Some decidable properties of linear dynamical systems

24.1.2022, One World Combinatorics on Words Seminar

Markus A. Whiteland[1]

Joint work with C. Baier, F. Funke, S. Jantsch, T. Karimov, E. Lefaucheux, F. Luca, J. Ouaknine, D. Purser, A. Varonka, J. Worrell

TECHNISCHE UNIVERSITÄT DRESDEN

MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

UNIVERSITY OF OXFORD

[1]Department of Mathematics, Université de Liège, Belgium

```
do {
    p = r = b + (2 * PTHRESH);
    if (r >= t) p = r = t;      /* too short to care about */
    else {
        while (((cmp(aTHX_ *(p-1), *p) > 0) == sense) &&
               ((p -= 2) > q)) {}
        if (p <= q) {
            /* b through r is a (long) run.
            ** Extend it as far as possible. */
            p = q = r;
            while (((p += 2) < t) &&
                   ((cmp(aTHX_ *(p-1), *p) > 0) == sense)) q = p;
            r = p = q + 2;      /* no simple pairs, no after-run */
        }
    }
    if (q > b) {                /* run of greater than 2 at b */
        gptr *savep = p;
        p = q += 2;
        /* pick up singleton, if possible */
        if ((p == t) &&
            ((t + 1) == last) &&
            ((cmp(aTHX_ *(p-1), *p) > 0) == sense))
            savep = r = p = q = last;
        p2 = NEXT(p2) = p2 + (p - b); ++runs;
        if (sense)
            while (b < --p) {
                const gptr c = *b;
                *b++ = *p;
                *p = c;
            }
        p = savep;
    }
    while (q < p) {             /* simple pairs */
        p2 = NEXT(p2) = p2 + 2; ++runs;
        const gptr c = *q++;
        *(q-1) = *q;
        *q++ = c;
        q += 2;
    }
    if (((b = p) == t) && ((t+1) == last)) {
        NEXT(p2) = p2 + 1; ++runs;
        b++;
    }
    q = r;
} while (b < t);
sense = !sense;
}
return runs;
```

Modeling software systems

Does this program work correctly?

Modeling software systems

```
    do {
        p = r = b + (2 * PTHRESH);
        if (r >= t) p = r = t;       /* too short to care about */
        else {
            while (((cmp(aTHX_ *(p-1), *p) > 0) == sense) &&
                   ((p -= 2) > q)) {}
            if (p <= q) {
                /* b through r is a (long) run.
                ** Extend it as far as possible. */
                p = q = r;
                while (((p += 2) < t) &&
                       ((cmp(aTHX_ *(p-1), *p) > 0) == sense)) q = p;
                r = p = q + 2;       /* no simple pairs, no after-run */
            }
        }
        if (q > b) {                 /* run of greater than 2 at b */
            gptr *savep = p;
            p = q += 2;
            /* pick up singleton, if possible */
            if ((p == t) &&
                ((t + 1) == last) &&
                ((cmp(aTHX_ *(p-1), *p) > 0) == sense))
                savep = r = p = q = last;
            p2 = NEXT(p2) = p2 + (p - b); ++runs;
            if (sense)
                while (b < --p) {
                    const gptr c = *b;
                    *b++ = *p;
                    *p = c;
                }
            p = savep;
        }
        while (q < p) {              /* simple pairs */
            p2 = NEXT(p2) = p2 + 2; ++runs;
            const gptr c = *q++;
            *(q-1) = *q;
            *q++ = c;
            q += 2;
        }
        if (((b = p) == t) && ((t+1) == last)) {
            NEXT(p2) = p2 + 1; ++runs;
            b++;
        }
        q = r;
    } while (b < t);
    sense = !sense;
}
return runs;
```

Does this program work correctly?

- Decomposing into simple blocks
- Modeling blocks separately
- Correctness of each part ensures correctness of the program

```c
do {
    p = r = b + (2 * PTHRESH);
    if (r >= t) p = r = t;        /* too short to care about */
    else {
        while (((cmp(aTHX_ *(p-1), *p) > 0) == sense) &&
               ((p -= 2) > q)) {}
        if (p <= q) {
            /* b through r is a (long) run.
            ** Extend it as far as possible. */
            p = q = r;
            while (((p += 2) < t) &&
                   ((cmp(aTHX_ *(p-1), *p) > 0) == sense)) q = p;
            r = p = q + 2;        /* no simple pairs, no after-run */
        }
    }
    if (q > b) {                  /* run of greater than 2 at b */
        gptr *savep = p;
        p = q += 2;
        /* pick up singleton, if possible */
        if ((p == t) &&
            ((t + 1) == last) &&
            ((cmp(aTHX_ *(p-1), *p) > 0) == sense))
            savep = r = p = q = last;
        p2 = NEXT(p2) = p2 + (p - b); ++runs;
        if (sense)
            while (b < --p) {
                const gptr c = *b;
                *b++ = *p;
                *p = c;
            }
        p = savep;
    }
    while (q < p) {               /* simple pairs */
        p2 = NEXT(p2) = p2 + 2; ++runs;
        const gptr c = *q++;
        *(q-1) = *q;
        *q++ = c;
        q += 2;
    }
    if (((b = p) == t) && ((t+1) == last)) {
        NEXT(p2) = p2 + 1; ++runs;
        b++;
    }
    q = r;
} while (b < t);
sense = !sense;
}
return runs;
```

Modeling software systems

Does this program work correctly?

- Decomposing into simple blocks
- Modeling blocks separately
- Correctness of each part ensures correctness of the program

Does this while loop terminate/halt?

Loops as Dynamical Systems

Algorithm

$x := 1$
$y := 2$
$z := 3$
while $(x, y, z) \neq (2, 1, 0)$ **do**
  $x' := 2x + y$
  $y' := x - y + 3z$
  $z' := 2x - y + 2z$
  $x := x'$
  $y := y'$
  $z := z'$
end while

Loops as Dynamical Systems

Algorithm

$x := 1$
$y := 2$
$z := 3$
while $(x, y, z) \neq (2, 1, 0)$ **do**
  $x' := 2x + y$
  $y' := x - y + 3z$
  $z' := 2x - y + 2z$
  $x := x'$
  $y := y'$
  $z := z'$
end while

$$\vec{x}_0 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix},$$

Algorithm

$x := 1$
$y := 2$
$z := 3$
while $(x, y, z) \neq (2, 1, 0)$ **do**
  $x' := 2x + y$
  $y' := x - y + 3z$
  $z' := 2x - y + 2z$
  $x := x'$
  $y := y'$
  $z := z'$
end while

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} := \begin{pmatrix} 2 & 1 & 0 \\ 1 & -1 & 3 \\ 2 & -1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\vec{x}_0 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad M = \begin{pmatrix} 2 & 1 & 0 \\ 1 & -1 & 3 \\ 2 & -1 & 2 \end{pmatrix}$$

Loops as Dynamical Systems

Algorithm

$x := 1$
$y := 2$
$z := 3$
while $(x, y, z) \neq (2, 1, 0)$ **do**
  $x' := 2x + y$
  $y' := x - y + 3z$
  $z' := 2x - y + 2z$
  $x := x'$
  $y := y'$
  $z := z'$
end while

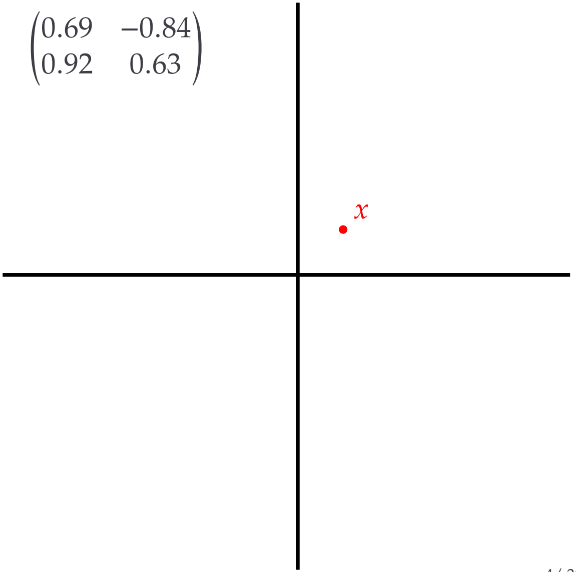
$$\vec{x}_0 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad M = \begin{pmatrix} 2 & 1 & 0 \\ 1 & -1 & 3 \\ 2 & -1 & 2 \end{pmatrix}$$

$$\text{Loop halts} \iff \exists n : M^n \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}.$$

Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d\times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$

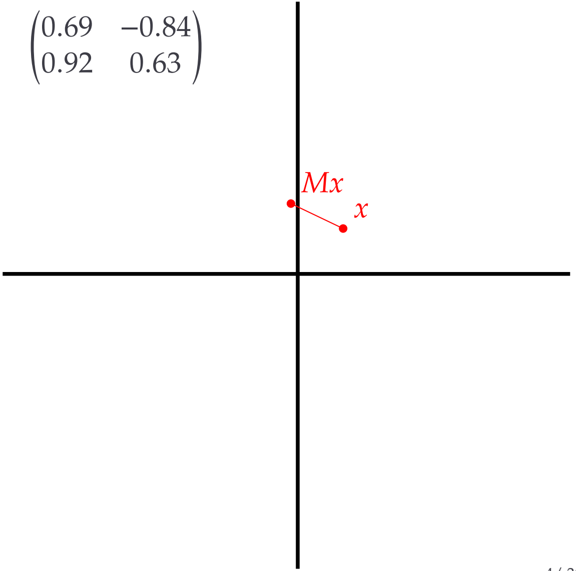
$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$

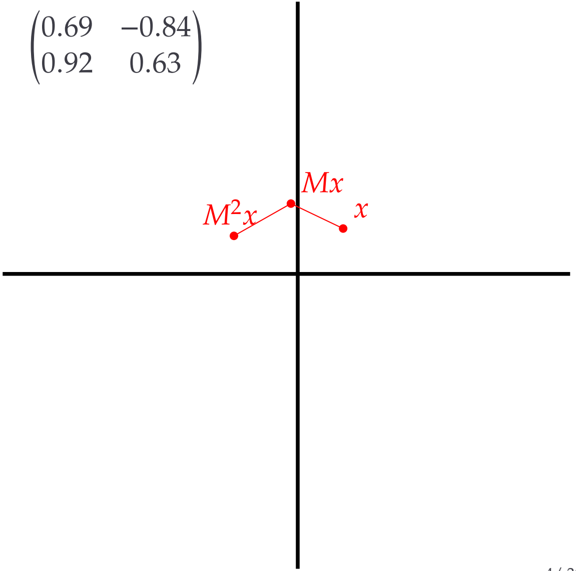
$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$

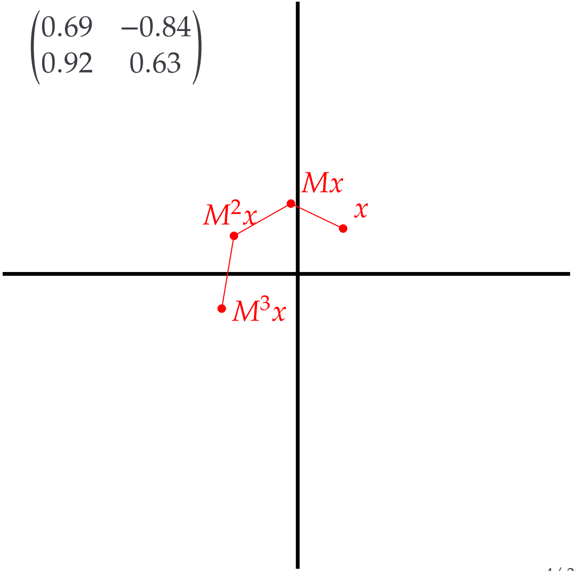
$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$

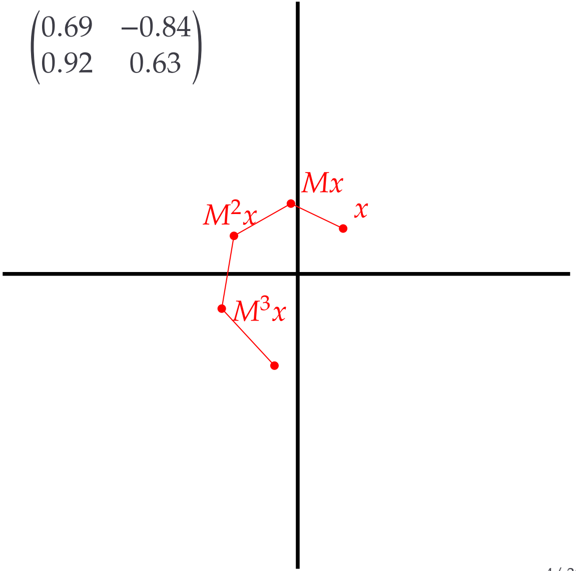
$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$

$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



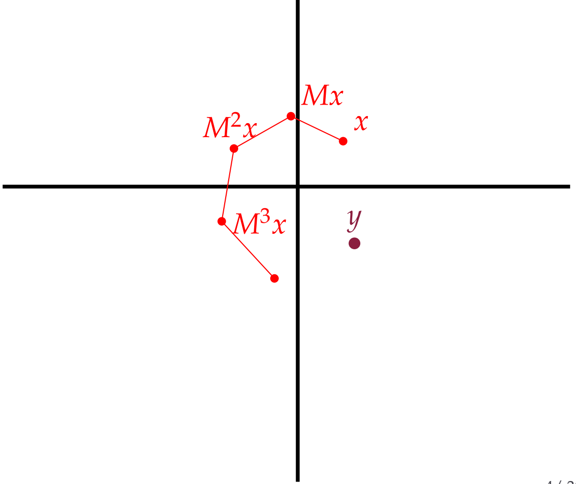
Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$
- A target point $y \in \mathbb{R}^d$.

Orbit Problem

Does the orbit contain the target?
$\exists n : M^n x = y$?

$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



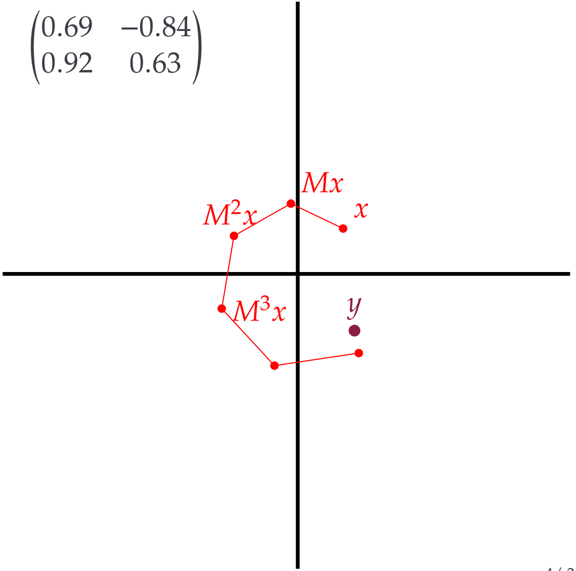
Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$
- A target point $y \in \mathbb{R}^d$.

Orbit Problem

Does the orbit contain the target? $\exists n : M^n x = y$?

$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



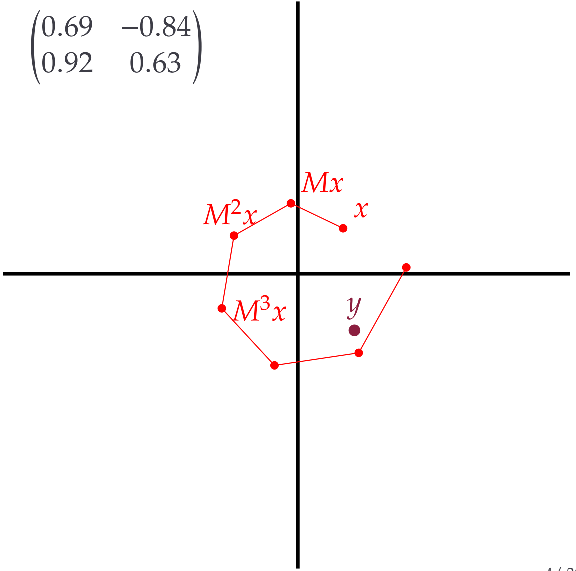
Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$
- A target point $y \in \mathbb{R}^d$.

Orbit Problem

Does the orbit contain the target? $\exists n : M^n x = y$?

$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



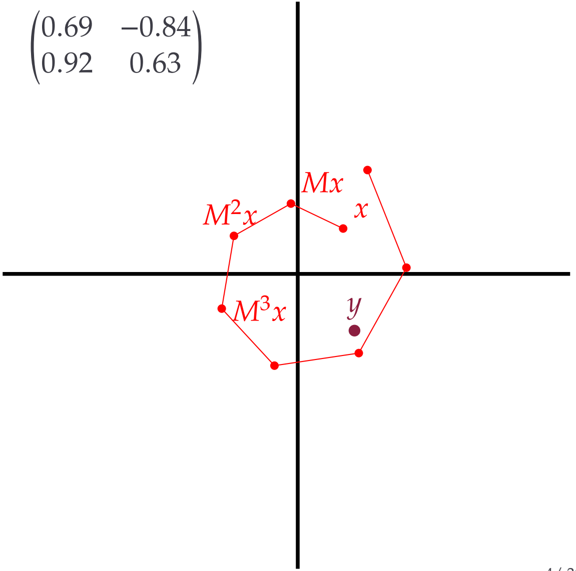
Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$
- A target point $y \in \mathbb{R}^d$.

Orbit Problem

Does the orbit contain the target? $\exists n : M^n x = y$?

$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$

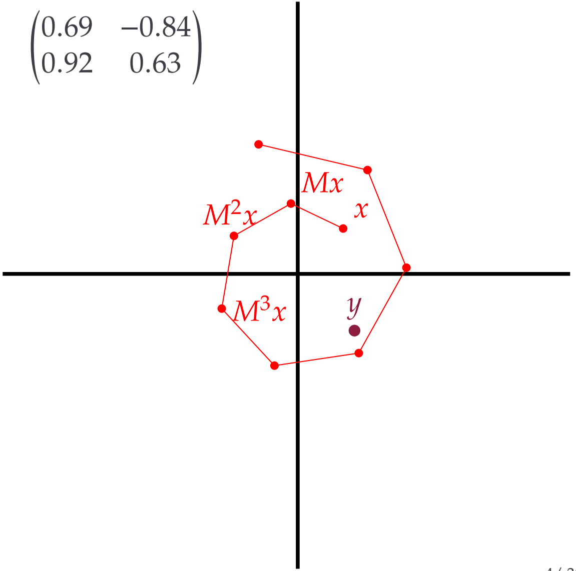


Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$
- A target point $y \in \mathbb{R}^d$.

Orbit Problem

Does the orbit contain the target? $\exists n : M^n x = y$?

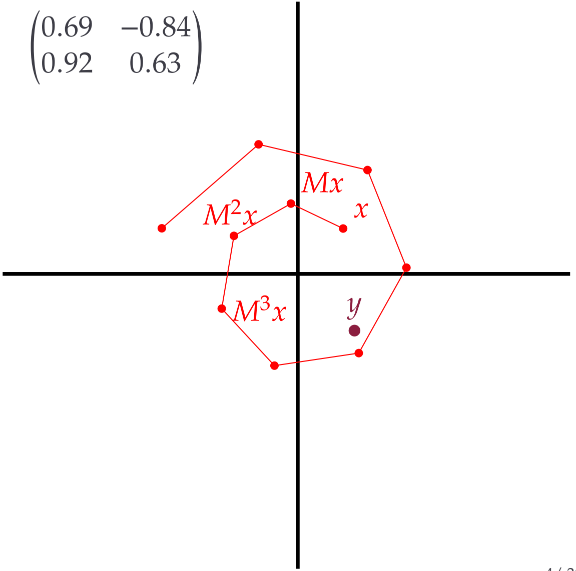


- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$
- A target point $y \in \mathbb{R}^d$.

Orbit Problem

Does the orbit contain the target?
$\exists n : M^n x = y$?

$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



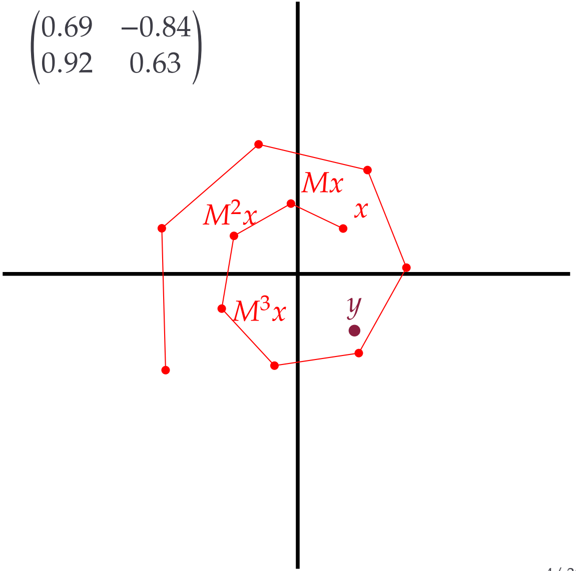
Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \ldots \rangle$
- A target point $y \in \mathbb{R}^d$.

Orbit Problem

Does the orbit contain the target?
$\exists n : M^n x = y$?

$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



Linear Dynamical Systems (LDS)

- Given a starting point $x \in \mathbb{R}^d$.
- A linear update map $M \in \mathbb{R}^{d \times d}$
- Defines an orbit $\langle x, Mx, M^2x, M^3x, \dots \rangle$
- A target point $y \in \mathbb{R}^d$.

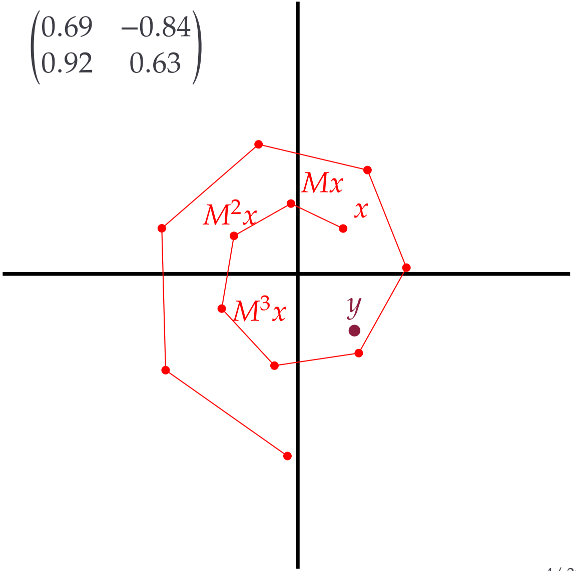
Orbit Problem

Does the orbit contain the target? $\exists n : M^n x = y$?

Theorem (Kannan–Lipton'86)

Orbit Problem is decidable for x, y, M with ***algebraic entries****.*
In $\mathtt{PTIME}$ for x, y, M over $\mathbb{Q}$.

$$\begin{pmatrix} 0.69 & -0.84 \\ 0.92 & 0.63 \end{pmatrix}$$



Reachability in LDS

Halting of Linear Dynamical Systems

Skolem's problem

Given x, u, M, decide

$$\exists n \in \mathbb{N} \text{ such that } u^\top M^n x = 0?$$

"Reaching a hyperplane."

- "Open" since the 1930s
- Decidable for instances with dimension ≤ 4.
  - Real algebraic entries.
- Open for systems with dimension ≥ 5
- Equivalent to zeros of linear recurrence sequences.

Halting of Linear Dynamical Systems

Skolem's problem

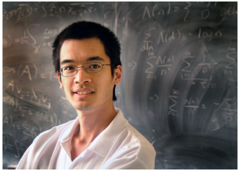
Given x, u, M, decide

$\exists n \in \mathbb{N}$ such that $u^\top M^n x = 0$?

"A mathematical embarrassment ..."

Richard Lipton

"Reaching a hyperplane."



"It is faintly outrageous that this problem is still open; it is saying that we do not know how to decide the Halting Problem even for 'linear' automata!"

Terence Tao

So what can we do?

Reachability problems

Given $x \in \mathbb{Q}^d$, $M \in \mathbb{Q}^{d \times d}$, and a set $T \subseteq \mathbb{R}^d$, decide: $\exists n \in \mathbb{N}$ such that $M^n x \in T$?

Linear loops: T encodes the **halting condition**.

So what can we do?

Reachability problems

Given $x \in \mathbb{Q}^d$, $M \in \mathbb{Q}^{d \times d}$, and a set $T \subseteq \mathbb{R}^d$, decide: $\exists n \in \mathbb{N}$ such that $M^n x \in T$?

Linear loops: T encodes the **halting condition**.

Reasonable targets T: **semialgebraic sets**.

So what can we do?

Reachability problems

Given $x \in \mathbb{Q}^d$, $M \in \mathbb{Q}^{d\times d}$, and a set $T \subseteq \mathbb{R}^d$, decide: $\exists n \in \mathbb{N}$ such that $M^n x \in T$?

Linear loops: T encodes the **halting condition**.

Reasonable targets T: **semialgebraic sets**.

Example (Skolem's problem; Not known to be decidable)

T is a hyperplane defined by $u \in \mathbb{Q}^d$: $\{z \in \mathbb{R}^d : (z, u) = 0\}$

So what can we do?

Reachability problems

Given $x \in \mathbb{Q}^d$, $M \in \mathbb{Q}^{d \times d}$, and a set $T \subseteq \mathbb{R}^d$, decide: $\exists n \in \mathbb{N}$ such that $M^n x \in T$?

Linear loops: T encodes the **halting condition**.

Reasonable targets T: **semialgebraic sets**.

Example (Skolem's problem; Not known to be decidable)

T is a hyperplane defined by $u \in \mathbb{Q}^d$: $\{z \in \mathbb{R}^d : (z, u) = 0\}$

Example (Positivity problem; Provably mathematically hard![1])

T is a halfspace defined by $u \in \mathbb{Q}^d$: $\{z \in \mathbb{R}^d : (z, u) \geq 0\}$.

[1] Decidability entails major breakthroughs in field of Diophantine approximation.
Ouaknine, Worrell: Positivity problems for low-order linear recurrence sequences. (SODA2014)

So what can we do?

Reachability problems

Given $x \in \mathbb{Q}^d$, $M \in \mathbb{Q}^{d\times d}$, and a set $T \subseteq \mathbb{R}^d$, decide: $\exists n \in \mathbb{N}$ such that $M^n x \in T$?

Linear loops: T encodes the **halting condition**.

Have to restrict T!

Reasonable targets T: **semialgebraic sets**.

Example (Skolem's problem; Not known to be decidable)

T is a hyperplane defined by $u \in \mathbb{Q}^d$: $\{z \in \mathbb{R}^d : (z, u) = 0\}$

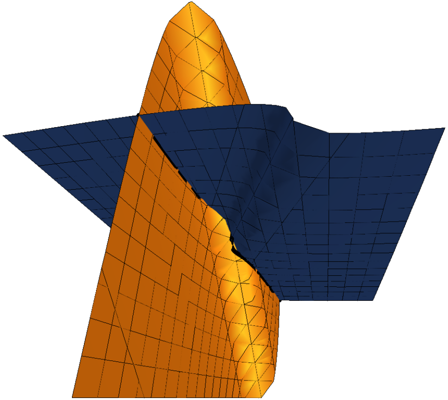
Example (Positivity problem; Provably mathematically hard![1])

T is a halfspace defined by $u \in \mathbb{Q}^d$: $\{z \in \mathbb{R}^d : (z, u) \geq 0\}$.

[1]Decidability entails major breakthroughs in field of Diophantine approximation.
Ouaknine, Worrell: Positivity problems for low-order linear recurrence sequences. (SODA2014)

Restricting the targets: dimension

Intrinsic dimension

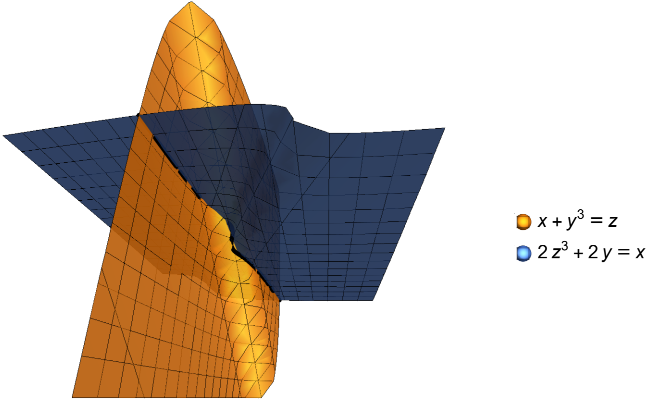


- $x + y^3 = z$
- $2z^3 + 2y = x$

In 3D ambient space:

- Surfaces: intrinsic dimension 2

Intrinsic dimension

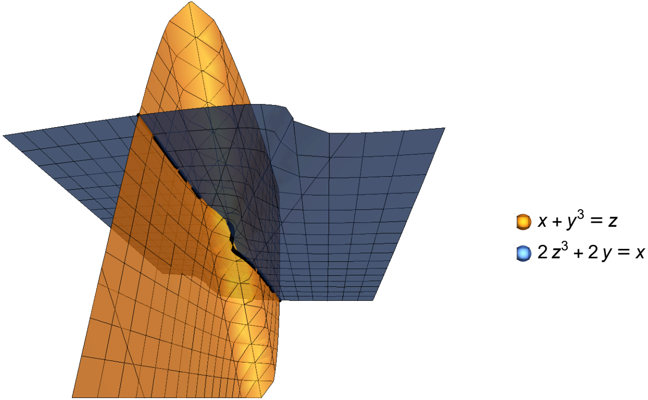


In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Restricting the targets: dimension

Intrinsic dimension

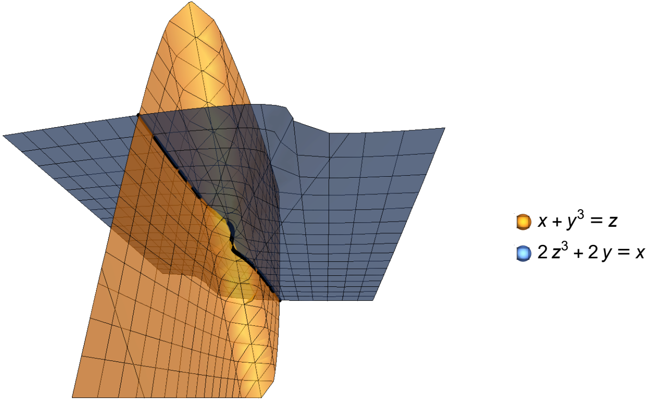


In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Restricting the targets: dimension

Intrinsic dimension

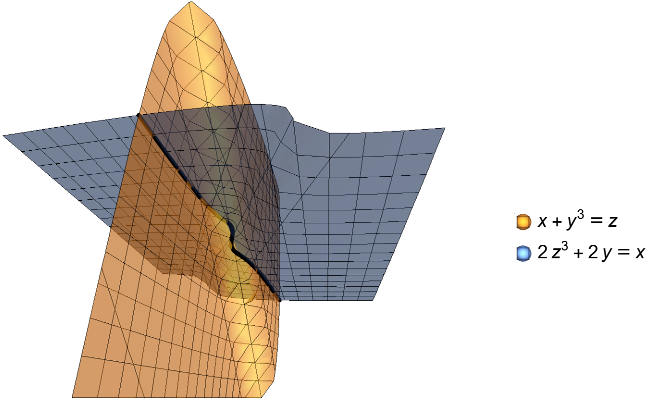


In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Restricting the targets: dimension

Intrinsic dimension

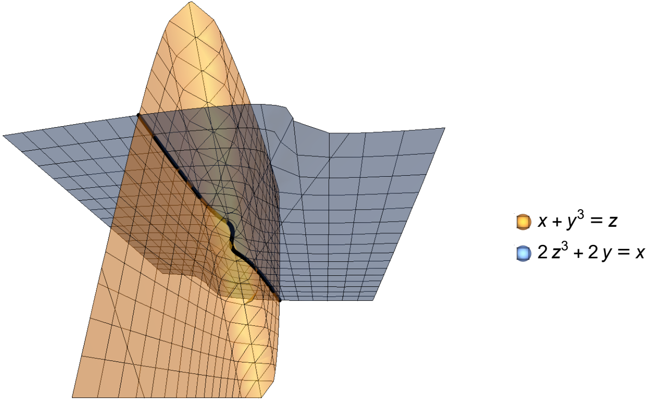


In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Restricting the targets: dimension

Intrinsic dimension

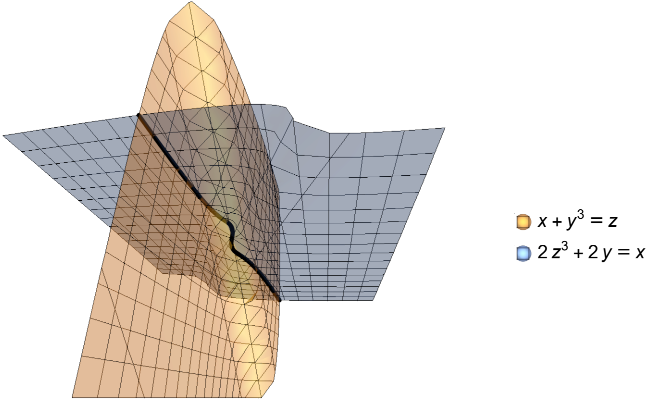


In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Restricting the targets: dimension

Intrinsic dimension

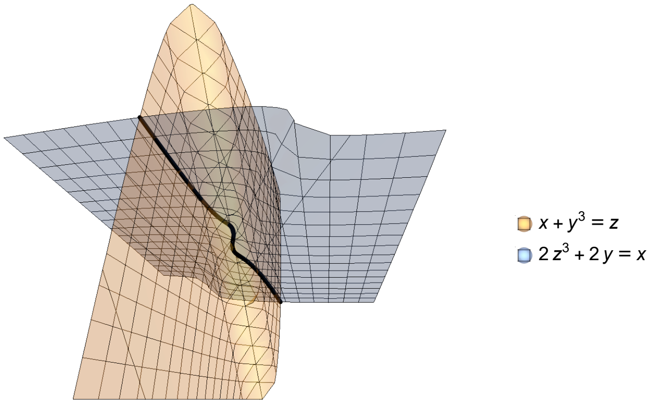


In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Restricting the targets: dimension

Intrinsic dimension

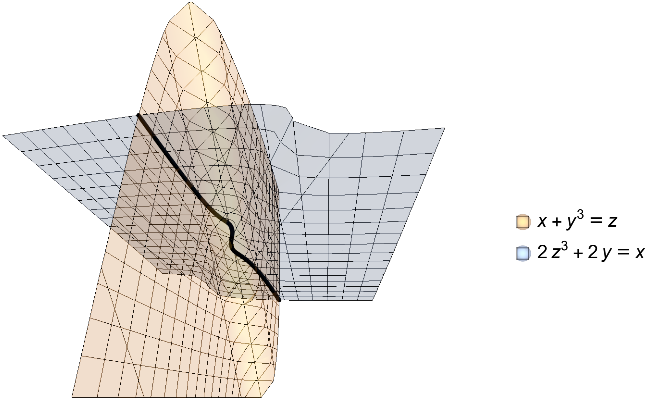


In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Restricting the targets: dimension

Intrinsic dimension

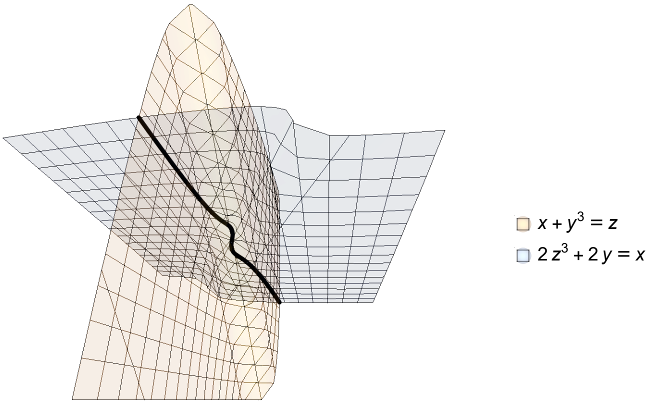


In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Restricting the targets: dimension

Intrinsic dimension



In 3D ambient space:

- Surfaces: intrinsic dimension 2
- Curves: intrinsic dimension 1

Examples: Our targets

Example

Consider the semialgebraic sets in ambient dimension 4:

$$T_1 = \{(t, u, v, w)\colon t + u + v - w = 0 \wedge (t^3 = u^2 \vee w \geq 3t^2 + u)\}$$

$$T_2 = \{(t, u, v, w)\colon t + u + 2v - 2w = 0 \wedge t^3 + v^2 + v > w\}$$

$$T_3 = \{(t, u, v, w) : t^4 - u^2 = 3 \wedge 2v^2 = w \wedge t^2 - 2u^3 = 4v\}$$

Examples: Our targets

Example

Consider the semialgebraic sets in ambient dimension 4:

$$T_1 = \{(t,u,v,w) \colon t+u+v-w=0 \wedge (t^3=u^2 \vee w \geq 3t^2+u)\}$$
$$T_2 = \{(t,u,v,w) \colon t+u+2v-2w=0 \wedge t^3+v^2+v > w\}$$
$$T_3 = \{(t,u,v,w) : t^4-u^2=3 \wedge 2v^2=w \wedge t^2-2u^3=4v\}$$

- T_1 and T_2 are contained in 3-dimensional subspaces.

Examples: Our targets

Example

Consider the semialgebraic sets in ambient dimension 4:

$$T_1 = \{(t, u, v, w) \colon t + u + v - w = 0 \wedge (t^3 = u^2 \vee w \geq 3t^2 + u)\}$$
$$T_2 = \{(t, u, v, w) \colon t + u + 2v - 2w = 0 \wedge t^3 + v^2 + v > w\}$$
$$T_3 = \{(t, u, v, w) : t^4 - u^2 = 3 \wedge 2v^2 = w \wedge t^2 - 2u^3 = 4v\}$$

- T_1 and T_2 are contained in 3-dimensional subspaces.
- T_3 has intrinsic dimension 1.

Examples: Our targets

Example

Consider the semialgebraic sets in ambient dimension 4:

$$T_1 = \{(t, u, v, w) \colon t + u + v - w = 0 \wedge (t^3 = u^2 \vee w \geq 3t^2 + u)\}$$
$$T_2 = \{(t, u, v, w) \colon t + u + 2v - 2w = 0 \wedge t^3 + v^2 + v > w\}$$
$$T_3 = \{(t, u, v, w) : t^4 - u^2 = 3 \wedge 2v^2 = w \wedge t^2 - 2u^3 = 4v\}$$

- T_1 and T_2 are contained in 3-dimensional subspaces.
- T_3 has intrinsic dimension 1.

Example

2D polytopes are semialgebraic sets contained in 3D subspaces.

Our results

Theorem

The reachability problem is decidable when T is a semialgebraic set

- *of* ***intrinsic dimension at most*** *1; or*
- *which is* ***contained in a subspace of (linear) dimension*** *3.*

Our results

Theorem

The reachability problem is decidable when T is a semialgebraic set

- *of* ***intrinsic dimension at most*** 1*; or*
- *which is* ***contained in a subspace of (linear) dimension*** 3*.*

Theorem (Skolem-hardness)

Skolem's problem at dimension 5 *reduces to reachability of T a dimension-*2 *set.*

Our results

Theorem

The reachability problem is decidable when T is a semialgebraic set

- *of* ***intrinsic dimension at most*** 1*; or*
- *which is* ***contained in a subspace of (linear) dimension*** 3*.*

Theorem (Skolem-hardness)

Skolem's problem at dimension 5 reduces to reachability of T a dimension-2 set.

Theorem (Chonev et al.[2])

- *Skolem's problem at dimension 5 reduces to reachability of a 3D polytope in $\mathbb{R}^4$.*
- *Reachability of a 4D polytope in $\mathbb{R}^4$ is* ***mathematically hard****.*

[2] Chonev, Ouaknine, Worrell: The Polyhedron-Hitting Problem. (SODA2015)

Semialgebraic targets of dimension 1

Proposition

Given x, M, and semialgebraic T of dimension 1, ***it is decidable whether*** *$\exists n \in \mathbb{N}$ such that $M^n x \in T$.*

Proof sketch.

We show that $\exists$ computable $N \in \mathbb{N}$ such that **if** such n exists, **then** $\exists n \leq N$.

1. Transform the system into Jordan normal form ($M = S^{-1}JS$).
   - Focus on invariant subspaces of M.
   - Case analysis depending on spectrum of M.

Semialgebraic targets of dimension 1

Proposition

Given x, M, and semialgebraic T of dimension 1, ***it is decidable whether*** $\exists n \in \mathbb{N}$ *such that* $M^n x \in T$.

Proof sketch.

We show that $\exists$ computable $N \in \mathbb{N}$ such that **if** such n exists, **then** $\exists n \leq N$.

1. Transform the system into Jordan normal form ($M = S^{-1}JS$).
   - Focus on invariant subspaces of M.
   - Case analysis depending on spectrum of M.
2. (Most cases) Solving instances of the Skolem problem.
3. (Corner cases) The system can be seen as a product of *arc hitting models / codings of rotations*.

$\square$

1. Transformation to Jordan normal form

- $M = S^{-1}JS$, where S is an invertible matrix and

$$J = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_s \end{pmatrix} \qquad \text{with} \quad J_i = \begin{pmatrix} \lambda_i & 1 & & \cdots \\ & \lambda_i & 1 & \cdots \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}$$

- λ_i an eigenvalue of M.

1. Transformation to Jordan normal form

- $M = S^{-1}JS$, where S is an invertible matrix and

$$J = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_s \end{pmatrix} \qquad \text{with} \quad J_i = \begin{pmatrix} \lambda_i & 1 & & \cdots \\ & \lambda_i & 1 & \cdots \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}$$

- λ_i an eigenvalue of M.

$M^n x \in T \Longleftrightarrow \underline{J^n(Sx) \in S(T)}$.

1. Transformation to Jordan normal form

- $M = S^{-1}JS$, where S is an invertible matrix and

$$J = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_s \end{pmatrix} \qquad \text{with} \quad J_i = \begin{pmatrix} \lambda_i & 1 & & \cdots \\ & \lambda_i & 1 & \cdots \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}$$

- λ_i an eigenvalue of M.

$M^n x \in T \iff \underline{J^n y \in T'}$.

1. Transformation to Jordan normal form

- $M = S^{-1}JS$, where S is an invertible matrix and

$$J^n y = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_s \end{pmatrix}^n \begin{pmatrix} \vec{y}_1 \\ \vec{y}_2 \\ \vdots \\ \vec{y}_s \end{pmatrix} \quad \text{with} \quad J_i^n \vec{y}_i = \begin{pmatrix} \lambda_i^n & n\lambda_i^{n-1} & \binom{n}{2}\lambda_i^{n-2} & \cdots \\ & \lambda_i^n & n\lambda_i^{n-1} & \cdots \\ & & \ddots & n\lambda_i^{n-1} \\ & & & \lambda_i^n \end{pmatrix} \begin{pmatrix} \vec{y}_{i,1} \\ \vec{y}_{i,2} \\ \vdots \\ \vec{y}_{i,s_i} \end{pmatrix}$$

- λ_i an eigenvalue of M.

$M^n x \in T \iff \underline{J^n y \in T'}$.

1. Transformation to Jordan normal form

- $M = S^{-1}JS$, where S is an invertible matrix and

$$J^n y = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_s \end{pmatrix}^n \begin{pmatrix} \vec{y}_1 \\ \vec{y}_2 \\ \vdots \\ \vec{y}_s \end{pmatrix} \quad \text{with} \quad J_i^n \vec{y}_i = \begin{pmatrix} \lambda_i^n & n\lambda_i^{n-1} & \binom{n}{2}\lambda_i^{n-2} & \cdots \\ & \lambda_i^n & n\lambda_i^{n-1} & \cdots \\ & & \ddots & n\lambda_i^{n-1} \\ & & & \lambda_i^n \end{pmatrix} \begin{pmatrix} \vec{y}_{i,1} \\ \vec{y}_{i,2} \\ \vdots \\ \vec{y}_{i,s_i} \end{pmatrix}$$

- λ_i an eigenvalue of M.

$M^n x \in T \iff \underline{J^n y \in T'}$.

Interesting cases: Assume λ_i and λ_j are **multiplicatively independent**;
$\lambda_i^a \lambda_j^b = 1 \implies a = b = 0$.

Project $J^n y$ to the coordinates $(\lambda_i^n y_{i,s_i}, \lambda_j^n y_{j,s_j})$.

T' is complex semialgebraic of dimension 1;
There exists a bivariate polynomial P such that: $J^n y \in T'$ implies

$$P(\lambda_i^n y_{i,s_i}, \lambda_j^n y_{j,s_j}) = 0$$

3. Solving instances of Skolem's problem

T' is complex semialgebraic of dimension 1;
There exists a bivariate polynomial P such that: $J^n y \in T'$ implies

$$P(\lambda_i^n y_{i,s_i}, \lambda_j^n y_{j,s_j}) = 0$$

The sequence $P(\lambda_i^n y_{i,s_i}, \lambda_j^n y_{j,s_j}) = \sum_{i,j}^d a_{i,j}(\lambda_i^i \lambda_j^j)^n$ is a linear recurrence sequence;

Reaching $T' \implies$ zero of this LRS.

3. Solving instances of Skolem's problem

T' is complex semialgebraic of dimension 1;
There exists a bivariate polynomial P such that: $J^n y \in T'$ implies

$$P(\lambda_i^n y_{i,s_i}, \lambda_j^n y_{j,s_j}) = 0$$

The sequence $P(\lambda_i^n y_{i,s_i}, \lambda_j^n y_{j,s_j}) = \sum_{i,j}^{d} a_{i,j} (\lambda_i^i \lambda_j^j)^n$ is a linear recurrence sequence;

Reaching $T' \implies$ zero of this LRS.

We get lucky: use known technology à la Shorey et al.[3] and Vereshchagin[4].

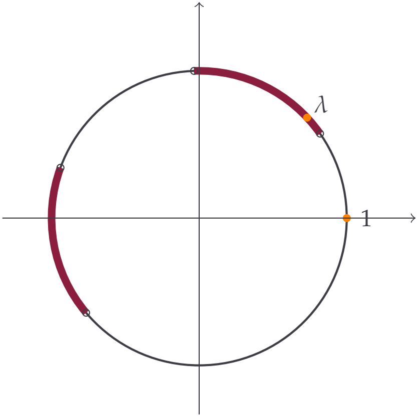
- (p-adic) Baker's theorem on linear forms in logarithms.

[3] Shorey, Tijdeman, Mignotte: *The distance between terms of an algebraic recurrence sequence*, (1984)
[4] Vereshchagin: *Occurrence of zero in a linear recursive sequence* (1985)

4. Corner cases; codings of rotations

Given $M = \lambda$ an algebraic number of modulus 1, $x = 1$, and open arcs $I_1, \ldots, I_k$ of unit circle in the complex plain (with algebraic endpoints)

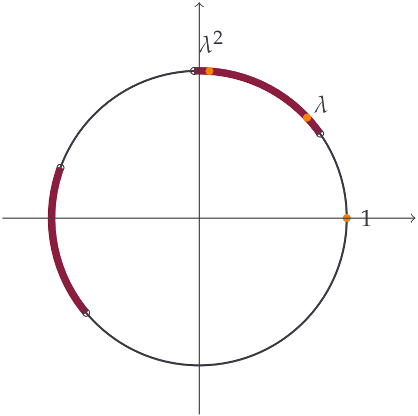
Decide if there exists $n \geq 0$ such that $\lambda^n \in \bigcup_{j=1}^{k} I_j$. (Almost trivial!)



4. Corner cases; codings of rotations

Given $M = \lambda$ an algebraic number of modulus 1, $x = 1$, and open arcs $I_1, \dots, I_k$ of unit circle in the complex plain (with algebraic endpoints)

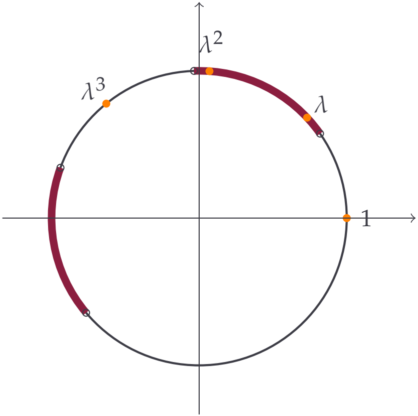
Decide if there exists $n \geq 0$ such that $\lambda^n \in \bigcup_{j=1}^{k} I_j$. (Almost trivial!)



4. Corner cases; codings of rotations

Given $M = \lambda$ an algebraic number of modulus 1, $x = 1$, and open arcs $I_1, \ldots, I_k$ of unit circle in the complex plain (with algebraic endpoints)

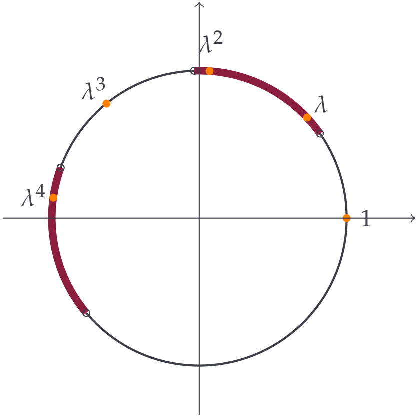
Decide if there exists $n \geq 0$ such that $\lambda^n \in \bigcup_{j=1}^{k} I_j$. (Almost trivial!)



4. Corner cases; codings of rotations

Given $M = \lambda$ an algebraic number of modulus 1, $x = 1$, and open arcs $I_1, \ldots, I_k$ of unit circle in the complex plain (with algebraic endpoints)

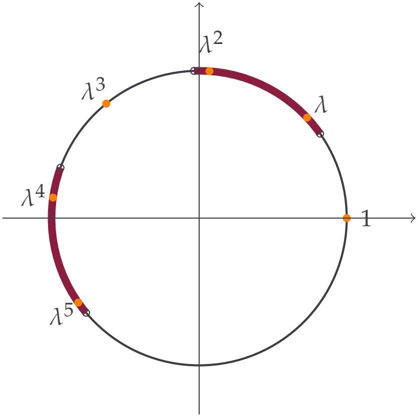
Decide if there exists $n \geq 0$ such that $\lambda^n \in \bigcup_{j=1}^{k} I_j$. (Almost trivial!)



4. Corner cases; codings of rotations

Given $M = \lambda$ an algebraic number of modulus 1, $x = 1$, and open arcs $I_1, \ldots, I_k$ of unit circle in the complex plain (with algebraic endpoints)

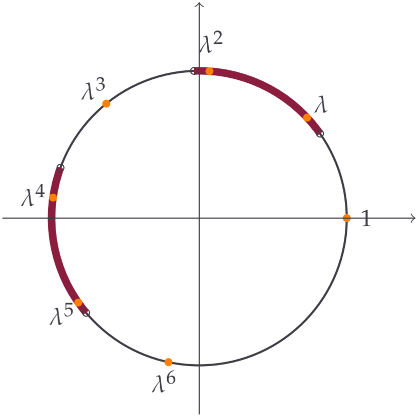
Decide if there exists $n \geq 0$ such that $\lambda^n \in \bigcup_{j=1}^{k} I_j$. (Almost trivial!)



4. Corner cases; codings of rotations

Given $M = \lambda$ an algebraic number of modulus 1, $x = 1$, and open arcs $I_1, \ldots, I_k$ of unit circle in the complex plain (with algebraic endpoints)

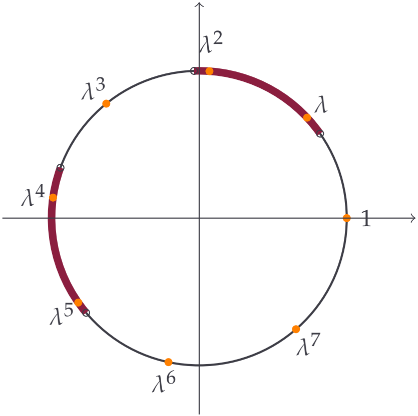
Decide if there exists $n \geq 0$ such that $\lambda^n \in \bigcup_{j=1}^{k} I_j$. (Almost trivial!)



4. Corner cases; codings of rotations

Given $M = \lambda$ an algebraic number of modulus 1, $x = 1$, and open arcs $I_1, \ldots, I_k$ of unit circle in the complex plain (with algebraic endpoints)

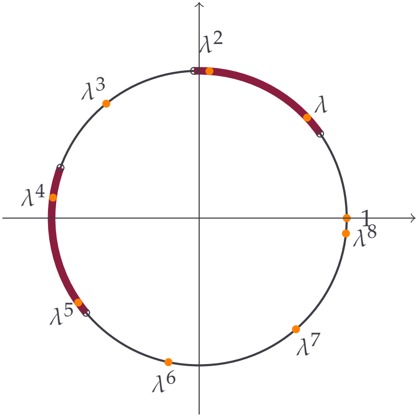
Decide if there exists $n \geq 0$ such that $\lambda^n \in \bigcup_{j=1}^{k} I_j$. (Almost trivial!)



4. Corner cases; codings of rotations

Given $M = \lambda$ an algebraic number of modulus 1, $x = 1$, and open arcs $I_1, \ldots, I_k$ of unit circle in the complex plain (with algebraic endpoints)

Decide if there exists $n \geq 0$ such that $\lambda^n \in \bigcup_{j=1}^{k} I_j$. (Almost trivial!)



Beyond reachability

Beyond reachability

```
real t,te,time;
assume(te=14 and 16<=t and t<=17);
while true {
  time := 0; - timer measuring duration in each mode

  while (t<=22) { - heating mode
    t := 15/16*t-1/16*te+1; time++;
  }
  time := 0;
  while (t>=18){ - cooling mode
    t := 15/16*t-1/16*te; time++;
  }
}
```

Figure 2. A thermostat system, composed of two simple loops inside a outer loop.

from: Jeannet, Schrammel, Sankaranarayanan: *Abstract Acceleration of General Linear Loops*, POPL2014

Beyond reachability

Algorithm

$t := 1$
$u := -1$
$v := 2$
$w := 0$
while `true` **do**
  $t := 3t + 2u - 5w$
  $u := u + 3w$
  $v := 4u + 3v + w$
  $w := t + u + 2v$
end while

Temporal analysis:
Do we get stuck in some set of bad states?

Beyond reachability

Algorithm

$t := 1$
$u := -1$
$v := 2$
$w := 0$
while `true` **do**
  $t := 3t + 2u - 5w$
  $u := u + 3w$
  $v := 4u + 3v + w$
  $w := t + u + 2v$
end while

Temporal analysis:
Do we get stuck in some set of bad states?

Define the predicates

$$P_1 : t + u + v - w = 0 \wedge (t^3 = u^2 \vee w \geq 3t^2 + u)$$
$$P_2 : t + u + 2v - 2w = 0 \wedge t^3 + v^2 + v > w$$
$$P_3 : t^4 - u^2 = 3 \wedge 2v^2 = w \wedge t^2 - 2u^3 = 4v$$

Beyond reachability

Algorithm

$t := 1$
$u := -1$
$v := 2$
$w := 0$
while true **do**
  $t := 3t + 2u - 5w$
  $u := u + 3w$
  $v := 4u + 3v + w$
  $w := t + u + 2v$
end while

Temporal analysis:
Do we get stuck in some set of bad states?

Define the predicates

$$P_1: t + u + v - w = 0 \wedge (t^3 = u^2 \vee w \geq 3t^2 + u)$$
$$P_2: t + u + 2v - 2w = 0 \wedge t^3 + v^2 + v > w$$
$$P_3: t^4 - u^2 = 3 \wedge 2v^2 = w \wedge t^2 - 2u^3 = 4v$$

Does the system satisfy the following LTL formula?

$$\mathbf{G}(P_1 \Rightarrow \mathbf{F}\neg P_2) \wedge \mathbf{F}(P_3 \vee \neg P_1)\,.$$

"whenever P_1 holds, then P_2 must eventually subsequently fail, and eventually either P_3 will hold or P_1 will fail"

Temporal properties of LDS

Let (M, x) be a LDS, let $T_1, \ldots, T_k$ be semialgebraic sets.

Definition (The characteristic word of the LDS (M, x) with respect to T_i)

$$\pi(M, x, T_1, \ldots, T_k) = a_0 a_1 a_2 \cdots \in \mathcal{P}(\{1, \ldots, k\})^{\mathbb{N}}$$

defined by

$$a_n = S \subseteq \{1, \ldots, k\}$$

if and only if $\forall i \in S \colon M^n x \in T_i$ and $\forall j \in \{1, \ldots, k\} \setminus S \colon M^n x \notin T_j$.

Temporal properties of LDS

Let (M, x) be a LDS, let $T_1, \ldots, T_k$ be semialgebraic sets.

Definition (The characteristic word of the LDS (M, x) with respect to T_i)

$$\pi(M, x, T_1, \ldots, T_k) = a_0 a_1 a_2 \cdots \in \mathcal{P}(\{1, \ldots, k\})^{\mathbb{N}}$$

defined by

$$a_n = S \subseteq \{1, \ldots, k\}$$

if and only if $\forall i \in S \colon M^n x \in T_i$ and $\forall j \in \{1, \ldots, k\} \setminus S \colon M^n x \notin T_j$.

Interested in temporal properties of the system; Translate to properties on $\pi(M, x, T_1, \ldots, T_k)$

Temporal properties of LDS

Let (M, x) be a LDS, let $T_1, \ldots, T_k$ be semialgebraic sets.

Definition (The characteristic word of the LDS (M, x) with respect to T_i)

$$\pi(M, x, T_1, \ldots, T_k) = a_0 a_1 a_2 \cdots \in \mathcal{P}(\{1, \ldots, k\})^{\mathbb{N}}$$

defined by

$$a_n = S \subseteq \{1, \ldots, k\}$$

if and only if $\forall i \in S \colon M^n x \in T_i$ and $\forall j \in \{1, \ldots, k\} \setminus S \colon M^n x \notin T_j$.

Interested in temporal properties of the system; Translate to properties on $\pi(M, x, T_1, \ldots, T_k)$

Properties given by formulas from Monadic Second-Order Logic (capture ω-regular properties).

MSO: Monadic Second-Order Logic

MSO over the structure $(\mathbb{N}, <)$ and a finite collection of predicates $P_1, \ldots, P_k : \mathbb{N} \to \{\mathtt{true}, \mathtt{false}\}$; MSO on infinite words: Predicates P_i describes the indices with letter i.

The grammar of formulas:

$\psi := P(i)$	(where $P(i)$ is a predicate on position i of the word)
$\psi := \exists i \in \mathbb{N} : \psi \mid \forall i \in \mathbb{N} : \psi$	(first-order quantification)
$\psi := \exists X \subseteq \mathbb{N} : \psi \mid \forall X \subseteq \mathbb{N} : \psi$	(monadic second-order quantification)
$\psi := i \in X \mid i \notin X$	(subset membership testing)
$\psi := i < j \mid i = j$	(index comparison)
$\psi := \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \psi \Rightarrow \psi$	(standard logical operations)
$\psi := i = 0 \mid i = 1 \mid i = 2 \mid \ldots$	(fixed values)

MSO

For us, the predicate P_i corresponds to those n: $M^n x \in T_i$.

Example

Examples of MSO formulas for model checking LDS:

- Reachability of target T_i: $\exists n : P_i(n)$.
- Eventually trapped inside T_i: $\exists n \forall m : m > n \implies P_i(m)$.
- In target T_i at every odd position (O = the set of odd natural numbers): $\exists O \subseteq \mathbb{N} : 1 \in O \wedge \forall x \in O, \exists y, z : (y \notin O \wedge z \in O \wedge x < y < z \wedge \nexists t : x < t < y \vee y < t < z) \wedge \forall x : x \in O \implies P_i(x)$.
- Whenever T_i is visited T_j is visited some point later: $\forall n : P_i(n) \implies \exists m > n : P_j(m)$.
- Any linear temporal logic (LTL) formula over predicates $P_1, \ldots, P_k$.

The MSO model-checking problem for LDS

Theorem

Let (M, x) be a LDS, $T_1, \ldots, T_k \subseteq \mathbb{R}^d$ each T_i of which is a semialgebraic set

- *of* ***(semialgebraic) dimension at most*** *1; or*
- *which is* ***contained in a subspace of (linear) dimension*** *3.*

Then it is decidable whether $\pi(M, x, T_1, \ldots, T_k)$ satisfies a given MSO formula ψ.

Key notion: Effective almost periodicity

Definition (Almost periodic words)

An infinite word $w \in \Sigma^\omega$ is **almost periodic** if for every factor $u \in \Sigma^*$, there exists $p \in \mathbb{N}$ such that either:

- u does not occur in w after the position p,
- or u occurs in every factor of w of length p.

Effectively almost periodic word: p can be computed for every factor u.

Key notion: Effective almost periodicity

Definition (Almost periodic words)

An infinite word $w \in \Sigma^\omega$ is **almost periodic** if for every factor $u \in \Sigma^*$, there exists $p \in \mathbb{N}$ such that either:

- u does not occur in w after the position p,
- or u occurs in every factor of w of length p.

Effectively almost periodic word: p can be computed for every factor u.

Theorem (Semënov'84)

Let w be an effectively almost periodic infinite word over alphabet Σ. The MSO theory over $(\mathbb{N}, <)$ extended with the unary predicates $\{P_a\}_{a \in \Sigma}$ remains decidable.

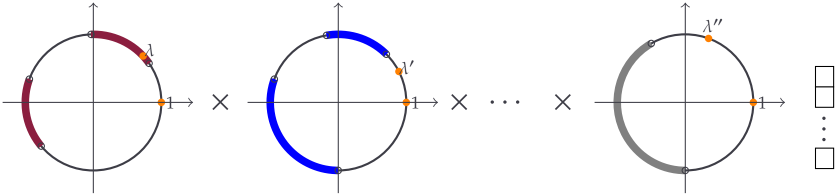
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \leq r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



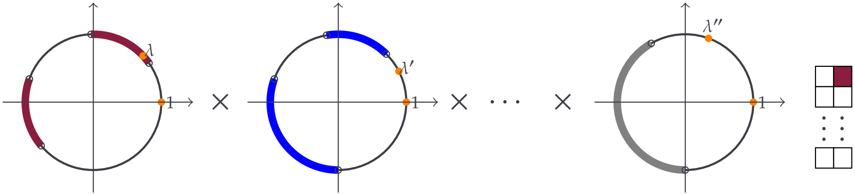
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \leq r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



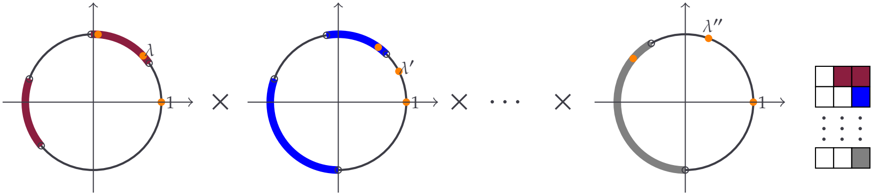
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \le r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



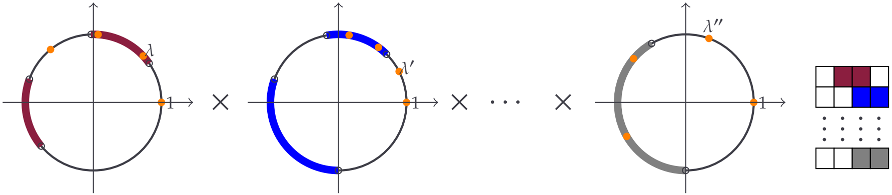
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \le r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



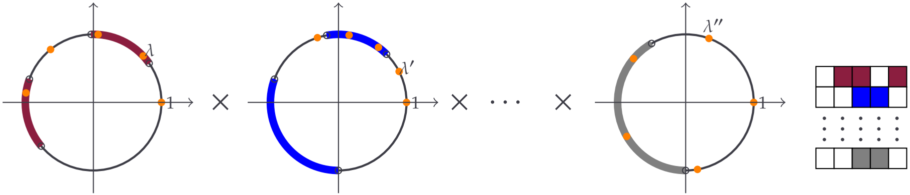
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \leq r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



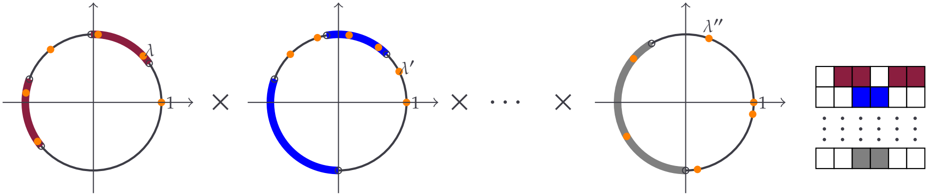
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \leq r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



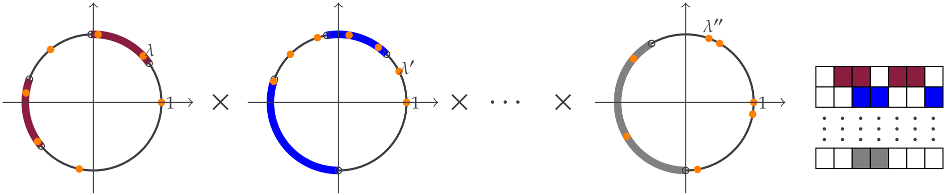
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \leq r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



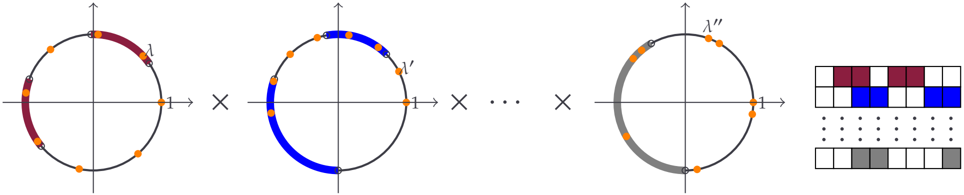
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \le r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



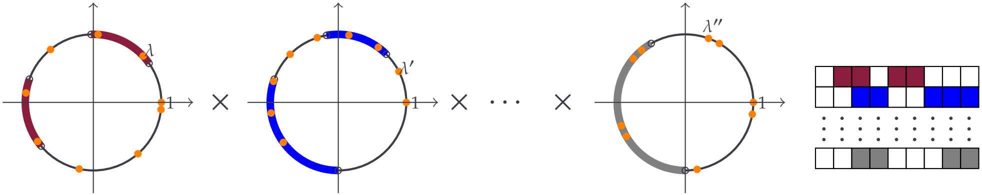
Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \leq r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*



Proof sketch

Analysis from decidability of reachability shows:

Observation

Let (M, x) be a LDS and T be a semialgebraic set of dimension 1 or a semialgebraic set contained in a subspace of linear dimension 3. There exists a computable $\ell \in \mathbb{N}$ such that for all $0 \leq r < \ell$ the word $\pi(M^\ell, M^r x, T)$ is either

- *eventually constant; or*
- *a coding of a rotation up to a finite computable prefix.*

Corollary

$\pi(M, x, T_0, \ldots, T_k)$ is an interleaving of codings of rotations on $\mathbb{T}^k$ (up to a finite prefix).

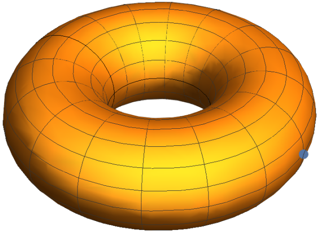
Coding of a rotation

Definition (Coding of a rotation on $\mathbb{T}^k$.)

Given a semialgebraic set $A_1 \subseteq \mathbb{T}^k$ and algebraic point $\vec{\lambda} = (\lambda_1, \ldots, \lambda_k) \in \mathbb{T}^k$. Let $A_0 = \mathbb{T}^k \setminus A_1$.
Coding: $\mathbf{w} = a_0 a_1 \cdots \in \{0,1\}^{\mathbb{N}}$ where $a_n = 1 \Leftrightarrow \vec{\lambda}^n = (\lambda_1^n, \ldots, \lambda_k^n) \in A_1$.

Proof sketch (of effective almost periodicity)

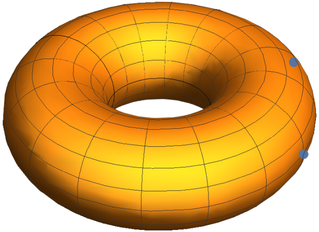
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

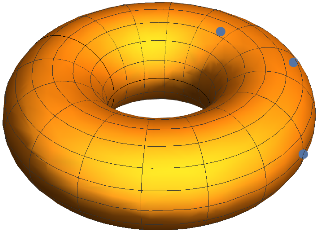
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5]Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

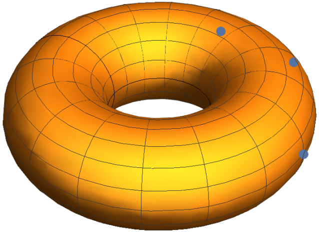
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

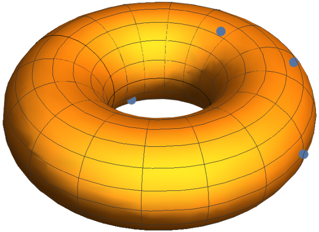
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

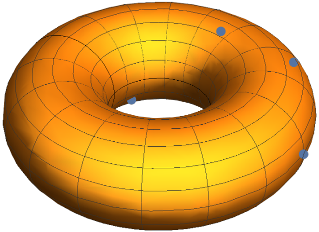
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n \in \mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

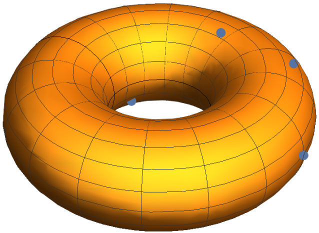
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5]Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

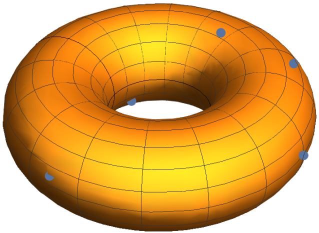
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5]Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

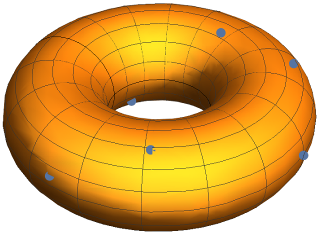
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

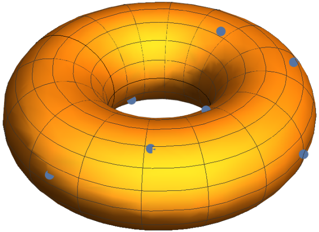
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

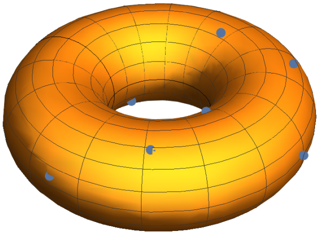
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n \in \mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

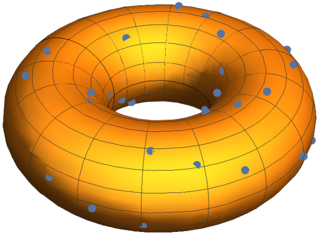
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

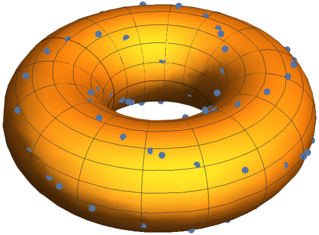
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

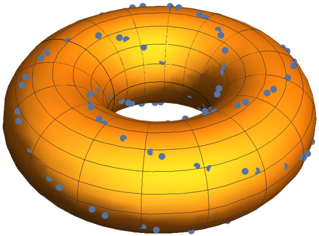
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

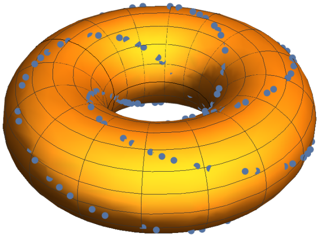
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

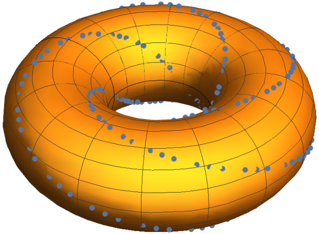
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n \in \mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5]Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

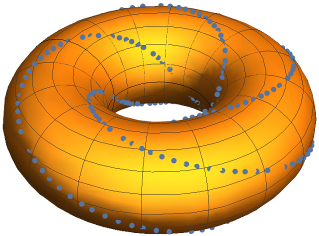
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5]Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

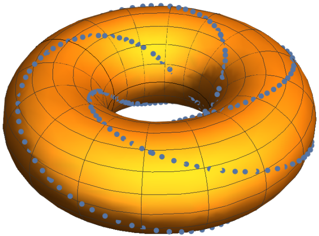
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

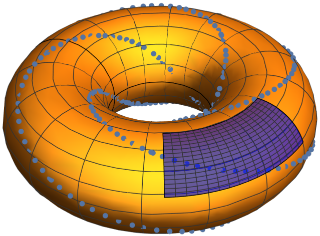
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

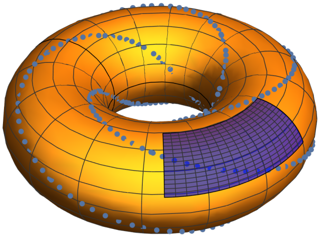
The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



[5]Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n \in \mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]

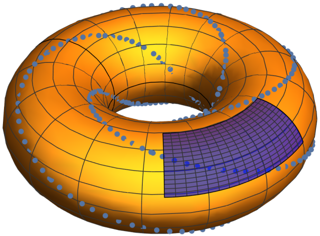


Factor $u = b_1 \cdots b_n$ occurs in $\mathbf{w}$ iff $C \cap A_{b_1} \cap \vec{\lambda}^{-1} A_{b_2} \cap \ldots \cap \vec{\lambda}^{-n} A_{b_n} \neq \emptyset$

[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



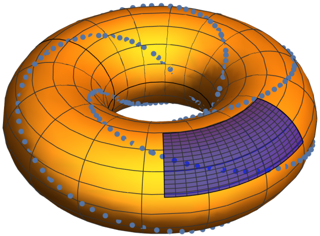
Factor $u = b_1 \cdots b_n$ occurs in $\mathbf{w}$ iff $C \cap A_{b_1} \cap \vec{\lambda}^{-1} A_{b_2} \cap \ldots \cap \vec{\lambda}^{-n} A_{b_n} \neq \emptyset$

Almost periodicity by compactness and guess-and-check.

[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Proof sketch (of effective almost periodicity)

The closure C of $\{(\lambda_1^n, \ldots, \lambda_k^n)\}_{n\in\mathbb{N}}$ is semialgebraic. Representation can be effectively computed![5]



Factor $u = b_1 \cdots b_n$ occurs in $\mathbf{w}$ iff $C \cap A_{b_1} \cap \vec{\lambda}^{-1} A_{b_2} \cap \ldots \cap \vec{\lambda}^{-n} A_{b_n} \neq \emptyset$

Almost periodicity by compactness and guess-and-check.

Effectiveness by Tarski: First-order theory of the reals is decidable.

[5] Ouaknine, Worrell: *On the Positivity Problem for Simple Linear Recurrence Sequences*, (ICALP 2014)

Parametric reachability

Parametric Loops as Parametric Dynamical Systems

Algorithm

$\vec{x} := \vec{a}$

while $\vec{x} \neq \vec{y}$ **do**
  $\vec{x} := M\vec{x}$
end while

Theorem (Kannan–Lipton'86)

*Orbit Problem is decidable for x, y, M with **algebraic entries**.*
In $\mathtt{PTIME}$ for x, y, M over $\mathbb{Q}$.

Parametric Loops as Parametric Dynamical Systems

Algorithm with **Input** $a_1, \ldots, a_k \in \mathbb{R}$

$\vec{x} := \vec{a}$

while $\vec{x} \neq \vec{y}$ **do**
  $\vec{x} := M\vec{x}$
end while

Theorem (Kannan–Lipton'86)

*Orbit Problem is decidable for x, y, M with **algebraic entries**.*
In $\mathtt{PTIME}$ for x, y, M over $\mathbb{Q}$.

Parametric Loops as Parametric Dynamical Systems

Algorithm with **Input** $a_1, \ldots, a_k \in \mathbb{R}$

$\vec{x} := \vec{a}(a_1, \ldots, a_k)$

while $\vec{x} \neq \vec{y}$ **do**
  $\vec{x} := M\vec{x}$
end while

Theorem (Kannan–Lipton'86)

*Orbit Problem is decidable for x, y, M with **algebraic entries**.*
In $\mathtt{PTIME}$ for x, y, M over $\mathbb{Q}$.

Algorithm with **Input** $a_1, \dots, a_k \in \mathbb{R}$

$\vec{x} := \vec{a}(a_1, \dots, a_k)$
$M := M(a_1, \dots, a_k)$

while $\vec{x} \neq \vec{y}$ **do**
  $\vec{x} := M\vec{x}$
end while

Theorem (Kannan–Lipton'86)

*Orbit Problem is decidable for x, y, M with **algebraic entries**.*
In $\mathtt{PTIME}$ for x, y, M over $\mathbb{Q}$.

Parametric Loops as Parametric Dynamical Systems

Algorithm with **Input** $a_1, \ldots, a_k \in \mathbb{R}$

$\vec{x} := \vec{a}(a_1, \ldots, a_k)$
$M := M(a_1, \ldots, a_k)$
$\vec{y} := \vec{y}(a_1, \ldots, a_k)$
while $\vec{x} \neq \vec{y}$ **do**
  $\vec{x} := M\vec{x}$
end while

Theorem (Kannan–Lipton'86)

*Orbit Problem is decidable for x, y, M with **algebraic entries**.*
In $\mathtt{PTIME}$ for x, y, M over $\mathbb{Q}$.

Parametric Loops as Parametric Dynamical Systems

Algorithm with **Input** $a_1, \ldots, a_k \in \mathbb{R}$

$\vec{x} := \vec{a}(a_1, \ldots, a_k)$
$M := M(a_1, \ldots, a_k)$
$\vec{y} := \vec{y}(a_1, \ldots, a_k)$
while $\vec{x} \neq \vec{y}$ **do**
  $\vec{x} := M\vec{x}$
end while

Theorem (Kannan–Lipton'86)

*Orbit Problem is decidable for x, y, M with **algebraic entries**.*
In `PTIME` *for x, y, M over $\mathbb{Q}$.*

For fixed input $\vec{a} = (a_1, \ldots, a_k)$:
loop halts $\iff \exists n : M(\vec{a})^n \vec{x}(\vec{a}) = \vec{y}(\vec{a})$.

- Halting depends on input.
- Halting on different n for different inputs.

Parametric Loops as Parametric Dynamical Systems

Algorithm with **Input** $a_1, \ldots, a_k \in \mathbb{R}$

$\vec{x} := \vec{a}(a_1, \ldots, a_k)$
$M := M(a_1, \ldots, a_k)$
$\vec{y} := \vec{y}(a_1, \ldots, a_k)$
while $\vec{x} \neq \vec{y}$ **do**
  $\vec{x} := M\vec{x}$
end while

For fixed input $\vec{a} = (a_1, \ldots, a_k)$:
loop halts $\iff \exists n : M(\vec{a}\,)^n \vec{x}(\vec{a}\,) = \vec{y}(\vec{a}\,)$.

- Halting depends on input.
- Halting on different n for different inputs.

Problem

Does there exist input $a_1, \ldots, a_k$ such that the loop halts?

$$\exists a_1, \ldots, a_k \in \mathbb{R},\ n \in \mathbb{N} : M(a_1, \ldots, a_k)^n \vec{x}(a_1, \ldots, a_k) = \vec{y}(a_1, \ldots, a_k)?$$

Problem statement

$\mathbb{Q}(z_1, \ldots, z_k)$: Rational functions in variables $z_1, \ldots, z_k$

Parametric Orbit Problem

Given

- initial vector $x \in \mathbb{Q}(z_1, \ldots, z_k)^d$,
- update matrix $M \in \mathbb{Q}(z_1, \ldots, z_k)^{d \times d}$, and
- target vector $y \in \mathbb{Q}(z_1, \ldots, z_k)^d$.

Problem statement

$\mathbb{Q}(z_1, \ldots, z_k)$: Rational functions in variables $z_1, \ldots, z_k$

Parametric Orbit Problem

Given

- initial vector $x \in \mathbb{Q}(z_1, \ldots, z_k)^d$,
- update matrix $M \in \mathbb{Q}(z_1, \ldots, z_k)^{d \times d}$, and
- target vector $y \in \mathbb{Q}(z_1, \ldots, z_k)^d$.

do there exist $\vec{a} = (a_1, \ldots, a_k) \in \mathbb{R}^k$ and $n \in \mathbb{N}$ such that

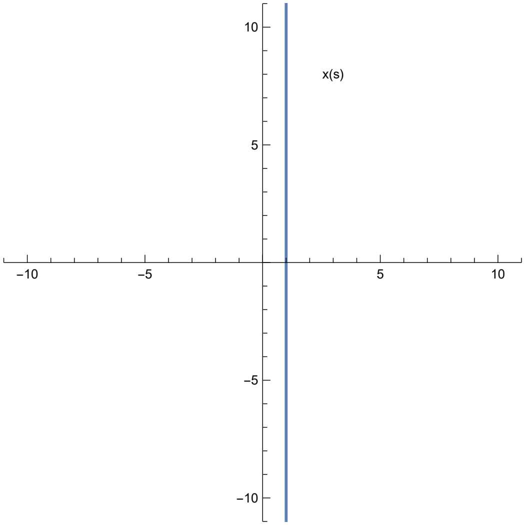
$$y(\vec{a}\,) = M(\vec{a}\,)^n x(\vec{a}\,)?$$

Parametric Linear Dynamical Systems

Input: $s \in \mathbb{R}$

$$x(s) = \begin{pmatrix} 1 \\ s \end{pmatrix}$$

$$M(s) = \begin{pmatrix} 1 & s-1 \\ 1 & -1 \end{pmatrix}$$

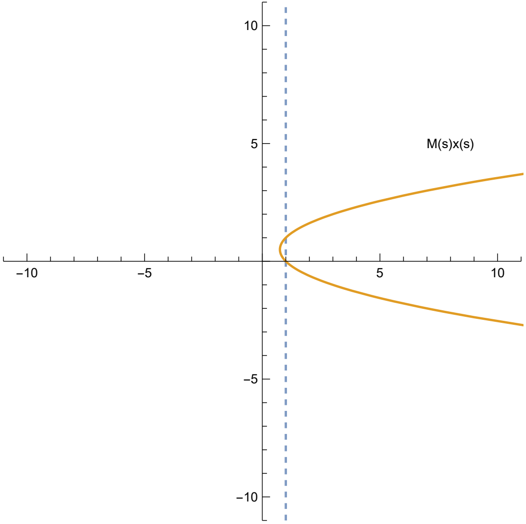


Parametric Linear Dynamical Systems

Input: $s \in \mathbb{R}$

$$x(s) = \begin{pmatrix} 1 \\ s \end{pmatrix}$$

$$M(s) = \begin{pmatrix} 1 & s-1 \\ 1 & -1 \end{pmatrix}$$

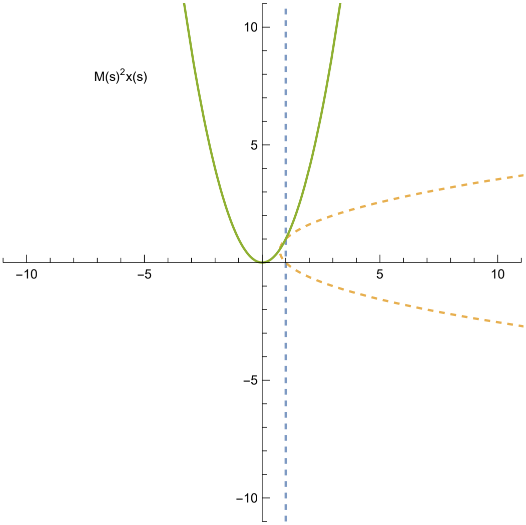


Parametric Linear Dynamical Systems

Input: $s \in \mathbb{R}$

$$x(s) = \begin{pmatrix} 1 \\ s \end{pmatrix}$$

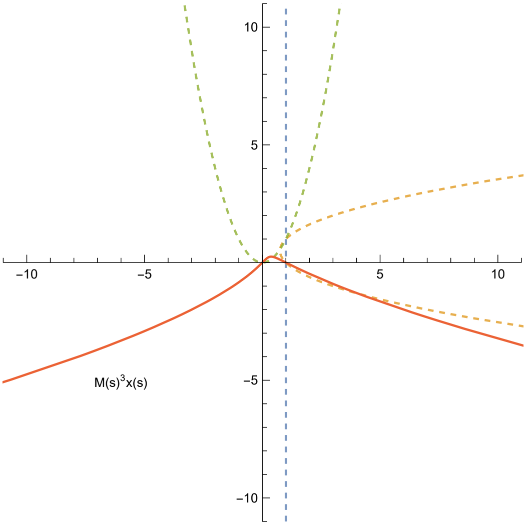
$$M(s) = \begin{pmatrix} 1 & s-1 \\ 1 & -1 \end{pmatrix}$$



Input: $s \in \mathbb{R}$

$$x(s) = \begin{pmatrix} 1 \\ s \end{pmatrix}$$

$$M(s) = \begin{pmatrix} 1 & s-1 \\ 1 & -1 \end{pmatrix}$$

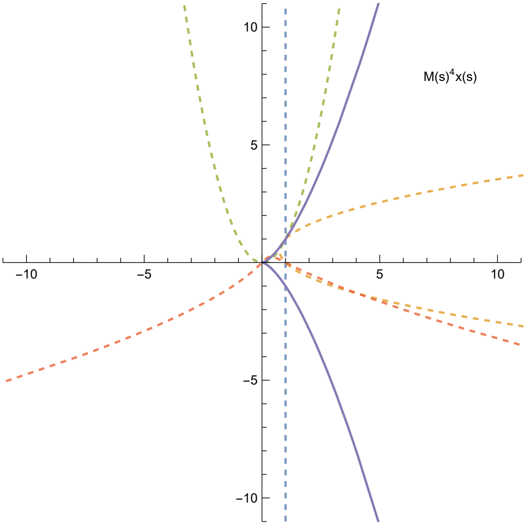


Parametric Linear Dynamical Systems

Input: $s \in \mathbb{R}$

$$x(s) = \begin{pmatrix} 1 \\ s \end{pmatrix}$$

$$M(s) = \begin{pmatrix} 1 & s-1 \\ 1 & -1 \end{pmatrix}$$

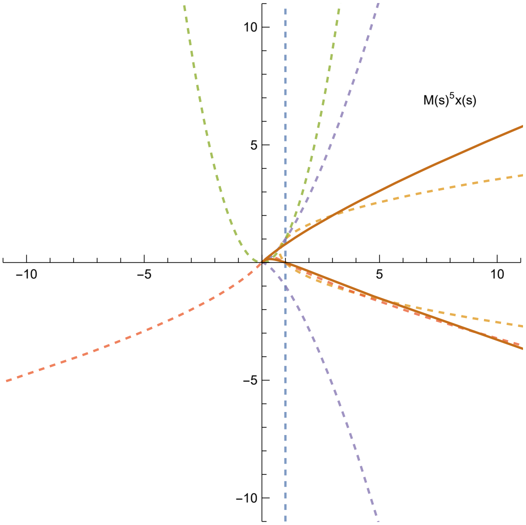


Parametric Linear Dynamical Systems

Input: $s \in \mathbb{R}$

$$x(s) = \begin{pmatrix} 1 \\ s \end{pmatrix}$$

$$M(s) = \begin{pmatrix} 1 & s-1 \\ 1 & -1 \end{pmatrix}$$

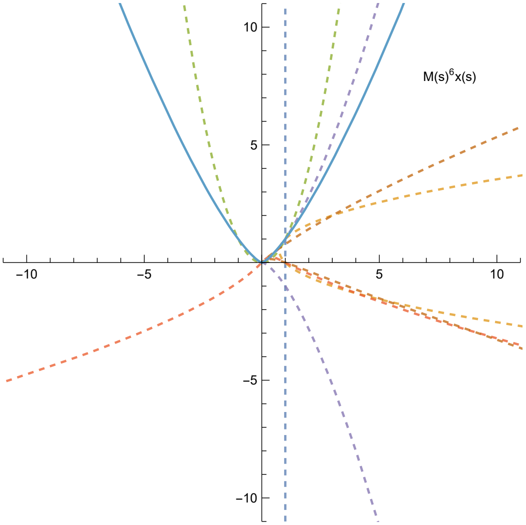


Parametric Linear Dynamical Systems

Input: $s \in \mathbb{R}$

$$x(s) = \begin{pmatrix} 1 \\ s \end{pmatrix}$$

$$M(s) = \begin{pmatrix} 1 & s-1 \\ 1 & -1 \end{pmatrix}$$



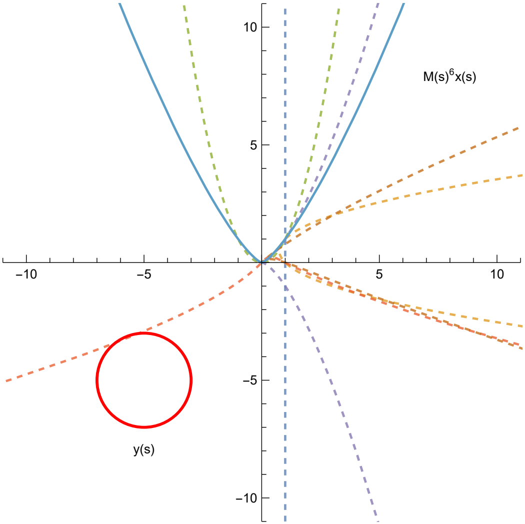
Parametric Linear Dynamical Systems

Input: $s \in \mathbb{R}$

$$x(s) = \begin{pmatrix} 1 \\ s \end{pmatrix}$$

$$M(s) = \begin{pmatrix} 1 & s-1 \\ 1 & -1 \end{pmatrix}$$

$$y(s) = \begin{pmatrix} -5 + 2\frac{1-s^2}{1+s^2} \\ -5 + 2\frac{2s}{1+s^2} \end{pmatrix}$$



Results

Parametric Orbit Problem

Given

- $x, y \in \mathbb{Q}(z_1, \ldots, z_k)^d$,
- $M \in \mathbb{Q}(z_1, \ldots, z_k)^{d \times d}$,

do there exist $\vec{a} = (a_1, \ldots, a_k) \in \mathbb{R}^k$ and $n \in \mathbb{N}$ such that

$$y(\vec{a}\,) = M(\vec{a}\,)^n x(\vec{a}\,)?$$

Parametric Orbit Problem

Given

- $x, y \in \mathbb{Q}(z_1, \ldots, z_k)^d$,
- $M \in \mathbb{Q}(z_1, \ldots, z_k)^{d \times d}$,

do there exist $\vec{a} = (a_1, \ldots, a_k) \in \mathbb{R}^k$ and $n \in \mathbb{N}$ such that

$$y(\vec{a}\,) = M(\vec{a}\,)^n x(\vec{a}\,)?$$

Theorem

- *Univariate systems ($k = 1$): Problem is decidable.*

Results

Parametric Orbit Problem

Given

- $x, y \in \mathbb{Q}(z_1, \ldots, z_k)^d$,
- $M \in \mathbb{Q}(z_1, \ldots, z_k)^{d \times d}$,

do there exist $\vec{a} = (a_1, \ldots, a_k) \in \mathbb{R}^k$ and $n \in \mathbb{N}$ such that

$$y(\vec{a}\,) = M(\vec{a}\,)^n x(\vec{a}\,)?$$

Theorem

- *Univariate systems ($k = 1$): Problem is decidable.*
- *Multivariate systems ($k \geq 2$): Problem is* ***Skolem hard****.*

Thank you!

Talk based on

- C. Baier, F. Funke, S. Jantsch, T. Karimov, E. Lefaucheux, F. Luca, J. Ouaknine, D. Purser, M.A. Whiteland, and J. Worrell:
  The Orbit Problem for Parametric Linear Dynamical Systems
  CONCUR 2021, #28, doi:10.4230/LIPIcs.CONCUR.2021.28
- T. Karimov, E. Lefaucheux, J. Ouaknine, D. Purser, A. Varonka, M.A. Whiteland, J. Worrell:
  What's Decidable about Linear Loops?
  POPL 2022, #65, doi:10.1145/3498727

The Bivariate Orbit Problem is hard!

$$\exists n \in \mathbb{N},\ a_1, a_2 \in \mathbb{R} : M(a_1, a_2)^n\, x(a_1, a_2) = y(a_1, a_2)$$

Proposition

Skolem-5 reduces to the Parametric Orbit Problem (with two variables).

The Bivariate Orbit Problem is hard!

$$\exists n \in \mathbb{N},\ a_1, a_2 \in \mathbb{R} : M(a_1, a_2)^n\, x(a_1, a_2) = y(a_1, a_2)$$

Proposition

Skolem-5 reduces to the Parametric Orbit Problem (with two variables).

Proof sketch.

$\vec{u} M^n \vec{x} = 0?$

The Bivariate Orbit Problem is hard!

$\exists n \in \mathbb{N},\ a_1, a_2 \in \mathbb{R} : M(a_1, a_2)^n\, x(a_1, a_2) = y(a_1, a_2)$

Proposition

Skolem-5 reduces to the Parametric Orbit Problem (with two variables).

Proof sketch.

$\vec{u} M^n \vec{x} = a\lambda^n + \overline{a\lambda^n} + b\gamma^n + \overline{b\gamma^n} + c$ with $1 \leq |\lambda| = |\gamma|$.

The Bivariate Orbit Problem is hard!

$$\exists n \in \mathbb{N},\ a_1, a_2 \in \mathbb{R} : M(a_1, a_2)^n\, x(a_1, a_2) = y(a_1, a_2)$$

Proposition

Skolem-5 reduces to the Parametric Orbit Problem (with two variables).

Proof sketch.

$\vec{u}M^n\vec{x} = A\Re(\lambda^n) + B\Im(\lambda^n) + C\Re(\gamma^n) + D\Im(\gamma^n) + E$ with $1 \leq |\lambda| = |\gamma|$.

The Bivariate Orbit Problem is hard!

$$\exists n \in \mathbb{N},\ a_1, a_2 \in \mathbb{R} : M(a_1, a_2)^n\, x(a_1, a_2) = y(a_1, a_2)$$

Proposition

Skolem-5 reduces to the Parametric Orbit Problem (with two variables).

Proof sketch.

$\vec{u}M^n\vec{x} = A\mathfrak{R}(\lambda^n) + B\mathfrak{I}(\lambda^n) + C\mathfrak{R}(\gamma^n) + D\mathfrak{I}(\gamma^n) + E$ with $1 \le |\lambda| = |\gamma|$.

- Construct **constant** matrix N and initial vector $\vec{x}'$.

The Bivariate Orbit Problem is hard!

$$\exists n \in \mathbb{N},\ a_1, a_2 \in \mathbb{R} : M(a_1, a_2)^n\, x(a_1, a_2) = y(a_1, a_2)$$

Proposition

Skolem-5 reduces to the Parametric Orbit Problem (with two variables).

Proof sketch.

$\vec{u}M^n\vec{x} = A\Re(\lambda^n) + B\Im(\lambda^n) + C\Re(\gamma^n) + D\Im(\gamma^n) + E$ with $1 \leq |\lambda| = |\gamma|$.

- Construct **constant** matrix N and initial vector $\vec{x}'$.
- $2D$ algebraic target set in $\mathbb{C}^4$; T: $\begin{cases} A\mathtt{x} + B\mathtt{y} + C\mathtt{z} + D\mathtt{w} + E = 0 \\ \mathtt{x}^2 + \mathtt{y}^2 = \mathtt{z}^2 + \mathtt{w}^2 \end{cases}$

The Bivariate Orbit Problem is hard!

$\exists n \in \mathbb{N},\ a_1, a_2 \in \mathbb{R} : M(a_1, a_2)^n\, x(a_1, a_2) = y(a_1, a_2)$

Proposition

Skolem-5 reduces to the Parametric Orbit Problem (with two variables).

Proof sketch.

$\vec{u}M^n\vec{x} = A\mathfrak{R}(\lambda^n) + B\mathfrak{I}(\lambda^n) + C\mathfrak{R}(\gamma^n) + D\mathfrak{I}(\gamma^n) + E$ with $1 \leq |\lambda| = |\gamma|$.

- Construct **constant** matrix N and initial vector $\vec{x}'$.
- $2D$ algebraic target set in $\mathbb{C}^4$; T: $\begin{cases} A\mathtt{x} + B\mathtt{y} + C\mathtt{z} + D\mathtt{w} + E = 0 \\ \mathtt{x}^2 + \mathtt{y}^2 = \mathtt{z}^2 + \mathtt{w}^2 \end{cases}$
  - the graph of a **target vector**: $T = \{y(a_1, a_2) : a_i \in \mathbb{R}\}$

The Bivariate Orbit Problem is hard!

$$\exists n \in \mathbb{N},\ a_1, a_2 \in \mathbb{R} : M(a_1, a_2)^n\, x(a_1, a_2) = y(a_1, a_2)$$

Proposition

Skolem-5 reduces to the Parametric Orbit Problem (with two variables).

Proof sketch.

$\vec{u}M^n\vec{x} = A\Re(\lambda^n) + B\Im(\lambda^n) + C\Re(\gamma^n) + D\Im(\gamma^n) + E$ with $1 \leq |\lambda| = |\gamma|$.

- Construct **constant** matrix N and initial vector $\vec{x}'$.
- $2D$ algebraic target set in $\mathbb{C}^4$; T: $\begin{cases} A\mathtt{x} + B\mathtt{y} + C\mathtt{z} + D\mathtt{w} + E = 0 \\ \mathtt{x}^2 + \mathtt{y}^2 = \mathtt{z}^2 + \mathtt{w}^2 \end{cases}$
    - the graph of a **target vector**: $T = \{y(a_1, a_2) : a_i \in \mathbb{R}\}$
- Then $\vec{u}M^n\vec{x} = 0$ if and only if $\exists a_1, a_2 \in \mathbb{R} : N^n\vec{x}' = y(a_1, a_2)$. $\square$