Fractional step methods for index-1 differential-algebraic equations

Prashanth K. Vijalapura [a]

[a] *Department of Civil and Environmental Engineering, 2108 Shattuck Ave., University of California, Berkeley, CA 94720-1716*

John Strain [b] Sanjay Govindjee [a,1]

[b] *Department of Mathematics 970 Evans Hall, #3840 University of California Berkeley, CA 94720*

Abstract

In the numerical solution of ordinary differential systems, the method of fractional steps (also known as operator splitting) yields high-order accurate schemes based on separate, computationally convenient treatments of distinct physical effects. Such schemes are equally desirable but much less accurate for semi-explicit index-1 differential-algebraic equations (DAEs). In the first half of this note, it is shown that naïve application to DAEs of standard splitting schemes suffers from order reduction: both first and second-order schemes are only first-order accurate for DAEs. In the second half of this note, a new family of higher-order splitting schemes for semi-explicit index-1 DAEs is developed. The new schemes are based on a deferred correction paradigm in which an error equation is solved numerically, and therefore inherit a simple computationally-convenient structure. Higher-order convergence of the new schemes is proved, and numerical results confirm the expected order of accuracy in addition to establishing efficiency.

Key words: operator splitting, fractional step methods, index-1 differential-algebraic equations, deferred correction, higher order methods.

[1] Corresponding author
Email address: sanjay@ce.berkeley.edu (Sanjay Govindjee)

1 Introduction

This paper explores new approaches to the construction and analysis of fractional step methods for solving differential-algebraic equations (DAEs). The method of fractional steps, or operator splitting, is often used as an efficient numerical integration technique for solving initial value problems in ordinary differential equations (ODEs) [1]. Operator splitting combines integration schemes for subproblems into an efficient scheme for the overall problem. For differential-algebraic equations, which combine algebraic constraints with ODEs, splitting schemes separate the algebraic constraints from the differential equations. For example, when the ODEs and constraints arise from distinct but coupled physical phenomena, splitting schemes can take full advantage of existing computer codes tuned for each subproblem.

This paper examines fractional step methods for index-1 DAEs in the most natural semi-explicit form. Common methods for general index-1 DAEs include one-step implicit Runge-Kutta (RK) methods [2,3] and multistep backward differentiation formulae (BDF) [2]. BDF methods require an expensive simultaneous integration of the ODEs and satisfaction of the constraints. Implicit RK methods are even more expensive, as they require solution of nonlinear systems whose size is the number of stages multiplied by the original size of the DAE. In the special case of semi-explicit index-1 DAEs, explicit Runge-Kutta methods [3] efficiently decouple the ODEs solver from the algebraic constraints. However, these methods are only conditionally stable and become inefficient for stiff DAEs. Some special-purpose splitting schemes preserve the dissipative structure of the DAE [4,5]. These schemes successfully avoid both the expense of fully implicit schemes and the conditional stability of explicit Runge-Kutta schemes, but lack generality. All these issues have led us to explore splitting schemes in greater detail than presently available in the literature.

The paper is organized as follows: In section 2, we show that the standard one-pass and two-pass symmetric splitting schemes which are respectively first- and second-order accurate for ODEs, are only first-order accurate for DAEs. In section 3, this "order reduction" is illustrated by a two-dimensional example. Order reduction is overcome by a new splitting scheme, based on deferred correction of a first-order scheme, which is introduced and analyzed in section 4. The deferred correction paradigm solves an error equation with the same structure as the original DAE, using the original first-order scheme. The resulting scheme is simple, efficient, and second-order accurate. It can be iterated to obtain efficient schemes with third and higher-order accuracy. Finally, section 5 presents numerical examples demonstrating the performance of our second and third-order accurate splitting schemes. In particular, an application to a large system of index-1 DAEs, arising from electrical circuit

simulation, illustrates efficiency of higher order splitting schemes over a highly optimized, state-of-the-art, fifth order Runge Kutta scheme–RADAU5.

2 Order reduction

2.1 The ODE case

In this section, we analyze the accuracy of operator splitting algorithms for ODEs [6]. Consider a first-order ordinary differential system written in partitioned form

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} f(x,y) \\ g(x,y) \end{pmatrix}. \tag{1}$$

For many coupled problems, the partitioned variables x and y describe different physical variables; for example in [5,7] they denote mechanical deformation and an auxiliary field, respectively.

A splitting scheme approximates the solution of Eqn. (1) by solving the following split equations in each time step:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} f(x,y) \\ 0 \end{pmatrix}}_{\Phi}; \qquad \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 \\ g(x,y) \end{pmatrix}}_{\Gamma}. \tag{2}$$

For example, we denote by $\Gamma_h \circ \Phi_h$ the one-pass splitting algorithm which evolves the solution from t_n to $t_{n+1} = t_n + h$ by (a) solving the first split ODE over one time step h with right-hand side Φ, and (b) solving the second split ODE over one time step h with right-hand side Γ, starting from the solution produced by (a). The other one-pass splitting algorithm $\Phi_h \circ \Gamma_h$ is similarly defined. The two-pass symmetric algorithms of [6] are $\Phi_{h/2} \circ \Gamma_h \circ \Phi_{h/2}$ and $\Gamma_{h/2} \circ \Phi_h \circ \Gamma_{h/2}$. They are symmetric because they take alternate half-steps of the two one-pass algorithms: for example,

$$\Phi_{h/2} \circ \Gamma_h \circ \Phi_{h/2} = (\Phi_{h/2} \circ \Gamma_{h/2}) \circ (\Gamma_{h/2} \circ \Phi_{h/2}).$$

The classical error analysis of one-pass algorithms leads to a splitting error of $O(h^2)$ per time step and first-order accuracy. The symmetric two-pass algorithms attain second-order accuracy because the splitting error per time step is $O(h^3)$ [8].

Note 1 *For finite-dimensional ODEs, Lipschitz continuity of the split evolution operators* Φ *and* Γ *implies convergence for the split solution. However,*

for infinite-dimensional ODEs or PDEs, stability requires that Φ and Γ generate bounded semi-groups. Unconditionally stable splitting schemes arising from dissipative dynamical systems [9] occur in many applications, notably in transient thermomechanical problems [5].

2.2 The DAE case

In this section, splitting and global errors are analyzed for the one-pass and two-pass algorithms applied to DAEs of the partitioned form

$$\begin{pmatrix} 0 \\ \dot{y} \end{pmatrix} = \begin{pmatrix} f(x,y) \\ g(x,y) \end{pmatrix} := \chi. \tag{3}$$

The DAEs are assumed to be of index 1[2] [2], meaning that the Jacobian matrix f_x of $f(x,y)$ with respect to x is invertible in a neighborhood of the solution to Eqn. (3). We split the partitioned equations into

$$\begin{pmatrix} 0 \\ \dot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} f(x,y) \\ 0 \end{pmatrix}}_{\Phi - Algebraic} \qquad \text{and} \qquad \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 \\ g(x,y) \end{pmatrix}}_{\Gamma - ODEs}.$$

so $\chi = \Phi + \Gamma$.

2.2.1 One-pass algorithms

The one-pass algorithm $\Gamma_h \circ \Phi_h$ first finds the algebraic variables $x = x_{n+1}$ that satisfy the algebraic constraint $f(x,y) = 0$ with $y = y_n$ fixed, and then evolves the ODE variables y through time h with $x = x_{n+1}$ fixed. The algorithm $\Phi_h \circ \Gamma_h$ is similarly defined, with evolution of y through h followed by choosing x to satisfy the algebraic constraints.

Since the DAE has index 1 by assumption, the implicit function theorem implies that a C^1 function φ exists such that

$$f(x,y) = 0 \qquad \text{implies} \qquad x = \varphi(y) \tag{5}$$

(near a solution (x,y)). Thus y satisfies a pure ODE

$$\dot{y} = g(\varphi(y), y). \tag{6}$$

[2] For coupled mechanical-auxiliary field problems, the index-1 assumption is equivalent to the natural assumption that the stiffness matrix associated with the mechanical degrees of freedom is invertible.

By Taylor expansion and the chain rule, the exact solution $y(t)$ satisfies

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + h\dot{y}(t_n) + \frac{h^2}{2!}\ddot{y}(t_n) + O(h^3) \\ &= y(t_n) + hg(\varphi(y(t_n)), y(t_n)) \\ &\quad + \frac{h^2}{2!}\left(g_x(\varphi(y(t_n)), y(t_n))\dot{\varphi}(y(t_n)) + g_y(\varphi(y(t_n)), y(t_n))\dot{y}(t_n)\right) + O(h^3) \end{aligned}$$

where g_x denotes the Jacobian matrix of partial derivatives of g with respect to the components of x. The exact solution $x(t)$ satisfies $x(t_{n+1}) = \varphi(y(t_{n+1}))$.

We now analyze the local error committed in one time step, starting from the exact value $y(t_n)$. The one-pass algorithm $\Phi_h \circ \Gamma_h$ generates an approximation $v(t)$ such that

$$\dot{v} = g(x(t_n), v) = g(\varphi(y(t_n)), v) \qquad t_n \le t < t_{n+1}, \tag{7}$$

and $v(t_n) = y(t_n)$. Thus by Taylor expansion and the chain rule,

$$\begin{aligned} v(t_{n+1}) &= v(t_n) + h\dot{v}(t_n) + \frac{h^2}{2!}\ddot{v}(t_n) + O(h^3) \\ &= y(t_n) + hg(\varphi(y(t_n)), y(t_n)) \\ &\quad + \frac{h^2}{2!}g_y(\varphi(y(t_n)), y(t_n))\dot{y}(t_n) + O(h^3). \end{aligned} \tag{8}$$

Hence the local error is

$$v(t_{n+1}) - y(t_{n+1}) = e(t_{n+1})h^2 + O(h^3) \tag{9}$$

where e is a smooth function.

Applying Φ_h yields a numerical approximation u to x which satisfies

$$\begin{aligned} u(t_{n+1}) &= \varphi(v(t_{n+1})) \\ &= \varphi(y(t_{n+1}) + h^2 e_{n+1} + O(h^3)) \\ &= \varphi(y(t_{n+1})) + h^2\varphi_y(y(t_{n+1}))e_{n+1} + O(h^3) \\ &= x(t_{n+1}) + h^2\varphi_y(y(t_{n+1}))e_{n+1} + O(h^3). \end{aligned} \tag{10}$$

From equations (9) and (10) it follows that $\|x(t_{n+1}) - u(t_{n+1})\| = O(h^2)$ and $\|y(t_{n+1}) - v(t_{n+1})\| = O(h^2)$. where $\|\cdot\|$ denotes any convenient error norm. In other words, the one-pass algorithm $\Gamma_h \circ \Phi_h$ commits second-order local errors within a time step, starting from the exact solution.

First-order convergence can then be easily proved. Consider the one-pass splitting algorithm as a one-step method for the ODE equivalent of the DAE, which

takes in solution values $x(t_n) = \varphi(y(t_n))$ and $y(t_n)$ and returns the approximation $v(t_{n+1})$ which solves (6) to order $O(h^2)$ within a time step. Assuming a standard Lipschitz condition $\|g_y(\varphi(y), y))\| \leq L$, Theorem 3.4 of [10], for example, shows that the global error $E = y(t) - v(t)$ satisfies:

$$\|E\| \leq h\frac{C}{L}\exp[L(t - t_0)], \tag{11}$$

where t_0 is the initial time and C is a constant independent of h. Eqn. (11) proves first-order accurate global convergence.

For the alternate splitting scheme $\Gamma_h \circ \Phi_h$, one obtains

$$u(t_{n+1}) = x(t_n) \quad (\text{updating } u(t_{n+1}) \text{ by } \Phi_h). \tag{12}$$

The approximation v then satisfies

$$\begin{aligned} \dot{v} &= g(u(t_{n+1}), v) \\ &= g(x(t_n), v). \end{aligned} \tag{13}$$

As above, $\|u(t_{n+1}) - x(t_{n+1})\| = O(h)$ and $\|v(t_{n+1}) - y(t_{n+1})\| = O(h^2)$. Even though the local error in x is $O(h)$ in each step, the order of accuracy of the $\Phi_h \circ \Gamma_h$ splitting scheme is also 1. This follows because evolution under Φ_h simply updates $u = \varphi(v)$ given v, and is therefore controlled by the errors in v alone. Indeed, the sequence of fractional steps is $\ldots (\Gamma_h \circ \Phi_h) \circ (\Gamma_h \circ \Phi_h) \circ (\Gamma_h \circ \Phi_h)$. Since the initial conditions satisfy $f(u(t_0), v(t_0)) = 0$, the first update Φ_h is redundant. Later in the sequence, Φ_h simply updates u given v, so the sequence is equivalent to

$$\ldots (\Phi_h \circ \Gamma_h) \circ (\Phi_h \circ \Gamma_h) \circ (\Phi_h \circ \underbrace{\Gamma_h}_{1^{st}step}). \tag{14}$$

We have already shown first-order global convergence for the latter sequence, as a consequence of which the former sequence is also globally convergent with first-order. The same conclusions hold even when variable stepsizes $h_1, h_2, \ldots$ are used in each time step. This completes the proof of first-order accuracy for the global error for the $\Gamma_h \circ \Phi_h$ split as well.

Before analyzing the order of the two-pass algorithm, we make a few observations.

Note 2 *The derivation provided here accounts only for errors due to operator splitting; exact time integration of the split flow operators is assumed. The additional discretization error due to approximate time stepping (in the linear case) is analyzed in [11].*

Note 3 *If the right hand sides g and f depend explicitly on time, so the DAE is non-autonomous, one can convert the system to an equivalent autonomous one by the standard technique: augment the ODEs by the equation $\dot{t} = 1$, with initial conditions $t = t_0$. However, in the later discussion of deferred correction this will not be possible, as the correction equations are always non-autonomous DAEs.*

Note 4 *The Lipschitz assumption is used only to ensure stability of the splitting scheme. If the sub-operators Φ and Γ are dissipative, stability of the splitting scheme is automatic and the proof extends even to the infinite dimensional case. (Then dissipativity guarantees that Φ and Γ generate contractive semigroups [12].)*

2.2.2 Two-Pass Algorithms

The main result of this section is that two-pass schemes are only first-order accurate for DAEs, even though they are second-order for ODEs.

Consider the two-pass algorithm $\Phi_{h/2} \circ \Gamma_{h/2} \circ \Phi_{h/2}$. The starting values for calculating the local truncation error are $y(t_n)$ and $x(t_n) = \varphi(y(t_n))$. The fractional step $\Phi_{h/2}$ gives $u(t_{n+\frac{1}{2}}) = \varphi(y(t_n)) = x(t_n)$. The fractional step Γ_h then gives

$$\begin{aligned} \dot{v} &= g(u(t_{n+\frac{1}{2}}), v) \\ &= g(x(t_n), v) \quad \text{for } t \in [t_n, t_{n+1}]. \end{aligned} \tag{15}$$

Finally, the fractional step $\Phi_{h/2}$ for the second pass updates $u(t_{n+1})$ as

$$u(t_{n+1}) = \varphi(v(t_{n+1})). \tag{16}$$

Since the first fractional step $\Phi_{h/2}$ is redundant, the updates given by (15) and (16) exactly correspond to the $\Gamma_h \circ \Phi_h$ case and as before lead to second-order splitting error. Consequently, as for the one-pass algorithm, the two-pass splitting scheme $\Phi_{h/2} \circ \Gamma_h \circ \Phi_{h/2}$ is only globally first-order accurate. This is in sharp contrast to the second-order accuracy of two-pass algorithms for ODEs.

The more interesting splitting error analysis occurs for the $\Gamma_{h/2} \circ \Phi_h \circ \Gamma_{h/2}$ sequence. The steps can be summarized as:

(1) $\underline{\Gamma_{h/2}}$: Update $v(t_{n+\frac{1}{2}})$ with $v(t_n) = y(t_n)$ by exactly solving

$$\dot{v} = g(\varphi(y(t_n)), v) \quad t \in [t_n, t_{n+\frac{1}{2}}]. \tag{17}$$

Repeating the earlier analysis, we obtain

$$v(t_{n+\frac{1}{2}}) - y(t_{n+\frac{1}{2}}) = h^2 e_{n+\frac{1}{2}} + O(h^3). \tag{18}$$

(2) $\underline{\Phi_h}$: Update $u(t_{n+1})$ exactly, using the implicit function theorem.

$$u(t_{n+1}) = \varphi(v(t_{n+\frac{1}{2}})). \tag{19}$$

The splitting error is calculated as follows:

$$\begin{aligned} u(t_{n+1}) &= \varphi(y(t_{n+\frac{1}{2}}) + h^2 e_{n+\frac{1}{2}} + O(h^3)) \\ &= \varphi(y(t_{n+1}) - \frac{h}{2}\dot{y}(t_{n+1}) + O(h^2)) \\ &= \varphi(y(t_{n+1})) + O(h) \\ &= x(t_{n+1}) + O(h), \end{aligned} \tag{20}$$

implying $\|u(t_{n+1}) - x(t_{n+1})\| = O(h)$.

(3) $\underline{\Gamma_{h/2}}$: Update $v(t_{n+1})$ with $v(t_{n+\frac{1}{2}})$ as the initial value by exactly solving,

$$\dot{v} = g(\varphi(v(t_{n+\frac{1}{2}})), v) \quad t \in [t_{n+\frac{1}{2}}, t_{n+1}]. \tag{21}$$

The splitting error in $v(t_{n+1})$ is found by expanding the exact solution around $t_{n+\frac{1}{2}}$.

$$\begin{aligned} y(t_{n+1}) &= y(t_{n+\frac{1}{2}}) + \frac{h}{2}\dot{y}(t_{n+\frac{1}{2}}) + \frac{h^2}{8}\ddot{y}(t_{n+\frac{1}{2}}) + O(h^3) \\ &= y(t_{n+\frac{1}{2}}) + \frac{h}{2} g(\varphi(y(t_{n+\frac{1}{2}})), y(t_{n+\frac{1}{2}})) + O(h^2). \end{aligned} \tag{22}$$

Similarly, expanding the approximate solution,

$$\begin{aligned} v(t_{n+1}) &= v(t_{n+\frac{1}{2}}) + \frac{h}{2}\dot{v}(t_{n+\frac{1}{2}}) + \frac{h^2}{8}\ddot{v}(t_{n+\frac{1}{2}}) + O(h^3) \\ &= v(t_{n+\frac{1}{2}}) + \frac{h}{2} g(\varphi(y(t_{n+\frac{1}{2}})), y(t_{n+\frac{1}{2}})) + O(h^2). \end{aligned} \tag{23}$$

Using $y(t_{n+\frac{1}{2}}) = v(t_{n+\frac{1}{2}}) + h^2 e_{n+\frac{1}{2}} + O(h^3)$ in Eq. (23), one can show that the h^2 terms do not cancel with the corresponding terms in the expansion of the exact solution. As a result, Eqs. (22) and (23) commit a splitting error of size $\|v(t_{n+1}) - y(t_{n+1})\| = O(h^2)$.

In order to find the global order of convergence, we observe that

$$\ldots(\Gamma_{h/2} \circ \Phi_h \circ \Gamma_{h/2}) \circ (\Gamma_{h/2} \circ \Phi_h \circ \Gamma_{h/2}) \circ (\Gamma_{h/2} \circ \Phi_h \circ \Gamma_{h/2}) \tag{24}$$

$$= \ldots(\Gamma_h \circ \Phi_h \circ \Gamma_h \circ \Phi_h \circ \Gamma_h \circ \Phi_h \circ \Gamma_{h/2}). \tag{25}$$

Consequently, the operation $\Gamma_{h/2}$ in the first step can be viewed as providing initial conditions accurate to $O(h^2)$ for the sequence $\ldots\Gamma_h\circ\Phi_h\circ\Gamma_h\circ\Phi_h\circ\Gamma_h\circ\Phi_h$. From the analysis of the one-pass algorithms, the latter sequence is globally first-order convergent, so initial conditions accurate to $O(h^2)$ will preserve the global order. The analysis extends to the variable stepsize case (the details

are omitted). Thus we have proved that the two-pass algorithms are only first-order accurate, in contrast to their second-order accuracy for ODEs.

Note 5 *An alternate approach overcomes order reduction in the two-pass by recasting the DAEs into the equivalent ODEs:*

$$f(x,y)=0 \Longrightarrow f_x\dot{x}+f_y\dot{y}=0 \Longrightarrow \begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}=\begin{pmatrix}(-f_x)^{-1}(f_y)g\\ g\end{pmatrix}. \tag{26}$$

The one-pass and two-pass algorithms for this system of ODEs attain first and second order global orders, respectively. In practice, this technique is rather expensive, as higher derivatives of the right-hand side are involved. In addition, the constraints are not satisfied exactly, and thus may drift over many time steps.

3 Numerical Example

The orders of accuracy derived above are verified through a simple numerical example. Consider the exactly-solvable DAE

$$\begin{pmatrix}0\\ \dot{y}\end{pmatrix}=\begin{pmatrix}x^3-y^2\\ x\end{pmatrix}, \tag{27}$$

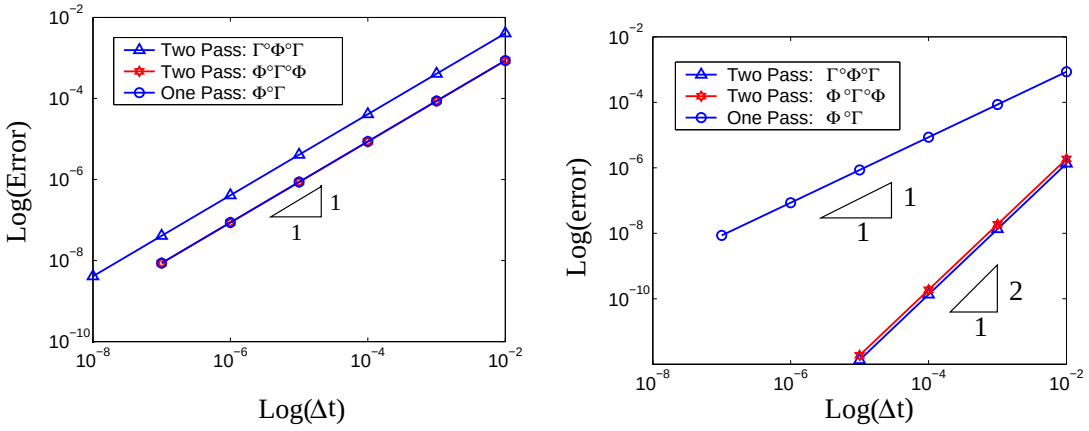
with initial conditions $x_0=1$ and $y_0=1$ at $t_0=0$ satisfying $x_0^3-y_0^2=0$. The equivalent ODE form is

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}=\begin{pmatrix}\frac{2y}{3x}\\ x\end{pmatrix}. \tag{28}$$

The exact solution satisfying the initial conditions is $(x_{ex}(t),y_{ex}(t))=((1+t/3)^2,(1+t/3)^3)$.

A single step of the one-pass splitting, $\Phi_h\circ\Gamma_h$, with exact integration, yields solutions

DAE Split	Equivalent ODE Split
$y_{n+1}=y_n+x_nh$	$y_{n+1}=y_n+x_nh$
$x_{n+1}=y_{n+1}^{\frac{2}{3}}$	$x_{n+1}=(x_n^2+\frac{4h}{3}y_{n+1})^{\frac{1}{2}}$



(a) (b)

Fig. 1. (a) Order reduction in DAE-based splitting vs. (b) no order reduction in ODE-based splitting.

Similarly, the two-pass splitting $\Gamma_{h/2} \circ \Phi_h \circ \Gamma_{h/2}$ yields solutions

DAE Split	Equivalent ODE Split
$y_{n+\frac{1}{2}} = y_n + x_n h/2$	$y_{n+\frac{1}{2}} = y_n + x_n h/2$
$x_{n+1} = y_{n+\frac{1}{2}}^{\frac{2}{3}}$	$x_{n+1} = (x_n^2 + \frac{4h}{3} y_{n+\frac{1}{2}})^{\frac{1}{2}}$
$y_{n+1} = y_{n+\frac{1}{2}} + x_{n+1} h/2$	$y_{n+1} = y_{n+\frac{1}{2}} + x_{n+1} h/2$

Fig. 1 (a) plots the logarithm of the DAE splitting errors versus the logarithm of the uniform stepsize, and exhibits first-order convergence for both one and two-pass algorithms. Fig. 1 (b) shows the errors for the ODE form. First and second-order convergence, agreeing with theory, is obtained for both one- and two-pass schemes.

4 Order Improvement by Deferred Correction

In this section, we employ the deferred correction paradigm to derive a new splitting scheme, and prove its second-order accuracy. The general paradigm of deferred correction is straightforward: Given an approximate solution v to a problem with exact solution y, derive an equation for the error $e = v - y$ and solve it numerically for an approximate error ϵ. The corrected solution $V = v - \epsilon$ is then more accurate than v, and the process can be repeated to generate schemes of arbitrarily high order if the solution is sufficiently smooth. The advantage of this paradigm is that the error equation has the same structure as the original equation, so any convenient low-order method can be used to compute both the original solution v and all subsequent approximate errors ϵ.

4.1 Deferred correction

Consider the general non-autonomous DAE

$$\begin{pmatrix} 0 \\ \dot{y} \end{pmatrix} = \begin{pmatrix} f(t,x,y) \\ g(t,x,y) \end{pmatrix}.$$

We generate a basic solution by the analogue of the simple first-order splitting $\Gamma_h \circ \Phi_h$, which applies the constraint and then the ODE solver. The resulting numerical solution $v(t)$ *exactly* satisfies the ODE

$$\dot{v} = g(t, \varphi(t, v_n), v) \qquad t_n \le t < t_{n+1} \tag{29}$$

where $f(t, \varphi(t,y), y) = 0$ and $v_n = v(t_n)$.

Thus the error $e = v - y$ satisfies the exact error equation

$$\dot{e} = g(t, \varphi(t, v_n), v(t)) - g(t, \varphi(t, v(t) - e(t)), v(t) - e(t)).$$

The first-order splitting scheme replaces $e(t)$ by e_n in the argument of the constraint solver φ only, yielding a second-order splitting scheme composed of a first-order step

$$\dot{v} = g(t, \varphi(t, V_n), v(t)), \qquad v(t_n) = V_n$$

on $t_n \le t < t_{n+1}$, followed by a correction step

$$\dot{\epsilon} = g(t, \varphi(t, V_n), v(t)) - g(t, \varphi(t, v(t)), v(t) - \epsilon(t)), \qquad \epsilon(t_n) = 0 \tag{30}$$

on $t_n \le t < t_{n+1}$. Here the corrected solution is $V(t) = v(t) - \epsilon(t)$ and we are correcting each time step before proceeding to the next. The initial value $\epsilon_n = 0$ has therefore been omitted in the argument of the second g in the correction step (30). The correction step retains the simplicity of the basic splitting scheme, because the correction equation is a pure ODE for the correction: The constraints are imposed only via the known approximate solution v.

Since the solution ϵ of Eq. (30) is a first-order accurate approximation of e and e is itself $O(h)$, we expect $\epsilon = e + O(h^2)$. In the next subsection, we prove that the corrected solution $V(t) = v(t) - \epsilon(t)$ is indeed second-order accurate.

4.2 Convergence Analysis

The convergence proof for the first-order splitting scheme contains the basic idea of the convergence proof for the second-order scheme we have just derived

by deferred correction, so we review it briefly first. The exact solution satisfies the ODE form

$$\dot{y} = g(t, \varphi(t, y), y)$$

while the numerical solution v satisfies

$$\dot{v} = g(t, \varphi(t, v_n), v(t)).$$

By subtraction, the error $e = v - y$ satisfies

$$\begin{aligned}\dot{e} &= g(t, \varphi(t_n, v(t_n)), v) - g(t, \varphi(t, y), y) \\ &= \overline{g_x \varphi_y}(v_n - y) + \overline{g_x \varphi_t}(t_n - t) + \overline{g_y}(v - y) \\ &= -\overline{g_x}\,\overline{f_x^{-1} f_y}(v_n - y) + \overline{g_x \varphi_t}(t_n - t) + \overline{g_y} e.\end{aligned}$$

Here we have denoted differentiation by subscripts and evaluation of the elements of a matrix or vector at possibly different unknown points by an overbar, in accordance with the multivariable mean value theorem [13]. For convenience write $v_n - y = v_n - y_n + y_n - y$, $A = -\overline{g_x}\,\overline{f_x^{-1} f_y}$, $b = \overline{g_x \varphi_t}$ and $C = \overline{g_y}$ to get

$$\dot{e} = Ae_n + A(y_n - y) + b(t_n - t) + Ce.$$

Assume derivative bounds $\|A\| \le \alpha$, $\|b\| \le \beta$ and $\|C\| \le \gamma$ and integrate to get

$$\begin{aligned}\|e(t)\| &= \|e_n + \int_{t_n}^{t} Ae_n ds + \int_{t_n}^{t} A(y_n - y) ds + \int_{t_n}^{t} b(t_n - s) ds + \int_{t_n}^{t} Ce(s) ds\| \\ &\le \|e_n\| + \alpha(t - t_n)(\|e_n\| + \|y_n - y\|) + \frac{\beta}{2}(t - t_n)^2 + \gamma \int_{t_n}^{t} \|e(s)\| ds.\end{aligned} \tag{31}$$

By Gronwall's inequality

$$0 \le u(t) \le a + b \int_0^t u(s) ds \Longrightarrow u(t) \le a \exp[bt]$$

and a Taylor expansion of y, this gives

$$\|e_{n+1}\| \le \exp[(\alpha + \gamma)h] \|e_n\| + \delta h^2$$

where δ bounds $(\alpha \|\dot{y}\| + \beta/2) \exp[\gamma h]$. Iterating this inequality gives

$$\|e_n\| \le \frac{\exp[(\alpha + \gamma)t_n] - 1}{(\alpha + \gamma)} \delta h$$

which proves convergence.

This proof resembles a standard convergence proof for e.g. Euler's method for ODEs [13], with the exception that the usual recurrence inequality, which

bounds the accumulated error at one step in terms of previous errors and local truncation errors, becomes the delay-differential inequality (31). Thus Gronwall's inequality is required, to derive a bound for e_{n+1} in terms of e_n.

The second-order proof is similar. By Taylor expansion, the error $\delta(t) = V(t) - y(t)$ satisfies the exact equation

$$\begin{aligned}\dot{\delta}(t) &= g(t, \varphi(t, v(t)), v(t) - \epsilon(t)) - g(t, \varphi(t, V(t) - \delta(t)), V(t) - \delta(t)) \\ &= A\epsilon(t) + (A + B)\delta(t)\end{aligned}$$

where $A = \overline{g_x \varphi_y}$ and $B = \overline{g_y}$. At the same time, the correction satisfies

$$\dot{\epsilon}(t) = A(V_n - v(t)) + B\epsilon(t)$$

where

$$\|v(t) - V_n\| \leq \int_{t_n}^{t} \|g(s, \varphi(s, V_n), v(s))ds\| \leq hG$$

with G a bound for the maximum of $\|g\|$. Consequently

$$\|\epsilon(t)\| \leq G\alpha h^2 + \beta \int_{t_n}^{t} \|\epsilon(s)\|ds \leq G\alpha h^2 \exp[\beta t]$$

by Gronwall, so

$$\|\delta(t)\| \leq \|\delta_n\| + \int_{t_n}^{t} \alpha \|\epsilon(s)\|ds + (\alpha + \beta) \int_{t_n}^{t} \|\delta(s)\|ds$$

and applying Gronwall again gives

$$\|\delta(t)\| \leq (\|\delta_n\| + G\alpha h^3 \exp[\beta h]) \exp[(\alpha + \beta)h].$$

By iteration, global second-order convergence

$$\|\delta_n\| \leq O(h^2)$$

follows immediately as usual. Thus the second-order splitting based on deferred correction produces a second-order accurate solution V.

A third-order scheme can be constructed by repeating the deferred correction step to find a second-order error and subtracting it. Third-order accuracy can then be proved by a very similar analysis. However, the Picard-like viewpoint of the next subsection permits a simpler proof.

4.3 A Picard-like viewpoint

The deferred correction scheme above computes a first-order solution v and then a correction ϵ, yielding a second-order solution $V = v - \epsilon$. Summing the

original and correction Eqs. (29) and (30) yields a simple second-order scheme for V itself:

$$\dot{V} = g(t, \varphi(t, v(t)), V(t)) \qquad t_n \leq t < t_{n+1}.$$

The constrained variables are simply lagged one iteration behind. Similarly, the jth-order solution v_j (where $v = v_1$ and $V = v_2$) produced by $j-1$ steps of deferred correction satisfies

$$\dot{v_j} = g(t, \varphi(t, v_{j-1}(t)), v_j(t)) \qquad t_n \leq t < t_{n+1}.$$

Using the integral form and Gronwall's inequality as above yields immediately that

$$v_j(t) - y(t) = O(h^{j+1})$$

for all j.

While the Picard-like version of our approach thus yields a simple high-order convergence proof, the deferred correction version above may be more convenient for practical implementation. It produces a natural error estimate for step size adaptation. We also note that all our analysis assumes that both constraints and ODEs are solved exactly in each timestep; the deferred correction formulation implies that we can use a simple first-order scheme such as explicit Euler or linearly implicit Euler without order reduction.

5 Numerical Examples

In this section, we present three more examples. First we repeat the order reduction example, using our deferred correction schemes. Second, we demonstrate that our schemes are more efficient than a standard scheme, for a high-dimensional transistor example with practical applications. Finally, we illustrate the application of our schemes to a stiff system.

5.1 Order confirmation

Our second and third-order deferred correction splitting schemes are applied to the example considered in section 3. The global splitting errors of the 2-norm of the two components at $t = 0.2$, are plotted as a function of stepsize in Fig. 2. It is clear from the plot that the correct orders of convergence are achieved–thus providing a simple verification of the analysis. Note that the deferred correction flows are integrated exactly in Fig. 2.

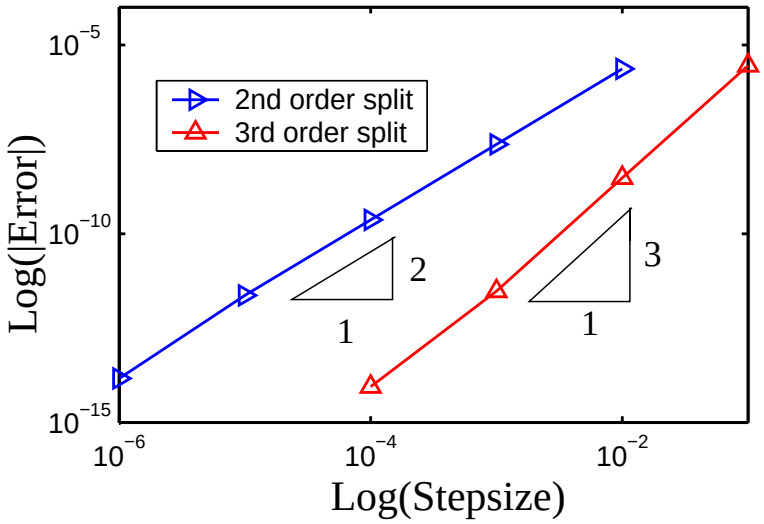


Fig. 2. Global error at $t = 0.2$ for the two dimensional example for second and third-order splitting schemes.

5.2 Transistor Amplifier Example

One of the primary applications for the techniques developed in this paper is in the simulation of electrical circuits. The challenge in simulating circuits with transistors, capacitors and resistors as circuit elements, comes from stiff oscillatory behavior of circuit potentials when subjected to an alternating voltage. The resulting index-1 DAEs are highly nonlinear. Transistor response introduces nonlinearities to alter the response amplitude while capacitors introduce the transient behavior.

As an example we consider the amplifier circuit shown in Fig. 3, which is representative of circuits with oscillatory response. This example is a modification of an amplifier example considered in reference [3]. It is a convenient numerical test case because the total amplification can be controlled through parameters for resistances, capacitances and transistors, while the number of transistors N can be varied systematically. This allows us to demonstrate the efficiency of our splitting schemes for problems with increasing sizes and similar response characteristics. Response frequency increases with problem size, making the circuit more challenging to simulate.

The governing equations are in terms of nodal potentials $U_j^{(i)}$. A linear transformation recasts the circuit equations into a semi-explicit index-1 form, in

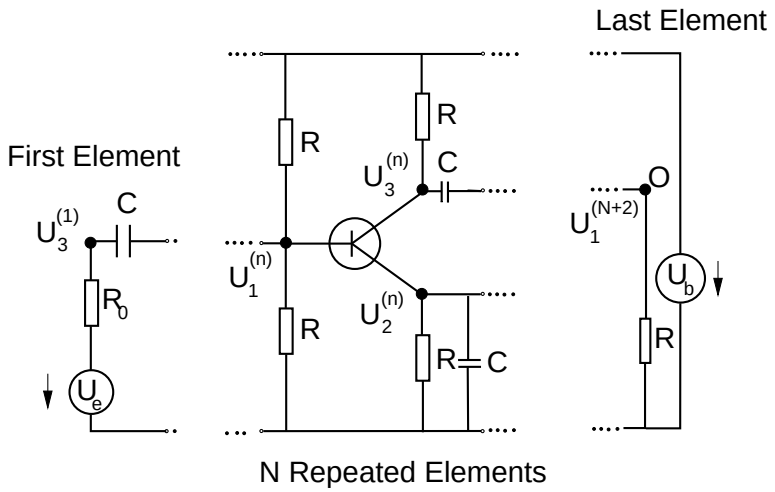


Fig. 3. Schematic of the Transistor Amplifier.

terms of transformed potentials $V_j^{(i)}$[3]. They are:

$$
\begin{aligned}
0 &= \frac{U_e}{R_0} + \frac{U_b}{R} - \frac{V_3^{(1)}}{R_0} - \frac{2}{R}(V_3^{(1)} + V_1^{(2)}) + (\alpha - 1) f(V_3^{(1)} + V_1^{(2)} - V_2^{(2)}) \\
C\dot{V}_1^{(n)} &= \frac{U_b}{R} - \frac{2}{R}(V_3^{(n-1)} + V_1^{(n)}) + (\alpha - 1) f(V_3^{(n-1)} + V_1^{(n)} - V_2^{(n)}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad n = 2, \ldots N+1 \\
C\dot{V}_2^{(n)} &= f(V_3^{(n-1)} + V_1^{(n)} - V_2^{(n)}) - \frac{V_2^{(n)}}{R} \qquad\qquad n = 2, \ldots N+1 \\
0 &= \frac{2U_b - V_3^{(n)}}{R} - \alpha f(V_3^{(n-1)} + V_1^{(n)} - V_2^{(n)}) + \\
&\quad - \frac{2}{R}(V_3^{(n)} + V_1^{(n+1)}) + (\alpha - 1) f(V_3^{(n)} + V_1^{(n+1)} - V_2^{(n+1)}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad n = 2, \ldots N \\
0 &= \frac{U_b - V_3^{(N+1)}}{R} - \alpha f(V_3^{(N)} + V_1^{(N+1)} - V_2^{(N+1)}) - \frac{V_1^{(N+2)} + V_3^{(N+1)}}{R} \\
C\dot{V}_1^{(N+2)} &= -\frac{V_1^{(N+2)} + V_3^{(N+1)}}{R}
\end{aligned}
\tag{32}
$$

These $3N+2$ equations have $3N+2$ unknown voltages $V_j^{(i)}$. Consistent initial conditions for this system of DAEs in terms of the $3N+2$ voltages are: $V_3^{(1)}(0) = 0$, $V_3^{(n)}(0) = U_b \quad n = 2, \ldots, N+1$, $V_1^{(n)}(0) = U_b/2 - V_3^{(n-1)}$, $V_2^{(n)}(0) = U_b/2 \quad n = 2, \ldots, N+1$, and $V_1^{(N+2)}(0) = -U_b$. The variables $V_3^{(n)}$, $n = 2, \ldots, N+1$, and $V_1^{(1)}$ are the algebraic variables, while the rest are ODE variables. The nonlinear transistor function f is given by $f(v) = \beta[\exp((v/U_f) - 1]$. Parameter values are $U_b = 6$, $\alpha = 0.99$, $\beta = 10^{-6}$, $R_0 = 10^3$, $R = 9 \times 10^3$ and, $C = 10^{-6}$. We consider $N = 100,\ 400,\ 700,\ 1000$ for our test suite of increasing problem sizes, with $U_f = 2.7 \times 10^{-1}$ for $N = 1000$ and $U_f = 2.6 \times 10^{-1}$ for the rest. A periodic input voltage signal $U_e(t) =$

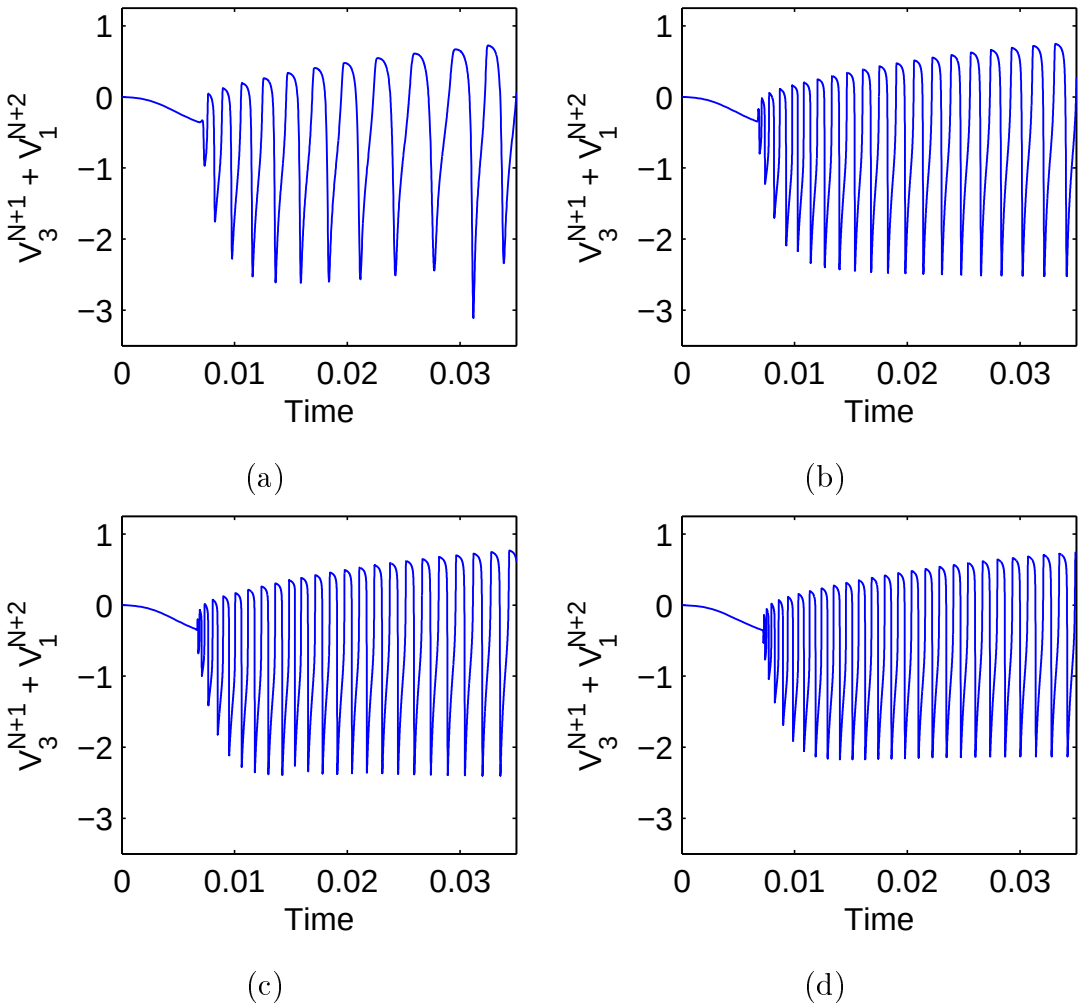


Fig. 4. Amplified voltage $U_1^{(N+2)} = V_3^{(N+1)} + V_1^{(N+2)}$ for the transistor amplifier: (a) $N = 100$, (b) $N = 400$, (c) $N = 700$ and (d) $N = 1000$.

$0.1\sin(200\pi t)$ is chosen and the error in the amplified nodal output voltage $U_1^{(N+2)} = V_3^{(N+1)} + V_1^{(N+2)}$ (shown at node O in Fig. 3) is measured.

Fig. 4 displays the output voltage at the end node (marked O in Fig. 3) for $N = 100,\ 400,\ 700,\ 1000$, in the interval $t \in [0,\ 0.035]$. The response frequency increases slowly with N, making the already oscillatory equations stiffer and harder to solve.

Fig. 5 plots errors in $V_3^{(N+1)} + V_1^{(N+2)}$ vs. CPU time (on a 3.1GHz Intel Pentium processor with 512MB RAM), on a log-log scale. The errors are plotted at $T_{max} = 0.20$ for $N = 100$, $T_{max} = 0.10$ for $N = 400$, $T_{max} = 0.07$ for $N = 700$, and $T_{max} = 0.035$ for $N = 1000$; these values of T_{max} result in CPU times of the same order of magnitude for all the cases.

We compare our second- and third-order splitting schemes with the highly

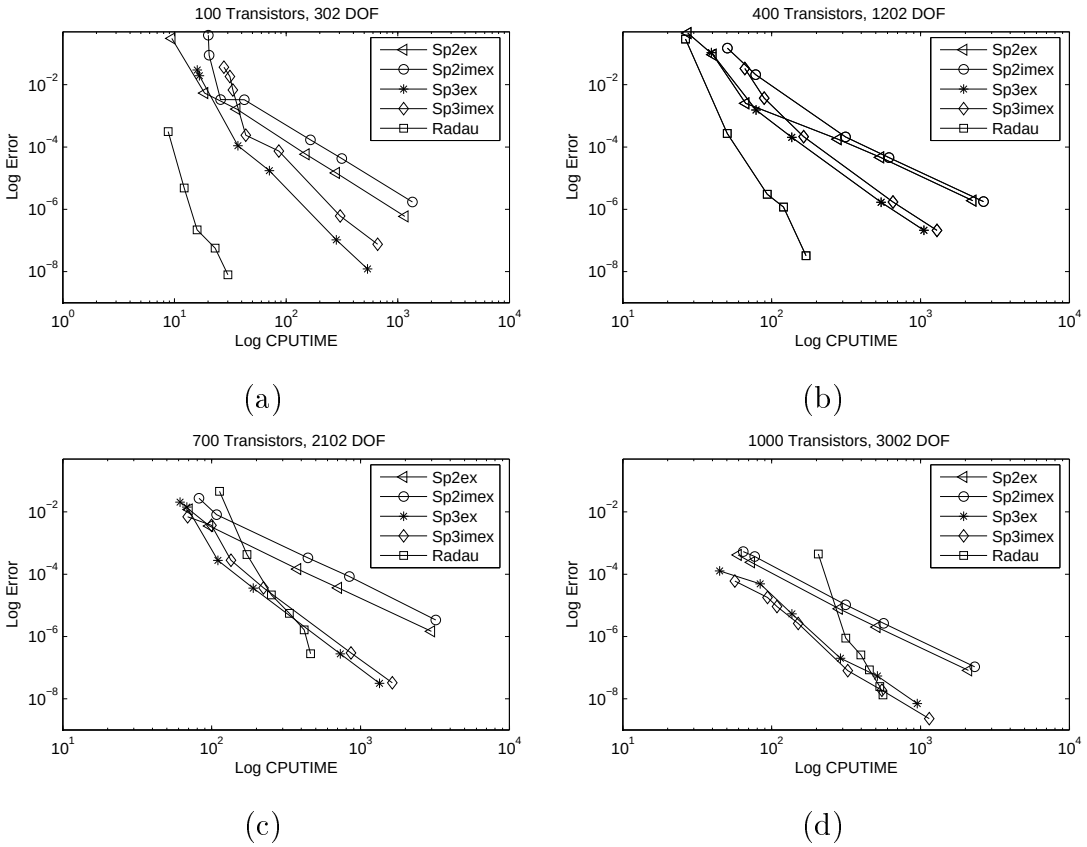


Fig. 5. Global error in output voltage ($V_3^{(N+1)} + V_1^{(N+2)}$) for the transistor-amplifier example with (a) $N = 100$, (b) $N = 400$, (c) $N = 700$ and (d) $N = 1000$.

optimized fifth-order implicit RK code RADAU5 of [14]. The ODEs are integrated by forward Euler (marked Sp2ex and Sp3ex in Fig. 5), and by linearized implicit Euler for the uncorrected ODEs (29), coupled with forward Euler for the error equation (30) (marked Sp2imex and Sp3imex in Fig. 5). We have used constant stepsizes for our simulations. RADAU5 experienced frequent and repeated stepsize failures for tolerances less than 10^{-9} (probably due to ill-conditioning of the Jacobian of the full system of equations). Thus the error is calculated by comparison with our third order splitting solution Sp3imex, accurate to 10-11 digits. The individual Jacobians of the constraints and the ODEs were well-conditioned, so splitting schemes could obtain close to full double precision accuracy.

Fig. 5 clearly demonstrates the main advantage of splitting: the decoupling of algebraic and differential equations. For $N = 100$, RADAU5 outperforms all our split implementations. At $N = 400$, with 1202 unknowns, RADAU5 still performs better than our split implementations, although the margin has decreased. At $N = 700$, with 2102 unknowns, the error-CPU curve of RADAU5 crosses those of the third-order splitting scheme at errors ranging from 10^{-4} to 10^{-5} and second-order splitting scheme at errors ranging from 10^{-2} to

10^{-3}. Thus for relaxed error tolerances, splitting schemes clearly outperform RADAU5. When $N = 1000$, Sp3imex outperforms RADAU5 for all tolerances greater then 10^{-8}, while RADAU5 terminated early due to repeated stepsize failures for most tolerances smaller than 10^{-8}. Hence over the entire range of tolerances where RADAU5 can provide a solution to the 1000 transistor problem, our third-order schemes outperformed RADAU5. For engineering accuracy of 4 digits, our third-order schemes are about twice as fast as the state-of-the-art, highly optimized, adaptive stepsize, fifth-order RADAU5 scheme. These results clearly demonstrate the efficiency gained by using our splitting schemes for large index-1 DAEs from circuit simulation.

5.3 Pendulum Example

As a final example, a stiff damped pendulum is considered. This example is interesting because the first-order splitting $\Gamma_h \circ \Phi_h$ is also dissipative. Thus we investigate stability and accuracy of the splitting schemes for large stepsizes. The governing equations presented in Ref. [3] are modified to include damping and an input excitation. The index-1 system is given by:

$$\begin{aligned} \dot{p} &= u \\ \dot{q} &= v \\ m\dot{u} &= -p\lambda - cu - f(t) \\ m\dot{v} &= -q\lambda - g - cv \\ 0 &= m(u^2 + v^2) - gq - l^2\lambda - pcu - qcv \end{aligned} \tag{33}$$

Here, values of mass $m = 5 \times 10^{-5}$, damping $c = 5 \times 10^{-3}$, length $l = 1$, acceleration due to gravity $g = 1$ are chosen. The tension λ in the pendulum rod is an algebraic variable, while the position coordinates p and q and their time derivatives u and v are ODE variables. A periodic input excitation $f(t) = 0.2\sin(0.75\pi t)$ is chosen. The pendulum system is dissipative due to damping. Since the tension λ is always positive or zero, replacing the current $\lambda(t)$ with $\lambda(t_n) \geq 0$ for $t \in [t_n, t_{n+1}]$ still renders the system dissipative. This replacement exactly corresponds to the first-order split $\Gamma_h \circ \Phi_h$. The exact solution has an initial transient phase followed by a periodic steady state solution. The stiffness ratio for the pendulum system is $\mathcal{O}(10^2)$ leading to an initial transient phase for $t \in [0,\ 0.4]$.

Using backward Euler, which is dissipative for the ODE system, one obtains a dissipative splitting scheme with no stepsize restrictions for stability. On the other hand, numerical experiments indicate that the second and third order deferred correction schemes are only conditionally stable. For the present case, the maximum fixed stepsize for the second-order scheme is 9×10^{-3}, and requires a CPU time of 0.13 to achieve an error of 1.45×10^{-5} at $t = 10$ in

component p. The third-order scheme has similar stepsize restrictions. If coarse accuracy is required, the first-order splitting scheme performs very well. For example, with $h = 10^{-1}$, one can obtain a solution with an error of 5×10^{-3} in component p in 10^{-2} CPU seconds at $t = 10$ even though the solution is grossly inaccurate in the transient phase. Thus for high stiffness ratios and situations where only coarse accuracy is required, a first-order dissipative splitting is recommended. If a dissipative split is not possible, fully implicit methods remain the most efficient approach to highly stiff DAE systems.

Finally, we remark that the following stepsize sequences were chosen for the first-order and the second-order splitting scheme, to maintain a constant error of $\mathcal{O}(10^{-5})$ for $t \in [0,\ 10]$. In the second-order splitting case, the maximum stepsize is close to the stability limit.

$$\text{First order implicit split: } h = \begin{cases} 10^{-6} & 0 \le t \le 0.12 \\ 10^{-6} + \frac{10^{-2}-10^{-6}}{0.88}(t - 0.12) & 0.12 \le t \le 1 \\ 10^{-2} & 1 \le t \le 10 \end{cases} \tag{34}$$

$$\text{Second Order Split: } h = \begin{cases} 1.25 \times 10^{-4} & 0 \le t \le 0.08 \\ 1.25 \times 10^{-4} + \frac{0.007875}{0.92}(t - 0.0.08) & 0.08 \le t \le 1 \\ 8 \times 10^{-3} & 1 \le t \le 10 \end{cases} \tag{35}$$

Using these values for the stepsize, the CPU time for second-order splitting is 2×10^{-2} seconds which is about 50 times smaller than the CPU time of first-order splitting. Higher order splitting schemes are more efficient when accuracy decides the stepsizes.

6 Conclusions

We have analyzed and demonstrated first-order convergence of standard ODE splitting schemes for semi-explicit index-1 DAEs, and employed a deferred correction paradigm to obtain efficient higher-order accurate operator splitting schemes for such DAEs.

Numerical examples exhibit the expected order reduction for the standard two-pass ODE splitting schemes and the theoretical orders of accuracy of our new deferred correction schemes, and show our schemes to be efficient in work and storage. While fully implicit RK methods like RADAU5 are useful for small problems, they become prohibitively expensive for large problems. Our analysis yields efficient methods for large problems where high-order splitting of constraints from differential equations can be highly effective.

References

[1] R. I. McLachlan, G. R. W. Quispel, Splitting methods, Acta Numerica 11 (2002) 341–434.

[2] K. E. Brenan, S. L. Campbell, L. R. Petzold, Numerical solution of initial-value problems in differential-algebraic equations, Elsevier Science Pub. Co., North-Holland, New York, 1989.

[3] E. Hairer, C. Lubich, M. Roche, Lecture notes in mathematics, in: The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods, Vol. 1409, Springer-Verlag, 1989.

[4] F. Armero, Formulation and finite element implementation of a multiplicative model of coupled pore-plasticity at finite strains under fully saturated conditions, Computer Methods in applied Mechanics and Engineering 171 (1999) 205–241.

[5] F. Armero, J. C. Simo, A new unconditionally stable fractional step method for nonlinear coupled thermomechanical problems, International Journal for Numerical Methods in Engineering 35 (1992) 737–756.

[6] G. Strang, On construction and comparison of difference schemes, SIAM Journal on Numerical Analysis 5 (1968) 506–517.

[7] O. C. Zienkiewicz, D. K. Paul, A. H. C. Chan, Unconditionally stable staggered solution procedure for soil pore fluid interaction problems, International Journal for Numerical Methods in Engineering 26 (1988) 1039–1055.

[8] J. M. Sanz-Serna, Geometric integration, in: The state of the art in Numerical Analysis, Clarendon Press, Oxford, 1997, pp. 121–143.

[9] A. Pazy, Semigroups of linear operators and applications to partial differential equations, Vol. 44 of Applied Mathematical sciences, Springer-Verlag, New York, 1983.

[10] E. Hairer, S. P. Norsett, G. Wanner, Solving ordinary differential equations I: Nonstiff problems, Springer-Verlag, New York, 1993.

[11] B. Sportisse, An analysis of operator splitting techniques in the stiff case, Journal of Computational Physics 161 (2000) 140–168.

[12] A. C. Chorin, T. J. R. Hughes, M. F. McCracken, J. E. Marsden, Product formulas and numerical algorithms, Communications on Pure and Applied Mathematics 31 (1978) 205–256.

[13] J. D. Lambert, Numerical Methods for Ordinary Differential Equations: The initial value problem, John Wiley and Sons, New York, 1993.

[14] E. Hairer, G. Wanner, Solving ordinary differential equations II: Stiff and Differential-Algebraic problems, Springer-Verlag, New York, 1993.