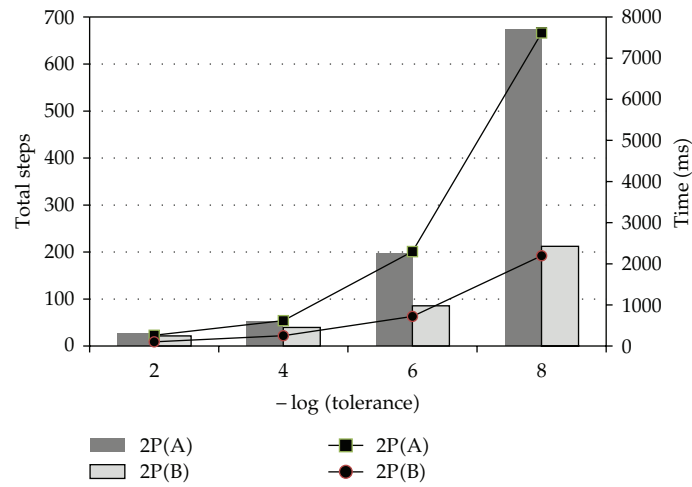| TOL	Method	TS	FS	MAXE	AVERR	FCN	TIME
10^{-2}	2P(A)	17	4	$4.64239e - 004$	$1.01900e - 004$	137	408
	2P(B)	6	0	$1.29794e - 003$	$2.48275e - 004$	47	264
10^{-4}	2P(A)	46	3	$5.27751e - 006$	$9.23287e - 007$	369	708
	2P(B)	16	1	$4.19030e - 005$	$7.96402e - 006$	131	452
10^{-6}	2P(A)	171	2	$2.35992e - 008$	$5.04232e - 009$	1369	2694
	2P(B)	55	1	$4.24624e - 007$	$6.60118e - 008$	429	950
10^{-8}	2P(A)	750	2	$4.50146e - 011$	$7.04651e - 012$	1741	3844
	2P(B)	173	1	$4.31085e - 009$	$6.26858e - 010$	1343	2588

Figure 6: Results of total steps and time for Problem 3.
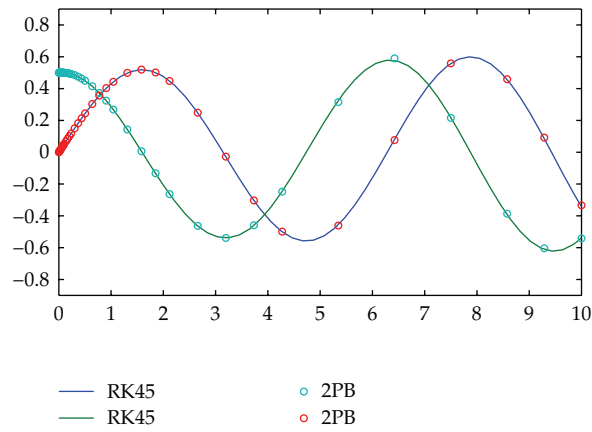

Figure 7: Solution of Problem 4 when TOL = 10^{-8}.

Table 4: Comparison between 2P(A) and 2P(B) methods for solving Problem 3.

TOL	Method	TS	FS	MAXE	AVERR	FCN	TIME
10^{-2}	2P(A)	23	7	$1.42058e-004$	$3.37359e-005$	185	296
	2P(B)	9	0	$1.47268e-003$	$4.24039e-004$	69	247
10^{-4}	2P(A)	54	2	$1.41231e-006$	$2.21945e-007$	433	597
	2P(B)	22	0	$2.73310e-005$	$9.95954e-006$	177	451
10^{-6}	2P(A)	201	1	$6.04402e-009$	$9.73203e-010$	1609	2252
	2P(B)	63	0	$9.51564e-007$	$2.58666e-007$	503	980
10^{-8}	2P(A)	666	1	$4.12406e-011$	$8.11194e-012$	5329	7700
	2P(B)	192	0	$1.25592e-008$	$2.60498e-009$	1537	2426

Notations

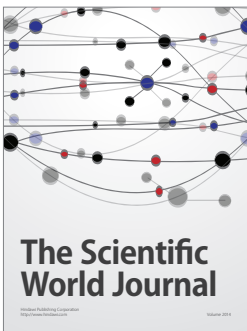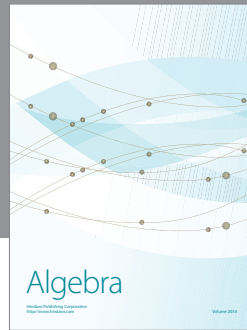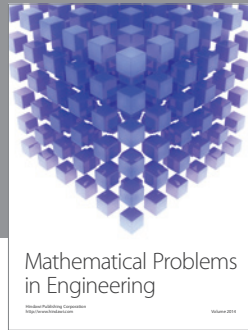|	
TOL:	Tolerance
MTD:	Method employed
TS:	Total steps taken
FS:	Total failure steps
MAXE:	Magnitude of the maximum error of the computed solution
AVERR:	The average error
FCN:	Total function calls
TIME:	The execution time taken in microseconds
2P(A):	Implementation of the direct block method in [9] by reducing the problem to system of first-order ODEs
2P(B):	Implementation of the direct two-point block one-step method by solving the problem directly
RSTEP:	The ratio steps, $\text{TS}_{2P(A)}/\text{TS}_{2P(B)}$
RTIME:	The ratio execution times, $\text{TIME}_{2P(A)}/\text{TIME}_{2P(B)}$

Acknowledgment

References

[1] J. B. Rosser, "A Runge-Kutta for all seasons," *SIAM Review*, vol. 9, pp. 417–452, 1967.

[2] L. F. Shampine and H. A. Watts, "Block implicit one-step methods," *Mathematics of Computation*, vol. 23, pp. 731–740, 1969.

[3] P. B. Worland, "Parallel methods for the numerical solution of ordinary differential equations," *Institute of Electrical and Electronics Engineers*, vol. 25, no. 10, pp. 1045–1048, 1976.

[4] S. Mehrkanoon, Z. A. Majid, and M. Suleiman, "A variable step implicit block multistep method for solving first-order ODEs," *Journal of Computational and Applied Mathematics*, vol. 233, no. 9, pp. 2387–2394, 2010.

[5] Z. A. Majid and M. B. Suleiman, "Implementation of four-point fully implicit block method for solving ordinary differential equations," *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 514–522, 2007.

[6] Z. A. Majid, N. A. Azmi, and M. Suleiman, "Solving second order ordinary differential equations using two point four step direct implicit block method," *European Journal of Scientific Research*, vol. 31, no. 1, pp. 29–36, 2009.

[7] Z. B. Ibrahim, M. B. Suleiman, and F. Ismail, "Fully implicit two point block backward difference formula for solving a first order initial value problems," *Science Putra Research Bulletin*, vol. 11, no. 2, pp. 14–17, 2003.

[8] P. C. Chakravarti and P. B. Worland, "A class of self-starting methods for the numerical solution of $y'' = f(x, y)$," *BIT*, vol. 11, pp. 368–383, 1971.

[9] Z. A. Majid, M. B. Suleiman, F. Ismail, and M. Othman, "2-point implicit block one-step method half Gauss-Seidel for solving first order ordinary differential equations," *Matematika*, vol. 19, pp. 91–100, 2003.

[10] S. O. Fatunla, "Block methods for second order ODEs," *International Journal of Computer Mathematics*, vol. 41, pp. 55–63, 1991.

[11] J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons, New York, NY, USA, 1973.

[12] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.