BIT manuscript No.
(will be inserted by the editor)

One-Sided Event Location Techniques in the Numerical Solution of Discontinuous Differential Systems

Luca Dieci · **Luciano Lopez**

Received: date / Accepted: date

Abstract In this short paper, event location techniques for a differential system the solution of which is directed towards a surface Σ defined as the 0-set of a smooth function h: $\Sigma = \{x \in \mathbb{R}^n \ : h(x) = 0 \ \}$ are considered. It is assumed that the exact solution trajectory hits Σ non-tangentially, and numerical techniques guaranteeing that the trajectory approaches Σ from one side only (i.e., does not cross it) are studied. Methods based on Runge Kutta schemes which arrive to Σ in a finite number of steps are proposed. The main motivation of this paper comes from integration of discontinuous differential systems of Filippov type, where location of events is of paramount importance.

Keywords Event surface · time reparametrization · Runge Kutta methods · monotone integration.

Mathematics Subject Classification (2000) 65L05 · 34A36

1 Introduction

2 Introduction

Let a given surface Σ be defined as

$$\Sigma = \{x \in \mathbb{R}^n |\ h(x) = 0\}\,,\ h : \mathbb{R}^n \to \mathbb{R}\ , \tag{2.1}$$

Work performed while the second author was in visit to the School of Mathematics at Georgia Tech, whose support is gratefully acknowledged. This work has been supported in part by the GNCS of INDAM .

Luca Dieci
School of Mathematics, Georgia Institute of Technology,Atlanta, GA 30332-0160, USA
E-mail: dieci@math.gatech.edu

Luciano Lopez
Dipartimento di Matematica, Universitá degli Studi di Bari “Aldo Moro”, Via E. Orabona 4, I-70125, Bari, Italy
E-mail: luciano.lopez@uniba.it

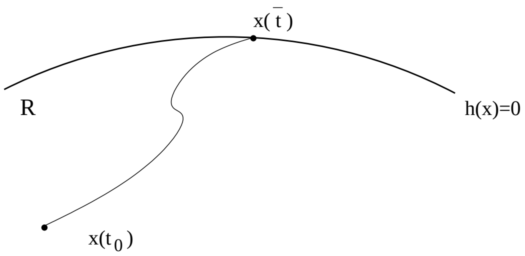

Fig. 2.1 Trajectory and event surface.

where $h \in C^k$, $k \geq 2$, and $\nabla h(x) \neq 0$ for all $x \in \Sigma$. At least locally, Σ separates region of phase space, and we let $R = \{x \in \mathbb{R}^n |\ h(x) < 0\ \}$. Such surface Σ is also called an *event set* and points $x \in \Sigma$ are also called *event points*.

Consider a differential system of the form

$$\begin{cases} x' = f(x)\ , \text{when } x \in R, \\ x(0) = x_0 \in R\ , \end{cases} \tag{2.2}$$

and write $x(t, x_0)$ for its solution. Also, define $H > 0$ by the relation $h(x_0) = -H$. Hereafter, the function f is assumed to be sufficiently smooth, so that the results later on (for example, Proposition 4.1 or Section 5) are meaningful.

In applications, quite often the function h is a (hyper-)plane:

$$h(x)\ :=\ d^T x + e\ ,\ d \in \mathbb{R}^n,\ e \in \mathbb{R}. \tag{2.3}$$

This situation arises especially in control engineering, where it is crucially important to make sure that a numerically computed trajectory lands exactly on Σ, to avoid undesired numerical chattering (e.g., see [2], [4], [11]). At the same time, there are applications where h is a more complicated nonlinear function, see the work on terminal sliding mode control theory (e.g., [19]) and in robotics (e.g., [10]).

Now, consider a solution trajectory of (2.2). The interesting case is when this trajectory is directed towards Σ (Σ is *attractive*) and reaches it in a finite time $\bar{t}$ (which of course depends on x_0), arriving on it non-tangentially; in other words, $x(\bar{t}, x_0) \in \Sigma$ and $(\nabla h)^T f \neq 0$ at $x(\bar{t}, x_0)$. What happens after we reach an event point is not our present concern (e.g., see [12], [1], [7], [8] and references there).

First of all, for points in R, we characterize attractivity of Σ as follows. *There exists a positive constant δ such that, for all $x \in R$ and sufficiently close to Σ, we have*

$$h_x^T(x) f(x) \geq \delta > 0\ . \tag{2.4}$$

Remark 2.1 Since along a solution trajectory we have

$$\frac{d}{dt} h(x) = h_x^T x' = h_x^T f\ ,$$

then (2.4) implies that the function h monotonically increases along a solution trajectory in R (and close to Σ), until eventually the trajectory hits Σ non-tangentially at the event point.

Although in most applications the function f is well defined in an open neighborhood of Σ, there are situations where one cannot extend f smoothly past the surface Σ; e.g., see [6], [10], [15], [17]. In this work, we are interested in numerical procedures in which the discontinuity surface is approached from one side and the numerical trajectory reaches Σ in a **finite** number of steps. We will call these procedures *exact event location methods*[1].

A plan of this paper is as follows. In Section 2 we review standard event location techniques, and briefly discuss a new one we considered. In Section 3 we give our main result, and propose a systematic way to reparametrize time in a manner which is conducive to exact event location methods for the case of event surfaces which are planar (see (2.3)) or quadratic. In Section 4, we adapt the results of Section 3, along the lines of [9], to deal with the case of general event surfaces. Finally, in Section 5 we report on numerical results for several test problems.

3 Review of event location techniques

Consider a grid: $0 = t_0 < t_1 < \ldots$, with $t_{k+1} = t_k + \tau_k$, for $k = 0, 1, \ldots$. Let x_j be the approximation to $x(t)$ at t_j, $j = 0, 1, \ldots, k$, obtained by a one-step or multistep method; we can assume that all these values of x_j, $j = 0, 1 \ldots, k$, are in R.

The straightforward idea of standard event location techniques is the following. When the numerical solution x_{k+1}, obtained by using the time step τ_k, lands on the other side of Σ, an event point is typically located by looking for a root of the scalar function $h(x_{k+1}(\tau)) = 0$.

Different methods come about from how τ is found. For example, if a continuous extension of the same order of the underlying scheme, or a polynomial interpolant, is available (as it is for some explicit Runge Kutta (RK) methods, or for multistep Adams-Bashforth methods, see [5], [3], [18]), then it is natural to use it in order to solve $h(x_{k+1}(\tau)) = 0$. Alternatively, one may make τ part of the unknowns to solve for and embed the constraint $h(x_{k+1}(\tau)) = 0$ in the construction of the step; for example, this is done for implicit RK methods in [13]. However, all of these methods most likely require evaluation of f at points outside of $R \cup \Sigma$, which is not desirable for us; moreover, there may also be multiple roots for the resulting nonlinear scalar function. The class of sub-diagonal explicit RK methods that we studied in [9] avoids these problems, but requires monitoring the position (with respect to Σ) of all the internal stages and this can be cumbersome.

Within the class of one-step schemes, an alternative idea to the methods mentioned above would be to use RK schemes with step-size dependent tableaux, and adjust the coefficients so to enforce the desired monotone behavior in the numerical trajectory. We exemplify this below, see Example 3.1, in the case of the event surface being a plane. The idea is to build a RK-like method for which $h(x_k)$ increases by a fixed positive value η at each step k. This way, we would reach the event surface in a finite number of steps.

[1] The word exact refers to locating an event point exactly, of course in general this will be a numerical approximation to the exact $x(\bar{t}, x_0)$

Example 3.1 (Explicit RK scheme with variable tableau) Restrict to the case of $h(x)$ as in (2.3), $h(x) = d^T x + e$, and consider the class of explicit second order Runge Kutta schemes. These are defined by the tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ c & c & 0 \\ \hline & 1-b & b \end{array} \quad , \tag{3.1}$$

subject to

$$cb = \frac{1}{2} \, . \tag{3.2}$$

We impose the monotonicity conditions

$$\begin{cases} h(x_k^2) = h(x_k) + \eta_2 & (a) \\ h(x_{k+1}) = h(x_k) + \eta & (b) \end{cases} \tag{3.3}$$

where $\eta_2 = \delta\eta$, and $\delta > 0$ (there is some freedom in specifying δ, see below). Now we can solve the nonlinear system (3.2)-(3.3). In fact, from (3.3.a)

$$c\tau = \frac{\eta_2}{d^T f(x_k)} \, ,$$

and so (using (3.2))

$$x_{k+1} = x_k + \tau[(1-b) f(x_k) + b f(x_k^2)] = x_k + \frac{2\eta_2 b}{d^T f(x_k)}[(1-b) f(x_k) + b f(x_k^2)] \, .$$

So, from (3.3.b), b is a root of the quadratic equation

$$[\frac{\beta}{\alpha} - 1]b^2 + b - \frac{1}{2}\frac{\eta}{\eta_2} \; = 0 \, , \tag{3.4}$$

where $\alpha = d^T f(x_k)$, and $\beta = d^T f(x_k^2)$. If $\frac{\beta}{\alpha} > 1$, there are two real roots of different sign and we only take the positive root, while if $\frac{\beta}{\alpha} < 1$, both roots are positive and one may take either one. [In the case $\alpha = \beta$ and $\eta_2 = \eta$, we get the familiar scheme with $b = \frac{1}{2}$ and $c = 1$, hence we would be inclined to take the root of (3.4) closer to $1/2$].

Similar considerations may be applied to explicit RK methods of higher order. However, this way of proceeding is not appealing as a general purpose methodology, and downright impractical if h is not of the form (2.3). □

4 Time reparametrization and RK schemes

Here we propose our new technique, whose simple idea is to exploit the monotonicity of h along solution trajectories. In fact, given (2.4), the function

$$s \; = \; h(x(t)) \tag{4.1}$$

is monotòne and can be used instead of t. We propose to do just this. From the chain rule,

$$dx/dt = f(x) \to (dx/ds)(ds/dt) = f(x) \to (dx/ds)(h_x^T(x)f(x)) = f(x)\ ,$$

and thus we obtain the system

$$\begin{cases} \frac{dx}{ds} \ = \ g(x)\ , \quad -H \le s \le 0\ , \\ x(-H) \ = \ x_0\ , \end{cases} \tag{4.2}$$

where

$$g(x) = \frac{f(x)}{h_x^T(x)f(x)}\ , \quad x \in R\ . \tag{4.3}$$

Observe that (2.4) becomes simply

$$h_x^T(x)g(x) = 1\ , \quad x \in R\ . \tag{4.4}$$

Of course, we have just reparametrized time, so that the two trajectories, $x(s,x_0)$ of (4.2) for $s \in [-H,0]$ and $x(t,x_0)$ of (2.2) for $t \in [0,\bar{t}]$, represent the same curve in state space. At first glance, little has changed, but in fact there are interesting computational advantages when working with the formulation (4.2) instead of (2.2), as we show below.

Consider the general p-stage RK scheme defined by the tableau

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}$$

where $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{p\times p}$, $\mathbf{c} = (c_i) \in \mathbb{R}^p$, $\mathbf{b} = (b_i) \in \mathbb{R}^p$. We will always assume the familiar conditions

$$\sum_{i=1}^{p} b_i = 1\ ,\ \ \sum_{j=1}^{p} a_{ij} = c_i\ ,\ \ i = 1,\ldots,p\ ,$$

and are further (exclusively) interested in the case of

$$0 \le c_i \le 1\ ,\ \ i = 1,\ldots,p\ .$$

Starting from x_0, one step of a RK method with time step $\sigma > 0$ applied to (4.2) (we reserve τ for the stepsize of a discretization of (2.2)), reads

$$\begin{aligned} x_1 \ &= \ x_0 + \sigma \textstyle\sum_{i=1}^{p} b_i g(x_0^i)\ , \\ x_0^i \ &= \ x_0 + \sigma \textstyle\sum_{j=1}^{p} a_{ij} g(x_0^j)\ , \quad i = 1,\ldots,p\ . \end{aligned} \tag{4.5}$$

The following result holds for any RK scheme, explicit or implicit, and suggests that when Σ is a plane there are advantages to using the formulation (4.2). Which RK formula to use will of course depend on whether (4.2) is stiff or not.

Theorem 4.1 *Let the event surface Σ be defined by $h(x) = 0$, with h given by (2.3). Consider a discretization of the interval $[-H, 0]$ as $s_0 = -H < s_1 < \ldots < s_{N-1} < s_N = 0$, where $s_{k+1} = s_k + \sigma_k$, with $k = 0, \ldots, N-1$, and let $\{x_k\}_{k=0}^N$ be the numerical solution of (4.2) obtained by any RK scheme as above. Then*

$$s_k = h(x_k)\ ,\ \text{for}\ k = 1, \ldots, N\ ,$$

and in particular $h(x_N) = 0$. Further, $h(x_k^i) = s_k + c_i\sigma_k$, for all $i = 1, \ldots, p$, and $k = 0, \ldots, N-1$. This result is independent of the stepsizes σ_k, $k = 0, \ldots, N-1$.

Proof Rewrite the system (4.2) in the form

$$\begin{cases} \frac{dx}{dv} = g(x)\ ,\text{when } x \in R, \\ \frac{ds}{dv} = 1\ , \end{cases} \tag{4.6}$$

with initial condition $x(-H) = x_0$, $s(-H) = s_0$. Let $y = \begin{pmatrix} x \\ s \end{pmatrix}$ and rewrite the problem as:

$$\frac{dy}{dv} = G(y), \quad \text{where } G(y) = \begin{pmatrix} g(x) \\ 1 \end{pmatrix}\ ,\ y(-H) = \begin{pmatrix} x_0 \\ s_0 \end{pmatrix}\ , \tag{4.7}$$

subject to the constraint

$$h(x) - s = 0\ .$$

In other words, we have rewritten the problem so to have a linear constraint for the variable y in (4.7) and can now proceed in a similar way to what is done in the context of geometric integration of ODEs on manifods (see [14]).

Consider a RK step:

$$\begin{cases} y_1 = y_0 + \sigma_k \sum_{i=1}^p b_i G(y_0 + \sigma_0 \sum_{j=1}^p a_{ij} y_k^j) \\ y_k^i = y_0 + \sigma_0 \sum_{j=1}^p a_{ij} G(y_k^j)\ ,\ i = 1, \ldots, p\ . \end{cases} \tag{4.8}$$

Rewrite this explicitly (using $\sum_i b_i = 1$):

$$\begin{pmatrix} x_1 \\ s_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ s_0 \end{pmatrix} + \sigma_0 \begin{pmatrix} \sum_{i=1}^p b_i g(x_0 + \sigma_0 \sum_{j=1}^p\ a_{ij} x_0^j) \\ 1 \end{pmatrix}$$

where for the stage values

$$\begin{cases} x_0^i = x_0 + \tau_0 \sum_{j=1}^p a_{ij} g(x_0 + \sigma_0 \sum_{j=1}^p a_{ij} x_0^j) \\ \\ s_0^i = s_0 + \tau_0 \sum_{j=1}^p a_{ij} = s_0 + c_i\sigma_0\ , \quad i = 1, \ldots, p\ . \end{cases}$$

For us, $h(x) = d^T x + e$, and $d^T g(x_0 + \tau_0 \sum_{j=1}^p a_{ij} x_0^j) = 1$, so

$$h(x_0^i) = d^T x_0 + e + \sigma_0 \sum_{j=1}^p a_{ij} d^T g(x_0 + \sigma_0 \sum_{j=1}^p a_{ij} x_0^j) = h(x_0) + \sigma_0 \sum_{j=1}^p a_{ij} = h(x_0) + c_i\sigma_0,$$

for $i = 1,\dots,p$, that is, $s_0^i = h(x_0^i)$. Also,

$$h(x_1) = h(x_0) + \sigma_0 \sum_{i=1}^{p} b_i d^T g(x_0 + \sigma_0 \sum_{j=1}^{p} a_{ij} x_0^j) = h(x_0) + \sigma_0$$

that is $h(x_1) = s_1$. The proof now follows by realizing that we can change the index 0 with the index k in the above. □

In case of a quadratic surface, we obtain a similar result for Gauss RK methods.

Theorem 4.2 *Assume that the event surface is defined by the* 0*-set of the function* $h(x) := x^T Ax + d^T x + e$, *where* $A \in \mathbb{R}^{n\times n}$. *Consider a discretization of the interval* $[-H,0]$ *as* $s_0 = -H < s_1 < \dots < s_{N-1} < s_N = 0$, *where* $s_{k+1} = s_k + \sigma_k$, *with* $k = 0,\dots,N-1$, *and let* $\{x_k\}_{k=0}^N$ *be the numerical solution of (4.2) obtained by a RK scheme satisfying the algebraic condition*

$$b_i b_j - b_i a_{ij} - b_j a_{ji} = 0 \ , \qquad i,j = 1,\dots,p \ . \tag{4.9}$$

Then

$$s_k = h(x_k) \ , \text{ for } k = 1,\dots,N \ ,$$

and in particular $h(x_N) = 0$.

Proof As in the proof of Theorem 4.1, we use the variable $y = \begin{pmatrix} x \\ s \end{pmatrix}$, so that we have (4.7) subject to the constraint

$$x^T Ax + d^T x + e - s = 0, \quad \text{or} \quad y^T My + u^T y + e = 0$$

where

$$M = \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix} \ , \quad u^T = [d^T \ -1] \ .$$

Suppose the constraint is satisfied at y_k (this is obviously true at y_0), and consider a step of a RK method for which (4.9) holds:

$$y_{k+1} = y_k + \sigma_k \sum_{i=1}^{p} b_i G(y_k^i)$$

where:

$$y_k^i = y_k + \sigma_k \sum_{j=1}^{p} a_{ij} G(y_k^j) \ , \qquad i = 1,\dots,p \, . \tag{4.10}$$

Evaluating the constraint at y_{k+1} we have:

$$y_{k+1}^T M y_{k+1} + u^T y_{k+1} + e =$$

$$\left(y_k^T + \sigma_k \sum_{i=1}^{p} b_i G^T(y_k^i)\right) M \left(y_k + \sigma_k \sum_{i=1}^{p} b_i G(y_k^i)\right) + u^T \left(y_k + \sigma_k \sum_{i=1}^{p} b_i G(y_k^i)\right) + e$$
$$= y_k^T M y_k + u^T y_k + e + \sigma_k \left[\sum_{i=1}^{p} b_i \left(G^T(y_k^i) M y_k + y_k^T M G(y_k^i)\right) + +u^T G(y_k^i)\right]$$
$$+\sigma_k^2 \sum_{i,j=1}^{p} b_i b_j G^T(y_k^i) M G(y_k^j) \, .$$

Now, using (4.10), write $y_k = y_k^i - \sum_j a_{ij} G(y_k^j)$ and use it in the two occurrences of y_k in the bracket multiplied by σ_k, to obtain that the constraint is preserved if

$$\sigma_k \sum_{i=1}^{p} b_i \left[G^T(y_k^i) M y_k^i + (y_k^i)^T M G(y_k^i) + u^T G(y_k^i)\right] +$$
$$\sigma_k^2 \sum_{i,j=1}^{p} [b_i b_j - b_i a_{ij} - b_j a_{ji}] \, G^T(y_k^j) M G(y_k^i) = 0 \ .$$

Using (4.9), then the result follows if we show

$$G^T(y_k^i) M y_k^i + (y_k^i)^T M G(y_k^i) + u^T G(y_k^i) = 0 \ .$$

Using the forms of G, M and u, this is equivalent to say that:

$$g^T(x_k^i) A x_k^i + (x_k^i)^T A g(x_k^i) + d^T g(x_k^i) - 1 = 0 \ ,$$

which can be further rewritten (recall (4.3)) as

$$f^T(x_k^i) A x_k^i + (x_k^i)^T A f(x_k^i) + d^T f(x_k^i) - h_x^T(x_k^i) f(x_k^i) = 0 \ ,$$

and this is trivially true since $h_x^T(x) y = y^T A x + y^T A^T x + y^T d$. □

Remark 4.1

(a) In the case of $h(x) = x^T A x + d^T x + e$, under the assumptions of Theorem 4.2 we have $h(x_k) = s_k$, but in general we do not have such property for the stage values. That is, in general, $s_k^i \neq h(x_k^i)$, and we cannot infer that $h(x_k) \le h(x_k^i) \le h(x_{k+1})$. This fact can be a concern in case the internal stages end up on the other side of Σ.

(b) The result of Theorem 4.2 holds when the stage values are computed **exactly**. In practice, the nonlinear system (4.10) will be solved only with a certain accuracy, say `tol`. In this case, we may have that $h(x_k) - s_k = \mathcal{O}(k \, \texttt{tol})$, $k = 1, 2, \ldots$.

Finally, in case the function $h(\cdot)$ is neither linear nor quadratic in x, no RK scheme on the problem (4.2) will maintain the relation $s_k = h(x_k)$ exactly. Before proposing a remedy to this case of general h, we have the following result which is an immediate consequence of standard error results.

Proposition 4.1 *Suppose that h_{xxx} is not identically* 0*. Consider the problem (4.2) with g sufficiently smooth. Let $-H = s_0 < s_1 < \ldots < s_N = 0$ be a segmentation with $s_{k+1} = s_k + \sigma$, $\sigma = H/N$, $k = 0,\ldots,N-1$, and suppose we use a RK integrator of order q. Then $h(x_N) = \mathscr{O}(\sigma^q)$.*

We end up this section with some important considerations relative to working with the formulation (4.2), and the **advantages** of adopting this formulation.

Remark 4.2

(a) The overall method appears to be quite a bit simpler than the traditional event methods working with the variable "t". Of course, we need that (2.4) hold, and in practice this limits the applicability of the change of variable $s = h(x(t))$ to a neighborhood of Σ where (2.4) holds.

(b) In case of linear or quadratic function h, the method locates exactly an event point on Σ. This is in contrast with standard methods which require a zero finding routine or the continuous extensions of the numerical solution, even when Σ is a plane. Further, in the case of Σ being a plane, and proceeding with any (explicit or implicit) RK scheme with fixed stepsize, our method will reach Σ in an a-priori determined number of steps. In case of Σ being a quadratic surface, the same is true for Gauss RK methods.

(c) When $h(\cdot)$ is linear in x, and (as we said) $0 \le c_i \le 1$, for all $i = 1,\ldots,p$, then $h(x_k^i) \le h(x_k) + \sigma_k$, and there is no need to reduce the stepsize σ_k in order to stay below the value $h(x_{k+1})$ (cfr. with [9], and see also Section 5).

(d) Naturally, a result like Proposition 4.1 is also valid for the problem in the original formulation (2.2). [In fact, this type of result is always true on a finite interval]. The difference is that the end-point of integration for (2.2) is not known a-priori, unlike the case of the formulation (4.2). Indeed, in our limited experience with constant stepsizes, the formulation (4.2) is always superior to the original formulation. We expected this when Σ is a plane. But, in fairness, there may be situations where the original formulation (2.2) is advantageous, as when the region of validity of (2.4) is just a small neighborhood of Σ. For this reason, one may think of a hybrid method, switching between integrating (4.2) and (2.2).

5 Monotonicity

In case the function $h(\cdot)$ is neither linear or quadratic in x, then the subdiagonal explicit RK schemes considered in [9], on the formulation (4.2)[2], can be made monotone (for stepsize σ sufficiently small). The argument is similar to one we used in [9], and we show it just for Heun method, a RK schemes of order 3 (in the process, we will show it also for forward Euler and the explicit midpoint methods).

[2] Of course, we are assuming that (2.4) holds

Heun method is defined by the following tableau:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} .$$

With stepsize σ, one step of Heun method reads

$$x_1 \;=\; x_0 + \sigma[\frac{1}{4}g(x_0) + \frac{3}{4}g(x_0 + \frac{2}{3}\sigma g(x_0 + \frac{1}{3}\sigma g(x_0)))] \,, \tag{5.1}$$

that is

$$\begin{aligned} x_1 =&\; x_0 + \sigma[\frac{1}{4}g(x_0) + \frac{3}{4}g(x_0^3)] \,, && (5.2)\\ x_0^3 =&\; x_0 + 2\frac{\sigma}{3}g(x_0^2) \,, && (5.3)\\ x_0^2 =&\; x_0 + \frac{\sigma}{3}g(x_0) \;, \text{ and } \quad \text{so} && (5.4)\\ x_0^3 =&\; x_0 + 2\frac{\sigma}{3}g(x_0 + \frac{\sigma}{3}g(x_0)) \,. && (5.5) \end{aligned}$$

So, x_0^2 is just a step of Euler method with stepsize $\sigma/3$, and x_0^3 is one step of the explicit midpoint method with stepsize $\frac{2\sigma}{3}$. We want these two values to be such that $h(x_0^2)$ and $h(x_0^3)$ are both in R (and eventually less than, or equal to, $h(x_1)$); conditions ensuring that this can be always achieved –for σ sufficiently small– are given next.

Recall that (4.4) holds for any value x in R. In particular, this implies that for y in a sufficiently small neighborhood of x, there exist a value $0 < c < 1$ such that

$$h_x^T(y)g(x) \geq 1 - c \,. \tag{5.6}$$

Moreover, assuming that h and g are sufficiently smooth, we can always bound $\|h_x\|$, $\|Dg\|$, and $\|g\|$, in a closed ball centered at any point $x \in R$. In particular, this means that we can always assume to have lower bounds for expressions like $h_x^T(x)Dg(y)g(z)$ for $x, y, z \in R$, close to one another.

Theorem 5.1 *Let $x_0^2(\sigma) = x_0 + \frac{\sigma}{3}g(x_0)$ and $x_0^3(\sigma) = x_0 + 2\frac{\sigma}{3}g(x_0 + \frac{\sigma}{3}g(x_0))$. Then, there exists $\sigma_0 > 0$, such that, for all $0 < \sigma \leq \sigma_0$, $h(x_0^2(\sigma))$ and $h(x_0^3(\sigma))$ are strictly increasing functions of σ.*

Proof We have

$$\frac{d}{d\sigma}\, h(x_0^2(\sigma)) = \frac{\sigma}{3}h_x^T(x_0^2(\sigma))g(x_0) \,.$$

Because of (5.6), for σ sufficiently small, one has $h_x^T(x_0^2(\sigma))g(x_0) \geq 1 - \gamma_1$, with $0 \leq \gamma_1 < 1$ and the result for x_0^2 follows.

Also,

$$\frac{d}{d\sigma}\, h(x_0^3(\sigma)) = \frac{2}{3}h_x^T(x_0^3(\sigma))\left[g\left(x_0 + \frac{\sigma}{3}g(x_0)\right) + \frac{\sigma}{3}Dg(x_0 + \frac{\sigma}{3}g\,(x_0))\,g(x_0)\right] \,.$$

Now, for all σ sufficiently small, because of (5.6), there exists $\gamma_2,\ 0 \le \gamma_2 \le \frac{1}{2}$ such that
$h_x^T(x_0^3(\sigma))g\left(x_0+\frac{\sigma}{3}g(x_0)\right) \ge 1-\gamma_2$. Now, let $\rho_2 \ge 0$ such that $h_x^T(x_0^3(\sigma))Dg(x_0+\frac{\sigma}{3}g(x_0))\,g(x_0) \ge -\rho_2$. Then, the result for x_0^3 follows from the requirement

$$\frac{2}{3}(1-\gamma_2)-\frac{2\sigma}{9}\rho_2 > 0\,,$$

which is certainly true for σ sufficiently small. □

For Heun method we also have a similar monotonicity result.

Theorem 5.2 *Let* $x_1(\sigma) = x_0+\sigma[\frac{1}{4}g(x_0)+\frac{3}{4}g(x_0^3(\sigma))]$. *Then, there exists* $\sigma_0 > 0$, *such that* $h(x_1(\sigma))$ *is a strictly increasing function of* σ*, for all* $0 < \sigma \le \sigma_0$.

Proof Note that because of Theorem 5.1, we can assume that the values of the functions $x_0^2(\sigma)$ and $x_0^3(\sigma)$ are in R.

Take the derivative of $x_1(\sigma)$:

$$\begin{aligned}\frac{d}{d\sigma}h(x_1(\sigma)) \;=\; \frac{1}{4}h_x^T(x_1(\sigma))\left[g(x_0)+3g(x_0^3(\sigma))\right] \\ +\;\frac{\sigma}{4}h_x^T(x_1(\sigma))Dg(x_0^3(\sigma))g(x_0^2(\sigma))+\frac{\sigma^2}{6}h_x^T(x_1(\sigma))Dg(x_0^3(\sigma))Dg(x_0^2(\sigma))g(x_0)\,.\end{aligned}$$

Because of (4.4), for all σ sufficiently small, there exists a constant $\gamma_3,\ 0 \le \gamma_3 \le \frac{1}{2}$ such that $h_x^T(x_1(\sigma))\left[g(x_0)+3g(x_0^3(\sigma))\right] \ge 1-\gamma_3$. Also, let $\rho_3 \ge 0$ and $\eta_3 > 0$ be such that

$$\begin{aligned}h_x^T(x_1(\sigma))Dg(x_0^3(\sigma))g(x_0^2(\sigma)) &\ge -\rho_3\,,\\ h_x^T(x_1(\sigma))Dg(x_0^3(\sigma))Dg(x_0^2(\sigma))g(x_0) &\ge -\eta_3\,.\end{aligned}$$

Then, the result will follow from the requirement

$$\frac{1}{4}(1-\gamma_3)-\frac{1}{2}\sigma\rho_3-\frac{1}{6}\sigma^2\eta_3 > 0\,,$$

which is surely true for σ sufficiently small. □

Remark 5.1

(a) In [9], we had derived a result similar to Theorem 5.2, relatively to the original formulation (2.2). The key difference is that, in that context, we needed a condition guaranteeing that $h_x^T(x_1(\tau))\left[f(x_0)+3f(x_0^3(\tau))\right]/4 \ge \delta - c > 0$, with δ from (2.4). Although this is doable, it is bound to place a strong stepsize restriction on τ in case $h_x^T(x)f(x)$ is small.

(b) A result like that in Theorem 5.2 can also be given for the classical RK scheme of order 4, following the arguments of the proof of Theorem 5.2 and [9].

6 Numerical tests

Here we report on some numerical experiments, comparing the results of numerical integration of (2.2) and (4.2). Numerical integration is done with the classic 4th order RK scheme (ERK4) and/or the implicit midpoint scheme of (Gauss RK of order 2, GRK2 below), always using constant step sizes. In each problem, f refers to the vector field of (2.2) and the function h is such that $\Sigma = \{x:\ h(x) = 0\}$.

Example 6.1 Take

$$f(x) = \begin{pmatrix} x_2 \\ -x_1 + \frac{1}{1.2-x_2} \end{pmatrix}, \quad h(x) = \alpha x_1 + x_2 - \beta \sin(x_1) - 0.4 .$$

Here, α and β control the nonlinearity of Σ, and the attractivity rate to it. We report on the results obtained with ERK4, in the two cases below (linear or nonlinear Σ):

$$\begin{array}{ll} (a) & \alpha = 1\ , \beta = 0\ , (x_1(0), x_2(0)) = (-0.2, -0.2)\ ; \\ (b) & \alpha = 20\ , \beta = 20\ , (x_1(0), x_2(0)) = (-0.5, -0.5)\ . \end{array}$$

(a). Taking $N = 80$ steps on the formulation (4.7), ERK4 gives $h(x) = -5.5 \times 10^{-17}$, whereas 80 steps of ERK4 on (2.2) give $h(x) = -0.0121$ and the 81-st step gives $h(x) = 0.004$. Increasing N while solving (2.2) gives only marginal improvements; e.g., after 725068 and 725069 steps, gives $h(x) = -4.249 \times 10^{-7}$ and $h(x) = 1.3613 \times 10^{-6}$, respectively.

(b). Taking $N = 160$ steps of ERK4 on (4.7) gives $h(x) = -5.2512 \times 10^{-8}$. ERK4 applied to the original problem reaches Σ with the same accuracy only if we use extremely small constant time steps τ; for instance, if we take $\tau = 10^{-5}$ after 98434 and 98435 steps, we get $h(x) = -4.5355 \times 10^{-5}$ and $h(x) = 1.0547 \times 10^{-4}$, respectively.

Example 6.2 Take

$$f(x) = \begin{pmatrix} x_2 \\ -x_1 + 1 \end{pmatrix}\ , \quad h(x_1, x_2) = x_1^2 + x_2^2 - 5 = 0\ , \quad (x_1(0), x_2(0)) = (-1, 1)\ .$$

After 80 steps, ERK4 applied to (4.2) gives $h(x) = 2.2087 \times 10^{-8}$. 80 steps of GRK2 on (4.2), solving the nonlinear equation with tolerance $\texttt{tol} = \texttt{eps}$, gives $h(x) \approx -6.2 \times 10^{-15}$. Notice the deterioration of $\mathscr{O}(80\texttt{tol})$, as we had anticipated in Remark 4.1-(ii).

Example 6.3 This is a discontinuous differential system which models earthquake phenomena (see [6]), and was used in [16] as a test problem. We have

$$f(x_1, x_2, x_3) = \begin{pmatrix} x_2 \\ 0.5(-4.1x_2 - 210.125x_1 - u(x_1, x_2) - 2\sin(14x_3)) \\ 1 \end{pmatrix},$$

where u is the discontinuous function (below, $c = 2.47 \times 10^6$):

$$u(x_1, x_2) = \begin{cases} 0 & \text{if} \quad x_1 < 0.005 \\ c(x_1 - 0.005)^{3/2} + 1.98(c(x_1 - 0.005)^{3/2})^{1/2} x_1 & \text{if} \ \ x_1 > 0.005,\ x_2 > 0 \\ c(x_1 - 0.005)^{3/2} & \text{if} \ \ x_1 > 0.005\ x_2 < 0 \end{cases}$$

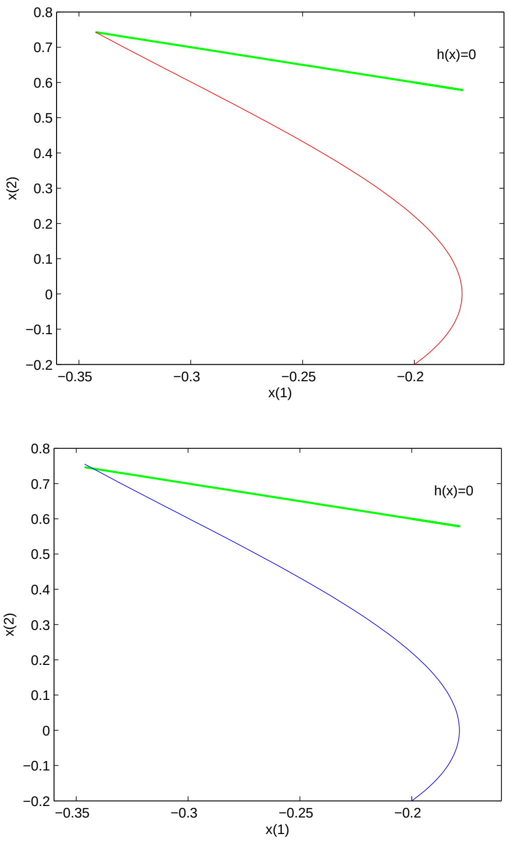

Fig. 6.1 Example 6.1-(a). ERK4 on (4.2), left, and (2.2), right.

Initial condition is $x(0) = (0.05, -0.2, 0)$. There are two discontinuity planes: $x_1 = 0.005$, and $x_2 = 0$, and –for the given initial condition– the trajectory reaches $\Sigma = \{x : \, h(x) = 0 : x_1 - 0.005 = 0\}$.

After $N = 500$ steps, ERK4 applied to (4.2) gives $h(x) \approx 1.01 \times 10^{-16}$. Here, the (relatively speaking) small stepsize of 9×10^{-5} is due to the stiffness of the problem. Using GRK2 (with tolerance $\texttt{tol} = 2.2 \times 10^{-13}$ to solve the nonlinear system), after $N = 50$ steps gives $h(x) = 1.8 \times 10^{-17}$.

7 Conclusions

Through a simple reparametrization of time, we have shown that any RK scheme produces numerical trajectories reaching exactly a planar event surface Σ which attracts nearby dynamics, in a finite number of steps. We have shown the same result for a quadratic surface Σ when using Gauss RK schemes. Further, we have discussed how to obtain monotóne schemes even when Σ is neither planar nor quadratic. Finally,

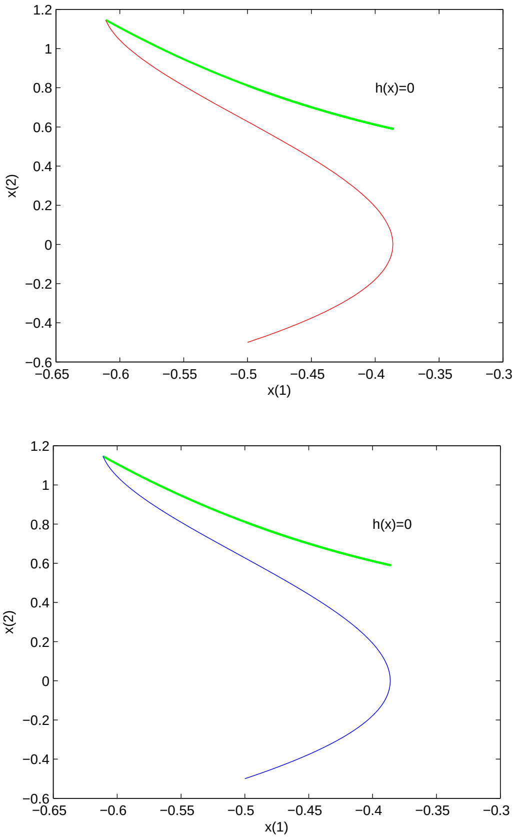

Fig. 6.2 Example 6.1-(b). ERK4 applied to (4.2), left, and (2.2), right.

we have given numerical evidence that RK schemes on the transformed problem outperform (in constant stepsize mode) the same schemes applied to the problem in the original time variable. We believe that our work will be particularly useful for the numerical integration of discontinuous Filippov-like systems, especially those arising in control engineering, where the discontinuity surface is typically a plane, to avoid undesired numerical chattering phenomena.

References

1. V. Acary and B. Brogliato. *Numerical Methods for Nonsmooth Dynamical Systems. Applications in Mechanics and Electronics*. Lecture Notes in Applied and Computational Mechanics. Springer-Verlag, Berlin, 2008.
2. V. Acary, B. Brogliato, and Y. Orlov. Chattering-free digital sliding-mode control with state observer and disturbance rejection. *IEEE Trans. Automat. Contr.*, pages 1087–1101, 2012.
3. M. Berardi and L. Lopez. On the continuous extension of Adams-Bashforth methods and the event location in discontinuous ODEs. *Applied Mathematics Letters*, 25 (6):995–999, 2012.

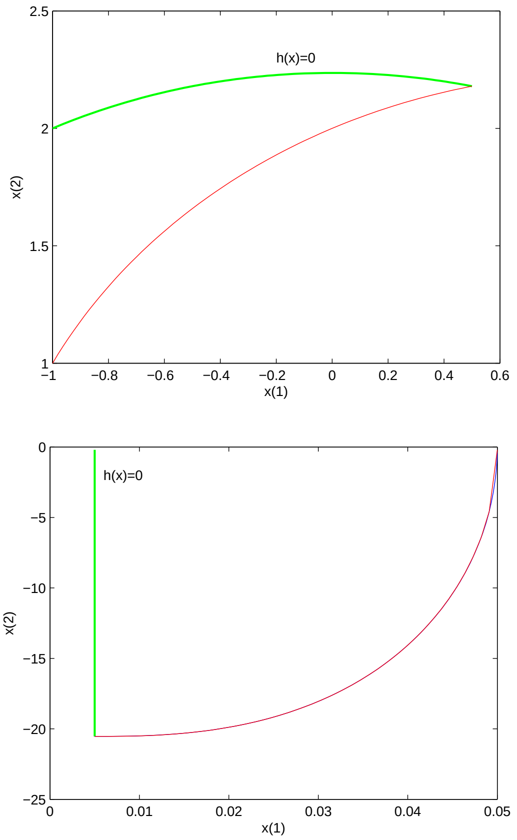

Fig. 6.3 Left: Example 6.2; ERK4 applied to (4.2). Right: Earthquake system (6.3); GRK2 on (4.2).

4. I. Boiko and L. Fridman. Analysis of chattering in continuous sliding-mode controllers. *IEEE Transactions on Automatic Control*, 50:1442–1446, 2005.
5. M. Calvo, J.L. Montijano, and L. Randez. On the solution of discontinuous IVPs by adaptive Runge-Kutta codes. *Numerical Algorithms*, 33:163–182, 2003.
6. K.T. Chau, X.X. Wei, X. Gou, and Shen C.Y. Experimental and theoretical simulations of seismic poudings between two adiacient structures. *Earthquake Engineering and Structure Dynamics*, 32:537–554, 2003.
7. L. Dieci and L. Lopez. Sliding motion in Filippov differential systems: Theoretical results and a computational approach. *SIAM J. Numer. Anal.*, 47:2023–2051, 2009.
8. L. Dieci and L. Lopez. A survey of numerical methods for IVPs of ODEs with discontinuous right-hand side. *Journal of Computational and Applied Mathematics*, 236:3967–3991, 2012.
9. L. Dieci and L. Lopez. Numerical Solution of Discontinuous Differential Systems: Approaching the Discontinuity from One Side. *Applied Numerical Mathematics*, 67:98–110, 2013.
10. J.M. Esposito and V. Kuman. A State Event Detection Algorithm for Numerically Simulating Hybrid Systems with Model Singularities. *ACM Transactions on Modeling and Computer Simulation*, 17, Issue 1:1–22, 2007.
11. Z. Feng and V.I. Utkin. Adaptive simulation and control of variable-structure control systems in sliding regimes. *Automatica*, 32:1037–1042, 1996.
12. A.F. Filippov. *Differential Equations with Discontinuous Right-Hand Sides*. Mathematics and Its Applications, Kluwer Academic, Dordrecht, 1988.

13. N. Guglielmi and E. Hairer. Computing breaking points in implicit delay differential equations. *Advances in Computational Mathematics*, 29:229–247, 2008.
14. E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: structure-preserving algorithms for ordinary differential equations*. Springer-Verlag, Berlin, 2006.
15. Y. Li and H. Wu. Global stability analysis for periodic solution in discontinuous neural networks with nonlinear growth activations. *Hindawi Publishing Corporation-Advances in Difference Equations*, doi:10.1155/20097798685:pages 14, 2009.
16. J.L. Montijano. Runge-Kutta methods for the numerical solution of discontinuous systems of Filippov type. *Talk given at Scicade Conference, Sep. 2013, Valladolid Spain*.
17. M. Najaf and R. Nikoukhah. Modeling and simulation of differential equations in Scicos. *Modelica, The Modelica Association*, September:177–185, 2006.
18. L.F. Shampine and S. Thompson. Event location for ordinary differential equations. *Computer and Mathematics with Applications*, 39:43–54, 2000.
19. M. Zhihong and X.H. Yu. Terminal sliding mode control of MIMO linear systems. *IEEE Trans. Circuits and Systems I*, 44 (11):1065–1070, 1997.