Hermite–Birkhoff–Obrechkoff 3-stage 6-step ODE Solver of order 14*

T. Nguyen-Ba[†] [‡] Rémi Vaillancourt[*§]

CRM-3214

April 2006

*This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and the Centre de recherches mathématiques of the Université de Montréal

[†]Department of Mathematics and Statistics, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5.

[‡]tnguyen@mathstat.uottawa.ca

[§]remi@uottawa.ca

Abstract

A 3-stage 6-step variable step Hermite–Birkhoff–Obrechkoff method of order 14, denoted by HBO14(3,6), is constructed for solving non-stiff systems of first-order differential equations of the form $y' = f(x, y)$, $y(x_0) = y_0$. Its formula uses y' and y'' as in Obrechkoff method. Forcing a Taylor expansion of the numerical solution to agree with an expansion of the true solution leads to multistep and Runge–Kutta type order conditions which are reorganized into linear Vandermonde-type systems. Fast algorithms are developed for solving these systems to obtain Hermite–Birkhoff interpolation polynomials in terms of generalized Lagrange basis functions. The new method has a larger region of absolute stability than Adams–Bashforth–Moulton methods of orders 10 to 13 in PECE mode. The stepsize is controlled by a local error estimator. HBO14(3,6) is superior to MATLAB's `ode113` in solving several problems often used to test higher order ode solvers based on the number of steps, cpu time, and maximum global error.

Mathematical Subject Classification: Primary 65L06; Secondary 65D05, 65D30

Keywords and phrases: general linear method for nonstiff ode's, Hermite–Birkhoff method, Obrechkoff method, Vandermonde-type systems, maximum global error, number of function evaluations, cpu time, MATLAB's `ode113`, comparing ode solvers

Résumé

On construit un solveur à 3 étages et à 6 pas d'ordre 14 de type Hermite–Birkhoff–Obrechkoff, noté HBO14(3,6), pour systèmes d'équations différentielles non raides $y' = f(x, y)$, $y(x_0) = y_0$, faisant usage de y' et y''. On dérive les conditions d'ordre de types Runge–Kutta et multipas en identifiant les coefficients du développement de Taylor de la solution numérique et de la solution exacte qu'on réorganise en systèmes de type Vandermonde. La solution de ces systèmes au moyen de nouveaux algorithmes rapides donne lieu à des polynômes d'interpolation d'Hermite–Birkhoff sur une base de fonctions de Lagrange généralisées. La nouvelle méthode admet un domaine de stabilité absolu plus grand que ceux des méthodes d'Adams–Bashforth–Moulton d'ordre 10 à 13 en mode PECE. Un estimateur de l'erreur locale contrôle le pas. HBO14(3,6) est supérieure à `ode113` de MATLAB sur des problèmes souvent utilisés pour tester les méthodes d'ordre élevé quant au nombre de pas, au temps machine et à l'erreur globale maximum.

1 Introduction

An explicit 6-step Obrechkoff method [21] and a 3-stage Runge–Kutta method of order 3 are cast into a 3-stage 6-step Hermite–Birkhoff–Obrechkoff method of order 14 named HBO14(3,6) (HBO14, for short, in this paper). The 6 steps consist in the current step and 5 backsteps. The method's name was chosen because it uses Hermite–Birkhoff interpolation polynomials and first and second order derivatives of y at step points like Obrechkoff methods. The link between the two types of methods is that values at off-step points are obtained by means of predictors which use values at previous points.

Milne [18] was perhaps the first to have advocated the use of multiderivative, multistep Obrechkoff formulae for the numerical solution of differential equations. More recently, Huang and Innanen [14] introduced a new form of the classical Adams–Cowell methods and multiderivative, multistep methods some of which having larger stability interval and smaller truncation error than classical multistep methods.

Scientific computation widely uses variable step, variable order Adams–Bashforth–Moulton multistep methods of orders 1 to 14 as implemented by Gear [9] and [10], Krogh [16], and up to order 13, by Shampine [22] in MATLAB's `ode113` ([2], [23]). The codes DVDQ of Krogh [15] and DIFSUB of Gear [10] prompted the recognition of the effectiveness of a variable order, variable step formulation of Adams methods. When the equation is expensive to evaluate, high-order solvers appear to be more efficient than lower order ones.

HBO14 is designed for solving nonstiff systems of first-order initial value problems of the form

$$y' = f(x, y), \qquad y(x_0) = y_0, \qquad \text{where} \quad ' = \frac{d}{dx}. \tag{1}$$

Forcing a Taylor expansion of the numerical solution to agree with an expansion of the true solution leads to multistep and Runge–Kutta type order conditions which are reorganized into linear Vandermonde-type systems. The solutions of these systems are obtained as generalized Lagrange basis functions by new fast algorithms

HBO14 has a smaller error constant and a larger interval of absolute stability than Adams–Bashforth–Moulton methods of orders 10 to 13 in PECE mode.

The performance of HBO14 and MATLAB's `ode113` was compared on several problems often used to test higher order ode solvers. It is seen that HBO14 requires fewer steps and lower cpu time, and has higher accuracy than `ode113`.

Section 2 introduces general variable step HBO14 of order 14. Order conditions are listed in Section 3. In Section 4 this general HBO14 is represented in terms of Vandermonde-type systems. In Section 5 new symbolic algorithms are derived to diagonalize the coefficient matrices of these systems as functions of the parameters of the systems. Section 6 defines a particular variable step HBO14. In sections 7, a particular HBO14 is constructed by a fast solution of the systems. Section 8 considers the region of absolute stability of a constant step HBO14 and its principal local truncation coefficients. Section 9 deals with the step control. In section 10, three criteria are used to compare the performance of the methods considered in this paper. Appendix A lists the algorithms and Appendix B describes the MATLAB programming.

2 General variable step HBO14

A Hermite–Birkhoff–Obrechkoff method is said to be a **general** variable step Hermite–Birkhoff–Obrechkoff method if its backstep and off-step points are variable parameters. If the off-step points are fixed, the method is said to be a **particular** variable step method.

A general HBO14 requires the following four formulae to perform the integration step from x_n to x_{n+1}. For notational simplicity, $c_1 = 0$ is used in the summations.

(P_2) A Hermite–Birkhoff polynomial of degree 12 is used as predictor P_2 to obtain y_{n+c_2} to order 12,

$$y_{n+c_2} = y_n + h_{n+1}\left(a_{21} f_{n+c_1} + \sum_{j=1}^{5} \beta_{2j} f_{n-j}\right) + h_{n+1}^2 \left(\sum_{j=0}^{5} \gamma_{2j} f'_{n-j}\right). \tag{2}$$

(P_3) A Hermite–Birkhoff polynomial of degree 13 is used as predictor P_3 to obtain y_{n+c_3} to order 12,

$$y_{n+c_3} = y_n + h_{n+1}\left(\sum_{j=1}^{2} a_{3j} f_{n+c_j} + \sum_{j=1}^{5} \beta_{3j} f_{n-j}\right) + h_{n+1}^2 \left(\sum_{j=0}^{5} \gamma_{3j} f'_{n-j}\right). \tag{3}$$

(IF) A Hermite–Birkhoff polynomial of degree 14 is used as integration formula IF to obtain y_{n+1} to order 14,

$$y_{n+1} = y_n + h_{n+1}\left(\sum_{j=1}^{2} b_{1j} f_{n+c_j} + b_{13} f_{n+c_3} + \sum_{j=1}^{5} \beta_{1j} f_{n-j}\right) + h_{n+1}^2 \left(\sum_{j=0}^{5} \gamma_{1j} f'_{n-j}\right). \tag{4}$$

(P$_4$) A Hermite–Birkhoff polynomial of degree 12 is used as step control predictor P$_4$ to obtain $\widetilde{y}_{n+1}$ to order 12,

$$\widetilde{y}_{n+1} = y_n + h_{n+1}\left(a_{41}f_n + a_{43}f_{n+1} + \sum_{j=1}^{4}\beta_{4j}f_{n-j}\right) + h_{n+1}^2\left(\sum_{j=0}^{5}\gamma_{4j}f'_{n-j}\right). \tag{5}$$

We note that f'_{n+1} is computed only once at x_{n+1}.

The off-step points of HBO14 satisfy the following Runge–Kutta type simplifying conditions:

$$c_i = \sum_{j=1}^{i-1} a_{ij} + \sum_{\ell=1}^{5}\beta_{i\ell}, \qquad i = 2, 3. \tag{6}$$

3 Order conditions for general HBO14

As in similar search for ODE solvers, we impose the following simplifying assumptions [6], [19] on HBO14:

$$\sum_{j=1}^{i-1} a_{ij}c_j^k + k!B_i(k+1) = \frac{1}{k+1}\,c_i^{k+1}, \qquad \begin{cases} i = 2, 3, \\ k = 0, 1, 2, \ldots, 12, \end{cases} \tag{7}$$

where

$$B_i(j) = \sum_{\ell=1}^{5}\left[\beta_{i\ell}\frac{\eta_{\ell+1}^{j-1}}{(j-1)!}\right] + \sum_{\ell=1}^{5}\left[\gamma_{i\ell}\frac{\eta_{\ell+1}^{j-2}}{(j-2)!}\right], \qquad \begin{cases} i = 2, 3, \\ j = 0, 1, 2, \ldots, 14, \end{cases} \tag{8}$$

and

$$\eta_j = -\frac{1}{h_{n+1}}\left(x_n - x_{n+1-j}\right) = -\frac{1}{h_{n+1}}\sum_{i=0}^{j-1} h_{n-i}, \qquad j = 2, 3, \ldots, 6. \tag{9}$$

Equation (9) will be frequently used in this paper without further reference.

There remain two sets of equations to be solved:

$$\sum_{i=1}^{3} b_{1i}c_i^k + k!B_1(k+1) = \frac{1}{k+1}, \qquad k = 0, 1, \ldots, 13, \tag{10}$$

$$\sum_{i=2}^{3} b_{1i}\left[\sum_{j=1}^{i-1} a_{ij}\frac{c_j^{12}}{12!} + B_i(13)\right] + B_1(14) = \frac{1}{14!}, \tag{11}$$

where

$$B_1(j) = \sum_{i=1}^{5}\beta_{1i}\frac{\eta_{i+1}^{j-1}}{(j-1)!} + \sum_{i=1}^{5}\gamma_{1i}\frac{\eta_{i+1}^{j-2}}{(j-2)!}, \qquad j = 1, \ldots, 15. \tag{12}$$

We note that equations (10), for $k = 0, 1, \ldots, 12$, are multistep type order conditions. On the other hand, equation (10) for $k = 13$ and equation (11) are Runge–Kutta type order conditions. The numbers $B_1(k)$, $B_2(k)$ and $B_3(k)$ are associated with IF, P$_2$ and P$_3$, respectively.

4 Vandermonde-type formulation of general HBO14

4.1 The integration formula IF

The 14-vector of the reordered coefficients of integration formula IF (4),

$$\boldsymbol{u}^1 = [b_{11}, \gamma_{10}, \beta_{11}, \gamma_{11}, \beta_{12}, \gamma_{12}, \beta_{13}, \gamma_{13}, \beta_{14}, \gamma_{14}, \beta_{15}, \gamma_{15}, b_{13}, b_{12}]^T,$$

is the solution of the Vandermonde-type system of order conditions

$$M^1\boldsymbol{u}^1 = \boldsymbol{r}^1. \tag{13}$$

where

$M^1 =$

$$\begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 & 1 & 1 \\ 0 & 1 & \eta_2 & 1 & \cdots & \eta_6 & 1 & c_3 & c_2 \\ 0 & 0 & \eta_2^2/2! & \eta_2 & \cdots & \eta_6^2/2! & \eta_6 & c_3^2/2! & c_2^2/2! \\ \vdots & & & & & & & & \vdots \\ 0 & 0 & \eta_2^{13}/13! & \eta_2^{12}/12! & \cdots & \eta_6^{13}/13! & \eta_6^{12}/12! & c_3^{13}/13! & c_2^{13}/13! \end{bmatrix} \tag{14}$$

and $\boldsymbol{r}^1 = r_1(1:14)$ has components

$$r_1(i) = 1/i!, \qquad i = 1, 2, \ldots, 14.$$

The leading error term of IF is

$$\left[\sum_{j=1}^{5} \beta_{1j} \frac{\eta_{j+1}^{14}}{14!} + \sum_{j=1}^{5} \gamma_{1j} \frac{\eta_{j+1}^{13}}{13!} + b_{13} \frac{c_3^{14}}{14!} + b_{12} \frac{c_2^{14}}{14!} - \frac{1}{15!}\right] h_{n+1}^{15} y_n^{15}.$$

4.2 The predictor P_2

The 12-vector of the reordered coefficients of predictor P_2 (2),

$$\boldsymbol{u}^2 = [a_{21}, \gamma_{20}, \beta_{21}, \gamma_{21}, \beta_{22}, \gamma_{22}, \beta_{23}, \gamma_{23}, \beta_{24}, \gamma_{24}, \beta_{25}, \gamma_{25}]^T,$$

is the solution of the system of order conditions

$$M^2 \boldsymbol{u}^2 = \boldsymbol{r}^2, \tag{15}$$

where

$$M^2 = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & \eta_2 & 1 & \cdots & \eta_6 & 1 \\ 0 & 0 & \eta_2^2/2! & \eta_2 & \cdots & \eta_6^2/2! & \eta_6 \\ \vdots & & & & & & \vdots \\ 0 & 0 & \eta_2^{13}/13! & \eta_2^{12}/12! & \cdots & \eta_6^{13}/13! & \eta_6^{12}/12! \end{bmatrix} \tag{16}$$

and $\boldsymbol{r}^2 = r_2(1:12)$ has components

$$r_2(i) = c_2^i/i!, \qquad i = 1, 2, \ldots, 12.$$

A truncated Taylor expansion of the right-hand side of (2) about x_n gives

$$\sum_{j=0}^{15} S_2(j) h_{n+1}^j y_n^{(j)}$$

with coefficients

$$S_2(j) = M^2(j, 1:12)\boldsymbol{u}^2 = r_2(j) = \frac{c_2^j}{j!}, \qquad j = 1, 2, \ldots, 12,$$

$$S_2(j) = \sum_{i=1}^{5} \beta_{2i} \frac{\eta_{i+1}^{j-1}}{(j-1)!} + \sum_{i=0}^{5} \gamma_{2i} \frac{\eta_{i+1}^{j-2}}{(j-2)!}, \qquad j = 13, 14, 15.$$

We note that P_2 is of order 12 since it satisfies the order conditions

$$S_2(j) = c_2^j/j!, \qquad j = 1, \ldots, 12,$$

and its leading error term is

$$\left[S_2(13) - \frac{c_2^{13}}{13!}\right] h_{n+1}^{13} y_n^{(13)}.$$

4.3 The predictor P_3

The 13-vector of the reordered coefficients of predictor P_3 (3),

$$\boldsymbol{u}^3 = [a_{31}, \gamma_{30}, \beta_{31}, \gamma_{31}, \beta_{32}, \gamma_{32}, \beta_{33}, \gamma_{33}, \beta_{34}, \gamma_{34}, \beta_{35}, \gamma_{35}, a_{32}]^T,$$

is the solution of the system of order conditions

$$M^3 \boldsymbol{u}^3 = \boldsymbol{r}^3, \tag{17}$$

where

$$M^3 = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 & 1 \\ 0 & 1 & \eta_2 & 1 & \cdots & \eta_6 & 1 & c_2 \\ 0 & 0 & \eta_2^2/2! & \eta_2 & \cdots & \eta_6^2/2! & \eta_6 & c_2^2/2! \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \eta_2^{12}/12! & \eta_2^{11}/11! & \cdots & \eta_6^{12}/12! & \eta_6^{11}/11! & c_2^{12}/12! \end{bmatrix}. \tag{18}$$

The first 12 components of $\boldsymbol{r}^3 = r_3(1:13)$ are

$$r_3(i) = c_3^i/i!, \qquad i = 1, 2, \ldots, 12,$$

and the 13th component is

$$r_3(13) = \frac{1}{b_{13}} \left[\frac{1}{14!} - b_{12} S_2(13) - B_1(14) \right],$$

which corresponds to the Runge–Kutta order condition (11).

A truncated Taylor expansion of the right-hand side of (3) about x_n gives

$$\sum_{j=0}^{15} S_3(j) h_{n+1}^j y_n^{(j)}$$

with coefficients

$$S_3(j) = M^3(j, 1:13)\boldsymbol{u}^3 = r_3(j) = \frac{c_3^j}{j!}, \qquad j = 1, 2, \ldots, 12,$$

$$S_3(j) = a_{32} S_2(j-1) + \sum_{i=1}^{5} \beta_{3i} \frac{\eta_{i+1}^{j-1}}{(j-1)!} + \sum_{i=1}^{5} \gamma_{3i} \frac{\eta_{i+1}^{j-2}}{(j-2)!} \qquad j = 13, 14, 15.$$

4.4 The step control predictor P_4

The 12-vector of the reordered coefficients of the step control predictor P_4 (5),

$$\boldsymbol{u}^4 = [a_{41}, \gamma_{40}, \beta_{41}, \gamma_{41}, \beta_{42}, \gamma_{42}, \beta_{43}, \gamma_{43}, \beta_{44}, \gamma_{44}, a_{43}, \gamma_{45}]^T,$$

is the solution of the system of order conditions

$$M^4 \boldsymbol{u}^4 = \boldsymbol{r}^4, \tag{19}$$

where

$$M^4 = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 & 1 & 0 \\ 0 & 1 & \eta_2 & 1 & \cdots & \eta_5 & 1 & c_3 & 1 \\ 0 & 0 & \eta_2^2/2! & \eta_2 & \cdots & \eta_5^2/2! & \eta_5 & c_3^2/2! & \eta_6 \\ \vdots & & & & & & & & \vdots \\ 0 & 0 & \eta_2^{11}/11! & \eta_2^{10}/10! & \cdots & \eta_5^{11}/11! & \eta_5^{10}/10! & c_3^{11}/11! & \eta_6^{10}/10! \end{bmatrix} \tag{20}$$

and $\boldsymbol{r}^4 = r_4(1:12)$ has components

$$r_4(i) = 1/i!, \qquad i = 1, 2, \ldots, 12.$$

5 Symbolic elementary matrix functions

Consider the matrices

$$M^\ell \in \mathbb{R}^{m_\ell \times m_\ell}, \qquad \ell = 1, 2, 3, 4, \tag{21}$$

of the Vandermonde-type systems (13), (15), (17), and (19), where

$$m_1 = 14, \quad m_2 = 12, \quad m_3 = 13, \quad m_4 = 12. \tag{22}$$

A fast solution of these systems in $O(m_\ell^2)$ operations will be achieved by decomposing $(M^\ell)^{-1}$ into the product of lower and upper bidiagonal matrices, one diagonal matrix and one upper tridiagonal matrix.

The purpose of this section is to construct elementary lower and upper bidiagonal symbolic matrix functions of the parameters of HBO14 to be used in Section 7 to diagonalize M_ℓ, $\ell = 1, 2, 3$.

Since the Vandermonde-type matrices M^ℓ can be decomposed into the product of a diagonal matrix containing reciprocals of factorials and a confluent Vandermonde matrix, the factorizations used in this paper hold following the approach of Björck and Pereyra [4], Krogh [16], Galimberti and Pereyra [8] and Björck and Elfving [3]. Pivoting is not needed in this decomposition because of the special structure of Vandermonde-type matrices.

5.1 Construction of lower bidiagonal matrices

We first describe the zeroing process of a general vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]^T$ with no zero elements. The lower bidiagonal matrix

$$L_k = \begin{bmatrix} I_{k-1} & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 1 & -\tau_{k+1} & & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & -\tau_m \end{bmatrix} \tag{23}$$

defined by the multipliers

$$\tau_i = \frac{x_{i-1}}{x_i} = -L_k(i,i), \qquad i = k+1, k+2, \ldots, m, \tag{24}$$

zeros the last $(m-k)$ components, $x_{k+1}, \ldots, x_m$, of $\boldsymbol{x}$.

For $k = 3, 4, \ldots, m_\ell - 1$, left multiplying $T = L^\ell_{k-1} \cdots L^\ell_4 L^\ell_3 M^\ell$ by L^ℓ_k zeros the last $(m_l - k)$ components of the kth column of T. Thus we obtain the upper triangular matrix

$$L^\ell M^\ell = L^\ell_{m_\ell - 1} \cdots L^\ell_4 L^\ell_3 M^\ell \tag{25}$$

in $(m_\ell - 3)$ steps. We note that L^ℓ does not change the first two rows of M^ℓ.

Process 1. At the kth step, starting with $k = 3$,

- $M^{\ell(k-1)} = L^\ell_{k-1} L^\ell_{k-2} \cdots L^\ell_3 M^\ell$ is an upper triangular matrix in columns 1 to $k-1$.
- The multipliers in L^ℓ_k are obtained from $M^{\ell(k-1)}(k+1 : m_l, k)$ since $M^\ell(i,k) \neq 0$ for $i = k+1, k+2, \ldots, m_l$.

Algorithm 1 in Appendix A describes this process.

5.2 Construction of initializing upper tridiagonal matrices U^ℓ_1 for M^ℓ

The second step in diagonalizing M^ℓ transforms the first two rows of $L^\ell M^\ell$ by right multiplication by an upper tridiagonal matrix U^ℓ_1 such that

$$L^\ell M^\ell U^\ell_1(1:2, 1:m_\ell) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{26}$$

The action of U^ℓ_1 amounts to take the divided difference of the odd-numbered columns of $L^\ell M^\ell$ plus the last column if m_ℓ is even (cf. [3]). The precise form of U^ℓ_1 is postponed until Section 7 since it is defined in terms of each M^ℓ.

5.3 Construction of upper bidiagonal matrices for M^ℓ, $\ell = 1, 2, 3$

We construct upper bidiagonal matrices U^ℓ_k, $k = 2, 3, \ldots, m_\ell - 1$ whose right multiplication on $L^\ell M^\ell U^\ell_1$ amounts to take divided differences of order k, for $k = 2, 3, \ldots, m_\ell - 1$, of the columns k to m_ℓ of the matrices on which they act.

Specifically, consider the two-row matrix:

$$\begin{aligned} &L^\ell M^\ell U^\ell_1 \cdots U^\ell_{k-1}(k : k+1, 1 : m_\ell) \\ &\qquad = \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 1 & \cdots & 1 & 1 \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & y_{k+1,k+1} & \cdots & y_{k+1,m_\ell-1} & y_{k+1,m_\ell} \end{bmatrix} \end{aligned} \tag{27}$$

and define the upper bidiagonal matrix

$$U^\ell_k = \begin{bmatrix} I_{k-1} & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 1 & -\sigma_{k+1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma_{k+1} & -\sigma_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{m_\ell-2} & -\sigma_{m_\ell-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{m_\ell-1} & -\sigma_{m_\ell} \\ 0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{m_\ell} \end{bmatrix} \tag{28}$$

by means of the divisors

$$\sigma_i = \frac{1}{y_{2,i} - y_{2,i-1}} = U^\ell_k(i,i), \qquad i = k+1, k+2, \ldots, m_\ell. \tag{29}$$

Then right multiplication of (27) by U_k^ℓ zeros the 1's in positions $k+1,\ldots,m_\ell$ in the first row of and puts 1's in positions $k+1,\ldots,m_\ell$ in the second row:

$$L^\ell M^\ell U_1^\ell \cdots U_{k-1}^\ell U_k^\ell(k:k+1,1:m_\ell) = \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 0 & \cdots & 0 & 0 \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & 1 & \cdots & 1 & 1 \end{bmatrix}. \tag{30}$$

Applying U_k^ℓ, $k=2,3,\ldots,m_\ell-1$, on the right of the upper triangular matrix $L^\ell M^\ell U_1^\ell$, we obtain the diagonal matrix

$$D^\ell = L^\ell M^\ell U^\ell = L_{m_\ell-1}^\ell \cdots L_4^\ell L_3^\ell M^\ell U_1^\ell U_2^\ell \cdots U_{m_\ell-1}^\ell \tag{31}$$

in $(m_\ell-2)$ steps.

Process 2. At the kth step, starting with $k=2$,

- $M^{\ell(k-1)} = (L^\ell M^\ell U_1^\ell)U_2^\ell \cdots U_{k-1}^\ell$ is a diagonal matrix in rows 1 to $k-1$.
- The divisors in U_k^ℓ are obtained from $M^{\ell(k-1)}(k+1,k+1:m_\ell)$ since $M^{\ell(k-1)}(k+1,j)-M^{\ell(k-1)}(k+1,j-1)\neq 0$ for $j=k+1,k+2,\ldots,m_\ell-1$.

Algorithm 2 in Appendix A describes this process.

5.4 Construction of upper bidiagonal matrices for M^4

The construction of upper bidiagonal matrices for M^4 differs slightly from the construction for M^ℓ, $\ell=1,2,3$, and requires a consideration of its own. For the matrix $L^4M^4U_1^4$ we construct upper bidiagonal matrices $U_2^4, U_3^4\ldots,U_{10}^4$ such that the upper triangular matrix $U^4 = U_1^4U_2^4\cdots U_9^4U_{10}^4$ transforms L^4M^4 into a matrix $W^4 = L^4M^4U^4$ with nonzero diagonal elements, $W^4(i,i)\neq 0$, $i=1,2,\ldots,12$, and nonzero second to last elements, $W^4(2:12,12)\neq 0$, in the last column, and zero elsewhere. We call such a matrix a "diagonal+last-column" matrix.

We describe the zeroing process of the upper bidiagonal matrix U_k^4 on the two-row matrix

$$L^4M^4U_1^4\cdots U_{k-1}^4(k:k+1,1:12) = \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 1 & \cdots & 1 & y_{k,12} \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & y_{k+1,k+1} & \cdots & y_{k+1,11} & y_{k+1,12} \end{bmatrix} \tag{32}$$

which differs from matrix (27) in the last column. The divisors

$$\sigma_i = \frac{1}{y_{2,i}-y_{2,i-1}} = U_k^4(i,i), \qquad i=k+1,k+2,\ldots,11, \tag{33}$$

define

$$U_k^4 = \begin{bmatrix} I_{k-1} & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 1 & -\sigma_{k+1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma_{k+1} & -\sigma_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{10} & -\sigma_{11} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{11} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \tag{34}$$

which differs from matrix (28) in the last column. Right-multiplication of (32) by U_k^4 zeros the 1's in position $k,\ldots,11$ in the first row and puts 1's in position $k+1,\ldots,11$ in the second row:

$$L^4M^4U_1^4\cdots U_{k-1}^4U_k^4(k:k+1,1:12) = \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 0 & \cdots & 0 & y_{k,12} \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & 1 & \cdots & 1 & y_{k+1,12} \end{bmatrix}. \tag{35}$$

Thus, $U^4 = U_1^4U_2^4\cdots U_{10}^4$ transforms the upper triangular matrix L^4M^4 of (25) into the diagonal+last-column matrix

$$W^4 = L^4M^4U_1^4U_2^4\cdots U_{10}^4 \tag{36}$$

in ten steps.

Process 3. At the kth step, starting with $k = 2$,

- $M^{4(k-1)} = L^4 M^4 U_1^4 U_2^4 \cdots U_{k-1}^4$ is a diagonal+last column matrix in rows 1 to $k-1$.
- The divisors in U_k^4 are obtained from $M^{4(k-1)}(k+1, k+1:12)$ since $M^{4(k-1)}(k+1, j) - M^{4(k-1)}(k+1, j-1) \neq 0$, $j = k+1, k+2, \ldots, 11$.

Algorithm 3 in Appendix A describes this process.

6 Particular variable step HBO14

In the general HBO14 of section 2, c_2 and c_3 are free parameters. After an appropriate choice of c_2 and c_3, one has a particular variable step HBO14 that depends only on h_{n+1} and the previous nodes, $x_n, x_{n-1}, \ldots, x_{n-5}$, which determine $\eta_2, \ldots, \eta_6$ in (9).

After extensive numerical experimentation, a good **particular** variable step HBO14 was obtained with

$$c_1 = 0, \qquad c_2 = \frac{2}{3}, \qquad c_3 = 1. \tag{37}$$

The remaining of this paper is concerned with the particular variable step HBO14 with coefficients c_j given in (37).

The procedure to advance integration from x_n to x_{n+1} is as follows.

(a) The stepsize, h_{n+1}, is obtained by formula (55) of section 9.

(b) The numbers $\eta_2, \ldots, \eta_6$, defined in (9), are calculated.

(c) The coefficients of integration formula IF, predictors P_2, P_3 and step control predictor P_4 are obtained successively as solutions of systems (13), (15), (17) and (19), respectively.

(d) The values y_{n+c_2}, y_{n+c_3}, y_{n+1}, and $\widetilde{y}_{n+1}$ are obtained by formulae (2)–(5).

(e) The step is accepted if $|y_{n+1} - \widetilde{y}_{n+1}|$ is smaller than the chosen tolerance and the program goes to (a) with n replaced by $n+1$. Otherwise the program returns to (a) and a new smaller stepsize h_{n+1} is computed.

7 Fast solution of particular HBO14

The elementary matrices L_k^ℓ and U_k^ℓ, $\ell = 1, 2, 3, 4$, are constructed only once as functions of $\eta_2, \ldots, \eta_6$. Then these functions are used by the fast Algorithms 4 and 5 in Appendix A to solve systems (13), (15), (17), and (19) at each integration step.

7.1 Solution of $M^1 \boldsymbol{u}^1 = \boldsymbol{r}^1$

Firstly, the elimination procedure of subsection 5.1 is applied to M^1 of (14) to construct 14×14 lower bidiagonal matrices L_k^1, $k = 3, \ldots, 13$, of the form (23) defined by the multipliers

$$\tau_i = \frac{i+2-k}{\mu_1(k)} = -L_k^1(i,i), \qquad i = k+1, k+2, \ldots, 14, \tag{38}$$

where

$$\mu_1(k) = \begin{cases} M^1(2,k), & k = 1, 3, 5, \ldots, 13, \\ M^1(3,k), & k = 2, 4, 6, \ldots, 12. \end{cases}$$

Left multiplying M^1 by L_k^1, $k = 3, \ldots, 13$, produces the upper triangular matrix $L^1 M^1 = L_{13}^1 \cdots L_3^1 M^1$ of the form (25), where $m_\ell = 14$.

Secondly, we construct the upper tridiagonal matrix U_1^1 which transforms the matrix M^1 into the matrix of first order divided differences $M^1 U_1^1$ of the odd-numbered columns of M^1 plus the last one where the divisors are taken from the second row of M^1. The diagonal entries of U_1^1 are ones, except for

$$U_1^1(3,3) = \frac{1}{\eta_2}, \quad U_1^1(5,5) = \frac{1}{\eta_3 - \eta_2}, \quad U_1^1(7,7) = \frac{1}{\eta_4 - \eta_3}, \quad U_1^1(9,9) = \frac{1}{\eta_5 - \eta_4},$$
$$U_1^1(11,11) = \frac{1}{\eta_6 - \eta_5}, \quad U_1^1(13,13) = \frac{1}{c_3 - \eta_6}, \quad U_1^1(14,14) = \frac{1}{c_2 - c_3},$$

and the non-diagonal entries are zeros, except for

$$U_1^1(i-2,i) = -U_1^1(i,i), \qquad i = 3,5,7,\ldots,13,$$
$$U_1^1(13,14) = -U_1^1(14,14).$$

The first two rows of $M^1U_1^1$ are as in (26).

Thirdly, the elimination procedure of subsection 5.3 is used to construct 14×14 upper bidiagonal matrices U_k^1, $k = 2,\ldots,13$, of the form (28) defined by the divisors

$$\sigma_i = \frac{k}{\mu_1(i) - \mu_1(i-k)} = U_k^1(i,i), \qquad i = k+1, k+2, \ldots, 14. \tag{39}$$

Right multiplying L^1M^1 by U_k^1, $k = 1,\ldots,13$, produces the diagonal matrix

$$D^1 = L_{13}^1 L_{12}^1 \cdots L_3^1 M^1 U_1^1 U_2^1 \cdots U_{13}^1,$$

where

$$D^1(i,i) = 1, \qquad i = 1,2,3,$$

and

$$D^1(i,i) = \frac{(i-1)!}{2[-\mu_1(3)]\,[-\mu_1(4)]\cdots[-\mu_1(i-1)]}, \qquad i = 4,5,\ldots,14.$$

Lastly, M^1 is decomposed into the product of elementary matrices:

$$M^1 = \left(L_{13}^1 L_{12}^1 \cdots L_3^1\right)^{-1} D^1 \left(U_1^1 U_2^1 \cdots U_{13}^1\right)^{-1}$$

and the solution of system (13) is

$$\boldsymbol{u}^1 = U_1^1 U_2^1 \cdots U_{13}^1 (D^1)^{-1} L_{13}^1 L_{12}^1 \cdots L_3^1\, \boldsymbol{r}^1, \tag{40}$$

where fast computation goes from right to left.

Process 4. Procedure (40) is implemented in the following steps:

- Algorithm 4 overwrites $\boldsymbol{r}^1 = r_1(1:m)$ with $U_2^1 \cdots U_{13}^1 (D^1)^{-1} L_{13}^1 L_{12}^1 \cdots L_3^1\, \boldsymbol{r}^1$ in $O(m^2)$ operations, where $m = 14$. The input is $M = M^1$; $m = 14$; $\boldsymbol{r} = \boldsymbol{r}^1$; $L_k = L_k^1$, $k = 3,4,\ldots,m-1$; $U_k = U_k^1$, $k = 2,\ldots,m-1$; and $D = D^1$.

- The following iteration overwrites $\boldsymbol{r}^1 = r_1(1:m)$ with $U_1^1\boldsymbol{r}^1$:

$$\begin{aligned} r_1(i) &= r_1(i)U_1^1(i,i), && i = 3,5,7,\ldots,m-1,\\ r_1(i) &= r_1(i) - r_1(i+2), && i = 1,3,5,\ldots,m-3,\\ r_1(14) &= r_1(14)U_1^1(14,14),\\ r_1(13) &= r_1(13) - r_1(14). \end{aligned}$$

7.2 Solution of $M^2\boldsymbol{u}^2 = \boldsymbol{r}^2$

Firstly, the elimination procedure of subsection 5.1 is applied to M^2 of (16) to construct 12×12 lower bidiagonal matrices L_k^2, $k = 3,\ldots,11$, of the form (23) defined by the multipliers

$$\tau_i = \frac{i+2-k}{\mu_2(k)} = -L_k^2(i,i), \qquad i = k+1, k+2, \ldots, 12, \tag{41}$$

where

$$\mu_2(k) = \begin{cases} M^2(2,k), & k = 1,3,5,\ldots,11,\\ M^2(3,k), & k = 2,4,6,\ldots,10. \end{cases}$$

Left multiplying M^2 by L_k^2, $k = 3,\ldots,11$, produces the upper triangular matrix $L^2M^2 = L_{11}^2 \cdots L_3^2 M^2$ of the form (25), where $m_\ell = 12$.

Secondly, we construct the upper tridiagonal matrix U_1^2 which transforms the matrix M^2 into the matrix of first order divided differences $M^2U_1^2$ of the odd-numbered columns of M^2 where the divisors are taken from the second row of M^2.

The diagonal entries of U_1^2 are ones, except for

$$U_1^2(3,3) = \frac{1}{\eta_2}, \quad U_1^2(5,5) = \frac{1}{\eta_3 - \eta_2}, \quad U_1^2(7,7) = \frac{1}{\eta_4 - \eta_3},$$
$$U_1^2(9,9) = \frac{1}{\eta_5 - \eta_4}, \quad U_1^2(11,11) = \frac{1}{\eta_6 - \eta_5},$$

and the non-diagonal entries are zeros, except for

$$U_1^2(i-2,i) = -U_1^2(i,i), \qquad i = 3,5,7,\ldots,11.$$

The first two rows of $M^2U_1^2$ are as in equation (26).

Thirdly, the elimination procedure of subsection 5.3 is used to construct 12×12 upper bidiagonal matrices U_k^2, $k = 2,\ldots,12$, of the form (28) defined by the divisors

$$\sigma_i = \frac{k}{\mu_2(i) - \mu_2(i-k)} = U_k^2(i,i), \qquad i = k+1, k+2, \ldots, 12. \tag{42}$$

Right multiplying L^2M^2 by U_k^2, $k = 1,\ldots,11$, produces the diagonal matrix

$$D^2 = L_{11}^2 L_{10}^2 \cdots L_3^2 M^2 U_1^2 U_2^2 \cdots U_{11}^2,$$

where

$$D^2(i,i) = 1, \qquad i = 1,2,3,$$

and

$$D^2(i,i) = \frac{(i-1)!}{2[-\mu_2(3)]\,[-\mu_2(4)]\cdots[-\mu_2(i-1)]}, \qquad i = 4,5,\ldots,12.$$

Finally, M^2 is decomposed into the product of elementary matrices:

$$M^2 = \left(L_{11}^2 L_{10}^2 \cdots L_3^2\right)^{-1} D^2 \left(U_1^2 U_2^2 \cdots U_{11}^2\right)^{-1},$$

and the solution of system (15) is

$$\boldsymbol{u}^2 = U_1^2 U_2^2 \cdots U_{11}^2 (D^2)^{-1} L_{11}^2 L_{10}^2 \cdots L_3^2\, \boldsymbol{r}^2, \tag{43}$$

where fast computation goes from right to left.

Process 5. Procedure (43) is implemented in the following steps:

- Algorithm 4 overwrites $\boldsymbol{r}^2 = r_2(1:m)$ with $U_2^2 \cdots U_{11}^2 (D^2)^{-1} L_{11}^2 L_{10}^2 \cdots L_3^2\, \boldsymbol{r}^2$ in $O(m^2)$ operations, where $m = 12$. The input is $M = M^2$; $m = 12$; $\boldsymbol{r} = \boldsymbol{r}^2$; $L_k = L_k^2$, $k = 3,4,\ldots,m-1$; $U_k = U_k^2$, $k = 2,\ldots,m-1$; and $D = D^2$.
- The following iteration overwrites $\boldsymbol{r}^2 = r_2(1:m)$ with $U_1^2\boldsymbol{r}^2$:

$$r_2(i) = r_2(i)U_1^2(i,i), \qquad i = 3,5,7,\ldots,m-1,$$
$$r_2(i) = r_2(i) - r_2(i+2), \qquad i = 1,3,5,\ldots,m-3.$$

7.3 Solution of $M^3\boldsymbol{u}^3 = \boldsymbol{r}^3$

Firstly, the elimination procedure of subsection 5.1 is applied to M^3 of (18) to construct 13×13 lower bidiagonal matrices L_k^3, $k = 3,\ldots,12$, of the form (23) defined by the multipliers

$$\tau_i = \frac{i+2-k}{\mu_3(k)} = -L_k^3(i,i), \qquad i = k+1, k+2, \ldots, 13, \tag{44}$$

where

$$\mu_3(k) = \begin{cases} M^3(2,k), & k = 1,3,5,\ldots,13, \\ M^3(3,k), & k = 2,4,6,\ldots,12. \end{cases} \tag{45}$$

Left multiplying M^3 by L_k^3, $k = 3,\ldots,13$, produces the upper triangular matrix $L^3M^3 = L_{12}^3 \cdots L_3^3 M^3$ of the form (25), where $m_\ell = 13$.

Secondly, we construct the upper tridiagonal matrix U_1^3 which transforms the matrix M^3 into the matrix of first order divided differences $M^3U_1^3$ of the odd-numbered columns of M^3 where the divisors are taken from the second row of M^3.

The diagonal entries of U_1^3 are ones, except for

$$U_1^3(3,3) = \frac{1}{\eta_2}, \quad U_1^3(5,5) = \frac{1}{\eta_3 - \eta_2}, \quad U_1^3(7,7) = \frac{1}{\eta_4 - \eta_3},$$
$$U_1^3(9,9) = \frac{1}{\eta_5 - \eta_4}, \quad U_1^3(11,11) = \frac{1}{\eta_6 - \eta_5}, \quad U_1^3(13,13) = \frac{1}{c_2 - \eta_6},$$

and the non-diagonal entries are zeros, except for

$$U_1^3(i-2,i) = -U_1^3(i,i), \qquad i = 3,5,7,\ldots,13.$$

The first two rows of $M^3U_1^3$ are as in equation (26).

Thirdly, the elimination procedure of subsection 5.3 is used to construct 13×13 upper bidiagonal matrices U_k^3, $k = 2,\ldots,12$, of the form (28) defined by the divisors

$$\sigma_i = \frac{k}{\mu_3(i) - \mu_3(i-k)} = U_k^3(i,i), \qquad i = k+1, k+2, \ldots, 13. \tag{46}$$

Right multiplying L^3M^3 by U_k^3, $k = 1,\ldots,12$, produces the diagonal matrix

$$D^3 = L_{12}^3 L_{11}^3 \cdots L_3^3 M^3 U_1^3 U_2^3 \cdots U_{12}^3,$$

where

$$D^3(i,i) = 1, \qquad i = 1,2,3,$$

and

$$D^3(i,i) = \frac{(i-1)!}{2[-\mu_3(3)]\,[-\mu_3(4)]\cdots[-\mu_3(i-1)]}, \qquad i = 4,5,\ldots,13.$$

Lastly, we have the following factorization of M^3 into elementary matrices:

$$M^3 = \left(L_{12}^3 L_{11}^3 \cdots L_3^3\right)^{-1} D^3 \left(U_1^3 U_2^3 \cdots U_{12}^3\right)^{-1},$$

and the solution of system (17) is

$$\boldsymbol{u}^3 = U_1^3 U_2^3 \cdots U_{12}^3 (D^3)^{-1} L_{12}^3 L_{11}^3 \cdots L_3^3 \boldsymbol{r}^3, \tag{47}$$

where fast computation goes from right to left.

Process 6. Procedure (47) is implemented in the following steps:

- Algorithm 4 overwrites $\boldsymbol{r}^3 = r_3(1:m)$ with $U_2^3 \cdots U_{12}^3 (D^3)^{-1} L_{12}^3 L_{11}^3 \cdots L_3^3 \boldsymbol{r}^3$ in $O(m^2)$ operations, where $m = 13$. The input is $M = M^3$; $m = 13$; $\boldsymbol{r} = \boldsymbol{r}^3$; $L_k = L_k^3$, $k = 3,4,\ldots,m-1$; $U_k = U_k^3$, $k = 2,\ldots,m-1$; and $D = D^3$.
- The following iteration overwrites $\boldsymbol{r}^3 = r_3(1:m)$ with $U_1^3\boldsymbol{r}^3$:

$$\begin{aligned} r_3(i) &= r_3(i)U_1^3(i,i), && i = 3,5,7,\ldots,m, \\ r_3(i) &= r_3(i) - r_3(i+2), && i = 1,3,5,\ldots,m-2. \end{aligned}$$

7.4 Solution of $M^4\boldsymbol{u}^4 = \boldsymbol{r}^4$

The solution procedure for the step control predictor P_4 differs slightly from the the procedure for IF, P_2 and P_3, and requires a treatment of its own.

Firstly, the elimination procedure of subsection 5.1 is applied to M^4 of (20) to construct 12×12 lower bidiagonal matrices L_k^4, $k = 3,\ldots,11$,

$$L_k^4 = \begin{bmatrix} I_{k-1} & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 1 & -\tau_{k+1} & & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & -\tau_{12} \end{bmatrix}, \tag{48}$$

defined by the multipliers

$$\tau_i = \frac{i+2-k}{\mu_4(k)} = -L_k^4(i,i), \qquad i = k+1, k+2, \ldots, 12, \tag{49}$$

where

$$\mu_4(k) = \begin{cases} M^4(2,k), & k = 1,3,5,\ldots,11, \\ M^4(3,k), & k = 2,4,6,\ldots,10. \end{cases} \tag{50}$$

Left multiplying M^4 by L_k^4, $k = 3,\ldots,11$, produces the upper triangular matrix $L^4M^4 = L_{11}^4 \cdots L_3^4 M^4$ of the form (25), where $m_\ell = 12$.

Secondly, we construct the upper tridiagonal matrix U_1^4 which transforms the matrix M^4 into the matrix of first order divided differences $M^4U_1^4$ of the odd-numbered columns of M^4 where the divisors are taken from the second row of M^4.

The diagonal entries of U_1^4 are ones, except for

$$U_1^4(3,3) = \frac{1}{\eta_2}, \quad U_1^4(5,5) = \frac{1}{\eta_3 - \eta_2}, \quad U_1^4(7,7) = \frac{1}{\eta_4 - \eta_3},$$
$$U_1^4(9,9) = \frac{1}{\eta_5 - \eta_4}, \quad U_1^4(11,11) = \frac{1}{c_3 - \eta_5},$$

and the non-diagonal entries are zeros, except for

$$U_1^4(i-2,i) = -U_1^4(i,i), \qquad i = 3,5,7,\ldots,11.$$

The first two rows of $M^4U_1^4$ are as in equation (26).

Thirdly, the elimination procedure of subsection 5.4 is used to construct 12×12 upper bidiagonal matrices U_k^4, $k = 2,\ldots,10$, defined by the divisors

$$\sigma_i = \frac{k}{\mu_4(i) - \mu_4(i-k)} = U_k^4(i,i), \qquad i = k+1, k+2, \ldots, 11, \tag{51}$$

Right multiplying L^4M^4 by U_k^4, $k = 1,\ldots,10$, produces a diagonal+last-column matrix W^4 of the form (36).

Next, a factored Gaussian elimination by means of $L_{13}^4 L_{12}^4$ diagonalizes W^4 as follows. First, $W^4(12,12)$ is set to 1 by the diagonal matrix L_{12}^4:

$$L_{12}^4(i,i) = 1, \qquad i = 1,\ldots,11,$$
$$L_{12}^4(12,12) = 1/W^4(12,12).$$

Then, the non-diagonal entries in the last column of $L_{12}^4 W^4$ are zeroed by the unit diagonal+last-column matrix L_{13}^4 whose last column has top 11 entries

$$L_{13}^4(1:11,12) = -(L_{12}^4 W^4)(1:11,12).$$

This procedure transforms M^4 into the diagonal matrix

$$D^4 = L_{13}^4 L_{12}^4 \cdots L_3^4 M^4 U_1^4 U_2^4 \cdots U_{10}^4,$$

where

$$D^4(i,i) = 1, \qquad i = 1,2,3,12,$$

and

$$D^4(i,i) = \frac{(i-1)!}{2[-\mu_4(3)]\,[-\mu_4(4)]\cdots[-\mu_4(i-1)]}, \qquad i = 4,5,\ldots,11.$$

Lastly, M^4 is decomposed into the product of elementary matrices,

$$M^4 = \left(L_{13}^4 L_{12}^4 \cdots L_3^4\right)^{-1} D^4 \left(U_1^4 U_2^4 \cdots U_{10}^4\right)^{-1},$$

and the solution of system (19) is

$$\boldsymbol{u}^4 = U_1^4 U_2^4 \cdots U_{10}^4 (D^4)^{-1} L_{13}^4 L_{12}^4 \cdots L_3^4\, \boldsymbol{r}^4, \tag{52}$$

where fast computation goes from right to left.

Procedure (52) is implemented in Algorithm 5 in Appendix A in $O(m^2)$ operations, where $m = 12$. The input is $M = M^4$; $m = 12$; $\boldsymbol{r} = \boldsymbol{r}^4$; $L_k = L_k^4$, $k = 3,4\ldots,13$; $U_k = U_k^4$, $k = 1,2,\ldots,10$; and $D = D^4$. The output is $\boldsymbol{u} = \boldsymbol{u}^4$.

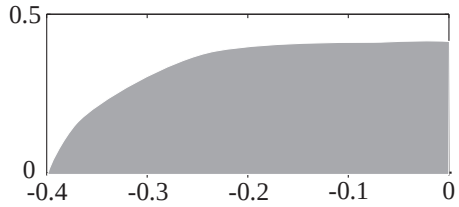

Figure 1: Upper-half of region of absolute stability, R, of HBO14 ($\alpha = -0.39$).

Remark 1. Formulae (2)–(5) can be put in matrix form. For instance, (4) can be written as

$$y_{n+1} = y_n + F^1 \boldsymbol{u}^1.$$

where

$$F^1 = h_{n+1}\Big[f_n, h_{n+1}f'_n, f_{n-1}, h_{n+1}f'_{n-1}, f_{n-2}, h_{n+1}f'_{n-2}, f_{n-3}, h_{n+1}f'_{n-3}, \\ f_{n-4}, h_{n+1}f'_{n-4}, f_{n-5}, h_{n+1}f'_{n-5}, f_{n+c_3}, f_{n+c_2}\Big],$$

and

$$\boldsymbol{u}^1 = [b_{11}, \gamma_{10}, \beta_{11}, \gamma_{11}, \beta_{12}, \gamma_{12}, \beta_{13}, \gamma_{13}, \beta_{14}, \gamma_{14}, \beta_{15}, \gamma_{15}.b_{13}, b_{12}]^T,$$

It is interesting to note the three decomposition forms of the system $F\boldsymbol{u}$:

$$\begin{array}{ll} F(UD^{-1}L\boldsymbol{r}) & \text{(generalized Lagrange interpolation)}, \\ (FUD^{-1})L\boldsymbol{r} & \text{(Krogh's modified divided differences)}, \\ (FUD^{-1}L)\boldsymbol{r} & \text{(Nordsieck's formulation)}. \end{array}$$

The first form is used in this paper, the second form for Vandermonde systems is found in [16], and the third form is found in [20].

8 Region of absolute stability and principal error term

To obtain the region of absolute stability, R, of HBO14 we apply the predictors $\mathrm{P}_2, \mathrm{P}_3$ and the integration formula IF, with constant h, to the linear test equation

$$y' = \lambda y, \qquad y_0 = 1.$$

This gives the difference equation and the corresponding characteristic equation

$$\sum_{j=0}^{6} \gamma_j y_{n+j} = 0, \qquad \sum_{j=0}^{6} \gamma_j r^j = 0, \tag{53}$$

respectively. A complex number λh is in R if the six roots of the characteristic equation satisfy the root condition $|r_s| \le 1$ and the multiple roots satisfy $|r_s| < 1$. The method used to find R is similar to the one used for k-step multistep methods (see [12, pp. 256–257]). The gray region in Fig. 1 is the upper-half of region R, which is symmetric with respect to the real axis. The interval of absolute stability is $(\alpha, 0) = (-0.39, 0)$. It is seen that HBO14 has a larger interval of absolute stability than the Adams–Bashforth–Moulton methods of orders 10 to 13 in PECE mode [22, pp. 139–140].

The principal error term of HBO14 is

$$\Big[\delta_1 \{f^{14}\} + 14\delta_2 \{\{f^{12}\}f\} + \delta_3 \{{}_2f^{13}\}_2 + \delta_4 \{{}_3f^{12}\}_3\Big] h^{15}, \tag{54}$$

where $\{f^{14}\}, \{\{f^{12}\}f\}, \{{}_2f^{13}\}_2, \{{}_3f^{12}\}_3$ are elementary differentials defined in [5], [17] and [11].

The principal local truncation coefficients of the principal error term are

$$[\delta_1, 14\delta_2, \delta_3, \delta_4] = \left[-\frac{1}{120066847}, -\frac{14}{3212565}, \frac{1}{8337285}, \frac{1}{229499}\right]$$

with ℓ_2-norm 6.16e-06, which is quite small.

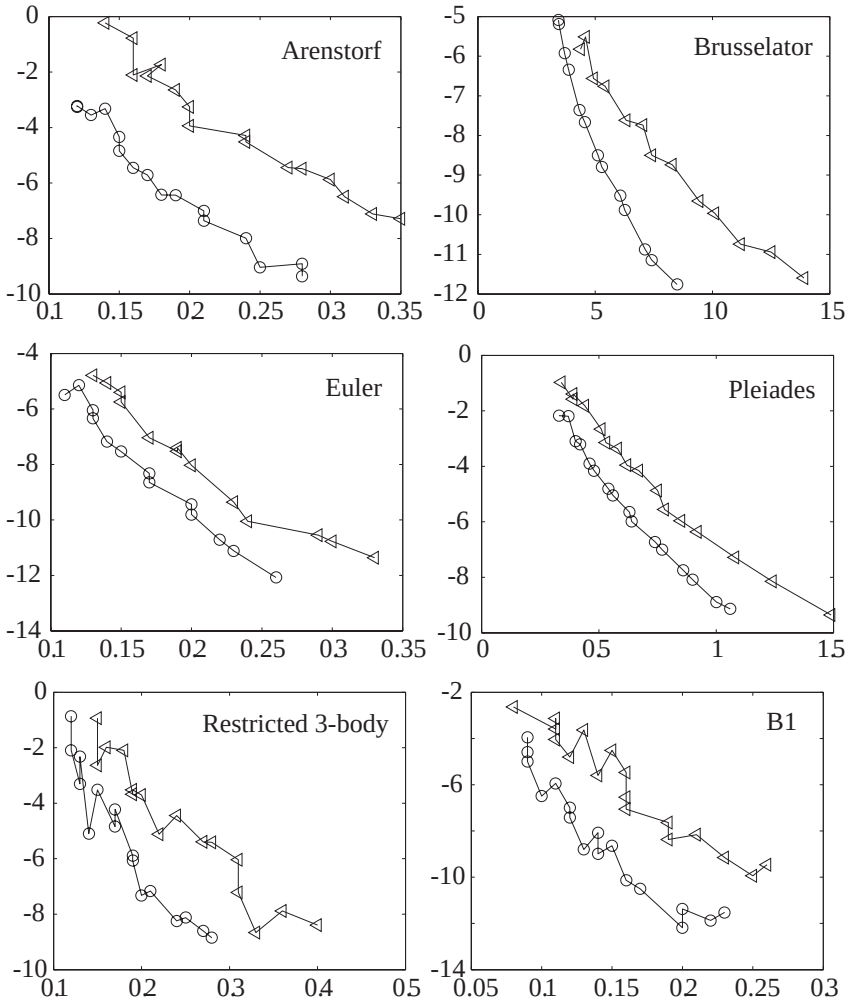

Figure 2: CPU (horizontal axis) versus $\log_{10}(|\mathrm{MGE}|)$ (vertical axis) for Arenstorf, Brusselator, Euler, Pleiades, restricted three-body and B1. HBO14 $\circ$ and `ode113` $\triangleright$.

9 Controlling stepsize

The estimate $y_n - \widetilde{y}_n$ and the current step h_n are used to calculate the next stepsize h_{n+1} by means of formula [13]

$$h_{n+1} = \min\left\{h_{\max}, \beta\, h_n \left[\frac{\text{tolerance}}{|y_n - \widetilde{y}_n|}\right]^{1/\kappa}, 4\, h_n\right\}, \tag{55}$$

with $\kappa = 13$ and safety factor $\beta = 0.81$.

10 Numerical results

The numerical performance of HBO14 and MATLAB's `ode113` is compared on the following problems: Arenstorf's orbits [1], the Brusselator and the Pleiades [11, pp. 244–249], Euler's equation and the restricted three-body problem [22, pp. 232–259], and the following nonstiff DETEST problems: growth problem of two conflicting populations B1, two-body problems D1–D5, and Van der Pol's equation E2 with $\epsilon = 1$ [13].

The six starting values for HBO14 were obtained by DP(5,4)7FM (see [7]) with initial step size, h_1, chosen by a method similar to steps (a) and (b) of [11, p. 169].

Computations were performed on a Mac with a dual 2.5 GHz PowerPC G5 and 4 GB DDR SSRAM running under Mac OS X Version 10.4.2 and MATLAB Version 7.0.4.352 (R14) Service Pack 2. Algorithms 4 and 5 were written in C and made into system-dependent MATLAB `mex` files for speed.

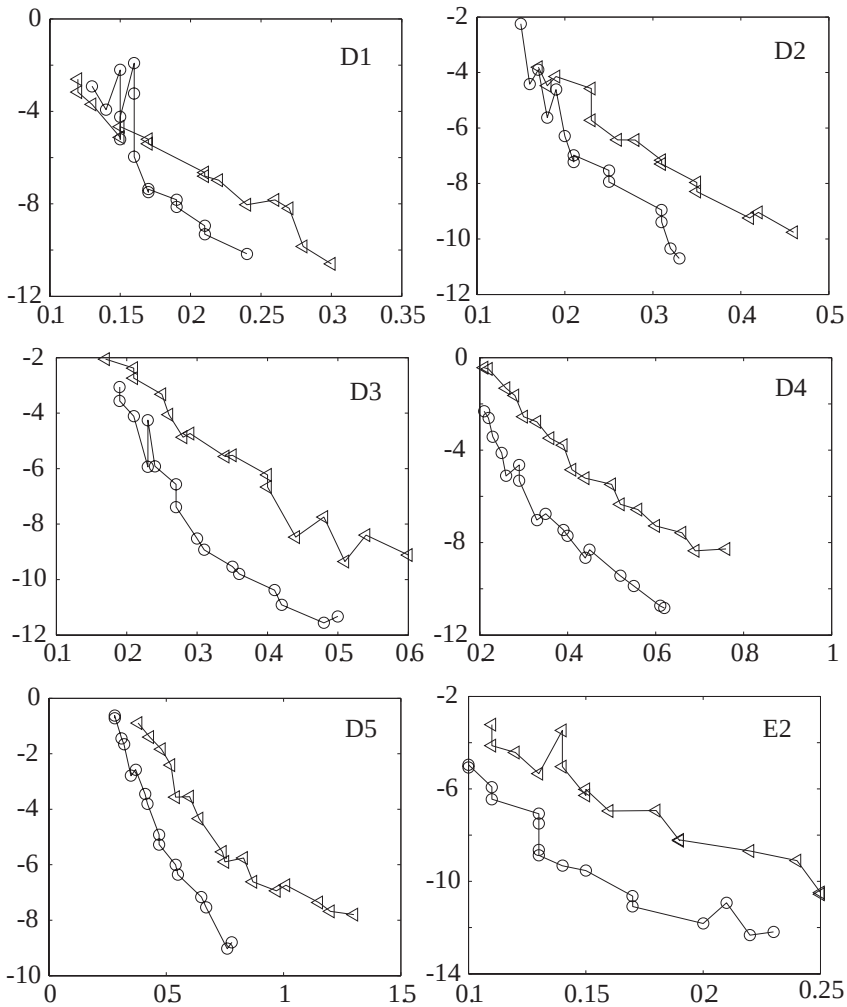

Figure 3: CPU (horizontal axis) versus $\log_{10}(|\mathrm{MGE}|)$ (vertical axis) for for D1 to D5 and E2. HBO14 $\circ$ and `ode113` $\triangleright$.

Table 1: CPU percentage efficiency gain, CPU PEG, of HBO14 over `ode113` for the listed problems.

Problem	CPU PEG	Problem	CPU PEG
Arenstorf	62%	D1	20%
Brusselator	39%	D2	34%
Euler	25%	D3	51%
Pleiades	36%	D4	63%
Restricted 3-body	48%	D5	69%
B1	49%	E2	43%

10.1 CPU against maximum global error

The maximum global error, MGE, was obtained from the errors corresponding to each integration step. These errors were calculated from the numerical value y_{n+1} of HBO14 and the "exact solution" obtained by MATLAB's `ode113` with stringent tolerance 5×10^{-14}.

The cpu time was obtained from the curves which fit, in a least-square sense, the data $(\log_{10}(|\mathrm{MGE}|), \log_{10}(\mathrm{CPU}))$ using, say, MATLAB's `polyfit`.

In Figs. 2 and 3, CPU (horizontal axis) is plotted versus $\log_{10}(|\mathrm{MGE}|)$ (vertical axis) for the problems considered in this paper.

10.2 CPU percentage efficiency gain of HBO14 against `ode113`

The cpu percentage efficiency gain (CPU PEG) is defined by formula (cf. Sharp [24]),

$$(\text{CPU PEG})_i = 100\left(\frac{\sum_j \mathrm{CPU}_{2,ij}}{\sum_j \mathrm{CPU}_{1,ij}} - 1\right), \tag{56}$$

where $\mathrm{CPU}_{1,ij}$ and $\mathrm{CPU}_{2,ij}$ are the CPU of methods 1 and 2, respectively, associated with problem i, and $j = -\log_{10}(|\mathrm{MGE}|)$.

The CPU PEG for the problems considered in this paper is listed in Table 1.

It is seen from Figs. 2 and 3 and Table 1 that HBO14 compares favorably with MATLAB's `ode113` on the basis of CPU versus MGE.

10.3 Maximum global error against tolerance

Table 2 lists several numerical results related to the step control for the problems in hand on the time interval $[0, t_f]$ with set tolerance (TOL), namely, cpu time (CPU), number of function evaluations (NFE), number of failed attempts (REJ) and maximum global error (MGE) of HBO14 and `ode113`. For HBO14, NFE splits into 75% and 25% between f and f', respectively.

It is seen from Table 2 that HBO14 has smaller maximum global error than `ode113` for each given tolerance.

11 Conclusion

A fast variable step 3-stage Hermite–Birkhoff–Obrechkoff method of order 14 with 5 backsteps was constructed by solving Vandermonde-type systems satisfying multistep and Runge–Kutta type order conditions. The stepsize is controlled by a local error estimator. This method, in its vectorized Lagrange form, was tested on the Brusselator, Euler's equation, Arenstorf's orbits, the restricted three-body problem, the Pleiades, and the following nonstiff DETEST problems: two-body problems of class D, the growth problem of two conflicting populations of class B and Van der Pol's equation of class E. It is seen from subsections 10.1 and 10.2 that the new method has lower global error, uses less cpu time and requires fewer function evaluations than the highly optimized MATLAB's `ode113` for the problems in hand.

Acknowledgment

The anonymous referee is deeply thanked for many constructive comments and corrections. Thanks are also due to Philip W. Sharp for helpful discussions and observations.

Table 2: For each problem, time interval $[0, t_f]$ and tolerance (TOL), the table lists cpu time (CPU), number of function evaluations (NFE), number of failed attempts (REJ) and maximum global error (MGE) in corresponding left and right columns for HBO14 and `ode113`, respectively.

		HBO14 and `ode113`							
Problem	TOL	CPU		NFE		REJ		MGE	
BRUS	10^{-04}	3.44	2.57	922	736	1	5	8.15e-06	2.64e-04
$t_f = 7.5$	10^{-10}	8.49	9.84	2498	2419	0	4	1.75e-12	2.22e-10
EULER	10^{-04}	0.10	0.09	602	485	12	4	3.65e-05	2.89e-03
$t_f \approx 52.15$	10^{-10}	0.22	0.24	1646	1637	0	4	1.95e-11	8.92e-11
AREN	10^{-04}	0.12	0.15	574	491	8	20	5.62e-04	2.22e-0
$t_f \approx 17.06$	10^{-10}	0.24	0.27	1562	1637	1	18	1.03e-08	3.29e-06
R-3-body	10^{-04}	0.13	0.11	590	513	15	20	4.98e-04	4.15e 00
$t_f \approx 6.19$	10^{-10}	0.25	0.27	1574	1632	1	21	7.56e-09	4.03e-06
PLEI	10^{-04}	0.33	0.31	870	765	17	22	6.73e-03	4.99e-01
$t_f = 3$	10^{-10}	0.86	0.85	2366	2408	0	23	1.79e-08	1.09e-06
B1	10^{-04}	0.10	0.09	450	377	8	12	5.24e-05	5.26e-03
$t_f = 20$	10^{-10}	0.16	0.19	1134	1208	4	12	7.36e-11	2.30e-08
E2	10^{-04}	0.10	0.11	526	448	8	15	8.28e-06	6.02e-04
$t_f = 20$	10^{-10}	0.21	0.22	1470	1437	16	10	1.18e-11	2.07e-09
D1	10^{-04}	0.16	0.10	750	450	0	7	1.22e-02	2.11e-02
$t_f = 16\pi$	10^{-10}	0.21	0.22	1318	1480	0	3	1.13e-09	1.06e-07
D2	10^{-04}	0.14	0.31	782	577	5	0	1.66e-02	4.97e-02
$t_f = 16\pi$	10^{-10}	0.31	0.31	1914	2035	0	12	1.10e-09	6.89e-08
D3	10^{-04}	0.17	0.14	958	785	17	26	2.29e-03	4.32e-01
$t_f = 16\pi$	10^{-10}	0.35	0.40	2534	2688	0	27	2.95e-10	5.98e-07
D4	10^{-04}	0.20	0.17	1230	1102	20	41	1.70e-02	2.81e 00
$t_f = 16\pi$	10^{-10}	0.45	0.50	3338	3463	0	36	4.94e-09	3.29e-06
D5	10^{-04}	0.28	0.26	1946	1662	42	77	1.90e-01	4.56e 00
$t_f = 16\pi$	10^{-10}	0.65	0.74	4966	5099	0	56	6.80e-08	2.91e-06

A Algorithms

Algorithm 1. ***This algorithm constructs lower bidiagonal matrices*** L_k ***as functions of*** c_2***,*** c_3 ***and*** η_j***,*** $j = 2:6$***.***

For $k = 3 : m-1$, do the following iteration:
For $i = m : -1 : k+1$, do the following two steps:

Step (1) $L_k(i,i) = -A(i-1,k)/A(i,k)$.

Step (2) For $j = k : m$, compute:

$$A(i,j) = A(i-1,j) + A(i,j)L_k(i,i).$$

Algorithm 2. ***This algorithm constructs upper bidiagonal matrices*** U_k ***(applied to*** $\mathrm{IF}, \mathrm{P}_2, \mathrm{P}_3$***) as functions of*** c_2***,*** c_3 ***and*** η_j***,*** $j = 2:6$***.***

For $k = 2 : m-1$, do the following iteration:
For $j = m : -1 : k+1$, do the following two steps:

Step (1) $U_k(j,j) = 1/[A(k+1,j) - A(k+1,j-1)]$.

Step (2) for $i = k : j$, compute

$$A(i,j) = (A(i,j) - A(i,j-1))U_k(j,j).$$

Algorithm 3. ***This algorithm constructs upper bidiagonal matrices*** U_k ***(applied to*** P_4***) as functions of*** c_2***,*** c_3 ***and*** η_j***,*** $j = 2:6$***.***

For $k = 2 : m-2$, do the following iteration:
For $j = m-1 : -1 : k+1$, do the following two steps:

Step (1) $U_k(j,j) = 1/[A(k+1,j) - A(k+1,j-1)]$.

Step (2) for $i = k : j$, compute

$$A(i,j) = (A(i,j) - A(i,j-1))U_k(j,j).$$

Algorithm 4. ***This algorithm overwrites*** $\boldsymbol{r} = r(1:m)$ ***with*** $U_2 \cdots U_{m-1}(D^1)^{-1}L_{m-1}L_{m-2}\cdots L_3\,\boldsymbol{r}$ ***in*** $O(m^2)$ ***operations for IF,*** $\mathbf{P}_2$ ***and*** $\mathbf{P}_3$.

Given $[\eta_2, \eta_3, \ldots, \eta_6]$ and $\boldsymbol{r} = r(1:m)$, the following algorithm overwrites $\boldsymbol{r}$ with $U_2 \cdots U_{m-1}(D^1)^{-1}L_{m-1}L_{m-2}\cdots L_3\,\boldsymbol{r}$.

Step (1) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $L_{m-1}L_{m-2}\cdots L_3\boldsymbol{r}$:
for $k = 3 : m-1$, compute

$$r(i) = r(i-1) + r(i)L_k(i,i), \qquad i = m : -1 : k+1.$$

Step (2) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_2U_3\cdots U_{m-1}D^{-1}\boldsymbol{r}$:

$$r(i) = r(i)/D(i,i), \qquad i = 1 : m.$$

For $k = m-1 : -1 : 2$, compute

$$\begin{aligned} r(i) &= r(i)U_k(i,i), & i &= k+1 : m, \\ r(i) &= r(i) - r(i+1), & i &= k : m-1. \end{aligned}$$

Algorithm 5. ***This algorithm solves the*** P_4 ***system*** P_4 ***in*** $O(m^2)$ ***operations.***

Given $[\eta_2, \eta_3, \ldots, \eta_6]$ and $\boldsymbol{r} = r(1:m)$, the following algorithm overwrites $\boldsymbol{r}$ with the solution $\boldsymbol{u} = u(1:m)$ of the system $M\boldsymbol{u} = \boldsymbol{r}$.

Step (1) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $L_{m-1}L_{m-2}\cdots L_3\boldsymbol{r}$:
for $k = 3 : m-1$, compute

$$r(i) = r(i-1) + r(i)L_k(i,i), \qquad i = m : -1 : k+1.$$

Step (2) First put

$$G(1:m) = M(1:m, m).$$

We obtain the coefficients of the last two row transformations, L_m and L_{m+1}, by means of the recursion: for $k = 3 : m-1$, compute

$$G(i) = G(i-1) + G(i)L_k(i,i), \qquad i = m : -1 : k+1.$$

Step (3) The following computation overwrites the newly obtained $\boldsymbol{r}$ with $L_{m+1}L_m\boldsymbol{r}$:

$$r(m) = r(m)/G(m),$$

and for $k = m-1 : -1 : 1$, compute

$$r(k) = r(k) - G(k)r(m).$$

Step (4) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_2U_3\cdots U_{m-2}D^{-1}\boldsymbol{r}$:

$$r(i) = r(i)/D(i,i), \qquad i = 1 : m.$$

For $k = m-2 : -1 : 2$, compute

$$\begin{aligned} r(i) &= r(i)U_k(i,i), & i &= k+1 : m-1, \\ r(i) &= r(i) - r(i+1), & i &= k : m-2. \end{aligned}$$

Step (5) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_1\boldsymbol{r}$:

$$\begin{aligned} r(i) &= r(i)U_1(i,i), & i &= 3 : 2 : m-1, \\ r(i) &= r(i) - r(i+2), & i &= 1 : 2 : m-3. \end{aligned}$$

Algorithms 4 and 5 use minimum storage since the solution is obtained by successively transforming the right-hand side into the solution vector. This is an advantage compared to generating $m \times m$ triangular matrices as an intermediate result at each integration step.

B Matlab programming

Algorithm 4 which solves systems IF, P_2 and P_3 was programmed in C and compiled by the MATLAB `mex` command into `mex` files, say, `IF.macmex`, `P2.macmex` and, `P3.macmex`.

Algorithm 5 which solves the P_4 system was programmed in C and compiled by the MATLAB `mex` command into a `mex` file, say, `SCP.macmex`.

At runtime, the data of differential equations were input. Then, `IF.macmex`, `P2.macmex`, `P3.macmex` and `SCP.macmex` were called and run to calculate the values of the coefficients of IF, P_2, P_3 and P_4 at each integration step until completion of the integration. CPU time and number of evaluations of $f(x, y)$ and $f'(x, y)$ for the runtime of Algorithms 4 and 5 were recorded.

The option MGE can be run.

MATLAB's `ode113` can be run with appropriate tolerance for comparison with HBO14.

The elementary matrix functions L_k^ℓ and U_k^ℓ, $\ell = 1, 2, 3, 4$, are constructed by Algorithms 1 and 2 as functions of η_j, for $j = 2, 3, \ldots, 6$. These algorithms are not needed at runtime since these matrix functions are already implemented in the MATLAB `mex` files `IF.macmex`, `P2.macmex`, `P3.macmex` and `SCP.macmex`.

References

[1] R. F. Arenstorf, *Periodic solutions of the restricted three-body problem representing analytic continuations of Keplerian elliptic motions*, Amer. J. Math., **LXXXV** (1963), 27–35.

[2] R. Ashino, M. Nagase, and R. Vaillancourt, *Behind and beyond the* MATLAB *ODE Suite*, Comput. & Math. with Applics., **40** (2000), 491–512.

[3] A. Björck and T. Elfving, *Algorithms for confluent Vandermonde systems*, Numer. Math., **21** (1973), 130–137.

[4] A. Björck and V. Pereyra, *Solution of Vandermonde systems of equations*, Math. Comp., **24** (1970), 893–903.

[5] J. C. Butcher, *Coefficients for the study of Runge–Kutta integration processes*, J. Aust. Math. Soc., **3** (1963), 185–201.

[6] J. C. Butcher, *A modified multistep method for the numerical integration of ordinary differential equations*, J. Assoc. Comput. Mach., **12** (1965), 124–135.

[7] J. R. Dormand and P. J. Prince, *A reconsideration of some embedded Runge–Kutta formulae*, J. Comput. Appl. Math., **15** (1986), 203–211.

[8] G. Galimberti and V. Pereyra, *Solving confluent Vandermonde systems of Hermite type*, Numer. Math., **18** (1971), 44–60.

[9] C. W. Gear, *The numerical integration of ordinary differential equations*, Math. Comp., **21** (1967), 146–156.

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

[11] E. Hairer, S. P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, Section III.8, Springer-Verlag, Berlin, 1993.

[12] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Springer-Verlag, Berlin, 1991.

[13] T. E. Hull, W. H. Enright, B. M. Fellen, and A. E. Sedgwick, *Comparing numerical methods for ordinary differential equations*, SIAM J. Numer. Anal., **9** (1972), 603–637.

[14] T. Y. Huang and K. Innanen, *A survey of multiderivative multistep integrators*, Astronomical J., **112**(3) (1996), 1254–1262.

[15] F. T. Krogh, *VODQ/SVDQ/DVDQ—variable order integrators for the numerical solution of ordinary differential equations*, TU Doc. No. CP-2308, NPO-11643, May 1969, Jet Propulsion Laboratory, Pasadena, CA.

[16] F. T. Krogh, *Changing stepsize in the integration of differential equations using modified divided differences*, in Proc. Conf. on the Numerical Solution of Ordinary Differential Equations, University of Texas at Austin 1972 (Ed. D.G. Bettis), Lecture Notes in Mathematics No. 362, Springer-Verlag, Berlin, 22–71, 1974.

[17] J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, Ch. 5, Wiley, London, 1973.

[18] W. E. Milne, *A note on the numerical integration of differential equations*, J. Res. Nat. Bur. Standards, **43** (1949), 537–542.

[19] T. Nguyen-Ba and R. Vaillancourt, *Hermite–Birkhoff differential equation solvers*, Scientific Proceedings of Riga Technical University, 5-th series: Computer Science, 46-th thematic issue, **21** (2004), 47–64.

[20] A. Nordsieck, *On numerical integration of ordinary differential equations*, Math. Comp., **16** (1962), 22–49.

[21] N. Obrechkoff, *Neue Quadraturformeln*, Abh. Preuss. Akad. Wiss. Math. Nat. Kl., No. 4, (1940), 1–20.

[22] L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, Freeman, San Francisco, CA, 1975.

[23] L. F. Shampine and M. W. Reichelt, *The Matlab ODE suite*, SIAM J. Sc. Comp., **18**(1) (1997), 1–22.

[24] P. W. Sharp, *Numerical comparison of explicit Runge–Kutta pairs of orders four through eight*, Trans. on Mathematical Software, **17** (1991), 387–409.