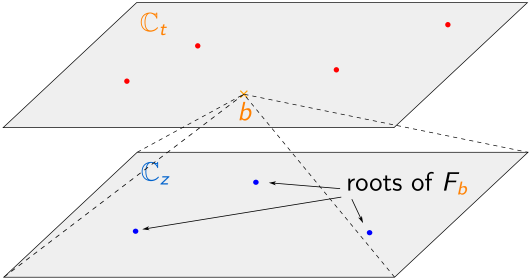
Braid monodromy computations using certified path tracking

Alexandre Guillemot
Joint work with Pierre Lairez
MATHEXP, Inria, France

Journées de géométrie algorithmique
October 14, 2025 | Roscoff

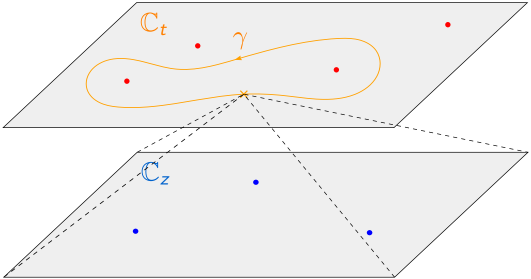
Motivation



Setup

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

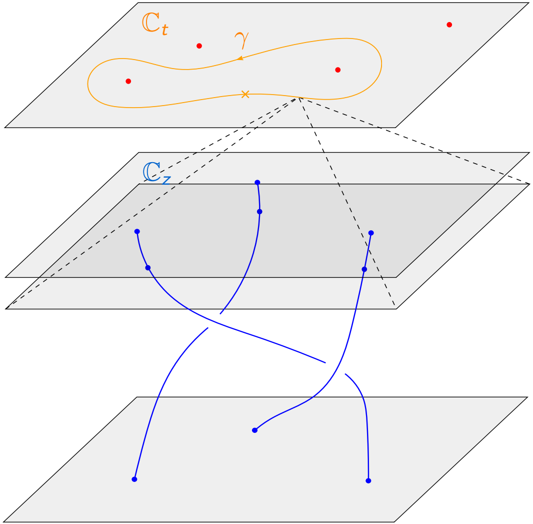
Motivation



Setup

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

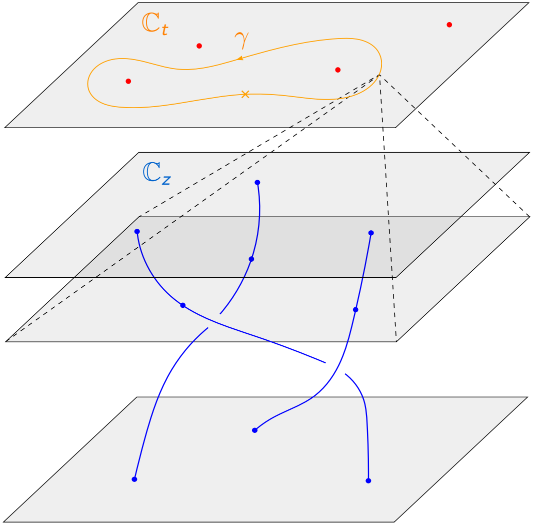
Motivation



Setup

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

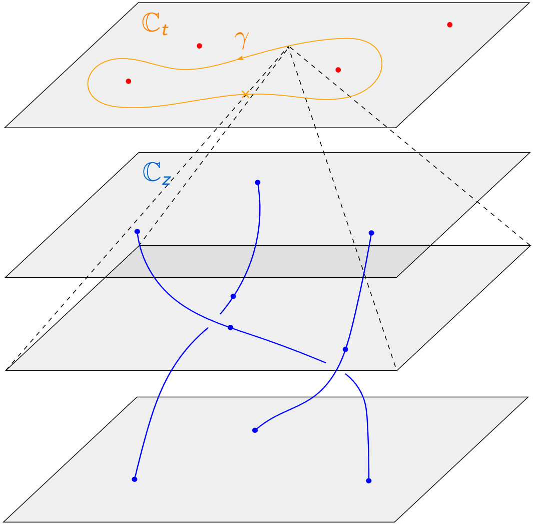
Motivation



Setup

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

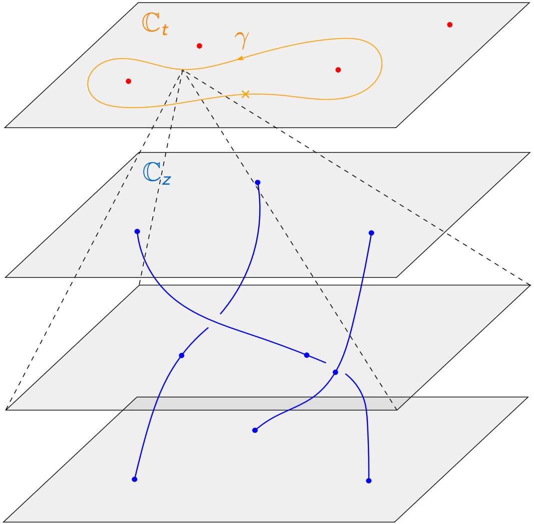
Motivation



Setup

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

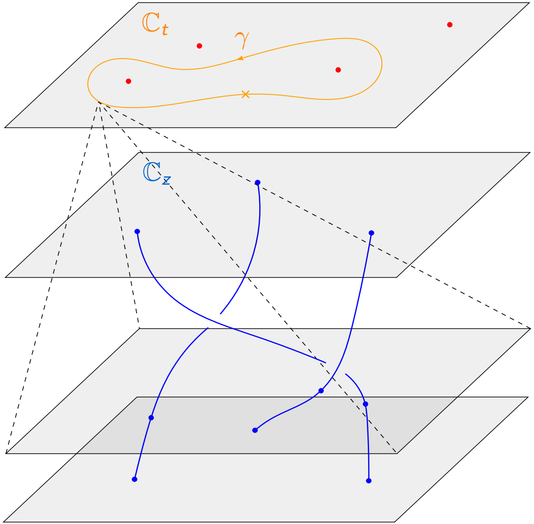
Motivation



Setup

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

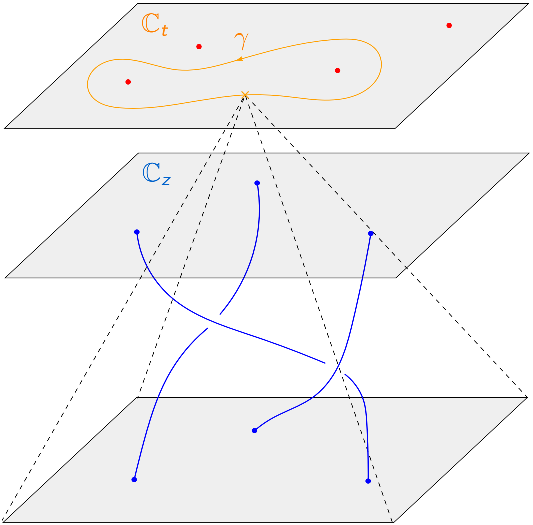
Motivation



Setup

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

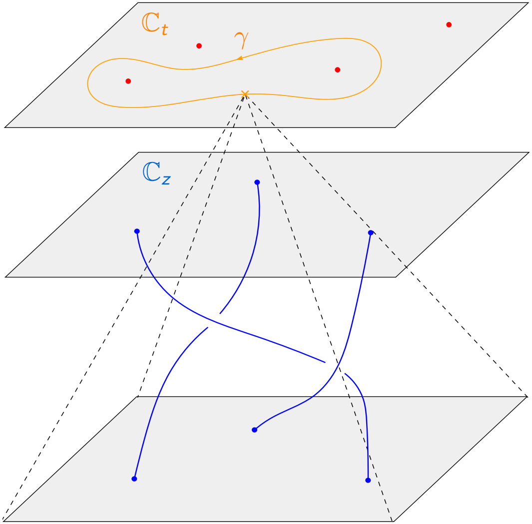
Motivation



Setup

- Let $g \in \mathbb{C}[t, z]$ $(n = \deg_z(g))$,
- 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



Setup

- Let $g \in \mathbb{C}[t, z]$ $(n = \deg_z(g))$,
- 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

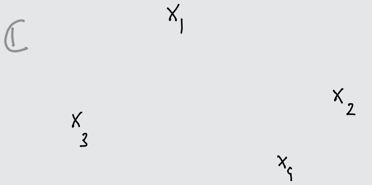
Input: g, γ

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

Configurations

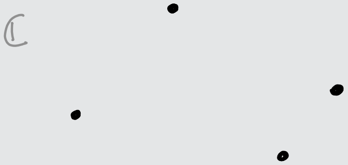
Ordered configurations

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



Configurations

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

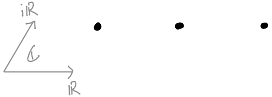


"Forget order" projection

$$\begin{array}{rccc} \Phi : & OC_n & \to & C_n \\ & (x_1, \dots, x_n) & \mapsto & \{x_1, \dots, x_n\}. \end{array}$$

Braid

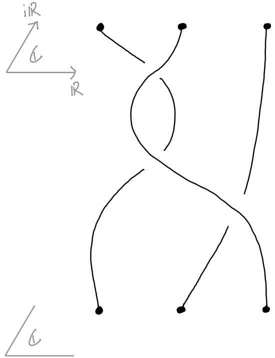
Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \ldots, n\}$.



Braid group

Braid

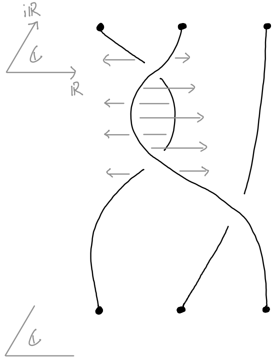
Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \ldots, n\}$.



Braid group

Braid

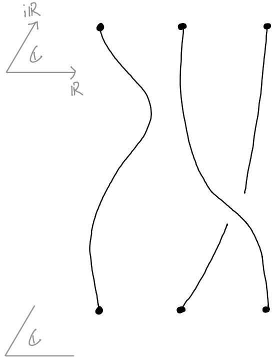
Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \dots, n\}$.



Braid group

Braid

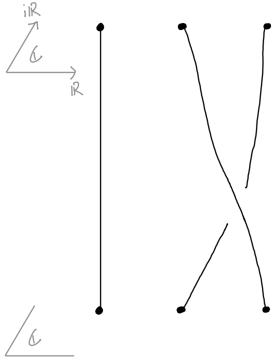
Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \dots, n\}$.



Braid group

Braid

Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \ldots, n\}$.

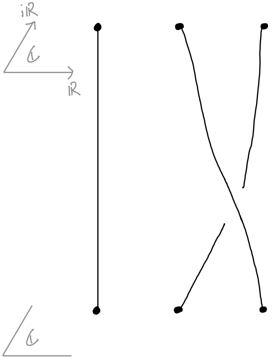


Braid group

Braid

Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \ldots, n\}$.

In practice, we will manipulate paths in OC_n.



Braid group

Braid

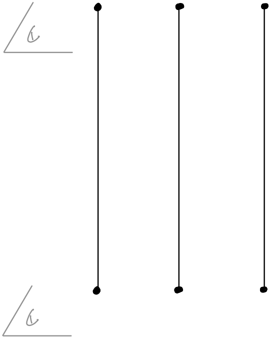
Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \dots, n\}$.

In practice, we will manipulate paths in OC_n.

Braid group B_n

id: class of the constant path equal to $\{1, \dots, n\}$.
Law: $[\beta_1][\beta_2] := [\beta_1 \cdot \beta_2]$.

Rk: this is $\pi_1(C_n, \{1, \dots, n\})$.



id_{B_3}

Braid

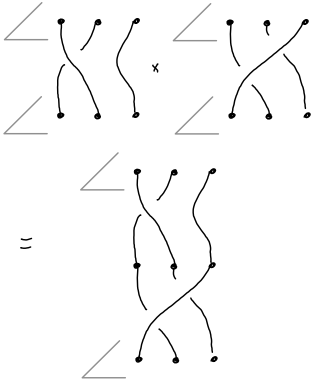
Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \ldots, n\}$.

In practice, we will manipulate paths in OC_n.

Braid group B_n

id: class of the constant path equal to $\{1, \ldots, n\}$.
Law: $[\beta_1][\beta_2] := [\beta_1 \cdot \beta_2]$.

Rk: this is $\pi_1(C_n, \{1, \ldots, n\})$.



Braid group

Braid

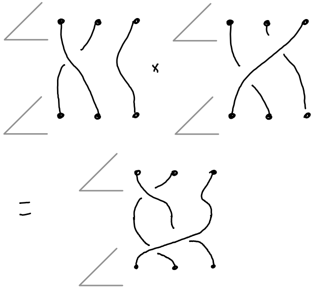
Homotopy class of a path $\beta : [0,1] \to C_n$ such that $\beta(0) = \beta(1) = \{1, \ldots, n\}$.

In practice, we will manipulate paths in OC_n.

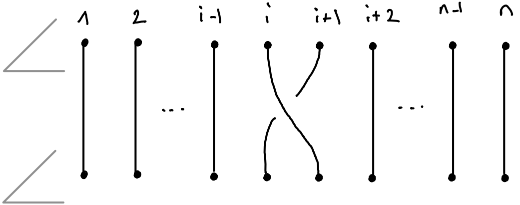
Braid group B_n

id: class of the constant path equal to $\{1, \ldots, n\}$.
Law: $[\beta_1][\beta_2] := [\beta_1 \cdot \beta_2]$.

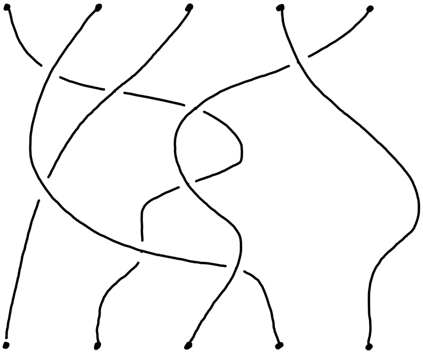
Rk: this is $\pi_1(C_n, \{1, \ldots, n\})$.



Artin's theorem



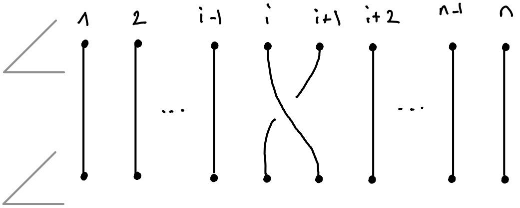
$\sigma_i \in B_n$



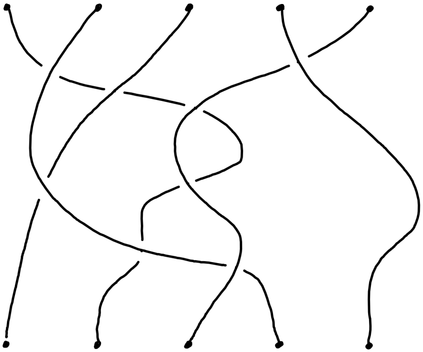
Theorem [Artin, 1947]

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

Artin's theorem



$\sigma_i \in B_n$



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

Theorem [Artin, 1947]

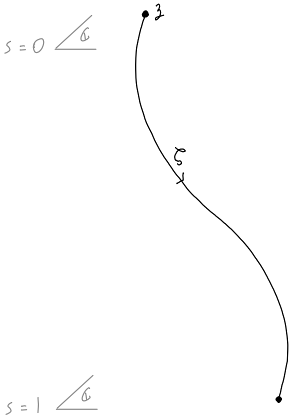
The σ_i's generate B_n (+ explicit relations).

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* $H(s, \zeta(s)) = 0$ *and* $\zeta(0) = z$. *Assume it is unique.*



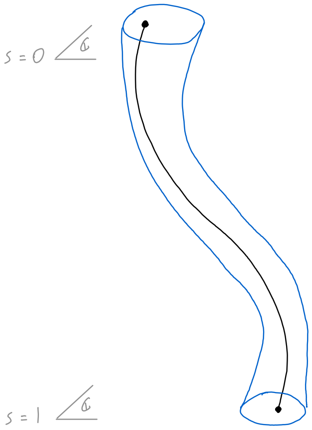
Main tool

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* $H(s, \zeta(s)) = 0$ *and* $\zeta(0) = z$. *Assume it is unique.*

Output: A tubular neighborhood isolating ζ.



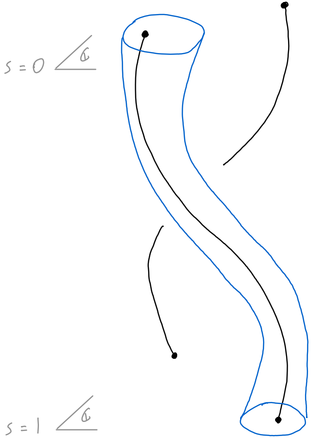
Main tool

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* $H(s, \zeta(s)) = 0$ *and* $\zeta(0) = z$. *Assume it is unique.*

Output: A tubular neighborhood isolating ζ.



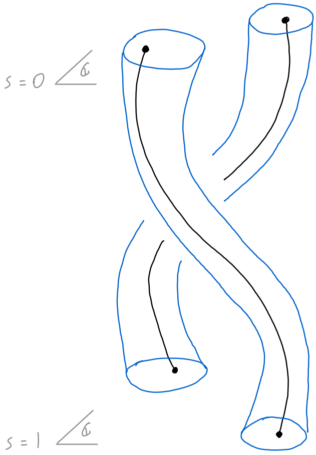
Certified homotopy continuation

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* $H(s, \zeta(s)) = 0$ *and* $\zeta(0) = z$*. Assume it is unique.*

Output: A tubular neighborhood isolating ζ.

We can to that for every solution at $s = 0$.



Main tool

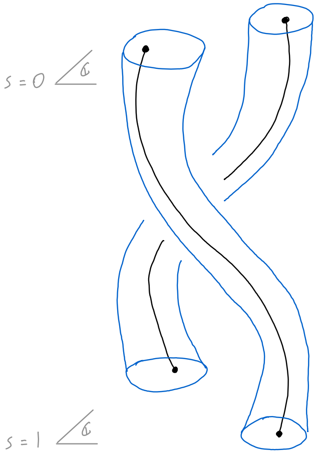
Certified homotopy continuation

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* $H(s, \zeta(s)) = 0$ *and* $\zeta(0) = z$. *Assume it is unique.*
Output: A tubular neighborhood isolating ζ.

We can to that for every solution at $s = 0$.

Application

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



Today's goal

We now assume $\zeta = (\zeta_1, \ldots, \zeta_n) : [0,1] \to OC_n$ inducing a loop in C_n i.e. $\Phi(\zeta(0)) = \Phi(\zeta(1))$.

Goal

Input : ζ (n disjoint tubular neighborhoods around $\zeta_1, \ldots, \zeta_n$).

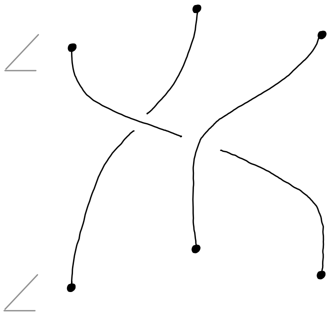
Output : A decomposition in standard generators of the braid induced by $\zeta_1, \ldots, \zeta_n$.

Today's goal

We now assume $\zeta = (\zeta_1, \ldots, \zeta_n) : [0,1] \to OC_n$ inducing a loop in C_n i.e. $\Phi(\zeta(0)) = \Phi(\zeta(1))$.

Goal

Input : ζ (n disjoint tubular neighborhoods around $\zeta_1, \ldots, \zeta_n$).
Output : A decomposition in standard generators of the braid induced by $\zeta_1, \ldots, \zeta_n$.



Overall strategy

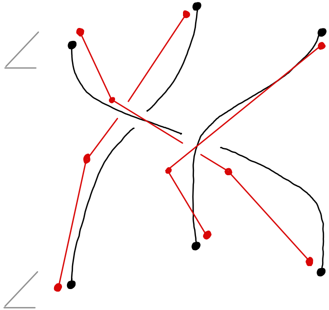
❗ We do not have access to ζ, not even to $\zeta(0)$.

Today's goal

We now assume $\zeta = (\zeta_1, \ldots, \zeta_n) : [0,1] \to OC_n$ inducing a loop in C_n i.e. $\Phi(\zeta(0)) = \Phi(\zeta(1))$.

Goal

Input : ζ (n disjoint tubular neighborhoods around $\zeta_1, \ldots, \zeta_n$).
Output : A decomposition in standard generators of the braid induced by $\zeta_1, \ldots, \zeta_n$.



Overall strategy

! We do not have access to ζ, not even to $\zeta(0)$.

1) Find a path $\tilde{\zeta}$ that has same associated braid.

Today's goal

We now assume $\zeta = (\zeta_1, \ldots, \zeta_n) : [0,1] \to OC_n$ inducing a loop in C_n i.e. $\Phi(\zeta(0)) = \Phi(\zeta(1))$.

Goal

Input : ζ (n disjoint tubular neighborhoods around $\zeta_1, \ldots, \zeta_n$).

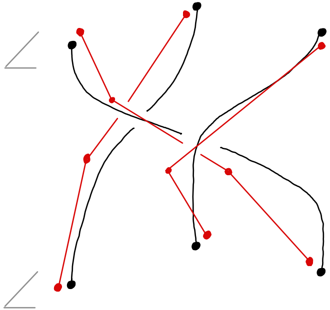
Output : A decomposition in standard generators of the braid induced by $\zeta_1, \ldots, \zeta_n$.

Overall strategy

! We do not have access to ζ, not even to $\zeta(0)$.

1) Find a path $\tilde{\zeta}$ that has same associated braid.

2) Decompose $\tilde{\zeta}$.

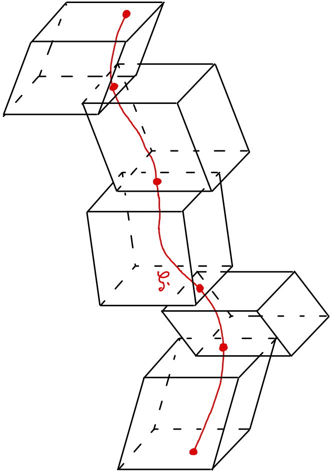


$\sigma_1 \sigma_2^{-1}$

Related work

SIROCCO [Marco-Buzunariz and Rodríguez, 2016]

- Tubular neighborhoods are piecewise **linear**.
- For each strand ζ_i, computes a piecewise linear path in the tube.
- "Intuitive" (! non generic cases) algorithm on the braid with piecewise linear strands.



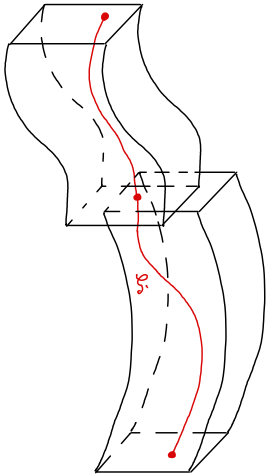
Related work

SIROCCO [Marco-Buzunariz and Rodríguez, 2016]

- Tubular neighborhoods are piecewise **linear**.
- For each strand ζ_i, computes a piecewise linear path in the tube.
- "Intuitive" (! non generic cases) algorithm on the braid with piecewise linear strands.

Algpath [G. and Lairez, 2024]

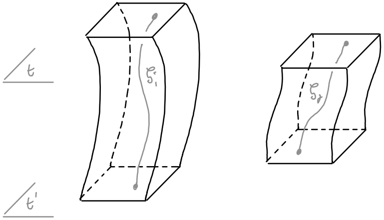
- Tubular neighborhoods are piecewise **cubic**.
- \+ Faster than SIROCCO.
- ! Finding a piecewise linear path in the tube requires additional work.



Strand separation interface

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

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



$\mathrm{sep}(i, j, t) = (t', \rightarrow)$

Cells

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

Definition

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

- $i \to_c j \iff (i, j) \in R$
- $i \to_c j \iff (i, j) \in I$

Cells

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

Definition

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

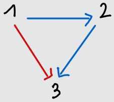
- for all $(i, j) \in R$, $\mathrm{Re}(x_i) < \mathrm{Re}(x_j)$,
- for all $(i, j) \in I$, $\mathrm{Im}(x_i) < \mathrm{Im}(x_j)$,

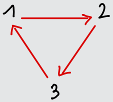
Notation

- $i \rightarrow_c j \iff (i, j) \in R$
- $i \rightarrow_c j \iff (i, j) \in I$

Examples

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





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

Properties of cells

Empty cells

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

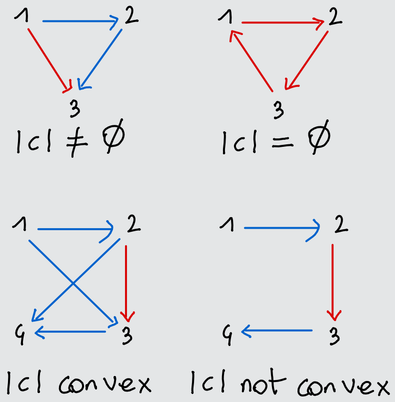
Convex cells

A (non-empty) cell is convex if and only if for all $i, j \in \{1, \ldots, n\}$, either $i {\color{blue}\rightarrow}^* j$ or $j {\color{blue}\rightarrow}^* i$ or $i {\color{red}\rightarrow}^* j$ or $j {\color{red}\rightarrow}^* i$. 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 $|c| \cap |c'|$.

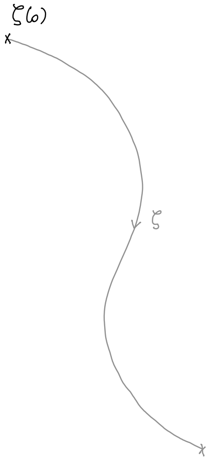
Examples



Step 1: compute a sequence of cells

Path to cells

Input: ζ (represented by tubular neighborhoods).

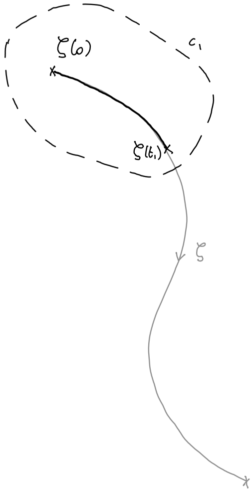


Step 1: compute a sequence of cells

Path to cells

Input: ζ (represented by tubular neighborhoods).
Output: a sequence of convex 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 convex cell c containing $\zeta(0)$.
- Associate to each edge a time of validity.
- When a relation expires, update it using sep and repair convexity.
- Repeat.

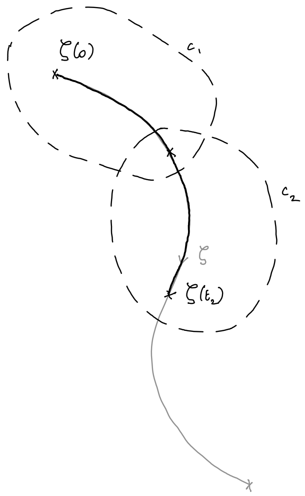


Step 1: compute a sequence of cells

Path to cells

Input: ζ (represented by tubular neighborhoods).
Output: a sequence of convex 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 convex cell c containing $\zeta(0)$.
- Associate to each edge a time of validity.
- When a relation expires, update it using sep and repair convexity.
- Repeat.

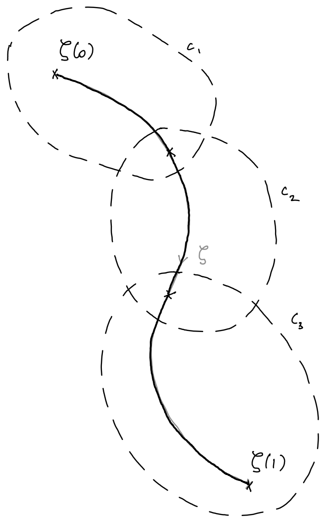


Step 1: compute a sequence of cells

Path to cells

Input: ζ (represented by tubular neighborhoods).
Output: a sequence of convex 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 convex cell c containing $\zeta(0)$.
- Associate to each edge a time of validity.
- When a relation expires, update it using sep and repair convexity.
- Repeat.



Step 2: linearize ζ

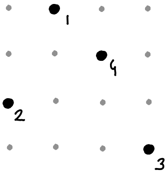
Definition

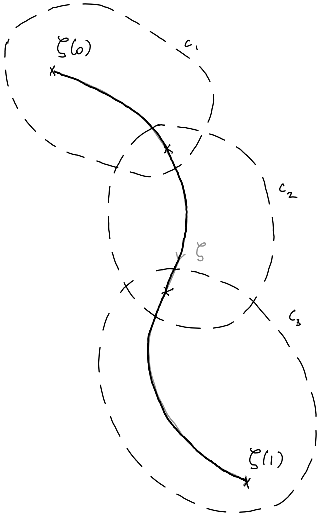
Let $\pi, \varphi \in \mathfrak{S}_n$. We define
$p_{\pi,\varphi} = (\pi(1) + \mathbf{i}\varphi(1), \cdots, \pi(n) + \mathbf{i}\varphi(n)) \in OC_n$.

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

$$\varphi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$

$p_{\pi,\varphi}$:





Step 2: linearize ζ

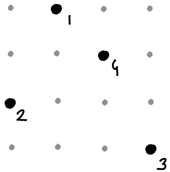
Definition

Let $\pi, \varphi \in \mathfrak{S}_n$. We define
$p_{\pi,\varphi} = (\pi(1) + \mathbf{i}\varphi(1), \cdots, \pi(n) + \mathbf{i}\varphi(n)) \in OC_n$.

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

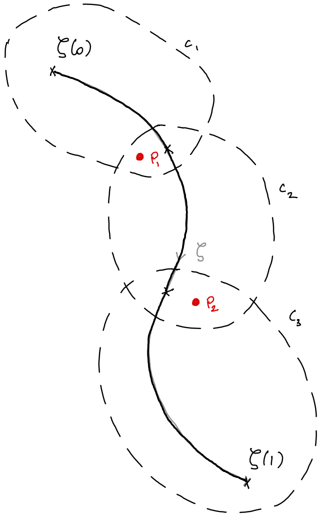
$$\varphi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$

$p_{\pi,\varphi}$:



Linearization of ζ

For each c_i, c_{i+1}, we compute π, φ such that $p_i = p_{\pi,\varphi}$ lies in the intersection $c_i \cap c_{i+1}$ (Hint: total order extending R and I).



Step 2: linearize ζ

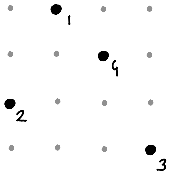
Definition

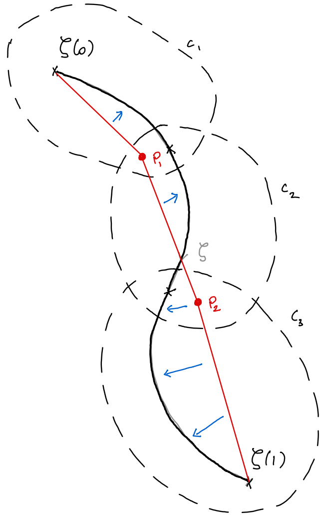
Let $\pi, \varphi \in \mathfrak{S}_n$. We define
$p_{\pi,\varphi} = (\pi(1) + \mathbf{i}\varphi(1), \cdots, \pi(n) + \mathbf{i}\varphi(n)) \in OC_n$.

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

$$e = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$

$p_{\pi,e}$:





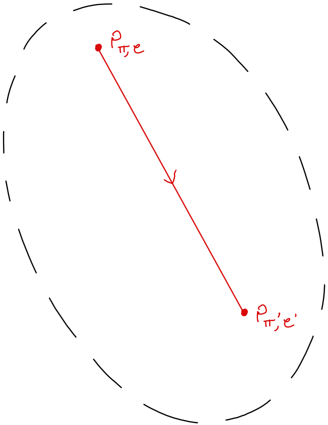
Linearization of ζ

For each c_i, c_{i+1}, we compute π, φ such that $p_i = p_{\pi,\varphi}$ lies in the intersection $c_i \cap c_{i+1}$ (Hint: total order extending R and I). The linear interpolation of the p_i is homotopic to ζ. Why? cells are convex!

Step 3: decomposition of the linearization

Reduction

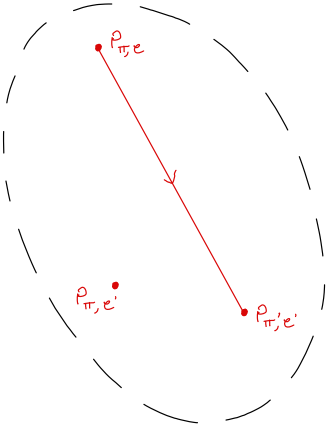
- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent.



Step 3: decomposition of the linearization

Reduction

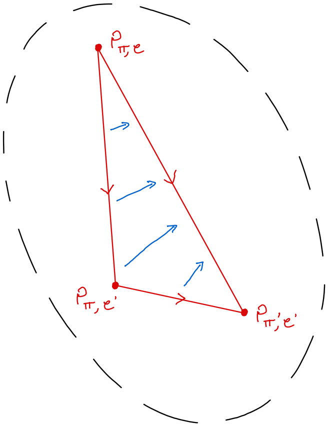
- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent.
- Assume $p_{\pi,\varphi}$ and $p_{\pi',\varphi'}$ both lie in a convex cell $c = (R, I)$. It means that π, π' extend R and φ, φ' extend I. **So $p_{\pi,\varphi'}$ also lies in c!**



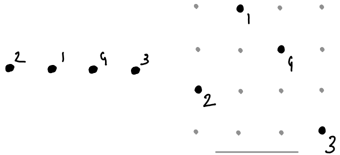
Step 3: decomposition of the linearization

Reduction

- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent.
- Assume $p_{\pi,\varphi}$ and $p_{\pi',\varphi'}$ both lie in a convex cell $c = (R, I)$. It means that π, π' extend R and φ, φ' extend I. **So $p_{\pi,\varphi'}$ also lies in c!**
- We compute the braid of $p_{\pi,\varphi} \to p_{\pi,\varphi'}$ then the braid of $p_{\pi,\varphi'} \to p_{\pi',\varphi'}$.

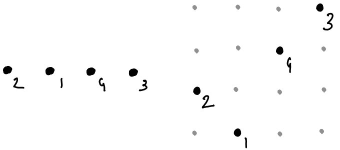


Step 3: decomposition of the linearization



$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

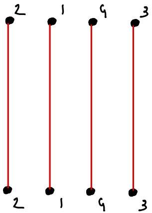
$$\mathcal{C} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$



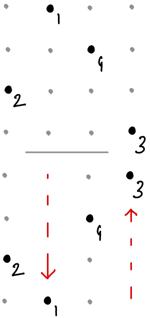
$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

$$\mathcal{C} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix}$$

Step 3: decomposition of the linearization



$1 \in B_4$



$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

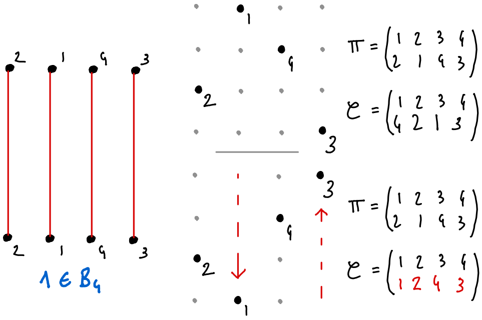
$$\varphi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$

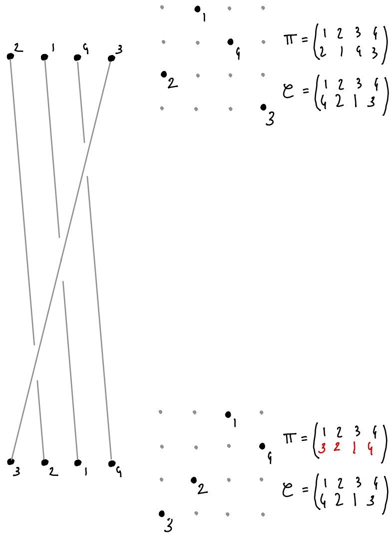
$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

$$\varphi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix}$$

$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.

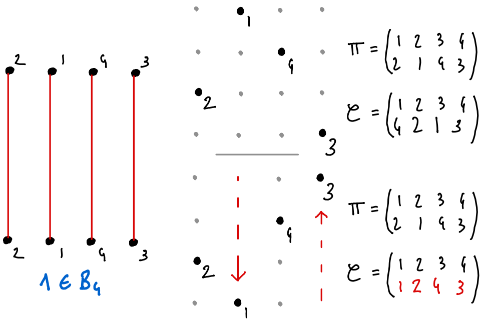
Step 3: decomposition of the linearization

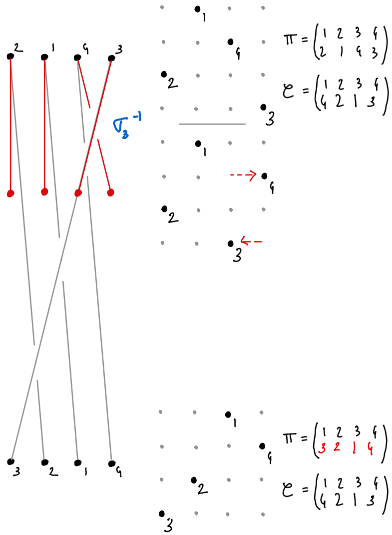




$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.

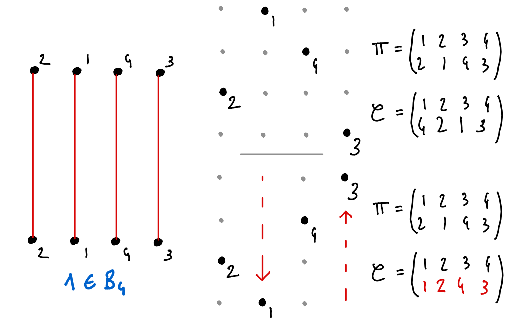
Step 3: decomposition of the linearization



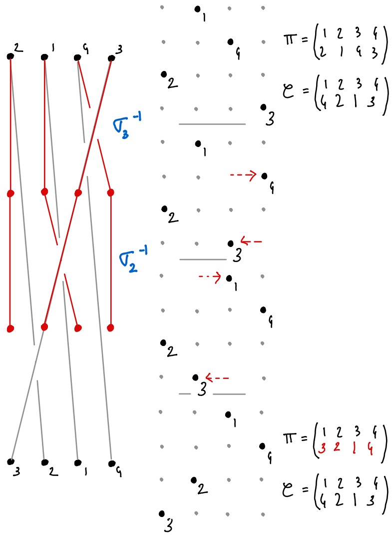


$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.

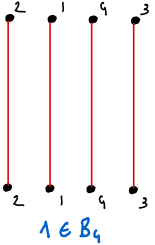
Step 3: decomposition of the linearization

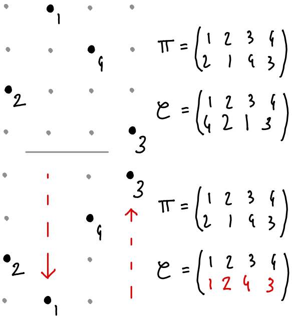


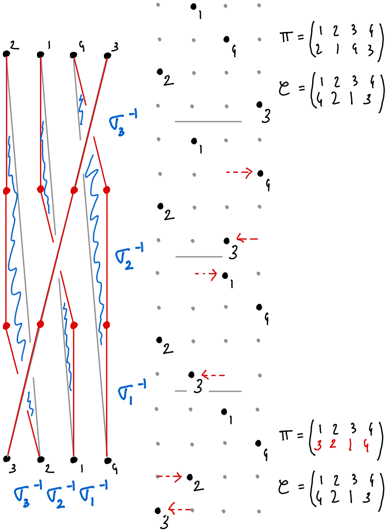
$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.



Step 3: decomposition of the linearization

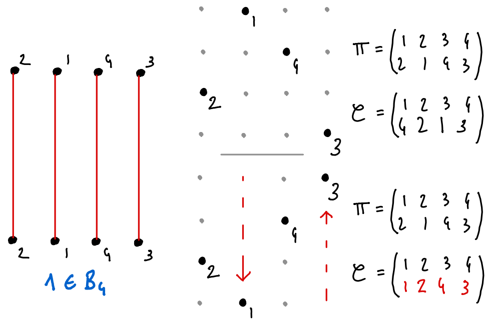


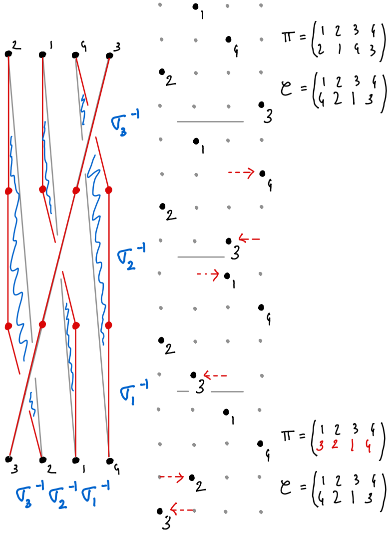




$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.

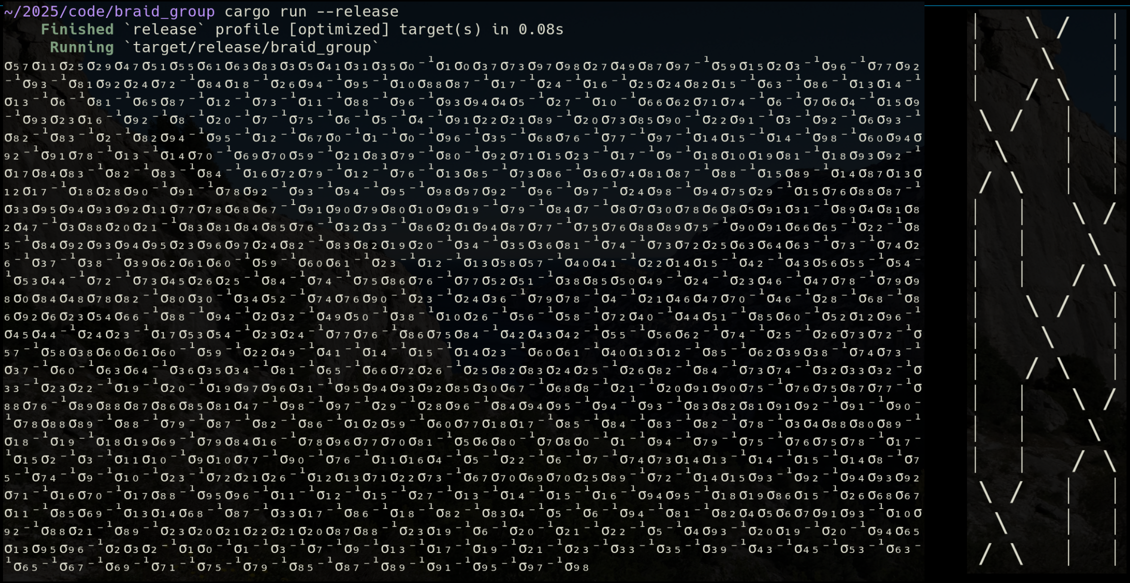
Step 3: decomposition of the linearization





$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.
$p_{\pi,\varphi'} \to p_{\pi',\varphi'}$: Decompose $\pi'\pi^{-1} = s_{i_1}\cdots s_{i_r}$ 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 φ'.

Conclusion


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