TECHNISCHE UNIVERSITÄT BERLIN

Efficient integration of matrix-valued non-stiff DAEs by half-explicit methods

Vu Hoang Linh Volker Mehrmann

Preprint 2011/16

Preprint-Reihe des Instituts für Mathematik
Technische Universität Berlin
http://www.math.tu-berlin.de/preprints

Preprint 2011/16 August 2011

Efficient integration of matrix-valued non-stiff DAEs by half-explicit methods

Vu Hoang Linh [*]   Volker Mehrmann [†]

August 29, 2011

Abstract

Numerical integration methods for nonlinear differential-algebraic equations (DAEs) in strangeness-free form are studied. In particular, half-explicit methods based on popular explicit methods like one-leg methods, linear multi-step methods, and Runge-Kutta methods are proposed and analyzed. Compared with well-known implicit methods for DAEs, these half-explicit methods demonstrate their efficiency particularly for a special class of semi-linear matrix DAEs which arise in the numerical computation of spectral intervals for DAEs. Numerical experiments illustrate the theoretical results.

Keywords: differential-algebraic equation, strangeness index, half-explicit methods, one-leg methods, linear multi-step methods, Runge-Kutta methods, spectral intervals.

AMS(MOS) subject classification: 65L07, 65L80

1 Introduction

Differential-algebraic equations are an important and convenient modeling concept in many different application areas such as multibody mechanics, circuit design, optimal control, chemical reactions, fluid dynamics, etc., see [4, 6, 9, 11, 12, 19, 20] and the references therein. In this work, we discuss efficient numerical integration

[*] Faculty of Mathematics, Mechanics and Informatics, Vietnam National University, 334, Nguyen Trai Str., Thanh Xuan, Hanoi, Vietnam. This work was supported by *Alexander von Humboldt Foundation*.

[†] Institut für Mathematik, MA 4-5, Technische Universität Berlin, D-10623 Berlin, Fed. Rep. Germany. Supported by *Deutsche Forschungsgemeinschaft*, through Collaborative Research Center 910, Control of self-organizing nonlinear systems.

methods for initial value problems associated with differential-algebraic equations (DAEs) of the form

$$\begin{array}{rcl} f(t,x(t),\dot{x}(t)) & = & 0 \\ g(t,x(t)) & = & 0, \end{array} \tag{1}$$

on an interval $\mathbb{I} = [t_0, t_f]$, together with an initial condition $x(t_0) = x_0$. Here we assume that $f = f(\cdot,\cdot,\cdot) : \mathbb{I} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^d$ and $g = g(\cdot,\cdot) : \mathbb{I} \times \mathbb{R}^n \to \mathbb{R}^a$, where $n = d + a$, are sufficiently smooth functions with bounded partial derivatives. Furthermore, we assume that (1) is strangeness-free, see [12, Definition 4.4], which means that the state x can be reordered and partitioned as $x = [x_1^T, x_2^T]^T$, where $x_1 : \mathbb{I} \times \mathbb{R}^d$, $x_2 : \mathbb{I} \times \mathbb{R}^a$ and the Jacobian g_{x_2} of g with respect to the variables x_2 is invertible in the neighborhood of the solution. Using the implicit function theorem, it has been shown in [12] that (1) can be locally transformed to a system of the form

$$\dot{x}_1 = \mathscr{L}(t, x_1), \quad x_2 = \mathscr{R}(t, x_1). \tag{2}$$

Strangeness-free DAEs of the from (1) have differentiation index one and they arise either directly in applications such as e. g., in circuit simulation or (under some weak assumptions) from the reduction process described in [12] applied to general implicit nonlinear DAEs

$$G(t, x, \dot{x}) = 0, \quad t \in \mathbb{I}. \tag{3}$$

Numerical methods for (1) or (3) are well analyzed in [4, 8, 9, 11, 19, 12] and several software packages for DAEs are available, see [12, Chapter 8]. In particular, it has been shown, see [12], that for strangeness-free DAEs of the form (1), well-known implicit methods like Runge-Kutta collocation methods and BDF methods are convergent of the same order as in the case of ordinary differential equations (ODEs).

In this paper we study half-explicit methods (HEMs). Such methods been suggested in [1, 2, 3, 11, 18] for the efficient integration of semi-explicit DAEs $\dot{x} = f(t,x,y),\ 0 = g(x,y)$ of differentiation index less than or equal to two. By exploiting the fact that the differential and the algebraic equations are separated, one applies an explicit integration scheme to the differential part and an implicit scheme (even simply the implicit Euler scheme) to the algebraic part. In every integration step this combination yields an algebraic system which uniquely determines the numerical solution. In general, the complexity of such methods is smaller than that of fully implicit schemes and the implementation is less complicated as well.

Here we will analyze the use of half-explicit methods applied to (1). Our main motivation arises from a special class of semi-linear matrix DAEs of the form

$$\begin{array}{rcl} E_1(t)\dot{X}(t) & = & F(t, X(t)), \\ 0 & = & A_2(t)X(t), \end{array} \tag{4}$$

where $E_1 : \mathbb{I} \to \mathbb{R}^{d\times n}$, $A_2 : \mathbb{I} \to \mathbb{R}^{a\times n}$ are continuous matrix valued functions, the unknown $X : \mathbb{I} \to \mathbb{R}^{n\times p}$ $(1 \le p \le d)$ and $F : \mathbb{I} \times \mathbb{R}^{n\times p} \to \mathbb{R}^{d\times p}$ are (nonlinear) matrix valued functions as well. The DAE (4) is strangeness-free if and only if the matrix $\bar{E}(t) = \begin{bmatrix} E_1(t) \\ A_2(t) \end{bmatrix}$ is invertible for all $t \in \mathbb{I}$.

Matrix DAEs of the form (4) arise in the stability analysis of DAEs via the numerical approximation of Lyapunov or Sacker-Sell spectral intervals by methods as developed recently in [14, 15]. In this application one has to solve DAEs of the form (4) on a very long interval $[0,t_f]$ with $t_f = O(10^3) - O(10^6)$. Furthermore, the exact solution satisfies some orthogonality condition in addition to the algebraic constraint explicitly given in (4). In order to approximate the spectral quantities accurately, the numerical solution must satisfy these conditions within machine precision [15]. Solving (4) by a well-known implicit scheme like BDF or Runge-Kutta methods presents a real challenge because in every step one has to solve a nonlinear matrix equation instead of the usual vector equation, and if one uses Newton's method, then the Jacobian of the vectorized matrix function G with respect to the components of X must be (approximately) available. In general, unfortunately, the (numerical) approximation of this Jacobian is very complicated and costly, since in this concrete problem no explicit formula of F is available. If a good approximation to the Jacobian is not available, then a slow fix-point iteration must be used instead.

We will show that, by using half-explicit methods, these challenges can be mastered, since only the solution of linear matrix equations in every time step is required, which can be solved efficiently with efficient methods from numerical linear algebra [7].

The outline of the paper is as follows. In the following section, we propose half-explicit one-leg methods and analyze their convergence. Sections 3 and 4 contain the realization and the analysis of half-explicit variants of linear multi-step methods and Runge-Kutta methods, respectively. Some numerical experiments illustrate the analysis in Section 5. We finish the paper with some conclusions.

Throughout the paper, we always assume that the initial value problem for (1) has a unique solution $x^* \in C^1(\mathbb{I}, \mathbb{R}^n)$. Linearizing (1) along x^* yields a linear DAE with coefficient functions

$$E(t) = \begin{bmatrix} E_1(t) \\ 0 \end{bmatrix} = \begin{bmatrix} f_{\dot{x}}(t, x^*, \dot{x}^*) \\ 0 \end{bmatrix}, \quad A(t) = \begin{bmatrix} A_1(t) \\ A_2(t) \end{bmatrix} = \begin{bmatrix} f_x(t, x^*, \dot{x}^*) \\ g_x(t, x^*) \end{bmatrix}. \tag{5}$$

We will frequently use this linearization in the analysis of the numerical methods presented in this paper, for consistency, stability and convergence, see [10] or [12] in the DAE framework.

2 Half-explicit one-leg methods for strangeness-free DAEs

In this section we discuss half-explicit one-leg methods which are special multistep methods. At the time $t = t_n$, we use k previous approximations $x_{n-1}, \dots, x_{n-k}$ for the computation of the approximation x_n to the solution value $x(t_n)$. Given real parameters α_j, β_j for $j = 0, 1, \dots, k$, $\alpha_0 \neq 0$, a one-leg method for the numerical solution of an initial value problem associated with the ODE

$$\dot{x} = f(t, x) \tag{6}$$

is given by

$$\sum_{j=0}^{k} \alpha_j x_{n-j} = hf\left(\sum_{j=0}^{k} \beta_j t_{n-j}, \sum_{j=0}^{k} \beta_j x_{n-j}\right). \tag{7}$$

Here, if $\beta_0 = 0$, then we have an explicit method, otherwise an implicit method. Only one function evaluation of f is used per step, this is the reason for the name *one-leg method*.

In order to have consistency for the scheme (7), we assume as in [17] that

$$\sum_{j=0}^{k} \alpha_j = 0,\ \sum_{j=0}^{k} \alpha_j \left(\sum_{l=1}^{k} l\beta_l - j\right) = 1,\ \sum_{j=1}^{k} \beta_j = 1.$$

Note that the last identity can always be achieved by a proper scaling of the β_i. The scheme (7) is *stable* if the associated characteristic polynomial

$$\rho(\lambda) = \sum_{j=0}^{k} \alpha_j \lambda^{n-j} \tag{8}$$

is stable, i. e., all the roots of $\rho(\lambda)$ lie in the closed unit disk and the roots of modulus one are simple. Then stability and consistency of order $p \geq 1$ implies the convergence of order p, see [12].

The parameter set of a one-leg method can be adopted from that of linear multistep methods such as Euler methods, Adams methods, or BDF (backward differentiation formula) methods. The analysis of explicit one-leg methods applied to ODEs is given e. g. in [21, 22]. For stiff ODEs and DAEs, however, one has to use implicit one-leg methods such as the implicit midpoint rule and BDF methods, see e. g. [4, 11, 12, 16, 17].

Here we adapt explicit one-leg methods in order to solve the strangeness-free DAE (1). For simplicity, in the analysis we assume that the mesh is uniform, i. e., that we have constant step-size. The analysis is easily extendable to the case of variable step-sizes.

If we apply an explicit one-leg discretization scheme to the differential part and evaluate the algebraic equation at $t = t_n$, then we have to solve the nonlinear system $F(x_n) = 0$ given by the equations

$$\begin{array}{lr} \text{(a)} & f\left(\sum_{j=1}^{k} \beta_j t_{n-j}, \sum_{j=1}^{k} \beta_j x_{n-j}, \frac{1}{h}\sum_{j=0}^{k} \alpha_j x_{n-j}\right) = 0, \\ \text{(b)} & g(t_n, x_n) = 0 \end{array} \tag{9}$$

for x_n. The Jacobian matrix at $t = t_n$ is

$$\frac{\partial F}{\partial x_n} = \left[\begin{array}{c} \frac{1}{h}\frac{\partial f}{\partial \dot{x}}|_{t=\bar{t}_n} \\ \frac{\partial g}{\partial x}|_{t=t_n} \end{array} \right],$$

where $\bar{t}_n = \sum_{j=1}^{k} \beta_j t_{n-j}$ is usually different from t_n. Note that, due to the consistency of the method, $\bar{t}_n$ remains close to t_n for sufficiently small h. Since the system is strangeness-free, the Jacobian matrix is boundedly invertible for sufficiently small h if the second block-row is Lipschitz-continuous with respect to t. Then the system (9) has a unique solution x_n which can be approximated by Newton's method see e. g. [5].

Note that unlike in the case of implicit methods, when we use the scheme (9), then the evaluation of $\partial f/\partial x$ at each step is avoided. Hence, if f and g are linear functions in $\dot{x}$ and x, respectively, which is the case for the matrix DAE (4), then (9) is a linear system for x_n.

For (4), we obtain

$$\begin{array}{rcl} E_1(\bar{t}_n)\frac{1}{h}\sum_{j=0}^{k} \alpha_j x_{n-j} & = & F\left(\bar{t}_n, \sum_{j=1}^{k} \beta_j x_{n-j}\right), \\ A_2(t_n)x_n & = & 0, \end{array}$$

which we write as the linear system

$$\alpha_0 \left[\begin{array}{c} E_1(\bar{t}_n) \\ A_2(t_n) \end{array} \right] x_n = \left[\begin{array}{c} -E_1(\bar{t}_n)\sum_{j=1}^{k} \alpha_j x_{n-j} + hF\left(\bar{t}_n, \sum_{j=1}^{k} \beta_j x_{n-j}\right) \\ 0 \end{array} \right]. \tag{10}$$

If one uses a direct solution method such as Gaussian elimination, then in each step $t = t_n$, only one LU factorization is needed to solve the linear matrix equation (10) instead of using Newton's method for a nonlinear system of essentially squared dimension.

For the analysis of the method we assume that the initial value problem for (1) has a unique solution $x^*(t)$ which is sufficiently smooth and the derivatives of x^* are bounded on $\mathbb{I}$. Furthermore, we assume that $g_x(t,x(t))$ is Lipschitz continuous with respect to t in a neighborhood of $(t,x^*(t))$, $t\in\mathbb{I}$. This is automatically satisfied if $\frac{d}{dt}g_x(t,x^*(t))$ is bounded on $\mathbb{I}$. In the following, we prove that the one-leg method (9) applied to (1) is convergent of order p provided that it is of order p and stable in the case of ODEs.

Theorem 1 *Suppose that the explicit one-leg method (7) as applied to ODEs (6) is convergent of order $p\geq 1$ (with starting values that are correct of order $\mathscr{O}(h^p)$). Then, the half-explicit scheme (9) as applied to DAEs (1) is convergent of order p as well, provided that the initial values $x_0,\ldots,x_{k-1}$ are consistent.*

Proof. We use the same strategy as in the proof for the convergence of BDF methods in [12]. Using the representation (2), we can split every x_i into $(x_{i,1},x_{i,2})$ and use

$$x_{i,2}=\mathscr{R}(t_i,x_{i,1}). \tag{11}$$

Introducing the linear operators L and D_h via

$$Lx_n=\sum_{j=1}^{k}\beta_j x_{n-j},\, D_h x_n=\frac{1}{h}\sum_{j=0}^{k}\alpha_j x_{n-j},$$

then, $\bar{t}_n=Lt_n$. Inserting, for $i=n,n-1,\ldots,n-k$, (11) into (9a), we obtain the system

$$f(\bar{t}_n,Lx_{n,1},L\mathscr{R}(t_n,x_{n,1}),D_h x_{n,1},D_h\mathscr{R}(t_n,x_{n,1}))=0. \tag{12}$$

We first show that close to the actual solution $x_1^*(t_n)$ of the implicit ODE

$$f(t,x_1,\mathscr{R}(t,x_1),\dot{x}_1,\mathscr{R}_t(t,x_1)+\mathscr{R}_{x_1}(t,x_1)\dot{x}_1)=0 \tag{13}$$

the system (12) uniquely determines the numerical approximation $x_{n,1}$.

To see this, we discretize the linearization (5) and apply the one-leg method to the problem

$$E(t)\dot{x}-A(t)x=E(t)\dot{x}^*(t)-A(t)x^*(t) \tag{14}$$

This gives the system

$$\begin{array}{rcl} E_1(\bar{t}_n)\frac{1}{h}\sum_{j=0}^{k}\alpha_j x_{n-j}-A_1(\bar{t}_n)\sum_{j=1}^{k}\beta_j x_{n-j} & = & E_1(\bar{t}_n)\dot{x}^*(\bar{t}_n)-A_1(\bar{t}_n)x^*(\bar{t}_n), \\ -A_2(t_n)x_n & = & -A_2(t_n)x^*(t_n). \end{array}$$

Rearranging the terms, we obtain the linear system

$$\begin{bmatrix} \alpha_0 E_1(\bar{t}_n) \\ A_2(t_n) \end{bmatrix} x_n=\begin{bmatrix} -E_1(\bar{t}_n)\sum_{j=1}^{k}\alpha_j x_{n-j}+hA_1(\bar{t}_n)\sum_{j=1}^{k}\beta_j x_{n-j}+hq_1(\bar{t}_n) \\ q_2(t_n) \end{bmatrix} \tag{15}$$

with $q_1(\bar{t}_n) := E_1(\bar{t}_n)\dot{x}^*(\bar{t}_n) - A_1(\bar{t}_n)x^*(\bar{t}_n)$, $q_2(t_n) := A_2(t_n)x^*(t_n)$.

Using an appropriate splitting and a linearization of the representation (2) yields an equation

$$x_{n,2} = R(t_n)x_{n,1} + s(t_n). \tag{16}$$

Since we have assumed that the starting values satisfy $x_{n-j} = x^*(t_{n-j}) + \mathscr{O}(h^p)$, $j = 1,\ldots,k$, we have

$$\begin{aligned} \sum_{j=1}^{k} \alpha_j x_{n-j} &= h\dot{x}^*(\bar{t}_n) - \alpha_0 x^*(t_n) + \mathscr{O}(h^p), \\ \sum_{j=1}^{k} \beta_j x_{n-j} &= x^*(\bar{t}_n) + \mathscr{O}(h^p). \end{aligned}$$

Inserting this into (15) and eliminating (16) then gives

$$\alpha_0 \left[E_{11}(\bar{t}_n) + E_{12}(\bar{t}_n)R(t_n)\right] x_{n,1} = \alpha_0 \left[E_{11}(\bar{t}_n) + E_{12}(\bar{t}_n)R(t_n)\right] x_1^*(t_n) + \mathscr{O}(h^p).$$

Since the problem is strangeness-free, it follows for sufficiently small h that the coefficient matrix $\alpha_0 \left[E_{11}(\bar{t}_n) + E_{12}(\bar{t}_n)R(t_n)\right]$ is bounded and boundedly invertible which implies that $x_1^*(t_n) - x_{n,1} = \mathscr{O}(h^p)$ and hence $x^*(t_n) - x_n = \mathscr{O}(h^p)$.

To show that the nonlinear system (12) fixes a numerical solution $x_{n,1}$ at least for sufficiently h, we apply the Newton-like method described in [12, Chapter 5] and the same argument used there. The iteration takes the form

$$\begin{aligned} &\tfrac{\alpha_0}{h} \left[E_{11}(\bar{t}_n) + E_{12}(\bar{t}_n)R(t_n)\right] (x_{n,1}^{m+1} - x_{n,1}^{m}) \\ &\quad = -f\Big(\bar{t}_n, Lx_{n,1}, L\mathscr{R}(t_n, x_{n,1}), \tfrac{1}{h}(\alpha_0 x_{n,1}^m + \textstyle\sum_{j=1}^k \alpha_j x_{n-j,1}), \\ &\quad \tfrac{1}{h}(\alpha_0 \mathscr{R}(t_n, x_{n,1}^m) + \textstyle\sum_{j=1}^k \alpha_j \mathscr{R}(t_{n-j}, x_{n-j,1}))\Big). \end{aligned} \tag{17}$$

Using as starting values the just constructed solution $x_{n,1}$ of (15), which is denoted by $x_{n,1}^0$, the same analysis as in [12, Chapter 5] shows that the iterates $x_{n,1}^m$ generated by (17) converge to a solution $x_{n,1}^*$ of (12). Furthermore, we have that $x_1^*(t_n) - x_{n,1}^* = \mathscr{O}(h^p)$, and setting $x_{n,2}^* = \mathscr{R}(t_n, x_{n,1}^*)$, one obtains a solution x_n^* of (9) satisfying $x^*(t_n) - x_n^* = \mathscr{O}(h^p)$.

In this way, we have shown that (12) locally defines a numerical solution $x_{n,1}^*$, provided that the iterates x_{n-j}, $j = 1,\ldots,k$, are close to the solution. Writing (12) in a simplified form as

$$\tilde{f}(t_n, x_{n,1}, \ldots, x_{n-k,1}; h) = 0, \tag{18}$$

this equation is locally solved via

$$x_{n,1} = \mathscr{S}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}; h). \tag{19}$$

and hence

$$\tilde{f}(t_n, \mathscr{S}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}; h), x_{n-1,1}, \ldots, x_{n-k,1}; h) \equiv 0. \tag{20}$$

As next step, we show that (19) indeed gives a convergent numerical method for the determination of the numerical solution x_1^* of (12). To this end, we define

$$\mathscr{X}_n = \begin{bmatrix} x_{n-1,1} \\ x_{n-2,1} \\ \vdots \\ x_{n-k,1} \end{bmatrix}, \quad \mathscr{X}(t_n) = \begin{bmatrix} x_1^*(t_{n-1}) \\ x_1^*(t_{n-2}) \\ \vdots \\ x_1^*(t_{n-k}) \end{bmatrix},$$

together with

$$\mathscr{F}(t_n, \mathscr{X}_n; h) = \begin{bmatrix} \mathscr{S}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}; h) \\ x_{n-1,1} \\ \vdots \\ x_{n-k+1,1} \end{bmatrix}.$$

For consistency, we must study $\mathscr{X}(t_{n+1}) - \mathscr{F}(t_n, \mathscr{X}(t_n); h)$ and, therefore, consider

$$x_1^*(t_n) - \mathscr{S}(t_n, x_1^*(t_{n-1}), \ldots, x_1^*(t_{n-k}); h).$$

Starting from

$$\begin{array}{l} f\Big(\bar{t}_n, \sum_{j=1}^k \beta_j x_1^*(t_{n-j}), \sum_{j=1}^k \beta_j \mathscr{R}(t_{n-j}, x_1^*(t_{n-j})), \\ \frac{1}{h}\sum_{j=0}^k \alpha_{n-j} x_1^*(t_{n-j}), \frac{1}{h}\sum_{j=0}^k \alpha_{n-j} \mathscr{R}(t_{n-j}, x_1^*(t_{n-j}))\Big) \\ = f\,(\bar{t}_n, x_1^*(\bar{t}_n) + \mathscr{O}(h^p), \mathscr{R}(\bar{t}_n, x_1^*(\bar{t}_n)) + \mathscr{O}(h^p), \dot{x}_1^*(\bar{t}_n) + \mathscr{O}(h^p), \\ \mathscr{R}_t(\bar{t}_n, x_1^*(\bar{t}_n)) + \mathscr{R}_{x_1}(\bar{t}_n, x_1^*(\bar{t}_n))\dot{x}_1^*(\bar{t}_n) + \mathscr{O}(h^p)\big) \\ = \mathscr{O}(h^p), \end{array}$$

we consider (18) in the perturbed form

$$\tilde{f}(t_n, x_{n,1}, \ldots, x_{n-k,1}; h) = \varepsilon.$$

For sufficiently small ε, this relation is still locally solvable for $x_{n,1}$ according to

$$x_{n,1} = \tilde{\mathscr{S}}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}; h, \varepsilon)$$

such that

$$\tilde{f}(t_n, \tilde{\mathscr{S}}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}; h, \varepsilon), x_{n-1,1}, \ldots, x_{n-k,1}; h, \varepsilon) \equiv 0. \tag{21}$$

Hence, we obtain

$$x_1^*(t_n) = \tilde{\mathscr{S}}(t_n, x_1^*(t_{n-1}), \dots, x_1^*(t_{n-k}); h, \varepsilon),$$

with $\varepsilon = \mathscr{O}(h^p)$. It follows that

$$\begin{aligned} & x_1^*(t_n) - \mathscr{S}(t_n, x_1^*(t_{n-1}), \dots, x_1^*(t_{n-k}); h) \\ & \quad = \tilde{\mathscr{S}}(t_n, x_1^*(t_{n-1}), \dots, x_1^*(t_{n-k}); h, \varepsilon) - \tilde{\mathscr{S}}(t_n, x_1^*(t_{n-1}), \dots, x_1^*(t_{n-k}); h, 0). \end{aligned}$$

We now estimate the derivative of $\tilde{\mathscr{S}_\varepsilon}$ of $\tilde{\mathscr{S}}$ with respect to ε. Differentiating (21) and using (12), we get

$$\frac{\alpha_0}{h}\left(f_{\dot{x}_1} + f_{\dot{x}_2}\mathscr{R}_{x_1}\right)\tilde{\mathscr{S}_\varepsilon} = I_d.$$

Hence, we get $\tilde{\mathscr{S}_\varepsilon} = \mathscr{O}(h)$ and $\tilde{\mathscr{S}}$ is Lipschitz continuous with respect to ε with Lipschitz constant $L_\varepsilon = \mathscr{O}(h)$. Using these results, we obtain the bound

$$\left\| x_1^*(t_n) - \mathscr{S}(t_n, x_1^*(t_{n-1}), \dots, x_1^*(t_{n-k}); h) \right\| \le L_\varepsilon \varepsilon = \mathscr{O}(h^{p+1}),$$

which means that the discretization method is consistent of order p.

To prove stability, we must study $\mathscr{F}(t_n, \mathscr{X}(t_n); h) - \mathscr{F}(t_n, \mathscr{X}_n; h)$. We again consider the first block

$$\mathscr{S}(t_n, x_1^*(t_{n-1}), \dots, x_1^*(t_{n-k}); h) - \mathscr{S}(t_n, x_{n-1,1}, \dots, x_{n-k,1}; h)$$

and determine the derivatives $\mathscr{S}_{x_{n-j,1}}$ of $\mathscr{S}$ with respect to $x_{n-j,1}$ for $j = 1, \dots, k$. Differentiating (20) and using (12), we get

$$\frac{\alpha_0}{h}\left(f_{\dot{x}_1} + f_{\dot{x}_2}\mathscr{R}_{x_1}\right)\mathscr{S}_{x_{n-j,1}} + \frac{\alpha_j}{h}\left(f_{\dot{x}_1} + f_{\dot{x}_2}\mathscr{R}_{x_1}\right) + \beta_j\left(f_{x_1} + f_{x_2}\mathscr{R}_{x_1}\right) = 0.$$

Since we have $\mathscr{S}_{x_{n-j,1}} = -\frac{\alpha_j}{\alpha_0} I_d + \mathscr{O}(h)$, a simple calculation shows that

$$\begin{aligned} & \mathscr{S}(t_n, x_1^*(t_{n-1}), \dots, x_1^*(t_{n-k}); h) - \mathscr{S}(t_n, x_{n-1,1}, \dots, x_{n-k,1}; h) \\ & \quad = \sum_{j=1}^{k}(-\frac{\alpha_j}{\alpha_0} I_d + \mathscr{O}(h))(x_1^*(t_{n-j}) - x_{n-j,1}). \end{aligned}$$

Thus

$$\mathscr{F}(t_n, \mathscr{X}(t_n); h) - \mathscr{F}(t_n, \mathscr{X}_n; h) = \begin{bmatrix} \sum_{j=1}^{k}(-\frac{\alpha_j}{\alpha_0} I_d + \mathscr{O}(h))(x_1^*(t_{n-j}) - x_{n-j,1}) \\ x_1^*(t_{n-1}) - x_{n-1,1} \\ \vdots \\ x_1^*(t_{n-k+1}) - x_{n-k+1,1} \end{bmatrix},$$

and we obtain the estimate

$$\|\mathscr{F}(t_n, \mathscr{X}(t_n); h) - \mathscr{F}(t_n, \mathscr{X}_n; h)\| \le (\left\|\mathscr{C}_\alpha \otimes I_d\right\| + Kh) \|\mathscr{X}(t_n) - \mathscr{X}_n\|,$$

where

$$\mathscr{C}_\alpha = \begin{bmatrix} -\frac{\alpha_1}{\alpha_0} & \cdots & -\frac{\alpha_{k-1}}{\alpha_0} & -\frac{\alpha_k}{\alpha_0} \\ 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 1 & 0 \end{bmatrix}.$$

If the underlying one-leg method is stable, then there exists a vector norm such that with the associated matrix norm, the inequality $\left\|\mathscr{C}_\alpha \otimes I_d\right\| \le 1$is satisfied. Hence, the discretization method (9) is stable as well. We conclude that the numerical solution $x_{n,1}$ by (12) converges to the exact solution x_1^* with order p. The convergence of the second component $x_{n,2}$ follows, since

$$\left\|x_2^*(t_n) - x_{n,2}\right\| = \left\|\mathscr{R}(t_n, x_1^*(t_n)) - \mathscr{R}(t_n, x_{n,1})\right\| \le L \left\|x_1^*(t_n) - x_{n,1}\right\| = \mathscr{O}(h^p)$$

for all $n = 0, 1, \ldots, N$, where L denotes the Lipschitz constant of $\mathscr{R}$ with respect to the second argument. This finishes the proof. □

Note, that to get accurate starting values $x_1, \ldots, x_{k-1}$, as in the ODE case, one may apply half-explicit j-step methods recursively with $j = 1, \ldots, k-1$, respectively.

Example 2 The simplest example of a one-leg method is the explicit Euler method with $\alpha_0 = 1, \alpha_1 = -1$ and $\beta_1 = 1$, which is of order 1. If we apply the resulting half-explicit method to the test DAE [13]

$$\begin{bmatrix} 1 & -\omega t \\ 0 & 0 \end{bmatrix} \dot{x} = \begin{bmatrix} \lambda & \omega(1 - \lambda t) \\ -1 & 1 + \omega t \end{bmatrix} x, \tag{22}$$

then with stepsize h we obtain the generalized stability function

$$R(z, w) = \frac{1 + z + w}{1 + w},$$

where $z = \lambda h$ and $w = \omega h$. Comparing this with the stability function of the implicit Euler method, see [13],

$$R(z, w) = \frac{1 - w}{1 - z - w},$$

we may conclude that the half-explicit method is feasible for non-stiff DAEs of the form (1), i. e., DAEs where the underlying ODE is non-stiff. For the test equation (22), this means that λ has negative, but not too large real part.

Example 3 A family of second order two-step methods discussed in [21] is defined by the coefficients

$$\alpha_0 = \frac{1}{\xi}, \alpha_1 = 1 - \frac{2}{\xi}, \alpha_2 = \frac{1}{\xi} - 1, \beta_1 = \frac{1}{2} + \frac{1}{\xi}, \beta_2 = \frac{1}{2} - \frac{1}{\xi},$$

where ξ is a parameter, $0 < \xi \leq 2$. If $\xi = 1$, then we have the one-leg variant of the well-known two-step Adams-Bashforth scheme.

3 Half-explicit linear multi-step methods

In this section we consider explicit linear multi-step methods applied to (6) as basis for the construction of half-explicit methods. These take the form

$$\sum_{j=0}^{k} \alpha_j x_{n-j} = h \sum_{j=1}^{k} \beta_j f_{n-j}, \quad f_{n-j} = f(t_{n-j}, x_{n-j}). \tag{23}$$

For simplicity, we assume that $\alpha_0 = 1$ and $\beta_1 \neq 0$ (if β_1 is not zero, then we use the first non-zero parameter among the β_i instead). To construct a half-explicit method for (1), the only question is how to implement this method for the differential part. Using the idea introduced for implicit multi-step methods for DAEs in [16], we proceed as follows. Let x_n and w_n be approximations of the exact solution $x(t_n)$ and its derivative $w(t_n) := \dot{x}(t_n)$, respectively. Now, suppose that we have already determined $x_{n-k}, \ldots, x_{n-1}$ and $w_{n-k}, \ldots, w_{n-2}$. The scheme (23) is equivalent to $\sum_{j=0}^{k} \alpha_j x_{n-j} = h \sum_{j=1}^{k} \beta_j w_{n-j}$, from which we get

$$w_{n-1} = \frac{1}{\beta_1} \left(\frac{1}{h} \sum_{j=0}^{k} \alpha_j x_{n-j} - \sum_{j=2}^{k} \beta_j w_{n-j} \right). \tag{24}$$

Using this approximate formula for w_{n-1}, we approximate the differential part at $t = t_{n-1}$ and the algebraic part at $t = t_n$. This results in a nonlinear system for x_n given by

$$\begin{aligned} f(t_{n-1}, x_{n-1}, w_{n-1}) &= 0 \\ g(t_n, x_n) &= 0, \end{aligned}$$

or equivalently

$$\begin{array}{ll} \text{(a)} & f\left(t_{n-1}, x_{n-1}, \frac{1}{\beta_1}\left(\frac{1}{h}\sum_{j=0}^{k} \alpha_j x_{n-j} - \sum_{j=2}^{k} \beta_j w_{n-j}\right)\right) = 0 \\ \text{(b)} & g(t_n, x_n) = 0. \end{array} \tag{25}$$

This system has a unique solution x_n for sufficiently small h which can be obtained by Newton's method. Applying (25) to (4), we obtain the linear system

$$\begin{bmatrix} E_1(t_{n-1}) \\ A_2(t_n) \end{bmatrix} x_n = \begin{bmatrix} -E_1(t_{n-1})\left(\sum_{j=1}^{k}\alpha_j x_{n-j} - h\sum_{j=2}^{k}\beta_j w_{n-j}\right) + h\beta_1 F(t_{n-1}, x_{n-1}) \\ 0 \end{bmatrix}. \tag{26}$$

So, similar as for the half-explicit one-leg methods, if we use a direct solver like Gaussian elimination, then we need only one LU factorization per step to solve the system (26) for x_n. The derivative approximation w_{n-1} that is needed for the next step is obtained by (24).

Besides the characteristic polynomial ρ defined in (8), let us introduce another characteristic polynomial

$$\sigma(\lambda) = \sum_{j=1}^{k} \beta_j \lambda^{n-j},$$

which is associated with the formula (24). We will see below that, to ensure the stability of the numerical scheme, this polynomial has to be stable as well. Moreover, no root of modulus is allowed to be a root of the first characteristic polynomial ρ. This guarantees that the product $\rho\sigma$ is stable.

We have the following convergence result for half-explicit linear multi-step methods.

Theorem 4 *Suppose that the explicit linear multi-step method (25) with $\beta_0 = 0$ applied to an ODE of the form (6) is convergent of order p, that the starting values are consistent and accurate of order p and that the product $\rho(\lambda)\sigma(\lambda)$ is stable. Then, the half-explicit scheme (25) applied to the DAE (1) is convergent of order p as well.*

Proof. We proceed in the same way as in the convergence analysis for half-explicit one-leg methods. All the arguments for proving the feasibility and the consistency are similar but with some slight differences due to the appearance of the derivative approximations w_{n-j}, $j = 2, \ldots, k$. We omit the details, but note that the derivative quantities are not split into two parts as the state values x_{n-j} and $x(t)$ are. Furthermore, since w_n is evaluated using (24), the order of accuracy of w_{n-1} is by one smaller than that of x_n. After eliminating the components $x_{n-j,2}$, $j \geq 0$ we obtain

$$x_{n,1} = \mathscr{S}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}, w_{n-2}, \ldots, w_{n-k}; h),$$

and

$$w_{n-1} = \mathscr{Q}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}, w_{n-2}, \ldots, w_{n-k}; h),$$

with solution operators $\mathscr{S}, \mathscr{Q}$.

We also define

$$\mathscr{X}_n = \begin{bmatrix} x_{n-1,1} \\ x_{n-2,1} \\ \vdots \\ x_{n-k,1} \\ w_{n-2} \\ \vdots \\ w_{n-k} \end{bmatrix}, \quad \mathscr{X}(t_n) = \begin{bmatrix} x_1^*(t_{n-1}) \\ x_1^*(t_{n-2}) \\ \vdots \\ x_1^*(t_{n-k}) \\ w^*(t_{n-2}) \\ \vdots \\ w^*(t_{n-k}) \end{bmatrix},$$

where w^* denotes the derivative of x^*, and

$$\mathscr{G}(t_n, \mathscr{X}_n; h) = \begin{bmatrix} \mathscr{S}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}; h) \\ x_{n-1,1} \\ \vdots \\ x_{n-k+1,1} \\ \mathscr{Q}(t_n, x_{n-1,1}, \ldots, x_{n-k,1}, w_{n-2}, \ldots, w_{n-k}; h) \\ w_{n-2} \\ \vdots \\ w_{n-k+1} \end{bmatrix}.$$

We focus on proving the stability of the scheme (25), (24). Elementary calculations show that the Jacobians satisfy

$$\mathscr{S}_{x_{n-j,1}} = -\frac{\alpha_j}{\alpha_0} I_d + \mathscr{O}(h),\ j \geq 1, \text{ and } \mathscr{S}_{w_{n-i}} = \mathscr{O}(h), i \geq 2.$$

Using (24), we then obtain

$$\mathscr{Q}_{x_{n-j,1}} = \mathscr{O}(1),\ j \geq 1, \text{ and } \mathscr{Q}_{w_{n-i}} = -\frac{\beta_i}{\beta_1} I_n + \mathscr{O}(h), i \geq 2,$$

and, by using the same argument as in the stability analysis for one-leg methods, we have

$$
\begin{aligned}
&\mathscr{G}(t_n, \mathscr{Z}(t_n); h) - \mathscr{G}(t_n, \mathscr{Z}_n; h) \\
&\quad = \begin{bmatrix}
\sum_{j=1}^{k}(-\frac{\alpha_j}{\alpha_0} I_d + \mathscr{O}(h))(x_1^*(t_{n-j}) - x_{n-j,1}) + \sum_{i=2}^{k} \mathscr{O}(h)(w^*(t_{n-i}) - w_{n-i}) \\
x_1^*(t_{n-1}) - x_{n-1,1} \\
\vdots \\
x_1^*(t_{n-k+1}) - x_{n-k+1,1} \\
\sum_{j=1}^{k} \mathscr{O}(1)(x_1^*(t_{n-j}) - x_{n-j,1}) + \sum_{i=2}^{k}(-\frac{\beta_j}{\beta_1} I_n + \mathscr{O}(h))(w^*(t_{n-i}) - w_{n-i,1}) \\
w^*(t_{n-2}) - w_{n-2} \\
\vdots \\
w^*(t_{n-k+1}) - w_{n-k+1}
\end{bmatrix}.
\end{aligned}
$$

Finally, we then obtain the estimate

$$
\|\mathscr{G}(t_n, \mathscr{Z}(t_n); h) - \mathscr{G}(t_n, \mathscr{Z}_n; h)\| \leq (\left\| \mathscr{C}_{\alpha\beta} \otimes I_{dn} \right\| + Kh) \, \|\mathscr{Z}(t_n) - \mathscr{Z}_n\|,
$$

where

$$
\mathscr{C}_{\alpha\beta} = \begin{bmatrix}
-\frac{\alpha_1}{\alpha_0} & \dots & -\frac{\alpha_{k-1}}{\alpha_0} & -\frac{\alpha_k}{\alpha_0} & 0 & \dots & 0 & 0 \\
1 & \ddots & 0 & 0 & 0 & \ddots & 0 & 0 \\
\vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & \dots & 1 & 0 & 0 & \ddots & 0 & 0 \\
\mathscr{O}(1) & \dots & \mathscr{O}(1) & \mathscr{O}(1) & -\frac{\beta_2}{\beta_1} & \dots & -\frac{\beta_{k-1}}{\beta_1} & -\frac{\beta_k}{\beta_1} \\
0 & \ddots & 0 & 0 & 1 & \ddots & 0 & 0 \\
\vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & \ddots & 0 & 0 & 0 & \dots & 1 & 0
\end{bmatrix}.
$$

The product $\rho\sigma$ is the characteristic polynomial associated with the matrix $\mathscr{C}_{\alpha\beta}$. If it is stable, then there exists a vector norm, so that with the associated matrix norm, the inequality $\left\| \mathscr{C}_{\alpha\beta} \otimes I_{dn} \right\| \leq 1$ holds. Hence, the discretization method (25) is stable and we conclude that the numerical solution $x_{n,1}$ converges to the exact solution x_1^* of order p and the second component $x_{n,2}$ converges of the same order. □

Table 1: Butcher tableau of explicit 2-stage Runge-Kutta method

0	0	0
c_2	a_{21}	0
	b_1	b_2

Example 5 *A family of second order two-step methods, discussed in [21], is defined by*

$$\alpha_0 = 1,\, \alpha_1 = \xi - 2,\, \alpha_2 = 1 - \xi, \beta_1 = \frac{\xi}{2} + 1, \beta_2 = \frac{\xi}{2} - 1,$$

where ξ is a parameter, $0 < \xi \leq 2$. It is easy to verify that for each $\xi \in (0,2]$, the method satisfies the conditions of Theorem 4 with convergence order $p = 2$.

4 Half-explicit Runge-Kutta methods

Given an explicit Runge-Kutta method, the corresponding half-explicit Runge-Kutta method can be constructed in the same way as in the case of linear multistep methods. For illustration, we consider a 2-stage explicit Runge-Kutta scheme given by Butcher tableau in Table 1.

Consider an interval $[t_{n-1}, t_n]$ and suppose that an approximation x_{n-1} to $x(t_{n-1})$ is given. Let $X_i \approx x(t_{n-1} + c_i h)$ be the stage approximation and let $K_i = \dot{X}_i$ be the approximations to the derivatives of X_i, $i = 1,2$. Then, the explicit Runge-Kutta scheme defined by Table 1 has the form

$$\begin{aligned} &\text{(a)} \qquad X_1 = x_{n-1}, \\ &\text{(b)} \qquad X_2 = x_{n-1} + h a_{21} K_1, \\ &\text{(c)} \qquad x_n = x_{n-1} + h(b_1 K_1 + b_2 K_2), \end{aligned} \tag{27}$$

with $c_1 = 0$ for explicit Runge-Kutta methods.

If we assume, furthermore, that

$$c_2 = a_{21},\, b_1 + b_2 = 1,\, c_2 b_2 = 1/2, \tag{28}$$

then it is well-known, see e. g. [10], that the explicit Runge-Kutta scheme (27) has convergence order $p = 2$. From (27(b)) and (27(c)), it follows that

$$K_1 = \frac{X_2 - x_{n-1}}{h a_{21}}, \quad K_2 = \frac{1}{b_2}\left[\frac{x_n - x_{n-1}}{h} - b_1 K_1\right].$$

To determine X_2 (and thus also K_1), we have to solve the system

$$f(t_{n-1},X_1,K_1)=0,\ g(t_{n-1}+c_2h,X_2)=0,$$

or equivalently

$$\begin{array}{lr} \text{(a)} & f(t_{n-1},X_1,\frac{X_2-x_{n-1}}{ha_{21}})=0, \\ \text{(b)} & g(t_{n-1}+c_2h,X_2)=0. \end{array} \tag{29}$$

Alternatively, one may use

$$g(t_{n-1}+c_2h,X_2)-g(t_{n-1},x_{n-1})=0 \tag{30}$$

instead of (29(b)).

Then, we obtain the next approximate x_n by solving the system

$$\begin{array}{rcl} f(t_{n-1}+c_2h,X_2,K_2) & = & 0, \\ g(t_n,x_n) & = & 0, \end{array}$$

or equivalently

$$\begin{array}{lr} \text{(a)} & f\left(t_{n-1}+c_2h,X_2,\frac{1}{b_2}\left[\frac{x_n-x_{n-1}}{h}-b_1K_1\right]\right)=0, \\ \text{(b)} & g(t_n,x_n)=0. \end{array} \tag{31}$$

Instead of (31(b)), we may also use

$$g(t_n,x_n)-g(t_{n-1},x_{n-1})=0. \tag{32}$$

Applying this procedure to (4), we have to solve consecutively two linear systems

$$\begin{array}{rcl} \left[\begin{array}{c} E_1(t_{n-1}) \\ A_2(t_{n-1}+c_2h) \end{array}\right] X_2 & = & \left[\begin{array}{c} E_1(t_{n-1})x_{n-1}+ha_{21}F(t_{n-1},x_{n-1}) \\ 0 \end{array}\right], \\ \left[\begin{array}{c} E_1(t_{n-1}+c_2h) \\ A_2(t_n) \end{array}\right] x_n & = & \left[\begin{array}{c} E_1(t_{n-1}+c_2h)\left[x_{n-1}+hb_1K_1\right]+hb_2F(t_{n-1}+c_2h,X_2) \\ 0 \end{array}\right]. \end{array}$$

For this method we have the following convergence result.

Theorem 6 *If the initial condition x_0 is consistent, then the half-explicit Runge-Kutta (HERK) method (31) (or alternatively using (32) for the algebraic equations) with order conditions (28) applied to (1) are convergent of order $p=2$.*

Proof. We first verify the feasibility of the schemes (29) and (31), i. e., that the nonlinear systems are uniquely solvable for X_2 and x_n, assuming that one is close to the actual solution of (1). This can be done without reducing the problem to the implicit ODE (13).

Consider first the system (29). Applying the same scheme (29) to the linear problem (14), we obtain the system

$$\begin{bmatrix} E_1(t_{n-1}) \\ A_2(t_{n-1}+c_2h) \end{bmatrix} X_2 = \begin{bmatrix} -E_1(t_{n-1})x_{n-1}+ha_{21}A_1(t_{n-1})x_{n-1}+ha_{21}q_1(t_{n-1}) \\ A_2(t_{n-1}+c_2h)x^*(t_{n-1}+c_2h) \end{bmatrix}, \tag{33}$$

where $q_1(t_{n-1}) = E_1(t_{n-1})\dot{x}^*(t_{n-1}) - A_1(t_{n-1})x^*(t_{n-1})$. Assuming that $x_{n-1} = x^*(t_{n-1}) + \mathscr{O}(h^2)$, it is easy to see that

$$-E_1(t_{n-1})x_{n-1}+ha_{21}A_1(t_{n-1})x_{n-1}+ha_{21}q_1(t_{n-1}) = E_1(t_{n-1})x^*(t_{n-1}+c_2h)+\mathscr{O}(h^2).$$

Keeping in mind that $c_2 = a_{21}$ due to (28), for sufficiently small h, the matrix

$$\begin{bmatrix} E_1(t_{n-1}) \\ A_2(t_{n-1}+c_2h) \end{bmatrix}$$

is boundedly invertible, which implies that the system (33) is uniquely solvable and $X_2 - x^*(t_{n-1}+c_2h) = \mathscr{O}(h^2)$. We use this value of X_2 as a starting value for the Newton-like iteration for solving (29), which takes the form

$$\begin{bmatrix} E_1(t_{n-1}) \\ A_2(t_{n-1}+c_2h) \end{bmatrix} (X_2^{m+1} - X_2^m) = -\begin{bmatrix} f(t_{n-1}, x_{n-1}, \frac{X_2^m - x_{n-1}}{a_{21}h} \\ g(t_{n-1}+c_2h, X_2^m) \end{bmatrix}. \tag{34}$$

By invoking [12, Theorem 5.7] once again, an analogous analysis shows that the iterates X_2^m generated by (34) converge to a solution X_2^*, which also fulfills $x^*(t_{n-1}+c_2h) - X_2^* = \mathscr{O}(h^2)$.

With the obtained X_2^*, repeating the above procedure for (31), we are also able to show that this nonlinear system has the unique solution x_n^* in the local sense, i. e., when we are close to the actual solution $x^*(t)$. Furthermore, the estimate $x^*(t_n) - x_n^* = \mathscr{O}(h^2)$ holds.

Next, we show that the discretization scheme has consistency order $p = 2$. The technique used in the proof of Theorem 1 can be used here as well. The key difference is that we must construct a finer estimate for the stage X_2 in order to obtain the desired bound for the local error. Furthermore, for the sake of simplicity, we avoid reducing (1) to the underlying ODE (13).

Let us denote by $X_2 = \mathscr{T}(t_n, x_{n-1}; h)$ the local solution of the system (29), which can be written in a simplified form as

$$\tilde{f}(t_n, X_2, x_{n-1}; h) = 0, \quad \tilde{g}(t_n, X_2; h) = 0, \tag{35}$$

or equivalently,

$$\tilde{f}(t_n, \mathscr{T}(t_n, x_{n-1}; h), x_{n-1}; h) = 0, \quad \tilde{g}(t_n, \mathscr{T}(t_n, x_{n-1}; h); h) = 0. \tag{36}$$

We need to estimate the stage local error $d_{X_2} = x^*(t_{n-1} + c_2 h) - \mathscr{T}(t_n, x^*(t_{n-1}); h)$. By Taylor expansion, we have

$$\begin{array}{ll} \text{(a)} & f\left(t_{n-1}, x^*(t_{n-1}), \frac{1}{ha_{21}}(x^*(t_{n-1} + c_2 h) - x^*(t_{n-1}))\right) = f_{\dot{x}} \frac{c_2 h}{2} \ddot{x}^*(t_{n-1}) + \mathscr{O}(h^2), \\ \text{(b)} & g\left(t_{x_{n-1}} + c_2 h, x^*(t_{n-1} + c_2 h)\right) = 0, \end{array} \tag{37}$$

where $f_{\dot{x}}$ is evaluated at $(t_{n-1}, x^*(t_{n-1}), \dot{x}^*(t_{n-1}))$. Then we can consider (35) in the perturbed form

$$\tilde{f}(t_n, X_2, x_{n-1}; h) = \varepsilon, \quad \tilde{g}(t_n, X_2; h) = 0.$$

where $\varepsilon = f_{\dot{x}} \frac{c_2 h}{2} \ddot{x}^*(t_{n-1}) + \mathscr{O}(h^2) = \mathscr{O}(h)$. For sufficiently small ε, this system is still locally solvable. Comparing (37) with

$$\begin{array}{rcl} f\left(t_{n-1}, x^*(t_{n-1}), \dfrac{1}{ha_{21}}(\mathscr{T}(t_n, x^*(t_{n-1}); h) - x^*(t_{n-1}))\right) & = & 0, \\ g\left(t_{n-1} + c_2 h, \mathscr{T}(t_n, x^*(t_{n-1}); h)\right) & = & 0, \end{array}$$

and application of the mean value theorem yields the linear system

$$\left[\begin{array}{c} \frac{1}{ha_{21}} f_{\dot{x}} \\ g_x \end{array} \right] d_{X_2} = \left[\begin{array}{c} f_{\dot{x}}(t_{n-1}, x^*(t_{n-1}), \dot{x}^*(t_{n-1})) \frac{c_2 h}{2} \ddot{x}^*(t_{n-1}) + \mathscr{O}(h^2) \\ 0 \end{array} \right],$$

where the partial derivatives $f_{\dot{x}}$ and g_x on the left hand side are evaluated at some points in the neighborhood of $(t_{n-1}, x^*(t_{n-1}), \dot{x}^*(t_{n-1}))$, and that of $(t_{n-1} + c_2 h, x^*(t_{n-1} + c_2 h))$, respectively. For sufficiently small h, the coefficient matrix of the linear system is boundedly invertible. It follows that $d_{X_2} = \mathscr{O}(h^2)$ and

$$f_{\dot{x}} d_{X_2} = f_{\dot{x}} \frac{c_2^2 h^2}{2} \ddot{x}^*(t_{n-1}) + \mathscr{O}(h^3), \tag{38}$$

where $f_{\dot{x}}$s on both sides are evaluated at the same point $(t_{n-1}, x^*(t_{n-1}), \dot{x}^*(t_{n-1}))$. It also follows that the estimate (38) holds when $f_{\dot{x}}$ is evaluated at $(t_{n-1} + c_2 h, x^*(t_{n-1} + c_2 h), \dot{x}^*(t_{n-1} + c_2 h))$. This will be used in the next step. From the method (27) we obtain the identity

$$\frac{1}{b_2}\left[\frac{x_n - x_{n-1}}{h} - b_1 K_1\right] = \frac{1}{b_2 h}\left[x_n - \frac{b_1}{a_{21}} X_2 + (\frac{b_1}{a_{21}} - 1) x_{n-1})\right]$$

Denoting the local solution of (31) by $\mathscr{S}(t_n, x_{n-1}, h)$, we want to bound the local error $d_n = x^*(t_n) - \mathscr{S}(t_n, x^*(t_{n-1}), h)$. Replacing x_n and X_2 in the left hand sides of (31) by $x^*(t_n)$ and $\mathscr{T}(t_n, x^*(t_{n-1}); h)$, respectively, and expanding into a Taylor series, we have

$$
\begin{aligned}
& f(t_{n-1}+c_2h, x^*(t_{n-1}+c_2h), \tfrac{1}{b_2h}\left[x^*(t_n) - \tfrac{b_1}{a_{21}}\mathscr{T}(t_n, x^*(t_{n-1}); h) + (\tfrac{b_1}{a_{21}} - 1)x^*(t_{n-1})\right]) \\
& = f(t_{n-1}+c_2h, x^*(t_{n-1}+c_2h), \tfrac{1}{b_2h}\left[x^*(t_n) - \tfrac{b_1}{a_{21}}(x^*(t_{n-1}+c_2h) - d_{X_2}) + (\tfrac{b_1}{a_{21}} - 1)x^*(t_{n-1})\right]) \\
& = f(t_{n-1}+c_2h, x^*(t_{n-1}+c_2h), \\
& \tfrac{1}{b_2h}\left[(1-\tfrac{c_2b_1}{a_{21}})\dot{x}^*(t_{n-1}+c_2h) + (\tfrac{1}{2} - c_2 + \tfrac{b_1c_2^2}{2a_{21}})h^2\dddot{x}^*(t_{n-1}^*+c_2h) + \tfrac{b_1}{a_{21}}d_{X_2}\right] + \mathscr{O}(h^2)) \\
& = f(t_{n-1}+c_2h, x^*(t_{n-1}+c_2h), \dot{x}^*(t_{n-1}+c_2h) \\
& \qquad + \tfrac{1}{b_2h}\left[(\tfrac{1}{2} - c_2 + \tfrac{b_1c_2}{2})h^2\dddot{x}^*(t_{n-1}^*+c_2h) + \tfrac{b_1}{a_{21}}d_{X_2}\right] + \mathscr{O}(h^2)) \\
& = \tfrac{1}{b_2h}f_{\dot{x}}\left[(\tfrac{1}{2} - c_2 + \tfrac{b_1c_2}{2})h^2\dddot{x}^*(t_{n-1}^*+c_2h) + \tfrac{b_1}{a_{21}}d_{X_2}\right] + \mathscr{O}(h^2) \\
& = \tfrac{h}{b_2}\left[(\tfrac{1}{2} - c_2 + \tfrac{b_1c_2}{2}) + \tfrac{b_1c_2}{2}\right] f_{\dot{x}}\dddot{x}^* + \mathscr{O}(h^2) = \tfrac{h}{b_2}(\tfrac{1}{2} - c_2b_2)f_{\dot{x}}\dddot{x}^* + \mathscr{O}(h^2) = \mathscr{O}(h^2).
\end{aligned}
$$

Here we have made use of the conditions in (28) and the estimate (38). In addition, we have

$$g(t_n, x^*(t_n)) = 0.$$

By definition, $\mathscr{S}(t_n, x^*(t_{n-1}); h)$ solves the system

$$
\begin{aligned}
& f(t_{n-1}+c_2h, x^*(t_{n-1}+c_2h), \\
& \tfrac{1}{b_2h}\left[\mathscr{S}(t_n, x^*(t_{n-1}); h) - \tfrac{b_1}{a_{21}}\mathscr{T}(t_n, x^*(t_{n-1}); h) + (\tfrac{b_1}{a_{21}} - 1)x^*(t_{n-1})\right]) = 0, \\
& g(t_n, \mathscr{S}(t_n, x^*(t_{n-1}); h)) = 0.
\end{aligned}
$$

Again applying the mean value theorem, we have the linear system

$$\begin{bmatrix} \frac{1}{b_2h}f_{\dot{x}} \\ g_x \end{bmatrix} d_n = \begin{bmatrix} \mathscr{O}(h^2) \\ 0 \end{bmatrix},$$

or equivalently

$$\begin{bmatrix} f_{\dot{x}} \\ g_x \end{bmatrix} d_n = \begin{bmatrix} \mathscr{O}(h^3) \\ 0 \end{bmatrix},$$

where the partial derivatives $f_{\dot{x}}$ and g_x are evaluated at some points in the neighborhood of $(t_{n-1}+c_2h, x^*(t_{n-1}+c_2h), \dot{x}^*(t_{n-1}+c_2h))$ and of $(t_n, x^*(t_n))$, respectively. For sufficiently small h, the coefficient matrix is boundedly invertible, which implies that $d_n = \mathscr{O}(h^3)$. This means that the schemes (29) or (31) are consistent of order $p = 2$. If (30) and (32) are used instead of (29(b)) and (31(b)), then the same consistency order is established in analogous way.

The stability analysis of HERK schemes (29) or (31) can be carried out similarly to that for half-explicit one-leg and half-explicit multi-step methods. We decompose x appropriately into (x_1, x_2) and then eliminate x_2 by solving the algebraic equation. We then obtain a corresponding discretization scheme for the underlying ODE (13). Because of the one-step property, it is even slightly less complicated to show the stability of the computation of x_1 using the same argument as in the proof of Theorem 1. Combined with the consistency results, this implies the convergence of x_1. Then, as a consequence, the convergence of x_2 follows.

In the case when we use (30) and (32) instead of (29(b)) and (31(b)), the stability analysis can be done directly, without splitting x and reducing to (13).

First, we show that there exists a constant $K_1 > 0$, which is independent of h, such that for sufficiently small h, the inequality

$$\left\|\mathscr{T}(t_n, x^*(t_{n-1}); h) - \mathscr{T}(t_n, x_{n-1}; h)\right\| \leq (1 + hK_1) \left\|x^*(t_{n-1}) - x_{n-1}\right\| \tag{39}$$

holds. To this end, we differentiate the first equation of (36) with respect to x_{n-1}, and using (29), we have

$$f_x + \frac{1}{ha_{21}} f_{\dot{x}}(\mathscr{T}_{x_{n-1}} + I_n) = 0.$$

Hence, it follows that

$$f_{\dot{x}} \mathscr{T}_{x_{n-1}} = f_{\dot{x}} + \mathscr{O}(h).$$

In addition, differentiating the relation

$$g(t_{n-1} + c_2 h, \mathscr{T}(t_n, x_{n-1}; h)) = g(t_{n-1}, x_{n-1}),$$

with respect to x_{n-1}, we obtain

$$g_x(t_{n-1} + c_2 h, \mathscr{T}(t_n, x_{n-1}; h)) \mathscr{T}_{x_{n-1}} = g_x(t_{n-1}, x_{n-1}) = g_x(t_{n-1} + c_2 h, \mathscr{T}(t_n, x_{n-1}; h)) + \mathscr{O}(h).$$

Gathering the terms, we obtain the linear system

$$\begin{bmatrix} f_{\dot{x}} \\ g_x \end{bmatrix} \mathscr{T}_{x_{n-1}} = \begin{bmatrix} f_{\dot{x}} \\ g_x \end{bmatrix} + \mathscr{O}(h).$$

For sufficiently small h, the coefficient matrix function is boundedly invertible, which implies that

$$\mathscr{T}_{x_{n-1}} = I_n + \mathscr{O}(h). \tag{40}$$

Then, we have

$$\begin{aligned} \mathscr{T}(t_n, x^*(t_{n-1}); h) - \mathscr{T}(t_n, x_{n-1}; h) &= \int_0^1 \mathscr{T}_{x_{n-1}}(t_n, x_{n-1} + s(x^*(t_{n-1}) - x_{n-1}); h) ds \\ &= (I_n + \mathscr{O}(h))(x^*(t_{n-1}) - x_{n-1}). \end{aligned}$$

So, inequality (39) immediately follows. Using the estimate (40), by an analogous argument, we can show that

$$\mathscr{S}_{x_{n-1}} = I_n + \mathscr{O}(h)$$

holds as well. Hence, for sufficiently small h, there exists a constant $K > 0$ such that

$$\left\|\mathscr{S}(t_n, x^*(t_{n-1}); h) - \mathscr{S}(t_n, x_{n-1}; h)\right\| \leq (1 + hK)\left\|x^*(t_{n-1}) - x_{n-1}\right\|.$$

This exactly means that the schemes (29), (31) are stable. Since the scheme is consistent of order $p = 2$ and stable, it follows that the two-stage HERK method applied to (1) is convergent of order $p = 2$. □

Example 7 The explicit two-stage Runge-Kutta method with the Butcher diagram 7

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1 - \frac{1}{2\alpha} & \frac{1}{2\alpha} \end{array}$$

where $\alpha \in (0, 1]$ is a parameter, is well-known to be of second order for ODEs. For $\alpha = 1/2$, we have the explicit midpoint rule, while with $\alpha = 1$, the explicit trapezoidal rule is obtained. The generalized stability function for the method as applied to the test DAE (22) is

$$R(z, w) = \frac{1}{1 + w(1 - \alpha)}\left[1 + z + z^2/2 + \alpha w\left(3 - 2\alpha + 2z(1 + \alpha) - \alpha w(3 - 2\alpha) + \alpha z^2\right)\right].$$

For $w = 0$, the stability function $R(z) = 1 + z + z^2/2$ is exactly the stability function of the explicit Runge-Kutta method (7) that is well analyzed in the numerical analysis of non-stiff ODEs, e. g., see [10].

A generalization of the construction to arbitrary high-stage half-explicit Runge-Kutta methods is straightforward. Consider an s-stage Runge-Kutta method given by Table 2. We assume that $a_{i+1,i} \neq 0$ for $i = 2, \ldots, s$ and $b_s \neq 0$. The first stage-approximation $X_1 = x_{n-1}$ is obviously available. The i-th stage-approximation X_i is obtained successively by solving the algebraic systems

$$\begin{aligned} &\text{(a)} \qquad f\left((t_{n-1} + c_i h, X_i, \tfrac{1}{a_{i+1,i}}\left[\tfrac{X_{i+1} - x_{n-1}}{h} - \textstyle\sum_{j=1}^{i-1} a_{i+1,j} K_j\right]\right) = 0, \\ &\text{(b)} \qquad g(t_{n-1} + c_{i+1} h, X_{i+1}) = 0, \end{aligned} \tag{41}$$

for $i = 1, \cdots, s-1$. Finally, the numerical solution x_n at time step $t = t_n$ is determined by the system

$$\begin{array}{ll} \text{(a)} & f\Big((t_{n-1} + c_s h, X_s, \tfrac{1}{b_s}\left[\tfrac{x_n - x_{n-1}}{h} - \sum_{i=1}^{s-1} b_i K_i\right]\Big) = 0, \\ \text{(b)} & \hfill g(t_n, x_n) = 0. \end{array} \qquad (42)$$

Applying this method to the special matrix-valued DAE system (4), these become simply linear systems of matrix equations,

$$\begin{bmatrix} E_1(t_{n-1}^{(i)}) \\ A_2(t_{n-1}^{(i)}) \end{bmatrix} X_{i+1} = \begin{bmatrix} E_1(t_{n-1}^{(i)}) \left[x_{n-1} + h\sum_{j=1}^{i-1} a_{i+1,j} K_j\right] + h a_{i+1,i} F(t_{n-1}^{(i)}, X_i) \\ 0 \end{bmatrix},$$

for $i = 1, \cdots, s-1$, and

$$\begin{bmatrix} E_1(t_{n-1}^{(s)}) \\ A_2(t_n) \end{bmatrix} x_n = \begin{bmatrix} E_1(t_{n-1}^{(s)}) \left[x_{n-1} + h\sum_{i=1}^{s-1} b_i K_i\right] + h b_s F(t_{n-1}^{(s)}, X_s) \\ 0 \end{bmatrix},$$

respectively, where $t_{n-1}^{(i)} = t_{n-1} + c_i h$, $i = 1, \cdots, s$.

Again when using direct solution methods, these linear systems can be solved efficiently by one LU factorization per system, i. e., a total of s LU-factorizations is needed. The convergence analysis for (41) and (42) can be carried out similar to the case $s = 2$, but more effort is needed.

Remark 8 The half-explicit Runge-Kutta methods proposed here for strangeness-free DAEs (1) can be considered as a generalization of the half-explicit Runge-Kutta methods for semi-explicit DAEs of index at most one analyzed in [2, 9]. However, their implementation is slightly different. For semi-explicit DAEs, not only the differential and the algebraic parts are separated, but also the derivative of the differential component is explicitly given, which is not the case with (1). Hence, the differential component of each stage is computed first and then the algebraic component follows by solving an algebraic system. Here the whole stage-approximation must be evaluated once by solving a larger algebraic system. In fact, in the case of semi-explicit systems, we have $\partial f / \partial \dot{x} = E_1 = \begin{bmatrix} I & 0 \end{bmatrix}$, hence the Jacobian of the algebraic system (29) has a special lower block-triangular form with the identity matrix in the left upper block. This fact makes the use of half-explicit Runge-Kutta methods simpler when they are applied to semi-explicit DAEs.

Remark 9 The implementation of the HERKs for (1) is similar to that of diagonally implicit Runge-Kutta (DIRK) methods applied to ODEs. In particular, the strictly lower triangular matrix $A = [a_{ij}]_{i,j=1}^{s}$ pretends to be lower triangular

Table 2: Butcher tableau of explicit s-stage Runge-Kutta method

$$\begin{array}{c|cccc} 0 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_s & a_{s1} & a_{s2} & \cdots & 0 \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

with nonzero diagonal by shifting up each row of A and b^T by one row. This explains why the HERKs are feasible and convergent of similar orders as implicit and stiffly accurate Runge-Kutta methods. In addition, all popular embedded Runge-Kutta solvers like Runge-Kutta-Fehlberg and Dormand-Prince methods, equipped with efficient error control and step-size selection, can be adapted to solving these strangeness-free DAEs.

5 Numerical experiments

Half-explicit methods as derived in the preceding sections have been implemented and applied to DAE examples. For illustration we present results for the half-explicit versions of the one-leg Adams-Bashforth method (HEOL) from Example 3, the two-step Adams-Bashforth method (HEAB) from Example 5, and the trapezoidal and midpoint Runge-Kutta methods (HETRA, HEMID) from Example 7.

Example 10 Our first test problem is a constructed DAE with a known exact solution. We consider the DAE

$$\begin{bmatrix} 1 & t \\ 0 & 0 \end{bmatrix} \dot{x} = \begin{bmatrix} x_1 x_2 + e^t + t\cos t - e^t \sin t \\ e^{-t} x_1 - x_2 + \sin t - 1 \end{bmatrix}, \quad t \geq 0, \tag{43}$$

together with the initial condition $x(0) = (0,\, 1)^T$. It is easy to check that the DAE is strangeness-free and that the exact unique solution is $x_1 = e^t$, $x_2 = \sin t$. We solve the initial value problem by the described HEOL and HERK methods on a uniform mesh with different stepsize h. The actual errors $\max |x_i(t_n) - x_{in}|$, $i = 1,2$, of different methods versus h are displayed. In addition, based on the actual errors, we also give numerical estimates for the convergence rate, which confirm the proved convergence orders.

Example 11 Our second test problem is a matrix-valued DAE of type (4), which arises from the continuous QR and SVD methods proposed for approximating

Table 3: Solution of initial value problem (43) with one-leg Adams-Bashforth method in Ex. 3

ξ	h	Error in x_1	Error order in x_1	Error in x_2	Error order in x_1
2	0.1	0.004791028	2.17	0.001762521	2.17
2	0.05	0.001067391	2.10	0.000392671	2.10
2	0.025	0.00024866	2.06	9.14768E-05	2.06
2	0.01	3.79341E-05	2.02	1.39552E-05	2.02
2	0.005	9.32463E-06	2.01	3.43034E-06	2.01
2	0.0025	2.31107E-06	2.01	8.50196E-07	2.01
1.5	0.1	0.005075127	2.25	0.001867035	2.25
1.5	0.05	0.001069938	2.16	0.000393608	2.16
1.5	0.025	0.000239792	2.09	8.82147E-05	2.09
1.5	0.01	3.55486E-05	2.04	1.30776E-05	2.04
1.5	0.005	8.64526E-06	2.02	3.18041E-06	2.02
1.5	0.0025	2.13074E-06	2.01	7.83855E-07	2.01
1	0.1	0.004108619	2.64	0.001511476	2.64
1	0.05	0.000657134	2.54	0.000241746	2.54
1	0.025	0.000112741	2.40	4.14753E-05	2.40
1	0.01	1.27928E-05	2.22	4.7062E-06	2.22
1	0.005	2.74791E-06	2.12	1.0109E-06	2.12
1	0.0025	6.30058E-07	2.07	2.31785E-07	2.07
0.5	0.1	0.004996419	0.73	0.00183808	0.73
0.5	0.05	0.003007934	1.57	0.001106557	1.57
0.5	0.025	0.001016036	1.81	0.000373779	1.81
0.5	0.01	0.000190678	1.93	7.01463E-05	1.93
0.5	0.005	5.01435E-05	1.96	1.84468E-05	1.96
0.5	0.0025	1.28518E-05	1.98	4.72791E-06	1.98

Table 4: Solution of initial value problem (43) with half-explicit Runge-Kutta method of Ex. 7

α	h	Error in x_1	Error order in x_1	Error in x_2	Error order in x_1
1	0.1	0.009389536	1.96	0.003454217	1.96
1	0.05	0.002411295	1.98	0.000887066	1.98
1	0.025	0.000610229	1.99	0.000224491	1.99
1	0.01	9.83115E-05	2.00	3.61668E-05	2.00
1	0.005	2.46325E-05	2.00	9.06178E-06	2.00
1	0.0025	6.16487E-06	2.00	2.26793E-06	2.00
0.5	0.1	0.004037535	1.96	0.001485326	1.96
0.5	0.05	0.00103773	1.98	0.000381759	1.98
0.5	0.025	0.000262811	1.99	9.66827E-05	1.99
0.5	0.01	4.23638E-05	2.00	1.55848E-05	2.00
0.5	0.005	1.06166E-05	2.00	3.90564E-06	2.00
0.5	0.0025	2.65735E-06	2.00	9.77584E-07	2.00

spectral intervals [14, 15]. We consider the computation of Lyapunov exponents of the DAE system given in [14, Example 38]. In Table 1 and Table 2 of that paper, the Lyapunov exponents, whose exact values are $\lambda_1 = 1$ and $\lambda_2 = -1$, are computed by the continuous *SVD* algorithm using the half-explicit Euler and the implicit Euler integrators, respectively. It has been shown in [14] that the half-explicit Euler integrator produces numerical results of almost the same accuracy as the implicit Euler integrator, but the former one requires less CPU time.

Here we test the continuous *QR* and *SVD* methods presented in [14, 15] combined with the half-explicit HEOL, HEAB, HETRA, and HEMID integrators. The numerical values of the Lyapunov exponents are computed on different $[0,T]$ intervals and with uniform step-sizes $h = 0.1$ and $h = 0.01$. The numerical results in Tables 5-7 reflect well the convergence order of the half-explicit methods. Furthermore, the dominance of the discretization error for the lower order method and a stepsize that is not sufficiently small ($h = 0.1$) and the dominance of interval truncation error for either high-order methods or small step-size, but with T that is not sufficiently large (e. g. $T = 100$), can be observed as well. Among the tested second-order integrators, the half-explicit Adams-Bashforth method is the fastest. However, if we consider both the accuracy and the CPU time, then the half-explicit trapezoidal method seems to be most recommendable.

Table 5: Lyapunov exponents computed via the continuous QR method and half-explicit Euler and half-explicit midpoint integrator

		QR-HEE			QR-HEMID		
T	h	λ_1	λ_2	CPU-time in s	λ_1	λ_2	CPU-time in s
100	0.1	0.9155	-0.9286	0.6250	0.9557	-0.9726	1.1563
100	0.01	0.9464	-0.9590	5.7813	0.9496	-0.9621	10.7188
500	0.1	0.9524	-0.9578	2.9375	0.9931	-1.0031	5.4063
500	0.01	0.9836	-0.9890	28.6250	0.9868	-0.9922	54.8125
1000	0.1	0.9584	-0.9592	5.7813	0.9991	-1.0045	10.7500
1000	0.01	0.9895	-0.9903	57.2031	0.9927	-0.9936	108.6094
5000	0.1	0.9638	-0.9642	29.0625	1.0048	-1.0097	54.4688
5000	0.01	0.9951	-0.9955	290.7344	0.9983	-0.9987	531.4844
10000	0.1	0.9646	-0.9649	57.4531	1.0056	-1.0104	106.6094
10000	0.01	0.9959	-0.9962	573.5938	0.9991	-0.9994	1062.6

Table 6: Lyapunov exponents computed via the continuous QR method with half-explicit one-leg and half-explicit Adams-Bashforth integrator

		QR-HEOL			QR-HEAB		
T	h	λ_1	λ_2	CPU-time in s	λ_1	λ_2	CPU-time in s
100	0.1	0.9713	-0.9866	1.0313	0.9351	-0.9476	0.7031
100	0.01	0.9571	-0.9697	9.5469	0.9494	-0.9619	6.0781
500	0.1	1.0117	-1.0191	4.8594	0.9720	-0.9774	3.0625
500	0.01	0.9948	-1.0003	48.1094	0.9865	-0.9919	30.5156
1000	0.1	1.0168	-1.0197	9.9063	0.9780	-0.9788	6.2969
1000	0.01	1.0007	-1.0016	97.5469	0.9925	-0.9933	61.4375
5000	0.1	1.0227	-1.0252	47.6094	0.9836	-0.9840	30.8281
5000	0.01	1.0063	-1.0068	475.0469	0.9980	-0.9985	304.5781
10000	0.1	1.0235	-1.0259	95.1875	0.9844	-0.9847	60.6250
10000	0.01	1.0072	-1.0075	945.5625	0.9989	-0.9991	604.3125

Table 7: Lyapunov exponents computed via the continuous QR and SVD methods with the half-explicit trapezoidal integrators

		QR-HETRA			SVD-HETRA		
T	h	λ_1	λ_2	CPU-time in s	λ_1	λ_2	CPU-time in s
100	0.1	0.9472	-0.9597	0.7969	0.9545	-0.9600	0.8281
100	0.01	0.9495	-0.9620	7.3750	0.9568	-0.9623	7.6094
500	0.1	0.9845	-0.9900	3.6563	0.9860	-0.9900	3.7969
500	0.01	0.9867	-0.9921	36.3281	0.9881	-0.9921	37.6094
1000	0.1	0.9906	-0.9914	7.2656	0.9913	-0.9914	7.5156
1000	0.01	0.9926	-0.9934	73.1250	0.9934	-0.9935	74.7188
5000	0.1	0.9962	-0.9966	36.2656	0.9963	-0.9966	37.7344
5000	0.01	0.9982	-0.9986	366.4219	0.9983	-0.9986	374.4375
10000	0.1	0.9970	-0.9973	72.7188	0.9971	-0.9973	74.8438
10000	0.01	0.9990	-0.9993	714.7188	0.9991	-0.9993	749.2656

6 Conclusion

In this paper we have discussed the use of half-explicit methods for solving general nonlinear DAEs in strangeness-free form. Half-explicit variants of explicit one-leg, linear multi-step, and Runge-Kutta methods are proposed and analyzed. These classes of methods offer an alternative choice and seem to be more efficient in solving non-stiff DAEs than the common implicit methods like BDF and Radau5. Particular efficiency is demonstrated when solving some semi-linear matrix-valued DAEs systems arising in the numerical computation of Lyapunov spectral intervals. As future work, a complete convergence analysis of high-order half-explicit Runge-Kutta methods as well as implementation of general half-explicit methods with error estimation and automatic step-size selection for solving general nonlinear strangeness-free DAEs would be of interest.

7 Acknowledgment

The authors thank Saskia Zurth for carrying out the numerical experiments for Example 10.

References

[1] M. Arnold. Half-explicit Runge-Kutta methods with explicit stages for differential-algebraic systems of index 2. *BIT*, 38:415–438, 1998.

[2] M. Arnold, K. Strehmel, and R. Weiner. Half-explicit Runge-Kutta methods for semi-explicit differential equations of index 1. *Numer. Math.*, 64:409–431, 1993.

[3] V. Brasey and E. Hairer. Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *SIAM J. Numer. Anal.*, 30:538–552, 1993.

[4] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential Algebraic Equations*. SIAM Publications, Philadelphia, PA, 2nd edition, 1996.

[5] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Springer-Verlag, Berlin, Germany, 2004.

[6] E. Eich-Soellner and C. Führer. *Numerical Methods in Multibody Systems*. Teubner Verlag, Stuttgart, Germany, 1998.

[7] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[8] E. Griepentrog and R. März. *Differential-Algebraic Equations and their Numerical Treatment*. Teubner Verlag, Leipzig, Germany, 1986.

[9] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Springer-Verlag, Berlin, Germany, 1989.

[10] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, Berlin, Germany, 2nd edition, 1993.

[11] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin, Germany, 2nd edition, 1996.

[12] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations. Analysis and Numerical Solution*. EMS Publishing House, Zürich, Switzerland, 2006.

[13] P. Kunkel and V. Mehrmann. Stability properties of differential-algebraic equations and spin-stabilized discretization. *Electr. Trans. Num. Anal.*, 26:383–420, 2007.

[14] V.H. Linh and V. Mehrmann. Approximation of spectral intervals and associated leading directions for linear differential-algebraic systems via smooth singular value decompositions. Preprint 711, DFG Research Center MATHEON, TU Berlin, Germany, 2010. `url: http://www.matheon.de/` .

[15] V.H. Linh, V. Mehrmann, and E. Van Vleck. *QR* methods and error analysis for computing Lyapunov and Sacker-Sell spectral intervals for linear differential-algebraic equations. *Adv. Comput. Math.*, 35:281–322, 2011.

[16] W. Liniger. Multistep and one-leg methods for implicit mixed differential algebraic systems. *IEEE Trans. Circ. and Syst.*, CAS-26:755–762, 1979.

[17] R. März. On one-leg methods for differential-algebraic equations. *Circ. Syst. Signal Proc.*, 2:87–95, 1986.

[18] A. Murua. Partitioned half-explicit Runge-Kutta methods for differential algebraic systems of index 2. *Computing*, 59:43–61, 1997.

[19] P. J. Rabier and W. C. Rheinboldt. *Theoretical and Numerical Analysis of Differential-Algebraic Equations*, volume VIII of *Handbook of Num. Analysis*. Elsevier Publications, Amsterdam, The Netherlands, 2002.

[20] R. Riaza. *Differential-algebraic systems. Analytical aspects and circuit applications*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ., 2008.

[21] N. N. Pham Thi, W. Hundsdorfer, and B. P. Sommeijer. Positivity for explicit two-step methods in linear multistep and one-leg form. *BIT*, 46:875–882, 2006.

[22] R.W.C.P. Verstappen and A.E.P. Veldman. Direct numerical simulation of turbulence at lower costs. *J. Engrg. Math.*, 32:143–159, 1997.