Hindawi Publishing Corporation
International Journal of Mathematics and Mathematical Sciences
Volume 2016, Article ID 9823147, 8 pages
http://dx.doi.org/10.1155/2016/9823147

Research Article

A 4-Point Block Method for Solving Higher Order Ordinary Differential Equations Directly

Nazreen Waeleh[1] and Zanariah Abdul Majid[2,3]

[1]*Faculty of Electronic & Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM), 76100 Melaka, Malaysia*
[2]*Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*
[3]*Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*

Correspondence should be addressed to Nazreen Waeleh; nazreen@utem.edu.my

Received 18 April 2016; Revised 23 June 2016; Accepted 30 June 2016

Academic Editor: Harvinder S. Sidhu

An alternative block method for solving fifth-order initial value problems (IVPs) is proposed with an adaptive strategy of implementing variable step size. The derived method is designed to compute four solutions simultaneously without reducing the problem to a system of first-order IVPs. To validate the proposed method, the consistency and zero stability are also discussed. The improved performance of the developed method is demonstrated by comparing it with the existing methods and the results showed that the 4-point block method is suitable for solving fifth-order IVPs.

1. Introduction

Many natural processes or real-world problems can be translated into the language of mathematics [1–4]. The mathematical formulation of physical phenomena in science and engineering often leads to a differential equation, which can be categorized as an ordinary differential equation (ODE) and a partial differential equation (PDE). This formulation will explain the behavior of the phenomenon in detail. The search for solutions of real-world problems requires solving ODEs and thus has been an important aspect of mathematical study. For many interesting applications, an exact solution may be unattainable, or it may not give the answer in a convenient form. The reliability of numerical approximation techniques in solving such problems has been proven by many researchers as the role of numerical methods in engineering problems solving has increased dramatically in recent years. Thus a numerical approach has been chosen as an alternative tool for approximating the solutions consistent with the advancement in technology.

Commonly, the formulation of real-world problems will take the form of a higher order differential equation associated with its initial or boundary conditions [4]. In the literature, a mathematical model in the form of a fifth-order differential equation, known as Korteweg-de Vries (KdV) equation, has been used to describe several wave phenomena depending on the values of its parameters [2, 3, 5, 6]. The KdV equation is a PDE and researchers have tackled the problem analytically and numerically. It is also noted that in certain cases by using different approaches the KdV might be transformed into a higher order ODE [7]. To date, there are a number of studies that have proposed solving fifth-order ODE directly [8, 9]. Hence, the purpose of the present paper is to solve directly the fifth-order IVPs with the implementation of a variable step size strategy. The fifth-order IVP with its initial conditions is defined as

$$y^{v} = f\left(x, y, y', y'', y''', y^{iv}\right), \quad y(a) = y_0,\ y'(a) = y_1,\ y''(a) = y_2,\ y'''(a) = y_3,\ y^{iv}(a) = y_4,\ x \in [a, b]. \tag{1}$$

Conventionally, (1) will be converted to a system of first-order ODEs by a simple change of variables. However, it will increase the computational cost in terms of function evaluation and thus will affect the computational time. This drawback is obviously seen when dealing with a higher order problem. Furthermore, [10] also has remarked that the block method is far more cost-effective when it is implemented in direct integration. Hence, several researchers [11–16] have shown an interest in the development of direct integration methods. A direct integration method of variable order and step size for solving systems of nonstiff higher order ODEs has been discussed in [11] whereby [12] has proposed an algorithm based on collocation of the differential system at selected grid points for direct solution of general second-order ODEs. In addition, [13] has used the Gaussian method in order to solve fourth-order differential equations directly. However, it requires a tedious computation as well, since it consists of higher order partial derivatives of Taylor series algorithm which supplies the starting values. Jator and Li [15] have proposed the linear multistep method (LMM) for solving general second-order IVPs directly. The method is self-starting, so it involves less computational time by avoiding incorporating subroutines to supply the starting values.

Thus far, a number of researchers have concerned themselves with developing a numerical method based on block features, and the characteristic feature of the block method is that in each application it generates a set of solutions concurrently [10]. Rosser [10] also has remarked that the implementation of block method in numerical computation will reduce the computational cost by reducing the number of function evaluations. Shampine and Watts [17] have constructed an A-stable implicit one-step block method and Cash [18] has studied block methods based upon the Runge-Kutta method for the numerical solution of nonstiff IVPs. Furthermore [19] has used the self-starting LMM to solve second-order ODEs in a block-by-block fashion and recently [20] has constructed a predictor-corrector scheme 3-point block method with the implementation of variable step size. This research is an extension of the work in [20] in which the solution is computed at three points concurrently and it shows the satisfactory numerical results obtained when solving general higher order ODEs.

An increasing amount of literature is devoted to variable step size implementations of numerical methods [11, 21, 22]. The practicality of varying the step size for block method has been justified by [10]. This strategy is an attempt to reduce the computational cost as well as maintaining the accuracy. The Falkner method with variable step size implementation for the numerical solution of second-order IVPs has been employed in [21]. Although the implementation of the method involves varying the step size and solving directly, the computation is still tedious since the coefficients of the formulae must be calculated every time the step size is changed. On the contrary, the present work will store all the integration coefficients in the code in order to avoid the tedious calculations of the divided differences.

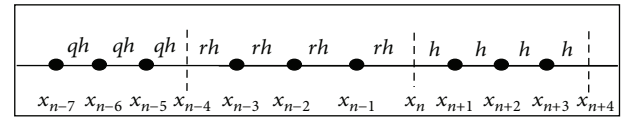

Figure 1: 4-point block method.

2. Methodology

2.1. Derivation of 4-Point Block Method. The basic approach of numerical methods for integration is performed by subdividing the interval of integration into certain subintervals. The proposed method was based on concurrent computation; hence the closed finite interval was subdivided into a series of blocks and each block contains four equal subintervals as illustrated in Figure 1.

Initially, (1) was integrated five times over the corresponding interval: $[x_n, x_{n+1}]$, $[x_n, x_{n+2}]$, $[x_n, x_{n+3}]$, $[x_n, x_{n+4}]$ for first, second, third, and fourth point, respectively. The integration was started by replacing the function $f(x, y, y', y'', y''', y^{\mathrm{iv}})$ with the interpolating function which was generated from Lagrange polynomials. A set of points $\{(x_{n-7}, f_{n-7}), \ldots, (x_n, f_n)\}, \{(x_{n-4}, f_{n-4}), \ldots, (x_{n+4}, f_{n+4})\}$ was interpolated for deriving predictor and corrector formulae, respectively. Let the Lagrange polynomial, $P_w(x)$, be written as

$$P_w(x) = L_{w,0}(x) f(x_{n+4}) + L_{w,1} f(x_{n+3}) + \cdots + L_{w,w} f(x_{n+4-w}) = \sum_{j=0}^{w} L_{w,j}(x) f(x_{n+4-j}), \tag{2}$$

where

$$L_{w,j}(x) = \prod_{\substack{i=0 \\ i \neq j}}^{w} \frac{(x - x_{n+4-i})}{(x_{n+4-j} - x_{n+4-i})} \quad \text{for each } j = 0, 1, \ldots, w. \tag{3}$$

Nine points were interpolated in (2) with w set to be eight for deriving the corrector and thus one point less for the predictor formula. Then, the integration process was proceeded by substituting $z = (x - x_{n+4})/h$ and $dx = hdz$ in (2). Consistent with the number of interpolation points involved in deriving the formulae, predictor and corrector formulae were obtained in terms of variables r and q. The variables r and q refer to the distance ratio between current and previous point as a result of implementation variable step size strategy in the proposed method.

In this work, the selection of the next step size could be increased by a factor of $(r = 0.5, q = 0.5)$ or maintained by $((r = 1, q = 1), (r = 1, q = 2), (r = 1, q = 0.5))$ and $(r = 2, q = 2)$ for halving the current step size. This step size changing technique was limited in order to minimize the number of coefficients to be stored as well as reducing the computational storage. The increment of step size was also limited to doubling in order to control the accuracy of

the computation [10]. The compact form of the 4-point block method is presented in

$$y_{n+d}^{(\mathrm{v}-g)} = \sum_{k=0}^{g-1} \frac{(dh)^k}{k!} y_n^{(\mathrm{v}-g+k)} + \frac{h^g}{(g-1)!} \sum_{j=s}^{t} \gamma_{d,j}^{g} f_{n+j}, \quad (4)$$

where $\gamma_{d,j}^{g}$ are the coefficients of the formulae to be calculated, d is the number of points ($d = 1, 2, 3, 4$), g is the number of times (1) will be integrated, and k is the number of terms when the equation is integrated. The values of $s = -7$, $t = 0$ and $s = -4$, $t = 4$ were considered for deriving the predictor and corrector formulae, respectively. After further simplification, the associated corrector formulae of the 4-point block method when $r = 1$ are represented below.

Integrate once:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{n+4}^{\mathrm{iv}} \\ y_{n+3}^{\mathrm{iv}} \\ y_{n+2}^{\mathrm{iv}} \\ y_{n+1}^{\mathrm{iv}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{n}^{\mathrm{iv}} \\ y_{n-1}^{\mathrm{iv}} \\ y_{n-2}^{\mathrm{iv}} \\ y_{n-3}^{\mathrm{iv}} \end{bmatrix} + \frac{h}{3628800} \left(\begin{bmatrix} -3233 & 36394 & -216014 & 1909858 \\ 4064 & -63232 & 1422272 & 4541696 \\ -29889 & 1312362 & 4667058 & 2789154 \\ 1040128 & 5779456 & 62464 & 8384512 \end{bmatrix} \begin{bmatrix} f_{n+4} \\ f_{n+3} \\ f_{n+2} \\ f_{n+1} \end{bmatrix} \right. \quad (5)$$

$$\left. + \begin{bmatrix} 2224480 & -425762 & 126286 & -25706 \\ 1391360 & -27904 & -15808 & 5888 \\ 2708640 & -782946 & 278478 & -63018 \\ -2324480 & 2363392 & -1012736 & 249856 \end{bmatrix} \begin{bmatrix} f_{n} \\ f_{n-1} \\ f_{n-2} \\ f_{n-3} \end{bmatrix} + \begin{bmatrix} 2497 & 0 & 0 & 0 \\ -736 & 0 & 0 & 0 \\ 6561 & 0 & 0 & 0 \\ -27392 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{n-4} \\ f_{n-5} \\ f_{n-6} \\ f_{n-7} \end{bmatrix} \right).$$

Integrate twice:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{n+4}''' \\ y_{n+3}''' \\ y_{n+2}''' \\ y_{n+1}''' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{n}''' \\ y_{n-1}''' \\ y_{n-2}''' \\ y_{n-3}''' \end{bmatrix} + h \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{n}^{\mathrm{iv}} \\ y_{n-1}^{\mathrm{iv}} \\ y_{n-2}^{\mathrm{iv}} \\ y_{n-3}^{\mathrm{iv}} \end{bmatrix}$$

$$+ \frac{h^2}{7257600} \left(\begin{bmatrix} -3057 & 34208 & -197216 & 1258488 \\ -3008 & 20480 & 370944 & 8341504 \\ -22599 & 659016 & 7027560 & 15769728 \\ 433152 & 8093696 & 12378112 & 26050560 \end{bmatrix} \begin{bmatrix} f_{n+4} \\ f_{n+3} \\ f_{n+2} \\ f_{n+1} \end{bmatrix} \right. \quad (6)$$

$$\left. + \begin{bmatrix} 2875850 & -444560 & 128472 & -25864 \\ 6492800 & -909312 & 244480 & -47104 \\ 10485450 & -1650456 & 478224 & -97200 \\ 11724800 & -622592 & -24576 & 32768 \end{bmatrix} \begin{bmatrix} f_{n} \\ f_{n-1} \\ f_{n-2} \\ f_{n-3} \end{bmatrix} + \begin{bmatrix} 2497 & 0 & 0 & 0 \\ 4416 & 0 & 0 & 0 \\ 9477 & 0 & 0 & 0 \\ -5120 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{n-4} \\ f_{n-5} \\ f_{n-6} \\ f_{n-7} \end{bmatrix} \right).$$

Integrate thrice:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y''_{n+4} \\ y''_{n+3} \\ y''_{n+2} \\ y''_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y''_{n} \\ y''_{n-1} \\ y''_{n-2} \\ y''_{n-3} \end{bmatrix} + h \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y'''_{n} \\ y'''_{n-1} \\ y'''_{n-2} \\ y'''_{n-3} \end{bmatrix} + \frac{h^2}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y^{\mathrm{iv}}_{n} \\ y^{\mathrm{iv}}_{n-1} \\ y^{\mathrm{iv}}_{n-2} \\ y^{\mathrm{iv}}_{n-3} \end{bmatrix}$$

$$+ \frac{h^3}{19958400} \left(\begin{bmatrix} -4872 & 53782 & -304397 & 1693482 \\ -29632 & 313088 & -1292928 & 25524992 \\ -63423 & 1004562 & 15809823 & 93765438 \\ 218112 & 20512768 & 74186752 & 202702848 \end{bmatrix} \begin{bmatrix} f_{n+4} \\ f_{n+3} \\ f_{n+2} \\ f_{n+1} \end{bmatrix} \right. \tag{7}$$

$$\left. + \begin{bmatrix} 5791735 & -751598 & 213153 & -42578 \\ 32478080 & -4915968 & 1396352 & -278272 \\ 77715045 & -11123082 & 3059613 & -599238 \\ 144240640 & -20774912 & 5701632 & -1114112 \end{bmatrix} \begin{bmatrix} f_{n} \\ f_{n-1} \\ f_{n-2} \\ f_{n-3} \end{bmatrix} + \begin{bmatrix} 4093 & 0 & 0 & 0 \\ 26688 & 0 & 0 & 0 \\ 56862 & 0 & 0 & 0 \\ 105472 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{n-4} \\ f_{n-5} \\ f_{n-6} \\ f_{n-7} \end{bmatrix} \right).$$

Integrate four times:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y'_{n+4} \\ y'_{n+3} \\ y'_{n+2} \\ y'_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y'_{n} \\ y'_{n-1} \\ y'_{n-2} \\ y'_{n-3} \end{bmatrix} + h \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y''_{n} \\ y''_{n-1} \\ y''_{n-2} \\ y''_{n-3} \end{bmatrix} + \frac{h^2}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y'''_{n} \\ y'''_{n-1} \\ y'''_{n-2} \\ y'''_{n-3} \end{bmatrix} + \frac{h^3}{6}$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 27 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y^{\mathrm{iv}}_{n} \\ y^{\mathrm{iv}}_{n-1} \\ y^{\mathrm{iv}}_{n-2} \\ y^{\mathrm{iv}}_{n-3} \end{bmatrix} + \frac{h^4}{159667200} \left(\begin{bmatrix} -25143 & 276056 & -1542812 & 7955976 \\ -437504 & 4788224 & -25525248 & 255250432 \\ -1358127 & 14924088 & 74244276 & 1604715624 \\ -3342336 & 190840832 & 1093402624 & 5052039168 \end{bmatrix} \begin{bmatrix} f_{n+4} \\ f_{n+3} \\ f_{n+2} \\ f_{n+1} \end{bmatrix} \right. \tag{8}$$

$$\left. + \begin{bmatrix} 679888370 & -66798970 & 18463200 & -3649810 \\ 455836160 & -15732736 & 19158016 & -3837952 \\ 1734575310 & -251470008 & 70150212 & -13839336 \\ 4381736960 & -643825664 & 180092928 & -35651584 \end{bmatrix} \begin{bmatrix} f_{n} \\ f_{n-1} \\ f_{n-2} \\ f_{n-3} \end{bmatrix} + \begin{bmatrix} 21649 & 0 & 0 & 0 \\ 369408 & 0 & 0 & 0 \\ 1318761 & 0 & 0 & 0 \\ 3407872 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{n-4} \\ f_{n-5} \\ f_{n-6} \\ f_{n-7} \end{bmatrix} \right).$$

Integrate five times:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{n+4} \\ y_{n+3} \\ y_{n+2} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_n \\ y_{n-1} \\ y_{n-2} \\ y_{n-3} \end{bmatrix} + h \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y'_n \\ y'_{n-1} \\ y'_{n-2} \\ y'_{n-3} \end{bmatrix} + \frac{h^2}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y''_n \\ y''_{n-1} \\ y''_{n-2} \\ y''_{n-3} \end{bmatrix} + \frac{h^3}{6}$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 27 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y'''_n \\ y'''_{n-1} \\ y'''_{n-2} \\ y'''_{n-3} \end{bmatrix} + \frac{h^4}{24} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \\ 81 & 0 & 0 & 0 \\ 256 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y^{\mathrm{iv}}_n \\ y^{\mathrm{iv}}_{n-1} \\ y^{\mathrm{iv}}_{n-2} \\ y^{\mathrm{iv}}_{n-3} \end{bmatrix} \tag{9}$$

$$+ \frac{h^5}{3632428800} \left(\begin{bmatrix} -397695 & 4349090 & -24084760 & 118367466 \\ -16628480 & 183347200 & -1022977536 & 7852902400 \\ -93592665 & 995920434 & -1796664240 & 79986003930 \\ -329908224 & 6980894720 & 40512389120 & 363276533800 \end{bmatrix} \begin{bmatrix} f_{n+4} \\ f_{n+3} \\ f_{n+2} \\ f_{n+1} \end{bmatrix} \right.$$

$$\left. + \begin{bmatrix} 679888370 & -66798970 & 18463200 & -3649810 \\ 18183101440 & -2522204160 & 720401920 & -144287744 \\ 109670461100 & -15912327690 & 4493865096 & -893148930 \\ 375644487700 & -54760308740 & 1534525440 & -3038248960 \end{bmatrix} \begin{bmatrix} f_n \\ f_{n-1} \\ f_{n-2} \\ f_{n-3} \end{bmatrix} + \begin{bmatrix} 348869 & 0 & 0 & 0 \\ 13889280 & 0 & 0 & 0 \\ 85522635 & 0 & 0 & 0 \\ 1095172096 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{n-4} \\ f_{n-5} \\ f_{n-6} \\ f_{n-7} \end{bmatrix} \right).$$

2.2. Order and Convergence of the Method. The matrix differential equation of the derived method is given as

$$\alpha Y_m = h\beta Y'_m + h^2 \lambda Y''_m + h^3 \mu Y'''_m + h^4 \sigma Y^{\mathrm{iv}}_m + h^5 \theta F_m, \tag{10}$$

where $\alpha, \beta, \lambda, \mu, \sigma$, and θ are the coefficients of the developed method. Consequently, the order of 4-point block method can be determined using the following formulae:

$$\begin{aligned} C_0 &= \sum_{j=0}^{19} \alpha_j, \\ C_1 &= \sum_{j=0}^{19} \left(j\alpha_j - \beta_j \right), \\ C_2 &= \sum_{j=0}^{19} \left(\frac{j^2}{2!}\alpha_j - j\beta_j - \lambda_j \right), \\ &\vdots \\ C_q &= \sum_{j=0}^{19} \left(\frac{j^q}{q!}\alpha_j - \frac{j^{q-1}}{(q-1)!}\beta_j - \frac{j^{q-2}}{(q-2)!}\lambda_j \right. \\ &\quad \left. - \frac{j^{q-3}}{(q-3)!}\mu_j - \frac{j^{q-4}}{(q-4)!}\sigma_j - \frac{j^{q-5}}{(q-5)!}\theta_j \right). \end{aligned} \tag{11}$$

As a result, a 4-point block method of order nine is developed with $C_{p+5} \neq 0$ and the error constant obtained is given as

$$\begin{aligned} C_{14} = \Bigg[& \frac{2497}{7257600}, \frac{40321}{239500800}, \frac{47357}{479001600}, \\ & \frac{2818273}{43589145600}, \frac{164971}{3632428800}, -\frac{23}{113400}, \frac{481}{1871100}, \\ & \frac{1187}{1871100}, \frac{95143}{85135050}, \frac{8753}{4729725}, \frac{113}{89600}, \frac{689}{985600}, \\ & \frac{2559}{1971200}, \frac{683073}{179379200}, \frac{497799}{44844800}, -\frac{94}{14175}, \\ & -\frac{568}{467775}, \frac{1088}{467775}, \frac{429568}{42567525}, \frac{532736}{14189175} \Bigg]^T. \end{aligned} \tag{12}$$

Hence, the consistency of 4-point block method is proven according to the definition in [23]. The analysis of zero stability for the developed method is tested using a similar approach as presented in [24] and the first characteristic polynomial obtained is $\rho(R) = R^3(R-1) = 0$. It is clearly seen that the roots are 0 and 1. Thus from Theorem 2.1 in [23], the convergence of the proposed method is asserted.

3. Implementation

Throughout this section, the implementation of 4-point block method for solving fifth-order IVPs will be explained in detail. The code starts by finding the values of y in the initial

Table 1: Numerical results for solving Problem 1.

TOL	MTD	TS	MAXE	AVERR	FC
10^{-2}	ode45	11	1.743 (−6)	2.244 (−6)	67
	4P1FI	23	1.640 (−4)	5.346 (−7)	210
	4PHODE	20	9.031 (−9)	1.611 (−10)	186
10^{-4}	ode45	11	1.641 (−6)	2.186 (−6)	67
	4P1FI	37	9.727 (−7)	6.988 (−9)	339
	4PHODE	27	7.059 (−10)	1.092 (−11)	243
10^{-6}	ode45	18	5.234 (−7)	8.596 (−7)	109
	4P1FI	50	5.582 (−9)	5.727 (−11)	447
	4PHODE	35	1.009 (−9)	3.401 (−11)	307
10^{-8}	ode45	44	5.485 (−9)	9.472 (−9)	265
	4P1FI	96	7.597 (−11)	2.881 (−13)	799
	4PHODE	42	1.838 (−10)	2.619 (−12)	363
10^{-10}	ode45	108	5.552 (−11)	9.551 (−11)	649
	4P1FI	788	1.834 (−14)	2.554 (−15)	6331
	4PHODE	50	1.023 (−10)	3.443 (−12)	439
KAYODE (a)		Not stated	1.638 (−6)	Not stated	Not stated
KAYODE (b)		100	5.082 (−7)	Not stated	Not stated

block using Euler method. However, it should be noted that Euler method will act only as a fundamental building block. Then the 4-point block method will be applied until the end of the interval. As stated earlier, the proposed method is implemented in the mode of predicting and correcting. In order to preserve the accuracy, the step will succeed if the local truncation error (LTE) is less than the specified error tolerance (TOL) such that

$$\text{LTE} = \left| y_{n+4}^{t} - y_{n+4}^{t-1} \right| < \text{TOL}, \tag{13}$$

where y_{n+4}^{t} and y_{n+4}^{t-1} are the corrector value of y at the last point for each block with t iterations. If (13) is satisfied, the new step size will be calculated via the step size increment formula. Otherwise, the current step size will be reduced by half. The step size increment formula is defined as

$$h_{\text{new}} = \delta \times h_{\text{old}} \times \left(\frac{\text{TOL}}{2 \times \text{LTE}} \right)^{1/m}, \tag{14}$$

where h_{new} and h_{old} denote the current and previous step size, respectively, with value of 0.5 for safety factor (δ) and m is the order of corrector formulae. To show the accuracy and efficiency of the proposed method, the computational errors will be reported, equal to

$$E_i = \left| \frac{y_i - y(x_i)}{A + B(y(x_i))} \right|. \tag{15}$$

Different values of A and B represent the type of error test which will be considered; namely, $A = 1$, $B = 0$ are for the absolute error test; $A = 1$, $B = 1$ represent the mixed error test; and $A = 0$, $B = 1$ correspond to the relative error test. Here we will use the mixed error test and the maximum error is calculated by

$$\text{MAXE} = \max_{1 \le i \le 4} (E_i). \tag{16}$$

4. Numerical Results

To illustrate the technique proposed in the preceding sections, two test problems are solved and the results obtained compared with the method proposed in [8, 9, 25] and the ODE solver in MATLAB ode45. The work done by [25] involved the solving of the first-order ODEs using 4-point block method using variable step size. This means that (1) needs to be reduced into a system of first-order IVPs whereby the methods proposed by [8, 9] are for solving (1) directly with the implementation of constant step size. The notations used in Tables 1 and 2 are listed below:

TOL: error tolerance limit;

MTD: method used;

TS: total steps taken;

MAXE: maximum error of the computed solution;

AVERR: average error of the computed solution;

FC: total function calls;

ode45: Runge-Kutta-Dormand-Prince ODE solver;

4PHODE: implementation of the 4-point block method in this research;

4P1FI: numerical results in [25];

KAYODE (a): numerical results in [8];

KAYODE (b): numerical results in [9].

Table 2: Numerical results for solving Problem 2.

TOL	MTD	TS	MAXE	AVERR	FC
10^{-2}	ode45	11	5.027 (−3)	3.806 (−3)	67
	4P1FI	58	2.528 (−7)	3.002 (−8)	490
	4PHODE	28	5.537 (−6)	2.245 (−7)	250
10^{-4}	ode45	13	1.967 (−4)	1.253 (−4)	79
	4P1FI	68	9.819 (−9)	1.117 (−9)	571
	4PHODE	36	4.193 (−7)	1.432 (−8)	315
10^{-6}	ode45	25	4.592 (−7)	1.870 (−7)	151
	4P1FI	86	1.219 (−10)	9.992 (−12)	715
	4PHODE	46	1.003 (−7)	4.290 (−9)	395
10^{-8}	ode45	58	1.861 (−9)	1.006 (−9)	349
	4P1FI	135	1.363 (−12)	1.597 (−13)	1107
	4PHODE	56	2.413 (−8)	1.012 (−9)	475
10^{-10}	ode45	142	4.555 (−11)	2.026 (−11)	853
	4P1FI	807	5.630 (−14)	1.250 (−14)	6483
	4PHODE	71	1.949 (−10)	7.564 (−12)	595
KAYODE (a)		Not stated	2.225 (−7)	Not stated	Not stated
KAYODE (b)		10	3.156 (−4)	Not stated	Not stated

Problem 1. Consider the following:

$$y^{(v)} = 2y'y'' - yy^{(iv)} - y'y''' - 8x + \left(x^2 - 2x - 3\right)e^x, \quad x \in [0,2],\ y(0) = 1,\ y'(0) = 1,\ y''(0) = 3,\ y'''(0) = 1,\ y^{(iv)}(0) = 1. \tag{17}$$

Solution is as follows: $y(x) = e^x + x^2$.

Source is [8].

Problem 2. Consider the following:

$$y^{(v)} = 6\left(2\left(y'\right)^3 + 6yy'y'' + y^2y'''\right), \quad x \in [1,3],\ y(1) = 1,\ y'(1) = -1,\ y''(1) = 2,\ y'''(1) = -6,\ y^{(iv)}(1) = 24. \tag{18}$$

Solution is as follows: $y(x) = 1/x$.
Source is [8].

5. Discussion

In this section the performances of 4PHODE, ode45, 4P1FI, KAYODE (a), and KAYODE (b) are discussed in terms of three parameters, namely, total steps taken, accuracy, and total function evaluations. It is apparent from these tables, mostly at tolerances 10^{-2}, 10^{-4}, and 10^{-6}, that 4PHODE gives better accuracy compared to ode45, whereas at tolerances 10^{-8} and 10^{-10}, ode45 is one decimal more accurate compared to 4PHODE for both problems. As both tables show, the total steps taken by the 4PHODE reduce by nearly half to ode45 at tolerance 10^{-10}. Although, at other tolerance, ode45 requires lesser steps to compute the solution, this result may be explained by the fact that the initial step size generated by 4PHODE is extremely small in order for the method control of the accuracy. From the data in Tables 1 and 2, it is apparent that the number of function calls is likely to be related to the number of steps taken.

In the comparison of block-by-block method, the numerical results obtained in Tables 1 and 2 demonstrate that the proposed method 4PHODE always requires fewer steps in converging to the exact solution compared to 4P1FI. This gain becomes more obvious as the tolerance decreases. However, the 4P1FI achieves better accuracy than 4PHODE. Even so, the maximum error for 4PHODE is still within the specified tolerance. Another issue that had to be addressed was the total number of function evaluations used by each method. The 4PHODE solved all the problems at much lower cost than 4P1FI and this superiority is most apparent at finer tolerance. This result is in agreement with [10] which states the drawback of using the reduction approach in the implementation of simultaneous computations.

Comparing the results with the method proposed by KAYODE (a) and KAYODE (b), for Problem 1, 4PHODE outperformed both KAYODE (a) and KAYODE (b) in terms of accuracy and total steps taken. While, for Problem 2, KAYODE (b) has lesser total steps taken, however, 4PHODE still has superiority in terms of accuracy.

6. Conclusion

The overall performance revealed that the 4-point block method is best to be implemented in a direct integration approach as it required much less storage than the reduction method while still maintaining an acceptable accuracy. Besides that, the results of this study also indicate that the developed method has better accuracy compared to the existing methods. Hence it can be said that 4PHODE is one of the alternative methods that can be used for solving fifth-order ODEs.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

Grateful acknowledgement is made to the Ministry of Higher Education, Malaysia, for financial support with Grant no. RAGS/2013/FKEKK/SG04/01/B00037.

References

[1] M. T. Darvishi, S. Kheybari, and F. Khani, "A numerical solution of the Korteweg-de Vries equation by pseudospectral method using Darvishi's preconditionings," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 98–105, 2006.

[2] L. Jin, "Application of variational iteration method to the fifth-order KdV equation," *International Journal of Contemporary Mathematical Sciences*, vol. 3, no. 5, pp. 213–221, 2008.

[3] L. Kaur, "Generalized (G′/G)—expansion method for generalized fifth order KdV equation with time-dependent coefficients," *Mathematical Sciences Letters*, vol. 3, no. 3, pp. 255–261, 2014.

[4] M. Suleiman, Z. B. Ibrahim, and A. F. N. Bin Rasedee, "Solution of higher-order ODEs using backward difference method," *Mathematical Problems in Engineering*, vol. 2011, Article ID 810324, 18 pages, 2011.

[5] U. Goktas and W. Hereman, "Symbolic computation of conserved densities for systems of nonlinear evolution equations," *Journal of Symbolic Computation*, vol. 24, no. 5, pp. 591–621, 1997.

[6] N. Khanal, R. Sharma, J. Wu, and J.-M. Yuan, "A dual-Petrov-Galerkin method for extended fifth-order Korteweg-de Vries type equations," *Discrete and Continuous Dynamical Systems*, pp. 442–450, 2009.

[7] J. Li and Z. Qiao, "Explicit soliton solutions of the Kaup-Kupershmidt equation through the dynamical system approach," *Journal of Applied Analysis and Computation*, vol. 1, no. 2, pp. 243–250, 2011.

[8] S. J. Kayode and D. O. Awoyemi, "A multiderivative Collocation method for 5th order ordinary differential equations," *Journal of Mathematics and Statistics*, vol. 6, no. 1, pp. 60–63, 2010.

[9] S. J. Kayode, "An order seven continuous explicit method for direct solution of general fifth order ordinary differential equations," *International Journal of Differential Equations and Applications*, vol. 13, no. 2, pp. 71–80, 2014.

[10] J. B. Rosser, "A Runge-Kutta for all seasons," *SIAM Review*, vol. 9, no. 3, pp. 417–452, 1967.

[11] M. B. Suleiman, "Solving nonstiff higher order ODEs directly by the direct integration method," *Applied Mathematics and Computation*, vol. 33, no. 3, pp. 197–219, 1989.

[12] D. O. Awoyemi, "A new sixth-order algorithm for general second order ordinary differential equations," *International Journal of Computer Mathematics*, vol. 77, no. 1, pp. 117–124, 2001.

[13] S. J. Kayode, "An efficient zero-stable numerical method for fourth-order differential equations," *International Journal of Mathematics and Mathematical Sciences*, vol. 2008, Article ID 364021, 10 pages, 2008.

[14] B. T. Olabode and Y. Yusuph, "A new block method for special third order ordinary differential equations," *Journal of Mathematics and Statistics*, vol. 5, no. 3, pp. 167–170, 2009.

[15] S. N. Jator and J. Li, "A self-starting linear multistep method for a direct solution of the general second-order initial value problem," *International Journal of Computer Mathematics*, vol. 86, no. 5, pp. 827–836, 2009.

[16] N. Waeleh, Z. A. Majid, F. Ismail, and M. Suleiman, "Numerical solution of higher order ordinary differential equations by direct block code," *Journal of Mathematics and Statistics*, vol. 8, no. 1, pp. 77–81, 2011.

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

[18] J. R. Cash, "Block Runge-Kutta methods for the numerical integration of initial value problems in ordinary differential equations. Part I. The nonstiff case," *Mathematics of Computation*, vol. 40, no. 161, pp. 175–191, 1983.

[19] S. N. Jator, "Solving second order initial value problems by a hybrid multistep method without predictors," *Applied Mathematics and Computation*, vol. 217, no. 8, pp. 4036–4046, 2010.

[20] N. Waeleh, Z. A. Majid, and F. Ismail, "A new algorithm for solving higher order IVPs of ODEs," *Applied Mathematical Sciences*, vol. 5, no. 53–56, pp. 2795–2805, 2011.

[21] J. Vigo-Aguiar and H. Ramos, "Variable stepsize implementation of multistep methods for y″=f(x, y, y′)," *Journal of Computational and Applied Mathematics*, vol. 192, no. 1, pp. 114–131, 2006.

[22] Z. A. Majid, N. A. Azmi, M. Suleiman, and Z. B. Ibrahaim, "Solving directly general third order ordinary differential equations using two-point four step block method," *Sains Malaysiana*, vol. 41, no. 5, pp. 623–632, 2012.

[23] J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons, London, UK, 1973.

[24] S. Ola Fatunla, "Block methods for second order ODEs," *International Journal of Computer Mathematics*, vol. 41, no. 1-2, pp. 55–63, 1991.

[25] 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.