

# Regular and singular $\beta$ -blocking of difference corrected multistep methods for nonstiff index-2 DAEs

Carmen Arévalo <sup>1</sup>

*Department of Scientific Computing and Statistics, Simón Bolívar University,  
Apartado 89000, Caracas 1080-A, Venezuela (camena@cesma.usb.ve)*

Claus Führer *and* Gustaf Söderlind <sup>2</sup>

*Numerical Analysis, Centre for Mathematical Sciences, Lund University, Box 118,  
S-221 00 Lund, Sweden (Claus.Fuhrer@na.lu.se; Gustaf.Soderlind@na.lu.se)*

---

## Abstract

There are several approaches to using nonstiff implicit linear multistep methods for solving certain classes of semi-explicit index 2 DAEs. Using  $\beta$ -blocked discretizations, [3], Adams–Moulton methods up to order 4 and difference corrected BDF [15] methods up to order 7 can be stabilized. As no extra matrix computations are required, this approach is an alternative to projection methods.

Here we examine some variants of  $\beta$ -blocking. We interpret earlier results as *regular*  $\beta$ -blocking and then develop *singular*  $\beta$ -blocking. In this nongeneric case the stabilized formula is *explicit*, although the discretization of the DAE as a whole is implicit. We investigate which methods can be stabilized in a broad class of implicit methods based on the BDF  $\rho$  polynomials. The class contains the BDF, Adams–Moulton and difference corrected BDF methods as well as other high order methods with small error constants. The stabilizing difference operator  $\tau$  is selected by a minimax criterion for the moduli of the zeros of  $\sigma + \tau$ . The class of explicit methods suitable as  $\beta$ -blocked methods is investigated. With singular  $\beta$ -blocking, Adams–Moulton methods up to order 7 can be stabilized with the stabilized method corresponding to the Adams–Bashforth methods.

*Key words:* differential algebraic equations (DAE),  $\beta$ -blocked methods, multistep methods, partitioned methods, half explicit methods, difference corrected multistep methods.

---

## 1 Introduction

We shall consider linear multistep discretizations of the general semi-explicit index 2 DAE in Hessenberg form,

$$\dot{x} = f(x, y) \tag{1a}$$

$$0 = g(x) \tag{1b}$$

where  $f : \mathbb{R}^{n_x+n_y} \rightarrow \mathbb{R}^{n_x}$ ,  $g : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$  and  $g_x f_y$  is invertible. Such problems occur in the simulation of e.g. electrical circuits and multibody mechanics.

Several different methods have been proposed for (1) with the goal of increasing computational accuracy and efficiency by using half-explicit or high-order methods [1–8,10,13,14]. In the multistep case, it is necessary to overcome special stability conditions to use high-order methods reaching the multistep order barrier. Partitioned multistep methods exploit the structure of (1) to treat the two variables or the two equations differently. This may result in a more or less conventional single discretization (cf. [1] for a general study, and [2,3] for  $\beta$ -blocked discretizations), or be based on alternating between two implicit discretizations [5,7]. In the former approach more of the classical theory remains applicable. In contrast, the latter approach requires a more complete reconsideration and yields a more complex computational scheme. A related approach is [10] which, using classical multistep theory, constructs two new methods of maximal order by choosing “optimal” method coefficients. Finally, projection methods make explicit use of  $g$  to stabilize a high-order solution by projecting it onto the constraints, but this requires additional matrix computations.

A multistep method  $(\rho, \sigma)$  is defined by its generating polynomials  $\rho(\zeta) = \sum_0^k \alpha_j \zeta^j$  and  $\sigma(\zeta) = \sum_0^k \beta_j \zeta^j$ . Using the forward shift operator  $E$ , we define the *difference operators*  $\rho = E^{-k} \rho(E)$ ,  $\sigma = E^{-k} \sigma(E)$ . Thus,

$$\rho x_n = \sum_{i=0}^k \alpha_{k-i} x_{n-i} \quad \sigma x_n = \sum_{i=0}^k \beta_{k-i} x_{n-i}.$$

In most cases, however, we shall represent the operators  $\rho$  and  $\sigma$  in terms of the backward difference operator  $\nabla = 1 - E^{-1}$ . The methods are normalized by the requirement  $\sigma 1 = 1$ .

We assume familiarity with our previous paper [3], which considered Euler–Lagrange equations,  $f(x, y) = f(x) - g_x^T(x)y$ , and introduced  $\beta$ -blocking to stabilize methods such as dcBDF and Adams–Moulton (AM) methods, which

---

<sup>1</sup> Partially funded by Simón Bolívar University Project DID–S1–CB–118 and CONICIT contract G-97-000592.

<sup>2</sup> Partially funded by TFR contracts 222/96–520 and 222/95–546.

are otherwise unstable when directly applied to index 2 problems. The instability is circumvented by adding a stabilizing term to the first equation,

$$h^{-1}\rho x_n = \sigma \left( f(x_n) - g_x^T(x_n)y_n \right) - g_x^T(x_n)\tau y_n \quad (2a)$$

$$0 = g(x_n), \quad (2b)$$

with  $\tau(\zeta) = \sum_0^k \gamma_j \zeta^j$  having the form  $\tau = c \cdot \nabla^k$ , where  $c$  is selected such that  $\sigma(\zeta) + \tau(\zeta)$  satisfies the root condition [11].

By taking  $\tau = -\beta_k \cdot \nabla^k$  we obtain an *explicit* stabilized formula  $(\rho, \sigma + \tau)$ . This case is non-generic, as the *implicit* methods obtained for values of  $c$  in a neighborhood of  $-\beta_k$  have  $\sigma + \tau$  operators not satisfying the root condition. We refer to the generic case as *regular*  $\beta$ -blocking, and the non-generic case as *singular*  $\beta$ -blocking. In this paper we shall develop the latter technique. While regular blocking cannot stabilize all AM methods of interest, singular blocking can be used to stabilize several more AM methods in which case the stabilized formulas  $(\rho, \sigma + \tau)$  correspond to the Adams–Bashforth (AB) methods.

In Section 2 we prove a general convergence theorem for  $\beta$ -blocked discretizations of index 2 DAEs and discuss some computational aspects. In Section 3 we establish a new class of nonstiff difference correction methods, explore which of the implicit (IDC) formulas can be regularly  $\beta$ -blocked and discuss appropriate choices of methods. Then we show how to obtain a class of explicit difference correction (EDC) methods corresponding to the IDC methods, and show how singular blocking corresponds to combining IDC and EDC methods. In Section 4 we verify the theoretical results numerically for a nonlinear test problem. We also point to the relation between  $\beta$ -blocked methods and the half-explicit Runge-Kutta methods which have recently been developed for nonstiff higher index DAEs, cf. [4,6,8,13,14].

## 2 Regular and singular $\beta$ -blocked discretizations

In view of (2), a simple generalization of  $\beta$ -blocking to the nonlinear semi-explicit index 2 DAE (1) is

$$h^{-1}\rho x_n = \sigma f(x_n, y_n) + f_y(x_n, y_n)\tau y_n \quad (3a)$$

$$0 = g(x_n). \quad (3b)$$

As this scheme requires extra computations of the Jacobian  $f_y$ , it is natural to seek alternative discretizations using only function evaluations. Let  $(\rho, \sigma)$  be implicit. Introducing  $\tilde{\sigma}x_n = \sum_1^k \beta_{k-i}x_{n-i} = (\sigma - \beta_k)x_n$ , (1) is discretized as

$$h^{-1}\rho x_n = \beta_k f(x_n, (1 + \beta_k^{-1}\tau)y_n) + \tilde{\sigma} f(x_n, y_n) \quad (4a)$$

$$0 = g(x_n). \quad (4b)$$

This discretization shows that in a  $\beta$ -blocked method the operator acting on  $x$  is  $\sigma$ , while the one acting on  $y$  is the corresponding  $\beta$ -blocked operator,  $\sigma + \tau$ . For Euler–Lagrange equations, the discretizations above reduce to (2). By (4) the regularly  $\beta$ -blocked discretization can also be written

$$h^{-1}\rho x_n = \beta_k f(x_n, y_n + \delta y_n) + \tilde{\sigma} f(x_n, y_n) \quad (5a)$$

$$0 = g(x_n). \quad (5b)$$

Thus, a regularly  $\beta$ -blocked method can, in principle, be based on a standard multistep method that solves for  $x_n$  and  $z_n = y_n + \delta y_n$  on each step, with a modification to calculate and discard  $\delta y_n$  before proceeding to the next step, thereby preventing unstable error propagation.

## 2.1 Stabilization objectives

In [3] it was shown that different  $\beta$ -blocked methods had different stability characteristics for the algebraic variable. Governed by the roots of the characteristic equation  $\sigma(\zeta) + \tau(\zeta) = 0$ , the error propagation in  $y$  varies both with the method  $(\rho, \sigma)$  and the stabilizer  $\tau$ . The dcBDF methods have all roots of  $\sigma(\zeta) + \tau(\zeta) = 0$  at the origin, while Adams methods always have nonzero roots for  $k \geq 2$ . The *stabilization objective* therefore affects the choice of method and vice versa.

There are several different ways to choose  $\tau$ . The strongest objective is to remove the instability completely rather than just provide sufficient damping. In terms of control theory, this is known as *deadbeat control*, [2], and corresponds to  $\sigma(\zeta) + \tau(\zeta) = 0$  having all roots at the origin.

**Theorem 1** *Deadbeat stabilization is equivalent to taking  $\tau = 1 - \sigma$ . The dcBDF methods are the only  $k$ -step methods having convergence order  $p = k + 1$  for the differential variables and deadbeat stabilization.*

**PROOF.** As  $\tau = c \cdot \nabla^k$  lest the order of the method drop, we have  $\tau(\zeta) = c \cdot (\zeta - 1)^k$ ; hence it follows that  $\tau(1) = 0$ , and therefore, by the normalization  $\sigma(1) = 1$ , deadbeat requires  $\sigma(\zeta) + \tau(\zeta) = \zeta^k$ . In terms of difference operators, this condition becomes  $\sigma = 1 - \tau$ . Hence  $\sigma = 1 - c \cdot \nabla^k$ . Order  $p = k + 1$  is then only attained by taking  $c = 1/(k + 1)$  and the BDF  $\rho$  operator, i.e., by the dcBDF methods.  $\square$

For AM methods the choice  $\tau = c \cdot \nabla^k$  retains the order but cannot achieve deadbeat stabilization; conversely the deadbeat stabilizer  $\tau = 1 - \sigma$  is  $O(h)$  for  $k \geq 2$ , and causes a loss of order. For methods other than the dcBDF one must therefore settle for a weaker stabilization objective. As a method is still useful as long as  $\sigma(\zeta) + \tau(\zeta)$  satisfies the root condition, we choose a minimax criterion for the roots as our objective in the regular  $\beta$ -blocking case.

## 2.2 Singular $\beta$ -blocked discretizations

Regular  $\beta$ -blocking attempts to construct an implicit method  $(\rho, \sigma + \tau)$ , with  $\tau = c \cdot \nabla^k$ , so that the  $k$ th degree polynomial  $\sigma(\zeta) + \tau(\zeta)$  has minimal roots. By contrast, singular  $\beta$ -blocking selects  $\tau = -\beta_k \cdot \nabla^k$ , causing  $(\rho, \sigma + \tau)$  to become explicit. In this case the discretization (4) becomes

$$h^{-1}\rho x_n = \beta_k f(x_n, (1 - \nabla^k)y_n) + \tilde{\sigma} f(x_n, y_n) \quad (6a)$$

$$0 = g(x_n). \quad (6b)$$

This leaves no choice for a free parameter  $c$ ; we require only that  $\sigma(\zeta) + \tau(\zeta)$  satisfies the root condition at  $c = -\beta_k$ . Note that, in spite of  $(\rho, \sigma + \tau)$  being explicit, the scheme (6) remains an implicit discretization.

## 2.3 Computational aspects

To illustrate singular  $\beta$ -blocking, we take the AM methods as an example. In the case  $k = 1$ , i.e. for the trapezoidal rule, singular blocking takes  $\tau = -\nabla/2$ . The scheme (6) takes the form of an AM/AB discretization,

$$h^{-1}\nabla x_n = \frac{1}{2} (f(x_n, y_{n-1}) + f(x_{n-1}, y_{n-1})) \quad (7a)$$

$$0 = g(x_n). \quad (7b)$$

Thus the effect of the blocking is that the algebraic variable is treated by an “explicit” method, in this case AB1, the explicit Euler method. The method works in the following way. Take  $n = 1$  and suppose that  $x_0$  is given; on this first step we solve this system simultaneously for  $x_1$  and  $y_0$ . The overall scheme is implicit and the determination of  $y_1$  is delayed until the computation of  $x_2$ .

The trapezoidal rule can be interpreted both as an AM method and a dcBDF method. In the latter case we use regular blocking to achieve deadbeat stabilization with  $\tau = +\nabla/2$ , which (in this case) is just the opposite stabilizer

compared with singular blocking. The scheme (4) takes the form

$$h^{-1}\nabla x_n = \frac{1}{2} \left( f(x_n, 2y_n - y_{n-1}) + f(x_{n-1}, y_{n-1}) \right) \quad (8a)$$

$$0 = g(x_n). \quad (8b)$$

This corresponds to using a dcBDF/BDF combination, i.e. the trapezoidal rule combined with the implicit Euler method. We note that the scheme requires initial values  $x_0$  and  $y_0$  and computes  $x_1$  and  $y_1$  simultaneously on the first step. By contrast, the singular blocking needs only the initial value  $x_0$  to get started; this may in some situations be more natural.

Both techniques above yield fully implicit discretizations. A computational process akin to the partitioned multistep [5], [1, p. 153] or Runge-Kutta methods [4,6,8,13,14] can be obtained if the nonlinear equation is solved by applying fixed point iteration to the differential variables and Newton's method to the algebraic variables. The computation is then split into two separate parts, iterating back and forth between the differential equation and the constraint. Thus, one would first predict  $x_n^{(0)}$  and then formally solve for  $x_n^{(1)}$  from (7a) in the singular case, and from (8a) in the regular case; the latter yields

$$x_n^{(1)} = x_{n-1} + \frac{h}{2} \left( f(x_n^{(0)}, 2\eta - y_{n-1}) + f(x_{n-1}, y_{n-1}) \right), \quad (9)$$

where  $\eta =: y_n^{(0)}$  is determined by solving the constraint equation  $0 = g(x_n^{(1)})$  using Newton's method. This is readily seen to be possible due to the index 2 assumption of  $g_x f_y$  being invertible. Finally the numerical value of  $x_n^{(1)}$  is obtained by evaluating (9) at the point  $(x_n^{(0)}, y_n^{(0)})$ . In this way the implicit part of the discretization requiring the solution of linear systems of equations is reduced to the constraint equation.

Likewise in the singular case we let

$$x_n^{(1)} = x_{n-1} + \frac{h}{2} \left( f(x_n^{(0)}, \eta) + f(x_{n-1}, \eta) \right) \quad (10)$$

and solve  $0 = g(x_n^{(1)})$  for  $\eta = y_{n-1}^{(0)}$ . In a similar way we obtain  $x_n^{(1)}$  by evaluating (10) at the point  $(x_n^{(0)}, y_{n-1}^{(0)})$ . The iteration can be organized in many different ways both for regularly and singularly blocked discretizations. A natural choice is to attempt to use a single Newton iteration for  $y^{(l-1)}$  in the constraint equation before  $x^{(l)}$  is updated from the discretization and then repeat the computation. Thus one would use one fixed point and one Newton iteration in tandem in the iterative solution process.

We note in (10) that the singular blocking technique requires two  $f$  evaluations per iteration. This is justified if the method permits larger steps. For example, in nonstiff computations one may expect around 25% larger steps with AM3

than with dcBDF3, due to the smaller error constants of the AM methods (cf. Section 3). Therefore, if the work of  $f$  evaluations is  $W_f$  and all other work (including evaluations of  $g$ , Jacobians, factorizations, equation solving, computation of divided differences, control strategies for stepsize and iteration, norms, error estimation, etc.) is  $W_o$ , then a simple model of total work is  $W_o + W_f$  for dcBDF3/BDF3 and  $W_o + 2W_f$  for AM3/AB3. But if the latter permits 25% larger steps, total work per integrated unit of time is the same if

$$W_o + W_f = \frac{4}{5}(W_o + 2W_f). \quad (11)$$

AM3/AB3 would therefore be more efficient if  $W_f \leq W_o/3$ , which is a fairly modest requirement. Higher order methods further relax this requirement as may minor differences in the iteration's rate of convergence. In conclusion, the number of  $f$ -evaluations alone does not reflect actual method performance.

#### 2.4 Convergence

The convergence theorem for  $\beta$ -blocked methods in [3] is now extended to (1) as follows.

**Theorem 2** *Let  $(\rho, \sigma)$  be a  $k$ -step linear multistep method of order  $p = k$  or  $p = k + 1$  and consider the discretizations (3), (4) and (6) of the index 2 problem (1). Further, assume that  $\sigma(\zeta) + \tau(\zeta)$  satisfies the root condition and that  $\tau y(t_n) = O(h^k)$  for any sufficiently smooth function. If the starting values are  $x_i - x(t_i) = O(h^p)$  and  $y_i - y(t_i) = O(h^k)$  for  $i = 0 : k - 1$ , then the discretizations (3), (4) and (6) are convergent such that, for  $nh = t$  constant,*

$$x_n - x(t) = O(h^p) \quad \text{and} \quad y_n - y(t) = O(h^k).$$

**PROOF.** The proof for the discretization (3) is similar to that of Theorem 2.1 in [3]. With the obvious changes in the definitions of  $A_j(h)$ ,  $Q(h)$  and  $P(h)$ , substituting  $-f_y$  for  $G^T$  and  $-g_x$  for  $G$ , the proof proceeds in the same manner. Since  $f(x_n, y_n + \tau y_n) - f(x_n, y_n) = f_y \tau y_n + O((\tau y_n)^2) = f_y \tau y_n + O(h^{2k})$ , stability as well as order is preserved also in (4).

The basic proof of convergence from [3] needs further modifications to apply to singularly blocked discretizations (6). These modifications do not concern the proof technique, which remains the same, but only the order in which the variables  $x_n$  and  $y_n$  are computed. Thus, as the blocked operator  $\sigma + \tau$  corresponds to an explicit method, the computation of  $y_n$  is delayed one step, as shown in the previous section. When this computational process is described in companion matrix form (cp. [3, p. 5]) there is one less error vector to consider: the error in  $y_n$  does not enter on the present step as  $y_n$  will not be

computed until the next. Therefore the “ $\beta$  block” matrix will also be of one dimension less, i.e., it will be a  $(k-1) \times (k-1)$  matrix, whose eigenvalues are the  $k-1$  zeros of the (explicit) blocked operator  $\sigma + \tau$ . Note that if  $k = 1$  as in (7), then the  $\beta$  block matrix is void and  $\sigma + \tau$  has no zeros. Thus, (7) proceeds with  $x_0$  as its single initial value, and once  $y_0$  is selected such that  $x_1$  satisfies the constraints it has no further influence on the solution. With this exception of vector and matrix dimensions, the convergence proof for singularly blocked discretizations is analogous to that of regularly blocked methods.  $\square$

### 3 Difference correction methods

We are interested in finding multistep methods  $(\rho, \sigma)$  that can be  $\beta$ -blocked by an appropriate  $\tau$ . To this end, we will construct a family of multistep methods based on applying difference corrections of different orders to the BDF methods. This creates a wide class of formulas including BDFs, AM methods and the dcBDFs as three particular cases.

Let  $G_\rho(\nabla)$  and  $G_\sigma(\nabla)$  denote the generating functions for the coefficients of  $\rho$  and  $\sigma$ , respectively, expressed in terms of the backward difference operator  $\nabla$ , in a method class  $(\rho, \sigma)$  proceeding with a time-step  $h$ . It is then well-known, cp. [12, p. 192ff], that

$$\frac{G_\rho(\nabla)}{G_\sigma(\nabla)} = -\log(1 - \nabla) = h \frac{d}{dt}. \quad (12)$$

Here the left equality holds in the sense of analytic functions, near  $\nabla = 0$  on the  $\log 1 = 0$  branch; cf. the requirement  $\rho(\zeta)/\sigma(\zeta) = \log \zeta + O((\zeta - 1)^{p+1})$  near  $\zeta = 1$ . The right equality holds in the sense of operator calculus, in which it represents Taylor’s theorem. Actual operators  $\rho$  and  $\sigma$  are obtained as truncations of the power series expansions of  $G_\rho(\nabla)$  and  $G_\sigma(\nabla)$ , respectively. For method construction purposes it is therefore not necessary to study the convergence of these operator series; it is immediately clear that  $h^{-1}\rho P(t) = \sigma \dot{P}(t)$  for all polynomials with  $\deg(P) \leq p$ , where  $p$  is the order of the rational approximation  $\rho(\zeta)/\sigma(\zeta) \approx \log \zeta$ .

For the BDF methods, it is well-known that the generating function for the coefficients of  $\rho^{\text{BDF}}$  is obtained by prescribing  $G_\sigma^{\text{BDF}}(\nabla) = \sigma^{\text{BDF}} = 1$ . Thus  $G_\rho^{\text{BDF}}(\nabla) = -\log(1 - \nabla) = \nabla + \nabla^2/2 + \nabla^3/3 + \dots$ . Similarly, the AM methods are obtained by fixing  $G_\rho^{\text{AM}}(\nabla) = \rho^{\text{AM}} = \nabla$ ; hence the generating function for  $\sigma^{\text{AM}}$  is  $G_\sigma^{\text{AM}}(\nabla) = -\nabla/\log(1 - \nabla) = 1 - \nabla/2 - \nabla^2/12 - \nabla^3/24 - \dots$ .

$\rho_i$	$\sigma_{ij} \quad (j = 0, \dots, 6)$						
$\nabla$	1	$-\frac{1}{2}\nabla$	$-\frac{1}{12}\nabla^2$	$-\frac{1}{24}\nabla^3$	$-\frac{19}{720}\nabla^4$	$-\frac{3}{160}\nabla^5$	$-\frac{863}{60480}\nabla^6$
$+\frac{1}{2}\nabla^2$	1		$-\frac{1}{3}\nabla^2$	$-\frac{1}{12}\nabla^3$	$-\frac{17}{360}\nabla^4$	$-\frac{23}{720}\nabla^5$	$-\frac{143}{6048}\nabla^6$
$+\frac{1}{3}\nabla^3$	1			$-\frac{1}{4}\nabla^3$	$-\frac{3}{40}\nabla^4$	$-\frac{11}{240}\nabla^5$	$-\frac{109}{3360}\nabla^6$
$+\frac{1}{4}\nabla^4$	1				$-\frac{1}{5}\nabla^4$	$-\frac{1}{15}\nabla^5$	$-\frac{3}{70}\nabla^6$
$+\frac{1}{5}\nabla^5$	1					$-\frac{1}{6}\nabla^5$	$-\frac{5}{84}\nabla^6$
$+\frac{1}{6}\nabla^6$	1						$-\frac{1}{7}\nabla^6$

Table 1

Implicit Difference Correction methods. To obtain the IDC $ij$  method, add the terms of  $\rho_i$  columnwise down to row  $i$ ; turn right, sum terms up to order  $j$  for  $\sigma_{ij}$ , starting with the unit operator in column 0. Methods ending in column  $j \neq 0$  have step number  $k = j$  and order  $p = k + 1$ . The error constant of each method equals the coefficient of the first neglected term in  $\sigma_{ij}$ . For stability  $i \leq 6$  is required, but further terms not shown to the right may be added at wish. The AM $k$  methods correspond to the top row (IDC1 $k$ ), the BDF $k$  to the left column (IDC $k$ 0) and the dcBDF $k$  to the diagonal (IDC $k$  $k$ ). All stability regions are bounded except for the BDFs and the trapezoidal rule, IDC11. Shaded methods can be regularly  $\beta$ -blocked.

### 3.1 Implicit difference correction methods

As all AM methods share the same  $\rho^{\text{AM}} = \nabla$ , they can be viewed as successively refined methods obtained by including additional difference corrections in the  $\sigma$  operator. The same technique can be used also when the  $\rho$  operator is taken from a higher order BDF method. Thus, if we let

$$\rho_i^{\text{BDF}} = \sum_{m=1}^i \frac{\nabla^m}{m}, \quad (13)$$

we can for each different value of  $i$  obtain a new family of methods, referred to as *implicit difference correction methods* (IDC methods), by prescribing

$$G_\rho^{\text{IDC}i(\cdot)}(\nabla) = \rho_i^{\text{BDF}}. \quad (14)$$

The generating function for the corresponding  $\sigma$  operators then becomes

$$G_\sigma^{\text{IDC}i(\cdot)}(\nabla) = -\frac{\rho_i^{\text{BDF}}}{\log(1 - \nabla)}. \quad (15)$$

**Definition 3** The IDC $ij$  method  $(\rho_i, \sigma_{ij})$  is defined by  $\rho_i = \rho_i^{\text{BDF}}$  and the  $\sigma_{ij}$  operator obtained by expanding  $-\rho_i^{\text{BDF}}/\log(1 - \nabla)$  in powers of  $\nabla$ , retaining powers up to order  $j \geq 0$ .

	AM2	AM3	IDC23	IDC24	IDC34	IDC45	IDC56
$c^*$	0.146	0.092	0.158	0.112	0.150	0.139	0.128
damping	0.33	0.68	0.50	0.86	0.66	0.81	0.96

Table 2

Optimal coefficient and corresponding damping of  $\beta$ -blockable IDC $ij$  methods. (Trivial data for the dcBDF methods are not included.)

**Theorem 4** *The IDC $ij$  methods are stable and convergent for  $1 \leq i \leq 6$ . For  $j \geq i$  the step number is  $k = j$  and the classical order is  $p = k + 1$ . For  $0 \leq j < i$  the methods are identical to the BDF methods for which  $p = k = i$ .*

**PROOF.** We only consider  $j \geq i$ . As the IDC methods are based on the BDF  $\rho$  operators which are known to be stable for  $k \leq 6$ , every IDC $ij$  method is stable and convergent for  $i \leq 6$ , independently of  $j$ . The order of the dcBDF methods IDC $jj$  is  $j+1$ , [3,15], and for each added term in  $\sigma$  the order increases by one, independently of  $i$ .  $\square$

As pointed out, the dcBDF $k$  can be regularly  $\beta$ -blocked for  $k = 1 : 6$ , and AM $k$  for  $k = 1 : 3$  by taking  $\tau = c \cdot \nabla^k$ . Like AM methods, not all the IDCs can be regularly  $\beta$ -blocked. To investigate when this is possible, we add  $\tau = c \cdot \nabla^k$  to each  $\sigma$  from Table 1 and compute  $c$  as the minimizer  $c^*$  of the largest modulus of a zero of  $\sigma(\zeta) + \tau(\zeta) = \sigma(\zeta) + c \cdot (\zeta - 1)^k$ . In Table 1 those IDC $ij$  methods which can be stabilized by regular  $\beta$ -blocking are shaded. These include all of the dcBDFs (orders 2–7), the first three AM formulas (orders 2–4) and five new methods: IDC23, IDC24, IDC34, IDC45 and IDC56 (orders 4–7). The optimal  $c^*$  and the magnitude of the largest zero of  $\sigma(\zeta) + \tau^*(\zeta)$  of these methods are displayed in Table 2. For a fixed order  $p$ , the size of the maximal root diminishes as the row number increases. At the same time the error constant increases; accuracy is traded for stability.

The investigation of IDC methods shows that there is a possibility to use regular  $\beta$ -blocking with several methods other than those considered in [3]. Good damping requires that the minimax root  $|\zeta^*|$  be well within the unit circle. An indication of how large values can be tolerated is given by the size of the extraneous roots (roots other than the principal root  $\zeta = 1$ ) of  $\rho(\zeta) = 0$ . For all methods except dcBDF/BDF,  $|\zeta^*|$  is larger than the maximum extraneous root of the corresponding  $\rho(\zeta)$ . Experience with BDF and dcBDF methods for ODEs shows that the extraneous roots rarely cause any difficulties, yet for BDF4, 5 and 6, the maximum modulus of the extraneous roots of  $\rho(\zeta)$  are as large as .561, .709 and .863, respectively. This indicates that all of the  $\beta$ -blocked methods above are acceptable, except possibly IDC56, cf. Table 2.

### 3.2 Explicit difference correction methods

We need a convention for the  $\sigma$  operators of explicit methods. Previously we split a  $k$ -step operator  $\sigma x_n = \beta_k x_n + \tilde{\sigma} x_n$ . Thus, for *explicit methods* we have  $\sigma = \tilde{\sigma}$ . Note that  $\hat{\sigma} := E\tilde{\sigma}$  has a representation in terms of a polynomial in  $\nabla$  of degree at most  $k - 1$ . Since  $\tilde{\sigma} = E^{-1}\hat{\sigma}$  and  $E^{-1} = 1 - \nabla$  it follows that  $\tilde{\sigma} = (1 - \nabla)\hat{\sigma}$ , where  $\hat{\sigma}$  fully characterizes the  $\tilde{\sigma}$  of an explicit method.

**Lemma 5** *Let  $(\rho_k, \sigma_k)$  be an implicit  $k$ -step method of order  $p = k$  or  $p = k + 1$ . Then the following factorization holds:*

$$\sigma_k - \beta_k \cdot \nabla^k = (1 - \nabla) \cdot \hat{\sigma}_k, \quad (16)$$

where  $(\rho_k, (1 - \nabla) \cdot \hat{\sigma}_k)$  is an explicit  $k$ -step method of order  $p = k$ .

**PROOF.** Only the statement about method orders needs to be proved. If the implicit method  $(\rho, \sigma)$  has convergence order at least  $k$  and we change  $\sigma$  by  $\beta_k \cdot \nabla^k = O(h^k)$ , then the order of consistency drops to  $k$ . It follows that the explicit method  $(\rho, (1 - \nabla)\hat{\sigma})$  has convergence order  $p = k$ .  $\square$

Some particular cases of Lemma 5 are of importance. For Adams methods, let  $\sigma^{\text{AM}k}$  and  $\hat{\sigma}^{\text{AB}k}$  denote the  $\sigma$  operators of the  $k$ -step Adams–Moulton and Adams–Bashforth methods, respectively. Then

$$\sigma^{\text{AM}k} - \beta_k^{\text{AM}k} \cdot \nabla^k = (1 - \nabla) \cdot \hat{\sigma}^{\text{AB}k}. \quad (17)$$

More generally, Lemma 5 asserts that to each implicit method there is an explicit method which can be obtained through (16). Thus, for each IDC method from Table 1, we can construct its explicit counterpart, the *explicit difference correction methods* (EDCs). These are still based on the BDF  $\rho$  polynomials, but the generating function for the coefficients of  $\hat{\sigma}^{\text{EDC}}$  is

$$G_{\hat{\sigma}}^{\text{EDC}i(\cdot)}(\nabla) = - \frac{\rho_i^{\text{BDF}}}{(1 - \nabla) \log(1 - \nabla)}. \quad (18)$$

The factor  $1 - \nabla$  in the denominator is a backward shift accounting for the explicitness of the methods.

**Definition 6** *The EDC $ij$  method  $(\rho_i, (1 - \nabla)\hat{\sigma}_{ij})$  is defined by  $\rho_i = \rho_i^{\text{BDF}}$  and the  $\hat{\sigma}_{ij}$  operator obtained by expanding  $-\rho_i^{\text{BDF}} / ((1 - \nabla) \log(1 - \nabla))$  in powers of  $\nabla$ , retaining powers up to order  $j \geq i - 1$ .*

**Theorem 7** *The EDC $ij$  methods are stable and convergent for  $1 \leq i \leq 6$ . For  $j \geq i - 1$  the step number is  $k = j + 1$  and the classical order is  $p = k$ .*

$\rho_i$	$\hat{\sigma}_{ij} \quad (j = 0, \dots, 6)$						
$\nabla$	1	$+\frac{1}{2}\nabla$	$+\frac{5}{12}\nabla^2$	$+\frac{3}{8}\nabla^3$	$+\frac{251}{720}\nabla^4$	$+\frac{95}{288}\nabla^5$	$+\frac{19087}{60480}\nabla^6$
$+\frac{1}{2}\nabla^2$	$\mathcal{E}_1$		$+\frac{2}{3}\nabla^2$	$+\frac{7}{12}\nabla^3$	$+\frac{193}{360}\nabla^4$	$+\frac{121}{240}\nabla^5$	$+\frac{14531}{30240}\nabla^6$
$+\frac{1}{3}\nabla^3$	$\mathcal{E}_2$			$+\frac{3}{4}\nabla^3$	$+\frac{27}{40}\nabla^4$	$+\frac{151}{240}\nabla^5$	$+\frac{401}{672}\nabla^6$
$+\frac{1}{4}\nabla^4$	$\mathcal{E}_3$				$+\frac{4}{5}\nabla^4$	$+\frac{11}{15}\nabla^5$	$+\frac{29}{42}\nabla^6$
$+\frac{1}{5}\nabla^5$	$\mathcal{E}_4$					$+\frac{5}{6}\nabla^5$	$+\frac{65}{84}\nabla^6$
$+\frac{1}{6}\nabla^6$	$\mathcal{E}_5$						$+\frac{6}{7}\nabla^6$

Table 3

Explicit Difference Correction EDC $ij$  methods ( $\rho_i, (1 - \nabla)\hat{\sigma}_{ij}$ ). As  $1 - \nabla$  is merely a backward shift, the methods can be directly read from the table by accounting for this shift. Add the terms of  $\rho_i$  columnwise down to row  $i$ ; turn right, sum terms up to order  $j$  for  $\hat{\sigma}_{ij}$ , starting with the extrapolation operator  $\mathcal{E}_{i-1}$ . Methods ending in column  $j \neq 0$  have step number  $k = j + 1$  and order  $p = k$ ; thus e.g. EDC33 is a 4-step, 4th order method. The error constant of each method equals the coefficient of the first neglected term in  $\hat{\sigma}_{ij}$ . The AB $k$  methods correspond to the top row.

**PROOF.** Stability and convergence follow from the BDF  $\rho$  operators. By applying Lemma 5 to the IDC methods for  $j \geq i$ , we obtain the factorization (note the index convention)

$$\sigma^{\text{IDC}ij} - \beta_k^{\text{IDC}ij} \cdot \nabla^k = (1 - \nabla) \cdot \hat{\sigma}^{\text{EDC}i,j-1}, \quad (19)$$

from which the order  $p = k$  follows.  $\square$

The EDC $ij$  methods are displayed in Table 3. Their structure is special. For example, EDC34 has

$$\hat{\sigma}^{\text{EDC}34} = 1 + \nabla + \nabla^2 + \frac{3}{4}\nabla^3 + \frac{27}{40}\nabla^4 = \mathcal{E}_2 + \frac{3}{4}\nabla^3 + \frac{27}{40}\nabla^4, \quad (20)$$

where the *extrapolation operator*  $\mathcal{E}_2 = \sum_{m=0}^2 \nabla^m = 1 + \nabla + \nabla^2$ ; these terms have accordingly been collected in column 0 of Table 3. In a similar way EDC32, which is the 3-step, third order Explicit Differentiation Formula EDF3 (the explicit counterpart to BDF3) has  $\hat{\sigma}^{\text{EDC}32} = \mathcal{E}_2$ . This is a second order approximation to the forward shift  $(1 - \nabla)^{-1}$ ; note that because EDF3 is an *explicit* method its  $\sigma$  operator must employ extrapolation on back values of  $f$  to match the derivative  $\dot{x}(t_n)$  which its (BDF3)  $\rho$  operator attempts to approximate to third order accuracy. The EDFs are displayed in the left column of Table 3; it should be noted that the dcBDF $k$  method (IDC $kk$ ) is turned by singular  $\beta$ -blocking into the the EDF $k$  (which equals EDC $k, k - 1$ ; cp. (19)), whereas regular (deadbeat) blocking turns it into the BDF $k$ .

$i,$	$j:$	0	1	2	3	4	5	6	7
1			–	0.33	0.47	0.63	0.81	0.98	1.16
2				0.50	0.50	0.61	0.76	0.93	1.10
3					0.58	0.66	0.77	0.91	1.07
4						0.71	0.81	0.93	1.06
5							0.85	0.97	1.09
6								1	1.13

Table 4

Magnitude of the largest zero of  $\sigma(\zeta) - \beta_k \cdot (\zeta - 1)^k$  for singularly  $\beta$ -blocked IDC $ij$  methods. The 8th order IDC methods cannot be stabilized; e.g. in AM7 the maximum root of the  $\beta$ -blocked operator  $\sigma^{\text{AB7}}(\zeta)$  is 1.16.

In singular blocking  $(1 + \beta_k^{-1}\tau)y_n = (1 - \nabla^k)y_n = \mathcal{E}_{k-1}y_{n-1}$ , as  $\tau = -\beta_k \nabla^k$ . Hence the general form of the singularly  $\beta$ -blocked discretization:

$$h^{-1}\rho x_n = \beta_k f(x_n, \mathcal{E}_{k-1}y_{n-1}) + \tilde{\sigma} f(x_n, y_n) \quad (21a)$$

$$0 = g(x_n). \quad (21b)$$

As the  $y$  argument in (21a) is just a  $(k-1)$ th order prediction of  $y_n$ , the singular (just like the regular) scheme approaches the plain multistep discretization as  $k$  increases. Therefore stability must eventually break down for some  $k$ , and we need to investigate the limitations of singular blocking.

From (19) we know that singular  $\beta$ -blocking transforms each  $\sigma^{\text{IDC}}$  into a  $\hat{\sigma}^{\text{EDC}}$ . Hence we must examine the  $\hat{\sigma}^{\text{EDC}}$  operators of Table 3. Computing the maximum moduli of the zeros of the characteristic equation  $\hat{\sigma}^{\text{EDC}}(\zeta) = 0$  yields the results shown in Table 4. There we see that the singular blocking is capable of stabilizing all AM methods up to the seventh order AM6. Likewise, all IDC methods shown in Table 1 can be stabilized, even if the damping (magnitude of max zero) is often rather weak. Since the AM methods have the smallest error constants and sufficient damping up to AM5, the suite AM1–5 is likely to be the best choice of singularly  $\beta$ -blocked formulas.

For ODEs, the most interesting method suites in the IDC Table 1 are the BDF, dcBDF and AM methods. The corresponding explicit methods are primarily useful as *predictors* for the above methods; they guarantee, through (19), that the predictor-corrector difference is proportional to  $\nabla^k x_n$ , a computationally desirable property in practical implementations, [15]. In connection with  $\beta$ -blocking, however, one seeks methods superior to the BDFs. For regular  $\beta$ -blocking the dcBDF methods are the most robust alternative, and for singular  $\beta$ -blocking an AM/AB combination is the most accurate. Full method suites are available in both cases, as are alternatives from the IDC/EDC class.

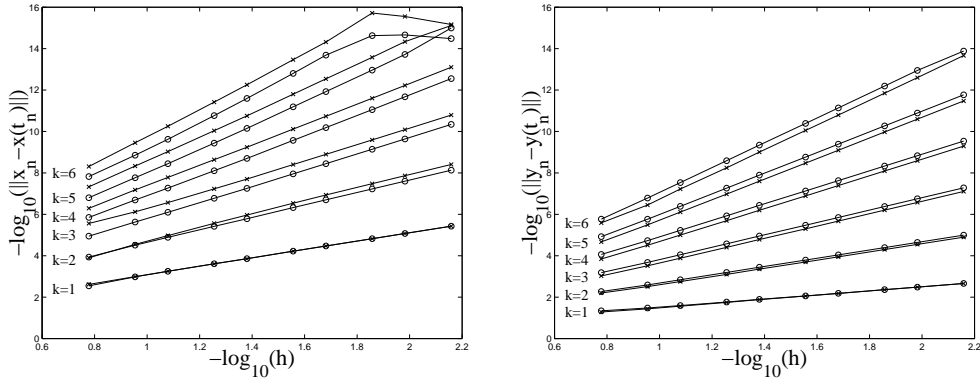


Fig. 1. Global error vs. stepsize for differential variable  $x_1$  (left) and algebraic variable  $y$  (right) for regularly  $\beta$ -blocked dcBDF (o) and singularly  $\beta$ -blocked Adams–Moulton ( $\times$ )  $k$ -step methods.

#### 4 Numerical verification

To verify the order of convergence of regularly and singularly  $\beta$ -blocked formulas, we have applied the dcBDF/BDF and AM/AB methods to the following nonlinear index 2 system,

$$\dot{x}_1 = -2\sqrt{x_1 y} - x_2 \quad (22a)$$

$$\dot{x}_2 = -y^2/x_2 \quad (22b)$$

$$0 = x_1 x_2 + x_2^2 \quad (22c)$$

with initial values  $x_1(0) = 1$ ,  $x_2(0) = -1$ ,  $y(0) = 1$  and analytic solution  $x_1(t) = e^{-t}$ ,  $x_2(t) = -e^{-t}$ ,  $y(t) = e^{-t}$ .

The problem was solved with the suites of regular dcBDF $k$ /BDF $k$  and singular AM $k$ /AB $k$  formulas for  $k = 1 : 6$  with constant stepsize, taking  $h = 1/N$  with  $N = 6, 9, 12, 18, 24, 36, 48, 72, 96$  and 144. The difference equations were solved by Newton iteration to full precision. The global errors in the differential variable  $x_1$  and the algebraic variable  $y$  were measured in the discretized  $L^1$  norm on the interval  $[0, 1]$  by taking the mean global error over all steps. This norm evens out minor fluctuations which often occur when errors are measured at a single point, thereby making it possible also to discern differences in error constants. Thus the test is constructed to be a numerical/mathematical verification of the properties of regular and singular  $\beta$ -blocking, rather than a software test.

Figure 1 shows the global errors in  $x_1$  and  $y$  on  $[0, 1]$  as functions of the stepsize  $h$ . In every case we observe the predicted order,  $k + 1$ , for the differential variable and  $k$  for the algebraic one, by reading the slope of the corresponding lines. The difference between each pair of formulas of the same order reflects the relative size of their error constants. As these are smaller for the AM

formulas than for dcBDFs, there is a small advantage of the singular  $\beta$ -blocked AM methods in the  $x$ -approximation. The opposite is true in the case of the algebraic variable  $y$ , as the BDFs are more accurate than the AB formulas. For the higher order discretizations, the singular  $\beta$ -blocking AM $k$  formulas are approximately 3 times as accurate in  $x$  as their corresponding regular  $\beta$ -blocked dcBDF $k$  discretizations, but the latter are approximately twice as accurate in  $y$  as their singular counterparts.

Finally, we remark that the numerical comparison [5] between a partitioned method based on alternating between Adams-Moulton and BDF methods, and the same methods stabilized by regular  $\beta$ -blocking, as described in [3] for the Euler-Lagrange equations, is performed on the same test problem (2D truck model) as was used in [3]. As expected, this comparison shows that the different approaches yield virtually identical accuracy, thus corroborating the results in [3] and the present paper. As for singular  $\beta$ -blocking vs. regular  $\beta$ -blocking, the difference in accuracy is only a matter of differences in error constants; hence this aspect is clearly demonstrated in the test above. True performance evaluation will however require tests which apply a full variable-step code to large-scale problems. Such tests are beyond the scope of this paper.

## Acknowledgements

The authors would like to thank Anne Kværnø, who in April 1997 invited us to Trondheim, Norway, where the central parts of this research were initiated.

## References

- [1] C. ARÉVALO AND G. SÖDERLIND, *Convergence of multistep discretizations of DAEs*, BIT Numerical Mathematics, 35 (1995), 143–168.
- [2] C. ARÉVALO, C. FÜHRER AND G. SÖDERLIND,  *$\beta$ -blocked multistep methods for Euler-Lagrange DAEs: Linear analysis*, ZAMM, 77 (1997), 609–617.
- [3] C. ARÉVALO, C. FÜHRER AND G. SÖDERLIND, *Stabilized multistep methods for index 2 Euler-Lagrange DAEs*, BIT Numerical Mathematics, 36 (1996), 1–13.
- [4] M. ARNOLD AND A. MURUA, *Non-stiff integrators for differential-algebraic systems of index 2*, Numerical Algorithms, 19 (1998), 25–41.
- [5] M. ARNOLD, *The stabilization of linear multistep methods for constrained mechanical systems*, Applied Numerical Mathematics, 28 (1998), 143–159.

- [6] M. ARNOLD, *Half-explicit Runge–Kutta methods with explicit stages for differential-algebraic systems of index 2*, BIT Numerical Mathematics, 38 (1998), 415–438.
- [7] M. ARNOLD, *Zur Theorie und zur numerischen Lösung von Anfangswertproblemen für differentiell-algebraische Systeme von höherem Index*, Fortschritt-Berichte VDI Reihe 20, Nr. 264. VDI-Verlag, Düsseldorf, 1998.
- [8] V. BRASEY AND E. HAIRER, *Half-explicit Runge–Kutta methods for differential-algebraic systems of index 2*, SINUM, 30 (1993), 538–552.
- [9] K.E. BRENNAN, S.L. CAMPBELL AND L.R. PETZOLD, *Numerical solution of initial-value problems in differential-algebraic equations*, SIAM, Philadelphia, 2nd edition, 1996.
- [10] Y. CAO AND Q. LI, *Highest order multistep formula for solving index 2 differential-algebraic equations*, BIT Numerical Mathematics, 38 (1998), 644–662.
- [11] E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Springer series in Computational Mathematics, 14, Springer–Verlag, New York, 2nd edition, 1996.
- [12] P. HENRICI, *Discrete Variable Methods in Ordinary Differential Equations II*, John Wiley, New York, 1962.
- [13] A. MURUA, *Partitioned half-explicit Runge–Kutta methods for differential algebraic systems of index 2*, Computing, 59 (1997) 43–61
- [14] A. OSTERMANN, *A half-explicit extrapolation method for differential-algebraic systems of index 3*, IMA J. Numer. Anal., 10 (1990), 171–180.
- [15] G. SÖDERLIND, *A multi-purpose system for the numerical integration of ODEs*, Appl. Math. Comp., 31 (1989), 346–360.