


# A Class of Hybrid Methods for Direct Integration of Fourth-Order Ordinary Differential Equations

Y. D. Jikantoro<sup>1,3</sup>  · F. Ismail<sup>1,2</sup> ·  
N. Senu<sup>1,2</sup> · Z. B. Ibrahim<sup>1,2</sup>

Received: 21 February 2017 / Revised: 27 May 2017 / Published online: 27 June 2017  
© Malaysian Mathematical Sciences Society and Penerbit Universiti Sains Malaysia 2017

**Abstract** A class of hybrid methods for solving fourth-order ordinary differential equations (HMFd) is proposed and investigated. Using the theory of B-series, we study the order of convergence of the HMFd methods. The main result is a set of order conditions, analogous to those for two-step hybrid method, which offers a better alternative to the usual ad hoc Taylor expansions. Based on the algebraic order conditions, a one-stage and two-stage explicit HMFd methods are constructed. Results from numerical experiment suggest the superiority of the new methods in terms of accuracy and computational efficiency over hybrid methods for special second ODEs, Runge–Kutta methods recently proposed for solving special fourth-order ODEs directly and some linear multistep methods proposed for the same purpose in the literature.

**Keywords** Hybrid method · Four-step method · B4-series · Order conditions · Fourth-order ordinary differential equations

**Mathematics Subject Classification** 65L05

---

Communicated by Ahmad Izani Md. Ismail.

---

✉ Y. D. Jikantoro  
jdauday@yahoo.ca

<sup>1</sup> Department of Mathematics, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia

<sup>2</sup> Institute for Mathematical Research, University Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia

<sup>3</sup> Department of Mathematics, Ibrahim Badamasi Babangida University, P.M.B 11, Lapai, Nigeria

## 1 Introduction

Fourth-order ordinary differential equations (ODEs) occur in a number of areas of applied sciences including quantum mechanics, fluid mechanics, elasticity, physics and engineering. It is a common knowledge that many classes of fourth-order ODEs defy analytical solution. That is, only a small class of the equations can be solved by analytical techniques. Hence, the need for numerical methods becomes imperative. Owing to this fact, a number of authors have proposed and investigated some numerical methods for solving the fourth-order ODEs, among which are linear multi-step and Runge–Kutta-related methods [1–4], where the fourth-order ODEs need to be transformed into an equivalent system of first-order ODEs for the numerical integration to proceed. The drawback of computational inefficiency of these methods prompted direct integrators for fourth-order ODEs such as cubic spline collocation tau method [5], logarithmic collocation method [6], cubic spline method for fourth-order obstacle problems [7], fourth-order initial and boundary value problems integrators [8]. Other such methods can be found in [9–11] and references therein.

In this paper, we seek to construct and investigate a class of efficient numerical integrators for a class of special fourth-order ODEs, which takes the form

$$y^{iv}(x) = f(x, y(x)), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad y''(x_0) = y''_0, \quad y'''(x_0) = y'''_0, \quad (1)$$

where  $y \in R^r$ ,  $f : R \times R^r \rightarrow R^r$  is a continuous vector value function. And the fact that  $f$  does not depend on  $y'$ ,  $y''$ ,  $y'''$  explicitly is the specialty associated with (1). Typical example of (1) is the ill-posed problem of a beam on elastic foundation, which finds an important engineering application. This problem has been studied in [11, 12].

In line with the direct numerical integrators of Runge–Kutta type for a class of special third-order ODEs [13], a direct numerical integrator of Runge–Kutta type was proposed for (1) recently [11], whose internal and update stages depend on the first, second and the third derivatives of the solution at each step. We propose a class of new integrators whose stages do not depend on the derivatives of the solution.

Section 2 is devoted to formulation of the proposed method. In Sect. 3, we present the theory of B-series and the associated rooted trees through which order conditions of the proposed method are derived. Local truncation error and order of convergence of the method are presented in Sect. 4. We present algebraic order conditions of the method in Sect. 5. As examples, explicit one-stage and two-stage HMFD methods are presented in Sect. 6. Stability and convergence analysis is presented in Sect. 7. Numerical experiment is presented in Sect. 8. And conclusion is given in Sect. 9.

## 2 Formulation of HMFD Method

To formulate the proposed HMFD method, we consider transforming (1) into a system of first-order initial value problem (IVP) as follows:

$$\begin{pmatrix} y(x) \\ u(x) \\ v(x) \\ w(x) \end{pmatrix}' = \begin{pmatrix} u(x) \\ v(x) \\ w(x) \\ f(x, y(x)) \end{pmatrix}, \quad \begin{pmatrix} y(x_0) \\ u(x_0) \\ v(x_0) \\ w(x_0) \end{pmatrix} = \begin{pmatrix} y_0 \\ y_0' \\ y_0'' \\ y_0''' \end{pmatrix}. \tag{2}$$

An s-stage Runge–Kutta method is defined by

$$\begin{aligned} Y_i &= y_n + h \sum_{j=1}^s \bar{a}_{i,j} f(x_n + c_j h, Y_j), \quad i = 1, \dots, s, \\ y_{n+1} &= y_n + h \sum_{j=1}^s \bar{b}_j f(x_n + c_j h, Y_j). \end{aligned} \tag{3}$$

Applying (3) to (2), the following equations are obtained

$$\begin{aligned} Y_i &= y_n + h \sum_{j=1}^s \bar{a}_{i,j} \bar{Y}_j, \\ \bar{Y}_i &= y_n' + h \sum_{j=1}^s \bar{a}_{i,j} \bar{\bar{Y}}_j, \\ \bar{\bar{Y}}_i &= y_n'' + h \sum_{j=1}^s \bar{a}_{i,j} \bar{\bar{\bar{Y}}}_j, \\ \bar{\bar{\bar{Y}}}_i &= y_n''' + h \sum_{j=1}^s \bar{a}_{i,j} f(x_n + c_j h, Y_j), \\ y_{n+1} &= y_n + h \sum_{i=1}^s \bar{b}_i \bar{Y}_i, \\ y_{n+1}' &= y_n' + h \sum_{i=1}^s \bar{b}_i \bar{\bar{Y}}_i, \\ y_{n+1}'' &= y_n'' + h \sum_{i=1}^s \bar{b}_i \bar{\bar{\bar{Y}}}_i, \\ y_{n+1}''' &= y_n''' + h \sum_{i=1}^s \bar{b}_i f(x_n + c_i h, Y_i). \end{aligned}$$

Eliminating  $\bar{Y}_i, \bar{\bar{Y}}_i$  and  $\bar{\bar{\bar{Y}}}_i$  in the equations above, we have

$$\begin{aligned}
 Y_i &= y_n + h \sum_{j=1}^s \bar{a}_{i,j} y'_n + h^2 \sum_{j,k=1}^s \bar{a}_{i,j} \bar{a}_{j,k} y''_n + h^3 \sum_{j,k,l=1}^s \bar{a}_{i,j} \bar{a}_{j,k} \bar{a}_{k,l} y'''_n \\
 &\quad + h^4 \sum_{j,k,l,m=1}^s \bar{a}_{i,j} \bar{a}_{j,k} \bar{a}_{k,l} \bar{a}_{l,m} f(x_n + c_m h, Y_m), \\
 y_{n+1} &= y_n + h \sum_{i=1}^s \bar{b}_i y'_n + h^2 \sum_{i,j=1}^s \bar{b}_i \bar{a}_{i,j} y''_n + h^3 \sum_{i,j,k=1}^s \bar{b}_i \bar{a}_{i,j} \bar{a}_{j,k} y'''_n \\
 &\quad + h^4 \sum_{i,j,k,l=1}^s \bar{b}_i \bar{a}_{i,j} \bar{a}_{j,k} \bar{a}_{k,l} f(x_n + c_l h, Y_l), \\
 y'_{n+1} &= y'_n + h \sum_{i=1}^s \bar{b}_i y''_i + h^2 \sum_{i,j=1}^s \bar{b}_i \bar{a}_{i,j} y'''_n + h^3 \sum_{i,j,k=1}^s \bar{b}_i \bar{a}_{i,j} \bar{a}_{j,k} f(x_n + c_k h, Y_k), \\
 y''_{n+1} &= y''_n + h \sum_{i=1}^s \bar{b}_i y'''_i + h^2 \sum_{i,j=1}^s \bar{b}_i \bar{a}_{i,j} f(x_n + c_j h, Y_j), \\
 y'''_{n+1} &= y'''_n + h \sum_{i=1}^s \bar{b}_i f(x_n + c_i h, Y_i).
 \end{aligned}$$

Denote

$$\begin{aligned}
 \sum_{j=1}^s \bar{a}_{i,j} &= c_i, \quad \sum_{j,k=1}^s \bar{a}_{i,j} \bar{a}_{j,k} = \frac{c_i(c_i + 1)}{2}, \quad \sum_{j,k,l=1}^s \bar{a}_{i,j} \bar{a}_{j,k} \bar{a}_{k,l} = \frac{c_i(c_i + 1)(c_i + 2)}{6}, \\
 \sum_{i=1}^s b_i &= \sum_{i,j=1}^s \bar{b}_i \bar{a}_{i,j} = \sum_{i,j,k=1}^s \bar{b}_i \bar{a}_{i,j} \bar{a}_{j,k} = 1, \quad i = 1, \dots, s.
 \end{aligned}$$

And

$$\sum_{k,l,m=1}^s \bar{a}_{i,k} \bar{a}_{k,l} \bar{a}_{l,m} \bar{a}_{m,j} = a_{i,j}, \quad \sum_{j,k,l=1}^s \bar{b}_j \bar{a}_{j,k} \bar{a}_{k,l} \bar{a}_{l,i} = b_i.$$

The above suppositions together with difference formula on the above equations lead to the proposed method in vector form as follows:

$$\begin{aligned}
 y_{n+1} &= 2(2y_n - 3y_{n-1} + 2y_{n-2}) - y_{n-3} + h^4 (\mathbf{b}^T \otimes \mathbf{I}) f(x_n + \mathbf{c}h, \mathbf{Y}), \\
 \mathbf{Y} &= \frac{1}{6} \{ \mathbf{C}_1 \otimes y_n - 3\mathbf{C}_2 \otimes y_{n-1} + 3\mathbf{C}_3 \otimes y_{n-2} - \mathbf{C}_4 \otimes y_{n-3} \} + h^4 (\mathbf{A} \otimes \mathbf{I}) f(x_n + \mathbf{c}h, \mathbf{Y}), \tag{4}
 \end{aligned}$$

where  $\mathbf{C}_1 = 6\mathbf{e} + 11\mathbf{c} + 6\mathbf{c}^2 + \mathbf{c}^3$ ,  $\mathbf{C}_2 = 6\mathbf{c} + 5\mathbf{c}^2 + \mathbf{c}^3$ ,  $\mathbf{C}_3 = 3\mathbf{c} + 4\mathbf{c}^2 + \mathbf{c}^3$ ,  $\mathbf{C}_4 = 2\mathbf{c} + 3\mathbf{c}^2 + \mathbf{c}^3$ ,  $\mathbf{b} = [b_1, \dots, b_m]^T$ ,  $\mathbf{c} = [c_1, \dots, c_m]^T$ ,  $\mathbf{e} = [1, \dots, 1]^T$ ,  $\mathbf{A} = [a_{i,j}]$ ,

**Table 1** General coefficients of HMFd methods

-3	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$\cdots$	$a_{1,m}$
-2	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$\cdots$	$a_{2,m}$
-1	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$\cdots$	$a_{3,m}$
0	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$\cdots$	$a_{4,m}$
$c_5$	$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$\cdots$	$a_{5,m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$c_m$	$a_{m,1}$	$a_{m,2}$	$a_{m,3}$	$\cdots$	$a_{m,m}$
	$b_1$	$b_2$	$b_3$	$\cdots$	$b_m$

$\mathbf{Y} = [Y_1, \dots, Y_m]^T$  and  $\mathbf{I}$  is identity matrix of  $m \times m$  dimension. The coefficients of the methods are summarized in Table 1.

### 3 B4-Series and Associated Rooted Trees

As we have in the case of RKT, RKN and RK methods for third-, second- and first-order ODEs, when working on the derivation of order conditions for HMFd methods for fourth-order ODEs, we need to consider the autonomous case of problem (1)

$$y^{iv}(x) = f(y(x)), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad y''(x_0) = y''_0, \quad y'''(x_0) = y'''_0. \quad (5)$$

In fact, the equation in (1) can be extended by one dimension  $v = x$  in order to rewrite the initial value problem (IVP) (1) equivalently as the following autonomous problem

$$\begin{aligned} v^{iv} &= 0, \quad v(x_0) = x_0, \quad v'(x_0) = 1, \quad v''(x_0) = 0, \quad v'''(x_0) = 0, \\ y^{iv} &= f(v, y), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad y''(x_0) = y''_0, \quad y'''(x_0) = y'''_0. \end{aligned} \quad (6)$$

Applying the scheme (4) to (6) gives

$$\begin{aligned} V_i &= \frac{1}{6} [C_1 v_n - 3C_2 v_{n-1} + 3C_3 v_{n-2} - C_4 v_{n-3}], \\ &= v_n + hc_i v'_n + \frac{h^2}{2} c_i (c_i + 1) v''_n + \frac{h^3}{6} c_i (c_i + 1) (c_i + 2) v'''_n, \\ Y_i &= \frac{1}{6} [C_1 y_n - 3C_2 y_{n-1} + 3C_3 y_{n-2} - C_4 y_{n-3}] + h^4 \sum_{j=1}^s a_{i,j} f(V_j, Y_j), \\ v_{n+1} &= 4z_n - 6z_{n-1} + 4v_{n-2} - v_{n-3}, = v_n + hv'_n + h^2 v''_n + h^3 v'''_n, \\ y_{n+1} &= 4y_n - 6y_{n-1} + 4y_{n-2} - y_{n-3} + h^4 \sum_{i=1}^s b_i f(V_i, Y_i). \end{aligned} \quad (7)$$

Substituting first part of (6) into (7) gives

$$\begin{aligned}
 V_i &= x_n + c_i h, \\
 v_{n+1} &= x_n + h, \\
 Y_i &= \frac{1}{6} [C_1 y_n - 3C_2 y_{n-1} + 3C_3 y_{n-2} - C_4 y_{n-3}] + h^4 \sum_{j=1}^s a_{i,j} f(x_n + c_j h, Y_j), \\
 y_{n+1} &= 4y_n - 6y_{n-1} + 4y_{n-2} - y_{n-3} + h^4 \sum_{i=1}^s b_i f(x_n + c_i h, Y_i). \tag{8}
 \end{aligned}$$

Observe the last two equations of (8), they look exactly like (4). This implies that the HMGD method (4) applied to (5) produces the same numerical solution like the one obtained when (4) is applied to (1). Thus, it's enough to discuss the numerical solution of (5). Hence, the method (4) takes the form

$$\begin{aligned}
 Y_i &= \frac{1}{6} [C_1 y_n - 3C_2 y_{n-1} + 3C_3 y_{n-2} - C_4 y_{n-3}] + h^4 \sum_{j=1}^s a_{i,j} f(Y_j), \\
 y_{n+1} &= 4y_n - 6y_{n-1} + 4y_{n-2} - y_{n-3} + h^4 \sum_{i=1}^s b_i f(Y_i). \tag{9}
 \end{aligned}$$

Continuous differentiation of the exact solution  $y(x)$  with respect to independent variable  $x$  gives the following:




$$\begin{aligned}
 y' &= y', \quad y'' = y'', \quad y''' = y''', \quad y^{iv} = f(y), \quad y^v = f'(y)y', \\
 y^{vi} &= f''(y)(y', y') + f(y)y'', \quad y^{vii} = f'''(y)(y', y', y') \\
 &\quad + 3f''(y)(y', y'') + f'(y)y''', \\
 y^{viii} &= f^{iv}(y)(y', y', y', y') + 6f'''(y)(y', y', y'') + 3f''(y)(y'', y'') \\
 &\quad + 4f'(y)(y', y''') + f'(y)f(y).
 \end{aligned}$$


### 3.1 Construction of Rooted Trees


Here, detail description of how the relevant trees in this paper are constructed is given. It's easy to associate each of the expressions of the derivatives of  $y$  above and those of higher orders with rooted trees. The relevant trees here consist of four type of vertices, namely small dot ( $\bullet$ ), big dot ( $\bullet$ ), small circle ( $\circ$ ) and big circle ( $\circ$ ) representing  $y'$ ,  $y''$ ,  $y'''$  and  $f$ , respectively. The 'branch' of a tree is a line joining its vertices according to naturally defined rules. The line simply indicates differentiation with respect to components of  $y$ ,  $y'$ ,  $y''$  or  $y'''$ . When a 'branch' grows to a small dot vertex, then it represents differentiation with respect to the component of  $y$ . It's with respect to the component of  $y'$  if the 'branch' grows to a big dot vertex. The differentiation is with respect to component of  $y''$  if the 'branch' grows to a small circle vertex. And




it's with respect to the component of  $y'''$  if the destination vertex is a big circle. This relationship could also be interpreted as parent–son relationship. For instance, small dot vertex gives birth to big dot vertex as its only son, because  $y'$  has only one nonzero derivative with respect to itself and has not with respect to  $y$ ,  $y''$  and  $y'''$ . The only son of big dot vertex on the other hand is small circle vertex, because  $y''$  has only one nonzero derivative with respect to itself and has non with respect to  $y$ ,  $y'$  and  $y'''$ . Similarly, small circle vertex's only son is big circle vertex. In the case of big circle vertex, it has many sons which are all small dot vertices. This is because  $f$  depends on  $y$  only and can have multiple derivatives with respect to component of  $y$  only, but not with  $y'$ ,  $y''$  and  $y'''$ . The relevant trees can recursively be defined by simply modifying the notation used for trees associated with second-order differential equations in [14].

**Definition 1** The set of trees  $T_4$  is defined recursively as follows:

- (i) the graph  $\bullet$  (root) is a member of  $T_4$ ; the graphs ,  and  are in  $T_4$ . And the null tree  $\theta$  is also a member of  $T_4$ ;
- (ii) if  $t_1, \dots, t_m \in T_4$ , then the graph  $t = [t_1, \dots, t_m]_4$  obtained by connecting the roots

of  $t_1, \dots, t_m$  to the big circle vertex of the graph  is also in  $T_4$ , where the small dot vertex at the bottom represents the root of the trees. The subscript 4 is a reminder that the tree is associated with fourth-order differential equations and that the

'stem'  upon which 'branches' grow consists of a string of four vertices.

Let  $\tau_1 = \bullet$ ,  $\tau_2 = \bullet$  ,  $\tau_3 = \bullet$   and  $\tau_4 = \bullet$   denote first-order, second-order, third-order and fourth-order trees, respectively.

**Definition 2** The order  $\rho : T_4 \rightarrow N$  of a tree  $t$  is defined recursively as follows:

- (i)  $\rho(\tau_1) = 1, \rho(\tau_2) = 2, \rho(\tau_3) = 3, \rho(\tau_4) = 4$ ;
- (ii) for  $t = [t_1, \dots, t_m]_4 \in T_4, \rho(t) = 4 + \sum_{i=1}^m \rho(t_i)$ . In nutshell, for each  $t \in T_4$ , the order  $\rho(t)$  of a tree  $t$  is the number of vertices of the tree. The set of all  $T_4$  of order  $q$  is denoted by  $T_{4q}$ .
- (iii)  $\alpha(\theta) = \alpha(\tau_1) = \alpha(\tau_2) = \alpha(\tau_3) = \alpha(\tau_4) = 1$ ;
- (iv) if  $t = [t_1^{\mu_1}, \dots, t_m^{\mu_m}]_4 \in T_4$ , with  $t_i$  distinct for different  $i$  and different from  $\theta$ , then

$$\alpha(t) = (\rho(t) - 4)! \prod_{i=1}^m \frac{1}{\mu_i!} \left( \frac{\alpha(t_i)}{\rho(t_i)!} \right)^{\mu_i} .$$

The first four vertices of a tree are its small dot vertex, big dot vertex, small circle vertex and big circle vertex. Any distinct labeling involves only the remaining  $\rho(t) - 4$

vertices. The first four vertices form the root and the 'stem' upon which the 'branches' (other vertices) grow.

It is important to associate each elementary differential  $F(t)$  with a corresponding  $t \in T_4$ .

**Definition 3** The function  $F$  on  $T_4$  is defined recursively by

- (i)  $F(\theta)(y, y', y'', y''') = y, F(\tau_1)(y, y', y'', y''') = y', F(\tau_2)(y, y', y'', y''') = y'', F(\tau_3)(y, y', y'', y''') = y'''$  and  $F(\tau_4)(y, y', y'', y''') = f(y)$ ;
- (ii) if  $t = [t_1, \dots, t_m]_4 \in T_4$ , then

$$F(t)(y, y', y'', y''') = f^{(m)}(y) (F(t_1)(y, y', y'', y'''), \dots, F(t_m)(y, y', y'', y''')).$$

**Definition 4** Let  $\beta : T_4 \rightarrow R$  be a mapping, with  $\beta(\theta) = 1$ . The B4-series with coefficient function  $\beta$  is a formal series of the form

$$\begin{aligned} B(\beta, y) &= y + h\alpha(\tau_1)\beta(\tau_1)y' + \frac{h^2}{2}\alpha(\tau_2)\beta(\tau_2)y'' + \frac{h^3}{6}\alpha(\tau_3)\beta(\tau_3)y''' + \dots \\ &= \sum_{t \in T_4} \frac{h^{\rho(t)}}{\rho(t)!} \alpha(t)\beta(t)F(t)(y, y', y'', y'''). \end{aligned} \tag{10}$$

In the derivation of order conditions, the following lemma, which states that  $h^4 f(\cdot)$  applied to a B4-series generates a B4-series [3, 13, 14], is very important.

**Lemma** Let  $B(\beta, y)$  be a B4-series with coefficient function  $\beta$ . Then,  $h^4 f(B(\beta, y))$  is also a B4-series, i.e.,

$$h^4 f(B(\beta, y)) = B(\beta^{iv}, y),$$

with

$$\beta^{iv}(\theta) = \beta^{iv}(\tau_1) = \beta^{iv}(\tau_2) = 0, \beta^{iv}(\tau_3) = \beta^{iv}(\tau_4) = 24$$

and for all other tree  $t = [t_1, \dots, t_m]_4 \in T_4$ ,

$$\beta^{iv}(t) = \rho(t)(\rho(t) - 1)(\rho(t) - 2)(\rho(t) - 3) \prod_{i=1}^m \beta(t_i).$$

*Proof* Since  $B(\beta, y) = y + O(h)$ , it is clear that  $h^4 f(B(\beta, y))$  can be expanded as a Taylor series about  $y$ . Furthermore,  $\beta^{iv}(\theta) = \beta^{iv}(\tau_1) = \beta^{iv}(\tau_2) = \beta^{iv}(\tau_3) = 0$ , because the expansion starts with  $h^4 f(y)$ , which also shows that  $\beta^{iv}(\tau_4) = 24$ . Proceeding as in the proof of Lemma in [11, 13, 14], we have



$$\begin{aligned}
 h^4 f(B(\beta, y)) &= h^4 \sum_{m \geq 0} \frac{1}{m!} f^{(m)}(y) (B(\beta, y) - y)^m \\
 &= h^4 \sum_{m \geq 0} \frac{1}{m!} \sum_{t_1 \in T_4 \setminus \theta} \dots \sum_{t_m \in T_4 \setminus \theta} \frac{h^{\rho(t_1) + \dots + \rho(t_m)}}{\rho(t_1)! \dots \rho(t_m)!} \alpha(t_1) \dots \alpha(t_m) \\
 &\quad \beta(t_1) \dots \beta(t_m) f^{(m)}(y) (F(t_1)(y, y' y'', y''') \dots F(t_m)(y, y' y'', y''')) \\
 &= \sum_{m \geq 0} \sum_{t_1 \in T_4 \setminus \theta} \dots \sum_{t_m \in T_4 \setminus \theta} \frac{h^{\rho(t)}}{(\rho(t) - 4)!} \alpha(t) \frac{\mu_1! \dots \mu_m!}{m!} \\
 &\quad \beta(t_1) \dots \beta(t_m) F(t)(y, y' y'', y''') \\
 &= \sum_{t \in T_4 \setminus \theta} \frac{h^{\rho(t)}}{(\rho(t) - 4)!} \alpha(t) \beta(t_1) \dots \beta(t_m) F(t)(y, y' y'', y'''),
 \end{aligned}$$

because  $\frac{m!}{\mu_1! \dots \mu_m!}$  is ways of ordering the labels  $t_1, \dots, t_m$  in  $[t_1, \dots, t_m]_4$ . Finally, with  $\beta^{iv}$  as defined in the statement of Lemma,

$$h^4 f(B(\beta, y)) = \sum_{t \in T_4 \setminus \theta} \frac{h^{\rho(t)}}{\rho(t)!} \alpha(t) \beta^{iv}(t) F(t)(y, y' y'', y''') = B(\beta^{iv}(t), y).$$

□

### 4 Local Truncation Error of HMFD and its Convergence Order

Like the hybrid methods presented in [14], to derive the order conditions of HMFD methods (four-step methods), we consider them as single step methods of the form

$$H_n = H_{n-1} + h\Phi(H_{n-1}, h), \tag{11}$$

where  $H_n$  is a well-defined numerical solution whose initial point  $H_0$  is generated by some starting procedure, see [14]. The first part of equation (4) can be written as a set of four equations by letting

$$F_n := \left( \frac{y_{n+1} - y_n}{h} \right),$$

so that

$$y_n = y_{n-1} + hF_{n-1}, \tag{12}$$

which implies that

$$F_n = 3F_{n-1} - 3hF_{n-2} + F_{n-3} + h^3 \left( \mathbf{b}^T \otimes \mathbf{I} \right) f(\mathbf{Y}). \tag{13}$$

Now, let

$$G_n := \left( \frac{F_{n+1} - F_n}{h} \right),$$

which implies that

$$F_n = F_{n-1} + hG_{n-1}. \tag{14}$$

Then, (13) becomes

$$G_n = 2G_{n-1} - G_{n-2} + h^2 \left( \mathbf{b}^T \otimes \mathbf{I} \right) f(\mathbf{Y}). \tag{15}$$

Now, by letting

$$U_n := \left( \frac{G_{n+1} - G_n}{h} \right),$$

we have

$$G_n = G_{n-1} + hU_{n-1}. \tag{16}$$

Hence, (15) gives,

$$U_n = U_{n-1} + h \left( \mathbf{b}^T \otimes \mathbf{I} \right) f(\mathbf{Y}). \tag{17}$$

The system of Eqs. (12), (14), (16) and (17) can be written as (11) with

$$H_n = \begin{pmatrix} y_n \\ F_n \\ G_n \\ U_n \end{pmatrix} \quad \text{and} \quad \Phi(H_{n-1}, h) = \begin{pmatrix} F_{n-1} \\ G_{n-1} \\ U_{n-1} \\ (\mathbf{b}^T \otimes \mathbf{I}) f(\mathbf{Y}) \end{pmatrix}.$$

The vector  $H_n$  is an approximation for  $w_n = w(x_n, h)$ , where  $w$  is the exact-value function defined by

$$w(x, h) = \begin{pmatrix} y(x) \\ \frac{y(x+h) - y(x)}{h} \\ \frac{F(x+h) - F(x)}{h} \\ \frac{G(x+h) - G(x)}{h} \end{pmatrix}. \tag{18}$$

The local truncation error of the HMF method at point  $x_n$  is defined by

$$d_n = w_n - w_{n-1} - h\Phi(w_{n-1}, h), \tag{19}$$

where

$$\Phi(w_{n-1}, h) = \begin{pmatrix} \frac{y(x_n) - y(x_{n-1})}{h} \\ \frac{F(x_n) - F(x_{n-1})}{h} \\ \frac{G(x_n) - G(x_{n-1})}{h} \\ (\mathbf{b}^T \otimes \mathbf{I}) f(\mathbf{Y}) \end{pmatrix}. \tag{20}$$

Suppose that each component of  $\mathbf{Y}$  in the second part of (4) can be expanded as a B4-series  $Y_i(x_n) = B(\psi_i, y(x_n))$ , we get

$$\begin{aligned} B(\psi_i, y(x_n)) &= \frac{1}{6} (c_i^3 + 6c_i^2 + 11c_i + 6) y(x_n) \\ &\quad - \frac{1}{2} (c_i^3 + 6c_i^2 + 5c_i) B((-1)^{\rho(t)}, y(x_n)) \\ &\quad + \frac{1}{2} (c_i^3 + 4c_i^2 + 3c_i) B((-2)^{\rho(t)}, y(x_n)) \\ &\quad - \frac{1}{6} (c_i^3 + 3c_i^2 + 2c_i) B((-3)^{\rho(t)}, y(x_n)) + h^4 \sum_j^s a_{i,j} f(Y_j). \end{aligned} \tag{21}$$

Applying Lemma in Sect. 3 on (21), we get

$$\begin{aligned} B(\psi_i, y(x_n)) &= \frac{1}{6} (c_i^3 + 6c_i^2 + 11c_i + 6) y(x_n) \\ &\quad - \frac{1}{2} (c_i^3 + 6c_i^2 + 5c_i) B((-1)^{\rho(t)}, y(x_n)) + \frac{1}{2} (c_i^3 + 4c_i^2 + 3c_i) \\ &\quad B((-2)^{\rho(t)}, y(x_n)) - \frac{1}{6} (c_i^3 + 3c_i^2 + 2c_i) B((-3)^{\rho(t)}, y(x_n)) \\ &\quad + h^4 \sum_j^s a_{i,j} f(Y_j) + \sum_j^s a_{i,j} B(\psi_j^{iv}(t_i), y(x_n)). \end{aligned} \tag{22}$$

It follows from (22) that, in vector form,

$$\psi(\theta) = \mathbf{e}, \quad \psi(\tau_1) = \mathbf{c}, \quad \psi(\tau_2) = \frac{1}{2}\mathbf{c}^2, \quad \text{and} \quad \psi(\tau_3) = \frac{1}{6}\mathbf{c}^3,$$

and  $\forall$  tree  $t \in T_4$  with  $\rho(t) \geq 4$ ,

$$\begin{aligned} \psi(t) = & \frac{1}{2} \left( \mathbf{c}^3 + 5\mathbf{c}^2 + 6\mathbf{c} \right) (-1)^{\rho(t)+1} - \left( \mathbf{c}^3 + 4\mathbf{c}^2 + 3\mathbf{c} \right) (-2)^{\rho(t)-1} \\ & + \frac{1}{2} \left( \mathbf{c}^3 + 3\mathbf{c}^2 + 2\mathbf{c} \right) (-3)^{\rho(t)-1} + \mathbf{A}\psi^{iv}(t). \end{aligned} \tag{23}$$

And for trees  $t = [t_1, \dots, t_m]_4 \in T_4$ , see Lemma in Sect. 3,

$$\psi_j^{iv}(t) = \rho(t)(\rho(t) - 1)(\rho(t) - 2)(\rho(t) - 3) \prod_{i=1}^m \psi_j(t_i). \tag{24}$$

By substituting (24) in (23), the coefficients  $\psi_j(t)$  can be generated recursively.

**Theorem** For exact starting values, the method (4) is said to be convergent of order  $p$  if and only if for all trees  $t \in T_4$ ,

$$\sum_{i=1}^s b_i \psi_i^{iv}(t) = 1 + 6(-1)^{\rho(t)} - 4(-2)^{\rho(t)} + (-3)^{\rho(t)}, \tag{25}$$

for  $\rho(t) \leq p + 1$  but not for some trees of order  $p + 2$ .

*Proof* The proof is similar to the proof of Theorem 1 in [14]. From (18)–(20), it is obvious that the first, second and third components of the local truncation error  $d_n$  vanish, and the fourth component is

$$\frac{1}{h^3} \left[ \begin{aligned} & B(1, y(x_n)) - 4y(x_n) + 6B((-1)^{\rho(t)}, y(x_n)) - 4B((-2)^{\rho(t)}, y(x_n)) + \\ & B((-3)^{\rho(t)}, y(x_n)) - \sum_{i=1}^s b_i B(\psi^{iv}(t), y(x_n)) \end{aligned} \right].$$

The method is of order  $p$  if  $p$  is the largest integer such that

$$d_n = O(h^{p+1}). \tag{26}$$

The only condition that makes (26) to hold for all  $n \geq 0$  is

$$\mathbf{b}^T \psi^{iv}(t) = 1 + 6(-1)^{\rho(t)} - (-2)^{\rho(t)+2} + (-3)^{\rho(t)}, \quad \text{for } \rho(t) \leq p + 1.$$

□

### 5 Algebraic Order Conditions

A relationship that exists between the coefficients of a numerical method which causes annihilation of successive terms in a Taylor series expansion of local truncation error of the method is termed order condition of the method [14]. To generate such relationships (order conditions) for HMFDM methods for trees of different orders, Eq. (25) together with (23) and (24) is used. It is worth noting that the 'order' referred to here is for the convergence of HMFDM methods not for the order of the rooted trees.

#### 5.1 Fourth-Order Tree



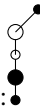
The only tree with order four in  $T_4$  is  $\tau_4 = [\theta]_4$ : , and the order condition for it is

$$\sum_{i=1}^s b_i = 1, \tag{27}$$

and (23) gives

$$\psi_i(\tau_4) = 24 \sum_{j=1}^s a_{i,j} - 6c_i^3 - 11c_i^2 - 6c_i.$$

#### 5.2 Fifth-Order Tree



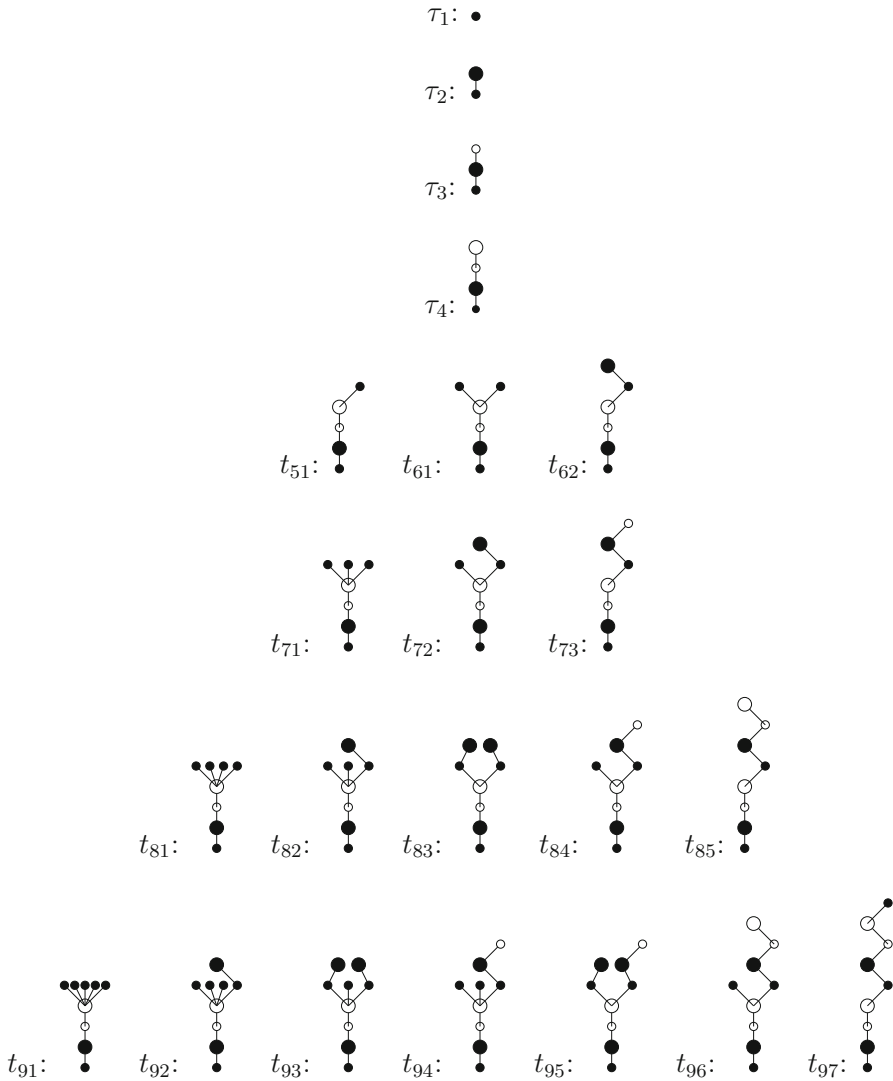
The tree in  $T_4$  with order 5 is  $t_{51} = [\tau_1]_4$ : . The corresponding order condition is

$$\sum_{i=1}^s b_i c_i = -1, \tag{28}$$

and from (24) we obtain  $\psi_i'''(t_{4,1}) = 120\psi_i(\tau_1)$ , which implies that from (23)

$$\psi_i(t_{51}) = 120 \sum_{j=1}^s a_{i,j} c_j + 25c_i^3 + 60c_i^2 + 36c_i.$$

The trees of order up to nine are constructed below in accordance with the description given in Sect. 3.



The corresponding order conditions of HMFD methods up to trees of order nine are presented in Table 2 below, where all the summations in the order conditions run from  $i, j, k, \dots$  to  $s$ .

Like the two-step hybrid methods, the summations in the equations of the order conditions of HMFD methods associated with any tree can easily be generated as follows: the stem of any tree of order  $\rho(t) \geq 4$  represents a component of vector  $\mathbf{b}$ ; each small dot vertex at the terminal donates a component of vector  $\mathbf{c}$ ; each branch from big circle vertex to a terminal big dot vertex contributes a component of  $\mathbf{c}^2$ ; each branch that grows from big circle vertex to a terminal small circle vertex contributes a component of vector  $\mathbf{c}^3$ , and each branch growing from big circle vertex to another big circle vertex contributes an element of matrix  $\mathbf{A}$ .

**Table 2** Order conditions

Tree $t$	$\rho(t)$	Order condition
$\tau_1$	1	
$\tau_2$	2	
$\tau_3$	3	
$\tau_4$	4	$\sum b_i = 1$
$t_{51}$	5	$\sum b_i c_i = -1$
$t_{61}, t_{6,2}$	6	$\sum b_i c_i^2 = \frac{4}{3}$
$t_{71}, t_{7,2}, t_{7,3}$	7	$\sum b_i c_i^3 = -2$
$t_{81}, t_{8,2}, t_{8,3}, t_{8,4}$	8	$\sum b_i c_i^4 = \frac{33}{10}$
$t_{85}$		$\sum b_i c_i a_{i,j} = -\frac{1}{720}$
$t_{91}, t_{9,2}, t_{9,3}, t_{9,4}, t_{9,5}$	9	$\sum b_i c_i^5 = -\frac{35}{6}$
$t_{96}$		$\sum b_i a_{i,j} c_j = -\frac{1}{720}$
$t_{97}$		$\sum b_i c_i a_{i,j} = \frac{1}{720}$

### 5.3 Simplifying Assumptions

It can be seen in Table 2 that the number of equations of order conditions of HMFD (like every other method, e.g., RK, RKN THM) increases with increase in the order of the trees, resulting in more equations to be solved for higher-order methods. But there exist certain relationships that naturally connect the coefficients of numerical methods which when explored appropriately can reduce the number of independent equations of order conditions. These relationships are called simplifying assumptions.

Suppose HMFD method (4) has a stage order  $q$ , so that  $Y_i(x_n) = B(\psi_i, y(x_n))$  differs from  $y(x_n + c_i h)$  by terms of order  $h^{q+1}$  [14], then

$$\psi_i(t) = c_i^{\rho(t)} \quad \text{for } \rho(t) \leq q.$$

Which implies that

$$\psi_i^{iv}(t) = \rho(t)(\rho(t) - 1)(\rho(t) - 2)(\rho(t) - 3)c_i^{\rho(t)-4},$$

and (23) gives the set of simplifying assumptions as

$$\sum_{j=1}^s a_{i,j} c_j^\lambda = \frac{c_i^{\lambda+4} - \frac{1}{2}(6c_i + 5c_i^2 + c_i^3)(-1)^{\lambda+5} + (3c_i + 4c_i^2 + c_i^3)(-2)^{\lambda+3} - \frac{1}{2}(2c_i + 3c_i^2 + c_i^3)(-3)^{\lambda+3}}{(\lambda+4)(\lambda+3)(\lambda+2)(\lambda+1)}, \tag{29}$$

where  $0 \leq \lambda \leq q - 4$ .

**Table 3** Coefficients of one-stage method (HMFDs1)

-3	0	0	0	0
-2	0	0	0	0
-1	0	0	0	0
0	0	0	0	0
	0	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

## 6 Construction of Explicit HMFD Methods

Having derived the order conditions for the proposed class of HMFD methods, we present in this section some explicit methods of the class. It is noteworthy that the proposed methods possess certain special properties, which are responsible for their computational efficiency. For example, although the methods are not self-starting, but after obtaining the starting values, the integration proceeds with  $s - 3$  function evaluation per step. These properties are given special consideration in the derivation process.

### 6.1 One-Stage Explicit HMFD Method

To construct a one-stage method, which is the first member of the proposed class of HMFD methods, algebraic order conditions (see Table 2) up to trees of order 7 are considered. Note that  $s \geq 4$  in any case of HMFD methods. Putting  $s = 4$  in the order conditions mentioned, we get

$$\begin{aligned} b_1 + b_2 + b_3 + b_4 &= 1, \\ b_1c_1 + b_2c_2 + b_3c_3 + b_4c_4 &= -1, \\ b_1c_1^2 + b_2c_2^2 + b_3c_3^2 + b_4c_4^2 &= \frac{4}{3}, \\ b_1c_1^3 + b_2c_2^3 + b_3c_3^3 + b_4c_4^3 &= -2, \end{aligned}$$

which is a system of four equations in four unknown parameters, see Table 1. This gives unique values of the unknowns. From Eq. (29) with  $\lambda = 0$ , the components of matrix **A** of the method are obtained. Summary of the coefficients is given in Table 3.

### 6.2 Two-Stage Explicit HMFD Method

To obtain a two-stage method,  $s = 5$  is considered in the order conditions up to trees of order nine in Table 2.

$$\begin{aligned} b_1 + b_2 + b_3 + b_4 + b_5 &= 1, \\ b_1c_1 + b_2c_2 + b_3c_3 + b_4c_4 + b_5c_5 &= -1, \\ b_1c_1^2 + b_2c_2^2 + b_3c_3^2 + b_4c_4^2 + b_5c_5^2 &= \frac{4}{3}, \end{aligned}$$



**Table 4** Coefficients of two-stage method (HMFDs2)

-3	0	0	0	0	0
-2	0	0	0	0	0
-1	0	0	0	0	0
0	0	0	0	0	0
1	$-\frac{1}{9887}$	$\frac{1}{8879}$	$\frac{87793536}{87786673}$	$-\frac{7871}{87786673}$	0
	$-\frac{1}{720}$	$\frac{1}{180}$	$\frac{1}{120}$	$\frac{1}{180}$	$-\frac{1}{720}$

$$\begin{aligned}
 b_1c_1^3 + b_2c_2^3 + b_3c_3^3 + b_4c_4^3 + b_5c_5^3 &= -2, \\
 b_1c_1^4 + b_2c_2^4 + b_3c_3^4 + b_4c_4^4 + b_5c_5^4 &= \frac{33}{10}, \\
 b_1c_1^5 + b_2c_2^5 + b_3c_3^5 + b_4c_4^5 + b_5c_5^5 &= -\frac{35}{6}.
 \end{aligned}$$

The system of equations above coupled with (29),  $\lambda = 0, 1$ , is solved. A unique set of coefficients of the two-stage method is obtained as given in Table 4.

### 7 Stability and Convergence Analysis

The update stage of the HMFD method (4), which is represented by its autonomous form (9), can be written as follows:

$$\sum_{i=0}^4 \gamma_i y_{n-i} - h^4 \sum_{i=0}^s b_i f(Y_i) = 0. \tag{30}$$

#### 7.1 Zero Stability

The HMFD method is said to be zero stable if the roots  $\xi_j, j = 1, 2, 3, 4$ , of the first characteristics polynomial  $\chi(\xi)$ , which is given by

$$\chi(\xi) = \sum_{i=0}^4 \gamma_i \xi^{4-i} = 0, \tag{31}$$

satisfy  $|\xi_j| \leq 1, j = 1, 2, 3, 4$  and for the roots with  $|\xi_j| = 1$ , the multiplicity does not exceed 1 (see [15, 16]).

#### 7.2 Consistency

The HMFD method is said to be consistent if it has order  $p \geq 1$ .

*Remark* We note that the first characteristics polynomial associated with (31) is

$$\chi(\xi) = \xi^4 - 4\xi^3 + 6\xi^2 - 4\xi + 1 = 0,$$

which implies that  $\xi = 1$  four times. Therefore, the HMFD method is zero stable. We also note from Table 2 that the minimum order  $p$  of the method is 4, which implies that it's consistent. Hence, we conclude that the HMFD method is convergent.

## 8 Numerical Experiment

Presented in this section is numerical experiment, in which the proposed methods alongside some existing codes are applied on several test problems. Efficiency of the codes is measured by plotting the  $\log_{10}$  of maximum errors recorded with different step lengths  $h$  in a given interval  $[a,b]$  against total function call for each code. While step length  $h$  is used in the integration with one-stage method,  $2h$ ,  $3h$  and  $4h$  are used for two-stage, three-stage and four-stage methods, respectively. The acronyms below are used in the paper:

- **HMFDs1**: the proposed one-stage explicit HMFD method derived in Sect. 6 of this paper;
- **HMFDs2**: the proposed two-stage explicit HMFD method derived in Sect. 6 of this paper;
- **RKFDs3**: three-stage explicit RKFD method presented in [11];
- **HMs4**: four-stage hybrid method for special second-order ODEs presented in [17];
- **THMs3**: three-stage explicit hybrid method given in [18];
- **AWM**: collocation method for solving fourth-order ODEs obtained in [6];
- **JTR**: Fourth-order ODEs integrator proposed in [8].

### 8.1 Implementation

To implement the proposed schemes in this paper, three starting points of the solution in addition to the given initial condition of the problem are required. These starting points can be obtained by any one-step scheme, for instance, RKFDs3 scheme in [11]. Once the starting points are generated, the next thing is to compute the components of vector  $\mathbf{Y}$ , update of the solution ( $y_{n+1}$ ) and the error ( $e_{n+1}$ ) of the method in the step. To compute these quantities at the next grid point, a fixed step length  $h$  is added to the previous grid point ( $x_{n+1} = x_n + h$ ). Here, only  $Y_4$  and  $Y_5$  are required to be computed if we are implementing the proposed HMFDs2 for example, because  $Y_1$ ,  $Y_2$ ,  $Y_3$  are readily available from the previous step, that is,

$$\begin{aligned} \text{for } n = 3 : Y_1^3 &= y_0 \quad (\text{given initial condition}); \\ Y_2^3 &= y_1 \quad (\text{computed by RKFDs3}); \\ Y_3^3 &= y_2 \quad (\text{computed by RKFDs3}); \\ & y_3 \quad (\text{computed by RKFDs3}); \end{aligned}$$

$$Y_4^3 = y_3 + \dots + h^4 \sum_{j=1}^3 a_{4,j} f(x_3 + c_j h, Y_j^3);$$

$$Y_5^3 = y_3 + \dots + h^4 \sum_{j=1}^4 a_{5,j} f(x_3 + c_j h, Y_j^3);$$

$$y_4; x_4 = x_3 + h; e_4 = |y(x_4) - y_4|;$$

for  $n = 4$  :  $Y_1^4 = y_1$  (already computed by RKFDs3 in the previous step);  
 $Y_2^4 = y_2$  (already computed by RKFDs3 in the previous step);  
 $Y_3^4 = y_3$  (already computed by RKFDs3 in the previous step);

$$Y_4^4 = y_4 + \dots + h^4 \sum_{j=1}^3 a_{4,j} f(x_4 + c_j h, Y_j^4);$$

$$Y_5^4 = y_4 + \dots + h^4 \sum_{j=1}^4 a_{5,j} f(x_4 + c_j h, Y_j^4);$$

$$y_5; x_5 = x_4 + h; e_5 = |y(x_5) - y_5|;$$

$$\vdots$$

The procedure continues until the value of  $x_n \geq b$ , where  $b$  is an upper limit of solution interval  $[a, b]$ . Maximum value of the errors ( $e_n$ ) computed at all steps is then recorded.

## 8.2 Test Problems

### Problem 1

$$y'''' = -4y, \quad 0 \leq x \leq 10,$$

$$y(0) = 1, \quad y'(0) = 1, \quad y''(0) = 2, \quad y'''(0) = 2,$$

$$y(x) = \exp(x) \sin(x). \quad \text{Source: [11]}$$

### Problem 2

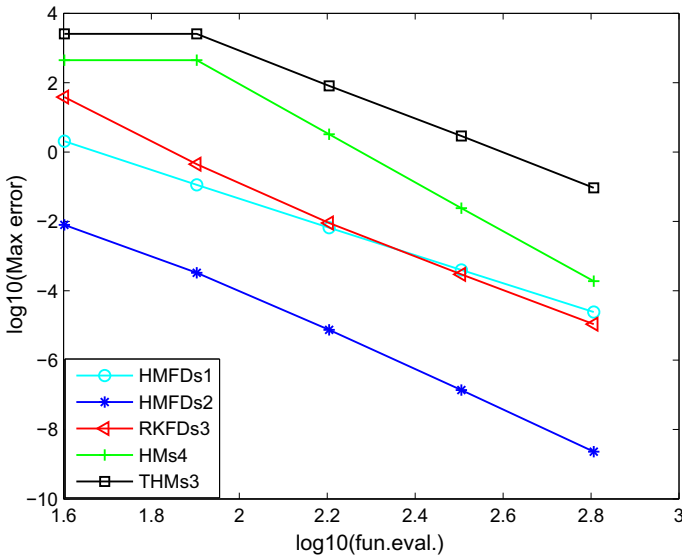
$$y'''' = y^2 + \cos^2(x) + \sin(x) - 1, \quad 0 \leq x \leq 5,$$

$$y(0) = 0, \quad y'(0) = 1, \quad y''(0) = 0, \quad y'''(0) = -1,$$

$$y(x) = \sin(x). \quad \text{Source: [11]}$$

### Problem 3

$$y'''' = \frac{3 \sin(y) (3 + 2 \sin^2(y))}{\cos^7(y)}, \quad 0 \leq x \leq \frac{\pi}{8},$$



**Fig. 1** Efficiency curves for problem 1,  $h = 2^{-i}$ ,  $i = 2, \dots, 6$

$$y(0) = 0, \quad y'(0) = 1, \quad y''(0) = 0, \quad y'''(0) = 1,$$

$$y(x) = \sin^{-1}(x). \quad \text{Source: [11]}$$

**Problem 4** the ill-posed problem of a beam on elastic foundation:

$$y'''' = x - y, \quad 0 \leq x \leq 5,$$

$$y(0) = 0, \quad y'(0) = 0, \quad y''(0) = 0, \quad y'''(0) = 0,$$

$$y(x) = 1 - 1/2 e^{-1/2 \sqrt{2}x} \cos\left(1/2 \sqrt{2}x\right) - 1/2 e^{1/2 \sqrt{2}x} \cos\left(1/2 \sqrt{2}x\right).$$

Source: [11, 12]

**Problem 5**

$$y_1'''' = e^{3x} y_2, \quad y_1(0) = 1, \quad y_1'(0) = -1, \quad y_1''(0) = 1, \quad y_1'''(0) = -1,$$

$$y_2'''' = 256e^{-x} y_3, \quad y_2(0) = 1, \quad y_2'(0) = -4, \quad y_2''(0) = 16, \quad y_2'''(0) = -64,$$

$$y_3'''' = 81e^{-x} y_4, \quad y_3(0) = 1, \quad y_3'(0) = -3, \quad y_3''(0) = 9, \quad y_3'''(0) = -27,$$

$$y_4'''' = 16e^{-x} y_1, \quad y_4(0) = 1, \quad y_4'(0) = -2, \quad y_4''(0) = 4, \quad y_4'''(0) = -8,$$

$$y_1(x) = e^{-x}, \quad y_2(x) = e^{-4x}, \quad y_3(x) = e^{-3x}, \quad y_4(x) = e^{-2x}, \quad 0 \leq x \leq 2.$$

Source: [11, 12]

The test problems are first transformed to equivalent systems of second-order equations for **HMs4** and **THMs3** to be applied. Figures 1, 2, 3, 4 and 5 show the outcome of numerical experiment where the computational efficiency of the proposed methods is

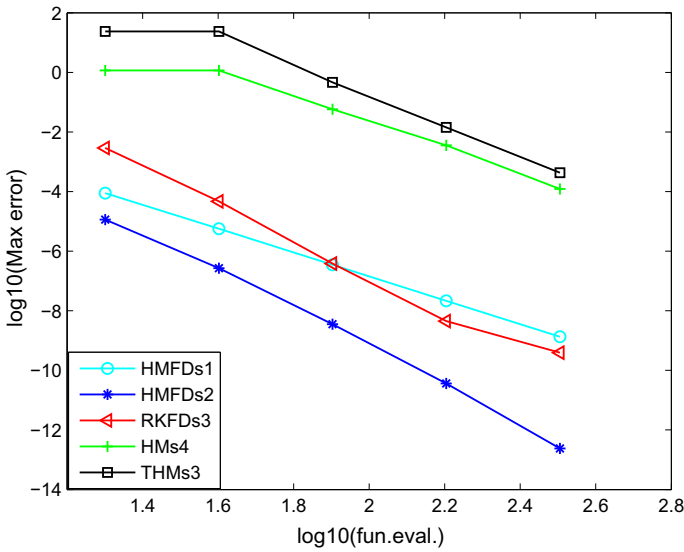


Fig. 2 Efficiency curves for problem 2,  $h = 2^{-i}$ ,  $i = 2, \dots, 6$

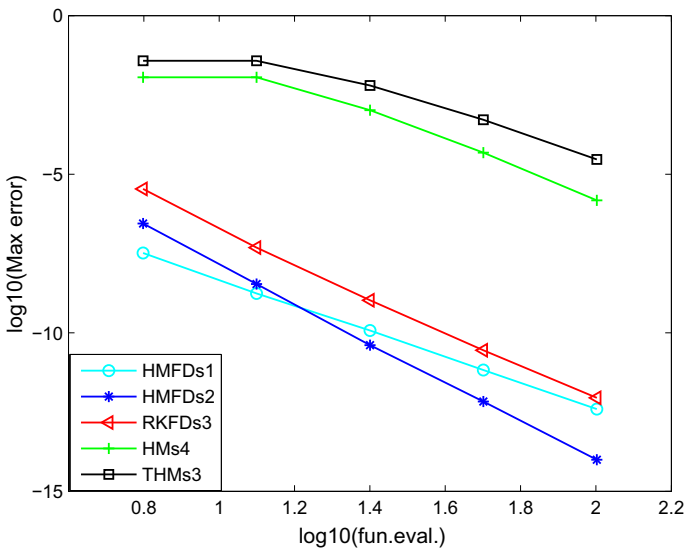


Fig. 3 Efficiency curves for problem 3,  $h = 2^{-i}$ ,  $i = 4, \dots, 8$

compared with the existing methods in the literature. For all the test problems solved, efficiency and accuracy of the proposed methods in this paper are more pronounced, especially for the two-stage method (HMFDs2).

Tables 5, 6, 7, 8 and 9 show comparison of numerical results of the proposed methods and those of fourth- order ODEs integrators of linear multistep type presented in [6] and [8], where maximum errors ( $|y(x_n) - y_n|$ ) for each method in the given intervals

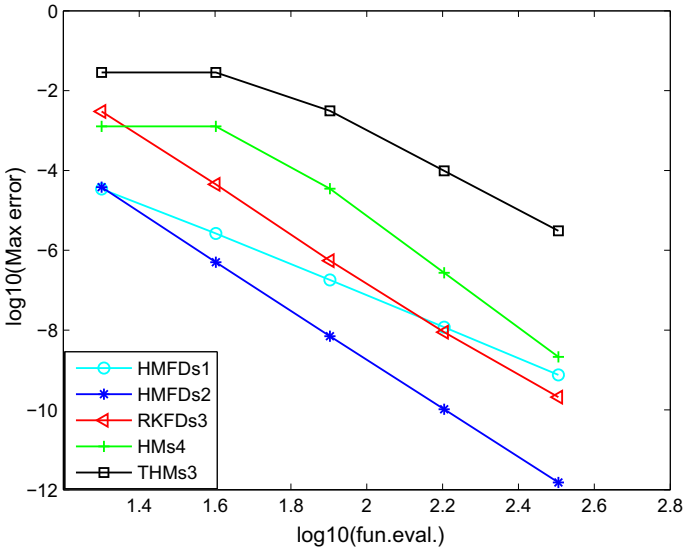


Fig. 4 Efficiency curves for problem 4,  $h = 2^{-i}$ ,  $i = 2, \dots, 6$

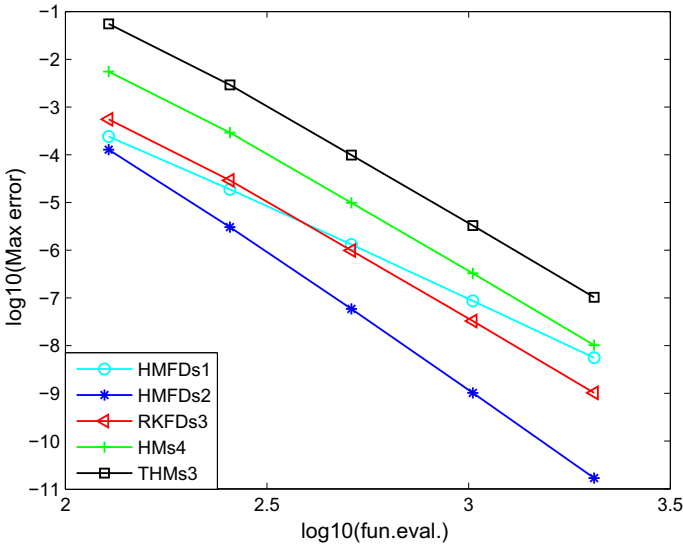


Fig. 5 Efficiency curves for problem 5

for each step-size  $h$  are recorded. It is obvious that for all the problems solved, our method records the smallest errors except for a few cases where the errors are almost equal with those of the multistep methods.

**Table 5** Results for problem 1

$h$	Method	Max.Error, $x \in [0, 10]$
1	HMFDs2	$1.94729 \times 10^{-01}$
	AWM	$3.71395 \times 10^{+01}$
	JTR	$6.58401 \times 10^{-01}$
1/2	HMFDs2	$7.90676 \times 10^{-03}$
	AWM	$6.70258 \times 10^{-01}$
	JTR	$2.63731 \times 10^{-02}$
1/4	HMFDs2	$3.30070 \times 10^{-04}$
	AWM	$4.98146 \times 10^{-02}$
	JTR	$1.10349 \times 10^{-03}$
1/8	HMFDs2	$7.42000 \times 10^{-06}$
	AWM	$3.19211 \times 10^{-03}$
	JTR	$2.48100 \times 10^{-05}$
1/16	HMFDs2	$3.66031 \times 10^{-07}$
	AWM	$1.98860 \times 10^{-04}$
	JTR	$4.56625 \times 10^{-07}$

**Table 6** Results for problem 2

$h$	Method	Max.Error, $x \in [0, 5]$
1/2	HMFDs2	$1.14600 \times 10^{-05}$
	AWM	$1.16425 \times 10^{-03}$
	JTR	$6.66600 \times 10^{-05}$
1/4	HMFDs2	$2.69815 \times 10^{-07}$
	AWM	$8.82900 \times 10^{-05}$
	JTR	$1.47237 \times 10^{-02}$
1/8	HMFDs2	$3.55471 \times 10^{-09}$
	AWM	$5.66000 \times 10^{-06}$
	JTR	$1.21883 \times 10^{-02}$
1/16	HMFDs2	$3.65268 \times 10^{-11}$
	AWM	$3.50063 \times 10^{-07}$
	JTR	$9.49966 \times 10^{-03}$
1/32	HMFDs2	$2.39235 \times 10^{-13}$
	AWM	$2.16257 \times 10^{-08}$
	JTR	$7.99706 \times 10^{-03}$

**Table 7** Results for problem 3

$h$	Method	Max.Error, $x \in [0, \pi/4]$
1/8	HMFDS2	$5.88720 \times 10^{-04}$
	AWM	$1.59700 \times 10^{-05}$
	JTR	$6.25610 \times 10^{-04}$
1/16	HMFDS2	$3.02000 \times 10^{-06}$
	AWM	$6.08000 \times 10^{-06}$
	JTR	$3.04000 \times 10^{-06}$
1/32	HMFDS2	$5.82518 \times 10^{-08}$
	AWM	$5.16395 \times 10^{-07}$
	JTR	$5.84913 \times 10^{-08}$
1/64	HMFDS2	$6.23889 \times 10^{-10}$
	AWM	$2.60858 \times 10^{-08}$
	JTR	$6.25523 \times 10^{-10}$
1/128	HMFDS2	$8.03367 \times 10^{-12}$
	AWM	$1.44112 \times 10^{-09}$
	JTR	$8.03988 \times 10^{-12}$

**Table 8** Results for problem 4

$h$	Method	Max.Error, $x \in [0, 10]$
1	HMFDS2	$1.99866 \times 10^{-01}$
	AWM	$1.84775 \times 10^{+00}$
	JTR	$6.69368 \times 10^{-01}$
1/2	HMFDS2	$2.62223 \times 10^{-03}$
	AWM	$5.17977 \times 10^{-02}$
	JTR	$8.76415 \times 10^{-03}$
1/4	HMFDS2	$3.58600 \times 10^{-05}$
	AWM	$1.94907 \times 10^{-03}$
	JTR	$1.19840 \times 10^{-04}$
1/8	HMFDS2	$5.18747 \times 10^{-07}$
	AWM	$2.60858 \times 10^{-04}$
	JTR	$1.73000 \times 10^{-06}$
1/16	HMFDS2	$7.77573 \times 10^{-09}$
	AWM	$8.60000 \times 10^{-06}$
	JTR	$2.59933 \times 10^{-08}$



**Table 9** Results for problem 5

$h$	Method	Max.Error, $x \in [0, 2]$
1/16	HMFDs2	$3.06000 \times 10^{-06}$
	AWM	$2.40340 \times 10^{-04}$
	JTR	$3.42000 \times 10^{-06}$
1/32	HMFDs2	$5.89104 \times 10^{-08}$
	AWM	$1.88300 \times 10^{-05}$
	JTR	$3.89350 \times 10^{-04}$
1/64	HMFDs2	$1.02158 \times 10^{-09}$
	AWM	$1.31000 \times 10^{-06}$
	JTR	$3.79050 \times 10^{-04}$
1/128	HMFDs2	$1.68130 \times 10^{-11}$
	AWM	$8.68127 \times 10^{-08}$
	JTR	$3.68740 \times 10^{-04}$
1/256	HMFDs2	$2.69602 \times 10^{-13}$
	AWM	$5.57561 \times 10^{-09}$
	JTR	$3.62040 \times 10^{-04}$

## 9 Conclusion

A class of four-step hybrid methods (HMFD) for direct numerical integration of special fourth-order ODEs is proposed. This class of hybrid methods is similar to the class of two-step hybrid methods for solving special second-order ODEs directly [14]. Unlike RKFD methods [11], HMFD methods do not depend on the derivatives of the solution. Using the theory of B-series with fourth-order ODEs trees that first appeared in [11], algebraic order conditions of the HMFD methods are derived. The order conditions are used to construct a one-stage and two-stage methods. Numerical results presented in Figs. 1, 2, 3, 4 and 5 reveal that the new methods proposed in this paper are more accurate and efficient when compared with the hybrid methods for solving special second-order ODEs as well as the RKFD methods recently proposed in [11]. Similarly, the results in Tables 5, 6, 7, 8 and 9 suggest the superiority of the proposed schemes in this paper over the linear multistep schemes presented in [6] and [8].

## References

1. Dormand, J.R.: Numerical Methods for Differential Equations: A Computational Approach, vol. 3, second edn. CRC Press, Boca Raton (1996)
2. Lambert, J.D.: Numerical Methods for Ordinary Differential Systems: The Initial Value Problem, third edn. Wiley, New York (1991)
3. Hairer, E., Nørsett, S.P., Wanner, G.: Solving Ordinary Differential Equations I: Nonstiff Problems, 2nd edn. Springer, Berlin (1993)
4. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations, 2nd edn. Wiley, England (2008)
5. Taiwo, O., Ogunlaran, O.: Numerical solution of fourth order linear ordinary differential equations by cubic spline collocation tau method. J. Math. Stat. **4**, 264–268 (2008)

6. Awoyemi, D.: Algorithmic collocation approach for direct solution of fourth-order initial-value problems of ordinary differential equations. *Int. J. Comput. Math.* **82**, 321–329 (2005)
7. Al-Said, E.A., Noor, M.A., Rassias, T.M.: Cubic splines method for solving fourth-order obstacle problems. *Appl. Math. Comput.* **174**, 180–187 (2006)
8. Jator, S.N.: Numerical integrators for fourth order initial and boundary value problems. *Int. J. Pure Appl. Math.* **47**, 563–576 (2008)
9. Kayode, S.J.: An efficient zero-stable numerical method for fourth-order differential equations. *Int. J. Math. Math. Sci.* (2008). doi:[10.1155/2008/364021](https://doi.org/10.1155/2008/364021)
10. Waeleh, N., Abdul Majid, Z., Ismail, F., Suleiman, M.: Numerical solution of higher order ordinary differential equations by direct block code. *J. Math. Stat.* **8**, 77–81 (2011)
11. Hussain, K., Ismail, F., Senu, N.: Solving directly special fourth-order ordinary differential equations using Runge–Kutta type method. *J. Comput. Appl. Math.* **306**, 179–199 (2016)
12. Dong, L., Alotaibi, A., Mohiuddine, S., Atluri, S.: Computational methods in engineering: a variety of primal & mixed methods, with global & local interpolations, for well-posed or ill-posed BCs. *CMES Comput. Model. Eng. Sci.* **99**, 1–85 (2014)
13. You, X., Chen, Z.: Direct integrators of Runge–Kutta type for special third-order ordinary differential equations. *Appl. Numer. Math.* **74**, 128–150 (2013)
14. Coleman, J.P.: Order conditions for a class of two-step methods for  $y'' = f(x, y)$ . *IMA J. Numer. Anal.* **23**, 197–220 (2003)
15. Ngwane, F., Jator, S.: Block hybrid method using trigonometric basis for initial value problems with oscillating solutions. *Numer. Algorithms* **63**, 713–725 (2013)
16. Ola Fatunla, S.: Block methods for second order odes. *Int. J. Comput. Math.* **41**, 55–63 (1991)
17. Jikantoro, Y., Ismail, F., Senu, N.: Higher order dispersive and dissipative hybrid method for the numerical solution of oscillatory problems. *Int. J. Comput. Math.* **93**, 929–941 (2016)
18. Franco, J.: A class of explicit two-step hybrid methods for second-order IVPs. *J. Comput. Appl. Math.* **187**, 41–57 (2006)