ABSTRACT. Variable-step variable-order 3-stage Hermite-Birkhoff-Obrechko methods of order 4 to 14, denoted by HBO(4-14)3, are constructed for solving non-stiff systems of first-order differential equations of the form $y' = f(x, y)$, $y(x_0) = y_0$. These methods use $y'$ and $y''$ as in Obrechko methods. Forcing a Taylor expansion of the numerical solution to agree with an expansion of the true solution leads to multistep-type and Runge-Kutta-type order conditions which are reorganized into linear Vandermonde-type systems. Fast and stable algorithms are developed for solving these systems to obtain Hermite-Birkhoff interpolation polynomials in terms of generalized Lagrange basis functions. The order and step-size of these methods are controlled by four local error estimators. When programmed in Matlab, HBO(4-14)3 is superior to Matlab’s ode113 in solving several problems often used to test higher order ODE solvers on the basis of the number of steps, CPU time, and maximum global error. It is also superior to the variable step 3-stage HBO(14)3 of order 14 on some problems. Programmed in C++, HBO(4-14)3 is superior to DP(8,7)13M in solving expensive equations over a long period of time.

1 Introduction In this paper, the 3-stage variable step (VS) HBO(14)3 of order 14 constructed in [19] is extended to variable-step variable-order (VSVO) 3-stage Hermite-Birkhoff-Obrechko methods of
order 4 to 14, denoted by HBO(4-14)3. Both of these methods are designed for solving nonstiff systems of first-order initial value problems of the form

\[ y' = f(x, y), \quad y(x_0) = y_0, \quad \text{where} \quad \frac{d}{dx}, \]

where the derivative \( f'(x, y) \) can be obtained analytically or recursively. There are many such problems, for instance in dynamical systems [2], [23].

The variable order HBO(4-14)3 has the remarkable property that the norm of the principal local truncation error coefficient decreases rapidly as the order increases.

The competitiveness of HBO(p)3 of order \( p \) comes from the surprisingly stable fast solution of ill-conditioned Vandermonde-type linear systems in \( O(p^3) \) operations (see [11, p. 187]) by taking the special structure of these systems into account, in contrast with Gaussian elimination with pivoting in \( O(p^3) \) operations, which requires more storage, is slower and, at times, leads to unstable solutions when the stepsize is very small.

It is seen that HBO(4-14)3, when programmed in Matlab, requires fewer steps, uses less CPU time, and has higher accuracy than Matlab's ode113 on several problems often used to test higher order ODE solvers. The VSVO HBO(4-14)3 is shown to perform better than the VS HBO(14)3 on some problems. When programmed in C++, HBO(4-14)3 is superior to DP(8,7)13M [22] in solving expensive equations over a long period of time.

HB methods of order 9, 10 and 11 have been studied in [18]. Three-stage HB(5-15)3 of order 5 to 15 and 4-stage HBO(5-14)4 of order 5 to 14 can be found in [20] and [21], respectively.

The contents of the paper is as follows. Section 2 defines VSVO HBO(4-14)3 of order 4 to 14. Order conditions are listed in Section 3. In Section 4, HBO(4-14)3 is represented in terms of Vandermonde-type systems. In Section 5, the reader is referred to Section 5 and the Appendix of [19] for the symbolic construction of elementary matrices which decompose the coefficient matrices of these systems. Section 6 deals with fast solutions of Vandermonde-type systems. Section 7 considers the regions of absolute stability of constant step HBO(4-14)3 and its principal local truncation error coefficients. Section 8 deals with step and order controls. Numerical results are listed in Section 9.

2 VSVO HBO(4-14)3 The defining formulae of HBO(p)3, of order \( p \), depends on its variable backstep points and three off-step points,
which in this paper are taken to be

\( c_1 = 0, \quad c_2 = \frac{2}{3}, \quad c_3 = 1. \)

This choice, which is based on extensive numerical experimentation, is simple and involves only one new point, thus reducing computation time.

An HBO(3) requires the following four formulae to perform the integration step from \( x_n \) to \( x_{n+1} \), where, for notational simplicity, \( c_1 = 0 \) is used in the summations and \( c_3 \) is used instead of its value 1 to keep track of it and distinguish the values \( y_{n+c_3} \) and \( f_{n+c_3} \) from \( y_{n+1} \) and \( f_{n+1} \), respectively. The “floor” of a real number \( q \), denoted by \( \lfloor q \rfloor \), is the largest integer smaller than or equal to \( q \).

A Hermite-Birkhoff polynomial (HBP) of degree \( p-2 \) is used as predictor \( P_2 \) to obtain \( y_{n+c_2} \) to order \( p-2 \),

\[
(3) \quad y_{n+c_2} = y_n + h_{n+1} \left( a_{21} f_{n+c_1} + \sum_{j=1}^{\lfloor (p-3)/2 \rfloor} \beta_{2j} f_{n-j} \right) + h_{n+1}^2 \left( \sum_{j=0}^{\lfloor (p-4)/2 \rfloor} \gamma_{2j} f'_{n-j} \right).
\]

An HBP of degree \( p-1 \) is used as predictor \( P_3 \) to obtain \( y_{n+c_3} \) to order \( p-2 \),

\[
(4) \quad y_{n+c_3} = y_n + h_{n+1} \left( \sum_{j=1}^{2} a_{3j} f_{n+c_j} + \sum_{j=1}^{\lfloor (p-3)/2 \rfloor} \beta_{3j} f_{n-j} \right) + h_{n+1}^2 \left( \sum_{j=0}^{\lfloor (p-4)/2 \rfloor} \gamma_{3j} f'_{n-j} \right).
\]

An HBP of degree \( p \) is used as integration formula IF to obtain \( y_{n+1} \) to order \( p \),

\[
(5) \quad y_{n+1} = y_n + h_{n+1} \left( \sum_{j=1}^{3} b_{1j} f_{n+c_j} + \sum_{j=1}^{\lfloor (p-3)/2 \rfloor} \beta_{1j} f_{n-j} \right) + h_{n+1}^2 \left( \sum_{j=0}^{\lfloor (p-4)/2 \rfloor} \gamma_{1j} f'_{n-j} \right).
\]
An HBP of degree $p$ is used as step control predictor $P_4$ to obtain $y_{n+1}$ to order $(p - 2)$,

$$
\bar{y}_{n+1} = y_n + h_{n+1} \left( \sum_{j=1}^{2} a_{4j} f_{n+c_j} + a_{43} f_{n+1} + \sum_{j=1}^{\lfloor (p-3)/2 \rfloor} \beta_{4j} f_{n-j} \right) \\
+ h_{n+1}^2 \left( \sum_{j=0}^{\lfloor (p-4)/2 \rfloor} \gamma_{4j} f'_{n-j} \right).
$$

We note that $f_n'$ is computed only once at each step $x_n$.

### 3 Order conditions for VSVO HBO(4-14)

As in similar search for ODE solvers, we impose the following Runge-Kutta-type simplifying assumptions [7], [18], [20] on HBO(4-14)3:

$$
\sum_{j=1}^{i-1} a_{ij} c_j + k! B_i(k + 1) = \frac{1}{k + 1} c^{k+1}_i, \quad \begin{cases} i = 2, 3, \\ k = 0, 1, 2, \ldots, p - 3, \end{cases}
$$

where

$$
B_i(j) = \sum_{\ell=1}^{\lfloor (p-3)/2 \rfloor} \beta_{i\ell} \frac{\eta_{\ell+1}^{j-1}}{(j-1)!} + \sum_{\ell=1}^{\lfloor (p-4)/2 \rfloor} \gamma_{i\ell} \frac{\eta_{\ell+1}^{j-2}}{(j-2)!}, \\
\begin{cases} i = 2, 3, \\ j = 0, 1, 2, \ldots, p, \end{cases}
$$

with $\eta_{p+1}^{-2} = 0$ and $\eta_{p+1}^{-1} = 0$ by notation, and

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

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

There remain two sets of equations to be solved:

$$
\sum_{i=1}^{3} b_{i1} c_i^k + k! B_1(k + 1) = \frac{1}{k + 1}, \quad k = 0, 1, \ldots, p - 1,
$$

$$
\sum_{i=2}^{3} b_{i1} \left[ \sum_{j=1}^{i-1} a_{ij} c_j^{p-2} + B_i(p - 1) \right] + B_1(p) = \frac{1}{p!}
$$
where
\[ B_1(j) = \sum_{i=1}^{[(p-3)/2]} \beta_{1i} \frac{\eta_{i+1}^{j-1}}{(j-1)!} + \sum_{i=1}^{[(p-4)/2]} \gamma_{1i} \frac{\eta_{i+1}^{j-2}}{(j-2)!}, \]
\[ j = 1, 2, \ldots, p+1. \]

We note that equations (10), for \( k = 0, 1, \ldots, p-2 \), are multistep-type order conditions. On the other hand, equation (10) for \( k = p - 1 \) and equation (11) are Runge-Kutta-type order conditions. The numbers \( B_1(k), B_2(k) \) and \( B_3(k) \) are associated with IF, \( P_2 \) and \( P_3 \), respectively.

4 Vandermonde-type formulation of HBO\((p)3\)

4.1 Integration formula IF The \( p \)-vector of the reordered coefficients of IF in (5),
\[ u^1 = [b_{11}, \gamma_{10}, b_{13}, b_{12}, \beta_{11}, \gamma_{11}, \beta_{12}, \gamma_{12}, \beta_{13}, \gamma_{13}, \ldots, \beta_{1_{[(p-3)/2]}}, \gamma_{1_{[(p-4)/2]}}]^T, \]
is the solution of the Vandermonde-type system of order conditions
\[ M^1 u^1 = r^1, \]
where
\[ M^1 = \begin{bmatrix}
1 & 0 & 1 & 1 & 1 & 0 & \cdots & 1 & 0 \\
0 & 1 & c_3 & c_2 & \eta_2 & 1 & \cdots & \eta_{[(p-3)/2]+1} & 1 \\
0 & 0 & \frac{c_3}{2!} & \frac{c_2}{2!} & \frac{\eta_2}{2!} & \eta_2 & \cdots & \frac{\eta_{[(p-3)/2]+1}}{2!} & \eta_{[(p-4)/2]+1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \frac{c_{p-1}^{p-1}}{(p-1)!} & \frac{c_{p-1}^{p-1}}{(p-1)!} & \frac{\eta_{p-1}^{p-1}}{(p-1)!} & \frac{\eta_{p-1}^{p-1}}{(p-1)!} & \cdots & \frac{\eta_{[(p-3)/2]+1}}{(p-1)!} & \frac{\eta_{[(p-4)/2]+1}}{(p-2)!} \\
\end{bmatrix} \]
and \( r^1 = r_1(1 : p) \) has components
\[ r_1(i) = \frac{1}{i!}, \quad i = 1, 2, \ldots, p. \]
The leading error term of IF is

\[
\frac{b_{13}}{p!} + \frac{c_{p}}{p!} + \sum_{j=1}^{(p-3)/2} \frac{\beta_{1j} \eta_{j}^{p+1}}{p!} + \sum_{j=1}^{(p-4)/2} \gamma_{1j} \frac{\eta_{j}^{p-1}}{(p-1)!} - \frac{1}{(p+1)!} h_{n+1}^{p+1} y_{n+1}^{(p+1)}.
\]

The detailed structure of columns 5 to \(p\) of \(M_1\) is as follows:

\[
M_1(i, j) = \begin{cases} 
  i! & \text{if } i = 1, 2, \ldots, p, \\
  0 & \text{if } j = 5, 7, \ldots, 2[p(p-1)/2]+1,
\end{cases}
\]

provided \(j + 1 \leq p\) in the second equation.

4.2 Predictor \(P_2\) The \((p-2)\)-vector of the reordered coefficients of \(P_2\) in (3),

\[
u_2 = \left[ a_{21}, \gamma_{20}, \beta_{21}, \gamma_{21}, \beta_{22}, \gamma_{22}, \beta_{23}, \gamma_{23}, \ldots, \beta_{2,[(p-3)/2]}, \gamma_{2,[(p-4)/2]} \right]^T,
\]

is the solution of the system of order conditions

\[M^2 u^2 = r^2,\]

where

\[M^2 = \begin{bmatrix}
1 & 0 & 1 & 0 & \cdots & 1 & 0 \\
0 & 1 & \eta_2 & 1 & \cdots & \eta_{[(p-3)/2]+1} & 1 \\
0 & 0 & \frac{\eta_{p-3}}{(p-3)!} & \eta_2 & \cdots & \eta_{[(p-3)/2]+1} & \eta_{[(p-4)/2]+1} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \frac{\eta_{p-3}}{(p-3)!} & \frac{\eta_{p-4}}{(p-4)!} & \cdots & \eta_{[(p-4)/2]+1} & \eta_{[(p-4)/2]+1}
\end{bmatrix}
\]

and \(r^2 = r_2(1 : p - 2)\) has components

\[r_2(i) = \frac{c_i^2}{i!}, \quad i = 1, 2, \ldots, p - 2.\]
The detailed structure of columns 3 to \( p \) of \( M^2 \in \mathbb{R}^{(p-2)\times(p-2)} \) is as follows:

\[
M^2(i, j) = \eta_{i-1}^{-j} / (i - 1)!,
\]

\[
M^2(i, j + 1) = \frac{d}{d\eta_{j+1}} M^2(i, j),
\]

\[
\begin{cases}
  i = 1, 2, \ldots, p - 2, \\
  j = 3, 5, \ldots, 2\lfloor (p - 3)/2 \rfloor + 1,
\end{cases}
\]

provided \( j + 1 \leq p - 2 \) in the second equation.

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

\[
\sum_{j=0}^{p+1} S_2(j) b_{n+1}^{j} y_n^{(j)}
\]

with coefficients

\[
(17) \quad S_2(j) = M^2(j, 1 : p - 2) u^2 = r_2(j) = \frac{c_j^2}{j!}, \quad j = 1, 2, \ldots, p - 2,
\]

\[
S_2(j) = \sum_{i=1}^{\lfloor (p-3)/2 \rfloor} \beta_{2i} \frac{\eta_{i+1}^{j-1}}{(j - 1)!} + \sum_{i=0}^{\lfloor (p-4)/2 \rfloor} \gamma_{2i} \frac{\eta_{i+1}^{j-2}}{(j - 2)!},
\]

\[
j = p - 1, p, p + 1.
\]

We see by (17) that \( P_2 \) is of order \( p - 2 \). The leading error term of \( P_2 \) is

\[
\left[ S_2(p - 1) - \frac{c_{p-1}^2}{(p - 1)!} \right] b_{n+1}^{p-1} y_n^{(p-1)}.
\]

### 4.3 Predictor \( P_3 \)

The \( (p - 1) \)-vector of the reordered coefficients of \( P_3 \) in (4),

\[
u^3 = [a_{31}, \gamma_{30}, a_{32}, \beta_{31}, \gamma_{31}, \beta_{32}, \gamma_{32}, \beta_{33}, \gamma_{33}, \ldots, \beta_{3,\lfloor (p-3)/2 \rfloor}, \gamma_{3,\lfloor (p-4)/2 \rfloor}]^T,
\]

is the solution of the system of order conditions

\[
(18) \quad M^3 u^3 = v^3,
\]
where

$$M^3 = \begin{bmatrix}
1 & 0 & 1 & 1 & 0 & \cdots & 1 & 0 \\
0 & 1 & c_2 & \eta_2 & 1 & \cdots & \eta_{[p-3]/2}+1 & 1 \\
0 & 0 & c_2^2 & \eta_2^2 & \eta_2 & \cdots & \eta_{[p-3]/2}+1 & \eta_{[p-4]/2}+1 \\
\vdots & & & & & & \vdots & \vdots \\
0 & 0 & c_3^{p-2}/(p-2)! & \eta_3^{p-2}/(p-2)! & \eta_3^{p-3}/(p-2)! & \cdots & \eta_{[p-3]/2}+1/(p-2)! & \eta_{[p-4]/2}+1/(p-3)! \\
\end{bmatrix}$$

and $r^3 = r_3(1:p-1)$ has components

$$r_3(i) = \frac{c_3^i}{i!}, \quad i = 1, 2, \ldots, p-2,$$

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

We see by (20) that $P_3$ is of order $p-2$. The last equation of (18) corresponds to order condition (11).

The detailed structure of columns 4 to $p-1$ of $M^3 \in \mathbb{R}^{(p-1) \times (p-1)}$ is as follows:

$$M^3(i, j) = \frac{\eta_{[j/2]}^{i-1}}{(i-1)!},$$

$$M^3(i, j+1) = \frac{d}{d\eta_{[j/2]}} M^3(i, j), \quad \begin{cases} i = 1, 2, \ldots, p-1, \\ j = 4, 6, \ldots, 2 \lfloor (p-1)/2 \rfloor, \end{cases}$$

provided $j + 1 \leq p - 1$ in the second equation.

A truncated Taylor expansion of the right-hand side of (4) about $x_n$ gives

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

with coefficients

$$S_3(j) = M^3(j, 1:p-1) u^3 = r_3(j) = \frac{c_3^j}{j!}, \quad j = 1, 2, \ldots, p-2,$$

$$S_3(j) = a_{32} S_2(j-1) + \sum_{i=1}^{\lfloor (p-3)/2 \rfloor} \beta_{3i} \frac{\eta_{i+1}^{j-1}}{(j-1)!}$$

$$+ \sum_{i=1}^{\lfloor (p-4)/2 \rfloor} \gamma_{3i} \frac{\eta_{i+1}^{j-2}}{(j-2)!}, \quad j = p-1, p, p+1.$$

4.4 Step control predictor $P_4$ The $p$-vector of the reordered coefficients of $P_4$ in (6) is

$$ \tilde{u}^4 = [a_{41}, \gamma_{40}, \beta_{41}, \gamma_{41}, \beta_{42}, \gamma_{42}, \ldots, \beta_{4,(p-3)/2}, \gamma_{4,(p-4)/2}, a_{43}, a_{42}]^T. $$

By setting $a_{43} = b_{13} + \omega_3$ and $a_{42} = b_{12} + \omega_2$, $\tilde{u}^4$ reduces to the $(p - 2)$-vector $u^4$ which is the solution of the system of order conditions

$$ M^4 u^4 = r^4, $$

where $M^4 = M^2$ and $r^4 = r_4(1 : p - 2)$ has components

$$ r_4(i) = \frac{1}{i!} - (b_{13} + \omega_3) \frac{r_{3,i-1}}{(i-1)!} - (b_{12} + \omega_2) \frac{r_{2,i-1}}{(i-1)!}, \quad i = 1, 2, \ldots, p - 2. $$

For arbitrary nonzero $\omega_3$ and $\omega_2$, $P_4$ yields $\tilde{y}_{n+1}$ to order $(p - 2)$. A good experimental choice is $\omega_3 = -0.025$ and $\omega_2 = 0.029$.

The column ordering in matrices (14), (16), and (19) makes it easy to go from order $p$ to order $p + 1$ simply by adding a bottom row and a far-right column. This ordering differs from the one used in [19].

5 Construction of elementary matrices A fast solution of Vandermonde-type systems (13), (15), (18) and (21), with coefficient matrices

$$ M^\ell \in \mathbb{R}^{m \times m}, \quad \ell = 1, 2, 3, 4, $$

where

$$ m_1 = p, \quad m_2 = p - 2, \quad m_3 = p - 1, \quad m_4 = p - 2. $$

can be achieved in $O(m_2^2)$ operations by decomposing $(M^\ell)^{-1}$ into the product of elementary matrices.

For $\ell = 1, 2, 3$, the construction of lower bidiagonal matrices $L^\ell$, initializing upper tridiagonal matrices $U^\ell$, and upper bidiagonal matrices $U^\ell$ for decomposing $(M^\ell)^{-1}$ is described in Section 5 and in the Appendix of [19] and will not be repeated here. This construction is most easily achieved by means of symbolic computation. The case $\ell = 4$ was reduced to the case $\ell = 2$ in Section 4.4.

The elementary matrix functions $L^\ell_k$ and $U^\ell_k$, $\ell = 1, 2, 3, 4$, are constructed only once as functions of $\eta_2, \ldots, \eta_6$. Then they are used by fast
Algorithm 4 in [19] to solve the systems $M^\ell u^\ell = r^\ell$, for $\ell = 1, 2, 3, 4$, in (13), (15), (18) and (21) at each integration step as described in Section 6.

Since the Vandermonde-type matrices $M^\ell$ can be decomposed into the product of a diagonal matrix containing reciprocals of factorials and a confluent Vandermonde matrix, the factorizations used in this paper hold following the approach of Björck and Pereyra [4], Krogh [15], Galimberti and Pereyra [10], and Björck and Elfving [3]. Thus pivoting is not needed in this stable $O(p^2)$ decomposition because of the special structure of Vandermonde-type matrices. This contrasts with the standard $O(p^3)$ Gaussian decomposition with pivoting which uses more storage, is slower and may become unstable in solving Vandermonde-type equations at very small stepsize.

6 Fast solution of Vandermonde-type systems for HBO($p$)3

In this section we describe $O(p^2)$ solutions to the Vandermonde-type systems for HBO($p$)3.

Firstly, the elimination procedure of Section 5.1 in [19] is applied to $M^\ell$ to construct $m_\ell \times m_\ell$ lower bidiagonal matrices $L_k^\ell$, $k = 3, \ldots, m_\ell - 1$, of the form

\[
L_k^\ell = \begin{bmatrix}
I_{k-1} & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 0 \\
0 & 1 & -\tau_{k+1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & 1 & -\tau_m
\end{bmatrix}
\]

defined by the multipliers

\[
\tau_i = \frac{i + 2 - k}{\mu_{\ell}(k)} = -L_k^\ell(i, i), \quad i = k + 1, k + 2, \ldots, m_\ell,
\]

where

\[
\mu_{\ell}(k) = \begin{cases}
M^\ell(2, k), & \text{if } M^\ell(1, k) = 1, \\
M^\ell(3, k), & \text{if } M^\ell(1, k) = 0, \\
\end{cases} \quad k = 3, 4, \ldots, m_\ell - 1.
\]

Left multiplying $M^\ell$ by $L_k^\ell$, $k = 3, \ldots, m_\ell - 1$, produces the upper triangular matrix

\[
L^\ell M^\ell = L_{m_\ell - 1}^\ell \cdots L_4^\ell L_3^\ell M^\ell
\]
Secondly, we construct the \( m_\ell \times m_\ell \) upper tridiagonal matrix \( U_1^\ell \) which transforms \( M^\ell \) into the matrix of first order divided differences \( M^\ell U_1^\ell \) of the columns of \( M^\ell \) whose first component is 1 and where the divisors are taken from the second row of \( M^\ell \). We note that \( U_1^4 = U_1^2 \) since \( M^4 = M^2 \).

For given \( p \) and \( \ell \), Table 1 lists the diagonal elements of \( U_1^\ell \) for \( i = 1, 2, \ldots, m_\ell \). The nonzero nondiagonal elements of \( U_1^\ell \) are given in the following three equations:

\[
\begin{align*}
U_1^1(i-2, i) &= -U_1^1(i, i), \\
& \quad i = 3, 7, 9, \ldots, m_1 - 1, \\
U_1^1(3, 4) &= -U_1^1(4, 4), \\
U_1^1(4, 5) &= -U_1^1(5, 5), \\
U_1^2(i-2, i) &= -U_1^2(i, i), \\
& \quad i = 3, 5, 7, \ldots, m_2 - 1, \\
U_1^3(i-2, i) &= -U_1^3(i, i), \\
& \quad i = 3, 6, 8, \ldots, m_3 - 1, \\
U_1^3(3, 4) &= -U_1^3(4, 4).
\end{align*}
\]
The remaining elements of \( U_1^\ell \) are zero. The first two rows of \( M^\ell U_1^\ell \) are

\[
L^\ell M^\ell U_1^\ell (1 : 2, 1 : m_{\ell}) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 1 \end{bmatrix}.
\]

Thirdly, the elimination procedure of Section 5.3 in [19] is used to construct \( m_{\ell} \times m_{\ell} \) upper bidiagonal matrices \( U_k^\ell \), \( k = 2, \ldots, m_{\ell} - 1 \), of the form

\[
U_k^\ell = \begin{bmatrix}
I_{k-1} & 0 & \cdots & 0 & \cdots & 0 & 0 \\
0 & 1 & -\sigma_{k+1} & 0 & \cdots & 0 & 0 \\
0 & 0 & \sigma_{k+1} & -\sigma_{k+2} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & \sigma_{m_{\ell}-2} & -\sigma_{m_{\ell}-1} & 0 \\
0 & 0 & 0 & \cdots & 0 & \sigma_{m_{\ell}-1} & -\sigma_{m_{\ell}} \\
0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{m_{\ell}}
\end{bmatrix}
\]

defined by the divisors

\[
\sigma_i = \frac{k}{\mu_{\ell}(i) - \mu_{\ell}(i-k)} = U_k^\ell(i, i), \quad i = k + 1, k + 2, \ldots, m_{\ell},
\]

where \( \mu_{\ell}(i) \) and \( \mu(i-k) \) are defined in (26). Right multiplying \( L^\ell M^\ell \)
by \( U_k^\ell \), \( k = 1, \ldots, m_{\ell} - 1 \), produces the diagonal matrix

\[
D^\ell = L^\ell_{m_{\ell}-1} L^\ell_{m_{\ell}-2} \cdots L^\ell_2 M^\ell U_1^\ell U_2^\ell \cdots U_{m_{\ell}-1}^\ell,
\]

where

\[
D^\ell(i, i) = 1, \quad i = 1, 2, 3,
\]

and

\[
D^\ell(i, i) = \frac{(i-1)!}{2^{\mu_{\ell}(3)} \cdots \mu_{\ell}(i-1)} = \frac{(i-1)!}{\mu_{\ell}(3) \cdots \mu_{\ell}(i-1)}, \quad i = 4, 5, \ldots, m_{\ell}.
\]

Lastly, \( M^\ell \) is decomposed into the product of elementary matrices:

\[
M^\ell = (L^\ell_{m_{\ell}-1} L^\ell_{m_{\ell}-2} \cdots L^\ell_3)^{-1} D^\ell (U_1^\ell U_2^\ell \cdots U_{m_{\ell}-1}^\ell)^{-1}
\]

and the solution of \( M^\ell u^\ell = r^\ell \) is

\[
u^\ell = U_1^\ell U_2^\ell \cdots U_{m_{\ell}-1}^\ell (D^\ell)^{-1} L_{m_{\ell}-1}^\ell L_{m_{\ell}-2}^\ell \cdots L_3^\ell r^\ell,
\]

where fast computation goes from right to left.
**Process 1.** Procedure (34) is implemented by the following two steps:

**Step 1** Algorithm 4 in [19] overwrites \( r^\ell = r_\ell(1 : m_\ell) \) with 
\[
U_2^\ell \cdots U_{m_{\ell-1}}^\ell (D^\ell)^{-1} L_{m_{\ell-1}}^\ell L_{m_{\ell-2}}^\ell \cdots L_3^\ell r^\ell
\]

in \( O(m_\ell^2) \) operations.

The input is \( M = M^\ell; m = m_\ell; r = r^\ell; L_k = L_k^\ell, k = 3, 4, \ldots, m_\ell - 1; U_k = U_k^\ell, k = 2, \ldots, m_\ell - 1; \) and \( D = D^\ell \).

**Step 2** For each value of \( \ell \), one of the following three cases is computed:

**Case 1** (\( \ell = 1 \)) The following iteration overwrites \( r^1 = r_1(1 : m_1) \) with \( U_1^1 r^1 \):

\[
\begin{align*}
r_1(3) &= r_1(3) U_1^1(3, 3), \\
r_1(4) &= r_1(4) U_1^1(4, 4), \\
r_1(i) &= r_1(i) U_1^1(i, i), & i = 5, 7, \ldots, \lfloor (m_1 + 1)/2 \rfloor - 1, \\
r_1(1) &= r_1(1) - r_1(3), \\
r_1(3) &= r_1(3) - r_1(4), \\
r_1(4) &= r_1(4) - r_1(5), & \text{if } m_1 \geq 5, \\
r_1(i) &= r_1(i) - r_1(i + 2), & i = 5, 7, \ldots, \lfloor (m_1 + 1)/2 \rfloor - 3.
\end{align*}
\]

**Case 2** (\( \ell = 2 \)) The following iteration overwrites \( r^2 = r_2(1 : m_2) \) with \( U_1^2 r^2 \):

\[
\begin{align*}
r_2(i) &= r_2(i) U_1^2(i, i), & i = 3, 5, 7, \ldots, \lfloor (m_2 + 1)/2 \rfloor - 1, \\
r_2(i) &= r_2(i) - r_2(i + 2), & i = 1, 3, 5, \ldots, \lfloor (m_2 + 1)/2 \rfloor - 3.
\end{align*}
\]

**Case 3** (\( \ell = 3 \)) The following iteration overwrites \( r^3 = r_3(1 : m_3) \) with \( U_1^3 r^3 \):

\[
\begin{align*}
r_3(3) &= r_3(3) U_1^3(3, 3), \\
r_3(i) &= r_3(i) U_1^3(i, i), & i = 4, 6, 8, \ldots, \lfloor m_3/2 \rfloor, \\
r_3(1) &= r_3(1) - r_3(3), \\
r_3(3) &= r_3(3) - r_3(4), & \text{if } m_3 \geq 4, \\
r_3(i) &= r_3(i) - r_3(i + 2), & i = 4, 6, 8, \ldots, \lfloor m_3/2 \rfloor - 2.
\end{align*}
\]
TRUONG NGUYEN-BA, ET AL.

<table>
<thead>
<tr>
<th>Order</th>
<th>α/4</th>
<th>α/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>HBO(p)3</td>
<td>ABM(p, p − 1)</td>
</tr>
<tr>
<td>4</td>
<td>−0.68</td>
<td>−0.97</td>
</tr>
<tr>
<td>5</td>
<td>−0.57</td>
<td>−0.70</td>
</tr>
<tr>
<td>6</td>
<td>−0.45</td>
<td>−0.52</td>
</tr>
<tr>
<td>7</td>
<td>−0.39</td>
<td>−0.39</td>
</tr>
<tr>
<td>8</td>
<td>−0.32</td>
<td>−0.30</td>
</tr>
<tr>
<td>9</td>
<td>−0.27</td>
<td>−0.22</td>
</tr>
<tr>
<td>10</td>
<td>−0.21</td>
<td>−0.17</td>
</tr>
<tr>
<td>11</td>
<td>−0.18</td>
<td>−0.13</td>
</tr>
<tr>
<td>12</td>
<td>−0.14</td>
<td>−0.11</td>
</tr>
<tr>
<td>13</td>
<td>−0.12</td>
<td>−0.03</td>
</tr>
<tr>
<td>14</td>
<td>−0.10</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 2: For order \( p \), the table lists the scaled abscissae of absolute stability for HBO(p)3 and ABM(p, p − 1).

7 Regions of absolute stability and principal error term  

The unscaled regions of absolute stability, \( R \), of HBO(p)3, \( p = 4, 5, \ldots, 14 \), shown in Figure 1 are obtained by applying the root condition (see [13, pp. 256–257]) to the characteristic equation

\[
\sum_{j=0}^{[(p-1)/2]} \gamma_j r^j = 0,
\]

which comes from an application of HBO(p)3 with constant \( h \) to the linear test equation

\[ y' = \lambda y, \quad y_0 = 1. \]

Let ABM(4-13) denote the family of ABM methods, ABM(p, p − 1), with predictor of order \( p - 1 \) and corrector of order \( p \) in PECE mode. Table 2 lists the scaled abscissae of absolute stability, \( \alpha/4 \) and \( \alpha/2 \), of HBO(4-14)3 and ABM(4-13) [24, p. 135–140], respectively. It is seen that HBO(p)3 has a larger scaled interval of absolute stability than ABM(p, p − 1) for \( p > 7 \).

The principal error term of HBO(p)3 is of the form

\[
\left[ \delta_1 \{ f^p \} + \delta_2 (p) \{ f^{-2} \} \{ f^p \} + \delta_3 \{ 2 f^{p-1} \} + \delta_4 \{ 3 f^{p-2} \} \right] h^{p+1},
\]
FIGURE 1: Unscaled regions of absolute stability of HBO(4-14).
where \( f^p \), \( \{ f^{p-2} \} \), \( \{ f^{p-1} \} \), \( \{ 3f^{p-2} \} \) are elementary differentials defined in [6], [16] and [12] and the principal local truncation error coefficients (PLTC) are \( [\delta_1, \delta_2, \delta_3, \delta_4] \). One may attempt to reduce the norm \( \| \text{PLTC} \|_2 \) by varying the parameters \( c_2 \) and \( c_3 \) in HBO(4-14)3.

The PLTC of ABM(\( p, p - 1 \)) are \( [\beta_kC_p, C_{p+1}] \) [17, p. 107]. Table 3 lists the principal local truncation error coefficients (PLTC) of HBO(3). Also listed are the scaled norms \( 4 \times \| \text{PLTC} \|_2 \) and \( 2 \times \| \text{PLTC} \|_2 \) for HBO(3) and ABM(\( p, p - 1 \)), respectively, the former being rapidly decreasing and being smaller that the latter.

### Table 3: For given order \( p \), the table lists the principal local truncation error coefficients (PLTC) of HBO(3) and the scaled norms \( 4 \times \| \text{PLTC} \|_2 \) and \( 2 \times \| \text{PLTC} \|_2 \) for HBO(3) and ABM(\( p, p - 1 \)), respectively.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_3 )</th>
<th>( \delta_4 )</th>
<th>HBO(3)</th>
<th>ABM(( p, p - 1 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>( \frac{1}{2} )</td>
<td>( \frac{1}{2} )</td>
<td>( \frac{1}{2} )</td>
<td>( \frac{1}{2} )</td>
<td>4.72e-02</td>
<td>2.86e-01</td>
</tr>
<tr>
<td>5</td>
<td>( \frac{1}{3} )</td>
<td>( \frac{1}{3} )</td>
<td>( \frac{1}{3} )</td>
<td>( \frac{1}{3} )</td>
<td>2.38e-02</td>
<td>2.44e-01</td>
</tr>
<tr>
<td>6</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>9.08e-03</td>
<td>2.18e-01</td>
</tr>
<tr>
<td>7</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>4.60e-03</td>
<td>2.00e-01</td>
</tr>
<tr>
<td>8</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>1.96e-03</td>
<td>1.83e-01</td>
</tr>
<tr>
<td>9</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>1.00e-03</td>
<td>1.75e-01</td>
</tr>
<tr>
<td>10</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>4.44e-04</td>
<td>1.65e-01</td>
</tr>
<tr>
<td>11</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>2.27e-04</td>
<td>1.57e-01</td>
</tr>
<tr>
<td>12</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>1.04e-04</td>
<td>1.51e-01</td>
</tr>
<tr>
<td>13</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>5.28e-05</td>
<td>1.45e-01</td>
</tr>
<tr>
<td>14</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
<td>2.46e-05</td>
<td>( \frac{1}{4} )</td>
</tr>
</tbody>
</table>

#### 8 Controlling stepsize and order
A variant of the procedure described in [24] is used to control the stepsize and order, \( p \), of VSVO HBO(4-14)3. For simplicity, in this section the order of the step control predictor \( P_4 \) will be denoted by \( q = p - 2 \).

- The program computes the maximum norm
  \[
  E_q = \| y_n - \bar{y}_{n,q} \|_\infty,
  \]
  where \( \bar{y}_{n,q} := \bar{y}_n \) is the value obtained by \( P_4 \).
- The stepsize \( h_{n+1} \) is obtained by the formula (see [14]):
  \[
  h_{n+1} = \min \left\{ h_{\text{max}}, \beta h_n \left( \frac{\text{tolerance}}{E_q} \right)^{1/\kappa}, 4 h_n \right\},
  \]

(37)
with $\kappa = p - 1$ and safety factor $\beta = 0.81$.

- The coefficients of IF, $P_2$, $P_3$ and $P_4$ are obtained successively as fast solutions of the linear systems (13), (15), (18) and (21).
- The step to $x_{n+1}$ is accepted if $E_q \leq$ tolerance, else it is rejected and the program returns to the previous step with smaller step $0.7 h_{n+1}$.
- If the step to $x_{n+1}$ is successful, besides $P_4$, three other step control predictors of orders $\rho = q \pm 1$ and $\rho = q - 2$, respectively, similar to $P_4$.

$$
\bar{y}_{n+1,\rho} = y_n + h_{n+1} \left[ \sum_{j=1}^{2} a_{4j} f_{n+c_j} + a_{43} f_{n+1} + \sum_{j=1}^{(\rho-1)/2} \beta_{4j} f_{n-j} \right] \\
+ h_{n+1}^2 \left[ \sum_{j=0}^{(\rho-2)/2} \gamma_{4j} f'_{n-j} \right],
$$

are used to control the order and stepsize by means of the following three maximum norms,

$$
E_{q \pm 1} = \| y_{n+1} - \bar{y}_{n+1,q,\pm 1} \|_\infty, \quad E_{q-2} = \| y_{n+1} - \bar{y}_{n+1,q-2} \|_\infty,
$$

which estimate the local error at $x_{n+1}$ had the step to $x_{n+1}$ been taken at orders $q \pm 1$ and $q - 2$, respectively.

To choose the lowest satisfactory order the following rules are used.

(a) The order is lowered if

$$
E_{q-1} \leq \text{min}\{E_q, E_{q+1}\} \quad \text{or} \quad E_q \geq \text{max}\{E_{q-1}, E_{q-2}\}.
$$

(b) The order is raised only if the following stronger conditions,

$$
E_{q+1} < E_q < \text{max}\{E_{q-1}, E_{q-2}\},
$$

are satisfied.

(c) When the order $q$ of $P_4$ is 12, $E_{q+1}$ is not available; thus, the order is lowered if

$$
E_q \geq \text{max}\{E_{q-1}, E_{q-2}\}.
$$

(d) When $q = 2$, the order is raised only if

$$
E_{q+1} < E_q.
$$
Then $\kappa$ and $E_q$ are reassigned according to the selected order $\kappa$. For example, if the order is to be lowered in the next step, $\kappa_{\text{new}} = \kappa_{\text{old}} - 1$ and $E_q = E_{q-1}$. The stepsize $h_{n+1}$ is then controlled by formula (37).

We summarize the procedure to advance integration from $x_n$ to $x_{n+1}$.

(a) The order $p$ is obtained by the above procedure. Then, the stepsize, $h_{n+1}$, is obtained by formula (37) with $\kappa = p - 1$.

(b) The numbers $\eta_2, \ldots, \eta_{[(p-3)/2]}+1$, defined in (9), are calculated.

(c) The coefficients of IF, $P_2, P_3$ and $P_4$ are obtained successively as solutions of systems (13), (15), (18) and (21), respectively.

(d) The values $y_{n+c_2}, y_{n+c_3}, y_{n+1},$ and $\tilde{y}_{n+1}$ are obtained by formulae (3)–(6).

(e) The step is accepted if $|y_{n+1} - \tilde{y}_{n+1}|$ is smaller than the chosen tolerance in which case the program goes to (a) with $n$ replaced by $n + 1$. Otherwise the program returns to (a) with the same order $p$ and smaller step $0.7h_{n+1}$.

9 Numerical results

9.1 Comparing HBO(4-14)3 against ode113 The numerical performance of HBO(4-14)3 programmed in Matlab and Matlab’s ode113 is compared on the following set of problems: Arenstorf’s orbits [1], the Brusselator and the Pleiades [12, pp. 244–249], Euler’s equation and the restricted three-body problem [24, pp. 232–259], and the following nonstiff DETEST problems: growth problem B1 of two conflicting populations, two-body problems D1–D5, and Van der Pol’s equation E2 with $\epsilon = 1$ [14].

HBO(4-14)3 is selfstarting with HBO(4)3. The initial step size, $h_1$, is chosen by a method similar to steps (a) and (b) of [12, p. 169].

Computations were performed on a Mac with a dual 2.5 GHz PowerPC G5 and 4 GB DDR SDRAM running under Mac OS X Version 10.4.6 and Matlab Version 7.0.4.352 (R14) Service Pack 2.

9.1.1 CPU against maximum global error The maximum global error is taken in the uniform norm,

$$MGE = \max_n \| y_n - z_n \|_{\infty},$$

where $y_n$ is the numerical value obtained by HBO(4-14)3 and $z_n$ is the “exact solution” obtained by Matlab’s ode113 with stringent tolerance $5 \times 10^{-14}$. 
FIGURE 2: CPU (horizontal axis) versus $\log_{10}(|MGE|)$ (vertical axis). HBO(4-14)3 ◇ and ode113 ▲.
TABLE 4: CPU percentage efficiency gain, CPU PEG, of HBO(4-14)3 over \texttt{ode113} for the listed problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>CPU PEG</th>
<th>Problem</th>
<th>CPU PEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arenstorf</td>
<td>42%</td>
<td>D1</td>
<td>12%</td>
</tr>
<tr>
<td>Brusselator</td>
<td>34%</td>
<td>D2</td>
<td>23%</td>
</tr>
<tr>
<td>Euler</td>
<td>13%</td>
<td>D3</td>
<td>37%</td>
</tr>
<tr>
<td>Pleiades</td>
<td>45%</td>
<td>D4</td>
<td>51%</td>
</tr>
<tr>
<td>Restricted 3-body</td>
<td>43%</td>
<td>D5</td>
<td>49%</td>
</tr>
<tr>
<td>B1</td>
<td>26%</td>
<td>E2</td>
<td>32%</td>
</tr>
</tbody>
</table>

In Figure 2, CPU (horizontal axis) is plotted versus $\log_{10}(|\text{MGE}|)$ (vertical axis) for six problems in hand.

9.1.2 CPU percentage efficiency gain of HBO(4-14)3 against \texttt{ode113}

The CPU percentage efficiency gain (CPU PEG) is defined by the formula (cf. Sharp [25]),

\begin{equation}
\text{(CPU PEG)}_i = 100 \left( \frac{\sum_j \text{CPU}_{2,ij}}{\sum_j \text{CPU}_{1,ij}} - 1 \right),
\end{equation}

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

The CPU time was obtained from the curves which fit the data ($\log_{10}(|\text{MGE}|), \log_{10}(\text{CPU})$) in a least-squares sense using, say, Matlab’s \texttt{polyfit}.

The CPU PEG for the set of problems in hand is listed in Table 4.

It is seen from Figure 2 and Table 4 that HBO(4-14)3 compares favorably with Matlab’s \texttt{ode113} on the basis of CPU versus $\log_{10}(|\text{MGE}|)$ for the set of problems in hand.

9.1.3 Maximum global error against tolerance

For six typical problems, time interval $[0, t_f]$ and given tolerance (TOL), Table 5 lists several numerical results related to the step control, namely: CPU time (CPU), number of function evaluations (NFE), number of failed attempts (REJ) and maximum global error (MGE) of HBO(4-14)3 and \texttt{ode113}. For HBO(4-14)3, NFE splits into 75% and 25% between $f$ and $f'$, respectively. Similar results hold for the other problems in hand.

For equivalent MGE, it is seen from Table 5 that HBO(4-14)3 uses less CPU time and fewer function evaluations than \texttt{ode113}. It is also
TABLE 5: For a given problem, time interval \([0, t_f]\) and tolerance (TOL), the table lists CPU time (CPU), number of functions evaluations (NFE), number of failed attempts (REJ) and maximum global error (MGE), in left column for HBO(4-14)3 and right column for \texttt{ode113}, respectively.

<table>
<thead>
<tr>
<th>Problem</th>
<th>TOL</th>
<th>CPU</th>
<th>NFE</th>
<th>CPU</th>
<th>NFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRUS</td>
<td>10^{-07}</td>
<td>3.12</td>
<td>2.57</td>
<td>360</td>
<td>736</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>5.55</td>
<td>4.95</td>
<td>1620</td>
<td>1359</td>
</tr>
<tr>
<td></td>
<td>10^{-12}</td>
<td>9.19</td>
<td>9.84</td>
<td>2780</td>
<td>2419</td>
</tr>
<tr>
<td>EULER</td>
<td>10^{-07}</td>
<td>0.17</td>
<td>0.15</td>
<td>1088</td>
<td>921</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>0.26</td>
<td>0.24</td>
<td>1860</td>
<td>1637</td>
</tr>
<tr>
<td>AREN</td>
<td>10^{-07}</td>
<td>0.12</td>
<td>0.15</td>
<td>612</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>0.28</td>
<td>0.27</td>
<td>1756</td>
<td>1637</td>
</tr>
<tr>
<td>PLEI</td>
<td>10^{-07}</td>
<td>0.36</td>
<td>0.31</td>
<td>964</td>
<td>765</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>0.93</td>
<td>0.85</td>
<td>2660</td>
<td>2408</td>
</tr>
<tr>
<td>E2</td>
<td>10^{-07}</td>
<td>0.11</td>
<td>0.11</td>
<td>572</td>
<td>448</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>0.22</td>
<td>0.22</td>
<td>1704</td>
<td>1457</td>
</tr>
<tr>
<td>I5</td>
<td>10^{-07}</td>
<td>0.34</td>
<td>0.30</td>
<td>2284</td>
<td>1992</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>0.52</td>
<td>0.50</td>
<td>3736</td>
<td>3463</td>
</tr>
</tbody>
</table>

seen that, for a given tolerance, the step control of both methods is problem dependent. There is a compromise between a sharper step control and the number of failed attempts. Similar results hold for the other problems considered in this paper.

9.2 Comparing HBO(4-14)3 against HBO(14)3 In this section, VSVO HBO(4-14)3 is shown to perform better than VS HBO(14)3 \cite{19} on some problems. The starting values for HBO(14)3 are obtained with DP(5,4)7FM \cite{8}.

In Figure 3, CPU is plotted versus \(\log_{10}(|MGE|)\) for Van der Pol’s equations for increasing \(\varepsilon = 3, 4, 5, 6\) \cite[pp. 111--115]{12}, with increasing CPU PEG = 6%, 17%, 26%, 27%, respectively.

In Figure 4, CPU is plotted versus \(\log_{10}(|MGE|)\) for DETEST’s B2, C1, E1, and E5 with CPU PEG = 37%, 24%, 12%, and 28%, respectively.

9.3 Comparing HBO(4-14)3 in C++ against DP(8,7)13M In Figure 5, CPU (horizontal axis) is plotted versus \(\log_{10}(|MGE|)\) for the Brusselator and the cubic wave equation \cite{5} as obtained by HBO(4-14)3 and DP(8,7)13M.

The CPU PEG’s, defined by formula (38), are 22% and 161% for
the Brusselator and the cubic wave over the intervals $[1e^{-4}, 1e^{-12}]$ and $[1e^{-2}, 1e^{-11}]$, respectively.

Similar to the test results in [9], it is seen from Figure 5 and the CPU PEG that for the cubic wave problem whose derivative evaluations are relatively expensive, HBO(4-14)3 wins over DP(8,7)13M.

10 Conclusion

A variable-step variable-order 3-stage Hermite-Birkhoff-Obrechko method, HBO(4-14)3, of order 4 to 14 was constructed by solving Vandermonde-type systems satisfying multistep-type and Runge-Kutta-type order conditions. The stepsize and order are controlled by four local error estimators. This method, in its vectorized Lagrange form, was tested on the Brusselator, Euler’s equation, Arenstorf’s orbits, the restricted three-body problem, the Pleiades, and the following nonstiff DETEST problems: two-body problems of class D, the growth problem of two conflicting populations of class B and Van der Pol’s equation of class E. Programmed in Matlab, HBO(4-14)3 was
FIGURE 4: CPU (horizontal axis) versus $\log_{10}(|\text{MGE}|)$ (vertical axis). HBO(4-14)○ and HBO(14)△.

FIGURE 5: CPU (horizontal axis) versus $\log_{10}(|\text{MGE}|)$ (vertical axis). HBO(4-14)○ and DP(8,7)△.
found to have lower global error, use less CPU time and require fewer function evaluations than the Matlab’s ode113. HBO(4-14)3 was also found to perform better than HBO(14)3 of order 14 on Van der Pol’s equations with $\varepsilon = 3, 4, 5, 6$, and DETEST’s problems B2, C1, E1 and E5. Programmed in C++, HBO(4-14)3 is superior to DP(8,7)13M in solving expensive equations over a long period of time.

Acknowledgment  The anonymous referee is deeply thanked for pointing out important modifications to this paper. Philip W. Sharp is heartily thanked for his suggestions which considerably improved the content and format of a preliminary version of this paper and for pointing out the cubic wave equation to the authors.

REFERENCES

5. P. J. Bryant, Nonlinear wave groups in deep water, manuscript.


**Department of Mathematics and Statistics, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5**

*E-mail address:* tnguyen@mathstat.uottawa.ca
*E-mail address:* hy.emails@gmail.com
*E-mail address:* zhangyujerry15@hotmail.com
*E-mail address:* remi@uottawa.ca