Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 5, No. 4, 2017, pp. 324-347

On second derivative 3-stage Hermite–Birkhoff–Obrechkoff methods for stiff ODEs: A-stable up to order 10 with variable stepsize

Truong Nguyen-Ba*
Department of Mathematics and Statistics,
University of Ottawa, Ottawa, Ontario, Canada K1N 6N5.
E-mail: trnguyen@uottawa.ca

Thierry Giordano
Department of Mathematics and Statistics,
University of Ottawa, Ottawa, Ontario, Canada K1N 6N5.
E-mail: giordano@uottawa.ca

Rémi Vaillancourt
Department of Mathematics and Statistics,
University of Ottawa, Ottawa, Ontario, Canada K1N 6N5.
E-mail: remi@uottawa.ca

Abstract Variable-step (VS) second derivative k-step 3-stage Hermite–Birkhoff–Obrechkoff (HBO) methods of order $p = (k+3)$, denoted by HBO(p) are constructed as a combination of linear k-step methods of order $(p-2)$ and a second derivative two-step diagonally implicit 3-stage Hermite–Birkhoff method of order 5 (DIHB5) for solving stiff ordinary differential equations. The main reason for considering this class of formulae is to obtain a set of k-step methods which are L-stable and are suitable for the integration of stiff differential systems whose Jacobians have some large eigenvalues lying close to the imaginary axis with negative real part. The approach, described in the present paper, allows us to develop L-stable k-step methods of order up to 10. Selected HBO(p) of order p, $p = 9, 10$, compare favorably with existing Cash L-stable second derivative extended backward differentiation formulae, SDEBDF(p), $p = 7, 8$ in solving problems often used to test stiff ODE solvers.

Keywords. Hermite–Birkhoff methods, generalized DIRK methods, A-stable, oscillatory stiff DETEST problems, confluent Vandermonde-type systems.

2010 Mathematics Subject Classification. 65L06, 65D05, 65D30.

1. Introduction

For solving stiff ordinary differential equations (ODE),

$$y' = f(t, y), \qquad y(t_0) = y_0, \qquad \text{where} \quad ' = \frac{d}{dt} \quad \text{and} \quad y \in \mathbb{R}^n, \tag{1.1}$$

a linear k-step method of order $p-2$ and a second derivative two-step diagonally implicit 3-stage Hermite–Birkhoff method of order 5 (DIHB5) are cast into a k-step 3-stage Hermite–Birkhoff–Obrechkoff (HBO) methods of order $p = (k+3)$, denoted

Received: 26 March 2017 ; Accepted: 24 September 2017.
* Corresponding author.

by HBO(p). The method's name was chosen because it uses Hermite–Birkhoff interpolation polynomials, first and second order derivatives of y like Obrechkoff methods [20]. Here, the DIHB5 is defined in Section 2 with $p = 5$ and step number $k = 2$, and has three degree of freedom since the coefficients in (2.4) are free parameters. And, following the approach of Cash [3], the abscissae c_i are allowed to be $0 \leq c_i \leq 2$, $i = 2, 3, 4$.

There is a variety of variable step (VS) methods designed to solve nonstiff and stiff systems of first-order differential equations (ODEs). Gear advocated a quasi-constant step size implementation in DIFSUB [9]. This software works with a constant step size until a change of step size is necessary or clearly advantageous. Then a continuous extension is used to get approximations to the solution at previous points in an equally spaced mesh. This was largely because constant mesh spacing is very helpful when solving stiff problems. Another possibility is fixed leading coefficient, which is seen in Petzold's popular code DASSL [21]. Finally, the actual mesh can be chosen by the code as done in MATLAB's `ode113`. This is the equivalent of a PECE Adams formula in contrast with the Adams–Moulton formula of DIFSUB and DASSL. In this paper, a fully variable step size implementation is used with actual mesh.

A more basic point about the implementation of a method is the choice of the form. The present method uses a generalized Lagrange form and much of the paper is devoted to computing the coefficients efficiently. Remark 6.1 in Subsection 6.2 connects the computation of coefficients for three well known forms: generalized Lagrange form, generalized Newton divided differences form (similar to Krogh's modified divided differences [16]) and Nordsieck form [19].

A brief survey of methods for the numerical integration of (1.1) reveals that there are many advances in the class of generalized linear multistep methods for stiff ODEs, methods like second derivative multistep methods (SDMM) proposed by Enright [5], second derivative extended backward differentiation formulas (SDEBDF) by Cash [3], second derivative BDF methods (SDBDF) by Hairer *et al.* [10], special classes of SDMM introduced by Ismail *et al.* [14], Hojjati *et al.* [12] and Khalsaraei *et al.* [15].

The first modification, introduced by Cash [3], was the SDEBDF in which one "super-future" point has been applied.

The current investigation and the results are offered as potentially useful additions to the contemporary repertory of variable step (VS) L-stable second derivative multistep solvers for stiff differential equations (ODEs). This paper explores an alternative way to improve the order of L-stable second derivative multistep methods.

Forcing a Taylor expansion of the numerical solution of HBO(p) methods 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. This approach allows us to develop L-stable methods of order up to 10, while we know well the difficulty of deriving multistep A-stable formulae of order greater than 2. A-stable linear multistep methods are limited to having maximum order 2 while high order, A-stable Runge-Kutta formulae can be very expensive to implement.

The extra stability is particularly important when integrating stiff differential systems whose Jacobians have some large eigenvalues lying close to the imaginary axis. Stiff oscillatory problems happen often in practice. In particular, they frequently develop when the method of lines technique is applied to a system of partial differential equations (PDE) that have some hyperbolic type of behaviour. Typical examples of such problems are the integro-differential equations describing the stiff beam problem [10], and advection-dominated PDE problems, as described, for example, in [11, 23]. A good description of the difficulties involved in integrating these hyperbolic type equations can be found in [10, pp. 12].

The selected HBO(p), $p = 9, 10$ compare favorably with SDEBDF(p), $p = 7, 8$, [3] in solving problems often used to test highly stable stiff ODE solvers on the basis of number of steps (NS), CPU time (CPU) and the error at the endpoint (EPE) of the interval of integration.

The paper is organized as follows: in Section 2, we introduce new general VS HBO(p) methods of order p. Order conditions of general VS HBO(p) are listed in Section 3. In Section 4, particular variable step HBO(p) are defined by fixing a set of parameters and are represented in terms of Vandermonde-type systems. In Section 5, symbolic elementary matrices are constructed as functions of the parameters of the methods in view of factoring the coefficient matrices of Vandermonde-type systems. Fast solution of Vandermonde-type systems for variable step HBO(p) is constructed in Section 6. Section 7 considers the regions of absolute stability of constant step L-stable HBO(p), $p = 5, 6, \ldots, 10$. Section 8 deals with the step control. In Section 9, we compare the numerical performance of L-stable methods considered in this paper. Appendix A lists the algorithms. Appendix B lists the coefficients of constant step L-stable HBO(p) methods of order $p = 9, 10$.

2. General variable step HBO(p) of order p

Variable step 3-stage HBO methods are constructed by the following formulae to perform integration from t_n to t_{n+1}.

Let h_{n+1} denote the step size. The abscissa vector $[c_1, c_2, c_3, c_4]^T$ defines the off-step points $t_n + c_j h_{n+1}$ with $c_1 = 0$ and $c_4 = 1$. Following the approach of Cash [3], c_i are allowed to be $0 \le c_i \le 2$, $i = 2, 3$.

Let $F_1 = f_n$ and $F_j := f(t_n + c_j h_{n+1}, Y_j)$, $j = 2, 3, 4$, denote the jth stage derivative.

With the initial stage value, $Y_1 = y_n$, HB polynomials are used as implicit predictors P_i to obtain the stage values Y_i to order $p - 2$,

$$
\begin{aligned}
Y_i = {} & h_{n+1} a_{ii} f(t_n + c_i h_{n+1}, Y_i) + h_{n+1}^2 \gamma_{ii} f'(t_n + c_i h_{n+1}, Y_i) \\
& + y_n + h_{n+1} \left[\sum_{j=0}^{p-4} \beta_{ij} f_{n-j} + \sum_{j=2}^{i-1} a_{ij} F_j \right] + h_{n+1}^2 \sum_{j=2}^{i-1} \gamma_{ij} F'_j, \quad i = 2, 3. \qquad (2.1)
\end{aligned}
$$

An HB polynomial is used as implicit integration formula IF to obtain y_{n+1} to order p,

$$\begin{aligned} y_{n+1} &= h_{n+1} b_4 f(t_n + h_{n+1}, y_{n+1}) + h_{n+1}^2 g_4 f'(t_n + h_{n+1}, y_{n+1}) \\ &\quad + y_n + h_{n+1}\left[\sum_{j=0}^{p-4} \beta_j f_{n-j} + \sum_{j=2}^{3} b_j F_j\right] + h_{n+1}^2 g_3 F_3'. \end{aligned} \tag{2.2}$$

An HB polynomial is used as implicit predictor P_4 to control the step size, h_{n+2}, and obtain $\widetilde{y}_{n+1}$ to order $p-2$,

$$\begin{aligned} \widetilde{y}_{n+1} &= h_{n+1} a_{44} f(t_n + h_{n+1}, y_{n+1}) + h_{n+1}^2 \gamma_{44} f'(t_n + h_{n+1}, y_{n+1}) \\ &\quad + y_n + h_{n+1}\left[\sum_{j=0}^{p-4} \beta_{4j} f_{n-j} + \sum_{j=2}^{3} a_{4j} F_j\right] + h_{n+1}^2 \gamma_{43} F_3'. \end{aligned} \tag{2.3}$$

Here, the forms (2.1) and (2.2) are used by the implicit algebraic equations system defining Y_i, $i = 2, 3$ and y_{n+1} to handle implicitness in the context of stiffness.

The distinct implicit algebraic equations systems (2.1) and (2.2) defining Y_i, $i = 2, 3$ and y_{n+1} are solved iteratively by the modified Newton–Raphson method similar to the usual resolution of system of implicit algebraic equations of BDF method [17, p. 11–13].

The following terminology will be useful. An HBO(p) method is said to be a *general* variable-step HBO method if its backstep and the coefficients

$$c_2, \quad c_3, \quad a_{22} = a_{33} = b_4, \tag{2.4}$$

in (2.1) and (2.2) are variable parameters. Hence, the general variable-step HBO method has three degrees of freedom ($c_2, c_3, a_{22} = a_{33} = b_4$). If the coefficients in (2.4) are fixed, the method is said to be a *particular* variable-step method. If the step size is constant, and hence the backsteps and the coefficients in (2.4) are fixed parameters, the method is said to be a *constant-step* method.

3. Order conditions of general HBO(p)

To derive the order conditions of 3-stage $(p-3)$-step HBO(p), we shall use the following expressions coming from the backsteps of the methods:

$$B_i(j) = \sum_{\ell=1}^{p-4} \beta_{i\ell} \frac{\eta_{\ell+1}^{j-1}}{(j-1)!}, \qquad \begin{cases} i = 2, 3, \\ j = 1, 2, \ldots, p, \end{cases} \tag{3.1}$$

and

$$\eta_j = -\frac{1}{h_{n+1}} (t_n - t_{n+1-j}) = -\frac{1}{h_{n+1}} \sum_{i=0}^{j-2} h_{n-i}, \quad j = 2, 3, \ldots, p-3. \tag{3.2}$$

In the sequel, η_j will be frequently used without explicit reference to (3.2).

Forcing an expansion of the numerical solution produced by formulae (2.1) and (2.2) to agree with the Taylor expansion of the true solution, we obtain multistep-and several Runge–Kutta(RK)-type order conditions that must be satisfied by 3-stage HBO(p) methods.

To reduce a large number of RK-type order conditions (see [18]), we impose the following simplifying assumptions:

$$\sum_{j=2}^{i} \gamma_{ij} \frac{c_j^{k-1}}{(k-1)!} + \sum_{j=2}^{i} a_{ij} \frac{c_j^k}{k!} + B_i(k+1) = \frac{c_i^{k+1}}{(k+1)!}, \quad \begin{cases} i = 2, 3, \\ k = 0, 1, \dots, p-3. \end{cases} \tag{3.3}$$

Thus, there remain only two sets of equations to be solved:

$$\sum_{i=3}^{4} g_i \frac{c_i^{k-1}}{(k-1)!} + \sum_{i=2}^{4} b_i \frac{c_i^k}{k!} + B(k+1) = \frac{1}{(k+1)!}, \quad k = 0, 1, \dots, p-1, \tag{3.4}$$

$$\sum_{i=3}^{4} g_i \frac{c_i^{p-2}}{(p-2)!} + \sum_{i=2}^{3} b_i \left[\sum_{j=2}^{i} \gamma_{ij} \frac{c_j^{p-3}}{(p-3)!} + \sum_{j=2}^{i} a_{ij} \frac{c_j^{p-2}}{(p-2)!} + B_i(p-1) \right] + b_4 \frac{c_4^{p-1}}{(p-1)!} + B(p) = \frac{1}{p!}, \tag{3.5}$$

where the backstep parts, $B(j)$, are defined by

$$B(j) = \sum_{\ell=1}^{p-4} \beta_\ell \frac{\eta_{\ell+1}^{j-1}}{(j-1)!}, \qquad j = 1, 2, \dots, p+1. \tag{3.6}$$

These order conditions are simply RK order conditions with backstep parts $B_i(\cdot)$ and $B(\cdot)$.

4. Vandermonde-type formulation of particular variable step HBO(p)

The general HBO(p) methods obtained in Section 3 contain free coefficients in (2.4), and depend on h_{n+1} and the previous nodes, $t_n, t_{n-1}, \dots, t_{n-(p-4)}$, which determine $\eta_2, \eta_3, \dots, \eta_{p-3}$ in (3.2). To obtain A-stability of particular HBO(p) methods, $p = 9, 10$, the coefficients listed in Table 1 were chosen. In Table 1, since $a_{22} = a_{33} = b_4$, only values of a_{22} are listed.

It is to be noted that, to obtain the coefficients in Table 1, the well-known *exhaustive search* method is used with possible candidates (c_2, c_3, a_{22}), for $c_2 = 0.1, 0.15, 0.2, \dots, 2.0$, $c_3 = 0.1, 0.15, 0.2, \dots, 2.0$ and a_{22} with the same increment (or decrement) size 0.05. The value of a_{22} which yields the largest α of $A(\alpha)$-stability is used as a starting value (the exhaustive search method can be repeated with new starting values of a_{22}).

The remaining of this paper is concerned with particular VS HBO(p) $p = 9, 10$, with coefficients c_i, $i = 2, 3$, and a_{22} given in Table 1.

TABLE 1. Coefficients c_i, $i = 2, 3$ and a_{22} $(= a_{33} = b_4)$ of particular VS HBO(p), $p = 9, 10$.

k	6	7
coeffs$\backslash p$	9	10
c_2	1.450000000000000e+00	2.000000000000000e+00
c_3	1.151000000000000e+00	1.401000000000001e+00
a_{22}	8.614213197969537e-01	9.614213197969360e-01

4.1. **Predictor** P_2**.** The $(p-2)$-vector of reordered coefficients of the predictor P_2 in (2.1) with $i = 2$,

$$\boldsymbol{u}^2 = [\beta_{20}, \beta_{21}, \ldots, \beta_{2,p-4}, \gamma_{22}]^T,$$

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

$$M^2 \boldsymbol{u}^2 = \boldsymbol{r}^2, \tag{4.1}$$

where

$$M^2 = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 0 \\ 0 & \eta_2 & \eta_3 & \cdots & \eta_{p-3} & 1 \\ 0 & \frac{\eta_2^2}{2!} & \frac{\eta_3^2}{2!} & \cdots & \frac{\eta_{p-3}^2}{2!} & c_2 \\ 0 & \frac{\eta_2^3}{3!} & \frac{\eta_3^3}{3!} & \cdots & \frac{\eta_{p-3}^3}{3!} & \frac{c_2^2}{2!} \\ \vdots & & & & & \vdots \\ 0 & \frac{\eta_2^{p-3}}{(p-3)!} & \frac{\eta_3^{p-3}}{(p-3)!} & \cdots & \frac{\eta_{p-3}^{p-3}}{(p-3)!} & \frac{c_2^{p-4}}{(p-4)!} \end{bmatrix}, \tag{4.2}$$

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

$$r_2(i) = \frac{c_2^i}{i!} - a_{22}\frac{c_2^{i-1}}{(i-1)!}, \qquad i = 1, 2, \ldots, p-2.$$

A truncated Taylor expansion of the right-hand side of (2.1) with $i = 2$ about t_n gives

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

with coefficients

$$\begin{aligned} S_2(j) &= a_{22}\frac{c_2^{j-1}}{(j-1)!} + M^2(j, 1 : p-2)\boldsymbol{u}^2 \\ &= a_{22}\frac{c_2^{j-1}}{(j-1)!} + r_2(j) = \frac{c_2^j}{j!}, \qquad j = 1, 2, \ldots, p-2, \\ S_2(j) &= a_{22} S_2(j-1) + \sum_{i=1}^{p-4} \beta_{2i}\frac{\eta_{i+1}^{j-1}}{(j-1)!}, \qquad j = p-1, p, p+1. \end{aligned}$$

We note that P_2 is of order $p-2$ since it satisfies the order conditions

$$S_2(j) = c_2^j / j!, \qquad j = 1, 2, \ldots, p-2,$$

and its leading error term is

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

4.2. **Integration formula** IF**.** The p-vector of reordered coefficients of the integration formula IF in (2.2),

$$\boldsymbol{u}^1 = [\beta_0, b_3, b_2, \beta_1, \beta_2, \ldots, \beta_{p-4}, g_3]^T,$$

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

$$M^1 \boldsymbol{u}^1 = \boldsymbol{r}^1, \tag{4.3}$$

where

$$M^1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 0 \\ 0 & c_3 & c_2 & \eta_2 & \eta_3 & \cdots & \eta_{p-3} & 1 \\ 0 & \frac{c_3^2}{2!} & \frac{c_2^2}{2!} & \frac{\eta_2^2}{2!} & \frac{\eta_3^2}{2!} & \cdots & \frac{\eta_{p-3}^2}{2!} & c_3 \\ \vdots & & & & & & & \vdots \\ 0 & \frac{c_3^{p-1}}{(p-1)!} & \frac{c_2^{p-1}}{(p-1)!} & \frac{\eta_2^{p-1}}{(p-1)!} & \frac{\eta_3^{p-1}}{(p-1)!} & \cdots & \frac{\eta_{p-3}^{p-1}}{(p-1)!} & \frac{c_3^{p-2}}{(p-2)!} \end{bmatrix}, \tag{4.4}$$

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

$$r_1(1) = 1 - b_4,$$

$$r_1(i) = \frac{1}{i!} - b_4 \frac{c_4^{i-1}}{(i-1)!} - g_4 \frac{c_4^{i-2}}{(i-2)!}, \qquad i = 2, 3, \ldots, p,$$

where $b_4 = a_{22}$ and $g_4 = \gamma_{22}$.

The leading error term of IF is

$$\left[g_4 \frac{c_4^{p-1}}{(p-1)!} + b_4 \frac{c_4^p}{p!} + \sum_{j=1}^{p-4} \beta_j \frac{\eta_{j+1}^p}{p!} + \sum_{j=2}^{3} b_j \frac{c_j^p}{p!} + g_3 \frac{c_3^{p-1}}{(p-1)!} - \frac{1}{(p+1)!}\right] h_{n+1}^{p+1} y_n^{p+1}.$$

4.3. **Predictor** P_3**.** We consider the $(p-1)$-vector of reordered coefficients of the predictor P_3 in (2.1) with $i = 3$,

$$\boldsymbol{u}^3 = [\beta_{30}, a_{32}, \beta_{31}, \ldots, \beta_{3,p-4}, \gamma_{32}]^T,$$

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

$$M^3 \boldsymbol{u}^3 = \boldsymbol{r}^3, \tag{4.5}$$

where

$$M^3 = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 0 \\ 0 & c_2 & \eta_2 & \eta_3 & \cdots & \eta_{p-3} & 1 \\ 0 & \frac{c_2^2}{2!} & \frac{\eta_2^2}{2!} & \frac{\eta_3^2}{2!} & \cdots & \frac{\eta_{p-3}^2}{2!} & c_2 \\ \vdots & & & & & & \vdots \\ 0 & \frac{c_2^{p-2}}{(p-2)!} & \frac{\eta_2^{p-2}}{(p-2)!} & \frac{\eta_3^{p-2}}{(p-2)!} & \cdots & \frac{\eta_{p-3}^{p-2}}{(p-2)!} & \frac{c_2^{p-3}}{(p-3)!} \end{bmatrix}. \tag{4.6}$$

The first $(p-2)$ components of $\boldsymbol{r}^3 = r_3(1:p-1)$ are

$$r_3(1) = c_3 - a_{33},$$
$$r_3(i) = \frac{c_3^i}{i!} - a_{33}\frac{c_3^{i-1}}{(i-1)!} - \gamma_{33}\frac{c_3^{i-2}}{(i-2)!}, \qquad i = 2, 3, \ldots, p-2,$$

the $(p-1)$th component is

$$r_3(p-1) = S_3(p-1) - a_{33}\frac{c_3^{p-2}}{(p-2)!} - \gamma_{33}\frac{c_3^{p-3}}{(p-3)!}, \tag{4.7}$$

where

$$S_3(p-1) = \frac{1}{b_3}\Bigg[\frac{1}{p!} - b_2 S_2(p-1) - b_4\frac{c_4^{p-1}}{(p-1)!} - g_3\frac{c_3^{p-2}}{(p-2)!} - g_4\frac{c_4^{p-2}}{(p-2)!} - B(p)\Bigg],$$

$a_{33} = a_{22}$ and $\gamma_{33} = \gamma_{22}$.

The equation for $r_3(p-1)$ in (4.7) corresponds to order condition (3.5).

A truncated Taylor expansion of the right-hand side of (2.1), with $i = 3$, about t_n gives

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

with coefficients

$$\begin{aligned} S_3(j) &= a_{33}\frac{c_3^{j-1}}{(j-1)!} + \gamma_{33}\frac{c_3^{j-2}}{(j-2)!} + M^3(j+1, 1:p-1)\boldsymbol{u}^3 \\ &= a_{33}\frac{c_3^{j-1}}{(j-1)!} + \gamma_{33}\frac{c_3^{j-2}}{(j-2)!} + r_3(j) = \frac{c_3^j}{j!}, \quad j = 1, 2, \ldots, p-2, \\ S_3(j) &= a_{33}S_3(j-1) + \gamma_{33}S_3(j-2) + a_{32}S_2(j-1) + \gamma_{32}S_2(j-2) \\ &\quad + \sum_{i=1}^{p-4} \beta_{3i}\frac{\eta_{i+1}^{j-1}}{(j-1)!}, \qquad j = p-1, p, p+1. \end{aligned}$$

4.4. **Step control predictor** P$_4$**.** The $(p-2)$-vector of reordered coefficients of predictor P$_4$ in (2.3),

$$\boldsymbol{u}^4 = [\beta_{40}, a_{42}, \beta_{41}, \ldots, \beta_{4,p-4}]^T,$$

is the solution of the system of order conditions

$$M^4\boldsymbol{u}^4 = \boldsymbol{r}^4, \tag{4.8}$$

where

$$M^4 = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & c_2 & \eta_2 & \eta_3 & \cdots & \eta_{p-3} \\ 0 & \frac{c_2^2}{2!} & \frac{\eta_2^2}{2!} & \frac{\eta_3^2}{2!} & \cdots & \frac{\eta_{p-3}^2}{2!} \\ \vdots & & & & & \vdots \\ 0 & \frac{c_2^{p-3}}{(p-3)!} & \frac{\eta_2^{p-3}}{(p-3)!} & \frac{\eta_3^{p-3}}{(p-3)!} & \cdots & \frac{\eta_{p-3}^{p-3}}{(p-3)!} \end{bmatrix}, \tag{4.9}$$

and $\boldsymbol{r}^4 = r_4(1 : p-2)$ has components

$$\begin{aligned} r_4(1) &= 1 - (b_4 + \omega_4) - (b_3 + \omega_3), \\ r_4(i) &= \frac{1}{i!} - (g_4 + \omega_4')\frac{c_4^{i-2}}{(i-2)!} - (b_4 + \omega_4)\frac{c_4^{i-1}}{(i-1)!} \\ &\qquad - (g_3 + \omega_3')\frac{c_3^{i-2}}{(i-2)!} - (b_3 + \omega_3)\frac{c_3^{i-1}}{(i-1)!}, \quad i = 2, 3, \dots, p-2. \end{aligned}$$

For arbitrary nonzero ω_3 and ω_3', P_4 yields $\widetilde{y}_{n+1}$ to order $(p-2)$. A good experimental choice is $\omega_3 = 0.025$, $\omega_3' = 0.025$, $\omega_4 = 0.025$, $\omega_4' = 0.025$.

The solutions $\boldsymbol{u}^\ell$, $\ell = 1, 2, 3, 4$, form generalized Lagrange basis functions for representing the HB interpolation polynomials.

5. Symbolic construction of elementary matrix functions

Consider the matrices

$$M^\ell \in \mathbb{R}^{m_\ell \times m_\ell}, \qquad \ell = 2, 1, 3, 4, \tag{5.1}$$

of the Vandermonde-type systems (4.1), (4.3), (4.5) and (4.8), where

$$m_2 = p-2, \qquad m_1 = p, \qquad m_3 = p-1, \qquad m_4 = p-2, \tag{5.2}$$

and p is the order of the method.

The purpose of this section is to construct elementary lower and upper triangular matrices as symbolic functions of the parameters of HBO(p). These matrices are most easily constructed by means of a symbolic software. These functions will be used in Section 6 to factor

- M^ℓ into a diagonal+last-1-column matrix, W_1^ℓ, $\ell = 2, 1, 3$, which will be further diagonalized by a Gaussian elimination,
- M^4 into the identity matrix I^4.

This decomposition will lead to a fast solution of the systems $M^\ell \boldsymbol{u}^\ell = \boldsymbol{r}^\ell$, $\ell = 2, 1, 3, 4$ in $\mathrm{O}(p^2)$ operations.

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 [2], Krogh [16], Galimberti and Pereyra [8] and Björck and Elfving [1]. Pivoting is not needed in this decomposition because of the special structure of Vandermonde-type matrices.

5.1. **Symbolic construction of lower bidiagonal matrices for M^ℓ, $\ell = 1,2,3,4$.** We first describe the zeroing process of a general vector $\boldsymbol{x} = [x_1, x_2, \dots, x_m]^T$ with no zero elements. The lower bidiagonal matrix

$$L_k = \begin{bmatrix} I_{k-1} & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & -\tau_{k+1} & 1 & & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & -\tau_m & 1 \end{bmatrix}, \tag{5.3}$$

defined by the multipliers $\tau_i = \frac{x_i}{x_{i-1}} = -L_k(i, i-1), \quad i = k+1, k+2, \dots, m$, zeros the last $(m-k)$ components, $x_{k+1}, x_{k+2}, \dots, x_m$, of $\boldsymbol{x}$. This zeroing process will be applied recursively on M^ℓ, $\ell = 1,2,3,4$, as follows. For $k = 2,3,\dots$, left multiplying $T_k^\ell = L_{k-1}^\ell L_{k-2}^\ell \cdots L_3^\ell L_2^\ell M^\ell$ by L_k^ℓ zeros the last $(m_\ell - k)$ components of the kth column of T_k^ℓ. Thus we obtain the upper triangular matrix

$$L^\ell M^\ell = L_{m_\ell - 1}^\ell \cdots L_3^\ell L_2^\ell M^\ell, \quad \ell = 1,2,3,4, \tag{5.4}$$

in $(m_\ell - 2)$ steps, for $\ell = 1,2,3,4$.

We note that L^ℓ does not change the first two rows of M^ℓ.

Process 1. At the kth step, starting with $k = 2$,

- $M^{\ell(k-1)} = L_{k-1}^\ell L_{k-2}^\ell \cdots L_2^\ell M^\ell$ is an upper triangular matrix in columns 1 to $k-1$,
- The multipliers in L_k^ℓ are obtained from $M^{\ell(k-1)}(k+1 : m_\ell, k)$ since $M^\ell(i,k) \neq 0$ for $i = k+1, k+2, \dots, m_\ell$.

Algorithm 1 in Appendix A describes this process. The input is $M = M^\ell$; $m = m_\ell$. The output is $L_k = L_k^\ell$, $k = 2,3,\dots, m_\ell - 1$, $\ell = 1,2,3,4$.

5.2. **Symbolic construction of upper bidiagonal matrices for M^ℓ, $\ell = 1,2,3,4$.** For matrix $L^\ell M^\ell$, $\ell = 1,2,3,4$, we construct recursively upper bidiagonal matrices $U_1^\ell, U_2^\ell \dots, U_{k_{\text{end}}^\ell}^\ell$ such that right multiplying $L^\ell M^\ell$ by the upper triangular matrix $U^\ell = U_1^\ell U_2^\ell \cdots U_{k_{\text{end}}^\ell}^\ell$ transforms $L^\ell M^\ell$ into a matrix $W_{\mathcal{C}_\ell}^\ell = L^\ell M^\ell U^\ell$ with nonzero diagonal elements, $W_{\mathcal{C}_\ell}^\ell(i,i) \neq 0$, $i = 1,2,\dots,m_\ell$, the last $\mathcal{C}_\ell$ nonzero columns $W_{\mathcal{C}_\ell}^\ell(1 : m_\ell, j) \neq 0$, $j = m_\ell - \mathcal{C}_\ell + 1, m_\ell - \mathcal{C}_\ell + 2, \dots, m_\ell$, and zero elsewhere. We call such a matrix a "diagonal+last-$\mathcal{C}_\ell$-column matrix". Here

$$\mathcal{C}_1 = 1, \qquad \mathcal{C}_2 = 1, \qquad \mathcal{C}_3 = 1, \qquad \mathcal{C}_4 = 0, \tag{5.5}$$

$$k_{\text{end}}^1 = m_1 - 2, \quad k_{\text{end}}^2 = m_2 - 2, \quad k_{\text{end}}^3 = m_3 - 2, \quad k_{\text{end}}^4 = m_4 - 1, \tag{5.6}$$

for M^ℓ, $\ell = 1,2,3,4$, respectively.

We describe the zeroing process of the upper bidiagonal matrix U_k^ℓ on the two-row matrix $(L^\ell M^\ell)(k : k+1, 1 : m_\ell)$:

$$
\begin{aligned}
&(L^\ell M^\ell)(k:k+1,1:m_\ell)U_1^\ell U_2^\ell \cdots U_{k-1}^\ell \\
&= \left[\begin{array}{cccccc} y_{k1} & \cdots & y_{k,k-1} & 1 & \cdots & 1 \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & \cdots & y_{k+1,m_\ell-\mathcal{C}_\ell} \end{array} \right. \\
&\qquad\qquad \left. \begin{array}{cccc} y_{k,m_\ell-\mathcal{C}_\ell+1} & y_{k,m_\ell-\mathcal{C}_\ell+2} & \cdots & y_{k,m_\ell} \\ y_{k+1,m_\ell-\mathcal{C}_\ell+1} & y_{k+1,m_\ell-\mathcal{C}_\ell+2} & \cdots & y_{k+1,m_\ell} \end{array} \right].
\end{aligned} \tag{5.7}
$$

The divisors $\sigma_i = \frac{1}{y_{k+1,i}-y_{k+1,i-1}} = U_k^\ell(i,i), \quad i = k+1, k+2, \ldots, m_\ell - \mathcal{C}_\ell$, define the upper bidiagonal matrix

$$
U_k^\ell = \begin{bmatrix}
I_{k-1} & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & -\sigma_{k+1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \sigma_{k+1} & -\sigma_{k+2} & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & & \ddots & \ddots & & \vdots & & & \vdots \\
0 & 0 & 0 & \cdots & \sigma_{m_\ell-\mathcal{C}_\ell-1} & -\sigma_{m_\ell-\mathcal{C}_\ell} & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & \sigma_{m_\ell-\mathcal{C}_\ell} & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1
\end{bmatrix}. \tag{5.8}
$$

Right multiplying (5.7) by U_k^ℓ zeros the 1's in position $k+1, k+2, \ldots, m_\ell - \mathcal{C}_\ell$ in the first row and puts 1's in position $k+1, k+2, \ldots, m_\ell - \mathcal{C}_\ell$ in the second row:

$$
\begin{aligned}
&(L^\ell M^\ell)(k:k+1,1:m_\ell)U_1^\ell U_2^\ell \cdots U_{k-1}^\ell U_k^\ell \\
&= \left[\begin{array}{cccccc} y_{k1} & \cdots & y_{k,k-1} & 1 & 0 & \cdots & 0 \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & 1 & \cdots & 1 \end{array} \right. \\
&\qquad\qquad \left. \begin{array}{cccc} y_{k,m_\ell-\mathcal{C}_\ell+1} & y_{k,m_\ell-\mathcal{C}_\ell+2} & \cdots & y_{k,m_\ell} \\ y_{k+1,m_\ell-\mathcal{C}_\ell+1} & y_{k+1,m_\ell-\mathcal{C}_\ell+2} & \cdots & y_{k+1,m_\ell} \end{array} \right].
\end{aligned} \tag{5.9}
$$

Thus, $U^\ell = U_1^\ell U_2^\ell \cdots U_{k_{\text{end}}^\ell}^\ell$ transforms the upper triangular matrix $L^\ell M^\ell$ into the diagonal+last-$\mathcal{C}_\ell$-column matrix

$$
W_{\mathcal{C}_\ell}^\ell = L^\ell M^\ell U_1^\ell U_2^\ell \cdots U_{k_{\text{end}}^\ell}^\ell, \tag{5.10}
$$

in k_{end}^ℓ steps. Here k_{end}^ℓ is defined in (5.6).

Process 2. At the kth step, starting with $k = 1$,

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

Algorithm 2 in Appendix A describes this process for M^ℓ, $\ell = 1, 2, 3, 4$. The input is $M = M^\ell$; $m = m_\ell$. The output is $U_k = U_k^\ell$, $k = 1, 2, \ldots, k_{\text{end}}^\ell$ where k_{end}^ℓ is defined in (5.6).

6. Fast solution of Vandermonde-type systems for particular variable step HBO(p)

Symbolic elementary matrix functions L_k^ℓ and U_k^ℓ, $\ell = 1, 2, 3, 4$, are constructed once as functions of η_j, for $j = 2, 3, \ldots, p-3$, by Algorithms 1 and 2 in Appendix A to factor

- for $\ell = 1, 2, 3$, M^ℓ into a diagonal+last-1-column matrix, W_1^ℓ which will be further diagonalized by a Gaussian elimination,
- M^4 into the identity matrix I^4.

These elementary matrix functions are used, first, to find, successively, the solution $\boldsymbol{u}^\ell$, $\ell = 2, 1, 3, 4$ in elementary matrix functions form and, then, to construct (a) fast Algorithm 3 in Appendix A, to solve systems (4.1), (4.3), (4.5) and (b) fast Algorithm 4 to solve system (4.8), at each integration step.

6.1. **Solution of $M^\ell \boldsymbol{u}^\ell = \boldsymbol{r}^\ell$, $\ell = 1, 2, 3$.** We let $m_1 = p, m_2 = p-2$ and $m_3 = p-1$ as defined in (5.2).

(1) The elimination procedure of Subsection 5.1 is applied to M^ℓ to construct $m_\ell \times m_\ell$ lower bidiagonal matrices L_k^ℓ, $k = 2, 3, \ldots, m_\ell - 1$, with multipliers $\tau_i = \frac{M^\ell(2,k)}{i-1} = -L_k^\ell(i, i-1), \quad i = k+1, k+2, \ldots, m_\ell$.
Left multiplying the coefficient matrix M^ℓ by the lower bidiagonal matrix $L^\ell = L_{m_\ell - 1}^\ell \cdots L_3^\ell L_2^\ell$ transforms M^ℓ into the upper triangular matrix $L^\ell M^\ell$ in column 1 to $m_\ell - 1$ of the form (5.4).

(2) The elimination procedure of Subsection 5.2 is used to construct $m_\ell \times m_\ell$ upper bidiagonal matrices U_k^ℓ, $k = 1, 2, \ldots, m_\ell - 2$, with multipliers

$$\sigma_i = \frac{k}{M^\ell(2,i) - M^\ell(2, i-k)} = U_k^\ell(i,i), \qquad i = k+1, k+2, \ldots, m_\ell - 1. \tag{6.1}$$

Right multiplying $L^\ell M^\ell$ by the upper triangular matrix $U^\ell = U_1^\ell U_2^\ell \cdots U_{m_\ell - 2}^\ell$ transforms $L^\ell M^\ell$ into a diagonal+last-1-column matrix W_1^ℓ of the form (5.10).

(3) Factored Gaussian eliminations, $L_{m_\ell+1}^\ell L_{m_\ell}^\ell$, will eliminate column m_ℓ of W_1^ℓ and transform W_1^ℓ into the identity matrix $I^\ell = L_{m_\ell+1}^\ell L_{m_\ell}^\ell W_1^\ell$ where L_k^ℓ, $k = m_\ell, m_\ell + 1$ have nonzero entries listed in Table 2 and zeros elsewhere.

Table 2. Nonzero entries of Gaussian elimination matrices L_k^ℓ, $k = m_\ell, m_\ell + 1$, $\ell = 1, 2, 3$.

Gaussian elimination matrices	
$L_{m_\ell}^\ell$	$L_{m_\ell+1}^\ell$
$L_{m_\ell}^\ell(i,i) = 1, \quad i = 1, 2, \ldots, m_\ell - 1$ $L_{m_\ell}^\ell(m_\ell, m_\ell) = 1/W^\ell(m_\ell, m_\ell)$	$L_{m_\ell+1}^\ell(i,i) = 1, \quad i = 1, 2, \ldots, m_\ell$ $L_{m_\ell+1}^\ell(1 : m_\ell - 1, m_\ell) = -W^\ell(1 : m_\ell - 1, m_\ell)$

This procedure transforms M^ℓ into the identity matrix

$$I^\ell = L_{m_\ell+1}^\ell L_{m_\ell}^\ell \cdots L_2^\ell M^\ell U_1^\ell U_2^\ell \cdots U_{m_\ell - 2}^\ell.$$

Thus we have the following factorization of M^ℓ into the product of elementary matrices:

$$M^\ell = \left(L^\ell_{m_\ell+1} L^\ell_{m_\ell} \cdots L^\ell_2\right)^{-1} \left(U^\ell_1 U^\ell_2 \cdots U^\ell_{m_\ell-2}\right)^{-1},$$

and the solution is

$$\boldsymbol{u}^\ell = U^\ell_1 U^\ell_2 \cdots U^\ell_{m_\ell-2} L^\ell_{m_\ell+1} L^\ell_{m_\ell} \cdots L^\ell_2\, \boldsymbol{r}^\ell, \tag{6.2}$$

where fast computation goes from right to left.

Procedure (6.2) is implemented in Algorithm 3 in Appendix A in $O(m_\ell^2)$ operations, $\ell = 1, 2, 3$. The input is $M = M^\ell$; $m = m_\ell$; $\boldsymbol{r} = \boldsymbol{r}^\ell$; $L_k = L^\ell_k$, $k = 2, 3, \dots, m_\ell + 1$; $U_k = U^\ell_k$, $k = 1, 2, \dots, m_\ell - 2$. The output is $\boldsymbol{u} = \boldsymbol{u}^\ell$, $\ell = 1, 2, 3$.

6.2. **Solution of $M^4\boldsymbol{u}^4 = \boldsymbol{r}^4$.** We let $m_4 = p - 2$ as defined in (5.2).

Similar to steps (1) and (2) of Subsection 6.1, the matrix $L^4 = L^4_{m_4-1} \cdots L^4_3 L^4_2$ transforms the coefficient matrix M^4 into the upper triangular matrix L^4M^4 in column 1 to $m_4 - 1$ of the form (5.4). Next, the right-product of the U^4_k, $k = 1, 2, \dots, m_4 - 1$, will transform L^4M^4 into the identity matrix I^4 of the form (5.10).

Thus we have the following factorization of M^4 into the product of elementary matrices:

$$M^4 = \left(L^4_{m_4-1} L^4_{m_4-2} \cdots L^4_2\right)^{-1} \left(U^4_1 U^4_2 \cdots U^4_{m_4-1}\right)^{-1},$$

and the solution is

$$\boldsymbol{u}^4 = U^4_1 U^4_2 \cdots U^4_{m_4-1} L^4_{m_4-1} \cdots L^4_2\, \boldsymbol{r}^4, \tag{6.3}$$

where fast computation goes from right to left.

Procedure (6.3) is implemented in Algorithm 4 in Appendix A in $O(m_4^2)$ operations. The input is $M = M^4$; $m = m_4$; $\boldsymbol{r} = \boldsymbol{r}^4$; $L_k = L^4_k$, $k = 2, 3, \dots, m_4 - 1$; $U_k = U^4_k$, $k = 1, 2, \dots, m_4 - 1$. The output is $\boldsymbol{u} = \boldsymbol{u}^4$.

Remark 6.1. Formulae (2.1) to (2.3) can be put in matrix form. For instance, (2.2) can be written as

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

where

$$F^1 = h_{n+1}\Big[f_n, F_3, F_2, f_{n-1}, f_{n-2}, \dots, f_{n-(p-4)}, h_{n+1}F_3'\Big],$$
$$G^1 = h_{n+1}\Big[f(t_n + h_{n+1}, y_{n+1}), h_{n+1}f'(t_n + h_{n+1}\Big],$$

and

$$\boldsymbol{u}^1 = [\beta_0, b_3, b_2, \beta_1, \beta_2, \dots, \beta_{p-4}, g_3]^T,$$
$$\boldsymbol{v}^1 = [b_4, g_4]^T.$$

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

$F(UL\boldsymbol{r})$ (generalized Lagrange interpolation),
$(FU)L\boldsymbol{r}$ (generalized divided differences),
$(FUL)\boldsymbol{r}$ (Nordsieck's formulation).

The first form is used in this paper, the form similar to the second form for Vandermonde systems is found in [16], and the third form is found in [19].

7. Regions of absolute stability

The regions $\mathcal{R}$ of constant step HBO(p), $p = 5, 6, \ldots, 10$, listed in Appendix B, are obtained by applying formulae (2.1) and (2.2) of the predictors P_i, $i = 2, 3$ and the integration formula IF with constant h to the linear test equation

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

This gives the following difference equation and corresponding characteristic equation

$$\sum_{j=0}^{k} \eta_j(z)\, y_{n+j} = 0, \qquad \sum_{j=0}^{k} \eta_j(z)\, r^j = 0, \tag{7.1}$$

respectively, where $k = p - 3$ is the number of steps of the method and $z = \lambda h$. A complex number z is in $\mathcal{R}$ if the k roots of the characteristic equation in (7.1) satisfy the root condition (see [17, pp. 70]).

The scanning method used to find $\mathcal{R}$ is similar to the one used for Runge–Kutta methods (see [17]).

The stability functions $\eta_j(z)$, $j = 0, 1, \ldots, k$ in (7.1) are rational functions of the form

$$\eta_k(z) = 1, \qquad \eta_j(z) = \frac{\sum_{\ell=0}^{5} n_{j\ell} z^\ell}{\sum_{\ell=0}^{6} d_{j\ell} z^\ell}, \qquad j = 0, 1, \ldots, k-1.$$

Hence, in the difference equation of (7.1), $y_{n+k} \to 0$ as $z \to \infty$. This implies that HBO(p), $p = 5, 6, \ldots, 10$ are L-stable since these methods are A-stable.

For each given step number k, Table 3 lists the α angles of $A(\alpha)$-stability for HBO(5–10), SDEBDF(5–10) (class 1) of Cash [3], SDMM(5–9) of Enright [5], SDMM(5–10) of Ismail *et al.* [14], SDBDF(5–10) of Hairer *et al.* [10, p. 270], SDMM(5–10) of Hojjati *et al.* [12] and SDMM(5–10) of Khalsarei *et al.* [15], respectively. It is seen that α of HBO methods compare favorably with α of the considered methods of comparable order p.

8. Controlling step size

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

$$h_{n+1} = \min\left\{ h_{\max}, \beta\, h_n \left[\frac{\text{tolerance}}{\|y_n - \widetilde{y}_n\|_\infty}\right]^{1/\kappa}, 4\, h_n \right\}, \tag{8.1}$$

with $\kappa = p_{\mathrm{SCP}} + 1$ and safety factor $\beta = 0.81$. Here p_{SCP} is the order of the step control predictor (SCP) (it is to be noted that the value $\widetilde{y}_n$ is obtained by the formula (2.3)).

The procedure to advance integration from t_n to t_{n+1} is as follows.

(a) The step size, h_{n+1}, is obtained by formula (8.1).
(b) The numbers $\eta_2, \eta_3, \ldots, \eta_{p-3}$, defined in (3.2), are calculated.

TABLE 3. For each given step number k, the table lists the order p, the α angles of $A(\alpha)$-stability for the listed methods.

HBO(p)			SDEBDF(p) class 1 of Cash [3]			SDMM(p) of Enright [5]			SDMM(p) of Ismail *et al.* [14]		
k	p	α	k	p	α	k	p	α	k	p	α
2	5	90.00°	2	5	90.00°	3	5	87.88°	4	5	89.90°
3	6	90.00°	3	6	90.00°	4	6	82.03°	5	6	87.30°
4	7	90.00°	4	7	90.00°	5	7	73.10°	6	7	84.20°
5	8	90.00°	5	8	90.00°	6	8	59.95°	7	8	80.00°
6	9	90.00°	6	9	89.51°	7	9	37.61°	8	9	71.00°
7	10	90.00°	7	10	88.57°				9	10	57.80°

SDBDF(p) of Hairer *et al.* [10, p. 270]			SDMM(p) of Hojjati *et al.* [12]			SDMM(p) of Khalsarei *et al.* [15]		
k	p	α	k	p	α	k	p	α
4	5	89.36°	3	5	90.00°	3	5	90.00°
5	6	86.35°	4	6	90.00°	4	6	90.00°
6	7	80.82°	5	7	89.80°	5	7	90.00°
7	8	72.53°	6	8	88.30°	6	8	90.00°
8	9	60.71°	7	9	85.30°	7	9	89.79°
9	10	43.39°	8	10	80.50°	8	10	88.33°

(c) The coefficients of predictors P_2, integration formula IF, P_3 and step control predictor P_4 are obtained successively as solutions of systems (4.1), (4.3), (4.5) and (4.8).
(d) The values Y_2, Y_3, y_{n+1}, and $\widetilde{y}_{n+1}$ are obtained by formulae (2.1) to (2.3).
(e) The step is accepted if $\|y_{n+1} - \widetilde{y}_{n+1}\|_\infty$ 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 step size h_{n+1} is computed.

9. Numerical results of comparing L-stable methods

The *error at the endpoint of the integration interval* (EPE, endpoint error) is taken in the uniform norm,

$$\mathrm{EPE} = \{\|y_{\mathrm{end}} - z_{\mathrm{end}}\|_\infty\},$$

where y_{end} is the numerical value obtained by the numerical method at the endpoint t_{end} of the integration interval and z_{end} is the "exact solution" obtained by MATLAB's `ode15s` with stringent tolerance 5×10^{-14}.

The necessary starting values at $t_1, t_2, \dots, t_{k-1}$ for HBO(p) were obtained by MATLAB's `ode15s` with stringent tolerance 5×10^{-14}.

Computations were performed on a PC with the following characteristics: Memory: 5.8 GB, Processor 0,1,. . . ,7: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz, Operating system: Ubuntu Release 11.04, Kernel Linux 2.6.38-12-generic, GNOME 2.32.1.

We compare L-stable methods on different types of problems. The following test problems are considered. Problems 1 and 2 are often used to test stiff ODE solvers. Problem 3 is representative of some stiff oscillatory problems which arise frequently in practice. In particular, they often arise when the method of lines technique is applied to a system of partial differential equations that have some hyperbolic type of behaviour. We have chosen Problems 3 and 4 where the eigenvalues of the Jacobian matrix lie close to the imaginary axis, since it is problems of this type that cause major difficulties to many existing codes.

(1) The Oregonator equation describing Belusov-Zhabotinskii reaction [7].

Problem 1.

$$\begin{aligned} y_1' &= 77.27(y_2 + y_1 - 8.375 \cdot 10^{-6} y_1^2 - y_1 y_2), && y_1(0) = 1,\\ y_2' &= (y_3 - (1 + y_1) y_2)/77.27, && y_2(0) = 2,\\ y_3' &= 0.161(y_1 - y_3), && y_3(0) = 3, \end{aligned} \tag{9.1}$$

with $t_{\text{end}} = 360$.

(2) The van der Pol's equation [10, pp. 4–6], [12].

Problem 2.

$$\begin{aligned} y_1' &= y_2, && y_1(0) = 2,\\ y_2' &= \mu^2[(1 - y_1^2) y_2 - y_1], && y_2(0) = 0, \end{aligned} \tag{9.2}$$

where $\mu = 500$ and with $t_{\text{end}} = 0.8$.

(3) The stiff DETEST problem B5 [6].

Problem 3.

$$\begin{aligned} y_1' &= -10 y_1 + \alpha y_2, && y_1(0) = 1,\\ y_2' &= -\alpha y_1 - 10 y_2 && y_2(0) = 1,\\ y_3' &= -4 y_3 && y_3(0) = 1,\\ y_4' &= -y_4 && y_4(0) = 1,\\ y_5' &= -0.5 y_5 && y_5(0) = 1,\\ y_6' &= -0.1 y_6 && y_6(0) = 1, \end{aligned} \tag{9.3}$$

with $\alpha = 1000$ and $t_{\text{end}} = 20$.

(4) As above with $\alpha = 1500$.

(5) A problem with large eigenvalues lying close to the imaginary axis [3].

Problem 4.

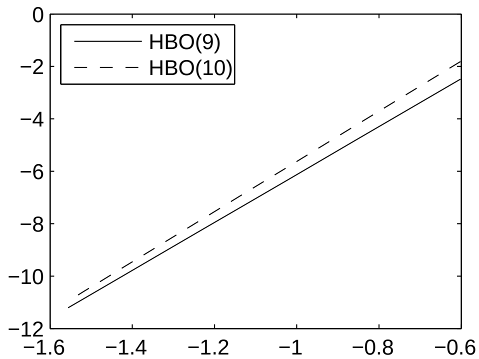
$$\begin{aligned} y_1' &= -\alpha y_1 - \beta y_2 + (\alpha + \beta - 1) e^{-t} && y_1(0) = 1,\\ y_2' &= \beta y_1 - \alpha y_2 + (\alpha - \beta - 1) e^{-t} && y_2(0) = 1,\\ y_3' &= 1 && y_3(0) = 0, \end{aligned} \tag{9.4}$$

with $\alpha = 1$, $\beta = 30$, fixed step $h = 0.09$ and $t_{\text{end}} = 20$. The exact solution is

$$y_1(t) = y_2(t) = e^{-t}, \quad y_3(t) = t.$$

(6) As above with $\beta = 42$ and fixed step $h = 1.00$.

FIGURE 1. $\log_{10}$(global error) versus $\log_{10} h$ at $t_n = 20$ for the listed HBO(p) applied to problem (9.2) with $\mu = 1$ over $t \in [0, 20]$ with constant stepsizes h.



9.1. **Numerical verification of the order p of HBO(p).** To show the relevance of the theoretical order of HBO(p), we have applied these methods with various constant stepsizes h on Problem (9.2) (van der Pol's equation) with $\mu = 1$ over $t \in [0, 20]$.

In Figure 1, the global error of y_1 and y_2 at $t_n = 20$,

$$\max\{|y_{1,n} - y_1(t_n)|, |y_{2,n} - y_2(t_n)|\} = O(h^p),$$

is plotted in a log-log scale for the listed HBO(p) methods applied to problem (9.2) over $t \in [0, 20]$ with different constant stepsizes h so that the curves appear as straight lines with slope p whenever the leading term of the global error is of order p. For HBO(p), the slopes of the straight lines which approximate the data in the least-squares sense are very close to p, which confirms the orders of the methods.

9.2. **Comparing number of steps of selected L-stable HBO(p) and highly efficient existing L-stable methods.** In our first tests, we numerically compare our most efficient methods HBO(p), $p = 9, 10$ with the highly efficient L-stable methods which are the second derivative extended backward differentiation formulae, of class 1, of order 7 and 8 [3], denoted by SDEBDF(p), $p = 7, 8$, on the basis of the EPE, endpoint error as a function of number of steps (NS). These classical SDEBDF(p) methods have been widely used to compare stiff ODE solvers.

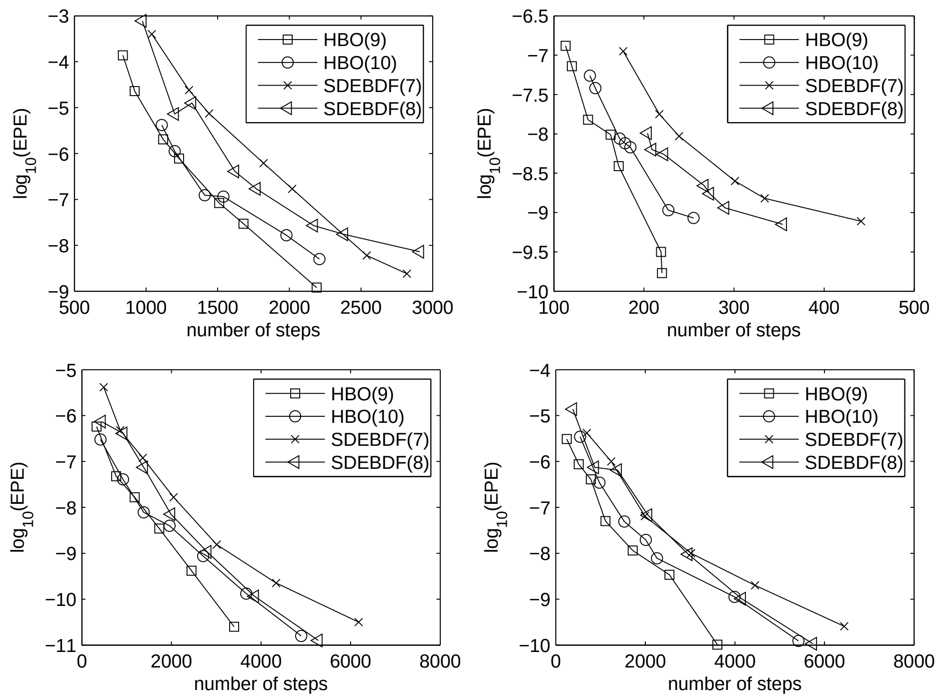
Figure 2 depicts the graph of $\log_{10}$ (EPE) (vertical axis) as a function of number of steps (NS) (horizontal axis) for the considered test problems. Here, similar to Cash [4], we use also Problem 3.

It is seen that, in general, HBO(p), $p = 9, 10$, compare favorably with SDEBDF(p), $p = 7, 8$.

The *number of steps percentage efficiency gain* (NS PEG) of method 1 over method 2 is defined by the formula (cf. Sharp [22]),

$$\text{NS PEG} = 100 \left[\frac{\sum_j \text{NS}_{2,j}}{\sum_j \text{NS}_{1,j}} - 1 \right], \tag{9.5}$$

FIGURE 2. $\log_{10}$ (EPE) (vertical axis) as a function of number of steps (NS) (horizontal axis) for Oregonator equation (top left), van der Pol's equation (top right), problem B5 with $\alpha = 1000$ (bottom left) and problem B5 with $\alpha = 1500$ (bottom right).



where $NS_{1,j}$ and $NS_{2,j}$ are the estimates of NS of methods 1 and 2, respectively, and $j = -\log_{10}(\text{EPE estimate})$. To compute $NS_{2,j}$ and $NS_{1,j}$ appearing in (9.5), we approximate the data $(\log_{10}(\text{EPE}), \log_{10}(\text{NS}))$ in a least-squares sense by MATLAB's `polyfit`. Then, for chosen integer values of the summation index j, we take $-\log_{10}(\text{EPE estimate}) = j$ and obtain $\log_{10}(\text{NS estimate})$ from the approximating curve, and finally the estimate of NS.

Table 4 lists the NS PEG of HBO(p), $p = 9, 10$, over SDEBDF(p), $p = 7, 8$, for the listed problems. It is seen that HBO(p), $p = 9, 10$, win.

9.3. **Comparing CPU time of selected L-stable HBO(p) and highly efficient existing L-stable methods.** In our second tests, programmed in C++ code, our most efficient methods HBO(p), $p = 9, 10$ and the highly efficient L-stable methods SDEBDF(p), $p = 7, 8$, are compared on the problems shown in Table 5 on the basis of EPE as a function of CPU time in seconds (CPU).

Table 4. NS PEG of HBO(p), $p = 9, 10$, over SDEBDF(p), $p = 7, 8$ for the listed problems.

Problem	NS PEG of HBO(9) over: SDEBDF(7)	NS PEG of HBO(9) over: SDEBDF(8)	NS PEG of HBO(10) over: SDEBDF(7)	NS PEG of HBO(10) over: SDEBDF(8)
Oregonator equation	39%	33%	32%	26%
van der Pol's equation	73%	37%	48%	17%
Prob. B5 with $\alpha = 1000$	66%	30%	37%	8%
Prob. B5 with $\alpha = 1500$	84%	56%	36%	16%

Table 5. CPU PEG of HBO(p), $p = 9, 10$, over SDEBDF(p), $p = 7, 8$ for the listed problems.

Problem	CPU PEG of HBO(9) over: SDEBDF(7)	CPU PEG of HBO(9) over: SDEBDF(8)	CPU PEG of HBO(10) over: SDEBDF(7)	CPU PEG of HBO(10) over: SDEBDF(8)
Oregonator equation	17%	33%	5%	-3%
van der Pol's equation	73%	37%	26%	-11%
Prob. B5 with $\alpha = 1000$	65%	30%	28%	-15%
Prob. B5 with $\alpha = 1500$	96%	56%	33%	-9%

The *CPU percentage efficiency gain* (CPU PEG) of method 1 over method 2 is defined by the formula (cf. Sharp [22]),

$$(\text{CPU PEG})_i = 100 \left[\frac{\sum_j \text{CPU}_{2,ij}}{\sum_j \text{CPU}_{1,ij}} - 1 \right], \tag{9.6}$$

where $\text{CPU}_{1,ij}$ and $\text{CPU}_{2,ij}$ are the estimates of CPU time of methods 1 and 2, respectively, associated with problem i, and estimate of EPE $= 10^{-j}$. The computation of $\text{CPU}_{2,j}$ and $\text{CPU}_{1,j}$ appearing in (9.6) is similar to the computation of $\text{NS}_{2,j}$ and $\text{NS}_{1,j}$ appearing in (9.5).

Table 5 lists the CPU PEG of HBO(p), $p = 9, 10$, over SDEBDF(p), $p = 7, 8$, for the listed problems. It is seen that HBO(9) wins. For HBO(10), the NS PEGs for HBO(10) in Table 4 are positive and tend to be larger than the CPU PEGs in Table 5 since the equations of the problems listed in Table 4 or 5 are relatively not expensive to evaluate. This would suggest that, when these equations become more expensive to evaluate, HBO(10) will be more efficient in CPU time.

Table 6 compares the efficiency of HBO(9) and HBO(10) on 4 considered problems (9.1), (9.2), (9.3), (9.4) under listed LT $= \log_{10}(\text{TOL})$. The comparison is based on CPU time in seconds, *number of steps* (NS), *number of rejected steps* (NRS) and *end point error* (EPE). It is seen that, to solve these 4 particular problems, in general, HBO(9) is more efficient.

9.4. **Comparing errors of methods on problems with large eigenvalues lying close to the imaginary axis.** Our final result is a comparison of the errors of HBO(p), $p = 9, 10$, and second derivative extended backward differentiation formulae of class 1 of order 9 and 10 [3], denoted by SDEBDF(p), $p = 9, 10$, on problems whose

Table 6. For 4 considered problems (9.1), (9.2), (9.3), (9.4) and LT $= \log_{10}(\mathrm{TOL})$, the table lists CPU time in seconds, number of steps (NS), number of rejected steps (NRS) and end point error (EPE) in corresponding left and right columns for HBO(9) and HBO(10), respectively.

		HBO(9) and HBO(10)							
Problem	LT	CPU time		NS		NRS		EPE	
Oregonator	-05	3.35e-02	3.63e-02	1125	1114	3	19	2.05e-06	4.18e-06
	-06	4.28e-02	4.32e-02	1510	1407	6	11	8.40e-08	1.22e-07
	-07	5.73e-02	5.60e-02	2188	1978	5	4	1.19e-09	1.65e-08
van der Pol	-07	1.92e-03	2.82e-03	138	173	0	2	1.53e-08	8.72e-09
	-08	2.33e-03	3.58e-03	172	227	0	4	3.86e-09	1.08e-09
	-9	2.95e-03	3.88e-03	219	255	1	0	3.15e-10	8.54e-10
B5	-03	1.30e-02	1.66e-02	768	918	2	2	4.77e-08	4.09e-08
$\alpha = 1000$	-05	2.89e-02	3.48e-02	1732	1959	1	1	3.43e-09	4.02e-09
	-07	5.69e-02	6.56e-02	3405	3669	0	0	2.58e-11	1.32e-10
B5	-02	8.75e-03	1.48e-02	514	822	3	2	8.72e-07	2.38e-07
$\alpha = 1500$	-04	2.84e-02	3.58e-02	1724	2013	1	0	1.16e-08	1.97e-08
	-06	5.94e-02	7.17e-02	3612	3988	0	0	1.03e-10	1.13e-09

Table 7. Error results obtained for the solution of Problem 4 (with $\alpha = 1$, $\beta = 30$, fixed step $h = 0.09$) as a function of step number k and t.

		HBO methods		SDEBDF methods	
		$\lvert$Error in $y_1\rvert$	$\lvert$Error in $y_2\rvert$	$\lvert$Error in $y_1\rvert$	$\lvert$Error in $y_2\rvert$
$k = 6$					
	$t = 10.0$	0.125×10^{-15}	0.356×10^{-15}	0.895×10^{-14}	0.116×10^{-12}
	$t = 15.0$	0.157×10^{-17}	0.382×10^{-17}	0.520×10^{-13}	0.143×10^{-12}
	$t = 20.0$	0.176×10^{-19}	0.429×10^{-19}	0.103×10^{-12}	0.675×10^{-13}
$k = 7$					
	$t = 10.0$	0.126×10^{-16}	0.594×10^{-16}	0.121×10^{-14}	0.361×10^{-13}
	$t = 15.0$	0.620×10^{-16}	0.530×10^{-16}	0.139×10^{-12}	0.164×10^{-12}
	$t = 20.0$	0.476×10^{-16}	0.135×10^{-17}	0.144×10^{-11}	0.933×10^{-12}

Jacobians have some large eigenvalues lying close to the imaginary axis. For this comparison, similar to Cash [3], we use Problem 4.

Table 7 presents error results obtained for the solution of Problem 4 (with $\alpha = 1$, $\beta = 30$, fixed step $h = 0.09$ suggested by Cash [3]) as a function of step number k and t. It is seen that HBO(p), $p = 9, 10$, and SDEBDF(p), $p = 9, 10$ remain stable for the integration of this problem.

Next, we present a numerical example which demonstrates the superior stability of the class of high order HBO(p), $p = 9, 10$. The problem integrated was Problem 4 with large eigenvalues lying close to the imaginary axis, $\alpha = 1$, β increased to 42 and fixed step $h = 1.00$. Table 8 shows HBO(p), $p = 9, 10$, remain stable while there is instability for SDEBDF(p), $p = 9, 10$.

TABLE 8. Error results obtained for the solution of Problem 4 (with $\alpha = 1$, $\beta = 42$, fixed step $h = 1.00$) as a function of step number k and t.

		HBO methods		SDEBDF methods	
		$\lvert$Error in $y_1\rvert$	$\lvert$Error in $y_2\rvert$	$\lvert$Error in $y_1\rvert$	$\lvert$Error in $y_2\rvert$
$k = 6$					
	$t = 10.0$	0.587×10^{-8}	0.169×10^{-8}	$0.603 \times 10^{+07}$	$0.254 \times 10^{+06}$
	$t = 15.0$	0.396×10^{-10}	0.146×10^{-10}	$0.569 \times 10^{+71}$	$0.614 \times 10^{+71}$
	$t = 20.0$	0.248×10^{-12}	0.976×10^{-13}	$0.138 \times 10^{+135}$	$0.115 \times 10^{+136}$
$k = 7$					
	$t = 10.0$	0.357×10^{-8}	0.289×10^{-8}	0.255×10^{-05}	0.555×10^{-05}
	$t = 15.0$	0.298×10^{-10}	0.233×10^{-10}	0.131×10^{-01}	0.168×10^{-01}
	$t = 20.0$	0.230×10^{-12}	0.859×10^{-13}	$0.737 \times 10^{+02}$	$0.190 \times 10^{+02}$

10. CONCLUSION

Second derivative multistep 3-stage Hermite–Birkhoff–Obrechkoff (HBO) methods of orders p were considered. It is seen that HBO(p) are L-stable up to order 10.

Selected HBO(p) of order p, $p = 9, 10$, compare positively with existing Cash modified extended backward differentiation formulae, SDEBDF(p), $p = 7, 8$ in solving differential equations problems often used to test highly stable stiff ODE solvers.

HBO(p) of order p, $p = 5, 6, \ldots, 10$, are members of general variable-step variable-order (VSVO) highly stable 3-stage k-step of order $p = k + 3$ which appear to be promising highly stable stiff ODE solvers in the light of the numerical results obtained in this paper.

11. ACKNOWLEDGMENT

Thanks are due to the reviewer whose deep and extended comments contributed to substantially improve the manuscript. Thanks are also due to Martín Lara for supplying the authors with his programs and sharing his experience. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

APPENDIX A. ALGORITHMS

Algorithm 1. This algorithm constructs $L_k(i, i-1)$ entries of lower bidiagonal matrices L_k (applied to IF, P_ℓ, $\ell = 2, 3, 4$) as functions of η_j, $j = 2, 3, \ldots$.

For $k = 2 : k_{\text{end}}$, do the following iteration:
For $i = m : -1 : k + 1$, do the following two steps:
Step (1) $L_k(i, i-1) = -M(i, k)/M(i-1, k)$.
Step (2) For $j = k : m$, compute:

$$M(i, j) = M(i, j) + M(i-1, j)L_k(i, i-1),$$

where $k_{\text{end}} = m_\ell - 1$, $\ell = 1, 2, 3, 4$ for IF, P_ℓ, $\ell = 2, 3, 4$.

Algorithm 2. This algorithm constructs diagonal entries $U_k(j, j)$ of upper bidiagonal matrices U_k (applied to IF, P_2, P_3 and P_4) as functions of η_j, $j = 2, 3, \ldots$.

For $k = 1 : k_{\text{end}}$, do the following iteration:
For $j = j_0 : -1 : k + 1$, do the following two steps:
Step (1) $U_k(j, j) = 1/[M(k+1, j) - M(k+1, j-1)]$.

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

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

where k_{end} values are defined in (5.6), $j_0 = m_\ell - 1$, $\ell = 1, 2, 3$ for IF, P_2, P_3 respectively and $j_0 = m_4$ for P_4.

Algorithm 3. This algorithm solves the systems for P_ℓ, $\ell = 1, 2, 3$ in $O(m^2)$ operations.

Given $[\eta_2, \eta_3, \ldots, \eta_{p-3}]$ 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 $L_{m-1}L_{m-2}\cdots, L_2\boldsymbol{r}$:
for $k = 2, 3, \ldots, m - 1$, compute

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

Step (2) This step forms the two matrices L_m and L_{m+1} which transform W_1^ℓ into the identity matrix $I^\ell = L_{m+1}L_mW_1^\ell$: this step computes the coefficients $G_m(i)$, $i = 1, 2, \ldots, m$ used to form the two matrices L_m and L_{m+1}, whose nonzero entries are listed in Table 2, as follows.
First set $G_m(1 : m)$,

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

The following computation overwrites $G_m(1 : m)$ with $L_{m-1}L_{m-2}\cdots\ L_2G_m(1 : m)$:
for $k = 2, 3, \ldots, m - 1$, compute

$$G_m(i) = G_m(i) + G_m(i - 1)L_k(i, i - 1), \quad i = m, m - 1, \ldots, k + 1.$$

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

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

next, for $k = m - 1, m - 2, \ldots, 1$, compute

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

Step (4) The following iteration overwrites $\boldsymbol{r} = r(1 : m)$ with $U_1U_2\cdots U_{m-2}\boldsymbol{r}$:
For $k = m - 2, m - 3, \ldots, 1$, compute

$$r(i) = r(i)U_k(i, i), \qquad i = k + 1, k + 2, \ldots, m - 1,$$

$$r(i) = r(i) - r(i + 1), \qquad i = k, k + 1, \ldots, m - 2.$$

Algorithm 4. This algorithm solves the systems for P_4 in $O(m^2)$ operations.

Given $[\eta_2, \eta_3, \ldots, \eta_{p-3}]$ 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 $L_{m-1}L_{m-2}\cdots L_2\boldsymbol{r}$:
for $k = 2, 3, \ldots, m - 1$, compute

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

Step (2) The following iteration overwrites $\boldsymbol{r} = r(1 : m)$ with $U_1U_2\cdots U_{m-1}\boldsymbol{r}$:
For $k = m - 1, m - 2, \ldots, 1$, compute

$$r(i) = r(i)U_k(i, i), \qquad i = k + 1, k + 2, \ldots, m,$$

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

Appendix B. Coefficients of HBO(p), $p = 9, 10$.

The appendix lists the coefficients of HBO(p), of order $p = 9, 10$, considered in this paper. It is to be noted that, in Table 9, since $a_{22} = a_{33} = b_4$ and $\gamma_{22} = \gamma_{33} = g_4$, only values of a_{22} and γ_{22} are listed.

TABLE 9. Coefficients of the implicit predictors P_i, $i = 2, 3$ and of the integration formulae of HBO(p), $p = 9, 10$.

k	6	7
coeffs$\backslash p$	9	10
c_2	1.450000000000000e+00	2.0
γ_{22}	-2.3103767125639274e-01	-2.7630285498304796e-01
a_{22}	8.6142131979695369e-01	9.6142131979693601e-01
β_{20}	4.3093866394931502e-01	1.3923420408193379e+00
β_{21}	6.0680052219178815e-01	1.0366637439360520e-01
β_{22}	-8.5099279806032546e-01	-1.4416141723243714e+00
β_{23}	5.7563546009809197e-01	1.7667276916004173e+00
β_{24}	-2.0406777596289427e-01	-1.0864846056891597e+00
β_{25}	3.0264607987070861e-02	3.5179100559806042e-01
β_{26}		-4.7849654194825481e-02
c_3	1.151000000000000e+00	1.401
γ_{32}	9.5140316545356249e-02	6.7204577435784785e-02
a_{32}	-1.8183754834295024e-01	-1.1236246851810028e-01
β_{30}	6.3162633555209435e-01	7.2383524894842388e-01
β_{31}	-3.3675269016743059e-01	-4.5569231676247846e-01
β_{32}	3.0716922073213720e-01	6.2715646248743961e-01
β_{33}	-1.8333126579366760e-01	-5.8014126364744922e-01
β_{34}	6.1581455752180346e-02	3.2368588228387790e-01
β_{35}	-8.8768275293166707e-03	-1.0020439557837188e-01
β_{36}		1.3301530989723063e-02
g_3	1.3288833164249580e-01	1.4887022016042095e-01
b_3	-1.8851980976917937e-01	-1.6116444980357206e-01
b_2	-5.1439833785719216e-02	-4.2323856760854671e-02
β_0	4.1668320798955982e-01	1.9106886517909408e-01
β_1	-5.1423205520101344e-02	9.6851663459148446e-02
β_2	1.7544794868273095e-02	-7.2341751929116349e-02
β_3	-5.1936846505160816e-03	3.6674997790626558e-02
β_4	1.0234654691055141e-03	-1.2535699142935602e-02
β_5	-9.6254398376063871e-05	2.5942475872990241e-03
β_6		-2.4533617662543620e-04

REFERENCES

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

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

[3] J. R. Cash, *Second derivative extended backward differentiation formulas for the numerical integration of stiff systems*, SIAM J. Numer. Anal., *18* (1981), 21–36.

[4] J. R. Cash, *A comparison of some codes for the stiff oscillatory problem*, Comput. Math. Appl., *36* (1998), 51–57.

[5] W. H. Enright, *Second derivative multistep methods for stiff ordinary differential equations*, SIAM J. Numer. Anal., *11* (1974), 321–331.

[6] W. H. Enright and J. D. Pryce, *Two Fortran packages for assessing initial value methods*, ACM Trans. Math., Software, *13* (1987), 1–27.

[7] J. Field and R. M. Noyes, *Oscillations in chemical systems. IV: Limit cycle behavior in a model of a real chemical reaction*, J. Chem. Phys., *60* (1974), 1877–1884.

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

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

[10] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, corr. sec. print., Springer-Verlag, Berlin, 2002.

[11] A. C. Hindmarsh, *Avoiding BDF stability barriers in the MOL solution of advection dominated problems*, Appl. Numer. Math., *17* (1995), 311–318.

[12] G. Hojjati, M. Y. Rahimi and S. M. Hosseini, *New second derivative multistep methods for stiff systems*, Appl. Math. Model., *30* (2006), 466–476.

[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] G. Ismail and I. Ibrahim, *New efficient second derivative multistep methods for stiff systems*, Appl. Math. Model., *23* (1999), 279–288.

[15] M. M. Khalsaraei, N. N. Oskuyi and G. Hojjati, *A class of second derivative multistep methods for stiff systems*, Acta Universitatis Apulensis, *30* (2012), 171–188.

[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, (1974), 22–71.

[17] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, Wiley, Chichester UK, 1991.

[18] T. Nguyen-Ba, E. Kengne and R. Vaillancourt, *One-step 4-stage Hermite–Birkhoff–Taylor ODE solver of order 12*, Can. Appl. Math. Q., *16*(1) (2008), 77–94.

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

[20] N. Obrechkoff, *Neue Quadraturformeln*, Abh. Preuss. Akad. Wiss. Math. Nat. Kl., *4* (1940), 1–20.

[21] L. R. Petzold, *A description of DASSL: A differential/algebraic system solver*, In Proceedings of IMACS World Congress, Montréal, Canada, 1982.

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

[23] R. F. Warming and M. Beam, *An extension of A-stability to alternating direction implicit methods*, BIT, *19* (1979), 395–417.