

# Decidability of the Reachability for a Family of Linear Vector Fields

Ting Gan<sup>1</sup>, Mingshuai Chen<sup>2</sup>, Liyun Dai<sup>1</sup>, Bican Xia<sup>1</sup>, and Naijun Zhan<sup>2</sup>✉

<sup>1</sup> LMAM & School of Mathematical Sciences, Peking University, Beijing, China  
{gant,dailiyun,xbc}@pku.edu.cn

<sup>2</sup> State Key Laboratory of Computer Science, Institute of Software,  
CAS, Beijing, China  
{chenms,znj}@ios.ac.cn

**Abstract.** The reachability problem is one of the most important issues in the verification of hybrid systems. Computing the reachable sets of differential equations is difficult, although computing the reachable sets of finite state machines is well developed. Hence, it is not surprising that the reachability of most of hybrid systems is undecidable. In this paper, we identify a family of vector fields and show its reachability problem is decidable. The family consists of all vector fields whose state parts are linear, while input parts are non-linear, possibly with exponential expressions. Such vector fields are commonly used in practice. To the best of our knowledge, the family is one of the most expressive families of vector fields with a decidable reachability problem. The decidability is achieved by proving the decidability of the extension of Tarski's algebra with some specific exponential functions, which has been proved in [21, 22] due to Strzebonski. In this paper, we propose another decision procedure, which is more efficient when all constraints are open sets. The experimental results indicate the efficiency of our approach, even better than existing approaches based on approximation and numeric computation in general.

**Keywords:** Tarski's algebra · Polynomial-exponential function · Reachability · Real root isolation · Cylindrical Algebraic Decomposition (CAD)

## 1 Introduction

Hybrid systems (HSs) integrate computation with physical processes: embedded computers and networks monitor and control physical processes and feedback loops continuously influence computations, which are known as Cyber-Physical Systems (CPSs) nowadays. Applications of CPS span over many safety-critical domains, including communication, healthcare, manufacturing, aerospace, transportation, etc. To guarantee the correctness of these systems is vital so that we can bet our lives on them, and challenging [24]. Therefore, formal methods have been widely used in the verification of HSs. The reachability problem of HSs is

to verify that unsafe states are not reachable from the set of initial states for a given HS, which is one of most important issues in the verification of HSs.

As HSs consist of deep interaction between continuous evolutions and discrete transitions, the reachability problem of most of HSs is undecidable [12], except for some simple cases, either their vector fields, *i.e.*, their continuous evolution parts, are quite simple such as timed automata [4] and multi-rate automata [3], or there are very restrictive constraints on their discrete transitions like o-minimal HSs [15].

In [16], Lafferriere et al. investigated vector fields of the following form

$$\dot{\xi} = A\xi + \mathbf{u}, \tag{1}$$

where  $\xi(t) \in \mathbb{R}^n$  is the state of the system at time  $t$ ,  $A \in \mathbb{R}^{n \times n}$  is the system matrix, and  $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^n$  is a piecewise continuous function which is called the *input*. They obtained the decidability of the reachability problems of the following three families of vector fields:

1.  $A$  is *nilpotent*, *i.e.*  $A^n = 0$ , and each component of  $\mathbf{u}$  is a polynomial;
2.  $A$  is *diagonalizable* with rational eigenvalues, and each component of  $\mathbf{u}$  is of the form  $\sum_{i=1}^m c_i e^{\lambda_i t}$ , where  $\lambda_i$ s are rationals and  $c_i$ s are subject to semi-algebraic constraints;
3.  $A$  is *diagonalizable* with purely imaginary eigenvalues, and each component of  $\mathbf{u}$  of the form  $\sum_{i=1}^m c_i \sin(\lambda_i t) + d_i \cos(\lambda_i t)$ , where  $\lambda_i$ s are rationals and  $c_i$ s and  $d_i$ s are subject to semi-algebraic constraints.

The above results are achieved by reducing the problems into Tarski’s algebra [23]. To the best of our knowledge, these results are the strongest ones on the decidability of the reachability problems of HSs obtained so far.

However, in practice, there are many linear HSs, whose reachability problem is out of the above three families. For instance,

*Example 1.* Consider an LDS  $\dot{\xi} = \begin{bmatrix} \sqrt{2} & & \\ & -\sqrt{2} & \\ & & -1 \end{bmatrix} \xi + \begin{bmatrix} 1-t \\ te^t \\ e^{-t} \end{bmatrix}$ . Let  $X = \{(x_1, x_2, x_3)^T \mid 1 - x_1^2 - x_2^2 - x_3^2 > 0\}$ ,  $Y = \{(y_1, y_2, y_3)^T \mid y_1 + y_2 + y_3 + 2 < 0\}$ . The problem we are concerning is to check if some state in  $Y$  is reachable from  $X$ . Obviously, this problem is not in any of the above three cases.

In this paper, we will generalize the case 2 above by allowing:

- $A$  is *diagonalizable* with *real* eigenvalues, and each component of  $\mathbf{u}$  is of the form  $\sum_{i=1}^m c_i e^{\lambda_i t}$ , where  $\lambda_i$ s are *reals* and  $c_i$ s are subject to semi-algebraic constraints.

Such extension is substantial, as the new family is strictly more expressive than the case 2, for instance, Example 1 falls into this family. In addition, the reachability problem cannot be reduced to Tarski’s algebra any more as in [16].

To obtain the decidability, we have to resort to the decidability of the extension of Tarski’s algebra with functions of the form

$$f(t, \mathbf{x}) = \sum_{i=0}^m f_i(t, \mathbf{x})e^{\lambda_i t}, \tag{2}$$

where  $m \in \mathbb{N}$ ,  $f_i(t, \mathbf{x}) \in \mathbb{R}[t, \mathbf{x}]$ ,  $\lambda_i \in \mathbb{R}$ ,  $i = 0, 1, \dots, m$  and the  $e$  is the natural logarithm. We denote the extension by  $\mathcal{T}_e$ .

Tarski’s algebra is the first-order theory of reals over the structure  $\langle \mathbb{R}; +, -, \cdot, 0, 1 \rangle$ , which is also called the elementary algebra and geometry. In [23], Tarski showed the decidability of Tarski’s algebra. But whether the extension of Tarski’s algebra with exponentiation is decidable (so-called “Tarski’s conjecture”) is still open. In [2], Weispfenning et al. gave a partial solution to Tarski’s conjecture by showing the decidability of the extension of Tarski’s algebra by allowing terms of the form  $f(t, \mathbf{x}, e^t)$ , where  $f(t, \mathbf{x}, y) \in \mathbb{R}[t, \mathbf{x}, y]$ . In [25], Xu et al. considered how to generalize Weispfenning et al’s approach by allowing functions of the form (2), but with the restriction that all the  $\lambda_i$ s are nonnegative integers. While in [21, 22], Strzebonski presented a decision procedure for  $\mathcal{T}_e$  under the assumption of *Schanuel’s conjecture*.

In this paper, we first show how to reduce the reachability problem under consideration to the decidability problem of  $\mathcal{T}_e$ , therefore obtain its decidability due to Strzebonski’s result [21, 22]. Then, we give another decision procedure based on a new real root isolation algorithm for functions of (2) using *Rolle’s theorem* and openCAD [20] or CAD [7], depending on if all constraints are open sets. The complexity of our algorithm is nearly same as Strzebonski’s in general, but more efficient than his if all constraints are open sets. The experimental results indicate the efficiency of our approach, which is better than existing approaches based on approximation and numeric computation in general, e.g., HSolver [18], FLOW\* [6], dReach [14], etc., given that most of them can only be used to compute reachable sets in bounded time.

## 2 Problem Description

In this section, we describe the problem to be solved.

As a convention, we use  $\mathbf{x}$  to stand for a vector variable  $(x_1, \dots, x_n)$ ,  $\mathbb{N}, \mathbb{Q}, \mathbb{R}$  for natural, rational and real numbers respectively,  $\mathbb{R}[\mathbf{x}]$  for the polynomial ring in  $\mathbf{x}$  with coefficients in  $\mathbb{R}$  in what follows.

A term with the form (2) is called *polynomial-exponential function* (PEF). A *linear differential system* (LDS) is of the form (1). We say an LDS is a *linear differential system with polynomial-exponential input* (LDS<sub>PEF</sub>) if every component of  $\mathbf{u}$  is a PEF.

A set  $X \subset \mathbb{R}^n$  is said *semi-algebraic* if it is defined as  $\{\mathbf{x} \in \mathbb{R}^n \mid p_1(\mathbf{x}) \triangleright 0, \dots, p_j(\mathbf{x}) \triangleright 0\}$ , for some polynomial  $p_1(\mathbf{x}), \dots, p_j(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ , where  $\triangleright \in \{\geq, >\}$ ,  $j \in \mathbb{N}$ . A semi-algebraic set  $X$  is said to be *open* if all  $\triangleright$  are instantiated to  $>$ .

Given an initial state  $\xi(0) = \mathbf{x}$ , the solution of (1) at time  $t \geq 0$  is denoted by  $\xi(t) = \Phi(\mathbf{x}, t)$ . Then the *backward reachable set*  $Pre(X)$  and the *forward reachable set*  $Post(X)$  of the LDS (1) from a given set  $X$  are defined as follows:

$$Pre(X) = \{\mathbf{y} \in \mathbb{R}^n \mid \exists \mathbf{x} \exists t : \mathbf{x} \in X \wedge t \geq 0 \wedge \Phi(\mathbf{y}, t) = \mathbf{x}\} \tag{3}$$

$$Post(X) = \{\mathbf{y} \in \mathbb{R}^n \mid \exists \mathbf{x} \exists t : \mathbf{x} \in X \wedge t \geq 0 \wedge \Phi(\mathbf{x}, t) = \mathbf{y}\} \tag{4}$$

Now, the problem under consideration is formulated as follows: Given an  $LDS_{PEF}$  in which  $A$  is diagonalizable with real eigenvalues, an initial set  $X$  and an unsafe set  $Y$ , the problem is to verify whether any unsafe state in  $Y$  is not reachable by some trajectory starting from  $X$ , *i.e.*, whether  $Post(X) \cap Y = \emptyset$ , or dually  $Pre(Y) \cap X = \emptyset$ , or

$$\mathcal{F}(X, Y) = \exists \mathbf{x} \exists \mathbf{y} \exists t : \mathbf{x} \in X \wedge \mathbf{y} \in Y \wedge t \geq 0 \wedge \Phi(\mathbf{x}, t) = \mathbf{y}.$$

### 3 Reduction to the Decidability of $\mathcal{T}_e$

For a given  $LDS_{PEF}$  (1) and a given initial state  $\mathbf{x}$ , the solution can be represented by

$$\xi(t) = \Phi(\mathbf{x}, t) = e^{At} \mathbf{x} + \int_0^t e^{A(t-\tau)} \mathbf{u}(\tau) d\tau, \tag{5}$$

where the matrix exponential  $e^{At}$  is defined as  $e^{At} = \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k$ .

Suppose the matrix  $A$  can be diagonalizable with real eigenvalues, then there exist a diagonal matrix  $D$  and an invertible matrix  $Q$  in  $\mathbb{R}^{n \times n}$  such that  $A = QDQ^{-1}$ . The matrix  $D$  is formed with the eigenvalues of  $A$  along the diagonal and the columns of  $Q$  form a basis of eigenvectors of  $A$ . For brevity, we denote

$$A = (a_{ij}), \quad Q = (q_{ij}), \quad Q^{-1} = (q_{ij}^-), \quad D = \text{diag}(\lambda_1, \dots, \lambda_n),$$

where  $a_{ij}, q_{ij}, q_{ij}^-$  and  $\lambda_1, \dots, \lambda_n$  all are reals.

So, the input of the  $LDS_{PEF}$  can be reformulated as

$$\mathbf{u} = (u_1, u_2, \dots, u_n)^T, \quad \text{where } u_i = \sum_{k=0}^{r_i} g_{ik}(t) e^{\mu_{ik}t} = \sum_{k=0}^{r_i} \sum_{l=0}^{d_{ik}} g_{ikl} \cdot t^l e^{\mu_{ik}t}, \quad i = 1, 2, \dots, n,$$

in which  $\mathbf{v}^T$  means the transposition of  $\mathbf{v}$ ,  $r_i \in \mathbb{N}$ ,  $\mu_{ik} \in \mathbb{R}$ ,  $g_{ik} \in \mathbb{R}[t]$ ,  $d_{ik} \in \mathbb{N}$  is the degree of  $g_{ik}$ ,  $g_{ikl} \in \mathbb{R}$ , for  $i = 1, \dots, n$ ,  $k = 1, \dots, r_i$ ,  $l = 1, \dots, d_{ik}$ .

Now, let us further discuss the solution given in formula (5) with an initial

state  $\mathbf{x} = (x_1, \dots, x_n)$ . First of all, note that  $e^{At} = e^{QDQ^{-1}t} = Q \begin{bmatrix} e^{\lambda_1 t} & & \\ & \ddots & \\ & & e^{\lambda_n t} \end{bmatrix} Q^{-1}$ .

Thus,

$$(e^{At})_{ij} = \sum_{k=1}^n q_{ik} q_{kj}^- e^{\lambda_k t}, \tag{6}$$

$$(e^{At} \mathbf{x})_i = \sum_{j=1}^n (e^{At})_{ij} x_j = \sum_{k=1}^n \left( \sum_{j=1}^n q_{ik} q_{kj}^- x_j \right) e^{\lambda_k t} = \sum_{k=1}^n \alpha_{ik}(\mathbf{x}) e^{\lambda_k t}, \tag{7}$$

where  $\alpha_{ik}(\mathbf{x}) = \sum_{j=1}^n q_{ik}q_{kj}^-x_j, i = 1, \dots, n$ .

Besides, for the second summand in the right side of (5), we have

$$\begin{aligned} \Psi(t) &= \int_0^t e^{A(t-\tau)} \mathbf{u}(\tau) d\tau, \\ (\Psi(t))_i &= \int_0^t (e^{A(t-\tau)} \mathbf{u}(\tau))_i d\tau = \sum_{k=1}^n (e^{At})_{ik} \int_0^t (e^{-A\tau} \mathbf{u}(\tau))_k d\tau \\ &= \sum_{k=1}^n (e^{At})_{ik} \int_0^t \sum_{j=1}^n \sum_{m=1}^n q_{km}q_{mj}^- e^{-\lambda_m\tau} \sum_{h=0}^{r_j} \sum_{l=0}^{d_{jh}} g_{jhl} \tau^l e^{\mu_{jh}\tau} d\tau \\ &= \sum_{k=1}^n \sum_{j=1}^n \sum_{m=1}^n \sum_{h=0}^{r_j} \sum_{l=0}^{d_{jh}} (e^{At})_{ik} q_{km}q_{mj}^- g_{jhl} \int_0^t \tau^l e^{(\mu_{jh}-\lambda_m)\tau} d\tau. \end{aligned}$$

Now, we consider the following two cases:

$$\begin{aligned} \mu_{jh} - \lambda_m \neq 0: & \text{ Then } \int_0^t \tau^l e^{(\mu_{jh}-\lambda_m)\tau} d\tau = \sum_{w=0}^l (-1)^w \frac{l!}{(\mu_{jh}-\lambda_m)^{w+1}(l-w)!} t^{(l-w)} \\ & e^{(\mu_{jh}-\lambda_m)t} + (-1)^{l+1} \frac{l!}{(\mu_{jh}-\lambda_m)^{l+1}}. \\ \mu_{jh} - \lambda_m = 0: & \text{ Then } \int_0^t \tau^l e^{(\mu_{jh}-\lambda_m)\tau} d\tau = \frac{t^{l+1}}{l+1}. \end{aligned}$$

So,  $(\Psi(t))_i$  can be reformulated as:  $(\Psi(t))_i = \sum_{j=0}^{c_i} \psi_{ij}(t)e^{\theta_{ij}t}$ , where  $\psi_{ij}(t) \in \mathbb{R}[t]$ ,  $\theta_{ij} \in \mathbb{R}$  and  $c_i \in \mathbb{N}$ , for  $i = 1, \dots, n, j = 1, \dots, c_i$ . Thus, we have  $(\Phi(\mathbf{x}, t))_i = \sum_{k=1}^n \alpha_{ik}(\mathbf{x})e^{\lambda_k t} + \sum_{j=0}^{c_i} \psi_{ij}(t)e^{\theta_{ij}t}$ , which is a PEF.

Therefore, (3) and (4) can be reformulated accordingly as

$$Pre(X) = \{\mathbf{y} \mid \exists \mathbf{x} \exists t : \mathbf{x} \in X \wedge t \geq 0 \wedge \bigwedge_{i=1}^n \sum_{j=1}^{s_i} \phi_{ij}(\mathbf{y}, t) e^{\nu_{ij}t} = x_i\} \tag{8}$$

$$Post(X) = \{\mathbf{y} \mid \exists \mathbf{x} \exists t : \mathbf{x} \in X \wedge t \geq 0 \wedge \bigwedge_{i=1}^n \sum_{j=1}^{s_i} \phi_{ij}(\mathbf{x}, t) e^{\nu_{ij}t} = y_i\} \tag{9}$$

Likewise, our problem considered in this paper is elaborated as: Given two semi-algebraic sets  $X = \{\mathbf{x} \in \mathbb{R}^n \mid p_1(\mathbf{x}) \triangleright 0, \dots, p_{J_1}(\mathbf{x}) \triangleright 0\}$ ,  $Y = \{\mathbf{y} \in \mathbb{R}^n \mid p_{J_1+1}(\mathbf{y}) \triangleright 0, \dots, p_J(\mathbf{y}) \triangleright 0\}$ , where  $\triangleright \in \{\geq, >\}$ , whether

$$\mathcal{F}(X, Y) = \exists \mathbf{x} \exists \mathbf{y} \exists t : \mathbf{x} \in X \wedge \mathbf{y} \in Y \wedge t \geq 0 \wedge \bigwedge_{i=1}^n \sum_{j=1}^{s_i} \phi_{ij}(\mathbf{x}, t) e^{\nu_{ij}t} = y_i \tag{10}$$

Hence, by Strzebonski’s result [21, 22], we can conclude that

**Theorem 1.** *The reachability problem (10) is decidable if  $\mathcal{T}_e$  is decidable.*

### 4 Decision Procedure for $\mathcal{T}_e$

In this section, we give a decision procedure for  $\mathcal{T}_e$  based on *cylindrical algebraic decomposition* (CAD), due to Collins [7].

The basic idea of CAD is: Given a set  $S$  of polynomials in  $\mathbb{R}[\mathbf{x}]$ , CAD is used to partition  $\mathbb{R}^n$  into connected semi-algebraic sets, called *cells*, such that each polynomial in  $S$  keeps constant *sign* (either +, - or 0) on each *cell*. As CAD plays a fundamental role in computer algebra and real algebraic geometry, in the literature, a numerous works are done on improvement of CAD, e.g., [5, 8, 10, 11, 13, 17]. When constraints are open sets, GCAD [20] or openCAD [11] is enough, which partitions the space  $\mathbb{R}^n$  into a set of *open cells* instead of *cells* (i.e., takes sample points from open cells only), such that on each of which every polynomial in  $S$  keeps constant nonzero *sign* (either + or -). For example, suppose  $f_1 = y - x, f_2 = y + x$ . The graphs of  $f_1 = 0$  and  $f_2 = 0$  decompose  $\mathbb{R}^2$  into 9 cells with different dimensions: four of which are 2-dimensional (open) cells (i.e.,  $f_1 \sim 0 \wedge f_2 \sim 0$ , where  $\sim \in \{>, <\}$ ); four of which are 1-dimensional cells (i.e.,  $f_1 \sim 0 \wedge f_2 = 0, f_1 = 0 \wedge f_2 \sim 0$ , where  $\sim \in \{>, <\}$ ); and one of which is 0-dimensional cell (i.e.,  $f_1 = 0 \wedge f_2 = 0$ ). Complete CAD takes at least one sample point from each of the 9 cells, while GCAD or openCAD takes at least one sample point only from each of the four 2-dimensional (open) cells. Formally,

**Definition 1.** For a polynomial  $f(x_1, \dots, x_n) \in \mathbb{R}[x_1, \dots, x_n]$ , a CAD (openCAD) defined by  $f$  under the order  $x_1 < x_2 < \dots < x_n$  is a set of sample points in  $\mathbb{R}^n$  obtained through the following three phases:

- Projection:** Apply CAD (openCAD) projection operator on  $f$  to get a set of projection polynomials  $\{f_n = f(x_1, \dots, x_n), f_{n-1}(x_1, \dots, x_{n-1}), \dots, f_1(x_1)\}$ ;
- Base:** Choose a rational point in each of the (open) intervals defined by the real roots of  $f_i$ ;
- Lifting:** Substitute each sample point in  $\mathbb{R}^{i-1}$  for  $(x_1, \dots, x_{i-1})$  in  $f_i$  to get a univariate polynomial  $f'_i(x_i)$ , and then, as in Base phase, choose sample points for  $f'_i(x_i)$ . Repeat this process for  $i$  from 2 to  $n$ .

Using CAD (openCAD), we develop a decision procedure for  $\mathcal{T}_e$  as follows:

- Step 1.** Check whether  $X \cap Y = \emptyset$ , if not, it's easy to see that (10) holds.
- Step 2.** Translate the problem to an openCAD solvable problem if  $X$  and  $Y$  are open sets, otherwise a CAD solvable problem. By (9),  $y_i(\mathbf{x}, t) = \sum_{j=1}^{s_i} \phi_{ij}(\mathbf{x}, t)e^{\nu_{ij}t}$ . So, we can replace  $p_j(\mathbf{y})$  with  $p_j(\mathbf{y}(\mathbf{x}, t))$ , which is polynomial in  $\mathbf{x}$  and polynomial-exponential in  $t$ , abbreviated as  $p_j(\mathbf{x}, t)$ , for  $j = J_1 + 1, \dots, J$ . Simply, we define  $p_j(\mathbf{x}, t)$  as  $p_j(\mathbf{x})$ , for  $j = 1, \dots, J_1$ . Thus,  $\mathcal{F}(X, Y)$  in (10) can be reformulated as  $\mathcal{F} = \exists \mathbf{x} \exists t \bigwedge_{j=1}^J p_j(\mathbf{x}, t) \triangleright 0 \wedge t \geq 0$ .
- Step 3.** Eliminate  $x_1, \dots, x_n$  one by one using CAD (openCAD) projection operator on  $\prod_{j=1}^J p_j$  and obtain a set of projection polynomials  $\{q_n(x_1, \dots, x_n, t) = \prod_{j=1}^J p_j, q_{n-1}(x_2, \dots, x_n, t), \dots, q_0(t)\}$ .
- Step 4.** Isolate the real roots of the resulted PEF  $q_0$  based on *Rolle's theorem*, which will be elaborated in the next section.

**Step 5.** Lift the solution using `openCAD` or `CAD` lifting procedure corresponding to Step 2 according to the order  $t, x_n, \dots, x_1$  based on  $\{q_0, \dots, q_n\}$ , and obtain a set  $S$  of *sample points*.

**Step 6.** Check if  $\mathcal{F}$  holds by testing if there exists  $\alpha$  in  $S$  such that  $\bigwedge_{j=1}^J p_j(\alpha) \triangleright 0$ .

In [22], Strzeboński presented another decision procedure for  $\mathcal{T}_e$  completely based on `CAD`. Our decision procedure differentiates from Strzeboński's in the following points:

- When all constraints are open sets, our method is based on `openCAD`, which requires less computation compared to the corresponding complete `CAD`. Therefore, our decision procedure is more efficient (see Sect. 6) in this case. But the two decision procedures share the same complexity in general case.
- In [22], an algorithm for isolating real roots of a given PEF based on *weak Fourier sequence* [21] is given. It is claimed that the algorithm is *complete* under the assumption of *Schanuel's conjecture* [19]. While, in this paper, we give another algorithm to isolate real roots of the resulted PEF  $q_0(t)$  based on *Rolle's theorem*. We prove that our approach is also *complete* under the assumption that  $q_0(t)$  does not have any multiple real roots, which can be implied by *Schanuel's conjecture*.

## 5 Isolating Real Roots of PEFs

In this section we give an algorithm `PEFIsolation` to isolate all real roots for a PEF.

**Definition 2.** Consider a PEF in  $t$  as

$$f(t) = \sum_{i=0}^s f_i(t)e^{\nu_i t}, \tag{11}$$

where  $s \in \mathbb{N}$ ,  $0 \neq f_i \in \mathbb{R}[t]$  and  $\nu_i \in \mathbb{R}$  are pairwise different. Real root isolation of the equation  $f(t) = 0$  is to obtain a set of intervals  $\{I_j = (a_j, b_j) \mid a_j, b_j \in \mathbb{R} \wedge a_j < b_j, j = 1, \dots, J\}$  such that  $I_i \cap I_j = \emptyset$  if  $i \neq j$ , in each  $I_j$  there exists only one real root of  $f(t)$ , and all real roots of  $f(t)$  are contained in  $\bigcup_{j=1}^J I_j$ .

Given an open interval  $I$ , *real root isolation* of  $f(t)$  over  $I$  can be defined similarly. Without loss of generality, in (11), we can assume

$$0 = \nu_0 < \nu_1 < \nu_2 < \dots < \nu_s, f_i(t) \not\equiv 0, \text{ for } i = 0, 1, \dots, s. \tag{12}$$

When  $s = 1$  or every  $\nu_i$  ( $0 \leq i \leq s$ ) is a positive integer, in [2] an algorithm named `ISOL` was proposed to isolate all real roots of  $f(t)$ . This algorithm can be easily extended to the case when all  $\nu_i$  ( $i = 0, \dots, s$ ) are rationals or there exists a nonzero real number  $\kappa$  such that for every  $0 \leq i \leq s$ ,  $\nu_i \kappa$  is a rational.

## 5.1 Lower and Upper Bounds on Real Roots

Similar to [2], we can prove the following theorem, which indicates that there is a lower and upper bound on real roots for any given PEF.

**Theorem 2 (upper bound).** *Let  $f(t)$  be a PEF of the form (11). Then we can obtain an upper bound  $C$  on its real roots through the following procedure:*

1. Find  $C_1 \geq 0$ ,  $M > 0$  such that for all  $t > C_1$ ,  $|f_s(t)| > \frac{1}{M}$ ;
2. Find  $C_2 \geq 0$  and  $k \in \mathbb{N}$  such that for all  $t > C_2$  and for all  $0 \leq i < s$ ,  $|f_i(t)| < \frac{t^k}{sM}$ ;
3. Find  $C_3 \geq 0$  such that for all  $t > C_3$ ,  $t^k < e^{(\nu_s - \nu_{s-1})t}$ ;
4. Set  $C = \max\{C_1, C_2, C_3\}$ .

*Proof.* Let  $t > C$ , then we have  $|f_s(t)| > \frac{1}{M}$ ,  $t^k < e^{(\nu_s - \nu_{s-1})t}$ ,  $|f_i(t)| < \frac{t^k}{sM}$ , for  $i = 0, \dots, s-1$ . Whence

$$\begin{aligned} |f_0(t) + \sum_{i=1}^{s-1} f_i(t)e^{\nu_i t}| &\leq |f_0(t)| + \sum_{i=1}^{s-1} |f_i(t)e^{\nu_i t}| < \frac{t^k}{sM} + \sum_{i=1}^{s-1} \frac{t^k}{sM} e^{\nu_i t} \\ &< \frac{t^k}{sM} e^{\nu_{s-1} t} + \sum_{i=1}^{s-1} \frac{t^k}{sM} e^{\nu_{s-1} t} = \frac{1}{M} t^k e^{\nu_{s-1} t} < \frac{1}{M} e^{\nu_s t} < |f_s(t)e^{\nu_s t}|. \end{aligned}$$

This implies  $f(t) \neq 0$  for any  $t \geq C$ . So  $C$  is an upper bound on the real roots of  $f(t)$ .  $\square$

In order to get a lower bound, a commonly used method is to replace  $f(t)$  with  $g(t) = f(-t)e^{\nu_s t}$ . Then, by Theorem 2, there is an upper bound  $B$  on the real roots of  $g(t) = 0$ . It's easy to see that  $-B$  is a lower bound on the real roots of  $f(t) = 0$ . Thus, we see that all roots of  $f(t) = 0$  are in the interval  $(-B, C)$ . In what follows, we denote by  $L(f) = -B, U(f) = C$ , the lower and upper bounds on the real roots of  $f(t)$ , respectively.

## 5.2 Algorithm

In this subsection, we present our algorithm `PEFIsolation` for isolating all real roots of a given nonzero PEF  $f(t)$  of the form (11).

**Definition 3.** *Let  $f(t)$  be a nonzero PEF of the form (11). We define*

$$\begin{aligned} \text{coff}(f) &= (f_0, f_1, \dots, f_s)^T, \quad \text{nu}(f) = (0, \nu_1, \dots, \nu_s)^T, \\ \text{deg}(f) &= (\text{deg}(f_0), \text{deg}(f_1), \dots, \text{deg}(f_s))^T, \end{aligned}$$

where  $\text{deg}(g)$  means the degree of  $g$ , and as a convention,  $\text{deg}(0) = -1$ . So, (11) can be shorten as  $f(t) = \text{coff}(f)^T \cdot e^{\text{nu}(f)t}$ , where  $e^{\text{nu}(f)t} = (1, e^{\nu_1 t}, \dots, e^{\nu_s t})^T$ ,  $\mathbf{a} \cdot \mathbf{b}$  stands for the inner product of the two vectors, i.e.,  $\sum_{i=1}^n a_i b_i$ .

From Definition 3, it follows

$$\begin{aligned} \text{coff}(f') &= (f'_0, f'_1 + \nu_1 f_1(t), \dots, f'_s + \nu_s f_s(t))^T, \quad \text{nu}(f') = (0, \nu_1, \dots, \nu_s)^T, \\ \text{deg}(f') &= (\max\{\text{deg}(f_0) - 1, -1\}, \text{deg}(f_1), \dots, \text{deg}(f_s))^T, \end{aligned}$$



where  $f'$  denotes the derivative of  $f$  w.r.t.  $t$ .

In the following, we will explain the basic idea behind **PEFI**solation through the following simple example.

*Example 2.* Consider  $\hat{f}(t) = t + 1 + e^{\sqrt{2}t} - (t + 2)e^{\sqrt{5}t}$ .

Firstly, in order to isolate the real roots of  $\hat{f}(t) = 0$ , we need to calculate the upper and lower bounds on all its real roots according to Theorem 2.

Regarding the upper bound of  $\hat{f}(t) = 0$ , we have: (i)  $C_1 = 0$  and  $M = 1$  as  $\forall t \geq 0. |t + 2| > 1$ ; (ii)  $C_2 = 4$  and  $k = 2$  as  $\forall t \geq 4. |t + 1| < \frac{t^2}{2} \wedge 1 < \frac{t^2}{2}$ ; (iii)  $C_3 = 12$  as  $\forall t \geq 12. t^2 < e^{(\sqrt{5}-\sqrt{2})t}$ . Thus, we obtain  $U(\hat{f}) = 12$ .

In order to obtain the lower bound, we have to calculate the upper bound  $U(g)$  of  $g(t) = \hat{f}(-t)e^{\sqrt{5}t}$ , i.e.,  $g(t) = t - 2 + e^{(\sqrt{5}-\sqrt{2})t} - (t - 1)e^{\sqrt{5}t}$ . Because (i)  $\bar{C}_1 = 3$  and  $M = 1$  as  $\forall t \geq 3. |t - 1| > 1$ ; (ii)  $\bar{C}_2 = 4$  and  $k = 2$  as  $\forall t \geq 4. |t - 2| < \frac{t^2}{2}$  and  $1 < \frac{t^2}{2}$ ; (iii)  $\bar{C}_3 = 1$  as  $\forall t \geq 1$  and  $t^2 < e^{\sqrt{2}t}$ , we obtain the upper bound  $U(g) = 4$ .

Therefore, the lower bound  $L(\hat{f}) = -U(g) = -4$  is obtained. Obviously, all real roots of  $\hat{f}(t) = 0$  should be in the interval  $(-4, 12)$ , which implies that we just need to isolate all real roots in  $(-4, 12)$ .

From *differential mean value theorem* (i.e., *Rolle's theorem*), we know there must exist at last one real root of  $f'(t) = 0$  between every two real roots of  $f(t) = 0$ , if  $f(t)$  is continuous differentiable. In order to obtain the real roots of  $f(t) = 0$ , we can try to get the real roots of  $f'(t) = 0$  first. Likewise, in order to obtain the real roots of  $f'(t) = 0$ , we can try to get the real roots of  $f''(t) = 0$  first. We can repeat the above procedure until the real solutions of the  $i$ th derivative of  $f(t)$  for some  $i$  can be achieved. Then, we lift the real solutions of the respective derivative in the inverse order until  $f(t)$  itself. We illuminate the procedure by continuing the running example.

At the beginning,

$$S_0 = \hat{f}(t) = t + 1 + e^{\sqrt{2}t} - (t + 2)e^{\sqrt{5}t},$$

$$\text{coeff}(S_0) = (t + 1, 1, -t - 2)^T, \text{nu}(S_0) = (0, \sqrt{2}, \sqrt{5})^T, \text{deg}(S_0) = (1, 0, 1)^T.$$

Then, we obtain the derivative of  $\hat{f}$  is

$$S_1 = \hat{f}'(t) = 1 + \sqrt{2}e^{\sqrt{2}t} - (\sqrt{5}t + 2\sqrt{5} + 1)e^{\sqrt{5}t},$$

$$\text{coeff}(S_1) = (1, \sqrt{2}, -\sqrt{5}t - 2\sqrt{5} - 1)^T, \text{nu}(S_1) = (0, \sqrt{2}, \sqrt{5})^T, \text{deg}(S_1) = (0, 0, 1)^T.$$

Furthermore, the derivative of  $\hat{f}'$  is

$$\hat{f}''(t) = 0 + 2e^{\sqrt{2}t} - (5t + 2\sqrt{5} + 10)e^{\sqrt{5}t},$$

$$\text{coeff}(\hat{f}'') = (0, 2, -5t - 2\sqrt{5} - 10)^T. \text{nu}(\hat{f}'') = (0, \sqrt{2}, \sqrt{5})^T, \text{deg}(\hat{f}'') = (-1, 0, 1)^T.$$

Clearly,  $\hat{f}''$  and the following  $S_2$  share the same real roots:

$$S_2 = \hat{f}''(t)e^{-\sqrt{2}t} = 2 - (5t + 2\sqrt{5} + 10)e^{(\sqrt{5}-\sqrt{2})t}, \tag{13}$$

$$\text{coeff}(S_2) = (0, 2, -5t - 2\sqrt{5} - 10)^T, \text{nu}(S_2) = (0, 0, \sqrt{5} - \sqrt{2})^T, \text{deg}(S_2) = (-1, 0, 1)^T.$$

Now, the derivative of  $S_2$  is

$$S_3 = S_2' = 0 + 0 + he^{(\sqrt{5}-\sqrt{2})t},$$

$$\text{coeff}(S_3) = (0, 0, h)^T, \text{nu}(S_3) = (0, 0, \sqrt{5} - \sqrt{2})^T, \text{deg}(S_3) = (-1, -1, 1)^T.$$

where  $h = -(5(\sqrt{5} - \sqrt{2})t + 15 + 10\sqrt{5} - 2\sqrt{10} - 10\sqrt{2})$ . Obviously,  $S_3 = 0$  if and only if  $h = 0$ , while the real zeros of  $h$  can be easily achieved by any real root isolation procedure for polynomials (e.g., cf. [9]).

*Remark 1.* In general, suppose  $S_i(t) = f_0(t) + \sum_{j=1}^J f_j(t)e^{\nu_j t}$  with  $0 \neq f_j(t) \in \mathbb{R}[t]$ ,  $0 < \nu_1 < \dots < \nu_J$ , and  $0 < J \in \mathbb{N}$ , then we define  $S_{i+1}(t) = S_i'(t)$  if  $f_0'(t) \neq 0$ ; otherwise,  $S_{i+1}(t) = S_i'(t)e^{-\nu_1 t} = (f_1'(t) + \nu_1 f_1(t)) + \sum_{j=2}^J (f_j'(t) + \nu_j f_j(t))e^{(\nu_j - \nu_1)t}$ . It's obvious that  $S_{i+1} = 0$  shares the same real roots of  $S_i'(t) = 0$ . We construct  $S_{i+1}$  from  $S_i$ , for  $i = 0, \dots$ . This procedure terminates when  $S_k$  is a polynomial for some  $k$ .

**Theorem 3.** Let  $f(t)$  be a PEF,  $f'(t)$  the derivative of  $f(t)$  w.r.t.  $t$ ,  $I = (a, b)$  a non-empty open interval, and  $\mathcal{L}_I(f') = \{I_j | j = 1, \dots, J\}$  a real root isolation of  $f'$  in  $I$ , in which  $I_j = (a_j, b_j)$  with  $a = b_0 < a_1 < b_1 < \dots < a_J < b_J < a_{J+1} = b$ . Furthermore,  $f(t)$  has no real roots in any closed interval  $[a_j, b_j]$ ,  $1 \leq j \leq J$ . Then,  $\{(b_j, a_{j+1}) | f(b_j)f(a_{j+1}) < 0, 0 \leq j \leq J\}$  is a real root isolation of  $f(t)$  in  $I$ .

*Proof.* Since  $f(t)$  has no real roots in any closed interval  $[a_j, b_j]$ ,  $1 \leq j \leq J$ , all real roots of  $f(t)$  are in  $\bigcup_{j=0}^J (b_j, a_{j+1})$  and  $f(b_j)f(a_{j+1}) \neq 0$ . Moreover,  $f(t)$  has at most one real root in each  $(b_j, a_{j+1})$ , otherwise, there must be at least one real root of  $f'(t) = 0$  on it by *Rolle's theorem*, which is a contradiction with the definition of  $\mathcal{L}_I(f')$ . So, if  $f(b_j)f(a_{j+1}) < 0$  then there exists only one real root of  $f(t)$  in  $(b_j, a_{j+1})$ , otherwise no real root of  $f(t)$  in  $(b_j, a_{j+1})$ . This completes the proof.  $\square$

Now, let's continue the running example. As  $e^{(\sqrt{5}-\sqrt{2})t} \neq 0$ , by  $S_3 = he^{(\sqrt{5}-\sqrt{2})t} = 0$ , it follows  $h(t) = 0$ . Thus,  $t = -\frac{15+10\sqrt{5}-2\sqrt{10}-10\sqrt{2}}{5(\sqrt{5}-\sqrt{2})} \in (-5, -4)$ . As  $(-5, -4) \cap (-4, 12) = \emptyset$ , there is no real root of  $S_3 = 0$  in  $(-4, 12)$ . Hence, we have  $\mathcal{L}(S_3) = \emptyset$ . In addition, from (13), we have

$$S_2(-4) = 2 + (10 - 2\sqrt{5})e^{-4(\sqrt{5}-\sqrt{2})} > 0, S_2(12) = 2 - (70 + 2\sqrt{5})e^{12(\sqrt{5}-\sqrt{2})} < 0.$$

Thus, there exists only one real root of  $S_2$  in  $(-4, 12)$  by Theorem 3. Clearly, the real root isolation of  $S_2$  in  $(-4, 12)$  is same as that of  $f''$ .

In order to construct  $\mathcal{L}_{(-4,12)}(S_1)$ , a real root isolation of  $S_1$  in  $(-4, 12)$ , from  $\mathcal{L}_{(-4,12)}(S_2)$  by Theorem 3, the condition that there is no real root of  $S_1$  in  $[a, b]$  for any  $(a, b)$  in  $\mathcal{L}_{(-4,12)}(S_2)$  should be guaranteed. This means that we have to refine the intervals in  $\mathcal{L}_{(-4,12)}(S_2)$  until the condition holds. This is achieved by Algorithm 2 below (see lines 2-13).

The following table is the bisection procedure (line 2-13) in Algorithm 2 to refine the interval  $(-4, 12)$ , in which '∃' (resp. '¬∃') means there exists (no) a real root in the observed interval.

	$(-4,12)$	$(-4,4)$	$(-4,0)$	$(-2,0)$	$(-2,-1)$
$S_2$	$\exists$	$\exists$	$\exists$	$\exists$	$\exists$
$S_1$	$\exists$	$\exists$	$\exists$	$\exists$	$\neg\exists$

Finally, a refined interval  $(a, b) = (-2, -1)$  is obtained, which satisfies the condition of Theorem 3. Thus,  $(-4, -2)$  and  $(-1, 12)$  are two intervals that may contain at most one real root of  $S_1(t) = 0$ . In addition, as  $S_1(-4)S_1(-2) > 0$  and  $S_1(-1)S_1(12) < 0$ ,  $(-1, 12)$  contains a real root of  $S_1(t) = 0$ , but  $(-4, -2)$  does not by Theorem 3. Thus, we get a real root isolation for  $S_1(t) = 0$  in  $(-4, 12)$ , i.e.,  $\mathcal{L}_{(-4,12)}(S_1) = \{(-1, 12)\}$ .

In order to compute  $\mathcal{L}_{(-4,12)}(S_0)$ , we repeat the above procedure, and finally obtain

$$\mathcal{L}_{(-4,12)}(\hat{f}) = \{(-4, -0.59375), (-0.390625, 12)\}.$$

Up to now, we have already explained the main idea of our approach how to isolate real roots of a PEF by the running example. This procedure is implemented in Algorithm 1, whose main steps are understood as follows:

- Step 1:** At line 1, compute the upper and lower bounds of  $f(t)$ ;
- Step 2:** At line 1, construct a sequence  $S_0(t) = f(t)$ ,  $S_1(t)$ ,  $S_2(t)$ , ...,  $S_r(t)$ , where  $S_i$  is a PEF which has the same real roots as the derivative of  $S_{i-1}$ ,  $i = 1, 2, \dots, r$ ,  $r \in \mathbb{N}$ , and  $S_r(t)$  is a polynomial in  $t$ .
- Step 3:** Isolate all real roots of  $S_r(t)$  by calling `UPIsolate(Sr(t))` in line 1. Note that there are lots of work on isolating real roots of a univariate polynomial (e.g., cf [9]).
- Step 4:** At line 1, for  $i = r - 1$  down to 0, construct a real root isolation of  $S_i$  from that of  $S_{i+1}$  using Theorem 3 by calling `PEFI`. Note that during this procedure, we use  $\mathcal{I}_1$  to record all subintervals in which  $f(t)$  has no real roots, while  $\mathcal{I}_2$  to record all subintervals in which  $f'(t)$  has no real roots. So, we only need to construct a real root isolation of  $S_i$  from that of  $S_{i+1}$  on the remainder part of the considered interval by excluding all subintervals in  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , and accordingly update  $\mathcal{I}_1$  and  $\mathcal{I}_2$  in each iteration, see the detail in Algorithm 2.

**Theorem 4 (Correctness of PEFI).** *Algorithm PEFI always terminates correctly.*

*Proof.* The termination of PEFI is obvious because  $f_1(t) = 0$  and  $f_2(t) = 0$  have no common real roots. Then we prove its correctness.

$\mathcal{I}'_1$  and  $\mathcal{I}'_2$  are updated in line 5 and line 7, respectively. Obviously, after every update, the properties of  $\mathcal{I}'_1$  and  $\mathcal{I}'_2$  still hold, i.e.,  $f_1(t)$  has no real roots in  $\cup\mathcal{I}'_1$ ,  $f_2(t)$  has no real roots in  $\cup\mathcal{I}'_2$ , and  $\cup\mathcal{I}'_1 \cap \cup\mathcal{I}'_2 = \emptyset$ . It is also easy to see that, after the second *for loop* at lines 15-18,  $\mathcal{L}'$  is a real root isolation of  $g_1(t)$  on  $(a, b) \setminus \cup(\mathcal{I}'_1 \cup \mathcal{I}'_2)$ . □

**Theorem 5 (Correctness of PEFIsolation).** *Algorithm PEFIsolation always terminates and returns a real root isolation for a given PEF  $f$ , if  $f$  does not have multiple real roots.*

---

**Algorithm 1.** PEFIsolation

---

```

Input:  $f(t)$ , a PEF of the form (11) with the assumption (12), which has no
multiple real roots
Output:  $\mathcal{L}$ , a real root isolation of  $f(t)$ 
1 Calculate a lower bound  $a$  and an upper bound  $b$  on real roots of  $f(t) = 0$ ;
2 set  $\mathcal{I}_1 \leftarrow \emptyset, \mathcal{I}_2 \leftarrow \emptyset$ ;
  /*  $\mathcal{I}_1$  records all closed subintervals of  $[a, b]$  in which  $f(t)$  has no
real roots, while  $\mathcal{I}_2$  records all closed subintervals of  $[a, b]$  in
which  $f'(t)$  has no real roots. */
3 Construct a sequence,  $S_0(t) = f(t), S_1(t), S_2(t), \dots, S_r(t)$ , where  $S_i$  is a
PEF, which shares the common real roots with the derivative of  $S_{i-1}$ ,
 $i = 1, 2, \dots, r, r \in \mathbb{N}$ , and  $S_r(t)$  is a polynomial;
4  $\mathcal{L}_{(a,b)}(S_r) := \text{UPIsolving}(S_r(t))$ , a real root isolation of  $S_r(t)$ ;
5 for  $i = r - 1; i \geq 0; i --$  do
6    $[\mathcal{I}_1, \mathcal{I}_2, \mathcal{L}] \leftarrow \text{PEFI}(S_0, S_1, S_i, S_{i+1}, (a, b), \mathcal{I}_1, \mathcal{I}_2, \mathcal{L})$ ;
7   for  $[c, d] \in \mathcal{I}_2$  do
8     if  $S_0(c)S_0(d) < 0$  then
9        $\mathcal{L} \leftarrow \mathcal{L} \cup \{(c, d)\}$ ;
10 return  $\mathcal{L}$ ;

```

---

*Proof.* Termination is immediately obtained from Theorem 4. Then we prove its correctness. After the *for loop* in line 2,  $\mathcal{L}$  is a real root isolation of  $S_0(t) = 0$  ( i.e.,  $f(t) = 0$ ) on  $(a, b) \setminus \cup(\mathcal{I}_1 \cup \mathcal{I}_2)$ . Because  $f'(t)$  has a constant nonzero sign in each interval of  $\mathcal{I}_2$ ,  $f(t)$  has at most one real root in each interval of  $\mathcal{I}_2$  and this can be decided by checking the signs of  $f(t)$  at two endpoints of the interval. Moreover, since there is no real root of  $f(t) = 0$  in  $\cup\mathcal{I}_1$ , so  $\mathcal{L}_2$  is a real root isolation of  $S_0(t)$  in  $(a, b)$ . □

### 5.3 Complexity Analysis of PEFIsolation

Here we give a rough complexity analysis of PEFIsolation. Suppose  $f(t) = f_0(t) + f_1(t)e^{\nu_1 t} + \dots + f_s(t)e^{\nu_s t}$ ,  $L(f)$  and  $U(f)$  are respectively a lower bound and an upper bound on real roots of  $f(t)$ ,  $\text{deg}(f) = (d_0, d_1, \dots, d_s)$ . PEFIsolation computes all real roots for a PEF chain  $f(t) = 0, f'(t) = 0, f''(t) = 0, \dots$ , totally,  $d_0 + \dots + d_{s-1} + s + 1$  such PEFs at most, with the corresponding degree. The last element in the chain is a polynomial with degree  $d_s$ , so it has at most  $d_s$  real roots. Clearly, for each function in the chain, the number of intervals in its real root isolation is at most  $d_0 + d_1 + \dots + d_s + s + 1$ . In addition, suppose the lower bound on the distances between real roots of  $S_i$  and those of  $S_{i+1}$  is  $\delta$ , then the *while loop* (line 3-13) in Algorithm 2 always terminates after the length of an interval is less than  $\delta$ . Since the length of every interval is less than or equal to  $U(f) - L(f)$ , the *while loop* must terminate in  $\log_2 \frac{U(f) - L(f)}{\delta}$  steps. In a summary, the complexity of PEFIsolation is about  $\mathcal{O}((\sum_{i=0}^s d_i + s + 1)^2 \log_2 \frac{U(f) - L(f)}{\delta})$ .

---

**Algorithm 2.** PEFI

---

**Input:** (1) PEFs  $f_1(t), f_2(t), g_1(t), g_2(t)$  s.t.  $f_2(t)$  and  $f'_1(t)$  share same real zeros,  $g_2(t)$  and  $g'_1(t)$  share same real zeros, and  $f_1(t)$  and  $f_2(t)$  have no common real zeros;

- (2) an open interval  $(a, b)$ ;
- (3)  $\mathcal{I}_1, \mathcal{I}_2$ , two sets of closed intervals contained in  $(a, b)$ , s.t.  $f_1(t)$  has no real zeros in  $\cup \mathcal{I}_1$ ,  $f_2(t)$  has no real zeros in  $\cup \mathcal{I}_2$ ,  $\cup \mathcal{I}_1 \cap \cup \mathcal{I}_2 = \emptyset$  ;
- (4)  $\mathcal{L}$ , a real root isolation of  $g_2(t)$  on  $(a, b) \setminus \cup(\mathcal{I}_1 \cup \mathcal{I}_2)$ .

**Output:** (1)  $\mathcal{I}'_1$  and  $\mathcal{I}'_2$  with the same properties as  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , respectively;

(2)  $\mathcal{L}_2$ , a real root isolation of  $g_1(t)$  on  $(a, b) \setminus \cup(\mathcal{I}'_1 \cup \mathcal{I}'_2)$ .

```

1   $\mathcal{I}'_1 \leftarrow \mathcal{I}_1, \mathcal{I}'_2 \leftarrow \mathcal{I}_2$ ;
2  for  $(l, u)$  in  $\mathcal{L}$  do
3      while  $0 \in g_1([l, u])$  do
4          if  $0 \notin f_1([l, u])$  then
5               $\mathcal{I}'_1 \leftarrow \mathcal{I}'_1 \cup \{[l, u]\}$ ; break;
6          if  $0 \notin f_2([l, u])$  and  $f_1(l)f_1(u) \neq 0$  then
7               $\mathcal{I}'_2 \leftarrow \mathcal{I}'_2 \cup \{[l, u]\}$ ; break;
8          if  $g_2(l)g_2(\frac{l+u}{2}) < 0$  then
9               $u \leftarrow \frac{l+u}{2}$ ;
10         else if  $g_2(\frac{l+u}{2}) = 0$  then
11              $l \leftarrow \frac{3l+u}{4}; u \leftarrow \frac{l+3u}{4}$ ;
12         else
13              $l \leftarrow \frac{l+u}{2}$ ;
14  $\mathcal{L}_1 \leftarrow \mathcal{L}; \mathcal{L}_2 \leftarrow \emptyset; \mathcal{L}_3 \leftarrow \{(a_1, b_1), \dots, (a_m, b_m)\}$ ;
    /* where  $a_1, b_1, \dots, a_m, b_m$  are the endpoints of the intervals in  $\mathcal{I}_1$ 
    and  $\mathcal{I}_2$  s.t.  $a \leq a_1 < b_1 < \dots < a_m < b_m \leq b$ ,  $(a_i, b_i) \subseteq (a, b) \setminus \cup(\mathcal{I}_1 \cup \mathcal{I}_2)$ 
    for  $i = 1, \dots, m$ , and  $\cup_{i=1}^m (a_i, b_i) = (a, b) \setminus \cup(\mathcal{I}_1 \cup \mathcal{I}_2)$ . */
15 for  $(c, d)$  in  $\mathcal{L}_3$  do
16      $\mathcal{L}_{1(c,d)} \leftarrow \{I \mid I \in \mathcal{L}_1 \text{ and } I \subset (c, d)\}$ ;
17     Obtain a real root isolation  $\mathcal{L}_{(c,d)}$  for  $g_1(t)$  on  $(c, d)$  from  $\mathcal{L}_{1(c,d)}$  by
    Theorem 3;  $\mathcal{L}_2 \leftarrow \mathcal{L}_2 \cup \mathcal{L}_{(c,d)}$ ;
18 return  $\mathcal{I}'_1, \mathcal{I}'_2, \mathcal{L}_2$ ;

```

---

## 6 Implementation and Experimental Results

We have implemented the proposed approach in *Mathematica* as a prototype, called *LinR*<sup>1</sup>, which takes a specific LDS reachability problem as input, and gives either *False* if the problem is not satisfiable, or *True* otherwise associated with some valid sample points.

*Remark 2.* When we implement the above algorithms, some optimizing strategies are adopted for improving efficiency. For example, if the input function can

---

<sup>1</sup> Both the tool and the case studies in this section can be found at <http://lcs.ios.ac.cn/~chenms/tools/LinR.tar.bz2>.

be factorized, then we isolate the real roots of each factor rather than the input function itself, and then refine the resulted intervals if necessary. We omit the implementation details here.

In the following, we report some experimental results with *LimR*.

*Example 3.* Let us continue Example 1 given in the introduction first.

Obviously,  $X \cap Y = \emptyset$ , and  $\xi(t) = \begin{bmatrix} x_1 e^{\sqrt{2}t} + \frac{\sqrt{2}t - \sqrt{2} + 1}{2} + \frac{\sqrt{2} - 1}{2} e^{\sqrt{2}t} \\ x_2 e^{-\sqrt{2}t} + \frac{(1 + \sqrt{2})t - 1}{3 + 2\sqrt{2}} e^t + \frac{e^{-\sqrt{2}t}}{3 + 2\sqrt{2}} \\ x_3 e^{-t} + t e^{-t} \end{bmatrix}$  is the solution of the LDS. Thus, the reachability problem becomes

$$\mathcal{F} = \exists x_1 \exists x_2 \exists x_3 \exists t. \Phi(x_1, x_2, x_3, t);$$

$$\Phi(x_1, x_2, x_3, t) = 1 - x_1^2 - x_2^2 - x_3^2 > 0 \wedge x_1 e^{\sqrt{2}t} + x_2 e^{-\sqrt{2}t} + x_3 e^{-t} + h(t) < 0 \wedge t > 0,$$

$$\text{where } h(t) = \frac{e^{-\sqrt{2}t}}{3 + 2\sqrt{2}} + t e^{-t} + \frac{\sqrt{2}t - \sqrt{2} + 5}{2} + \frac{(1 + \sqrt{2})t - 1}{3 + 2\sqrt{2}} e^t + \frac{\sqrt{2} - 1}{2} e^{\sqrt{2}t}.$$

Then, using *Brown's projection operator* [11] to eliminate  $x_1, x_2, x_3$  successively (**Step 3** in Sect. 4), we have

$$q_3(x_1, x_2, x_3, t) = (x_1^2 + x_2^2 + x_3^2 - 1)(ax_1 + bx_2 + cx_3 + h)$$

$$q_2(x_2, x_3, t) = a(x_2^2 + x_3^2 - 1) \\ (-a^2 + a^2 x_2^2 + a^2 x_3^2 + b^2 x_2^2 + 2bcx_2 x_3 + 2bhx_2 + c^2 x_3^2 + 2chx_3 + h^2),$$

$$q_1(x_3, t) = a(x_3 - 1)(x_3 + 1)(a^2 + b^2)(2chx_3 + h^2 - b^2 + b^2 x_3^2 + c^2 x_3^2) \\ (-a^2 + a^2 x_3^2 + 2chx_3 + h^2 - b^2 + b^2 x_3^2 + c^2 x_3^2),$$

$$q_0(t) = ab(c - h)(c + h)(a^2 + b^2)(b^2 + c^2)(b^2 + c^2 - h^2)(a^2 + b^2 + c^2) \\ (a^2 + b^2 + c^2 - h^2),$$

where  $a = e^{\sqrt{2}t}$ ,  $b = e^{-\sqrt{2}t}$  and  $c = e^{-t}$ .

Isolate all real roots of  $q_0(t) = 0$  in  $(0, +\infty)$  (as we only care  $t > 0$ ) (**Step 4** in Sect. 4), and obtain  $\mathcal{L}(q_0) = \{(1.08, 1.29)\}$ .

Lift the real root isolation in the order  $t, x_3, x_2, x_1$  successively using the *openCAD* lifting procedure (**Step 5** in Sect. 4), finally, we obtain 48 sample points,  $\{-0.835, -0.212, 0.184, 2.\}$  satisfies  $\Phi$ , which implies that the safety property is not satisfied with the counter example starting from  $(-0.835, -0.212, 0.184, 2) \in X$ , and ending at time  $t = 2$ .

*Example 4 (Adapted from [1]).* Consider a vessel of water containing a radioactive isotope, to be used as a tracer for the food chain, which consists of aquatic plankton varieties phytoplankton  $A$  and zooplankton  $B$ . Let  $\xi_1(t)$  be the isotope concentration in the water,  $\xi_2(t)$  the isotope concentration in  $A$  and  $\xi_3(t)$  the isotope concentration in  $B$ . The dynamics of the vessel is modelled as

$$\dot{\xi} = A\xi, \text{ where } A = \begin{bmatrix} -3 & 6 & 5 \\ 2 & -12 & 0 \\ 1 & 6 & -5 \end{bmatrix}. \text{ The initial radioactive isotope concentrations}$$

$$\xi_1(0) = x_1 > 0, \xi_2(0) = 0, \xi_3(0) = 0.$$

The safety property of our concern is whether  $\forall t > 0 \xi_1(t) \geq \xi_2(t) + \xi_3(t)$ . To this end, we consider a more general problem: For which  $n_1, n_2 \in \mathbb{N}$  s.t.  $\mathcal{F}(n_1, n_2) = \exists x_1 > 0 \exists t > 0 \xi_1(t) < n_1 \xi_2(t) + n_2 \xi_3(t)$  holds.

It is easy to see that the matrix  $A$  is diagonalizable with eigenvalues  $0, -10 + \sqrt{6}, -10 - \sqrt{6}$ . When  $(n_1, n_2) = (1, 1)$ , using the method in Sect. 4, we obtain two sample points for  $(x_1, t)$ , i.e.,  $(-0.1, 1), (0.1, 1)$ . But none of them satisfies  $\mathcal{F}(1, 1)$ , which simply implies the safety property holds. When  $(n_1, n_2) = (2, 2)$ , similarly, we obtain four sample points for  $(x_1, t)$ , i.e.,  $(-0.1, 0), (0.1, 0), (-0.1, 1), (0.1, 1)$ , in which  $(0.1, 1)$  satisfies  $\mathcal{F}(2, 2)$ . It can be proved that  $\xi_i(t) \geq 0$  for any  $t > 0$  and  $i = 1, 2, 3$ . So, it is clear that, if  $\mathcal{F}(n_1, n_2)$  holds,  $\mathcal{F}(m_1, m_2)$  holds for  $m_1 \geq n_1$  and  $m_2 \geq n_2$ . Then, by checking some pairs of  $(n_1, n_2) \in \mathbb{N} \times \mathbb{N}$  in a similar way as above, we conclude that all pairs  $(n_1, n_2) \in \mathbb{N} \times \mathbb{N}$  satisfy  $\mathcal{F}(n_1, n_2)$ , except for the pairs  $\{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (3, 0), (3, 1), (4, 0), (5, 0)\}$ .

*Example 5 (Adapted from [1]).* Consider a typical home with attic, basement and insulated main floor. Let  $x_3(t), x_2(t), x_1(t)$  be the temperature in the attic, main living area and basement respectively, and  $t$  is the time in hours. Assume it is winter time, the outside temperature is nearly  $35^\circ F$ , and the basement earth temperature is nearly  $45^\circ F$ . Suppose a small electric heater is turned on, and it provides a  $20^\circ F$  rise per hour. We want to verify that the temperature in main living area will never reach too high (maybe  $70^\circ F$ ). Analyze the changing temperatures in the three levels using Newton’s cooling law and given the value of the cooling constants, we obtain the model as follows:

$$\begin{aligned} \dot{x}_1 &= \frac{1}{2}(45 - x_1) + \frac{1}{2}(x_2 - x_1), \dot{x}_2 = \frac{1}{2}(x_1 - x_2) + \frac{1}{4}(35 - x_2) + \frac{1}{4}(x_3 - x_2) + 20, \\ \dot{x}_3 &= \frac{1}{4}(x_2 - x_3) + \frac{3}{4}(35 - x_3), \end{aligned}$$

with the initial set  $X = \{(x_1, x_2, x_3)^T \mid 1 - (x_1 - 45)^2 - (x_2 - 35)^2 - (x_3 - 35)^2 > 0\}$  and the unsafe set  $Y = \{(y_1, y_2, y_3)^T \mid y_2 - 70 > 0\}$ . The safety property we are concerning is to check if some state in  $Y$  is reachable from  $X$ , which holds by using *LinR*.

The above three examples are verified by *LinR*. Both the time and memory costs on a 64-bit Linux computer with a 2.93GHz Intel Core-i7 processor and 4GB of RAM are shown in Table 1. Besides, we have also compared on the same platform with the performances of Strzebonski’s approach (i.e., CT1D) [22], as well as verification tools dReach [14], HSolver [18], and Flow\* [6] on these examples. Note that, both dReach and Flow\* cannot handle unbounded model checking, and even for BMC, they are less efficient than our tool in many cases (see Example 3 and Example 4)<sup>2</sup>. In particular, Flow\* accepts only rectangular initial set, i.e. each variable needs to be specified within a closed interval and polynomial constraints are not allowed, and thus we tried different cube to approximate the spherical initial set in Example 3, while none of them can derive a desired result (“unsafe”). As for HSolver, due to the rejection of “sqrt”, we simplify the original model by replacing all the irrational numbers with their approximate decimals, however, 2 of the 3 examples still can not be answered by HSolver in reasonable time and memory.

<sup>2</sup> Here, we set the time bounds 2s, 2s, and 5s resp. for Examples 3, 4, and 5 when using dReach and Flow\*.

**Table 1.** Evaluation results of different methods

LDS	Time (sec)					Memory (kb)				
	<i>LinR</i>	<i>CT1D</i>	<i>dReach</i>	<i>HSolver</i>	<i>Flow*</i>	<i>LinR</i>	<i>CT1D</i>	<i>dReach</i>	<i>HSolver</i>	<i>Flow*</i>
Example 3	1.35	×	37.36	–	–	112	×	3812	–	–
Example 4	0.03	0.20	0.71	–	–	131	2018	3816	–	–
Example 5	1.68	×	0.05	0.72	16.50	166	×	3812	1076932	113492

× : the verification fails by non-termination within reasonable amount of time (10 hours)

– : the verification fails because of giving an answer as “safety unknown”

**Table 2.** Time consumption (in milliseconds) on Example 3.4 from [16]

LinR	CT1D	QEPCAD	dReach	HSolver	Flow*
39	33	57	110	–	–

*Remark 3* In the above examples, all constraints are open sets. Actually, more general initial and unsafe sets, i.e., when either  $Pre(X)$  or  $Post(X)$  is not *open semi-algebraic*, can be coped with in our approach also, as we have implemented CAD in the algorithms. For the Example 3.4 in [16], where  $A$  is diagonalizable with rational eigenvalues and  $Pre(X)$  and  $Post(X)$  are both closed sets, it takes 57 milliseconds using Lafferriere et al’s approach based on quantifier elimination by QEPCAD. In contrast, LinR takes 39 milliseconds, and CT1D takes 33 milliseconds. In brief, our approach shares nearly same complexity as Stzebonski’s in general case, but is still better than other approaches, see Table 2 (QEPCAD stands for Lafferriere et al’s approach).

## 7 Conclusion

In this paper, we proved the decidability of a family of vector fields whose state parts are linear, while input parts are non-linear, (possibly) with exponential expressions. Such vector fields are commonly used in practice. To the best of our knowledge, this family is one of the most expressive families of vector fields with a decidable reachability problem. The decidability is achieved by reduction to the decidability of the extension of Tarski’s algebra with some specific exponential functions. Experimental results indicate our algorithm is more efficient than Strzebonski’s if all constraints are open sets, better than existing approaches based on approximation and numeric computation in general, e.g., HSolver [18], dReach [14], FLOW\* [6], etc.

In addition, similar as in [22], our decision procedure can be extended without any substantial change to deal with logarithm functions as well as the composition of logarithmic and exponential functions.

**Acknowledgements.** The first, third and fourth authors are supported partly by NSFC under grants 11290141 and 11271034; the second and fifth authors are supported partly by “973 Program” under grant No. 2014CB340701, by NSFC under grants



91118007 and 91418204, by CDZ project CAP (GZ 1023), and by the CAS/SAFEA International Partnership Program for Creative Research Teams.

## References

1. <http://www.math.utah.edu/~gustafso/s2013/2250/systemsExamplesTheory2008.pdf>
2. Achatz, M., McCallum, S., Weispfenning, V.: Deciding polynomial-exponential problems. In: ISSAC 2008 (2008)
3. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. *Theoret. Comput. Sci.* **138**(1), 3–34 (1995)
4. Alur, R., Dill, D.L.: A theory of timed automata. *Theoret. Comput. Sci.* **126**(2), 183–235 (1994)
5. Brown, C.W.: Improved projection for cylindrical algebraic decomposition. *J. Symb. Comput.* **32**(5), 447–465 (2001)
6. Chen, X., Abraham, E., Sankaranarayanan, S.: Flow\*: an analyzer for non-linear hybrid systems. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 258–263. Springer, Heidelberg (2013)
7. Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Brakhage, H. (ed.) Automata Theory and Formal Languages 2nd GI Conference. LNCS, vol. 33, pp. 134–183. Springer, Heidelberg (1975)
8. Collins, G.E., Hong, H.: Partial cylindrical algebraic decomposition for quantifier elimination. *J. Symb. Comput.* **12**(3), 299–328 (1991)
9. Collins, G.E., Loos, R.: Real zeros of polynomials. In: Computer Algebra - Symbolic and Algebraic Computation, pp. 83–94. Springer (1982)
10. Davenport, J.H., Heintz, J.: Real quantifier elimination is doubly exponential. *J. Symb. Comput.* **5**(1), 29–35 (1988)
11. Han, J., Dai, L., Xia, B.: Constructing fewer open cells by gcd computation in cad projection. In: ISSAC 2014, pp. 240–247. ACM (2014)
12. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What’s decidable about hybrid automata? *J. Comput. Syst. Sci.* **57**(1), 94–124 (1998)
13. Hong, H.: An improvement of the projection operator in cylindrical algebraic decomposition. In: ISSAC 1990, pp. 261–264. ACM (1990)
14. Kong, S., Gao, S., Chen, W., Clarke, E.: dreach: Delta-reachability analysis for hybrid systems. In: TACAS 2015 (2015)
15. Lafferriere, G., Pappas, G.J., Sastry, S.: O-minimal hybrid systems. *MCSS* **13**(1), 1–21 (2000)
16. Lafferriere, G., Pappas, G.J., Yovine, S.: Symbolic reachability computation for families of linear vector fields. *J. Symb. Comput.* **32**, 231–253 (2001)
17. McCallum, S.: An improved projection operation for cylindrical algebraic decomposition of three-dimensional space. *J. Symb. Comput.* **5**(1), 141–161 (1988)
18. Ratschan, S., She, Z.: Safety verification of hybrid systems by constraint propagation based abstraction refinement. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 573–589. Springer, Heidelberg (2005)
19. Richardson, D.: How to recognize zero. *J. Symb. Comput.* **24**, 627–645 (1997)
20. Strzeboński, A.: Solving systems of strict polynomial inequalities. *J. Symb. Comput.* **29**(3), 471–480 (2000)
21. Strzeboński, A.: Real root isolation for exp-log functions. In: ISSAC 2008, pp. 303–314 (2008)

22. Strzeboński, A.: Cylindrical decomposition for systems transcendental in the first variable. *J. Symb. Comput.* **46**, 1284–1290 (2011)
23. Tarski, A.: *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkeley (1951)
24. Wing, J.: How can we provide people with cyber-physical systems they can bet their lives on? *Computing Research News*, 20(1) (2008)
25. Xu, M., Li, Z.-B., Yang, L.: Quantifier elimination for a class of exponential polynomial formulas. *J. Symb. Comput.* (2015). doi:[10.1016/j.jsc.2014.09.015](https://doi.org/10.1016/j.jsc.2014.09.015)