



Time Reparametrization and Event Location for Discontinuous Differential Algebraic Equations

L. Lopez¹ · S. Maset²

Received: 12 July 2023 / Revised: 4 April 2024 / Accepted: 12 April 2024 /
Published online: 21 May 2024
© The Author(s) 2024

Abstract

In this paper, we consider numerical methods for the event location of differential algebraic equations. The event corresponds to cross a discontinuity surface, beyond which another differential algebraic equation holds. The methods are based on a particular change of the independent variable time, called time reparametrization or time transformation, reducing the equation to another equation where the event time is known in advance. From a numerical point of view, these methods never cross the discontinuity surface and reach it in a fixed number of steps. The methods works also for differential algebraic equations of index higher than one.

Keywords Discontinuous differential algebraic equations · Event location · Time reparametrization · Diagonally implicit stiffly accurate Runge-Kutta methods

1 Introduction

This paper deals with differential algebraic equations (DAEs) with a discontinuity surface, where, during the numerical integration of the equation, it is required the accurate computation of the *event point* corresponding to reaching the discontinuity surface. Several real systems may be modeled by DAEs of this type, see for instance the applications in Chemical Engineering and Electrical Systems [1, 3, 21, 23]. In recent years, a growing interest has been observed concerning the theoretical aspects (see for example [10, 19, 20]), together with the numerical questions that arise in such DAEs (see for example [10, 15, 16, 19, 20, 22]). DAEs with a discontinuity surface are called in several way: *non-smooth DAEs*, *hybrid DAEs*, *discontinuous DAEs*, *DAEs of Filippov type*. Here, we use the term Discontinuous DAEs (DDAEs).

✉ S. Maset
maset@units.it

L. Lopez
luciano.lopez@uniba.it

¹ Dipartimento di Matematica, Università di Bari, Bari, Italy

² Dipartimento di Matematica, Informatica e Geoscienze, Università di Trieste, Trieste, Italy

In the following, we consider a DAE of the form:

$$\begin{cases} y'(t) = f(y(t), z(t)), & t \geq t_0, \\ g(y(t), z(t)) = 0, & t \geq t_0, \\ (y(t_0), z(t_0)) = (y_0, z_0), \end{cases} \tag{1}$$

where $y(t) \in \mathbb{R}^{d_1}$, $z(t) \in \mathbb{R}^{d_2}$ and $f : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}^{d_1}$ and $g : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}^{d_2}$ are sufficiently smooth functions. The initial value $(y_0, z_0) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$ is assumed to be consistent, i.e. $g(y_0, z_0) = 0$.

Moreover, we consider the state space $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$ as partitioned in the three subsets:

$$\begin{aligned} S^- &= \{(y, z) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : h(y, z) < 0\} \\ \Sigma &= \{(y, z) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : h(y, z) = 0\} \\ S^+ &= \{(y, z) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : h(y, z) > 0\}. \end{aligned}$$

where $h : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}$ is a sufficiently smooth function. The DAE (1) is assumed to hold in $S^- \cup \Sigma$ and the initial value (y_0, z_0) is assumed to belong to S^- , while, in general, in $S^+ \cup \Sigma$ a different DAE is assigned. So, the DAE (1) is a DDAE with discontinuity surface Σ .

We suppose that the DAE (1) has a unique solution (y, z) and this solution meets the discontinuity surface Σ at a certain time $t^* > 0$: we have $(y(t), z(t)) \in S^-$ for $t \in [t_0, t^*)$ and $(y(t^*), z(t^*)) \in \Sigma$, i.e.

$$h(y(t^*), z(t^*)) = 0.$$

The time t^* is called the *event time* and the state $(y(t^*), z(t^*))$ is called the *event point*. The *event location* is the determination of the event time t^* and the event point $(y(t^*), z(t^*))$.

The accurate evaluation of the event point is an important task in the integration of DDAEs, because, once evaluated the event point $(y(t^*), z(t^*))$, one may decided what to do next. In general, after the event point, the solution can cross or slide on the discontinuity surface (see for instance [7]). In [18], it is studied how to compute the event time and the event point by means of the standard numerical methods for DAEs. In this paper, we compute the event time and the event point by means of a time reparametrization producing a new DAE where the event time is known a priori.

It is important for the event location to use numerical methods satisfying the following three properties:

- (a) the numerical event point $(y_\tau(t^*), z_\tau(t^*))$ lies on the discontinuity surface, i.e. $h(y_\tau(t^*), z_\tau(t^*)) = 0$;
- (b) the numerical event point $(y_\tau(t^*), z_\tau(t^*))$ is consistent, i.e. $g(y_\tau(t^*), z_\tau(t^*)) = 0$;
- (c) the numerical method is *one-side*, i.e. up to reaching the discontinuity surface the method uses values of f and g only at points of $S^- \cup \Sigma$.

By using suitable numerical methods in the integration of the new DAE obtained by the time reparametrization, all the properties (a), (b) and (c) above can be satisfied. On the other hand, this cannot be achieved by standard numerical methods for DAEs.

Moreover, regarding the property (c), it is important that an one-side method can reach the discontinuity surface in an arbitrary number of steps fixed a priori, without needing to reduce the stepsize as one approaches the discontinuity surface. This is obtained in the numerical integration of the new DAE.

It is worthwhile to remark that the time reparametrization technique works also for DAEs of index higher than one.

The plan of the present paper is as follows. We start by recalling the standard numerical methods for DAEs and we show how to determine the event time and the event point by these methods. Then, we discuss one-side numerical methods and show the difficulties that the standard numerical methods encounter in satisfying the one-side property. After this, we describe the time reparametrization technique and then the related numerical methods fulfilling the properties (a), (b) and (c) above. Finally, some numerical tests conclude our paper.

2 Background on Standard Numerical Methods and Event Location for DAEs

In this section, we recall the standard numerical methods used for integrating DAEs, namely semi-implicit methods, implicit RK methods and Rosenbrock methods (for reference see [11, 12]). Moreover, we show how the event location is implemented for these methods (for reference see [18]).

The methods are applied over a mesh

$$t_0 < t_1 < t_2 < \dots$$

of stepsizes

$$\tau_{n+1} = t_{n+1} - t_n, \quad n = 0, 1, 2, \dots$$

In the following, y_n and z_n denote the numerical approximations of the differential and algebraic variables y and z , respectively, at the mesh point t_n .

During the numerical event location, the numerical integration continues up to mesh points t_{n^*} and t_{n^*+1} such that

$$h(y_{n^*}, z_{n^*}) < 0 \text{ and } h(y_{n^*+1}, z_{n^*+1}) > 0.$$

After the individuation of such mesh points, the numerical event location determines the numerical event time t_τ^* and numerical event point (y_τ^*, z_τ^*) .

2.1 Semi-implicit Methods

By using the usual notations for RK methods, a *semi-implicit method* is defined by the scheme

$$y_{n+1} = y_n + \tau_{n+1} \sum_{i=1}^v b_i f(y_{ni}, z_{ni}) \tag{2}$$

$$g(y_{n+1}, z_{n+1}) = 0, \tag{3}$$

where the stage values $(y_{ni}, z_{ni}), i = 1, \dots, v$, are successively obtained by

$$y_{ni} = y_n + \tau_{n+1} \sum_{j=1}^{i-1} a_{ij} f(y_{nj}, z_{nj}) \tag{4}$$

$$g(y_{ni}, z_{ni}) = 0. \tag{5}$$

Here, b_i and a_{ij} are weights and coefficients, respectively, of a ν -stage explicit RK method. Observe that the implicitness of this scheme consists in solving at each step the $\nu + 1$ non-linear equations (3)-(5) of dimension d_2 . By construction (see (3)), the numerical solution (y_n, z_n) is consistent, i.e.

$$g(y_n, z_n) = 0, \quad n = 0, 1, 2, \dots$$

The numerical event time t_τ^* and event point (y_τ^*, z_τ^*) are obtained by solving the nonlinear system in the unknowns t_τ^* and z_τ^* :

$$\begin{cases} g(y_\tau^*, z_\tau^*) = 0 \\ h(y_\tau^*, z_\tau^*) = 0, \end{cases} \tag{6}$$

where

$$y_\tau^* = \eta(t_\tau^*) = y_{n^*} + \tau_{n^*+1} \sum_{i=1}^{\nu} b_i \left(\frac{t_\tau^* - t_{n^*}}{\tau_{n^*+1}} \right) f(y_{n^*i}, z_{n^*i}),$$

with

$$\eta(t_n + \theta \tau_{n+1}) = y_n + \tau_{n+1} \sum_{i=1}^{\nu} b_i(\theta) f(y_{ni}, z_{ni}), \quad \theta \in [0, 1],$$

a continuous extension of the explicit RK method. By construction (see (6)), the numerical event point (y_τ^*, z_τ^*) lies on the discontinuity surface (point a) in the introduction) and it is consistent (point b) in the introduction). The numerical event location requires the solution of the non-linear system (6) of dimension $d_2 + 1$. The computational cost of solving such a system is a fraction of the computational cost of a step of the semi-implicit method, where $\nu + 1$ systems of dimension d_2 need to be solved.

2.2 Implicit RK Methods

Implicit RK methods are given by

$$\begin{aligned} y_{n+1} &= y_n + \tau_{n+1} \sum_{i=1}^{\nu} b_i f(y_{ni}, z_{ni}) \\ z_{n+1} &= \left(1 - \sum_{i,j=1}^{\nu} b_i \omega_{ij} \right) z_n + \sum_{i,j=1}^{\nu} b_i \omega_{ij} z_{nj}, \end{aligned} \tag{7}$$

where the stage values $(y_{ni}, z_{ni}), i = 1, \dots, \nu$, are obtained by solving the non-linear system of equations

$$\begin{aligned} y_{ni} &= y_n + \tau_{n+1} \sum_{j=1}^{\nu} a_{ij} f(y_{nj}, z_{nj}), \quad i = 1, \dots, \nu, \\ g(y_{ni}, z_{ni}) &= 0, \quad i = 1, \dots, \nu. \end{aligned} \tag{8}$$

Here, b_i and a_{ij} are weights and coefficients, respectively, of a ν -stage implicit RK method and ω_{ij} are the entries of the inverse of the RK matrix (a_{ij}) . At each step of this implicit scheme, the non-linear system (8) of dimension $\nu(d_1 + d_2)$ has to be solved. If the implicit RK method is *stiffly accurate*, i.e.

$$a_{\nu i} = b_i, \quad i = 1, \dots, \nu, \tag{9}$$

then the numerical solution (y_n, z_n) is consistent, since $(y_{n\nu}, z_{n\nu}) = (y_{n+1}, z_{n+1})$ holds.

The numerical event time t_τ^* and event point (y_τ^*, z_τ^*) are obtained by

$$\begin{aligned} t_\tau^* &= t_n^* + \tau^* \\ y_\tau^* &= y_n^* + \tau^* \sum_{i=1}^{\nu} b_i f(y_i^*, z_i^*) \\ z_\tau^* &= \left(1 - \sum_{i,j=1}^{\nu} b_i \omega_{ij} \right) z_n^* + \sum_{i,j=1}^{\nu} b_i \omega_{ij} z_j^*, \end{aligned} \tag{10}$$

where the stepsize τ^* and the stage values $(y_i^*, z_i^*), i = 1, \dots, \nu$, satisfy

$$\begin{aligned} y_i^* &= y_n^* + \tau^* \sum_{j=1}^{\nu} a_{ij} f(y_j^*, z_j^*), \quad i = 1, \dots, \nu, \\ g(y_i^*, z_i^*) &= 0, \quad i = 1, \dots, \nu, \\ h(y_\tau^*, z_\tau^*) &= 0. \end{aligned} \tag{11}$$

As in case of semi-implicit methods, the numerical event point (y_τ^*, z_τ^*) lies on the discontinuity surface (point a) in the introduction). Moreover, if the implicit RK method is stiffly accurate, then the numerical event point is consistent (point b) in the introduction). The numerical event location requires to solve the non-linear system (11) of dimension $\nu(d_1 + d_2) + 1$. So, the computational cost of solving such a non-linear system is essentially the same as the computational cost of a step of the implicit RK method, where a system of dimension $\nu(d_1 + d_2)$ needs to be solved.

2.3 Rosenbrock Methods

Rosenbrock methods read

$$\begin{aligned} y_{n+1} &= y_n + \sum_{i=1}^{\nu} b_i l_{ni} \\ z_{n+1} &= z_n + \sum_{i=1}^{\nu} b_i k_{ni}, \end{aligned} \tag{12}$$

where $(l_{ni}, k_{ni}), i = 1, \dots, \nu$, are successively obtained by solving the linear system

$$\begin{aligned} &\begin{bmatrix} I - \tau_{n+1}\gamma_{ii} f_y^n & -\tau_{n+1}\gamma_{ii} f_z^n \\ -\tau_{n+1}\gamma_{ii} g_y^n & -\tau_{n+1}\gamma_{ii} g_z^n \end{bmatrix} \begin{bmatrix} l_{ni} \\ k_{ni} \end{bmatrix} \\ &= \begin{bmatrix} \tau_{n+1}f(y_{ni}, z_{ni}) + \tau_{n+1} \sum_{j=1}^{i-1} \gamma_{ij} (f_y^n l_{nj} + f_z^n k_{nj}) \\ \tau_{n+1}g(y_{ni}, z_{ni}) + \tau_{n+1} \sum_{j=1}^{i-1} \gamma_{ij} (g_y^n l_{nj} + g_z^n k_{nj}) \end{bmatrix} \end{aligned} \tag{13}$$

with $(y_{ni}, z_{ni}), i = 1, \dots, \nu$, on the right-hand side successively and explicitly given by

$$\begin{aligned} y_{ni} &= y_n + \sum_{j=1}^{i-1} a_{ij} l_{nj} \\ z_{ni} &= z_n + \sum_{j=1}^{i-1} a_{ij} k_{nj}. \end{aligned} \tag{14}$$

Here, b_i, a_{ij} and γ_{ij} are weights and coefficients of a ν -stage Rosenbrock method and f_y^n, f_z^n, g_y^n and g_z^n denote the jacobian matrices of f and g at (y_n, z_n) . The implicitness of this method consists in solving at each step ν linear systems of dimension $d_1 + d_2$. In general, the numerical solution (y_n, z_n) is not consistent.

The numerical event time t_τ^* and event point (y_τ^*, z_τ^*) are obtained by solving the equation

$$h(y_\tau^*, z_\tau^*) = 0, \tag{15}$$

where

$$\begin{aligned} y_\tau^* &= \eta(t_\tau^*) = y_{n^*} + \sum_{i=1}^{\nu} b_i \left(\frac{t_\tau^* - t_{n^*}}{\tau_{n^*+1}} \right) l_{ni} \\ z_\tau^* &= \mu(t_\tau^*) = z_{n^*} + \sum_{i=1}^{\nu} b_i \left(\frac{t_\tau^* - t_{n^*}}{\tau_{n^*+1}} \right) k_{ni}, \end{aligned}$$

with

$$\begin{aligned} \eta(t_n + \theta\tau) &= y_n + \sum_{i=1}^{\nu} b_i(\theta) l_{ni} \\ \mu(t_n + \theta\tau) &= z_n + \sum_{i=1}^{\nu} b_i(\theta) k_{ni}, \quad \theta \in [0, 1], \end{aligned}$$

a continuous extension of the Rosenbrock method. As in case of semi-implicit methods and implicit RK methods, the event point (y_τ^*, z_τ^*) belongs to the discontinuity surface by construction (point a) in the introduction). But, unlike semi-implicit methods and stiffly accurate implicit RK methods, it is not consistent in general (point b) in the introduction). The equation (15) is a scalar non-linear equation in the unknown numerical event time t_τ^* . The cost, for solving it, is a small fraction of the cost of a step of the Rosenbrock method, where ν linear system of dimension $d_1 + d_2$ need to be solved.

3 One-side Methods

There are DDAEs where the functions f and g in (1) are defined only in $S^- \cup \Sigma$ and not in S^+ (see the example of DDAE given in [21]). In such situations, one needs to integrate the DAE (1) by a *one-side* numerical method, namely a method where the values of f and g are evaluated only at points of $S^- \cup \Sigma$, while values of f and g at points of S^+ cannot be computed.

The one-sidedness is an important property that numerical methods for DDAEs have to satisfy (see point c) in the introduction).

For Discontinuous Ordinary Differential Equations (DODEs), numerical methods with this property based on one-step and multistep schemes have been developed in [6, 8]. However, such schemes can become expensive because they need to adapt the stepsize in order to approach the discontinuity surface. On the other hand, for DODEs, the methods studied in [9, 17] and based on a transformation of the time variable are one-side and they can reach the discontinuity surface in a number of steps fixed a priori, without needing to adapt the stepsize. In this paper, we propose for DDAEs similar one-side methods based on a time transformation.

Now, we briefly show the difficulties when we ask to be one-side to the standard numerical methods for DAEs. Here, we are thinking about methods that adapt the stepsize in approaching the discontinuity surface in order to be one-side.

In case of semi-implicit methods (2), (3), (4), (5), the one-side property requires

$$(y_{ni}, z_{ni}) \in S^- \cup \Sigma, \quad n = 0, 1, \dots, n^* \text{ and } i = 1, \dots, \nu,$$

and

$$(y_{n+1}, z_{n+1}) \in S^- \cup \Sigma, \quad n = 0, 1, \dots, n^*.$$

This cannot be obtained, since $(y_{n^*+1}, z_{n^*+1}) \in S^+$. However, in the particular situation where $h(y, z) = h(y)$, i.e h is a function of the sole variable y , the one-side property could be obtained first by checking for $h(y_{n+1}) > 0$ and then, only in the situation where $h(y_{n+1}) \leq 0$, to compute z_{n+1} by $g(y_{n+1}, z_{n+1}) = 0$.

In case of implicit RK methods (7), (8), the one-side property requires

$$(y_{ni}, z_{ni}) \in S^- \cup \Sigma, \quad n = 0, 1, \dots, n^* \text{ and } i = 1, \dots, \nu.$$

This cannot be achieved if a stiffly accurate implicit RK method is used to obtain a consistent numerical solution, since $(y_{n^*}, z_{n^*}) = (y_{n^*+1}, z_{n^*+1}) \in S^+$ for $n = n^*$. However, if the numerical integration up to t_{n^*+1} is accomplished by a non-stiffly-accurate method and the stiffly accurate method is used only in the event location (10), (11), the one-side property could be obtained.

In case of Rosenbrock methods (12), (13), (14), the one-side property require

$$(y_{ni}, z_{ni}) \in S^- \cup \Sigma, \quad n = 0, 1, \dots, n^* \text{ and } i = 1, \dots, \nu,$$

and

$$(y_n, z_n) \in S^- \cup \Sigma, \quad n = 0, 1, \dots, n^*.$$

This shows that techniques adapting the step in order to obtain a one-side method can be used for Rosenbrock methods. However, recall that these methods have the drawback of providing an event point which is not consistent in general (point b) in the introduction).

By summarizing, we can say that the numerical event locations by the three standard numerical methods presented above satisfy the property (a) in the introduction, but they fail to satisfy simultaneously the other two properties (b) and (c). On the other hand, the three properties (a), (b) and (c) can be satisfied simultaneously by numerically integrating the DAE obtained from the time transformation presented in the next section. Moreover, by this numerical integration, the discontinuity surface can be reached in an arbitrary number of steps fixed a priori.

4 Reparametrization of the Time

We present another approach to the event location for DDAEs, different from the use of standard numerical methods, called *time reparametrization* or *time transformation*. This approach has been introduced in [9] and [17] for DODEs and it has been also used in different contexts (see for instance [2, 4, 5, 13]).

We assume to have

$$\frac{d}{dt}h(y(t), z(t)) > 0, \quad t \in [t_0, t^*],$$

which holds sufficiently close to the event time. This means that the time reparametrization should be used when the solution is definitely approaching the discontinuity surface.

4.1 The Time Transformation and the *s*-time DAE

Recall that t^* is the event time for the DAE (1). Fix $s_0 < 0$ and let

$$\alpha : [s_0, 0] \rightarrow [t_0, t^*]$$

be a C^1 function such that $\alpha(s_0) = t_0$ and $\alpha(0) = t^*$. We call α a *time transformation*.

Now, we consider in the DAE (1) the *time reparametrization* (change of variable)

$$t = \alpha(s)$$

and we set

$$(Y(s), Z(s)) := (y(\alpha(s)), z(\alpha(s))), \quad s \in [s_0, 0].$$

Observe that Y and Z are, respectively, the differential and algebraic variables in the new s -time, whereas y and z are the differential and algebraic variables in the old t -time.

The time transformation α is chosen in order to have

$$h(y(\alpha(s)), z(\alpha(s))) = h(Y(s), Z(s)) = \kappa(s), \quad s \in [s_0, 0], \tag{16}$$

where

$$\kappa : [s_0, 0] \rightarrow [h(y_0, z_0), 0]$$

is a given C^1 strictly increasing function with $\kappa(s_0) = h(y_0, z_0) < 0$ and $\kappa(0) = 0$. In this manner, in advance we prescribe in the s -time how the negative values of the function h have to increase in order to reach zero at the event time. The simplest choice for $\kappa(s)$ is $\kappa(s) = s$ and then the initial s -time is $s_0 = h(y_0, z_0)$. Other forms of $\kappa(s)$ are possible: for example, $\kappa(s) = s^m$ with $m \geq 2$ is used in [17].

The original t -time DAE (1) is transformed in a new s -time DAE where the time-transformation α is one of the unknowns.

Theorem 1 *The functions Y and Z and the time transformation α satisfy*

$$\left\{ \begin{array}{l} Y'(s) = \alpha'(s) f(Y(s), Z(s)), \quad s \in [s_0, 0], \\ g(Y(s), Z(s)) = 0, \quad s \in [s_0, 0], \\ h(Y(s), Z(s)) = \kappa(s), \quad s \in [s_0, 0], \\ (Y(s_0), Z(s_0), \alpha(s_0)) = (y_0, z_0, t_0). \end{array} \right. \tag{17}$$

Proof For $s \in [s_0, 0]$, we have

$$\begin{aligned} Y'(s) &= \alpha'(s) y'(\alpha(s)) = \alpha'(s) f(y(\alpha(s)), z(\alpha(s))) \\ &= \alpha'(s) f(Y(s), Z(s)) \end{aligned}$$

and

$$0 = g(y(\alpha(s)), z(\alpha(s))) = g(Y(s), Z(s)).$$

Moreover, (16) holds, the initial values for Y and Z at s_0 are the initial values y_0 and z_0 for y and z at t_0 and the initial value for α at s_0 is t_0 . □

In this approach, where the time t is transformed in the time s and the event time becomes known to be 0 a priori, the event location consists in integrating the DAE (17) in the unknown Y, Z and α over the interval $[s_0, 0]$: the event time is $\alpha(0)$ and the event point is $(Y(0), Z(0))$.

Observe that (17) is a DAE where α' , not α , is an unknown scalar algebraic variable. By introducing the fictitious differential equation $\alpha'(s) = \beta(s)$, α becomes a new differential variable and β is a new algebraic variable. The DAE (17) takes the form

$$\left\{ \begin{array}{l} Y'(s) = \beta(s) f(Y(s), Z(s)), \quad s \in [s_0, 0], \\ \alpha'(s) = \beta(s), \quad s \in [s_0, 0], \\ g(Y(s), Z(s)) = 0, \quad s \in [s_0, 0], \\ h(Y(s), Z(s)) = \kappa(s), \quad s \in [s_0, 0], \\ (Y(s_0), Z(s_0), \alpha(s_0)) = (y_0, z_0, t_0). \end{array} \right. \tag{18}$$

Observe that the algebraic scalar variable β appears in the differential equations, but not in the algebraic equations.

4.2 Index and Form of the s-time DAE

A general DAE is said of *index* m , m positive integer, if m differentiations of the algebraic equations provide explicit differential equations for the algebraic variables by reducing the DAE to an ODE.

Thus, by following this definition, if the original t -time DAE (1) has index 1, then the s -time DAE (18) is of index 2, and if the t -time DAE has index higher than 1, then the s -time DAE (18) has the same index as the t -time DAE. In fact, by two differentiations of the algebraic equation

$$h(Y(s), Z(s)) = \kappa(s)$$

we can obtain an explicit differential equation for β .

Suppose that the t -time DAE (1) has index 1. Since the algebraic equation $g(y, z) = 0$ can be written as $z = G(y)$ for some function G , the s -time DAE (18) reads

$$\begin{cases} Y'(s) = \beta(s) f(Y(s), G(Y(s))), & s \in [s_0, 0], \\ \alpha'(s) = \beta(s), & s \in [s_0, 0], \\ h(Y(s), G(Y(s))) = \kappa(s), & s \in [s_0, 0], \\ (Y(s_0), \alpha(s_0)) = (y_0, t_0). \end{cases} \tag{19}$$

The DAE (19) is an *Hessenberg index-2* DAE, i.e. a DAE where the algebraic variables appear only in the differential equations.

If the t -time DAE (1) is of index higher than 1, then the s -time DAE (18) inherits this higher index and it has in addition the scalar differential variable α and the scalar algebraic variable β , which appears only in the differential equations. In particular, if the t -time DAE (1) is an *Hessenberg index-2* DAE

$$\begin{cases} y'(t) = f(y(t), z(t)), & t \geq t_0, \\ g(y(t)) = 0, & t \geq t_0, \\ y(t_0) = y_0, \end{cases} \tag{20}$$

then the s -time DAE (18) becomes

$$\begin{cases} Y'(s) = \beta(s) f(Y(s), Z(s)), & s \in [s_0, 0], \\ \alpha'(s) = \beta(s), & s \in [s_0, 0], \\ g(Y(s)) = 0, & s \in [s_0, 0], \\ h(Y(s), Z(s)) = \kappa(s), & s \in [s_0, 0], \\ (Y(s_0), \alpha(s_0)) = (y_0, t_0). \end{cases} \tag{21}$$

The s -time DAE (21) has index 2, but in general it is not *Hessenberg index-2*, since the algebraic variable Z appears in the second algebraic equation. Of course, in the particular case $h(y, z) = h(y)$, the s -time DAE (21) is *Hessenberg index-2*.

In the next two subsections, we see two alternative approaches for the solution of (17), different from viewing it as a DAE of index (at least) 2. One is to reduce it to an ODE by a unique differentiation (not two differentiations) of the algebraic equation. The other is to reduce it to a DAE with the sole algebraic equation of the original DAE.

4.3 Reduction to an ODE

The DAE (17) can be reduced to an ODE by differentiating the algebraic equations.

Theorem 2 *The functions Y and Z and the time transformation α satisfy the ODE*

$$\begin{cases} Y'(s) = \frac{\kappa'(s)}{B(Y(s), Z(s))f(Y(s), Z(s))} f(Y(s), Z(s)), & s \in [s_0, 0], \\ Z'(s) = \frac{\kappa'(s)}{B(Y(s), Z(s))f(Y(s), Z(s))} A(Y(s), Z(s)) f(Y(s), Z(s)), & s \in [s_0, 0], \\ \alpha'(s) = \frac{\kappa'(s)}{B(Y(s), Z(s))f(Y(s), Z(s))}, & s \in [s_0, 0], \\ (Y(s_0), Z(s_0), \alpha(s_0)) = (y_0, z_0, t_0), \end{cases} \tag{22}$$

where

$$A(Y(s), Z(s)) = -g_z(Y(s), Z(s))^{-1} g_y(Y(s), Z(s)) \in \mathbb{R}^{d_2 \times d_1}$$

and

$$B(Y(s), Z(s)) = h_y(Y(s), Z(s)) + h_z(Y(s), Z(s)) A(Y(s), Z(s)) \in \mathbb{R}^{1 \times d_1},$$

with g_z, g_y, h_y, h_z derivatives (jacobian matrices) of the functions g and h with respect to the variables y and z .

Proof For $s \in [s_0, 0]$, we have

$$\frac{d}{ds} g(Y(s), Z(s)) = g_y(Y(s), Z(s))Y'(s) + g_z(Y(s), Z(s))Z'(s) = 0 \tag{23}$$

and

$$\frac{d}{ds} h(Y(s), Z(s)) = h_y(Y(s), Z(s))Y'(s) + h_z(Y(s), Z(s))Z'(s) = \kappa'(s). \tag{24}$$

Hence, we obtain

$$Z'(s) = A(Y(s), Z(s))Y'(s)$$

by (23) and

$$B(Y(s), Z(s))Y'(s) = B(Y(s), Z(s))\alpha'(s) f(Y(s), Z(s)) = \kappa'(s)$$

and then

$$\alpha'(s) = \frac{\kappa'(s)}{B(Y(s), Z(s)) f(Y(s), Z(s))}$$

by (24). Now, (22) easily follows. □

Although the ODE (22) is equivalent to the DAE (17), it has a much more complicated form than (17) and it involves g_y , the inverse of g_z, h_y and h_z .

4.4 Reduction to a DAE

In the particular case where $h(y, z) = h(y)$, the DAE (17) is equivalent to a DAE with the same algebraic equation of the original DAE (1). This is obtained by differentiating the scalar algebraic equation involving h .

Theorem 3 Suppose $h(y, z) = h(y)$. The functions Y and Z satisfy the DAE

$$\begin{cases} Y'(s) = \frac{\kappa'(s)}{h_y(Y(s))f(Y(s), Z(s))} f(Y(s), Z(s)), & s \in [s_0, 0], \\ g(Y(s), Z(s)) = 0, & s \in [s_0, 0], \\ (Y(s_0), Z(s_0)) = (y_0, z_0), \end{cases} \tag{25}$$

and the time-transformation α satisfy

$$\begin{cases} \alpha'(s) = \frac{\kappa'(s)}{h_y(Y(s))f(Y(s), Z(s))}, & s \in [s_0, 0], \\ \alpha(s_0) = t_0. \end{cases} \tag{26}$$

Proof For $s \in [s_0, 0]$, we have,

$$\kappa'(s) = h_y(Y(s)) Y'(s) = h_y(Y(s)) \alpha'(s) f(Y(s), Z(s))$$

and then

$$\alpha'(s) = \frac{\kappa'(s)}{h_y(Y(s)) f(Y(s), Z(s))}$$

in the differential equation of the DAE (17). □

Observe that (25) is a DAE in the sole variables Y and Z and it has the same algebraic equation of the DAE (1). The time transformation α satisfies the pure quadrature problem (26), which can be solved once Y and Z are determined. This reformulation of (17) as a DAE similar to the original DAE (1) only involves the derivative h_y .

We have seen that, when the t -time DAE has index 1, the s -time DAE (18) has index 2. On the other hand, the index 1 of the t -time DAE turns out to be preserved in the DAE (25).

Remark 4 When the DAE (1) is an ODE, i.e. $f(y, z) = f(y)$ and the algebraic equation is not present, the equation (17) remains a DAE and the equation (18) remains an Hessenberg index-2 DAE. Thus, we pay for the simplification of the problem, achieved through the time reparametrization, with the change in the nature of the equation, even in the simplest case of an ODE. Observe that in [17], the DAE (17) corresponding to an ODE (1) is reduced to (25), which is an ODE.

5 Numerical Solution in the s -time

In this section we present numerical methods for the s -time DAE (18). When we consider such methods as methods for the numerical event location in the original t -time DAE (1), they turn out to be methods satisfying all three properties (a), (b) and (c) in the introduction. It is important to remark that none of the standard methods presented in Section 2 can do this.

We consider the situation where the s -time DAE (18) is Hessenberg index-2. This happens when the original t -time DAE has index 1 or has the Hessenberg index-2 form (20) and $h(y, z) = h(y)$.

The s -time Hessenberg index-2 DAE reads

$$\begin{cases} Y'(s) = \beta(s)f(Y(s), Z(s)), & s \in [s_0, 0], \\ \alpha'(s) = \beta(s), & s \in [s_0, 0], \\ g(Y(s), Z(s)) = 0, & s \in [s_0, 0], \\ h(Y(s), Z(s)) = \kappa(s), & s \in [s_0, 0], \\ (Y(s_0), \alpha(s_0), Z(s_0)) = (y_0, t_0, Z_0) \end{cases} \tag{27}$$

where

- the first algebraic equation reads $Z(s) = G(Y(s))$ for a t -time DAE of index 1;
- the algebraic equations read $g(Y(s)) = 0$ and $h(Y(s)) = \kappa(s)$ for a t -time Hessenberg index-2 DAE.

We integrate the Hessenberg index-2 DAE (27), over a mesh

$$s_0 < s_1 < \dots < s_N = 0 \tag{28}$$

of stepsizes

$$\tau_{n+1} = s_{n+1} - s_n, \quad n = 0, 1, \dots, N - 1,$$

by RK methods (A, b, c) as illustrated in Chapter VII.4 of [12]. For a general Hessenberg index-2 DAE

$$\begin{cases} U'(s) = \mathcal{F}(U(s), V(s)), & s \in [s_0, 0], \\ \mathcal{G}(s, U(s)) = 0, & s \in [s_0, 0], \\ U(s_0) = U_0, \end{cases}$$

where U is the vector of the differentiable variables and V is the vector of the algebraic variables, the RK numerical integration over the mesh (28) takes the form

$$\begin{aligned} U_{n+1} &= U_n + \tau_{n+1} \sum_{i=1}^v b_i \mathcal{F}(U_{ni}, V_{ni}) \\ V_{n+1} &= V_n + \tau_{n+1} \sum_{i=1}^v b_i l_{ni}, \end{aligned} \tag{29}$$

where the stage values $(U_{ni}, V_{ni}), i = 1, \dots, v$, are determined by the non-linear equations

$$\begin{aligned} U_{ni} &= U_n + \tau_{n+1} \sum_{j=1}^v a_{ij} \mathcal{F}(U_{nj}, V_{nj}), \quad i = 1, \dots, v, \\ \mathcal{G}(s_{ni}, U_{ni}) &= 0, \quad j = 1, \dots, v, \end{aligned} \tag{30}$$

with $s_{ni} = s_n + c_i \tau_{n+1}$, and the derivatives $l_{ni}, i = 1, \dots, v$, are determined by the linear equations

$$V_{ni} = V_n + \tau_{n+1} \sum_{j=1}^v a_{ij} l_{nj}, \quad i = 1, \dots, v, \tag{31}$$

once the system (30) is solved,

About the convergence of the scheme (29), (30), (31), one can refer to the results in Chapter VII.4 in [12], in particular Table 4.1.

When the original t -time DAE has index 1, the RK numerical integration of the s -time DAE is

$$\begin{aligned}
 Y_{n+1} &= Y_n + \tau_{n+1} \sum_{i=1}^{\nu} b_i \beta_{ni} f(Y_{ni}, Z_{ni}) \\
 \alpha_{n+1} &= \alpha_n + \tau_{n+1} \sum_{i=1}^{\nu} b_i \beta_{ni} \\
 g(Y_{n+1}, Z_{n+1}) &= 0,
 \end{aligned} \tag{32}$$

where the stage values $(Y_{ni}, Z_{ni}, \beta_{ni}), i = 1, \dots, \nu$, are obtained by solving the non-linear system

$$\begin{aligned}
 Y_{ni} &= Y_n + \tau_{n+1} \sum_{j=1}^{\nu} a_{ij} \beta_{nj} f(Y_{nj}, Z_{nj}), \quad i = 1, \dots, \nu, \\
 g(Y_{ni}, Z_{ni}) &= 0, \quad i = 1, \dots, \nu, \\
 h(Y_{ni}, Z_{ni}) &= \kappa(s_n + c_i \tau_{n+1}), \quad i = 1, \dots, \nu,
 \end{aligned} \tag{33}$$

of dimension $\nu(d_1 + d_2 + 1)$.

When the original t -time DAE is Hessenberg index-2 and $h(y, z) = h(y)$, the RK numerical integration of the s -time DAE is

$$\begin{aligned}
 Y_{n+1} &= Y_n + \tau_{n+1} \sum_{i=1}^{\nu} b_i \beta_{ni} f(Y_{ni}, Z_{ni}) \\
 \alpha_{n+1} &= \alpha_n + \tau_{n+1} \sum_{i=1}^{\nu} b_i \beta_{ni} \\
 Z_{n+1} &= Z_n + \tau_{n+1} \sum_{i=1}^{\nu} b_i l_{ni},
 \end{aligned} \tag{34}$$

where, first, the stage values $(Y_{ni}, Z_{ni}, \beta_{ni}), i = 1, \dots, \nu$, are obtained by solving the non-linear system

$$\begin{aligned}
 Y_{ni} &= Y_n + \tau_{n+1} \sum_{j=1}^{\nu} a_{ij} \beta_{nj} f(Y_{nj}, Z_{nj}), \quad i = 1, \dots, \nu, \\
 g(Y_{ni}) &= 0, \quad i = 1, \dots, \nu, \\
 h(Y_{ni}) &= \kappa(s_n + c_i \tau_{n+1}), \quad i = 1, \dots, \nu,
 \end{aligned} \tag{35}$$

of dimension $\nu(d_1 + d_2 + 1)$ and, then, the derivatives $l_{ni}, i = 1, \dots, \nu$, are obtained by solving the linear system

$$Z_{ni} = Z_n + \tau_{n+1} \sum_{j=1}^{\nu} a_{ij} l_{nj}, \quad i = 1, \dots, \nu, \tag{36}$$

of dimension νd_2 .

The numerical integration of (27) provides the numerical event time α_N and the numerical event point (Y_N, Z_N) . Observe that there is no equation for β_{n+1} in (32) or (34), since we are not interested in the nodal values β_n but only in the stage values β_{ni} . In this manner, we can avoid to assign an initial value β_0 for the variable β .

If the s -time DAE is not Hessenberg index-2, then other RK schemes need to be used. In case of DAEs of index 3, see for example [14].

5.1 Diagonally Implicit Stiffly Accurate RK Methods

Suppose that a stiffly accurate RK method (recall (9)) is used for the integration (32), (33) or (34), (35), (36) of the s -time DAE (27). We have

$$(Y_{n+1}, Z_{n+1}) = (Y_{nv}, Z_{nv}), \quad n = 0, 1, \dots, N - 1,$$

and then the method consists only in the equations (33) or (35) and the equation for α_{n+1} .

The next theorem contains a fundamental result of our study

Theorem 5 *If the t -time DAE (1) is of index 1, or it is Hessenberg index-2 and $h(y, z) = h(y)$, and the numerical event location for such DAE is accomplished with an integration of the s -time DAE (27) by a stiffly accurate RK method, then the properties (a), (b) and (c) in the introduction are satisfied.*

Proof By construction (see (33) and (35)), we have a consistent numerical solution, i.e.

$$g(Y_{n+1}, Z_{n+1}) = g(Y_{nv}, Z_{nv}) = 0, \quad n = 0, 1, \dots, N - 1.$$

In particular, we have $g(Y_N, Z_N) = 0$ and so the numerical event point (Y_N, Z_N) is consistent (point b) in the introduction).

Moreover, we have

$$h(Y_{ni}, Z_{ni}) = \kappa(s_n + c_i \tau_{n+1}) \leq 0, \quad n = 0, 1, \dots, N - 1 \text{ and } i = 1, \dots, v,$$

i.e.

$$(Y_{ni}, Z_{ni}) \in S^- \cup \Sigma, \quad n = 0, 1, \dots, N - 1 \text{ and } i = 1, \dots, v. \tag{37}$$

Thus, the method is one-side (point c) in the introduction), since the values of f and g are required only at points (37).

Finally, we have

$$h(Y_{n+1}, Z_{n+1}) = \kappa(s_{n+1}), \quad n = 0, 1, \dots, N - 1.$$

In particular, we have $h(Y_N, Z_N) = 0$ and thus the numerical event point (Y_N, Z_N) lies on the discontinuity surface (point a) in the introduction). □

Observe that the one-side stiffly accurate RK method reaches the discontinuity surface, i.e. it reaches the final s -time $s = 0$, in an arbitrary number of steps fixed a priori, since there are no restrictions posed on the mesh (28).

5.1.1 Diagonally Implicit RK Methods

To avoid a large computational cost, we can use a *diagonally implicit stiffly accurate* RK method, instead of a fully implicit stiffly accurate RK method. Remind that a *diagonally implicit* RK method is a RK method such that

$$a_{ij} = 0, \quad i, j = 1, \dots, v \text{ with } i < j.$$

So, in (33) or (35), the stage values $(Y_{ni}, Z_{ni}, \beta_{ni}), i = 1, \dots, v$, are obtained by successively solving, for $i = 1, \dots, v$, the non-linear system

$$\begin{aligned}
 Y_{ni} &= Y_n + \tau_{n+1} \sum_{j=1}^{i-1} a_{ij} \beta_{nj} f(Y_{nj}, Z_{nj}) + \tau_{n+1} a_{ii} \beta_{ni} f(Y_{ni}, Z_{ni}) \\
 g(Y_{ni}, Z_{ni}) &= 0 \\
 h(Y_{ni}, Z_{ni}) &= \kappa(s_n + c_i \tau_{n+1}),
 \end{aligned}$$

or

$$\begin{aligned}
 Y_{ni} &= Y_n + \tau_{n+1} \sum_{j=1}^{i-1} a_{ij} \beta_{nj} f(Y_{nj}, Z_{nj}) + \tau_{n+1} a_{ii} \beta_{ni} f(Y_{ni}, Z_{ni}) \\
 g(Y_{ni}) &= 0 \\
 h(Y_{ni}) &= \kappa(s_n + c_i \tau_{n+1}),
 \end{aligned}$$

of dimension $d_1 + d_2 + 1$, where the stage value $(Y_{ni}, Z_{ni}, \beta_{ni})$ are the unknowns and the stage-values $(Y_{nj}, Z_{nj}, \beta_{nj}), j = 1, \dots, i - 1$, have been already computed.

The computational saving is given, at any step, by the solution of v non-linear systems of dimension $d_1 + d_2 + 1$, rather than a unique non-linear system of dimension $v(d_1 + d_2 + 1)$.

5.1.2 Some Methods

The simplest diagonally implicit stiffly accurate RK method is the implicit Euler method.

Two-stage diagonally implicit stiffly accurate RK methods of order two (for ODEs) are given by the tableau

$$\begin{array}{c|c}
 a & a \\
 1 & 1 - b \quad b \\
 \hline
 & 1 - b \quad b
 \end{array}$$

where

$$a = \frac{\frac{1}{2} - b}{1 - b}.$$

For $b = \frac{1}{2}$, we have the trapezoidal rule. For $b = 0$, we have the implicit midpoint rule.

Table 1 Hessenberg index-2 global orders of convergence for methods in Section 5.1.2

Method	Variables y	Variables z
Implicit Euler method (Radau IIA with $s = 1$)	One	One
Trapezoidal rule (Lobatto IIA with $s = 2$)	Two	Two
Implicit midpoint rule (Gauss with $s = 1$)	Two	One
Method (38) (SDIRK (IV.6.18))	Two	One

As another example of a diagonally implicit accurately stiff RK method, we give the following five-stage method of order four (for ODEs)

$$\begin{array}{c|cccc}
 \frac{1}{4} & \frac{1}{4} & & & \\
 \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & & \\
 \frac{11}{20} & \frac{17}{50} & -\frac{1}{25} & \frac{1}{4} & \\
 \frac{1}{2} & \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} \\
 1 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} \frac{1}{4} \\
 \hline
 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} \frac{1}{4}
 \end{array} \tag{38}$$

appearing in [12, IV.(6.16)]. Indeed, since all the diagonal coefficients a_{ii} are equal, this method is a singly diagonally implicit RK (SDIRK) method.

The global orders of convergence when the implicit Euler method, the trapezoidal rule, the implicit midpoint rule and the method (38) are applied to Hessenberg index-2 DAE are given in Table 1 derived from Table 4.1 in [12].

6 Numerical Tests

We consider three numerical examples for testing our approach to the numerical event location for DDAEs by the time reparametrization. Indeed, the first and third examples are event locations problems for DAEs, where one can imagine that the event equation could constitute a surface of discontinuity. Whereas, the second example comes from a model formulated as a DDAE.

6.1 The First Test

The first example is the three-dimensional DAE of index 1

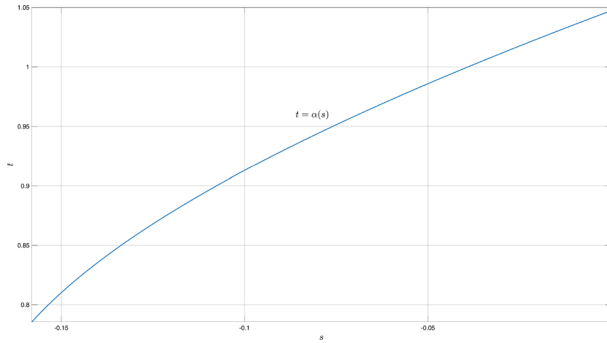


Fig. 1 Time transformation for the Hessenberg index-2 DAE relevant to the DAE (39)

$$\begin{cases} y_1'(t) = -2y_2(t), & t \geq \frac{\pi}{4}, \\ y_2'(t) = -z(t)^2 + y_1(t), & t \geq \frac{\pi}{4}, \\ y_1(t)^2 + y_2(t)^2 + z(t)^2 - 1 = 0, & t \geq \frac{\pi}{4}, \\ (y(\frac{\pi}{4}), z(\frac{\pi}{4})) = (\frac{1}{4}, \frac{1}{4}, \frac{\sqrt{2}}{2}) \end{cases} \tag{39}$$

whose solution is

$$(y(t), z(t)) = (\cos^2 t, \cos t \sin t, \sin t).$$

with the event equation

$$h(y, z) = -y_1(t) - y_2(t) - z(t) + v = 0,$$

where

$$v = \cos^2 \frac{\pi}{3} + \cos \frac{\pi}{3} \sin \frac{\pi}{3} + \sin \frac{\pi}{3}$$

In this situation, the event time is $t^* = \frac{\pi}{3}$ and the event point is

$$(y^*, z^*) = (\cos^2 \frac{\pi}{3}, \cos \frac{\pi}{3} \sin \frac{\pi}{3}, \sin \frac{\pi}{3}).$$

The numerical event location is accomplished by integrating the Hessenberg index-2 DAE (18), relevant to the DAE (39), with $\kappa(s) = s$. The implicit Euler method, the trapezoidal rule and the method (38), are used for the numerical integration. We consider uniform meshes (28) with

$$N_k = 2^k, \quad k = 0, 1, \dots, 10,$$

subintervals.

In Fig. 1, we see the time-transformation (computed with the method (38) over the mesh with $k = 10$).

In Fig. 2, we see, for the three methods, the logarithm in base 10 of the errors

$$errt_k = |\alpha_N - t^*|, \quad erry_k = \|Y_N - y^*\|_2 \text{ and } errz_k = |Z_N - z^*|, \quad k = 0, 1, \dots, 10,$$

in red dashed lines marked with \diamond . The black solid lines marked with \star show convergences with order one, two, three and four.

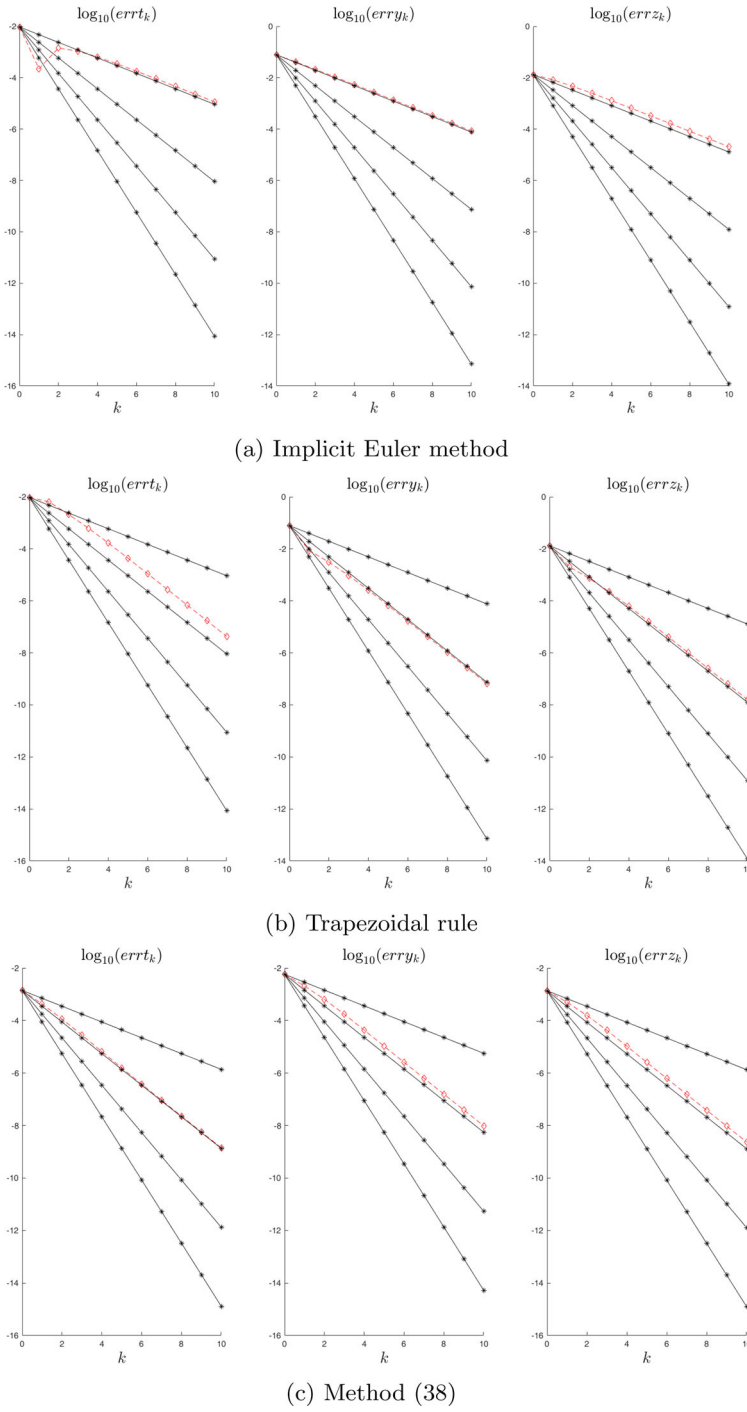


Fig. 2 Errors in the integration of the Hessenberg index-2 DAE relevant to the DAE (39)

The orders of convergence one, two and two in the differential variables (α included) for implicit Euler method, trapezoidal rule and method (38), respectively, given in Table 1 are here confirmed. The order of convergence of the algebraic variable Z is the order of the differential variables since $Z = G(Y)$ holds.

6.2 The Second Test

The second example is the DAE of index 1

$$\begin{cases} y_1'(t) = F_1 - z(t) - k_c \frac{y_1(t)y_2(t)}{V}, & t \geq 0 \\ y_2'(t) = F_2 - k_c \frac{y_1(t)y_2(t)}{V}, & t \geq 0, \\ y_3'(t) = k_c \frac{y_1(t)y_2(t)}{V}, & t \geq 0, \\ z(t) = k_g X (P(y(t)) - P_{\text{out}}), & t \geq 0, \\ (y_1(0), y_2(0), y_3(0)) = (0.72, 95, 0), \end{cases} \tag{40}$$

where

$$P(y(t)) = \frac{y_1(t)RT}{V - \frac{y_2(t)}{\rho_l} - \frac{y_3(t)}{\rho_a}},$$

describing the gas-phase in a model of soft-drink production (see [7]). The event equation is

$$h(y(t), z(t)) = \frac{y_2(t)}{\rho_l} + \frac{y_3(t)}{\rho_a} - V_d = 0,$$

and involves only $y(t)$. After the event, we have a transition to the liquid-phase described by another DAE. About constants and parameters in (40), the following values

$$\begin{aligned} F_1 &= 0.5, \quad F_2 = 7.5, \quad k_c = \frac{0.433}{4000}, \quad V = 10, \\ k_g &= 3, \quad X = 1, \quad P_{\text{out}} = 1 \\ R &= 0.0820574587, \quad T = 293, \quad \rho_a = 16, \quad \rho_l = 50, \\ V_d &= 2.25 \end{aligned}$$

in suitable units are used.

The numerical event location is accomplished by integrating the Hessenberg index-2 DAE (18), relevant to the DAE (40), with $\kappa(s) = s$. We use the three methods of the previous test over the same meshes.

In Fig. 3, we see the time-transformation.

In Fig. 4, we see the logarithm in base 10 of the errors as in the previous test. For the computation of the errors, the exact event time and event point are estimated by integrating with the method (38) over a uniform mesh with $N = 2^{13}$ subintervals, one thousand time more subintervals than the more refined mesh used: We obtain the values

$$\begin{aligned} t^* &= 2.333036718967131 \\ y^* &= (3.767995595486393 \cdot 10^{-1}, 1.124967285180228 \cdot 10^2, \end{aligned}$$

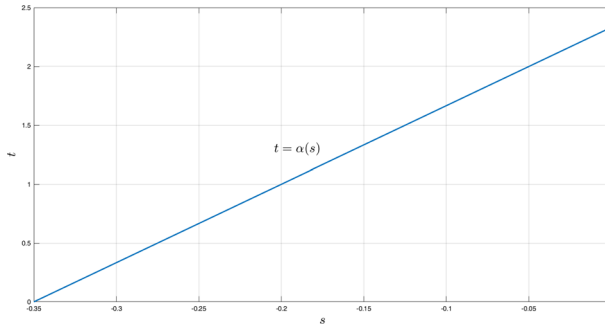


Fig. 3 Time transformation for the Hessenberg index-2 DAE relevant to the DAE (40)

$$z^* = 5.068373375540564 \cdot 10^{-1} \cdot 1.046874232710747 \cdot 10^{-3}$$

For the implicit Euler method, the trapezoidal rule and the method (38) the observed orders are one, two and four, respectively. The method (38) has, on this problem, the order four for ODEs, not the order two of Table 1 for Hessenberg index-2 DAEs. This could be due to the fact that, since h depends only on the differentiable variable and the t -time DAE is of index 1, the s -time DAE can be reduced to a DAE of index 1 as described in Sect. 4.4, where the convergence order is the convergence order for ODEs.

6.3 The Third Test

The third example is given by the second order Hessenberg index-2 DAE describing the motion of the simple pendulum in cartesian coordinates x and y (the x -axis is horizontal and the y -axis is vertical downward)

$$\begin{cases} x''(t) = -n(t)x(t), & t \geq 0, \\ y''(t) = -n(t)y(t) + g, & t \geq 0, \\ x(t)^2 + y(t)^2 = 1, & t \geq 0, \end{cases} \tag{41}$$

where the algebraic variable $n(t)$ appearing only in the differential equations is the tension of the pendulum rod. Here, we are assuming that the rod has unit length and the pendulum bob has unit mass.

We consider the initial condition

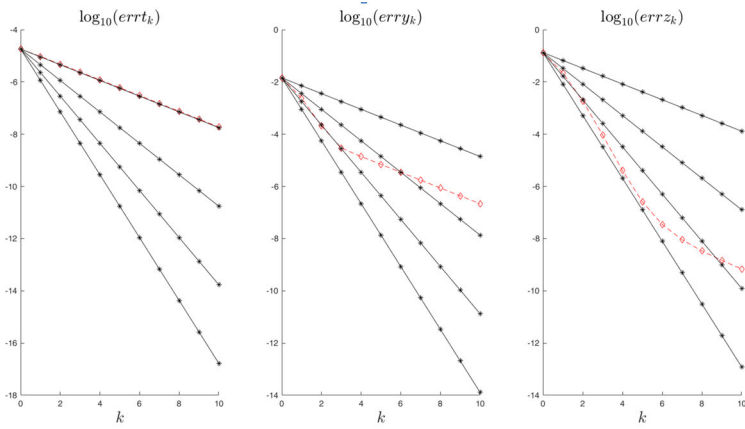
$$\begin{cases} (x(0), y(0)) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right) \\ (x'(0), y'(0)) = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right) \end{cases}$$

i.e. the pendulum starts with unit speed at 45° with respect to the vertical axis, and the event equation

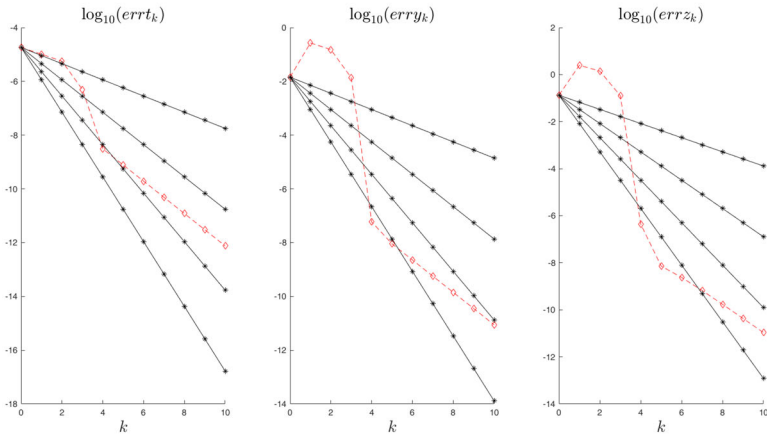
$$x(t) = 0,$$

i.e. the event is when the pendulum bob reaches the lowest point in the motion.

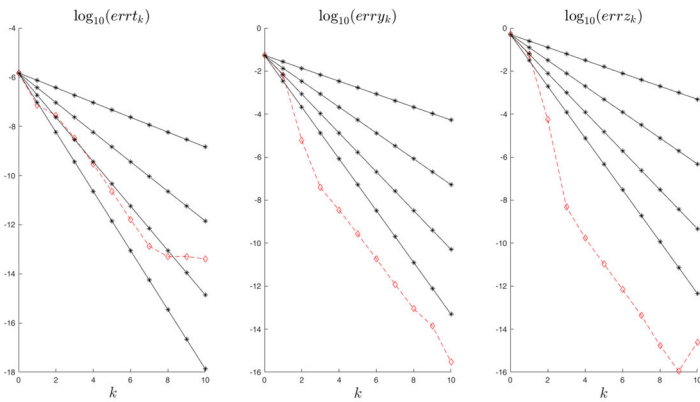
The DAE (41) has index 3 (by three differentiations of the algebraic equation we obtain a differential equation for n) and then the DAE (18) also has index 3. Thus, suitable numerical methods for the index 3 (see [14]) should be used for its integration. However, the DAE (41)



(a) Implicit Euler method



(b) Trapezoidal rule



(c) Method (38)

Fig. 4 Errors in the integration of the Hessenberg index-2 DAE relevant to the DAE (40)

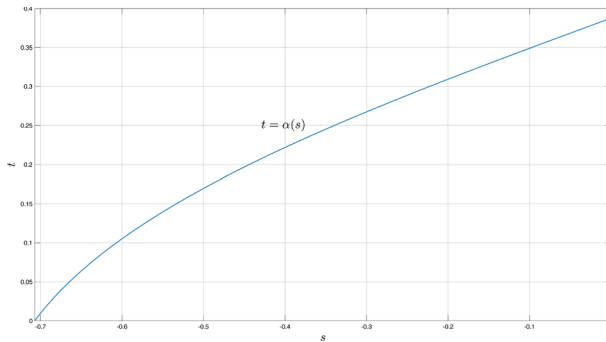


Fig. 5 Time transformation for the s -time Hessenberg index-2 DAE relevant to the t -time DAE (41)

can be reduced to an Hesseberg index-2 DAE by differentiating only one time the algebraic equation. We obtain

$$\begin{cases} x''(t) = -n(t)x(t), & t \geq 0, \\ y''(t) = -n(t)y(t) + g, & t \geq 0, \\ x(t)x'(t) + y(t)y'(t) = 0, & t \geq 0, \end{cases} \tag{42}$$

The numerical event location is accomplished by integrating the DAE (18), relevant to the first order version of the second order DAE (42), with $\kappa(s) = s$. Unlike the previous tests, where the original t -time DAE was of index 1, here the t -time DAE is Hessenberg index-2. The s -time DAE is Hessenberg index-2, since $h(y, z) = h(y)$.

For the numerical integration of the s -time DAE, we use the same methods with the same meshes of the two previous tests.

In Fig. 5, we see the time-transformation.

In Fig. 6, we see the logarithm in base 10 of the errors of t^* , $x'(t^*)$ and $n(t^*)$. The exact values of t^* , $x'(t^*)$ and $n(t^*)$ can be easily determined by well-known physics: we have

$$\begin{aligned} t^* &= 3.875000113579756 \cdot 10^{-1} \\ x'(t^*) &= -2.597415052147026 \\ n(t^*) &= 1.655656495311994 \cdot 10^1. \end{aligned}$$

The implicit Euler method appears to converge with order one as predicted in Table 1. The trapezoidal rule appears to converge with order two for the differential variables as predicted in Table 1, but it does not converge for the algebraic variable although the order is two for such variables in Table 1. The method (38) appears to converge with order two, also for the algebraic variable although the order is one for such variables in Table 1.

Just for a comparison, by replacing the trapezoidal rule with the implicit midpoint rule, we obtain the errors in Fig. 7. The orders agree with Table 1, although the order for $x'(t^*)$ is four rather than two.

7 Conclusion

In this paper, we have presented a new approach to the event location for DDAEs based on a time reparametrization reducing the event location to the integration of an Hessenberg index-2 DAE up to a known final time. By integrating this DAE by a diagonally implicit stiffly accurate RK method, we obtain a not-so-computationally-expensive method for the

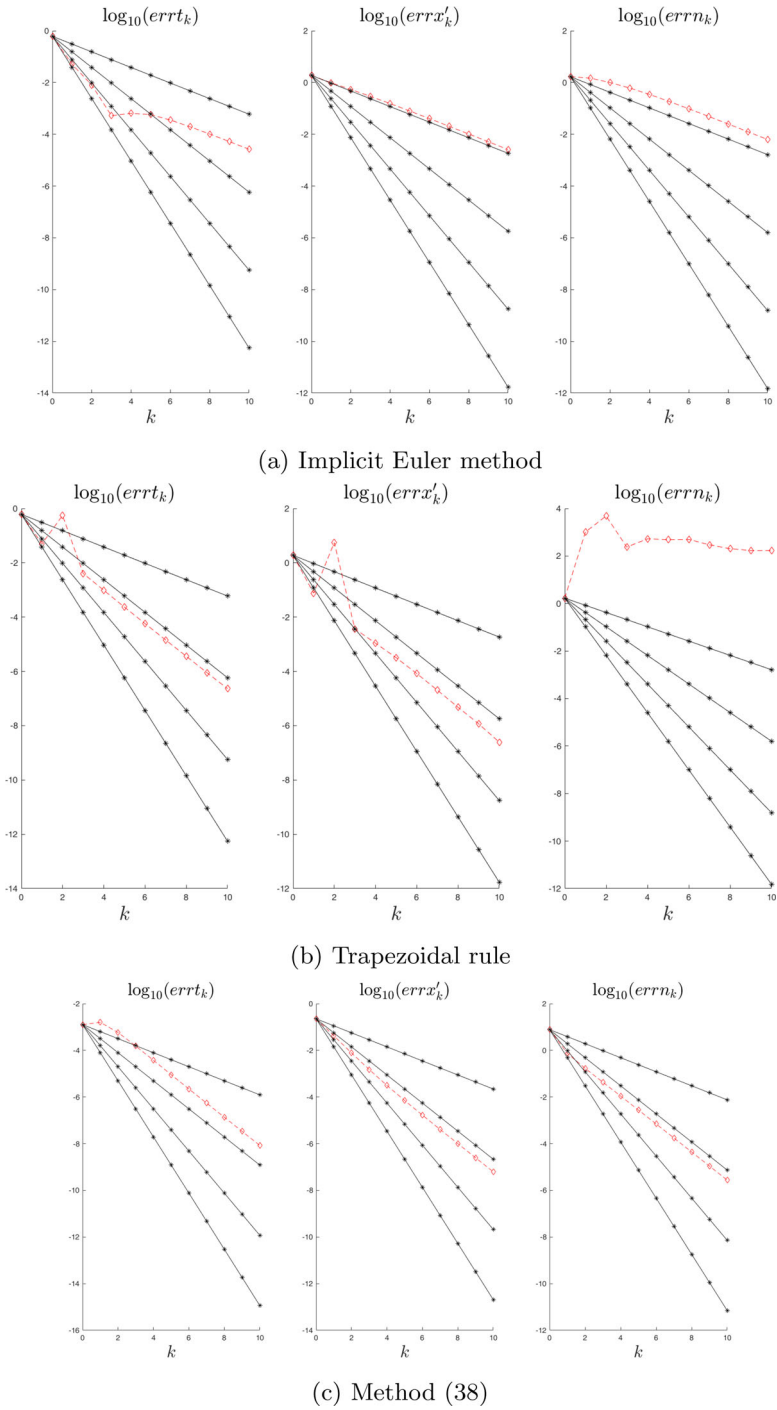


Fig. 6 Errors in the integration of the Hessenberg index-2 DAE relevant to the DAE (42)

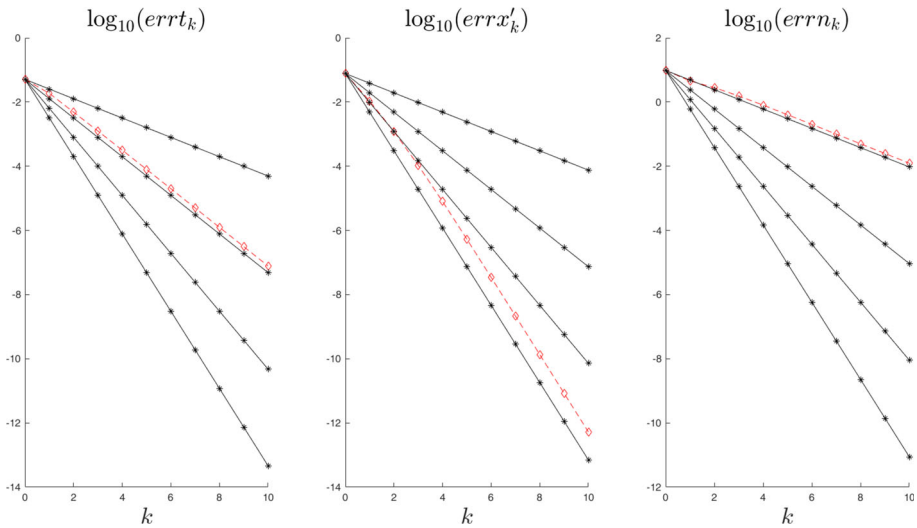


Fig. 7 Errors in the integration of the Hessenberg index-2 DAE relevant to the DAE (42) by the implicit midpoint rule

numerical event location satisfying the properties (a), (b) and (c) in the introduction. None of the standard methods for the numerical event location of DDAEs satisfies all three of these properties. The approach can be used also for DAEs of index higher than 1.

Acknowledgements The authors acknowledge that this research was supported by funds from the Italian MUR (Ministero dell'Università e della Ricerca) within the PRIN 2017 Project Discontinuous dynamical systems: theory, numerics and applications; and by the INdAM Research group GNCS (Gruppo Nazionale di Calcolo Scientifico).

Funding Open access funding provided by Università degli Studi di Trieste within the CRUI-CARE Agreement. The authors have no relevant financial or non-financial interests to disclose.

Data availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Agrawal, J., Moudgalya, K.M., Pani, A.K.: Sliding motion of discontinuous dynamical systems described by semi-implicit index one differential algebraic equations. *Chem. Eng. Sci.* **61**, 4722–4731 (2006)
2. Amodio, P., Brugnano, L., Iavernaro, F.: Arbitrary high-order methods for one-sided direct event location in discontinuous differential problems with nonlinear event function. *Appl. Numer. Math.* **179**, 39–49 (2022)
3. Biak, M., Hanus, T., Janovska, D.: Some applications of Filippov's dynamical systems. *J. Comput. Appl. Math.* **254**, 132–143 (2013)
4. Brunner, H., Maset, S.: Time transformation for delay differential equations. *Discr. Contin. Dyn. Syst.* **25**, 751–775 (2009)
5. Brunner, H., Maset, S.: Time transformation for state-dependent delay differential equations. *Commun. Pure Appl. Anal.* **9**, 23–45 (2010)
6. Berardi, M., Lopez, L.: On the continuous extension of Adams-Bashforth methods and the event location in discontinuous ODEs. *Appl. Math. Lett.* **25**, 995–999 (2015)
7. Dieci, L., Elia, C., Lopez, L.: On Filippov solutions of discontinuous DAEs of index 1. *Commun. Nonlinear Sci. Numer. Simulat.* **95**, 1105656 (2021)
8. Dieci, L., Lopez, L.: Numerical solution of discontinuous differential systems: approaching the discontinuity surface from one side. *Appl. Numer. Math.* **67**, 98–110 (2013)
9. Dieci, L., Lopez, L.: One-sided direct event location techniques in the numerical solution of discontinuous differential systems. *BIT Numer. Math.* **55**, 987–1003 (2015)
10. Galan, D.S., Barton, P.I.: Dynamic optimization of hybrid systems. *Comput. Chem. Eng.* **22 Suppl.**, S183–S190 (1998)
11. Hairer, E., Lubich, C., Roche, M.: The numerical solution of differential-algebraic systems by Runge-Kutta methods. Springer (1989)
12. Hairer, E., Wanner, G.: Solving ordinary differential equations II. Stiff Differential-Algebraic problems. Springer, Berlin (1996)
13. Henon, M.: On the numerical computation of Poincaré maps. *Physica* **5D**, 412–414 (1982)
14. Jay, L.: Convergence of Runge-Kutta methods for differential-algebraic systems of index 3. *Appl. Numer. Math.* **17**, 97–118 (1995)
15. Kunkel, P., Mehrmann, V.: Numerical solution of hybrid systems of differential-algebraic equations. *Comput. Methods Appl. Mech. Eng.* **197**, 693–705 (2008)
16. Kunkel, P., Mehrmann, V.: Regular solutions of DAE hybrid systems and regularization techniques. *BIT Numer. Math.* **58**, 1049–1077 (2018)
17. Lopez, L., Maset, S.: Time-transformations for the event location in discontinuous ODEs. *Math. Comput.* **87**, 2321–2341 (2018)
18. Lopez, L., Maset, S.: Numerical event location techniques in discontinuous differential algebraic equations. *Appl. Numer. Math.* **178**, 98–122 (2022)
19. Mao, G., Petzold, L.R.: Efficient integration over discontinuities for differential-algebraic systems. *Comput. Math. Appl.* **43**, 65–79 (2002)
20. Majer, C., Marquardt, W., Gilles, E.D.: Reinitialization of DEAs after discontinuities. *Comput. Chem. Eng.* **6**, 8507–8512 (1995). (Suppl.,)
21. Najafi, M., Nikoukhah, R.: Modeling and simulation of differential equations in Scicos. Modelica, The Modelica Association: 177–185, (2006)
22. Park, T., Barton, P.I.: State event location in differential-algebraic models. *ACM Trans. Model. Comput. Simul.* **6**, 137–165 (1996)
23. Stechliniski, P., Patrascu, M., Barton, P.I.: Nonsmooth differential-algebraic equations in chemical engineering. *Comput. Chem. Eng.* **114**, 52–68 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.