Braid computations using certified path tracking

Alexandre Guillemot, joint work with Pierre Lairez
MATHEXP, Université Paris–Saclay, Inria, France

May 20, 2025 | Laboratoire de Mathématiques de Versailles

$$F$$

Parametrized polynomial system

Certified homotopy continuation

Input: F

Point in $\mathbb{C}^n$

$$F_0(\zeta_0) = 0$$

Parametrized polynomial system

Certified homotopy continuation

Input: F, ζ_0

Introduction

Unique continuous extension

$$F_t(\zeta_t) = 0, \quad \forall t \in [0, 1]$$

Parametrized polynomial system

Certified homotopy continuation

Input: F, ζ_0

Unique continuous extension

$$F_t(\zeta_t) = 0, \quad \forall t \in [0, 1]$$

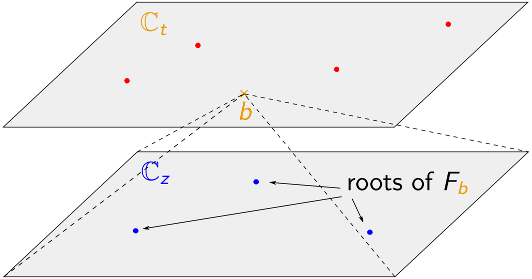
Parametrized polynomial system

Certified homotopy continuation

Input: F, ζ_0

Output: A "certified approximation" of ζ

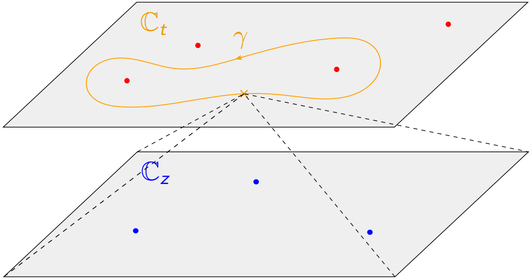
Motivation: braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- define $F_t(z) = g(t, z)$.
- Let $b \in \mathbb{C} \backslash \Sigma$ be a base point,

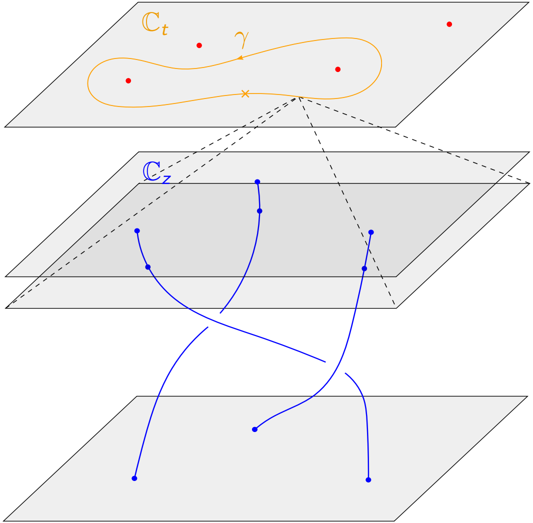
Motivation: braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- define $F_t(z) = g(t, z)$.
- 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.

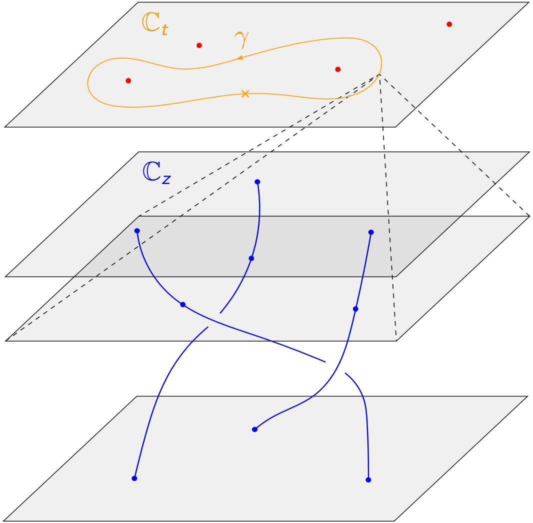
Motivation: braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- define $F_t(z) = g(t, z)$.
- 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 F_t when t moves along γ defines a braid.

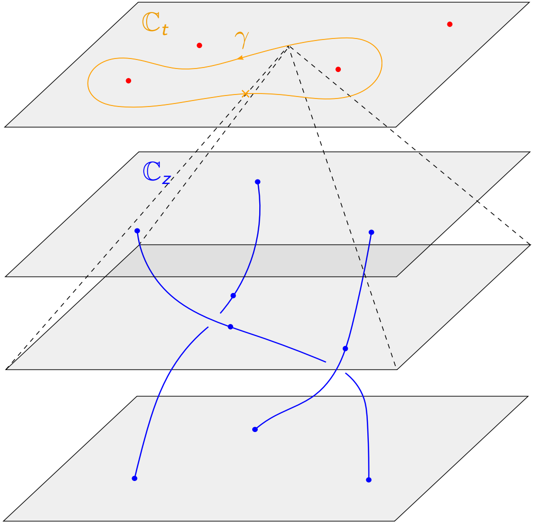
Motivation: braid computations



Setup

- Let $g \in \mathbb{C}[t,z]$,
- define $F_t(z) = g(t,z)$.
- 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 F_t when t moves along γ defines a braid.

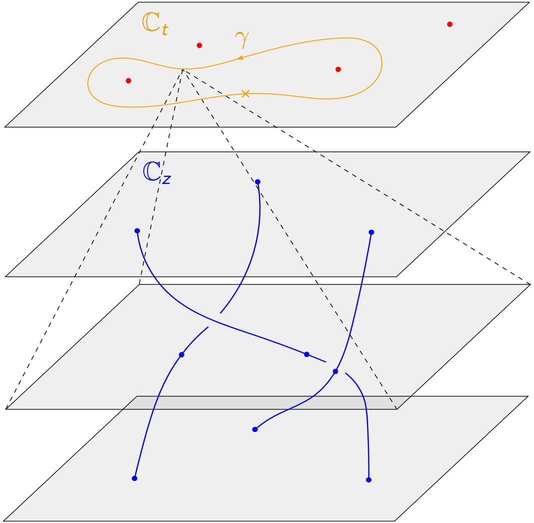
Motivation: braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- define $F_t(z) = g(t, z)$.
- 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 F_t when t moves along γ defines a braid.

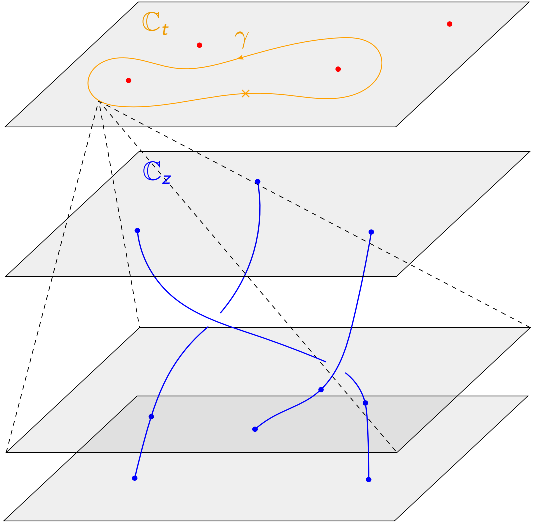
Motivation: braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- define $F_t(z) = g(t, z)$.
- 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 F_t when t moves along γ defines a braid.

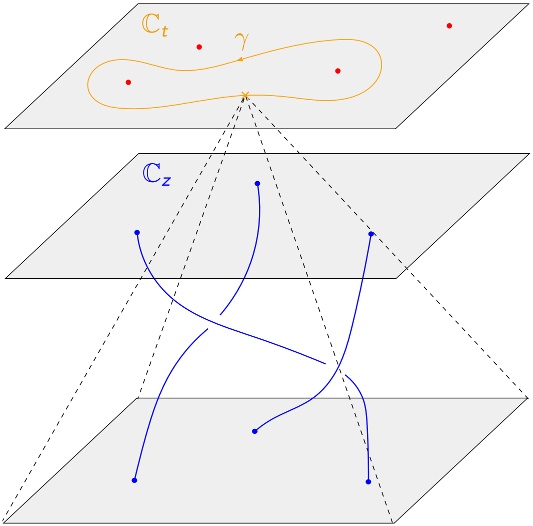
Motivation: braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- define $F_t(z) = g(t, z)$.
- 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 F_t when t moves along γ defines a braid.

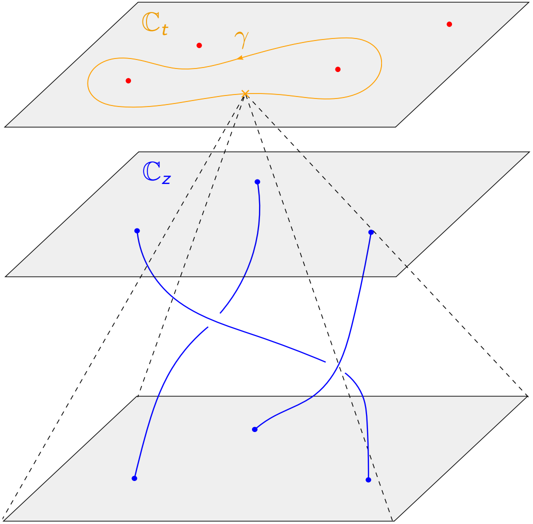
Motivation: braid computations



Setup

- Let $g \in \mathbb{C}[t, z]$,
- define $F_t(z) = g(t, z)$.
- 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 F_t when t moves along γ defines a braid.

Motivation: braid computations



Setup

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

Algorithmic goal

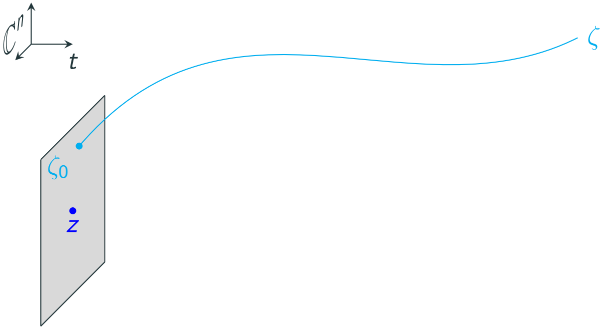
Input: g, γ

Output: the associated braid

Tool: certified path tracking

Generic corrector-predictor loop

Recall: for all $t \in [0,1]$, $F_t(\zeta_t) = 0$

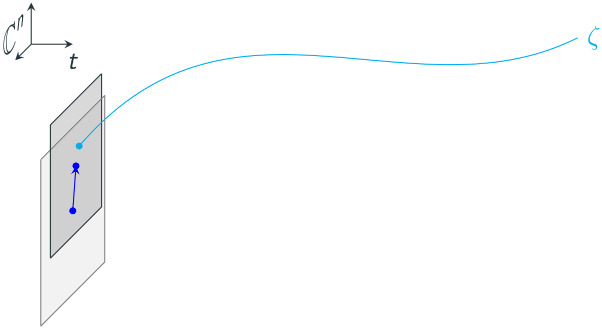


def $\texttt{track}(F, z)$:

$t \leftarrow 0$; $\quad L \leftarrow []$
while $t < 1$:
$\quad z \leftarrow \texttt{refine}(F_t, z)$
$\quad \delta \leftarrow \texttt{validate}(F, t, z)$
$\quad t \leftarrow t + \delta$
$\quad$ append (t, z) to L
return L

Generic corrector-predictor loop

Recall: for all $t \in [0,1]$, $F_t(\zeta_t) = 0$

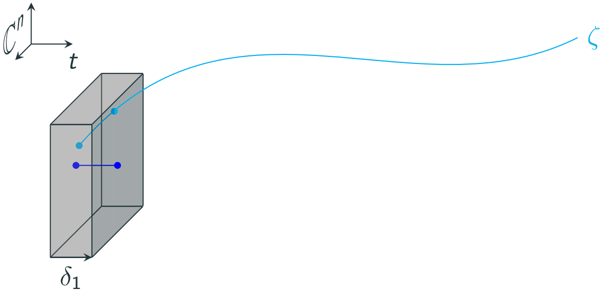


def $\texttt{track}(F, z)$**:**

$t \leftarrow 0; \quad L \leftarrow []$
while $t < 1$**:**
  $z \leftarrow \texttt{refine}(F_t, z)$
  $\delta \leftarrow \texttt{validate}(F, t, z)$
  $t \leftarrow t + \delta$
  append (t, z) to L
return L

Generic corrector-predictor loop

Recall: for all $t \in [0,1]$, $F_t(\zeta_t) = 0$



def $\texttt{track}(F, z)$**:**

$t \leftarrow 0; \quad L \leftarrow []$

while $t < 1$**:**

  $z \leftarrow \texttt{refine}(F_t, z)$

  $\delta \leftarrow \texttt{validate}(F, t, z)$

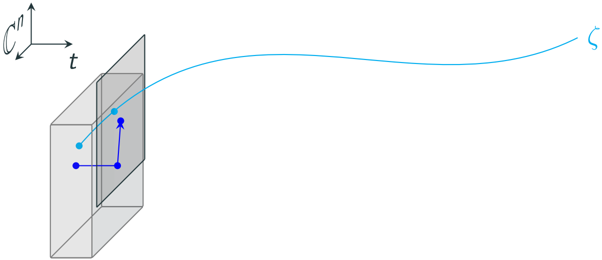
  $t \leftarrow t + \delta$

  append (t, z) to L

return L

Generic corrector-predictor loop

Recall: for all $t \in [0, 1]$, $F_t(\zeta_t) = 0$

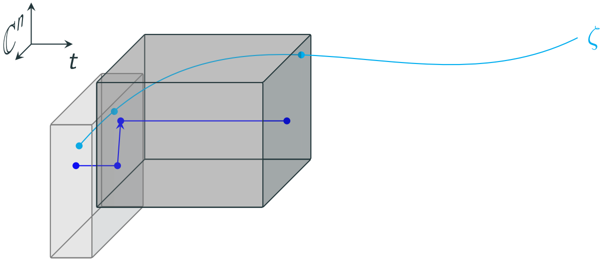


def track(F, z):

$t \leftarrow 0$; $L \leftarrow []$
while $t < 1$:
    $z \leftarrow$ refine(F_t, z)
    $\delta \leftarrow$ validate(F, t, z)
    $t \leftarrow t + \delta$
    append (t, z) to L
return L

Generic corrector-predictor loop

Recall: for all $t \in [0,1]$, $F_t(\zeta_t) = 0$



def `track`(F, z):

$t \leftarrow 0$; $\quad L \leftarrow []$

while $t < 1$:

    $z \leftarrow \texttt{refine}(F_t, z)$

    $\delta \leftarrow \texttt{validate}(F, t, z)$

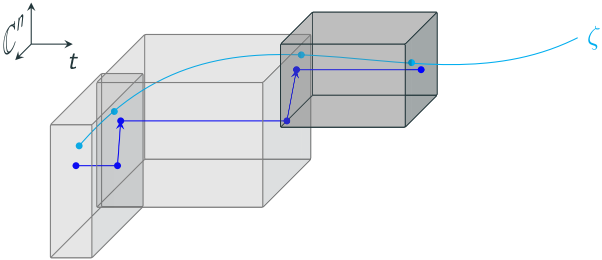
    $t \leftarrow t + \delta$

    append (t, z) to L

return L

Generic corrector-predictor loop

Recall: for all $t \in [0, 1]$, $F_t(\zeta_t) = 0$



def `track`(F, z)**:**

$t \leftarrow 0$; $L \leftarrow []$

while $t < 1$**:**

  $z \leftarrow$ `refine`(F_t, z)

  $\delta \leftarrow$ `validate`(F, t, z)

  $t \leftarrow t + \delta$

  append (t, z) to L

return L

Main tool: interval arithmetic

Problem

Given $f \in \mathbb{R}[x]$, I and J intervals, check $f(\mathrm{I}) \subseteq \mathrm{J}$.

Sufficient solution

- Define interval binary operations $\boxplus$ and $\boxtimes$ that take two intervals, give an interval and is such that for all $x \in \mathrm{A}$, $y \in \mathrm{B}$,

$$x + y \in \mathrm{A} \boxplus \mathrm{B},\ xy \in \mathrm{A} \boxtimes \mathrm{B}$$

- Write f as a composition of binary operations and replace each operation by its interval counterpart (**interval extension**, denoted by $\square f$), then plug I and check if the result is contained in J (as $f(I) \subseteq \square f(I)$).

❗ This is only a sufficient condition

Main tool: interval arithmetic

Rational enpoints interval arithmetic

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

Main tool: interval arithmetic

Rational enpoints interval arithmetic

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

$f = x^2 - x + 2$, $I = [0, 1]$

- If we decompose f as $(x \cdot x - x) + 2$, we get $[1, 3]$.
- If we decompose f as $x \cdot (x - 1) + 2$, we get $[1, 2]$.
- Actually, $f([0, 1]) = [1.75, 2]$.

! Decomposition dependant, dependency issue, overestimation

! Coefficient swell (we will adress this problem later...)

Moore boxes, the datastructure for isolating boxes

Root isolation criterion [Krawczyk, 1969], [Moore, 1977], [Rump, 1983]

- $f : \mathbb{C}^n \to \mathbb{C}^n$ polynomial, $\rho \in (0, 1)$,
- $z \in \mathbb{C}^n$, $A \in \mathbb{C}^{n\times n}$, $B \subseteq \mathbb{C}^n$ a ball of center 0,

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

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

Then f has a unique zero in $z + \rho B$.

Moore boxes, the datastructure for isolating boxes

Root isolation criterion [Krawczyk, 1969], [Moore, 1977], [Rump, 1983]

- $f : \mathbb{C}^n \to \mathbb{C}^n$ polynomial, $\rho \in (0, 1)$,
- $z \in \mathbb{C}^n$, $A \in \mathbb{C}^{n\times n}$, $B \subseteq \mathbb{C}^n$ a ball of center 0,

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

Then f has a unique zero in $z + \rho B$.

Root isolation criterion [Krawczyk, 1969], [Moore, 1977], [Rump, 1983]

- $f : \mathbb{C}^n \to \mathbb{C}^n$ polynomial, $\rho \in (0, 1)$,
- $z \in \mathbb{C}^n$, $A \in \mathbb{C}^{n\times n}$, $B \subseteq \mathbb{C}^n$ a ball of center 0,

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

Then f has a unique zero in $z + \rho B$.

Proof sketch

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

Definition

A ρ-Moore box for f is a triple (z, B, A) which satisfies Moore's criterion.

Refine

Input: $f : \mathbb{C}^n \to \mathbb{C}^n$ polynomial; z, B, A a $\frac{7}{8}$-Moore box for f.

Output: A $\frac{1}{8}$-Moore box for f with same associated zero as z, B, A.

def $\texttt{refine}(f, z, B, A)$**:**

$U \leftarrow A$

while not $-A \cdot \square f(z) + [I - A \cdot \square Jf(z + B)]\, B \subseteq \frac{1}{8} B$**:**

  if left term is small compared to $\frac{1}{8} B$**:**

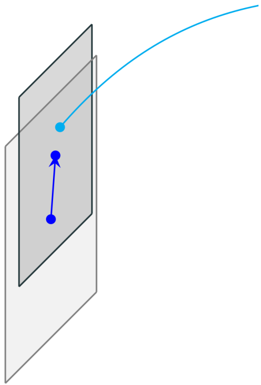
    shrink B

  else:

    $z \leftarrow z - Uf(z)$

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

return z, B, A

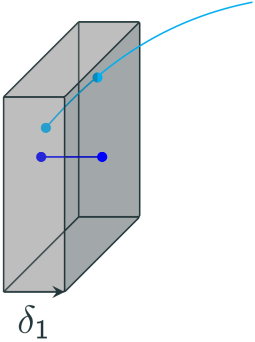


Validate

Input: $F : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$ polynomial; $t \in [0, 1]$; z, B, A a $\frac{1}{8}$-Moore box for F_t.
Output: $\delta > 0$ such that for all $s \in [t, t + \delta]$, z, B, A is a $\frac{7}{8}$-Moore box for F_s.

def `validate`(F, t, z, B, A):

$\delta \leftarrow 1$
$T \leftarrow [t, t + \delta]$
while $-A \cdot \square F_T(z) + [I - A \cdot \square JF_T(z + B)]\, B \not\subseteq \frac{7}{8} B$:
    $\delta \leftarrow \frac{\delta}{2}$
    $T \leftarrow [t, t + \delta]$
return δ



Important aspects that I did not adress

Model

- ✗ Rational enpoints: coefficient swell
- ✗ 64-bits float interval arithmetic: bad theoritical properties
- ✓ Arbitrary precision interval arithmetic: precision management allows for termination

  It requires careful handling!

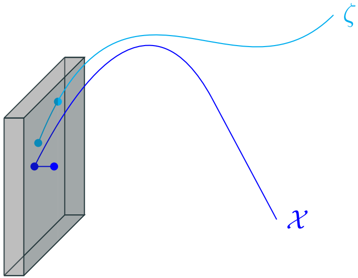
Important aspects that I did not adress

Model

- ✗ Rational enpoints: coefficient swell
- ✗ 64-bits float interval arithmetic: bad theoritical properties
- ✓ Arbitrary precision interval arithmetic: precision management allows for termination
  It requires careful handling!

Predictor

- ✗ Check that (z, B, A) is a Mb for F_t on T



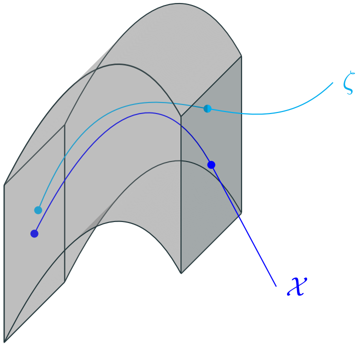
Important aspects that I did not adress

Model

- ✗ Rational enpoints: coefficient swell
- ✗ 64-bits float interval arithmetic: bad theoritical properties
- ✓ Arbitrary precision interval arithmetic: precision management allows for termination
  It requires careful handling!

Predictor

- ✗ Check that (z, B, A) is a Mb for F_t on T
- ✓ Check that $(\mathcal{X}(t), B, A)$ is a Mb for F_t on T.
- ! Replacing T by $\mathcal{X}(T)$ in `validate` does not work: computing directly $F_T(\mathcal{X}(T))$ suffers from the dependency problem (though $F_s(\mathcal{X}(s)) \approx 0$)
- 💡 Taylor models: “expand then evaluate”



Implementation

Algpath is a Rust implementation of the presented algorithm.

Generic implementation with respect to the interval arithmetic. Can use:

- 64-bit float endpoints. ➕ fast ➖ cannot increase precision
- Arb [1] (adaptive precision interval arithmetic library). ➕ can manage precision ➖ slower than float endpoints

Mixed precision

At each iteration of the main loop, chooses between float endpoints or Arb for the next corrector predictor round, depending on precision. ➕ little overhead over float endpoints

Predictor

Hermite's cubic predictor (degree 3)

[1] Johansson. *"Arb: Efficient Arbitrary-Precision Midpoint-Radius Interval Arithmetic"*.

Benchmarks

name	dim	degree	HomotopyContinuation.jl time (s)	HomotopyContinuation.jl failures	HomotopyContinuation.jl max steps	Algpath time (s)	Algpath max prec.	Algpath max steps
dense	1	1000	**11**		100	**27 min**	59	24 k
dense	1	2000	**48**	**3**	79	**2 h**	62	71 k
katsura	21	2^{20}	**5 h**		468	**89 h**	65	13 k
resultants	3	$16 \cdot 4^2$	**7.6**		128	**160**	58	3510
resultants	2	$40 \cdot 5$	**4.2**	**200**		**13 min**	69	2205
structured *	3	20^3	**3.7**	**12**	164	**6.3**	56	634
structured *	3	30^3	**3.5**	**92**	133	**97**	71	820
W_{20}	1	20	**2.4**	**20**		**3 h**	118	2316 k
clustered (10, 5, 10)	1	50	**3.8**		153	**153**	70	17 k

Figure 1: Total degree homotopy benchmarks. A * means that only 100 random roots were tracked.

[2] Breiding and Timme. *"HomotopyContinuation.Jl: A Package for Homotopy Continuation in Julia"*.

Introduction to braids

Definition

The configuration space on n elements C_n is $\{c \subset \mathbb{C} : \mathrm{card}(c) = n\}$ (we call its elements configurations). The (geometric) braid group is $B_n = \pi_1(C_n, \{1, \cdots, n\})$.

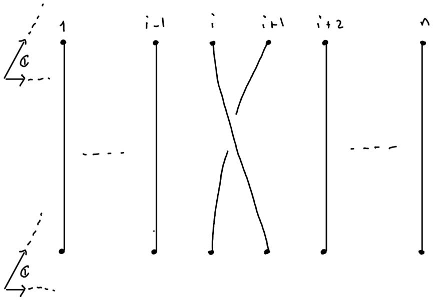


Figure 2: Standard generator σ_i

Theorem [Artin, 1947]

The σ_i's generate B_n (+ explicit relations).

Goal

Given some data representing a braid (e.g. output of Algpath), decompose it in terms of the σ_i's.

Computing a decomposition

Input: $\gamma_1, \cdots, \gamma_n$ continuous paths in $\mathbb{C}$ such that $\gamma_i(t) \neq \gamma_j(t)$ for all $i \neq j$ and $t \in [0,1]$
Output: a decomposition in standard generators of the braid induced by the input

Naive idea

Project on the real axis, process crossings in order.

- ! Multiple/non transversal crossings
- ! The output of Algpath is a piecewise tubular neighborhood around each strand. Each piece follows a degree three curve.

Work in progress

- ✓ Refine the naive approach
- ⚙ Prove correction and termination
- ⚙ Implement and pipe the output of Algpath

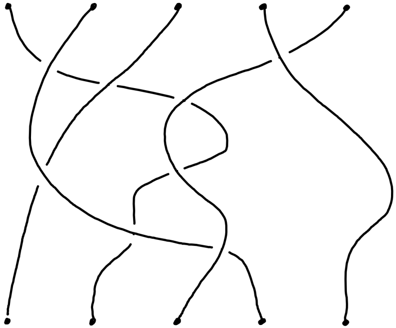


Figure 3: $\sigma_4\sigma_1^{-1}\sigma_2^{-1}\sigma_3^{-1}\sigma_3\sigma_1\sigma_2\sigma_3^{-1}$

References i

Artin, E. (1947). Theory of Braids. *Annals of Mathematics*, *48*(1), 101–126.

Breiding, P., & Timme, S. (2018). HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In J. H. Davenport, M. Kauers, G. Labahn, & J. Urban (Eds.), *Mathematical Software – ICMS 2018* (pp. 458–465). Springer International Publishing.

Johansson, F. (2017). Arb: Efficient Arbitrary-Precision Midpoint-Radius Interval Arithmetic. *IEEE Transactions on Computers*, *66*(8), 1281–1292.

Krawczyk, R. (1969). Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing*, *4*(3), 187–201.

Moore, R. E. (1977). A Test for Existence of Solutions to Nonlinear Systems. *SIAM Journal on Numerical Analysis*, *14*(4), 611–615. Retrieved February 19, 2024, from https://www.jstor.org/stable/2156481

Rump, S. M. (1983). SOLVING ALGEBRAIC PROBLEMS WITH HIGH ACCURACY. In U. W. Kulisch & W. L. Miranker (Eds.), *A New Approach to Scientific Computation* (pp. 51–120). Academic Press.

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: $f^{\triangleright}(z) = f^{\odot}(z) - f^{\odot}(z_0)$