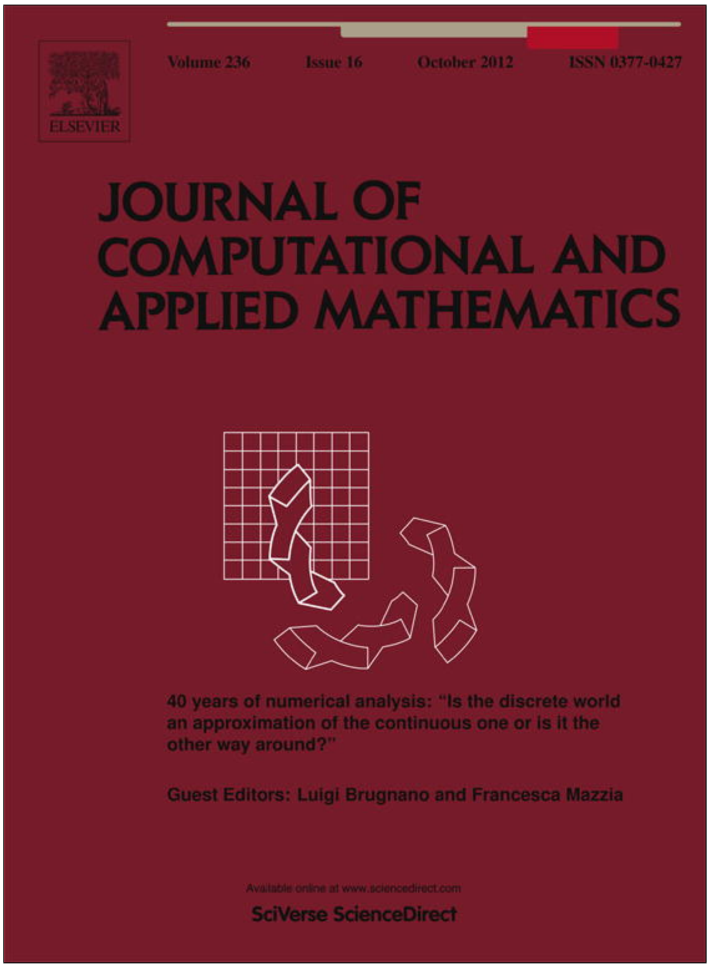
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



Volume 236 Issue 16 October 2012 ISSN 0377-0427

ELSEVIER

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

40 years of numerical analysis: "Is the discrete world an approximation of the continuous one or is it the other way around?"

Guest Editors: Luigi Brugnano and Francesca Mazzia

Available online at www.sciencedirect.com

SciVerse ScienceDirect

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Journal of Computational and Applied Mathematics 236 (2012) 3967–3991

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A survey of numerical methods for IVPs of ODEs with discontinuous right-hand side

Luca Dieci[a], Luciano Lopez[b,*]

[a] School of Mathematics, Georgia Tech Institute, Atlanta, GA 30332-0160, USA

[b] Dipartimento di Matematica, Universitá degli Studi di Bari "Aldo Moro", Via E. Orabona 4, I-70125, Bari, Italy

ARTICLE INFO

Article history:
Received 30 November 2011
Received in revised form 26 January 2012

MSC:
60H40
60H07

Keywords:
Discontinuous ODEs
One-step and multistep methods
Event driven
Time-stepping methods

ABSTRACT

This work is dedicated to the memory of Donato Trigiante who has been the first teacher of Numerical Analysis of the second author. The authors remember Donato as a generous teacher, always ready to discuss with his students, able to give them profound and interesting suggestions.

Here, we present a survey of numerical methods for differential systems with discontinuous right hand side. In particular, we will review methods where the discontinuities are detected by using an event function (so-called *event driven methods*) and methods where the discontinuities are located by controlling the local errors (so-called *time-stepping methods*). Particular attention will be devoted to discontinuous systems of Filippov's type where sliding behavior on the discontinuity surface is allowed.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction and examples

Differential systems with discontinuous right-hand sides appear pervasively in applications of various nature (see, e.g. [1–8]). For a sample of references in the context of control, see e.g. [9,10], and in the context of biological systems, see e.g. [11,12,4,8]; for works on the class of complementarity systems, see [13], for works from the point of view of bifurcations of dynamical-systems, see [14–18]; of course, see the classical Refs. [19,20,9,10] for a thorough theoretical introduction to these systems.

Because of their ubiquity in applications of biological and engineering nature, discontinuous differential systems are receiving a lot of attention. To witness, we mention the recent books [21,22] which deal with specific questions of bifurcations and numerical simulations for discontinuous differential systems. Indeed, many studies on discontinuous systems rely on simulation, and the book [21] has a nice collection of different case studies for which specific numerical techniques have been devised.

Consider the initial value problem (IVP)

$$x' = f(x), \qquad x(0) \text{ given} \tag{1.1}$$

where $f : \mathbb{R}^s \to \mathbb{R}^s$ is a given s-dimensional vector field.

When solving (1.1) numerically, traditional convergence analysis of the numerical methods relies on the assumption that the right-hand side f (hence the solution) is sufficiently smooth. However, the local truncation error analysis – that forms the basis of most stepsize control techniques – fails to be valid if there is not sufficient smoothness (locally). In particular,

[*] Corresponding author.
E-mail addresses: dieci@math.gatech.edu (L. Dieci), lopezl@dm.uniba.it (L. Lopez).

0377-0427/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2012.02.011

Author's personal copy

a numerical method may become either inaccurate or inefficient, or both, in region where discontinuities of the solution or its derivatives occur.

One strategy for treating discontinuities is simply to ignore them and to rely on the local error estimator to ensure that the error remains acceptably small. Methods of this type are known as *time stepping methods*. A different strategy is to locate the discontinuities using an *event function* $h : \mathbb{R}^s \to \mathbb{R}$, which defines a discontinuity surface $\Sigma = \{x \in \mathbb{R}^s | h(x) = 0\}$ in the state space of the differential system. (Of course, this requires knowledge of h.) Thus, when the numerical solution reaches Σ an *event point* will be located and one will restart at this point; methods of this type are known as *event driven methods*.

It is easy to appreciate that there will be classes of discontinuous problems for which it will be better to apply time-stepping methods than event driven methods and vice versa. For this reason, we will review both classes of methods in this work.

Of course, one alternative course of action to the above methods is to regularize (or smoothing) the system (e.g., see [23]). Undoubtedly, this leads to simplifications in the theory since existence and uniqueness of solutions may be derived from the classical theory of ODEs. However, small integration steps are usually required during the numerical simulation of the regularized system due to the large derivatives that replace the structural changes in the system; indeed, from a numerical point of view, the regularized system becomes quite *stiff*. Furthermore, it may also happen that regularization will lead to changing the dynamics of the original nonsmooth system (see [24]). These shortcomings notwithstanding, regularizing the system is often a reasonable thing to do in order to perform a preliminary exploration of the problem at hand.

In this paper, we give a brief, but complete, review of the main numerical techniques for solving discontinuous ODEs. We start our review with a few simple examples of differential systems with discontinuous right hand side.

Example 1.1. This example was proposed by Gear and Østerby in [25]. The discontinuity is caused by the independent variable reaching a certain value, and thus the problem is actually nonautonomous. Using the same values as in [25], we have the IVP

$$x' = f(x) = \begin{cases} 0 & \text{when } t < 40.33 \\ 100 & \text{when } t \geq 40.33, \end{cases} \tag{1.2}$$

with initial condition $x(0) = 40.33$. When t reaches the value $t = 40.33$, the vector field f changes discontinuously from the value 0 to the value 100. Obviously, one could integrate separately two IVPs, one up to $t = 40.33$ and one past it. Nevertheless, in [25] the authors are interested in showing the impact of the discontinuity on the performance of the multistep codes of Hindmarsh, [26], ignoring the breakpoint at $t = 40.33$. We report on their findings.

From the time the code first attempts to overstep the discontinuity, until it finally succeeds, it uses 118 function evaluations and takes 97 steps, 18 of which are rejected. The local error tolerance was 10^{-5} relative to x, corresponding to 4×10^{-4} in absolute measure. If the discontinuity is known to the code, the code uses no extra step to locate it. On the other hand, if the discontinuity has to be located (for instance, by a bisection type procedure), in the worse case 23 step halvings are sufficient to step past the discontinuity.

Example 1.2. Consider the following scalar discontinuous IVP

$$x' = f(x) = \begin{cases} -1, & x > 0, \\ -10, & x \leq 0, \end{cases}$$

with initial condition $x(0) = 1$, to be integrated in $[0, 2]$. The exact solution is

$$x(t) = 1 - t, \quad \text{for } 0 \leq t \leq 1; \quad \text{and} \quad x(t) = -10(t-1), \quad \text{for } 1 \leq t \leq 2,$$

and so the vector field changes from -1 to -10 at $x = 0$. We integrate this problem using the explicit midpoint method (EMM),

$$x_{k+1} = x_k + \tau f\left(x_k + \frac{\tau}{2} f(x_k)\right), \quad \text{for } k = 0, 1, \dots, N-1,$$

on the time interval $[0, 2]$. In Fig. 1, we show the global errors, at $t = 2$, of EMM applied with constant and decreasing stepsize $\tau = \frac{2}{N-1}$, for $N = 10, 20, 40, 80, 160, \dots$, and the semi-log plot of the error, from which we observe that EMM behaves like a first order method. In Fig. 2, instead, we show the stepsize sequence chosen by the `MATLAB` routine `ODE23` with `ATOL= RTOL=1.E-8`. Clearly, in proximity of the discontinuity, the code takes very small steps.

Example 1.3 (*Brick on Frictional Ramp*). This example was used by David Stewart in [27]. It describes a brick moving on a inclined ramp (see Fig. 3). Two forces act on the brick: gravity g, which would make the brick slide downward, and a friction force F which opposes sliding. According to *Coulomb Law*: "The friction equals the normal contact force times the coefficient of friction ν". In formulas, the equation of motion becomes:

$$mv'(t) = mg \sin\alpha - \nu mg \cos\alpha \,\text{sgn}[v(t)],$$

Author's personal copy

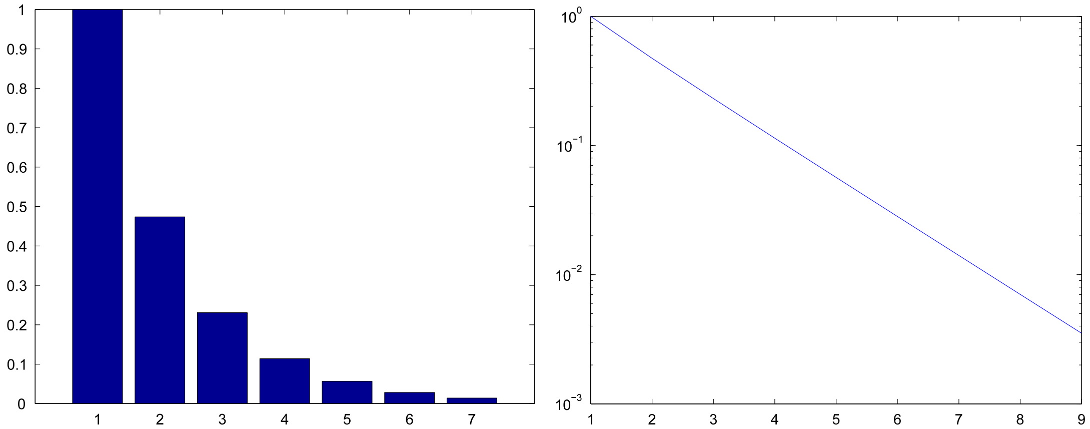


Fig. 1. Global errors (left) and semi-log global errors (right) at $t = 2$ for $\tau = 2/(N-1)$ and $k = 1, 2, 3, 4, 5, 6, 7$.

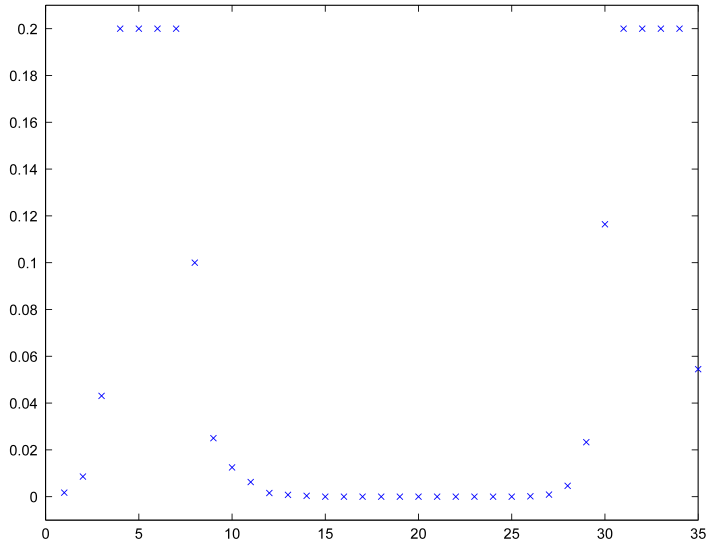


Fig. 2. Stepsize sequence against the time steps for ODE23 routine.

where "sgn" is the sign function defined to be 0 at 0:

$$\mathrm{sgn}[v] = \begin{cases} 1, & v > 0, \\ 0, & v = 0, \\ -1, & v < 0. \end{cases}$$

The key issue is whether or not one of the following two conditions is satisfied (see Fig. 4):

(i) $\sin\alpha - \nu\cos\alpha > 0$,

(ii) $\sin\alpha - \nu\cos\alpha < 0$.

Case (i) is not hard to understand: $v(t)$ increases for ever. Case (ii) is more interesting, because if $v(0) > 0$, then v decreases to 0, and if $v(0) < 0$, then v increases to 0. Then, what happens when $v = 0$? Formally, we would have $v' > 0$, which seemingly would make v grow, hence become positive, but then it would have to immediately decrease to 0. On physical grounds, we expect that the brick will stop and remain with $v = 0$ for ever. Thus, we notice that this system shows two different types of behavior: trajectories that transverse the line $v = 0$ and trajectories that remain at equilibrium on the line $v = 0$.

In the above examples, we had vector fields which become discontinuous at some point; this is the classical Filippov case (see [20]), and it is the case that we will consider in this work, so we will have the solution x continuous but x' will have a jump at the discontinuity point. (In general, discontinuous behavior may also occur in one of f's derivatives, see [25].)

Thus, we will focus on the model

$$x' = f(x) = \begin{cases} f_1(x) & \text{when } h(x) < 0 \\ f_2(x) & \text{when } h(x) > 0, \end{cases} \tag{1.3}$$

Author's personal copy

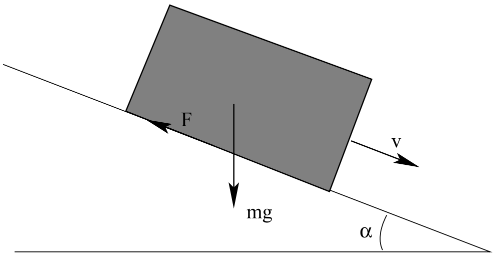


Fig. 3. Example 1.3. Moving brick.

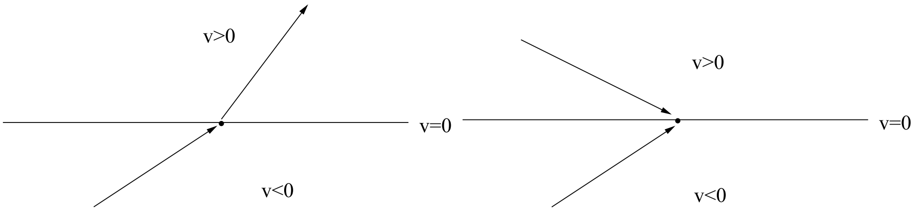


Fig. 4. Example 1.3. Orientation of the vector fields: cases (i) and (ii).

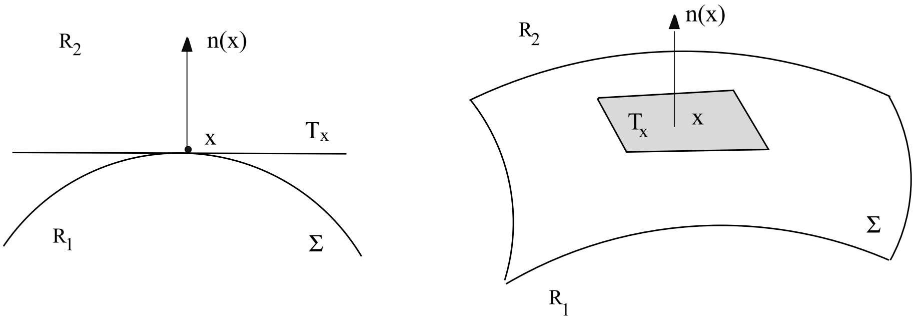


Fig. 5. The surface, tangent plane and normal vector in 1D and 2D.

where the *event function* $h : \mathbb{R}^s \to \mathbb{R}$ is known, and an initial condition $x(t_0) = x_0$ is given such that, say, $h(x_0) < 0$. In particular, at least locally, the state space $\mathbb{R}^s$ is split into two regions R_1 and R_2 by a surface Σ, where R_1, R_2, and Σ, are without loss of generality implicitly characterized as

$$\Sigma = \left\{x \in \mathbb{R}^s \mid h(x) = 0\right\}, \qquad R_1 = \left\{x \in \mathbb{R}^s \mid h(x) < 0\right\}, \qquad R_2 = \left\{x \in \mathbb{R}^s \mid h(x) > 0\right\}, \tag{1.4}$$

so that $\mathbb{R}^s = R_1 \cup \Sigma \cup R_2$. We will assume that $h \in C^k$, $k \geq 2$, and that the gradient of h at $x \in \Sigma$ never vanishes, $h_x(x) \neq 0$ for all $x \in \Sigma$. In the previous examples, the function h was actually linear (Σ was a plane). In (1.3), the right-hand side $f(x)$ can be assumed to be smooth in R_1 and R_2 separately, but it will be usually discontinuous across Σ, that is $f_1(x) \neq f_2(x)$, $x \in \Sigma$.

Many numerical methods known in literature assume that trajectories cross the surface Σ as they reach it, and that there are finitely many such crossing points (see for instance [28,25,29,30]). For this reason, we will start our review by considering discontinuous differential systems of the form (1.3) for which the *transversality condition* below is satisfied. Then, we will also consider systems in which other kinds of behaviors may appear, for example sliding motion on Σ.

Definition 1 (*Transversality at $x \in \Sigma$*). For $x \in \Sigma$, there exists $\delta > 0$ such that:

$$h_x^T(x) f_1(x) \geq \delta > 0, \qquad h_x^T(x) f_2(x) \geq \delta > 0. \tag{1.5}$$

Naturally, (1.5) reflects our choice of labeling of R_1 and R_2 done in (1.4) (see Fig. 5).

Transversality, at $x \in \Sigma$, implies that all trajectories reach the switching surface from below (or above), and then cross it. There is no solution which slides on Σ (see Section 7), and no spontaneous jump at $x \in \Sigma$.

Our definition of transversality guarantees that (locally) Σ is reached in finite time and it is particularly useful in the analysis of numerical methods, see below. In principle, we could also call *transversal* the case of

$$h_x^T(x) f_1(x) > 0, \qquad h_x^T(x) f_2(x) > 0, \tag{1.6}$$

which is the definition often adopted in the literature (e.g. [31]).

Author's personal copy

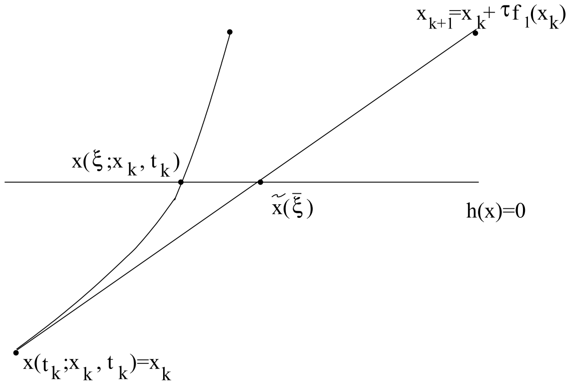


Fig. 6. Local error.

2. Explicit Euler method: local error order reduction

Here we consider the discontinuous differential system (1.3) and the explicit Euler method for its numerical solution, under the transversality assumption (1.5). As we will see, the method retains its usual order 1.

For didactical reasons, here we analyze Euler method rather than a general one-step method (see Section 3 and [28]), since it requires only minimal technical machinery and it clarifies precisely how things go in general. In the present context, the key features are:

- There is one (or several) special step(s) where the local truncation is of first order, rather than second.
- This loss of local order will not impact negatively error accumulation.

Let x_k and x_{k+1} be the approximations of the exact solution respectively at t_k and t_{k+1}, with step $\tau = t_{k+1} - t_k$. Suppose that $h(x_k)h(x_{k+1}) < 0$; this indicates that an event occurs (for the numerical scheme) in the time interval (t_k, t_{k+1}). For $t \in (t_k, t_{k+1})$, denote by $x(t; x_k, t_k)$ the exact solution of the local system $x' = f(x)$, $x(t_k) = x_k$, and let $\tilde{x}(t) = x_k + (t - t_k)f_1(x_k)$ be the continuous extension of the Euler method. Let $\bar{\xi} \in (t_k, t_{k+1})$ be the (unique) value for which $h(\tilde{x}(\bar{\xi})) = 0$. Let us further assume that there exists (unique) $\xi \in (t_k, t_{k+1})$ for which $h(x(\xi; x_k, t_k)) = 0$ (see Fig. 6).

The local truncation error is given by:

$$l_{k+1} = x(t_{k+1}; x_k, t_k) - x_{k+1} = x(t_{k+1}; x_k, t_k) - x_k - \tau f_1(x_k).$$

Taylor's expansion of the (local) exact solution gives

$$x(t_{k+1}; x_k, t_k) = x(\xi; x_k, t_k) + (t_{k+1} - \xi)f_2(x(\xi; x_k, t_k)) + O((t_{k+1} - \xi)^2),$$
$$x(\xi; x_k, t_k) = x_k + (\xi - t_k)f_1(x_k) + O((\xi - t_k)^2).$$

Hence, from the last two formulas we have:

$$\begin{aligned} l_{k+1} &= x(\xi; x_k, t_k) + (t_{k+1} - \xi)f_2(x(\xi; x_k, t_k)) - x_k - \tau f_1(x_k) + O((t_{k+1} - \xi)^2) \\ &= (\xi - t_k)f_1(x_k) + (t_{k+1} - \xi)f_2(x(\xi; x_k, t_k)) - \tau f_1(x_k) + O((\xi - t_k)^2) + O((\xi - t_{k+1})^2) \\ &= (t_{k+1} - \xi)[f_2(x(\xi; x_k, t_k)) - f_1(x_k)] + O(\tau^2). \end{aligned}$$

Now, rewrite

$$f_1(x_k) = f_1(x(\xi; x_k, t_k)) + Df_1(x(\xi; x_k, t_k))(x_k - x(\xi; x_k, t_k)) + O(\|x_k - x(\xi; x_k, t_k)\|^2) = f_1(x(\xi; x_k, t_k)) + O(\xi - t_k),$$

where Df_1 is the Jacobian matrix of f_1. Then, the local error expression becomes

$$l_{k+1} = (t_{k+1} - \xi)[f_2(x(\xi; x_k, t_k)) - f_1(x(\xi; x_k, t_k))] + O(\tau^2),$$

from which

$$\|l_{k+1}\| \leq \tau J + O(\tau^2), \tag{2.1}$$

where

$$J = \|f_1(x(\xi; x_k, t_k)) - f_2(x(\xi; x_k, t_k))\|,$$

is the jump of the vector field at the discontinuity point.

If we wish to give the local truncation error l_{k+1} in terms of the jump of the vector field at the numerical solution $\tilde{x}(\bar{\xi})$, we observe that:

(a) $x(\xi; x_k, t_k) = x_k + (\xi - t_k)f_1(x_k) + O((\xi - t_k)^2)$,
(b) $\tilde{x}(\bar{\xi}) = x_k + (\bar{\xi} - t_k)f_1(x_k)$,

Author's personal copy

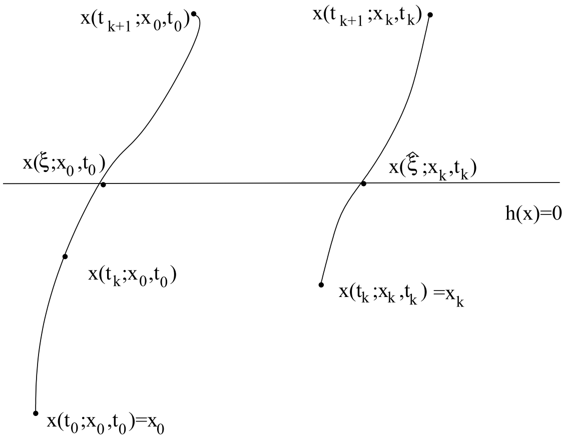


Fig. 7. Gobal error.

from which

$$x(\xi;x_k,t_k)=\tilde{x}(\bar{\xi})+(\xi-\bar{\xi})f_1(x_k)+O((\xi-t_k)^2).$$

Thus,

$$\begin{aligned} l_{k+1} &= (t_{k+1}-\bar{\xi})[f_2(\tilde{x}(\bar{\xi}))-f_1(\tilde{x}(\bar{\xi}))]+(\bar{\xi}-\xi)[f_2(\tilde{x}(\bar{\xi}))-f_1(\tilde{x}(\bar{\xi}))]+O(\tau^2), \\ &= (t_{k+1}-\bar{\xi})[f_2(\tilde{x}(\bar{\xi}))-f_1(\tilde{x}(\bar{\xi}))]+O(\tau^2), \end{aligned}$$

since $\bar{\xi}-\xi=O(\tau^2)$ (see below for a verification of this fact). Therefore:

$$\|l_{k+1}\|\leq\tau J_1+O(\tau^2),$$

where $J_1=\|f_2(\tilde{x}(\bar{\xi}))-f_1(\tilde{x}(\bar{\xi}))\|$.

Finally, let us verify that $\xi-\bar{\xi}=O(\tau^2)$. Let us observe that

$$\begin{aligned} &h(\tilde{x}(\bar{\xi}))=h(x_k)+h_x^T(x_k)(\tilde{x}(\bar{\xi})-x_k)+O(\|\tilde{x}(\bar{\xi})-x_k\|^2), \quad \text{and so} \\ &0=h(x_k)+(\bar{\xi}-t_k)h_x^T(x_k)f_1(x_k)+O((\bar{\xi}-t_k)^2), \end{aligned}$$

since $h(\tilde{x}(\bar{\xi}))=0$ and $\tilde{x}(\bar{\xi})=x_k+(\bar{\xi}-t_k)f_1(x_k)$.

Similarly, we have

$$\begin{aligned} &h(x(\xi;x_k,t_k))=h(x_k)+h_x^T(x_k)(x(\xi;x_k,t_k)-x_k)+O(\|x(\xi;x_k,t_k)-x_k\|^2) \\ &0=h(x_k)+(\xi-t_k)h_x^T(x_k)f_1(x_k)+O((\xi-t_k)^2), \end{aligned}$$

since $h(x(\xi;x_k,t_k))=0$ and $x(\xi;x_k,t_k)=x_k+(\xi-t_k)f_1(x_k)+O((\xi-t_k)^2)$.

Comparing these expressions, we get

$$0=(\xi-\bar{\xi})h_x^T(x_k)f_1(x_k)+O(\tau^2),$$

and then $\xi-\bar{\xi}=O(\tau^2)$, using $h_x^T(x_k)f_1(x_k)\neq 0$ which is inferred from (1.5).

2.1. Global error

Now, we will estimate the global error E_{k+1} at t_{k+1}. The global error (see Fig. 7) at t_{k+1} is given by

$$E_{k+1}=x(t_{k+1};x_0,t_0)-x_{k+1},$$

which may be written as

$$E_{k+1}=A_{k+1}+l_{k+1},$$

where

$$A_{k+1}=x(t_{k+1};x_0,t_0)-x(t_{k+1};x_k,t_k), \qquad l_{k+1}=x(t_{k+1};x_k,t_k)-x_{k+1},$$

and l_{k+1} is the local error.

Now, let ξ such that $h(x(\xi;x_0,t_0))=0$ and let $\hat{\xi}$ such that $h(x(\hat{\xi};x_k,t_k))=0$.

Author's personal copy

First, we are going to bound the quantity A_{k+1}.

$$\begin{aligned} A_{k+1} &= x(\xi; x_0, t_0) + (t_{k+1} - \xi) f_2(x(\xi; x_0, t_0)) - x(\hat{\xi}; x_k, t_k) - (t_{k+1} - \hat{\xi}) f_2(x(\hat{\xi}; x_k, t_k)) + O(\tau^2) \\ &= x(t_k; x_0, t_0) + (\xi - t_k) f_1(x(t_k; x_0, t_0)) - x_k - (\hat{\xi} - t_k) f_1(x_k) \\ &\quad + (t_{k+1} - \xi) f_2(x(\xi; x_0, t_0)) - (t_{k+1} - \hat{\xi}) f_2(x(\hat{\xi}; x_k, t_k)) + O(\tau^2) \\ &= E_k + (\xi - t_k)[f_1 x(t_k; x_0, t_0) - f_1(x_k)] + (t_{k+1} - \xi)[f_2 x(\xi; x_0, t_0) - f_2(x(\hat{\xi}; x_k, t_k))] \\ &\quad + (\xi - \hat{\xi}) f_1(x_k) + (\xi - \hat{\xi}) f_2(x(\hat{\xi}; x_k, t_k)) + O(\tau^2). \end{aligned}$$

The functions f_1 and f_2 are Lipschitz and bounded in the regions $R_1 \cup \Sigma$ and $R_2 \cup \Sigma$, respectively, so we may assume that there exist positive constants M_1, M_2, L_1, L_2 for which

(a) $\|f_1(x_k)\| \le M_1$,

(b) $\|f_2(x(\hat{\xi}; x_k, t_k))\| \le M_2$,

(c) $\|f_2(x(\xi; x_0, t_0)) - f_2(x(\hat{\xi}; x_k, t_k))\| \le L_2 \|x(\xi; x_0, t_0) - x(\hat{\xi}; x_k, t_k)\|$,

(d) $\|f_1(x(t_k; x_0, t_0)) - f_1(x_k)\| \le L_1 \|x(t_k; x_0, t_0) - x_k\|$,

and from (c) it follows

$$\begin{aligned} \text{(e) } \|f_2(x(\xi; x_0, t_0)) - f_2(x(\hat{\xi}; x_k, t_k))\| &\le L_2 \|x(\xi; x_0, t_0) - x(\hat{\xi}; x_k, t_k)\| \\ &\le L_2 \|x(t_k; x_0, t_0) + (\xi - t_k) f_1(x(t_k; x_0, t_0)) \\ &\quad - x_k - (\hat{\xi} - t_k) f_1(x_k) + O(\tau^2)\| \\ &\le L_2 \|E_k\| + L_1 L_2 (\xi - t_k) \|E_k\| + L_2 |\xi - \hat{\xi}| \, \|f_1(x_k)\| + O(\tau^2). \end{aligned}$$

Thus, using (a)–(e) it follows that

$$\begin{aligned} \|A_{k+1}\| &\le \|E_k\| + (\xi - t_k) L_1 \|E_k\| + (t_{k+1} - \xi) \left\{ L_2 \|E_k\| + L_1 L_2 (\xi - t_k) \|E_k\| + L_2 |\xi - \hat{\xi}| \, \|f_1(x_k)\| \right\} \\ &\quad + |\xi - \hat{\xi}| M_1 + |\xi - \hat{\xi}| M_2 + O(\tau^2) \\ &\le \left[1 + \tau L + L^2 \tau^2 / 2\right] \|E_k\| + 2|\xi - \hat{\xi}| M + t_{k+1} - \xi L_2 |\xi - \hat{\xi}| M_1 + O(\tau^2), \end{aligned}$$

where $L = \max\{L_1, L_2\}$, $M = \max\{M_1, M_2\}$.

Finally, we note that $|\xi - \hat{\xi}| = O(\tau)$, simply because Euler method has order 1. Therefore, we get the following bound

$$\|A_{k+1}\| \le \left[1 + \tau L + L^2 \tau^2 / 2\right] \|E_k\| + O(\tau). \tag{2.2}$$

Hence, since

$$\|E_{k+1}\| \le \|A_{k+1}\| + \|l_{k+1}\|,$$

we have

$$\|E_{k+1}\| = O(\tau).$$

We remark that there are two contributions to the $O(\tau)$ term in the global error. The first comes from $|\xi - \hat{\xi}| = O(\tau)$, and this is due to having used a first order method (the estimate improves to $O(\tau^p)$ for a method of order p). The second contribution, however, comes from the jump and this cannot be improved by using some higher order method.

Finally, we observe that for the next step we have

$$\|E_{k+2}\| \le (1 + \tau L_2) \|E_{k+1}\| + \|l_{k+2}\|,$$

where, now, $\|l_{k+2}\| = O(\tau^2)$, because we are in a smooth region. However, since $\|E_{k+1}\| = O(\tau)$, then of course $\|E_{k+2}\| = O(\tau)$ as well; it is noteworthy that this first order behavior does not deteriorate further (of course, we need to have just a few jumps, otherwise we will accumulate order τ contributions).

3. Mannshardt's work

We now review Mannshardt's pioneering work on the behavior of numerical schemes for discontinuous systems. As we will clarify below, Mannshardt work may be regarded as both a *time stepping* and an *event driven* method (see [28]).

The author assumes to have a discontinuous system in the form (1.3) and that the transversality condition (1.5) on Σ is satisfied. Obviously, the transversality condition implies that each discontinuity point ξ is a transition point, that is $h(x(t))$ changes sign at $t = \xi : h(x(\xi - \tau)) h(x(\xi + \tau)) < 0$, where τ is small. Of course, however, ξ depends on the solution and is not known a priori.

Author's personal copy

Similarly to what we did above for Euler method, Mannshardt observed that a Runge–Kutta (RK) method remains convergent after having crossed the discontinuity, but only with order 1. However, he also realized that one can avoid this order breakdown, if the discontinuity point is located with sufficient accuracy, and a restart process is enacted at the discontinuity point. In this sense, Mannshardt's work is in the category of event driven techniques.

For solving the discontinuous system (1.3) Mannshardt considered a one-step method:

$$x_{k+1} = x_k + \tau_k \phi(\tau_k; x_k), \quad k \geq 0,$$

where $\phi(\cdot)$ is the increment function, τ_k is the stepsize (bounded by τ), and $t_{k+1} = t_k + \tau_k$, for all $k \geq 0$. Now, let $\phi_1(\cdot)$ be the increment function of a one-step method of order p (for smooth problems) and use ϕ_1 to integrate the smooth differential system $x' = f_1(x)$ in the region R_1. In particular, consider the continuous approximation

$$\chi_1(t) = x_k + (t - t_k)\phi_1(t - t_k; x_k), \quad t \in (t_k, t_{k+1}),$$

from which

$$\chi_1(t_{k+1}) = x_k + \tau_k \phi_1(\tau_k; x_k) = x_{k+1}.$$

Since $h(x_k) < 0$, if we have $h(x_{k+1}) < 0$, then we continue to integrate, otherwise there exists a value $\bar{\xi} \in (t_k, t_{k+1})$ such that $h(\chi_1(\bar{\xi})) = 0$. In order to preserve the order p of the entire procedure, we need that the difference between the discontinuity point ξ of the theoretical solution and the one $\bar{\xi}$ of the numerical approximation to be of order p, that is $\xi - \bar{\xi} = O(\tau^p)$. We further notice that it is sufficient to compute an approximation $\tilde{\xi}$ of $\bar{\xi}$ for which

$$\tilde{\xi} - \bar{\xi} = O(\tau^{p+1}). \tag{3.1}$$

And, to find $\tilde{\xi}$, one can employ Newton's method applied to the function $\gamma(t) = h(\chi_1(t))$:

$$\xi_{i+1} = \xi_i - \frac{\gamma(\xi_i)}{\gamma'(\xi_i)} = \xi_i - \frac{h(\chi_1(\xi_i))}{h_x^T(\chi_1(\xi_i))f_1(\chi_1(\xi_i))}, \quad \text{for } i = 0, \ldots, \quad \text{and} \quad \xi_0 = t_k. \tag{3.2}$$

To reduce the expense due to function evaluations, Mannshardt proposed a simplified Newton method (where the denominator remains fixed), that is:

$$\xi_{i+1} = \xi_i - \frac{\gamma(\xi_i)}{\gamma'(t_k)} = \xi_i - \frac{h(\chi_1(\xi_i))}{h_x^T(x_k)f_1(x_k)}, \quad \text{for } i = 0, \ldots, \quad \text{and} \quad \xi_0 = t_k.$$

Mannshardt further proved that if one takes ξ_p instead of $\tilde{\xi}$ the estimate (3.1) holds.

Next, let $\tilde{x} = \chi_1(\tilde{\xi})$ be the numerically computed point on the discontinuity surface, with an error of order $O(\tau^p)$ with respect the exact solution $x(\xi; x_0, t_0)$. Then, we can restart with initial condition $(\tilde{\xi}, \tilde{x})$, solving the differential system in the region R_2. That is, we can consider the scheme

$$\chi_2(t) = \tilde{x} + (t - \tilde{\xi})\phi_2(t - \tilde{\xi}; \tilde{x}), \quad \text{for } t \in (\tilde{\xi}, t_{k+1}),$$

and let

$$x_{k+1} = \chi_2(t_{k+1})$$

where ϕ_2 is an increment function of order p (of course, this may be chosen to be the same as ϕ_1).

The key result of Mannshardt is that "*The order of this combined method is* p, *if* ϕ_1 *and* ϕ_2 *are of order* p (*applied to smooth systems*) *and* $\tilde{\xi} - \bar{\xi} = O(\tau^{p+1})$".

Mannshardt needs to assume that there is only one event in the time interval $[t_k, t_{k+1}]$, which is a reasonable assumption. In practice, if the number of events is finite in a finite time interval, then one can always choose a sufficiently small time step so that there is at most one event per step.

Finally, we notice that since the event point is computed with accuracy $O(\tau^{p+1})$, this method can be thought of as belonging to the class of time-stepping methods. (In principle, in order to have a real event driven method, the event point should be computed ideally to machine precision.)

4. Gear–Østerby method

In 1984, Gear and Østerby proposed a method in which the local error estimate is used to locate the event (see [25]). They do not assume that there is an event function h, but suppose that the values of the vector field f are given, say by a *black box routine*. Because they attempt to control the local error, their method may be regarded as one of the first *time stepping* methods in the literature.

Their idea is as follows. In a standard multistep code for ODEs, the local error is estimated at each step to decide whether to accept or reject the step, and whether to try a different step and/or order in the next step. The presence of a discontinuity is signaled by a very large value of the local error estimate resulting in the rejection of the step and a drastic reduction of the stepsize, and possibly also of the order. Back in the smooth region, it will be possible to build up the stepsize (and order)

Author's personal copy

until the code again attempts to step over the discontinuity. The stepsize is reduced again and this process may be repeated several times before the code successfully passes the trouble spot.

Here, we will recall just the case of ODEs where f is discontinuous at certain points, although in [25] the authors treat more general cases of discontinuity of f. Gear and Østerby identify four main tasks in a numerical procedure:

(1) *Detecting* the discontinuity.
(2) *Locating* the discontinuity.
(3) *Passing* the discontinuity.
(4) *Rebuilding* the data after the discontinuity is passed.

The numerical method used is a predictor–corrector in PEC mode, where both the predictor and corrector have order p. Applied to a smooth problem $x'(t) = f(x(t))$, the predictor is given by:

$$\sum_{j=0}^{m} \alpha_j^P x_{k+j} = \tau \sum_{j=0}^{m-1} \beta_j^P f(x_{k+j}), \qquad \alpha_m^P = 1,$$

while the corrector is:

$$\sum_{j=0}^{m} \alpha_j x_{k+j} = \tau \sum_{j=0}^{m} \beta_j f(x_{k+j}), \qquad \alpha_m = 1, \tag{4.1}$$

and x_{k+j} is the numerical solution at $t_{k+j} = t_k + j\tau$, $j = 0, \dots, m$.
Standard theory for the smooth case tells us that the local discretization error is

$$C_{p+1}\, \tau^{p+1}\, x^{(p+1)}(t_k) + O(\tau^{p+2}),$$

where C_{p+1} is the error constant of the corrector.

We shall first see how the discontinuity impacts the local error and the local error estimate. As already remarked, Gear and Østerby do not assume knowledge of an event function h, so that the specific discontinuity(-ies) become a function of the integration time and of the specific trajectory one is following. Nevertheless, we still suppose to be integrating a system like (1.3), with initial condition $x_0 \in R_1$, and f will change discontinuously when t reaches the value $t = \xi$ (ξ is a function of x_0, but this dependence will be omitted).

Let us now assume that the discontinuity time (say, ξ) is in $[t_{k+m-1}, t_{k+m}]$ for both the exact solution and the numerical method. So, we can write (locally) the differential system as:

$$x'(t) = f(x) = \begin{cases} f_1(x) & \text{when } t < \xi, \\ f_2(x) & \text{when } t \geq \xi, \end{cases} \tag{4.2}$$

with $f_1(\cdot) \neq f_2(\cdot)$. [We stress once more that ξ is not known in advance.] We will further assume that f_1 extends smoothly in R_2, that is f_1 has continuous derivatives of order $p+1$ also past Σ, and that the transversality condition (1.5) is satisfied.

Next, we build a smooth "solution", denoted by $x_c(t)$, such that $x_c'(t) = f_1(x_c(t))$ for $t \geq \xi$ and $x_c(t) = x(t)$ for $t < \xi$. Then, the predicted value x_{k+m}^P will not be affected by the discontinuity: x_{k+m}^P is actually an approximation to $x_c(t_{k+m})$ (see Fig. 8).

On the other hand, the corrector uses the function value $f_2(x_{k+m}^P)$, which we rewrite as

$$f_2(x_{k+m}^P) = f_1(x_{k+m}^P) + [f_2(x_{k+m}^P) - f_1(x_{k+m}^P)],$$

so that the corrected value rewrites as

$$x_{k+m} = \sum_{j=0}^{m-1} [-\alpha_j x_{k+j} + \tau \beta_j f_1(x_{k+j})] + \tau \beta_m f_1(x_{k+m}^P) + \tau \beta_m [f_2(x_{k+m}^P) - f_1(x_{k+m}^P)]. \tag{4.3}$$

From this, we observe that the quantity

$$\sum_{j=0}^{m-1} [-\alpha_j x_{k+j} + \tau \beta_j f_1(x_{k+j})] + \tau \beta_m f_1(x_{k+m}^P)$$

is the predictor–corrector value at t_{k+m} for the smooth problem $x_c'(t) = f_1(x_c(t))$. This is easy to handle, the key term to estimate/bound is $f_2(x_{k+m}^P) - f_1(x_{k+m}^P)$.

Now, if Milne's device is adopted, the local error estimate is a multiple of the corrector–predictor difference, that is

$$\begin{aligned} \gamma(x_{k+m} - x_{k+m}^P) &= \gamma \sum_{j=0}^{m-1} [(\alpha_j^P - \alpha_j) x_{k+j} + \tau(\beta_j^P - \beta_j) f_1(x_{k+j})] \\ &\quad + \gamma \tau \beta_m f_1(x_{k+m}^P) + \gamma \tau \beta_m [f_2(x_{k+m}^P) - f_1(x_{k+m}^P)] \\ &= C_{p+1}\, \tau^{p+1}\, x_c^{(p+1)}(t_k) + O(\tau^{p+2}) + \gamma \tau \beta_m [f_2(x_{k+m}^P) - f_1(x_{k+m}^P)], \end{aligned}$$

Author's personal copy

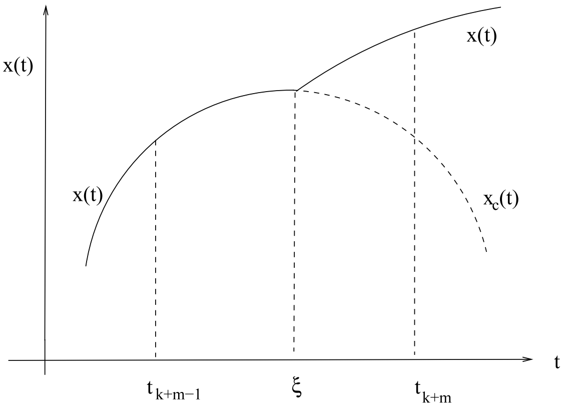


Fig. 8. Smooth continuation of x at the discontinuity point.

where the constant γ is $C_{p+1}/(C^P_{p+1}-C_{p+1})$, with C^P_{p+1} the constant error of the predictor (usually $\alpha_j=\alpha^P_j$, for $j=0,1,\dots,m$).

To estimate the term $f_2(x^P_{k+m})-f_1(x^P_{k+m})$, we observe that (again, $x(\cdot)$ and $x_c(\cdot)$ are the local solutions)

$$f_1(x^P_{k+m})=f_1(x(\xi))+Df_1(x(\xi))(x^P_{k+m}-x(\xi))+\cdots$$
$$f_2(x^P_{k+m})=f_2(x(\xi))+Df_2(x(\xi))(x^P_{k+m}-x(\xi))+\cdots$$

where $Df_1(x(\xi))$ and $Df_2(x(\xi))$ are the Jacobian matrices of f_1 and f_2 at $x(\xi)$.

Since $t_{k+m}-\xi=\theta\tau$, where $\theta\in(0,1)$, it follows that $x^P_{k+m}-x(\xi)=\theta\tau f_1(x(\xi))+O(\tau^2)$, so that

$$f_2(x^P_{k+m})-f_1(x^P_{k+m})=J+O(\tau), \tag{4.4}$$

where $J=[f_2(x(\xi))-f_1(x(\xi))]$. So, the local error estimate becomes

$$\gamma(x_{k+m}-x^P_{k+m})=\gamma\beta_m\tau J+O(\tau^2)+C_{p+1}\,\tau^{p+1}\,x_c^{(p+1)}(t_k)+O(\tau^{p+2})$$

and the dominant term, $O(\tau)$, is due to the discontinuity of f.

The idea of Gear and Østerby is that a variable step code, which determines the stepsize by using the local error estimate, will reduce the stepsize drastically because of the discontinuity, whereby automatically enforcing accuracy in spite of the decrease in smoothness of the solution.

To find out what the local error actually is, notice that

$$f_2(x(t))=f_1(x(t))+J+O(\tau),\quad t\in(\xi,t_{k+m}),$$

therefore

$$x(t_{k+m})=x_c(t_{k+m})+\theta\tau J+O(\tau^2).$$

The predictor produces an approximation of $x_c(t_{k+m})$ such that

$$x^P_{k+m}=x_c(t_{k+m})+O(\tau^{p+1}),$$

while the corrector will take the new value of f into account.

Thus, from (4.3) and (4.4) we have

$$x_{k+m}=x_c(t_{k+m})+\beta_m\tau J+O(\tau^2)+O(\tau^{p+1}),$$

so that the local error is:

$$x(t_{k+m})-x_{k+m}=(\theta-\beta_m)\tau J+O(\tau^2)+O(\tau^{p+1})$$

and, being $\beta_m\in(0,1]$, an upper bound for the dominant part of the local error is given by $\tau\|J\|$.
This expression may be used to ensure that the local error will be kept below a specified tolerance ϵ when taking a step across the discontinuity; we define the passing stepsize

$$\tau_{\text{pass}}=\frac{\epsilon}{\|J\|},$$

so that if $\tau<\tau_{\text{pass}}$, the local error when passing the discontinuity will be less than ϵ.

Detecting a discontinuity. The presence of a discontinuity will be indicated by a large value of the local error estimate (LEE). In turn, if a local error per step strategy is used, this will prompt a reduction in the stepsize according to a formula like

$$\tau_{\text{new}}=\left|\frac{\epsilon}{\|\text{LEE}\|}\right|^{1/(p+1)}\times\tau_{\text{old}}.$$

Author's personal copy

An easy detection check for a discontinuity is thus that

$$\tau_{\text{new}} \ll \tau_{\text{old}},$$

or that the local error estimate is much greater than the tolerance ϵ. However, we have to notice that these tests may become enforced even if f is not discontinuous but just varies rapidly. Once we have found that there is a discontinuity of f for $t \in (t_{k+m-1}, t_{k+m})$, then we need to locate it.

Locating a discontinuity. If an event function $h : \mathbb{R}^s \to \mathbb{R}$ exists, such that $h(x(t)) = 0$ at a discontinuity point in (t_{k+m-1}, t_{k+m}), then, if the time step is sufficiently small, we will have $h(x_{k+m-1})h(x_{k+m}) < 0$, and we can use a Newton's type method, or any other root finding method, to locate ξ approximatively. For example, multistep methods for discontinuous ODEs where an event function is used to detect the discontinuities have been used in [32,33,30].

Instead, Gear and Østerby are interested in the case in which no event function is known and the discontinuity points have to be found using values of f only. In this case, a bisection type strategy on the stepsize may be used to reduce the dimension of the interval containing ξ.

For efficiency reasons, when the stepsize is halved we should use the Nordsieck array technique in order to avoid to compute new function evaluations (see [34]).

Passing the discontinuity. Since we employ the same method as before the discontinuity point ξ was located, and we have halved the stepsize τ a number of times, say j, we expect that the local error estimate will be well below the tolerance ϵ as long as we are to the left of ξ. Even at first order, the local error should not exceed $4^{-j}\epsilon$. Therefore, if a step passes the test after at least two step halvings, say if

$$\epsilon/10 \le \|\text{LEE}\| \le \epsilon,$$

then we have passed the discontinuity and we can proceed to the restarting task.

Restarting, rebuilding data. Because of the discontinuity, the past values of x and f do not correspond to those of smooth functions, but differ by terms of order $O(\tau)$ and $O(1)$ respectively, although the local error is kept less than the tolerance ϵ. Thus, because Euler explicit/implicit methods do not use past values, a safe strategy is to use explicit Euler method as predictor and implicit Euler or the trapezoidal rule as corrector. Then, we can continue the integration using the same process adopted in variable-step, variable-order, multistep methods. Of course, for this to work we must have passed the discontinuity and must be using the correct branch of the function f.

5. Runge–Kutta time-stepping methods

Here we review the basics for the use of Runge–Kutta methods when the discontinuities are located by monitoring the local truncation errors. Runge–Kutta methods for discontinuous ODEs based on the control of the local truncation error have been proposed by several authors, see for instance [35–37,29].

Consider the discontinuous system (4.2), and take an embedded pair of explicit s-stage Runge–Kutta methods of order p and $\hat{p}$ (and below we will think of the usual case when $\hat{p} = p - 1$ or $\hat{p} = p + 1$), given by the Butcher array $A = (a_{ij}) \in \mathbb{R}^{s\times s}$, $c = (c_i) \in \mathbb{R}^s$, $b = (b_i) \in \mathbb{R}^s$, $\bar{b} = (\bar{b}_i) \in \mathbb{R}^s$; see for instance the routines ODE23 or ODE45 routines of the MATLAB ODESUITE package. Suppose that (t_k, t_{k+1}) is the interval where the discontinuity occurs, that $x(t_k) - x_k = O(\tau^{p^*})$ where $p^* = \max\{p, \hat{p}\}$, and that the transversality condition (1.5) holds. Then, on a successful step the solution is advanced with the higher order method:

$$x_{k+1} = x_k + \tau \sum_{i=1}^{s} b_i f(X_{ki}),$$

where:

$$X_{ki} = x_k + \tau \sum_{j=1}^{i-1} a_{ij} f(X_{kj}), \quad i = 1, \ldots, s.$$

Remark 5.1. We are supposing there is a nonempty subset I_1 of the index set $\{1, \ldots, s\}$ such that $f(X_{kj}) = f_1(X_{kj})$ when $j \in I_1$; that is, X_{kj}, with $j \in I_1$, is the approximation of the solution at a value of $t \in (t_k, \xi)$. Similarly, there is a nonempty subset $I_2 = \{1, \ldots, s\} - I_1$ such that $f(X_{kj}) = f_2(X_{kj})$ when $j \in I_2$; that is, X_{kj}, with $j \in I_2$ is the approximation of the solution at a value of $t \in (\xi, t_{k+1})$.

An estimate of the local truncation error (LEE) may be computed as the difference of the solutions of orders p and $\hat{p}$, that is:

$$\text{LEE} = \tau \sum_{i=1}^{s} (b_i - \bar{b}_i) f(X_{ki}).$$

Author's personal copy

The error estimate LEE is compared to the accuracy level ϵ which is derived from a user prescribed accuracy parameter `tol` by a mixture of relative and absolute criterion:

$$\epsilon = \texttt{tol}\max\{1, \|x\|_\infty\}.$$

If the error estimate is smaller than the prescribed level ϵ,

$$\text{LEE} < \epsilon, \tag{5.1}$$

then the numerical solution is advanced using the time step τ. Otherwise, the stepsize is reduced according to the usual criterion

$$\tau_{\text{new}} = C\left|\frac{\epsilon}{\|\text{LEE}\|}\right|^{1/(p^*)} \times \tau_{\text{old}},$$

where $0 < C < 1$ is some constant to ensure a cautious stepsize choice. Hence a new estimate LEE is computed and tested. For smooth systems, the quantity

$$\frac{\text{LEE}}{\tau} = \sum_{i=1}^{s} (b_i - \bar{b}_i) f(X_{ki})$$

approaches 0 when $\tau \to 0$, while for discontinuous systems (see Remark 5.1) this quantity approaches the jump of the vector field f at the discontinuity point. Thus, the algorithm can only pass the accuracy test (5.1) when the stepsize τ is decreased to

$$\tau_{\text{pass}} = \frac{\epsilon}{\|J\|},$$

where $J = \lim_{\tau\to 0}\frac{\text{LEE}}{\tau}$.

6. Event location procedure

Event location arises naturally in many models as a way of dealing with discontinuous behaviors. A nice survey of methods that have been proposed for dealing with it may be found in [38]. The task also arises in the solution of delay differential equations because of discontinuities that are induced by a lack of smoothness and propagated by the delays [39]. Even the popular `ODESUITE` of `Matlab` codes provides *event detection* as an option.

When an event function $h(x)$ exists, in order to derive an effective procedure, the routine which locates the discontinuity point $\bar{\xi}$ should be inexpensive but very accurate. For instance, in the simplified Newton method (3.2), the evaluation of $\chi_1(\tilde{\xi}_i)$ needs additional evaluations of the vector field f_1, and this could lead to an expensive procedure with a large number of function evaluations. An interpolation or a continuous extension of the numerical solution given by a one-step method can be used to locate the discontinuity point $\tilde{x}$ in the discontinuity interval (t_k, t_{k+1}). For instance, consider a s-stage Runge–Kutta method (of order p) given by the Butcher array $A = (a_{ij}) \in \mathbb{R}^{s\times s}$, $c = (c_i) \in \mathbb{R}^s$, $b = (b_i) \in \mathbb{R}^s$ and its continuous extension:

$$x_{k+1}(\sigma) = x_k + \tau\sum_{i=1}^{s} b_i^*(\sigma)K_i, \quad 0 \le \sigma \le 1,$$

where:

$$K_i = f_1\left(x_k + \tau_k\sum_{j=1}^{i-1} a_{ij}K_j\right), \quad i = 1, \dots, s$$

and $b_i^*(\sigma)$ are polynomials in σ such that $b_i^*(1) = b_i$ and

$$x_{k+1}(\sigma) - x(t_k + \sigma\tau) = O(\tau^{p^*+1}), \quad 0 \le \sigma \le 1,$$

with $p^* \le p$. Thus, in order to locate $\bar{\xi}$ we may find a root of $h(x_{k+1}(\sigma))$. Such a continuous extension does not require additional function evaluations, but it must provide a point on Σ with an error which preserves the accuracy of the underlying one-step method, that is of order $O(\tau^p)$; this means that p^* has to be equal to p (see for instance [37,40,41]). The theory of continuous Runge–Kutta methods tells us that for explicit Runge–Kutta methods of order up to $p = 3$ there is a continuous extension of the same order requiring no additional function evaluations (e.g., see [42]).

Often, in the context of linear multistep methods a continuous approximation of the numerical solution derives directly from the numerical scheme, which is based on polynomial interpolation of the vector field values; e.g., see the Adams formulas. So, in these cases, if an event function exists, it is possible to use such a continuous numerical solution to find the event point in the discontinuity interval without additional function evaluations (see [30]).

Author's personal copy

Table 1
$\gamma_i(\sigma)$ for the continuous approximation of AB methods.

i	0	1	2	3	4
$\gamma_i(\sigma)$	σ	$\frac{\sigma^2}{2}$	$\frac{\sigma^3}{6}+\frac{\sigma^2}{4}$	$\frac{\sigma^4}{24}+\frac{\sigma^3}{6}+\frac{\sigma^2}{6}$	$\frac{\sigma^5}{120}+\frac{\sigma^4}{16}+\frac{11}{72}\sigma^3+\frac{\sigma^2}{8}$

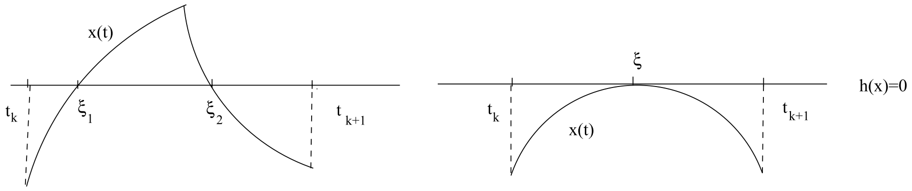


Fig. 9. Two event points in the discontinuity interval.

For instance, if the discontinuity interval is (t_k, t_{k+1}), then the m-step Adams–Bashforth method (AB), which is a method of order $p = m$, reads:

$$x_{k+1} = x_k + \tau \sum_{i=0}^{m-1} \gamma_i^* \nabla^i f_k, \tag{6.1}$$

where $\nabla^0 f_k = f_k$, $\nabla f_k = f_k - f_{k-1}$ and $\nabla^i f_k = \nabla(\nabla^{i-1} f_k)$, while $\gamma_i^* = (-1)^i \int_0^1 \binom{-r}{i} dr$, for $i = 0, 1, \dots, m-1$, (see [34]).

Thus, if we need to approximate the solution at $t_k + \sigma\tau$, for $\sigma \in (0, 1)$, it is natural to use the following continuous approximation:

$$x_{k+1}(\sigma\tau) = x_k + \tau \sum_{i=0}^{m-1} \gamma_i(\sigma) \nabla^i f_k, \quad \sigma \in (0, 1), \tag{6.2}$$

where $\gamma_i(\sigma) = (-1)^i \int_0^\sigma \binom{-r}{i} dr$, for $i = 0, 1, \dots, m-1$ (see Table 1 for the first few $\gamma_i(\sigma)$ functions). The numerical method in (6.2) may be interpreted as a variable stepsize AB method where the size of the interpolation grid is constant and equal to τ, except in the last step, when it is equal to $\sigma\tau$. It is easy to see that the local truncation error of the numerical solution (6.2) is $O(\tau^{m+1})$ for every $\sigma \in (0, 1)$. Thus, an event point may be computed as a root of the scalar polynomial (of degree m) $H(\sigma) = h(x_{k+1}(\sigma\tau))$. Of course, evaluation of (6.2) at a value $\sigma \in (0, 1)$ does not require additional function evaluations with respect to the ones need to compute x_{k+1} as noted in [43].

We conclude this discussion on event detection by cautioning that several critical situations can appear in the detection of the discontinuity points. For instance, an event may occur as a double root, or multiple events can occur if the discontinuity interval $[t_k, t_k + \tau]$ is not sufficiently small. In such cases the numerical method may fail to notice that an event has occurred because there is no change sign for $h(x)$ (see Fig. 9); because of this, the assumption that events are isolated is a more serious matter in practice than it might at first seem. From the practical and mathematical points of view, one could very small time steps or require some form of monotonicity for the numerical solution on the discontinuity interval in order to guarantee only one intersection with Σ. In [44], Carver suggests an interesting approach to event location: to add the differential equation

$$\frac{d}{dt} z = z_x^T f, \quad \text{where } z = h(x),$$

to the original differential system. This way, the stepsize will be selected not only to take into account changes in the solution $x(t)$ but also in the variable $z(t) = h(x(t))$.

When the differential discontinuous system has many discontinuity points in the time interval, then the system is said to have a *chattering behavior* (see [45,5,46]). Then, the use of a event location routine can lead to an expensive procedure and a time-stepping method may be preferable.

Example 6.1 (*Chattering*)**.** Consider the following IVP

$$\begin{cases} x_1' = -x_1 + x_2, & x_1(0) = 0, \\ x_2' = -\omega^2 x_1 - \text{sign}(x_1), & x_2(0) = 0.2, \end{cases} \tag{6.3}$$

for $\omega = 10$, where $\text{sign}(0) = [-1, 1]$. Here, we have a discontinuous system exhibiting only transversal intersections on the discontinuity line $x_1 = 0$. There is chattering behavior, caused by accumulation of event points (see Fig. 10). In this example, the solution behaves as a damped oscillator, and it approaches the origin as $t \to \infty$ with a chattering behavior around the discontinuity line $x_1 = 0$.

Author's personal copy

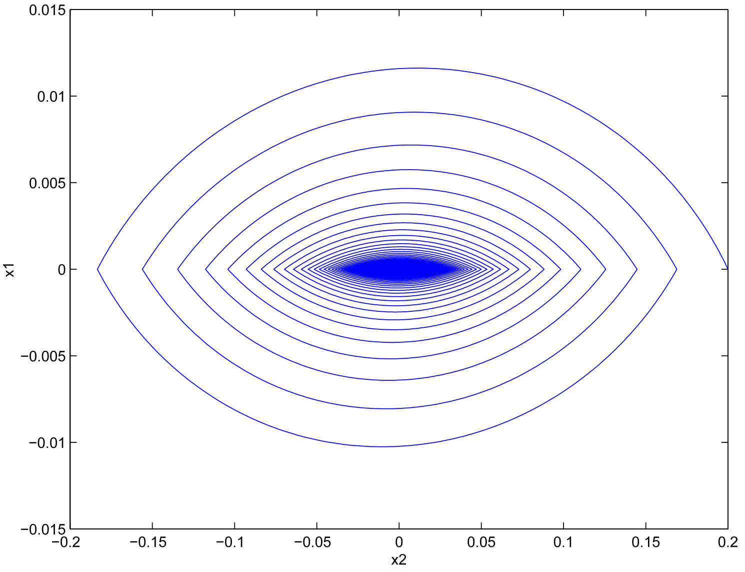


Fig. 10. Example 6.1. Chattering behavior.

7. Numerical solution of discontinuous Filippov systems

In the discontinuous differential system (1.3), $f(x)$ is not well defined when x is on the discontinuity surface Σ. A way to define the vector field on Σ is to consider the Filippov approach, that is the set valued extension $F(x)$ below:

$$x' \in F(x) = \begin{cases} f_1(x), & x \in R_1 \\ \overline{\text{co}}\,\{f_1(x), f_2(x)\}\,, & x \in \Sigma \\ f_2(x), & x \in R_2, \end{cases} \tag{7.1}$$

where $\overline{\text{co}}(A)$ denotes the smallest closed convex set containing A. In our particular case:

$$\overline{\text{co}}\,\{f_1, f_2\} = \left\{f_F : x \in \mathbb{R}^n \to \mathbb{R}^n : f_F = (1-\alpha)f_1 + \alpha f_2,\ \alpha \in [0, 1]\right\}. \tag{7.2}$$

The extension of a discontinuous system (1.3) into a convex differential inclusion (7.1) is known as *Filippov convexification*. Existence of solutions of (7.1) can be guaranteed with the notion of upper semi-continuity of set-valued functions (see [19,20]). A solution in the sense of Filippov is an absolutely continuous function $x : [0, \tau] \to \mathbb{R}^n$ such that $x'(t) \in F(x(t))$ for almost all $t \in [0, \tau]$.

Now, consider a trajectory of (1.3), and suppose that $x_0 \notin \Sigma$, and thus, without loss of generality, we can think that $x_0 \in R_1$, that is $h(x_0) < 0$. The interesting case is when, starting with x_0, the trajectory of $x' = f_1(x)$, $x(0) = x_0$, reaches Σ (in finite time). At this point, one must decide what happens next. Loosely speaking, there are two possibilities: (a) we leave Σ and enter into R_2 (or, less likely, we re-enter in R_1); (b) we remain in Σ with a well defined vector field. Filippov devised a very powerful 1st order theory which helps decide what to do in this situation, and how to define the vector field in case (b). We summarize it below.

Let $x \in \Sigma$ and let $n(x)$ be the unit normal to Σ at x, that is $n(x) = \frac{h_x(x)}{\|h_x(x)\|}$. Let $n^T(x)f_1(x)$ and $n^T(x)f_2(x)$ be the components of $f_1(x)$ and $f_2(x)$ onto the normal direction (see Fig. 11).

Transversal intersection. (See Definition 1.) In case in which, at $x \in \Sigma$, we have

$$\left(n^T(x)f_1(x)\right)\left(n^T(x)f_2(x)\right) > 0, \tag{7.3}$$

then we will leave Σ. We will enter R_1, when $n^T(x)f_1(x) < 0$, and will enter R_2, when $n^T(x)f_1(x) > 0$. In the former case we will have (1.3) with $f = f_1$, in the latter case with $f = f_2$. Any solution of (1.3) with initial condition not in Σ, reaching Σ at a time $\bar{t}$, and having a transversal intersection there, exists and is unique.

Sliding mode. In case in which, at $x \in \Sigma$, we have

$$\left(n^T(x)f_1(x)\right) \cdot \left(n^T(x)f_2(x)\right) < 0, \tag{7.4}$$

then we have a so-called sliding mode through x.

An *attracting Sliding Mode* occurs if

$$n^T(x)f_1(x) > 0 \quad \text{and} \quad n^T(x)f_2(x) < 0, \quad x \in \Sigma, \tag{7.5}$$

Author's personal copy

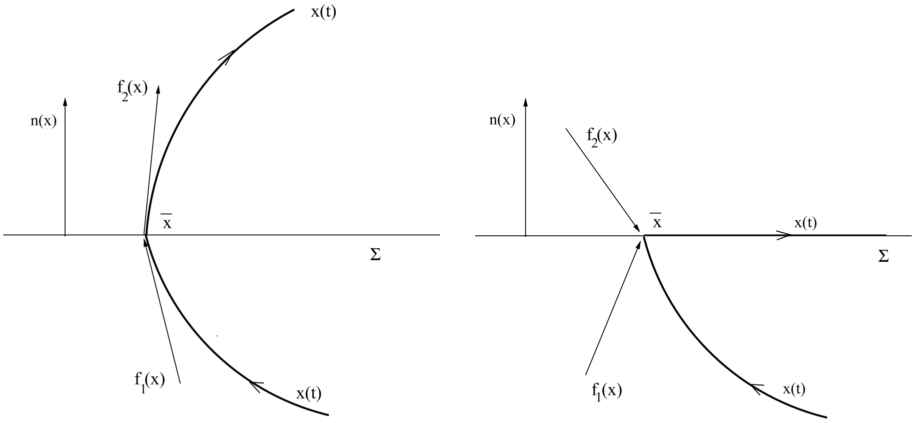


Fig. 11. Vector field orientation: transversal and sliding case.

where the inequality signs depend of course on the definition of $R_{1,2}$ in (1.4). When (7.5) is satisfied at $x_0 \in \Sigma$, a solution trajectory which reaches x_0 does not leave Σ, and will therefore have to move along Σ: *sliding motion*. During the sliding motion the solution will continue along Σ with time derivative f_F given by:

$$f_F(x) = (1 - \alpha(x))f_1(x) + \alpha(x)f_2(x). \tag{7.6}$$

Here, $\alpha(x)$ is the value for which $f_F(x)$ lies in the tangent plane T_x of $h(x)$ at x, that is the value for which $n^T(x)f_F(x) = 0$. This gives

$$\alpha(x) = \frac{n^T(x)f_1(x)}{n^T(x)(f_1(x) - f_2(x))}. \tag{7.7}$$

Observe that a solution having an attracting sliding mode exists and is unique, in forward time.

We have a *repulsive sliding mode* when

$$n^T(x)f_1(x) < 0 \quad \text{and} \quad n^T(x)f_2(x) > 0, \quad x \in \Sigma. \tag{7.8}$$

Repulsive sliding modes do not lead to uniqueness (at any instant of time one may leave with f_1 or f_2), and we will not further consider repulsive sliding motion in this work.

Summarizing, in this section we consider solutions of (1.3) which will exhibit either transversal intersection or attractive sliding mode on Σ. These will generally be continuous, but not differentiable, functions. Moreover, we will henceforth focus on the case in which we reach Σ coming from R_1 and we will restrict to the case in which f_1 reaches Σ not tangentially. To be precise, we will characterize the attractivity of Σ from R_1 by the following assumption:

There exists a strictly positive constant δ, such that for all $x \in R_1 \cup \Sigma$, and sufficiently close to Σ, we have

$$h_x^T(x)f_1(x) \geq \delta > 0. \tag{7.9}$$

Observe that, since (for a trajectory in R_1)

$$\frac{d}{dt}h(x) = h_x^T(x)x' = h_x^T(x)f_1(x),$$

then (7.9) implies that the function h monotonically increases along a solution trajectory in R_1 (and close to Σ), until eventually the trajectory hits Σ non-tangentially. A "discrete analog" of this property is the key to producing appropriate numerical schemes.

7.1. A numerical method

The discussion in this section is done with reference to the model problem (1.3), and we will use the notation therein. We will be mainly concerned with the different tasks of a numerical procedure for a discontinuous differential system of Filippov type where different behaviors (transversal intersections, sliding motions, exits from the discontinuity surface, etc.) are allowed. A MATLAB code for the numerical solution of Filippov systems, based on an event driven method, may be found in [47].

A numerical procedure for this type of problem will need to accomplish the following tasks (see [48]):

(i) *Integration outside Σ*;
(ii) *Accurate location of points on Σ reached by a trajectory*;
(iii) *Control of the transversality or sliding condition as one reaches Σ*;
(iv) *Integration on Σ (sliding mode)*;
(v) *Control of exit conditions and decision of whether or not we should leave Σ*.

Author's personal copy

In order to explain each task, as model scheme we consider the explicit midpoint rule, which is a 2nd order Runge–Kutta scheme with a simple continuous extension of the same order. The extension itself is useful to find the event points, that now will be both *entry* and *exit* points to the surface Σ. The basic scheme for $x' = f_1(x)$, with stepsize τ, and initial value $x_0 \in R_1$, has the form

$$x_1 = x_0 + \tau f_1(x_{02}), \qquad x_{02} = x_0 + \frac{\tau}{2} f_1(x_0), \tag{7.10}$$

and the second order continuous extension is

$$x_1(\sigma) = x_0 + \sigma\left[\left(1 - \frac{\sigma}{\tau}\right) f_1(x_0) + \frac{\sigma}{\tau} f_1(x_{02})\right], \quad \forall \sigma \in [0, \tau]. \tag{7.11}$$

7.2. Integration outside Σ

Integration of (1.3) while the solution remains in R_1, or R_2, is not different than standard numerical integration of a smooth differential system. Therefore, the only interesting case to consider is when, while integrating (say) the system with f_1, we end up reaching/crossing the surface Σ, that is until $h(x_0)h(x_1) < 0$. If we have $h(x_1) = 0$, it means that the discontinuity (x_1) point is already found.

7.3. Location of points on the surface Σ

We have to find a root $\bar{\xi}$ of the scalar function

$$H(\sigma) = h(x_1(\sigma)), \tag{7.12}$$

$x_1(\sigma)$ given by (7.11) where σ will belong to the interval $(0, \tau)$. It is desirable to find a root $\bar{\xi}$ within machine precision, to ensure that the point $x_1(\bar{\xi})$ is on Σ and avoid numerical oscillations during integration on Σ. Of course, a simple bisection approach can be used, but we eventually resorted (in order to have a faster convergence) to the secant method:

$$\xi_{i+1} = \xi_i - \frac{(\xi_i - \xi_{i-1})}{H(\xi_i) - H(\xi_{i-1})} H(\xi_i), \quad i \geq 0, \qquad \xi_0 = 0, \qquad \xi_1 = \tau.$$

Remark 7.1. As we have observed in Section 6, by using the continuous extension (7.11), we avoid computing the vector field f_1 except at points where we did for the original scheme.

7.4. Integration on Σ

Once we have a point $\bar{x}$ on Σ, we have to decide if we will need to cross Σ or slide on it. Letting

$$g_i(x) = n^T(x) f_i(x), \quad i = 1, 2,$$

then we check if $g_1(\bar{x})g_2(\bar{x})$ is (strictly) positive or negative. If $g_1(\bar{x})g_2(\bar{x}) > 0$, then we integrate the system:

$$x' = f_2(x), \qquad x(0) = \bar{x}. \tag{7.13}$$

Instead, if $g_1(\bar{x})g_2(\bar{x}) < 0$, we will have an attractive sliding mode and integrate the system:

$$x' = f_F(x), \qquad x(0) = \bar{x}, \tag{7.14}$$

where $f_F(x)$ is the sliding Filippov vector field.

Suppose that $\bar{x}$ is on Σ and that we have a sliding mode solution. When we compute the approximation x_1 of the solution $x(\tau)$ by the explicit midpoint method, in general the vector x_1 will not lie on Σ. To remedy this situation, we project the value x_1 back onto Σ, so to avoid that the numerical solution leaves it. Moreover, even the intermediate stage value $x_0 + \frac{\tau}{2} f_F(x_0)$ in general will not be on Σ, and thus before computing x_1 we project the stage value onto Σ as well. Succinctly, one step of the projected midpoint scheme on Σ is expressed as:

1. $\hat{x}_{02} = x_0 + \frac{\tau}{2} f_F(x_0)$;
2. $x_{02} = P(\hat{x}_{02})$;
3. $\hat{x}_1(\tau) = x_0 + \tau f_F(x_{02})$;
4. $x_1(\tau) = P(\hat{x}_1(\tau))$;

where $P(y)$ denotes the Euclidean projection onto Σ. In a similar way, we define the projected continuous extension of the method as

$$x_1(\sigma) = P\left(x_0 + \sigma\left[\left(1 - \frac{\sigma}{\tau}\right) f_1(x_0) + \frac{\sigma}{\tau} f_1(x_{02})\right]\right), \quad \sigma \in [0, \tau], \tag{7.15}$$

where it is understood that the value x_{02} is the projected value.

It is worth observing that the projection operator does not change the overall order of the method which remains 2. (Of course, if $h(x)$ is linear with respect to x, that is if Σ is flat, then no projection is required because the numerical solution $x_1(\tau)$ will automatically remain on Σ.) The issue of how to do the projection, and its associated expense, is discussed in Section 7.6.

Author's personal copy

7.5. Exit conditions

While we integrate on Σ, we will monitor if we have to continue sliding on it, or if we need to leave Σ. Once the point x_1 on Σ has been computed, we need to check the first order exit conditions: that is, if $g_1(x_1)g_1(x_0) < 0$ or $g_2(x_1)g_2(x_0) < 0$. If neither of these is true, we continue integrating on Σ. To fix ideas, suppose, instead, that $g_1(x_1)g_1(x_0) < 0$. In this case, we seek a zero of the function

$$g_1(x_1(\sigma)), \quad \sigma \in [0, \tau], \text{ with } x_1(\sigma) \text{ from (7.15).}$$

Notice that the function $g_1(x_1(\sigma))$ depends continuously on σ and changes sign at the endpoints. As before, we may use the secant method to find a root. Once this zero is found, sat at $\bar{\sigma}$, we will leave Σ and proceed integrating in R_1 (assuming that $g_2(x_1(\bar{\sigma})) < 0$). Similar reasoning applies if it is g_2 to change sign at x_0 and x_1.

7.6. Projection on Σ

The projection on Σ is done in the standard way (e.g., see [38,49]), with some simplifications due to the specific nature of our problem.

If $\hat{x}$ is a point close to Σ, then the projected vector $x = P(\hat{x})$ on Σ is the solution of the following constrained minimization problem

$$\min_{x\in\Sigma} e(x), \quad \text{where } e(x) = \frac{1}{2}(\hat{x} - x)^T(\hat{x} - x).$$

By using the Lagrange's multiplier's method, we have to find the root of

$$G(x, \lambda) = \begin{pmatrix} e_x(x) + \lambda h_x(x) \\ h(x) \end{pmatrix},$$

where $\lambda \in \mathbb{R}$. Consider Newton's method to compute the root of $G(x, \lambda)$:

$$G'(x^k, \lambda^k) \begin{pmatrix} \Delta x^k \\ \Delta\lambda^k \end{pmatrix} = -G(x^k, \lambda^k), \quad k \geq 0,$$

where $\Delta x^k = x^{k+1} - x^k$, $\Delta\lambda^k = \lambda^{k+1} - \lambda^k$, for $k \geq 0$, and

$$G'(x, \lambda) = \begin{pmatrix} I + \lambda h_{xx}(x) & h_x(x) \\ h_x^T(x) & 0 \end{pmatrix},$$

where h_{xx} is the Hessian matrix of h.

To avoid having to solve a true linear system at each k, we actually use the following simplified Newton iteration

$$\begin{pmatrix} I & h_x(x^k) \\ h_x^T(x^k) & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta\lambda^k \end{pmatrix} = -\begin{pmatrix} \hat{x} - x^k + \lambda^k h_x(x^k) \\ h(x^k) \end{pmatrix};$$

this is legitimate, since we expect that the value of λ will be close to 0 and a few iterates are typically needed to converge to the point on Σ. Observe that the linear system we solve has a coefficient matrix with a simple structure and a simple factorization: $\begin{pmatrix} I & b \\ b^T & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ b^T & 1 \end{pmatrix} \begin{pmatrix} I & b \\ 0 & -b^T b \end{pmatrix}$.

Remark 7.2. As alternative to projecting onto Σ, in order to remain close to Σ one may modify the system as in

$$x' = f_F(x) - \mu h(x) h_x(x), \qquad x(0) = \bar{x},$$

(μ is a suitable positive constant) which is a technique used in numerical methods for differential–algebraic equations (see [50]).

7.7. The oscillatory behavior of an explicit method

We need to stress that if, during the sliding mode, the numerical solution obtained by the explicit midpoint scheme (or by any other explicit method) is not projected back onto Σ, then oscillations around Σ will appear (numerical chattering behavior).

Example 7.3. In order to observe this phenomenon, let us consider the brick problem of Example 1.3 and suppose that sliding conditions are satisfied. Let us fix two different initial conditions $v_0 = \pm 1$, $m = 1$, $\theta = \pi/6$ and $\nu = 1$. In Fig. 12 we show the numerical approximations of $v(t)$, corresponding to the initial conditions $v_0 = 1$ and $v_0 = -1$, obtained by using the explicit midpoint method applied with stepsize $\tau = 1/100$. Similar results are obtained with smaller stepsize.

Author's personal copy

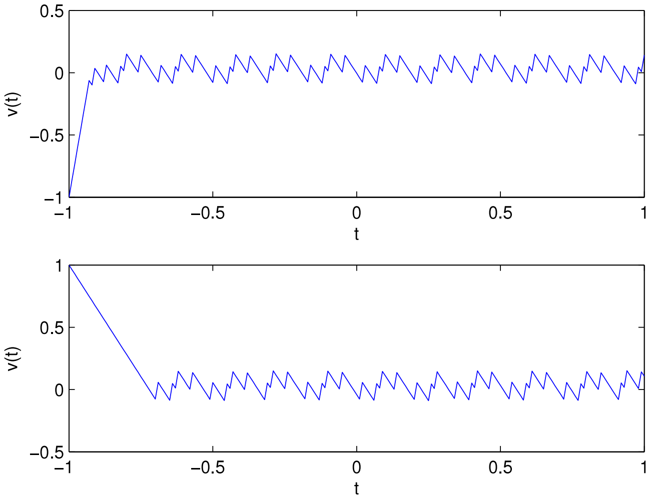


Fig. 12. Attractive sliding without projection.

Now, we are going to show that these oscillations will disappear when $\tau \to 0$. More precisely, below we will show that *the average of the numerical vector field approaches the Filippov sliding vector field when* $\tau \to 0$.

Utkin in [10] clarified this behavior for the explicit Euler scheme. Below, we prove the result for the explicit midpoint method. To proceed, we need to assume that the vector field $f_1(x)$ $(f_2(x))$ may be evaluated at points above (below), but close to, the discontinuity surface Σ. This is the main difference with respect the explicit Euler method where this extra assumption is not required.

Let $\delta > 0$ be sufficiently small and consider the surfaces

$$\Sigma_{\pm\delta} = \left\{x \in \mathbb{R}^n \mid h(x) = \pm\delta\right\}. \tag{7.16}$$

Let x_0 be an initial point on $\Sigma_{-\delta}$ (that is $h(x_0) = -\delta$) and consider one step of the explicit midpoint method with stepsize τ:

$$x_1 = x_1(\tau), \quad x_1(\tau) = x_0 + \tau f_1\left(x_0 + \frac{\tau}{2} f_1(x_0)\right). \tag{7.17}$$

Now, if x_1 is in the region R_2, then we would be getting the next approximation x_2 as

$$x_2 = x_2(\tau), \quad x_2(\tau) = x_1 + \tau f_2\left(x_1 + \frac{\tau}{2} f_2(x_1)\right). \tag{7.18}$$

Let us suppose that Σ is attracting all along the numerical trajectory $x_1(\chi)$, as expressed by the following condition:

$$h_x^T(x_1(\chi))x_1'(\chi) > 0, \quad \chi \in [0, \tau]. \tag{7.19}$$

Similarly, let us suppose that Σ is attracting all along the numerical trajectory $x_2(\chi)$ as expressed by:

$$h_x^T(x_2(\chi))x_2'(\chi) < 0, \quad \chi \in [0, \tau]. \tag{7.20}$$

Notice that, because of the attractivity of the discontinuity surface Σ, the exact solution satisfies conditions similar to (7.19) and (7.20); see (7.9).

By Taylor's expansion, we can write

$$h(x_1(\tau)) = h(x_0) + \tau h_x^T(x_1(\chi))x_1'(\chi), \quad \text{where } x_1(\chi) \in [x_0, x_1(\tau)], \tag{7.21}$$

and now we consider the value τ_1 which moves x_0 on Σ_δ, that is

$$\tau_1 = \frac{2\delta}{h_x^T(x_1(\chi_1))x_1'(\chi_1)}, \quad x_1(\chi_1) \in [x_0, x_1(\tau_1)], \tag{7.22}$$

and τ_1 is strictly positive because of (7.19).

Author's personal copy

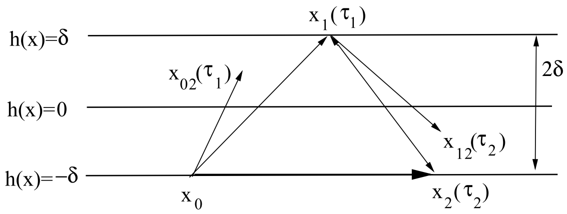


Fig. 13. Explicit midpoint method around a discontinuity surface.

We now apply the explicit midpoint method at $x_1(\tau_1)$, which is a point in the region R_2:

$$x_2(\tau) = x_1(\tau_1) + \tau f_2\left(x_1(\tau_1) + \frac{\tau}{2} f_2(x_1(\tau_1))\right). \tag{7.23}$$

Again, by Taylor's expansion

$$h(x_2(\tau)) = h(x_1(\tau_1)) + \tau h_x^T(x_2(\chi))x_2'(\chi), \quad \text{where } x_2(\chi) \in [x_1(\tau_1), x_2(\tau)], \tag{7.24}$$

and we consider the value τ_2 which moves $x_1(\tau_1)$ on $\Sigma_{-\delta}$, that is:

$$\tau_2 = \frac{-2\delta}{h_x^T(x_2(\chi_2))x_2'(\chi_2)}, \quad x_2(\chi_2) \in [x_1(\tau_1), x_2(\tau_2)], \tag{7.25}$$

and τ_2 is strictly positive because of (7.20); see Fig. 13.

Observe that

$$\tau_1 + \tau_2 = 2\delta \frac{h_x^T(x_2(\chi_2))x_2'(\chi_2) - h_x^T(x_1(\chi_1))x_1'(\chi_1)}{h_x^T(x_1(\chi_1))x_1'(\chi_1) \cdot h_x^T(x_2(\chi_2))x_2'(\chi_2)}, \tag{7.26}$$

and therefore

$$\begin{aligned}
\frac{x_2(\tau_2) - x_0}{\tau_2 + \tau_1} &= \frac{x_2(\tau_2) - x_1(\tau_1) + x_1(\tau_1) - x_0}{\tau_2 + \tau_1} \\
&= \frac{1}{2\delta} \frac{h_x^T(x_1(\chi_1))x_1'(\chi_1) \cdot h_x^T(x_2(\chi_2))x_2'(\chi_2)}{h_x^T(x_2(\chi_2))x_2'(\chi_2) - h_x^T(x_1(\chi_1))x_1'(\chi_1)} \\
&\quad \times \left[\frac{2\delta}{h_x^T(x_1(\chi_1))x_1'(\chi_1)} f_1(x_{02}(\tau_1)) - \frac{2\delta}{h_x^T(x_2(\chi_2))x_2'(\chi_2)} f_2(x_{12}(\tau_2))\right] \\
&= \frac{h_x^T(x_2(\chi_2))x_2'(\chi_2)}{h_x^T(x_2(\chi_2))x_2'(\chi_2) - h_x^T(x_1(\chi_1))x_1'(\chi_1)} f_1(x_{02}(\tau_1)) \\
&\quad - \frac{h_x^T(x_1(\chi_1))x_1'(\chi_1)}{h_x^T(x_2(\chi_2))x_2'(\chi_2) - h_x^T(x_1(\chi_1))x_1'(\chi_1)} f_2(x_{12}(\tau_2))
\end{aligned}$$

where $x_{02}(\tau_1) = x_0 + \frac{\tau_1}{2} f_1(x_0)$, and $x_{12}(\tau_2) = x_1(\tau_1) + \frac{\tau_2}{2} f_2(x_1(\tau_1))$.

Now, when $\delta \to 0$, then $x_0 \in \Sigma$, $\tau_1, \tau_2 \to 0$ and $x_{02}(\tau_1), x_{12}(\tau_2), x_1(\tau_1), x_1(\chi_1), x_2(\tau_2), x_2(\chi_2) \to x_0, x_1'(\chi_1) \to f_1(x_0), x_2'(\chi_2) \to f_2(x_0)$. Hence, it follows that

$$\lim_{\delta \to 0} \frac{x_2(\tau_2) - x_0}{\tau_2 + \tau_1} = \frac{h_x^T(x_0)f_2(x_0)}{h_x^T(x_0)f_2(x_0) - h_x^T(x_0)f_1(x_0)} f_1(x_0) - \frac{h_x^T(x_0)f_1(x_0)}{h_x^T(x_0)f_2(x_0) - h_x^T(x_0)f_1(x_0)} f_2(x_0),$$

that is

$$\lim_{\delta \to 0} \frac{x_2(\tau_2) - x_0}{\tau_2 + \tau_1} = (1 - \alpha(x_0))f_1(x_0) + \alpha(x_0)f_2(x_0),$$

where

$$\alpha(x_0) = \frac{h_x^T(x_0)f_1(x_0)}{h_x^T(x_0)f_1(x_0) - h_x^T(x_0)f_2(x_0)},$$

given by (7.7). Thus

$$\lim_{\delta \to 0} \frac{x_2(\tau_2) - x_0}{\tau_2 + \tau_1} = f_F(x_0)$$

which is the Filippov's sliding vector as claimed.

Author's personal copy

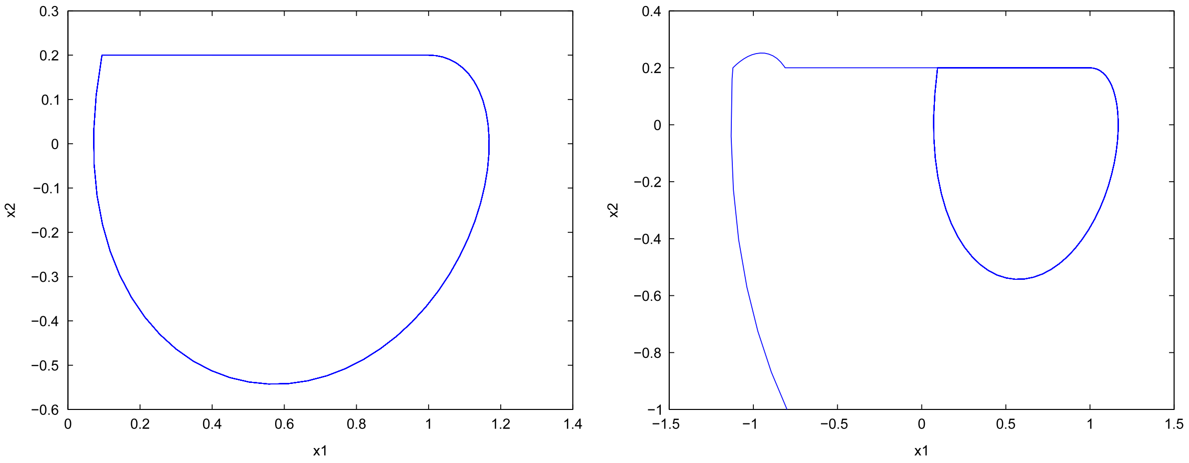


Fig. 14. Limit cycle with sliding segment (left) and approaching it through crossing and sliding (right).

Example 7.4 (*Sliding on a Line-Segment*)**.** This simple example is one which we can understand by hand calculation and it is helpful to illustrate the different tasks of our numerical procedure. It is an example in the same flavor of a problem in [24,17] (the so-called *stick-slip* system). We have the two-dimensional system

$$x' = \begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{cases} f_1(x), & h(x) < 0, \\ f_2(x), & h(x) > 0, \end{cases}$$

with

$$f_1(x) = \begin{pmatrix} x_2 \\ -x_1 + \dfrac{1}{1.2 - x_2} \end{pmatrix}, \qquad f_2(x) = \begin{pmatrix} x_2 \\ -x_1 - \dfrac{1}{0.8 + x_2} \end{pmatrix},$$

and the surface Σ is defined by the zero set of $h(x) = x_2 - 0.2$. We notice that $h_x(x) = [0\ 1]^T$, and thus on Σ we have

$$h_x^T(x)f_1(x) = -x_1 + 1, \qquad h_x^T(x)f_2(x) = -x_1 - 1,$$

and so there will be an attractive sliding mode on Σ when $x_1 \in (-1, 1)$. The sliding vector field on Σ is

$$f_F(x) = \begin{pmatrix} x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}.$$

Thus, on Σ, the x_1-component of the solution will grow linearly until reaching the value $x_1 = 1$, at which point the trajectory will leave Σ, with vector field f_1. In Fig. 14 we show, in the phase space, the numerically computed limit cycle for this problem, as well as a typical trajectory which reaches the limit cycle through previous crossing of Σ and sliding on it. Obviously, at the points in which we enter the surface Σ there is lack of differentiability of the solution, whereas at the value $x_1 = 1$, the solution leaves the surface differentiably.

Remark 7.5. Recently, Filippov's systems with solutions sliding on the intersection of more than one discontinuity surface have also been studied. In this case, the still unsettled challenging task is to be able to define the sliding vector field on the discontinuity surface, since Filippov's theory leads to ambiguous sliding vector fields (see [48,51]).

8. One-sided methods

In some instances, the vector field f_1 (respectively, f_2) cannot be extended smoothly outside $R_1 \cup \Sigma$ (respectively, $R_2 \cup \Sigma$), or, even if it may be extended, the physical features of the model may prohibit evaluation of f_1 above (respectively, f_2 below) Σ. See, for instance, the interesting examples in Mechanics and Robotics given in [32,33,52].

Example 8.1. The following dynamical system exemplifies this situation:

$$x' = \begin{cases} x\big(1-t\big)^{(2k+1)/2}, & k = 0, 1, 2, \ldots, \text{ when } t \le 1 \\ 0, \quad \text{when } t > 1. \end{cases} \tag{8.1}$$

Note that the vector field in (8.1) has k continuous derivatives at $t = 1$. The example is an extension of one considered in [53] where the authors considered the case of $k = 0$.

Author's personal copy

This system can be written in the form of (1.3) by letting $x_1 = x$, $x_2 = t$:

$$f_1(x_1, x_2) = \begin{pmatrix} x_1(1-x_2)^{\frac{2k+1}{2}} \\ 1 \end{pmatrix}, \qquad f_2(x_1, x_2) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{8.2}$$

with discontinuity function $h(x_1, x_2) = x_2 - 1$. Of course, f_1 cannot be computed when $x_2 > 1$.

Thus, one may be unwilling to consider numerical methods (unlike, say, [38,28,30]) that require that f_1 extends smoothly outside $R_1 \cup \Sigma$. For this reason, below we consider *one-sided* numerical schemes in which we do not need to compute the vector field f_1 outside $R_1 \cup \Sigma$. In particular, we will study numerical procedures in which the discontinuity surface is approached from one side. These procedures compute the event or discontinuity points, and therefore they belong to the class of *event driven* methods and make sense only if on the time interval of interest there are finitely many event points.

Below, we review results from [54] where the class of general explicit Runge–Kutta (ERK) schemes has been studied and conditions under which these methods approach the discontinuity surface from one side have been derived. As illustration of this general result, we will consider a subclass of *sub-diagonal* ERK methods, that are methods for which, in the Butcher's tableau, only the entries in the first sub-diagonal are nonzero. This specific class of ERK schemes allows for recursive arguments of proof, as well as modularity in the implementation of the schemes.

8.1. General Explicit Runge–Kutta methods

Let us consider the general explicit Runge–Kutta (ERK) scheme defined by the Butcher's tableau

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & 0 & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & \cdots & b_s \end{array}.$$

To simplify the discussion we start with x_0 in the region R_1 and suppose that x_1 the numerical solution given by the ERK method with stepsize τ is in R_2; that is $(0, \tau)$ is the discontinuity interval. One step of the ERK method starting with x_0 may be written as:

$$x_1 = x_0 + \tau \sum_{i=1}^{s} b_i f_1(y_i), \tag{8.3}$$

with

$$y_1 = x_0, \qquad y_i = x_0 + \tau \sum_{j=1}^{i-1} a_{i,j} f_1(y_j), \quad i = 2, \ldots, s. \tag{8.4}$$

Of course all points x_1 and $y_2, \ldots, y_s$ may be seen as functions of the stepsize τ, that is $x_1 = x_1(\tau)$ and $y_i = y_i(\tau)$, $i = 2, \ldots, s$.

The main idea is to make sure that f_1 can be evaluated at all internal stages $y_2, \ldots, y_s$, so that the numerical solution $x_1(\tau)$ may be computed. Clearly, as long as all stage values y_j's and x_1 are in R_1, the numerical integration can proceed. Otherwise, we need to consider several different cases.

Case 1. Let us suppose (see Fig. 15 on the left):

(1.a) $h(y_i(\sigma)) \leq 0$ for $0 \leq \sigma \leq \tau$ and $i = 2, \ldots, s$;
(1.b) $h(x_1(\tau)) > 0$.

In this case, the numerical solution with stepsize τ, $x_1(\tau)$, is above Σ, while all internal stage values (for all $\sigma \in [0, \tau]$) are below Σ. This is the simplest case. Letting $x_1(\sigma) = x_0 + \sigma \sum_{i=1}^{s} b_i f_1(y_i(\sigma))$, $\sigma \in [0, \tau]$, one can use a root finding routine (say, bisection or the secant method) to compute a value η such that the scalar function $H(\sigma) = h(x_1(\sigma))$ vanishes, that is $x_1(\eta)$ on Σ.

Finally, we notice that the value $\eta \in [0, \tau]$ which gives $h(x_1(\eta)) = 0$ is unique if

$$\frac{d}{d\sigma} h(x_1(\sigma)) = h_x^T(x_1(\sigma)) x_1'(\sigma) > 0, \quad \forall \sigma \in [0, \tau], \tag{8.5}$$

and we recognize this formula as the numerical realization of (7.9).

Case 2. Let us suppose (see Fig. 15 on the right):

(2.a) $h(y_i(\sigma)) \leq 0$ for $0 \leq \sigma \leq \tau$ and $i = 2, \ldots, s-1$;
(2.b) $h(y_s(\tau)) > 0$.

Author's personal copy

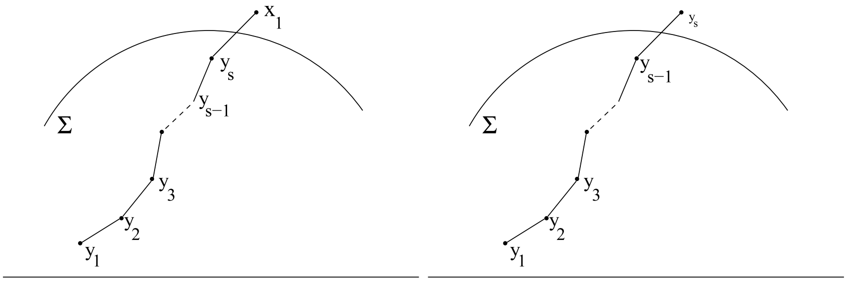


Fig. 15. General explicit RK method: case 1 and 2.

In other words, we have that the last stage value $y_s(\tau)$ is above Σ, while all the previous stage values are below Σ for all $\sigma \in [0, \tau]$. Let us now assume that

$$\frac{d}{d\sigma} h(y_s(\sigma)) = h_x^T(y_s(\sigma))y_s'(\sigma) > 0, \quad \forall\sigma \in [0, \tau],$$

which we again recognize as a monotonicity condition on the stage value function y_s (see (7.9)). Then, there exists a unique $\eta \in (0, \tau)$ such that $h(y_s(\eta)) = 0$ and further $h(y_s(\sigma)) < 0$, for all $\sigma \in [0, \eta)$. With this value of η, we compute $x_1(\eta) = x_0 + \eta \sum_{i=1}^{s} b_i f_1(y_i(\eta))$ and we need to distinguish between two subcases:

(2.c) if $h(x_1(\eta)) > 0$, we are back to the situation treated in Case 1;
(2.d) if $h(x_1(\eta)) \leq 0$, then we either continue the integration with vector field f_1 (if $h(x_1(\eta)) < 0$), or stop since we have found the sought point on Σ.

The analysis of the other possibilities proceed along similar lines.

Case 3. Suppose that:

(3.a) $h(y_i(\sigma)) \leq 0$ for $0 \leq \sigma \leq \tau$ and $i = 2, \dots, s-2$;
(3.b) $h(y_{s-1}(\tau)) > 0$.

Assume that (again a monotonicity condition for the stage function y_{s-1}):

$$\frac{d}{d\sigma} h(y_{s-1}(\sigma)) = h_x^T(y_{s-1}(\sigma))y_{s-1}'(\sigma) > 0, \quad \forall\sigma \in [0, \tau].$$

Then, there exists a unique $\bar\eta \in (0, \tau)$ such that $h(y_{s-1}(\bar\eta)) = 0$. Similarly to before, we have to distinguish between two subcases:

(3.c) if $h(y_s(\bar\eta)) \leq 0$, then we can form $x_1(\bar\eta)$; if $h(x_1(\bar\eta)) > 0$, we will assume that $h(y_s(\sigma)) \leq 0$ for $\sigma \in (0, \bar\eta)$ in order to compute $\hat\eta$ such that $h(x_1(\hat\eta))$ vanishes;
(3.d) if $h(y_s(\bar\eta)) > 0$, then we go back to case 2.

All other cases, until the situation where $y_2(\tau)$ is above Σ, and $y_1(\tau)$ (that is, x_0) is below Σ may be treated in much the same way. We stress that, as long as appropriate monotonicity assumptions hold for the stage value functions, any explicit Runge–Kutta method can approach the discontinuity surface from one side. Next, we exemplify what properties are needed of the vector field, in order to make sure that these monotonicity properties hold. We do this for the Explicit Euler scheme and for the Explicit Midpoint Rule.

Explicit Euler method. Let us consider the explicit Euler method, ERK1. In the region R_1, one step of ERK1 with stepsize τ reads

$$x_1 = x_0 + \tau f_1(x_0). \tag{8.6}$$

If x_1 is in R_1 we continue to integrate, otherwise we are above Σ. Consider the function $x_1(\sigma) = x_0 + \sigma f_1(x_0)$, $0 \leq \sigma \leq \tau$. Trivially, this is a monotone function. It is a simple observation that the function $h(x_1(\sigma))$ changes sign in $[0, \tau]$, and therefore there must be a value $\eta \in [0, \tau]$ where this function has a zero. If we want to make sure that this is the only root of $h(x_1(\sigma))$ for $\sigma \in [0, \tau]$, then we need that the straight line segment $x_1(\sigma)$ intersect Σ just once; this requires a control on the curvature of Σ with respect to the stepsize and the attractivity rate δ of (7.9). This is the content of the next result.

Theorem 8.2. *Let $x_0 \in R_1$ and close to Σ. Let $\tau > 0$ be the stepsize of the method and let $x_1(\sigma) = x_0 + \sigma f_1(x_0)$, $0 \leq \sigma \leq \tau$ (so that $x_1(\tau) = x_1$). Let τ be sufficiently small, and assume that there exist two strictly positive constants δ and ρ such that*

(S1) $h_x^T(x_0)f_1(x_0) \geq \delta$;
(S2) $[f_1(x_0)]^T h_{xx}(x_1(\sigma))f_1(x_0) \geq -\rho$, *for all* $\sigma \in [0, \tau]$;
(S3) $\delta - \rho\tau > 0$.

Then, the function $h(x_1(\sigma))$ is strictly increasing for $\sigma \in [0, \tau]$. In particular:

(i) *if* $h(x_1) \leq 0$, *then* $h(x_1(\sigma)) \leq 0$ *for all* $\sigma \in [0, \tau]$;
(ii) *if* $h(x_1) > 0$, *then there exists a unique* $\eta \in (0, \tau)$ *such that* $h(x_1(\eta)) = 0$.

Author's personal copy

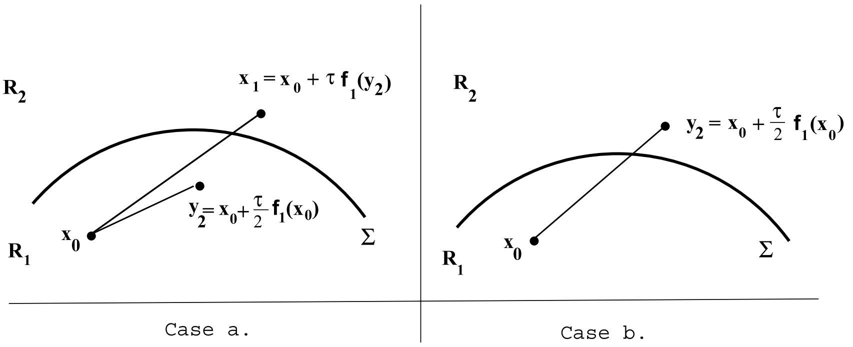


Fig. 16. Different cases for the midpoint method.

Explicit midpoint method. Next, let us consider the explicit midpoint method, ERK2. In the region R_1, one step of ERK2 with stepsize τ reads

$$x_1 = x_0 + \tau f_1(y_2), \quad \text{with } y_2 = x_0 + \frac{\tau}{2} f_1(x_0), \tag{8.7}$$

and notice that y_2 is one step of Euler method (8.6) with stepsize $\tau/2$ (see Fig. 16).

Now, suppose we have $x_0 \in R_1$, close to Σ. If $h(x_0 + \frac{\tau}{2} f_1(x_0)) \le 0$ and also $h(x_1) \le 0$, we continue integrating this system. Otherwise, we will have to distinguish between the following two cases:

(a) $h(y_2) \le 0$ but $h(x_0 + \tau f_1(y_2)) > 0$,
(b) $h(y_2) > 0$.

Case (a). Define $y_2(\sigma) = x_0 + \frac{\sigma}{2} f_1(x_0)$, for $\sigma \in [0, \tau]$, and assume that $y_2(\sigma) \in R_1$, for all $\sigma \in [0, \tau]$; this can be guaranteed under conditions much like those of Theorem 8.2, namely: $h_x^T(x_0) f_1(x_0) \ge \delta$, $[f_1(x_0)]^T h_{xx}(y_2(\sigma)) f_1(x_0) \ge -\rho$ and $\delta - \rho\tau/2 > 0$. In this case, take the function $H(\sigma) = h(x_1(\sigma))$, $\sigma \in [0, \tau]$, where $x_1(\sigma) = x_0 + \sigma f_1(y_2(\sigma))$. Observe that $H(\sigma)$ is a smooth function, taking values of opposite sign at the endpoints of the interval $[0, \tau]$. As a consequence, there must be a (first) value, call it η, where $H(\eta) = 0$. If we want that this is the unique value in $[0, \tau]$ where H vanishes, we can give sufficient conditions to guarantee that the function $H(\sigma)$ is monotone for $\sigma \in [0, \tau]$. The theorem below is such a result.

Theorem 8.3. *Consider case* (a), *and assume that $h(y_2(\sigma)) \le 0$, for all $\sigma \in [0, \tau]$. Further, assume that there are constants $\delta_2 > 0$ and $\rho_2 > 0$ and let $\tau > 0$ be sufficiently small such that the following conditions hold:*

(S1) $h_x^T(x_1(\sigma)) f_1(y_2(\sigma)) \ge \delta_2$, *for all* $\sigma \in [0, \tau]$;
(S2) $h_x^T(x_1(\sigma)) Df_1(y_2(\sigma)) f_1(x_0) \ge -\rho_2$, *for all* $\sigma \in [0, \tau]$;
(S3) $\delta_2 - \frac{\tau}{2}\rho_2 > 0$.

Then, the function $h(x_1(\sigma))$ is strictly increasing for $\sigma \in [0, \tau]$. In particular, there exists a unique $\eta \in (0, \tau)$ such that $h(x_1(\eta)) = 0$.

Case (b). Now, the stage value y_2 is already on the other side of Σ, and thus we cannot properly form x_1. So, we first seek a value $\eta \in (0, \tau)$ such that $y_2(\eta) \in \Sigma$. Then, if $x_1(\eta) = x_0 + \eta f_1(y_2(\eta))$ is above Σ, we are back to case (a) relatively to the stepsize η, and therefore the fact that there will exist a (unique) value $\eta \in [0, \eta]$ for which $h(x_1(\eta)) = 0$ can rest on Theorem 8.3. On the other hand, if $x_1(\eta)$ is below Σ, we continue integrating.

8.2. One-sided multistep methods

We conclude this review by pointing out that attempts have been made in [32,33,52] to develop also one-sided methods based on multistep schemes.

Here the goal is to choose the (variable) stepsize $\tau_k = t_{k+1} - t_k$ so to have a stable equilibrium point on the surface $h(x) = 0$. In particular, consider the Adams–Bashforth schemes of order m:

$$x_{k+1} = x_k + \tau_k \sum_{i=0}^{m-1} \gamma_i^* \nabla^i f_k, \tag{8.8}$$

so that

$$h(x_{k+1}) = h\left(x_k + \tau_k \sum_{i=0}^{m-1} \gamma_i^* \nabla^i f_k\right).$$

Author's personal copy

Now, provided that the event function is invertible, one may select

$$\tau_k = \frac{-x_k + h^{-1}(\gamma h(x_k))}{\sum_{i=0}^{m-1} \gamma_i^* \nabla^i f_k},$$

yielding the difference equation $h_{k+1} = \gamma h_k$ which has solution $h_k = h_0\gamma^k$ and converges exponentially to $h = 0$ provided $0 \le \gamma < 1$.

Of course, this requires being able to compute the inverse h^{-1}, which is often an unrealistic assumption. However, if $h(x) = b^T x + a$, then we have:

$$h(x_{k+1}) = h(x_k) + \tau_k b^T \sum_{i=0}^{m-1} \gamma_i^* \nabla^i f_k,$$

which is essentially a Taylor expansion in τ_k about x_k. Hence, one may select

$$\tau_k = \frac{(\gamma - 1)h(x_k)}{\sum_{i=0}^{m-1} \gamma_i^* [b^T \nabla^i f_k]}.$$

For more general event functions, see [32,33,52].

9. Conclusions

We gave a brief review of works on numerical integration of differential equations with discontinuous right-hand side. The topic has received attention for many years chiefly in the control and electrical engineering communities. Our hope is that this review will encourage other workers in numerical analysis to get closer to this challenging problem.

References

[1] G. Bartolini, F. Parodi, V.I.A. Utkin, T. Zolezzi, The simplex method for nonlinear sliding mode control, Mathematical Problems in Engineering 4 (1999) 461–487.

[2] E.K.P. Chong, S. Hui, S.H. Zak, An analysis of a class of neural networks for solving linear programming problems, IEEE Transactions on Automatic Control 44 (1999) 1995–2006.

[3] M.P. Glazos, S. Hui, S.H. Zak, Sliding modes in solving convex programming problems, SIAM Journal on Control and Optimization 36 (1998) 680–697.

[4] J.L. Gouze, T. Sari, A class of piecewise linear differential equations arsing in biological models, Dynamical Systems 17 (2002) 299–319.

[5] K.H. Johansson, A.E. Barabanov, K.J. Astrom, Limit cycles with chattering in relay feedback systems, IEEE Transactions on Automatic Control 247 (2002) 1414–1423.

[6] K.H. Johansson, A. Rantzer, K.J. Astrom, Fast swiyches in relay feedback systems, Automatica 35 (1999) 539–552.

[7] J. Malmborg, B. Bernhardsson, Control and simulation of hybrid systems, Communications in Nonlinear Science and Numerical Simulation 30 (1997) 337–347.

[8] E. Plathe, S. Kjøglum, Analysis and genetic proporties of gene regulatory networks with graded response functions, Physica D 201 (2005) 150–176.

[9] V.I. Utkin, Sliding Modes and Their Application in Variable Structure Systems, MIR Publisher, Moskow, 1978.

[10] V.I. Utkin, Sliding Mode in Control and Optimization, Springer, Berlin, 1992.

[11] R. Casey, H. de Jong, J.L. Gouze, Piecewise-linear models of genetics regulatory networks: equilibria and their stability, Journal of Mathematical Biology 52 (2006) 27–56.

[12] H. de Jong, J.L. Gouze, C. Hernandez, M. Page, T. Sari, J. Geiselmann, Qualitative simulation of genetic regulatory networks using piecewise-linear models, Bulletin of Mathematical Biology 66 (2004) 301–340.

[13] W.P.M.H. Heemels, J.M. Schumacher, S. Weiland, Linear complementarity systems, SIAM Journal on Applied Mathematics 60 (4) (2000) 1234–1269.

[14] M. di Bernardo, P. Kowalczyk, A. Nordmark, Bifurcations of dynamical systems with sliding: derivation of normal-form mappings, Physica D 170 (2002) 175–205.

[15] Y.A. Kuznetsov, S. Rinaldi, A. Gragnani, One-parameter bifurcations in planar filippov systems, International Journal of Bifurcation and Chaos 13 (2003) 2157–2188.

[16] P. Kowalczyk, M. di Bernardo, Two-parameter degenerate sliding bifurcations in filippov systems, Physica D 204 (2005) 204–229.

[17] R.I. Leine, H. Nijmeijer, Dynamics and Bifurcations in Non-Smooth Mechanical Systems, in: Lecture Notes in Applied and Computational Mechanics, vol. 18, Springer-Verlag, Berlin, 2004.

[18] R.I. Leine, D.H. van Campen, B.L. van de Vrande, Bifurcations in nonlinear discontinuous systems, Nonlinear Dynamics 23 (2000) 105–164.

[19] J.P. Aubin, A. Cellina, Differential Inclusions, Springer-Verlag, Berlin, 1984.

[20] A.F. Filippov, Differential Equations with Discontinuous Right–Hand Sides, in: Mathematics and Its Applications, Kluwer Academic, Dordrecht, 1988.

[21] V. Acary, B. Brogliato, Numerical Methods for Nonsmooth Dynamical Systems. Applications in Mechanics and Electronics, in: Lecture Notes in Applied and Computational Mechanics, Springer-Verlag, Berlin, 2008.

[22] M. di Bernardo, C.J. Budd, A.R. Champneys, P. Kowalczyk, Piecewise-smooth Dynamical Systems. Theory and Applications, in: Applied Mathematical Sciences, vol. 163, Springer-Verlag, Berlin, 2008.

[23] J. Llibre, P.R. da Silva, M.A. Teixeira, Regularization of discontinuous vector fields via singular perturbation, Journal of Dynamics and Differential Equations 19 (2009) 309–331.

[24] R.I. Leine, Bifurcations in discontinuous mechanical systems of filippov's type, Ph.D. Thesis, Techn. Univ. Eindhoven, The Netherlands, 2000.

[25] C.W. Gear, O. Østerby, Solving ordinary differential equations with discontinuities, ACM Transactions on Mathematical Software 10 (1984) 23–24.

[26] A. Hindmarsh, GEAR: Ordinary differential solver, Tech. Rep. UCID-30001, Revision 3, Lawrence Livermore National Laboratories, Livermore California, 1974.

[27] D.E. Stewart, Rigid-body dynamics with friction and impact, SIAM Review 42 (2000) 3–39.

[28] R. Mannshardt, One-step methods of any order for ordinary differential equations with discontinuous right-hand sides, Numerische Mathematik 31 (1978) 131–152.

Author's personal copy

[29] T. Holzhueter, Simulation of relay control systems using MATLAB/SIMULINK, Control Engineering Practice 6 (1998) 1089–1096.
[30] L.F. Shampine, S. Thompson, Event location for ordinary differential equations, Computer and Mathematics with Applications 39 (2000) 43–54.
[31] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, second revised ed., Springer-Verlag, Berlin, 1987.
[32] J.M. Esposito, V. Kuman, An asynchronous integration and event detection algorithm for simulating Multi-Agent hybrid systems, ACM Transactions on Modeling and Computer Simulation 14 (4) (2004) 363–388.
[33] J.M. Esposito, V. Kuman, A state event detection algorithm for numerically simulating hybrid systems with model singularities, ACM Transactions on Modeling and Computer Simulation 17 (1) (2007) 1–22.
[34] J.D. Lambert, Numerical Methods for Ordinary Differential Systems: The Initial Value Problem, John Wiley, London, 1991.
[35] M. Calvo, J.I. Montijano, L. Randez, On the solution of discontinuous IVPs by adaptive Runge–Kutta codes, Numerical Algorithms 33 (2003) 163–182.
[36] M. Calvo, J.I. Montijano, L. Randez, The numerical solution of discontinuous IVPs by Runge–Kutta codes: a review, Boletín de la Sociedad Espanõla Mathemática Aplicada 44 (2008) 33–53.
[37] W.H. Enright, K.R. Jackson, S.P. Nørsett, P.G. Thomsen, Effective solution of discontinuous IVPs using Runge–Kutta formula pair with interpolants, Applied Mathematics and Computation 27 (1988) 313–335.
[38] E. Eich-Soellner, C. Fuhrer, Numerical Methods in Multibody Dynamics, B.G. Teubner, Stuttgart, Germany, 1998.
[39] N. Guglielmi, E. Hairer, Computing breaking points in implicit delay differential equations, Advances in Computational Mathematics 29 (2008) 229–247.
[40] G. Grabner, R. Kittinger, A. Kecskemethy, An integration Runge–Kutta and polynomial root finding method for reliable event-driven multibody simulation, in: Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, IFAC, Seville 3–5, 2003.
[41] E. Hairer, C. Lubich, G. Wanner, Solving Ordinary Differential Equations II: Stiff Problems, second revised ed., Springer-Verlag, Berlin, 2010.
[42] A. Bellen, M. Zennaro, Numerical Methods for Delay Differential Equations, Clarendon Press, Oxford, 2003.
[43] M. Berardi, L. Lopez, On the continuous extension of Adams-Bashforth methods and the event location in discontinuous ODEs, Applied Mathematics Letters 25 (6) (2012) 995–999.
[44] M.B. Carver, Efficient implementation over discontinuities in ordinary differential equation simulations, Mathematics and Computers in Simulation 20 (1978) 190–196.
[45] L. Fridman, Chattering in sliding mode systems and singular perturbation, in: Proc. Int. Symp. on Nonlinear Control Systems, 1995, pp. 725–730.
[46] A.B. Nordmark, P.T. Piiroinen, Simulation and stability analysis of impacting systems with complete chattering, Nonlinearity 58 (1) (2009) 85–106.
[47] P.T. Piiroinen, Y.A. Kuznetsov, An event-driven method to simulate Filippov systems with accurate computing of sliding motions, ACM Transactions on Mathematical Software 34 (13) (2008) 1–24.
[48] L. Dieci, L. Lopez, Sliding motion in Filippov differential systems: theoretical results and a computational approach, SIAM Journal on Numerical Analysis 47 (2009) 2023–2051.
[49] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations, Springer-Verlag, Berlin, 2006.
[50] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Society for Industrial and Applied Mathematics (1998).
[51] L. Dieci, L. Lopez, Sliding motion on discontinuity surfaces of high co-dimension. A construction for selecting a Filippov vector field, Numerische Mathematik 117 (2011) 779–811.
[52] J.M. Esposito, V. Kuman, G.J. Pappas, Accurate event detection for simulating hybrid systems, in: M.D. Di Benedetto, A. Sangiovanni-Vincitelli (Eds.), HSCC 2001, in: LNCS, vol. 2034, Springer-Verlag, Berlin, Heidelberg, 2001, pp. 204–217.
[53] M. Najaf, A. Azil, R. Nikoukhah, Implementation of continuous-time dynamics in scicos, in: Vlatka Hlupic Alexander Verbraeck Ed., Proceedings 15th European Simulation Symposium, SCS European Council / SCS Europe BVBA, 2003.
[54] L. Dieci, L. Lopez, Numerical solution of discontinuous differential systems: approaching the discontinuity from one side, Applied Numerical Mathematics (2011).