Acta Universitatis Apulensis No. 41/2015
ISSN: 1582-5329 pp. 177-220
http://www.uab.ro/auajournal/ doi: 10.17114/j.aua.2015.41.14

VARIABLE-STEP 4-STAGE HERMITE–BIRKHOFF SOLVER OF ORDER 9 AND 10 FOR STIFF ODES

T. Nguyen-Ba, T. Giordano, R. Vaillancourt

Abstract. Variable-step (VS) 4-stage k-step Hermite–Birkhoff (HB) methods of order $p = (k+2)$, $p = 9, 10$, denoted by HB(p), are constructed as a combination of linear k-step methods of order $(p-2)$ and a diagonally implicit one-step 4-stage Runge–Kutta method of order 3 (DIRK3) for solving stiff ordinary differential equations. 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 confluent Vandermonde-type systems. This approach allows us to develop $L(\alpha)$-stable methods of order up to 10. Fast algorithms are developed for solving these systems in $O(p^2)$ operations to obtain HB interpolation polynomials in terms of generalized Lagrange basis functions. The stepsizes of these methods are controlled by a local error estimator. HB(p) of order $p = 9$ and 10 compare favorably with existing Cash modified extended backward differentiation formulae of order 7 and 8, MEBDF(7-8), in solving problems often used to test higher order stiff ODE solvers on the basis of number of steps and error at the endpoint of the integration interval.

2010 *Mathematics Subject Classification*: Primary: 65L06; Secondary: 65D05, 65D30.

Keywords: general linear method for stiff ODE's, Hermite–Birkhoff method, endpoint error, number of steps, stiff DETEST problems, confluent Vandermonde-type systems.

1. Introduction

In this paper, we shall be concerned with solving stiff systems of first-order ordinary differential equations of the form

$$y' = f(t, y), \qquad y(t_0) = y_0, \qquad \text{where} \quad ' = \frac{d}{dt} \quad \text{and} \quad y \in \mathbb{R}^n. \tag{1}$$

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 [13]. 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 [22]. 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 formulas of DIFSUB and DASSL. In this paper, a fully variable step size implementation is used with actual mesh. A brief survey of methods for the numerical integration of (1) reveals that many of the advances in the class of general linear multistep methods for stiff ODEs, methods like extended backward differentiation formula (EBDF), modified extended backward differentiation formula (MEBDF), adaptive extended backward differentiation formula (AEBDF) and hybrid backward differentiation formula (HBDF) [5, 6, 7, 8, 15, 9], are based on backward differentiation formula (BDF). These methods are A-stable or A(α)-stable. The first modification introduced by Cash [5] was the EBDF in which one superfuture point has been applied.

In this paper, methods with four off-step points are presented. A linear k-step method of order $p-2$ and a diagonally implicit one-step 4-stage Runge–Kutta method of order 3 (DIRK3) are cast into a k-step 4-stage Hermite–Birkhoff method of order $p = k + 2$, named HB(p), $p = 9, 10$, because it uses Hermite–Birkhoff interpolation polynomials, for solving stiff ordinary differential equations (ODE) (1). Here, the DIRK3 is defined in Section 2 with $p = 3$ and step number $k = 1$. This method is similar to the diagonally implicit one-step Runge–Kutta methods (DIRK) found in [1] except that, following the approach of Cash [5], the abscissae c_i are allowed to be $0 \leq c_i \leq 2$, $i = 2, 3, 4$. The methods which we shall derive will be observed to require more work per step, but to have higher orders of accuracy and better stability characteristics, than existing methods.

Forcing a Taylor expansion of the numerical solution of HB(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.

It was found experimentally that, generally, increasing the number of backstep points is efficient in increasing the accuracy of HB methods and the stability of HB methods increases with the number of off-step points. The HB(p), considered here, are $L(\alpha)$-stable methods of order up to 10.

It was also found that, with a given fixed number of off-step points, increased speed is generally achieved by higher order HB methods.

HB(p), $p = 9, 10$ compare favorably with MEBDF(p), $p = 7, 8$, [5, 6] on problems often used to test higher order ODE solvers for stiff ODEs on the basis of the number of steps and the error at the endpoint of the interval of integration. In Section 2, we introduce new general VS HB(p) methods of order p. Order conditions of general VS HB(p) are listed in Section 3. In Section 4, particular variable step HB(p), $p = 9, 10$ 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 particular variable step HB(p) is constructed in Section 6. Section 7 considers the regions of absolute stability of constant step HB(p), $p = 3, 4, \dots, 10$. Section 8 deals with the step control. In Section 9, we compare the numerical performance of the methods considered in this paper. Appendix A lists the algorithms. Appendix B lists the coefficients of DIRK3 and constant step HB(p) methods of order $p = 4, 5, \dots, 10$.

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

General 4-stage HB methods are constructed, as a subclass of general linear methods, by the following four formulae to perform integration from t_n to t_{n+1}.

Let h_{n+1} denote the step size. The abscissa vector $[c_1, c_2, \dots, c_5]^T$ defines the off-step points $t_n + c_j h_{n+1}$ with $c_1 = 0$ and $c_5 = 1$. Following the approach of Cash [5], c_i are allowed to be $0 \le c_i \le 2$, $i = 2, 3, 4$.

Let $F_1 = f_n$ and $F_j := f(t_n + c_j h_{n+1}, Y_j)$, $j = 2, 3, 4, 5$, denote the jth stage derivative.

With the initial stage value, $Y_1 = y_n$, HB polynomials of degree $k + i - 1$ are used as predictors P_i to obtain the stage values Y_i to order $p - 2$,

$$Y_i = h_{n+1} a_{ii} f(t_n + c_i h_{n+1}, Y_i) + \sum_{j=0}^{p-3} \alpha_{ij} y_{n-j} + h_{n+1} \left[\sum_{j=1}^{i-1} a_{ij} F_j \right], \quad i = 2, 3, 4. \tag{2}$$

An HB polynomial of degree $k + 3$ is used as implicit integration formula IF to obtain y_{n+1} to order p,

$$y_{n+1} = h_{n+1} b_5 f(t_n + h_{n+1}, y_{n+1}) + \sum_{j=0}^{p-3} \alpha_j y_{n-j} + h_{n+1} \left[\sum_{j=2}^{4} b_j F_j \right]. \tag{3}$$

An HB polynomial of degree $k+3$ is used as implicit predictor P_5 to control the stepsize, h_{n+2}, and obtain $\widetilde{y}_{n+1}$ to order $p-2$,

$$\widetilde{y}_{n+1} = h_{n+1} a_{55} f(t_n + h_{n+1}, y_{n+1}) + \sum_{j=0}^{p-3} \alpha_{5j} y_{n-j} + h_{n+1}\left[\sum_{j=2}^{4} a_{5j} F_j\right]. \tag{4}$$

Here, the forms (2)–(3) are used by the implicit algebraic equations system defining Y_i, $i = 2, 3, 4$ and y_{n+1} to handle implicitness in the context of stiffness.

The distinct implicit algebraic equations systems (2)–(3) defining Y_i, $i = 2, 3, 4$ and y_{n+1} are solved exactly sequentially.

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

$$a_{22} = a_{33} = a_{44} = b_5, \qquad a_{32}, \tag{5}$$

in (2)–(3) are variable parameters. Hence, the general variable-step HB method has five degrees of freedom $(c_2, c_3, c_4, a_{22} = a_{33} = a_{44} = b_5, a_{32})$. If the off-step points and the coefficients in (5) are fixed, the method is said to be a *particular* variable-step method. If the stepsize is constant, and hence the backsteps, off-steps and the coefficients in (5) are fixed parameters, the method is said to be a *constant-step* method.

3. Order conditions of general HB(p)

To derive the order conditions of 4-stage ($p-2$)-step HB(p), we shall use the following expressions coming from the backsteps of the methods:

$$B_i(j) = \sum_{\ell=1}^{p-3} \alpha_{i\ell} \frac{\eta_{\ell+1}^j}{j!}, \qquad \begin{cases} i = 2, 3, 4, \\ j = 1, 2, \dots, p, \end{cases} \tag{6}$$

and

$$\eta_j = -\frac{1}{h_{n+1}}(t_n - t_{n+1-j}) = -\frac{1}{h_{n+1}} \sum_{i=0}^{j-2} h_{n-i}, \qquad j = 2, 3, \dots, p-2. \tag{7}$$

In the sequel, η_j will be frequently used without explicit reference to (7).

Forcing an expansion of the numerical solution produced by formulae (2)–(3) to agree with the Taylor expansion of the true solution, we obtain multistep- and several RK-type order conditions that must be satisfied by 4-stage HB(p) methods.

First, we need to satisfy the following set of multistep-type consistency conditions:

$$\sum_{j=0}^{k-1} \alpha_{ij} = 1, \qquad i = 2, 3, 4, \qquad \text{and} \qquad \sum_{j=0}^{k-1} \alpha_j = 1. \tag{8}$$

Second, to reduce a large number of RK-type order conditions (see [20]), we impose the following simplifying assumptions:

$$\sum_{j=2}^{i} a_{ij} c_j^k + k! B_i(k+1) = \frac{1}{k+1}\, c_i^{k+1}, \qquad \begin{cases} i = 2, 3, 4, \\ k = 0, 1, \dots, p-3. \end{cases} \tag{9}$$

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

$$\sum_{i=2}^{5} b_i c_i^k + k! B(k+1) = \frac{1}{k+1}, \qquad k = 0, 1, \dots, p-1, \tag{10}$$

$$\sum_{i=2}^{4} b_i \left[\sum_{j=2}^{i} a_{ij} \frac{c_j^{p-2}}{(p-2)!} + B_i(p-1) \right] + b_5 \frac{c_5^{p-1}}{(p-1)!} + B(p) = \frac{1}{p!}, \tag{11}$$

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

$$B(j) = \sum_{\ell=1}^{p-3} \alpha_\ell \frac{\eta_{\ell+1}^{j}}{j!}, \qquad j = 1, 2, \dots, p+1. \tag{12}$$

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 HB(p)

The general HB(p) methods obtained in Section 3 contain free coefficients: c_i, $i = 2, 3, 4$, coefficients in (5), and depend on h_{n+1} and the previous nodes, $t_n, t_{n-1}, \dots, t_{n-(p-3)}$, which determine $\eta_2, \eta_3, \dots, \eta_{p-2}$ in (7). For simplicity and to obtain large regions of absolute stability, $\mathcal{R}$, of particular variable-step HB(p) methods, the coefficients listed in (14) and Table 1 were chosen and, following the approach of Butcher and Chen [4], the following condition is imposed on the coefficients:

$$b_4(a_{41}a_{22}a_{33} - a_{42}a_{21}a_{33} + a_{43}a_{21}a_{32} - a_{43}a_{22}a_{31}) + b_2(a_{44}a_{21}a_{33}) + b_3(a_{44}a_{22}a_{31} - a_{44}a_{21}a_{32}) = 0. \tag{13}$$

Table 1: Coefficients of implicit predictors P_i, $i = 2, 3, 4$ and integration formula of k-step Hermite–Birkhoff solver of order $p = 9, 10$.

k	7	8
coeffs\\p	9	10
a_{32}	-1.8268922342457146e-02	-1.2644364453523351e-02
b_5	3.8669248231767694e-01	3.5644917896211648e-01

The remaining of this paper is concerned with particular VS HB(9) and HB(10) with coefficients c_i, $i = 1, 2, \ldots, 5$ given by

$$\begin{aligned} c_1 = 0, \quad c_2 = 1.2791616119701035, \quad c_3 = 0.38776891003998121,\\ c_4 = 1.1997368881525279, \quad c_5 = 1.0, \end{aligned} \tag{14}$$

a_{32} and $a_{ii} = b_5$, $i = 2, 3, 4$ given in Table 1.

4.1. Integration formula IF

The $(p+1)$-vector of reordered coefficients of the integration formula IF in (3),

$$\boldsymbol{u}^1 = [\alpha_0, \alpha_1, \ldots, \alpha_{p-3}, b_4, b_3, b_2]^T,$$

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

$$M^1 \boldsymbol{u}^1 = \boldsymbol{r}^1, \tag{15}$$

where

$$M^1 = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ 0 & \eta_2 & \eta_3 & \cdots & \eta_{p-2} & 1 & 1 & 1 \\ 0 & \frac{\eta_2^2}{2!} & \frac{\eta_3^2}{2!} & \cdots & \frac{\eta_{p-2}^2}{2!} & c_4 & c_3 & c_2 \\ \vdots & & & & & & & \vdots \\ 0 & \frac{\eta_2^p}{p!} & \frac{\eta_3^p}{p!} & \cdots & \frac{\eta_{p-2}^p}{p!} & \frac{c_4^{p-1}}{(p-1)!} & \frac{c_3^{p-1}}{(p-1)!} & \frac{c_2^{p-1}}{(p-1)!} \end{bmatrix}, \tag{16}$$

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

$$\begin{aligned} r_1(1) &= 1, \\ r_1(i) &= \frac{1}{(i-1)!} - b_5 \frac{c_5^{i-2}}{(i-2)!}, \qquad i = 2, 3, \ldots, p+1. \end{aligned}$$

The leading error term of IF is

$$\left[b_5\frac{c_5^p}{p!}+\sum_{j=1}^{p-3}\alpha_j\frac{\eta_{j+1}^{p+1}}{(p+1)!}+\sum_{j=2}^{4}b_j\frac{c_j^p}{p!}-\frac{1}{(p+1)!}\right]h_{n+1}^{p+1}y_n^{p+1}.$$

4.2. Predictor P_2

The $(p-1)$-vector of reordered coefficients of the predictor P_2 in (2) with $i=2$,

$$\boldsymbol{u}^2=[\alpha_{20},a_{21},\alpha_{21},\dots,\alpha_{2,p-3}]^T,$$

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

$$M^2\boldsymbol{u}^2=\boldsymbol{r}^2, \tag{17}$$

where

$$M^2=\begin{bmatrix}1 & 0 & 1 & 1 & \cdots & 1\\ 0 & 1 & \eta_2 & \eta_3 & \cdots & \eta_{p-2}\\ 0 & 0 & \frac{\eta_2^2}{2!} & \frac{\eta_3^2}{2!} & \cdots & \frac{\eta_{p-2}^2}{2!}\\ \vdots & & & & & \vdots\\ 0 & 0 & \frac{\eta_2^{p-2}}{(p-2)!} & \frac{\eta_3^{p-2}}{(p-2)!} & \cdots & \frac{\eta_{p-2}^{p-2}}{(p-2)!}\end{bmatrix}, \tag{18}$$

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

$$\begin{aligned} r_2(1)&=1,\\ r_2(2)&=c_2-a_{22},\\ r_2(i)&=\frac{c_2^{i-1}}{(i-1)!}-a_{22}\frac{c_2^{i-2}}{(i-2)!}, \qquad i=3,4,\dots,p-1.\end{aligned}$$

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

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

with coefficients

$$
\begin{aligned}
S_2(j) &= a_{22}\frac{c_2^{j-1}}{(j-1)!} + M^2(j+1, 1:p-1)\boldsymbol{u}^2\\
&= a_{22}\frac{c_2^{j-1}}{(j-1)!} + r_2(j+1) = \frac{c_2^j}{j!}, \qquad j = 1, 2, \dots, p-2,\\
S_2(j) &= a_{22}S_2(j-1) + \sum_{i=1}^{p-3}\alpha_{2i}\frac{\eta_{i+1}^j}{j!}, \qquad j = p-1, p, p+1.
\end{aligned}
$$

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

$$
S_2(j) = c_2^j/j!, \qquad j = 0, 1, \dots, p-2,
$$

and its leading error term is

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

4.3. Predictor P_3

We consider the $(p+1)$-vector of coefficients of the predictor P_3 in (2) with $i = 3$,

$$
\widetilde{\boldsymbol{u}}^3 = [a_{33}, \alpha_{30}, \alpha_{31}, \dots, \alpha_{3,p-3}, a_{31}, a_{32}]^T.
$$

By setting $a_{33} = b_5$ and a_{32} equal to the values in Table 1, $\widetilde{\boldsymbol{u}}^3$ reduces to the $(p-1)$-vector $\boldsymbol{u}^3$ which is the solution of the system of order conditions

$$
M^3\boldsymbol{u}^3 = \boldsymbol{r}^3, \tag{19}
$$

where $M^3 = M^2$ and $\boldsymbol{r}^3 = r_3(1:p-1)$ has components

$$
\begin{aligned}
r_3(1) &= 1,\\
r_3(i) &= \frac{c_3^{i-1}}{(i-1)!} - a_{33}\frac{c_3^{i-2}}{(i-2)!} - a_{32}\frac{c_2^{i-2}}{(i-2)!}, \qquad i = 2, 3, \dots, p-1.
\end{aligned}
$$

A truncated Taylor expansion of the right-hand side of (2), with $i = 3$, about t_n gives

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

with coefficients

$$S_3(j) = a_{33}\frac{c_3^{j-1}}{(j-1)!} + a_{32}\frac{c_2^{j-1}}{(j-1)!} + M^3(j+1, 1:p-1)\boldsymbol{u}^3$$
$$= a_{33}\frac{c_3^{j-1}}{(j-1)!} + a_{32}\frac{c_2^{j-1}}{(j-1)!} + r_3(j+1) = \frac{c_3^j}{j!}, \qquad j = 1, 2, \dots, p-2,$$
$$S_3(j) = a_{33}S_3(j-1) + a_{32}S_2(j-1) + \sum_{i=1}^{p-3}\alpha_{3i}\frac{\eta_{i+1}^j}{j!}, \qquad j = p-1, p, p+1.$$

4.4. Predictor P_4

The $(p+1)$-vector of reordered coefficients of the predictor P_4 in (2) with $i = 4$,

$$\boldsymbol{u}^4 = [\alpha_{40}, a_{41}, \alpha_{41}, \dots, \alpha_{4,p-3}, a_{43}, a_{42}]^T,$$

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

$$M^4\boldsymbol{u}^4 = \boldsymbol{r}^4, \tag{20}$$

where

$$M^4 = \begin{bmatrix} 1 & 0 & 1 & 1 & \cdots & 1 & 0 & 0 \\ 0 & 1 & \eta_2 & \eta_3 & \cdots & \eta_{p-2} & 1 & 1 \\ 0 & 0 & \frac{\eta_2^2}{2!} & \frac{\eta_3^2}{2!} & \cdots & \frac{\eta_{p-2}^2}{2!} & c_3 & c_2 \\ 0 & 0 & \frac{\eta_2^3}{3!} & \frac{\eta_3^3}{3!} & \cdots & \frac{\eta_{p-2}^3}{3!} & \frac{c_3^2}{2!} & \frac{c_2^2}{2!} \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \frac{\eta_2^{p-1}}{(p-1)!} & \frac{\eta_3^{p-1}}{(p-1)!} & \cdots & \frac{\eta_{p-2}^{p-1}}{(p-1)!} & \frac{c_3^{p-2}}{(p-2)!} & \frac{c_2^{p-2}}{(p-2)!} \\ 0 & a_{22}a_{33}b_4 & 0 & 0 & \cdots & 0 & (a_{21}a_{32}b_4 - a_{22}a_{31}b_4) & -a_{21}a_{33}b_4 \end{bmatrix}. \tag{21}$$

The first $(p-1)$ components of $\boldsymbol{r}^4 = r_4(1:p+1)$ are

$$r_4(1) = 1,$$
$$r_4(i) = \frac{c_4^{i-1}}{(i-1)!} - a_{44}\frac{c_4^{i-2}}{(i-2)!}, \qquad i = 2, 3, \dots, p-1,$$

the pth component is

$$r_4(p) = \frac{1}{b_4}\left[\frac{1}{p!} - b_2S_2(p-1) - b_3S_3(p-1) - b_5\frac{c_5^{p-1}}{(p-1)!} - B(p)\right] - a_{44}\frac{c_4^{p-2}}{(p-2)!}, \tag{22}$$

which corresponds to the order conditions (11), and the $(p+1)$th component is

$$r_4(p+1) = -a_{44}(a_{21}a_{33}b_2 - a_{21}a_{32}b_3 + a_{22}a_{31}b_3), \tag{23}$$

which corresponds to the condition (13).

4.5. Step control predictor $\mathbf{P}_5$

We consider the $(p+2)$-vector of the coefficients of predictor P_5 in (4),

$$\widetilde{\boldsymbol{u}}^5 = [a_{55}, \alpha_{50}, \alpha_{51}, \dots, \alpha_{5,p-3}, a_{52}, a_{53}, a_{54}]^T.$$

By setting $a_{55} = b_5 + \omega_5$, $a_{54} = b_4 + \omega_4$ and $a_{52} = b_2 + \omega_2$, $\widetilde{\boldsymbol{u}}^5$ reduces to the $(p-1)$-vector $\boldsymbol{u}^5$ which is the solution of the system of order conditions

$$M^5\boldsymbol{u}^5 = \boldsymbol{r}^5, \tag{24}$$

where

$$M^5 = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 0 \\ 0 & \eta_2 & \eta_3 & \cdots & \eta_{p-2} & 1 \\ 0 & \frac{\eta_2^2}{2!} & \frac{\eta_3^2}{2!} & \cdots & \frac{\eta_{p-2}^2}{2!} & c_3 \\ 0 & \frac{\eta_2^3}{3!} & \frac{\eta_3^3}{3!} & \cdots & \frac{\eta_{p-2}^3}{3!} & \frac{c_3^2}{2!} \\ \vdots & & & & & \vdots \\ 0 & \frac{\eta_2^{p-2}}{(p-2)!} & \frac{\eta_3^{p-2}}{(p-2)!} & \cdots & \frac{\eta_{p-2}^{p-2}}{(p-2)!} & \frac{c_3^{p-3}}{(p-3)!} \end{bmatrix}, \tag{25}$$

and $\boldsymbol{r}^5 = r_5(1:p-1)$ has components

$$\begin{aligned} r_5(1) &= 1, \\ r_5(i) &= \frac{1}{(i-1)!} - (b_5+\omega_5)\frac{c_5^{i-2}}{(i-2)!} - (b_4+\omega_4)\frac{c_4^{i-2}}{(i-2)!} - (b_2+\omega_2)\frac{c_2^{i-2}}{(i-2)!}, \\ &\qquad i = 2, 3, \dots, p-1. \end{aligned}$$

For arbitrary nonzero ω_4 and ω_2, P_5 yields $\widetilde{y}_{n+1}$ to order $(p-2)$. A good experimental choice is $\omega_5 = 0.025$, $\omega_4 = 0.025$ and $\omega_2 = -10^{-12}$.

The solutions $\boldsymbol{u}^\ell$, $\ell = 1, 2, \dots, 5$, form generalized Lagrange basis functions for representing the HB interpolation polynomials.

5. Symbolic construction of elementary matrix functions

Consider the matrices

$$M^{\ell} \in \mathbb{R}^{m_\ell \times m_\ell}, \qquad \ell = 1, 2, 4, 5, \tag{26}$$

of the Vandermonde-type systems (15), (17), (20) and (24), where

$$m_1 = p+1, \qquad m_2 = p-1, \qquad m_4 = p+1, \qquad m_5 = p-1, \tag{27}$$

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 HB(p). These matrices are most easily constructed by means of a symbolic software. These functions will be used in Section 6 to factor

- M^1 into a diagonal+last-three-column matrix, W_3^1, which will be further diagonalized by a Gaussian elimination,
- M^2 into the identity matrix,
- M^4 into a diagonal+last-row+last-two-column matrix W_3^4, which will be further diagonalized by a Gaussian elimination,
- M^5 into a diagonal+last-column matrix W_1^5 which will be further diagonalized by a Gaussian elimination.

This decomposition will lead to a fast solution of the systems $M^{\ell}\boldsymbol{u}^{\ell} = \boldsymbol{r}^{\ell}$, $\ell = 1, 2, 4, 5$ 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 [3], Krogh [18], Galimberti and Pereyra [12] and Björck and Elfving [2]. 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}, $\ell = 1, 2, 4, 5$

5.1.1. Symbolic construction of lower bidiagonal matrices for M^{ℓ}, $\ell = 1, 5$

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

$$L_k = \begin{bmatrix} I_{k-1} & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & -\tau_{k+1} & 1 & & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & -\tau_m & 1 \end{bmatrix}, \tag{28}$$

defined by the multipliers

$$\tau_i = \frac{x_i}{x_{i-1}} = -L_k(i, i-1), \qquad i = k+1, k+2, \dots, m, \tag{29}$$

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, $\ell = 1, 5$, 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

$$L^1 M^1 = L_{m_1-3}^1 \cdots L_3^1 L_2^1 M^1, \tag{30}$$

$$L^5 M^5 = L_{m_5-1}^5 \cdots L_3^5 L_2^5 M^5, \tag{31}$$

in $(m_1 - 4)$ and $(m_5 - 2)$ steps respectively.

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.

With $\ell = 1$, the input is $M = M^1$; $m = m_1$. The output is $L_k = L_k^1$, $k = 2, 3, \dots, m_1 - 3$.

With $\ell = 5$, the input is $M = M^5$; $m = m_5$. The output is $L_k = L_k^5$, $k = 2, 3, \dots, m_5 - 1$.

5.1.2. Symbolic construction of lower bidiagonal matrices for M^2

The zeroing process by means of lower bidiagonal matrix (28) defined by the multipliers (29) will be applied recursively on M^2, as follows. For $k = 3, 4, \dots, m_2 - 1$, left multiplying $T_k^2 = L_{k-1}^2 L_{k-2}^2 \cdots L_4^2 L_3^2 M^2$ by L_k^2 zeros the last $(m_2 - k)$ components of the kth column of T_k^2. Thus we obtain the upper triangular matrix

$$L^2 M^2 = L_{m_2-1}^2 \cdots L_4^2 L_3^2 M^2, \tag{32}$$

in $(m_2 - 3)$ steps respectively.

We note that L^2 does not change the first three rows of M^2.

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

- $M^{2(k-1)} = L^2_{k-1}L^2_{k-2}\cdots L^2_3 M^2$ is an upper triangular matrix in columns 1 to $k-1$.
- The multipliers in L^2_k are obtained from $M^{2(k-1)}(k{+}1 : m_2, k)$ since $M^2(i,k) \neq 0$ for $i = k+1, k+2, \dots, m_2$.

Algorithm 1 in Appendix A describes this process.

The input is $M = M^2$; $m = m_2$. The output is $L_k = L^2_k$, $k = 3, 4, \dots, m_2 - 1$.

5.1.3. Symbolic construction of lower bidiagonal matrices for M^4

We first describe the zeroing process of a general vector $\boldsymbol{x} = [x_1, x_2, \dots, x_{m-1}, 0]^T$ whose first $(m-1)$ components are non zero elements. The lower bidiagonal matrix

$$L_k = \begin{bmatrix} I_{k-1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & & 0 & 0 \\ 0 & -\tau_{k+1} & 1 & & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & -\tau_{m-1} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{33}$$

defined by the multipliers

$$\tau_i = \frac{x_i}{x_{i-1}} = -L_k(i, i-1), \qquad i = k+1, k+2, \dots, m-1, \tag{34}$$

zeros the last $(m-1-k)$ components, $x_{k+1}, x_{k+2}, \dots, x_{m-1}$, of $\boldsymbol{x}$. This zeroing process will be applied recursively on M^4 as follows. For $k = 3, 4, \dots, m_4 - 2$, left multiplying $T^4_k = L^4_{k-1}\cdots L^4_4 L^4_3 M^4$ by L^4_k zeros the last $(m_4 - 1 - k)$ components of the kth column of T^4_k. Thus we obtain the upper triangular matrix in row 1 to $m_4 - 1$ and in column 1 to $m_4 - 2$,

$$L^4 M^4 = L^4_{m_4-2}\cdots L^4_4 L^4_3 M^4, \tag{35}$$

in $(m_4 - 4)$ steps.

We note that L^4 does not change the first three rows and the last row of M^4.

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

- $M^{4(k-1)} = L^4_{k-1}L^4_{k-2}\cdots L^4_3 M^4$ is an upper triangular matrix in row 1 to $m_4 - 1$ and in columns 1 to $k-1$.

- The multipliers in L_k^4 are obtained from $M^{4(k-1)}(k+1 : m_4-1, k)$ since $M^4(i,k) \neq 0$ for $i = k+1, k+2, \dots, m_4-1$.

Algorithm 1 in Appendix A describes this process.

The input is $M = M^4$; $m = m_4$. The output is $L_k = L_k^4$, $k = 3, 4, \dots, m_4-2$.

5.2. Symbolic construction of elementary upper triangular matrices

5.2.1. Symbolic construction of upper bidiagonal matrices for M^1

For matrix L^1M^1, we construct recursively upper bidiagonal matrices $U_1^1, U_2^1 \dots, U_{m_1-4}^1$ such that right multiplying L^1M^1 by the upper triangular matrix $U^1 = U_1^1U_2^1 \cdots U_{m_1-4}^1$ transforms L^1M^1 into a matrix $W_3^1 = L^1M^1U^1$ with nonzero diagonal elements, $W_3^1(i,i) \neq 0$, $i = 1, 2, \dots, m_1$, the last three nonzero columns $W_3^1(1 : m_1, j) \neq 0$, $j = m_1-2, m_1-1, m_1$, and zero elsewhere. We call such a matrix a "diagonal+last-three-column matrix" matrix.

We describe the zeroing process of the upper bidiagonal matrix U_k^1 on the two-row matrix $(L^1M^1)(k : k+1, 1 : m_1)$:

$$(L^1M^1)(k : k+1, 1 : m_1)U_1^1U_2^1 \cdots U_{k-1}^1$$
$$= \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & \cdots & 1 & y_{k,m_1-2} & y_{k,m_1-1} & y_{k,m_1} \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & \cdots & y_{k+1,m_1-3} & y_{k+1,m_1-2} & y_{k+1,m_1-1} & y_{k+1,m_1} \end{bmatrix}. \tag{36}$$

The divisors

$$\sigma_i = \frac{1}{y_{k+1,i} - y_{k+1,i-1}} = U_k^1(i,i), \qquad i = k+1, k+2, \dots, m_1-3, \tag{37}$$

define the upper bidiagonal matrix

$$U_k^1 = \begin{bmatrix} I_{k-1} & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & -\sigma_{k+1} & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{k+1} & -\sigma_{k+2} & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & & \ddots & \ddots & & & & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{m_1-4} & -\sigma_{m_1-3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{m_1-3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{38}$$

Right multiplying (36) by U_k^1 zeros the 1's in position $k+1, k+2, \dots, m_1-3$ in the first row and puts 1's in position $k+1, k+2, \dots, m_1-3$ in the second row:

$$(L^1M^1)(k : k+1, 1 : m_1)U_1^1U_2^1 \cdots U_{k-1}^1U_k^1$$
$$= \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 0 & \cdots & 0 & y_{k,m_1-2} & y_{k,m_1-1} & y_{k,m_1} \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & 1 & \cdots & 1 & y_{k+1,m_1-2} & y_{k+1,m_1-1} & y_{k+1,m_1} \end{bmatrix}. \tag{39}$$

Thus, $U^1 = U_1^1 U_2^1 \cdots U_{m_1-4}^1$ transforms the upper triangular matrix L^1M^1 into the diagonal+last-three-column matrix

$$W_3^1 = L^1 M^1 U_1^1 U_2^1 \cdots U_{m_1-4}^1 \tag{40}$$

in $(m_1 - 4)$ steps.

Process 4. At the kth step, starting with $k = 1$,

- $M^{1(k)} = L^1 M^1 U_1^1 U_2^1 \cdots U_k^1$ is a diagonal+last-three-column matrix in rows 1 to k.
- The divisors in U_k^1 are obtained from $M^{1(k-1)}(k+1, k : m_1-3)$ since $M^{1(k-1)}(k+1, j) - M^{1(k-1)}(k+1, j-1) \neq 0$, $j = k+1, k+2, \dots, m_1 - 3$.

Algorithm 2 in Appendix A describes this process for M^1. The input is $M = M^1$; $m = m_1$. The output is $U_k = U_k^1$, $k = 1, 2, \dots, m_1 - 4$.

The next two subsections, Subsection 5.2.2 and Subsection 5.2.3 describe the construction of elementary upper triangular matrices U_k^2, $k = 1, 2, \dots, m_2 - 1$ for M^2.

5.2.2. Construction of initializing upper tridiagonal matrix U_1^2 for M^2

For matrix L^2M^2, first, we construct the initializing upper tridiagonal matrix U_1^2 such that right multiplying L^2M^2 by U_1^2 transforms L^2M^2 into a matrix whose first two rows are of the form:

$$(L^2 M^2 U_1^2)(1:2, 1:m_2) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ y_{21} & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}. \tag{41}$$

We describe the zeroing process of the upper tridiagonal matrix U_1^2 on the first-two-row matrix $(L^2M^2)(1 : 2, 1 : m_2)$:

$$(L^2 M^2)(1:2, 1:m_2) = \begin{bmatrix} 1 & 0 & 1 & \cdots & 1 \\ y_{21} & 1 & y_{23} & \cdots & y_{2,m_2} \end{bmatrix}. \tag{42}$$

The divisors

$$\sigma_3 = \frac{1}{y_{2,3} - y_{2,1}} = U_1^2(3,3), \qquad \sigma_i = \frac{1}{y_{2,i} - y_{2,i-1}} = U_1^2(i,i), \quad i = 4, 5, \dots, m_2, \tag{43}$$

define the elementary upper tridiagonal matrix U_1^2 of the form

$$U_k^1 = \begin{bmatrix} 1 & 0 & -\sigma_3 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & -\sigma_4 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \sigma_4 & -\sigma_5 & \cdots & 0 \\ \vdots & \vdots & & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \sigma_{m_2-1} & -\sigma_{m_2} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \sigma_{m_2} \end{bmatrix}. \tag{44}$$

Right multiplying (42) by U_1^2 zeros the 1's in position $3, 4, \ldots, m_2$ in the first row and puts 1's in position $3, 4, \ldots, m_2$ in the second row: the resulting $(L^2M^2)(1 : 2, 1 : m_2)U_1^2$ is of the form (41).

5.2.3. Symbolic construction of upper bidiagonal matrices for M^2

For matrix L^2M^2, we construct recursively upper bidiagonal matrices $U_2^2, U_3^2, \ldots, U_{m_2-1}^2$ such that right multiplying L^2M^2 by the upper triangular matrix $U^2 = U_1^2U_2^2 \cdots U_{m_2-1}^2$ transforms L^2M^2 into the identity matrix $I^2 = L^2M^2U^2$.

We describe the zeroing process of the upper bidiagonal matrix U_k^2 on the two-row matrix $(L^2M^2)(k : k+1, 1 : m_2)$:

$$(L^2M^2)(k : k+1, 1 : m_2)U_1^2U_2^2 \cdots U_{k-1}^2 = \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 1 & \cdots & 1 \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & y_{k+1,k+1} & \cdots & y_{k+1,m_2} \end{bmatrix}. \quad (45)$$

The divisors

$$\sigma_i = \frac{1}{y_{k+1,i} - y_{k+1,i-1}} = U_k^2(i,i), \qquad i = k+1, k+2, \ldots, m_2, \quad (46)$$

define the upper bidiagonal matrix

$$U_k^2 = \begin{bmatrix} I_{k-1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & -\sigma_{k+1} & 0 & \cdots & 0 \\ 0 & 0 & \sigma_{k+1} & -\sigma_{k+2} & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{m_2-1} & -\sigma_{m_2} \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{m_2} \end{bmatrix}. \quad (47)$$

Right multiplying (45) by U_k^2 zeros the 1's in position $k+1, k+2, \ldots, m_2$ in the first row and puts 1's in position $k+1, k+2, \ldots, m_2$ in the second row:

$$(L^2M^2)(k : k+1, 1 : m_2)U_1^2U_2^2 \cdots U_{k-1}^2U_k^2 = \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 0 & \cdots & 0 \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & 1 & \cdots & 1 \end{bmatrix}. \quad (48)$$

Thus, right multiplying L^2M^2 by the upper triangular matrix $U^2 = U_1^2U_2^2 \cdots U_{m_2-1}^2$ transforms the upper triangular matrix L^2M^2 into the identity matrix

$$I^2 = L^2M^2U_1^2U_2^2 \cdots U_{m_2-1}^2 \quad (49)$$

in $(m_2 - 1)$ steps.

Process 5. At the kth step, starting with $k = 1$,

- $M^{2(k)} = L^2 M^2 U_1^2 U_2^2 \cdots U_k^2$ is the identity matrix in rows 1 to k.
- The divisors in U_k^2 are obtained from $M^{2(k-1)}(k+1, k : m_2)$ since $M^{2(k-1)}(k+1, j) - M^{2(k-1)}(k+1, j-1) \neq 0$, $j = k+1, k+2, \dots, m_2$.

Algorithm 3 in Appendix A describes this process for M^2. The input is $M = M^2$; $m = m_2$. The output is $U_k = U_k^2$, $k = 1, 2, \dots, m_2 - 1$.

The next two subsections, Subsection 5.2.4 and Subsection 5.2.5 describe the construction of elementary upper triangular matrices U_k^4, $k = 1, 2, \dots, m_4 - 3$ for M^4.

5.2.4. Construction of initializing upper tridiagonal matrix U_1^4 for M^4

For matrix L^4M^4, first, we construct the initializing upper tridiagonal matrix U_1^4 such that right multiplying L^4M^4 by U_1^4 transforms L^4M^4 into a matrix whose first two rows are of the form:

$$(L^4M^4U_1^4)(1:2, 1:m_4) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ y_{21} & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}. \tag{50}$$

We describe the zeroing process of the upper tridiagonal matrix U_1^4 on the first-two-row matrix $(L^4M^4)(1:2, 1:m_4)$:

$$(L^4M^4)(1:2, 1:m_4) = \begin{bmatrix} 1 & 0 & 1 & \cdots & 1 & 0 & 0 \\ y_{21} & 1 & y_{23} & \cdots & y_{2,m_4-2} & 1 & 1 \end{bmatrix}. \tag{51}$$

The divisors

$$\sigma_3 = \frac{1}{y_{2,3} - y_{2,1}} = U_1^4(3,3), \qquad \sigma_i = \frac{1}{y_{2,i} - y_{2,i-1}} = U_1^4(i,i), \quad i = 4, 5, \dots, m_4 - 2, \tag{52}$$

define the elementary upper tridiagonal matrix U_1^4 of the form

$$U_k^1 = \begin{bmatrix} 1 & 0 & -\sigma_3 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & -\sigma_4 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_4 & -\sigma_5 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & & & \ddots & \ddots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \sigma_{m_4-3} & -\sigma_{m_4-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \sigma_{m_4-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{53}$$

Right multiplying (51) by U_1^4 zeros the 1's in position $3, 4, \dots, m_4 - 2$ in the first row and puts 1's in position $3, 4, \dots, m_4 - 2$ in the second row: the resulting $(L^4M^4)(1:2, 1:m_4)U_1^4$ is of the form (50).

5.2.5. Symbolic construction of elementary upper bidiagonal matrices for M^4

For matrix L^4M^4, we construct recursively upper bidiagonal matrices $U_2^4, U_3^4, \ldots, U_{m_4-3}^4$ such that right multiplying L^4M^4 by the upper triangular matrix $U^4 = U_1^4U_2^4 \cdots U_{m_4-3}^4$ transforms L^4M^4 into a matrix $W_3^4 = L^4M^4U^4$ with nonzero diagonal elements, $W_3^4(i,i) \neq 0$, $i = 1, 2, \ldots, m_4$, the last nonzero row $W_3^4(m_4, j)$, $j = 1, 2, \ldots, m_4 - 1$, the last two nonzero columns $W_3^4(1 : m_4, j) \neq 0$, $j = m_4 - 1, m_4$, and zero elsewhere. We call such a matrix a "diagonal+last-row+last-two-column" matrix. Here the last nonzero row $W_3^4(m_4, j)$, $j = 1, 2, \ldots, m_4$ contains some nonzero entries and zero elsewhere.

We describe the zeroing process of the upper bidiagonal matrix U_k^4 on the two-row matrix $(L^4M^4)(k : k+1, 1 : m_4)$:

$$
\begin{aligned}
&(L^4M^4)(k : k+1, 1 : m_4)U_1^4U_2^4 \cdots U_{k-1}^4 \\
&\qquad = \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & \cdots & 1 & y_{k,m_4-1} & y_{k,m_4} \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & \cdots & y_{k+1,m_4-2} & y_{k+1,m_4-1} & y_{k+1,m_4} \end{bmatrix}. \qquad (54)
\end{aligned}
$$

The divisors

$$
\sigma_i = \frac{1}{y_{k+1,i} - y_{k+1,i-1}} = U_k^4(i,i), \qquad i = k+1, k+2, \ldots, m_4 - 2, \tag{55}
$$

define the upper bidiagonal matrix

$$
U_k^4 = \begin{bmatrix}
I_{k-1} & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 \\
0 & 1 & -\sigma_{k+1} & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \sigma_{k+1} & -\sigma_{k+2} & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & & \ddots & \ddots & & & \vdots \\
0 & 0 & 0 & \cdots & \sigma_{m_4-3} & -\sigma_{m_4-2} & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & \sigma_{m_4-2} & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1
\end{bmatrix}. \tag{56}
$$

Right multiplying (54) by U_k^4 zeros the 1's in position $k+1, k+2, \ldots, m_4 - 2$ in the first row and puts 1's in position $k+1, k+2, \ldots, m_4 - 2$ in the second row:

$$
\begin{aligned}
&(L^4M^4)(k : k+1, 1 : m_4)U_1^4U_2^4 \cdots U_{k-1}^4U_k^4 \\
&\qquad = \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 0 & \cdots & 0 & y_{k,m_4-1} & y_{k,m_4} \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & 1 & \cdots & 1 & y_{k+1,m_4-1} & y_{k+1,m_4} \end{bmatrix}. \qquad (57)
\end{aligned}
$$

Thus, $U^4 = U_1^4U_2^4 \cdots U_{m_4-3}^4$ transforms the upper triangular matrix L^4M^4 into the diagonal+last-row+last-two-column matrix

$$
W_3^4 = L^4M^4U_1^4U_2^4 \cdots U_{m_4-3}^4 \tag{58}
$$

in (m_4-3) steps.
Process 6. At the kth step, starting with $k=1$,

- $M^{4(k)} = L^4M^4U_1^4U_2^4\cdots U_k^4$ is a diagonal+last-row+last-two-column matrix in rows 1 to k.
- The divisors in U_k^4 are obtained from $M^{4(k-1)}(k+1, k : m_4-2)$ since $M^{4(k-1)}(k+1, j) - M^{4(k-1)}(k+1, j-1) \neq 0$, $j = k+1, k+2, \dots, m_4-2$.

Algorithm 3 in Appendix A describes this process for M^4. The input is $M = M^4$; $m = m_4$. The output is $U_k = U_k^4$, $k = 1, 2, \dots, m_4-3$.

5.2.6. Symbolic construction of upper bidiagonal matrices for M^5

For matrix L^5M^5, we construct recursively upper bidiagonal matrices $U_1^5, U_2^5 \dots, U_{m_5-2}^5$ such that right multiplying L^5M^5 by the upper triangular matrix $U^5 = U_1^5U_2^5\cdots U_{m_5-2}^5$ transforms L^5M^5 into a matrix $W_1^5 = L^5M^5U^5$ with nonzero diagonal elements, $W_1^5(i,i) \neq 0$, $i = 1, 2, \dots, m_5$, nonzero $W_1^5(1 : m_5, m_5) \neq 0$, 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^5 on the two-row matrix $(L^5M^5)(k : k+1, 1 : m_5)$:

$$(L^5M^5)(k : k+1, 1 : m_5)U_1^5U_2^5\cdots U_{k-1}^5$$
$$= \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 1 & \cdots & 1 & 1 & y_{k,m_5} \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & y_{k+1,k+1} & \cdots & y_{k+1,m_5-2} & y_{k+1,m_5-1} & y_{k+1,m_5} \end{bmatrix}. \quad (59)$$

The divisors

$$\sigma_i = \frac{1}{y_{k+1,i} - y_{k+1,i-1}} = U_k^5(i,i), \qquad i = k+1, k+2, \dots, m_5-1, \quad (60)$$

define the upper bidiagonal matrix

$$U_k^5 = \begin{bmatrix} I_{k-1} & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -\sigma_{k+1} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \sigma_{k+1} & -\sigma_{k+2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & & \ddots & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{m_5-3} & -\sigma_{m_5-2} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{m_5-2} & -\sigma_{m_5-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{m_5-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (61)$$

Right-multiplying (59) by U_k^5 zeros the 1's in position $k+1, k+2, \dots, m_5-1$ in the first row and puts 1's in position $k+1, k+2, \dots, m_5-1$ in the second row:

$$(L^5M^5)(k : k+1, 1 : m_5)U_1^5U_2^5U_3^5\cdots U_{k-1}^5U_k^5$$
$$= \begin{bmatrix} y_{k1} & \cdots & y_{k,k-1} & 1 & 0 & \cdots & 0 & 0 & y_{k,m_5} \\ y_{k+1,1} & \cdots & y_{k+1,k-1} & y_{k+1,k} & 1 & \cdots & 1 & 1 & y_{k+1,m_5} \end{bmatrix}. \quad (62)$$

Thus, $U_1^5 U_2^5 \cdots U_{m_5-2}^5$ transforms the upper triangular matrix $L^5 M^5$ into a diagonal+last-column matrix

$$W_1^5 = (L^5 M^5) U_1^5 U_2^5 \cdots U_{m_5-2}^5 \tag{63}$$

in $(m_5 - 2)$ steps.

Process 7. At the kth step, starting with $k = 1$,

- $M^{5(k)} = (L^5 M^5) U_1^5 U_2^5 \cdots U_k^5$ is a diagonal+last-column matrix in rows 1 to k.
- The divisors in U_k^5 are obtained from $M^{5(k-1)}(k+1, k : m_5-1)$ since $M^{5(k-1)}(k+1, j) - M^{5(k-1)}(k+1, j-1) \neq 0$, $j = k+1, k+2, \ldots, m_5 - 1$.

Algorithm 2 in Appendix A describes this process. The input is $M = M^5$; $m = m_5$. The output is $U_k = U_k^5$, $k = 1, 2, \ldots, m_5 - 2$.

6. Fast solution of Vandermonde-type systems for particular HB(p)

Symbolic elementary matrix functions L_k^ℓ and U_k^ℓ, $\ell = 1, 2, 4, 5$, are constructed once as functions of η_j, for $j = 2, 3, \ldots, p-2$ by Algorithms 1, 2 and 3 in Appendix A:

- Algorithms 1 and 2 to factor M^1 into a diagonal+last-three-column matrix, W_3^1, which will be further diagonalized by a Gaussian elimination,
- Algorithms 1 and 3 to factor M^2 into the identity matrix,
- Algorithms 1 and 3 to factor M^4 into a diagonal+last-row+last-two-column matrix, W_3^4, which will be further diagonalized by a Gaussian elimination,
- Algorithms 1 and 2 to factor M^5 into a diagonal+last-column matrix, W_1^5, which will be further diagonalized by a Gaussian elimination.

These elementary matrix functions are used, first, to find the solution $\boldsymbol{u}^\ell$, $\ell = 1, 2, 4, 5$ in elementary matrix functions form and, then, to construct fast Algorithms 4, 5, 6 and 7, in Appendix A, to solve systems (15), (17), (20) and (24) at each integration step.

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

We let $m_1 = p + 1$ as defined in (27).

(1) The elimination procedure of Subsection 5.1.1 is applied to M^1 to construct $m_1 \times m_1$ lower bidiagonal matrices L_k^1, $k = 2, 3, \ldots, m_1 - 3$, with multipliers

$$\tau_i = \frac{M^1(2, k)}{i - 1} = -L_k^1(i, i-1), \qquad i = k+1, k+2, \ldots, m_1. \tag{64}$$

Left multiplying the coefficient matrix M^1 by the lower triangular matrix $L^1 = L^1_{m_1-3} \cdots L^1_3 L^1_2$ transforms M^1 into the upper triangular matrix $L^1 M^1$ in column 1 to $m_1 - 3$ of the form (30).

(2) The elimination procedure of Subsection 5.2.1 is used to construct $m_1 \times m_1$ upper bidiagonal matrices U^1_k, $k = 1, 2, \dots, m_1 - 4$, with multipliers

$$\sigma_i = \frac{k}{M^1(2,i) - M^1(2,i-k)} = U^1_k(i,i), \qquad i = k+1, k+2, \dots, m_1 - 3. \tag{65}$$

Right multiplying $L^1 M^1$ by the upper triangular matrix $U^1 = U^1_1 U^1_2 \cdots U^1_{m_1-4}$ transforms $L^1 M^1$ into a diagonal+last-three-column matrix W^1_3 of the form (40).

(3) A factored Gaussian elimination, $L^1_{m_1-1} L^1_{m_1-2}$, will transform W^1_3 into a diagonal+last-two-column matrix $W^1_2 = L^1_{m_1-1} L^1_{m_1-2} W^1_3$ as follows. First, $W^1_3(m_1-2, m_1-2)$ is set to 1 by the diagonal matrix $L^1_{m_1-2}$ whose entries are zeros, except for,

$$\begin{aligned} L^1_{m_1-2}(i,i) &= 1, \qquad i = 1, 2, \dots, m_1 - 3, \\ L^1_{m_1-2}(m_1-2, m_1-2) &= 1/W^1_3(m_1-2, m_1-2), \\ L^1_{m_1-2}(i,i) &= 1, \qquad i = m_1 - 1, m_1. \end{aligned}$$

Then the non-diagonal entries in the column $m_1 - 2$ of $L^1_{m_1-2} W^1_3$ are zeroed by the unit diagonal+column-(m_1-2) matrix $L^1_{m_1-1}$ whose entries are zeros, except for,

$$\begin{aligned} L^1_{m_1-1}(1:m_1-3, m_1-2) &= -W^1_3(1:m_1-3, m_1-2), \\ L^1_{m_1-1}(i,i) &= 1, \qquad i = 1, 2, \dots, m_1, \\ L^1_{m_1-1}(i, m_1-2) &= -W^1_3(i, m_1-2), \qquad i = m_1 - 1, m_1. \end{aligned}$$

(4) A factored Gaussian elimination, $L^1_{m_1+1} L^1_{m_1}$, will transform W^1_2 into a diagonal+last-column matrix $W^1_1 = L^1_{m_1+1} L^1_{m_1} W^1_2$ as follows. First, $W^1_2(m_1-1, m_1-1)$ is set to 1 by the diagonal matrix $L^1_{m_1}$ whose entries are zeros, except for,

$$\begin{aligned} L^1_{m_1}(i,i) &= 1, \qquad i = 1, 2, \dots, m_1 - 2, \\ L^1_{m_1}(m_1-1, m_1-1) &= 1/W^1_2(m_1-1, m_1-1), \\ L^1_{m_1}(m_1, m_1) &= 1. \end{aligned}$$

Then the non-diagonal entries in column $m_1 - 1$ of $L^1_{m_1} W^1_2$ are zeroed by the unit diagonal+column-(m_1-1) matrix $L^1_{m_1+1}$ whose entries are zeros, except for,

$$\begin{aligned} L^1_{m_1+1}(1:m_1-2, m_1-1) &= -W^1_2(1:m_1-2, m_1-1),\\ L^1_{m_1+1}(i,i) &= 1, \qquad i=1,2,\dots,m_1,\\ L^1_{m_1+1}(m_1, m_1-1) &= -W^1_2(m_1, m_1-1).\end{aligned}$$

(5) A factored Gaussian elimination, $L^1_{m_1+3}L^1_{m_1+2}$, will transform W^1_1 into the identity matrix $I^1 = L^1_{m_1+3}L^1_{m_1+2}W^1_1$ as follows. First, $W^1_1(m_1,m_1)$ is set to 1 by the diagonal matrix $L^1_{m_1+2}$ whose entries are zeros, except for,

$$\begin{aligned} L^1_{m_1+2}(i,i) &= 1, \qquad i=1,2,\dots,m_1-1,\\ L^1_{m_1+2}(m_1,m_1) &= 1/W^1_1(m_1,m_1).\end{aligned}$$

Then the non-diagonal entries in the last column of $L^1_{m_1+2}W^1_1$ are zeroed by the unit diagonal+last-column matrix $L^1_{m_1+3}$ whose entries are zeros, except for,

$$\begin{aligned} L^1_{m_1+3}(i,i) &= 1, \qquad i=1,2,\dots,m_1,\\ L^1_{m_1+3}(1:m_1-1, m_1) &= -W^1_1(1:m_1-1, m_1).\end{aligned}$$

We now obtain the following procedure which transforms M^1 into the identity matrix

$$I^1 = L^1_{m_1+3}L^1_{m_1+2}\cdots L^1_2 M^1 U^1_1 U^1_2\cdots U^1_{m_1-4}.$$

Thus we have the following factorization of M^1 into the product of elementary matrices:

$$M^1 = \left(L^1_{m_1+3}L^1_{m_1+2}\cdots L^1_2\right)^{-1}\left(U^1_1U^1_2\cdots U^1_{m_1-4}\right)^{-1},$$

and the solution is

$$\boldsymbol{u}^1 = U^1_1U^1_2\cdots U^1_{m_1-4}L^1_{m_1+3}L^1_{m_1+2}\cdots L^1_2\,\boldsymbol{r}^1, \tag{66}$$

where fast computation goes from right to left.

Procedure (66) is implemented in Algorithm 4 in Appendix A in $O(m_1^2)$ operations. The input is $M = M^1$; $m = m_1$; $\boldsymbol{r} = \boldsymbol{r}^1$; $L_k = L^1_k$, $k = 2,3,\dots,m_1+3$; $U_k = U^1_k$, $k = 1,2,\dots,m_1-4$. The output is $\boldsymbol{u} = \boldsymbol{u}^1$.

It is to be noted that, by using Algorithm 2, the new $\sigma_i = \frac{k}{M^1(2,i)-M^1(2,i-k)} = U^1_k(i,i)$ in (65) is found for integration formula IF instead of $\sigma_i = \frac{1}{M^1(2,i)-M^1(2,i-k)} = U^1_k(i,i)$ of the usual Newton divided differences. Similar result is found for predictor P_i, $i = 2,3,4$.

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

We let $m_2 = p - 1$ as defined in (27).

(1) The elimination procedure of Subsection 5.1.2 is applied to M^2 to construct $m_2 \times m_2$ lower bidiagonal matrices L_k^2, $k = 3, 4, \dots, m_2 - 1$, with multipliers

$$\tau_i = \frac{M^2(2,k)}{i-1} = -L_k^2(i, i-1), \qquad i = k+1, k+2, \dots, m_2. \tag{67}$$

The matrix $L^2 = L_{m_2-1}^2 \cdots L_4^2 L_3^2$ transforms the coefficient matrix M^2 into the upper triangular matrix $L^2 M^2$ of the form (32).

(2) The elimination procedure of Subsection 5.2.2 is used to construct a $m_2 \times m_2$ initializing upper tridiagonal matrice U_1^2 with multipliers

$$\begin{aligned} \sigma_3 &= \frac{1}{M^2(2,3) - M^2(2,3-2)} = U_1^2(3,3), \\ \sigma_i &= \frac{1}{M^2(2,i) - M^2(2,i-1)} = U_1^2(i,i), \qquad i = 4, 5, \dots, m_2. \end{aligned} \tag{68}$$

(3) Then, the elimination procedure of Subsection 5.2.3 is used to construct $m_2 \times m_2$ upper bidiagonal matrices U_k^2, $k = 2, 3, \dots, m_2 - 1$ with multipliers

$$\sigma_i = \frac{k}{M^2(2,i) - M^2(2,i-k)} = U_k^2(i,i), \qquad i = k+1, k+2, \dots, m_2. \tag{69}$$

We now obtain the following procedure which transforms M^2 into the identity matrix:

$$I^2 = L_{m_2-1}^2 L_{m_2-2}^2 \cdots L_3^2 M^2 U_1^2 U_2^2 \cdots U_{m_2-1}^2.$$

Thus we have the following factorization of M^2 into the product of elementary matrices:

$$M^2 = \left(L_{m_2-1}^2 L_{m_2-2}^2 \cdots L_3^2\right)^{-1} \left(U_1^2 U_2^2 \cdots U_{m_2-1}^2\right)^{-1},$$

and the solution is

$$\boldsymbol{u}^2 = U_1^2 U_2^2 \cdots U_{m_2-1}^2 L_{m_2-1}^2 L_{m_2-2}^2 \cdots L_3^2\, \boldsymbol{r}^2, \tag{70}$$

where fast computation goes from right to left.

Procedure (70) is implemented in Algorithm 5 in Appendix A in $O(m_2^2)$ operations. The input is $M = M^2$; $m = m_2$; $\boldsymbol{r} = \boldsymbol{r}^2$; $L_k = L_k^2$, $k = 3, 4, \dots, m_2 - 1$; $U_k = U_k^2$, $k = 1, 2, \dots, m_2 - 1$. The output is $\boldsymbol{u} = \boldsymbol{u}^2$.

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

We let $m_4 = p+1$ as defined in (27).

(1) The elimination procedure of Subsection 5.1.3 is applied to M^4 to construct $m_4 \times m_4$ lower bidiagonal matrices L_k^4, $k = 3, 4, \dots, m_4 - 2$, with multipliers

$$\tau_i = \frac{M^4(2,k)}{i-1} = -L_k^4(i, i-1), \qquad i = k+1, k+2, \dots, m_4 - 1. \tag{71}$$

The matrix $L^4 = L_{m_4-2}^4 \cdots L_4^4 L_3^4$ transforms the coefficient matrix M^4 into the upper triangular matrix L^4M^4 in columns 1 to $m_4 - 2$ of the form (35).

(2) The elimination procedure of Subsection 5.2.4 is used to construct a $m_4 \times m_4$ initializing upper tridiagonal matrix U_1^4, with multipliers,

$$\begin{aligned} \sigma_3 &= \frac{1}{M^4(2,3) - M^4(2,3-2)} = U_1^4(3,3), \\ \sigma_i &= \frac{1}{M^4(2,i) - M^4(2,i-1)} = U_1^4(i,i), \qquad i = 4, 5, \dots, m_4 - 2. \end{aligned} \tag{72}$$

(3) Then, the elimination procedure of Subsection 5.2.5 is used to construct $m_4 \times m_4$ elementary upper triangular matrices U_k^4, $k = 2, 3, \dots, m_4 - 3$, with multipliers,

$$\sigma_i = \frac{k}{M^4(2,i) - M^4(2,i-k)} = U_k^4(i,i), \qquad i = k+1, k+2, \dots, m_4 - 2. \tag{73}$$

Right multiplying L^4M^4 by the upper triangular matrix $U_1^4U_2^4 \cdots U_{m_4-3}^4$, will transform L^4M^4 into a diagonal+last-row+last-two-column matrix W_3^4 of the form (58).

(4) A $m_4 \times m_4$ lower bidiagonal matrix $L_{m_4-1}^4$, constructed with zero entries, except for,

$$\begin{aligned} L_{m_4-1}^4(i,i) &= 1, \qquad i = 1, 2, \dots, m_4, \\ L_{m_4-1}^4(m_4, j) &= -(M^4U_1^4U_2^4 \cdots U_{m_4-3}^4)(m_4, j), \qquad j = 2, 3, \dots, m_4 - 2, \end{aligned} \tag{74}$$

will transform W_3^4 into a matrix $W_2^4 = L_{m_4-1}^4 W_3^4$ with nonzero diagonal elements, $W_2^4(i,i) \neq 0$, $i = 1, 2, \dots, m_4$, the last two nonzero columns $W_2^4(1 : m_4, j) \neq 0$, $j = m_4 - 1, m_4$, and zero elsewhere. We call such a matrix a "diagonal+last-two-column" matrix.

(5) A factored Gaussian elimination, $L^4_{m_4+1}L^4_{m_4}$, will transform W^4_2 into a diagonal+last-column matrix $W^4_1 = L^4_{m_4+1}L^4_{m_4}W^4_2$ as follows. First, $W^4_2(m_4-1, m_4-1)$ is set to 1 by the diagonal matrix $L^4_{m_4}$ whose entries are zeros, except for,

$$\begin{aligned} L^4_{m_4}(i,i) &= 1, \qquad i = 1, 2, \dots, m_4-2, \\ L^4_{m_4}(m_4-1, m_4-1) &= 1/W^4_2(m_4-1, m_4-1), \\ L^4_{m_4}(m_4, m_4) &= 1. \end{aligned}$$

Then the non-diagonal entries in column $m_4 - 1$ of $L^4_{m_4}W^4_2$ are zeroed by the unit diagonal+column-(m_4-1) matrix $L^4_{m_4+1}$ whose entries are zeros, except for,

$$\begin{aligned} L^4_{m_4+1}(1 : m_4-2, m_4-1) &= -W^4_2(1 : m_4-2, m_4-1), \\ L^4_{m_4+1}(i,i) &= 1, \qquad i = 1, 2, \dots, m_4, \\ L^4_{m_4+1}(m_4, m_4-1) &= -W^4_2(m_4, m_4-1). \end{aligned}$$

(6) A factored Gaussian elimination, $L^4_{m_4+3}L^4_{m_4+2}$, will transform W^4_1 into the identity matrix $I^4 = L^4_{m_4+3}L^4_{m_4+2}W^4_1$ as follows. First, $W^4_1(m_4, m_4)$ is set to 1 by the diagonal matrix $L^4_{m_4+2}$ whose entries are zeros, except for,

$$\begin{aligned} L^4_{m_4+2}(i,i) &= 1, \qquad i = 1, 2, \dots, m_4-1, \\ L^4_{m_4+2}(m_4, m_4) &= 1/W^4_1(m_4, m_4). \end{aligned}$$

Then the non-diagonal entries in the last column of $L^4_{m_4+2}W^4_1$ are zeroed by the unit diagonal+last-column matrix $L^4_{m_4+3}$ whose entries are zeros, except for,

$$\begin{aligned} L^4_{m_4+3}(i,i) &= 1, \qquad i = 1, 2, \dots, m_4, \\ L^4_{m_4+3}(1 : m_4-1, m_4) &= -W^4_1(1 : m_4-1, m_4). \end{aligned}$$

We now obtain the following procedure which transforms M^4 into the identity matrix:

$$I^4 = L^4_{m_4+3}L^4_{m_4+2}\cdots L^4_3 M^4 U^4_1 U^4_2 \cdots U^4_{m_4-3}.$$

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

$$M^4 = \left(L^4_{m_4+3}L^4_{m_4+2}\cdots L^4_3\right)^{-1}\left(U^4_1U^4_2\cdots U^4_{m_4-3}\right)^{-1},$$

and the solution is

$$\boldsymbol{u}^4 = U^4_1U^4_2\cdots U^4_{m_4-3}L^4_{m_4+3}L^4_{m_4+2}\cdots L^4_3\,\boldsymbol{r}^4, \tag{75}$$

where fast computation goes from right to left.

Procedure (75) is implemented in Algorithm 6 in Appendix A in $O(m_4^2)$ operations. The input is $M = M^4$; $m = m_4$; $\boldsymbol{r} = \boldsymbol{r}^4$; $L_k = L_k^4$, $k = 3, 4, \dots, m_4 + 3$; $U_k = U_k^4$, $k = 1, 2, \dots, m_4 - 3$. The output is $\boldsymbol{u} = \boldsymbol{u}^4$.

6.4. Solution of $M^5\boldsymbol{u}^5 = \boldsymbol{r}^5$

We let $m_5 = p - 1$ as defined in (27).

(1) The elimination procedure of Subsection 5.1.1 is applied to M^5 to construct $m_5 \times m_5$ lower bidiagonal matrices L_k^5, $k = 2, 3, \dots, m_5 - 1$, with multipliers

$$\tau_i = \frac{M^5(2,k)}{i-1} = -L_k^5(i, i-1), \qquad i = k+1, k+2, \dots, m_5. \tag{76}$$

The matrix $L^5 = L_{m_5-1}^5 \cdots L_3^5 L_2^5$ transforms the coefficient matrix M^5 into the upper triangular matrix L^5M^5 in column 1 to $m_5 - 1$ of the form (31).

(2) The elimination procedure of Subsection 5.2.6 is used to construct $m_5 \times m_5$ upper bidiagonal matrices U_k^5, $k = 1, 2, \dots, m_5 - 2$, with multipliers

$$\sigma_i = \frac{k}{M^5(2,i) - M^5(2,i-k)} = U_k^5(i,i), \qquad i = k+1, k+2, \dots, m_5 - 1. \tag{77}$$

The right-product of the U_k^5, $k = 1, 2, \dots, m_5 - 2$, will transform L^5M^5 into a diagonal+last-column matrix W_1^5 of the form (63).

(3) A factored Gaussian elimination, $L_{m_5+1}^5 L_{m_5}^5$, will transform W_1^5 into the identity matrix $I^5 = L_{m_5+1}^5 L_{m_5}^5 W_1^5$ as follows. First, $W_1^5(m_5, m_5)$ is set to 1 by the diagonal matrix $L_{m_5}^5$ whose entries are zeros, except for,

$$L_{m_5}^5(i,i) = 1, \qquad i = 1, 2, \dots, m_5 - 1,$$
$$L_{m_5}^5(m_5, m_5) = 1/W_1^5(m_5, m_5).$$

Then the non-diagonal entries in the last column of $L_{m_5}^5 W_1^5$ are zeroed by the unit diagonal+last-column matrix $L_{m_5+1}^5$ whose entries are zeros, except for,

$$L_{m_5+1}^5(i,i) = 1, \qquad i = 1, 2, \dots, m_5,$$
$$L_{m_5+1}^5(1 : m_5 - 1, m_5) = -(W_1^5)(1 : m_5 - 1, m_5).$$

This procedure transforms M^5 into the identity matrix

$$I^5 = L_{m_5+1}^5 L_{m_5}^5 \cdots L_2^5 M^5 U_1^5 U_2^5 \cdots U_{m_5-2}^5.$$

Thus we have the following factorization of M^5 into the product of elementary matrices:

$$M^5 = \left(L^5_{m_5+1} L^5_{m_5} \cdots L^5_2\right)^{-1} \left(U^5_1 U^5_2 \cdots U^5_{m_5-2}\right)^{-1},$$

and the solution is

$$\boldsymbol{u}^5 = U^5_1 U^5_2 \cdots U^5_{m_5-2} L^5_{m_5+1} L^5_{m_5} \cdots L^5_2 \, \boldsymbol{r}^5, \tag{78}$$

where fast computation goes from right to left.

Procedure (78) is implemented in Algorithm 7 in Appendix A in $O(m_5^2)$ operations. The input is $M = M^5$; $m = m_5$; $\boldsymbol{r} = \boldsymbol{r}^5$; $L_k = L^5_k$, $k = 2, 3, \ldots, m_5 + 1$; $U_k = U^5_k$, $k = 1, 2, \ldots, m_5 - 2$. The output is $\boldsymbol{u} = \boldsymbol{u}^5$.

Remark 1. *Formulae* (2)–(4) *can be put in matrix form. For instance,* (3) *can be written as*

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

where

$$F^1 = \Big[h_{n+1} f(t_n + h, y_{n+1}), y_n, y_{n-1}, \ldots, y_{n-(p-3)}, h_{n+1} F_2, h_{n+1} F_3, h_{n+1} F_4\Big],$$

and

$$\boldsymbol{v}^1 = [b_5, \alpha_0, \alpha_1, \ldots, \alpha_{p-3}, b_2, b_3, b_4]^T,$$

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

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

The first form is used in this paper, the form similar to the second form for Vandermonde systems is found in [18], and the third form is found in [21].

7. Regions of absolute stability

The regions of absolute stability, $\mathcal{R}$, of constant step HB(p), $p = 4, 5, \ldots, 10$, listed in Appendix B, with coefficients c_i, $i = 1, 2, \ldots, 5$ given by (14), can be obtained by applying formulae (2)–(3) of the predictors P_i, $i = 2, 3, 4$ and the integration formula IF with constant h to the linear test equation

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

Table 2: For each given step number k, the table lists the order p, the α angles of $A(\alpha)$-stability for HB(p), MEBDF(p) and BDF(p), respectively.

HB(p)			MEBDF(p)			BDF(p)		
k	p	α	k	p	α	k	p	α
						1	1	90.00°
			1	2	90.00°	2	2	90.00°
			2	3	90.00°	3	3	86.03°
2	4	90.00°	3	4	90.00°	4	4	73.35°
3	5	90.00°	4	5	88.36°	5	5	51.84°
4	6	83.65°	5	6	83.07°	6	6	17.84°
5	7	80.52°	6	7	74.48°			
6	8	80.52°	7	8	61.98°			
7	9	78.68°	8	9	42.87°			
8	10	64.28°						

This gives the following difference equation and corresponding characteristic equation

$$\sum_{j=0}^{k} \eta_j(z)\, y_{n+j} = 0, \qquad \sum_{j=0}^{k} \eta_j(z)\, r^j = 0, \tag{79}$$

respectively, where $k = p - 2$ 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 (79) satisfy the root condition (see [19, pp. 70]). The scanning method used to find $\mathcal{R}$ is similar to the one used for Runge–Kutta methods (see [19]).

The stability functions $\eta_j(z)$, $j = 0, 1, \dots, k$ in (79) are rational functions of the form

$$\eta_k(z) = 1, \qquad \eta_j(z) = \frac{\sum_{\ell=0}^{3} n_{j\ell} z^\ell}{\sum_{\ell=0}^{4} d_{j\ell} z^\ell}, \qquad j = 0, 1, \dots, k-1.$$

Here $\eta_{k-1}(z)$ is of this form since the condition (13) is satisfied. Hence, in the difference equation of (79), $y_{n+k} \to 0$ as $z \to \infty$. This implies that HB(p), $p = 4, 5, \dots, 10$ are L-stable or $L(\alpha)$-stable according to whether these methods are A-stable or $A(\alpha)$-stable, respectively.

Table 2 lists the α angles of $A(\alpha)$-stability of HB(4–10), MEBDF(2–9) [14, p. 270] and BDF(1–6) [14, p. 251], respectively. It is seen that α of HB(p), $p = 4, 5, \dots, 10$ compare favorably with α of MEBDF(p), $p = 3, 4, \dots, 9$ which decrease faster than α of HB(p) for $p \geq 7$.

8. Controlling stepsize

The estimate $\|y_n - \widetilde{y}_n\|_\infty$ and the current step h_n are used to calculate the next stepsize h_{n+1} by means of formula [17]

$$h_{n+1} = \min\left\{h_{\max}, \beta\, h_n \left[\frac{\text{tolerance}}{\|y_n - \widetilde{y}_n\|_\infty}\right]^{1/\kappa}, 4\, h_n\right\}, \tag{80}$$

with $\kappa = p-1$ and safety factor $\beta = 0.81$.

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

(a) The stepsize, h_{n+1}, is obtained by formula (80) with $\kappa = p-1$.

(b) The numbers $\eta_2, \eta_3, \dots, \eta_{p-2}$, defined in (7), are calculated.

(c) The coefficients of integration formula IF, predictors P_2, P_3, P_4 and step control predictor P_5 are obtained successively as solutions of systems (15), (17), (19), (20) and (24).

(d) The values Y_2, Y_3, Y_4, y_{n+1}, and $\widetilde{y}_{n+1}$ are obtained by formulae (2)–(4).

(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 stepsize h_{n+1} is computed.

9. Numerical results

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

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

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 HB(p) were obtained by MATLAB's `ode15s` with stringent tolerance 5×10^{-14}.

We consider four following test problems:

(1) The Robertson chemical reaction [23, pp. 178–182].

Problem 1. Robertson chemical reaction:

$$\begin{aligned} y_1' &= -0.04y_1 + 10^4 y_2 y_3, & y_1(0) &= 1,\\ y_2' &= 0.04y_1 - 10^4 y_2 y_3 - 3\times 10^7 y_2^2, & y_2(0) &= 0,\\ y_3' &= 3\times 10^7 y_2^2, & y_3(0) &= 0, \end{aligned} \tag{81}$$

with $t_{\text{end}} = 400$.

(2) The stiff DETEST problem D1 [10].

Problem 2.

$$\begin{aligned} y_1' &= 0.2\,(y_2 - y_1)\,, & y_1(0) &= 0, \\ y_2' &= 10y_1 - (60 - 0.123y_3)\,y_2 + 0.125y_3, & y_2(0) &= 0, \\ y_3' &= 1, & y_3(0) &= 0, \end{aligned} \tag{82}$$

with $t_{\text{end}} = 400$.

(3) The Oregonator equation describing Belusov-Zhabotinskii reaction [11].

Problem 3. The Oregonator model describing Belusov-Zhabotinskii reaction

$$\begin{aligned} y_1' &= 77.27(y_2 + y_1 - 8.375 \cdot 10^{-6} y_1^2 - y_1 y_2), & y_1(0) &= 1, \\ y_2' &= (y_3 - (1 + y_1)y_2)/77.27, & y_2(0) &= 2, \\ y_3' &= 0.161(y_1 - y_3), & y_3(0) &= 3, \end{aligned} \tag{83}$$

with $t_{\text{end}} = 20$.

(4) The van der Pol's equation [14, pp. 4–6], [16].

Problem 4.

$$\begin{aligned} y_1' &= y_2, & y_1(0) &= 2, \\ y_2' &= \mu^2[(1 - y_1^2)y_2 - y_1], & y_2(0) &= 0, \end{aligned} \tag{84}$$

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

Similar to Hojjati et al. [16], we numerically compare our new methods with MEBDF(p), $p = 7, 8$, on the basis of the EPE, endpoint error as a function of number of steps (NS).

Table 3, 4, 5 and 6 list endpoint errors (EPEs) as a function of number of steps (NS) of HB(p), $p = 9, 10$ and MEBDF(p), $p = 7, 8$ for Robertson chemical reaction problem (81), stiff problem D1 (82), Oregonator problem (83) and van der Pol's equation (84), respectively.

It is seen that, in general, HB(p), $p = 9, 10$, compare favorably with MEBDF(p), $p = 7, 8$, at stringent tolerance.

The *NS percentage efficiency gain* (NS PEG) is defined by the formula (cf. Sharp [24]),

$$\text{NS PEG} = 100 \left[\frac{\sum_j \text{NS}_{2,j}}{\sum_j \text{NS}_{1,j}} - 1 \right], \tag{85}$$

Table 3: Endpoint errors (EPE) as a function of number of steps (NS) of HB(p), $p = 9, 10$ and MEBDF(p), $p = 7, 8$ for Robertson chemical reaction problem (81).

	Error in		Error in	
number of steps	HB(9)	HB(10)	MEBDF(7)	MEBDF(8)
51	2.14e-07	2.42e-06	2.33e-05	1.02e-04
55	6.99e-08	4.05e-08	1.17e-05	4.19e-05
62	1.19e-08	5.33e-09	3.94e-06	1.02e-05
70	1.96e-09	6.17e-10	1.31e-06	2.43e-06
81	2.26e-10	5.91e-11	3.46e-07	4.33e-07
95	2.13e-11	9.37e-12	8.12e-08	6.60e-08
112	1.86e-12	4.05e-12	1.82e-08	9.45e-09

Table 4: Endpoint errors (EPE) as a function of number of steps (NS) of HB(p), $p = 9, 10$ and MEBDF(p), $p = 7, 8$ for stiff problem D1 (82).

	Error in		Error in	
number of steps	HB(9)	HB(10)	MEBDF(7)	MEBDF(8)
34	8.37e-07	6.21e-07	9.89e-05	5.83e-05
41	1.87e-07	5.08e-08	2.76e-05	1.53e-05
52	2.79e-08	8.09e-09	5.46e-06	2.78e-06
64	5.29e-09	3.87e-10	1.33e-06	6.29e-07
81	8.03e-10	6.43e-11	2.66e-07	1.16e-07

Table 5: Endpoint errors (EPE) as a function of number of steps (NS) of HB(p), $p = 9, 10$ and MEBDF(p), $p = 7, 8$ for Oregonator problem (83).

	Error in		Error in	
number of steps	HB(9)	HB(10)	MEBDF(7)	MEBDF(8)
31	8.77e-04	8.40e-04	3.12e-02	8.68e-02
38	1.79e-04	1.45e-05	1.00e-02	2.01e-02
51	1.79e-05	1.56e-06	1.94e-03	2.43e-03
78	6.48e-07	1.39e-07	1.82e-04	1.15e-04
125	1.63e-08	1.13e-08	1.31e-05	3.89e-06
158	2.61e-09	3.02e-10	3.56e-06	7.22e-07

Table 6: Endpoint errors (EPE) as a function of number of steps (NS) of HB(p), $p = 9, 10$ and MEBDF(p), $p = 7, 8$ for van der Pol's equation (84).

	Error in		Error in	
number of steps	HB(9)	HB(10)	MEBDF(7)	MEBDF(8)
35	6.60e-06	2.66e-06	5.52e-02	1.01e-02
41	2.54e-06	3.20e-07	2.15e-02	4.55e-03
46	1.27e-06	3.19e-07	1.08e-02	2.55e-03
56	3.87e-07	3.42e-09	3.35e-03	9.48e-04
61	2.31e-07	1.01e-09	2.01e-03	6.16e-04
81	4.17e-08	1.10e-08	3.72e-04	1.48e-04
146	1.19e-09	2.09e-09	1.11e-05	7.62e-06
185	2.86e-10	6.64e-10	2.71e-06	2.31e-06

Table 7: NS PEG of HB(p), $p = 9, 10$, over MEBDF(p), $p = 7, 8$, for the listed problems.

	NS PEG of HB(9) over:		NS PEG of HB(10) over:	
Problem	MEBDF(7)	MEBDF(8)	MEBDF(7)	MEBDF(8)
Robertson equation	93%	76%	93%	75%
problem D1	118%	91%	159%	127%
Oregonator equation	214%	108%	238%	233%
van der Pol's equation	357%	401%	481%	537%

where $NS_{1,j}$ and $NS_{2,j}$ are the NS of methods 1 and 2, respectively, and $j = -\log_{10}(\text{EPE})$. To compute $NS_{2,j}$ and $NS_{1,j}$ appearing in (85), 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 $-\log_{10}(\text{EPE}) = j$ and obtain $\log_{10}(\text{NS})$ from the approximating curve, and finally NS PEG.

Table 7 lists the NS PEG of HB(p), $p = 9, 10$, over MEBDF(p), $p = 7, 8$, for four problems. It is seen that HB(p), $p = 9, 10$, win.

As an example, for van der Pol's equation, HB(p), $p = 9, 10$ take 159 steps, compared to 576 steps used by MEBDF(p), $p = 7, 8$ to obtain an EPE of 7.0e-10, approximatively.

10. Conclusion

Variable-step (VS) 4-stage k-step Hermite–Birkhoff (HB) methods of order $p = (k+2)$, $p = 9, 10$, denoted by HB(p), are constructed as a combination of linear k-step methods of order $(p-2)$ and a diagonally implicit one-step 4-stage Runge–Kutta method of order 3 (DIRK3) for solving stiff ordinary differential equations. 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 confluent Vandermonde-type systems. This approach allows us to develop $L(\alpha)$-stable methods of order up to 10. Fast algorithms are developed for solving these systems in $O(p^2)$ operations to obtain HB interpolation polynomials in terms of generalized Lagrange basis functions. The stepsizes of these methods are controlled by a local error estimator. HB(p) of order $p = 9$ and 10 compare favorably with existing Cash modified extended backward differentiation formulae of order 7 and 8, MEBDF(7-8), in solving problems often used to test higher order stiff ODE solvers on the basis of number of steps and error at the endpoint of the integration interval.

Acknowledgements. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

A. Algorithms

Definition 1. ***Algorithm 1 constructs*** $L_k(i, i-1)$ ***entries of lower bidiagonal matrices*** L_k ***(applied to IF,*** P_i***,*** $i = 2, 4, 5$***) as functions of*** η_j***,*** $j = 2, 3, \dots, p-2$***.***

For $k = k_0 : k_{\text{end}}$, do the following iteration:

For $i = i_0 : -1 : k+1$, 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 $k_0 = 2, k_{\text{end}} = m-3, i_0 = m$ for IF, $k_0 = 3, k_{\text{end}} = m-1, i_0 = m$ for P_2 $k_0 = 3, k_{\text{end}} = m-2, i_0 = m$ for P_4 and $k_0 = 2, k_{\text{end}} = m-1, i_0 = m$ for P_5.

Definition 2. ***Algorithm 2 constructs diagonal entries*** $U_k(j, j)$ ***of upper bidiagonal matrices*** U_k ***(applied to IF and*** P_5***) as functions of*** η_j***,*** $j = 2, 3, \dots, p-2$***.***

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 $k_{\text{end}} = m-4, j_0 = m-3$ for IF, and $k_{\text{end}} = m-2, j_0 = m-1$ for P$_5$.

Definition 3. ***Algorithm 3 constructs diagonal entries $U_k(j,j)$ of upper bidiagonal matrices U_k (applied to $\boldsymbol{P}_2$ and $\boldsymbol{P}_4$) as functions of η_j, $j = 2,3,\dots,p-2$.***

(Section 1: for initializing U_1)
For $k = 1$, do the following iteration:

For $j = j_0 : -1 : 4$, 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

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

For $j = 3$, do the following two steps:
Step (1) $U_k(j,j) = 1/[M(k+1,j) - M(k+1,j-2)]$.
Step (2) for $i = k : j$, compute

$$M(i,j) = (M(i,j) - M(i,j-2))U_k(j,j).$$

(Section 2: for U_k, $k = 2 : k_{\text{end}}$)
For $k = 2 : k_{\text{end}}$, 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 $k_{\text{end}} = m-1, j_0 = m$ for P$_2$, and $k_{\text{end}} = m-3, j_0 = m-2$ for P$_4$.

Definition 4. ***Algorithm 4 solves the systems for IF in $O(m^2)$ operations.***

Given $[\eta_2, \eta_3, \dots, \eta_{p-2}]$ 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 $L_{m-3}L_{m-4}\cdots L_2\boldsymbol{r}$:
for $k = 2,3,\dots,m-3$, compute

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

Step (2) This step forms the two matrices L_{m-2} and L_{m-1}: it computes the coefficients $G_{m-2}(i)$, $i = 1,2,\dots,m$ used to form the two matrices L_{m-2} and L_{m-1} which transform W_3^1 into a diagonal+last-two-column matrix $W_2^1 = L_{m-1}L_{m-2}W_3^1$:
First set $G_{m-2}(1:m)$,

$$G_{m-2}(1:m) = M(1:m, m-2).$$

The following computation overwrites $G_{m-2}(1:m)$ with $L_{m-3}L_{m-4}\cdots L_2G_{m-2}(1:m)$:
for $k = 2,3,\dots,m-3$, compute

$$G_{m-2}(i) = G_{m-2}(i) + G_{m-2}(i-1)L_k(i,i-1), \quad i = m, m-1, \dots, k+1.$$

Step (3) The following computation overwrites the newly obtained $\boldsymbol{r}$ with $L_{m-1}L_{m-2}\boldsymbol{r}$:

$$r(m-2) = r(m-2)/G_{m-2}(m-2),$$

next, for $k = m, m-1, m-3, m-4, \ldots, 1$, compute

$$r(k) = r(k) - G_{m-2}(k)r(m-2).$$

Step (4) This step forms the two matrices L_m and L_{m+1}: it computes the coefficients $G_{m-1}(i)$, $i = 1, 2, \ldots, m$ used to form the two matrices L_m and L_{m+1} which transform W_2^1 into a diagonal+last-column matrix $W_1^1 = L_{m+1}L_mW_2^1$:

First set $G_{m-1}(1:m)$,

$$G_{m-1}(1:m) = M(1:m, m-1).$$

The following computation overwrites $G_{m-1}(1:m)$ with $L_{m-3}L_{m-4}\cdots L_2G_{m-1}(1:m)$: for $k = 2, 3, \ldots, m-3$, compute

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

The following computation overwrites the newly obtained $G_{m-1}(1:m)$ with $G_{m-1}(1:m) = L_{m-1}L_{m-2}G_{m-1}(1:m)$:

$$G_{m-1}(m-2) = G_{m-1}(m-2)/G_{m-2}(m-2),$$

next, for $k = m, m-1, m-3, m-4, \ldots, 1$, compute

$$G_{m-1}(k) = G_{m-1}(k) - G_{m-2}(k)G_{m-1}(m-2).$$

Step (5) The following computation overwrites the newly obtained $\boldsymbol{r}$ with $L_{m+1}L_m\boldsymbol{r}$:

$$r(m-1) = r(m-1)/G_{m-1}(m-1),$$

next, compute

$$r(m) = r(m) - G_{m-1}(m)r(m-1),$$

and for $k = m-2, m-3, \ldots, 1$, compute

$$r(k) = r(k) - G_{m-1}(k)r(m-1).$$

Step (6) This step forms the two matrices L_{m+2} and L_{m+3}: it computes the coefficients $G_m(i)$, $i = 1, 2, \ldots, m$ used to form the two matrices L_{m+1} and L_{m+2} which transform W_1^1 into the identity matrix $I^1 = L_{m+3}L_{m+2}W_1^1$:

First set $G_m(1:m)$:

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

The following computation overwrites $G_m(1:m)$ with $L_{m-3}L_{m-4}\cdots L_2G_m(1:m)$:
for $k = 2, 3, \ldots, m-3$, compute

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

The following computation overwrites the newly obtained $G_m(1:m)$ with $L_{m-1}L_{m-2}G_m(1:m)$:

$$G_m(m-2) = G_m(m-2)/G_{m-2}(m-2),$$

next, for $k = m, m-1, m-3, m-4, \dots, 1$, compute

$$G_m(k) = G_m(k) - G_{m-2}(k)G_m(m-2),$$

The following computation overwrites the newly obtained $G_m(1:m)$ with $L_{m+1}L_mG_m(1:m)$:

$$G_m(m-1) = G_m(m-1)/G_{m-1}(m-1),$$

next, for $k = m, m-2, m-3, m-4, \dots, 1$, compute

$$G_m(k) = G_m(k) - G_{m-1}(k)G_m(m-1).$$

Step (7) The following computation overwrites the newly obtained $\boldsymbol{r}$ with $L_{m+3}L_{m+2}\boldsymbol{r}$:

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

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

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

Step (8) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_1U_2\cdots U_{m-4}\boldsymbol{r}$:
For $k = m-4, m-5, \dots, 1$, compute

$$r(i) = r(i)U_k(i,i), \qquad i = k+1, k+2, \dots, m-3,$$
$$r(i) = r(i) - r(i+1), \qquad i = k, k+1, \dots, m-4.$$

Definition 5. ***Algorithm 5 solves the systems for P_2 in $O(m^2)$ operations.***

Given $[\eta_2, \eta_3, \dots, \eta_{p-2}]$ 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 $k = 3, 4, \dots, m-1$, compute

$$r(i) = r(i) + r(i-1)L_k(i,i-1), \quad i = m, m-1, \dots, k+1.$$

Step (2) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_2U_3\cdots U_{m-1}\boldsymbol{r}$:
For $k = m-1, m-2, \dots, 2$, compute

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

Next, the following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_1\boldsymbol{r}$:
For $k = 1$, compute

$$r(i) = r(i)U_k(i,i), \qquad i = k+1, k+2, \dots, m,$$
$$r(1) = r(1) - r(3),$$
$$r(i) = r(i) - r(i+1), \qquad i = k+2, k+3, \dots, m-1.$$

Definition 6. ***Algorithm 6 solves the systems for P_4 in $O(m^2)$ operations.***

Given $[\eta_2, \eta_3, \dots, \eta_{p-2}]$ 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 $L_{m-2}L_{m-3}\cdots L_3\boldsymbol{r}$:
For $k = 3, 4, \dots, m-2$, compute

$$r(i) = r(i) + r(i-1)L_k(i, i-1), \quad i = m-1, m-2, \dots, k+1.$$

Step (2) This step forms the matrix L_{m-1}: it computes the coefficients vector $H(1:m)$ used to form the matrix L_{m-1} which transforms W_3^4 into a diagonal+last-two-column matrix $W_2^4 = L_{m-1}W_3^4$:
First set H equal to the last row of M,

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

The following computation overwrites H with HU_1:
for $k = 1$, compute

$$\begin{aligned} H(i) &= (H(i) - H(i-1))U_k(i,i), \quad i = m-2, m-3, \dots, k+3, \\ H(k+2) &= (H(k+2) - H(k))U_k(k+2, k+2). \end{aligned}$$

The following computation overwrites H with $HU_2U_3 \dots U_{m-3}$:
For $k = 2, 3, \dots, m-3$, compute

$$H(i) = (H(i) - H(i-1))U_k(i,i), \quad i = m-2, m-3, \dots, k+1.$$

Step (3) The following computation overwrites the newly obtained $\boldsymbol{r}$ with $L_{m-1}\boldsymbol{r}$:
First set P_{rH},

$$P_{rH} = 0.$$

Compute the product P_{rH} and, next, compute $r(m)$:

$$\begin{aligned} P_{rH} &= P_{rH} + r(i)H(i), \qquad i = 2, 3, \dots, m-2, \\ r(m) &= -P_{rH} + r(m). \end{aligned}$$

Step (4) This step forms the two matrices L_m and L_{m+1}: it computes the coefficients vector $G_{m-1}(i)$, $i = 1, 2, \dots, m$ used to form the two matrices L_m and L_{m+1} which transform W_2^4 into a diagonal+last-column matrix $W_1^4 = L_{m+1}L_mW_2^4$:
First set $G_{m-1}(1:m)$,

$$G_{m-1}(1:m) = M(1:m, m-1).$$

The following computation overwrites $G_{m-1}(1:m)$ with $L_{m-2}L_{m-3}\cdots\ L_3G_{m-1}(1:m)$: for $k = 3, 4, \dots, m-2$, compute

$$G_{m-1}(i) = G_{m-1}(i) + G_{m-1}(i-1)L_k(i, i-1), \quad i = m-1, m-2, \dots, k+1.$$

The following computation overwrites the newly obtained vector G_{m-1} with $L_{m-1}G_{m-1}$:
First set P_{GH},

$$P_{GH} = 0.$$

Compute the product P_{GH} and, next, compute $G_{m-1}(m)$:

$$\begin{aligned} P_{GH} &= P_{GH} + G_{m-1}(i)H(i), \qquad i = 2, 3, \dots, m-2, \\ G_{m-1}(m) &= -P_{GH} + G_{m-1}(m). \end{aligned}$$

Step (5) The following computation overwrites the newly obtained $\boldsymbol{r}$ with $L_{m+1}L_m\boldsymbol{r}$:

$$r(m-1) = r(m-1)/G_{m-1}(m-1),$$

next, for $k = m, m-2, m-3, \dots, 1$, compute

$$r(k) = r(k) - G_{m-1}(k)r(m-1).$$

Step (6) This step forms the two matrices L_{m+2} and L_{m+3}: it computes the coefficients $G_m(i)$, $i = 1, 2, \dots, m$ used to form the two matrices L_{m+2} and L_{m+3} which transform W_1^4 into the identity matrix $I^4 = L_{m+3}L_{m+2}W_1^4$:

First set $G_m(1:m)$:

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

The following computation overwrites $G_m(1:m)$ with $L_{m-2}L_{m-3}\cdots L_3G_m(1:m)$:

For $k = 3, 4, \dots, m-2$, compute

$$G_m(i) = G_m(i) + G_m(i-1)L_k(i, i-1), \quad i = m-1, m-2, \dots, k+1.$$

The following computation overwrites the newly obtained vector G_m with $L_{m-1}G_m$:

First set P_{GH},

$$P_{GH} = 0.$$

Compute the product P_{GH} and, next, compute $G_m(m)$:

$$\begin{aligned} P_{GH} &= P_{GH} + G_m(i)H(i), \qquad i = 2, 3, \dots, m-2, \\ G_m(m) &= -P_{GH} + G_m(m). \end{aligned}$$

The following computation overwrites the newly obtained $G_m(1:m)$ with $L_{m+1}L_mG_m(1:m)$:

$$G_m(m-1) = G_m(m-1)/G_{m-1}(m-1),$$

next, for $k = m, m-2, m-3, m-4, \dots, 1$, compute

$$G_m(k) = G_m(k) - G_{m-1}(k)G_m(m-1).$$

Step (7) The following computation overwrites the newly obtained $\boldsymbol{r}$ with $L_{m+3}L_{m+2}\boldsymbol{r}$:

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

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

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

Step (8) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_1U_2U_3\cdots U_{m-3}\boldsymbol{r}$:

(8.1) The following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_2U_3\cdots U_{m-3}\boldsymbol{r}$:

For $k = m-3, m-4, \dots, 2$, compute

$$\begin{aligned} r(i) &= r(i)U_k(i,i), \qquad i = k+1, k+2, \dots, m-2, \\ r(i) &= r(i) - r(i+1), \qquad i = k, k+1, \dots, m-3. \end{aligned}$$

(8.2) Next, the following iteration overwrites $\boldsymbol{r} = r(1:m)$ with $U_1\boldsymbol{r}$:

For $k = 1$, compute

$$\begin{aligned} r(i) &= r(i)U_k(i,i), \qquad i = k+2, k+3, \dots, m-2, \\ r(1) &= r(1) - r(3), \\ r(i) &= r(i) - r(i+1), \qquad i = k+2, k+3, \dots, m-3. \end{aligned}$$

Definition 7. ***Algorithm 7 solves the systems for*** $\boldsymbol{P}_5$ ***in*** $O(m^2)$ ***operations.***

Given $[\eta_2, \eta_3, \dots, \eta_{p-2}]$ 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 $k = 2, 3, \dots, m-1$, compute

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

Step (2) This step forms the two matrices L_m and L_{m+1}: it computes the coefficients $G_m(i)$, $i = 1, 2, \dots, m$ used to form the two matrices L_m and L_{m+1} which transform W_1^5 into the identity matrix I^5: $I^5 = L_{m+1}L_mW_1^5$.
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 $k = 2, 3, \dots, m-1$, compute

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

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 $k = m-1, m-2, \dots, 1$, 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 $k = m-2, m-3, \dots, 1$, compute

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

B. Coefficients of HB(p), $p = 4, 5, \dots, 10$.

The appendix lists the coefficients of HB(p), of order $p = 4, 5, \dots, 10$, with coefficients c_i, $i = 1, 2, \dots, 5$ given by (14) and considered in this paper. It is to be noted that, in Table 8–10, since $a_{22} = a_{33} = a_{44} = b_5$, only a_{22} are listed.

Table 8: Coefficients of the implicit predictors P_i, $i = 2, 3, 4$ and of the integration formula of HB(4).

k	2
coeffs\ p	4
a_{22}	4.6349043784767707e-01
a_{21}	1.2661672524404526e+00
α_{20}	5.4950392168197382e-01
α_{21}	4.5049607831802618e-01
a_{32}	-1.8530834291876901e-02
a_{31}	-2.1887246599062252e-01
α_{30}	1.1616817724748036e+00
α_{31}	-1.6168177247480350e-01
a_{43}	9.2532001902408567e-01
a_{42}	-1.2991041785648558e-01
a_{41}	-1.1719553264875242e-01
α_{40}	1.0580323817860031e+00
α_{41}	-5.8032381786003208e-02
b_4	-5.8502664620919764e-01
b_3	7.3620641073684356e-01
b_2	3.6285323782546275e-01
α_0	1.0224765597992143e+00
α_1	-2.2476559799214289e-02

References

[1] Alexander, R., *Diagonally implicit Runge–Kutta methods for stiff ODEs*, SIAM J. Numer. Anal. 14(1977), 1006–1021.

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

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

[4] Butcher, J. C., Chen, D. J. L., *A new type of singly-implicit Runge–Kutta method*, Appl. Numer. Math. 34(2000), 179–188.

[5] Cash, J. R., *On the integration of stiff systems of ODEs using extended backward differentiation formula*, Numer. Math. 34(1980), 235–246.

[6] Cash, J. R., *The integration of stiff initial value problems in ODEs using modified extended backward differentiation formula*, Comput. Math. Appl. 9(1983), 645–657.

[7] Cash, J. R., Considine, S., *An MEBDF code for stiff initial value problems*, ACM Trans. Math. Software 18(1992), 142–160.

[8] Cash, J. R., *Modified extended backward differentiation formulae for the numerical solution of stiff initial value problems in ODEs and DAEs*, J. Comput. Appl. Math. 125(2000), 117–130.

[9] Ebadi, M., Gokhale, M. Y., *Hybrid BDF methods for the numerical solutions of ordinary differential equations*, Numer. Algorithms 55(2010), 1–17 . DOI 10.1007/s11075-009-9354-4

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

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

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

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

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

[15] Hojjati, G., Rahimi, M. Y., Hosseini, S. M., *A-EBDF: an adaptive method for numerical solution of stiff systems of ODEs*, Math. Comput. Simulation 66(2004), 33–41.

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

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

[18] Krogh, F. T., *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.

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

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

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

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

[23] Robertson, H. H., *The solution of a set of reaction rate equations*, In: J. Walsh ed: Numer. Anal., an Introduction, Academic Press (1966), 178–182.

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

Truong Nguyen-Ba, Thierry Giordano and Rémi Vaillancourt
Department of Mathematics and Statistics, University of Ottawa,
Ottawa, Ontario, Canada K1N 6N5
email: *trnguyen@uottawa.ca*, *giordano@uottawa.ca* and *remi@uottawa.ca*

Table 9: Coefficients of the implicit predictors P_i, $i = 2, 3, 4$ and of the integration formulae of HB(p), $p = 5, 6, 7$.

k	3	4	5
coeffs\\p	5	6	7
a_{22}	4.6349043784767707e-01	4.6155581379386562e-01	4.4584126788465805e-01
a_{21}	1.4003497954748405e+00	1.2821415800023641e+00	1.1809598520943367e+00
α_{20}	3.4823010713039193e-01	5.7502414183048167e-01	9.1481034721048538e-01
α_{21}	7.1886116438680214e-01	3.4131085592548177e-01	-4.3812261641398953e-01
α_{22}	-6.7091271517193962e-02	1.2777008083146502e-01	8.5659166780677842e-01
α_{23}		-4.4105078587428515e-02	-4.0569638322242141e-01
α_{24}			7.2416984619147140e-02
a_{32}	-3.0849563760214662e-02	-3.4791032567112530e-02	-3.0417325207035724e-02
a_{31}	-2.3979061907940127e-01	-2.0114174241918126e-01	-1.7570780227865077e-01
α_{30}	1.2272947421546210e+00	1.1821653030018511e+00	1.1468949085655049e+00
α_{31}	-2.5967082927732166e-01	-2.0042250866454406e-01	-1.2131102878253612e-01
α_{32}	3.2376087122700813e-02	1.6494979555944228e-02	-5.8044015259916808e-02
α_{33}		1.7622261067487407e-03	4.0494252246431703e-02
α_{34}			-8.0341167694837432e-03
a_{43}	8.6550584976656608e-01	8.5024803239176061e-01	8.9926334191083679e-01
a_{42}	-1.2094398621868979e-01	-1.2632597125244538e-01	-1.2382213043693883e-01
a_{41}	-9.5828504441028375e-02	-6.6163354324848211e-02	-1.3840058562750060e-01
α_{40}	1.1139780059745155e+00	1.1381648831163653e+00	1.2213905025684662e+00
α_{41}	-1.4044292075102807e-01	-2.1560838560463660e-01	-3.9512851149057066e-01
α_{42}	2.6464914776512558e-02	9.7144489404372855e-02	2.6050923462671782e-01
α_{43}		-1.9700986916101452e-02	-1.0433995063411622e-01
α_{44}			1.7568724929502764e-02
b_4	-4.4932378943708634e-01	-3.9309941410892663e-01	-3.3680433035369617e-01
b_3	7.0309903482509495e-01	6.8674782107319698e-01	6.8053670737153726e-01
b_2	2.4792697121265941e-01	2.0229434482024061e-01	1.6361881436910350e-01
α_0	1.0380122948604162e+00	1.0495685791492901e+00	1.0572992300634656e+00
α_1	-4.1217244169177658e-02	-5.7458711632632553e-02	-7.0102502172306555e-02
α_2	3.2049493087613640e-03	8.7131202390179352e-03	1.5387946419020668e-02
α_3		-8.2298775567559324e-04	-2.8577658465862143e-03
α_4			2.7309153640663492e-04

Table 10: Coefficients of the implicit predictors P_i, $i = 2, 3, 4$ and of the integration formulae of HB(p), $p = 8, 9, 10$.

k	6	7	8
coeffs\\p	8	9	10
a_{22}	4.2533683882410295e-01	3.8669248231767694e-01	3.5644917896211648e-01
a_{21}	1.1691175426503642e+00	1.8068140479185923e+00	2.5399256921202902e+00
α_{20}	1.1820658725415993e+00	2.2585956016814793e-01	-1.0676245134148956e+00
α_{21}	-1.3457907856374380e+00	-3.9202677101680204e-02	2.1897810218978102e+00
α_{22}	2.1897810218978102e+00	2.1093991104430829e+00	8.8466046491689831e-01
α_{23}	-1.4600218816659654e+00	-2.1897810218978102e+00	-2.1897810218978102e+00
α_{24}	5.0824170797589130e-01	1.2116697568163834e+00	1.8736987013081039e+00
α_{25}	-7.4275935111897506e-02	-3.6508178347831965e-01	-9.0533024426837816e-01
α_{26}		4.7137055050195838e-02	2.4285835102429823e-01
α_{27}			-2.8262759566026571e-02
a_{32}	-2.7820033747103474e-02	-1.8268922342457146e-02	-1.2644364453523351e-02
a_{31}	-9.4495480223432252e-02	-1.4283473085755846e-01	-1.7206642996978982e-01
α_{30}	1.0148354422389918e+00	1.1658484444977595e+00	1.2895079102810389e+00
α_{31}	1.5741558417488954e-01	-9.9141780515115380e-02	-3.5161810938268645e-01
α_{32}	-3.3031526045982507e-01	-2.0690958345228419e-01	-1.7232691644707614e-03
α_{33}	2.2529826789821539e-01	2.4091961693981062e-01	1.5871739214940811e-01
α_{34}	-7.8742717125058329e-02	-1.3701582178785510e-01	-1.5342353763525574e-01
α_{35}	1.1508683272786733e-02	4.1709443076754824e-02	7.7139532052879739e-02
α_{36}		-5.4103187590702640e-03	-2.1077864145246719e-02
α_{37}			2.4779458443329057e-03
a_{43}	8.9538742002184812e-01	1.0954037811396611e+00	1.3231666530204849e+00
a_{42}	-1.2088516509822729e-01	-1.0848858678504342e-01	-9.9919961109895900e-02
a_{41}	1.2342483090987948e-01	-1.1810131700162743e-01	-5.0145251499123311e-01
α_{40}	8.2624798878744277e-01	8.3103215300350552e-01	9.8854551131180979e-01
α_{41}	2.1951979318994586e-01	3.6702700935554078e-01	3.5619383494601842e-01
α_{42}	-2.7622002080726051e-02	-3.2129860931787940e-01	-7.3962838738604464e-01
α_{43}	-3.5784041350335596e-02	1.7120416512594636e-01	6.7541016661566800e-01
α_{44}	2.1587935740775669e-02	-5.8539351153030231e-02	-4.0752380683909584e-01
α_{45}	-3.9496742871028313e-03	1.1563207882864681e-02	1.6050895203004736e-01
α_{46}		-9.8857489694740381e-04	-3.7433269662979912e-02
α_{47}			3.9269989845767796e-03
b_4	-2.8478390776767987e-01	-2.0711184580076136e-01	-1.5094507140981844e-01
b_3	6.8006098992011321e-01	6.8844479542865034e-01	6.9747851629196012e-01
b_2	1.3073344134353643e-01	8.5129921834428163e-02	5.3103287011206937e-02
α_0	1.0615542018171131e+00	1.0596434936095855e+00	1.0549616923905276e+00
α_1	-7.8442822347043528e-02	-7.7046772382444131e-02	-6.9806707506548230e-02
α_2	2.1782023749097362e-02	2.3441103972377812e-02	1.9864711237223571e-02
α_3	-5.9010736889115135e-03	-7.7722557134681704e-03	-6.5456198489904126e-03
α_4	1.1089941130658875e-03	2.0658402730010452e-03	1.8785469224632662e-03
α_5	-1.0132364332147836e-04	-3.6178282039571599e-04	-4.0536116900096033e-04
α_6		3.0373061343575776e-05	5.6492230588381594e-05
α_7			-3.7542562630236547e-06