On Implicit Euler and Related Methods for High-Order High-Index DAEs*

J. Sand

Department of Computer Science, University of Copenhagen,
Universitetsparken 1, DK–2100 Copenhagen, Denmark.
E-mail: *datjs@diku.dk*

Abstract

The Implicit Euler method is seldom used to solve differential-algebraic equations (DAEs) of differential index $r \geq 3$, since the method in general fails to converge in the first $r-2$ steps after a change of stepsize. However, if the differential equation is of order $d = r - 1 \geq 1$, an alternative variable-step version of the Euler method can be shown uniformly convergent. For $d = r - 1$, this variable-step method is equivalent to the Implicit Euler except for the first $r - 2$ steps after a change of stepsize. Generalization to DAEs with differential equations of order $d > r - 1 \geq 1$, and to variable-order formulas is discussed.

Key words: Linear multistep method, Backward Differentiation Formula, differential-algebraic equation, differential index, initial value problem, divided difference.

1 Introduction

According to ([3], p.46), those systems of differential-algebraic equations (DAEs) which arise most commonly in applications are the index one systems, the semi-explicit index two systems and the index three systems in Hessenberg form. However, systems of arbitrarily high index may occur naturally in mathematical models ([3], p.150), and thus methods for such systems are of interest. In this paper we consider DAEs of order $d \geq r - 1 \geq 1$, where r is the (differential) index.

Several codes for DAEs (e.g. DASSL([8]), LSODI([7]) and SPRINT([2])) have been based on the Backward Differentiation Formulas (BDFs), of which the first order formula (Implicit Euler) plays a central role - at least in the beginning of the integration. However, it has been known for a long time (cf. e.g.

*Tech. Rep. 01/03, Dept. of Computer Sci., Univ. of Copenhagen, 2001.

This report is an extended version of a paper appearing in APNUM.

[5,6,4]) that Implicit Euler in general fails to converge in the first $r-2$ steps after a change of stepsize, where the initial point may be regarded as one of the positions, where the stepsize is changed (from 0 to a positive value). In [1] an algorithm for correcting the numerical values after stepchanges was derived for $r=3$. However, the algorithm assumes the DAE to depend linearly on the algebraic variables, and consecutive stepchanges seem to worsen the corrected values. In this paper we will derive an alternative variable-step version of the Implicit Euler method applicable to d'th order DAEs of index $r \in [2, d+1]$, and for $(d,r)=(2,3)$ we may compare the errors produced by this method to those of Implicit Euler with/without correction, listed in Table 8.3 of [1]:

Table 1
Comparison with results listed in Table 8.3 of [1].

Step no.	Stepsize $\times 10^3$	Value of alg. var.	Absolute error of approximation (Euler)	(Corrected)	(Alt. Euler)
1	1.000	−4.0080	**2.0080**	**0.0044**	0.0080
2	1.000	−4.0160	0.0080	0.0080	0.0120
3	0.200	−4.0176	**8.0303**	**0.0391**	0.0057
4	0.040	−4.0179	**8.0348**	**0.2121**	0.0012
5	0.008	−4.0180	**8.0357**	**1.0725**	0.0003
6	0.008	−4.0181	0.0001	0.0001	0.0001
7	0.016	−4.0182	**1.0047**	**0.0001**	0.0002
8	0.032	−4.0185	**1.0048**	**0.0002**	0.0004
9	0.064	−4.0190	**1.0052**	**0.0003**	0.0007
10	0.064	−4.0195	0.0006	0.0006	0.0008

Consider the initial value problem

$$y^{(d)} = f(t, y, y', .., y^{(d-1)}, \lambda), \quad y^{(j)}(t_0) = \eta_j, \quad j = 0, 1, .., d-1, \tag{1}$$

$$0 = g(t, y, y', .., y^{(d+1-r)}), \quad r \in [2, d+1], \tag{2}$$

Most often, high-order ordinary differential equations (ODEs) are solved by transforming the equation to a system of first-order ODEs, and then by applying some of the many methods for first-order ODEs, e.g. the Implicit Euler. It thus seems natural to consider the following 'Implicit Euler method' for producing the approximations $(y_{0,n}, \lambda_n)$ to the values $(y(t_n), \lambda(t_n))$, $n = 1, 2, 3, ..,$ of the DAE-solution:

$$\begin{aligned} (y_{j,n} - y_{j,n-1})/(t_n - t_{n-1}) &= y_{j+1,n}, \quad j = 0, 1, .., d-2, \\ (y_{d-1,n} - y_{d-1,n-1})/(t_n - t_{n-1}) &= f(t_n, y_{0,n}, y_{1,n}, .., y_{d-1,n}, \lambda_n), \\ 0 &= g(t_n, y_{0,n}, y_{1,n}, .., y_{d+1-r,n}), \end{aligned} \tag{3}$$

where $y_{j,0} = \eta_j$ for $j = 0, 1, .., d-1$.

However, methods for systems of first-order ODEs are designed to estimate each component of the solution with the *same* order of accuracy, and for low

order methods (such as Implicit Euler) the accuracy of the $y(t_n)$-estimate is in general too low for producing reasonable estimates of the *derivatives* of y and thus of λ.

Another approach is to exchange the 'equation order reduction' and the discretization. If we thus discretize the DAE (1),(2) by using divided differences, and then write the *discretized* equations as a system of equations, we obtain for the approximations $(y_{j,n}, \lambda_n) \approx (j!y[t_n, t_{n-1}, .., t_{n-j}], \lambda(t_n))$

$$\begin{array}{ll} (y_{j,n} - y_{j,n-1})/((t_n - t_{n-1-j})/(j+1)) = y_{j+1,n}, & j = 0, 1, .., d-2, \\ (y_{d-1,n} - y_{d-1,n-1})/((t_n - t_{n-d})/d) = f(t_n, y_{0,n}, y_{1,n}, .., y_{d-1,n}, \lambda_n), & \\ 0 = g(t_n, y_{0,n}, y_{1,n}, .., y_{d+1-r,n}), & \end{array} \tag{4}$$

where $y_{j,0} = \eta_j$ for $j = 0, 1, .., d-1$, and t_m is interpreted as t_0 for m negative.

We notice that (4) only differs from (3) in $d-1$ steps after a change of stepsize, and for $r = d+1 \geq 3$ this corresponds to the case, where (3) fails to converge. Hence, one might expect (4) to remedy this lack of convergence. However, as seen in Example 1 below, (4) must be modified for $r \in [2, d]$, since the accuracy of the $y(t_n)$-estimate may then be affected by the lower accuracy of the estimates of the derivatives via the algebraic condition.

Example 1 *Consider the following DAE of order $d = 3$ and index $r = 3$:*

$$\begin{aligned} y^{(3)}(t) &= \lambda(t), & y^{(0)}(0) = y^{(1)}(0) = y^{(2)}(0) &= 1, \\ 0 &= a\exp(t) + by^{(0)}(t) - (a+b)y^{(1)}(t), & |a| + |b| &> 0, \end{aligned}$$

for which the solution is $y(t) = \lambda(t) = \exp(t)$. Applying method (4), the approximations in the first grid point $t_1 = h$ will satisfy the equations

$$6(y_{0,1} - 1 - h - \frac{1}{2}h^2)/h^3 = \lambda_1$$

$$0 = a\exp(h) + by_{0,1} - (a+b)(y_{0,1} - 1)/h$$

Hence,

$$\lambda_1 = \begin{cases} 3!\exp[h, 0, 0, 0] \text{ if } & a + b = 0 \\ 3h^{-1} + \mathcal{O}(1) & \text{otherwise,} \end{cases}$$

and we have no (uniform) convergence for $a + b \neq 0$.

On the other hand, if $y^{(1)}(h)$ in the constraint *is approximated by using the third-order BDF formula*

$$y^{(1)}(t_n) \approx y[t_n, t_{n-1}] + (t_n - t_{n-1})\{y[t_n, t_{n-1}, t_{n-2}] + (t_n - t_{n-2})y[t_n, t_{n-1}, .., t_{n-3}]\},$$

$y_{0,n}$ will for $n \geq 3$ become a BDF3-solution of the ODE

$$(a+b)y'(t) = by(t) + a\exp(t),$$

and for constant *stepsize h, λ_n will for $n \geq 6$ become a BDF3-solution of*

$$(a+b)\lambda'(t) = b\lambda(t) + a(3!)\exp[t, t-h, t-2h, t-3h].$$

Hence, we obtain convergence for fixed *stepsize, provided the starting values $y_{j,0}$ are chosen $\mathcal{O}(h^{4-j})$-accurate, as this implies $\mathcal{O}(h)$-accuracy of λ_3, λ_4 and λ_5.*

For variable *stepsize, however, third order accuracy of $y_{0,n}$ does not necessarily imply first order accuracy of λ_n, and one might think of using the BDF4-formula in the constraint, assuming that an $\mathcal{O}(H)$-accurate estimate of the initial value $\lambda(0)$ is known, as well as $\mathcal{O}(H^{4-j})$-estimates of $y^{(j)}(0)$, where H is a finite upper bound of the stepsizes. We will, however, leave this possibility for further research.* □

Example 1 indicates that method (4) should be modified for $r \in [2, d]$ in the following way:

$$\begin{aligned} (y_{j,n} - y_{j,n-1})/((t_n - t_{n-1-j})/(j+1)) &= y_{j+1,n}, \qquad j = 0, 1, .., d-2, \\ (y_{d-1,n} - y_{d-1,n-1})/((t_n - t_{n-d})/d) &= f(t_n, y_{0,n}, y_{1,n}, .., y_{d-1,n}, \lambda_n), \\ 0 &= g(t_n, y_{0,n}, p_n'(t_n), .., p_n^{(d+1-r)}(t_n)), \end{aligned} \tag{5}$$

where $y_{j,0} = \eta_j$ for $j = 0, 1, .., d-1$, t_m is interpreted as t_0 for m negative, and p_n is an interpolation polynomial, which - in case the BDFd-formula is used for estimating $y'(t_n)$ - reads

$$\sum_{i=0}^{d-1} \prod_{j=0}^{i-1} \left(\frac{t - t_{n-j}}{j+1} \right) y_{i,n} + \prod_{j=0}^{d-1} \left(\frac{t - t_{n-j}}{j+1} \right) f(t_n, y_{0,n}, y_{1,n}, .., y_{d-1,n}, \lambda_n).$$

In Section 2 we list the assumptions on the DAE (1), (2), ensuring a unique local DAE-solution, and show that for fixed $n \geq 1$ (5) has a unique solution within a neighbourhood of the DAE-solution provided the previous numerical values $(y_{j,n-i}, \lambda_{n-i})$, $i \geq 1$, are sufficiently accurate, satisfying the algebraic condition to a certain accuracy, and the stepsizes are sufficiently small with bounded ratios. In Section 3 we restrict ourselves to the case $d = r - 1 \geq 1$ and show that for sufficiently accurate starting values and small stepsizes with bounded ratios, the numerical values will remain accurate, since the method (5) is then (uniformly) convergent. In Section 4 we outline how method (5) may be generalized to variable-step variable-order methods based on the BDFs. As

an example, a method based on the first- and second-order BDFs is derived and tested on a problem with $(d, r) = (2, 3)$. In Section 5 a much more simple variable-step variable-order method is seen to give results similar to those in Section 4.

2 Existence and Uniqueness of Solutions to (1),(2) and (5)

Let $g^{(i)},\ i = 0, 1, .., r-1$, be formally defined as

$$g^{(i)}(t, y(t), y'(t), .., y^{(d+1-r+i)}(t)) = \left(\frac{d}{dt}\right)^{(i)} g(t, y(t), y'(t), .., y^{(d+1-r)}(t)).$$

The assumptions on the DAE (1),(2) can then be written as follows.

ASSUMPTIONS.

(1) *f and $g^{(r-1)}$ are C^1-functions with bounded and Lipschitz-continuous partial derivatives on open sets Ω_1, Ω_2, containing $v_0 = (t_0, \eta_0, .., \eta_{d-1}, \lambda_0)$ and $(t_0, \eta_0, .., \eta_{d-1}, f(v_0))$, respectively, where λ_0 is the unique value of $\lambda(t_0)$ (cf. (3) below).*

(2) *The initial values $\eta_0, .., \eta_{d-1}$ are consistent with the equations*

$$0 = g^{(i)}(t_0, \eta_0, .., \eta_{d+1-r+i}), \quad i = 0, 1, .., r-2.$$

(3) *There exists a unique solution $\lambda(t_0) = \lambda_0$ to the equation*

$$0 = g^{(r-1)}(t_0, \eta_0, .., \eta_{d-1}, f(t_0, \eta_0, .., \eta_{d-1}, \lambda(t_0))),$$

or a solution $\lambda(t_0) = \lambda_0$ is given as initial value.

(4) *The matrix*

$$\frac{\partial}{\partial \lambda(t)} g^{(r-1)}(t, y_0^{[2]}(t), .., y_{d-1}^{[2]}(t), f(t, y_0^{[1]}(t), .., y_{d-1}^{[1]}(t), \lambda(t)))$$

is regular with bounded inverse for all $v(t) = (t, y_0^{[1]}(t), .., y_{d-1}^{[1]}(t), \lambda(t)) \in \Omega_1$, $(t, y_0^{[2]}(t), .., y_{d-1}^{[2]}(t), f(v(t))) \in \Omega_2$. □

Since the low derivatives $y^{(i)}(t),\ i = 0, 1, .., d-1$, may be formally expressed in terms of $y^{(d)}(t)$:

$$y^{(i)}(t) = \sum_{j=0}^{d-1-i} \frac{(t-t_0)^j}{j!} \eta_{i+j} \ + \ \int_{t_0}^{t} \frac{(t-s)^{d-1-i}}{(d-1-i)!} y^{(d)}(s) ds, \tag{6}$$

the assumptions above are easily seen to ensure a unique local solution to the DAE by considering the following iteration for $k = 0, 1, ...$

$$\begin{aligned} y_0^{(d)}(t) &\equiv 0, \\ y_{k+1}^{(d)}(t) = &\quad f(t, y_k(t), y_k'(t), .., y_k^{(d-1)}(t), \lambda_k(t)) \\ 0 = &\frac{d}{dt} g^{(r-1)}(t, y_{k+1}(t), .., y_{k+1}^{(d-1)}(t), y_{k+1}^{(d)}(t)), \quad \lambda_k(t_0) = \lambda_0. \end{aligned}$$

where $y_m^{(i)}(t)$ denotes (6) with $y_m^{(d)}$ substituted for $y^{(d)}$, $m = k, k+1$.

As concerns the solution of (5), we note that

$$\begin{aligned} p_n(t) &= p_{n-1}(t) + q_{n-1}(t) d! (p_n - p_{n-1})[t_n, .., t_{n-d}], \\ q_{n-1}(t) &= \prod_{j=0}^{d-1} \left(\frac{t - t_{n-1-j}}{j+1} \right). \end{aligned} \tag{7}$$

Since $q_{n-1}(t) = 0$ for $t = t_{n-1}, t_{n-2}, .., t_{n-d}$, we thus have a discrete analogue to (6):

$$\begin{aligned} i! p_n[t_n, .., t_{n-i}] = &\sum_{j=0}^{d-1-i} \prod_{k=i}^{i+j-1} \left(\frac{t_n - t_{n-1-k}}{k+1} \right) (i+j)! p_{n-1}[t_{n-1}, .., t_{n-1-i-j}] \\ &+ \prod_{k=i}^{d-1} \left(\frac{t_n - t_{n-1-k}}{k+1} \right) d! p_n[t_n, .., t_{n-d}], \end{aligned} \tag{8}$$

and we try to find a solution of (5) by simple functional iteration:

$$\begin{aligned} p_{n,0}(t) &\equiv p_{n-1}(t), \\ d! p_{n,k+1}[t_n, .., t_{n-d}] &= f(t_n, p_{n,k}[t_n], .., (d-1)! p_{n,k}[t_n, .., t_{n-d+1}], \lambda_{n,k}) \\ 0 &= g(t_n, p_{n,k+1}(t_n), .., p_{n,k+1}^{(d+1-r)}(t_n)) / q_{n-1}^{(d+1-r)}(t_n), \quad k = 0, 1, ... \end{aligned} \tag{9}$$

where $p_{n,k+1}^{(i)}(t)$ denotes the i'th derivative of (7) with $p_{n,k+1}[t_n, .., t_{n-d}]$ substituted for $p_n[t_n, .., t_{n-d}]$, and the i'th order divided difference is found from the d'th order through (8).

Lemma 2 *Assume that the unique solution of* (1),(2), *ensured by our* ASSUMPTIONS, *exists for* $t \in [t_0, t_{n-1} + H]$, *where* $H < \infty$ *is an upper bound of the stepsizes* $t_i - t_{i-1}$, $i \geq 1$, *and that the DAE-solution remains within* Ω_1, Ω_2.

If for $j = 0, 1, .., d-1$, $m = 1, 2, .., n-1$,

(i) $y_{j,0} = y^{(j)}(t_0) + \mathcal{O}(H)(t_1 - t_0)^{d-j}$,
(ii) $y_{j,m} = y^{(j)}(t_m) + \mathcal{O}(H), \quad \lambda_m = \lambda(t_m) + \mathcal{O}(H)$,
(iii) $g(t_m, p_m(t_m), .., p_m^{(d+1-r)}(t_m))/q_{m-1}^{(d+1-r)}(t_m) = \mathcal{O}(H)$,
(iv) $(t_{m+1} - t_m)/(t_m - t_{m-1}) \in [\gamma, \Gamma]$ *for* $0 < \gamma \leq \Gamma < \infty$.

then the iteration (9) *converges for sufficiently small* H *to the solution of* (5) *satisfying* $\lambda_n = \lambda_{n-1} + \mathcal{O}(H)$.

PROOF. First we prove that for sufficiently small H, a unique $\lambda_{n,0} = \lambda_{n-1} + \mathcal{O}(H)$ exists, and that $\|(p_{n,1} - p_{n-1})[t_n, .., t_{n-d}]\|$ is $\mathcal{O}(H)$. Then we show, by induction in $k \geq 1$, the existence of a unique $\lambda_{n,k}$ satisfying $\|\lambda_{n,k} - \lambda_{n,k-1}\| = \mathcal{O}(H)\|(p_{n,k} - p_{n,k-1})[t_n, .., t_{n-d}]\|$, and that $\|(p_{n,k+1} - p_{n,k})[t_n, .., t_{n-d}]\| = \mathcal{O}(H)\|(p_{n,k} - p_{n,k-1})[t_n, .., t_{n-d}]\|$. Hence, for sufficiently small H, the Cauchy sequence $(\lambda_{n,k}, p_{n,k+1}[t_n, .., t_{n-d}])_k$ will converge to a fixpoint $(\lambda_n, p_n[t_n, .., t_{n-d}])$ of (9), since f and g are continuous. That (9) has no other fixpoints with $\lambda_n = \lambda_{n-1} + \mathcal{O}(H)$ follows from the boundedness of $(\partial g^{(r-1)}/\partial\lambda)^{-1}$ and the partial derivatives of f, which is valid for sufficiently small H.

Let $k \geq 0$ and $p_{n,k}$ be given with $\|(p_{n,k} - p_{n-1})[t_n, .., t_{n-d}]\|$ being $\mathcal{O}(H)$. In order to find $\lambda_{n,k} = \lambda_{n-1} + \mathcal{O}(H)$, we use the iterative scheme

$$\lambda_{n,k}^{[j+1]} = \lambda_{n,k}^{[j]} - \left[\frac{\partial G_{n,k}}{\partial\lambda}(\lambda_{n,k}^{[0]})\right]^{-1} G_{n,k}(\lambda_{n,k}^{[j]}), \quad j = 0, 1, .., \quad \lambda_{n,k}^{[0]} = \lambda_{n-1}, \quad (10)$$

where

$$G_{n,k}(\lambda) = g(t_n, (p_{n-1}^{(i)}(t_n) + q_{n-1}^{(i)}(t_n)\Delta f_{n,k}(\lambda))_{i=0}^{d+1-r})/q_{n-1}^{(d+1-r)}(t_n),$$

and $\Delta f_{n,k}(\lambda)$ denotes

$$f(t_n, (s!p_{n,k}[t_n, .., t_{n-s}])_{s=0}^{d-1}, \lambda) - f(t_{n-1}, (s!p_{n-1}[t_{n-1}, .., t_{n-1-s}])_{s=0}^{d-1}, \lambda_{n-1}).$$

Since $p_{n,k}(t)$ is defined in (7) with $p_{n,k}[t_n, .., t_{n-d}]$ substituted for $p_n[t_n, .., t_{n-d}]$, we find, for $s = 0, 1, .., d-1$, that $p_{n,k}[t_n, .., t_{n-s}] - p_{n-1}[t_{n-1}, .., t_{n-1-s}]$ equals

$$(t_n - t_{n-1-s})p_{n-1}[t_n, .., t_{n-1-s}] + \prod_{i=s}^{d-1}(t_n - t_{n-1-i})\mathcal{O}(H) = \mathcal{O}(H). \quad (11)$$

Hence,

$$\frac{\partial G_{n,k}}{\partial\lambda}(\lambda_{n,k}^{[0]}) = \sum_{i=0}^{d+1-r} \frac{q_{n-1}^{(i)}(t_n)}{q_{n-1}^{(d+1-r)}(t_n)} M_{i,n-1}(t_n),$$

where

$$M_{i,n-1}(t_n) = \frac{\partial g}{\partial y^{(i)}}(t_n, (p_{n-1}^{(s)}(t_n) + \mathcal{O}(H))_{s=0}^{d+1-r}) \frac{\partial f}{\partial \lambda}(t_n, (y_{s,n-1} + \mathcal{O}(H))_{s=0}^{d-1}, \lambda_{n-1}),$$

and

$$\frac{q_{n-1}^{(i)}(t_n)}{q_{n-1}^{(d+1-r)}(t_n)} \leq \frac{(t_n - t_{n-1-i}) \cdots (t_n - t_{n-d}) d!/(d-i)!}{(t_n - t_{n-2-d+r}) \cdots (t_n - t_{n-d})(d+1-r)!} = \mathcal{O}(H) \quad i = 0, 1, .., d-r.$$

Due to (ii) in the lemma, and ASSUMPTION 4, we may thus assume that

$$\left\| \left[\frac{\partial G_{n,k}}{\partial \lambda}(\lambda_{n,k}^{[0]}) \right]^{-1} \right\| \leq M, \tag{12}$$

where M is a constant independent of k. Hence, if $G_{n,k}(\lambda_{n,k}^{[0]}) = \mathcal{O}(H)$ it will follow from the scheme (10) that $\lambda_{n,k}^{[1]} = \lambda_{n-1} + \mathcal{O}(H)$.

$$G_{n,k}(\lambda_{n,k}^{[0]}) = \tilde{g}_{n-1}(t_n)/q_{n-1}^{(d+1-r)}(t_n) + \mathcal{O}(H),$$

where

$$\tilde{g}_{n-1}(t) = g(t, p_{n-1}(t), .., p_{n-1}^{(d+1-r)}(t)),$$

and for $n \geq r+1$ we obtain from (ii), (iii) and (7) with $n = n-1$

$$\begin{aligned} \tilde{g}_{n-1}(t_{n-i}) &= g(t_{n-i}, p_{n-i}(t_{n-i}), .., p_{n-i}^{(d+1-r)}(t_{n-i})) + \mathcal{O}(H) \sum_{s=2}^{i} q_{n-s}^{(d+1-r)}(t_{n-i}) \\ &= \mathcal{O}(H) q_{n-r-1}^{(d+1-r)}(t_{n-1}), \quad i = 1, 2, .., r. \end{aligned}$$

Thus (iv) implies that the C^r-function $\tilde{g}_{n-1}(t)$ satisfies

$$\tilde{g}_{n-1}(t_n)/q_{n-1}^{(d+1-r)}(t_n) =$$
$$\left[\sum_{i=1}^{r} \prod_{s=1}^{i-1} (t_n - t_{n-s}) g_{n-1}[t_{n-1}, .., t_{n-i}] + \mathcal{O}(\prod_{s=1}^{r} (t_n - t_{n-s})) \right] / q_{n-1}^{(d+1-r)}(t_n) = \mathcal{O}(H).$$

Due to (i), the result above is also valid for $n \in [1, r]$, but we leave this as an exercise for the reader. Having proved that $\lambda_{n,k}^{[1]} = \lambda_{n-1} + \mathcal{O}(H)$, we may now conclude the existence of $\lambda_{n,k} = \lambda_{n-1} + \mathcal{O}(H)$ by showing that the iterative

scheme (10) is strongly contractive. The uniqueness of $\lambda_{n,k}$ for small H follows from (12).

Subtracting the equation in (10) from the one with $j = j-1$, we have, by induction in $j \geq 1$, that

$$\begin{aligned}\|\lambda_{n,k}^{[j+1]} - \lambda_{n,k}^{[j]}\| &\leq M \parallel G_{n,k}(\lambda_{n,k}^{[j]}) - G_{n,k}(\lambda_{n,k}^{[j-1]}) - \left[\frac{\partial G_{n,k}}{\partial \lambda}(\lambda_{n-1})\right](\lambda_{n,k}^{[j]} - \lambda_{n,k}^{[j-1]})\| \\ &\leq \mathcal{O}(HM)\|\lambda_{n,k}^{[j]} - \lambda_{n,k}^{[j-1]}\|,\end{aligned}$$

since $G_{n,k}$ is a C^1-function, and $\lambda_{n,k}^{[j-1]},\ \lambda_{n,k}^{[j]}$ stays within a certain neighbourhood of λ_{n-1}.

Returning to the outer iteration (9), we note that, for $k = 0$, the uniform Lipschitz continuity of the C^1-function f implies that $\|(p_{n,1} - p_{n-1})[t_n, .., t_{n-d}]\|$ is $\mathcal{O}(H)$. If H is sufficiently small, we may thus find a unique $\lambda_{n,1} = \lambda_{n-1} + \mathcal{O}(H)$, satisfying $G_{n,1}(\lambda) = 0$. Subtracting $G_{n,0}(\lambda_{n,0})$ from $G_{n,1}(\lambda_{n,1})$ we obtain

$$\begin{aligned}0 &= [g(t_n, (p_{n,2}^{(i)}(t_n))_{i=0}^{d+1-r}) - g(t_n, (p_{n,1}^{(i)}(t_n))_{i=0}^{d+1-r})]/q_{n-1}^{(d+1-r)}(t_n) \\ &= \left\{\sum_{i=0}^{d+1-r} \frac{q_{n-1}^{(i)}(t_n)}{q_{n-1}^{(d+1-r)}(t_n)} \int_0^1 \frac{\partial g}{\partial y^{(i)}}(t_n, ((\theta p_{n,2}^{(s)} + (1-\theta)p_{n,1}^{(s)})(t_n))_{s=0}^{d+1-r})d\theta\right\} \cdot \\ &\quad \{f(t_n, (s!p_{n,1}[t_n, .., t_{n-s}])_{s=0}^{d-1}, \lambda_{n,1}) - f(t_n, (s!p_{n,0}[t_n, .., t_{n-s}])_{s=0}^{d-1}, \lambda_{n,0})\} \\ &= \left\{\mathcal{O}(H) + \int_0^1 \frac{\partial g}{\partial y^{(d+1-r)}}(t_n, ((\theta p_{n,2}^{(s)} + (1-\theta)p_{n,1}^{(s)})(t_n))_{s=0}^{d+1-r})d\theta\right\} \cdot \\ &\quad \{\mathcal{O}(H)\|(p_{n,1} - p_{n,0})[t_n, .., t_{n-d}]\| + \\ &\quad \int_0^1 \frac{\partial f}{\partial \lambda}(t_n, (s!p_{n,1}[t_n, .., t_{n-s}])_{s=0}^{d-1}, \theta\lambda_{n,1} + (1-\theta)\lambda_{n,0})d\theta(\lambda_{n,1} - \lambda_{n,0})\Bigg\}.\end{aligned}$$

Using ASSUMPTION 4 and the fact that g is a C^1-function, we thus have

$$\|\lambda_{n,1} - \lambda_{n,0}\| = \mathcal{O}(H)\|(p_{n,1} - p_{n,0})[t_n, t_{n-1}, .., t_{n-d}]\|. \tag{13}$$

From (9) and (8) with subscript n replaced by $n,1$ and $n,0$, it thus follows from the Lipschitz-continuity of f that

$$\|(p_{n,2} - p_{n,1})[t_n, t_{n-1}, .., t_{n-d}]\| = \mathcal{O}(H)\|(p_{n,1} - p_{n,0})[t_n, t_{n-1}, .., t_{n-d}]\|. \tag{14}$$

Hence, we may find a unique $\lambda_{n,2} = \lambda_{n-1} + \mathcal{O}(H)$ satisfying $G_{n,2}(\lambda_{n,2}) = 0$, and since (13),(14) can be generalized to all consecutive iterates, the lemma follows by induction. □

3 Uniform Convergence of Method (5) in case r = d + 1

Since the purpose of this section is to prove condition (ii) of Lemma 2 for all $m \geq 1$ (provided the solution remains within Ω_1, Ω_2), we may as well use a formulation similar to Lemma 2.

Theorem 3 *Consider the case $r = d + 1$. Assume that the unique solution of* (1),(2)*, ensured by our* ASSUMPTIONS*, exists for $t \in [t_0, t_{N-1} + H]$, where $H < \infty$ is an upper bound of the stepsizes $t_i - t_{i-1}$, $i \geq 1$, and that the DAE-solution remains within Ω_1, Ω_2. If*

(i) $y_{j,0} = y^{(j)}(t_0) + \mathcal{O}(H)(t_1 - t_0)^{d-j}$ *for* $j = 0, 1, .., d-1$,
(ii) $(t_{n+1} - t_n)/(t_n - t_{n-1}) \in [\gamma, \Gamma]$ *for* $0 < \gamma \leq \Gamma < \infty$, $n = 1, 2, .., N-1$.

then, for sufficiently small H, (5) *has a unique solution satisfying $\lambda_n = \lambda(t_n) + \mathcal{O}(H)$ for all t_n, $n = 1, 2, .., N$, and*

$$\|y_{j,n} - j!y[t_n, t_{n-1}, .., t_{n-j}]\| =$$

$$\mathcal{O}(H)(H + t_n - t_0)^{d-j}[1 + K(H + t_n - t_0)\ \exp((K + \mathcal{O}(H))(t_n - t_0))],$$

for $j = 0, 1, .., d-1$. The constant $K = d + L_f(1 + ML_g(1 + L_f))$ depends on the bounds L_f, L_g of the partial derivatives of f and $g^{(d)}$ and on the bound M of $[\partial g^{(d)}/\partial\lambda(t)]^{-1}$ on Ω_1, Ω_2 (cf. the ASSUMPTIONS*).*

The error bounds of $y_{j,n}, j = 0, 1, .., d-1$, are also valid if the algebraic constraint is replaced by

$$g(t_n, y_{0,n}) = \mathcal{O}(H) \prod_{j=1}^{d} (t_n - t_{n-j}). \tag{15}$$

PROOF. The theorem is clearly valid for $n = 0$. Assume that it holds for $n \leq n - 1$. Then according to Lemma 2 a unique $\lambda_n = \lambda(t_n) + \mathcal{O}(H)$ exists. Defining the errors

$$e_{j,n} = j!y[t_n, t_{n-1}, .., t_{n-j}] - y_{j,n}, \quad j = 0, 1, .., d-1,$$

we obtain from (5) the inequalities

$$\|e_{j,n}\| \leq \|e_{j,n-1}\| + \left(\frac{t_n - t_{n-1-j}}{j+1}\right) \|e_{j+1,n}\|, \quad j = 0, 1, .., d-2, \tag{16}$$

$$\|e_{d-1,n}\| \leq \|e_{d-1,n-1}\| + \left(\frac{t_n - t_{n-d}}{d}\right) \|d!y[t_n, .., t_{n-d}] - f(t_n, (y_{i,n})_{i=0}^{d-1}, \lambda_n)\| \tag{17}$$

$$\leq \|e_{d-1,n-1}\| + \left(\frac{t_n - t_{n-d}}{d}\right)(\mathcal{O}(H) + L_f(\sum_{i=0}^{d-1}\|e_{i,n}\| + \|\lambda(t_n) - \lambda_n\|)).$$

Hence, since $(1-x)^{-1} = \exp(x + \mathcal{O}(x^2))$ for all small $x > 0$, we obtain by summation

$$\sum_{j=0}^{d-1}\|e_{j,n}\| \leq \tag{18}$$

$$\leq \sum_{j=0}^{d-1}\|e_{j,n-1}\| + \left(\frac{t_n - t_{n-d}}{d}\right)(\mathcal{O}(H) + (d + L_f)\sum_{j=0}^{d-1}\|e_{j,n}\| + L_f\|\lambda(t_n) - \lambda_n\|) \leq$$

$$\exp\left((d + L_f + \mathcal{O}(H))\left(\frac{t_n - t_{n-d}}{d}\right)\right)$$

$$\left(\sum_{j=0}^{d-1}\|e_{j,n-1}\| + \left(\frac{t_n - t_{n-d}}{d}\right)(\mathcal{O}(H) + L_f\|\lambda(t_n) - \lambda_n\|)\right).$$

For sufficiently small H we may thus assume that the bounds L_f, L_g and M are applicable on the line from the DAE-solution to the numerical solution at t_n. We shall make use of this and prove that

$$\|\lambda(t_n) - \lambda_n\| \leq \mathcal{O}(H) + ML_g(1 + L_f + \mathcal{O}(H))\sum_{j=0}^{d-1}\|e_{j,n}\|. \tag{19}$$

It will then follow from the first inequality of (18) that

$$\begin{aligned}\sum_{j=0}^{d-1}\|e_{j,n}\| &\leq \sum_{j=0}^{d-1}\|e_{j,n-1}\| + \left(\frac{t_n - t_{n-d}}{d}\right)(\mathcal{O}(H) + (K + \mathcal{O}(H))\sum_{j=0}^{d-1}\|e_{j,n}\|) \\ &\leq \exp\left((K + \mathcal{O}(H))\left(\frac{t_n - t_{n-d}}{d}\right)\right)(\sum_{j=0}^{d-1}\|e_{j,n-1}\| + \left(\frac{t_n - t_{n-d}}{d}\right)\mathcal{O}(H)) \\ &\leq \exp((K + \mathcal{O}(H))(t_n - t_0))(H + t_n - t_0)\mathcal{O}(H).\end{aligned}$$

Inserting this bound in (19) and (17) we obtain

$$\|e_{d-1,n}\| \leq$$

$$\|e_{d-1,n-1}\| + \left(\frac{t_n - t_{n-d}}{d}\right)\mathcal{O}(H)[1 + K(H + t_n - t_0)\exp((K + \mathcal{O}(H))(t_n - t_0))] \leq$$

$$\|e_{d-1,0}\| + \sum_{i=1}^{n}\left(\frac{t_i - t_{i-d}}{d}\right)\mathcal{O}(H)[1 + K(H + t_n - t_0)\exp((K + \mathcal{O}(H))(t_n - t_0))] \leq$$

$$\leq \mathcal{O}(H)(H + t_n - t_0)[1 + K(H + t_n - t_0)\exp((K + \mathcal{O}(H))(t_n - t_0))].$$

For $j = d-2, d-3, .., 0$, we obtain the error bound of $y_{j.n}$ by a similar substitution into (16) of the error bound of $y_{j+1,n}$, and the theorem will thus follow from (19).

In order to prove (19) we consider the function

$$\tilde{g}_n(t) = g(t, p_n(t)),$$

where p_n is the polynomial defined in connection with (5). From (7) we know that $p_n(t_{n-j}) = p_{n-j}(t_{n-j})$ for $j = 0, 1, .., d$. Since the ratio between consecutive stepsizes are bounded, it thus follows from (15) (and (i) in case $n \leq d$) that

$$\tilde{g}_n[t_n, t_{n-1}, .., t_{n-d}] = \mathcal{O}(H).$$

Since $\tilde{g}_n$ is a C^{d+1}-function there exists a $t_n^* \in [t_{n-d}, t_n]$ such that

$$g^{(d)}(t_n^*, p_n(t_n^*), .., p_n^{(d)}(t_n^*)) = \mathcal{O}(H).$$

Let $r_{i,n}$ denote the polynomials

$$r_{i,n} = \prod_{j=0}^{i-1}(t - t_{n-j})/i!, \qquad i = 0, 1, .., d.$$

We may then write

$$\begin{aligned} p_n^{(s)}(t_n^*) &= y^{(s)}(t_n^*) + \mathcal{O}(H) - \sum_{i=s}^{d-1} r_{i,n}^{(s)}(t_n^*)e_{i,n} - \\ &\quad - r_{d,n}^{(s)}(t_n^*)[f(t_n, (j!y[t_n, .., t_{n-j}])_{j=0}^{d-1}, \lambda(t_n)) - f(t_n, (y_{j,n})_{j=0}^{d-1}, \lambda_n)] \\ &= y^{(s)}(t_n^*) + \mathcal{O}(H) - e_{s,n} + \mathcal{O}(H(\sum_{i=0}^{d-1}\|e_{i,n}\| + \|\lambda(t_n) - \lambda_n\|)) \end{aligned}$$

for $s = 0, 1, .., d-1$, whereas $p_n^{(d)}(t_n^*)$ is

$$y^{(d)}(t_n^*) + \mathcal{O}(H) - [f(t_n, (j!y[t_n, .., t_{n-j}])_{j=0}^{d-1}, \lambda(t_n)) - f(t_n, (y_{j,n})_{j=0}^{d-1}, \lambda_n)].$$

Using the boundedness of $[\partial g^{(d)}/\partial\lambda(t)]^{-1}$ and the partial derivatives of f and $g^{(d)}$ on the line from the DAE-solution to the numerical solution at t_n, (19) and thus the theorem follows by induction in n. □

4 Generalization to Variable-Step Variable-Order BDFs

It is outside the scope of this paper to extend the convergence result of Section 3 to a variable-step variable-order method. However, in this section we will present such a method (based on the BDFs) and show that this method at least improves the results of the first- and second-order BDFs (with/without corrections) listed in [1], and that the second-order formula seems to have global error $\mathcal{O}(H^2)$ even for variable stepsizes.

When applying the BDF method of order k (i.e. BDFk) for estimation of $y^{(j+1)}(t_n)$ (y being the solution of (1),(2)), one would like the result to be exact, if y is a polynomial of degree at most $k+j$. For $j=0$ this is achieved by using the ordinary BDFk:

$$\sum_{i=1}^{k}\prod_{m=1}^{i-1}(t_n - t_{n-m})y_0[t_n, t_{n-1}, .., t_{n-i}] = y_{1,n}.$$

However, for polynomials y of degree larger than k, the differentiation formula is **not** exact, and the BDFk has to be modified in order to produce exact values of $y^{(j+1)}(t_n)$ for polynomials y of degree $k+j$, $j = 1, 2, .., d-1$:

$$\sum_{i=1}^{k}\alpha_{i,n}^{[j+1]}y_j[t_n, t_{n-1}, .., t_{n-i}] = y_{j+1,n}, \tag{20}$$

where the coefficients $\alpha_{i,n}^{[j+1]}$ are to satisfy

$$\sum_{i=1}^{k}\alpha_{i,n}^{[j+1]}y_j[t_n, t_{n-1}, .., t_{n-i}] = \frac{(s+j)!}{(s-1)!}t_n^{s-1} \text{ for } y_0(t) = t^{s+j}, \quad s = 1, 2, .., k. \tag{21}$$

In particular, **if** $y_{j,n}, y_{j,n-1}, .., y_{j,n-k}$ all have been produced by such a modified BDFk **or** by formulas of at least this order of accuracy, the coefficients $\alpha_{i,n}^{[j+1]}$ may easily be obtained from the ordinary BDFk-coefficients:

$$\alpha_{i,n}^{[j+1]} = \prod_{m=1}^{i-1}(t_n - t_{n-m}), \quad i = 1, 2, ..k-1, \tag{22}$$

and <u>for $y_0(t) = t^{k+j}$</u>:

$$\alpha_{k,n}^{[j+1]} = \left\{\frac{(k+j)!}{(k-1)!}t_n^{k-1} - \sum_{i=1}^{k-1}\prod_{m=1}^{i-1}(t_n - t_{n-m})y_j[t_n, .., t_{n-i}]\right\} / y_j[t_n, .., t_{n-k}]. \tag{23}$$

Otherwise, the solution of (21) may require somewhat more work, and it may not even exist for all combinations of stepsizes. However, since the formulas are to be examined for $y_0(t) = t^{s+j},\ s = 1, 2, .., k,\ j = 1, .., d-1$, this work may be of value in the derivation of the $(k+d-1)$'st degree polynomial p_n needed in the algebraic constraint if $r \in [2, d]$ (cf. (5)). p_n should satisfy

$$p_n(t_{n-i}) = y_{0,n-i}, \quad i = 0, 1, .., k-1, \ \text{ and}$$

$$\sum_{i=1}^{k} \alpha_{i,n}^{[j+1]} y_j[t_n, t_{n-1}, .., t_{n-1}] = y_{j+1,n} \ \text{ for } \ y_0(t) = p_n(t), \quad j = 0, 1, .., d-1.$$

Example 4 *We would like to solve the problem introduced in* [1] *by means of the variable-step variable-order formula* (20)*, with* k *equal to* 1 *and* 2*. The problem, which was solved by merely first-order formulas in Table* 1 *is of index* $r = 3$*, and it describes the position of a particle on a circular track. It reads*

$$\begin{aligned} \begin{pmatrix} x'' \\ y'' \end{pmatrix} &= 2 \begin{pmatrix} y \\ -x \end{pmatrix} + \lambda \begin{pmatrix} x \\ y \end{pmatrix}, \qquad \begin{pmatrix} x & x' \\ y & y' \end{pmatrix}(1) = \begin{pmatrix} \sin(1) & 2\cos(1) \\ \cos(1) & -2\sin(1) \end{pmatrix}, \\ 0 &= x^2 + y^2 - 1, \end{aligned}$$

and the solution is $(x(t), y(t), \lambda(t)) = (\sin(t^2), \cos(t^2), -4t^2)$*. The* ASSUMPTIONS *in Section* 2 *are satisfied, and the parasitic numerical solution that we find in each step is easily removed, since the point* $y_{0,n}$ *is almost opposite to the* $\mathcal{O}(H)$*-solution on the unit circle.*

In the following we will use the notation BDF(2,1)*, BDF*(1,1) *for BDF*1*-formulas modified for estimation of* $y^{(2)}(t_n)$*, whereas BDF*(3,1,2)*, BDF*(1,2,2)*, BDF*(2,2,2)*, BDF*(1,1,2)*, BDF*(2,1,2) *are used for modified BDF2-formulas. The interpretation is the following:*

BDF (2,1)***:*** *This BDF*1*-formula expects* $y_{1,n-1}$ *to be exact for* $y_0(t) = t^2$*, and it is used for changing from BDF*2 *to BDF*1*, as well as for taking the very first step.*

BDF (1,1)***:*** *Used for proceeding a BDF*1*-solution.*

BDF (3,1,2)***:*** *This BDF*2*-formula expects* $y_{1,n-2}$ *to be exact for* $y_0(t) = t^3$*, and* $y_{1,n-1}$ *to be produced by BDF*(2,1)*. The formula may be used in the second step only.*

BDF $(\mathbf{k_{n-2}, k_{n-1}, 2})$***,*** $\mathbf{k_{n-2}, k_{n-1} \in \{1, 2\}}$***:*** *The BDF*2*-formula to be used when* $y_{1,n-2}$ *and* $y_{1,n-1}$ *were produced by our modified BDF*k_{n-2}*- and BDF*k_{n-1}*-formulas, respectively.*

We note that for BDF$(k_{n-1}$,1*),* $y_0(t) = t^2$ *will imply* $y_{1,n-1} = t_{n-1} + t_{n-2}$ *if* $k_{n-1} = 1$*, and otherwise* $2t_{n-1}$*. Since* $y_{1,n} = t_n + t_{n-1}$*, we obtain the following*

BDF(1,1):
For $y_0(t) = t^2$ *we have* $y_1[t_n, t_{n-1}] = (t_n - t_{n-2})/(t_n - t_{n-1})$, *and it follows from* (23) *with* $k = 1, j = 1$ *that* $y^{(2)}(t_n)$ *is to be estimated by*

$$\begin{aligned} f(t_n, y_{0,n}, y_{1,n}, \lambda_n) &= y_{2,n} = 2(t_n - t_{n-1})/(t_n - t_{n-2}) y_1[t_n, t_{n-1}] \\ &= \frac{y_{1,n} - y_{1,n-1}}{(t_n - t_{n-2})/2}. \end{aligned}$$

BDF(2,1):
For $y_0(t) = t^2$ *we have* $y_1[t_n, t_{n-1}] = 1$, *and we obtain from* (23) *with* $k = 1, j = 1$ *that* $y^{(2)}(t_n)$ *is to be estimated by*

$$\begin{aligned} f(t_n, y_{0,n}, y_{1,n}, \lambda_n) &= y_{2,n} = 2 y_1[t_n, t_{n-1}] \\ &= \frac{y_{1,n} - y_{1,n-1}}{(t_n - t_{n-1})/2}. \end{aligned}$$

Similarly, we note that for $BDF(k_{n-2}, k_{n-1}, 2)$, $y_0(t) = t^3$ *will imply* $y_{1,n-2} = 3t_{n-2}^2$ *if* $k_{n-2} = 3$, $y_{1,n-2} = 2t_{n-2}^2 + t_{n-2}(t_{n-3} + t_{n-4}) - t_{n-3}t_{n-4}$ *if* $k_{n-2} = 2$, *and otherwise* $t_{n-2}^2 + t_{n-2}t_{n-3} + t_{n-3}^2$. *Hence,*

BDF(2,2,2):
For $y_0(t) = t^3$ *we obtain*

$$y_1[t_n, t_{n-1}] = [2(t_n + t_{n-1} + t_{n-2})(t_n - t_{n-1}) + (t_n - t_{n-3})(t_{n-1} - t_{n-2})]/(t_n - t_{n-1}),$$

and $y_1[t_n, t_{n-1}, t_{n-2}] =$

$$\frac{(t_n - t_{n-3})(2t_n - t_{n-1} - t_{n-2})/(t_n - t_{n-1}) - (t_{n-1} - t_{n-4})(t_{n-2} - t_{n-3})/(t_{n-1} - t_{n-2})}{(t_n - t_{n-2})}.$$

From (22),(23) *with* $k = 2, j = 1$ *it thus follows that* $y^{(2)}(t_n)$ *is to be estimated by*

$$f(t_n, y_{0,n}, y_{1,n}, \lambda_n) = y_1[t_n, t_{n-1}] + \frac{\alpha_{2,n}^{[2]}}{(t_n - t_{n-2})}(y_1[t_n, t_{n-1}] - y_1[t_{n-1}, t_{n-2}]),$$

where $\alpha_{2,n}^{[2]}/(t_n - t_{n-2})$ *is*

$$\frac{(t_{n-1} - t_{n-2})[2(3t_n - t_{n-1} - t_{n-2} - t_{n-3})(t_n - t_{n-1}) - (2t_n - t_{n-1} - t_{n-2})(t_n - t_{n-3})]}{(2t_n - t_{n-1} - t_{n-2})(t_n - t_{n-3})(t_{n-1} - t_{n-2}) - (t_n - t_{n-1})(t_{n-1} - t_{n-4})(t_{n-2} - t_{n-3})}.$$

If there has been no stepchanges since t_{n-4}, we thus obtain the usual BDF2. However, for certain unfortunate combinations of stepsizes this formula does not exist!

BDF(1,1,2):
From (21) with $k=2, j=1$ we obtain two equations for $\alpha_{1,n}^{[2]}$ and $\alpha_{2,n}^{[2]}$, which can be simplified by definition of $\beta_2 = \alpha_{2,n}^{[2]}/((t_n - t_{n-2})(t_{n-1} - t_{n-2}))$ and $\beta_1 = (\alpha_{1,n}^{[2]} + (t_{n-1} - t_{n-2})\beta_2)/(t_n - t_{n-1})$. We find that

$$\begin{bmatrix} 2t_n - t_{n-1} - t_{n-2} & -(t_{n-1} - t_{n-3}) \\ (2t_n - t_{n-1} - t_{n-2})(t_n + t_{n-1} + t_{n-2}) & -(t_{n-1} - t_{n-3})(t_n + t_{n-1} + t_{n-2}) \end{bmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 6t_n \end{pmatrix}.$$

Hence, $y^{(2)}(t_n)$ is to be estimated by

$$f(t_n, y_{0,n}, y_{1,n}, \lambda_n) = \beta_1(y_{1,n} - y_{1,n-1}) - \beta_2(y_{1,n-1} - y_{1,n-2}),$$

where

$$\beta_2 = \frac{2(2t_n - t_{n-1} - t_{n-2})}{(t_{n-1} - t_{n-3})(t_n - t_{n-3})}, \quad \beta_1 = \frac{2(3t_n - t_{n-1} - t_{n-2} - t_{n-3})}{(2t_n - t_{n-1} - t_{n-2})(t_n - t_{n-3})}.$$

BDF(1,2,2):
As in the case of BDF(1,1,2), we solve the equations (21) and obtain the formula

$$f(t_n, y_{0,n}, y_{1,n}, \lambda_n) = \beta_1(y_{1,n} - y_{1,n-1}) - \beta_3(y_{1,n-1} - y_{1,n-2}),$$

where β_1 is the same as for BDF(1,1,2), and

$$\beta_3 = \frac{2(\beta_1(t_n - t_{n-1}) - 1)}{2t_{n-1} - t_{n-2} - t_{n-3}}.$$

BDF(3,1,2):
From (21) with $k=2, j=1$ we obtain two equations whose solution gives the formula

$$f(t_n, y_{0,n}, y_{1,n}, \lambda_n) = \frac{t_n - t_{n-2}}{-\alpha_{2,n}^{[2]}/4} y_1[t_n, t_{n-1}] + \alpha_{2,n}^{[2]} y_1[t_n, t_{n-1}, t_{n-2}],$$

where $\alpha_{2,n}^{[2]} = 2(2t_n - t_{n-1} - t_{n-2})$.

BDF(2,1,2):
Since BDF(3,1,2) is used as starting method, in case second order is needed

in the second step, the formula BDF(2,1,2) *will probably be used very little. For the sake of completeness, we will, however, find it from the equations* (21) *and discover that it does not even exist for constant stepsize!*

$$f(t_n, y_{0,n}, y_{1,n}, \lambda_n) = \beta_4(y_{1,n} - y_{1,n-1}) - \beta_5(y_{1,n-1} - y_{1,n-2}),$$

where

$$\beta_5 = \frac{2(2t_n - t_{n-1} - t_{n-2})}{(t_n - t_{n-3})(t_{n-1} - t_{n-2}) - (t_{n-1} - t_{n-4})(t_{n-2} - t_{n-3})},$$
$$\beta_4 = \frac{\beta_5(t_{n-1} - t_{n-2}) + 2}{2t_n - t_{n-1} - t_{n-2}}.$$

Since we solve a problem with $r = d + 1$, *we need not insert the derivatives of a polynomial* $p_n(t)$ *in the algebraic constraint. However, in case the reader would like to test the methods on a problem with* $(d, r) = (2, 2)$, *we note that*

$$p_n(t) = y_{0,n} + (t - t_n)[y_{1,n} + \frac{1}{2}(t - t_{n-1})f(t_n, y_{0,n}, y_{1,n}, \lambda_n)]$$

for BDF(1,1), *BDF*(2,1) *and*

$$p_n(t) = y_{0,n} + (t - t_n)\left\{y_0[t_n, t_{n-1}] + (t - t_{n-1})\left\{p_n[t_n, .., t_{n-2}] + \frac{1}{2}(t - t_{n-2})\cdot \right.\right.$$
$$\left.\left.\frac{f(t_n, y_{0,n}, y_{1,n}, \lambda_n) - 2p_n[t_n, .., t_{n-2}]}{2t_n - t_{n-1} - t_{n-2}}\right\}\right\}, \quad p_n[t_n, .., t_{n-2}] = \frac{y_{1,n} - y_0[t_n, t_{n-1}]}{t_n - t_{n-1}},$$

for BDF$(i, j,2)$, $i, j = 1, 2$.

We compare our results to those listed in Table 8.2 *of* [1]*:*

Table 2
Comparison with results in Table 8.2 of [1]. The results are errors in the estimated algebraic variable λ, and second-order formulas are used except for the first step.

	Absolute errors for stepsize 0.005			Absolute errors for stepsize 0.01		
t_n	(BDF1&2)	(Corrected)	(20), $k \leq 2$	(BDF1&2)	(Corrected)	(20), $k \leq 2$
1.005	**2.0400**	**0.0403**	0.0402			
1.010	**4.0190**	**0.0190**	0.0010	**2.0810**	**0.0812**	0.0809
1.015	**1.0120**	**0.0119**	0.0010			
1.020	0.0012	0.0012	0.0009	**4.0350**	**0.0360**	0.0041
1.030	0.0013	0.0013	0.0009	**1.0280**	**0.0280**	0.0041
1.040	0.0013	0.0013	0.0009	0.0052	0.0052	0.0038
1.050	0.0014	0.0014	0.0010	0.0054	0.0054	0.0038

In order to estimate the global errors of the second-order formulas for variable *stepsizes, we first took* 10 *steps of the sizes shown in Table* 1. *Then we used the pseudo-random number generator rand() of of* MAPLE *to generate stepsizes approx.* 10 *times smaller than those of Table* 1, *dividing each stepsize by rand*(5..15)*-integers until a t-value in Table* 1 *was reached, and thus a new stepsize was to be divided by rand()-integers.*

As seen below, the second-order formula BDF(2,2,2) *(started by BDF*(2,1), *BDF*(3,1,2) *and BDF*(1,2,2)*) seems to have global error* $\mathcal{O}(H^2)$.

Table 3
Results for variable-step second-order formulas, started by a first-order step.

Step no.	h_n $\times 10^3$	Absolute error of λ_n (BDF1&2)	(20),$k \le 2$	Step no.	Absolute error of λ_n (20),$k \le 2$, for h_n/rand()
1	1.000	2.0080	$8.0 \cdot 10^{-3}$	12	$2.5 \cdot 10^{-7}$
2	1.000	4.0040	$4.0 \cdot 10^{-5}$	22	$2.8 \cdot 10^{-7}$
3	0.200	0.3202	$3.1 \cdot 10^{-5}$	31	$3.0 \cdot 10^{-7}$
4	0.040	0.0083	$1.6 \cdot 10^{-5}$	38	$3.0 \cdot 10^{-7}$
5	0.008	0.0008	$2.5 \cdot 10^{-5}$	43	$3.0 \cdot 10^{-7}$
6	0.008	0.0009	$2.7 \cdot 10^{-5}$	52	$3.0 \cdot 10^{-7}$
7	0.016	0.0001	$2.7 \cdot 10^{-5}$	62	$3.0 \cdot 10^{-7}$
8	0.032	0.0001	$2.7 \cdot 10^{-5}$	71	$3.0 \cdot 10^{-7}$
9	0.064	0.0002	$2.7 \cdot 10^{-5}$	83	$3.0 \cdot 10^{-7}$
10	0.064	0.0001	$2.7 \cdot 10^{-5}$	93	$3.0 \cdot 10^{-7}$
Step no.	h_n $\times 10^3$	$\|y_{1,n} - y'(t_n)\|_2$ (BDF1&2)	(20),$k \le 2$	Step no.	$\|y_{1,n} - y'(t_n)\|_2$ (20),$k \le 2$, for h_n/rand()
1	1.000	$2.0 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	12	$7.8 \cdot 10^{-8}$
2	1.000	$1.1 \cdot 10^{-5}$	$6.3 \cdot 10^{-6}$	22	$6.5 \cdot 10^{-8}$
3	0.200	$1.1 \cdot 10^{-5}$	$6.5 \cdot 10^{-6}$	31	$7.5 \cdot 10^{-8}$
4	0.040	$1.1 \cdot 10^{-5}$	$6.6 \cdot 10^{-6}$	38	$7.6 \cdot 10^{-8}$
5, .., 8	$\cdots$	$1.1 \cdot 10^{-5}$	$6.7 \cdot 10^{-6}$	43, .., 71	$7.6 \cdot 10^{-8}$
9	0.064	$1.1 \cdot 10^{-5}$	$6.6 \cdot 10^{-6}$	83	$7.6 \cdot 10^{-8}$
10	0.064	$1.1 \cdot 10^{-5}$	$6.6 \cdot 10^{-6}$	93	$7.6 \cdot 10^{-8}$
Step no.	h_n $\times 10^3$	$\|y_{0,n} - y(t_n)\|_2$ (BDF1&2)	(20),$k \le 2$	Step no.	$\|y_{0,n} - y(t_n)\|_2$ (20),$k \le 2$, for h_n/rand()
1	1.000	$1.0 \cdot 10^{-6}$	$2.7 \cdot 10^{-9}$	12	$7.6 \cdot 10^{-11}$
2	1.000	$1.3 \cdot 10^{-6}$	$8.6 \cdot 10^{-9}$	22	$1.4 \cdot 10^{-10}$
3	0.200	$1.3 \cdot 10^{-6}$	$9.9 \cdot 10^{-9}$	31	$1.6 \cdot 10^{-10}$
4, .., 6	$\cdots$	$1.3 \cdot 10^{-6}$	$1.0 \cdot 10^{-8}$	38, .., 52	$1.6 \cdot 10^{-10}$
7	0.016	$1.3 \cdot 10^{-6}$	$1.0 \cdot 10^{-8}$	62	$1.6 \cdot 10^{-10}$
8	0.032	$1.3 \cdot 10^{-6}$	$1.1 \cdot 10^{-8}$	71	$1.6 \cdot 10^{-10}$
9	0.064	$1.3 \cdot 10^{-6}$	$1.1 \cdot 10^{-8}$	83	$1.7 \cdot 10^{-10}$
10	0.064	$1.3 \cdot 10^{-6}$	$1.1 \cdot 10^{-8}$	93	$1.7 \cdot 10^{-10}$

In order to check the robustness *of the variable-order variable-stepsize method, the computations above were repeated with somewhat arbitrary order changes at the 10 basic t-values. We note that the global error now seems to be $\mathcal{O}(H)$, since the first-order formula is used rather often (in approximately 40% of the steps).*

Table 4
Results for variable-step variable-order methods.

Order	Step no.	h_n $\times 10^3$	Absolute error of λ_n (BDF1&2)	(20),$k \leq 2$	Step no.	Absolute error of λ_n (20),$k \leq 2$, h_n/rand()
1	1	1.000	2.0080	0.008009	12	0.001592
2	2	1.000	4.0040	0.000040	22	0.000004
1	3	0.200	1.9963	0.038403	31	0.000233
1	4	0.040	8.0348	0.001142	38	0.000061
2	5	0.008	10.3796	0.000020	43	0.000004
1	6	0.008	2.0091	0.000108	52	0.000014
2	7	0.016	3.0134	0.000020	62	0.000004
2	8	0.032	0.6698	0.000020	71	0.000004
2	9	0.064	0.0002	0.000020	83	0.000004
2	10	0.064	0.0001	0.000020	93	0.000004

Order	Step no.	h_n $\times 10^3$	$\|y_{1,n} - y'(t_n)\|_2$ (BDF1&2)	(20),$k \leq 2$	Step no.	$\|y_{1,n} - y'(t_n)\|_2$ (20),$k \leq 2$, h_n/rand()
1	1	1.000	2.0×10^{-3}	2.2×10^{-3}	12	3.2×10^{-4}
2	2	1.000	1.1×10^{-5}	6.3×10^{-6}	22	8.7×10^{-7}
1	3	0.200	4.0×10^{-4}	4.5×10^{-4}	31	3.2×10^{-5}
1	4	0.040	7.8×10^{-5}	9.2×10^{-5}	38	1.3×10^{-5}
2	5	0.008	1.1×10^{-5}	5.0×10^{-6}	43	8.9×10^{-7}
1	6	0.008	1.7×10^{-5}	2.1×10^{-5}	52	2.8×10^{-6}
2	7	0.016	1.1×10^{-5}	5.0×10^{-6}	62	9.0×10^{-7}
2	8	0.032	1.1×10^{-5}	5.0×10^{-6}	71	9.0×10^{-7}
2	9	0.064	1.1×10^{-5}	5.0×10^{-6}	83	8.9×10^{-7}
2	10	0.064	1.1×10^{-5}	5.0×10^{-6}	93	8.9×10^{-7}

Order	Step no.	h_n $\times 10^3$	$\|y_{0,n} - y(t_n)\|_2$ (BDF1&2)	(20),$k \leq 2$	Step no.	$\|y_{0,n} - y(t_n)\|_2$ (20),$k \leq 2$, h_n/rand()
1	1	1.000	1.0×10^{-6}	2.7×10^{-9}	12	4.6×10^{-10}
2	2	1.000	1.3×10^{-6}	8.6×10^{-9}	22	1.3×10^{-9}
1	3	0.200	1.4×10^{-6}	9.5×10^{-9}	31	1.4×10^{-9}
1	4	0.040	1.4×10^{-6}	9.7×10^{-9}	38	1.4×10^{-9}
2	5	0.008	1.4×10^{-6}	9.8×10^{-9}	43	1.4×10^{-9}
1	6	0.008	1.4×10^{-6}	9.8×10^{-9}	52	1.5×10^{-9}
2	7	0.016	1.4×10^{-6}	9.9×10^{-9}	62	1.5×10^{-9}
2	8	0.032	1.4×10^{-6}	1.0×10^{-8}	71	1.5×10^{-9}
2	9	0.064	1.4×10^{-6}	1.0×10^{-8}	83	1.6×10^{-9}
2	10	0.064	1.4×10^{-6}	1.1×10^{-8}	93	1.6×10^{-9}

5 Generalization to Another Variable-Step Variable-Order Method

Due to the popularity of the BDFs, readers may find the previous section interesting, although formulas based on the BDFs may not be the best choice for *high-order* DAEs. If we follow the approach leading to (5) (postponing the 'equation order reduction'), we find the following variable-step variable-order method for (1),(2) in case $r = d + 1$:

$$\begin{aligned} p^{(d)}_{n,d-1+k_n}(t_n) &= f\,(t_n, y_{0,n}, p'_{n,k_n}(t_n), p''_{n,1+k_n}(t_n), .., p^{(d-1)}_{n,d-2+k_n}(t_n), \lambda_n), \\ 0 &= g(t_n, y_{0,n}), \\ p_{n,s}(t) &= \sum_{i=0}^{s} \prod_{j=0}^{i-1} (t - t_{n-j}) y_0[t_n, t_{n-1}, .., t_{n-i}], \quad s = k_n, .., d - 1 + k_n. \end{aligned} \tag{24}$$

Method (24) is much easier implemented for $d = 2, k_n \leq 2$, than the family of BDF-formulas derived in Example 4, and it turns out that the results in Tables 3 and 4 are *very* similar to the results produced by (24) applied to the same test problem. Hence, we will not list the results corresponding to Tables 3,4, but instead show the similarity of the *local* errors of the second-order formulas (24) and (20) ($k_n \equiv 2$), with *exact* initial step. These results indicate that Theorem 3 may be generalized to cover some of the higher-order formulas (24), and we hope to show this in the near future. As regards zero-stability for fixed stepsize, we may note that the formulas with $k_n \leq 5$ are zero-stable for *all* $d = r - 1 \geq 1$, whereas the formula with $k_n = 6$ is 'only' stable for $4 \geq d = r - 1 \geq 1$.

Table 5
Comparison of local errors of the second-order formulas (20) and (24).

Step	h_n	Absolute error of λ_n		Abs. error of λ_n for $h_n := 10 \times h_n$	
no.	$\times 10^3$	(20),$k_n \equiv 2$	(24), $k_n \equiv 2$	(20),$k_n \equiv 2$	(24),$k_n \equiv 2$
1	1.000	0	0	0	0
2	1.000	2.7×10^{-6}	2.7×10^{-6}	2.6×10^{-4}	2.6×10^{-4}
3	0.200	1.0×10^{-6}	2.7×10^{-6}	2.8×10^{-5}	2.6×10^{-4}
4	0.040	1.0×10^{-5}	3.1×10^{-8}	1.1×10^{-3}	7.9×10^{-5}
5	0.008	4.0×10^{-7}	8.0×10^{-8}	1.1×10^{-4}	8.8×10^{-5}
6	0.008	8.4×10^{-8}	8.4×10^{-8}	8.8×10^{-5}	8.8×10^{-5}
7	0.016	8.5×10^{-8}	8.4×10^{-8}	8.8×10^{-5}	8.8×10^{-5}
8	0.032	8.2×10^{-8}	8.3×10^{-8}	8.7×10^{-5}	8.8×10^{-5}
9	0.064	7.6×10^{-8}	7.7×10^{-8}	8.7×10^{-5}	8.8×10^{-5}
10	0.064	7.2×10^{-8}	7.1×10^{-8}	8.6×10^{-5}	8.7×10^{-5}

Table 5 (continued)

Step no.	h_n $\times 10^3$	$\|y_{0,n} - y(t_n)\|_2$ (20),$k_n \equiv 2$	(24),$k_n \equiv 2$	$\|y_{0,n} - y(t_n)\|_2$ for $h_n := 10 \times h_n$ (20),$k_n \equiv 2$	(24),$k_n \equiv 2$
1	1.000	0	0	0	0
2	1.000	1.3×10^{-11}	1.3×10^{-11}	1.3×10^{-7}	1.3×10^{-7}
3	0.200	1.7×10^{-11}	1.7×10^{-11}	1.7×10^{-7}	1.7×10^{-7}
4	0.040	1.8×10^{-11}	1.8×10^{-11}	1.8×10^{-7}	1.8×10^{-7}
5	0.008	1.8×10^{-11}	1.8×10^{-11}	1.8×10^{-7}	1.8×10^{-7}
6	0.008	1.8×10^{-11}	1.8×10^{-11}	1.9×10^{-7}	1.9×10^{-7}
7	0.016	1.8×10^{-11}	1.8×10^{-11}	1.9×10^{-7}	1.9×10^{-7}
8	0.032	1.9×10^{-11}	1.9×10^{-11}	2.0×10^{-7}	2.0×10^{-7}
9	0.064	2.0×10^{-11}	2.1×10^{-11}	2.1×10^{-7}	2.1×10^{-7}
10	0.064	2.2×10^{-11}	2.2×10^{-11}	2.2×10^{-7}	2.2×10^{-7}

References

[1] C. Arévalo and P. Lötstedt, *Improving the accuracy of BDF methods for index 3 differential-algebraic equations*, BIT 35 (1995), pp. 297–308.

[2] M. Berzins, P.M. Dew and R.M. Furzeland, *Developing PDE software using the method of lines and differential algebraic integrators*, Appl. Numer. Math. 5 (1989), pp.375–397.

[3] K.E. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Classics in Applied Mathematics, Vol.14, SIAM, 1996.

[4] K.E. Brenan and B.E. Engquist, *Backward differentiation approximations of nonlinear differential/algebraic systems*, Math. Comp. 51 (1988), pp. 659–676.

[5] C.W. Gear, H.H. Hsu and L.R. Petzold, *Differential-algebraic equations revisited*, in Proc. ODE Meeting, Oberwolfach, Germany, 1981.

[6] C.W. Gear and L.R. Petzold, *Singular implicit ordinary differential equations and constraints*, Report No. UIUCDCS-R-82-1110, Dept. of Computer Sci., Univ. of Illinois at Urbana-Champaign, 1982.

[7] A.C. Hindmarsh, *LSODE and LSODI, two new initial value ordinary differential equation solvers*, ACM-SIGNUM Newsletters 15 (1980), pp. 10–11.

[8] L.R. Petzold, *A description of DASSL: A differential/algebraic system solver*, in Scientific Computing, R.S. Stepleman et al, eds., North-Holland, Amsterdam, 1983, pp. 65–68.