

An implicit algorithm for validated enclosures of the solutions to variational equations for ODEs

Irmina Walawska and Daniel Wilczak*

Faculty of Mathematics and Computer Science
Jagiellonian University
Łojasiewicza 6, 30-348 Kraków, Poland
{Irmina.Walawska, Daniel.Wilczak}@ii.uj.edu.pl

March 29, 2018

Abstract

We propose a new algorithm for computing validated bounds for the solutions to the first order variational equations associated to ODEs. These validated solutions are the kernel of numerics computer-assisted proofs in dynamical systems literature. The method uses a high-order Taylor method as a predictor step and an implicit method based on the Hermite-Obreshkov interpolation as a corrector step. The proposed algorithm is an improvement of the C^1 -Lohner algorithm proposed by Zgliczyński and it provides sharper bounds.

As an application of the algorithm, we give a computer-assisted proof of the existence of an attractor set in the Rössler system, and we show that the attractor contains an invariant and uniformly hyperbolic subset on which the dynamics is chaotic, that is, conjugated to subshift of finite type with positive topological entropy.

MSC: 65G20, 65L05.

Keywords: validated numerics, initial value problem, variational equations, uniform hyperbolicity, chaos

*This research is partially supported by the Polish National Science Center under Maestro Grant No. 2014/14/A/ST1/00453.

1 Introduction.

The aim of this paper is to provide an algorithm that computes validated enclosures for the solutions to the following set of initial value problems

$$\begin{cases} \dot{x}(t) &= f(x(t)), \\ \dot{V}(t) &= Df(x(t)) \cdot V(t), \\ x(0) &\in [x_0], \\ V(0) &\in [V_0], \end{cases} \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a smooth function (usually analytic in the domain) and $[x_0] \subset \mathbb{R}^n$, $[V_0] \subset \mathbb{R}^{n^2}$ are sets of initial conditions. In contrast to standard numerical methods, one step of a validated algorithm for (1) produces sets $[x_1]$ and $[V_1]$ that guarantee to contain $x(h) \in [x_1]$ and $V(h) \in [V_1]$ for all initial conditions $x(0) \in [x_0]$ and $V(0) \in [V_0]$, where $h > 0$ is a time step (usually variable) of the method. The computations are performed in interval arithmetics [25] in order to obtain guaranteed bounds on the expressions we evaluate.

The equation for $V(t)$ is called the variational equation associated with an ODE. Solutions $V(t)$ give us an information about sensitivities of trajectories with respect to initial conditions. They proved to be useful in finding periodic solutions, proving their existence and analysis of their stability [3, 4, 12, 13, 15, 16]. They are used to estimate invariant manifolds of periodic orbits [7, 8]. Derivatives with respect to initial conditions are used to prove the existence of connecting orbits [2, 39, 43] or even (non)uniformly hyperbolic and chaotic attractors [40, 41]. First and higher-order derivatives with respect to initial conditions can be used to study some bifurcation problems [17, 43, 44]. This wide spectrum of applications is our main motivation for developing an efficient algorithm that produces sharp bounds on the solutions to (1).

In principle, the problem (1) can be solved by any algorithm capable to compute validated solution to IVP for ODEs. There are several available algorithms and their implementations — just to mention a few of them: VNODE-LP [27, 28, 29, 30], COSY Infinity [5, 22, 23], CAPD [6], Valencia-IVP [35]. The above mentioned ODE solvers are internally higher-order methods with respect to the initial state, which means that they use at least partial information about the derivatives with respect to initial conditions to reduce the *wrapping effect*. Therefore, direct application of these solvers uses a higher effective dimension (the internal dimension of the solver) than the dimension of the phase space. In the case of the codes VNODE-LP and CAPD, this effective dimension is $(n(n+1))^2$, which dramatically decreases the performance of these methods when applied directly to the extended system (1). This key observation motivated developing the \mathcal{C}^1 -Lohner algorithm [47], which takes into account the block structure of (1) and works in n^2 effective dimension. Even in low dimensions, it is orders of magnitude faster than direct application of a \mathcal{C}^0 solver to the variational equations. However, it does not use derivatives of $V(t)$ with respect to other coefficients of V (i.e. second order derivatives of the original system) to better control the wrapping effect. This is why it usually produces

worse estimations than those obtained from direct application of a \mathcal{C}^0 solver to the extended system.

In this paper, we propose a new algorithm for computation of validated solutions to (1). Our algorithm consists of two steps. First, the high-order Taylor method is used as a predictor step. Then, an implicit method based on the Hermite-Obreshkov (HO) formula is used to compute tighter bounds for the variational equations. This last step is motivated by the very famous and efficient algorithm proposed by Nedialkov and Jackson [29] and implemented by Nedialkov in the VNODE-LP package [28]. We name the proposed algorithm \mathcal{C}^1 -HO because it computes bounds for the first order variational equations and it is based on the Hermite-Obreshkov interpolation formula.

Our algorithm, by its construction, cannot produce worse estimations than the \mathcal{C}^1 -Lohner algorithm. Complexity analysis (see Section 3) shows that, in low dimensions, it is slower than the \mathcal{C}^1 -Lohner algorithm by the factor 9/8 only. This lack of performance is compensated by a significantly smaller truncation error of the method. This allows to take larger time steps when computing the trajectories and thus our algorithm appears to be slightly faster than the \mathcal{C}^1 -Lohner in real applications — see Section 5 for the case study.

We would like to emphasize that the proposed method can be directly extended to the nonautonomous case without increasing the effective dimension of the problem. For simplicity in the notation, we consider the autonomous case only. In [6], we provide an implementation of the \mathcal{C}^1 -HO algorithm for the nonautonomous case.

As an application of the proposed algorithm we give a computer-assisted proof of the following new result concerning the Rössler system [36].

Theorem 1 *For the parameter values $a = 5.7$ and $b = 0.2$ the system*

$$\begin{cases} \dot{x} &= -y - z \\ \dot{y} &= x + by \\ \dot{z} &= b + z(x - a) \end{cases} \quad (2)$$

admits a compact, connected invariant set \mathcal{A} that is an attractor. There is an invariant subset $\mathcal{H} \subset \mathcal{A}$ on which the dynamics is uniformly hyperbolic and chaotic, that is, conjugated to a subshift of finite type with positive topological entropy.

Verification that an ODE is chaotic is not an easy task in general. After development of rigorous ODE solvers there appeared numerous computer-assisted proofs of the existence of chaos in classical low-dimensional systems — just to mention two pioneering results [24, 46]. To the best of our knowledge there are only two computer-assisted proofs of the existence of chaotic and (non)-uniformly hyperbolic attractors for ODEs [40, 41]. These results became possible with development of suitable theory and the algorithms capable to integrate variational equations. In [41] the \mathcal{C}^1 -Lohner algorithm implemented in the CAPD library [6] was used.

The paper is organized as follows. In Section 1.1, we introduce the symbols and notation used in the paper. Section 2 contains description of the algorithm

and the proof of its correctness. In Section 3, we analyze the complexity of the \mathcal{C}^1 -HO algorithm and we compare it to the complexity of the \mathcal{C}^1 -Lohner algorithm. In Section 4, we compare the bounds obtained by the \mathcal{C}^1 -Lohner and \mathcal{C}^1 -HO algorithms on several examples. In Section 5, we give a more detailed statement and proof of Theorem 1. We discuss also how the computing time depends on the choice of the \mathcal{C}^1 -Lohner and \mathcal{C}^1 -HO algorithms to integrate variational equations.

1.1 Notation.

By I we denote the identity matrix of the dimension clear from the context. By Df we denote the derivative of a smooth function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. By $D_x f$ we denote the partial derivative of f with respect to x .

The local flow induced by an ordinary differential equation (ODE) $\dot{x}(t) = f(x(t))$ will be denoted by φ , i.e. $\varphi(\cdot, x) = x(\cdot)$ is the unique solution passing through x at time zero. We will often identify an element $x \in \mathbb{R}^n$ with the function $x(\cdot) = \varphi(\cdot, x)$. We denote $\psi(t, x, V) = D_x \varphi(t, x) \cdot V$. Clearly $\psi(\cdot, x, V)$ is a solution to the first-order variational equation associated with an ODE with the initial conditions x and V .

Let $f : \mathbb{R} \rightarrow \mathbb{R}^n$ be a smooth function. By $f^{(i)}(x)$ we denote the vector of i th derivatives of f . Normalized derivatives (Taylor coefficients) will be denoted by $f^{[i]}(x) = \frac{1}{i!} f^{(i)}(x)$. We apply this notation to derivatives and Taylor coefficients of the flows φ and ψ taken with respect to the time variable

$$\begin{aligned}\varphi^{[i]}(t, x) &:= (\varphi^{[i]}(\cdot, x))(t), \\ \psi^{[i]}(t, x, V) &:= (\psi^{[i]}(\cdot, x, V))(t).\end{aligned}$$

Interval objects will be always denoted in square brackets, for instance $[a] = [\underline{a}, \bar{a}]$ is an interval, and $[v] = ([v_1], \dots, [v_n])$ is an interval vector. Matrices or interval matrices will be denoted by capital letters, for example $[A]$. Vectors and scalars will be always denoted by small letters. We also identify Cartesian product of intervals $[v_1] \times \dots \times [v_n]$ with a vector of intervals $([v_1], \dots, [v_n])$. Thus interval vectors can be seen as subsets of \mathbb{R}^n . The same identification will apply to interval matrices.

The midpoint of an interval $[a] = [\underline{a}, \bar{a}]$ will be denoted by $\hat{a} = (\underline{a} + \bar{a})/2$. The same convention will be used to denote the midpoint of an interval vector or an interval matrix; for example for $[v] = ([v_1], \dots, [v_n])$ we put $\hat{v} = (\hat{v}_1, \dots, \hat{v}_n)$. Sometimes, we will denote the midpoint of products of interval objects by $\text{mid}([V][r])$.

Throughout this article, interval vectors or interval matrices marked with tilde $[\tilde{y}]$, $[\tilde{V}]$ will always refer to rough enclosures for the solutions to the IVP problem (1) — see Section 2.1 for details.

2 The algorithm.

Consider the initial value problem (1) and assume that we have already proved the existence of the solutions at time t_k , and we have computed rigorous bounds $[x_k]$ and $[V_k]$ for $\varphi(t_k, [x_0]) \subset [x_k]$, $\psi(t_k, [x_0], [V_0]) \subset [V_k]$, respectively. Let us fix a time step $h_k > 0$. A rigorous numerical method for (1) consists usually of the following two steps:

- computation of a rough enclosure. In this step, the algorithm validates that solutions indeed exist over the time interval $[t_k, t_k + h_k]$, and it produces sets $[\tilde{y}]$ and $[\tilde{V}]$, called rough enclosures, which satisfy

$$\varphi([0, h_k], [x_k]) \subset [\tilde{y}] \quad \text{and} \quad (3)$$

$$\psi([0, h_k], [x_k], I) \subset [\tilde{V}]. \quad (4)$$

- computation of tighter bounds $[x_{k+1}]$, $[V_{k+1}]$ satisfying $\varphi(t_k + h_k, [x_0]) \subset [x_{k+1}]$ and $\psi(t_k + h_k, [x_0], [V_0]) \subset [V_{k+1}]$.

In the sequel, we give details of each part of the proposed algorithm.

2.1 Computation of a rough enclosure.

This section is devoted to describe a method for finding rough enclosures (3-4). The key observation is that the equation for V in (1) is linear in V , which implies that the following identity holds

$$\psi(t_k + h_k, x_0, V_0) = \psi(h_k, \varphi(t_k, x_0), I) \cdot \psi(t_k, x_0, V_0)$$

provided all quantities are well defined. This implies that

$$\psi(t_k + h_k, [x_0], [V_0]) \subset \psi(h_k, [x_k], I) \cdot [V_k]. \quad (5)$$

Hence, it is sufficient to use I as an initial condition for the variational equations when computing a rough enclosure $[\tilde{V}]$.

One approach to find rough enclosures $[\tilde{y}]$ and $[\tilde{V}]$ is to compute them separately. Given a set $[\tilde{y}]$ satisfying (3) and computed by any algorithm [26, 30], we can try to find an interval matrix $[\tilde{V}]$ such that

$$I + [0, h_k] Df([\tilde{y}]) \cdot [\tilde{V}] \subset [\tilde{V}]. \quad (6)$$

If we succeed, then the interval matrix $[\tilde{V}]$ satisfies (4). This method is known as the First Order Enclosure (FOE). It has, however, at least one significant disadvantage. If we already know that the solutions to the main equations exist over the time step h_k ($[\tilde{y}]$ has been computed) there is no reason to shorten this time step because solutions to variational equation also exist over the same time range. However, this shortening might be necessary to fulfill the condition (6). To avoid this drawback, Zgliczyński proposes [47] a method based on logarithmic

norms that always computes an enclosure $[\tilde{V}]$ satisfying (4) for the same time step h_k , provided we are able to find an enclosure $[\tilde{y}]$ satisfying (3). This type of enclosure is also used in the \mathcal{C}^r -Lohner algorithm [45] for higher order variational equations.

Another strategy is to use the High Order Enclosure (HOE) method [10, 27, 30]. The authors propose to predict a rough enclosure of the form

$$[\tilde{y}] = \sum_{i=0}^m [0, h_k]^i \varphi^{[i]}(0, [x_k]) + [\varepsilon], \quad (7)$$

where $[\varepsilon]$ is an interval vector centered at zero. The inclusion

$$[0, h_k]^{m+1} \varphi^{[m+1]}(0, [\tilde{y}]) \subset [\varepsilon] \quad (8)$$

implies that the set $[\tilde{y}]$ is indeed a rough enclosure, i.e. it satisfies (3). If the inclusion (8) is not satisfied, then we can always find $\bar{h}_k < h_k$ such that (8) holds with this time step, and thus $[\tilde{y}]$ is a rough enclosure for the time step \bar{h}_k . This strategy is very efficient because we do not need to recompute $[\tilde{y}]$. Furthermore, with quite high order m and a reasonable algorithm for time step prediction, we usually have $\bar{h}_k/h_k = \left(\frac{\|\varepsilon\|}{\|\varphi^{[m+1]}(0, [\tilde{y}])\|} \right)^{1/(m+1)} \approx 1$.

The above method can be used to find simultaneously two enclosures $([\tilde{y}], [\tilde{V}])$ for the entire system (1). We predict $[\tilde{y}]$ as in (7) and

$$[\tilde{V}] = \sum_{i=0}^m [0, h_k]^i \psi^{[i]}(0, [x_k], I) + [E]. \quad (9)$$

Then we check simultaneously (8) and

$$[0, h_k]^{m+1} \psi^{[m+1]}(0, [\tilde{y}], I) [\tilde{V}] \subset [E]. \quad (10)$$

Due to linearity of the equation for variational equations we can consider two strategies when (10) is not satisfied. We can

1. either shorten the time step as we do in (8) or
2. compute $[\tilde{V}^0]$ from the logarithmic norms for the same time step h_k as proposed in [47] and set $[E] = [0, h_k]^{m+1} \psi^{[m+1]}(0, [\tilde{y}], I) [\tilde{V}^0]$. Then $[\tilde{V}]$ computed as in (9) with such $[E]$ satisfies (4).

We would like to emphasize that in both cases we do not need to recompute the coefficients $\psi^{[i]}(0, [\tilde{y}], I)$ which is very expensive. The first strategy is recommended when the tolerance per one step is specified which means that there is a maximal norm of $[E]$ which should not be exceeded. The second strategy applies when the fixed time step is used (by the user choice or application specific reason).

Algorithm 1: Predictor.

Input : m - natural number (order of the Taylor method)
 h_k - positive real number (current time step)
 $[x_k], [\tilde{y}]$ - interval vectors
 $[\tilde{V}]$ - interval matrix
Output: $([x_{k+1}^0], [r^0], [V^0], [R^0]) \subset \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n^2} \times \mathbb{R}^{n^2}$
Compute:
 $[A] \leftarrow \sum_{i=0}^m h_k^i \psi^{[i]}(0, [x_k], I);$
 $y_0 \leftarrow \sum_{i=0}^m h_k^i \varphi^{[i]}(0, \hat{x}_k);$
 $[y] \leftarrow \sum_{i=0}^m h_k^i \varphi^{[i]}(0, [x_k]);$
 $[r^0] \leftarrow [0, h_k]^{m+1} \varphi^{[m+1]}(0, [\tilde{y}]);$
 $[x_{k+1}^0] \leftarrow (y_0 + [A]([x_k] - \hat{x}_k)) \cap [y] + [r^0];$
 $[R^0] \leftarrow [0, h_k]^{m+1} \psi^{[m+1]}(0, [\tilde{y}], I)[\tilde{V}];$
 $[V^0] \leftarrow [A] + [R^0];$
return $([x_{k+1}^0], [r^0], [V^0], [R^0]);$

Our tests show that the \mathcal{C}^1 version of (HOE) gives better results than the approach proposed by Zgliczyński, which uses logarithmic norms. Since computing a rough enclosure (4) is not the main topic of the paper we omit details here.

In what follows we assume that we have a routine that returns three quantities: $h_k, [\tilde{y}], [\tilde{V}]$ for which the properties (3) and (4) hold.

2.2 The predictor step.

We give a short description of the \mathcal{C}^1 -Lohner algorithm [47] which will be used as a predictor step in the \mathcal{C}^1 -HO algorithm.

Lemma 2 *Assume $h_k, [x_k], [\tilde{y}], [\tilde{V}]$ are such that (3) and (4) hold. Then the quantities $([x_{k+1}^0], [r^0], [V^0], [R^0])$ computed by the Algorithm 1 satisfy*

$$\varphi(h_k, [x_k]) \subset [x_{k+1}^0], \quad (11)$$

$$\psi(h_k, [x_k], I) \subset [V^0], \quad (12)$$

$$[0, h_k]^{m+1} \varphi^{[m+1]}(0, [\tilde{y}]) \subset [r^0] \quad \text{and} \quad (13)$$

$$[0, h_k]^{m+1} \psi^{[m+1]}(0, [\tilde{y}], [\tilde{V}]) \subset [R^0]. \quad (14)$$

Proof: The Taylor theorem with Lagrange remainder implies that for all $x_k \in [x_k]$ and each component $j = 1, \dots, n$:

$$\varphi_j(h_k, x_k) = \sum_{i=0}^m h_k^i \varphi_j^{[i]}(0, x_k) + h_k^{m+1} \varphi_j^{[m+1]}(\tau_j, x_k) \quad (15)$$

for some $\tau_j \in (0, h_k)$. By the assumptions $\varphi(\tau_j, x_k) \in [\tilde{y}]$ and by the group property of the flow, we have

$$\varphi_j^{[m+1]}(\tau_j, x_k) = \varphi_j^{[m+1]}(0, \varphi(\tau_j, x_k)) \in \varphi_j^{[m+1]}(0, [\tilde{y}]).$$

Therefore

$$\varphi(h_k, x_k) \in \sum_{i=0}^m h_k^i \varphi_j^{[i]}(0, x_k) + h_k^{m+1} \varphi_j^{[m+1]}(0, [\tilde{y}]) \subset [y] + [r^0].$$

Since $[x_k]$ is convex, we can apply the mean value form to the polynomial part of (15) and obtain that for $x_k \in [x_k]$ there holds

$$\sum_{i=0}^m h_k^i \varphi_j^{[i]}(0, x_k) \in \sum_{i=0}^m h_k^i \varphi_j^{[i]}(0, \hat{x}_k) + [A](x_k - \hat{x}_k) \subset y_0 + [A]([x_k] - \hat{x}_k).$$

Gathering the above together, we obtain

$$\varphi(h_k, [x_k]) \subset (y_0 + [A]([x_k] - \hat{x}_k)) \cap [y] + [r^0] = [x_{k+1}^0]. \quad (16)$$

In a similar way, we deduce that for $x_k \in [x_k]$ and for each component $j, c = 1, \dots, n$ there holds

$$\psi_{j,c}^{[m+1]}(\tau_{j,c}, x_k, I) \in \psi_{j,c}^{[m+1]}(0, [\tilde{y}], [\tilde{V}])$$

for $j, c = 1, \dots, n$, and in consequence

$$\psi(h_k, [x_k], I) \subset [A] + [R^0] = [V^0].$$

■

2.3 The corrector step.

The goal of this section is to set forth a one-step method that refines the results obtained from the predictor step and returns tighter rigorous bounds for the solution to the ODE and its associated variational equation (1). The method combines the algorithm by Nedialkov and Jackson [29] based on the Hermite-Obreshkov interpolation formula with the C^1 -Lohner algorithm for variational equations proposed by Zgliczyński [47]. For reader's convenience, we recall here the key ideas of the Hermite-Obreshkov method.

For natural numbers p, q, i such that $i \leq q$, let

$$c_i^{q,p} = \binom{q}{i} / \binom{p+q}{i}.$$

For a smooth function $u : \mathbb{R} \rightarrow \mathbb{R}^n$ and real numbers h, t we define

$$\Psi_{q,p}(h, u, t) = \sum_{i=0}^q c_i^{q,p} h^i u^{[i]}(t).$$

Using this notation, the Hermite-Obreshkov [32] formula reads

$$\Psi_{q,p}(-h, u, h) = \Psi_{p,q}(h, u, 0) + (-1)^q c_q^{q,p} h^{p+q+1} R(h, u), \quad (17)$$

where

$$R(h, u) = \left(u_1^{[p+q+1]}(\tau_1), \dots, u_n^{[p+q+1]}(\tau_n) \right), \quad \tau_i \in (0, h), i = 1, \dots, n.$$

The key observation which was the main motivation to develop rigorous numerical method based on this formula is that the coefficient $c_q^{q,p} = \binom{p+q}{q}^{-1}$ can be very small for $p = q$. Thus, this formula can have significantly smaller remainder than the Lagrange remainder used in the Taylor series method.

Now we would like to apply (17) to the flows φ and $\psi := D_x \varphi$. Let $[x_k]$ be a set of initial conditions and assume that from the predictor step we have computed $([x_{k+1}^0], [r^0], [V^0], [R^0])$ satisfying (11–14).

Let us fix positive integers p, q such that $m = p + q$, $x_k \in [x_k]$ and put $x_{k+1} = \varphi(h, x_k)$. The formula (17) applied to this case reads

$$\sum_{i=0}^q c_i^{q,p} (-h_k)^i \varphi^{[i]}(0, x_{k+1}) = \sum_{i=0}^p c_i^{p,q} h_k^i \varphi^{[i]}(0, x_k) + \varepsilon,$$

where $\varepsilon \in (-1)^q c_q^{q,p} [r^0]$. Identifying vectors x_k, x_{k+1} with unique solutions $x_k(\cdot), x_{k+1}(\cdot)$ to the ODE passing through them at time zero, we obtain the equivalent but shorter form

$$\Psi_{q,p}(-h_k, x_{k+1}, 0) = \Psi_{p,q}(h_k, x_k, 0) + \varepsilon. \quad (18)$$

Take the midpoints $\hat{x}_{k+1}^0 \in [x_{k+1}^0]$, $\hat{x}_k \in [x_k]$. Since interval vectors are convex sets, and the local flow is a smooth function in both variables, we can apply the mean-value form to both sides of (18) to obtain

$$\Psi_{q,p}(-h_k, \hat{x}_{k+1}^0, 0) + J_-(x_{k+1} - \hat{x}_{k+1}^0) = \Psi_{p,q}(h_k, \hat{x}_k, 0) + J_+(x_k - \hat{x}_k) + \varepsilon$$

for some

$$\begin{aligned} J_- &\in [D_x \Psi_{q,p}(-h_k, [x_{k+1}^0], 0)] \quad \text{and} \\ J_+ &\in [D_x \Psi_{p,q}(h_k, [x_k], 0)]. \end{aligned}$$

We obtained a linear equation for x_{k+1}

$$J_-(x_{k+1} - \hat{x}_{k+1}^0) = J_+(x_k - \hat{x}_k) + (\Psi_{p,q}(h_k, \hat{x}_k, 0) - \Psi_{q,p}(-h_k, \hat{x}_{k+1}^0, 0)) + \varepsilon \quad (19)$$

in which the matrices J_{\pm} are unknown, but they can be rigorously bounded. Denoting

$$\begin{aligned} [\delta] &= \Psi_{p,q}(h_k, \hat{x}_k, 0) - \Psi_{q,p}(-h_k, \hat{x}_{k+1}^0, 0), \\ [\varepsilon] &= (-1)^q c_q^{q,p} [r^0], \\ [J_-] &= [D_x \Psi_{q,p}(-h_k, [x_{k+1}^0], 0)], \\ [J_+] &= [D_x \Psi_{p,q}(h_k, [x_k], 0)], \\ [S] &= I - \hat{J}_-^{-1} [J_-], \\ [r] &= \hat{J}_-^{-1} ([\delta] + [\varepsilon]) + [S]([x_{k+1}^0] - \hat{x}_{k+1}^0) \end{aligned}$$

and applying the interval Krawczyk operator [1, 18, 31] to the linear system (19), we obtain that for $x_k \in [x_k]$,

$$\varphi(h_k, x_k) = x_{k+1} \in \hat{x}_{k+1}^0 + \left(\hat{J}_-^{-1}[J_+] \right) ([x_k] - \hat{x}_k) + [r] \quad (20)$$

which is the main evaluation formula in the interval Hermite-Obreshkov method for IVPs presented in [29]. Note that this formula has exactly the same structure as (16) used in the predictor step.

Each coefficients of $[S]$ is an interval containing zero and its diameter tends to zero with $h_k \rightarrow 0$. The vector $[\delta]$ is almost a point vector, and $[\varepsilon]$ can be made as small as we need (manipulating the time step h_k). Therefore, the total error accumulated in $[r]$ is usually very thin in comparison to the size of the set $[x_k]$ we propagate. Thus, the main source of overestimation when evaluating (20) comes from the propagation of the product $\left(\hat{J}_-^{-1}[J_+] \right) ([x_k] - \hat{x}_k)$. There is a wide literature on how to reduce this wrapping effect for such propagation (see [26] for a survey), and we will give some details concerning this issue in Section 2.4.

In what follows we argue that, with a little additional cost, we can compute a possibly tighter enclosure for the solutions to variational equation than the bound $[V^0]$ obtained from the predictor step. Let us fix $x_k \in [x_k]$ and let $V = \psi(h_k, x_k, I)$. Applying (17) to the solutions to the variational equation, we obtain

$$\sum_{i=0}^q c_i^{q,p} (-h_k)^i \psi^{[i]}(0, x_{k+1}, V) = \sum_{i=0}^p c_i^{p,q} h_k^i \psi^{[i]}(0, x_k, I) + E,$$

where $E \in (-1)^q c_q^{q,p} [R^0]$. Since ψ is linear in V , we obtain that the matrix $V = \psi(h_k, x_k, I)$ belongs to the solution set to the linear equation

$$[J_-]V = [J_+] + [E],$$

where $[E] = (-1)^q c_q^{q,p} [R^0]$. Note that from the predictor step we already know that $V \in [V^0]$. Applying the interval Krawczyk operator [1, 18, 31] to this linear system we obtain

$$V \in \hat{J}_-^{-1}([J_+] + [E]) + (I - \hat{J}_-^{-1}[J_-])[V^0] = \hat{J}_-^{-1}([J_+] + [E]) + [S][V^0].$$

Due to linearity of the variational equation, we can reuse the matrices $[J_-]$, $[J_+]$, $[S]$ and \hat{J}_-^{-1} computed in the corrector step for φ . Thus the additional cost is just a few matrix additions and multiplications. Algorithm 2 and Lemma 3 summarize the above considerations.

Lemma 3 *Assume that h_k , $[x_k]$, $[\tilde{y}]$, $[\tilde{V}]$ are such that (3) and (4) hold and that the quadruple $([x_{k+1}^0], [r^0], [V^0], [R^0])$ is returned by the predictor step (Algorithm 1). Then the quantities $([x_{k+1}], [V])$ computed by Algorithm 2 satisfy*

$$\varphi(h_k, [x_k]) \subset [x_{k+1}] \quad \text{and} \quad (21)$$

$$\psi(h_k, [x_k], I) \subset [V]. \quad (22)$$

Algorithm 2: Corrector.

Input : p, q - positive integers
 h_k - positive real number
 $[x_k]$ - interval vectors
 $([x_{k+1}^0], [r^0], [V^0], [R^0])$ - from the predictor step with
 $m = p + q$
Output: $([x_{k+1}], [V])$
Compute:
 $[\delta] \leftarrow \Psi_{p,q}(h_k, \hat{x}_k, 0) - \Psi_{q,p}(-h_k, \hat{x}_{k+1}^0, 0);$
 $[\varepsilon] \leftarrow (-1)^q c_q^{q,p}[r^0];$
 $[J_-] \leftarrow [D_x \Psi_{q,p}(-h_k, [x_{k+1}^0], 0)];$
 $[J_+] \leftarrow [D_x \Psi_{p,q}(h_k, [x_k], 0)];$
 $[S] \leftarrow I - \hat{J}_-^{-1}[J_-];$
 $[r] \leftarrow \hat{J}_-^{-1}([\delta] + [\varepsilon]) + [S]([x_{k+1}^0] - \hat{x}_{k+1}^0);$
 $[R] \leftarrow \hat{J}_-^{-1}([J_+] + (-1)^q c_q^{q,p}[R^0]);$
 $[x_{k+1}] \leftarrow (\hat{x}_{k+1}^0 + (\hat{J}_-^{-1}[J_+])([x_k] - \hat{x}_k) + [r]) \cap [x_{k+1}^0];$
 $[V] \leftarrow ([R] + [S][V^0]) \cap [V^0];$
return $([x_{k+1}], [V]);$

We would like to emphasize that by its construction the proposed algorithm always returns tighter bounds than the \mathcal{C}^1 -Lohner algorithm because the result obtained from the corrector step is intersected with the bound obtained from the predictor step.

2.4 Propagation of product of interval objects.

It is well known that evaluation of expressions in interval arithmetic can produce large overestimation due to dependency of variables and the wrapping effect [1, 20, 25, 31]. To reduce this undesirable drawback we follow the ideas from [20, 26, 29, 47], and we represent subsets of \mathbb{R}^n and \mathbb{R}^{n^2} in the forms (doubletons in [26] terminology)

$$[x_k] = x_k + C_k[r_k] + B_k[s_k] \quad \text{and} \quad (23)$$

$$[V_k] = V_k + A_k[R_k] + Q_k[S_k]. \quad (24)$$

The initial conditions $([x_0], [V_0])$ of (1) are assumed to be already in the form (23–24). The parallelepipeds $x_k + C_k[r_k]$ and $V_k + A_k[R_k]$ are used to store the main part of the sets $[x_k]$ and $[V_k]$, respectively. The terms $B_k[s_k]$ and $Q_k[S_k]$ are used to collect all usually thin quantities that appear during the computation.

According to (5), the bound for $\psi(t_k + h_k, [x_0], [V_0])$ can be computed as

$$[V_{k+1}] = [V][V_k],$$

where $[V]$ satisfies (22). Substituting the representation (24) we obtain

$$[V_{k+1}] \subset [V] (V_k + A_k[R_k] + Q_k[S_k]) \cap (V_{k+1} + A_{k+1}[R_{k+1}] + Q_{k+1}[S_{k+1}]),$$

where the new representation is computed as follows

$$\begin{aligned} [\Delta A] &= ([V] - \widehat{V})(V_k + A_k[R_k]), \\ V_{k+1} &= \widehat{V}V_k, \\ A_{k+1} &= \widehat{V}A_k, \\ [S_{k+1}] &= (Q_{k+1}^{-1}[V][Q_k])[S_k] + Q_{k+1}^{-1}[\Delta A] \quad \text{and} \\ [R_{k+1}] &= [R_k]. \end{aligned}$$

In principle, the matrix Q_{k+1} can be chosen as any invertible matrix. The numerical experiments [20, 26, 29, 47] show that one of the most efficient strategies in reducing the wrapping effect is to compute Q_{k+1} as an orthogonal matrix from the QR decomposition of the point matrix $\widehat{V}Q_k$. Note, that even if the matrix Q_{k+1} is a point matrix, the inverse Q_{k+1}^{-1} must be computed rigorously in interval arithmetic.

Similar strategy is used for propagation of products in

$$\begin{aligned} [x_{k+1}] &\subset \widehat{x}_{k+1}^0 + \left(\widehat{J}_-^{-1}[J_+]\right)([x_k] - \widehat{x}_k) + [r] \\ &= \widehat{x}_{k+1}^0 + \left(\widehat{J}_-^{-1}[J_+]\right)(C_k[r_k] + B_k[s_k]) + [r] \end{aligned}$$

— see [20, 26, 29, 47] for details.

3 Complexity.

In this section, we explain why the \mathcal{C}^1 -HO algorithm may perform better than the \mathcal{C}^1 -Lohner algorithm, even if it has higher computational complexity. A large numerical and theoretical study were performed to compare the Interval Hermite-Obreshkov method (IHO) with the Interval Taylor Series Method (ITS) [27]. It has been shown that, with the same step size and order, the IHO method is more stable and produces smaller enclosures than the ITS method on constant coefficient problems. Furthermore, the IHO method allows the use of a much larger stepsize than the ITS method, thus saving computation time during the whole integration. However, comparing these two methods in the nonlinear case is not as simple as in the constant coefficient case. Our goal is to predict the benefits of performing additional calculations required by the IHO method applied to (1).

3.1 Cost of \mathcal{C}^1 -Lohner and \mathcal{C}^1 -HO methods per step.

We assume that both predictor (Algorithm 1) and corrector (Algorithm 2) have the same order. That is, if the order of the predictor is m , we consider the

corrector step with p and q such that $m = p + q$. In what follows we list the most time-consuming items of the predictor and corrector, which are the core of their computational complexity.

In the analysis give below, we count the number of operations which are really executed by the implementation, rather than the possible theoretical and asymptotic complexity. Therefore, we assume that the product of two square interval matrices is computed by the naive algorithm (three nested loops or equivalent), which executes exactly n^3 interval multiplications.

We would like to emphasize, that the rigorous integration of a differential equation is a very difficult task even in quite low dimensions. Thus, dimensions used in practice are usually less than 20. Computer-assisted proofs for 100-dimensional systems are actually the state of the art — see for instance [15]. Therefore, the use of asymptotically fast algorithms for matrix multiplications, such as the Strassen algorithm [38] or the Coppersmith-Winograd [9] algorithm, does not make any sense.

Let us denote by c_f the cost of evaluating the vector field (1). For the \mathcal{C}^1 -Lohner step we need the following operations (predictor step and propagation of doubleton representations)

- simultaneous computation of $\varphi^{[i]}(0, [x_k])$ and $\psi^{[i]}(0, [x_k], I)$ up to order m . This is performed by means of automatic differentiation techniques, and it takes $c_f(2n+1)(m+1)(m+2)/2$ multiplications — see [34],
- simultaneous computation of $\varphi^{[i]}(0, [\tilde{y}])$ and $\psi^{[i]}(0, [\tilde{y}], I)$ up to order $m+1$. This is performed by means of the automatic differentiation techniques and it takes $c_f(2n+1)(m+2)(m+3)/2$ multiplications — see [34],
- 13 matrix by matrix multiplications, 2 point matrix inversions and 2 point matrix QR decompositions. Approximate QR decomposition of a point matrix is much cheaper than the product of interval matrices and we may assume that it takes $O(n^3)$ with a constant less than one (in terms of interval multiplications). The inversion of a point matrix which is very close to orthogonal is performed by means of the interval Krawczyk operator [18] and takes n^3 (one multiplication) because an approximate result is already known (transposition of an approximate orthogonal matrix). Thus, the total cost of all matrix operations listed above is at most $17n^3$.

We did not list cheaper operations like additions, intersections of interval objects, matrix by vector products. All polynomial evaluations perform in total $O(n^2m)$ interval multiplications, and they add significant cost to linear systems ($c_f = 0$) and to nonlinear systems but with very small number of nonlinear terms ($c_f \ll n$). Thus, we skip them.

To sum up, the total costs of the \mathcal{C}^1 -Lohner step is

$$C_{\text{LO}}(n, m) \simeq c_f(2n+1)(m+2)^2 + 17n^3.$$

In the \mathcal{C}^1 -HO method, we can reuse the Taylor coefficients of φ and ψ computed in the predictor step, which are needed for computing the $[J_+]$ matrix and $\Psi_{p,q}(h_k, \hat{x}_k, 0)$. Thus, the additional cost is

- computation of $\psi^{[i]}(0, [x_{k+1}^0], I)$ up to order q . This is performed by means of automatic differentiation techniques, and it takes $c_f(2n+1)(q+1)(q+2)/2$ operations — see [34],
- computation of $\Psi_{q,p}(-h_k, \hat{x}_{k+1}^0, 0)$ takes $c_f(q+1)(q+2)/2$,
- rigorous inversion of the point matrix \hat{J}_- takes at most $2n^3$ (one non-rigorous inverse and one interval matrix multiplication in the Krawczyk method) and
- 4 interval matrix multiplications require in total $4n^3$ operations.

The total additional cost of the \mathcal{C}^1 -HO step is at most

$$c_f(n+1)(q+1)(q+2) + 6n^3.$$

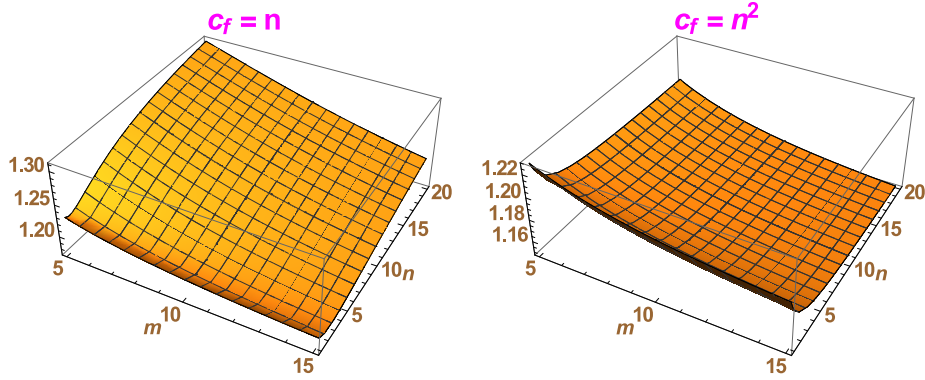


Figure 1: Plot of $C_{\text{HO}}(n, m)/C_{\text{LO}}(n, m)$ for $c_f = n$ and $c_f = n^2$, respectively.

Assume that m is an even number and take $q = p = \frac{m}{2}$. Then the above additional cost of the \mathcal{C}^1 -HO method is approximately

$$\frac{1}{4}c_f(n+1)(m+2)(m+4) + 6n^3.$$

Hence, total computational complexity of the \mathcal{C}^1 -HO method is

$$C_{\text{HO}}(n, m) \simeq C_{\text{LO}}(n, m) + \frac{1}{4}c_f(n+1)(m+2)(m+4) + 6n^3.$$

In general, the complexity depends on the cost of the vector field evaluation c_f which can be arbitrary. In Fig. 1 we plot the graph of $C_{\text{HO}}/C_{\text{LO}}$ for two cases. The case $c_f = n$ means that the number of nonlinear terms in the vector field

is equal to the dimension of the problem. We observe that, if order m of the method is much smaller than the dimension n , then the complexity is dominated by the matrix operations and we have

$$\lim_{n \rightarrow \infty} C_{\text{HO}}(n, m)/C_{\text{LO}}(n, m) = \frac{23}{17} \approx 1.35294.$$

for all fixed values of m . We observe, however, that for reasonable dimensions and orders, this factor is much smaller than the limit value.

A model example for the $c_f = n^2$ case is a second order polynomial vector field with nonzero coefficients in the quadratic terms. In this case we have

$$\lim_{n \rightarrow \infty} C_{\text{HO}}(n, m)/C_{\text{LO}}(n, m) = \frac{9m^2 + 38m + 132}{8m^2 + 3m + 100}.$$

The above analysis shows that the additional cost of the \mathcal{C}^1 -HO method in a typical nonlinear case approaches 1/8. In the next section, we argue that this extra cost of the \mathcal{C}^1 -HO method is compensated by the larger time steps this method can perform without losing the accuracy.

3.2 Maximal allowed time step for a fixed error tolerance.

To obtain insights into the compared methods, we ask the following question: given an acceptable tolerance ε per step, what is the maximal time step h of both methods that guarantees achieving this constraint. For the \mathcal{C}^1 -Lohner method, we have to solve the following inequality

$$\left\| h^{m+1} \varphi^{[m+1]}(0, [\tilde{y}]) \right\| \leq \varepsilon.$$

In general, it is very difficult to answer this question because $[\tilde{y}] = [\tilde{y}(h)]$ depends on h . If $[x_k]$ is a point and ε is very small, we can assume that the vector field is almost constant near $[x_k]$ and thus $\varphi^{[m+1]}(0, [\tilde{y}]) \approx \varphi^{[m+1]}(0, [x_k])$. Since $[x_k] \subset [\tilde{y}]$, we always have $\|\varphi^{[m+1]}(0, [x_k])\| \leq \|\varphi^{[m+1]}(0, [\tilde{y}])\|$. With this simplification, we obtain an upper bound for the time step

$$h_{\text{LO}} := h = \sqrt[m+1]{\frac{\varepsilon}{\|\varphi^{[m+1]}(0, [x_k])\|}}.$$

For the \mathcal{C}^1 -HO method we obtain the following upper bound for the time step

$$h_{\text{HO}} := h = \sqrt[m+1]{\binom{m}{\lceil \frac{m}{2} \rceil} \frac{\varepsilon}{\|\varphi^{[m+1]}(0, [x_k])\|}},$$

where by $\lceil m/2 \rceil$ we denote the smallest integer not smaller than $m/2$. Denote

$$g(m) := h_{\text{HO}}/h_{\text{LO}} = \sqrt[m+1]{\binom{m}{\lceil \frac{m}{2} \rceil}}. \quad (25)$$

It is easy to show that

$$\lim_{m \rightarrow \infty} g(m) = 2.$$

In Fig. 2, we observe that the values of $g(m)$ rapidly grow for small values of m . This is important from practical point of view — even for small order $m = 6$ the \mathcal{C}^1 -HO method allows up to 53% larger time steps than the \mathcal{C}^1 -Lohner method. For $m = 16$ this is 74%. For larger values of the tolerance ε , the computed enclosure $[\tilde{y}]$ for $h = h_{\text{LO}}$ is usually significantly smaller than that computed for $h = h_{\text{HO}}$, which affects the norm $\|\varphi^{[m+1]}(0, [\tilde{y}])\|$. Therefore, the value $g(m)$ is a theoretical upper bound for the possible growth ratio of the time step in the \mathcal{C}^1 -HO method achievable when $\varepsilon \rightarrow 0$.

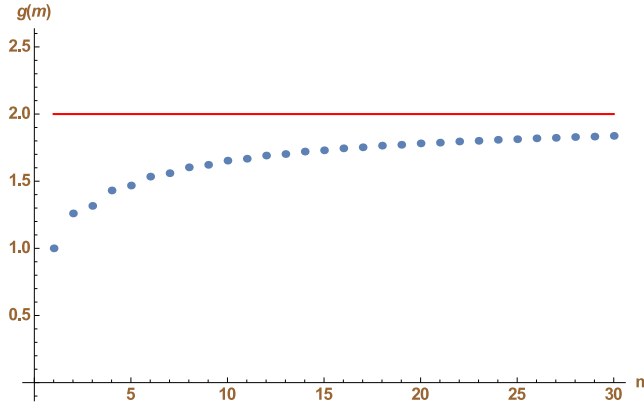


Figure 2: Plot of the theoretical maximal factor of maximal time step in the \mathcal{C}^1 -HO and \mathcal{C}^1 -Lohner methods for a fixed tolerance — see 25.

4 Benchmarks.

In this section, we present the results of a comparison of the \mathcal{C}^1 -Lohner algorithm and the \mathcal{C}^1 -HO algorithm. The structure of the tests is as follows. For a given ODE

- we take an initial condition u which is an approximate periodic orbit for the system;
- we integrate the variational equations along this periodic orbit using the \mathcal{C}^1 -Lohner and \mathcal{C}^1 -HO algorithms with the same algorithm for rough enclosure (HOE), the same order $m = p + q$ of the methods and a constant time step h ;
- we compare the logarithm of the maximal diameter of the interval matrix $[V_k]$ (diameter of the widest component) computed by means of the two algorithms; and

- we repeat the above two steps six times: for two different orders of the numerical methods each for three different time steps.

Fixing the time steps allows us to compare the size of the enclosures returned by the two algorithms over the same time step. This will allow us to conclude that the \mathcal{C}^1 -HO algorithm can take larger time steps than the \mathcal{C}^1 -Lohner algorithm without significant loss of accuracy. The comparison of the two algorithms with variable time steps will be given in Section 5.

The above test is performed for four ODEs: the Lorenz [21] system, the Hénon-Heiles system [14], the Planar Circular Restricted Three Body Problem (PCR3BP), and a 10-dimensional moderately stiff ODE. Below we give initial conditions and discuss obtained results.

The Lorenz system [21] for “classical” parameters is given by

$$\begin{cases} \dot{x} &= 10(y - x), \\ \dot{y} &= x(28 - z) - y, \\ \dot{z} &= xy - \frac{8}{3}z. \end{cases} \quad (26)$$

The Hénon-Heiles system [14] is a hamiltonian ODE given by

$$\begin{cases} \ddot{x} &= -x(1 + 2y), \\ \ddot{y} &= x^2 - y(1 + y). \end{cases} \quad (27)$$

The PCR3BP is a mathematical model that describes motion of a small body with negligible mass in the gravitational influence of two big bodies. The motion is restricted to the plane, and the two main primaries rotate around their common mass centre. The equations for motion of the small body is then given by

$$\begin{cases} \ddot{x} - 2\dot{y} = D_x\Omega(x, y), \\ \ddot{y} + 2\dot{x} = D_y\Omega(x, y), \end{cases} \quad (28)$$

where

$$\Omega(x, y) = \frac{1}{2}(x^2 + y^2) + \frac{1 - \mu}{\sqrt{(x + \mu)^2 + y^2}} + \frac{\mu}{(x - 1 + \mu)^2 + y^2}.$$

The parameter μ stands for the relative mass of the two main bodies. For our tests we fixed $\mu = 0.0009537$, which corresponds to the Sun-Jupiter system.

The last ODE is the Galerkin projection of the following infinite dimensional ODE

$$\dot{a}_k = k^2(1 - \nu k^2)a_k - k \sum_{n=1}^{k-1} a_n a_{k-n} + 2k \sum_{n=1}^{\infty} a_n a_{n+k} \quad (29)$$

onto (a_1, \dots, a_{10}) variables. The above system describes solutions to the one-dimensional Kuramoto-Sivashinsky PDE [19, 37] under periodic and odd boundary conditions, see [48, 49] for derivation.

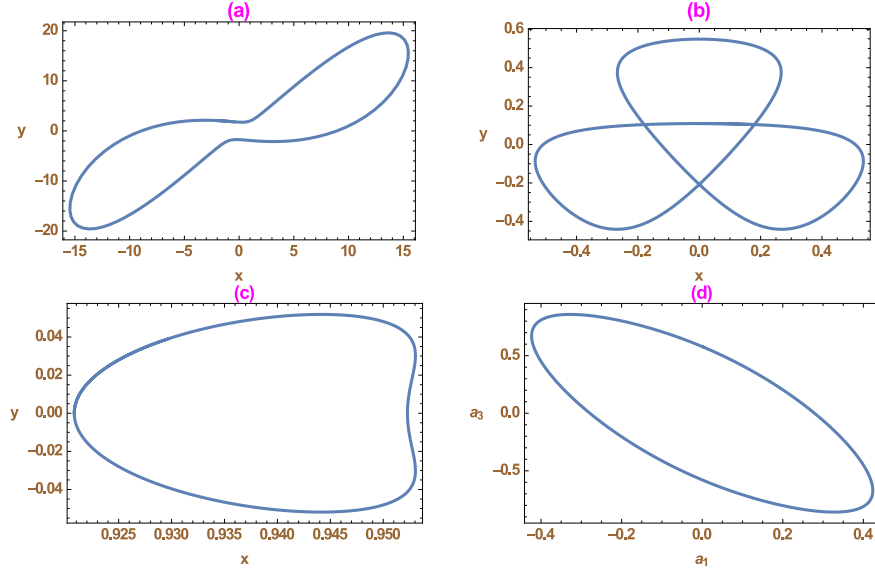


Figure 3: Approximate periodic orbits for (a) the Lorenz system (26), (b) Hénon-Heiles hamiltonian (27), (c) the PCR3BP (28) and (d) a the 10-dimensional Galerkin projection of the Kuramoto-Sivashinsky equation (29), respectively.

We have chosen initial conditions that are close to periodic orbits of these systems (see Fig. 3)

$$\begin{aligned}
 u_{\text{Lorenz}} &= (-2.1473681756955529387, 2.078047612582596404, 27), \\
 u_{\text{Hénon-Heiles}} &= (0.0, 0.10903, 0.5677233993382853, 0.0), \\
 u_{\text{PCR3BP}} &= (0.92080349132074, 0.0, 0.0, 0.1044476727069111) \quad \text{and} \\
 u_{\text{KS}} &= \begin{bmatrix} 0.2012106 \\ 1.2899797585174486 \\ 0.2012106 \\ -0.37786628185377774 \\ -0.042309451521292417 \\ 0.043161614695331821 \\ 0.0069402112803455653 \\ -0.0041564870501656455 \\ -0.00079448972725675504 \\ 0.00033160609117820303 \end{bmatrix}.
 \end{aligned}$$

For the two Hamiltonian systems, the coordinates are given in the order (x, y, \dot{x}, \dot{y}) . The orbit u_{PCR3BP} is the well known L_1 Lyapunov orbit for the Sun-Jupiter-Oterma system. In [48], a computer assisted proof of the existence of a periodic solution for the full infinite dimensional system (29) is given. The projection of

this periodic orbit onto the first 10 coordinates is very close to the point u_{KS} . In fact, due to very strong dissipation, the variables with high indexes have very small impact on the dynamics of (29). The system becomes very stiff even for relative small dimension of the Galerkin projection.

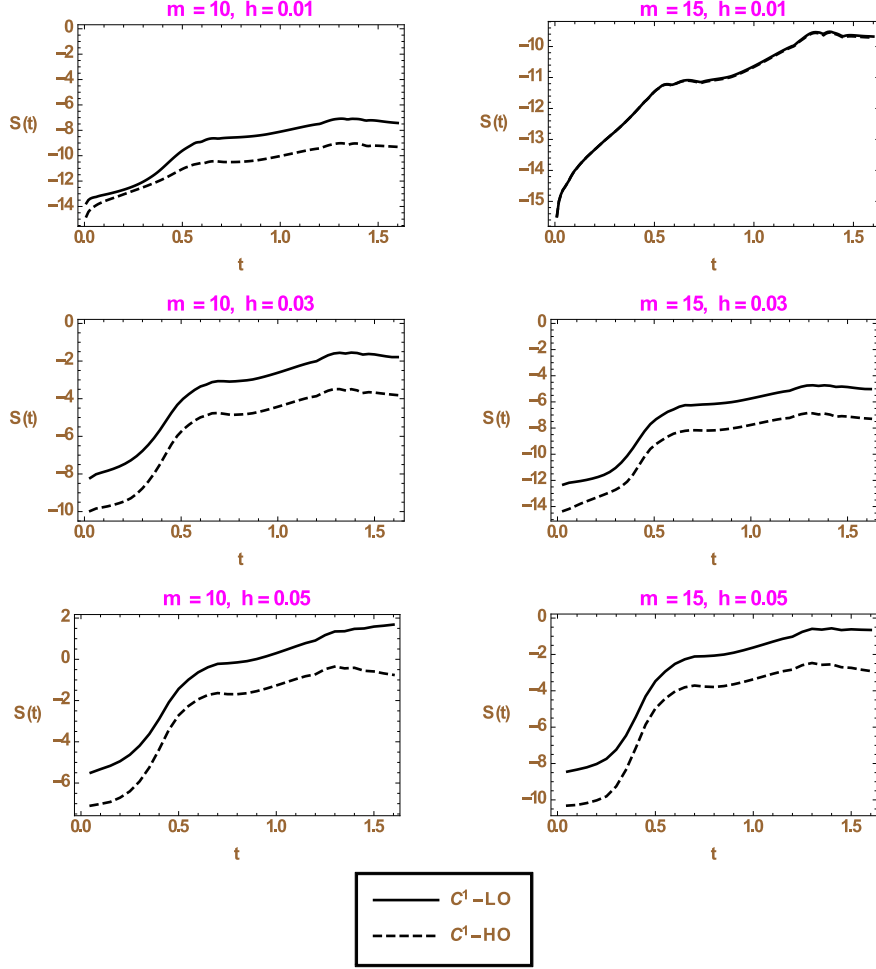


Figure 4: The results of the tests for the Lorenz system. We plot $S(t) = \log_{10} \text{diam}([V(t)])$ along trajectory of the point u_{Lorenz} integrated with order m and with fixed time step h .

In Figs. 4–7 we present results of our numerical experiments. On these figures we show plot of $S(t) = \log_{10} \text{diam}([V(t)])$ along an approximate periodic trajectory, where $\text{diam}([V(t)])$ is the largest width of coefficient in the interval matrix $[V(t)]$. We can see that in each case the C^1 -HO method does not return worse results than the C^1 -Lohner algorithm. This is due to its construction,

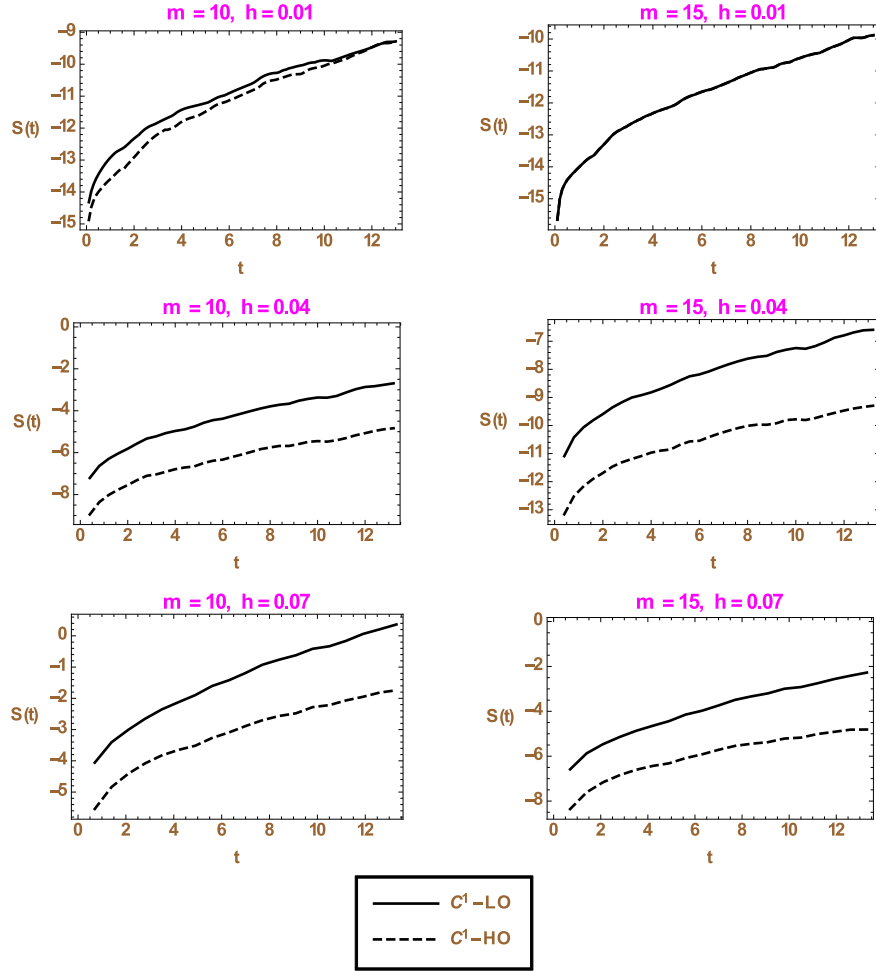


Figure 5: The results of the tests for the Hénon-Heiles system. We plot $S(t) = \log_{10} \text{diam}([V(t)])$ along trajectory of the point $u_{\text{Hénon-Heiles}}$ integrated with order m and with fixed time step h .

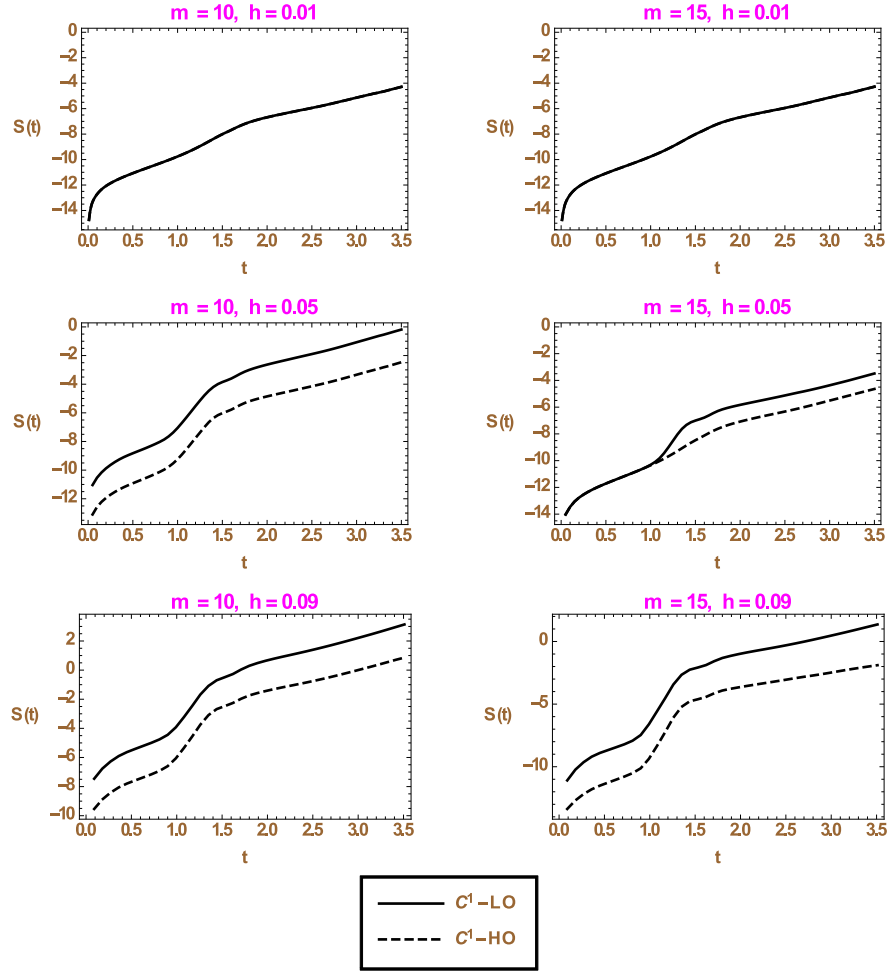


Figure 6: The results of the tests for the PCR3BP. We plot $S(t) = \log_{10} \text{diam}([V(t)])$ along trajectory of the point u_{PCR3BP} integrated with order m and with fixed time step h .

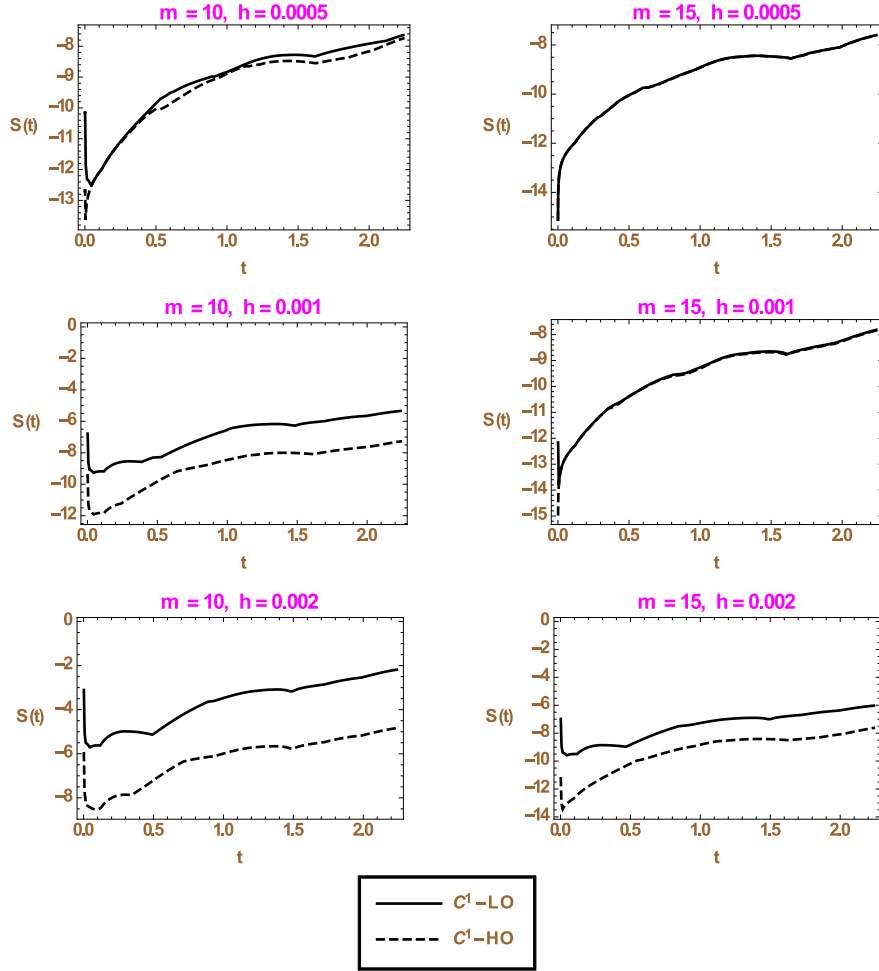


Figure 7: The results of the tests for the 10-dimensional Galerkin projection of the Kuramoto-Sivashinsky equation. We plot $S(t) = \log_{10} \text{diam}([V(t)])$ along trajectory of the point u_{KS} integrated with order m and with fixed time step h .

because the bounds computed in the corrector step are intersected with the estimates from the predictor step, which is used in the \mathcal{C}^1 -Lohner algorithm. Indeed, in the Algorithm 2 we have

$$[V] \leftarrow ([R] + [S][V^0]) \cap [V^0].$$

Looking at the columns of Figs. 4, 5 and 6, we observe that in each case the advantage of the \mathcal{C}^1 -HO method increases when the time step is enlarged, and the obtained bounds can be orders of magnitude tighter. This is due to the fact that the \mathcal{C}^1 -HO method has $c_q^{q,p}$ times tighter truncation error than the Taylor method. To give some numbers, let us take $m = 20$ which is a typical order used in computations. Then $p = q = 10$ and $c_q^{q,p} = c_{10}^{10,10} \approx 5.4 \cdot 10^{-6}$.

Increasing the order of the method makes the truncation error of the \mathcal{C}^1 -Lohner method smaller when the time step is fixed. Therefore, in the right columns of each figure we observe that the \mathcal{C}^1 -Lohner method performs much better than in the left column. The \mathcal{C}^1 -HO method, however, still returns tighter enclosures and is capable to take even larger time steps without significant loss of accuracy. This is especially important for stiff problems, where the time steps used by a nonstiff solver cannot be large and thus integration over large time interval is very expensive. We would like to emphasize, that the maximal possible time step that a rigorous ODE solver can take is limited mainly by the possibility of finding a rough enclosure over the time step. To the best of our knowledge, the HOE algorithm [30], which is nonstiff, is one of the most efficient. Therefore construction of a general rigorous stiff ODE solver without extra knowledge of the system is a challenge. Remarkable exceptions are solvers for infinite-dimensional strongly dissipative systems [11, 48], where the structure of the system is used to construct a dedicated so-called dissipative enclosure.

In Fig. 7 we can see that the \mathcal{C}^1 -HO method can perform much larger time steps keeping very good accuracy of computed bounds.

The bounds obtained by the \mathcal{C}^1 -HO method are tighter than those returned by the \mathcal{C}^1 -Lohner algorithm, but as we observed in Section 3, the \mathcal{C}^1 -HO method is computationally more expensive. In the next section, we argue that this extra cost per step is compensated by the larger time steps we can take.

5 Applications.

In this section, we present an application of the proposed algorithm to a computer-assisted proof of a new result concerning the Rössler system [36]. We focus on the comparison of the time of computation needed to prove this result, when the \mathcal{C}^1 -Lohner algorithm and the \mathcal{C}^1 -HO algorithm are used to integrate variational equations, which are necessary to prove this theorem.

The classical Rössler system [36] is given by (2). When the two parameters a, b vary, the system exhibits wide spectrum of bifurcations. This system admits period doubling bifurcations [44], which lead to chaotic dynamics [46]. In [33], the existence of two periodic orbits was proved by means of the Conley index

theory. All the above results about the system (2) are computer assisted and use rigorous ODE solvers.

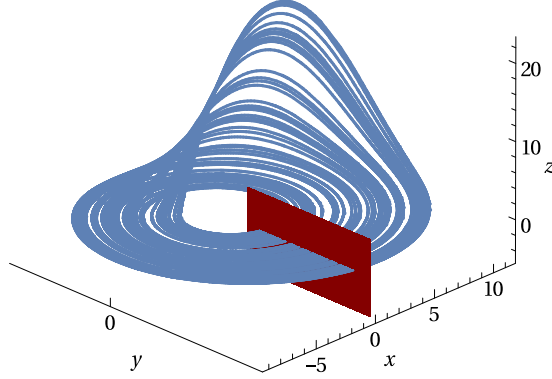


Figure 8: Typical chaotic trajectory of the system (2) and a slice of the Poincaré section Π .

Let $\Pi = \{(x, y, z) \in \mathbb{R}^3 : x = 0 \text{ and } \dot{x} > 0\}$ be a Poincaré section (see Fig. 8) and let $P : \Pi \rightarrow \Pi$ be the Poincaré map. Since the x coordinate is equal to zero on Π , we use only two coordinates (y, z) to describe points on Π .

Theorem 4 *Let $l_B = -10.7$, $r_B = -2.3$, $l_M = -8.4$, $r_M = -7.6$, $l_N = -5.7$, $r_N = -4.6$, $Z = [0.028, 0.034]$ and let*

$$\begin{aligned} B &= [l_B, r_B] \times Z, \\ M &= [l_M, r_M] \times Z \text{ and} \\ N &= [l_N, r_N] \times Z. \end{aligned}$$

For the classical parameter values $a = 0.2$, $b = 5.7$ the following statements hold.

- *The system (2) admits an attractor. The set B is a trapping region for the Poincaré map, i.e. P is well defined on B and $P(B) \subset B$. In particular, there exists a maximal invariant set $\mathcal{A} = \bigcup_{n \geq 0} P^n(B)$ for the map P that is compact and connected.*
- *The maximal invariant set for P^2 in $N \cup M$, denoted by $\mathcal{H} = \text{inv}(P^2, N \cup M) \subset \mathcal{A}$, is uniformly hyperbolic; in particular it is robust under perturbations of the system. The dynamics of P^2 on \mathcal{H} is chaotic in the sense that $P^2|_{\mathcal{H}}$ is conjugated to the Bernoulli shift on two symbols.*

Proof: The tools used in a computer-assisted proof of Theorem 4 are well known, and we summarize them here.

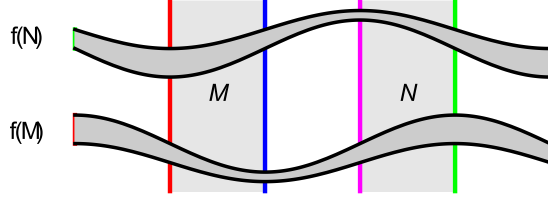


Figure 9: Geometric conditions that guarantee the existence of chaotic dynamics — see (31).

Trapping region. Verification that B is a trapping region for P reduces to checking the inclusion

$$P(B) \subset B.$$

We uniformly subdivided the set B onto $N = 160$ pieces of the form $B_i = [y_i, y_{i+1}] \times Z$, $y_i = l_B + i \cdot (r_B - l_B)/N$. Then we verified that

$$\bigcup_{i=1}^N P(B_i) \subset B. \quad (30)$$

We used a rigorous ODE solver of order 25 from the CAPD library which implements the \mathcal{C}^0 Hermite-Obreshkov algorithm proposed in [29]. Rigorous enclosure for $P(B)$ returned by our routine is shown in Fig. 10.

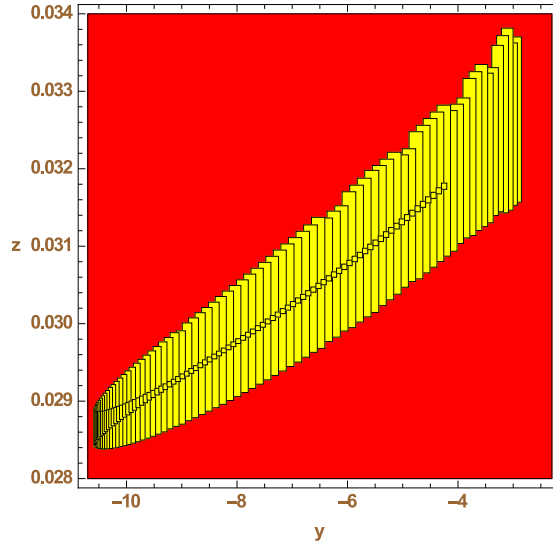


Figure 10: The set B (in red) and a rigorous enclosure for $P(B)$ (in yellow) obtained as the union of enclosures $\bigcup_{i=1}^{160} P(B_i)$ — see (30).

Chaos. Semiconjugacy of $P^2|_{\mathcal{H}}$ to the Bernoulli shift is proved by means of the method of covering relations — the same as in [46] but applied to different sets. It is sufficient to check the following geometric conditions

$$\begin{aligned} \pi_y P^2(y, z) &< l_M && \text{for } (y, z) \in \{l_M\} \times Z, \\ \pi_y P^2(y, z) &> r_N && \text{for } (y, z) \in \{r_M\} \times Z, \\ \pi_y P^2(y, z) &< l_M && \text{for } (y, z) \in \{r_N\} \times Z \quad \text{and} \\ \pi_y P^2(y, z) &> r_N && \text{for } (y, z) \in \{l_N\} \times Z, \end{aligned} \quad (31)$$

where π_y denotes the canonical projection onto the y coordinate. The geometry of these conditions is shown in Fig. 9. For the precise statement of a general theorem concerning, the method of covering we refer to [46].

The conditions (31) have been verified in direct computation. We did not need to subdivide any of the four edges of N and M that appear in (31). Rigorous bounds on $P^2(\{l_M\} \times Z)$, $P^2(\{r_M\} \times Z)$, $P^2(\{l_N\} \times Z)$ and $P^2(\{r_N\} \times Z)$, returned by our routine, are shown in Fig. 11.

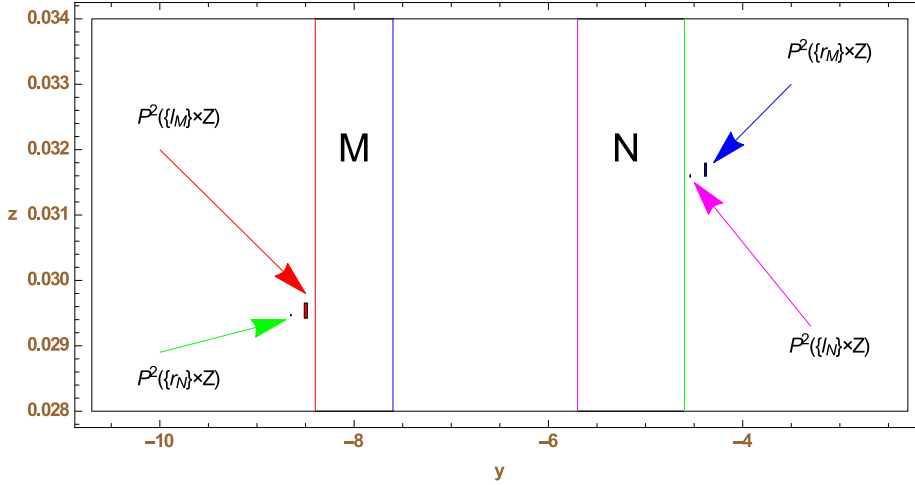


Figure 11: The sets M and N and rigorous enclosures of the images of their exit edges — see (31).

Hyperbolicity and full conjugacy. Uniform hyperbolicity of \mathcal{H} is proved by means of the cone condition introduced in [17]. Here we use our algorithm for integration of variational equations. Derivatives with respect to initial conditions are necessary for computation of the derivative of Poincaré map P^2 . Let Q be a diagonal matrix $Q = \text{Diag}(\lambda, \mu)$ with arbitrary coefficients satisfying $\lambda > 0$ and $\mu < 0$. It has been shown [41] that if for all $(y, z) \in N \cup M$ the matrix

$$DP^2(y, z)^T \cdot Q \cdot DP^2(y, z) - Q \quad (32)$$

is positive definite, then the maximal invariant set for P^2 in $N \cup M$ is uniformly hyperbolic. In our computations we used $\lambda = 1$ and $\mu = -1000$.

We uniformly subdivided both sets N and M onto 48 and 32 equal pieces, respectively (only y coordinate was subdivided). Then, each rectangle was submitted to our routine that integrates the first order variational equations and computes derivative of the Poincaré map P^2 . Given a rigorous bound of the derivative, we checked successfully the condition (32). Note that in the case of 2×2 matrix it is easy to check positive definiteness of a matrix by the Sylvester criterion. ■

5.1 Comparison of time of computation.

In the section, we discuss how the CPU-time needed for verification of the uniform hyperbolicity in Theorem 4 depends on the choice of the algorithm used to integrate variational equations. To this end, we did the following numerical experiment. For fixed parameters

- m — the order of numerical method,
- tol — truncation error per one step of the numerical method,
- Alg — the algorithm used to integrate variational equations (\mathcal{C}^1 -Lohner or \mathcal{C}^1 -HO algorithm)

we compute the following three numbers

- $g_N(m, tol), g_M(m, tol)$ — minimal natural numbers, such that using algorithm Alg , the method of order m with the tolerance tol we were able to check the cone condition (32) subdividing uniformly the sets N, M onto g_N and g_M parts, respectively,
- $t(Alg)$ — CPU time of checking the cone condition on both sets N and M with the algorithm and parameters as above.

Let us emphasize that the vector field of the Rössler system (2) contains only one nonlinear term. Hence, we have $c_f = 1$, and this is almost the worst linear case for the \mathcal{C}^1 -HO method when the complexity is dominated by the matrix operations and the expected time savings from the \mathcal{C}^1 -HO method are smaller — see analysis in Section 3.

In Table 1, we present results from this experiment. We see that in each case the \mathcal{C}^1 -HO algorithm is faster than the \mathcal{C}^1 -Lohner algorithm. Higher computational complexity of the \mathcal{C}^1 -HO algorithm is compensated by significantly smaller truncation error. Therefore, a routine that predicts the time step (the same routine was used in both cases) returns larger time steps for the \mathcal{C}^1 -HO algorithm, and in consequence, the total computing time is smaller. We also notice that in some cases decreasing the tolerance increases the number of subdivisions g_N and g_M needed to check the cone condition — see for instance the row with $m = 14$ and $tol = 10^{-14}$. This is a consequence of many heuristics made in the implementation (for instance reorganization of doubleton representation after reaching some threshold values). These heuristics make the algorithm

m	tol	\mathcal{C}^1 -Lohner			\mathcal{C}^1 -HO			$\frac{t(\text{LO})}{t(\text{HO})}$
		g_M	g_N	$t(\text{LO})$	g_M	g_N	$t(\text{HO})$	
10	10^{-10}	39	33	0.90	48	32	0.82	1.10
10	10^{-12}	38	31	1.29	37	31	1.02	1.26
10	10^{-14}	25	31	1.69	23	30	1.20	1.41
10	10^{-16}	25	28	2.25	24	25	1.67	1.35
14	10^{-10}	45	52	1.13	48	48	0.84	1.34
14	10^{-12}	42	39	1.22	41	36	0.90	1.35
14	10^{-14}	47	33	1.65	49	33	1.21	1.36
14	10^{-16}	36	32	1.70	36	31	1.38	1.23
18	10^{-10}	63	77	1.67	62	56	1.20	1.40
18	10^{-12}	48	56	1.41	63	49	1.26	1.12
18	10^{-14}	44	43	1.76	47	38	1.18	1.50
18	10^{-16}	40	37	1.91	54	34	1.44	1.33
22	10^{-10}	151	95	3.36	101	67	1.93	1.74
22	10^{-12}	78	78	2.44	59	58	1.81	1.35
22	10^{-14}	52	61	2.05	61	49	1.56	1.32
22	10^{-16}	45	41	1.80	47	47	1.53	1.17

Table 1: Comparison of \mathcal{C}^1 -Lohner and \mathcal{C}^1 -HO algorithms.

discontinuous with respect to parameters. Moreover, decreasing the tolerance increases number of time steps needed to compute a full trajectory. This may result in weaker control of unavoidable wrapping effect.

6 Conclusions.

Since the \mathcal{C}^1 -Lohner algorithm appeared [47] it has been proved to be very useful in rigorous analysis of ODEs. In this paper, we proposed an efficient alternative for this algorithm and we provided free implementation of both \mathcal{C}^1 -Lohner and \mathcal{C}^1 -HO algorithms available as a module of the CAPD library [6]. Numerical tests show that the \mathcal{C}^1 -HO algorithm is slightly faster than the widely used \mathcal{C}^1 -Lohner algorithm. We have shown that the \mathcal{C}^1 -HO algorithm may be faster in practical applications. This is not very important when the total time of computation is counted in seconds, as we have seen in Section 5. Any progress matters, however, if a problem requires hundreds or thousands CPU hours: for example verification of the existence of an uniformly hyperbolic attractor of the Smale-Williams type [41] or the coexistence of chaos and hyperchaos in the 4D Rössler system [2, 42]. In the computation reported in [2, 42], the proposed \mathcal{C}^1 -HO algorithm has been used.

In [45], an algorithm for integration of higher order variational equations is presented. Ideas from Section 2 can be directly used to design and implement an algorithm, let us call it \mathcal{C}^r -HO, with the \mathcal{C}^r -Lohner method as a predictor step. This requires encoding rather than theoretical effort and, we hope, this implementation will be available soon as part of the CAPD library [6].

References

- [1] Alefeld, G., 1994. Inclusion methods for systems of nonlinear equations—the interval Newton method and modifications. In: Topics in validated computations (Oldenburg, 1993). Vol. 5 of Stud. Comput. Math. North-Holland, Amsterdam, pp. 7–26.
- [2] Barrio, R., Martínez, M. A., Serrano, S., Wilczak, D., 2015. When chaos meets hyperchaos: 4D Rössler model. *Physics Letters A* 379 (38), 2300–2305.
- [3] Barrio, R., Rodríguez, M., 2014. Systematic computer assisted proofs of periodic orbits of Hamiltonian systems. *Communications in Nonlinear Science and Numerical Simulation* 19 (8), 2660–2675.
- [4] Barrio, R., Rodríguez, M., Blesa, F., 2012. Computer-assisted proof of skeletons of periodic orbits. *Computer Physics Communications* 183 (1), 80–85.
- [5] Berz, M., Makino, K., 1999. New methods for high-dimensional verified quadrature. *Reliable Computing* 5 (1), 13–22.

- [6] CAPD, 2013. Computer Assisted Proofs in Dynamics, a package for rigorous numerics. <http://capd.ii.uj.edu.pl>.
- [7] Capiński, M. J., 2012. Computer assisted existence proofs of Lyapunov orbits at L2 and transversal intersections of invariant manifolds in the Jupiter-Sun PCR3BP. *SIAM J. Applied Dynamical Systems* 11 (4), 1723–1753.
- [8] Capiński, M. J., Wasieczko-Zajac, A., 2015. Geometric proof of strong stable/unstable manifolds with application to the restricted three body problem. *Top. Meth. Non. Anal.* 46 (1), 363–399.
- [9] Coppersmith, D., Winograd, S., 1990. Computational algebraic complexity editorial matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation* 9 (3), 251–280.
- [10] Corliss, G. F., Rihm, R., 1996. Validating an a priori enclosure using high-order Taylor series. In: *In Scientific Computing, Computer Arithmetic, and Validated Numerics*. Akademie Verlag, pp. 228–238.
- [11] Cyranka, J., 2013. Efficient and Generic Algorithm for Rigorous Integration Forward in Time of dPDEs: Part I. *J. Sci. Comp.* 59 (1), 28–52.
- [12] Galias, Z., 2006. Counting low-period cycles for flows. *International Journal of Bifurcation and Chaos* 16 (10), 2873–2886.
- [13] Galias, Z., Tucker, W., May 2008. Rigorous study of short periodic orbits for the Lorenz system. In: *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*. pp. 764–767.
- [14] Hénon, M., Heiles, C., 1964. The applicability of the third integral of motion: Some numerical experiments. *Astronom. J.* 69, 73–79.
- [15] Kapela, T., Simó, C., 2007. Computer assisted proofs for nonsymmetric planar choreographies and for stability of the eight. *Nonlinearity* 20 (5), 1241.
- [16] Kapela, T., Zgliczyński, P., 2003. The existence of simple choreographies for the n-body problem — a computer-assisted proof. *Nonlinearity* 16 (6), 1899.
- [17] Kokubu, H., Wilczak, D., Zgliczyński, P., 2007. Rigorous verification of cocoon bifurcations in the Michelson system. *Nonlinearity* 20 (9), 2147–2174.
- [18] Krawczyk, R., 1969. Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. *Computing*, 187–201.
- [19] Kuramoto, Y., Tsuzuki, T., 1976. Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Progress of Theoretical Physics* 55 (2), 356–369.

- [20] Lohner, R. J., 1992. Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In: Computational ordinary differential equations (London, 1989). Vol. 39 of Inst. Math. Appl. Conf. Ser. New Ser. Oxford Univ. Press, New York, pp. 425–435.
- [21] Lorenz, E., 1963. Deterministic nonperiodic flow. J. Atmospheric Sci. 20, 130–141.
- [22] Makino, K., Berz, M., 2006. Cosy infinity version 9. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 558 (1), 346–350, proceedings of the 8th International Computational Accelerator Physics Conference ICAP 2004.
- [23] Makino, K., Berz, M., 2009. Rigorous integration of flows and ODEs using Taylor models. In: Proceedings of the 2009 Conference on Symbolic Numeric Computation. SNC '09. ACM, New York, NY, USA, pp. 79–84.
- [24] Mischaikow, K., Mrozek M., 1995. Chaos in the Lorenz equations: a computer assisted proof. Bull. Amer. Math. Soc., 32, 66–72.
- [25] Moore, R. E., 1966. Interval analysis. Prentice-Hall, Inc., Englewood Cliffs, N.J.
- [26] Mrozek, M., Zgliczyński, P., 2000. Set arithmetic and the enclosing problem in dynamics. Ann. Polon. Math. 74, 237–259.
- [27] Nedialkov, N., Jackson, K., Corliss, G., 1999. Validated solutions of initial value problems for ordinary differential equations. Applied Mathematics and Computation 105 (1), 21–68.
- [28] Nedialkov, N. S., 2006. VNODE-LP: A validated solver for initial value problems in ordinary differential equations. Tech. Rep. Technical Report CAS-06-06-NN.
- [29] Nedialkov, N. S., Jackson, K. R., 1998. An interval Hermite-Obreschkoff method for computing rigorous bounds on the solution of an initial value problem for an ordinary differential equation. Developments in Reliable Computing 5, 289–310.
- [30] Nedialkov, N. S., Jackson, K. R., Pryce, J. D., 2001. An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. Reliable Computing 7 (6), 449–465.
- [31] Neumaier, A., 1990. Interval methods for systems of equations. Vol. 37 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge.
- [32] Obreschkoff, N., 1940. Neue Quadraturformeln. Abh. Preuss. Akad. Wiss. Math.-Nat. Kl. 1940 (4), 20.

- [33] Pilarczyk, P., 2003. Topological-numerical approach to the existence of periodic trajectories in ODE's. *Discrete Contin. Dyn. Syst. (suppl.)*, 701–708, dynamical systems and differential equations (Wilmington, NC, 2002).
- [34] Rall, L. B., Corliss, G. F., 1996. An introduction to automatic differentiation. In: *Computational differentiation* (Santa Fe, NM, 1996). SIAM, Philadelphia, PA, pp. 1–18.
- [35] Rauh, A., Brill, M., Günther, C., Sep. 2009. A novel interval arithmetic approach for solving differential-algebraic equations with Valencia-IVP. *Int. J. Appl. Math. Comput. Sci.* 19 (3), 381–397.
- [36] Rössler, O. E., 1976. An equation for continuous chaos. *Phys. Lett. A* 57 (5), 397–398.
- [37] Sivashinsky, G., 1977. Nonlinear analysis of hydrodynamic instability in laminar flames — I. derivation of basic equations. *Acta Astronautica* 4 (11), 1177–1206.
- [38] Strassen, V., 1969. Gaussian elimination is not optimal. *Numerische Mathematik* 13 (4), 354–356.
- [39] Szczelina, R., Zgliczyński, P., 2013. A homoclinic orbit in a planar singular ODE — a computer assisted proof. *SIAM J. App. Dyn. Sys.* 12 (3), 1541–1565.
- [40] Tucker, W., 2002. A rigorous ODE solver and Smale's 14th problem. *Found. Comput. Math.* 2 (1), 53–117.
- [41] Wilczak, D., 2010. Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system. *SIAM J. App. Dyn. Sys.* 9 (4), 1263–1283.
- [42] Wilczak D., Serrano S., Barrio R., 2016. Coexistence and Dynamical Connections between Hyperchaos and Chaos in the 4D Rössler System: A Computer-Assisted Proof. *SIAM J. Appl. Dyn. Syst.* 15 (1), 356–390.
- [43] Wilczak, D., Zgliczyński, P., 2009a. Computer assisted proof of the existence of homoclinic tangency for the Hénon map and for the forced damped pendulum. *SIAM J. App. Dyn. Sys.* 8 (4), 1632–1663.
- [44] Wilczak, D., Zgliczyński, P., 2009b. Period doubling in the Rössler system – a computer assisted proof. *Foundations of Computational Mathematics* 9 (5), 611–649.
- [45] Wilczak, D., Zgliczyński, P., 2011. C^r -Lohner algorithm. *Schedae Informaticae* 20, 9–46.
- [46] Zgliczyński, P., 1997. Computer assisted proof of chaos in the Rössler equations and in the Hénon map. *Nonlinearity* 10 (1), 243–252.

- [47] Zgliczyński, P., 2002. C^1 -Lohner algorithm. *Foundations of Computational Mathematics* 2 (4), 429–465.
- [48] Zgliczyński, P., 2004. Rigorous numerics for dissipative partial differential equations II. Periodic orbit for the Kuramoto-Sivashinsky PDE: a computer-assisted proof. *Foundations of Computational Mathematics* 4 (2), 157–185.
- [49] Zgliczyński, P., Mischaikow, K., 2001. Rigorous numerics for partial differential equations: The Kuramoto-Sivashinsky equation. *Foundations of Computational Mathematics* 1 (3), 255–288.