DISODE45, A Matlab Runge-Kutta solver for Piecewise Smooth IVPs of Filippov type

M. Calvo [1], J. I. Montijano [2] and L. Rández[3] *

Departamento Matemática Aplicada
Pza. San Francisco s/n
Universidad de Zaragoza.
50009-Zaragoza, Spain.

[1] *email: calvo@unizar.es*
[2] email: monti@unizar.es
[3] *email: randez@unizar.es*

Abstract

In this paper an adaptive Runge-Kutta code, based on the DOPRI5(4) pair for solving Initial Value Problems for differential systems with Piecewise Smooth solutions (PWS) is presented and the algorithms used in the code are described. The code automatically detects and locates accurately the switching points of the PWS, restarting the integration after each discontinuity. Further, in the case of Filippov systems, algorithms to handle properly sliding mode regimes in an automatic way are included. The code requires the user to provide a description of the IVP and the functions defining the hypersurfaces where the switching points are located, and it returns the discrete approximated solution together with the switching points. Several numerical experiments are presented to illustrate the reliability and efficiency of the code.

Keywords: Discontinuous Initial Value Problems. Adaptive Runge–Kutta methods. Detection of discontinuities. Filippov discontinuous systems;

AMS classification: 65L06; 65L05.

*This work was supported by project DGI MTM2013–2562

1 Introduction

We consider IVPs for differential systems with PieceWise Smooth solutions (PWS) that are defined by

$$y' = f(t, y), \quad y(t_0) = y_0 \in \mathbb{R}^m, \quad t \in [t_0, t_f], \tag{1}$$

where the vector field $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ contains bounded discontinuities either in f itself or in some its derivatives on a smooth event hypersurface defined by $g(t, y) = 0$ (switching surface), so that $f(t, y)$ can be locally written in the form:

$$f(t, y) = \begin{cases} f_-(t, y) & \text{for} \quad g(t, y) < 0, \\ f_+(t, y) & \text{for} \quad g(t, y) > 0, \end{cases} \tag{2}$$

with the sufficiently smooth functions f_- and f_+ satisfying a local Lipschitz condition with respect to y in a tubular domain around the solution of (1) in their definition domain. These PWS systems are also called switching systems, and some authors consider them as hybrid systems (see for example, [**?**, **?**]).

PWS systems appear in electrical circuits, mechanical systems with friction or impacts [**?**, **?**], modelling of traffic [**?**, **?**], vehicle control [**?**, **?**], neural networks [**?**] and biology [**?**].

If f is not smooth, standard codes for the numerical solution of IVPs perform inefficiently and may provide a very poor approximation, as has been shown in [**?**, **?**, **?**]. To overcome those difficulties, different algorithms that make the integration more reliable and accurate have been proposed. Thus, in [**?**, **?**, **?**] the case when there is no additional information about the location of the discontinuities is considered, whereas algorithms for problems with a known smooth hypersurface of discontinuity have been proposed in [**?**, **?**, **?**, **?**]. In all these cases, it is assumed that the solution crosses the hypersurface of discontinuity and a transversality condition is satisfied. However, there are dynamical systems for which the solution reaches the discontinuity surface and stays on it for some time and then goes out again. This happens in the so called sliding mode regime for Filippov systems [**?**] in which a switching surface attracts, in finite time, nearby dynamics, so that trajectories become constrained to remain on this surface. Such systems appear, for example, in mechanical and electromechanical systems under certain control techniques, ecology and population models [**?**, **?**].

Numerical methods to deal with non-smooth Filippov systems have been studied in [**?**]. Piiroinen and Kuznetsov propose in [**?**] a code to solve these systems based on the event driven facility of `odepack` in Matlab [**?**, **?**]. In this code, the user must provide the different functions that define the vector field of the differential system (f_+ and f_- in (2)). Then, if, for example, $g(t_0, y_0) > 0$, the code advances the numerical integration of the system $y' = f_+(t, y)$ until a switching point (t_d, y_d) such that $g(t_d, y_d) = 0$ is reached.

The integration continues from this point by solving $y' = f_-(t, y)$ if the discontinuity is transversal, or the associated Filippov system otherwise. The code requires that the two functions f_+ and f_- are defined around the switching point, even if they are only used in their respective regions.

However, there are problems for which the functions f_+ and f_- of (2) are not defined at both sides of the event hypersurface. In such situations, algorithms that are able to reach the discontinuity point from one side of the switching surface are required. In [**?**] a method based on extrapolation techniques is given. More recently, Dieci and Lopez [**?**, **?**, **?**] propose the so called one-side methods to deal with this situation.

The aim of this paper is to present a code, based on the pair of explicit RK formulas of Dormand and Prince [**?**] of orders 4 and 5, for the numerical solution of PWS systems. The main features of the code are:

1. Automatic integration of co-dimension 1 Filippov systems.
2. Integration of problems with as many switching surfaces as required.
3. Automatic management of vector fields for which their defining functions are defined only at one side of the switching surface.
4. Minimum information of the differential system required to the user.
5. Reliable integration.

The paper is organized as follows: In §2 the notations and basic assumptions that will be used in the rest of the paper are presented. Also, the classes of systems that the code can integrate are given. In §3 algorithms to detect the presence of a discontinuity and its accurate location are proposed. In §4, the algorithms used to integrate the system at a sliding region are described. Finally, in §5 some numerical experiments to test the behaviour of the algorithms implemented in `DISODE45` are presented.

2 Notations and basic assumptions

A point $(t_d, y_d) \in [t_0, t_f] \times \mathbb{R}^m$ is a switching point of the solution $y = y(t)$ of (1)–(2) if $y_d = y(t_d)$ and $g(t_d, y_d) = 0$. We will denote by $\nabla g(t, y)$ the gradient vector of $g(t, y)$ with respect to y. Moreover, for each vector $u \in \mathbb{R}^m$ we define the scalar operator

$$\mathrm{D}g(u) \equiv \partial_t g + \nabla g \cdot u$$

The class of problems under consideration has the following features:

- The system can have several switching surfaces, although for simplicity we consider here only one event hypersurface $g(t, y) = 0$. We also assume that the function g splits a neighbourhood of a switching point into two disjoint regions in such a way that at one region $g(t, y) > 0$ and at the other region $g(t, y) < 0$.

- We may have transversal discontinuity points (t_d, y_d) that satisfy the so called transversality condition (see for example, [**?**], eq. (1.6)) at (t_d, y_d)

$$\mathrm{D}g(f_+)\ \mathrm{D}g(f_-) > 0. \tag{3}$$

  This implies that at time $t = t_d$ the solution $y(t)$ is not tangent to the switching surface and crosses it. On the other hand the switching points can also not be transversal in such a way that the solution reaches the switching surface and stays on it for some time, and therefore the solution will contain an infinite number of switching points. The code admits several switching surfaces, but it admits only sliding motion on co-dimension 1 discontinuity surfaces (for theoretical analysis of higher co-dimension surfaces see for example, [**?**, **?**]). When a high-codimension motion is found, the code stops the integration and an error message is returned to the user.

  At a sliding region it is assumed that the integration advances under the Filippov vector field $f_{\mathcal{F}}$ defined by

$$f_{\mathcal{F}}(t, y) = \alpha_- f_-(t, y) + \alpha_+ f_+(t, y) \text{ for all } (t, y) \text{ with } g(t, y) = 0, \tag{4}$$

  with

$$\alpha_- = \alpha_-(t, y) = \frac{\mathrm{D}g(f_+)(t, y)}{\nabla g(t, y) \cdot (f_+(t, y) - f_-(t, y))}, \quad \alpha_+ = 1 - \alpha_-. \tag{5}$$

- The function f_+ (respectively f_-) may not be defined at the region $g(t, y) < 0$ (respectively $g(t, y) > 0$). This implies that the time advancing scheme must ensure that these functions are evaluated only in their definition domains.

Furthermore, the code allows us the integration of IVPs with impulses in which at certain times t_d that satisfy $g(t_d, y(t_d)) = 0$, the states y_d are externally modified from y_d to y_d^*. In a similar way, the code admits problems with a switching vector field. There are situations where the ODE changes to a different one when the solution $y = y(t)$ reaches a point (t_d, y_d) satisfying $g(t_d, y(t_d)) = 0$ (see for example, [**?**, pp. 38], temperature control system). Such problems can be transformed to IVPs with impulses.

To advance the numerical solution of (1)–(2), from (t_n, y_n) to (t_{n+1}, y_{n+1}), $t_{n+1} = t_n + h_n$, we consider the embedded pair of explicit formulas of orders 5 and 4 with $s = 7$ stages proposed by Dormand and Prince [**?**]. Note that this pair computes the values of the vector field f at the stage values $(t_{n,i} = t_n + c_i h_n, Y_{n,i})$, $i = 1, \ldots, 7$ with $c_1 = 0 < c_2 < \ldots < c_6 = c_7 = 1$.

Further, `DISODE45` uses the continuous extension $y_{n+1}(t_n + \theta h_n)$ of the numerical solution between the points t_n and $t_n + h_n$ of uniform order 5 proposed in [**?**] that requires two additional stages, as well as the natural fourth order continuous extension $\hat{y}_{n+1}(t_n + \theta h_n)$ [**?**].

3 Detecting and locating a discontinuity

3.1 Detecting a discontinuity

Since two points (t_a, y_a) and (t_b, y_b) are (locally) on different sides of the switching surface g if $g(t_a, y_a)\ g(t_b, y_b) < 0$, once a step from (t_n, y_n) to $(t_{n+1} = t_n + h_n, y_{n+1})$ has been completed, we can check whether or not the numerical solution has crossed the event surface by examining the sign of the product $g(t_n, y_n)\ g(t_{n+1}, y_{n+1})$. In such a case we can assume that a discontinuity is located in the interval $[t_n, t_{n+1}]$, which will be true for sufficiently small h_n. However, a positive sign of the product does not allow us to conclude that there is no discontinuity point within the step. It may happen that (t_n, y_n) and (t_{n+1}, y_{n+1}) are both at the same side of the surface, but there is some intermediate point $(t_n + \theta h_n, y(t_n + \theta h_n))$, $0 < \theta < 1$ on the other side of the surface. Such situations have motivated the search for more reliable algorithms that can detect the discontinuity in difficult situations (see for example, [**?**]).

The detection of a possible switching point in the numerical solution from (t_n, y_n) provided by DOPRI5(4) with the predicted step size h_n can be improved by noting that this pair requires the evaluation of the vector field at the intermediate stage points $(t_{n,i}, Y_{n,i})$, $i = 1, \ldots, 7$ with $Y_{n,i} \simeq y(t_n + c_i h_n)$. If $g(t_n, y_n)\ g(t_{n,i}, Y_{n,i}) < 0$ for some i-th stage in the current step, the method is using values of the vector field on different sides of the event surface.

Further, `DISODE45` includes a variable named `eventcontrol` that lets the user reinforce the reliability of the algorithm in detecting possible discontinuities. The larger the variable `eventcontrol` is, the more reliable the detection algorithm is. By default (`eventcontrol`$= 0$), the code only checks at the end points t_n and t_{n+1}. If the user sets `eventcontrol`$= 1$, the code checks at the stage points $t_{n,i}$. For `eventcontrol`$= k > 1$ the fourth-order continuous extension $\hat{y}_{n+1}(t)$ is used and the signs of the products $g(t_n, y_n)\ g(t_n + \theta_i h_n, \hat{y}_{n+}(t_n + \theta_i h_n))$ are checked with $\theta_i = i/(1 + 6k)$ for $i = 1, \ldots, 1 + 6k$.

3.2 Locating the discontinuity

Suppose that at the starting point (t_n, y_n) of the step $t_n \to t_{n+1} = t_n + h_n$ we have $g(t_n, y_n) > 0$ and for some i $(1 \le i \le 5)$ we detect a stage $Y_{n,i+1}$, where $g(t_n + c_{i+1} h_n, Y_{n,i+1}) < 0$. We expect to find a switching point in the interval $[t_n + c_i h_n, t_n + c_{i+1} h_n]$ and we reduce the size of the step by choosing $h_n^* < h_n$ and advance the solution from t_n to $t_{n+1} = t_n + h^*$ such that the discontinuity, t_{dis}, belongs to the interval $[t_n + h^*, t_n + \beta h^*]$ with $\beta > 1$. In this situation, we use the fifth-order continuous extension, which provides a reliable extrapolated approximation to the solution $y(t)$ on the interval $[t_n + h^*, t_n + 1.3h^*]$ (see for example, [**?**] or [**?**]). For this reason β can not be greater than 1.3. We have taken $\beta = 1.3$. Then, the conditions

$g(t_n, y_n)\ g(t_{n+1}, y_{n+1}) > 0$ and $g(t_n, y_n)\ g(t_n + 1.3h^*, y_{n+1}(t_n + 1.3h^*)) < 0$ tell us that there exist an intermediate point t_{dis} for which $g(t_{dis}, y_{n+1}(t_{dis})) = 0$.

The discontinuity point can be obtained by solving on $\theta \in [1, 1.3]$ the scalar nonlinear equation $g(t_n + \theta h^*, y_{n+1}(t_n + \theta h^*)) = 0$. To do that, we have chosen an iterative scheme which is a combination of the secant and bisection methods. Let us note that the function $y_{n+1}(t_n + \theta h^*)$ is a polynomial in θ of degree 8.

The method we are using here to locate the discontinuity is in fact a one-sided method. Other methods of this type have been reported by Dieci and Lopez [**?**, **?**, **?**].

After some calculations, we see that t_{dis} is an approximation to the exact switching time t_d of order 5 whenever $\frac{\mathrm{d}}{\mathrm{d}t} g(t, y(t))|_{t=t_d} = \mathrm{D}g(f)(t_d, y(t_d)) \neq 0$. This derivative represents the rate of change of the event function along the flow field, and Carver in [**?**] noticed that it is a critical quantity in event detection. This condition means that the solution $y(t)$ does not approximate the discontinuity $(t_d, y(t_d))$ in a direction tangent to the surface $g(t, y)$. Moreover, the error in the discontinuity point will be large if the direction is close to the tangent.

3.3 Determining the type of discontinuity

After numerically obtaining a discontinuity point (t_{dis}, y_{dis}) of the solution, we need to know if it satisfies the transversality condition (3) or corresponds to a sliding point in order to determine how the integration will proceed. The numerical integration will continue at the other side of the switching surface in the case of a transversal point, otherwise it will stay on the switching surface until an exit point is reached. There are two difficulties: First we are assuming that the user provides only the vector field f to `DISODE45` and the functions f_+ and f_- are not given explicitly. Therefore, to check the transversality condition we must compute f_+ and f_- through f. Second, even if we had expressions for both functions, we should evaluate them at the discontinuity point (t_{dis}, y_{dis}). But since this point has been obtained numerically, it is highly unlikely that $g(t_{dis}, y_{dis}) = 0$ exactly and therefore this point will be in practice at one side of the switching surface, and one of the two functions can not be defined at this numerical discontinuity point.

To overcome these problems we propose the following algorithm: Assuming that $g(t_{dis}, y_{dis}) > 0$ (the other case is similar)

1. If $g(t_{dis}, y_{dis})\ g(t_n, y_n) > 0$, we compute with the continuous extension another point (t^+_{dis}, y^*) with $t^+_{dis} = t_{dis} + \varepsilon$ and $y^* = y_{n+1}(t^+_{dis})$ such that $g(t^+_{dis}, y^*)\ g(t_n, y_n) < 0$, that is, t_{dis} is on the other side of the switching surface, with $\varepsilon > 0$ as small as possible (in the code we have

used $\varepsilon = 10\texttt{eps}$, where ε is the round-off unit). Then we define

$$\begin{array}{l} y^+ = y_{dis},\ f_+ = f(t_{dis}, y_{dis}),\ y^- = y^*,\ f_- = f(t_{dis}^+, y^*) \quad \text{if } g(t_n, y_n) > 0, \\ y^- = y_{dis},\ f_- = f(t_{dis}, y_{dis}),\ y^+ = y^*,\ f_+ = f(t_{dis}^+, y^*) \quad \text{if } g(t_n, y_n) < 0. \end{array}$$

2. If $g(t_{dis}, y_{dis}) \cdot g(t_n, y_n) < 0$, we compute with the continuous extension another point (t_{dis}^-, y^*) with $t_{dis}^- = t_{dis} - \varepsilon$ and $y^* = y_{n+1}(t_{dis}^-)$ such that $g(t_{dis}^-, y^*)\ g(t_n, y_n) > 0$. Then we define

$$\begin{array}{l} y^+ = y_{dis},\ f_+ = f(t_{dis}, y_{dis}),\ y^- = y^*,\ f_- = f(t_{dis}^-, y^*) \quad \text{if } g(t_n, y_n) > 0, \\ y^- = y_{dis},\ f_- = f(t_{dis}, y_{dis}),\ y^+ = y^*,\ f_+ = f(t_{dis}^-, y^*) \quad \text{if } g(t_n, y_n) < 0. \end{array}$$

3. If $\mathrm{D}g(f_+)\ \mathrm{D}g(f_-) > 0$, the discontinuity is transversal, and the integration is restarted from the point (t_{dis}, y_{dis}) if $g(t_{dis}, y_{dis})\ g(t_n, y_n) < 0$ or from the point (t_{dis}^+, y^*) otherwise.

4. If $\mathrm{D}g(f_+)\ \mathrm{D}g(f_-) < 0$, the discontinuity is not transversal and we enter into a sliding region at the point (t_{dis}, y_{dis}).

4 Integrating along a sliding region

We start from a point (t_{dis}, y_{dis}) such that $\mathrm{D}g(f_+)(t_{dis}, y_{dis})\ \mathrm{D}g(f_-)(t_{dis}, y_{dis}) < 0$. Now we must integrate the differential system

$$y' = f_{\mathcal{F}}(t, y), \quad y(t_{dis}) = y_{dis},$$

with $f_{\mathcal{F}}(t, y)$ given by (4)–(5), until we reach a point (t, y) such that

$$\mathrm{D}g(f_+)(t, y)\ \mathrm{D}g(f_-)(t, y) = 0. \tag{6}$$

Such a point, if it exists, will be called an exit point.

The Runge-Kutta method we use requires the evaluation of the intermediate stages $Y_{n,i}$ associated with the vector field $f_{\mathcal{F}}$. However, some of the intermediate stages may not belong to the manifold, that is $g(t_n + c_i h_n, Y_{n,i}) \neq 0$, and the vector field $f_{\mathcal{F}}$ may not be defined at these internal stages. To overcome this difficulty, we force the intermediate stages $Y_{n,i}$ and the final approximation y_{n+1} to stay on the manifold by using orthogonal projection [?] and following a technique similar to the one used with DAEs. Starting from a point (t_n, y_n) belonging to the manifold, we advance a step by

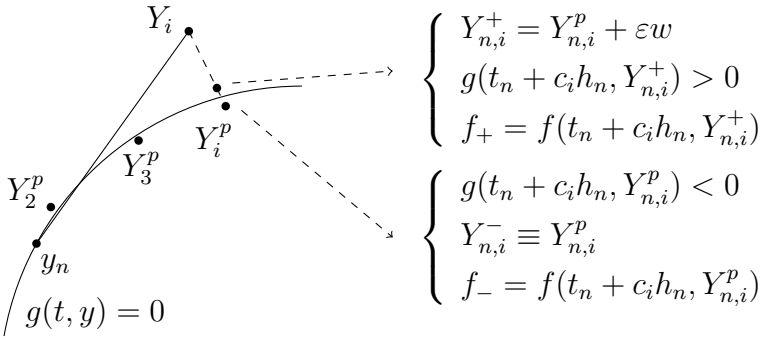
$$\begin{array}{l} Y_{n,1} = y_n \in \mathcal{M}, \quad Y_{n,1}^p = Y_{n,1} \\ Y_{n,i} = y_n + h_n \sum_{j=1}^{i-1} a_{ij} f_F(Y_{n,j}^p), \quad i = 2, \ldots, 6 \\ Y_{n,i}^p = Y_{n,i} + \lambda_i w, \text{ with } \lambda_i \text{ such that } g(t_n + c_i h_n, Y_{n,i}^p) = 0 \\ \tilde{y}_{n+1} = y_n + h_n \sum_{i=1}^{6} b_i f_F(Y_{n,i}^p) \\ y_{n+1} = \tilde{y}_{n+1} + \lambda_7 w, \text{ with } \lambda_7 \text{ such that } g(t_{n+1}, y_{n+1}) = 0 \end{array}$$

where $w = \nabla g(t_n, y_n)/\|\nabla g(t_n, y_n)\|$. Note that this projection does not change the order of accuracy of the method (see [**?**]).

To evaluate the functions $f_+(t, y)$ and $f_-(t, y)$ in the appropriate definition domains, we use a similar approach to the one used for the classification of the discontinuity:

- Once we have computed an intermediate stage $(t_{n,i}, Y_{n,i})$, projecting onto the manifold, we have $Y^p_{n,i}$ such that $g(t_n + c_i h_n, Y^p_{n,i}) \simeq 0$ but not exactly zero. Let us suppose that $g(t_n + c_i h_n, Y^p_{n,i}) < 0$ (the other case is identical). We take $Y^-_{n,i} = Y^p_{n,i}$ and $f_- = f(t_n + c_i h_n, Y^p_{n,i})$. Next, we compute another value, at the other side of the manifold, using $Y^+_{n,i} = Y^p_{n,i} + \varepsilon w$ with ε as small as possible and the appropriate sign so that $g(t_n + c_i h_n, Y^p_{n,i} + \varepsilon w) > 0$. Then, we take $Y^+_{n,i} = Y^p_{n,i} + \varepsilon w$ and $f_+ = f(t_n + c_i h_n, Y^p_{n,i} + \varepsilon w)$.

- We evaluate the Filippov vector field at $Y^p_{n,i}$ using $f_{\mathcal{F}}(t_n + c_i h_n, Y^p_{n,i}) = (1 - \alpha) f_+ + \alpha f_-$ and we can proceed to the next stage $Y_{n,i+1}$.



4.1 Exit from a sliding region

An exit point is reached when condition (6) is satisfied. The detection and location of exit points follow a procedure similar to the one for discontinuity points.

- We detect a possible exit point when, at a step $(t_n, y_n) \to (t_n + h_n, y_{n+1})$, for some $i \leq 6$,

$$\mathrm{D}g(f_+)(t_n + c_i h_n, Y^p_{n,i})\ \mathrm{D}g(f_-)(t_n + c_i h_n, Y^p_{n,i}) > 0,$$
$$\mathrm{D}g(f_+)(t_n, y_n)\ \mathrm{D}g(f_-)(t_n, y_n) < 0.$$

  Then we compute a rough approximation t_{exit} to the exiting point.

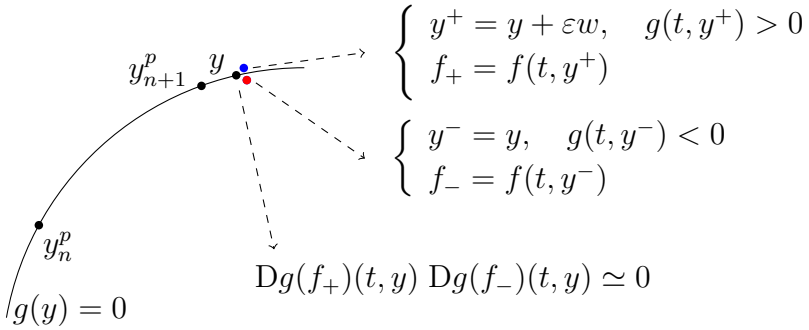
- We advance from (t_n, y_n) a step of size $h^* < h_n$ such that $t_{exit} - (t_n + h^*) < \gamma h^*$. The factor $\gamma \leq 0.3$ guarantees that the extrapolation of the continuous extension gives an approximation with small error. We have chosen the value $\gamma = 0.15$ in the code.

- We compute, by a combination of the secant and bisection methods, the value of θ such that

$$\mathrm{D}g(f_+)(t_n + \theta h^*, y^+_{n+\theta})\ \mathrm{D}g(f_-)(t_n + \theta h^*, y^-_{n+\theta}) = 0$$

  where $y^+_{n+\theta}$, $y^-_{n+\theta}$ are two points very close to the projection $y^p_{n+\theta}$ of the continuous approximation $y_{n+1}(t_n + \theta h^*)$ onto the switching surface.

- We restart the integration (to a non sliding region) from $y^-_{n+\theta} = y^p_{n+\theta}$ if $\mathrm{D}g(w)(t_n + \theta h^*, y^p_{n+\theta})\mathrm{D}g(f)(t_n + \theta h^*, y^p_{n+\theta}) < 0$. This condition implies that at this point, the vector field $f(t, y)$ takes the dynamics out of the manifold. We restart from $y^+_{n+\theta}$ otherwise.



5 Numerical examples

To illustrate the performance of the proposed algorithms, we present the numerical results obtained when four initial value problems with discontinuities, covering different situations, are integrated with `DISODE45`.

Problem 1. This is based on a 2-dimensional problem used in [**?**], with some modifications so that there is a unique switching surface $g(y_1, y_2) = 0$ defined by a non linear function and the vector field is defined by two functions $f_+(y_1, y_2)$ which is not defined for $g(y_1, y_2) < 0$ and $f_-(y_1, y_2)$ which is not defined for $g(y_1, y_2) > 0$.

$$\begin{aligned} &y_1' = y_2 - \sin(2y_1), \\ &y_2' = 2\cos(2y_1)(y_2 - \sin(2y_1)) - y_1 + u(y_1, y_2), \\ &y_1(0) = -0.75, y_2(0) = -1 - \sin(1.5), \quad t \in [0, 30], \end{aligned}$$

with the scalar function, u, given by

$$u(y_1, y_2) = \frac{-\text{sign}(g(y_1, y_2))}{1 + |g(y_1, y_2)|^{3/2}}, \qquad g(y_1, y_2) = y_2 - 0.2 - \sin(2y_1).$$

The exact solution has a first discontinuity point at $t = 0.72319254, y = (-1.08023276, 0.2 + \sin(-2 * 1.08023276))$ that satisfies the transversality condition. Then the solution continues very close to the switching surface (see Figure 1, bottom plot) and at $t = 1.49648739, y = (-0.917378, 0.2 + \sin(-2 * 0.917378))$ the solution enters a sliding region and leaves it at $t = 11.083377, y = (1.0, 0.2 + \sin(2))$. From here the solution follows a periodic cycle, entering and going out the sliding region two times more. The phase plot of the solution in the (y_1, y_2)-plane is depicted in Figure 1, where the numerical solution is the continuous line and the switching surface is the dashed line.

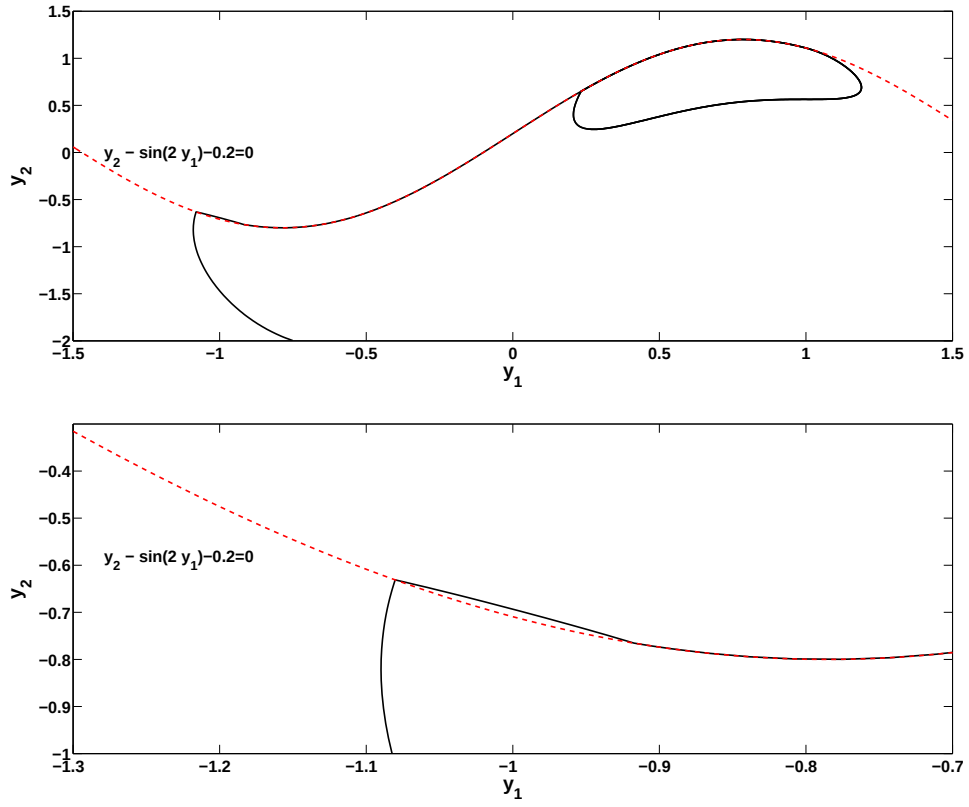


Figure 1: Phase diagram for Problem 1.

Problem 2. This is a simplified model of structural pounding used in the study of the effects of earthquakes [**?**, **?**].

$$2y'' = -4.1\, y' - 210.125\, y - u(y, y') - f(t), \qquad t \in [0, 3]$$

with $f(t) = 2\sin(14\,t)$ and

$$u(y,y') = \begin{cases} 0 & \text{if } y < 0.005 \\ c\cdot(y-0.005)^{3/2} + & \\ \qquad 1.98\sqrt{2c\sqrt{y-0.005}}\,y' & \text{if } y > 0.005 \text{ and } y' > 0 \\ c\cdot(y-0.005)^{3/2} & \text{if } y > 0.005 \text{ and } y' < 0 \\ c = 2.47\times 10^6 & \end{cases}$$

This problem has two switching surfaces $g_1(t,y,y') = y - 0.005$ and $g_2(t,y,y') = y'$. Moreover, due to the powers $1/2$ and $3/2$, the function defining the vector field at the region $g_1(t,y,y') > 0$ is not defined when $g_1(t,y,y') < 0$.

For $t \in [0,3]$ the solution crosses the surface g_1 twelve times and the surface g_2 six times. In all the cases the discontinuity is transversal.

The phase diagram for this problem (y versus y') is depicted in Figure 2 (the dashed lines correspond to the switching surfaces).

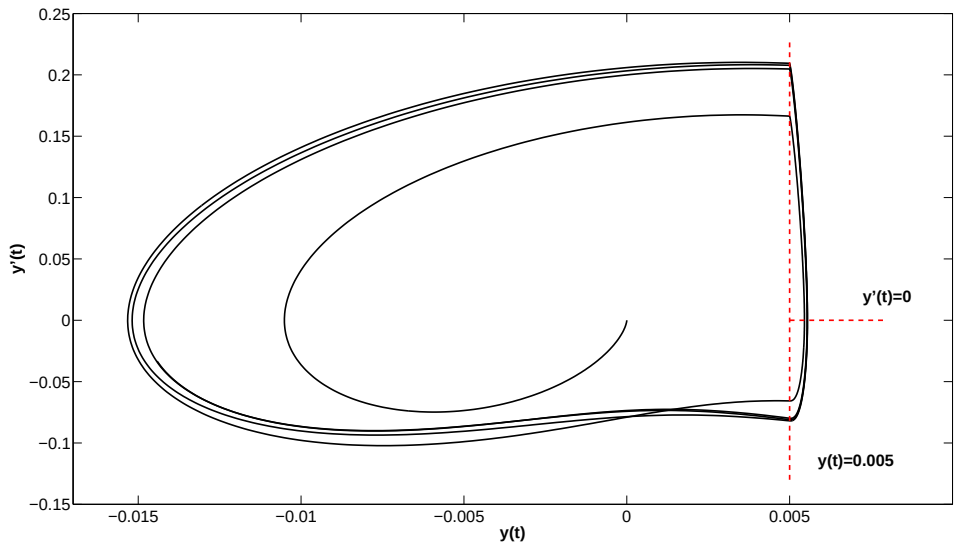


Figure 2: Phase diagram for Problem 2

Problem 3. A relay feedback system (see [**?**, pp. 29], [**?**])

$$\begin{aligned} y_1' &= -(2\zeta\omega + 1)y_1 + y_2 - \operatorname{sign}(y_1), \\ y_2' &= -(2\zeta\omega + \omega^2)y_1 + y_3 + 2\operatorname{sign}(y_1), \\ y_3' &= -\omega^2 y_1 - \operatorname{sign}(y_1), \\ y_1(0) &= 0, y_2(0) = 0.2, y_3(0) = 0.06, \quad t \in [0, 4\pi]. \end{aligned}$$

with $\omega = 25$ and $\zeta = 0.05$.

This problem has a unique switching surface $g(y_1, y_2, y_3) = y_1$. When the time t increases from 0 to 4π the exact solution enters and exits a sliding region fourteen times. The phase diagram for this problem (y_1, y_2, y_3) is depicted in Figure 3. At some points the solution stays at the sliding area

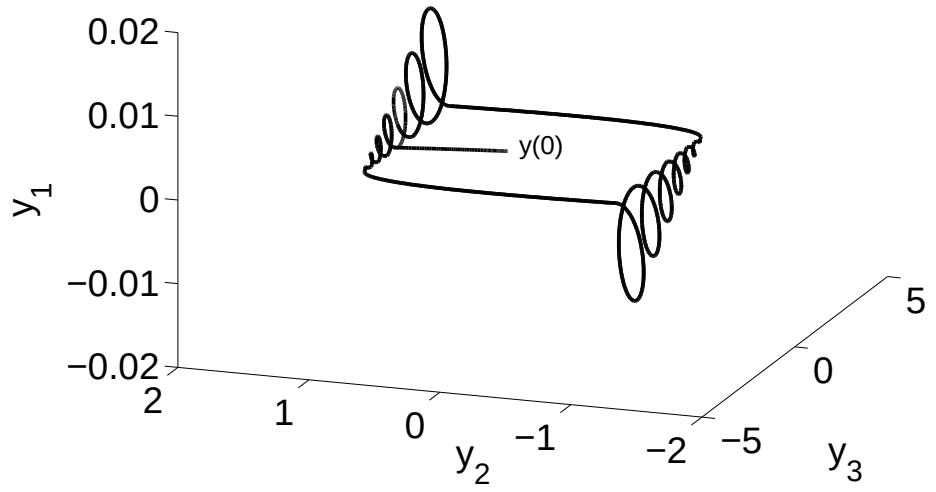


Figure 3: Phase diagram for Problem 3

for a very short time. These regions correspond to the points $y(t)$ where $y_1(t) = 0$ and they can be seen in Figure 4. Due to the discrete form of the numerical approximation there is a danger that "small" sliding regions may go undetected unless a robust detection code is used. In `DISODE45` this is done by setting the parameter `eventcontrol`= 3

Problem 4. This problem models two masses linked by a spring and moving on a surface with Coulomb friction [**?**, pp. 162]. Both masses are equal, but made of different material, thus both their frictional coefficients, μ_1 and μ_2, and the associated forces, F_1 and F_2 are different. The surface where the masses move has two separate parts, which makes the friction coefficients change.

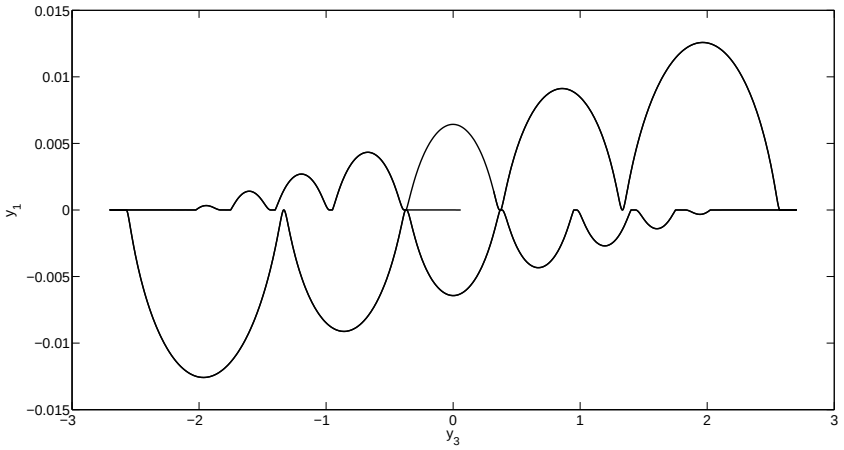
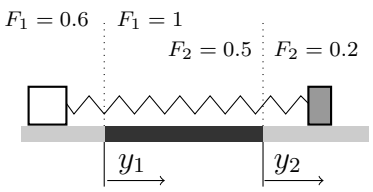


Figure 4: Projection onto the $y_2 = 0$ plane of the phase diagram for Problem 3.

$$y_1'' = -k(y_1 - y_2) - F_1 \operatorname{sign}(y_1'),$$
$$y_2'' = -k(y_2 - y_1) - F_2 \operatorname{sign}(y_2'),$$
$$F_1 = \begin{cases} 0.6 & \text{if } y_1 \leq 0 \\ 1 & \text{if } y_1 > 0 \end{cases}$$
$$F_2 = \begin{cases} 0.5 & \text{if } y_2 \leq 0 \\ 0.2 & \text{if } y_2 > 0 \end{cases}$$
$$y_1(0) = -2, y_2(0) = 3, y_1'(0) = 0, y_2'(0) = 0$$
$$k = 1, \quad t \in [0, 6].$$



This problem has four switching surfaces, $g_1(y, y') = y_1$, $g_2(y, y') = y_2$, $g_3(y, y') = y_1'$ and $g_4(y, y') = y_2'$. The solution, with these initial conditions, first passes through several transversal discontinuities and after some time it enters a sliding region onto $g_1(y, y') = 0$. Inside this sliding region, it crosses transversal discontinuities three times and finally reaches a co-dimension 2 sliding region $y_1' = y_2' = 0$, where neither masses move, and the system remains stationary.

The components of the solution $y_1(t)$ and $y_2(t)$ of this problem as well as their derivatives are depicted in Figure 5. The 18 discontinuity points are indicated by means of small circles.

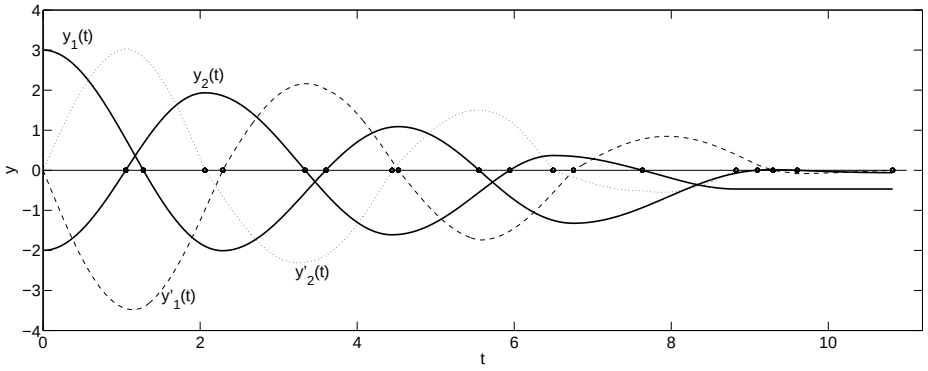


Figure 5: Phase diagram for Problem 4

5.1 Summary

All problems have been integrated with the code `DISODE45` with error tolerances ranging from 10^{-2} to 10^{-9} (we used the same absolute and relative error tolerance).

For each problem we have computed the numerical discontinuity points $(t_d, y_d) \simeq (t_{dis}, y_{dis})$ and we have obtained their maximum errors

$$Err_{t_d} = \max_{disc} |t_d - t_{dis}|, \quad Err_{y_d} = \max_{disc} \|y_d - y_{dis}\|_2.$$

We have also computed the norm of the global error at the end point of the integration interval GE.

The results for problems 1 to 4 are collected in Tables **??** to **??**. To get more insight about the performance of the code we include also in the tables the following information:

Nfcn:	Number of evaluations of the vector field f,
Ngn:	Number of evaluations of switching functions g,
Nac:	Number of accepted steps,
Nre:	Number of rejected steps,
NacS:	Number of accepted steps in sliding regions,
NreS:	Number of rejected steps in sliding regions.

In all the cases the code was able to locate accurately all the discontinuity points and to integrate all the problems successfully.

As mentioned above, for Problem 3, some of the sliding regions are "small" and the discrete integrator can not detect them. This happens for error tolerances from 10^{-3} to 10^{-6}. To avoid this, we integrated the problem with the option `eventcontrol=3` that checks if there are any discontinuity inside every step at 7 uniformly distributed points. With this option, all the discontinuities are properly located. For error tolerances 10^{-7} to 10^{-9} the

Table 1: Problem 1

Tol	Nfcn	Ngn	Nac	Nre	NacS	NreS	Err_{t_d}	Err_{y_d}	GE
$1.e-3$	940	5470	76	2	20	3	$2.3e-1$	$5.8e-2$	$7.4e-2$
$1.e-4$	1010	5674	83	2	22	3	$2.9e-3$	$1.4e-3$	$7.9e-4$
$1.e-5$	1218	7823	87	8	30	6	$5.1e-4$	$1.9e-5$	$1.5e-4$
$1.e-6$	1410	8627	109	4	35	6	$6.6e-5$	$3.6e-6$	$1.9e-5$
$1.e-7$	1660	9691	135	8	43	9	$5.6e-6$	$4.2e-7$	$1.7e-6$
$1.e-8$	2220	13045	176	12	61	9	$4.4e-7$	$3.8e-8$	$1.3e-7$
$1.e-9$	2980	17011	242	17	91	10	$3.8e-8$	$3.7e-9$	$1.1e-8$

Table 2: Problem 2

Tol	Nfcn	Ngn	Nac	Nre	Err_{t_d}	Err_{y_d}	GE
$1.e-3$	1065	1148	113	7	$2.2e-4$	$8.2e-4$	$3.0e-4$
$1.e-4$	1118	1092	118	13	$1.1e-4$	$7.5e-5$	$1.8e-4$
$1.e-5$	1243	1006	135	19	$2.4e-5$	$6.0e-5$	$2.7e-5$
$1.e-6$	1666	1241	184	44	$8.9e-6$	$2.5e-5$	$1.0e-5$
$1.e-7$	2280	1519	260	71	$9.2e-7$	$2.7e-6$	$1.1e-6$
$1.e-8$	3268	1868	400	89	$8.6e-8$	$2.5e-7$	$6.6e-8$
$1.e-9$	4657	2336	613	106	$8.0e-9$	$2.3e-8$	$8.6e-9$

option `eventcontrol=0` works well. This is the reason why the number of evaluations of the switching function is smaller for these three tolerances. For problem 4 with error tolerance 10^{-3} the numerical solution reaches the co-dimension 2 sliding region at $t = 9.228$ instead of at $t = 10.8197$, and the last discontinuity points are not detected. This is due to the fact that from $t = 9.2$ the values of $y_1'(t)$ and $y_2'(t)$ are very close to zero and the allowed errors of approximately 10^{-3} make the numerical solution enter the co-dimension 2 sliding area earlier than the exact solution. This phenomenon can not be avoided by using a more robust detection control but only by using a more stringent error tolerance in the integration. For error tolerances less than or equal to 10^{-4} all the discontinuity points are located.

Conclusions

Reliable algorithms implemented in the adaptive code `DISODE45` for the numerical integration of PieceWise Smooth differential problems have been presented. This code is based on the embedded pair of formulas DOPRI5(4)7FM

Table 3: Problem 3

Tol	Nfcn	Ngn	Nac	Nre	NacS	NreS	Err_{t_d}	Err_{y_d}	GE
$1.e-3$ *	3826	32804	94	8	72	48	$8.1e-3$	$2.1e-2$	$9.1e-3$
$1.e-4$ *	4420	38427	157	19	93	41	$1.4e-3$	$1.7e-3$	$2.8e-3$
$1.e-5$ *	4964	39406	242	39	91	30	$3.2e-4$	$6.7e-4$	$6.3e-4$
$1.e-6$ *	5075	39034	298	50	95	23	$4.3e-5$	$9.7e-5$	$5.6e-5$
$1.e-7$	6694	37295	453	69	128	23	$4.1e-6$	$1.1e-5$	$3.4e-6$
$1.e-8$	8798	43928	708	76	154	19	$4.8e-7$	$1.1e-6$	$1.8e-7$
$1.e-9$	11092	44691	1102	69	181	11	$4.9e-8$	$1.3e-7$	$6.7e-9$

* `eventcontrol=3` was used. Otherwise not all the discontinuities are detected

Table 4: Problem 4

Tol	Nfcn	Ngn	Nac	Nre	NacS	NreS	Err_{t_d}	Err_{y_d}	GE
$1.e-3$	824	4922	44	0	14	0	3.3e-3	4.1e-4	$3.5e-4$
$1.e-4$	791	4483	45	0	15	0	$2.3e-3$	$2.2e-4$	$2.1e-4$
$1.e-5$	874	5082	52	0	18	0	$5.5e-4$	$5.4e-5$	$5.4e-5$
$1.e-6$	1190	7910	68	0	26	9	$9.1e-5$	$8.9e-6$	$8.9e-6$
$1.e-7$	1577	10750	95	0	35	24	$1.2e-5$	$1.2e-6$	$1.2e-6$
$1.e-8$	2098	14019	136	0	49	30	$1.4e-6$	$1.4e-7$	$1.4e-7$
$1.e-9$	2729	16722	205	0	65	36	$1.5e-7$	$1.5e-8$	$1.5e-8$

of orders 5 and 4 due to Dormand and Prince. These algorithms allow the code to detect, accurately locate and cross the discontinuities. They also let the code deal with co-dimension 1 sliding mode regime for Filippov systems. With these algorithms, the user must provide the definition of the vector field and the functions that define the switching surfaces. Numerical examples show that the code is reliable and efficient.

References

[1] V. Acary and B. Brogliato, *Numerical Methods for Nonsmooth Dynamical Systems*. Lecture Notes in Applied and Computational Mechanics, Vol. 35. Springer-Verlag, Berlin.

[2] R. Ashino, M. Nagase, and R. Vaillancourt, Behind and beyond the MATLAB ODE suite. *Comput. Math. Appl.* 40, 4-5 (2000), 491–512.

[3] Martin Biák, Tomáš Hanus, and Drahoslava Janovská, Some applications of Filippov's dynamical systems. *J. Comput. Appl. Math.* 254 (2013), 132–143.

[4] K. T. Cahu, X. X. Wei, X. Guo, and C. Y. Shen. 2003. Experimental and theoretical simulations of seismic poundings between two adjacents structures. *Earthq. Eng. Struct. Dyn.* 32 (Jan. 2003), 537–554.

[5] M. Calvo, J. I. Montijano, and L. Rández. 1990. A fifth-order interpolant for the Dormand and Prince Runge-Kutta method. *J. Comput. Appl. Math.* 29, 1 (1990), 91–100.

[6] M. Calvo, J. I. Montijano, and L. Rández. 2003. On the solution of discontinuous IVPs by adaptive Runge-Kutta codes. *Numer. Algorithms* 33, 1-4 (2003), 163–182.

[7] M. B. Carver. 1978. Efficient integration over discontinuities in ordinary differential equation simulations. *Math. Comput. Simulation* 20, 3 (1978), 190–196.

[8] M. di Bernardo, C. J. Budd, A. R. Champneys, and P. Kowalczyk. 2008. *Piecewise-smooth dynamical systems*. Applied Mathematical Sciences, Vol. 163. Springer-Verlag London, Ltd., London. xxii+481 pages. Theory and applications.

[9] L. Dieci, C. Elia, and L. Lopez. 2013. A Filippov sliding vector field on an attracting co-dimension 2 discontinuity surface, and a limited loss-of-attractivity analysis. *J. Differential Equations* 254, 4 (2013), 1800–1832.

[10] Luca Dieci and Luciano Lopez. 2011. Sliding motion on discontinuity surfaces of high co-dimension. A construction for selecting a Filippov vector field. *Numer. Math.* 117, 4 (2011), 779–811.

[11] Luca Dieci and Luciano Lopez. 2012. A survey of numerical methods for IVPs of ODEs with discontinuous right-hand side. *J. Comput. Appl. Math.* 236, 16 (2012), 3967–3991.

[12] Luca Dieci and Luciano Lopez. 2013. Numerical solution of discontinuous differential systems: approaching the discontinuity surface from one side. *Appl. Numer. Math.* 67 (2013), 98–110.

[13] Luca Dieci and Luciano Lopez. 2015. One-sided direct event location techniques in the numerical solution of discontinuous differential systems. *BIT* 55, 4 (2015), 987–1003.

[14] J. R. Dormand and P. J. Prince. 1980. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* 6, 1 (1980), 19–26.

[15] D. Ellison. 1981. Efficient automatic integration of ordinary differential equations with discontinuities. *Math. Comput. Simulation* 23, 1 (1981), 12–20.

[16] W. H. Enright, K. R. Jackson, S. P. Nørsett, and P. G. Thomsen. 1988. Effective solution of discontinuous IVPs using a Runge-Kutta formula pair with interpolants. *Appl. Math. Comput.* 27, 4, part II (1988), 313–335.

[17] J. M. Esposito and V. Kumar. 2007. A State Event Detection Algorithm for Numerically Simulating Hybrid Systems with Model singularitie. *Appl. Math. Comput.* 17 (2007), 1–22.

[18] A. F. Filippov. 1988. *Differential equations with discontinuous righthand sides*. Mathematics and its Applications (Soviet Series), Vol. 18. Kluwer Academic Publishers Group, Dordrecht. x+304 pages.

[19] C. W. Gear and O. Østerby. 1984. Solving ordinary differential equations with discontinuities. *ACM Trans. Math. Software* 10, 1 (1984), 23–44.

[20] E. Hairer, S. P. Nørsett, and G. Wanner. 2000. *Solving Ordinary Differential Equations I, Nonstiff Problems* (second corrected printing ed.). Springer-Verlag, Berlin.

[21] E. Hairer and G. Wanner. 1981. *Solving Ordinary Differential Equations II, Stiff and Differential–Algebraic Problems*. Springer-Verlag, Berlin.

[22] C. W. Hou, J. Xu, and H. Zhong. 2007. Freeway Traffic Control Using Iterative Learning Control-Based Ramp Metering and Speed Signaling. *IEEE Trans. Vehic. Tech.* 56, 2 (2007), 466–477.

[23] R. Jankovski. 2005. Non-linear viscoelastic modelling of earthquake-induced structural pounding. *Earthq. Eng. Struct. Dyn.* 34 (2005), 595–611.

[24] Yingwei Li and Huaiqin Wu. 2009. Global stability analysis for periodic solution in discontinuous neural networks with nonlinear growth activations. *Adv. Difference Equ.* (2009), Art. ID 798685, 14.

[25] Albert C. J. Luo. 2009. *Discontinuous dynamical systems on time-varying domains.* Higher Education Press, Beijing; Springer, Berlin. x+228 pages.

[26] Reinhold Mannshardt. 1978/79. One-step methods of any order for ordinary differential equations with discontinuous right-hand sides. *Numer. Math.* 31, 2 (1978/79), 131–152.

[27] P. G. O'Regan. 1970. Step size adjustement at discontinuities for fourth order Runge-Kutta methods. *Comput. J.* 13, 4 (1970), 401–404.

[28] Diana M. Ovalle, Javier García, and Francisco Periago. 2011. Analysis and numerical simulation of a nonlinear mathematical model for testing the manoeuvrability capabilities of a submarine. *Nonlinear Anal. Real World Appl.* 12, 3 (2011), 1654–1669.

[29] Petri T. Piiroinen and Yuri A. Kuznetsov. 2008. An event-driven method to simulate Filippov systems with accurate computing of sliding motions. *ACM Trans. Math. Software* 34, 3 (2008), Art. 13, 24.

[30] Oscar Rosas-Jaimes and Luis Alvarez-Icaza. 2007. Vehicle density and velocity estimation on highways for on-ramp metering control. *Nonlinear Dyn.* 49 (2007), 555–566.

[31] Lawrence F. Shampine and Mark W. Reichelt. 1997. The MATLAB ODE suite. *SIAM J. Sci. Comput.* 18, 1 (1997), 1–22.

[32] Arjan van der Schaft and Hans Schumacher. 2000. *An introduction to hybrid dynamical systems.* Lecture Notes in Control and Information Sciences, Vol. 251. Springer-Verlag London, Ltd., London. xiv+174 pages.