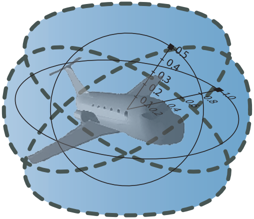
15-819/18-879: Hybrid Systems Analysis & Theorem Proving

03: Numerical versus Symbolic Analysis

André Platzer

aplatzer@cs.cmu.edu
Carnegie Mellon University, Pittsburgh, PA



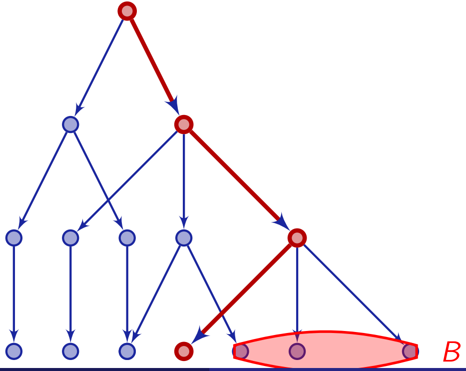
Outline

Outline

Model Checking in a Nutshell

Definition (Model Checking Problem)

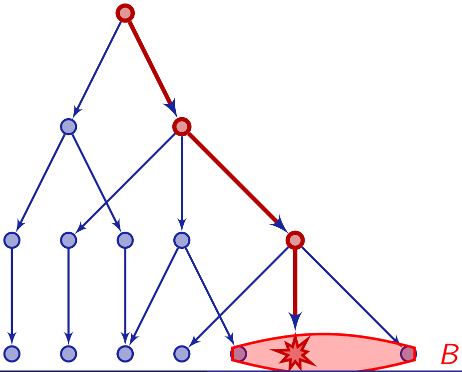
Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.



Model Checking in a Nutshell

Definition (Model Checking Problem)

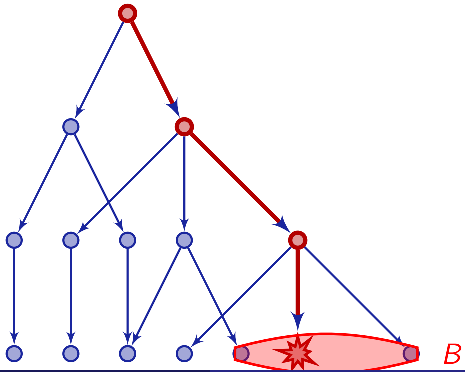
Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.



Model Checking in a Nutshell

Definition (Image Computation)

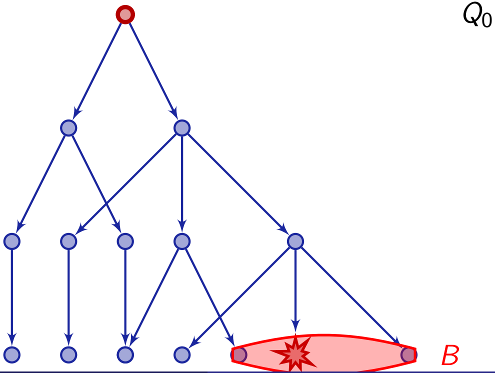
$$Post_A(Y) \;:=\; \{q^+ \in Q \;:\; q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Model Checking in a Nutshell

Definition (Image Computation)

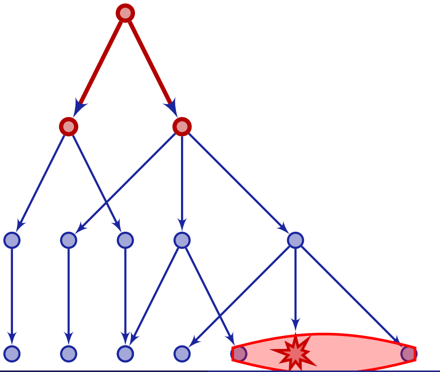
$$Post_A(Y) := \{q^+ \in Q \ : \ q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Model Checking in a Nutshell

Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Q_0

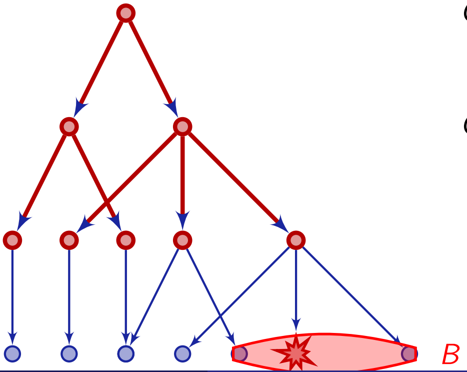
$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

Model Checking in a Nutshell

Definition (Image Computation)

$$Post_A(Y) \;:=\; \{q^+ \in Q \;:\; q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$

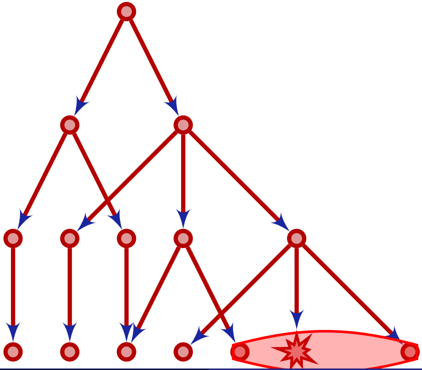


$$\begin{array}{l} Q_0 \\ \quad\downarrow Post_A(Q_0) \\ Q_1 = Post_A(Q_0) \\ \quad\downarrow Post_A(Q_1) \\ Q_2 = Post_A^2(Q_0) \end{array}$$

Model Checking in a Nutshell

Definition (Image Computation)

$$Post_A(Y) \;:=\; \{q^+ \in Q \;:\; q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$

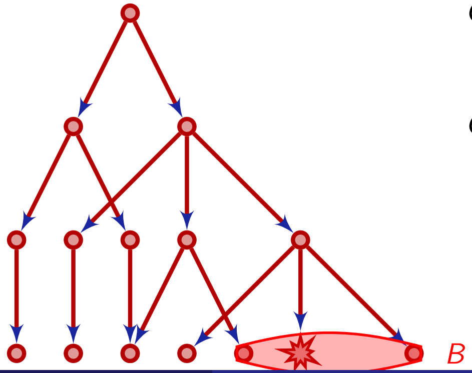


$$Q_0 \xrightarrow{Post_A(Q_0)} Q_1 = Post_A(Q_0) \xrightarrow{Post_A(Q_1)} Q_2 = Post_A^2(Q_0) \xrightarrow{Post_A(Q_2)} Q_3 = Post_A^3(Q_0)$$

Model Checking in a Nutshell

Definition (Image Computation)

$$
\begin{aligned}
Post_A(Y) &:= \{q^+ \in Q \;:\; q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\} \\
Post_A^*(Y) &:= \mu Z.(Y \cup Z \cup Post_A(Z))
\end{aligned}
$$



Q_0

$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

$Post_A(Q_1)$

$Q_2 = Post_A^2(Q_0)$

$Post_A(Q_2)$

$Q_3 = Post_A^3(Q_0)$

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Can we use this for hybrid systems?

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure.

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure. Faster algorithms exist with OBDD, BMC, ...

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure. Faster algorithms exist with OBDD, BMC, ...

Proposition (Semidecision)

For (computable) countably infinite-state systems, naïve MC gives a (slow) semidecision procedure.

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure. Faster algorithms exist with OBDD, BMC, ...

Proposition (Semidecision)

For (computable) countably infinite-state systems, naïve MC gives a (slow) semidecision procedure. Faster algorithms depend on problem

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure. Faster algorithms exist with OBDD, BMC, ...

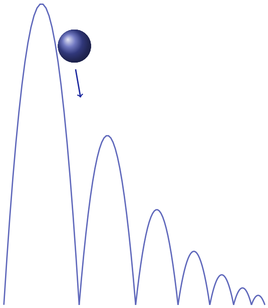
Proposition (Semidecision)

For (computable) countably infinite-state systems, naïve MC gives a (slow) semidecision procedure. Faster algorithms depend on problem

Hybrid systems have uncountable state spaces

(Uncountably) infinite state spaces require extra care

PHAVer Tool Example



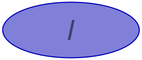
PHAVer Tool Example

```
g:=1; // constant for gravity
automaton bouncing_ball
contr_var: x, v;
synclabs: jump;
loc state:
 while x>=0 & x<=10 & v<=10 & v>=-10 wait {x'==v&v'==-g};
 // transitions
 when x==0&v<0 sync jump do {v'==-v*0.5&x'==x} goto state;
 when x==0&-0.1<v&v<0.1 sync jump do {v'==v&x'==x} goto frz
loc frz:
    while x>=0 & x<=10 & v<=10 & v>=-10 wait {x'==0&v'==0};
initially: state & x==2 & v==0;
end

reg=bouncing_ball.reachable;
// reg=bouncing_ball.is_reachable(frz);
reg.print("out_reach",2); // output reachable set
```

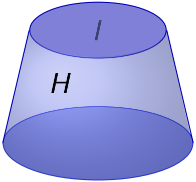
What analysis is doable at all?

Image Computation in Hybrid Systems



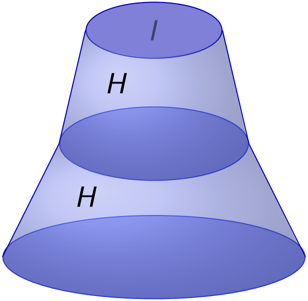
- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

Image Computation in Hybrid Systems



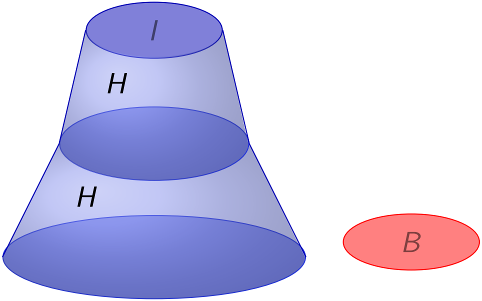
- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

Image Computation in Hybrid Systems



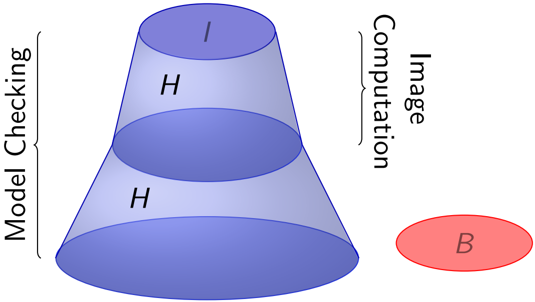
- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

Image Computation in Hybrid Systems



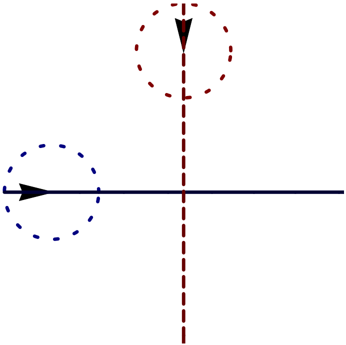
- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

Image Computation in Hybrid Systems

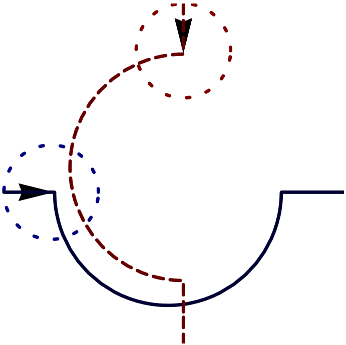


- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

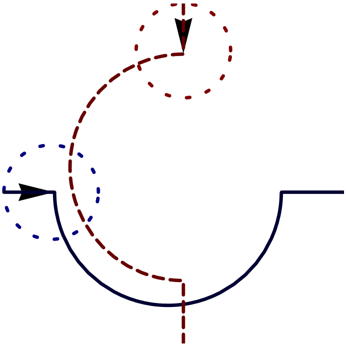
Air Traffic Management

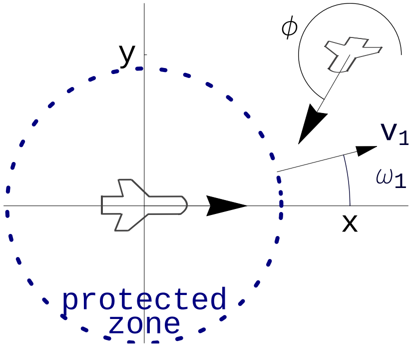


Air Traffic Management

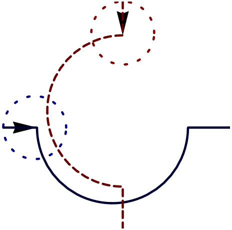


Air Traffic Management

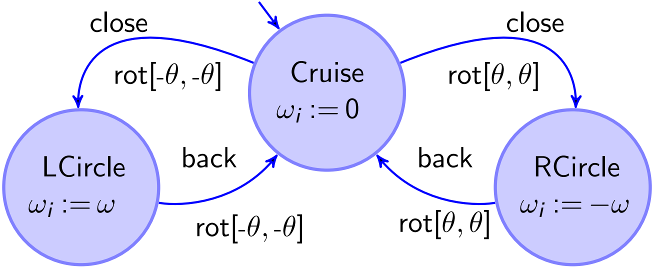




ATM: Roundabout Maneuver Automaton



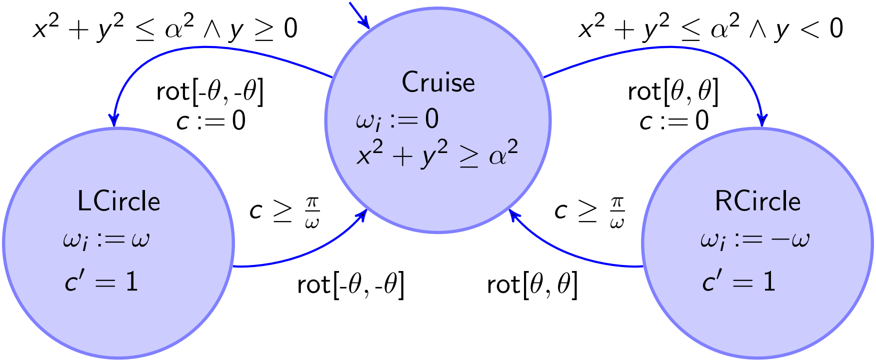
$$\begin{bmatrix} x' & = -v_1 & +v_2\cos\phi & +\omega_1 y \\ y' & = & v_2\sin\phi & -\omega_1 x \\ \phi' & = & \omega_2 & -\omega_1 \end{bmatrix}$$



Details

ATM: Roundabout Maneuver Automaton

$$\begin{bmatrix} x' = -v_1 + v_2 \cos\phi + \omega_1 y \\ y' = v_2 \sin\phi - \omega_1 x \\ \phi' = \omega_2 - \omega_1 \end{bmatrix}$$



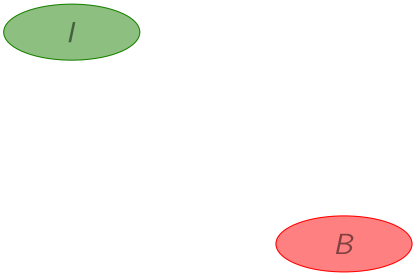
Return

Outline

AMC: Approximation Refinement Model Checking

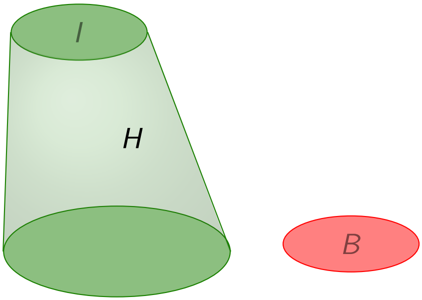
AMC(B reachable from I in H):

1. $A :=$ approx(H) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check(B reachable from I in $A + \epsilon$)
4. B not reachable $\Rightarrow$ H safe



AMC(B reachable from I in H):

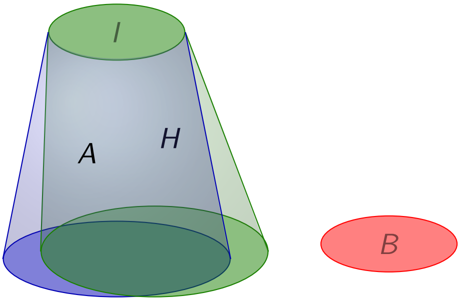
1. $A :=$ approx(H) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check(B reachable from I in $A + \epsilon$)
4. B not reachable $\Rightarrow$ H safe



AMC: Approximation Refinement Model Checking

AMC(B reachable from I in H):

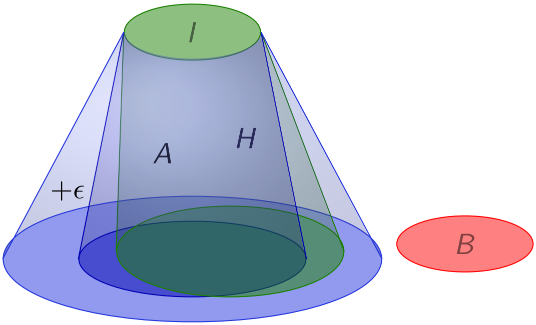
1. $A :=$ approx(H) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check(B reachable from I in $A + \epsilon$)
4. B not reachable $\Rightarrow$ H safe



AMC: Approximation Refinement Model Checking

AMC(B reachable from I in H):

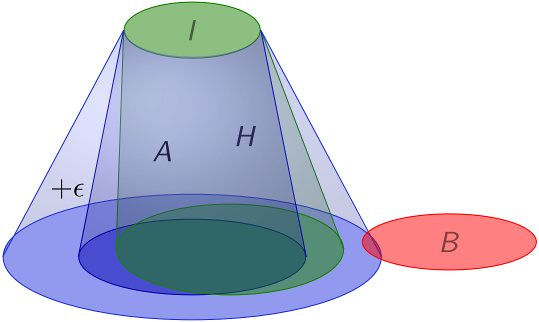
1. $A :=$ approx(H) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check(B reachable from I in $A + \epsilon$)
4. B not reachable $\Rightarrow$ H safe



AMC: Approximation Refinement Model Checking

AMC(B reachable from I in H):

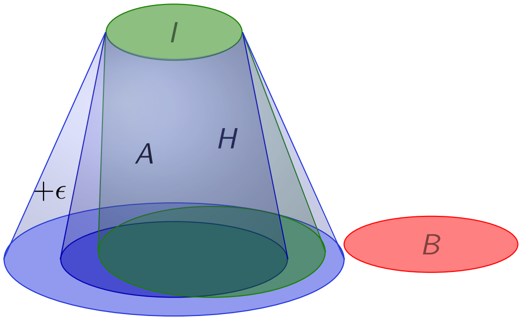
1. $A :=$ approx(H) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check(B reachable from I in $A + \epsilon$)
4. B not reachable $\Rightarrow$ H safe



AMC: Approximation Refinement Model Checking

AMC(B reachable from I in H):

1. $A :=$ approx(H) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check(B reachable from I in $A + \epsilon$)
4. B not reachable $\Rightarrow$ H safe



AMC: Exact Image Computation

AMC(B reachable from I in H):

1. $A :=$ approx(H) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check(B reachable from I in $A+\epsilon$)
4. B not reachable $\Rightarrow$ H safe

Proposition

check and blur can be implemented for

- *I and B semialgebraic (propositional combinations of $p \geq 0$)*
- *A with polynomial flows over $\mathbb{R}$*
- *+Piecewise definitions*
- *+Rational extensions (e.g. multivariate rational splines)*

AMC: Image Approximation

AMC(B reachable from I in H):

1. $A :=$ approx(H) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check(B reachable from I in $A + \epsilon$)
4. B not reachable $\Rightarrow$ H safe

Proposition

approx exists for all uniform errors $\epsilon > 0$ when

- *using polynomials to build A*
- *Flows $\varphi \in C(D, \mathbb{R}^n)$ of H*
- *$D \subset \mathbb{R} \times \mathbb{R}^n$ compact closure of an open set*

Approximation can solve problems without effective exact solution

Weierstraßian Flow Approximation

Proposition

approx exists for all uniform errors $\varepsilon > 0$:

- $\varphi \in C(D, \mathbb{R}^n)$ *on compact closure* $D \subset \mathbb{R} \times \mathbb{R}^n$ *of an open set*

$\Rightarrow$ $\forall \varepsilon > 0 \ \exists p \in \mathbb{R}[t, x_1, \ldots, x_n]^n \, \forall Y \subseteq \mathbb{R}^n$

$$Post_{\varphi|_D}(Y) \ \subseteq \ \mathcal{U}_\varepsilon(Post_{p|_D}(Y))$$

Weierstraßian Flow Approximation

Proposition

approx exists for all uniform errors $\varepsilon > 0$*:*

- $\varphi \in C(D, \mathbb{R}^n)$ *on compact closure* $D \subset \mathbb{R} \times \mathbb{R}^n$ *of an open set*

$\Rightarrow$ $\forall \varepsilon > 0 \ \exists p \in \mathbb{R}[t, x_1, \ldots, x_n]^n \, \forall Y \subseteq \mathbb{R}^n$

$$Post_{\varphi|_D}(Y) \ \subseteq \ \mathcal{U}_\varepsilon(Post_{p|_D}(Y))$$

Where $\mathcal{U}_\varepsilon(Y)$ is the ε ball around set Y:

$$\mathcal{U}_\varepsilon(Y) := \{x \ : \ \|x - y\| < \varepsilon \text{ for some } y \in Y\}$$

Weierstraßian Flow Approximation

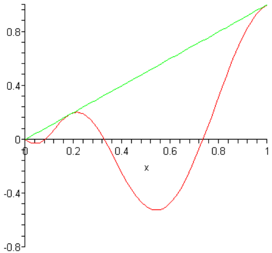
Theorem (Stone-Weierstraß Approximation)

Polynomials uniformly approximate cont. functions on compact domains:

- $\varphi \in C(D, \mathbb{R}^n)$ *on compact domain* $D \subset \mathbb{R} \times \mathbb{R}^n$

$\Rightarrow$ $\forall \varepsilon > 0 \ \exists p \in \mathbb{R}[t, x_1, \dots, x_n]^n \, \forall (t, x) \in D$

$$\|\varphi(t; x) - p(t, x)\| < \varepsilon$$



Weierstraßian Flow Approximation

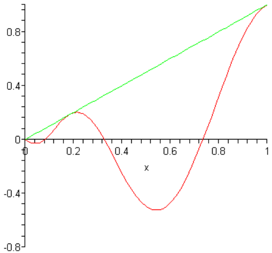
Theorem (Stone-Weierstraß Approximation)

Polynomials uniformly approximate cont. functions on compact domains:

- $\varphi \in C(D, \mathbb{R}^n)$ *on compact domain* $D \subset \mathbb{R} \times \mathbb{R}^n$

$\Rightarrow$ $\forall \varepsilon > 0 \ \exists p \in \mathbb{R}[t, x_1, \dots, x_n]^n \, \forall (t, x) \in D$

$$\|\varphi(t; x) - p(t, x)\| < \varepsilon$$



Existence of solutions may be computationally insufficient

Exact Image Computation: Polynomials

Proposition

check and blur can be implemented for

- I, D, B *definable in* $\mathrm{FOL}_{\mathbb{R}}$, *i.e., semialgebraic*
- A *with polynomial flows over* $\mathbb{R}$

Exact Image Computation: Polynomials

Proposition

check and *blur* can be implemented for

- I, D, B definable in $\mathrm{FOL}_{\mathbb{R}}$, i.e., semialgebraic
- A with polynomial flows over $\mathbb{R}$

Proof.

Inductive consequence of $\mathcal{U}_{\varepsilon}(Post_{p|_D}(Y))$ being definable in $\mathrm{FOL}_{\mathbb{R}}$, thus being decidable: Let Y, D be defined by $\mathrm{FOL}_{\mathbb{R}}$ formulas F_Y, F_D.

Exact Image Computation: Polynomials

Proposition

check and *blur* can be implemented for

- I, D, B definable in $\mathrm{FOL}_{\mathbb{R}}$, *i.e., semialgebraic*
- A *with polynomial flows over* $\mathbb{R}$

Proof.

Inductive consequence of $\mathcal{U}_\varepsilon(Post_{p|_D}(Y))$ being definable in $\mathrm{FOL}_{\mathbb{R}}$, thus being decidable: Let Y, D be defined by $\mathrm{FOL}_{\mathbb{R}}$ formulas F_Y, F_D.

1. "$z \in Post_{p|_D}(Y)$" is definable as:

$$\exists x\, \exists t{\geq}0\, (F_Y(x) \wedge \forall 0{\leq}s{\leq}t\, F_D(s, p(s,x)) \wedge z = p(t,x))$$

Exact Image Computation: Polynomials

Proposition

check and *blur* can be implemented for

- I, D, B definable in $\mathrm{FOL}_{\mathbb{R}}$, i.e., semialgebraic
- A with polynomial flows over $\mathbb{R}$

Proof.

Inductive consequence of $\mathcal{U}_\varepsilon(Post_{p|_D}(Y))$ being definable in $\mathrm{FOL}_{\mathbb{R}}$, thus being decidable: Let Y, D be defined by $\mathrm{FOL}_{\mathbb{R}}$ formulas F_Y, F_D.

1. "$z \in Post_{p|_D}(Y)$" is definable as:

$$\exists x\, \exists t{\geq}0\, (F_Y(x) \land \forall 0{\leq}s{\leq}t\, F_D(s, p(s,x)) \land z = p(t,x))$$

2. "$z \in \mathcal{U}_\varepsilon(Y)$" is definable in $\mathrm{FOL}_{\mathbb{R}}$, thus decidable:

$$\exists y\, (F_Y y \land \sum_{i=1}^{n} (y_i - z_i)^2 < \varepsilon^2)$$

Exact Image Computation: Piecewise Polynomials

Proposition

check and *blur* can be implemented for

- I, D, B definable in $\mathrm{FOL}_{\mathbb{R}}$, *i.e., semialgebraic*
- A *with piecewise polynomial flows over* $\mathbb{R}$

Exact Image Computation: Piecewise Polynomials

Proposition

check and *blur* can be implemented for

- I, D, B definable in $\mathrm{FOL}_{\mathbb{R}}$, *i.e., semialgebraic*
- *A with piecewise polynomial flows over* $\mathbb{R}$

Proof.

$s : D \longrightarrow \mathbb{R}$ consists of polynomial pieces $p_i : D_i \longrightarrow \mathbb{R}$ for disjoint definable D_i with $D = D_1 \cup \ldots \cup D_n$. Then, we define $\mathcal{U}_\varepsilon(Post_{s|_D}(Y))$:

Exact Image Computation: Piecewise Polynomials

Proposition

check and *blur* can be implemented for

- I, D, B definable in $\mathrm{FOL}_{\mathbb{R}}$, i.e., semialgebraic
- A with *piecewise* polynomial flows over $\mathbb{R}$

Proof.

$s : D \to \mathbb{R}$ consists of polynomial pieces $p_i : D_i \to \mathbb{R}$ for disjoint definable D_i with $D = D_1 \cup \ldots \cup D_n$. Then, we define $\mathcal{U}_\varepsilon(Post_{s|_D}(Y))$:

1. "$z = s(x)$" is definable:

$$\bigvee_{i=1}^{n} (x \in D_i \wedge p_i(x) = t)$$

Exact Image Computation: Piecewise Polynomials

Proposition

check and *blur* can be implemented for

- I, D, B definable in $\mathrm{FOL}_{\mathbb{R}}$, i.e., semialgebraic
- A with *piecewise* polynomial flows over $\mathbb{R}$

Proof.

$s : D \to \mathbb{R}$ consists of polynomial pieces $p_i : D_i \to \mathbb{R}$ for disjoint definable D_i with $D = D_1 \cup \ldots \cup D_n$. Then, we define $\mathcal{U}_\varepsilon(Post_{s|_D}(Y))$:

1. "$z = s(x)$" is definable:
$$\bigvee_{i=1}^{n} (x \in D_i \land p_i(x) = t)$$
2. Decompose image computation using:
$$Post_{s|_D}(Y) = \bigcup_{i=1}^{n} Post_{p_i|_{D_i}}(Y) \quad \text{and} \quad \mathcal{U}_\varepsilon(X \cup Y) = \mathcal{U}_\varepsilon(X) \cup \mathcal{U}_\varepsilon(Y)$$

Exact Image Computation: Rational Functions

Proposition

check and *blur* can be implemented for

- I, D, B *definable in* $\mathrm{FOL}_{\mathbb{R}}$*, i.e., semialgebraic*
- A *with* *rational* *flows over* $\mathbb{R}$

Exact Image Computation: Rational Functions

Proposition

check and *blur* can be implemented for

- I, D, B definable in $\mathrm{FOL}_{\mathbb{R}}$, i.e., semialgebraic
- A with *rational* flows over $\mathbb{R}$

Proposition (Rational Tarski)

Tarski's theorem extends to rational functions.

Exact Image Computation: Rational Functions

Proposition

check and blur can be implemented for

- I, D, B *definable in* $\mathrm{FOL}_{\mathbb{R}}$, *i.e., semialgebraic*
- A *with rational flows over* $\mathbb{R}$

Proposition (Rational Tarski)

Tarski's theorem extends to rational functions.

Proof.

Repeatedly remove rational expressions (using field of fractions form):

$$p(x)/q(x) = 0 \quad \equiv \quad p(x) = 0 \land q(x) \neq 0$$

$\square$

Exact Image Computation: Rational Functions

Proposition

check and *blur* can be implemented for

- I, D, B *definable in* $\mathrm{FOL}_{\mathbb{R}}$, *i.e., semialgebraic*
- A *with rational flows over* $\mathbb{R}$

Proposition (Rational Tarski)

Tarski's theorem extends to rational functions.

Proof.

Repeatedly remove rational expressions (using field of fractions form):

$$
\begin{aligned}
p(x)/q(x) = 0 &\equiv p(x) = 0 \land q(x) \neq 0 \\
p(x)/q(x) > 0 &\equiv (p(x) > 0 \land q(x) > 0) \lor (p(x) < 0 \land q(x) < 0)
\end{aligned}
$$

□

Logical foundation for effective image computation operations

Outline

Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective, D effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

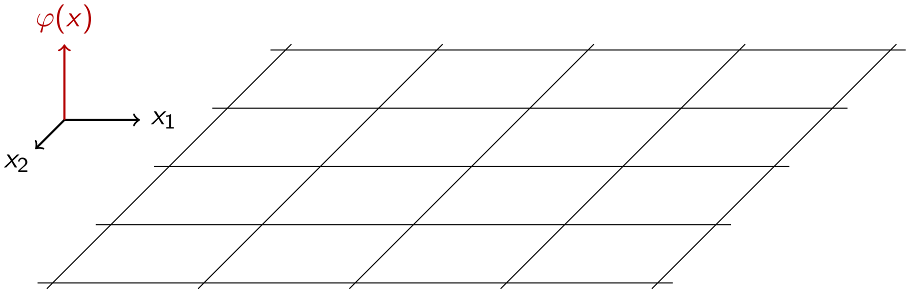
$\Rightarrow$ *approx computable, hence image computation decidable*

Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective, D effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*

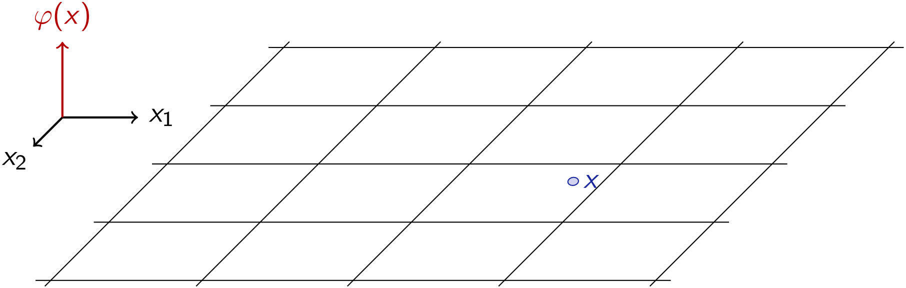


Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective,* D *effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*

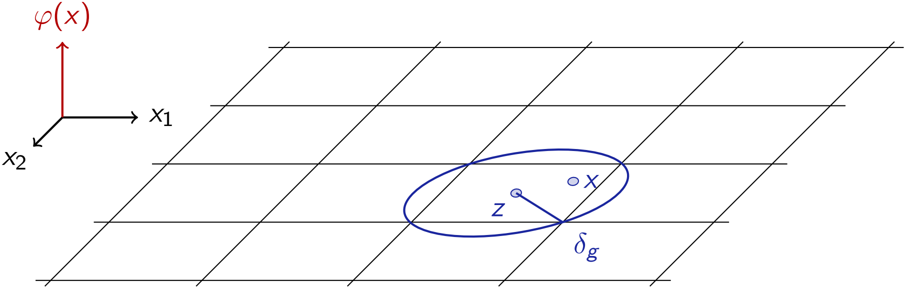


Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective,* D *effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*

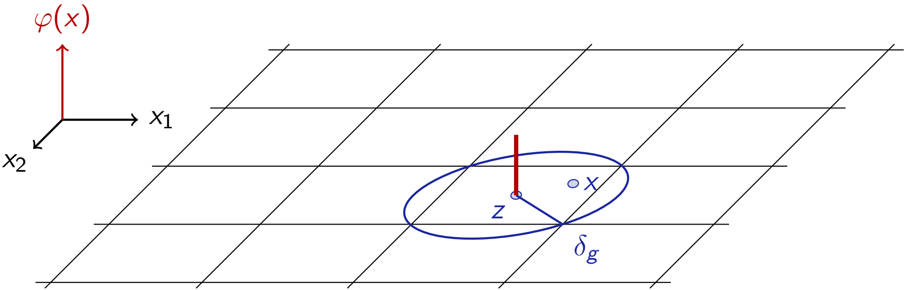


Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective,* D *effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*

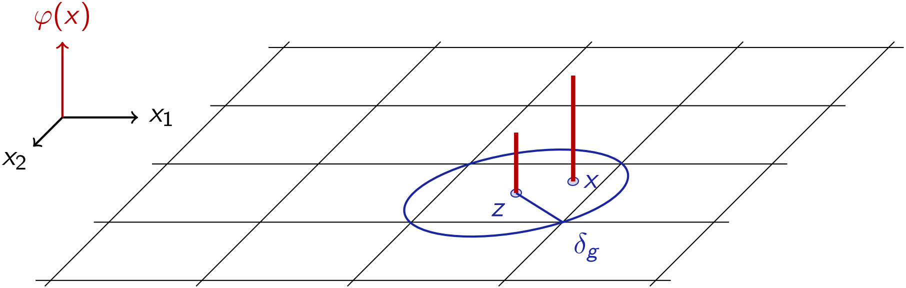


Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective,* D *effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

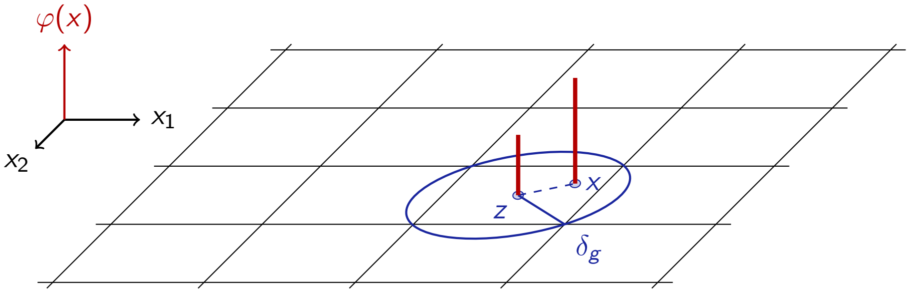
$\Rightarrow$ *approx computable, hence image computation decidable*



Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective,* D *effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

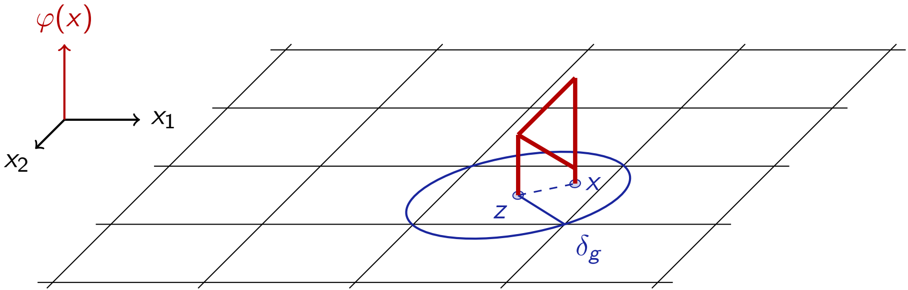
$\Rightarrow$ *approx computable, hence image computation decidable*



Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective,* D *effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*

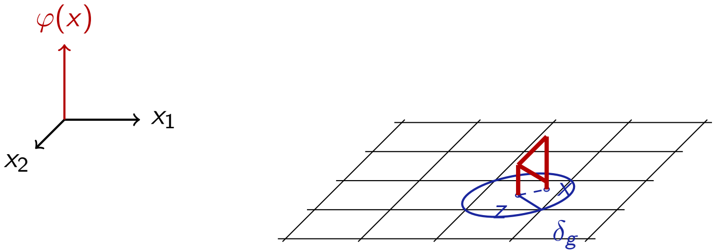


Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective,* D *effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*

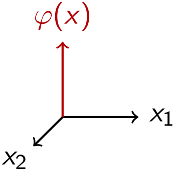


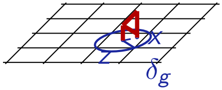
Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective, D effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*



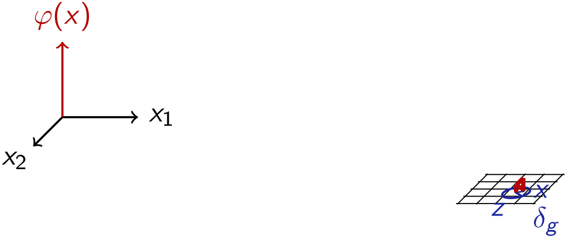


Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective,* D *effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*



Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.
- φ arbitrarily effective, i.e., $\forall \delta_c > 0 \; \exists f_{\delta_c} : D \to \mathbb{R}^1$ effective such that $\forall y \in D \; \|\varphi(y) - f_{\delta_c}(y)\| < \delta_c$.

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.
- φ arbitrarily effective, i.e., $\forall \delta_c > 0 \ \exists f_{\delta_c} : D \to \mathbb{R}^1$ effective such that $\forall y \in D \ \ \|\varphi(y) - f_{\delta_c}(y)\| < \delta_c$.
- Let $z \in D$ be a point on a δ_g-grid with distance $\|x - z\| < \delta_g$.

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.
- φ arbitrarily effective, i.e., $\forall \delta_c > 0 \ \exists f_{\delta_c} : D \to \mathbb{R}^1$ effective such that $\forall y \in D \ \ \|\varphi(y) - f_{\delta_c}(y)\| < \delta_c$.
- Let $z \in D$ be a point on a δ_g-grid with distance $\|x - z\| < \delta_g$.
- Assume D convex on grid cell. Thus by MVT $\exists \xi \in S[x, z]$

$$\|\varphi(x) - \varphi(z)\| = \|\varphi'(\xi)(x - z)\| = \|\varphi'(\xi)\| \cdot \|(x - z)\| < b\delta_g$$

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.
- φ arbitrarily effective, i.e., $\forall \delta_c > 0 \ \exists f_{\delta_c} : D \to \mathbb{R}^1$ effective such that $\forall y \in D \ \ \|\varphi(y) - f_{\delta_c}(y)\| < \delta_c$.
- Let $z \in D$ be a point on a δ_g-grid with distance $\|x - z\| < \delta_g$.
- Assume D convex on grid cell. Thus by MVT $\exists \xi \in S[x, z]$

$$\|\varphi(x) - \varphi(z)\| = \|\varphi'(\xi)(x - z)\| = \|\varphi'(\xi)\| \cdot \|(x - z)\| < b\delta_g$$

- φ arbitrarily effective at grid point z, hence

$$\|\varphi(x) - f_{\delta_c}(z)\| \leq \|\varphi(x) - \varphi(z)\| + \|\varphi(z) - f_{\delta_c}(z)\| < b\delta_g + \delta_c$$

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.
- φ arbitrarily effective, i.e., $\forall \delta_c > 0 \ \exists f_{\delta_c} : D \to \mathbb{R}^1$ effective such that $\forall y \in D \ \ \|\varphi(y) - f_{\delta_c}(y)\| < \delta_c$.
- Let $z \in D$ be a point on a δ_g-grid with distance $\|x - z\| < \delta_g$.
- Assume D convex on grid cell. Thus by MVT $\exists \xi \in S[x, z]$

$$\|\varphi(x) - \varphi(z)\| = \|\varphi'(\xi)(x - z)\| = \|\varphi'(\xi)\| \cdot \|(x - z)\| < b\delta_g$$

- φ arbitrarily effective at grid point z, hence

$$\|\varphi(x) - f_{\delta_c}(z)\| \leq \|\varphi(x) - \varphi(z)\| + \|\varphi(z) - f_{\delta_c}(z)\| < b\delta_g + \delta_c$$

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.
- φ arbitrarily effective, i.e., $\forall \delta_c > 0 \ \exists f_{\delta_c} : D \to \mathbb{R}^1$ effective such that $\forall y \in D \ \ \|\varphi(y) - f_{\delta_c}(y)\| < \delta_c$.
- Let $z \in D$ be a point on a δ_g-grid with distance $\|x - z\| < \delta_g$.
- Assume D convex on grid cell. Thus by MVT $\exists \xi \in S[x, z]$

$$\|\varphi(x) - \varphi(z)\| = \|\varphi'(\xi)(x - z)\| = \|\varphi'(\xi)\| \cdot \|(x - z)\| < b\delta_g$$

- φ arbitrarily effective at grid point z, hence

$$\|\varphi(x) - f_{\delta_c}(z)\| \leq \|\varphi(x) - \varphi(z)\| + \|\varphi(z) - f_{\delta_c}(z)\| < b\delta_g + \delta_c$$

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.
- φ arbitrarily effective, i.e., $\forall \delta_c > 0 \;\exists f_{\delta_c} : D \to \mathbb{R}^1$ effective such that $\forall y \in D \;\; \|\varphi(y) - f_{\delta_c}(y)\| < \delta_c$.
- Let $z \in D$ be a point on a δ_g-grid with distance $\|x - z\| < \delta_g$.
- Assume D convex on grid cell. Thus by MVT $\exists \xi \in S[x, z]$

$$\|\varphi(x) - \varphi(z)\| = \|\varphi'(\xi)(x - z)\| = \|\varphi'(\xi)\| \cdot \|(x - z)\| < b\delta_g$$

- φ arbitrarily effective at grid point z, hence

$$\|\varphi(x) - f_{\delta_c}(z)\| \leq \|\varphi(x) - \varphi(z)\| + \|\varphi(z) - f_{\delta_c}(z)\| < b\delta_g + \delta_c \overset{!}{<} \epsilon$$

Bounded Flow Approximation: Proof

Proof.

- Working component-wise, assume range $\mathbb{R}^1$ for φ.
- Separately consider connected components of D. Let $\epsilon > 0, x \in D$.
- φ arbitrarily effective, i.e., $\forall \delta_c > 0\ \exists f_{\delta_c} : D \to \mathbb{R}^1$ effective such that $\forall y \in D\ \ \|\varphi(y) - f_{\delta_c}(y)\| < \delta_c$.
- Let $z \in D$ be a point on a δ_g-grid with distance $\|x - z\| < \delta_g$.
- Assume D convex on grid cell. Thus by MVT $\exists \xi \in S[x, z]$

$$\|\varphi(x) - \varphi(z)\| = \|\varphi'(\xi)(x - z)\| = \|\varphi'(\xi)\| \cdot \|(x - z)\| < b\delta_g$$

- φ arbitrarily effective at grid point z, hence

$$\|\varphi(x) - f_{\delta_c}(z)\| \leq \|\varphi(x) - \varphi(z)\| + \|\varphi(z) - f_{\delta_c}(z)\| < b\delta_g + \delta_c \overset{!}{<} \epsilon$$

- Approximate by step functions $f_{\delta_c}(z)$ on $\pm\delta_g/2$ hypercube around z.

Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective, D effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*

Bounded Flow Approximation

Proposition (Effective Weierstraß approximation)

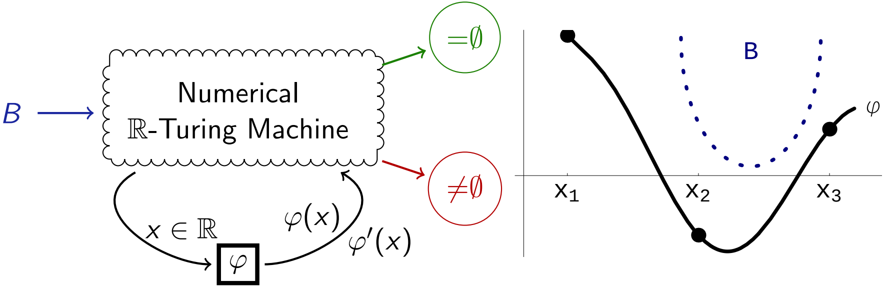
- *Flows* $\varphi \in C^1(D, \mathbb{R}^n)$ *arbitrarily effective, D effective*
- *Bounds* $b := \max_{x \in D} \|\varphi'(x)\|$

$\Rightarrow$ *approx computable, hence image computation decidable*

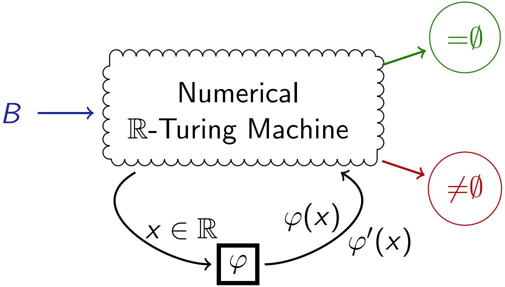
Only need to find the bound b . . .

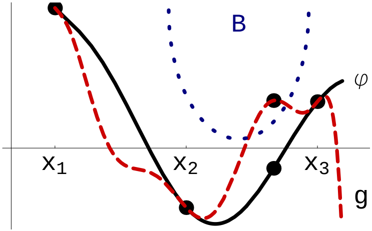
Finding bounds is easier than verification?

Continuous Image Computation

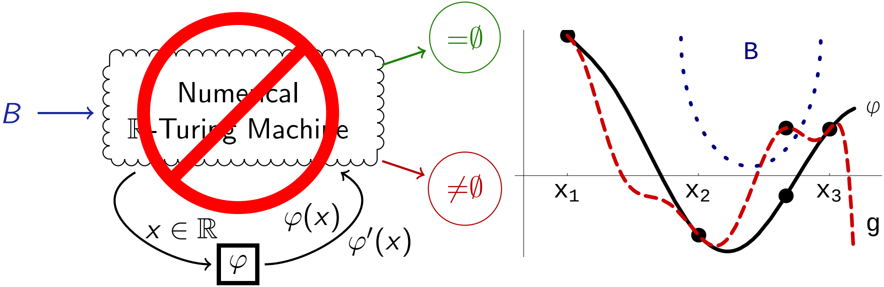


Continuous Image Computation





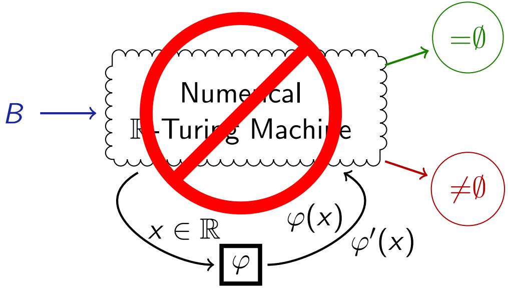
Continuous Image Computation

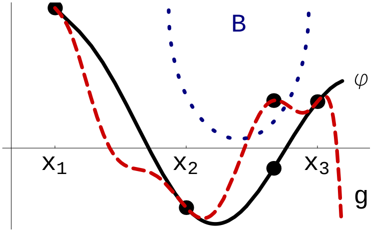


Proposition (Image computation undecidable for...)

- *arbitrarily effective flow* $\varphi \in C^k(D \subseteq \mathbb{R}^n, \mathbb{R}^m)$; *D, B effective*
- *tolerate error* $\epsilon > 0$ *in decisions*

Continuous Image Computation

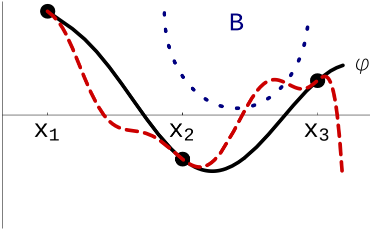




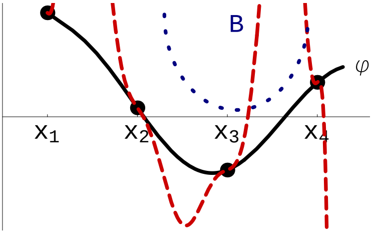
Proposition (Image computation undecidable for. . .)

- *arbitrarily effective flow* $\varphi \in C^k(D \subseteq \mathbb{R}^n, \mathbb{R}^m)$*; D, B effective*
- *tolerate error* $\epsilon > 0$ *in decisions*
- φ *smooth polynomial function with* $\mathbb{Q}$*-coefficients*

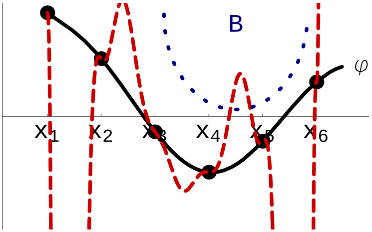
Probabilistic Model Checking



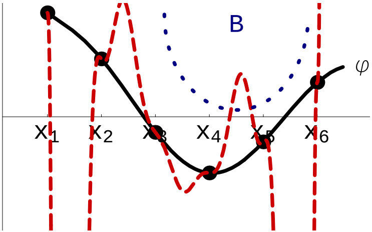
Probabilistic Model Checking



Probabilistic Model Checking



Probabilistic Model Checking

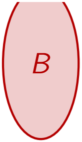


Proposition

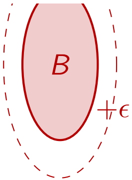
- $P(\|\varphi'\|_\infty > b) \to 0$ *as* $b \to \infty$
- φ *evaluated on finite subset* $X = \{x_i\}$ *of open or compact* D

$\Rightarrow$ $P(\mathit{decision\ correct}) \to 1$ *as* $\|d(\cdot, X)\|_\infty \to 0$

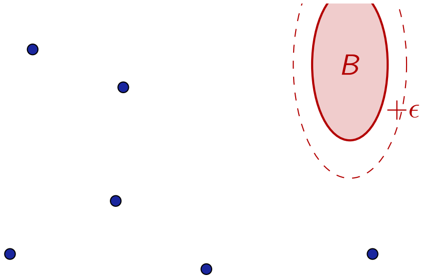
Probabilistic Model Checking: Proof Illustration



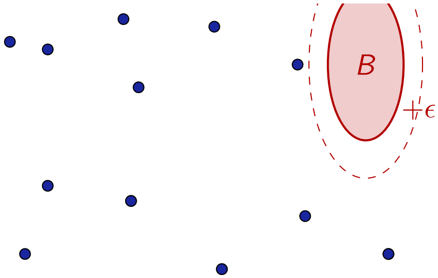
Probabilistic Model Checking: Proof Illustration



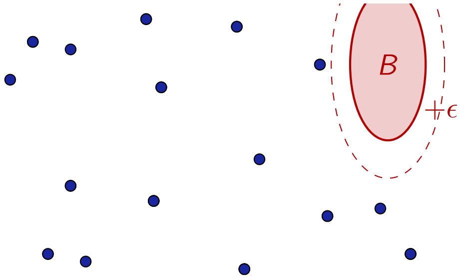
Probabilistic Model Checking: Proof Illustration



Probabilistic Model Checking: Proof Illustration



Probabilistic Model Checking: Proof Illustration



Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.

$\square$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.

$\square$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:

$\square$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \ \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.

$\square$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \ \ \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).

□

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \ \ \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).
- By MVT $\exists \xi$ between x and x_i

$$\|\varphi(x) - \varphi(x_i)\| = \|\varphi'(\xi)(x - x_i)\| = \|\varphi'(\xi)\| \cdot \|x - x_i\|$$

$\square$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \;\; \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).
- By MVT $\exists \xi$ between x and x_i

$$\| \underbrace{\varphi(x)}_{\in B} - \underbrace{\varphi(x_i)}_{\notin \mathcal{U}_\epsilon(B)} \| = \|\varphi'(\xi)(x - x_i)\| = \|\varphi'(\xi)\| \cdot \|x - x_i\|$$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \ \ \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).
- By MVT $\exists \xi$ between x and x_i

$$\epsilon \leq \| \underbrace{\varphi(x)}_{\in B} - \underbrace{\varphi(x_i)}_{\notin \mathcal{U}_\epsilon(B)} \| = \|\varphi'(\xi)(x - x_i)\| = \|\varphi'(\xi)\| \cdot \|x - x_i\|$$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \;\; \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).
- By MVT $\exists \xi$ between x and x_i

$$\epsilon \leq \| \underbrace{\varphi(x)}_{\in B} - \underbrace{\varphi(x_i)}_{\notin \mathcal{U}_\epsilon(B)} \| = \|\varphi'(\xi)(x - x_i)\| = \|\varphi'(\xi)\| \cdot \underbrace{\|x - x_i\|}_{\leq \nu}$$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \;\; \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).
- By MVT $\exists \xi$ between x and x_i

$$\epsilon \le \| \underbrace{\varphi(x)}_{\in B} - \underbrace{\varphi(x_i)}_{\notin \mathcal{U}_\epsilon(B)} \| = \|\varphi'(\xi)(x - x_i)\| = \|\varphi'(\xi)\| \cdot \underbrace{\|x - x_i\|}_{\le \nu}$$

$$\frac{\epsilon}{\nu} \le \|\varphi'(\xi)\|$$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \;\; \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).
- By MVT $\exists \xi$ between x and x_i

$$\epsilon \leq \| \underbrace{\varphi(x)}_{\in B} - \underbrace{\varphi(x_i)}_{\notin \mathcal{U}_\epsilon(B)} \| = \|\varphi'(\xi)(x - x_i)\| = \|\varphi'(\xi)\| \cdot \underbrace{\|x - x_i\|}_{\leq \nu}$$

$$\frac{\epsilon}{\nu} \leq \|\varphi'(\xi)\| \leq \|\varphi'\|_\infty$$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D \ \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).
- By MVT $\exists \xi$ between x and x_i

$$\epsilon \leq \| \underbrace{\varphi(x)}_{\in B} - \underbrace{\varphi(x_i)}_{\notin \mathcal{U}_\epsilon(B)} \| = \|\varphi'(\xi)(x - x_i)\| = \|\varphi'(\xi)\| \cdot \underbrace{\|x - x_i\|}_{\leq \nu}$$

$$\frac{\epsilon}{\nu} \leq \|\varphi'(\xi)\| \leq \|\varphi'\|_\infty \quad \text{becomes arbitrarily improbable}$$

Probabilistic Model Checking: Proof

Proof. (for problem with tolerance $\epsilon > 0$).

- Let $X \subseteq D$ set of points where φ is evaluated and $\nu := \|d(\cdot, X)\|_\infty$.
- If $\varphi(x_i) \in \mathcal{U}_\epsilon(B)$ for a $x_i \in X$, output "$\neq\emptyset$" correct with tolerance ϵ.
- Otherwise, output "$=\emptyset$" wrong with probability $p \to 0$ for $\nu \to 0$:
- Suppose $\exists x \in D\ \ \varphi(x) \in B$. Let $x_i \in X$ have smallest distance to x.
- Assume $S[x, x_i] \subseteq D$ (use a $\nu > 0$ with $\mathcal{U}_\nu(x) \subseteq D$ as D open).
- By MVT $\exists\xi$ between x and x_i

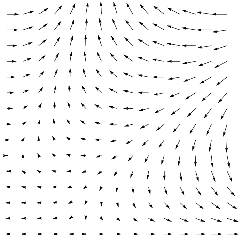
$$\epsilon \leq \|\underbrace{\varphi(x)}_{\in B} - \underbrace{\varphi(x_i)}_{\notin \mathcal{U}_\epsilon(B)}\| = \|\varphi'(\xi)(x - x_i)\| = \|\varphi'(\xi)\| \cdot \underbrace{\|x - x_i\|}_{\leq \nu}$$

$$\frac{\epsilon}{\nu} \leq \|\varphi'(\xi)\| \leq \|\varphi'\|_\infty \quad \text{becomes arbitrarily improbable}$$

- Because $P(\|\varphi'\|_\infty \geq \frac{\epsilon}{\nu}) \to 0$ for $\nu \to 0$ by premise, as ϵ is a constant independent of ν and $\frac{\epsilon}{\nu} \to \infty$ as $\nu \to 0$.

$\square$

Differential Flow Approximation



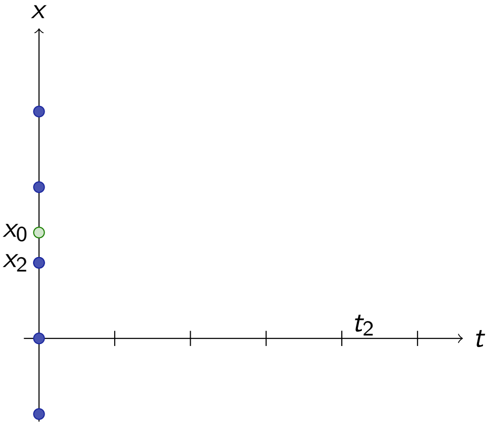
φ solves
$x'(t) = f(t, x)$

Proposition

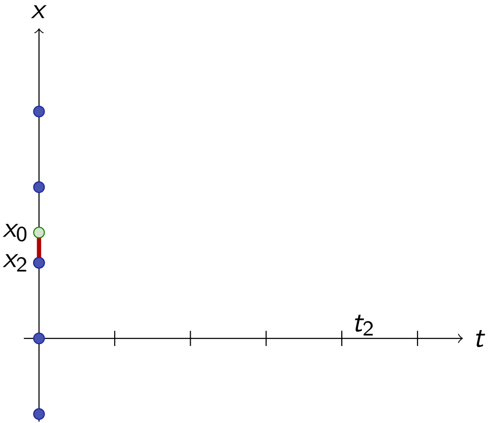
- *Flow φ is solution of $x'(t) = f(t, x)$*
- $f \in C([a, b] \times \mathbb{R}^n, \mathbb{R}^n)$
- *ℓ-Lipschitz-continuous:* $\|f(t, x_1) - f(t, x_2)\| \leq \ell \|x_1 - x_2\|$

$\Rightarrow$ *Continuous image computation decidable*

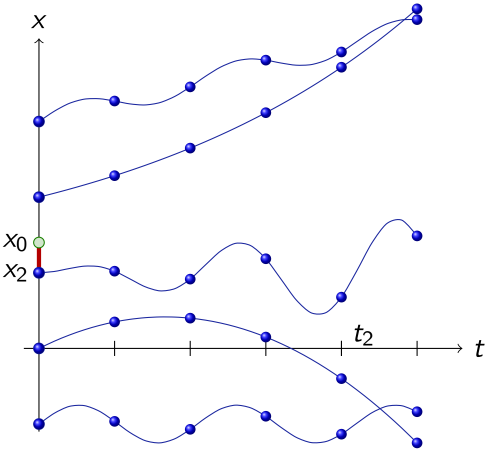
Differential Flow Approximation: Proof Illustration



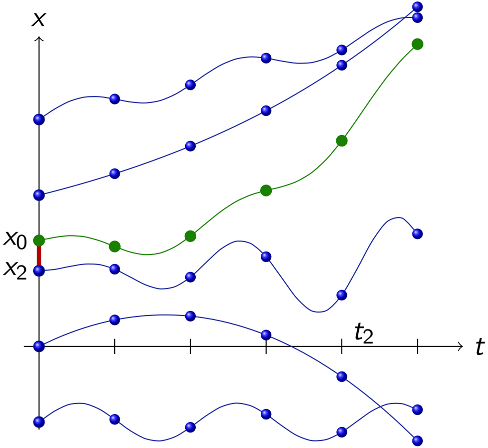
Differential Flow Approximation: Proof Illustration



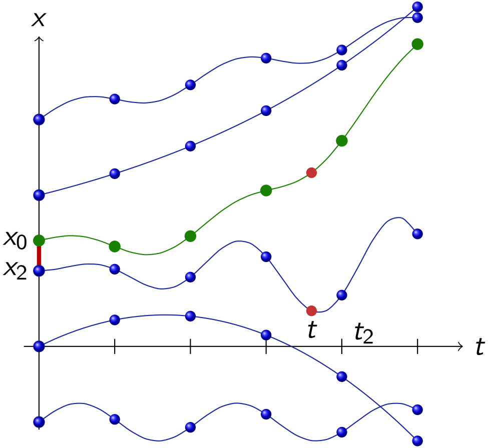
Differential Flow Approximation: Proof Illustration



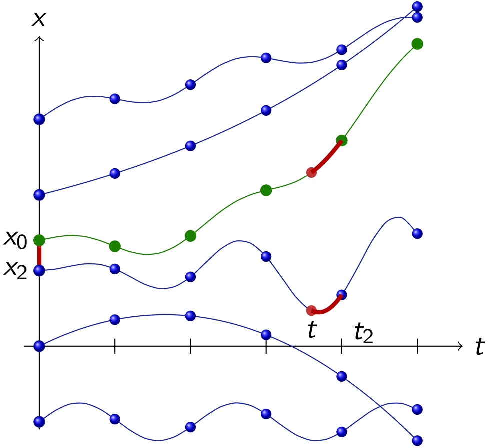
Differential Flow Approximation: Proof Illustration



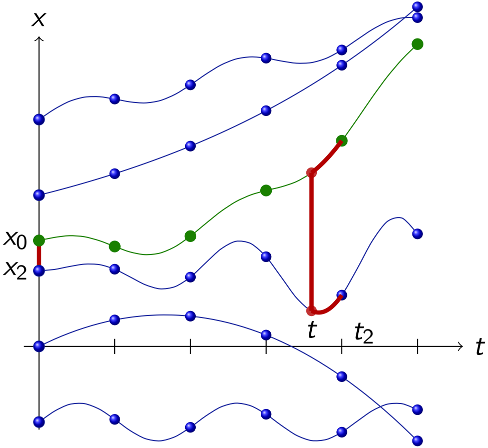
Differential Flow Approximation: Proof Illustration



Differential Flow Approximation: Proof Illustration



Differential Flow Approximation: Proof Illustration



Differential Flow Approximation: Proof

Proof.

- Let $\epsilon > 0$. For (t, x_0) let (t_2, x_2) be the closest points on a mesh.

Differential Flow Approximation: Proof

Proof.

- Let $\epsilon > 0$. For (t, x_0) let (t_2, x_2) be the closest points on a mesh.

$\Rightarrow$ Flow $\varphi(t; x_0)$ arbitrarily close to mesh values $\varphi(t_2; x_2)$, which can be approximated numerically:

$$\begin{aligned}
\|\varphi(t; x_0) - \varphi(t_2; x_2)\| &\leq \|\varphi(t; x_0) - \varphi(t; x_2)\| + \|\varphi(t; x_2) - \varphi(t_2; x_2)\| \\
&\leq e^{\ell|t-t_0|}\|x_0 - x_2\| + \|\varphi'(\xi; x_2)\| \cdot |t - t_2| \\
&= \underbrace{e^{\ell|t-t_0|}}_{\text{bounded}}\|x_0 - x_2\| + \underbrace{\|f(\xi, \varphi(\xi; x_2))\|}_{\text{bounded}} \cdot |t - t_2|
\end{aligned}$$

by corollary of Picard-Lindelöf and MVT with $\xi \in (t, t_2)$.

Differential Flow Approximation: Proof

Proof.

- Let $\epsilon > 0$. For (t, x_0) let (t_2, x_2) be the closest points on a mesh.

$\Rightarrow$ Flow $\varphi(t; x_0)$ arbitrarily close to mesh values $\varphi(t_2; x_2)$, which can be approximated numerically:

$$\begin{aligned}\|\varphi(t;x_0) - \varphi(t_2;x_2)\| &\leq \|\varphi(t;x_0) - \varphi(t;x_2)\| + \|\varphi(t;x_2) - \varphi(t_2;x_2)\| \\ &\leq e^{\ell|t-t_0|}\|x_0 - x_2\| + \|\varphi'(\xi;x_2)\| \cdot |t - t_2| \\ &= \underbrace{e^{\ell|t-t_0|}}_{\text{bounded}}\|x_0 - x_2\| + \underbrace{\|f(\xi, \varphi(\xi;x_2))\|}_{\text{bounded}} \cdot |t - t_2| \overset{!}{<} \frac{\epsilon}{2}\end{aligned}$$

by corollary of Picard-Lindelöf and MVT with $\xi \in (t, t_2)$.

- Factors bounded on compact domain in bounded time; f Lipschitz.

Differential Flow Approximation: Proof

Proof.

- Let $\epsilon > 0$. For (t, x_0) let (t_2, x_2) be the closest points on a mesh.

$\Rightarrow$ Flow $\varphi(t; x_0)$ arbitrarily close to mesh values $\varphi(t_2; x_2)$, which can be approximated numerically:

$$\begin{aligned}
\|\varphi(t;x_0) - \varphi(t_2;x_2)\| &\leq \|\varphi(t;x_0) - \varphi(t;x_2)\| + \|\varphi(t;x_2) - \varphi(t_2;x_2)\| \\
&\leq e^{\ell|t-t_0|}\|x_0 - x_2\| + \|\varphi'(\xi;x_2)\| \cdot |t - t_2| \\
&= \underbrace{e^{\ell|t-t_0|}}_{\text{bounded}}\|x_0 - x_2\| + \underbrace{\|f(\xi, \varphi(\xi;x_2))\|}_{\text{bounded}} \cdot |t - t_2| \overset{!}{<} \frac{\epsilon}{2}
\end{aligned}$$

by corollary of Picard-Lindelöf and MVT with $\xi \in (t, t_2)$.

- Factors bounded on compact domain in bounded time; f Lipschitz.
- Lipschitz-continuous one-step methods of order p for mesh quantity $\varphi(t_2; x_2)$ with global discretization error $< \frac{\epsilon}{2}$ when refining mesh.

Differential Flow Approximation: Proof

Proof.

- Let $\epsilon > 0$. For (t, x_0) let (t_2, x_2) be the closest points on a mesh.

$\Rightarrow$ Flow $\varphi(t; x_0)$ arbitrarily close to mesh values $\varphi(t_2; x_2)$, which can be approximated numerically:

$$\begin{aligned}
\|\varphi(t;x_0) - \varphi(t_2;x_2)\| &\leq \|\varphi(t;x_0) - \varphi(t;x_2)\| + \|\varphi(t;x_2) - \varphi(t_2;x_2)\| \\
&\leq e^{\ell|t-t_0|}\|x_0 - x_2\| + \|\varphi'(\xi;x_2)\| \cdot |t - t_2| \\
&= \underbrace{\color{red}{e^{\ell|t-t_0|}}}_{\text{bounded}}\|x_0 - x_2\| + \underbrace{\|f(\xi,\varphi(\xi;x_2))\|}_{\text{bounded}} \cdot |t - t_2| \overset{!}{<} \frac{\epsilon}{2}
\end{aligned}$$

by corollary of Picard-Lindelöf and MVT with $\xi \in (t, t_2)$.

- Factors bounded on compact domain in bounded time; f Lipschitz.
- Lipschitz-continuous one-step methods of order p for mesh quantity $\varphi(t_2; x_2)$ with global discretization error $<\frac{\epsilon}{2}$ when refining mesh.

Approximation Errors

Exponential terms in approximation error computations are bad

$$\|\varphi(t;x_0) - \varphi(t_2;x_2)\| \leq e^{\ell|t-t_0|}\|x_0 - x_2\| + \|f(\xi,\varphi(\xi;x_2))\| \cdot |t - t_2|$$

Approximation Errors

Exponential terms in approximation error computations are bad

$$\|\varphi(t; x_0) - \varphi(t_2; x_2)\| \leq e^{\ell|t-t_0|}\|x_0 - x_2\| + \|f(\xi, \varphi(\xi; x_2))\| \cdot |t - t_2|$$

but tight!

Example

$$x' = \ell x$$

is ℓ-Lipschitz-continuous with unique global solution$\varphi(t; x_0) = x_0 e^{\ell(t-t_0)}$

$$\|\varphi(t; x_0) - \varphi(t; x_2)\| = \|e^{\ell(t-t_0)}(x_0 - x_2)\| = e^{\ell|t-t_0|}\|x_0 - x_2\|$$

Approximation Errors

Exponential terms in approximation error computations are bad

$$\|\varphi(t;x_0) - \varphi(t_2;x_2)\| \le e^{\ell|t-t_0|}\|x_0 - x_2\| + \|f(\xi,\varphi(\xi;x_2))\| \cdot |t - t_2|$$

but tight!

Example

$$x' = \ell x$$

is ℓ-Lipschitz-continuous with unique global solution $\varphi(t;x_0) = x_0 e^{\ell(t-t_0)}$

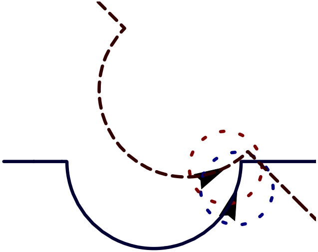
$$\|\varphi(t;x_0) - \varphi(t;x_2)\| = \|e^{\ell(t-t_0)}(x_0 - x_2)\| = e^{\ell|t-t_0|}\|x_0 - x_2\|$$

Outline

Experiments with Roundabout ATC

Counterexamples with distances ≈0.0016mi after 3 refinements

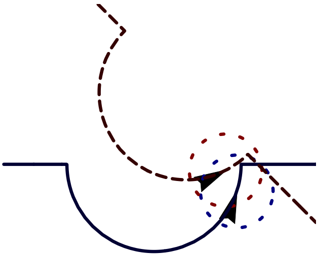
in absolute coords



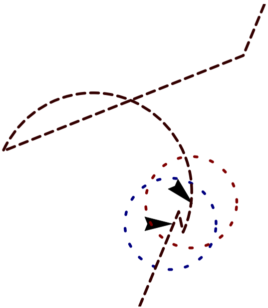
Experiments with Roundabout ATC

Counterexamples with distances ≈0.0016mi after 3 refinements

in absolute coords



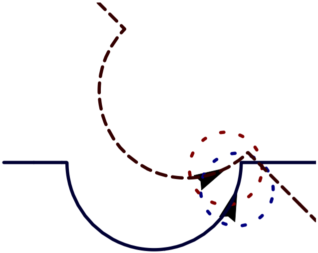
relative coords



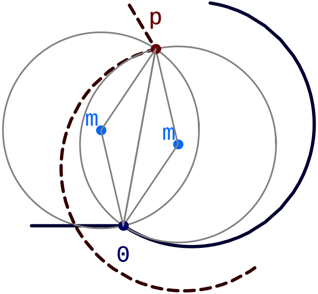
Experiments with Tangential Roundabout ATC

Solution: adaptively choose rotation using tangential construction

classical



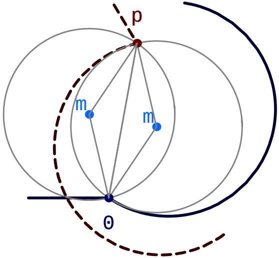
tangential



No more counterexamples found

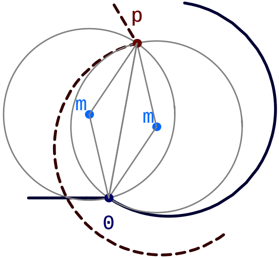
Experimental Results: Tangential Roundabout ATM

$$\alpha^2 = \|m - 0\|^2$$
$$\alpha^2 = \|m - p\|^2$$
$$\gamma_1 = \angle(m - 0)$$
$$\gamma_2 = \angle(m - p)$$



Experimental Results: Tangential Roundabout ATM

$$\alpha^2 = \|m - 0\|^2$$
$$\alpha^2 = \|m - p\|^2$$
$$\gamma_1 = \angle(m - 0)$$
$$\gamma_2 = \angle(m - p)$$

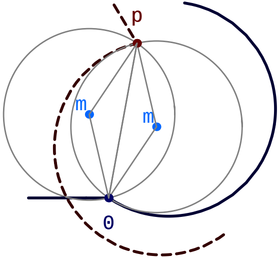


Solutions for θ_j using any $k, \ell \in \{1, 2\}$:

$$\angle\left((-1)^{j+1}\frac{x^3 + xy^2 + (-1)^{j+k}\imath\sqrt{x^2(x^2+y^2)(4\alpha^2 - x^2 - y^2)}}{x(x - \imath y)}\right) + (-1)^{\ell}\frac{\pi}{2}$$

Experimental Results: Tangential Roundabout ATM

$$\alpha^2 = \|m - 0\|^2$$
$$\alpha^2 = \|m - p\|^2$$
$$\gamma_1 = \angle(m - 0)$$
$$\gamma_2 = \angle(m - p)$$

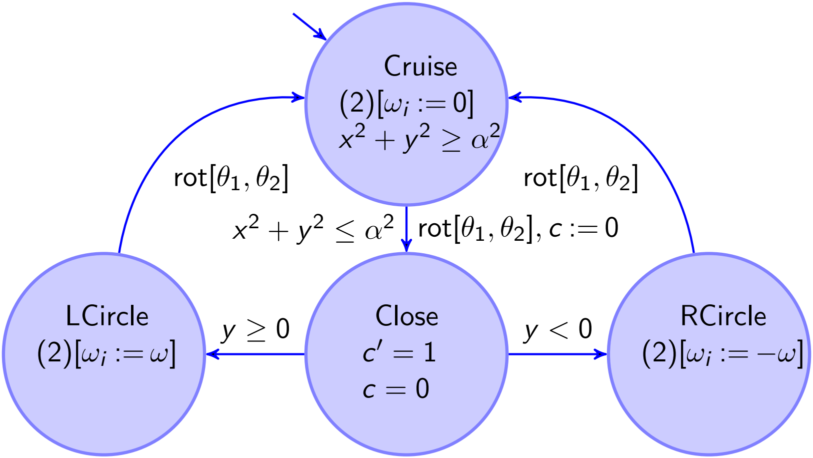


Solutions for θ_j using any $k, \ell \in \{1, 2\}$:

$$\angle\left((-1)^{j+1}\frac{x^3 + xy^2 + (-1)^{j+k}\imath\sqrt{x^2(x^2+y^2)(4\alpha^2 - x^2 - y^2)}}{x(x - \imath y)}\right) + (-1)^\ell \frac{\pi}{2}$$

$$\min_{k,\ell} \max(|\theta_1 - 0|, |\theta_2 - \phi|)$$

Tangential Roundabout Maneuver Automaton



Return

Outline

Summary

- Image computation in hybrid systems model checking
  1. approx uniformly
  2. blur by uniform error
  3. check for B

flows	approx / image computation
continuous	uniform approx exists, but...
smooth	undecidable by evaluation
bounded by b	decidable
bound probabilities	probabilistically decidable
ODE ℓ-Lipschitz	decidable

- Combine numerical algorithms with symbolic analysis
- Roundabout maneuver unsafe
- Solution: adaptively choose rotations by tangential construction

Possible Extensions for Projects

- Extend tangential roundabout maneuver
  - Determine speed/thrust bounds
  - Position discrepancies caused by imprecise tracking
  - Verify liveness: aircraft finally on original route
  - Full curve dynamics
- Combine numerical algorithms with symbolic analysis ...
- Improved model checker
- Multivariate rational spline approximation

A. Chutinan and B. H. Krogh.
Computational techniques for hybrid system verification.
IEEE T. Automat. Contr., 48(1):64–75, 2003.

E. M. Clarke, A. Fehnker, Z. Han, B. H. Krogh, J. Ouaknine, O. Stursberg, and M. Theobald.
Abstraction and counterexample-guided refinement in model checking of hybrid systems.
Int. J. Found. Comput. Sci., 14(4):583–604, 2003.

C. Piazza, M. Antoniotti, V. Mysore, A. Policriti, F. Winkler, and B. Mishra.
Algorithmic algebraic model checking I: Challenges from systems biology.
In K. Etessami and S. K. Rajamani, editors, *CAV*, volume 3576 of *LNCS*, pages 5–19. Springer, 2005.

A. Platzer and E. M. Clarke.
The image computation problem in hybrid systems model checking.
In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *HSCC*, volume 4416 of *LNCS*, pages 473–486. Springer, 2007.