Braid monodromy computations using certified path tracking

Alexandre Guillemot & Pierre Lairez
MATHEXP, Université Paris–Saclay, Inria, France

Aca 2025
July 17, 2025 | Cultural Conference Center of Heraklion, Greece

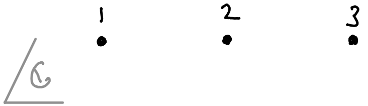
Warmup: braid group

Definitions

$OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \forall i \neq j,\ x_i \neq x_j\}$.

$C_n = \{\text{subsets of size } n \text{ in } \mathbb{C}\}$.

$(x_1, \cdots, x_n) \in OC_n \mapsto \{x_1, \cdots, x_n\} \in C_n$.



Warmup: braid group

Definitions

$OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \forall i \neq j,\ x_i \neq x_j\}$.

$C_n = \{\text{subsets of size } n \text{ in } \mathbb{C}\}$.

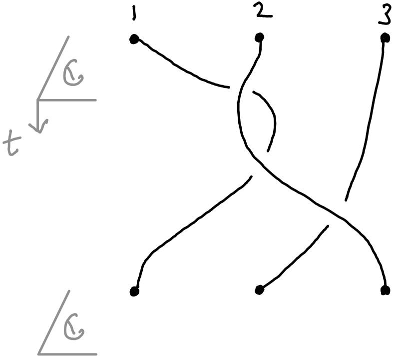
$(x_1, \cdots, x_n) \in OC_n \mapsto \{x_1, \cdots, x_n\} \in C_n$.

Braids

A braid is a homotopy class of a path

$\beta : [0,1] \to C_n$ such that

$\beta(0) = \beta(1) = \{1, \cdots, n\}$,



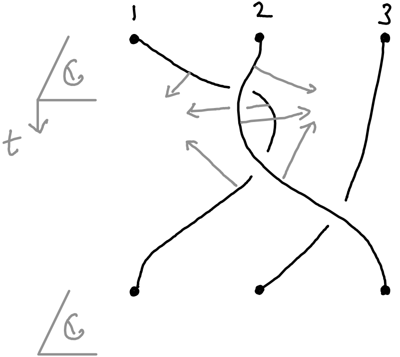
Warmup: braid group

Definitions

$OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \forall i \neq j,\, x_i \neq x_j\}$.
$C_n = \{$subsets of size n in $\mathbb{C}\}$.
$(x_1, \cdots, x_n) \in OC_n \mapsto \{x_1, \cdots, x_n\} \in C_n$.

Braids

A braid is a **homotopy class** of a path
$\beta : [0, 1] \to C_n$ such that
$\beta(0) = \beta(1) = \{1, \cdots, n\}$,



Warmup: braid group

Definitions

$OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \forall i \neq j,\ x_i \neq x_j\}.$

$C_n = \{\text{subsets of size } n \text{ in } \mathbb{C}\}.$

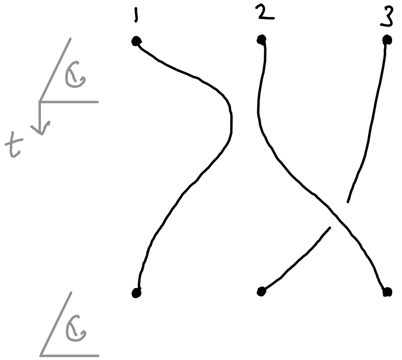
$(x_1, \cdots, x_n) \in OC_n \mapsto \{x_1, \cdots, x_n\} \in C_n.$

Braids

A braid is a **homotopy class** of a path

$\beta : [0,1] \to C_n$ such that

$\beta(0) = \beta(1) = \{1, \cdots, n\},$



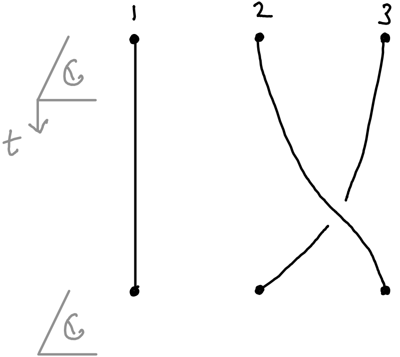
Warmup: braid group

Definitions

$OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \forall i \neq j,\ x_i \neq x_j\}.$
$C_n = \{\text{subsets of size } n \text{ in } \mathbb{C}\}.$
$(x_1, \cdots, x_n) \in OC_n \mapsto \{x_1, \cdots, x_n\} \in C_n.$

Braids

A braid is a **homotopy class** of a path
$\beta : [0, 1] \to C_n$ such that
$\beta(0) = \beta(1) = \{1, \cdots, n\},$



Warmup: braid group

Definitions

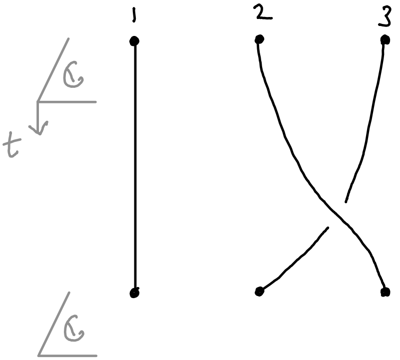
$OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \forall i \neq j,\, x_i \neq x_j\}$.
$C_n = \{\text{subsets of size } n \text{ in } \mathbb{C}\}$.
$(x_1, \cdots, x_n) \in OC_n \mapsto \{x_1, \cdots, x_n\} \in C_n$.

Braids

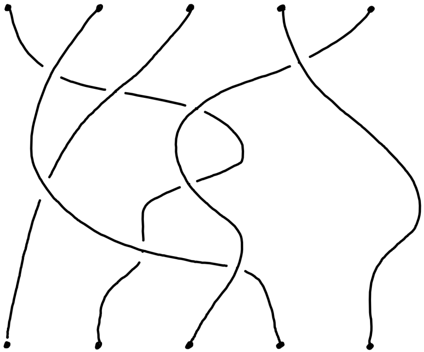
A braid is a **homotopy class** of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \cdots, n\}$,

Remark

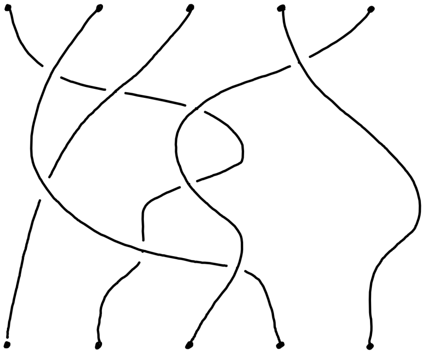
A path $\zeta = (\zeta_1, \cdots, \zeta_n) : [0,1] \to OC_n$ induces a braid. **If $\zeta' : [0,1] \to OC_n$ is homotopic to ζ, they have the same associated braid**

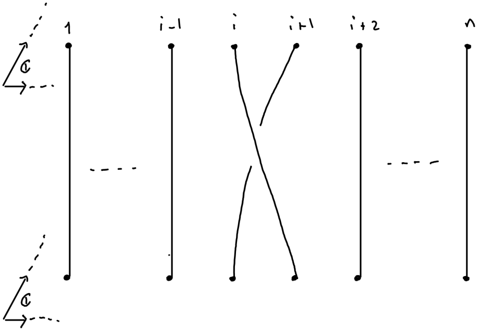


Examples and Artin's theorem



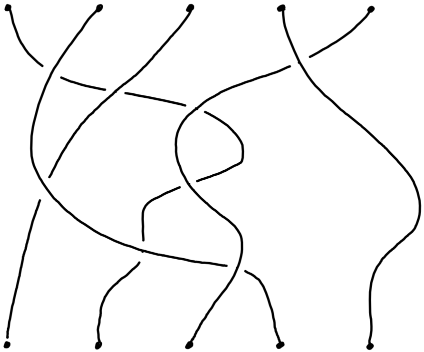
Examples and Artin's theorem

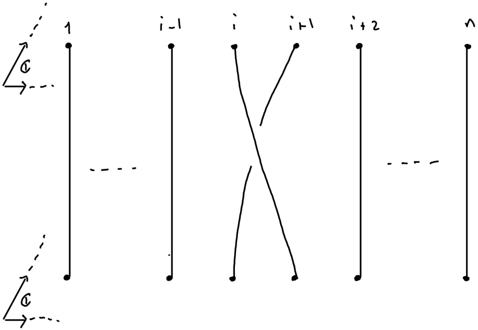




Standard generator σ_i

Examples and Artin's theorem



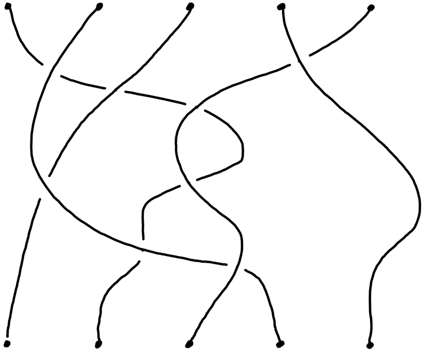


Standard generator σ_i

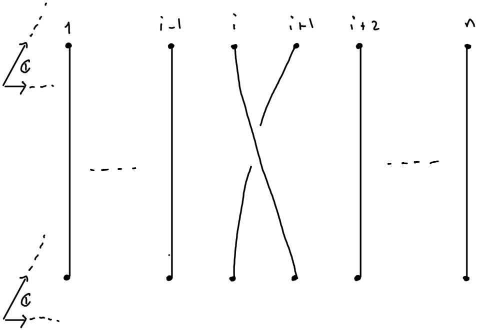
Theorem [Artin, 1947]

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

Examples and Artin's theorem



$\sigma_4\sigma_1^{-1}\sigma_2^{-1}\sigma_3^{-1}\sigma_3\sigma_1\sigma_2\sigma_3^{-1}$

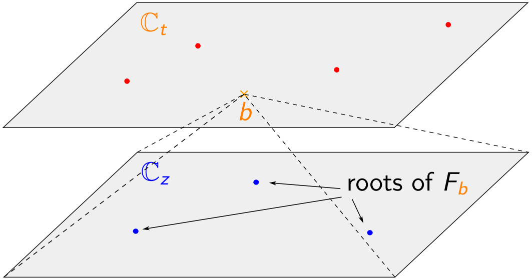


Standard generator σ_i

Theorem [Artin, 1947]

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

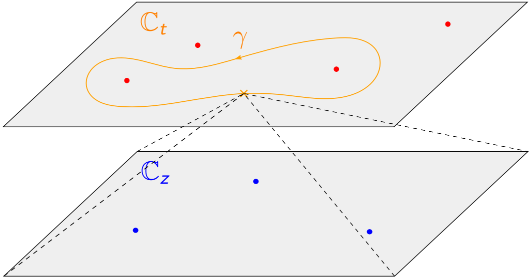
Another example



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,

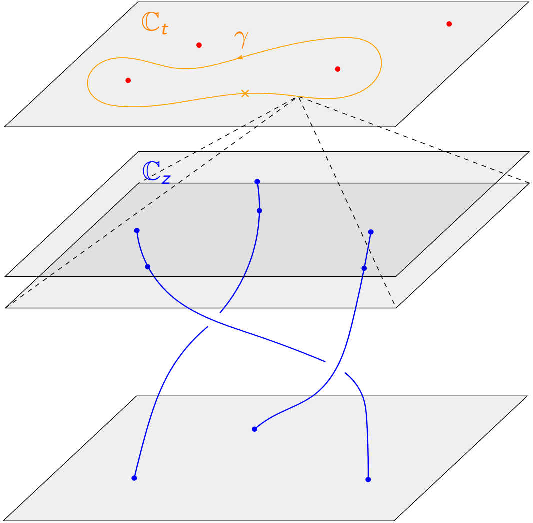
Another example



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.

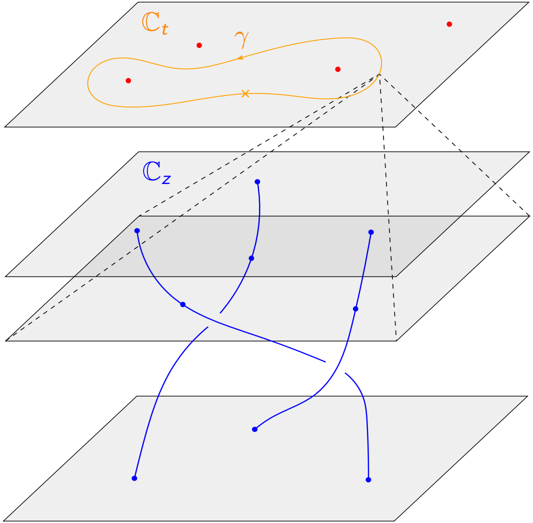
Another example



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.

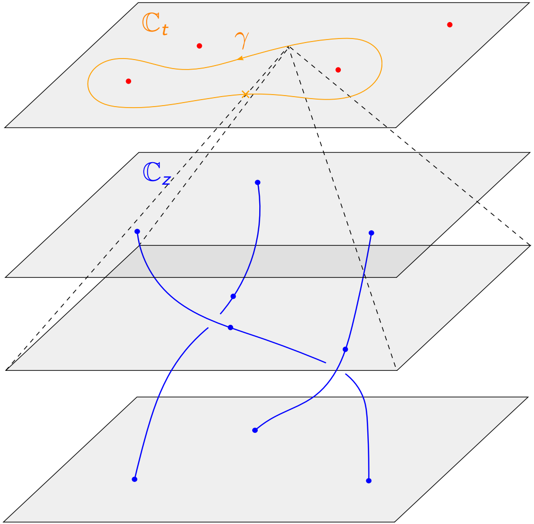
Another example



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.

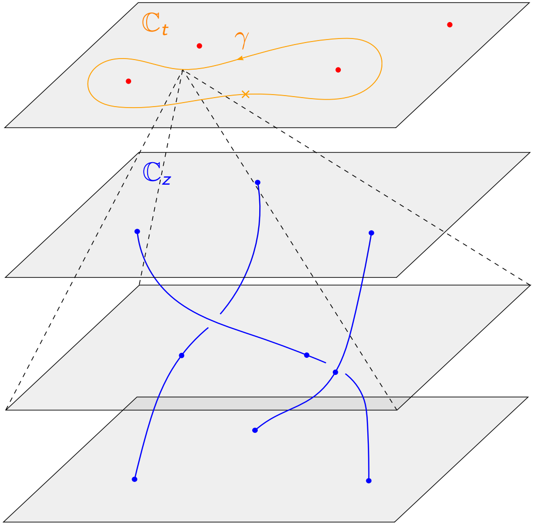
Another example



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.

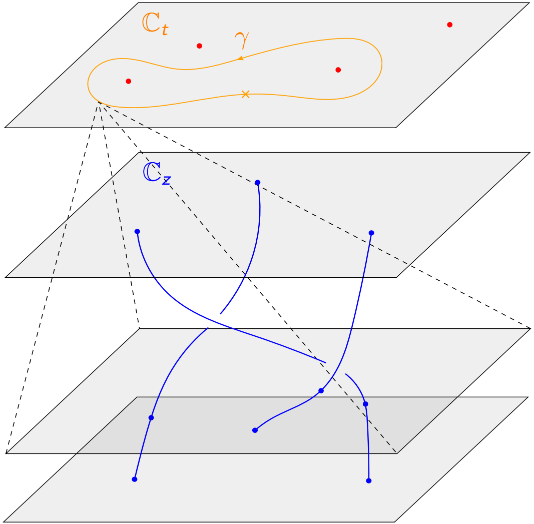
Another example



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.

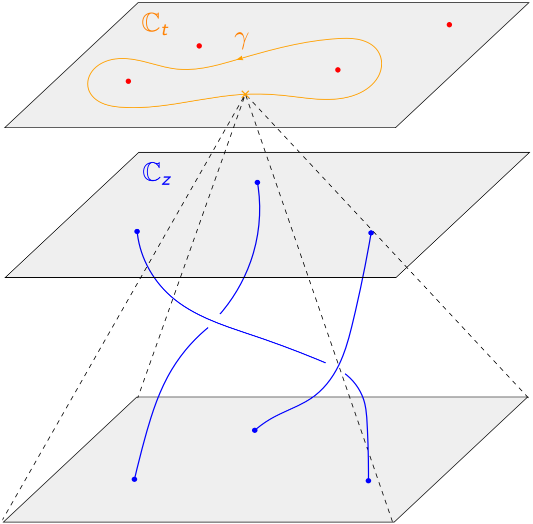
Another example



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.

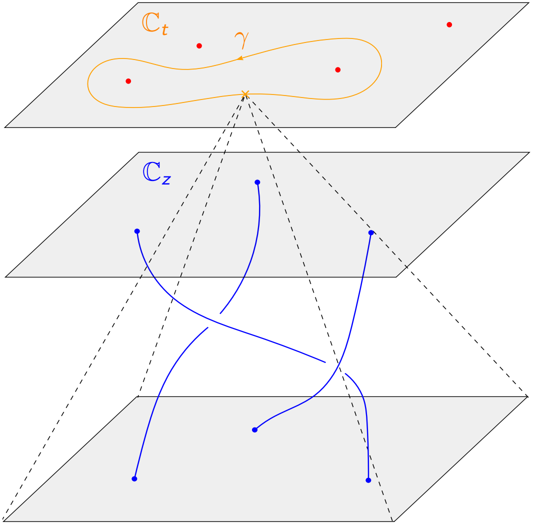
Another example



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.

Another example



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.

Algorithmic goal

Input: g, γ

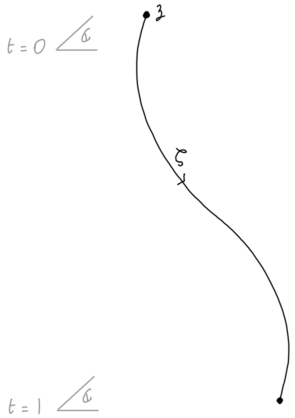
Output: the associated braid in terms of Artin's generators

Main tool

Certified homotopy continuation

Input: $H : [0,1] \times \mathbb{C}^r \to \mathbb{C}^r$ and $z \in \mathbb{C}^r$ such that $H(0,z) = 0$.

There exists $\zeta : [0,1] \to \mathbb{C}^r$ *such that* $H(t, \zeta(t)) = 0$ *and* $\zeta(0) = z$. *Assume it is unique.*



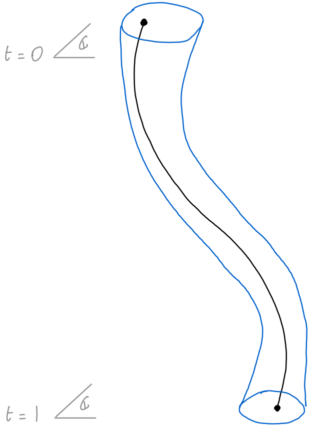
Main tool

Certified homotopy continuation

Input: $H : [0,1] \times \mathbb{C}^r \to \mathbb{C}^r$ and $z \in \mathbb{C}^r$ such that $H(0,z) = 0$.

There exists $\zeta : [0,1] \to \mathbb{C}^r$ *such that* $H(t, \zeta(t)) = 0$ *and* $\zeta(0) = z$. *Assume it is unique.*

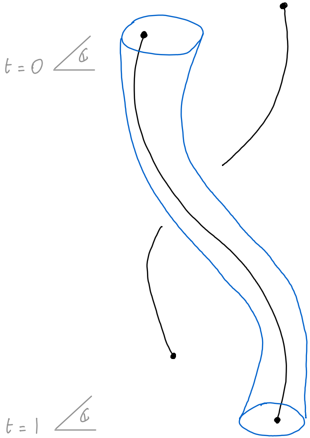
Output: A tubular neighborhood isolating ζ.



Main tool

Certified homotopy continuation

Input: $H : [0,1] \times \mathbb{C}^r \to \mathbb{C}^r$ and $z \in \mathbb{C}^r$ such that $H(0,z) = 0$.
There exists $\zeta : [0,1] \to \mathbb{C}^r$ *such that* $H(t, \zeta(t)) = 0$ *and* $\zeta(0) = z$. *Assume it is unique.*
Output: A tubular neighborhood isolating ζ.



Main tool

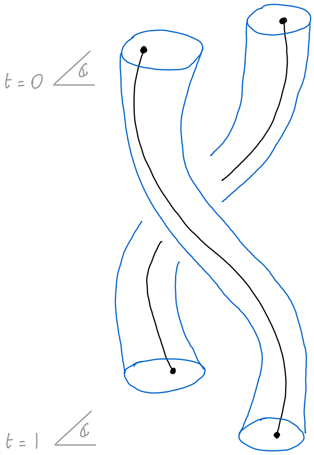
Certified homotopy continuation

Input: $H : [0,1] \times \mathbb{C}^r \to \mathbb{C}^r$ and $z \in \mathbb{C}^r$ such that $H(0,z) = 0$.

There exists $\zeta : [0,1] \to \mathbb{C}^r$ such that $H(t, \zeta(t)) = 0$ and $\zeta(0) = z$. Assume it is unique.

Output: A tubular neighborhood isolating ζ.

We can to that for every solution at $t = 0$



Main tool

Certified homotopy continuation

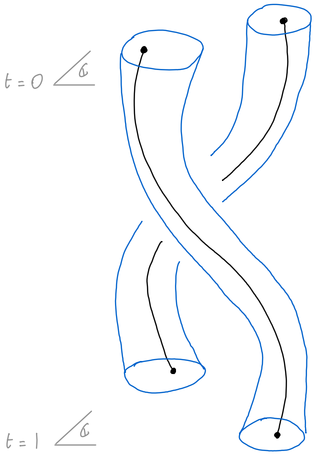
Input: $H : [0,1] \times \mathbb{C}^r \to \mathbb{C}^r$ and $z \in \mathbb{C}^r$ such that $H(0,z) = 0$.
There exists $\zeta : [0,1] \to \mathbb{C}^r$ *such that* $H(t, \zeta(t)) = 0$ *and* $\zeta(0) = z$. *Assume it is unique.*
Output: A tubular neighborhood isolating ζ.

We can to that for every solution at $t = 0$

Application

Take $g \in \mathbb{C}[t,z]$ from last slide and $\gamma : [0,1] \to \mathbb{C} \backslash \Sigma$ ($n = \deg_z(g)$). Apply certified homotopy continuation to $H(t,z) = g(\gamma(t), z)$.

Goal: use Algpath [G. and Lairez, 2024] for this step



Related work

Certified homotopy continuation

- Kearfott, R. B., & Xing, Z. (1994). An Interval Step Control for Continuation Methods.
- van der Hoeven, J. (2015). *Reliable homotopy continuation*.
- Xu, J., Burr, M., & Yap, C. (2018). An Approach for Certifying Homotopy Continuation Paths: Univariate Case.
- Duff, T., & Lee, K. (2024). Certified homotopy tracking using the Krawczyk method.

Braid computations

- Rodriguez, J. I., & Wang, B. (2017). **Numerical computation of braid groups**.
- Marco-Buzunariz, M. Á., & Rodríguez, M. (2016). **SIROCCO: a library for certified polynomial root continuation**.

Braid algorithm

We now assume $\zeta = (\zeta_1, \cdots, \zeta_n) : [0,1] \to OC_n$.

Goal

Input : n disjoint tubular neighborhoods around $\zeta_1, \cdots, \zeta_n$

Output : A decomposition in standard generators of the braid induced by $\zeta_1, \cdots, \zeta_n$

Braid algorithm

We now assume $\zeta = (\zeta_1, \cdots, \zeta_n) : [0,1] \to OC_n$.

Goal

Input : n disjoint tubular neighborhoods around $\zeta_1, \cdots, \zeta_n$

Output : A decomposition in standard generators of the braid induced by $\zeta_1, \cdots, \zeta_n$

Interface

We assume a function $\mathrm{sep}(i, j, t)$ that returns $t' \in (t, 1]$ and a symbol in $\star \in \{\to, \leftarrow, \to, \leftarrow\}$, such that for all $s \in [t, t']$,

- $\mathrm{Re}(\zeta_i(s)) < \mathrm{Re}(\zeta_j(s))$ if $\star = \to$,
- $\mathrm{Re}(\zeta_i(s)) > \mathrm{Re}(\zeta_j(s))$ if $\star = \leftarrow$,
- $\mathrm{Im}(\zeta_i(s)) < \mathrm{Im}(\zeta_j(s))$ if $\star = \to$,
- $\mathrm{Im}(\zeta_i(s)) > \mathrm{Im}(\zeta_j(s))$ if $\star = \leftarrow$,

Easy to implement in practice thanks to interval arithmetic !

Cells

Recall: $OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \forall i \neq j,\, x_i \neq x_j\}$.

Definition

A cell is a pair $c = (R, I)$ of relations on $\{1, \cdots, n\}$. We associate to it a topological space $|c| \subseteq OC_n$ which points are $(x_1, \cdots, x_n) \in OC_n$ such that

- for all $(i,j) \in R$, $\mathrm{Re}(x_i) < \mathrm{Re}(x_j)$,
- for all $(i,j) \in I$, $\mathrm{Im}(x_i) < \mathrm{Im}(x_j)$,

Notation

- $i \color{blue}{\rightarrow} j \iff (i,j) \in R$
- $i \color{red}{\rightarrow} j \iff (i,j) \in I$

Cells

Recall: $OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \forall i \neq j,\ x_i \neq x_j\}$.

Definition

A cell is a pair $c = (R, I)$ of relations on $\{1, \cdots, n\}$.
We associate to it a topological space $|c| \subseteq OC_n$ which points are $(x_1, \cdots, x_n) \in OC_n$ such that

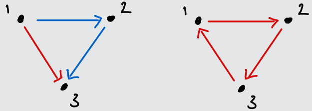
- for all $(i, j) \in R$, $\mathrm{Re}(x_i) < \mathrm{Re}(x_j)$,
- for all $(i, j) \in I$, $\mathrm{Im}(x_i) < \mathrm{Im}(x_j)$,

Notation

- $i \color{blue}{\rightarrow} j \iff (i, j) \in R$
- $i \color{red}{\rightarrow} j \iff (i, j) \in I$

Examples

$c = (\varnothing, \varnothing)$: $|c| = OC_n$,



$(1, 2, 3+i) \in |c|$ $|c| = \phi$

Properties of cells

Empty cells

A cell is empty if and only if there is a cycle in R or in I.

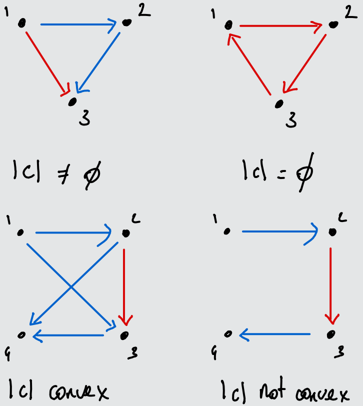
Convex cells

A (non-empty) cell is convex if and only if for all $i, j \in \{1, \cdots, n\}$, either $i \rightarrow^* j$ or $j \rightarrow^* i$ or $i \rightarrow^* j$ or $j \rightarrow^* i$. We call this graph property "monochromatic semi-connectedness" (m.s.c. for short).

Intersection of cells

Given $c = (R, I)$ and $c' = (R', I')$ two cells, the space associated to $(R \cup R', I \cup I')$ is $|c| \cap |c'|$.

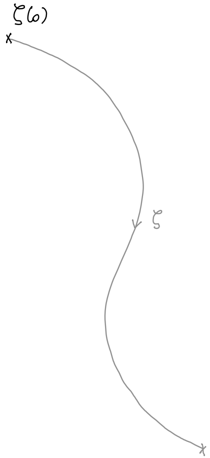
Examples



Step 1: compute a sequence of cells

Path to cells

Input: ζ (represented by tubular neighborhoods)

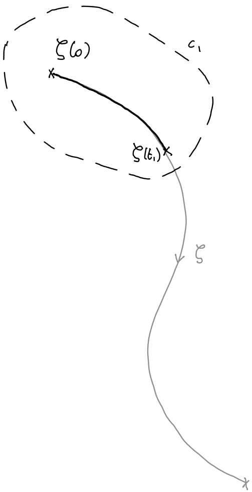


Step 1: compute a sequence of cells

Path to cells

Input: ζ (represented by tubular neighborhoods)
Output: a sequence of m.s.c. cells $c_1, \cdots, c_r$ such that there exists $0 = t_0 < \cdots < t_r = 1$ and for any $s \in [t_{i-1}, t_i]$, $\zeta(s) \in c_i$
Idea:

- Start with an initial m.s.c cell c containing $\zeta(0)$.
- Associate to each edge a time of validity.
- When a relation expires, update it using sep and repair m.s.c.
- Repeat



Step 1: compute a sequence of cells

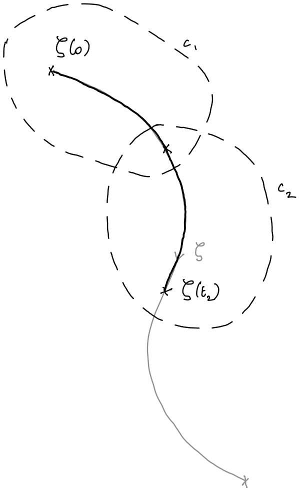
Path to cells

Input: ζ (represented by tubular neighborhoods)

Output: a sequence of m.s.c. cells $c_1, \cdots, c_r$ such that there exists $0 = t_0 < \cdots < t_r = 1$ and for any $s \in [t_{i-1}, t_i]$, $\zeta(s) \in c_i$

Idea:

- Start with an initial m.s.c cell c containing $\zeta(0)$.
- Associate to each edge a time of validity.
- When a relation expires, update it using sep and repair m.s.c.
- Repeat



Step 1: compute a sequence of cells

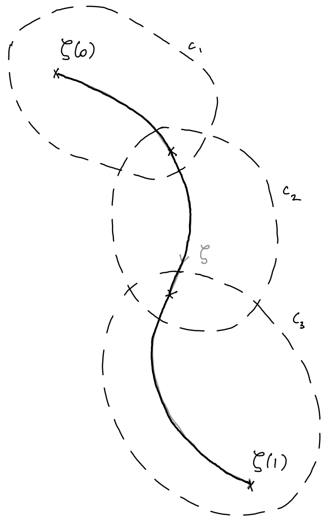
Path to cells

Input: ζ (represented by tubular neighborhoods)

Output: a sequence of m.s.c. cells $c_1, \cdots, c_r$ such that there exists $0 = t_0 < \cdots < t_r = 1$ and for any $s \in [t_{i-1}, t_i]$, $\zeta(s) \in c_i$

Idea:

- Start with an initial m.s.c cell c containing $\zeta(0)$.
- Associate to each edge a time of validity.
- When a relation expires, update it using sep and repair m.s.c.
- Repeat

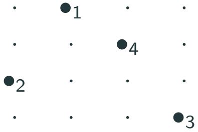


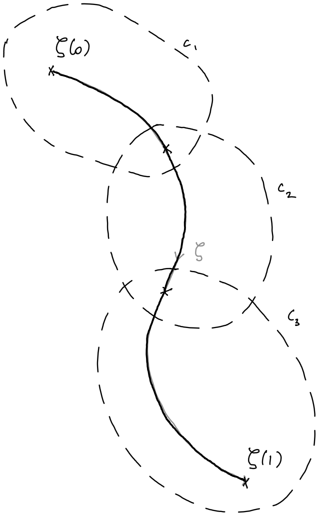
Step 2: linearize ζ

Definition

Let $\rho, \iota \in \mathfrak{S}_n$. We define
$\omega_{\rho,\iota} = (\rho(1) + \mathbf{i}\iota(1), \cdots, \rho(n) + \mathbf{i}\iota(n)) \in OC_n$

$$\omega_{(1\,2)(3\,4),(1\,4\,3)} =$$



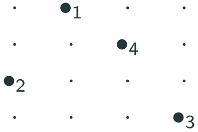


Step 2: linearize ζ

Definition

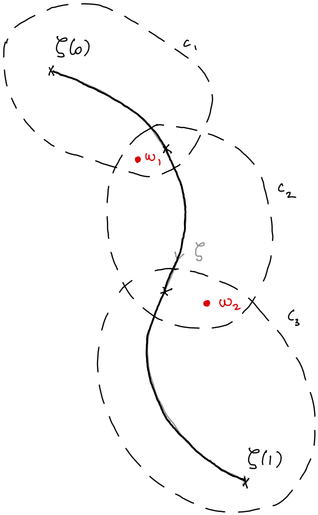
Let $\rho, \iota \in \mathfrak{S}_n$. We define
$\omega_{\rho,\iota} = (\rho(1) + \mathbf{i}\iota(1), \cdots, \rho(n) + \mathbf{i}\iota(n)) \in OC_n$

$\omega_{(1\,2)(3\,4),(1\,4\,3)} =$



Linearization of ζ

For each c_i, c_{i+1}, we compute ρ, ι such that $\omega_i = \omega_{\rho,\iota}$ lies in the intersection $c_i \cap c_{i+1}$ (Hint: total order extending R and I).

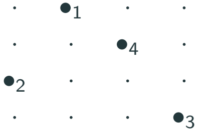


Step 2: linearize ζ

Definition

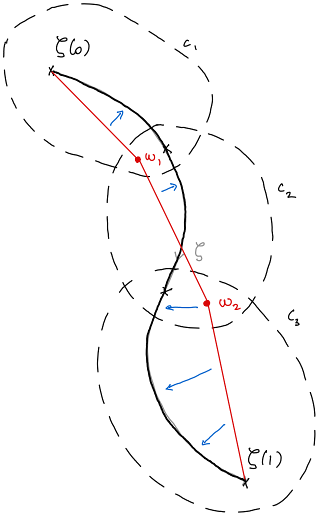
Let $\rho, \iota \in \mathfrak{S}_n$. We define
$\omega_{\rho,\iota} = (\rho(1) + \mathbf{i}\iota(1), \cdots, \rho(n) + \mathbf{i}\iota(n)) \in OC_n$

$\omega_{(1\,2)(3\,4),(1\,4\,3)} =$



Linearization of ζ

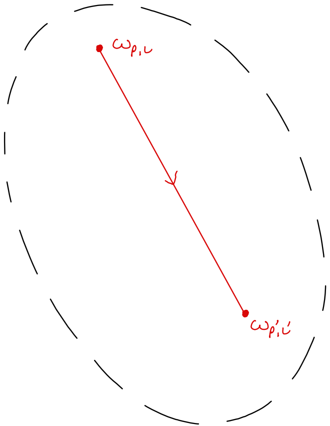
For each c_i, c_{i+1}, we compute ρ, ι such that $\omega_i = \omega_{\rho,\iota}$ lies in the intersection $c_i \cap c_{i+1}$ (Hint: total order extending R and I). The linear interpolation of the ω_i is homotopic to ζ. Why ? m.s.c cells are convex !



Step 3: decomposition of the linearization

Reduction

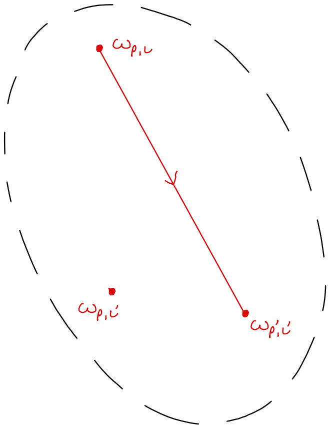
- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent



Step 3: decomposition of the linearization

Reduction

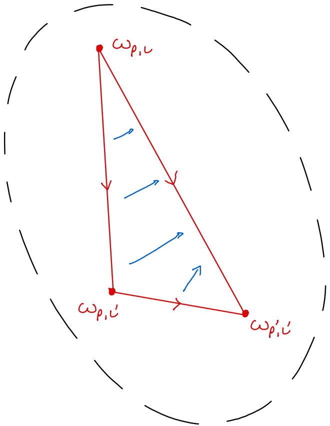
- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent
- Assume $\omega_{\rho,\iota}$ and $\omega_{\rho',\iota'}$ both lie in a m.s.c cell $c = (R, I)$. It means that ρ, ρ' extend R and ι, ι' extend I. **So $\omega_{\rho,\iota'}$ also lies in c!**



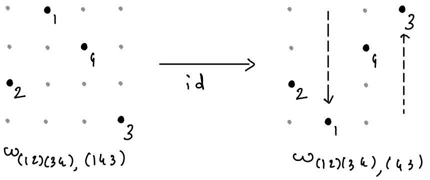
Step 3: decomposition of the linearization

Reduction

- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent
- Assume $\omega_{\rho,\iota}$ and $\omega_{\rho',\iota'}$ both lie in a m.s.c cell $c = (R, I)$. It means that ρ, ρ' extend R and ι, ι' extend I. **So $\omega_{\rho,\iota'}$ also lies in c!**
- We compute the braid of $\omega_{\rho,\iota} \to \omega_{\rho,\iota'}$ then the braid of $\omega_{\rho,\iota'} \to \omega_{\rho',\iota'}$



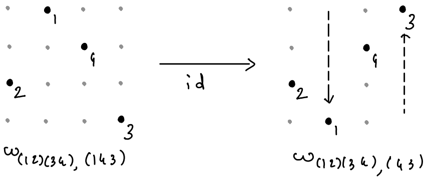
Step 3: decomposition of the linearization

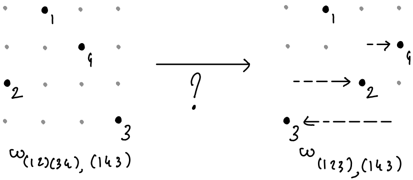


$\omega_{\rho,\iota} \to \omega_{\rho,\iota'}$

The induced braid is trivial, as the real part of the strands is constant.

Step 3: decomposition of the linearization

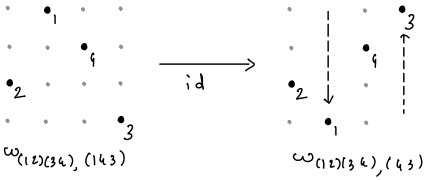




$\omega_{\rho,\iota} \to \omega_{\rho,\iota'}$

The induced braid is trivial, as the real part of the strands is constant.

Step 3: decomposition of the linearization

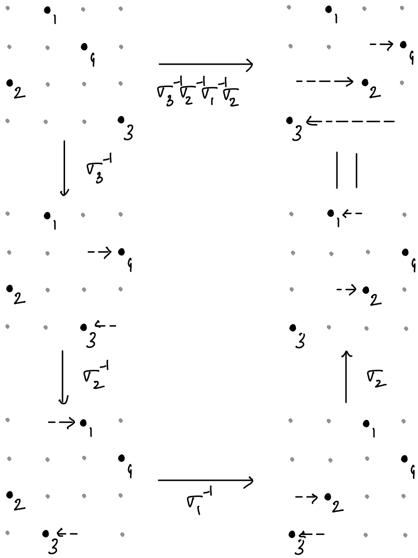


$\omega_{\rho,\iota} \to \omega_{\rho,\iota'}$

The induced braid is trivial, as the real part of the strands is constant.

$\omega_{\rho,\iota'} \to \omega_{\rho',\iota'}$

Let $\rho'\rho^{-1} = s_{i_1} \cdots s_{i_r}$ be a decomposition in elementary transpositions. Output $\sigma_{i_1}^{\varepsilon_1} \cdots \sigma_{i_r}^{\varepsilon_r}$ with $\varepsilon_1, \cdots, \varepsilon_r \in \{\pm 1\}$ computed using ι'.



Conclusion

```
~/2025/code/braid_group cargo run --release
    Finished `release` profile [optimized] target(s) in 0.08s
     Running `target/release/braid_group`
```

$$\sigma_{57}\sigma_{11}\sigma_{25}\sigma_{29}\sigma_{47}\sigma_{51}\sigma_{55}\sigma_{61}\sigma_{63}\sigma_{83}\sigma_{3}\sigma_{5}\sigma_{41}\sigma_{31}\sigma_{35}\sigma_{0}^{-1}\sigma_{1}\sigma_{0}\sigma_{37}\sigma_{73}\sigma_{97}\sigma_{98}\sigma_{27}\sigma_{49}\sigma_{87}\sigma_{97}^{-1}\sigma_{59}\sigma_{15}\sigma_{2}\sigma_{3}^{-1}\sigma_{96}^{-1}\sigma_{77}\sigma_{92}^{-1}\sigma_{93}^{-1}\sigma_{81}\sigma_{92}\sigma_{24}\sigma_{72}^{-1}\sigma_{84}\sigma_{18}^{-1}\sigma_{26}\sigma_{94}^{-1}\sigma_{95}^{-1}\sigma_{10}\sigma_{88}\sigma_{87}^{-1}\sigma_{17}^{-1}\sigma_{24}^{-1}\sigma_{16}^{-1}\sigma_{25}\sigma_{24}\sigma_{82}\sigma_{15}^{-1}\sigma_{63}^{-1}\sigma_{86}^{-1}\sigma_{13}\sigma_{14}^{-1}\sigma_{13}^{-1}\sigma_{6}^{-1}\sigma_{81}^{-1}\sigma_{65}\sigma_{87}^{-1}\sigma_{12}^{-1}\sigma_{73}^{-1}\sigma_{11}^{-1}\sigma_{88}^{-1}\sigma_{96}^{-1}\sigma_{93}\sigma_{94}\sigma_{4}\sigma_{5}^{-1}\sigma_{27}^{-1}\sigma_{10}^{-1}\sigma_{66}\sigma_{62}\sigma_{71}\sigma_{74}^{-1}\sigma_{6}^{-1}\sigma_{7}\sigma_{6}\sigma_{4}^{-1}\sigma_{15}\sigma_{9}^{-1}\sigma_{93}\sigma_{23}\sigma_{16}^{-1}\sigma_{92}^{-1}\sigma_{8}^{-1}\sigma_{20}^{-1}\sigma_{7}^{-1}\sigma_{75}^{-1}\sigma_{6}^{-1}\sigma_{5}^{-1}\sigma_{4}^{-1}\sigma_{91}\sigma_{22}\sigma_{21}\sigma_{89}^{-1}\sigma_{20}\sigma_{73}\sigma_{85}\sigma_{90}^{-1}\sigma_{22}\sigma_{91}^{-1}\sigma_{3}^{-1}\sigma_{92}^{-1}\sigma_{6}\sigma_{93}^{-1}\sigma_{82}^{-1}\sigma_{83}^{-1}\sigma_{2}^{-1}\sigma_{82}\sigma_{94}^{-1}\sigma_{95}^{-1}\sigma_{12}^{-1}\sigma_{67}\sigma_{0}^{-1}\sigma_{1}^{-1}\sigma_{0}^{-1}\sigma_{96}^{-1}\sigma_{35}^{-1}\sigma_{68}\sigma_{76}^{-1}\sigma_{77}^{-1}\sigma_{97}^{-1}\sigma_{14}\sigma_{15}^{-1}\sigma_{14}^{-1}\sigma_{98}^{-1}\sigma_{60}\sigma_{94}\sigma_{92}^{-1}\sigma_{91}\sigma_{78}^{-1}\sigma_{13}^{-1}\sigma_{14}\sigma_{70}^{-1}\sigma_{69}\sigma_{70}\sigma_{59}^{-1}\sigma_{21}\sigma_{83}\sigma_{79}^{-1}\sigma_{80}^{-1}\sigma_{92}\sigma_{71}\sigma_{15}\sigma_{23}^{-1}\sigma_{17}^{-1}\sigma_{9}^{-1}\sigma_{18}\sigma_{10}\sigma_{19}\sigma_{81}^{-1}\sigma_{18}\sigma_{93}\sigma_{92}^{-1}\sigma_{17}\sigma_{84}\sigma_{83}^{-1}\sigma_{82}^{-1}\sigma_{83}^{-1}\sigma_{84}^{-1}\sigma_{16}\sigma_{72}\sigma_{79}^{-1}\sigma_{12}^{-1}\sigma_{76}^{-1}\sigma_{13}\sigma_{85}^{-1}\sigma_{73}\sigma_{86}^{-1}\sigma_{36}\sigma_{74}\sigma_{81}\sigma_{87}^{-1}\sigma_{88}^{-1}\sigma_{15}\sigma_{89}^{-1}\sigma_{14}\sigma_{87}\sigma_{13}\sigma_{12}\sigma_{17}^{-1}\sigma_{18}\sigma_{28}\sigma_{90}^{-1}\sigma_{91}^{-1}\sigma_{78}\sigma_{92}^{-1}\sigma_{93}^{-1}\sigma_{94}^{-1}\sigma_{95}^{-1}\sigma_{98}\sigma_{97}\sigma_{92}^{-1}\sigma_{96}^{-1}\sigma_{97}^{-1}\sigma_{24}\sigma_{98}^{-1}\sigma_{94}\sigma_{75}\sigma_{29}^{-1}\sigma_{15}\sigma_{76}\sigma_{88}\sigma_{87}^{-1}\sigma_{33}\sigma_{95}\sigma_{94}\sigma_{93}\sigma_{92}\sigma_{11}\sigma_{77}\sigma_{78}\sigma_{68}\sigma_{67}^{-1}\sigma_{91}\sigma_{90}\sigma_{79}\sigma_{80}\sigma_{10}\sigma_{9}\sigma_{19}^{-1}\sigma_{79}^{-1}\sigma_{84}\sigma_{7}^{-1}\sigma_{8}\sigma_{7}\sigma_{30}\sigma_{78}\sigma_{6}\sigma_{8}\sigma_{5}\sigma_{91}\sigma_{31}^{-1}\sigma_{89}\sigma_{4}\sigma_{81}\sigma_{82}\sigma_{47}^{-1}\sigma_{3}\sigma_{88}\sigma_{20}\sigma_{21}^{-1}\sigma_{83}\sigma_{81}\sigma_{84}\sigma_{85}\sigma_{76}^{-1}\sigma_{32}\sigma_{33}^{-1}\sigma_{86}\sigma_{2}\sigma_{1}\sigma_{94}\sigma_{87}\sigma_{77}^{-1}\sigma_{75}\sigma_{76}\sigma_{88}\sigma_{89}\sigma_{75}^{-1}\sigma_{90}\sigma_{91}\sigma_{66}\sigma_{65}^{-1}\sigma_{22}^{-1}\sigma_{85}^{-1}\sigma_{84}\sigma_{92}\sigma_{93}\sigma_{94}\sigma_{95}\sigma_{23}\sigma_{96}\sigma_{97}\sigma_{24}\sigma_{82}^{-1}\sigma_{83}\sigma_{82}\sigma_{19}\sigma_{20}^{-1}\sigma_{34}^{-1}\sigma_{35}\sigma_{36}\sigma_{81}^{-1}\sigma_{74}^{-1}\sigma_{73}\sigma_{72}\sigma_{25}\sigma_{63}\sigma_{64}\sigma_{63}^{-1}\sigma_{73}^{-1}\sigma_{74}\sigma_{26}^{-1}\sigma_{37}^{-1}\sigma_{38}^{-1}\sigma_{39}\sigma_{62}\sigma_{61}\sigma_{60}^{-1}\sigma_{59}^{-1}\sigma_{60}\sigma_{61}^{-1}\sigma_{23}^{-1}\sigma_{12}^{-1}\sigma_{13}\sigma_{58}\sigma_{57}^{-1}\sigma_{40}\sigma_{41}^{-1}\sigma_{22}\sigma_{14}\sigma_{15}^{-1}\sigma_{42}^{-1}\sigma_{43}\sigma_{56}\sigma_{55}^{-1}\sigma_{54}^{-1}\sigma_{53}\sigma_{44}^{-1}\sigma_{72}^{-1}\sigma_{73}\sigma_{45}\sigma_{26}\sigma_{25}^{-1}\sigma_{84}^{-1}\sigma_{74}^{-1}\sigma_{75}\sigma_{86}\sigma_{76}^{-1}\sigma_{77}\sigma_{52}\sigma_{51}^{-1}\sigma_{38}\sigma_{85}\sigma_{50}\sigma_{49}^{-1}\sigma_{24}^{-1}\sigma_{23}\sigma_{46}^{-1}\sigma_{47}\sigma_{78}^{-1}\sigma_{79}\sigma_{98}\sigma_{0}\sigma_{84}\sigma_{48}\sigma_{78}\sigma_{82}^{-1}\sigma_{80}\sigma_{30}^{-1}\sigma_{34}\sigma_{52}^{-1}\sigma_{74}\sigma_{76}\sigma_{90}^{-1}\sigma_{23}^{-1}\sigma_{24}\sigma_{36}^{-1}\sigma_{79}\sigma_{78}^{-1}\sigma_{4}^{-1}\sigma_{21}\sigma_{46}\sigma_{47}\sigma_{70}^{-1}\sigma_{46}^{-1}\sigma_{28}^{-1}\sigma_{68}^{-1}\sigma_{86}\sigma_{92}\sigma_{6}\sigma_{23}\sigma_{54}\sigma_{66}^{-1}\sigma_{88}^{-1}\sigma_{94}^{-1}\sigma_{2}\sigma_{32}^{-1}\sigma_{49}\sigma_{50}^{-1}\sigma_{38}^{-1}\sigma_{10}\sigma_{26}^{-1}\sigma_{56}^{-1}\sigma_{58}^{-1}\sigma_{72}\sigma_{40}^{-1}\sigma_{44}\sigma_{51}^{-1}\sigma_{85}\sigma_{60}^{-1}\sigma_{52}\sigma_{12}\sigma_{96}^{-1}\sigma_{45}\sigma_{44}^{-1}\sigma_{24}\sigma_{23}^{-1}\sigma_{17}\sigma_{53}\sigma_{54}^{-1}\sigma_{23}\sigma_{24}^{-1}\sigma_{77}\sigma_{76}^{-1}\sigma_{86}\sigma_{75}\sigma_{84}^{-1}\sigma_{42}\sigma_{43}\sigma_{42}^{-1}\sigma_{55}^{-1}\sigma_{56}\sigma_{62}^{-1}\sigma_{74}^{-1}\sigma_{25}^{-1}\sigma_{26}\sigma_{73}\sigma_{72}^{-1}\sigma_{57}^{-1}\sigma_{58}\sigma_{38}\sigma_{60}\sigma_{61}\sigma_{60}^{-1}\sigma_{59}^{-1}\sigma_{22}\sigma_{49}^{-1}\sigma_{41}^{-1}\sigma_{14}^{-1}\sigma_{15}^{-1}\sigma_{14}\sigma_{23}^{-1}\sigma_{60}\sigma_{61}^{-1}\sigma_{40}\sigma_{13}\sigma_{12}^{-1}\sigma_{85}^{-1}\sigma_{62}\sigma_{39}\sigma_{38}^{-1}\sigma_{74}\sigma_{73}^{-1}\sigma_{37}^{-1}\sigma_{60}^{-1}\sigma_{63}\sigma_{64}^{-1}\sigma_{36}\sigma_{35}\sigma_{34}^{-1}\sigma_{81}^{-1}\sigma_{65}^{-1}\sigma_{66}\sigma_{72}\sigma_{26}^{-1}\sigma_{25}\sigma_{82}\sigma_{83}\sigma_{24}\sigma_{25}^{-1}\sigma_{26}\sigma_{82}^{-1}\sigma_{84}^{-1}\sigma_{73}\sigma_{74}^{-1}\sigma_{32}\sigma_{33}\sigma_{32}^{-1}\sigma_{33}^{-1}\sigma_{23}\sigma_{22}^{-1}\sigma_{19}^{-1}\sigma_{20}^{-1}\sigma_{19}\sigma_{97}\sigma_{96}\sigma_{31}^{-1}\sigma_{95}\sigma_{94}\sigma_{93}\sigma_{92}\sigma_{85}\sigma_{30}\sigma_{67}^{-1}\sigma_{68}\sigma_{8}^{-1}\sigma_{21}^{-1}\sigma_{20}\sigma_{91}\sigma_{90}\sigma_{75}^{-1}\sigma_{76}\sigma_{75}\sigma_{87}\sigma_{77}^{-1}\sigma_{88}\sigma_{76}^{-1}\sigma_{89}\sigma_{88}\sigma_{87}\sigma_{86}\sigma_{85}\sigma_{81}\sigma_{47}^{-1}\sigma_{98}^{-1}\sigma_{97}^{-1}\sigma_{29}^{-1}\sigma_{28}\sigma_{96}^{-1}\sigma_{84}\sigma_{94}\sigma_{95}^{-1}\sigma_{94}^{-1}\sigma_{93}^{-1}\sigma_{83}\sigma_{82}\sigma_{81}\sigma_{91}\sigma_{92}^{-1}\sigma_{91}^{-1}\sigma_{90}^{-1}\sigma_{78}\sigma_{88}\sigma_{89}^{-1}\sigma_{88}^{-1}\sigma_{79}^{-1}\sigma_{87}^{-1}\sigma_{82}^{-1}\sigma_{86}^{-1}\sigma_{1}\sigma_{2}\sigma_{59}^{-1}\sigma_{60}\sigma_{77}\sigma_{18}\sigma_{17}^{-1}\sigma_{85}^{-1}\sigma_{84}^{-1}\sigma_{83}^{-1}\sigma_{82}^{-1}\sigma_{78}^{-1}\sigma_{3}\sigma_{4}\sigma_{88}\sigma_{80}\sigma_{89}^{-1}\sigma_{18}^{-1}\sigma_{19}^{-1}\sigma_{18}\sigma_{19}\sigma_{69}^{-1}\sigma_{79}\sigma_{84}\sigma_{16}^{-1}\sigma_{78}\sigma_{96}\sigma_{77}\sigma_{70}\sigma_{81}^{-1}\sigma_{5}\sigma_{6}\sigma_{80}^{-1}\sigma_{7}\sigma_{8}\sigma_{0}^{-1}\sigma_{1}^{-1}\sigma_{94}^{-1}\sigma_{79}^{-1}\sigma_{75}^{-1}\sigma_{76}\sigma_{75}\sigma_{78}^{-1}\sigma_{17}^{-1}\sigma_{15}\sigma_{2}^{-1}\sigma_{3}^{-1}\sigma_{11}\sigma_{10}^{-1}\sigma_{9}\sigma_{10}\sigma_{77}^{-1}\sigma_{90}^{-1}\sigma_{76}^{-1}\sigma_{11}\sigma_{16}\sigma_{4}^{-1}\sigma_{5}^{-1}\sigma_{22}^{-1}\sigma_{6}^{-1}\sigma_{7}^{-1}\sigma_{74}\sigma_{73}\sigma_{14}\sigma_{13}^{-1}\sigma_{14}^{-1}\sigma_{15}^{-1}\sigma_{14}\sigma_{8}^{-1}\sigma_{75}^{-1}\sigma_{74}^{-1}\sigma_{9}^{-1}\sigma_{10}^{-1}\sigma_{23}^{-1}\sigma_{72}\sigma_{21}\sigma_{26}^{-1}\sigma_{12}\sigma_{13}\sigma_{71}\sigma_{22}\sigma_{73}^{-1}\sigma_{67}\sigma_{70}\sigma_{69}\sigma_{70}\sigma_{25}\sigma_{89}^{-1}\sigma_{72}^{-1}\sigma_{14}\sigma_{15}\sigma_{93}^{-1}\sigma_{92}^{-1}\sigma_{94}\sigma_{93}\sigma_{92}\sigma_{71}^{-1}\sigma_{16}\sigma_{70}^{-1}\sigma_{17}\sigma_{88}^{-1}\sigma_{95}\sigma_{96}^{-1}\sigma_{11}^{-1}\sigma_{12}^{-1}\sigma_{15}^{-1}\sigma_{27}^{-1}\sigma_{13}^{-1}\sigma_{14}^{-1}\sigma_{15}^{-1}\sigma_{16}^{-1}\sigma_{94}\sigma_{95}^{-1}\sigma_{18}\sigma_{19}\sigma_{86}\sigma_{15}^{-1}\sigma_{26}\sigma_{68}\sigma_{67}\sigma_{11}^{-1}\sigma_{85}\sigma_{69}^{-1}\sigma_{13}\sigma_{14}\sigma_{68}^{-1}\sigma_{87}^{-1}\sigma_{33}\sigma_{17}^{-1}\sigma_{86}^{-1}\sigma_{18}^{-1}\sigma_{82}^{-1}\sigma_{83}\sigma_{4}^{-1}\sigma_{5}^{-1}\sigma_{6}^{-1}\sigma_{94}^{-1}\sigma_{81}^{-1}\sigma_{82}\sigma_{4}\sigma_{5}\sigma_{6}\sigma_{7}\sigma_{91}\sigma_{93}^{-1}\sigma_{10}\sigma_{92}^{-1}\sigma_{88}\sigma_{21}^{-1}\sigma_{89}^{-1}\sigma_{23}\sigma_{20}\sigma_{21}\sigma_{22}\sigma_{21}\sigma_{20}\sigma_{87}\sigma_{88}^{-1}\sigma_{23}\sigma_{19}^{-1}\sigma_{6}^{-1}\sigma_{20}^{-1}\sigma_{21}^{-1}\sigma_{22}^{-1}\sigma_{5}^{-1}\sigma_{4}\sigma_{93}^{-1}\sigma_{20}\sigma_{19}^{-1}\sigma_{20}^{-1}\sigma_{94}\sigma_{65}\sigma_{13}\sigma_{95}\sigma_{96}^{-1}\sigma_{2}\sigma_{3}\sigma_{2}^{-1}\sigma_{1}\sigma_{0}^{-1}\sigma_{1}^{-1}\sigma_{3}^{-1}\sigma_{7}^{-1}\sigma_{9}^{-1}\sigma_{13}^{-1}\sigma_{17}^{-1}\sigma_{19}^{-1}\sigma_{21}^{-1}\sigma_{23}^{-1}\sigma_{33}^{-1}\sigma_{35}^{-1}\sigma_{39}^{-1}\sigma_{43}^{-1}\sigma_{45}^{-1}\sigma_{53}^{-1}\sigma_{63}^{-1}\sigma_{65}^{-1}\sigma_{67}^{-1}\sigma_{69}^{-1}\sigma_{71}^{-1}\sigma_{75}^{-1}\sigma_{79}^{-1}\sigma_{85}^{-1}\sigma_{87}^{-1}\sigma_{89}^{-1}\sigma_{91}^{-1}\sigma_{95}^{-1}\sigma_{97}^{-1}\sigma_{98}$$