BIT
2000, Vol. 40, No. 3, pp. 497–512

0006-3835/00/4003-0497 $15.00
© Swets & Zeitlinger

DIAGONALIZABLE EXTENDED BACKWARD DIFFERENTIATION FORMULAS *

J. E. FRANK and P. J. VAN DER HOUWEN †

CWI, Kruislaan 413, P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands
email: J.E.Frank@cwi.nl, P.J.van.der.Houwen@cwi.nl

Abstract.

We generalize the extended backward differentiation formulas (EBDFs) introduced by Cash and by Psihoyios and Cash so that the system matrix in the modified Newton process can be block-diagonalized, enabling an efficient parallel implementation. The purpose of this paper is to justify the use of diagonalizable EBDFs on parallel computers and to offer a starting point for the development of a variable stepsize-variable order method. We construct methods which are L-stable up to order $p = 6$ and which have the same computational complexity per processor as the conventional BDF methods. Numerical experiments with the order 6 method show that a speedup factor of between 2 and 4 on four processors can be expected.

AMS subject classification: 65L06.

Key words: Initial-value problems, extended BDFs, parallelism.

1 Introduction.

In [7] we discussed the parallel implementation of the extended backward differentiation formulas (EBDFs) introduced by Cash in [2] and [3] for the numerical solution of initial value problems for stiff differential equations of the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}, \mathbf{f} \in \mathbb{R}^d, \quad t \geq t_0. \tag{1.1}$$

The parallel approach described in [7] is based on block-diagonalization of the system matrix in the modified Newton process used for solving the implicit EBDF relations. The system matrix is of the form $I - (A \otimes hJ)$, where h is the stepsize, I is the identity matrix, the matrix A is determined by the EBDF method coefficients, and J is an approximation to the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{y}$. Since *exact* block-diagonalization is not possible due to defectiveness of the matrix A, we applied *approximate* block-diagonalization. The resulting block system for the stages can then be efficiently solved in parallel on a number of processors equal to the number of stages. Although the rate of convergence is

*Received July 1999. Revised November 1999. Communicated by Stig Skelboe.

†Work carried out under project MAS 2.3—"Numerical Algorithms for Initial Value Problems". The investigations reported in this paper were partly supported by STW and TU Delft.

less than that of true modified Newton, the experiments in [7] show a speedup on a three-processor configuration of between 2 and 3.

The same parallel approach can be applied to the more general EBDF methods which have recently been proposed by Psihoyios and Cash [15]. These more general EBDF methods also lead to defective coefficient matrices A in the modified Newton process, but have the property that they can be made L-stable up to order $p = 6$ (the original EBDFs are L-stable up to order $p = 4$). However, approximate block-diagonalization is now much less accurate than in the case of the original EBDF methods. The aim of this paper is to construct methods which are L-stable up to order $p = 6$ with a nondefective matrix A, so that exact block-diagonalization is possible.

The code MEBDFDAE developed by Cash and Considine [4] is a variable stepsize, variable order implementation of the MEBDF method [3] suitable for integrating differential algebraic equations. This code performed quite well in a comparison with a number of other solvers in [14]. The extension of the results of this paper to differential algebraic equations is the subject of future research.

In Section 2, we define a family of EBDF-type methods which generalizes the Cash and Psihoyios–Cash methods. The order conditions, the global error for the Prothero–Robinson test equation, and stability conditions are derived. Section 3 discusses the sequential and parallel implementation of these methods and in Section 4 we derive L-stable, nondefective EBDF methods of order up to $p = 6$. Per processor, the computational complexity of these methods is comparable to that of the conventional BDF methods. Finally, Section 5 reports numerical experiments for the sixth-order method. These experiments indicate that a speedup factor in the range of 2 to 4 on four processors can be expected.

2 EBDF-type methods.

The generalizations of the EBDF methods to be discussed in this paper are of the form

$$
\begin{aligned}
&(B \otimes I)\mathbf{Y}_{n+1} - h(C \otimes I)\mathbf{F}(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_{n+1}) = (E \otimes I)\mathbf{V}_n, \\
&\mathbf{V}_n = (\mathbf{y}_{n-s+1}^T, \ldots, \mathbf{y}_n^T)^T.
\end{aligned} \tag{2.1}
$$

Here, $\otimes$ denotes the Kronecker product, h is the stepsize $t_{n+1} - t_n$, $\mathbf{e}$ and $\mathbf{c}$ are r-dimensional vectors, $\mathbf{e} = (1, \ldots, 1)^T$, $\mathbf{c} = (c_1, \ldots, c_r)^T$ with $c_r = 1$. I is the d by d identity matrix, B and C are r by r lower triangular matrices and E is an r by s matrix. The unknown stage vector $\mathbf{Y}_{n+1}$ contains r stages $\mathbf{y}_{n+c_i}$ of dimension d, representing numerical approximations at the points $t_n + c_i h$, and $\mathbf{F}(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_{n+1})$ contains the r right-hand side values $\mathbf{f}(t_n + c_i h, y_{n+c_i})$. Since B and C are lower triangular, the first $r - 1$ stage equations may be considered to be implicit *predictor formulas* providing the internal stage values $\mathbf{y}_{n+c_i}$, $i = 1, \ldots, r-1$, needed in the last stage equation. This last stage equation will be referred to as the *corrector equation* defining the output or step point value $y_{n+c_r} = y_{n+1}$.

We shall call (2.1) an *EBDF-type method*, because it can be viewed as a generalization of the original three-stage EBDF and MEBDF methods of Cash and the four-stage version recently discussed by Psihoyios and Cash. Note that the one-stage versions with $c_1 = 1$ assume the form of the conventional BDF methods.

Given the abscissa vector $\mathbf{c} = (c_i)$, the matrices B, C and E can be determined such that the ith stage equation in (2.1) is consistent of order p_i provided that $p_i + 1$ free coefficients are available for that equation. To formulate the consistency conditions, we first write (1.1) in autonomous form by adding the equation $dy_{d+1}/dt = 1$, so that (2.1) also becomes autonomous. Next, we introduce the abscissa vector for the back-values $\mathbf{b} := (1-s, 2-s, \ldots, 0)^T$, and the component-wise notation $g(\mathbf{v})$ associated with a scalar function $g : \mathbb{R} \to \mathbb{R}$ to denote the vector with components $g(v_i)$, where $\mathbf{v} = (v_i)$. An elementary derivation leads to the order equations

$$\text{(2.2)} \qquad \mathbf{e}_i^T E \mathbf{b}^j = \mathbf{e}_i^T (B\mathbf{c}^j - jC\mathbf{c}^{j-1}), \quad j = 0, \ldots, p_i, \quad i = 1, \ldots, r,$$

where $\mathbf{e}_i$ is the ith unit vector and where we define $0^0 = 1$. If (2.2) is satisfied, then the *stage order* of (2.1) is defined by $\bar{p} := \min\{p_i\}$. In general, the output value $\mathbf{y}_{n+c_r} = \mathbf{y}_{n+1}$ has *nonstiff* order of accuracy $p = \bar{p}$. However, if the first $r-1$ entries of the last row of the matrix B in (2.1) vanish (as will be the case for the methods of Section 4) and if $p_r = \bar{p} + 1$ (as will henceforth be assumed), then the nonstiff order of accuracy is equal to $\bar{p} + 1$.

We also study the global error of the EBDF-type method (2.1) when applied to the Prothero–Robinson equation $dy(t)/dt = \lambda y(t) + \phi(t)$, where ϕ is a given function. By means of this test equation we can obtain insight into the behavior of the error components in the integration of the general ODE system (1.1) by interpreting λ as an eigenvalue of the matrix J, where J denotes the Jacobian of the ODE system.

In [8] we proved the following result:

THEOREM 2.1. *Let $\bar{p}$ be the stage order of the EBDF-type method* (2.1). *Then, the global error of the Prothero-Robinson equation behaves according to* $\varepsilon_n \equiv \mathbf{y}(t_n) - \mathbf{y}_n = O(z^{-1}h^{\bar{p}+1})$ *as* $h \to 0$ *and* $z = h\lambda \to \infty$.

If the *stiff order* of accuracy is defined by the order of ε_n in h as $z \to \infty$, then we conclude from this theorem that the stiff order of EBDF-type methods is $\bar{p}+1$. In [14] the MEBDFDAE code of Cash and Considide [4] is compared with a number of codes based on standard BDF methods—for which the stiff order is equal to the nonstiff order [10], but which are less stable than EBDF-type methods—and Radau IIA methods—for which the nonstiff order of an s stage method is $2s-1$, but for which the stiff order is only $s+1$ [10]. The discussion of this section assumes only the general form (2.1) of EBDF-type methods, and thus applies directly to MEBDF; we think the high accuracy in the presence of stiffness helps to explain the good relative performance of MEBDFDAE observed in [14].

Finally, we mention a stability result for EBDF-type methods [8].

THEOREM 2.2. *Let B be nonsingular and let the row vectors of the matrix $B^{-1}E$ be denoted by $\mathbf{w}_i^T$. The EBDF-type method* (2.1) *is zero-stable if the equation $\zeta^s = \mathbf{w}_r^T\mathbf{\Gamma}(\zeta)$ has one simple root $\zeta_1 = 1$ and $s-1$ roots ζ_i, $i = 2,\dots,s-1$, on the unit disk with only simple roots on the unit circle.*

We note that this theorem holds for any general linear method of the form (2.1) such that the output (step point) value is given by one of the stages, regardless of the structures of the matrices B, C and E.

3 Sequential and parallel iteration.

The solution of (2.1) can be obtained by successively solving r subsystems, each of dimension d (recall that B and C are assumed to be lower triangular). If a (modified) Newton method is applied, then the iteration scheme for the ith stage $\mathbf{y}_{n+c_i}$ of $\mathbf{Y}_{n+1}$ assumes the form

$$(3.1)\quad (I - h\tilde{C}_{ii}J)(\mathbf{y}_{n+c_i}^{(j)} - \mathbf{y}_{n+c_i}^{(j-1)}) = -\mathbf{y}_{n+c_i}^{(j-1)} + h\tilde{C}_{ii}\mathbf{f}(t_{n+c_i}, \mathbf{y}_{n+c_i}^{(j-1)}) + h\sum_{k=1}^{i-1}\tilde{C}_{ik}\mathbf{f}(t_{n+c_k}, y_{n+c_k}) + \sum_{k=1}^{s}\tilde{E}_{ik}\mathbf{y}_{n-s+k}, \quad j = 1,\dots,m_i,$$

where $\tilde{C}_{ik}$ and $\tilde{E}_{ik}$ denote the entries of the matrices $B^{-1}C$ and $B^{-1}E$, respectively, J is an approximation to the Jacobian matrix of the right-hand side function in (1.1) at t_{n+1}, and $\mathbf{y}_{n+c_i}^{(0)}$ is an initial approximation to $\mathbf{y}_{n+c_i}$. This amounts to the solution of $\bar{m}r$ linear systems per step, where $\bar{m}$ denotes the (average) number of Newton iterations needed in the r subsystems. This approach will be called *sequential* iteration.

If, however, a parallel computer system is available, then one may attempt to solve the r stages more efficiently in parallel on r processors. In [7] we developed for the original EBDF and MEBDF methods of Cash a highly parallel iterative method for solving the implicit relations in (2.1). This parallel approach can also be applied to methods of the form (2.1) with more general matrices B, C and E. It is based on the approximate block-diagonalization of the modified Newton method applied to the full (block) system (2.1). Let us define the residue function

$$(3.2)\qquad \mathbf{R}_n(\mathbf{Y}) := \mathbf{Y} - h(B^{-1}C\otimes I)\mathbf{F}(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}) - (B^{-1}E\otimes I)\mathbf{V}_n.$$

Then, solving (2.1) by m modified Newton iterations amounts to

$$(3.3)\qquad (I - B^{-1}C\otimes hJ)(\mathbf{Y}_{n+1}^{(j)} - \mathbf{Y}_{n+1}^{(j-1)}) = -\mathbf{R}_n(\mathbf{Y}_{n+1}^{(j-1)}), \quad j = 1,\dots,m.$$

The MEBDF methods were developed in [3] to avoid factoring two Jacobians by forcing the diagonal of $B^{-1}C$ to be a constant. In parallel the extra factorizations come for free, and a constant diagonal is actually undesirable: if we use an abscissa vector of the form $\mathbf{c} = (1, 2, \dots, r-1, 1)^T$ and assume the same zero structure of the matrices B, C and E as in the original EBDF and MEBDF methods, then the matrix $B^{-1}C$ is defective, so that we cannot directly diagonalize (3.1) by applying a similarity transformation. One option is to replace the matrix $B^{-1}C$ in (3.3) by a diagonalizable approximation A^*, for example, by $A^* = \mathrm{diag}(B^{-1}C)$. The rate of convergence will be less than that of the modified Newton method, however. In the case of the three-stage EBDF and MEBDF methods, the loss in rate of convergence is modest (see the experiments in [7]) because the diagonalizable approximation is quite accurate. In fact, even with the simple choice $A^* = \mathrm{diag}(B^{-1}C)$, we obtained surprisingly fast convergence. However, for higher-stage methods, where diagonalizable approximations are less accurate, the rate of convergence is expected to decrease significantly.

3.1 Nondefective methods.

Rather than applying approximate block-diagonalization, we follow an alternative approach in which the abscissa vector is changed to the form

$$\mathbf{c} = (c_1, 2, 3, \dots, r-1, 1)^T$$

and in which we choose $c_1 \neq 1$ such that $B^{-1}C$ is no longer a defective matrix (except for the degenerate case $s < r-2$). We shall call such EBDF methods *nondefective EBDF methods.* Nondefective EBDF methods can directly be diagonalized by the transformation $\tilde{\mathbf{Y}}^{(j)} = (Q^{-1} \otimes I)\mathbf{Y}^{(j)}$, where Q is such that $Q^{-1}(B^{-1}C)Q = D$ with D diagonal. This yields the transformed iteration method

$$(3.4)\qquad \begin{aligned} (I - D \otimes hJ)(\tilde{\mathbf{Y}}^{(j)} - \tilde{\mathbf{Y}}^{(j-1)}) &= -\,(Q^{-1} \otimes I)\mathbf{R}_n\left((Q \otimes I)\tilde{\mathbf{Y}}^{(j-1)}\right), \\ j = 1, \dots, m, \quad \mathbf{Y}_{n+1} &= (Q \otimes I)\tilde{\mathbf{Y}}^{(m)} \end{aligned}$$

and will be called *parallel* iteration. We emphasize that (3.4) is algebraically equivalent to (3.3).

The introduction of the free parameter c_1 in the abscissa vector $\mathbf{c}$ preserves the attractive property that all stage values, except for the first one, can be reused in the initial approximation $\mathbf{Y}^{(0)}$ needed in the succeeding time step. The first stage can in turn be approximated by interpolating between the abundant history and stage values. In fact, setting $c_1 = 1$ has no additional advantages, because it 'duplicates' the output value at t_{n+1}.

3.2 Convergence condition.

Defining the iteration error $\varepsilon^{(j)} := \mathbf{Y}^{(j)} - \mathbf{Y}_{n+1}$, we derive for (3.3) the error recursion

$$\begin{aligned} \varepsilon^{(j)} &= hK\Phi(\varepsilon^{(j-1)}), \quad j = 1, \dots, m, \\ K &:= (I - B^{-1}C \otimes hJ)^{-1}(B^{-1}C \otimes I), \\ \Phi(\varepsilon) &:= \mathbf{F}(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_{n+1} + \varepsilon) - \mathbf{F}(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_{n+1}) - (I \otimes J)\varepsilon. \end{aligned}$$

CWI-Centrum voor Wiskunde en Informatica Amsterdam

Let $\Phi(\varepsilon)$ have at $\varepsilon = 0$ a Lipschitz constant L_Φ with respect to the Euclidean norm and let the problem be dissipative, i.e. $\mu_2[J] \le 0$, where $\mu_2[\cdot]$ denotes the logarithmic norm associated with the Euclidean norm. Then, by applying the matrix version of von Neumann's theorem (see [10, p. 356]), we conclude that for dissipative problems

$$\begin{aligned} &\|\varepsilon^{(j)}\|_2 \le hL_\Phi L_K \|\varepsilon^{(j-1)}\|_2, \\ &L_K = \max\{\|(I - zB^{-1}C)^{-1}B^{-1}C\|_2 : \mathrm{Re}(z) \le 0\}. \end{aligned} \tag{3.5}$$

Hence, for dissipative problems, a sufficient condition for convergence is

$$h \le \frac{1}{L_\Phi L_K}.$$

Thus, difference in convergence of two EBDF-type methods is mainly determined by differences in the upper bound L_K.

3.3 Analysis of computational expenses.

Finally, the computational expenses of (3.1) when implemented on one processor (sequential iteration of the subsystems) are compared with those of (3.4) implemented on r processors (parallel iteration). In (3.1) we define $\bar{m} := r^{-1}(m_1 + \cdots + m_r)$ and we denote the number of distinct diagonal entries of C by r_0. Table 3.1 lists the numbers of floating point operations to advance the solution one time step using a fixed stepsize. In this table, C_f and C_J respectively denote the average numbers of operations needed to compute a component of f and an entry of J.

Table 3.1: Operation costs per processor to advance the solution one time step.

	Sequential iteration	Parallel iteration
Once per Jacobian update		
Jacobian evaluation	$C_J d^2$	$\frac{1}{r} C_J d^2$
System matrix	$r_0 d$	d
LUD of system matrix	$\frac{2}{3} r_0 d^3$	$\frac{2}{3} d^3$
Once per time step		
Right-hand side	$(C_f + 2s + 1)rd - (C_f + 2)d$	$(2s - 1)d$
Per Newton iteration		
Forward/backward	$2rd^2$	$2d^2$
Updates	$r(C_f + 5)d$	$(C_f + r + 4)d$
Transformations	–	$2rd$

For a linear problem, only one Newton iteration is needed. Hence, assuming that the costs of building and factoring the Jacobian are negligible, it follows from Table 3.1 that the parallel speedup can be estimated by

$$S = r\frac{(2 - r^{-1})C_f + 2d + 2s + 6 - 2r^{-1}}{C_f + 2d + 2s + 3r + 3}.$$

At the other extreme, assume a very stiff nonlinear problem such that the Jacobian must be evaluated once per step. Then, we obtain

$$S = r\frac{\bar{m}(2d + C_f + 5) + (C_f + 2s + 1) + r^{-1}(C_Jd - C_f - 2) + r_0r^{-1}(1 + \frac{2}{3}d^2)}{m(2d + C_f + 3r + 4) + (2s - 1) + r^{-1}C_Jd + (1 + \frac{2}{3}d^2)},$$

from which the following observations can be made:

- If the evaluation of the Jacobian dominates the computation, then $S \approx r$.
- If factoring the Jacobian dominates the computation, then $S \approx r_0$.
- If the iterations dominate the computation, then $S \approx r\bar{m}m^{-1}$.

4 Construction of nondefective EBDF methods.

We shall construct nondefective versions of the original three-stage and four-stage EBDF-type methods given in [2] and [15].

4.1 Three-stage methods.

We consider methods of the form (2.1) with $r = 3$ and

$$c = \begin{pmatrix} c_1 \\ 2 \\ 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 0 \\ B_{21} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & 0 & 0 \\ 0 & C_{22} & 0 \\ C_{31} & C_{32} & C_{33} \end{pmatrix},$$

$$E = \begin{pmatrix} E_{11} & E_{12} & \cdots & E_{1s} \\ 0 & E_{22} & \cdots & E_{2s} \\ E_{31} & E_{32} & \cdots & E_{3s} \end{pmatrix}. \tag{4.1}$$

Given the abscissa c_1 and one of the parameters C_{3j}, the remaining entries in the arrays in (4.1) can be computed by means of the order conditions (2.2) such that $p_1 = p_2 = s$ and $p_3 = s + 1$. Hence, the order of accuracy (both stiff and nonstiff) is $p = s + 1$. The cases $\{c_1 = 1, C_{31} = 0\}$ and $\{c_1 = 1, C_{33} = C_{11} = C_{22}\}$ respectively define the original EBDF and MEBDF methods. For future reference, Table 4.1 lists for $p = 3, \ldots, 6$ the MEBDF values of the angle of unconditional stability α; the parameters D_1 and D_2 determining the rectangle $\{z : -D_1 \leq \Re(z) \leq 0, -D_2 \leq \Im(z) \leq D_2\}$ containing the region of instability in the left half-plane; and the maximal modulus of the characteristic roots ζ in this region of instability. For larger values of p, the angle α quickly decreases, so that the resulting integration methods are less useful for solving general stiff problems.

As we already observed, the MEBDF methods of Table 4.1 are defective, so that direct diagonalization is not possible. Therefore, we used the two free parameters c_1 and C_{31} to construct a nondefective, zero-stable and L(α)-stable EBDF method with (i) a relatively large α and (ii) a well-conditioned transformation matrix Q. Requiring that Q be lower triangular with unit diagonal entries, we found by a straightforward numerical search the results listed in

Table 4.1: Three-stage MEBDF methods of Cash with $c_1 = 1$, $C_{33} = C_{11} = C_{22}$.

p	3	4	5	6
α	90°	90°	88.4°	83.1°
(D_1, D_2)	(0,0)	(0,0)	(0.040, 1.8)	(0.246, 2.6)
$\|\zeta\|_{\max}$	1	1	1.029	1.121

Table 4.2 (for the L-stable third- and fourth- order methods, the generating matrices $B^{-1}C$, $B^{-1}E$, D and Q needed in (3.4) are given in the Appendix to this paper). We mention only that there is a lot of freedom in choosing the parameters c_1 and C_{31} to determine L-stable 3-stage methods satisfying (i) and (ii). For the 4-stage methods of the next subsection, the L-stable parameter space is much more restricted.

Table 4.2: Three-stage, nondefective EBDF methods of the form (4.1).

p	3	4	5	6
c_1	5/4	5/4	5/4	5/4
C_{31}	0	0	2/7	3/13
$\|\|Q\|\|_\infty$	6.1	7.9	6.6	8.3
α	90°	90°	88.5°	83.9°
(D_1, D_2)	(0, 0)	(0, 0)	(0.04, 2.1)	(0.24, 3.9)
$\|\zeta\|_{\max}$	1	1	1.029	1.121

4.2 Higher-stage methods.

The original EBDF and MEBDF methods have $c = (1, 2, 1)^T$, so that there is one 'future point' at $t_n + 2h$. This method can be interpreted as the successive application of the s-step BDF formula at $t_n + h$ and $t_n + 2h$ for predicting the future point value at $t_n + 2h$ needed in the $(s + 1)$-step (M)EBDF corrector formula. More generally, we may introduce further future points by using $c = (1, 2, 3, \ldots, r - 1, 1)^T$. Considering only the stability of the corrector formula (last stage equation), we verified experimentally that up to order 18 the maximal order of L-stable formulas increases by 2 and the maximal order of L(α)-stable formulas increases by 3 with each additional future point. Of course, the use of BDF predictors will reduce the stability of the overall method, but we may still hope for improvement: Psihoyios and Cash [15] have shown that there exist L-stable 4-stage methods of order 6. However, just as in the case of the three-stage EBDF, choosing $c = (1, 2, 3, \ldots, r - 1, 1)^T$ yields defective matrices $B^{-1}C$. Therefore, we shall consider abscissae vectors of the form

$$c = (c_1, 2, 3, \ldots, r - 1, 1)^T, \tag{4.2a}$$

where c_1 is a free parameter. According to the structure of the original (M)EBDF methods, we impose the following sparsity pattern on the matrices B, C and E:

$$B := \begin{pmatrix} 1 & & & & \\ * & 1 & & & \\ \vdots & \vdots & \ddots & & \\ * & * & \cdots & 1 & \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \qquad C := \begin{pmatrix} * & & & & \\ & * & & & \\ & & \ddots & & \\ & & & * & \\ * & * & \cdots & * & * \end{pmatrix},$$

$$E := \begin{pmatrix} * & * & \cdots & * & \cdots & * \\ & * & \cdots & * & \cdots & * \\ & & \ddots & & & \vdots \\ & & & * & \cdots & * \\ * & * & \cdots & * & \cdots & * \end{pmatrix}. \tag{4.2b}$$

The entries in the matrices B, C and E can be determined such that the first $r-1$ stage equations in (4.2b) are consistent of order s. The last stage equation contains $r+s$ free parameters, so that it can be made consistent of order $r+s-1$. Since the order of accuracy of (4.2a, 4.2b) cannot exceed $s+1$, we shall choose the entries in the corrector equation such that it is consistent of order $s+1$, leaving $r-2$ free parameters. Together with the free parameter c_1, we obtain an $(r-1)$-parameter family of EBDF-type methods with stage order $\bar{p} = s$ and order of accuracy $p = s+1$. From this family, we want nondefective, L-stable methods, again under the condition of zero-stability and a well-conditioned transformation matrix Q.

Let us consider the case of four stages ($r = 4$) with three free parameters. As already mentioned, Psihoyios and Cash have considered the defective case $c_1 = 1$ and shown that L-stable, sixth-order methods exist for a particular choice of the remaining two free parameters. For example, they verified that the parameters $C_{41} = 1/10$ and $C_{43} = 1/20$ generate an L-stable method with $p = s+1 = 6$. This motivated us to search for nondefective, L- and zero-stable methods by choosing $c_1 \neq 1$. A numerical search produced for $p = s+1 = 5$ the values $c_1 = 3/2$, $C_{41} = 3/10$, $C_{43} = 7/50$ giving $\|Q\|_\infty \approx 31.5$ and for $p = s+1 = 6$ the values $c_1 = 6/5$, $C_{41} = 11/100$, $C_{43} = 1/20$, giving $\|Q\|_\infty \approx 167.5$. The corresponding generating matrices $B^{-1}C$, $B^{-1}E$, D and Q needed in (3.4) are given in the Appendix.

Together with the conventional BDF methods of order $p = 1$ and $p = 2$, and the three-stage nondefective EBDF methods of order $p = 3$ and $p = 4$ derived in the preceding subsection, we now have L-stable methods up to order six, all having a comparable effective computational complexity per step, provided that we employ three processors for $p = 3, 4$ and four processors for $p = 5, 6$.

5 Numerical experiments.

Preliminary numerical experiments have been conducted using a constant stepsize implementation in MATLAB. Due to the difficulty of determining the free coefficients for optimal stability, a variable stepsize implementation should be based on interpolation of back-values to maintained evenly-spaced data. One could also recompute the coefficients at each timestep for fixed values of the free parameters, but this is likely to lead to a loss of L-stability at high order. In fact, this paper is meant as a starting point for the development of a variable stepsize-variable order code.

In the numerical experiments we compare two methods from the three-parameter family of four-stage, 6th-order EBDF-type methods of the form (4.2a, 4.2b) with free parameters c_1, C_{41} and C_{43}. The first method is due to Psihoyios and Cash and is defined by $c_1 = 1$, $C_{41} = 0.10$, $C_{43} = 0.05$. It is L-stable, but defective, so that sequential iteration has to be applied (see Section 2.4). The second method is defined by $c_1 = 1.2$, $C_{41} = 0.11$, $C_{43} = 0.05$. It also is L-stable, but nondefective, so that the parallel iteration method (3.4) can be applied. In the following, we call these methods the Defective and Nondefective EBDF methods, respectively. The values of the parameter L_K in (3.5) are $L_K \approx 1.88$ for the Defective EBDF method and $L_K \approx 1.68$ for the Nondefective EBDF method, so we would expect the methods to have similar convergence behaviors. In addition to these methods, we reproduced the results from [7] obtained for the original three-stage, 6th-order EBDF method of Cash when iterated by the diagonal iteration method (3.2) with $A^* = \operatorname{diag}(B^{-1}C)$, to be referred to as Diagonal EBDF. By mutual comparison of the three methods we can see what we have gained by the introduction of nondefective EBDF methods.

Following [7] the initial iterates for the iteration processes are obtained by taking the most recent approximation available or, if not yet available (in the case of the future value at t_{n+r-1} and at t_{n+c_1}), by 6-point extrapolation of already computed approximations. The Jacobian matrix J is evaluated in each step using the future-point-approximation to y_{n+1} from the preceding step. The starting values were obtained either from the exact solution (if available) or by applying the 5th-order Radau IIA method with a 5 times smaller stepsize and using 10 Newton iterations per step.

Three of the test problems are the same as in [7], viz. the Kaps problem [13]

$$\begin{aligned} &\frac{dy_1}{dt} = -1002y_1 + 1000y_2^2, \\ &\frac{dy_2}{dt} = y_1 - y_2(1 + y_2), \\ &y_1(0) = y_2(0) = 1, \quad 0 \le t \le 5; \end{aligned} \tag{5.1}$$

the eight-dimensional 'High Irradiance RESponse' problem given in [10, p. 157]:

$$\text{HIRES on the interval } [5, 321.8122], \tag{5.2}$$

where the initial conditions at $t = 5$ were obtained by applying the RADAU 5 code [11] on $[0, 5]$; and the non-autonomous Robertson problem, modified to remove the transient phase and make the problem suitable for fixed stepsize integration:

$$(5.3)\qquad \begin{aligned} y_1' &= -0.04y_1 + 10^4 y_2 y_3 - 0.96\mathrm{e}^{-t}, & y_1(0) &= 1, \\ y_2' &= 0.04y_1 - 10^4 y_2 y_3 - 10^7 (y_2)^2 - 0.04\mathrm{e}^{-t}, & y_2(0) &= 0, \quad 0 \le t \le 1. \\ y_3' &= 3 \times 10^7 (y_2)^2 + \mathrm{e}^{-t}, & y_3(0) &= 0, \end{aligned}$$

The fourth test problem is the 15-dimensional circuit analysis problem due to Horneber [12] and extensively discussed in [6] and [9]. In our implementation, we used the specification given in [14]:

(5.4) Ring modulator on the interval $[0, 10^{-3}]$ with $C_s = 10^{-9}$.

In our numerical experiments, we denoted the number of steps by N, the number of iterations in each iteration process by m, and the total number of iterations by M (not including the iterations needed to compute the starting values). Note that for fixed values of m and N, a serial implementation of Defective EBDF requires—per processor—four times as many right-hand side evaluations and forward-backward substitutions as a 4-processor parallel implementation of the Nondefective EBDF method, because Defective EBDF solves four subsystems per step. Hence, we would expect the value of M to be four times greater for Defective EBDF. The accuracy is given by the number of significant correct digits *scd*; that is, we write the maximal *absolute* end point error in the form 10^{-scd}. In the tables of results, we shall indicate negative *scd*-values by *.

As a basis for comparison of parallel performance, we list here the rough speedup results obtained by two other parallel ODE solvers. Bendtsen [1] reports speedups of between 3 and 5 on a 9-processor implementation of an eighth order multiple implicit Runge–Kutta method. In [5], de Swart reports speedups of between 2 and 4 from a 4-processor implementation of the fifth order Radau IIA method.

5.1 Fixed numbers of iterations.

Tables 5.1, 5.2 and 5.3 list for given values of m and N the resulting *scd*-values for the first three problems (5.1)–(5.3). These results show that the three methods converge to solutions with comparable accuracy. Furthermore, the convergence rate is for Diagonal EBDF slightly less than for the other two methods.

5.2 Variable number of iterations.

For our dynamic iteration strategy, we used the stopping criterion described in [10, p. 130], replacing the tolerance parameter *Tol* with an estimate of the local truncation error *LTE* (see [7] for details on this modification.) At the timestep $(n+1)$, *LTE* was defined to be the difference between the order p approximation to $\mathbf{y}_n$ from the last stage of $\mathbf{Y}_n$ and—in the case of Defective EBDF—the order $p-1$ approximation from the first stage of $\mathbf{Y}_n$ or—in the Nondefective EBDF

Table 5.1: Values of *scd* for the Kaps problem (5.1).

N	Method	$m=1$	$m=2$	$\cdots$	$m=\infty$
10	Defective EBDF	5.0		$\cdots$	5.0
	Nondefective EBDF	5.2		$\cdots$	5.2
	Diagonal EBDF	*	4.7	$\cdots$	4.5
20	Defective EBDF	6.8		$\cdots$	6.8
	Nondefective EBDF	6.9		$\cdots$	6.9
	Diagonal EBDF	*	6.4	$\cdots$	6.3
40	Defective EBDF	8.5		$\cdots$	8.5
	Nondefective EBDF	8.8		$\cdots$	8.8
	Diagonal EBDF	*	8.2	$\cdots$	8.1

Table 5.2: Values of *scd* for the HIRES problem (5.2).

N	Method	$m=1$	$m=2$	$m=3$	$m=4$	$\cdots$	$m=\infty$
10	Defective EBDF	3.1	3.0	2.0	3.2	$\cdots$	3.1
	Nondefective EBDF	3.4	3.0	2.0	2.9	$\cdots$	2.8
	Diagonal EBDF	*	2.8	2.5	2.7	$\cdots$	2.7
20	Defective EBDF	2.6	3.7	3.9	3.9	$\cdots$	3.8
	Nondefective EBDF	3.6	3.7	3.6	3.6	$\cdots$	3.6
	Diagonal EBDF	*	3.6	3.4	3.3	$\cdots$	3.3
40	Defective EBDF	5.0	4.8	4.9		$\cdots$	4.9
	Nondefective EBDF	4.4	4.7	4.8		$\cdots$	4.8
	Diagonal EBDF	*	4.4	4.3		$\cdots$	4.3

Table 5.3: Values of *scd* for the Robertson problem (5.3).

N	Method	$m=1$	$m=2$	$\cdots$	$m=\infty$
10	Defective EBDF	7.6		$\cdots$	7.6
	Nondefective EBDF	7.7		$\cdots$	7.7
	Diagonal EBDF	7.8	7.9	$\cdots$	7.9
20	Defective EBDF	9.3		$\cdots$	9.3
	Nondefective EBDF	9.3		$\cdots$	9.3
	Diagonal EBDF	*	9.6	$\cdots$	9.6
40	Defective EBDF	11.0		$\cdots$	11.0
	Nondefective EBDF	11.0		$\cdots$	11.0
	Diagonal EBDF	*	11.3	$\cdots$	11.3

case—the order $p-1$ approximation from the second stage of $\mathbf{Y}_{n-1}$. All further iteration strategy parameters are the same as in [7].

For the three most difficult problems (5.2), (5.3) and (5.4), we performed experiments in which the number of steps was chosen such that a prescribed *scd*-value was obtained. For these problems, the *maximal* number of Newton iterations in the subsequent iteration processes was limited to 10. Tables 5.4, 5.5 and 5.6 list the total number of iterations M needed to obtain a given *scd*-value. From these values, we may conclude that the two parallel methods Nondefective EBDF and Diagonal EBDF need about two to four times fewer iterations then the sequential method Defective EBDF.

Table 5.4: Values of M for the HIRES problem (5.2).

Method	$scd = 4$	5	6	7
Defective EBDF	96	168	573	928
Nondefective EBDF	73	102	195	343
Diagonal EBDF	83	133	189	241

Table 5.5: Values of M for the Robertson problem (5.3).

Method	$scd = 8$	9	10	11	12	13
Defective EBDF	36	63	107	169	265	415
Nondefective EBDF	10	18	29	47	74	123
Diagonal EBDF	9	17	29	49	74	114

Table 5.6: Values of M for the Ring modulator (5.4).

Method	$scd = 6$	7	8
Defective EBDF	49900	72400	104500
Nondefective EBDF	14800	22300	33800
Diagonal EBDF	20000	29700	42800

Acknowledgements.

The authors are grateful to Jeff Cash for his interest and his comments on this work, and for drawing our attention to [15].

A Coefficients of some nondefective EBDF methods.

For reference we provide the coefficient matrices of the L-stable nondefective EBDF methods considered in this paper. For each method we give the matrices $B^{-1}C$, $B^{-1}E$ and Q needed for parallel implementation of (3.4). Obviously, the diagonal matrix D needed for the implementation of 3.4 is given by $D = \text{diag}(B^{-1}C)$. The coefficients listed are exact, expressed in fractional form, and were determined by Maple.

The 3-stage L-stable method of order $p = 3$ is defined by $c_1 = 5/4$, $C_{31} = 0$.

The method coefficients and transformation matrix are given by

(A.1)

$$B^{-1}C = \begin{pmatrix} \frac{45}{56} & 0 & 0 \\ \frac{72}{77} & \frac{6}{11} & 0 \\ 0 & -\frac{4}{23} & \frac{22}{23} \end{pmatrix}, \quad B^{-1}E = \begin{pmatrix} -\frac{25}{56} & \frac{81}{56} \\ -\frac{40}{77} & \frac{117}{77} \\ -\frac{5}{23} & \frac{28}{23} \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 \\ \frac{192}{53} & 1 \\ \frac{43008}{10441} & \frac{11}{26} \end{pmatrix}$$

A 3-stage L-stable method of order $p = 4$ is defined by $c_1 = 5/4$, C_{31}
The method coefficients and transformation matrix are given by

$$B^{-1}C = \begin{pmatrix} \frac{585}{908} & 0 & 0 \\ \frac{192}{227} & \frac{6}{13} & 0 \\ 0 & -\frac{18}{197} & \frac{150}{197} \end{pmatrix}, \quad B^{-1}E = \begin{pmatrix} \frac{2025}{7264} & -\frac{4225}{3632} & \frac{13689}{7264} \\ \frac{1080}{2951} & -\frac{4204}{2951} & \frac{6075}{2951} \\ \frac{17}{197} & -\frac{99}{197} & \frac{279}{197} \end{pmatrix}$$

(A.2)

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ \frac{3328}{719} & 1 & 0 \\ \frac{18130944}{5022215} & \frac{39}{128} & 1 \end{pmatrix}.$$

A 4-stage L-stable method of order $p = 5$ is defined by $c_1 = 3/2$, $C_{41} =$
$C_{43} = 7/50$. The method coefficients and transformation matrix are given

$$B^{-1}C = \begin{pmatrix} \frac{315}{496} & 0 & 0 & 0 \\ \frac{864}{1147} & \frac{12}{37} & 0 & 0 \\ \frac{2768}{3441} & \frac{32}{37} & \frac{4}{9} & 0 \\ \frac{3}{10} & -\frac{3059487}{4001600} & \frac{7}{50} & \frac{5279163}{4001600} \end{pmatrix},$$

(A.3)

$$B^{-1}E = \begin{pmatrix} -\frac{1225}{3968} & \frac{6075}{3968} & -\frac{11907}{3968} & \frac{11025}{3968} \\ -\frac{420}{1147} & \frac{2043}{1147} & -\frac{3884}{1147} & \frac{3408}{1147} \\ -\frac{12110}{30969} & \frac{2118}{1147} & -\frac{3907}{1147} & \frac{91382}{30969} \\ \frac{2153579}{24009600} & -\frac{3413921}{8003200} & \frac{4631823}{8003200} & \frac{3640463}{4801920} \end{pmatrix},$$

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{4608}{1901} & 1 & 0 & 0 \\ \frac{24616704}{1617751} & -\frac{36}{5} & 1 & 0 \\ -\frac{38599642812960}{45767552496101} & \frac{145802607}{81838795} & -\frac{5042016}{31506067} & 1 \end{pmatrix}.$$

A 4-stage L-stable method of order $p = 6$ is defined by $c_1 = 6/5$, $C_{41} = 11/100$, $C_{43} = 1/20$. The method coefficients and transformation matrix are given by

$$B^{-1}C = \begin{pmatrix} \frac{16016}{32525} & 0 & 0 & 0 \\ \frac{40625}{49438} & \frac{15}{38} & 0 & 0 \\ \frac{39040625}{41626796} & \frac{30375}{31996} & \frac{180}{421} & 0 \\ \frac{11}{100} & -\frac{120153318}{388515625} & \frac{1}{20} & \frac{1497086157}{1554062500} \end{pmatrix},$$

$$B^{-1}E = \begin{pmatrix} \frac{569184}{4065625} & -\frac{10469888}{12196875} & \frac{9018009}{4065625} & -\frac{12719616}{4065625} & \frac{32064032}{12196875} \\ \frac{5775}{24719} & -\frac{101768}{74157} & \frac{82350}{24719} & -\frac{105400}{24719} & \frac{227750}{74157} \\ \frac{5549775}{20813398} & -\frac{46526500}{31220097} & \frac{70906923}{20813398} & -\frac{42611025}{10406699} & \frac{90894625}{31220097} \\ -\frac{211339877}{6216250000} & \frac{939457771}{4662187500} & -\frac{168763034}{388515625} & \frac{333046763}{1554062500} & \frac{19629003023}{18648750000} \end{pmatrix}$$

(A.4)

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1015625}{120733} & 1 & 0 & 0 \\ \frac{7376452890625}{53619698494} & -\frac{405}{14} & 1 & 0 \\ -\frac{4755875950106507681468 75}{5105209189934884057295 8} & \frac{241922892409}{78349451754} & -\frac{32713015625}{350542022097} & 1 \end{pmatrix}.$$

REFERENCES

1. C. Bendtsen, *A parallel stiff ODE solver based on MIRKs*, Adv. in Comp. Math., 7 (1997), pp. 27–36.
2. J. R. Cash, *On the integration of stiff ODEs using extended backward differentiation formulae*, Numer. Math., 34 (1980), pp. 235–246.
3. J. R. Cash, *The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae*, Comput. Math. Appl., 5 (1983), pp. 645–657.
4. J. R. Cash and S. Considine, *An MEBDF code for stiff initial value problems*, ACM Trans. Math. Software, 18 (1992), pp. 142–155. Code available at `http://www.ma.ic.ac.uk/~jcash/IVP_software/finaldae/readme.html`.
5. J. J. B. de Swart, *Parallel Software for Implicit Differential Equations*, PhD thesis, University of Amsterdam, 1997.
6. G. Denk and P. Rentrop, *Mathematical models in electric circuit simulation and their numerical treatment*, in Proc. of the Conference "Numerical Treatment of Differential Equations", Halle (DDR). Teubner, 1989.
7. J. E. Frank and P. J. van der Houwen, *Parallel iteration of backward differentiation formulas*, to appear in IMA J. Numer. Anal., 2000.
8. J. E. Frank and P. J. van der Houwen, *Diagonalizable extended backward differentiation formulas*, Tech. Report MAS-R9917, CWI, Amsterdam, 1999. Available at `http://www.cwi.nl/static/publications/reports/MAS-1999.html`.
9. E. Hairer, C. Lubich, and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge–Kutta Methods*, Lecture Notes in Mathematics, 1409. Springer-Verlag, Berlin, 1989.
10. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations, II. Stiff and Differential-Algebraic Problems.* Springer-Verlag, Berlin, 1991.
11. E. Hairer and G. Wanner, *Radau.* Available at `ftp://ftp.unige.ch/pub/doc/math/stiff/radau.f`, 1998.
12. E. H. Horneber, *Analysis of of nonlinear RCLÜ circuits by means of mixed potential functions and a systematic representation of nonlinear dynamic circuit analysis*, (in German) PhD thesis, University of Kaiserslautern, 1976.
13. P. Kaps, *Rosenbrock-type methods*, in Numerical methods for stiff initial value problems, G. Dahlquist and R. Jeltsch, eds., Bericht nr. 9. Inst. für Geometrie und Praktische Mathematik der RWTH Aachen, 1981.
14. W. M. Lioen and J. J. B. de Swart, *Test set for IVP solvers*, Release 2.0. Available at `http://www.cwi.nl/cwi/projects/IVPtestset/`, 1998.
15. G.-Y. Psihoyios and J. R. Cash, *A stability result for general linear methods with characteristic function having real poles only*, BIT, 38 (1998), pp. 612–617.