From validated numerics to certified homotopies

Alexandre Guillemot, Pierre Lairez

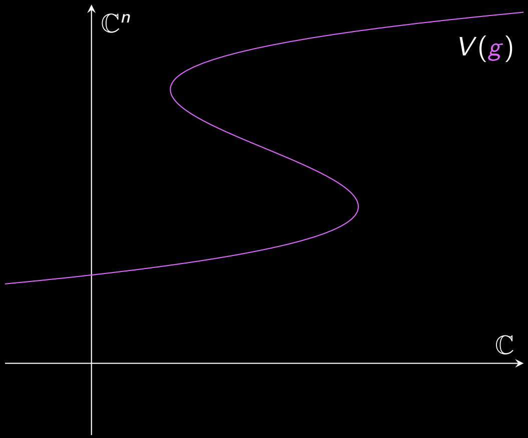
May 22, 2024

Introduction

Introduction

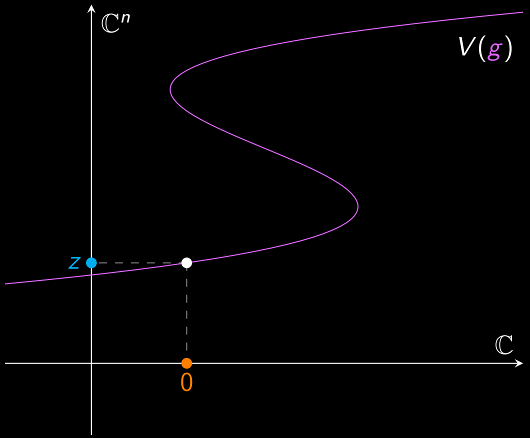
Setup

- Let $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial map.
- Notation : $g_t : \mathbb{C}^n \to \mathbb{C}^n$ defined by $g_t(z) = g(t, z)$.



Setup

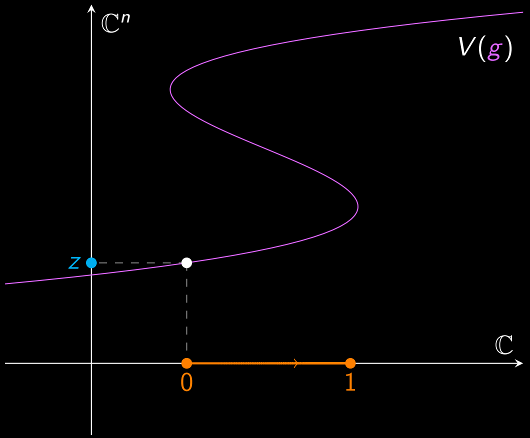
- Let $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial map.
- Notation : $g_t : \mathbb{C}^n \to \mathbb{C}^n$ defined by $g_t(z) = g(t, z)$.
- Let $z \in \mathbb{C}^n$ such that $g_0(z) = 0$.



Introduction

Setup

- Let $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial map.
- Notation : $g_t : \mathbb{C}^n \to \mathbb{C}^n$ defined by $g_t(z) = g(t, z)$.
- Let $z \in \mathbb{C}^n$ such that $g_0(z) = 0$.

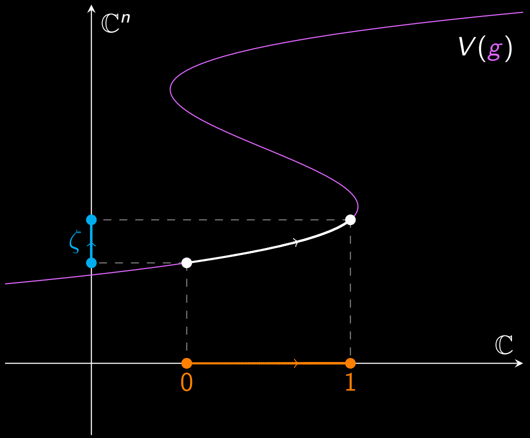


Introduction

Setup

- Let $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial map.
- Notation : $g_t : \mathbb{C}^n \to \mathbb{C}^n$ defined by $g_t(z) = g(t, z)$.
- Let $z \in \mathbb{C}^n$ such that $g_0(z) = 0$.

$\rightsquigarrow$ Moving the parameter from 0 to 1 induces $\zeta : [0, 1] \to \mathbb{C}^n$ s.t. $\zeta(0) = z$ and $g_t(\zeta(t)) = 0$.

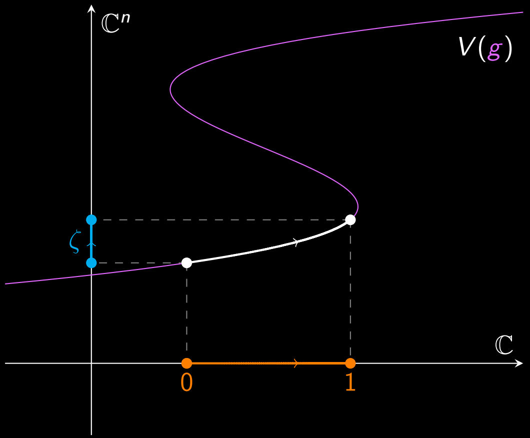


Setup

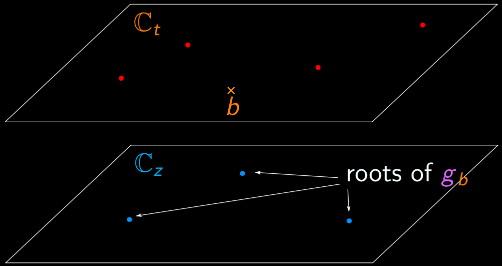
- Let $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial map.
- Notation : $g_t : \mathbb{C}^n \to \mathbb{C}^n$ defined by $g_t(z) = g(t, z)$.
- Let $z \in \mathbb{C}^n$ such that $g_0(z) = 0$.

$\rightsquigarrow$ Moving the parameter from 0 to 1 induces $\zeta : [0, 1] \to \mathbb{C}^n$ s.t. $\zeta(0) = z$ and $g_t(\zeta(t)) = 0$.

- Goal : "Track" ζ, with some topological guarantees.



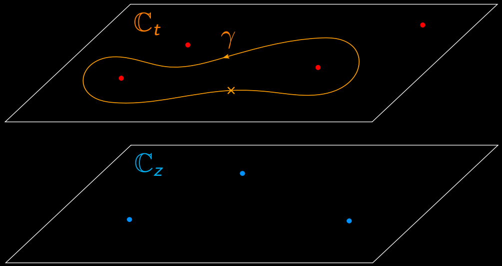
Motivation : braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- let $b \in \mathbb{C} \setminus \Sigma$ be a base point,

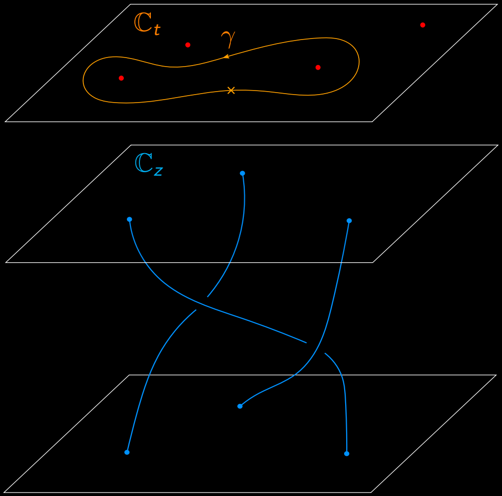
Motivation : braid computations



Setup

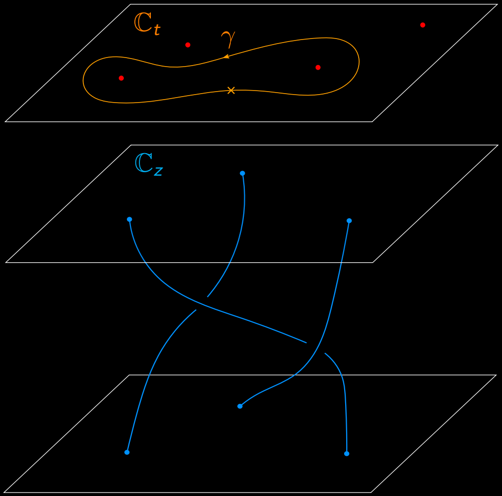
- Let $g \in \mathbb{C}[t, z]$,
- let $b \in \mathbb{C}\backslash\Sigma$ be a base point,
- let $\gamma : [0, 1] \to \mathbb{C}\backslash\Sigma$ be a loop starting at b.

Motivation : braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- let $b \in \mathbb{C} \backslash \Sigma$ be a base point,
- let $\gamma : [0, 1] \to \mathbb{C} \backslash \Sigma$ be a loop starting at b.
- The displacement of all roots of g_t when t moves along γ defines a braid.



Setup

- Let $g \in \mathbb{C}[t, z]$,
- let $b \in \mathbb{C} \backslash \Sigma$ be a base point,
- let $\gamma : [0, 1] \to \mathbb{C} \backslash \Sigma$ be a loop starting at b.
- The displacement of all roots of g_t when t moves along γ defines a braid.
- Goal : "compute" this braid.
- An efficient **certified** path tracking algorithm is required !

Previous works

Non certified path trackers

- **PHCpack**, *Jan Verschelde*, 1999.
- **Bertini**, *Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, Charles W. Wampler*, 2013.
- **HomotopyContinuation.jl**, *Paul Breiding and Sascha Timme*, 2017.

Previous works

Non certified path trackers

- **PHCpack**, *Jan Verschelde*, 1999.
- **Bertini**, *Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, Charles W. Wampler*, 2013.
- **HomotopyContinuation.jl**, *Paul Breiding and Sascha Timme*, 2017.

Certified path trackers using explicit theoritical bounds

- **Certified Numerical Homotopy Tracking**, *Carlos Beltrán and Anton Leykin*, 2012.
- **An epsilon-delta bound for plane algebraic curves and its use for certified homotopy continuation of systems of plane algebraic curves**, *Stefan Kranich*, 2016.

Previous works

Non certified path trackers

- **PHCpack**, *Jan Verschelde*, 1999.
- **Bertini**, *Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, Charles W. Wampler*, 2013.
- **HomotopyContinuation.jl**, *Paul Breiding and Sascha Timme*, 2017.

Certified path trackers using explicit theoritical bounds

- **Certified Numerical Homotopy Tracking**, *Carlos Beltrán and Anton Leykin*, 2012.
- **An epsilon-delta bound for plane algebraic curves and its use for certified homotopy continuation of systems of plane algebraic curves**, *Stefan Kranich*, 2016.

Certified path trackers using interval arithmetic

- **An Interval Step Control for Continuation Methods**, *R. Baker Kearfott and Zhaoyun Xing*, 1994.
- **Reliable homotopy continuation**, *Joris van der Hoeven*, 2011.
- **SIROCCO: A Library for Certified Polynomial Root Continuation**, *Miguel Ángel Marco-Buzunariz and Marcos Rodríguez*, 2016.

Path certification

Root isolation

Root isolation criterion (Krawczyk, Moore, Rump)

- $f : \mathbb{C}^n \to \mathbb{C}^n$ polynomial,
- $z \in \mathbb{C}^n$, $r \in \mathbb{R}_{>0}$, $A \in \mathbb{C}^{n\times n}$,
- $\rho \in (0, 1)$,

such that for all $u, v \in B_r$,

$$-Af(z) + [\mathrm{I}_n - A \cdot Jf(z + u)]v \in B_{\rho r}.$$

Then there exists a unique $\tilde{z} \in z + B_{\rho r}$ s.t. $f(\tilde{z}) = 0$.

Root isolation

Root isolation criterion (Krawczyk, Moore, Rump)

- $f : \mathbb{C}^n \to \mathbb{C}^n$ polynomial,
- $z \in \mathbb{C}^n$, $r \in \mathbb{R}_{>0}$, $A \in \mathbb{C}^{n\times n}$,
- $\rho \in (0, 1)$,

such that for all $u, v \in B_r$,

$$-Af(z) + [\mathrm{I}_n - A \cdot Jf(z + u)]v \in B_{\rho r}.$$

Then there exists a unique $\tilde{z} \in z + B_{\rho r}$ s.t. $f(\tilde{z}) = 0$.

Proof sketch

We show that $\varphi : z + B_{\rho r} \to \mathbb{C}^n$ defined by $\varphi(w) = w - Af(w)$ is a ρ-contraction map with values in $z + B_{\rho r}$.

Moore boxes

Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be polynomial map and $\rho \in (0,1)$.

Moore boxes

A ρ-*Moore box* for f is a triple (z, r, A) satisfying

$$-Af(z) + [\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r \subseteq B_{\rho r}.$$

Moore boxes

Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be polynomial map and $\rho \in (0, 1)$.

Moore boxes

A ρ-*Moore box* for f is a triple (z, r, A) satisfying

$$-Af(z) + [\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r \subseteq B_{\rho r}.$$

Moore boxes are our data structure to represent zeros of f.

Moore boxes

Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be polynomial map and $\rho \in (0, 1)$.

Moore boxes

A ρ-*Moore box* for f is a triple (z, r, A) satisfying

$$-Af(z) + [\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r \subseteq B_{\rho r}.$$

Moore boxes are our data structure to represent zeros of f.

In practice ...

We use interval arithmetic to check that a triple (z, r, A) is a Moore box (sufficient condition).

Interval arithmetic

We choose a set $\mathbb{F} \subset \mathbb{R}$ of representable numbers.

Interval space

- $\square\mathbb{R} = \{[a, b] \subseteq \mathbb{R} \mid a \leq b,\ a, b \in \mathbb{F}\}$,
- $\square\mathbb{C}$: pairs of elements of $\square\mathbb{R}$,
- $\square\mathbb{R}^n$ resp. $\square\mathbb{C}^n$: vectors of boxes in $\mathbb{R}$ resp. $\mathbb{C}$ of size n.

Interval arithmetic

We choose a set $\mathbb{F} \subset \mathbb{R}$ of representable numbers.

Interval space

- $\square\mathbb{R} = \{[a, b] \subseteq \mathbb{R} \mid a \leq b,\ a, b \in \mathbb{F}\}$,
- $\square\mathbb{C}$: pairs of elements of $\square\mathbb{R}$,
- $\square\mathbb{R}^n$ resp. $\square\mathbb{C}^n$: vectors of boxes in $\mathbb{R}$ resp. $\mathbb{C}$ of size n.

Interval extension

Let $f : \mathbb{C}^n \to \mathbb{C}^m$. An *interval extension* of f is a map $\square f : \square\mathbb{C}^n \to \square\mathbb{C}^m$ such that

$$\forall Z \in \square\mathbb{C}^n,\ f(Z) \subseteq \square f(Z).$$

Interval arithmetic

We choose a set $\mathbb{F} \subset \mathbb{R}$ of representable numbers.

Interval space

- $\square\mathbb{R} = \{[a, b] \subseteq \mathbb{R} \mid a \leq b,\ a, b \in \mathbb{F}\}$,
- $\square\mathbb{C}$: pairs of elements of $\square\mathbb{R}$,
- $\square\mathbb{R}^n$ resp. $\square\mathbb{C}^n$: vectors of boxes in $\mathbb{R}$ resp. $\mathbb{C}$ of size n.

Interval extension

Let $f : \mathbb{C}^n \to \mathbb{C}^m$. An *interval extension* of f is a map $\square f : \square\mathbb{C}^n \to \square\mathbb{C}^m$ such that

$$\forall Z \in \square\mathbb{C}^n,\ f(Z) \subseteq \square f(Z).$$

By picking effective extensions of binary operations, we are able to effectively build an extension $\square f$ given a (polynomial) map f.

Interval certification of a Moore box

Algorithm

def $M(\square f, \square Jf, z, r, A, \rho)$:
  if $-A \cdot \square f(z) + [\mathrm{I}_n - A \cdot \square Jf(z + B_r)]B_r \subseteq B_{\rho r}$: # Interval arithmetic computation
    return `True`
  else:
    return `False`

Interval certification of a Moore box

Algorithm

```
def M(□f, □Jf, z, r, A, ρ):
    if −A · □f(z) + [I_n − A · □Jf(z + B_r)]B_r ⊆ B_ρr:  # Interval arithmetic computation
        return True
    else:
        return False
```

Proposition

Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial map. If $\square f$ is an extension of f, $\square Jf$ is an extension of Jf and $M(\square f, \square Jf, z, r, A, \rho)$ returns True, then (z, r, A) is a ρ-Moore box for f.

Interval certification of a Moore box

Algorithm

def $M(\Box f, \Box Jf, z, r, A, \rho)$:

  if $-A \cdot \Box f(z) + [\mathrm{I}_n - A \cdot \Box Jf(z + B_r)]B_r \subseteq B_{\rho r}$: # Interval arithmetic computation

    return `True`

  else:

    return `False`

Proposition

Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial map. If $\Box f$ is an extension of f, $\Box Jf$ is an extension of Jf and $M(\Box f, \Box Jf, z, r, A, \rho)$ returns `True`, then (z, r, A) is a ρ-Moore box for f.

Proof

Fundamental property of interval arithmetic !

Algorithm

Expected input and output

Input

- A polynomial map $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- a zero $\tilde{z}$ of g_0 (represented by a $\frac{7}{8}$-Moore box (z, r, A)).

Expected input and output

Input

- A polynomial map $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- a zero $\tilde{z}$ of g_0 (represented by a $\frac{7}{8}$-Moore box (z, r, A)).

Let $\zeta : [0, 1] \to \mathbb{C}^n$ s.t. $\zeta(0) = \tilde{z}$ and $g_t(\zeta(t)) = 0$ for all $t \in [0, 1]$.

Expected input and output

Input

- A polynomial map $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- a zero $\tilde{z}$ of g_0 (represented by a $\frac{7}{8}$-Moore box (z, r, A)).

Let $\zeta : [0, 1] \to \mathbb{C}^n$ s.t. $\zeta(0) = \tilde{z}$ and $g_t(\zeta(t)) = 0$ for all $t \in [0, 1]$.

Output

Moore boxes (z_t, r_t, A_t) for g_t isolating $\zeta(t)$, this for all $t \in [0, 1]$.

Expected input and output

Input

- A polynomial map $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- a zero $\tilde{z}$ of g_0 (represented by a $\frac{7}{8}$-Moore box (z, r, A)).

Let $\zeta : [0,1] \to \mathbb{C}^n$ s.t. $\zeta(0) = \tilde{z}$ and $g_t(\zeta(t)) = 0$ for all $t \in [0,1]$.

Output

Moore boxes (z_t, r_t, A_t) for g_t isolating $\zeta(t)$, this for all $t \in [0,1]$.

Output in practice

- Subintervals $I_1, \cdots, I_k$ covering $[0,1]$,
- triples $(z_1, r_1, A_1), \cdots, (z_k, r_k, A_k)$

such that (z_ℓ, r_ℓ, A_ℓ) is a $\frac{7}{8}$-Moore box for g on I_ℓ.

General strategy

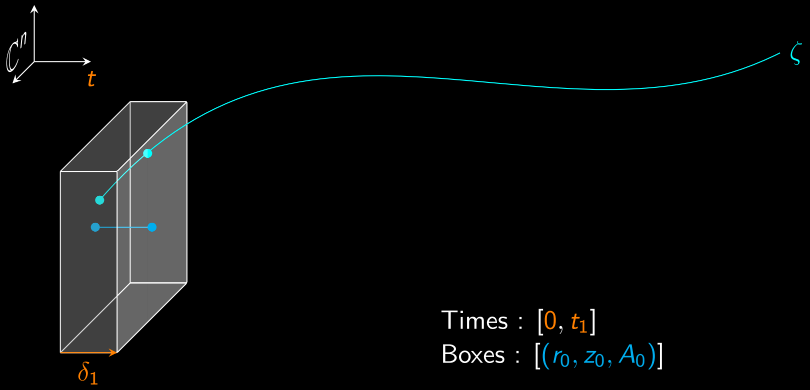
Times : [0]
Boxes : []

General strategy

Times : [0]

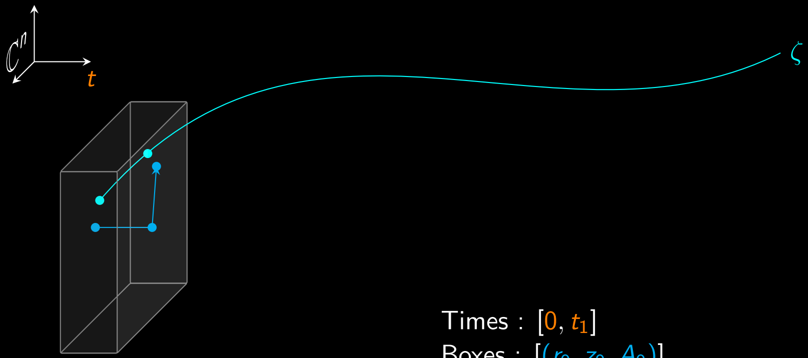
Boxes : []

General strategy



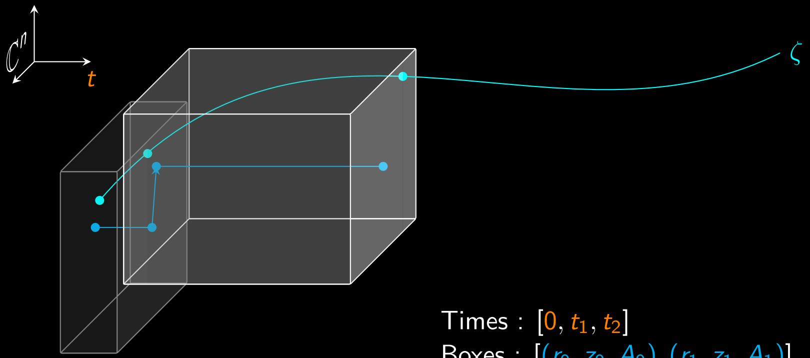
Times : $[0, t_1]$
Boxes : $[(r_0, z_0, A_0)]$

General strategy



Times : $[0, t_1]$
Boxes : $[(r_0, z_0, A_0)]$

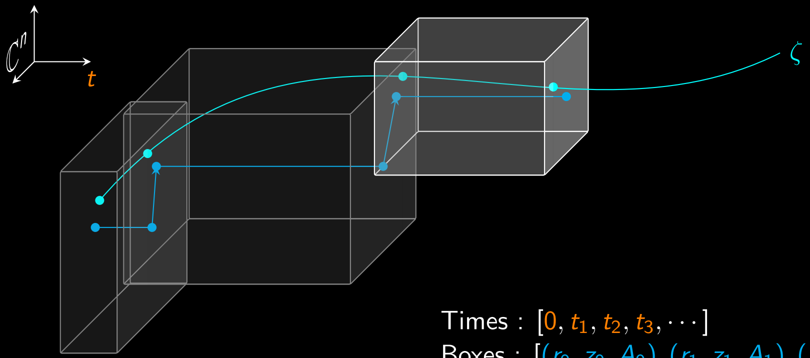
General strategy



Times : $[0, t_1, t_2]$

Boxes : $[(r_0, z_0, A_0), (r_1, z_1, A_1)]$

General strategy



Times : $[0, t_1, t_2, t_3, \cdots]$

Boxes : $[(r_0, z_0, A_0), (r_1, z_1, A_1), (r_2, z_2, A_2), \cdots]$

Validating boxes on intervals

How can we check that (z, r, A) is a ρ-Moore box for g_s, for all $s \in T \subseteq \mathbb{R}$?

Answer : build

- $\square g_T : \square\mathbb{C}^n \to \square\mathbb{C}^n$ that is, for all $s \in T$, an extension of g_s,
- $\square J g_T : \square\mathbb{C}^n \to \square\mathbb{C}^{n\times n}$ that is, for all $s \in T$, an extension of $J g_s$,

and check $M(\square g_T, \square J g_T, z, r, A, \rho)$.

Validating boxes on intervals

How can we check that (z, r, A) is a ρ-Moore box for g_s, for all $s \in T \subseteq \mathbb{R}$?

Answer : build

- $\square g_T : \square\mathbb{C}^n \to \square\mathbb{C}^n$ that is, for all $s \in T$, an extension of g_s,
- $\square J g_T : \square\mathbb{C}^n \to \square\mathbb{C}^{n\times n}$ that is, for all $s \in T$, an extension of $J g_s$,

and check $M(\square g_T, \square J g_T, z, r, A, \rho)$.

How can we build such $\square g_T$ and $\square J g_T$?

Answer : specify to T the first variable in $\square g$ an extension of g : $\square g_T(Z) = \square g(T, Z)$.
Indeed, for all $s \in T$, if $Z \in \square\mathbb{C}$ and $z \in Z$ then

$$g_s(z) = g(t, z) \in \square g(T, Z) = \square g_T(Z).$$

Refining the approximation

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Refining the approximation

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Idea

After thickening, we have (z, r, A) a Moore box for g on $[t, t'] \subseteq [0, 1]$. In particular, it is a Moore box for $g_{t'}$, so we can perform quasi Newton iterations using A before thickening again.

Refining the approximation

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Idea

After thickening, we have (z, r, A) a Moore box for g on $[t, t'] \subseteq [0, 1]$. In particular, it is a Moore box for $g_{t'}$, so we can perform quasi Newton iterations using A before thickening again.

Remaning problems to address

- When thickening, how do we choose the radius of the box and for how much time ?
- When should we stop iterating quasi Newton steps while refining ?
- How to ensure termination in a **realistic computational model** (MPFI) ?

Naive computational model

We assume that $\mathbb{F} = \mathbb{R}$. Moreover, we assume the following on our extensions :

Assumptions

Let $f : \mathbb{C}^n \to \mathbb{C}^m$ and let $\square f$ be an extension of f.

- **Naturalilty :** for all $z \in \mathbb{R}$, $\square f([z, z]) = f(z)$,
- **Continuity :** for all $(Z_n)_n$ sequence in $\square \mathbb{C}^n$ converging to $Z \in \square \mathbb{C}^n$, $\square f(Z_n)$ converges to $\square f(Z)$ when $n \to \infty$ (for the induced Hausdorff distance).

Naive computational model

We assume that $\mathbb{F} = \mathbb{R}$. Moreover, we assume the following on our extensions :

Assumptions

Let $f : \mathbb{C}^n \to \mathbb{C}^m$ and let $\square f$ be an extension of f.

- **Naturalilty :** for all $z \in \mathbb{R}$, $\square f([z, z]) = f(z)$,
- **Continuity :** for all $(Z_n)_n$ sequence in $\square\mathbb{C}^n$ converging to $Z \in \square\mathbb{C}^n$, $\square f(Z_n)$ converges to $\square f(Z)$ when $n \to \infty$ (for the induced Hausdorff distance).

This is not reasonable, but we will see later how to relax these constraints !

Thicken

Remark

If (z, r, A) is a $\frac{1}{8}$-Moore box for g_t, then there exists $\delta > 0$ such that it is a $\frac{7}{8}$-Moore box for g on $[t, t + \delta]$.

Thicken

Remark

If (z, r, A) is a $\frac{1}{8}$-Moore box for g_t, then there exists $\delta > 0$ such that it is a $\frac{7}{8}$-Moore box for g on $[t, t + \delta]$.

Idea

- If (z, r, A) is a $\frac{1}{8}$-Moore box for g_t, try to certify it on $[t, t + \delta]$ for decreasing δ.

Thicken

Remark

If (z, r, A) is a $\frac{1}{8}$-Moore box for g_t, then there exists $\delta > 0$ such that it is a $\frac{7}{8}$-Moore box for g on $[t, t+\delta]$.

Idea

- If (z, r, A) is a $\frac{1}{8}$-Moore box for g_t, try to certify it on $[t, t+\delta]$ for decreasing δ.
- So the input of the thickening step should not only be z close to $\zeta(t)$ but a Moore box (z, r, A) for g_t enclosing this zero.

Thicken

Remark

If (z, r, A) is a $\frac{1}{8}$-Moore box for g_t, then there exists $\delta > 0$ such that it is a $\frac{7}{8}$-Moore box for g on $[t, t+\delta]$.

Idea

- If (z, r, A) is a $\frac{1}{8}$-Moore box for g_t, try to certify it on $[t, t+\delta]$ for decreasing δ.
- So the input of the thickening step should not only be z close to $\zeta(t)$ but a Moore box (z, r, A) for g_t enclosing this zero.
- This forces the output of the refining step to be a $\frac{1}{8}$-Moore box. We keep this problem for later...

Thicken

Input

- $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- $t \in [0, 1]$,
- (z, r, A) a $\frac{1}{8}$-Moore box for g_t.

Thicken

Input

- $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- $t \in [0, 1]$,
- (z, r, A) a $\frac{1}{8}$-Moore box for g_t.

Output

$\delta > 0$ s.t. for all $s \in T = [t, t + \delta]$,
(z, r, A) is a $\frac{7}{8}$-Moore box for g_s.

Thicken

Input

- $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- $t \in [0, 1]$,
- (z, r, A) a $\frac{1}{8}$-Moore box for g_t.

Output

$\delta > 0$ s.t. for all $s \in T = [t, t + \delta]$, (z, r, A) is a $\frac{7}{8}$-Moore box for g_s.

Procedure

```
def thicken(g, t, δ_hint, z, r, A):
    δ ← δ_hint;    T ← [t, t + δ]
    while not M(□g_T, □Jg_T, z, r, A, 7/8):
        δ ← δ/2;    T ← [t, t + δ]
    return δ
```

Refine

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Refine

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Output

(z', r', A') a $\frac{1}{8}$-Moore box for f with same associated zero as (z, r, A).

Refine

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Output

(z', r', A') a $\frac{1}{8}$-Moore box for f with same associated zero as (z, r, A).

More precisely ...

The input satisfies

$$-Af(z) + [\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r \subseteq \frac{7}{8}B_r.$$

Our goal : shrink the l.h.s to make it fit into $\frac{1}{8}B_r$.

Refine

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Refine

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Heuristic

$$-Af(z) + [\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r \subseteq \frac{1}{8} B_r$$

Refine

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Heuristic

$$-Af(z) + [\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r \subseteq \frac{1}{8} B_r$$

- We set A to always be $Jf(z)^{-1}$.

Refine

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Heuristic

$$\underbrace{-Af(z)}_{\xrightarrow[\text{q.n. iters}]{} 0} + [\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r \subseteq \frac{1}{8} B_r$$

- We set A to always be $Jf(z)^{-1}$.
- By performing quasi Newton iterations, we are able to make the term $-Af(z)$ go to zero.

Refine

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Heuristic

$$\underbrace{-Af(z)}_{\xrightarrow[\text{q.n. iters}]{}0} + \underbrace{[\mathrm{I}_n - A \cdot Jf(z + B_r)]}_{\xrightarrow[r \to 0]{}0} B_r \subseteq \frac{1}{8} B_r$$

- We set A to always be $Jf(z)^{-1}$.
- By performing quasi Newton iterations, we are able to make the term $-Af(z)$ go to zero.
- By reducing r, we are able to make the term $[\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r$ fit into any εB_r.

Refine

Reminder

In a ρ-Moore box (z, r, A), the quasi Newton iteration $\varphi(w) = w - Af(w)$ is a ρ-contraction map, and the limit of iterated compositions of φ gives the associated zero $\tilde{z}$.

Heuristic

$$\underbrace{-Af(z)}_{\xrightarrow[\text{q.n. iters}]{}0} + \underbrace{[\mathrm{I}_n - A \cdot Jf(z + B_r)]}_{\xrightarrow[r \to 0]{}0} B_r \subseteq \frac{1}{8} B_r$$

- We set A to always be $Jf(z)^{-1}$.
- By performing quasi Newton iterations, we are able to make the term $-Af(z)$ go to zero.
- By reducing r, we are able to make the term $[\mathrm{I}_n - A \cdot Jf(z + B_r)]B_r$ fit into any εB_r.

Idea : a balance between reductions of r and quasi Newton iterations.

Refine

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Output

(z', r', A') a $\frac{1}{8}$-moore box for f with same associated zero as (z, r, A).

Procedure

```
def refine(f, z, r, A):
    U ← A
    while not M(□f, □Jf, z, r, A, 1/8):
        if −A · □f(z) ⊆ (1/64) B_r:
            r ← r/2
        else:
            z ← z − Uf(z) # Naturality
            A ← Jf(z)^−1
    return z, r, A
```

Track

Procedure

```
def track(g, z, r, A):
    t ← 0;  δ ← 1;
    𝒯 ← [0];  ℬ ← [];
    while t < 1:
        z, r, A ← refine(g_t, z, r, A)
        δ ← thicken(g, t, 2δ, z, r, A)
        t ← t + δ
        𝒯.append(t);  ℬ.append((z, r, A));
    return 𝒯, ℬ
```

Track

Procedure

```
def track(g, z, r, A):
    t ← 0;  δ ← 1;
    𝒯 ← [0];  ℬ ← [];
    while t < 1:
        z, r, A ← refine(g_t, z, r, A)
        δ ← thicken(g, t, 2δ, z, r, A)
        t ← t + δ
        𝒯.append(t);  ℬ.append((z, r, A));
    return 𝒯, ℬ
```

Remaining questions

- Correction ?
- Termination ?
- In which **computational model** ?

Track

Procedure

def $\mathit{track}(g, z, r, A)$:
    $t \leftarrow 0;\ \ \delta \leftarrow 1;$
    $\mathcal{T} \leftarrow [0];\ \ \mathcal{B} \leftarrow [];$
    while $t < 1$:
        $z, r, A \leftarrow \mathit{refine}(g_t, z, r, A)$
        $\delta \leftarrow \mathit{thicken}(g, t, 2\delta, z, r, A)$
        $t \leftarrow t + \delta$
        $\mathcal{T}.\mathit{append}(t);\ \ \mathcal{B}.\mathit{append}((z, r, A));$
    return $\mathcal{T}, \mathcal{B}$

Remaining questions

- Correction ?
- Termination ?
- In which **computational model** ?

Answers

- By construction, correction is quite direct and does not depend on the model.
- What about termination ?
- We would like to state it in a realistic model !

Computational model

Which model to use ?

We need to pick a suitable set $\mathbb{F}$ and respective extensions $\boxplus$ and $\boxtimes$ of $+$ and $\times$.

Computational model

Which model to use ?

We need to pick a suitable set $\mathbb{F}$ and respective extensions $\boxplus$ and $\boxtimes$ of $+$ and $\times$.

We can pick ...

- $\mathbb{F} = \mathbb{Q}$,
- $[a, b] \boxplus [c, d] = [a + c, b + d]$,
- $[a, b] \boxtimes [c, d] = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}]$.

Pros and cons

Pros : naturality and continuity.

Cons : prohibited numerator and denominator growth.

Computational model

Which model to use ?

We need to pick a suitable set $\mathbb{F}$ and respective extensions $\boxplus$ and $\boxtimes$ of $+$ and $\times$.

We can pick ...

- $\mathbb{F} = \{\text{IEEE-754 64-bits floating-point numbers}\}$,
- $[a, b] \boxplus [c, d] = [\underline{a + c}, \overline{b + d}]$,
- $[a, b] \boxtimes [c, d] = [\min\{\underline{ac}, \underline{ad}, \underline{bc}, \underline{bd}\}, \max\{\overline{ac}, \overline{ad}, \overline{bc}, \overline{bd}\}]$.

Pros and cons

Pros : fast.

Cons : termination does not hold.

Computational model

Which model to use ?

We need to pick a suitable set $\mathbb{F}$ and respective extensions $\boxplus$ and $\boxtimes$ of $+$ and $\times$.

We can pick ...

- $\mathbb{F} = \{m2^e \in \mathbb{R} : m, e \in \mathbb{Z}\}$,
- $[a, b] \boxplus_u [c, d] = [\underline{a + c}, \overline{b + d}]$,
- $[a, b] \boxtimes_u [c, d] = [\min\{\underline{ac}, \underline{ad}, \underline{bc}, \underline{bd}\}, \max\{\overline{ac}, \overline{ad}, \overline{bc}, \overline{bd}\}]$.

Where $u \in (0, 1)$ is the roundoff unit that can be changed at will.

Pros and cons

Pros : possibility to separate close points by increasing the precision.

Cons : not exactly continuous neither natural.

Thicken, adaptative version

Input

- $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- $t \in [0, 1]$,
- (z, r, A) a $\frac{1}{8}$-Moore box for g_t.

Output

$\delta > 0$ s.t. for all $s \in T = [t, t + \delta]$, (z, r, A) is a $\frac{7}{8}$-Moore box for g_s.

Procedure

```
def thicken(g, t, δ_hint, z, r, A):
    δ ← δ_hint;    T ← [t, t + δ]
    while not M(□g_T, □Jg_T, z, r, A, 7/8):
        δ ← δ/2;    T ← [t, t + δ]
    return δ
```

Thicken, adaptative version

Input

- $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- $t \in [0, 1]$,
- (z, r, A) a $\frac{1}{8}$-Moore box for g_t.

Output

$\delta > 0$ s.t. for all $s \in T = [t, t + \delta]$, (z, r, A) is a $\frac{7}{8}$-Moore box for g_s.

Procedure

```
def thicken(g, t, δ_hint, z, r, A):
    δ ← δ_hint;   T ← [t, t + δ]
    while not M(□g_T, □Jg_T, z, r, A, 7/8):
        δ ← δ/2;   T ← [t, t + δ]
        if δ < u:
            inscrease working precision
    return δ
```

Thicken, adaptative version

Input

- $g : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$,
- $t \in [0, 1]$,
- (z, r, A) a $\frac{1}{8}$-Moore box for g_t.

Output

$\delta > 0$ s.t. for all $s \in T = [t, t + \delta]$, (z, r, A) is a $\frac{7}{8}$-Moore box for g_s.

Proposition

thicken terminates and is correct.

Algorithm

```
def thicken(g, t, δ_hint, z, r, A):
    δ ← δ_hint;    T ← [t, t + δ]
    while not M(□g_T, □Jg_T, z, r, A, 7/8):
        δ ← δ/2;    T ← [t, t + δ]
        if δ < u:
            inscrease working precision
    return δ
```

Refine, adaptative version

Procedure

def *refine*(f, z, r, A):
    $U \leftarrow A$
    while not $M(\Box f, \Box Jf, z, r, A, \frac{1}{8})$:
        if $-A\Box f(z) \subseteq \frac{1}{64} B_r$: # l.h.s is small
            $r \leftarrow \frac{r}{2}$
        else: # l.h.s is big
            $z \leftarrow z - Uf(z)$
            $A \leftarrow Jf(z)^{-1}$
    return z, r, A

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Output

(z', r', A') a $\frac{1}{8}$-moore box for f with same associated zero as (z, r, A).

Refine, adaptative version

Procedure

```
def refine(f, z, r, A):
    U ← A;  s ← r
    while not M(□f, □Jf, z, r, A, 1/8):
        if −A□f(z) ⊆ (1/64) B_r:  # l.h.s is small
            r ← r/2
            if r < (1/128) s:
                reduce u enough so that u = o(r)
        else:  # l.h.s is big
            z ← z − Uf(z)
        A ← Jf(z)^{−1}
    return z, r, A
```

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Output

(z', r', A') a $\frac{1}{8}$-moore box for f with same associated zero as (z, r, A).

Refine, adaptative version

Procedure

```
def refine(f, z, r, A):
    U ← A;  s ← r
    while not M(□f, □Jf, z, r, A, 1/8):
        if −A□f(z) ⊆ (1/64) B_r:  # l.h.s is small
            r ← r/2
            if r < (1/128) s:
                reduce u enough so that u = o(r)
        else:  # l.h.s is big
            δ ← U□f(z)
            if width(z − δ) > (1/40) ||δ||_□:
                reduce u
            else:
                z ← mid(z − δ)
        A ← Jf(z)^{−1}
    return z, r, A
```

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Output

(z', r', A') a $\frac{1}{8}$-moore box for f with same associated zero as (z, r, A).

Refine, adaptative version

Algorithm

```
def refine(f, z, r, A):
    U ← A;  s ← r
    while not M(□f, □Jf, z, r, A, 1/8):
        if −A□f(z) ⊆ (1/64) B_r:  # l.h.s is small
            r ← r/2
            if r < (1/128) s:
                reduce u enough so that u = o(r)
        else:  # l.h.s is big
            δ ← U□f(z)
            if width(z − δ) > (1/40) ||δ||_□:
                reduce u
            else:
                z ← mid(z − δ)
        A ← Jf(z)^{−1}
    return z, r, A
```

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Output

(z', r', A') a $\frac{1}{8}$-moore box for f with same associated zero as (z, r, A).

Proposition

refine terminates and is correct.

Refine, adaptative version

Algorithm

def $refine(f, z, r, A)$:

$U \leftarrow A$; $s \leftarrow r$

while not $M(\square f, \square Jf, z, r, A, \frac{1}{8})$:

if $-A\square f(z) \subseteq \frac{1}{64} B_r$: # l.h.s is small

$r \leftarrow \frac{r}{2}$

if $r < \frac{1}{128} s$:

reduce u enough so that $u = o(r)$

else: # l.h.s is big

$\delta \leftarrow U\square f(z)$

if $width(z - \delta) > \frac{1}{40} \|\delta\|_\square$:

reduce u

else:

$z \leftarrow mid(z - \delta)$

$A \leftarrow Jf(z)^{-1}$

return z, r, A

Input

- $f : \mathbb{C}^n \to \mathbb{C}^n$,
- (z, r, A) a $\frac{7}{8}$-Moore box for f.

Output

(z', r', A') a $\frac{1}{8}$-moore box for f with same associated zero as (z, r, A).

Proposition

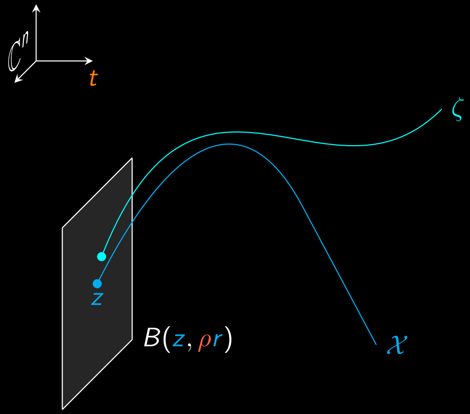
refine terminates and is correct.

Theorem

track terminates and is correct.

Optimizing the prediction

Thickening with a predictor

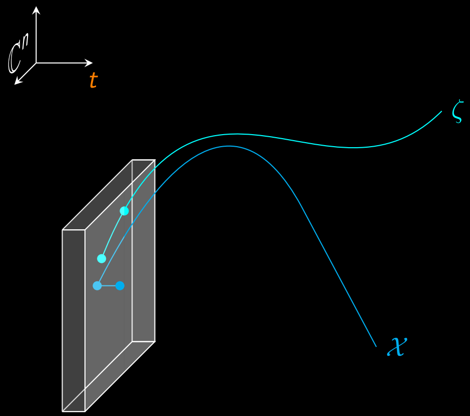


Predictor

A map $\mathcal{X} : \mathbb{R} \to \mathbb{C}^n$ such that $\mathcal{X}(0) = z$.

In practice, one should have $\mathcal{X}(s) \approx \zeta(t + s)$ around 0.

Thickening with a predictor



Predictor

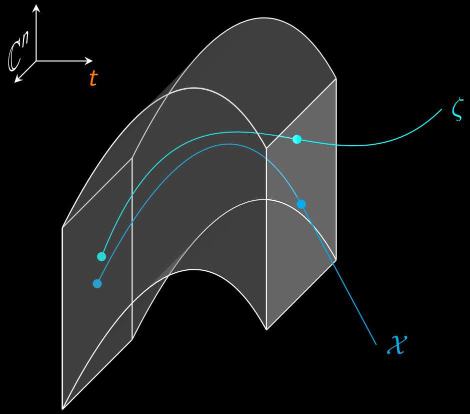
A map $\mathcal{X} : \mathbb{R} \to \mathbb{C}^n$ such that $\mathcal{X}(0) = z$.

In practice, one should have $\mathcal{X}(s) \approx \zeta(t + s)$ around 0.

Certifying the prediction

Pb : check that for all $s \in [0, \delta]$, (z, r, A) is a ρ-Moore box for g_{t+s}.
Soln : try $M(\square g_T, \square J g_T, z, r, A, \rho)$, where $T = [t, t + \delta]$.

Thickening with a predictor



Predictor

A map $\mathcal{X} : \mathbb{R} \to \mathbb{C}^n$ such that $\mathcal{X}(0) = z$.

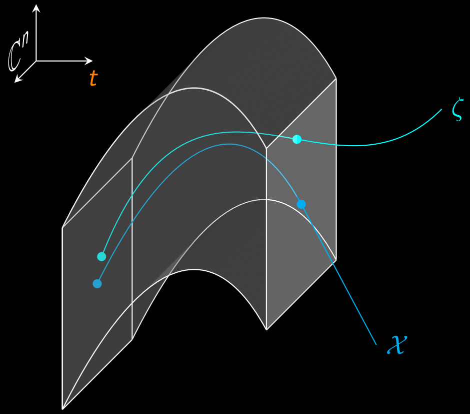
In practice, one should have $\mathcal{X}(s) \approx \zeta(t+s)$ around 0.

Certifying the prediction

Pb : check that for all $s \in [0, \delta]$, $(\mathcal{X}(s), r, A)$ is a ρ-Moore box for g_{t+s}.

Soln : try $M(\square g_T, \square J g_T, \square \mathcal{X}([0,\delta]), r, A, \rho)$, where $T = [t, t+\delta]$.

Thickening with a predictor



Predictor

A map $\mathcal{X} : \mathbb{R} \to \mathbb{C}^n$ such that $\mathcal{X}(0) = z$.

In practice, one should have $\mathcal{X}(s) \approx \zeta(t + s)$ around 0.

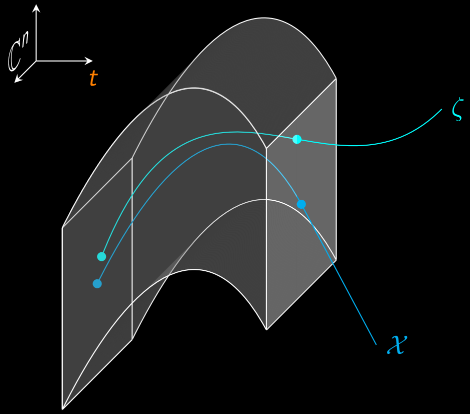
Certifying the prediction

Pb : check that for all $s \in [0, \delta]$, $(\mathcal{X}(s), r, A)$ is a ρ-Moore box for g_{t+s}.

Soln : try $M(\square g_T, \square J g_T, \square \mathcal{X}([0, \delta]), r, A, \rho)$, where $T = [t, t + \delta]$.

This is too strong !

Thickening with a predictor



Predictor

A map $\mathcal{X} : \mathbb{R} \to \mathbb{C}^n$ such that $\mathcal{X}(0) = z$.

In practice, one should have $\mathcal{X}(s) \approx \zeta(t+s)$ around 0.

Certifying the prediction

Pb : check that for all $s \in [0, \delta]$, $(\mathcal{X}(s), r, A)$ is a ρ-Moore box for g_{t+s}.
Soln : try $M(\Box g_T, \Box J g_T, \Box \mathcal{X}([0,\delta]), r, A, \rho)$, where $T = [t, t+\delta]$.

This is too strong !

Way around the dependency problem : Taylor models !

Taylor models with relative remainder

Definition

- An interval $S \in \square\mathbb{R}$ containing zero,
- a polynomial $P(\eta) = A_0 + A_1\eta + \cdots + A_{d+1}\eta^{d+1}$ where $A_i \in \square\mathbb{C}$.

d is the order of the Taylor model.

Taylor models with relative remainder

Definition

- An interval $S \in \square\mathbb{R}$ containing zero,
- a polynomial $P(\eta) = A_0 + A_1\eta + \cdots + A_{d+1}\eta^{d+1}$ where $A_i \in \square\mathbb{C}$.

d is the order of the Taylor model.

Definition

A Taylor model (S, P) encloses a function $f : \mathbb{R} \to \mathbb{C}$ if for all $s \in S$, there exists $a_i \in A_i$, for all $0 \leq i \leq d+1$ s.t. $f(s) = a_0 + a_1 s + \cdots + a_{d+1} s^{d+1}$

Taylor models with relative remainder

Definition

- An interval $S \in \square\mathbb{R}$ containing zero,
- a polynomial $P(\eta) = A_0 + A_1\eta + \cdots + A_{d+1}\eta^{d+1}$ where $A_i \in \square\mathbb{C}$.

d is the order of the Taylor model.

Definition

A Taylor model (S, P) encloses a function $f : \mathbb{R} \to \mathbb{C}$ if for all $s \in S$, there exists $a_i \in A_i$, for all $0 \leq i \leq d+1$ s.t. $f(s) = a_0 + a_1 s + \cdots + a_{d+1} s^{d+1}$

Remark

If $J \subseteq S$, then $f(J) \subseteq P(J)$.

Arithmetic

Reduction

Let (S, P) be a Taylor model of order d.

Goal : reduce it order to $d-1$, s.t. if (S, P) encloses a function, so does its reduction.

Solution : replace $A_d\eta^d + A_{d+1}\eta^{d+1}$ by $(A_d \boxplus (A_{d+1} \boxtimes I))\eta^d$.

Arithmetic

Reduction

Let (S, P) be a Taylor model of order d.

Goal : reduce it order to $d-1$, s.t. if (S, P) encloses a function, so does its reduction.

Solution : replace $A_d\eta^d + A_{d+1}\eta^{d+1}$ by $(A_d \boxplus (A_{d+1} \boxtimes I))\eta^d$.

Operations

Let (S, P) and (S, Q) be Taylor models of order d.

Sum : Component-wise sum using $\boxplus$. Compatible with sums of enclosed functions.

Product : Usual product formula, gives a Taylor model of order $2d+1$, then reduce it to make it of order d. Compatible with products of enclosed functions.

Back to our problem

Recall what we want

(z, r, A) is a $\frac{1}{8}$-Moore box for g_t, $\mathcal{X} : \mathbb{R} \to \mathbb{C}^n$ polynomial s.t. $\mathcal{X}(0) = z$, $\delta > 0$. We want to check that for all $s \in [0, \delta]$,

$$-Ag_{t+s}(\mathcal{X}(s)) + [\mathrm{I}_n - A \cdot Jg_{t+s}(\mathcal{X}(s) + B_r)]B_r \subseteq \frac{7}{8}B_r.$$

Back to our problem

Recall what we want

(z, r, A) is a $\frac{1}{8}$-Moore box for g_t, $\mathcal{X} : \mathbb{R} \to \mathbb{C}^n$ polynomial s.t. $\mathcal{X}(0) = z$, $\delta > 0$. We want to check that for all $s \in [0, \delta]$,

$$-Ag_{t+s}(\mathcal{X}(s)) + [\mathrm{I}_n - A \cdot Jg_{t+s}(\mathcal{X}(s) + B_r)]B_r \subseteq \frac{7}{8}B_r.$$

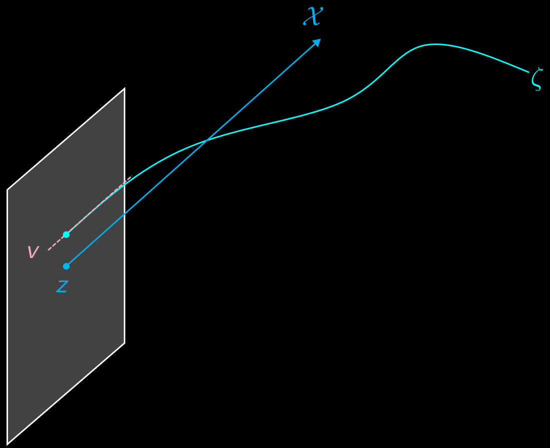
Solution using Taylor models

- Compute an order d taylor model $\mathcal{K}$ on $[0, \delta]$ of

$$-Ag_{t+\eta}(\mathcal{X}(\eta)) + [\mathrm{I}_n - A \cdot Jg_{t+\eta}(\mathcal{X}(\eta) + B_r)]B_r.$$

  This is just Taylor model arithmetic !
- Check that $\mathcal{K}([0, \delta]) \subseteq \frac{7}{8}B_r$ (interval arithmetic).

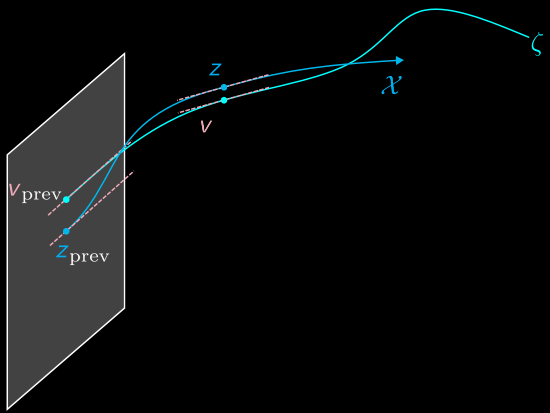
Choosing the right predictor



Tangent predictor

Idea : $-A \cdot \frac{\partial}{\partial t} g(t, z)$ is a good approximation of $\zeta'(t)$. Use it to do a order 1 correction.

Choosing the right predictor



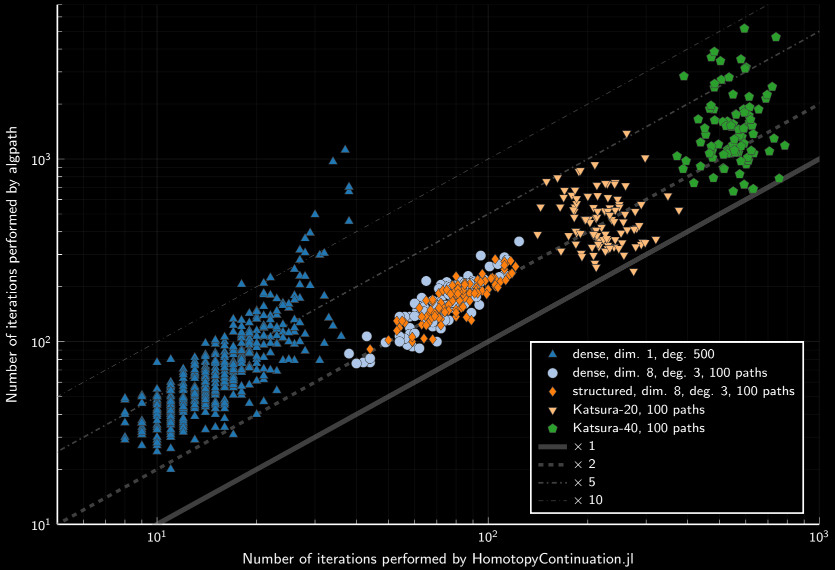
Tangent predictor

Idea : $-A \cdot \frac{\partial}{\partial t} g(t, z)$ is a good approximation of $\zeta'(t)$. Use it to do a order 1 correction.

Hermite predictor

Idea : use previous point z_{prev} and previous tangent vector v_{prev}, z and v to do a Hermite cubic spline approximation.

Implementation and benchmarks



References

Beltrán, Carlos and Anton Leykin. "Certified Numerical Homotopy Tracking". In: *Experimental Mathematics* 21.1 (Mar. 2012). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10586458.2011.606184, pp. 69–83.

Breiding, Paul and Sascha Timme. "HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia". en. In: *Mathematical Software – ICMS 2018*. Ed. by James H. Davenport et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 458–465.

Duff, Timothy and Kisun Lee. *Certified homotopy tracking using the Krawczyk method*. arXiv:2402.07053 [cs, math]. Feb. 2024.

Hauenstein, Jonathan D. and Alan C. Liddell. "Certified predictor–corrector tracking for Newton homotopies". In: *Journal of Symbolic Computation* 74 (May 2016), pp. 239–254.

Hoeven, Joris van der. *Reliable homotopy continuation*. Research Report. LIX, Ecole polytechnique, Jan. 2015.

Kearfott, R. Baker and Zhaoyun Xing. "An Interval Step Control for Continuation Methods". In: *SIAM Journal on Numerical Analysis* 31.3 (June 1994). Publisher: Society for Industrial and Applied Mathematics, pp. 892–914.

Kranich, Stefan. *An epsilon-delta bound for plane algebraic curves and its use for certified homotopy continuation of systems of plane algebraic curves*. arXiv:1505.03432 [math]. May 2016.

Marco-Buzunariz, Miguel Ángel and Marcos Rodríguez. "SIROCCO: A Library for Certified Polynomial Root Continuation". en. In: *Mathematical Software – ICMS 2016*. Ed. by Gert-Martin Greuel et al. Vol. 9725. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 191–197.

Moore, R. E. "A Test for Existence of Solutions to Nonlinear Systems". In: *SIAM Journal on Numerical Analysis* 14.4 (1977). Publisher: Society for Industrial and Applied Mathematics, pp. 611–615.

Verschelde, Jan. "Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation". In: *ACM Transactions on Mathematical Software* 25.2 (June 1999), pp. 251–276.

Test data

We tested systems of the form $g_t(z) = tf^{\odot}(z) + (1-t)f^{\triangleright}(z)$ ($f^{\triangleright}$ is the start system, $f^{\odot}$ is the target system).

Test data

We tested systems of the form $g_t(z) = tf^{\odot}(z) + (1-t)f^{\triangleright}(z)$ ($f^{\triangleright}$ is the start system, $f^{\odot}$ is the target system).

Target systems

- Dense : $f_i^{\odot}$'s of given degree with random coefficients
- Structured : $f_i^{\odot}$'s of the form $\pm 1 + \sum_{i=1}^{5} \left(\sum_{j=1}^{n} a_{i,j} z_j \right)^d$, $a_{i,j} \in_R \{-1, 0, 1\}$
- Katsura family (sparse - high dimension - low degree)

Test data

We tested systems of the form $g_t(z) = tf^{\odot}(z) + (1-t)f^{\triangleright}(z)$ ($f^{\triangleright}$ is the start system, $f^{\odot}$ is the target system).

Target systems

- Dense : $f_i^{\odot}$'s of given degree with random coefficients
- Structured : $f_i^{\odot}$'s of the form $\pm 1 + \sum_{i=1}^{5} \left(\sum_{j=1}^{n} a_{i,j} z_j \right)^d$, $a_{i,j} \in_R \{-1, 0, 1\}$
- Katsura family (sparse - high dimension - low degree)

Start systems

- Total degree homotopies : $f_i^{\triangleright}$'s of the form $\gamma_i(z_i^{d_i} - 1)$, $\gamma_i \in_R \mathbb{C}$, $d_i = \deg f_i^{\odot}$
- Newton homotopies : $f^{\triangleright}(z) = f^{\odot}(z) - f^{\odot}(z_0)$

Benchmarks table

			HomotopyContinuation.jl			algpath			Macaulay2		
name	dim.	max deg	fail.	med.	time (s)	fail.	med.	time (s)	fail.	med.	time (s)
dense	1	10		6	1.8		11	< 0.1		629	0.2
dense	1	30		10	2.0		23	< 0.1		830 k	18 min
dense	1	50		12	1.9		30	0.7		$>$ 1 h	
dense	2	10		22	2.6		53	0.7		33 k	158
dense	2	30		24	6.4		85	72		$>$ 1 h	
dense	2	50		27	33		117	12 min		$>$ 1 h	
katsura	11	2		100	6.7		177	30		21 k	30 min
katsura	21	2		209	4 h	483	427	101 h		not benchmarked	
katsura *	41	2		554	24	9	1371	13 min		$>$ 1 h	
dense *	8	3		73	6.3		157	19		21 k	243
structured *	8	3		81	3.9		182	2.3		36 k	305
structured ^{N}	10	10		53	3.1		123	< 0.1		$>$ 1 h	
structured ^{N}	20	20		$>$ 8 GB			1591	1.5		$>$ 8 GB	
structured ^{N}	30	30		$>$ 8 GB			1989	5.2		$>$ 8 GB	